

**Design and Cavitation Performance  
of Contra-Rotating Propellers**

by

Dimitrios Laskos

Bachelor of Science in Marine Engineering  
Hellenic Naval Academy, 2002

Submitted to the Department of Mechanical Engineering  
in Partial Fulfillment of the Requirements for the Degrees of

NAVAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

©2010 Dimitrios Laskos. 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 in any medium now known or hereafter created.

Signature of Author \_\_\_\_\_  
Department of Mechanical Engineering  
May 5, 2010

Certified by \_\_\_\_\_  
Chryssostomos Chryssostomidis  
Doherty Professor of Ocean Science and Engineering  
Professor of Mechanical and Ocean Engineering  
Thesis Supervisor

Certified by \_\_\_\_\_  
Richard W. Kimball  
Thesis Supervisor

Accepted by \_\_\_\_\_  
David E. Hardt  
Professor of Mechanical Engineering  
Chairman, Departmental Committee on Graduate Students

This page intentionally left blank

# **Design and Cavitation Performance of Contra-Rotating Propellers**

by

Dimitrios Laskos

Submitted to the Department of Mechanical Engineering on May 5, 2010  
in Partial Fulfillment of the Requirements for the Degrees of  
Naval Engineer  
and  
Master of Science in Mechanical Engineering

## **Abstract**

Improvement of the propulsive efficiency of ships has always been one of the main objectives for naval architects and marine engineers. Contra-Rotating propellers (CRP) are propulsor configurations offering higher efficiency compared to conventional single propellers by recovering the rotational energy in the propeller slipstream. The application of this type of propulsive device to modern ships becomes even more attractive, considering the recent developments in electric propulsion and the increased emphasis on fuel economy. Propeller design codes are therefore expected to include CRP design capabilities.

This thesis describes two methods for designing CRP in the context of lifting-line theory, along with a procedure for predicting the cavitation performance of conventional propellers and CRP. All of the above methods have been implemented numerically and integrated into a computer program developed in MATLAB®.

Comparisons of numerical predictions of efficiency between single and contra-rotating propellers, which confirm the superiority of the latter are presented. Physical insight into the increased efficiency of CRP is also obtained by presenting results for the velocity fields induced by these propulsor configurations. In addition, the predicted cavitation patterns, observed on conventional and contra-rotating propellers operating in uniform and non-uniform wakes, show the advantage of CRP with respect to the occurrence of cavitation.

Thesis Supervisor: Chrysostomos Chrysostomidis

Title: Doherty Professor of Ocean Science and Engineering

Professor of Mechanical and Ocean Engineering

Thesis Supervisor: Richard W. Kimball

Title: Lecturer, Department of Mechanical Engineering

This page intentionally left blank

*Αφιερώνεται στον άνθρωπο που μου έχει συμπαρασταθεί με όλες του τις δυνάμεις κατά τη διάρκεια αυτών των τριών ετών, και όχι μόνο... Η ζωντάνια, ο ενθουσιασμός και η αγάπη που προσφέρει δίνουν ξεχωριστό νόημα και αξία στη ζωή μου...*

*Στη Φανή...*

This page intentionally left blank

## **Acknowledgements**

I would like to express my gratitude to my thesis advisor, and director of the MIT Sea Grant College Program, Professor Chryssostomos Chryssostomidis, for supporting me and giving me the opportunity to work on such an interesting topic.

I am also grateful to Professor Richard Kimball for his guidance during the course of my thesis work. His class has been the basis for my understanding of the propeller design theory.

Special thanks to Dr. Brenden Epps for his valuable feedback on the numerical implementation of the CRP design theory.

For providing the MATLAB® codes for calculating minimum pressure envelopes with the use of XFOIL, I would like to thank Chris Peterson.

For their mentoring and support during the three years of my studies I am grateful to:

Captain Patrick J. Keenan, USN

Captain Mark S. Welsh, USN

Commander Trent R. Gooding, USN

I would also like to thank the Hellenic Navy for sponsoring my graduate studies at MIT.

# **Table of Contents**

<b>Abstract.....</b>	<b>3</b>
<b>Acknowledgements .....</b>	<b>7</b>
<b>Table of Contents .....</b>	<b>8</b>
<b>List of Figures.....</b>	<b>10</b>
<b>List of Tables .....</b>	<b>12</b>
<b>Introduction.....</b>	<b>13</b>
1.1 Historical Background and Motivation .....	13
1.2 Vortex Lattice Lifting Line Theory.....	15
<b>Design Methods for CRP .....</b>	<b>18</b>
2.1 Criteria for Optimum Circulation Distribution .....	19
2.2 Lifting Line Methods for CRP .....	21
2.3 Two CRP Design Methods.....	23
2.3.1 ‘Uncoupled’ Method .....	24
2.3.2 ‘Coupled’ Method .....	28
2.4 Interaction Velocities .....	32
2.4.1 Axial Interaction Velocities .....	33
2.4.2 Tangential Interaction Velocities .....	36
2.5 Blade Design .....	37
<b>Numerical Predictions .....</b>	<b>41</b>
3.1 Single Case Comparison .....	42
3.2 Parametric Design Results .....	48
3.3 Comparison between the Two Methods.....	53
<b>Additional Considerations .....</b>	<b>55</b>
4.1 Effect of Lagrange Multipliers.....	55
4.2 Axial Spacing.....	58



4.3	Finite Hub Loading .....	59
4.4	Hub Drag Cancellation.....	60
	<b>Cavitation on Marine Propellers .....</b>	<b>65</b>
5.1	Introduction .....	65
5.2	Blade Section and Cavitation Performance.....	66
5.3	Methods for Calculating Pressure Coefficients.....	70
5.3.1	VLM .....	70
5.3.2	XFOIL .....	72
5.4	Comparison between VLM-XFOIL.....	75
	<b>Cavitation Analysis Results.....</b>	<b>77</b>
6.1	Steady Cavitation in Uniform Wake .....	77
6.2	Unsteady Cavitation .....	79
6.3	Remarks on Cavitation Performance of CRP.....	86
	<b>Conclusions and Future Work.....</b>	<b>89</b>
7.1	Conclusions .....	89
7.2	Recommendations for future work.....	90
	<b>References .....</b>	<b>93</b>
	<b>Appendices.....</b>	<b>97</b>
	Appendix A: CRP Variational Optimization Equations .....	97
	Appendix B: MATLAB® functions .....	99
	B1. ‘Uncoupled’ Method .....	99
	B2. ‘Coupled’ Method .....	104
	B3. Calculation of Pressure Distributions.....	134
	Appendix C: NACA a=0.8 and NACA 66(TMB mod) section .....	142

## **List of Figures**

Figure 1-1: Lifting Line Lattice of bound and free, trailing vortices.....	16
Figure 1-2: Force elements on lifting line.....	17
Figure 2-1: Comparison of optimum circulation distributions for various thrust loadings .....	20
Figure 2-2: Relative Velocities at Blade Sections .....	22
Figure 2-3: Contra-Rotating Propeller Set.....	24
Figure 2-4: Flow Chart for 'Uncoupled' Method .....	27
Figure 2-5: Flow Chart for 'Coupled' Method .....	31
Figure 2-6: Representation of the interacting components of a CRP set.....	35
Figure 2-7: Axial velocity profiles at different axial locations.....	36
Figure 2-8: Geometry of CRP set designed with the current code .....	39
Figure 3-1: Circulation Distributions as predicted by the 'Coupled' method .....	43
Figure 3-2: Circulation Distributions as predicted by the 'Uncoupled' method.....	44
Figure 3-3: Axial induced velocities far downstream / 'Coupled' method.....	46
Figure 3-4: Axial induced velocities far downstream / 'Uncoupled' method.....	46
Figure 3-5: Tangential induced velocities far downstream / 'Coupled' method .....	47
Figure 3-6: Tangential induced velocities far downstream / 'Uncoupled' method .....	47
Figure 3-7: Efficiency for various propellers as a function of thrust loading for $J_s=0.89$ and $C_d=0.0085$ .....	49
Figure 3-8: Propulsor configurations which have been studied.....	50
Figure 3-9: Ideal Efficiency as a function of thrust loading for $J_s=0.89$ .....	51
Figure 3-10: Efficiency as a function of advance coefficient with $C_d=0.0085$ .....	52
Figure 3-11. Efficiency of CRP for viscous-inviscid case ( $C_t=0.69$ ).....	53
Figure 4-1: Efficiency as a function of the Lagrange multipliers.....	55

Figure 4-2: Circulation Distributions for different values of $\lambda_Q$ .....	56
Figure 4-3: Total tangential induced velocities far downstream .....	57
Figure 4-4: Total axial induced velocities far downstream .....	57
Figure 4-5: Effect of Axial Spacing on Efficiency and Geometric Pitch Ratio.....	58
Figure 4-6: Vortex on a lifting line and its image.....	60
Figure 4-7: Hub Loaded Circulation Distributions as predicted by the ‘Coupled’ method.....	63
Figure 5-1: Minimum pressure envelope for NACA 66 section (TMB modified) with the NACA $a=0.8$ camberline having a maximum camber ratio of 0.01 and a maximum thickness ratio of 0.08.....	67
Figure 5-2: Minimum pressure envelopes using vortex lattice method (VLM code) for NACA 66 sections (TMB modified) with the NACA $a=0.8$ camberline having a maximum camber ratio of 0.01.....	69
Figure 5-3: Pressure distributions obtained with VLM for a NACA 66 (TMB modified) thickness and NACA $\alpha=0.8$ meanline with a maximum thickness ratio of 0.08 and a maximum camber ratio of 0.01, for two different angles of attack.....	72
Figure 5-4: Pressure distributions obtained with XFOIL for a NACA 66 (TMB modified) thickness and NACA $\alpha=0.8$ meanline with a maximum thickness ratio of 0.08 and a maximum camber ratio of 0.01, for two different angles of attack.....	74
Figure 5-5: Comparison of Minimum Pressure Envelopes between VLM and XFOIL for sections having a maximum camber ratio of 0.06 but different thickness ratios .....	76
Figure 6-1: Geometry and steady cavitation patterns of a conventional propeller .....	78
Figure 6-2: Geometry and steady cavitation pattern of a CRP set using VLM .....	79
Figure 6-3: Circumferential axial wake distribution.....	80
Figure 6-4: Variation in the angle of attack due to non-uniform axial wake.....	82
Figure 6-5: Geometry and unsteady cavitation patterns of a conventional propeller using VLM .....	83
Figure 6-6: Geometry and Unsteady Cavitation patterns of a CRP set using VLM.....	85
Figure 6-7: Comparison between SRP and CRP efficiency curves at the design point for a thrust loading of 0.69 .....	87
Figure 6-8: Comparison of tip loading between SRP - CRP for a thrust loading of 0.69 .....	88

## **List of Tables**

Table 3-1: Characteristics of CRP set used in comparing the two design methods .....	43
Table 3-2: Chord Distribution for the 4-bladed single propeller .....	50
Table 4-1: Comparison between efficiencies of hub-loaded and hubless variants .....	64
Table 6-1: SR propeller characteristics used in steady cavitation analysis .....	77
Table 6-2: SR propeller characteristics used in unsteady cavitation analysis .....	82
Table 6-3: Comparison between SRP–CRP design for operation in a non-uniform axial wake .....	86

# **Chapter 1**

## **Introduction**

Increase of the propulsive efficiency of ships has always been the objective of naval architects and propeller designers. Several propulsor configurations have been developed aiming at improving the propulsive efficiency. Contra-rotating propellers are one of the most attractive propulsive devices, which can be expected to improve the open water efficiency remarkably by reducing the rotational kinetic energy losses. Despite the hydrodynamic advantages and the possible improvement of the propulsive efficiency that the CRP concept could offer, application to ships has been limited. A reasonable explanation can be given by considering the mechanical complexity, the increased installation cost and the high maintenance requirements associated with the installation of this ‘unconventional’ propulsor configuration in mechanically driven ships.

The development of the podded propulsion during the past few years has removed the need for complex transmission systems and has brought the concept of CRP back into the daylight. In addition, advances in the field of electric propulsion are expected to further increase the application of CRP systems.

### **1.1 Historical Background and Motivation**

The concept dates back to the Swedish naval architect John Ericsson who applied contra-rotating propellers to a shallow draught boat in order to overcome directional instability problems arising from unbalanced forces produced by single propellers. Their principal application since then has been to torpedoes, where torque cancellation is necessary to prevent spinning and to secure directional stability. Over the past several decades model tests and computer simulations have yielded significant predicted power reductions for application of CRP to surface ships, the main reason being the recovery of the rotational energy losses originating from a forward propeller by a counter rotating aft propeller. However, their application to ships was limited, as already

mentioned, mainly due to the mechanical complexity of the drive train which has to include a gearbox, and the bearing of the inner shaft within the outer one.

One major application of CRP system to full-scale ship was made in Japan in 1988 (Nishiyama 1990), after the forementioned issues related to the mechanical complications had been tackled. The sea trial results showed reduction in power compared to the performance of the conventional propeller originally fitted to this ship of 15 percent. Given the increased emphasis on fuel economy, the number of ships equipped with contra-rotating propellers is expected to grow.

Therefore, propeller design and analysis methods, capable of treating CRP receive increased attention. In spite of the development of elaborate lifting surface methods, as well as the introduction of surface panel methods, lifting line theory still plays an essential role in propeller design and particularly in the preliminary design stage. According to Kerwin (1986), the hydrodynamic design of a propeller can be thought to consist of two steps:

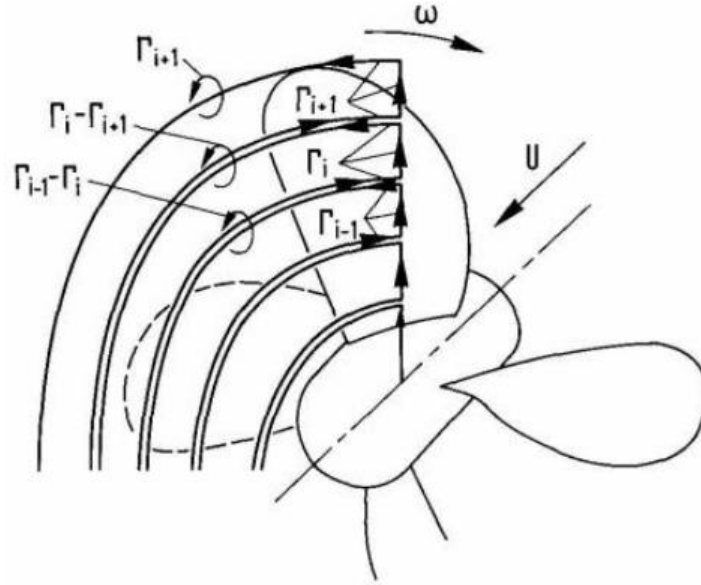
1. Establishment of the radial and chordwise distribution of circulation over the blades which will produce the desired thrust
2. Determination of the blade shape that will produce the prescribed circulation distribution

However, within the limitations of lifting line theory, only the radial distribution of circulation can be obtained since the blades are modeled by straight lines carrying point vortex elements. The radial circulation values are distributed chordwise by assuming that the propeller blade consists of hydrofoil sections with specific camber and thickness distributions. In this work the focus is placed on determining optimum radial circulation distributions for CRP by presenting two numerical methods in the context of lifting line theory. Both methods were coded in MATLAB®, a widely used high-performance language for technical computing. The computer codes borrow from the vortex-lattice lifting line formulation, utilized in the open-source propeller design code ‘OpenProp’ (R.W. Kimball et al. 2008). OpenProp evolved from the MIT Propeller Vortex Lattice Lifting Line Program (PVL) developed by Kerwin (2001), which was later translated into the GUI-based MATLAB® program called MPVL (Chung 2007). The ultimate goal of this work is the extension of the OpenProp’s capabilities so that it can design contra-rotating propeller sets and perform a basic cavitation analysis. Before discussing the CRP

design theory and its detailed numerical implementation, it is necessary to describe the vortex lattice lifting line theory as applied to the case of the single propeller design.

## 1.2 Vortex Lattice Lifting Line Theory

In the framework of vortex lattice lifting line theory, the propeller blades are represented by straight, radial lifting lines with the blades having equal angular spacing and identical loading. The inflow to the propeller disk is assumed to vary radially but is constant in the circumferential sense. Since all blades have the same circulation distribution, one blade is designated as the key blade. The span of the key blade is divided into  $M$  panels extending from the hub root to the blade tip. The radial distribution of bound circulation  $\Gamma(r)$  is approximated by a set of  $M$  vortex elements of constant (but not identical) strength extending from  $r_V(m)$  to  $r_V(m + 1)$ , where  $r_V(m)$  denotes the radius of the  $m_{th}$  vortex point on the lifting line. A discrete trailing (free) vortex line is shed at each of the panel boundaries, with strength equal to the difference in strengths of the adjacent bound vortices. Alternatively, the vortex system can be thought to comprise a set of  $M$  horseshoe vortex elements, each consisting from a bound vortex segment and two free vortex lines (see Fig.1-1) which are of helical shape as we will see later. Taking into account all blades, each horseshoe element actually represents a set of  $Z$  identical elements of equal strength, one emanating from each blade. The velocity field induced at the lifting line by this system of vorticity is computed using the efficient asymptotic formulas developed by Wrench (1957).



**Figure 1-1: Lifting Line Lattice of bound and free, trailing vortices (Breslin & Andersen 1994)**

Vortex elements shed by the propeller blade rotating about a fixed point at angular velocity  $\omega$  in a stream  $U$  are in principle convected by the resultant relative velocity composed of  $U$ ,  $r\omega$  plus the axial, tangential and radial components (self-induced velocity components) induced at the shed element by all members of the vortex array. Thus the trajectory of vortices shed from any radial blade element is not a true helix as the induced velocities vary with distance from the propeller. Only in the ultimate wake (some two-three diameters downstream) a true helical pattern is achieved as the axial inductions achieve their asymptotic values and the radial component vanishes. Moreover, as the vortices act on each other the sheet of vorticity shed from all blade elements as in the flow abaft wings is unsteady and wraps up into two concentrated vortices, a straight one streaming aft of the hub and one inboard of the tip radius.

Once the radial distribution of circulation  $\Gamma(r)$  and the total inflow velocity distribution  $V(r)$  have been determined, the total force acting at any radius  $r$  is given by the Kutta-Joukowski law:

$$dF(r) = \rho V(r) \Gamma(r) \quad (1)$$

and is directed at right angles to the total inflow velocity (see Fig.1-2). The differential axial and tangential forces acting at any radius  $r$  can then be calculated using the hydrodynamic pitch angle  $\beta_i$ :



$$dF_a = \rho V \Gamma \cos \beta_i \quad (2)$$

$$dF_t = \rho V \Gamma \sin \beta_i \quad (3)$$

The radial velocity components do not contribute to either the thrust or the torque elements since they are parallel to the lifting line.

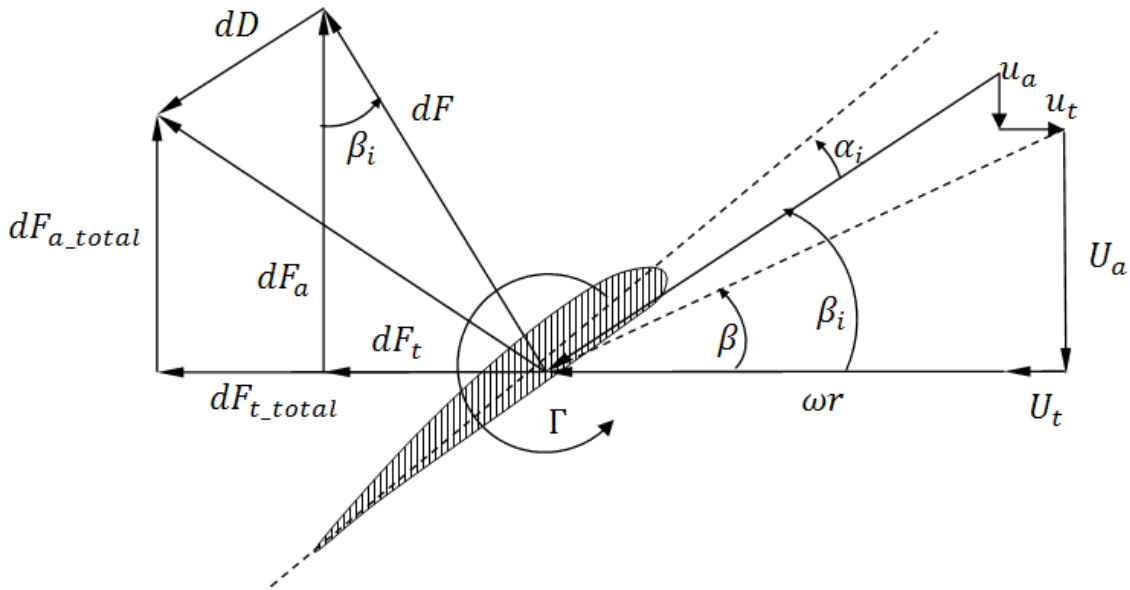


Figure 1-2: Force elements on lifting line

Simplifications which are often applied to the lifting line theory involve the geometry of the propeller wake to be purely helical, with a pitch at each radius determined either by the undisturbed inflow in the lightly loaded case (linear theory) or by the induced flow at the lifting line in the moderately loaded case. In the present application of the CRP design theory the moderately loaded model is implemented.

The lightly loaded propeller is analogous to the wing where the trajectories of the trailing vortices are assumed to be independent of the wing loading. In addition, the trailing vortex rolling-up process is neglected due to the extremely large computational burden and the fact that the precise details of the deformed trailing vortex wake are not critical in determining the flow at the blades.

This page intentionally left blank

# **Chapter 2**

## **Design Methods for CRP**

This chapter deals with the different approaches for designing contra-rotating propellers using lifting line theory. First, the optimum loading criteria for single propellers are presented. A review of the various methods for designing CRP is given next. Section 2.3 describes the numerical implementation of two design methods for CRP. A detailed description of the interaction velocities calculation procedure, which is an important part for the design of interacting components, is given in section 2.4. Finally, the process of determining the shape of the blade once the loading distribution has been calculated, is explained.

### **2.1 Criteria for Optimum Circulation Distribution**

As mentioned, lifting line theory is the basis for propeller design since it provides the radial distribution of loading or circulation. This distribution is obtained by use of criteria for optimum efficiency or modifications of such a distribution, for example to reduce the hub or the tip loading, avoid cavitation, high vibratory forces and noise, etc.

Betz(1919) first derived the optimum circulation distribution criterion for propellers operating in uniform wake by using Munk's 'displacement law' that states that the total force on a lifting surface is unchanged if an element of bound circulation is displaced in a streamwise direction. His result suggested that the ultimate forms of the vortices far downstream for an optimum circulation distribution are true helices and is expressed as

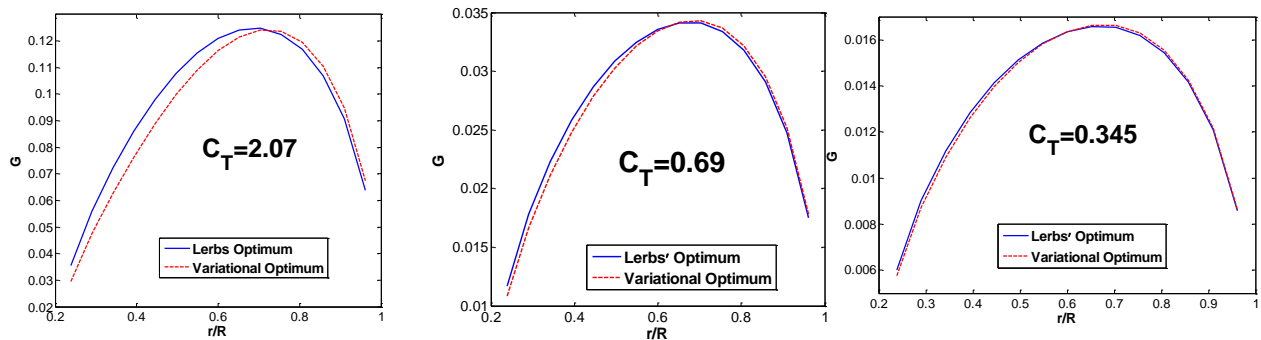
$$\tan\beta_i = \tan\beta/\sqrt{l^*} \quad (4)$$

where  $l^*$  is a dimensionless constant depending on the required thrust produced by the propeller. The condition for non-uniform or wake-adapted inflow was given by Lerbs (1952) by extending Betz's work after including the thrust deduction and the wake fractions in his computations. Lerbs' method remains one of the universally accepted procedures for establishing the radial

distribution of circulation and calculating the propeller efficiency in the context of lifting line theory. Several other criteria were developed afterwards but all gave a distribution of the hydrodynamic pitch angle  $\tan\beta_i$  of the form

$$\tan\beta_i = k \cdot F(r) \tag{5}$$

where  $k$  is an unknown factor related to the required thrust and  $F$  a function depending on the optimum criterion. A different procedure for determining optimum circulation distributions has been developed by Kerwin, Coney and Hsin (1986). Instead of deriving optimum criteria corresponding to Betz or Lerbs a numerical version of their derivation using calculus of variations and Lagrange multipliers, but working with the unknown circulations, is used. Interestingly, both methods yield similar results and in the limit of light loading the variational optimum approaches Lerbs' optimum. Optimum circulation distributions plots for a single four bladed propeller operating in uniform wake for three different thrust loadings are given in Fig.2-1 below. It is evident that the variational optimum approaches the Lerbs' optimum as the thrust loading becomes lighter.



**Figure 2-1: Comparison of optimum circulation distributions for various thrust loadings**

Another method based on a genetic algorithm has been presented by Caponnetto et al.(1997). This method is based on a trial and error procedure through which several different propellers, all satisfying the design requirements are sequentially analyzed and their efficiency is calculated. Each time a more efficient propeller is found and it is used as a base for a new candidate optimum propeller, modified by a small random perturbation. Thus, neither a closed analytical form for describing efficiency nor an optimality criterion is required, even though several thousand iterations would be necessary in order to obtain a 'smooth' propeller geometry.

## 2.2 Lifting Line Methods for CRP

A contra-rotating propeller is defined to consist of two, coaxial, open propellers positioned a short distance apart and rotating in opposite directions. The first lifting line method for predicting optimum circulation distributions for CR propellers with equal number of blades was outlined by Lerbs (1955). His method was an extension of his lifting line method for single rotation (SR) propellers with the inclusion of the mutual interaction velocities as well as the self induced velocities. Lerbs in his theory first assumed that the axial distance between the fore and aft propellers was zero which he later corrected for the actual spacing. Thus he was able to design a so-called 'equivalent' propeller which produced one half the total thrust. He also determined the interaction velocities between the two propellers by using weighting factors applied to the self-induced velocities. Morgan (1960) derived once again Lerbs' theory for CR propellers with any combination of number of blades for both the wake-adapted and the free-running cases. In a complimentary paper, Morgan and Wrench (1965) rederived the integro-differential equation for the equivalent circulation distribution of a CRP set, and most importantly, derived efficient and accurate formulas for the evaluation of the velocity induction factors. CRP theory has been developed since that time as a logical extension of the foregoing concepts underlying the classical vortex theory for SR propellers, but several additional approximations have to be introduced.

In a CRP set not only must the self induced velocities arising from the induction of the trailing vortices on the lifting lines be calculated but also the interaction velocities on one propeller due to the presence of the other must be taken into account. The mutual interactions between forward and aft propellers give rise to time dependent flow and forces. In particular, the aft propeller blades rotate through the vortex sheets in the slipstream of the forward propeller. The forward propeller is also subjected to the circumferentially varying flow disturbance generated by the aft propeller. The theory for time-average (or steady) forces rests on a fundamental approximation. The onset flow to each propeller is divided into a circumferential average component (which may vary radially and axially) and components which are periodic (harmonics). It is assumed that the average component of velocity produces the steady forces on the propeller while the periodic components produce alternating forces with zero mean.

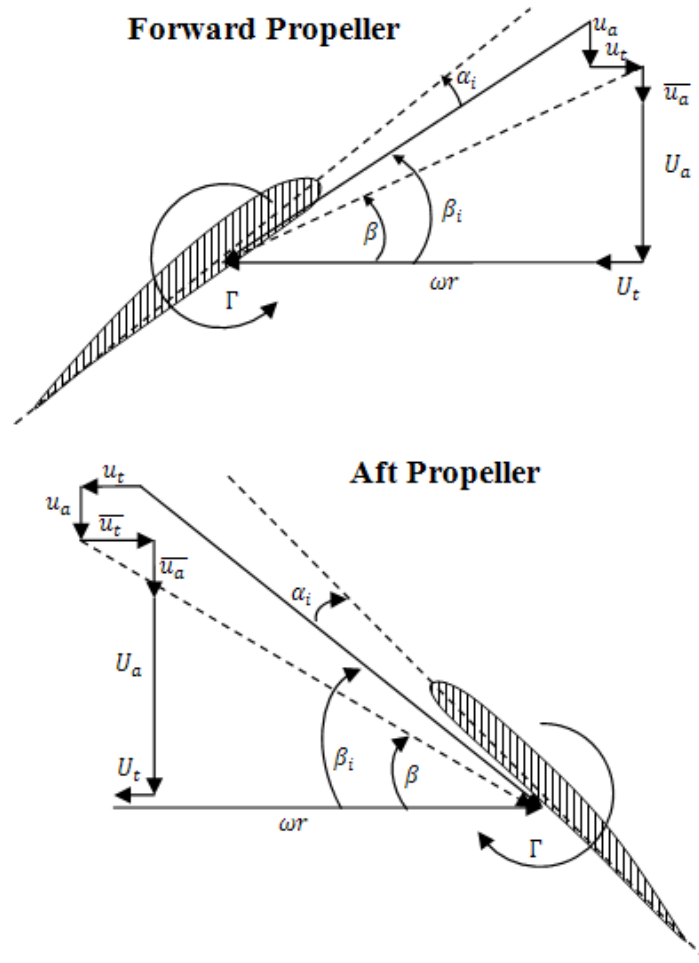


Figure 2-2: Relative Velocities at Blade Sections

Thus, the forward and aft propellers are usually regarded as SR propellers operating in steady, axisymmetric flows in which the onset flow to each propeller is modified to include the average axial, radial, and tangential components of the velocity field induced by the other propeller. This necessarily involves an iterative procedure in which the loadings and induced velocities of each propeller are successively determined until a converged solution is reached. Velocity diagrams at the forward and the aft propeller blade sections are shown in Fig.2-2. Both the self-induced and the interaction velocity components,  $u_a$ ,  $u_t$  and  $\bar{u}_a$ ,  $\bar{u}_t$  respectively are included.

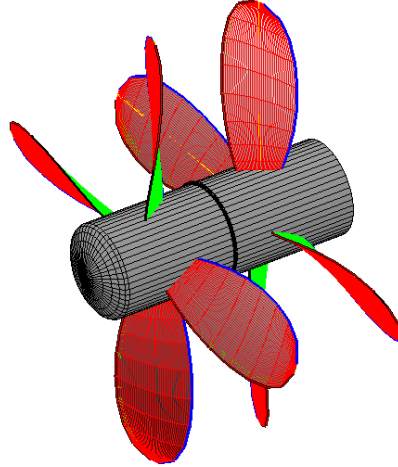
More recently, Kerwin, et al.(1986) presented a more rigorous lifting line method for the design of multi-component propulsors which could model CRP as an integrated propulsive unit and not by coupling single propeller codes in an iterative way. The coupled integro-differential equations for circulation distributions are solved simultaneously by using a variational optimization

scheme. An extensive list of the different design methods for CRP is given by Cox and Reed (1988). One of the latest contributions to the optimization and design of contra-rotating propellers has been made by Caponetto (2000). His technique is also based on an iterative procedure where one propeller is designed at a time, but the genetic algorithm, described in the section 2.1, is used.

It has become obvious by now that two different methods for determining optimum circulation distributions for contra-rotating propellers can be identified in general. The first one, referred to from now on as the ‘Uncoupled’ method, treats the components of the contra-rotating propeller set as if they were SR propellers. Optimum circulation distribution criteria for SR propellers can therefore be utilized in order to obtain the solution for the CRP set. The second one, referred to as the ‘Coupled’ method, treats the two propellers as a single unit and is the extension of the variational optimization approach developed by Kerwin, et al. to the case of two-component propulsors. The specifics of the numerical implementation of these two methods are given next.

### **2.3 Two CRP Design Methods**

Before presenting the detailed description of the ‘Coupled’ and the ‘Uncoupled’ methodologies, the underlying general assumptions should be listed first. In the current formulation of the optimum circulation distribution methods for CRP the major assumption is that the streamlines do not contract. Consequently, there’s no need for the aft propeller diameter to be smaller than the forward one since the tip vortices from the latter do not impinge on the former. Furthermore, tracking of the contracting streamlines is avoided, thus making the computation algorithms lighter. Hence, both propellers are assumed to have the same diameter. The hub diameter is also the same for both components. A representation of such a configuration can be seen in Fig.2-3.



**Figure 2-3: Contra-Rotating Propeller Set**

### 2.3.1 ‘Uncoupled’ Method

As already mentioned this method decouples the circulation distributions for the forward and aft propellers by eliminating the requirement of designing an equivalent propeller as described in Morgan’s and Lerbs’ model for the CRP interactions. Either Lerbs’ or Kerwin’s (variational optimization) methods for single propulsors can be used for setting up the system of equations for the bound circulation values on the two propellers. In the present work the variational optimization is implemented so that there is consistency in the way the circulation values are determined by both methods (‘Uncoupled’ - ‘Coupled’) and the results of the comparison capture the differences exclusively due to the way the CRP set is modeled (a combination of two SR propellers or an integrated propulsor with two components), even though the implementation of Lerbs’ optimization method is expected to yield similar results (see section 2.1).

According to the variational optimization scheme for a single propeller as described by Coney (1989) an auxiliary function

$$H = Q + \lambda(T - T_r) \quad (6)$$

is formed. The goal is to find a set of discrete circulation values  $\Gamma(1) \dots \Gamma(M)$  applied at the  $M$  control points on the lifting line, such that the torque,



$$Q = \rho Z \sum_{m=1}^M [V_a(m) + u_a^*(m)] r(m) \Gamma(m) \Delta_r \quad (7)$$

is minimized subject to the constraint that the thrust,

$$T = \rho Z \sum_{m=1}^M [V_t(m) + \omega r + u_t^*(m)] r(m) \Gamma(m) \Delta_r \quad (8)$$

has a prescribed value  $T_r$ . The parameters  $V_a$ ,  $V_t$ ,  $u_a^*$  and  $u_t^*$  correspond to the total axial inflow, total tangential inflow, induced axial and induced tangential velocities respectively. The partial derivatives of  $H$  with respect to the unknown  $M$  circulation values and the Lagrange multiplier  $\lambda$  are set to zero. The resulting system of  $M + 1$  equations is linearized by assuming that the Lagrange multiplier is known where it forms quadratic terms with the circulations and solved for the  $M + 1$  unknowns. Initially setting the Lagrange multiplier equal to -1 proved to be a suitable estimate. The solution yields the optimum circulation distribution and the value for  $\lambda$ . The variational optimization for single component propulsors has been implemented numerically by Epps (2009).

The coupling between the ‘two SR propellers’ is provided entirely by the interaction velocities between them and the resulting equations are subjected to two constraints, the total required thrust produced by the set and the torque ratio between the elements of the set,

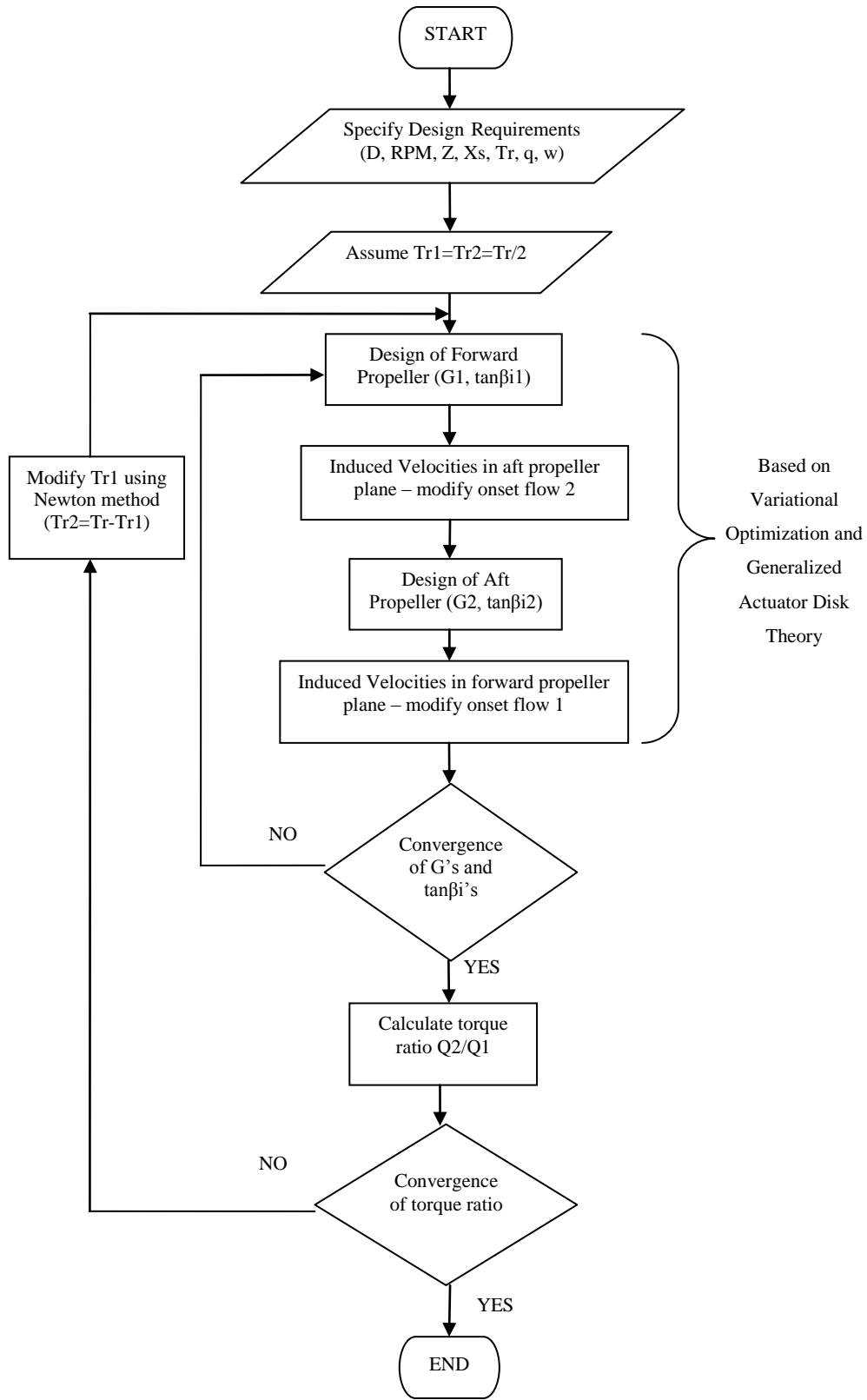
$$T_{r1} + T_{r2} = T_r \quad (9)$$

and

$$q = Q_2/Q_1 \quad (10)$$

These systems of equations are solved using an iterative scheme where one system is solved at a time as if it was solved for a conventional SR propeller. The interaction velocities induced on the forward propeller by the aft are initially assumed to be zero and the linear system of equations for the unknown circulations and the Lagrange multiplier for the forward propeller are solved. Once the solution for the forward propeller is obtained, the interaction velocities induced by the

forward propeller on the aft one are computed. These interaction velocities are then added to the onset flow for the aft propeller and the linear system of equations for that propeller is then solved as if it were a SR propeller. The interaction velocities induced on the forward by the aft component are then computed and the new circulation values on the forward component for the updated onset flow are then determined. This iterative procedure is repeated until convergence for the forward and aft circulation distributions is achieved. Once convergence for the circulation distributions  $\Gamma_1(r), \Gamma_2(r)$  is reached a check is performed on whether or not the obtained solution satisfies the torque requirement. Matching of the specified total thrust and torque ratio is accomplished by first using an initial guess with the total thrust being equally divided into the two components,  $T_{r1} = T_{r2} = T_r/2$ , and then applying Newton's method to find the thrust ratio which produces the required torque ratio  $q$ . This procedure is implemented numerically in the *UncoupledCRP.m* MATLAB function. The corresponding flow chart is illustrated in Fig.2-4.



**Figure 2-4: Flow Chart for 'Uncoupled' Method**

The fact that the induced velocities due to the aft propeller acting on the forward one are generally small, especially as the axial separation  $X_s$  increases, insures that this iterative scheme converges very quickly. In order to compute the circumferential mean axial interaction velocity components the analytic expressions from generalized actuator disk theory developed by Hough and Ordway (1965) are used. As far as the computation of the tangential interaction velocity components is concerned, a direct application of Kelvin's theorem for the conservation of circulation is utilized. A detailed description of the procedure for calculation the interaction velocities is given in section 2.4. The 'Uncoupled' method presented above is similar to the lifting line design method developed by Caster and LaFone (1975). Their method however uses Lerbs' criterion for determining optimum circulation distributions instead of the variational optimization. In addition, Caster and LaFone used Kerwin's field point velocity program for calculating the average axial and tangential interaction velocities (J.E. Kerwin & Leopold 1964).

### 2.3.2 'Coupled' Method

This method was developed by Kerwin, Coney and Hsin (1986) and is an extension of the variational optimization approach for single propeller design. The optimization procedure enables one to determine both the division of loading between CRP components and the radial distribution of loading (circulation) on each component simultaneously since the two propellers comprising the set are regarded as a unit. This method can also be applied to other multi component propulsors, such as propellers with pre or post swirl stators or vane wheels, provided a computational scheme for calculating the interaction velocities exists. The same technique for calculating the interaction velocity components, as in the 'Uncoupled' method, is being used. Following Coney, in the case of two propulsor components, the goal is to find the discrete circulation values  $\Gamma_1(1) \dots \Gamma_1(M_1), \Gamma_2(1) \dots \Gamma_2(M_2)$  such that the total power  $P = \omega_1 \cdot Q_1 + \omega_2 \cdot Q_2$ , absorbed by the propulsor is minimized. The propulsor is additionally required to develop a prescribed thrust  $T_r$ . In addition, two component propulsors are often constrained to have a specific division of torque between the components. Therefore, a torque ratio,  $q = Q_2/Q_1$ , is also specified. These three conditions are used to form an auxiliary function H,

$$H = (\omega_1 Q_1 + \omega_2 Q_2) + \lambda_T (T_1 + T_2 - T_r) + \lambda_Q (q Q_1 - Q_2) \quad (11)$$

After expressing the thrust and torque of the individual components of the CRP set in terms of the circulation values, the partial derivatives of H with respect to the unknown circulation values and the Lagrange multipliers are set equal to zero:

$$\begin{aligned}
\frac{\partial H}{\partial \Gamma_j(i)} = 0 &= \omega_1 \frac{\partial Q_1}{\partial \Gamma_j(i)} + \omega_2 \frac{\partial Q_{21}}{\partial \Gamma_j(i)} + \lambda_T \left[ \frac{\partial T_1}{\partial \Gamma_j(i)} + \frac{\partial T_2}{\partial \Gamma_j(i)} \right] \\
&+ \lambda_Q \left[ q \frac{\partial Q_1}{\partial \Gamma_j(i)} - \frac{\partial Q_2}{\partial \Gamma_j(i)} \right] = \\
&= (\omega_1 + q\lambda_Q) \frac{\partial Q_1}{\partial \Gamma_j(i)} + (\omega_2 - \lambda_Q) \frac{\partial Q_2}{\partial \Gamma_j(i)} + \lambda_T \left[ \frac{\partial T_1}{\partial \Gamma_j(i)} + \frac{\partial T_2}{\partial \Gamma_j(i)} \right],
\end{aligned} \tag{12}$$

*for j = 1,2 and i = 1, ..., M<sub>j</sub>*

and,

$$\frac{\partial H}{\partial \lambda_T} = 0 = (T_{I_1} + T_{V_1}) + (T_{I_2} + T_{V_2}) - T_r \tag{13}$$

$$\frac{\partial H}{\partial \lambda_Q} = 0 = q(Q_{I_1} + Q_{V_1}) - (Q_{I_2} + Q_{V_2}) \tag{14}$$

The expressions for the inviscid and viscous thrust and torque terms acting on a CRP component can be written as:

$$T_{I_j} = \rho Z_j \sum_{n=1}^{M_j} \left[ V_{t_j}(n) + \omega_j r_j(n) + u_{t_j}^*(n) \right] \Gamma_j(n) \Delta r_j \tag{15}$$

$$Q_{I_j} = \rho Z_j \sum_{n=1}^{M_j} \left[ V_{a_j}(n) + u_{a_j}^*(n) \right] r_j(n) \Gamma_j(n) \Delta r_j \tag{16}$$

$$T_{V_j} = -\frac{1}{2} \rho Z_j \sum_{n=1}^{M_j} V_j^*(n) \left[ V_{a_j}(n) + u_{a_j}^*(n) \right] c_j(n) C_{D_{V_j}}(n) \Delta r_j \tag{17}$$

$$Q_{V_j} = \frac{1}{2} \rho Z_j \sum_{n=1}^{M_j} V_j^*(n) \left[ V_{t_j}(n) + \omega_j r_j(n) + u_{t_j}^*(n) \right] r_j(n) c_j(n) C_{D_{vj}}(n) \Delta r_j \quad (18)$$

for  $j = 1, 2$

where  $C_{D_v}$  and  $c$  are the 2D viscous drag coefficients and the chordlengths of the blade sections respectively. The velocity induced on a given control point is a summation of the velocities induced by the individual horseshoe vortices and is given by:

$$u_{a_j}^*(n) = \sum_{k=1}^K \sum_{m=1}^M \Gamma_k(m) u_{a_{j,k}}^*(n, m) \quad (19)$$

$$u_{t_j}^*(n) = \sum_{k=1}^K \sum_{m=1}^M \Gamma_k(m) u_{t_{j,k}}^*(n, m), \quad (20)$$

where  $u_{a_{j,k}}^*(n, m), u_{t_{j,k}}^*(n, m)$  are the axial and tangential velocities respectively induced at control point  $n$  of component  $j$  by the horseshoe vortex of unit strength surrounding control point  $m$  of component  $k$ . Whenever  $j = k$ ,  $u_{a_{j,k}}^*$  and  $u_{t_{j,k}}^*$  are the self-induced velocities, otherwise they are the interaction velocities.

This is a non-linear system of  $M_1 + M_2 + 2$  equations for the  $M_1$  unknown circulation values on component 1, the  $M_2$  unknown values on component 2 and the two Lagrange multipliers. Kerwin, et al. solved the resulting system of non-linear equations by freezing the Lagrange multipliers where they formed quadratic terms with the circulation values. The only terms in which the Lagrange multipliers were allowed to contribute to the system of equations were those where they forms products with the onset flow velocities  $V_a$  and  $V_t$ , and the angular velocities  $\omega_1$  and  $\omega_2$ . The final, linear system can be found in Appendix A.

At this point we should note that Kerwin, et al's linearization represents an arbitrary linearization of the non-linear equations. An exact linearization of the equations for the circulation and Lagrange multipliers should involve using the vector equivalent of a Taylor series expansion of the non-linear equations in terms of the unknowns. The exact linearization of the inner problem

has been performed by Cox and Reed (1988) and is the most complete lifting line theory which can be developed within the assumptions of the theory.

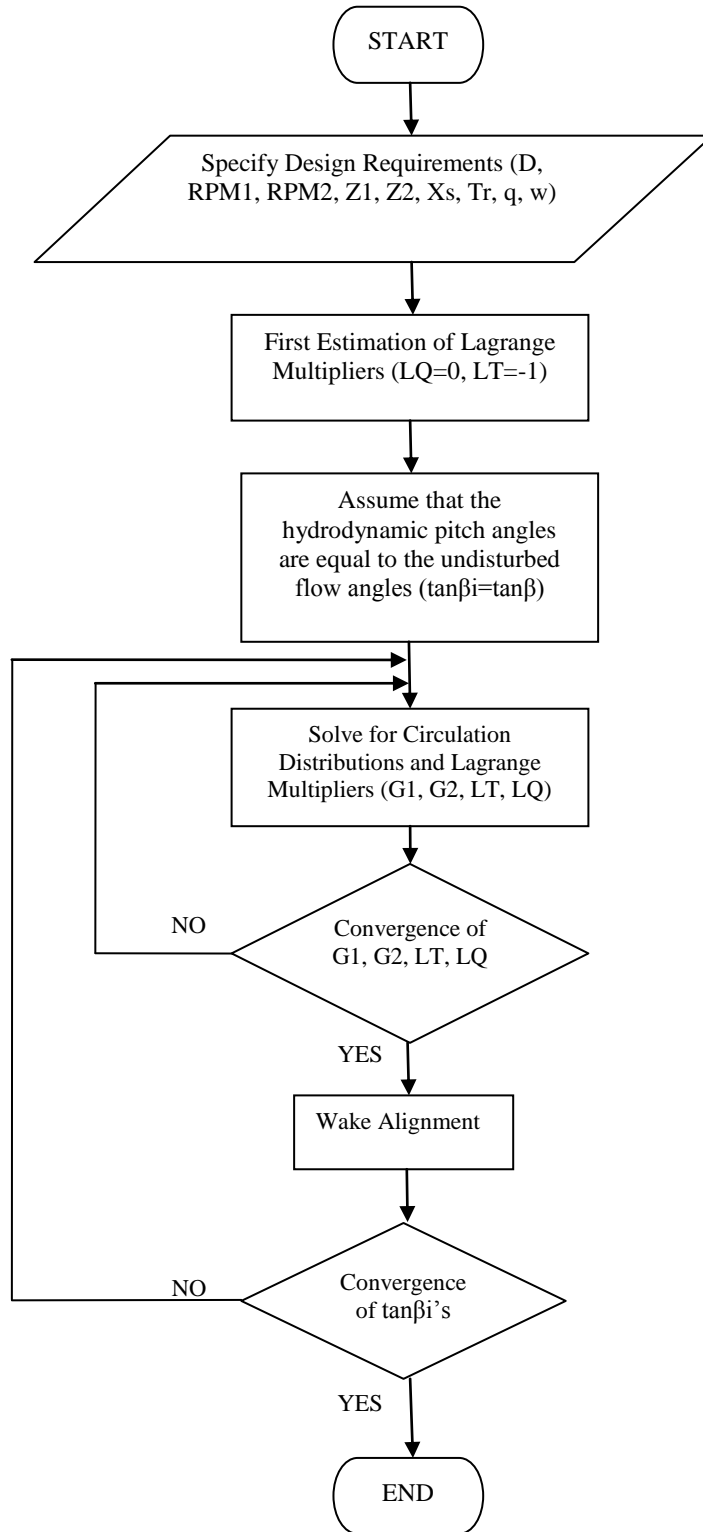


Figure 2-5: Flow Chart for 'Coupled' Method

In the current formulation, like in the variational optimization for SR propellers, Kerwin, et al. had two nested problems to solve. The inner problem was the determination of the optimum circulation distributions for the forward and aft propellers; it was solved for fixed value of the hydrodynamic pitch angles,  $\tan\beta_i$ , on the forward and aft propellers. The outer problem was that of determining the appropriate distributions of  $\tan\beta_i$  for both propellers; it was solved by selecting an initial distribution of  $\tan\beta_i$ , usually,  $\tan\beta$ , and solving the inner problem for the optimum circulation distributions. The induced velocities due to these circulation distributions were then used to determine new  $\tan\beta_i$  distributions, which were used to solve again the inner problem for the optimum circulation distributions. This process was repeated as many times as necessary until the  $\tan\beta_i$  and the circulation distributions had converged. Fig.2-5 shows the flow chart for the ‘Coupled’ method.

Interestingly, by setting the torque ratio  $q = Q_2/Q_1$ , the number of blades  $Z_2$  and the rotational speed of the aft propeller  $\omega_2$  equal to zero, one recovers the optimum circulation distribution for the single propeller, thus minimization of the objective functions  $H = \omega_1 Q_1 + \lambda_T (T_1 - T_r)$  and  $H = Q + \lambda(T - T_r)$  yields the same results, as expected.

Initial estimates of zero for  $\lambda_Q$  and -1 for  $\lambda_T$  were suitable according to Coney (1989). However, if the values of the Lagrange multipliers are not updated each time the linear system of equations is solved, and values other than zero and -1 are used, the results are found to differ significantly. In particular, the obtained circulation distributions appear to be very sensitive to the values of the Lagrange multipliers, whereas the efficiency is not. Plots of propeller efficiency and circulation distributions as a function of  $\lambda_Q$  and  $\lambda_T$  will be presented in the next section.

With respect to the element arrangement of the vortex lattice model, both constant and cosine spacing were implemented successfully. However, for cosine spacing with more than 15 panels the algorithm would not converge. Therefore, constant spacing was employed in both methods.

## 2.4 Interaction Velocities

The major difference between SR and CR propellers in terms of modeling is the fact that for CRPs the velocity field is much more complex, since both the self-induced and the interaction velocities between the components of the set must be included in the optimization equations.



Therefore, a detailed knowledge of the velocity field induced by a propeller at axial locations other than the propeller plane itself is necessary.

An analytical solution for the steady (circumferentially average) velocity fields induced by a lightly loaded propeller of arbitrary blade number and circulation distribution has been developed by Hough and Ordway (1965). In their formulation, the propeller is represented by a conventional vortex system consisting of bound radial vortex lines and a helical sheet of vortices trailing from each line. The induced velocities at any field point are then determined by the Biot-Savart law. In order to achieve that, they first calculated the velocity fields induced by an actuator disk with uniform radial distribution of load and then constructed by superposition the solution for general load distributions by integrating the constant-loading solutions.

The selection of this method for calculating the induced velocity fields is further justified by the work of Hsin (1987), who compared three different methods for computing the circumferential average induced velocity for multi-component propulsors and found Hough's and Ordway's method to be the most computationally efficient. Conway (1995) also derived the solutions given by Hough and Ordway by using a different method based on construction of the velocity and potential fields induced by a vortex ring.

In the case of CRPs, only the axial and the tangential interaction velocities are of interest since the radial component does not contribute to the development of forces on the propeller blades. Besides, the contraction of the wake is ignored which effectively implies that there is no motion of the fluid particles in the radial direction. A detailed description of the interaction velocities calculation procedure follows.

#### **2.4.1 Axial Interaction Velocities**

In the calculation of the axial velocity, only the free trailing helical vortices are taken into account since there is no contribution from the bound vorticity. Kerwin, et al. applied the formulas developed by Hough and Ordway to the case of moderately loaded propellers. Following Coney's notation, the local axial velocity induced at the  $m_{th}$  control point of component  $j$  by the  $p_{th}$  trailing vortex with unit strength of component  $l$  is given by:

$$\bar{u}_{a_j,l}(m, p) = \frac{Z_l}{\pi r_l(p) \tan(\beta_l(p))} C_1 \quad (21)$$

with:

$$C_1 = \begin{cases} \pi + \frac{X_f}{2\sqrt{r_{c_j}(m)r_{v_l}(p)}} Q_{-\frac{1}{2}}(q) + \frac{\pi}{2} \Lambda_0(s, t), & r_{c_j}(m) \leq r_{v_l}(p) \\ \frac{X_f}{2\sqrt{r_{c_j}(m)r_{v_l}(p)}} Q_{-\frac{1}{2}}(q) - \frac{\pi}{2} \Lambda_0(s, t), & r_{c_j}(m) > r_{v_l}(p) \end{cases} \quad (22)$$

where the parameters included in the definition of  $C_1$  are given by:

$$q = 1 + \frac{X_f^2 + (r_{c_j}(m) - r_{v_l}(p))^2}{2r_{c_j}(m)r_{v_l}(p)} \quad (23)$$

$$s = \sin^{-1} \left[ \frac{X_f}{\sqrt{X_f^2 + (r_{c_j}(m) - r_{v_l}(p))^2}} \right] \quad (24)$$

$$t = \sqrt{\frac{4r_{c_j}(m)r_{v_l}(p)}{X_f^2 + (r_{c_j}(m) - r_{v_l}(p))^2}} \quad (25)$$

$Q_{-\frac{1}{2}}$  is the Legendre function of the second kind and half integer order and  $\Lambda_0$  is the Heumann's Lambda function. A representation of the interacting components of a CRP set can be seen in Fig.2-6.

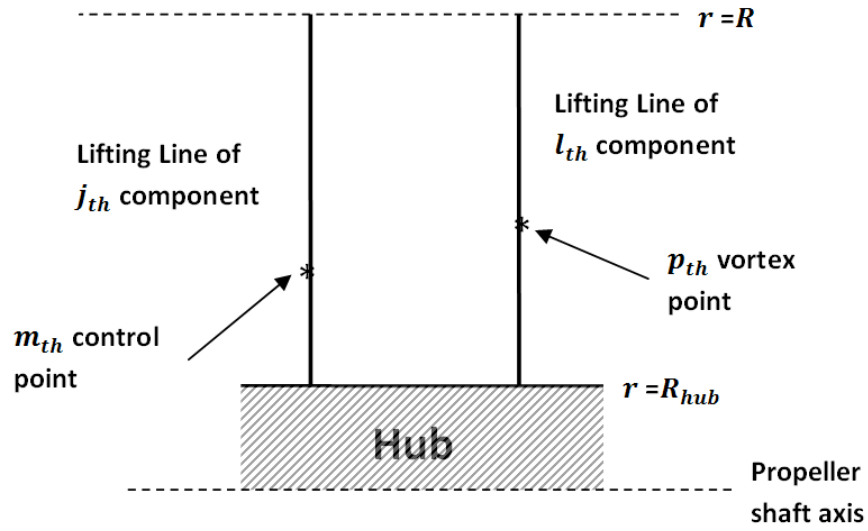


Figure 2-6: Representation of the interacting components of a CRP set

In the formulation of the optimization for the circulation distribution, the bound circulation values are set to be the unknowns. Therefore, the axial interaction velocities should be expressed in terms of the strength of the bound vortex segments (at the  $M$  control points) and not in terms of the strength of the trailing vortices (at the  $M+1$  vortex points). Thus, an axial interaction velocity ‘influence function’ on the  $m_{th}$  control point of component  $j$  induced by the  $p_{th}$  bound vortex segment of unit strength of component  $l$  is defined as:

$$\bar{u}_{a_{jl}}^*(m, p) = \bar{u}_{a_{j,l}}(m, p) - \bar{u}_{a_{j,l}}(m, p + 1) \quad (26)$$

The above formulas for the calculation of the ‘trailing vortex influence function’  $\bar{u}_{a_{j,l}}(m, p)$  and the ‘horseshoe influence function’  $\bar{u}_{a_{jl}}^*(m, p)$  have been implemented in the *Hough.m* and the *Horseshoe\_int.m* MATLAB functions respectively.

An example of the axial velocity fields predicted by these expressions is presented next. The radial circulation distribution of the propeller inducing these fields follows roughly a typical parabolic shape. The radial variation of the axial velocity at various axial stations is given in Fig.2-7.

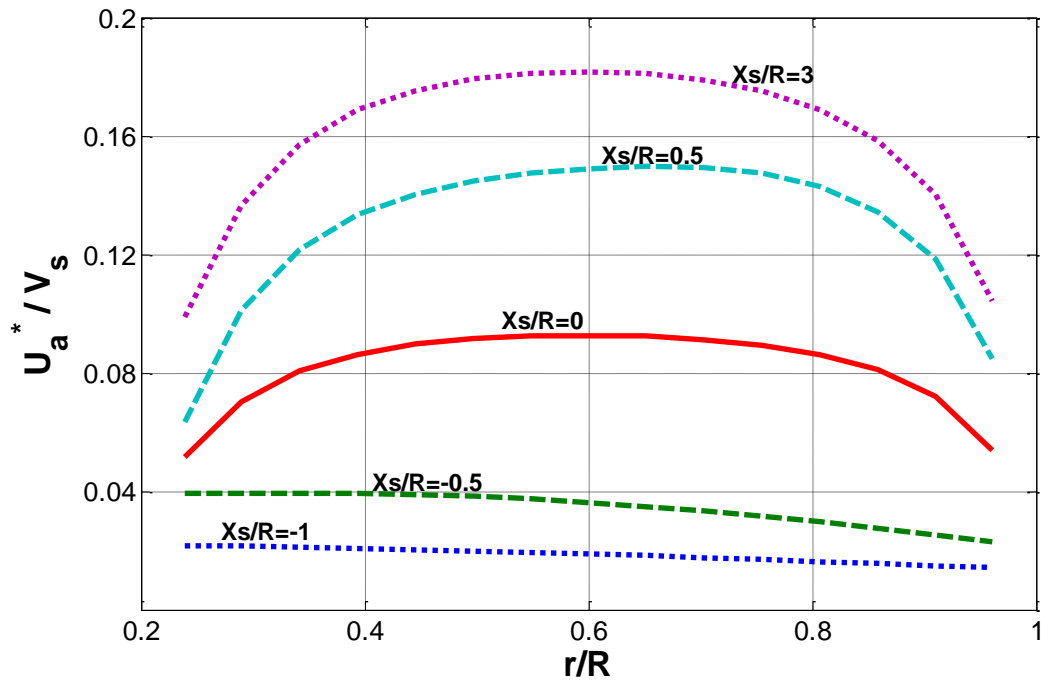


Figure 2-7: Axial velocity profiles at different axial locations

The axial velocity appears to decay fast as we move upstream ( $X_s < 0$ ) and increases just as fast downstream ( $X_s > 0$ ). At a plane located three blade radii downstream it approaches twice its value at the propeller plane, as momentum theory would have predicted. Understanding these trends is crucial for the CRP design where the axial separation between the components of the set has a significant effect on the axial interaction velocity fields experienced by the forward and aft propellers.

### 2.4.2 Tangential Interaction Velocities

Contrary to the axial velocity component, both the bound and the trailing vorticity contribute to the tangential induced velocity. Hough's and Ordway's formula for the total tangential velocity shows that it vanishes everywhere outside the slipstream of the horseshoe vortex and is proportional to the bound circulation values and inversely proportional to the radial distance from the propeller hub. The resulting expression for this velocity is the same as the one obtained by a direct application of Kelvin's theorem using a circular path about the propeller shaft axis.

Based on the notation used by Coney, a horseshoe vortex of strength  $\Gamma$  with lattice points at radii  $r_{v_l}(p-1)$  and  $r_{v_l}(p)$  of one component will induce a tangential circumferential mean velocity on a control point  $r_{c_j}(m)$  of the other component, of:

$$\bar{u}_{t_{j,l}}^*(m, p) = \begin{cases} 0, & S > 0, & -\infty \leq X_f \leq \infty \\ 0, & S \leq 0, & X_f < 0 \\ \frac{-Z_l \Gamma(p)}{2\pi r_{c_j}(m)}, & S < 0, & X_f > 0 \end{cases} \quad (27)$$

where,

$$S = \left( r_{v_l}(p-1) - r_{c_j}(m) \right) \left( r_{v_l}(p) - r_{c_j}(m) \right) \quad (28)$$

The above algorithm for the calculation of the tangential circumferential mean velocity has also been implemented numerically in the *Horseshoe\_int.m* function in a non-dimensional form (Appendix B).

## 2.5 Blade Design

Until this point, the first step for the design of CRP has been accomplished by developing methods for calculating optimum circulation distributions. Once these loading distributions have been determined, the second step involves finding the shape of the blades that will generate the prescribed loadings for the forward and the aft propellers. In general, this is accomplished by combining the lifting line results with theoretical or experimental two-dimensional section data characterized by standardized meanline and thickness types. In the design code used throughout this thesis the procedure for determining the shape of the blades can be described as follows:

First, the required lift coefficients for the blade sections at each radial location are determined based on the non-dimensional circulation values:

$$C_l = \frac{2\Gamma(r)}{V^*(r)c(r)} \quad (29)$$

where  $V^*$  denotes the total inflow velocity,  $c$  is the chord length and  $\Gamma$  the dimensional circulation which can be expressed in terms of the blade radius  $R$ , the ship speed  $V_s$  and the non-dimensional circulation  $G$  :

$$\Gamma(r) = 2\pi R V_s G(r) \quad (30)$$

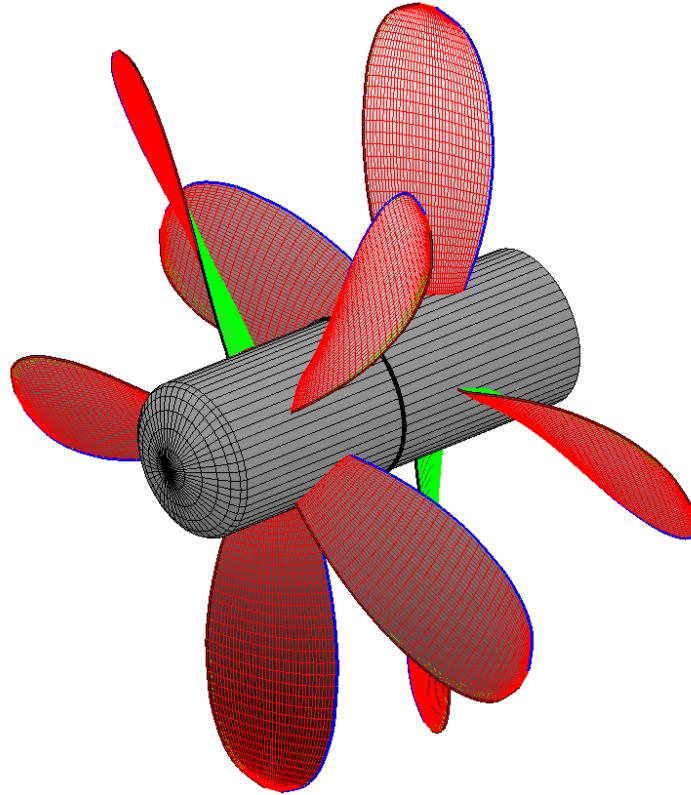
Second, the blade sections are assumed to follow a specific type of meanline and thickness distribution. Typical distributions for propeller applications are the NACA  $a = 0.8$  meanline and the NACA 66(TMB modified) thickness. The properties of this kind of profile will be discussed extensively in Chapter 5, where cavitation considerations are introduced.

Third, the exact blade geometry and orientation are obtained by requiring a shock free entry with each section operating at its ideal angle of attack. The required values for the maximum camber ratio and the ideal angle of attack at each radial location for the NACA  $a = 0.8$  meanline are:

$$f_0/c = 0.0679 \cdot C_l \quad (31)$$

$$a_{ideal} = 1.54 \cdot C_l \quad (32)$$

Finally, the propeller geometry is defined by placing all blades sections such that the nose-tail pitch angle equals the sum of the hydrodynamic pitch angle and the ideal angle of attack. The procedure for determining the propeller geometry discussed above is integrated in the *Geometry.m* MATLAB function. Fig.2-8 shows a sample rendering of a contra-rotating propeller set designed with the current code.



**Figure 2-8: Geometry of CRP set designed with the current code**

We have seen that the maximum camber ratio and the ideal angle of attack have been calculated assuming that the blade sections operate as two-dimensional airfoil sections, which is a reasonable assumption if the blade aspect ratio is high. Unfortunately, marine propellers are forced to have low aspect-ratio blades, the result being that the induced velocity and hence the total inflow velocity will vary along the chord length of the blade sections. In order to account for these discrepancies of lifting line theory and also include the effect of the thickness, lifting-surface corrections are frequently introduced. Morgan, Silovic and Denny (1968) have developed such surface correction factors and their findings indicate that, if three dimensional effects are included, the actual camber and ideal angle of attack are generally greater than two dimensional values at the same lift coefficient.

As far as the radial chord length distribution is concerned, this is a direct user input in the code. The chord lengths are necessary for determining the viscous components of thrust and torque on the lifting line. They are also used for defining the geometry of the complete propeller in the final stage of the design process. The selection of the chord lengths is a tradeoff between the

propeller efficiency and the cavitation performance along with the structural strength. Small chord lengths are expected to improve the efficiency by inducing low viscous losses. On the other hand, the blades would then become more vulnerable to the occurrence of cavitation due to increased loading.

Blade rake and skew are also direct inputs in the part of the code that generates the propeller geometry. Nevertheless, their effects on the efficiency cannot be captured by lifting line codes which place vortices on straight radial lines. If the impact of skewed or raked blades on the efficiency, the cavitation and the vibratory forces is of interest, lifting surface codes should be implemented. In the preliminary design stage though, a simple lifting line approach, as the one used throughout this thesis, is sufficient.



# Chapter 3

## Numerical Predictions

Before moving on to the presentation of characteristic CRP design examples and comparisons, the different definitions of efficiency will be given. The propulsive efficiency can be divided into three components; hull efficiency, relative rotating efficiency and open water efficiency. The hull efficiency is defined as:

$$n_H = \frac{1 - t}{1 - w} \quad (33)$$

The expression  $(1 - t)$  is the thrust deduction factor and  $w$  is the wake fraction. The relative rotative efficiency is given by:

$$n_R = \frac{Q}{Q_o} \quad (34)$$

where  $Q$  and  $Q_o$  are the torques behind the hull and in open water respectively. Open water efficiency for a single propeller is defined as:

$$n_o = \frac{T \cdot V_A}{2\pi n Q_o} \quad (35)$$

where  $T, V_A, n$  are the thrust, the speed of advance and the rotational speed respectively.

Although it is necessary to consider the total propulsive efficiency, the propeller designer has mainly influence on the open water efficiency. In this work only the open water efficiency will be of interest, even though the wake fraction and the thrust deduction are also expected to be different between ships equipped with contra-rotating versus single propellers. Sasaki, et al. (1998) compared the self propulsion factors for ships equipped with CRP and single propulsors and developed a method for estimating these factors.

Before presenting the results from both methods it is necessary to define the open water efficiency for a CRP set. Efficiency of a two stage propulsor is given as:

$$n = \frac{T_1 \cdot V_{A1} + T_2 \cdot V_{A2} - F_h \cdot V_S}{\omega_1 \cdot Q_1 + \omega_2 \cdot Q_2} \quad (36)$$

where  $T, \omega, Q$  are the thrust, rotational speed and torque for each of the propellers and  $V_{A1}, V_{A2}$  are the volumetric mean effective inflows for the forward and aft components respectively defined by:

$$V_A = \frac{2}{(R^2 - r_h^2)} \int_{r_h}^R r V_a(r) dr \quad (37)$$

The hub drag term  $F_h$ , is given by

$$F_h = -\frac{\rho}{16\pi} \left( \ln \frac{r_h}{r_0} + 3 \right) [Z_1 \Gamma_1(1) - Z_2 \Gamma_2(1)]^2 \quad (38)$$

This drag force is experienced due to the presence of a concentrated hub vortex created when the circulation at the hub is not zero. In CRP configurations the hub drag can be reduced or even eliminated if the circulation shed onto the hub from each propeller has the same magnitude. A detailed discussion about the presence of the hub will be given in Chapter 4.

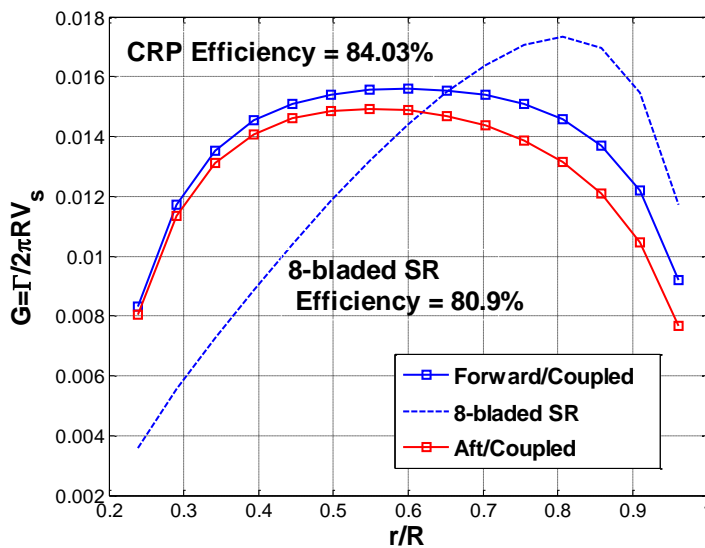
### 3.1 Single Case Comparison

A comparison between the results obtained by the ‘Coupled’ and the ‘Uncoupled’ methods is performed next. Figures of optimum circulation distributions, axial and tangential induced velocities for a pair of free-running contra-rotating propellers are presented. The propellers are rotating at the same speed at an advance coefficient of  $J_s=0.89$  and are required to develop a thrust corresponding to  $C_t=0.69$ .

**Table 3-1: Characteristics of CRP set used in comparing the two design methods**

Number of blades	$Z_1=Z_2=4$
Blade Diameter	$D_1=D_2=D$
Hub Diameter	$D_{hub}/D=0.2$
Thrust Loading/Coef.	$C_T=0.69/K_t=0.214$
Torque Ratio	$Q_2/Q_1=q=1$
Advance Ratios	$J_{s1}=J_{s2}=0.89$
Axial Separation	$X_s/D=0.25$

In all of the cases considered the presence of the hub as a solid boundary is not taken into account, therefore the circulation values at the hub and the associated hub drag forces are equal to zero. Both propellers of the set are four bladed and are required to absorb equal torque, thus the torque ratio is set to one. The axial separation distance between the components is equal to a quarter of the propeller diameter. Viscous forces are neglected. The characteristics of the CRP set are given in Table 3-1. Figs 3-1 and 3-2 give the circulation distributions for the forward and the aft propellers. Also shown is an optimum circulation distribution for a single eight bladed propeller operating at the same advance ratio and required to generate the same thrust. Each blade of the SR propeller produces approximately the same thrust as the blades of the CRP set. Constant radial spacing with 15 panels has been implemented in all cases.



**Figure 3-1: Circulation Distributions as predicted by the 'Coupled' method**

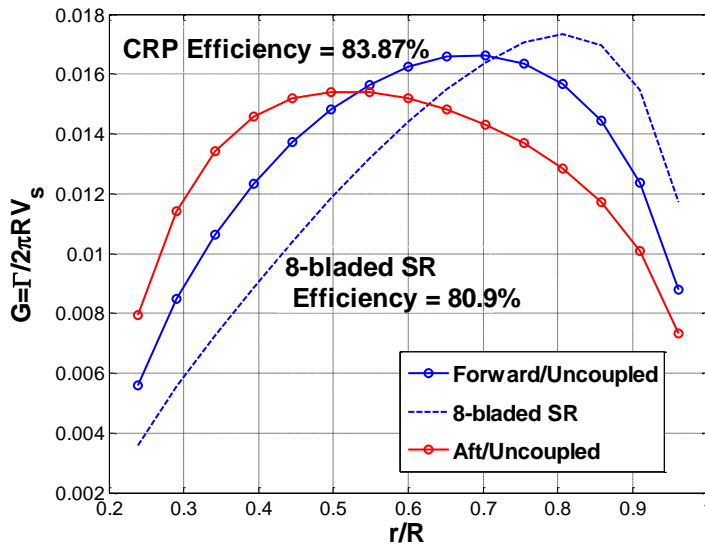


Figure 3-2: Circulation Distributions as predicted by the 'Uncoupled' method

The results for the ‘Coupled’ CRP design method are characterized by an inward shift in the circulation distributions for both the forward and the aft propellers of the set compared to the SR propeller (see Fig.3-1). On the other hand, from the application of the ‘Uncoupled’ method we obtain an inward shifted circulation distribution for the aft propeller (Fig.3-2) while the circulation distribution for the forward one is relatively unaffected. However, the efficiencies obtained by the two methods are comparable (84.03% versus 83.87%). The fact that the two methods specify different circulation distributions as the optima can be attributed to the different optimization routines used. In the ‘Coupled’ method the inward shifted distributions are obtained under the assumption of uniform onset flow, whereas in the ‘Uncoupled’ method the shift of the circulation distributions (always compared to the SR propeller) is due to the modified onset flows, mainly by the interaction velocities. The fact that the interaction velocities on the forward propeller induced by the aft are small provides an explanation for the relatively unaffected circulation distribution on the forward propeller.

Before attempting to explain the efficiency increase of the contra-rotating propulsor configurations a short reference to the energy losses related to the propeller action is necessary. The hydrodynamic energy losses associated with the action of a propeller are due to increases in the kinetic energy of the water passing through the propeller disk area. These losses have two components, axial and rotational. The development of thrust results from axial acceleration of

the water and causes an increase in the axial kinetic energy, while the shaft torque is transferred to the water causing induced rotational velocities and a rotational energy loss. Taken together, these two losses result in the ideal efficiency of the propeller. Moreover, in an ideal fluid the efficiency will approach 100% as the propeller diameter approaches infinity since, as the mass flow increases, the increase in speed for a required increase in momentum becomes smaller. In a real fluid with viscous drag losses there will be an optimum, finite diameter at which the efficiency will have a maximum value for a given thrust and advance coefficient. Inversely, within the limitations of propeller-hull clearances, for a given diameter and thrust loading there will be an optimum advance coefficient for which the highest efficiency will be realized. In the current work, the presentation of efficiency curves for a wide range of thrust loadings and advance coefficients for both CR and SR propellers will show the agreement with the fore mentioned theoretical concepts.

Returning to the comparison between the two optimization methods, the axial and tangential induced velocities far downstream (five diameters downstream of the forward propeller plane) by the same propulsor configurations are given in Figs 3-3~3-6. Some physical insight into the increased efficiency of the CRP pairs over that of the SR propeller (efficiency of the 8-bladed SR propulsor is 80.9%) can be obtained by comparing these velocity components to the one corresponding to the ideal case of the actuator disk.

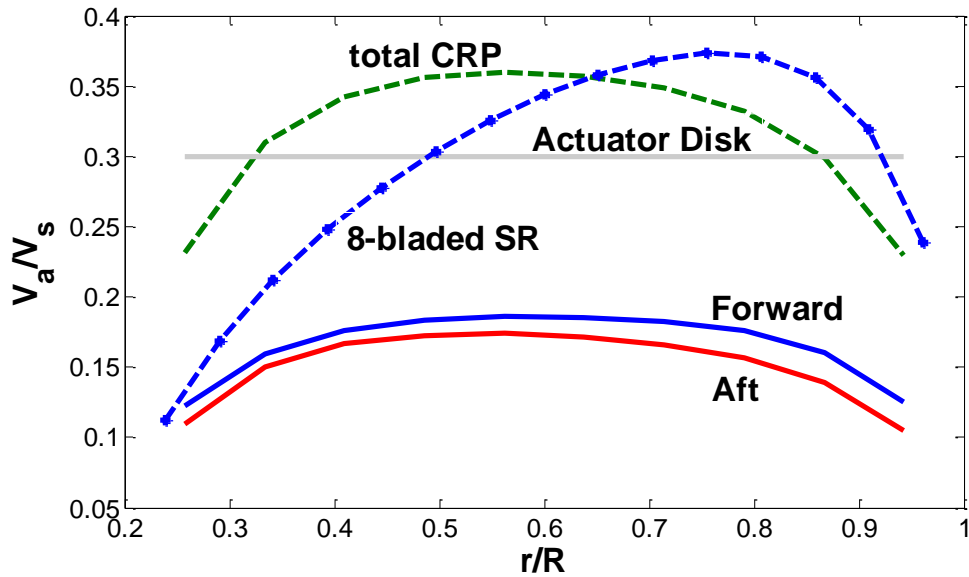


Figure 3-3: Axial induced velocities far downstream / 'Coupled' method

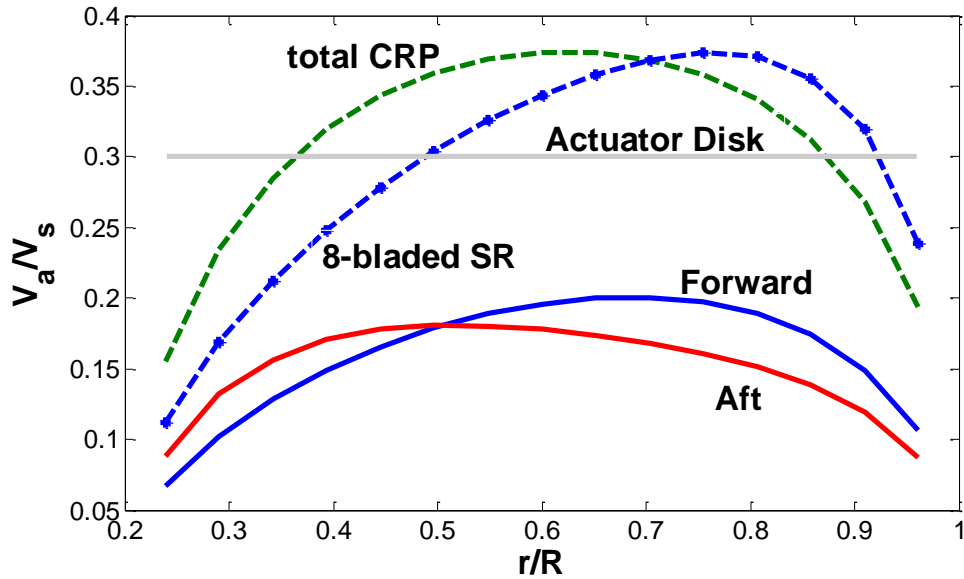


Figure 3-4: Axial induced velocities far downstream / 'Uncoupled' method

The total axial induced velocities of the contra-rotating pairs are more uniform and less concentrated toward the tip than that of the single propeller. This observation is connected with the increased efficiency. These velocities are also closer to the constant value predicted by actuator disk theory. In the limit of light loading we expect the circumferential mean axial induced velocities to approach that of the actuator disk.

The fact that the two methods yield slightly different results in terms of the hydrodynamic efficiency of the CRP sets can partly be attributed to the different forms of the total axial induced velocities. An axial velocity distribution which is closer to the ideal can merely account for the slightly better efficiency of the ‘Coupled’ method.

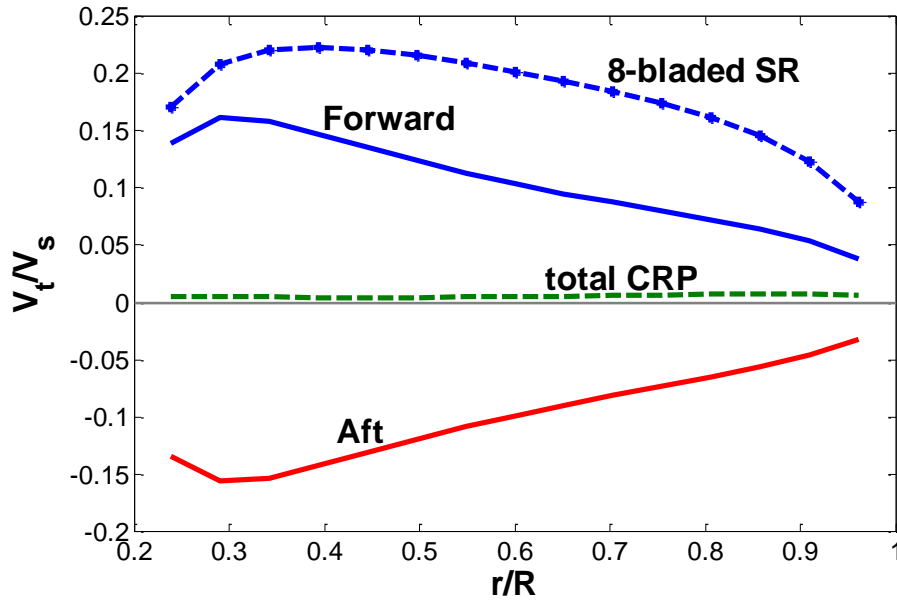


Figure 3-5: Tangential induced velocities far downstream / 'Coupled' method

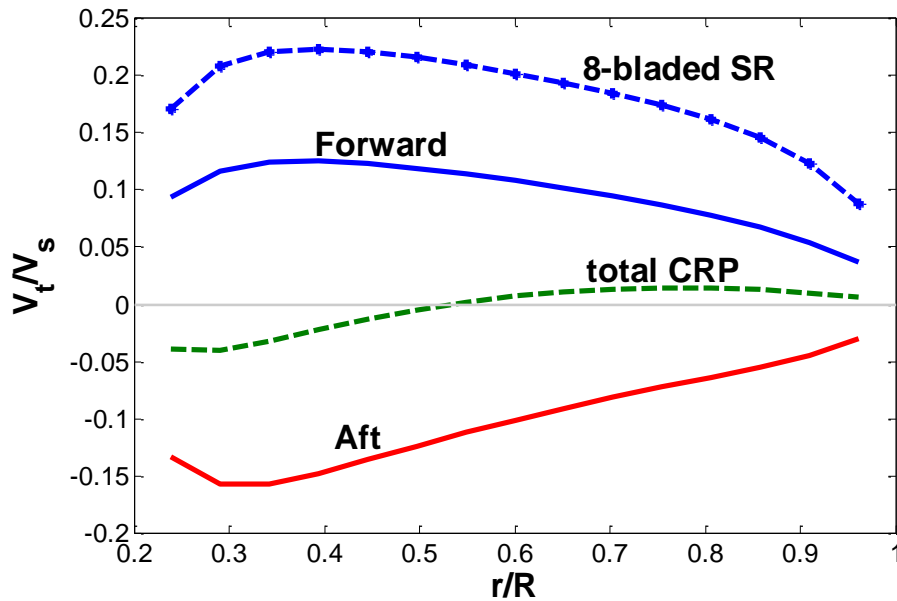


Figure 3-6: Tangential induced velocities far downstream / 'Uncoupled' method

Likewise, the almost complete cancellation of the tangential induced velocities far downstream of the CR propeller planes is another explanation of the efficiency gains over the conventional single propeller. The difference in mean tangential (swirl) velocities for a CRP set compared to a SR propeller is significant and is the primary reason for its increased hydrodynamic efficiency. A decomposition of the propeller efficiency calculated for representative vessels covering a wide range of thrust loadings suggests that even though the rotational energy loss is small compared to the axial, it has a significant effect on the propeller efficiency (Glover 1987). According to the generalized actuator disk theory, the induced tangential velocity is proportional to the bound circulation values and the number of blades and independent of the axial distance downstream of the propeller plane. Since the ‘Coupled’ method gives similar circulation distributions on both components, the total swirl velocity is closer to zero than that predicted by the ‘Uncoupled’ method. This fact also accounts for the slightly better efficiency of the former method.

### **3.2 Parametric Design Results**

Until this point, only results corresponding to a unique combination of thrust loading and advance coefficient have been presented. Values of propeller efficiency covering a wide range of thrust loadings and advance ratios are therefore necessary for understanding the mechanisms governing the action of contra-rotating propellers and the potential benefits over conventional single component propulsors. The calculated efficiencies obtained by both methods for the same CRP sets with torque ratio equal to unity are shown in Fig.3-7.



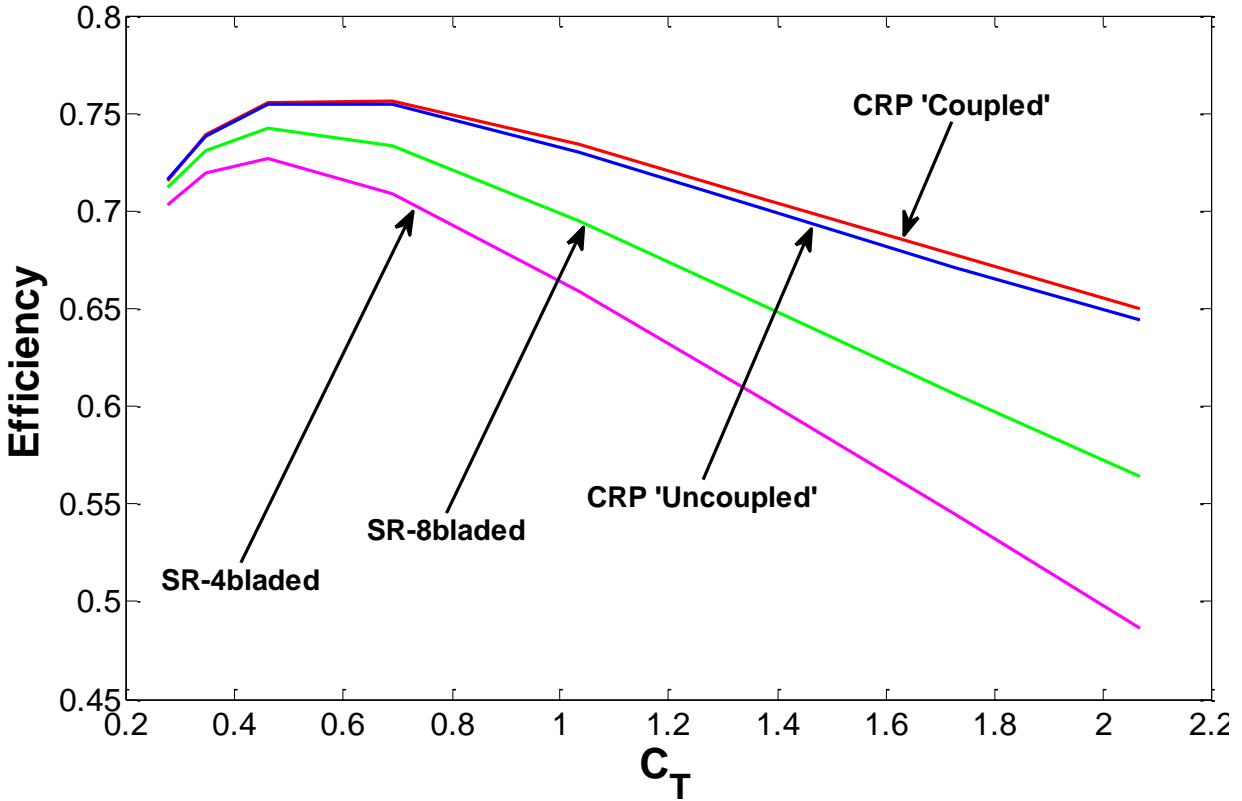
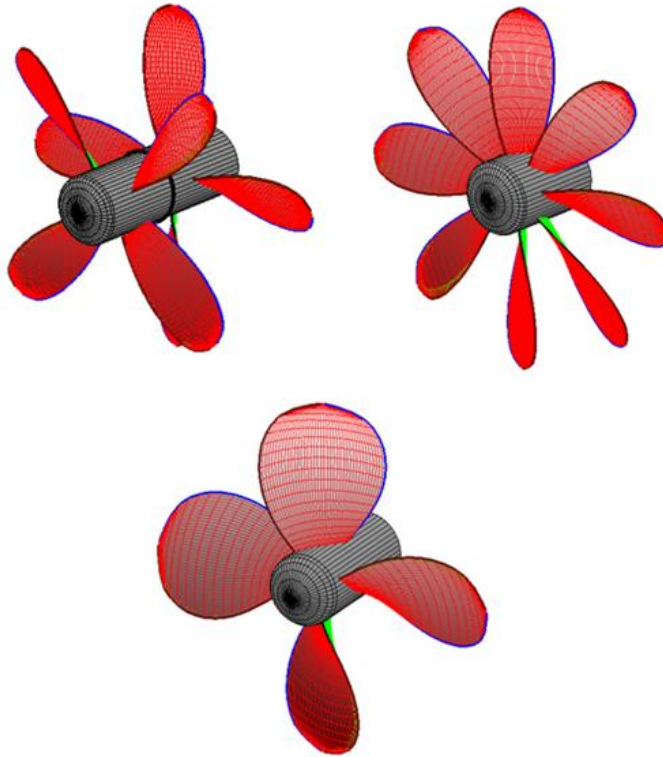


Figure 3-7: Efficiency for various propellers as a function of thrust loading for  $J_s=0.89$  and  $C_d=0.0085$

In this case viscous forces are taken into account by introducing a constant drag coefficient value of 0.0085. Also included are the results for a 4-bladed and an 8-bladed free-running single propellers. All propulsors have the same diameter as before and the same chord distribution, given in Table 3-2, and are required to operate at an advance coefficient of 0.89. However, the chord lengths have been halved for the eight-bladed configurations (the CRP pair and the 8-bladed SR propeller) so that the expanded area ratio is the same among all. Fig. 3-8 shows the geometries of the three different configurations.



**Figure 3-8: Propulsor configurations which have been studied**

**Table 3-2: Chord Distribution for the 4-bladed single propeller**

r/R	c/D	r/R	c/D
0.2	0.174	0.7	0.347
0.3	0.202	0.8	0.334
0.4	0.229	0.9	0.280
0.5	0.275	0.95	0.218
0.6	0.312	0.98	0.144
0.7	0.337	1.0	0

All of the curves in Fig. 3-7 clearly demonstrate that there is an optimum thrust loading value for which propulsor configurations reach their maximum efficiency. Furthermore, another common characteristic is that the efficiency increases as the thrust loading decreases up to a maximum value and then, as the thrust loading further decreases, the efficiency degrades. Comparing to Fig. 3-9, which gives the efficiency for the inviscid case, we conclude that for lightly loaded propellers the frictional losses constitute a significant part of the total losses, but they become less significant at higher thrust loadings. All of the propulsor alternatives show the trend of

decreased efficiency with increased thrust loading, something that also describes the actuator disk. However, the percent difference in efficiency between that of the actuator disk and the single propellers increases with increasing thrust. The CRP configurations follow curves of lower decrease rate which effectively makes them even more efficient compared to SR propellers as the thrust loading increases. Both the 'Coupled' and the 'Uncoupled' methods yield similar results over a wide range of thrust. The small difference can mainly be attributed to the better cancellation of the swirl velocities obtained by the 'Coupled' method.

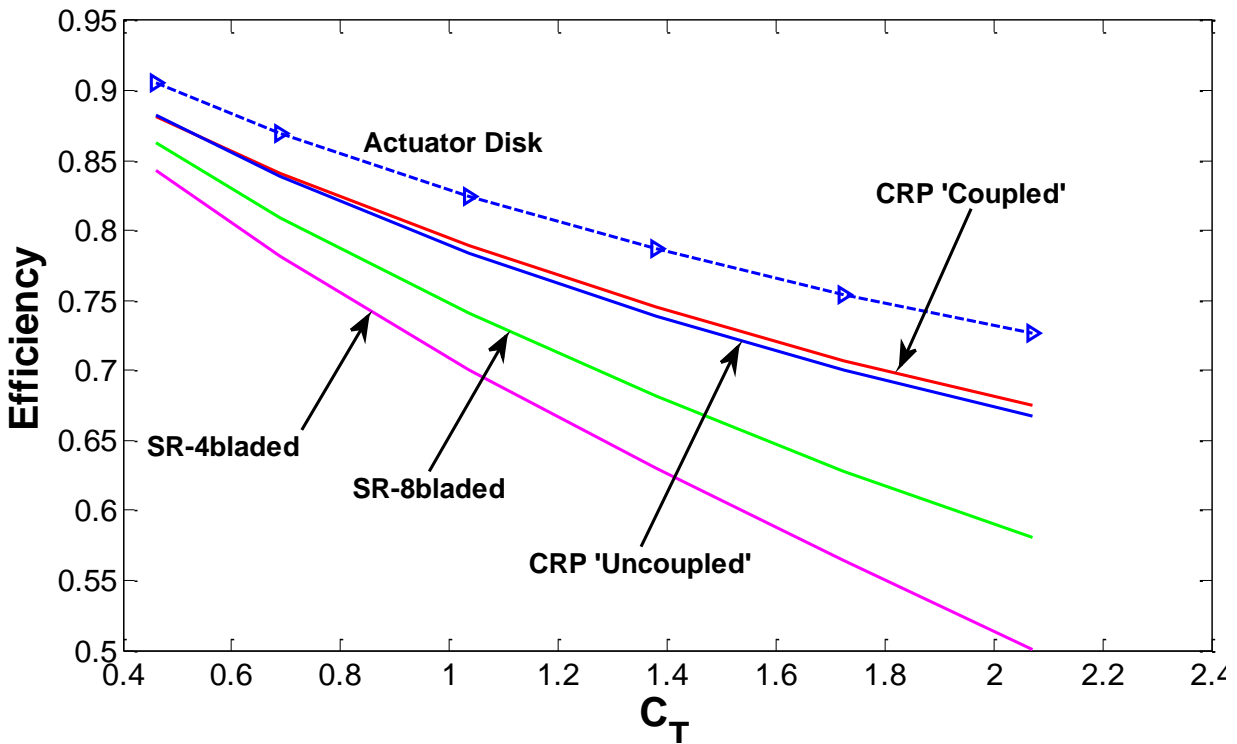
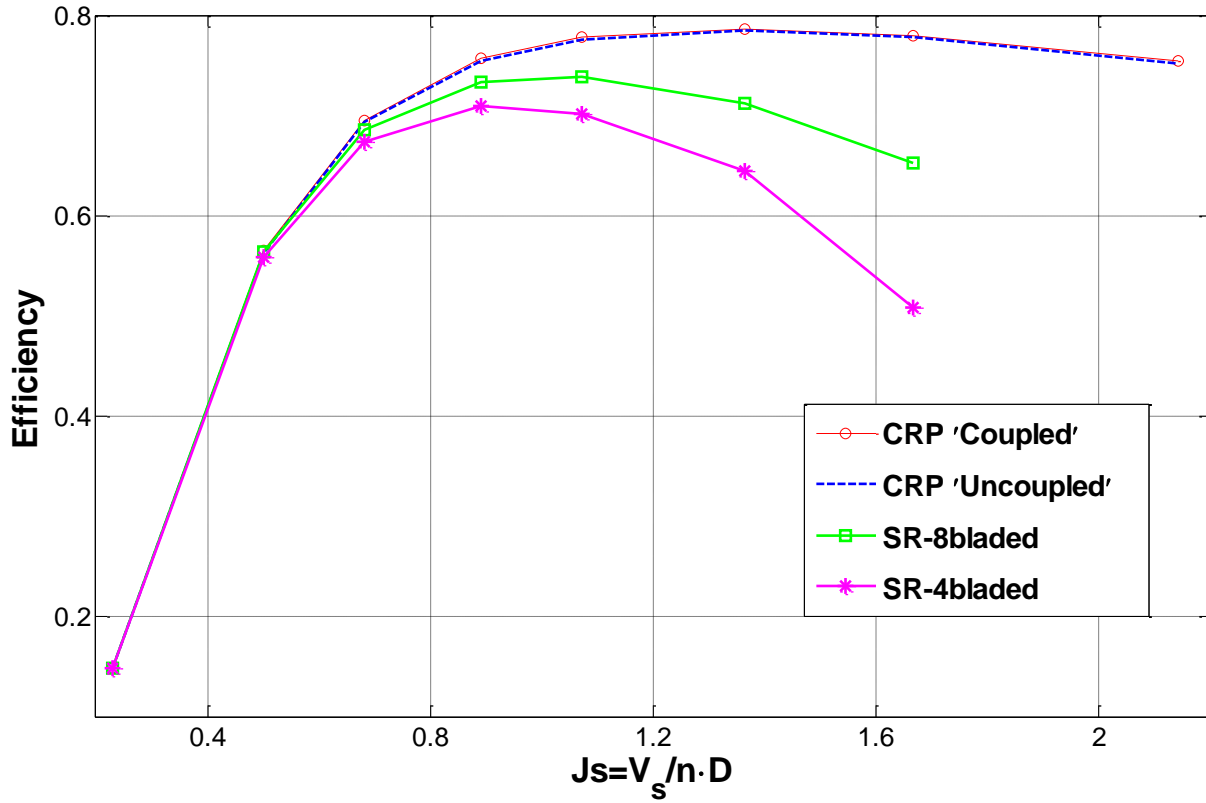


Figure 3-9: Ideal Efficiency as a function of thrust loading for  $J_s=0.89$

Fig. 3-10 shows the real efficiency (including viscous drag) of the same propulsor configurations as a function of the advance coefficient for a constant thrust loading of  $C_t=0.69$ . As the advance coefficient  $J_s$  approaches zero the rotational speed increases and the efficiency decreases due to high frictional losses.



**Figure 3-10: Efficiency as a function of advance coefficient with  $C_d=0.0085$**

Fig. 3-11 gives the results for the CRP sets for both the viscous and the inviscid cases. In the limit of zero  $J_s$  (infinite rotational speed) the propellers behave as an actuator disk. Once again the efficiency gain of the CRP sets increases as the rotational speed decreases. The hydrodynamic losses in this regime become dominant but the swirl cancellation has a favorable effect on efficiency.

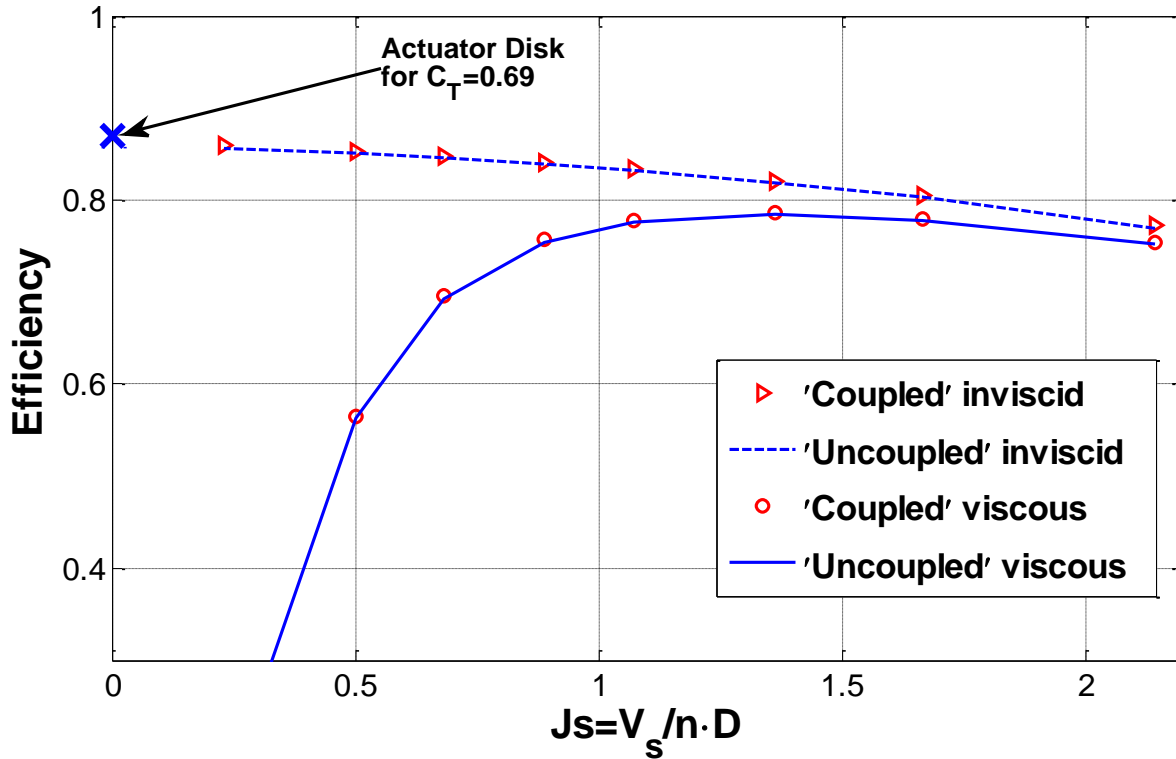


Figure 3-11. Efficiency of CRP for viscous-inviscid case ( $C_t=0.69$ )

### 3.3 Comparison between the Two Methods

The preceding sections dealt with the detailed description of the two circulation optimization methods and the presentation of the results. Their distinguishing characteristics are given here in order to become clear which method captures more of the CRP properties.

First of all, the 'Coupled' method yields slightly better results in terms of efficiency compared to the 'Uncoupled' one. As explained this is attributed to the more effective cancellation of the tangential induced velocity field. Secondly, the 'Coupled' method could easily handle other types of multi-component propulsors such as pre and post-swirl propeller-stator combinations. The only necessary modifications in the code would relate to the calculation of the interaction velocities. Kerwin, Coney and Hsin (1986) have successfully applied this procedure for designing a variety of multi component propulsors. Another feature that favors the 'Coupled' method is that the system of equations can even be solved for designing a conventional single propeller without any changes in the code. In that case of course all of the input parameters for the aft propeller would have to be set to zero as explained in section 2.3.

However, the most important property of this method is that it can handle the case of finite hub loading, something which will be covered in detail in section 4.3. The presence of the hub can be ‘captured’ by the optimization algorithm in the form of an additional drag term which is added to the total thrust requirement (see eq. 11). On the other hand, the ‘Uncoupled’ method is computationally ‘simpler’ since a procedure for the design of the single propeller is iteratively executed. This means that optimization methods other than the variational, such as Lerbs’ method, can be used instead. Nevertheless, finite hub loadings could not be modeled with the ‘Uncoupled’ method, at least in its current form.

With respect to the run times of the respective codes they are approximately the same. The functions for calculating the interaction velocities are the most time demanding and since these are used in both methods, the computing times are comparable. As for the limitations regarding the type of radial spacing and the number of panels on the lifting lines, convergence could not be achieved in either method for cosine spacing with more than 15 panels. This led to the adoption of constant spacing.

For all of the reasons mentioned above, and particularly for the ‘loaded hub’ capability, it was decided that the ‘Coupled’ method should be used in the CRP design process.

# Chapter 4

## Additional Considerations

### 4.1 Effect of Lagrange Multipliers

The values of the Lagrange multipliers  $\lambda_Q$  and  $\lambda_T$  used in the ‘Coupled’ method affect the ‘optimum’ circulation distributions considerably. Consequently, the induced velocity field far downstream is also very sensitive to the multipliers. On the other hand, the effect on the propeller efficiency is very weak. Fig. 4-1 shows the efficiency predicted by the ‘Coupled’ method for different combinations of  $\lambda_Q$  and  $\lambda_T$ . The characteristics of the CRP set are the same as those given in Table 3-1 and the frictional losses are neglected. For the range of  $\lambda_Q$  and  $\lambda_T$  considered, the efficiency appears to be relatively independent of  $\lambda_T$  and only slightly influenced by  $\lambda_Q$ .

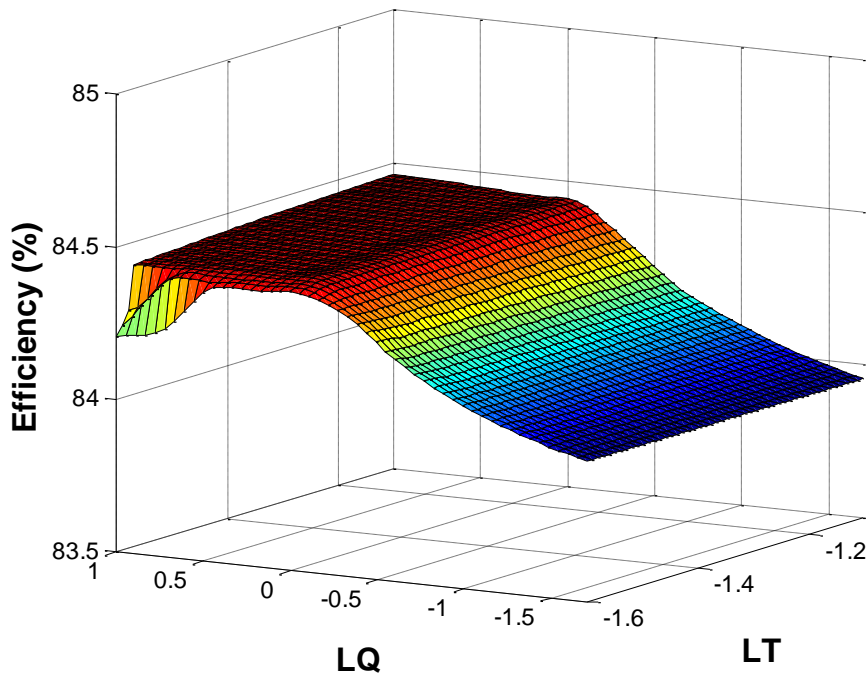


Figure 4-1: Efficiency as a function of the Lagrange multipliers

The situation is reversed as far as the circulation distribution is concerned. Fig. 4-2 gives the circulation distribution on the forward and aft propellers for different values of  $\lambda_Q$  when  $\lambda_T$  is equal to -1.38.

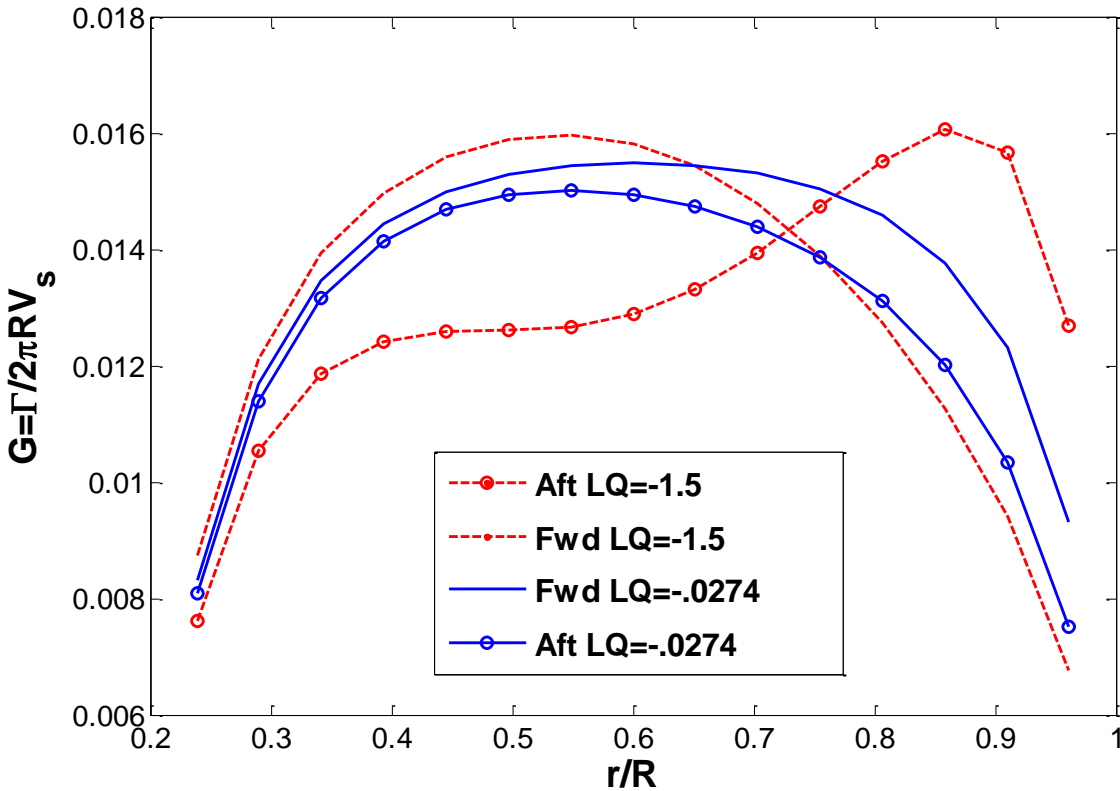


Figure 4-2: Circulation Distributions for different values of  $\lambda_Q$

Likewise, the axial and tangential induced velocity fields far downstream show a strong dependence on the Lagrange multipliers (see Figs 4-3~4-4). An explanation for the fact that the efficiency is relatively unaffected (84.12% versus 84.49%) can be given by comparing these velocity fields. The tangential induced velocity corresponding to the first case, for which  $\lambda_Q = -0.0274$ , is closer to the actuator disk result (Fig. 4-3). Conversely, the axial induced velocity distribution in the second case, for which  $\lambda_Q = -1.5$ , is closer to the ideal, implying lower rotational losses (Fig. 4-4). Therefore, the Lagrange multipliers can be considered as a means of modifying the blade loading by ‘redistributing’ the hydrodynamic losses without sacrificing the propeller efficiency.



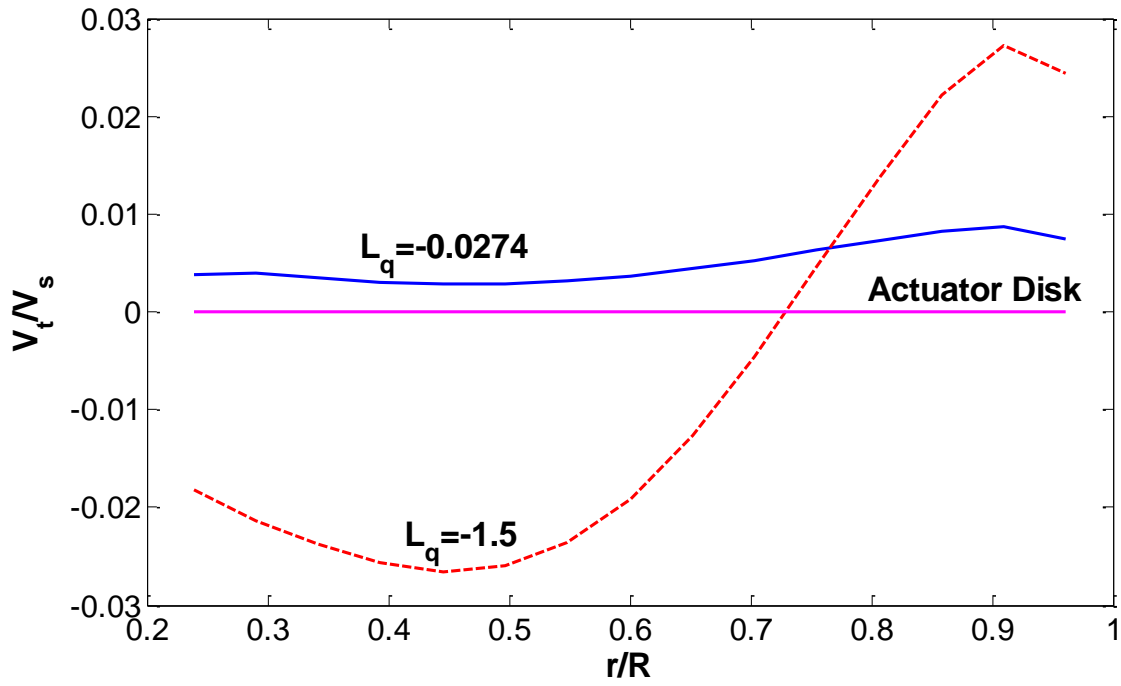


Figure 4-3: Total tangential induced velocities far downstream

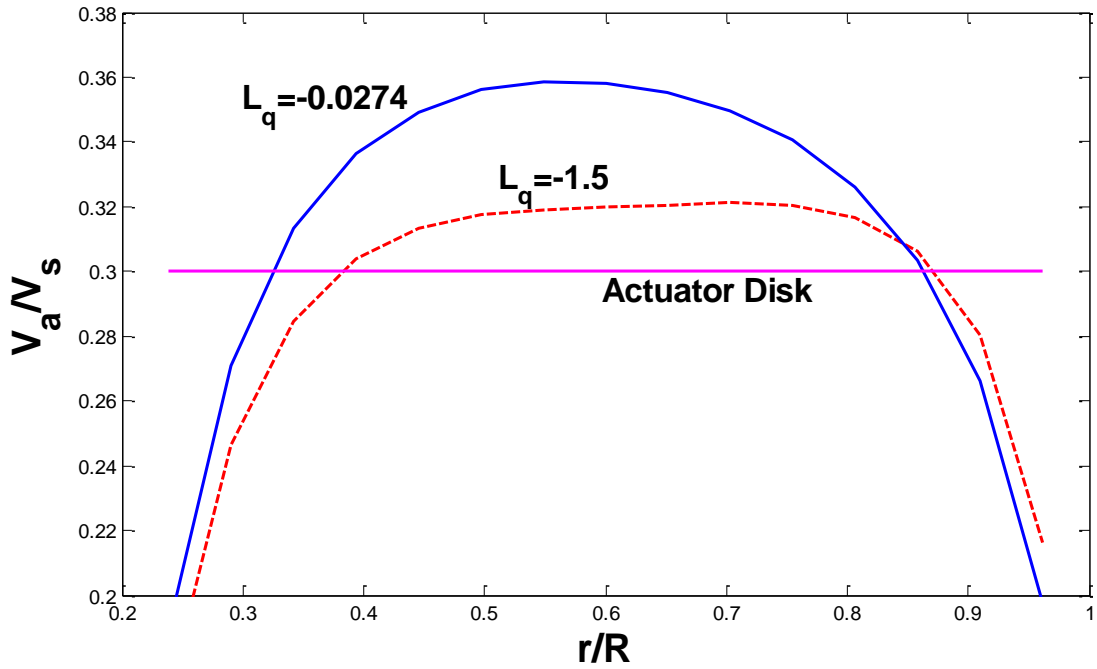


Figure 4-4: Total axial induced velocities far downstream

## 4.2 Axial Spacing

The effects of the axial clearance between the propellers on the performance and the geometric characteristics of the CRP set are investigated in this section. The particulars and the design conditions for the CRP set used for this purpose are the same as those presented in section 3.1. Only the separation distance was allowed to vary with the axial spacing ratio  $X_s/R$  ranging from 0.25 to 1.5. The geometric pitch ratios at  $r=0.7R$ , as well as the efficiency are given in Fig. 4-5.

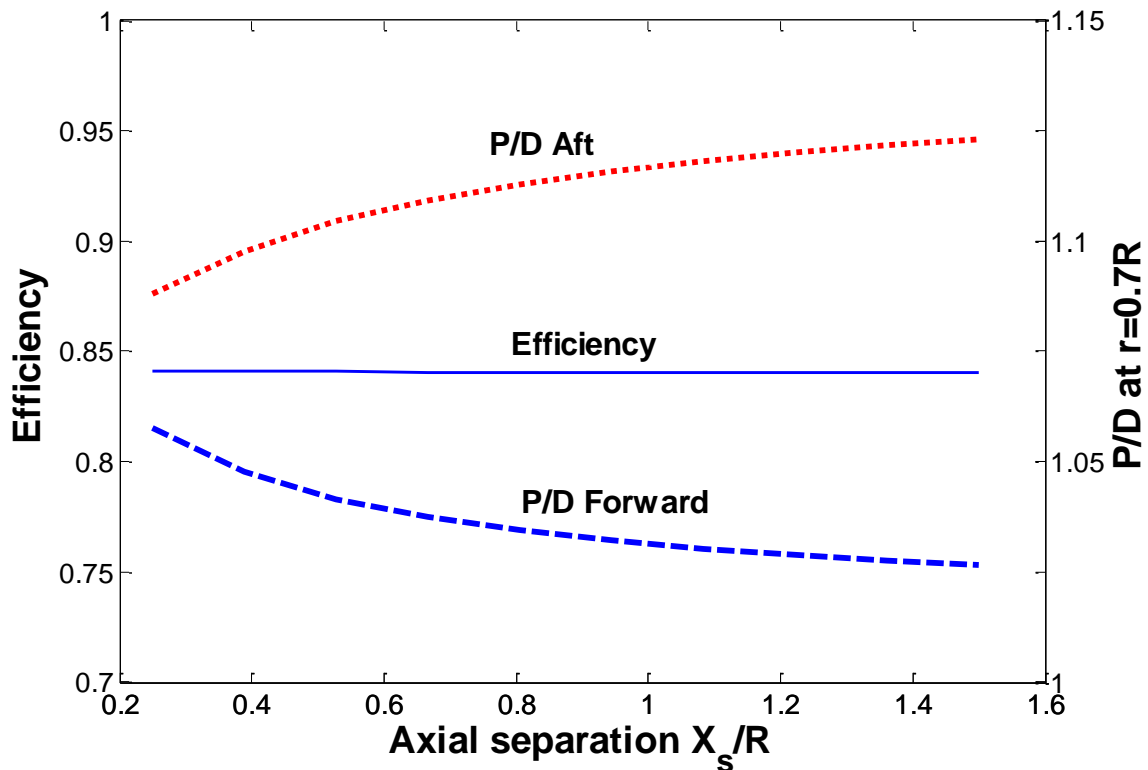


Figure 4-5: Effect of Axial Spacing on Efficiency and Geometric Pitch Ratio

Over the range of the axial spacing ratios there is clearly no effect on the propeller efficiency. On the other hand, the change in the required geometric pitch ratios is considerable. As the separation increases, the design pitch of the aft propeller will have to increase in order to adapt to the higher axial interaction velocities induced by the forward propeller. In contrast, the axial velocities induced on the forward propeller by the aft one decrease as the separation between them increases resulting in lower geometric pitch values.

In practice, the CRP components are usually placed as close as possible in order to avoid interferences with other hull appendages and due to strength and vibration considerations. Moreover, small axial clearances are beneficial from a cavitation standpoint, since it is harder for the cavitating tip vortices of the aft propeller to hit the surface of the rudder and erode it.

### 4.3 Finite Hub Loading

In all of the results presented so far the circulation distribution vanishes at the blade root. This is a consequence of the fact that the presence of the hub is neglected and the blades are modeled as free wings. No modifications in the flow characteristics and in the loading distribution at the inner radii are introduced. The only indication of the existence of a hub is the fact that the lifting lines start at a radial distance  $r = r_{hub}$  and not at  $r = 0$ .

Lerbs (1955) considered the hub as an infinitely long cylinder and concluded that a finite value of circulation at the hub was physically impossible because then a circulation discontinuity at  $r = r_{hub}$  would arise. His arguments though were later rejected by experimental findings which confirmed that finite circulation at the hub was a physically attainable result.

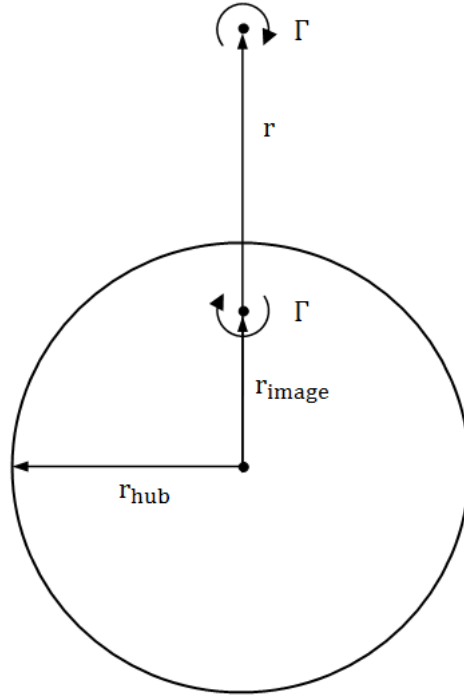
Kerwin and Leopold (1964) proposed using a circular solid boundary for representing the hub. This method is based on the fact that two-dimensional vortices of equal and opposite strength located on the same lifting line, induce no net radial velocity on a circle of radius  $r_{hub}$  when:

$$r_{image} = \frac{r_{hub}^2}{r} \quad (39)$$

where  $r$  is the radius of the outer vortex (on the lifting line) and  $r_{image}$  is the radius of the inner (image) vortex with a pitch given by:

$$\tan\beta_{image} = \frac{r}{r_{image}} \tan\beta \quad (40)$$

Hence, the hub is represented by using a set of image vortices located inside the hub boundary (see Fig. 4-6). The axial and the tangential velocities induced by these image vortices are then added to the inductions of the lifting line vortices, resulting in finite circulation values at the hub.



**Figure 4-6: Vortex on a lifting line and its image**

Except for the modifications in the self – induced velocities  $u_{\alpha}^*$  and  $u_t^*$ , in the case of contra-rotating propellers the effect of the image vortices on the axial interaction velocity  $\bar{u}_{\alpha j l}^*(m, p)$  is also included. As far as the tangential interaction component is concerned, no change is expected since the control points on the lifting line lie outside the ‘fictitious slipstream’ of the image vortices. The numerical implementation of the hub modeling by using the ‘Coupled’ method, and the corresponding modifications in the calculation of the interaction velocities are included in the *Horseshoe\_int.m* and the *Horseshoe\_self.m* MATLAB functions (Appendix B).

#### **4.4 Hub Drag Cancellation**

The presence of the hub is also responsible for the formation of a concentrated vortex extending infinitely downstream from the end of the hub. This hub vortex gives rise to a drag force due to the decreased pressure inside the vortex. According to Wang (1985) the drag force can be divided into two parts; the force in the area inside the hub vortex core and the force in the area outside of it. By modeling this vortex as a Rankine type vortex Wang was able to find an expression for the total force induced on the hub by a semi-infinite vortex:

$$F_h = -\frac{\rho}{16\pi} \left( \ln \frac{r_h}{r_0} + 3 \right) (Z\Gamma_0)^2 \quad (41)$$

where  $r_h$  is the hub radius,  $r_0$  is the radius of the vortex core,  $Z$  the number of blades and  $\Gamma_0$  the circulation value at the hub boundary. From the above expression it is obvious that as the radius of the vortex core decreases, the drag force increases with a consequent reduction in efficiency. In the current numerical implementation the ratio of the hub radius over the vortex radius is set to one if the hub is present.

For a single propeller this drag term is nonzero if the circulation at the hub is finite. This explains the traditional practice of unloading the propeller near the blade root instead of keeping the optimum circulation distribution. However, Coney has shown that if the hub drag term is neglected, the hub loaded circulation distribution gives a slightly increased efficiency over the ‘hubless’ propeller.

In the case of contra-rotating propellers the hub drag can be reduced or even eliminated by requiring that each component sheds equal and opposite total circulation onto the hub. This can be understood by looking at the expression for the drag force which is:

$$F_h = -\frac{\rho}{16\pi} \left( \ln \frac{r_h}{r_0} + 3 \right) (Z_1\Gamma_{0_1} - Z_2\Gamma_{0_2})^2 \quad (42)$$

where the indices 1,2 correspond to the forward and the aft propellers of the set respectively. Under the discrete vortex lattice model the circulation values  $\Gamma_{0_1}, \Gamma_{0_2}$  at the hub would correspond to the values at the innermost control points  $\Gamma_1(1), \Gamma_2(1)$ , the exact location of which will depend on the type of spacing used (cosine or constant).

The presence of the hub should also be reflected in the formulation of the optimization equations for the circulation distribution. A hub drag term is then included in the objective function  $H$  (compare with eq.11):

$$H = (\omega_1 Q_1 + \omega_2 Q_2) + \lambda_T (T_1 + T_2 - F_h - T_r) + \lambda_Q (q Q_1 - Q_2) \quad (43)$$

Consequently, each of the partial derivatives of the objective function H with respect to the circulation values will have to include two additional terms pertaining to the finite circulation values at the two innermost locations on the lifting lines for the forward and the aft propellers:

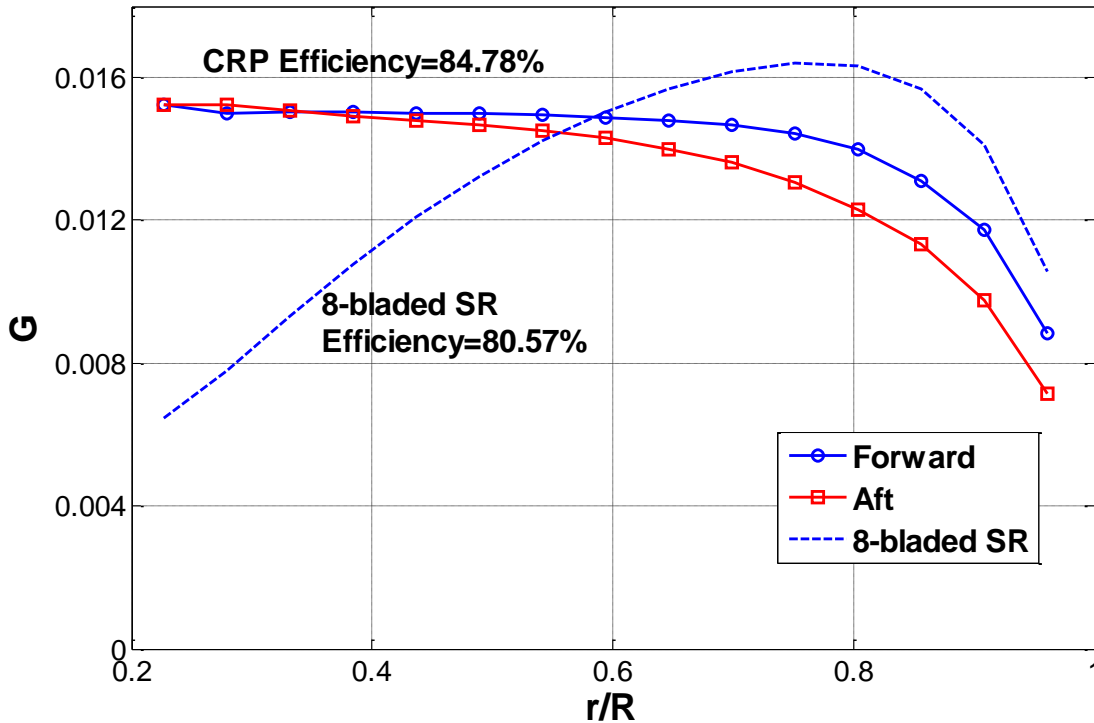
$$\frac{\partial H}{\partial \Gamma_1(1)} = \dots - \frac{\rho}{16\pi} \left( \ln \frac{r_h}{r_0} + 3 \right) [2Z_1^2 \Gamma_1(1) - 2Z_1 Z_2 \Gamma_2(1)] + \dots \quad (44)$$

$$\frac{\partial H}{\partial \Gamma_2(1)} = \dots - \frac{\rho}{16\pi} \left( \ln \frac{r_h}{r_0} + 3 \right) [-2Z_1 Z_2 \Gamma_1(1) + 2Z_2^2 \Gamma_2(1)] + \dots \quad (45)$$

The trust constraint equation will also include the hub drag term such that:

$$\frac{\partial H}{\partial \lambda_T} = 0 = (T_{I_1} + T_{V_1}) + (T_{I_2} + T_{V_2}) - F_h - T_r \quad (46)$$

If the above modifications are applied to the design procedure for a CRP set, the resulting optimum circulation distributions are expected to display finite values at the hub such that the net circulation at this point vanishes. In addition, if the CRP components have the same number of blades, the innermost circulation values will be equal and opposite. This trend is observed in the results for a CRP set including the effects of the hub. In this case the advance coefficient is 0.89, and the required thrust loading is 0.69. Both the forward and the aft propellers are 4-bladed, the torque ratio is set to unity and viscous forces are neglected.



**Figure 4-7: Hub Loaded Circulation Distributions as predicted by the ‘Coupled’ method**

Fig. 4-7 shows that the maximum values of the circulation distributions for the CRP pair appear near the hub. The distribution for a single 8-bladed propeller required to generate the same thrust and having the same rotational speed as each of the CRP components is also plotted. The characteristics of the above propulsor alternatives are the same to the ones studied in Chapter 3, being only different in that the hub was then ignored (compare with Fig. 3-1). Table 4-1 presents the results for these four cases. Interestingly, the efficiency is not expected to alter significantly when the hub is loaded, even though the corresponding circulation distributions differ considerably. A possible explanation would be that the inner radii contribute little to the overall propeller forces as a result of the low rotational speeds in this region.

**Table 4-1: Comparison between efficiencies of hub-loaded and hubless variants**

	SRP	CRP	% increase
Hubless	80.9	84.03	3.87
Hub Loaded	80.57	84.78	5.22

The hub-loaded CRP set is slightly more efficient than the unloaded variant. On the contrary, inclusion of the hub as a solid boundary leads to minor decrease in efficiency for the single propeller. The increase in the efficiency gain with a contra-rotating propeller when the hub is present can be explained by looking at the hub drag force for these two cases. The effect of the hub on the efficiency will always be negative for the single propeller due to the drag force associated with it. As for the CRP set, this drag force can be even be eliminated by designing the blades such that the net circulation at the hub vanishes.



# Chapter 5

## Cavitation on Marine Propellers

### 5.1 Introduction

Cavitation is a phenomenon observed in high speed flows where the local absolute pressure of the fluid reaches its vapor pressure at the ambient temperature. Cavitation occurs on nearly all ship propellers, especially when these are highly loaded, and can lead to extensive problems if not acknowledged in an early design stage. The most frequently occurring problems are two; pitting and erosion on propeller blades and appendages produced by vapor bubble collapse, and high frequency noise and vibration in the afterbody due to cavitation-induced pressure fluctuations on the hull. The latter is of great importance to naval vessels where the acoustic detection of ships and submarines is a major concern.

In order to study the behavior of a propulsor with regard to cavitation and to get an estimate of the cavitation pattern on its blades, experiments with models are usually conducted in cavitation tunnels prior to construction. In the preliminary design stage however, cavitation tunnel tests are far too expensive and time-consuming to be justified. Therefore numerical simulations are performed to reveal the approximate cavitation patterns on propeller blades. In the current work such a numerical method for obtaining the pressure distribution on the blades is described. Comparison with the local cavitation numbers for different blade radii will provide a first estimation of the blade areas which are prone to cavitation inception.

To predict the inception of cavitation we are interested in finding the position on the body where the local pressure drops below the vapour pressure. Alternatively we can compare the so-called cavitation number  $\sigma$  ( $\sigma_V$ ) with the minimum pressure coefficient ( $C_{Pmin}$ ). The cavitation number or cavitation index is given by:

$$\sigma_V = \frac{p_{atm} + \rho gh - p_V}{\frac{1}{2}\rho V^2} \quad (47)$$

where  $p_{atm}$  is the ambient pressure acting on the surface of the fluid,  $g$  the gravity acceleration,  $h$  the submergence height,  $p_V$  the vapor pressure depending on temperature and  $V$  the total inflow velocity at the point where the cavitation number is evaluated. This index is a measure of the tendency of the fluid to cavitate. The larger the value of  $\sigma_V$ , the smaller the likelihood for cavitation.

On the other hand, the minimum pressure coefficient is a measure of the maximum difference between the ambient pressure and the local pressure on the body under examination. It is defined as:

$$C_{Pmin} = \frac{p_{min} - p_{\infty}}{\frac{1}{2}\rho V^2} \quad (48)$$

where  $p_{min}$  is the minimum pressure at some point on the body,  $p_{\infty}$  is the ambient pressure equal to  $p_{atm} + \rho gh$  and the denominator corresponds to the stagnation pressure as in the definition of the cavitation index.

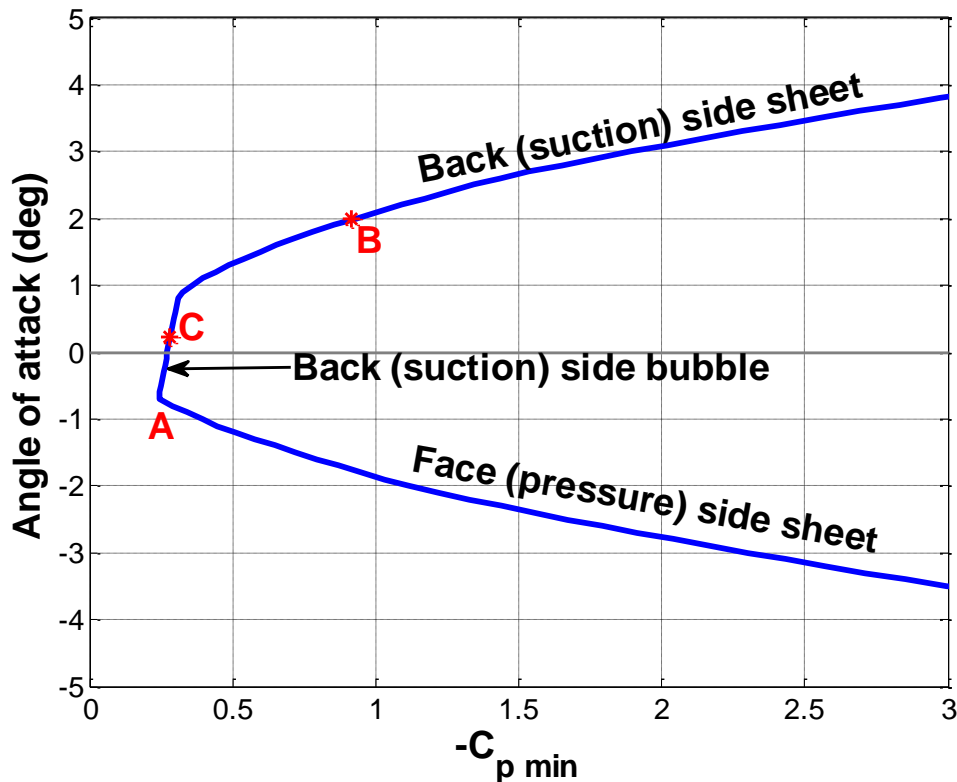
The criterion for the inception of cavitation is  $p_{min} < p_V$  which can alternatively be expressed as  $-C_{Pmin} > \sigma_V$  if the previous definitions are used. The latter expression is more useful since plots of the pressure coefficient over the section chord are usually available for blade sections used in propeller design.

## 5.2 Blade Section and Cavitation Performance

The National Advisory Committee for Aeronautics (NACA) in the early 1930's started developing and systematically testing aerofoil shapes after using a method for combining meanline and thickness distributions to obtain the desired cambered wing sections. An extensive list of experimental data for such airfoil sections is provided by Abbot and von Doenhoff (1959). Some of these sections have been adopted for the design of marine propellers because of their

relatively flat pressure on the suction side which results in relatively wide cavitation-free envelopes.

Fig.5-1 shows the minimum pressure envelope for a section with a NACA 66 (TMB modified nose and tail) thickness and a NACA  $\alpha=0.8$  camberline with a maximum camber ratio  $f_0/c$  equal to 0.01 and a maximum thickness ratio  $t_0/c$  of 0.08. This envelope is similar to the ones developed by Brockett (1966), but a vortex lattice method (VLM code) has been used for determining the minimum pressure coefficients for a range of angles of attack instead of an analytical expression. The NACA  $\alpha=0.8$  loading and associated camber has been widely adopted by aeronautical engineers and later by naval architects and propeller designers.



**Figure 5-1: Minimum pressure envelope for NACA 66 section (TMB modified) with the NACA  $\alpha=0.8$  camberline having a maximum camber ratio of 0.01 and a maximum thickness ratio of 0.08**

From the diagram, four primary areas are identified: the cavitation free area inside the ‘bucket’ and the areas where back side sheet, back side bubble and face side cavitation can be expected. The NACA 66 (TMB modified) profile was designed so that the pressure on most of the suction

side from the leading edge is very close to a flat pressure distribution. However, due to a decomposition of thickness and camber when using linear wing section theory, the pressure on the pressure side of the profile is not well controlled. The cavitation bucket has a sharp corner at point A, which limits the width of the bucket. Points B and C correspond to cases of back sheet and back bubble cavitation, for which the respective pressure distributions will be presented in the next section.

Whether or not bubble cavitation on the suction side of the blade and sheet cavitation at the leading edge on the pressure side are erosive, they have the reputation of being erosive and should be avoided at the design stage. Therefore, the blade sections are designed with a certain pitch and camber combination, so that there is always enough margin against pressure side cavitation. The consequence of this design philosophy is that the design operation point can no longer be placed at the angle of attack of shock-free entry of the profile, but on the side of back (suction) cavitation.

In this thesis however, only the propeller performance at the design point is of interest, and the most common in practice chordwise load distribution is considered for the blade sections. The sections are therefore required to operate in the shock free condition, being oriented at the ideal angle of attack. This means that the lift is obtained exclusively due to camber and the local spikes in the pressure distribution near the leading edge are minimized.

On the other hand, if a propeller performance analysis at off design points were to be performed, the blade sections would operate at a relatively wide range of angles of attack. In that case the design of blade sections would be challenging, since the thickness ratio of the sections would have to be chosen based on strength and cavitation considerations. The variation in the operating angle of attack would then be known or could be estimated. It would be necessary to find a thickness ratio and an average operating angle of attack such that the thickness is not less than what the strength considerations would permit, and such that  $-C_{p\ min}$  is less than the cavitation index over the range of angle of attack variations. A collection of cavitation buckets for different thickness ratios could then be used for selecting the appropriate sections at each radial position on the propeller blade. Fig. 5-2 shows the minimum pressure envelopes for NACA 66 sections with the NACA  $\alpha=0.8$  camberline having the same camber ratio ( $f_0/c = 0.01$ ) but different thickness ratios ranging from 0.02 to 0.2.

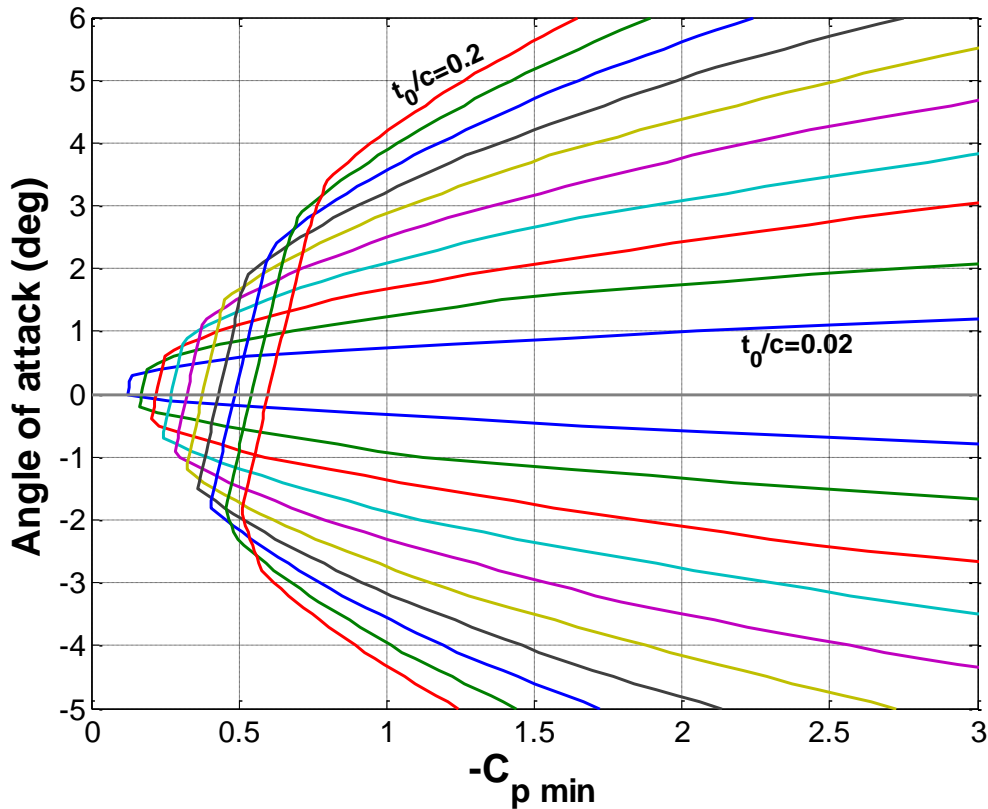


Figure 5-2: Minimum pressure envelopes using vortex lattice method (VLM code) for NACA 66 sections (TMB modified) with the NACA  $a=0.8$  camberline having a maximum camber ratio of 0.01

It is evident that the greater the thickness ratio, the greater the maximum pressure drop (shallower bucket depth) and the more likely is that back bubble cavitation will form. Thicker sections, however, are less sensitive to variations in the angle of attack, that's why they appear to have wider cavitation-free envelopes. Conversely, as the sections become thinner, the margin against bubble cavitation on the back side increases (deeper bucket) but at the same time the width of the envelope narrows.

In the current work, however, we do not intend to address the geometric design of the blade sections in order to minimize cavitation, but to analyze the propeller blades with respect to the likelihood of cavitation inception. We are therefore interested not only in calculating the minimum pressure coefficients of the blade sections, but also in determining the pressure distribution along the section chord at various radial positions from the propeller hub to the blade tip. To calculate the distribution of the pressure coefficient along the chord, two different

methods have been used; a linearized two-dimensional vortex lattice method in which point vortices placed on the chord line are used to represent the 2-D blade section and, a panel method based on the surface distribution of singularity elements. After the pressure coefficients are determined, a comparison with the local cavitation numbers at each radial location is performed and areas on the blade surface where cavitation is predicted to occur are identified. Color plots of the pressure distribution and the cavitating areas on the propeller blades are produced.

In the following section the two different methods for calculating the pressure distribution are described in detail.

### 5.3 Methods for Calculating Pressure Coefficients

#### 5.3.1 VLM

This method is based on the classical linearized theory for thin 2-D foil sections. A numerical implementation has been presented by Kerwin (2001) and more recently Chung (2007) rewrote the numerical code (VLM) in MATLAB®.

According to the theory of thin wing sections the chordwise velocity distribution is considered to be composed of three separate and independent components as follows:

- The distribution corresponding to the load distribution of the mean line at its ideal angle of attack.
- The distribution corresponding to the velocity distribution over the basic thickness form at zero angle of attack
- The distribution corresponding to the additional load distribution associated with the angle of attack.

After taking into account the contributions from all three components, the complete expression for the non-dimensional velocity on the section surface becomes:

$$\frac{q(x)}{U} = 1 + \frac{u_t(x)}{U} + \frac{u_c(x)}{U} + (\alpha - \alpha_{ideal}) \sqrt{\frac{c-x}{x}} \quad (49)$$

where  $U$  is the incident velocity,  $u_t(x)$  and  $u_c(x)$  the perturbation velocities due to thickness and camber respectively,  $c$  the chordlength and  $x$  the distance from the leading edge. This expression of course cannot predict the local behavior of the flow near the leading edge (notice that for  $x$  equal to zero the surface velocity becomes infinite), something that would have a significant effect on the accuracy of any cavitation inception predictions. In order to solve this problem, a variant of Lighthill's leading edge correction is applied to the total computed velocity prior to computing the pressure coefficient, in the form of a multiplicative factor:

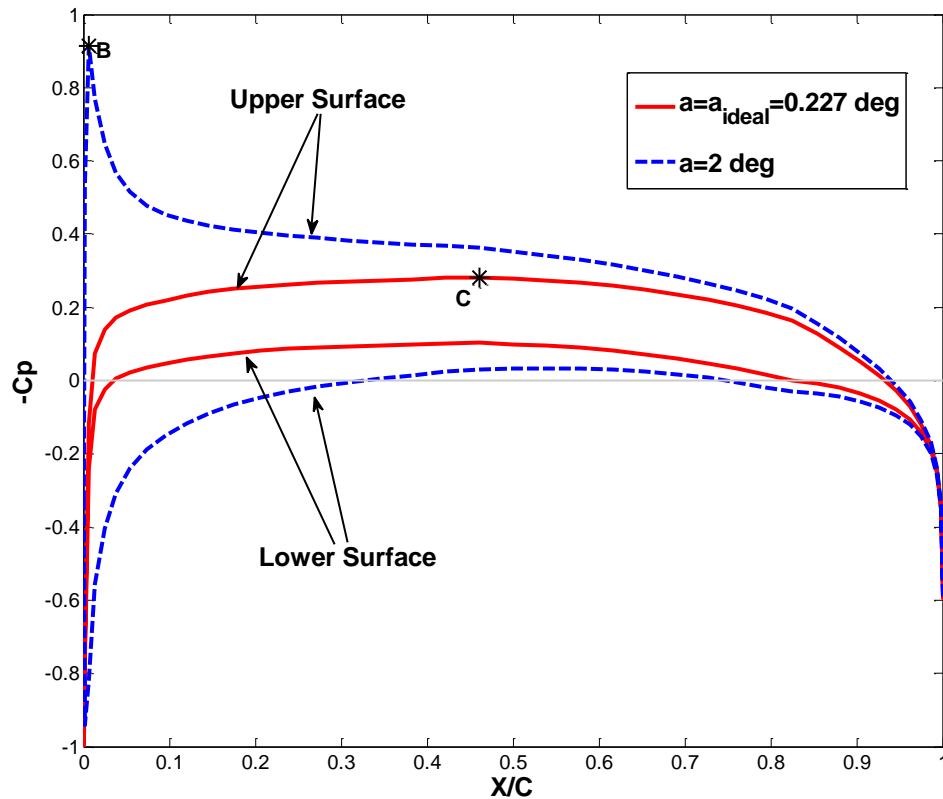
$$\frac{q(x)}{U} = \left[ 1 + \frac{u_t(x)}{U} + \frac{u_c(x)}{U} + (\alpha - \alpha_{ideal}) \sqrt{\frac{c-x}{x}} \right] \left( 1 / \sqrt{1 + \left( \frac{dy}{dx} \right)^2} \right) \quad (50)$$

with  $1 / \sqrt{1 + \left( \frac{dy}{dx} \right)^2} = \sqrt{x / \left( x + \frac{r_l}{2} \right)}$ , where  $r_l$  is the leading edge radius depending on the specific foil profile examined.

The non-dimensional pressure coefficient can then be evaluated along the chord by using the following relationship:

$$C_p(x) = \frac{p - p_\infty}{\frac{1}{2} \rho U^2} = 1 - \left( \frac{q(x)}{U} \right)^2 \quad (51)$$

Given the thickness and the camber distributions of the foil, the respective perturbation velocities can be calculated. If, additionally, the angle of attack that the foil operates at is provided, the previous equations can be used for obtaining the pressure coefficients over the chordlength. In VLM the maximum camber ratio and the angle of attack are entered indirectly, by specifying the value for the ideal lift coefficient and the difference between  $\alpha$  and  $\alpha_{ideal}$ . An example of the output of VLM can be seen in Fig. 5-3. In this case the foil section has a NACA 66 (TMB modified) thickness distribution with  $t_0/c$  equal to 0.08 and a NACA  $\alpha=0.8$  camberline with  $f_0/c$  equal to 0.01.



**Figure 5-3: Pressure distributions obtained with VLM for a NACA 66 (TMB modified) thickness and NACA  $\alpha=0.8$  meanline with a maximum thickness ratio of 0.08 and a maximum camber ratio of 0.01, for two different angles of attack**

It is clear that at the ideal angle of attack the concept of the shock-free operation is verified; there is no sudden pressure drop at the leading edge and the minimum pressure (maximum value of  $-C_p$ ) occurs at point C, near the mid-chord. At an angle of attack of two degrees, a sudden pressure drop is observed near the leading edge (point B in Fig.5-3). Points B and C can also be seen lying on the minimum pressure envelope illustrated in Fig. 5-1.

### 5.3.2 XFOIL

The second method for calculating the pressure coefficient over the surface of a given blade is based on the use of XFOIL, a panel code for viscous/inviscid analysis and mixed inversed design of subcritical airfoils.



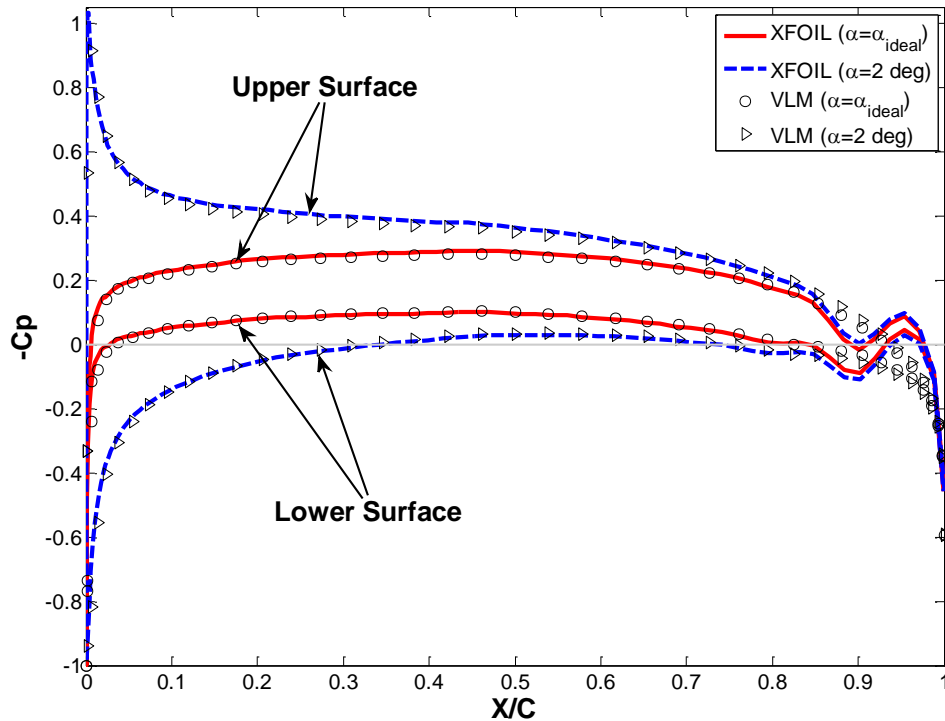
In the inviscid formulation employed in XFOIL, a general two-dimensional inviscid airfoil flow field is constructed by the superposition of a freestream flow, a vortex sheet of strength  $\gamma$  on the airfoil surface, and a source sheet of strength  $\sigma$  on the airfoil surface and the wake. The airfoil contour and wake trajectory are then discretized into flat panels with each foil panel having a linear vorticity distribution. Each foil and wake panel also has a constant source strength. The details of the different formulations and the capabilities of the code are described by Drela (1989).

XFOIL is a menu driven program which requires interactive user input and manipulation, and generates various output plots to allow graphical display and interface by the user. A modified version of XFOIL has been successfully used as an analysis tool for calculating minimum pressure envelopes of blade sections (Peterson 2008). Comparisons with results from analytical expressions for the pressure distribution revealed the increased accuracy of XFOIL over the method developed by Brockett. In that version, all the plot and screen output utilities had been disabled in order to increase calculation speed. Additionally, command line arguments could be read in, rather than direct inputs from the operator via menu options. That allowed the rapid execution of the code by an external program, MATLAB® in particular, such that a future integration of XFOIL into the open-source propeller design and analysis code OpenProp could be easily performed. In the context of the present work, the same modified version has been used in order to calculate the chordwise pressure distribution for the blade sections.

At this point reference to the way that the foil geometry is entered into XFOIL should be made. The geometry can either be loaded or selected from a list of standardized NACA geometries. If the former method is used, it is required that foil geometry is specified by defining the X-Y locations along the foil surface from trailing edge, along the upper surface, around the leading edge, and back to the trailing edge along the lower surface in a counterclockwise sense. Functions that can translate camber and thickness distributions into this kind of format have been developed by Peterson as part of his Master's thesis.

The pressure distributions calculated with XFOIL on the surface of the same foil which was examined in the previous section are presented in Fig. 5-4. Also plotted are the results obtained with VLM. Even though the two methods are fundamentally different, they yield nearly identical predictions for the most part of the foil. The irregularities given by XFOIL, observed near the

trailing edge, can be attributed to the way the section was modeled and the number of panels used. Nevertheless, they are not expected to alter the foil's 'behavior' with respect to the occurrence of cavitation since minimum pressure values will be experienced near the leading edge.



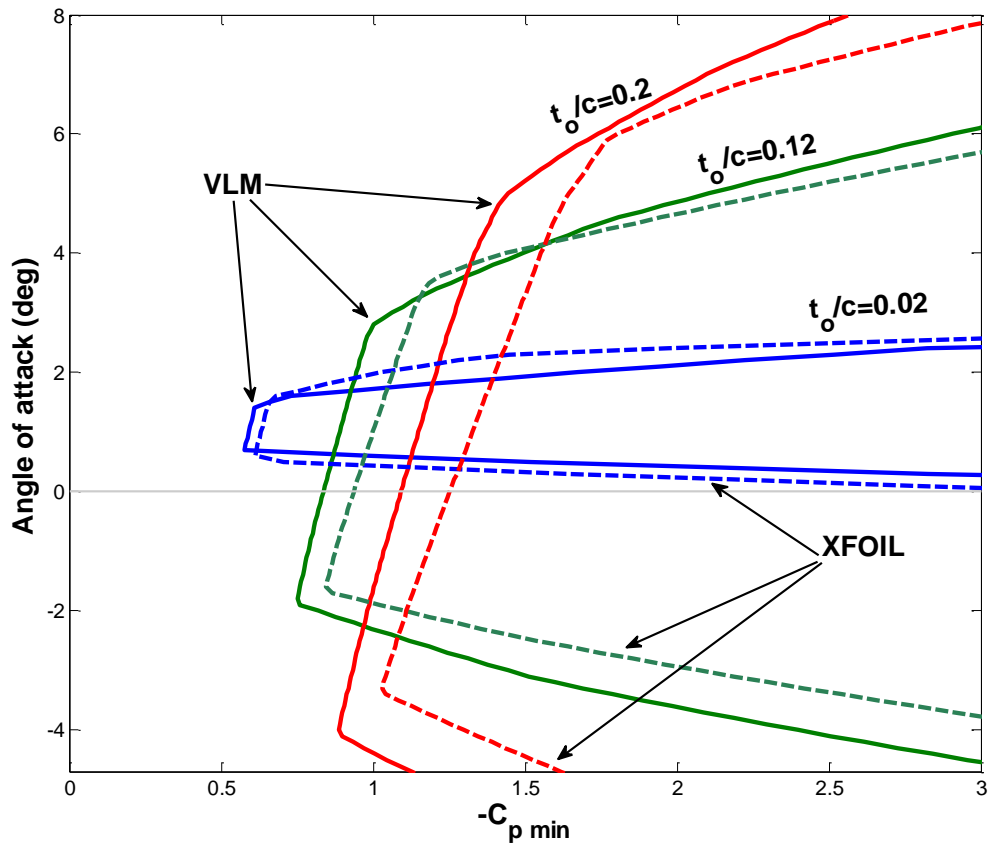
**Figure 5-4: Pressure distributions obtained with XFOIL for a NACA 66 (TMB modified) thickness and NACA  $\alpha=0.8$  meanline with a maximum thickness ratio of 0.08 and a maximum camber ratio of 0.01, for two different angles of attack**

The distribution of pressure coefficient over the blade's surface is obtained in the current implementation by analyzing individual sections at various locations along the blade span. At each radial location the exact foil section geometry is fed into XFOIL, as the lift coefficient and the corresponding maximum camber ratio have been determined from the geometry generation routine described in section 2.5. The required inputs for XFOIL include the maximum thickness and camber ratios and the operating angle of attack (set equal to the ideal as explained earlier).

## 5.4 Comparison between VLM-XFOIL

A comparison between VLM and XFOIL is performed next. Cavitation bucket diagrams for the NACA 66(TMB modified) and meanline NACA  $a=0.8$  section, generated by both methods for foil sections having the same camber distribution but different maximum thickness ratios is given in Fig. 5-5. This type of section is the one used throughout this work and its meanline and thickness distributions can be found in Appendix C.

The cavitation-free envelopes as predicted by the VLM method appear to be wider, especially as the foil's thickness increases. Moreover, XFOIL gives greater maximum pressure drops than VLM (higher values for  $-C_p$ ), which effectively means that the sections are more susceptible to suction side bubble cavitation. In the figure below this is understood by observing the near vertical segments of the envelopes, which correspond to the region of operation when the minimum pressure occurs in the vicinity of the mid-chord and set the limit for suction side bubble cavitation; the segments acquired with VLM are placed to the left of the respective segments for XFOIL implying larger margins against suction side bubble cavitation. Given that the current propeller geometry module places blade sections at their ideal angle of attack, we expect that XFOIL will give predictions for larger cavitating blade areas compared to VLM.



**Figure 5-5: Comparison of Minimum Pressure Envelopes between VLM and XFOIL for sections having a maximum camber ratio of 0.06 but different thickness ratios**

Apart from the noticeable differences observed in Fig. 5-5, VLM and XFOIL require different inputs in order to predict pressure distributions. As already mentioned XFOIL needs the exact foil geometry to be entered in the form of two dimensional coordinates. On the other hand, VLM requires that the foil is described by its meanline and thickness distribution type, with the maximum thickness ratio  $t_0/c$  being entered directly and the maximum camber ratio expressed by the corresponding value of the ideal lift coefficient. As far as the operating condition is concerned, the angle of attack is an input for XFOIL, while the deviation from the ideal angle of attack is used in VLM.

# Chapter 6

## Cavitation Analysis Results

### 6.1 Steady Cavitation in Uniform Wake

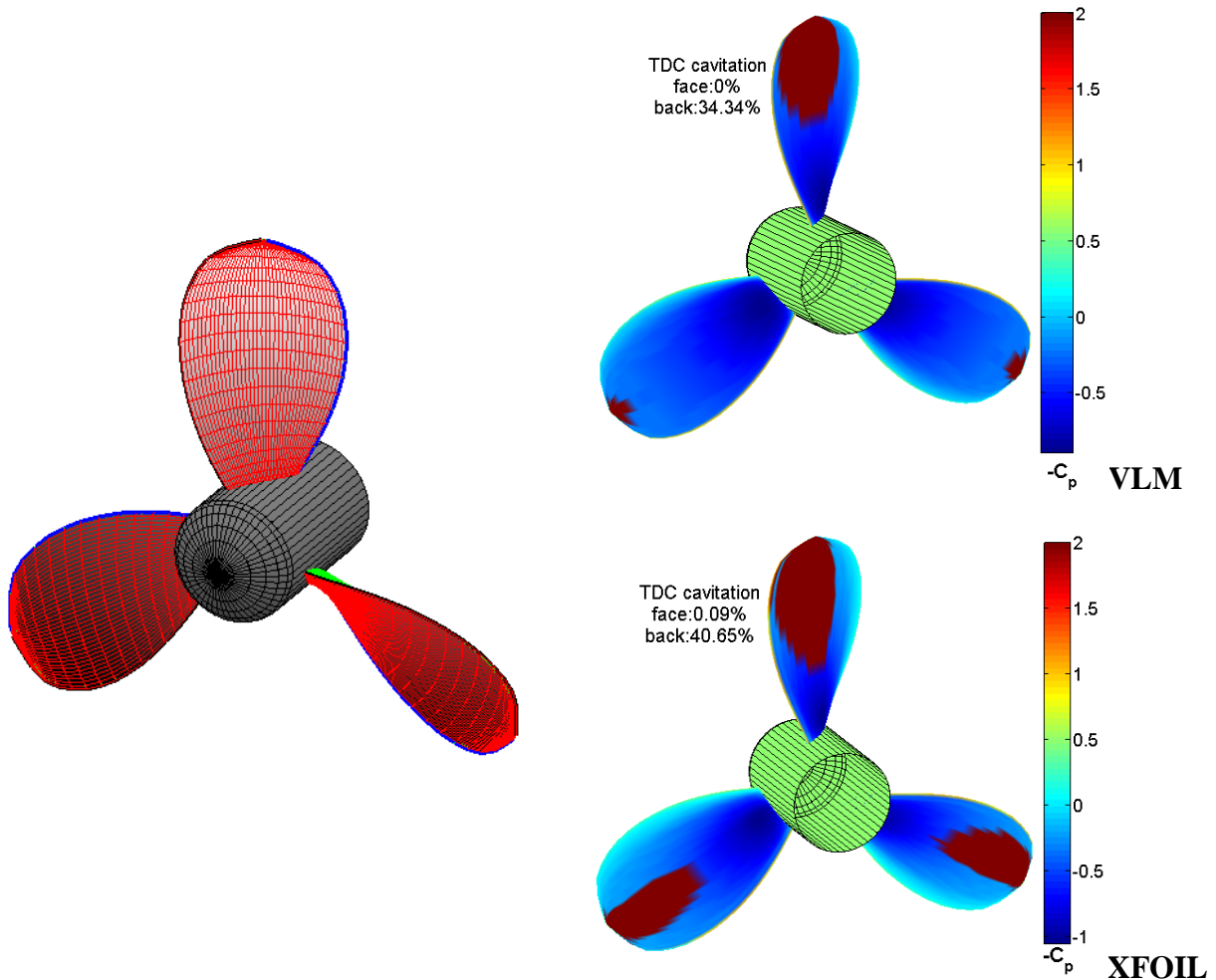
When the propeller is assumed to operate in a uniform axial wake (open water conditions) the propeller blades experience steady flow conditions as they rotate about the propeller axis. The design of the blades is then performed by requiring a shock free entry, a condition which is satisfied around the propeller disk given that the flow is considered to be independent of the angular position. The occurrence of cavitation in this case is exclusively due to high blade loadings and can be observed in the mid-chord neighborhood of the suction side, where the pressure attains its minimum value. On the other hand, no cavitation occurs near the leading edge, since the pressure distribution on the blade sections does not exhibit local peaks near this area. Fig. 6-1 shows the cavitation patterns on the blades of a free-running conventional propeller, the characteristics of which are given in Table 6-1.

**Table 6-1: SR propeller characteristics used in steady cavitation analysis**

Number of blades	$Z=3$
Blade Diameter	2.5 m
Hub Diameter	0.5 m
Thrust Loading	$C_t=0.932$
Advance coefficient	$J_s=0.738$
Shaft Centerline Depth	$H=2$ m

The particulars of the propeller have been selected such that cavitation would occur and its patterns on the blade's surface could be observed. Comparing the results between VLM and XFOIL, the former method yields smaller cavitating areas (34.3 % compared to 40.6 %). The percentage that appears on the 3D color plots corresponds to the blade positioned at the 12

o'clock location. Areas where the local pressure is lower than the vapor pressure are marked in red, whereas the color for the rest of the surface corresponds to the local pressure coefficient.



**Figure 6-1: Geometry and steady cavitation patterns of a conventional propeller**

The difference between the predictions given by VLM and XFOIL can be attributed to the difference in the width of the corresponding minimum pressure envelopes. However, both patterns illustrate bubble cavitation in the mid-chord region of the suction side of the blade sections, as expected. Increasing the chord lengths of the blade sections would help mitigate the phenomenon by decreasing the loading of the blades. This improvement in the cavitation performance though would come at the cost of lower hydrodynamic efficiency due to increased viscous losses.

A contra-rotating six-bladed propeller set required to generate the same thrust and operating under the same conditions (see Table 6-1) was also examined using VLM. In this case, though, the total expanded area ratio is larger by 33% compared to the conventional propeller. Figure 6-2 shows that cavitation will be significantly decreased on the forward propeller. The fact that the aft propeller blades will not develop cavitation at all is explained by the reduced circulation values compared to the forward one, which leads to aft blade sections having lower lift coefficients.

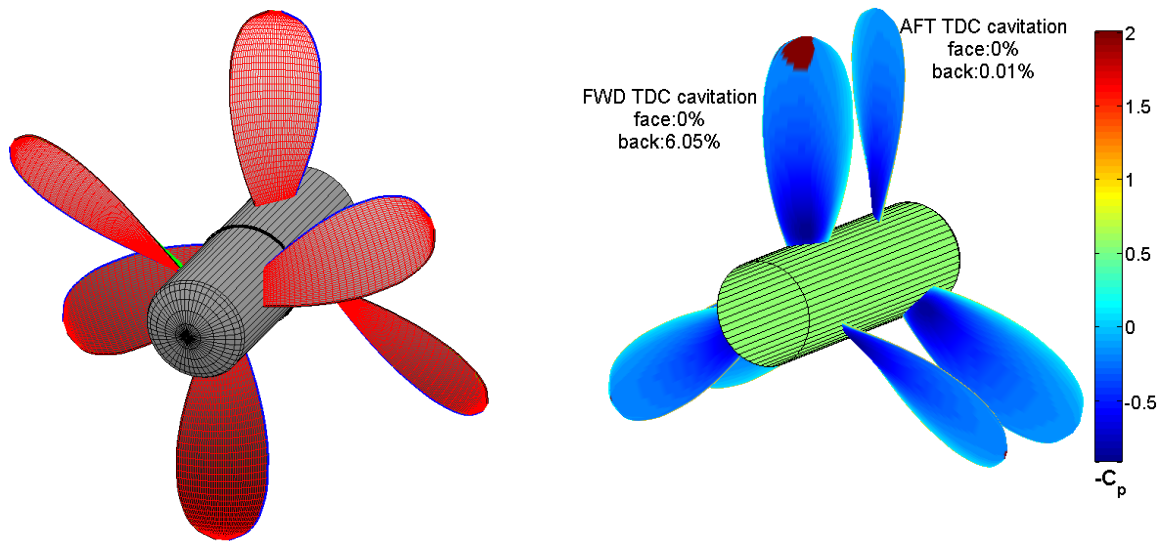


Figure 6-2: Geometry and steady cavitation pattern of a CRP set using VLM

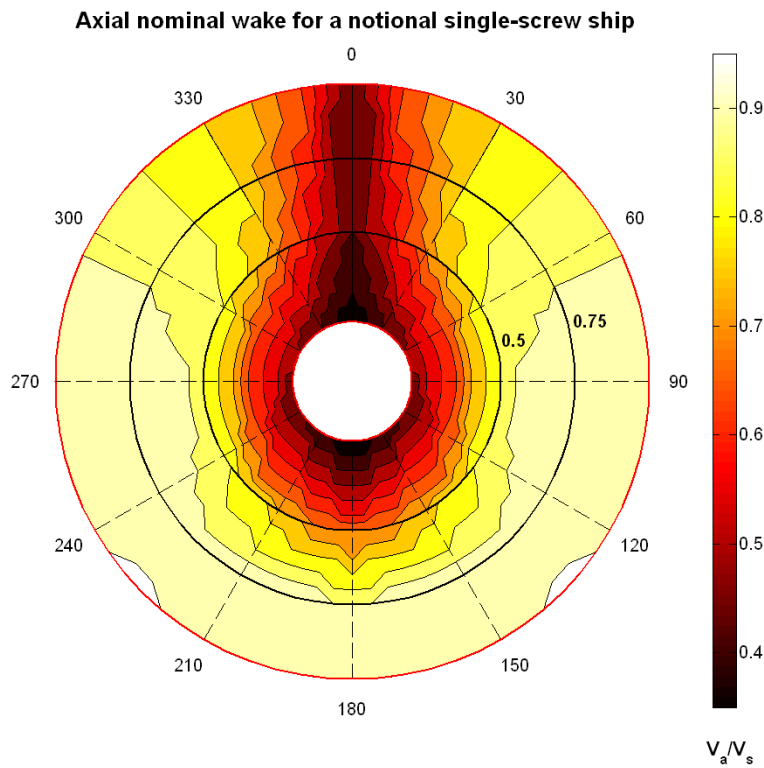
Aside from the improvement in its cavitation performance, the CRP configuration also exhibits higher efficiency, despite the increased expanded area ratio. An efficiency gain of 6 percent over the conventional propeller is achieved.

## 6.2 Unsteady Cavitation

In the previous section operation under open-water conditions, in which the propellers advanced into undisturbed water, was assumed. In practice, however, propellers are placed behind a ship's hull where the inflow conditions are far from uniform. Flow measurements at the actual propeller locations yield wakes that vary both temporally and spatially. Even if the temporal variation is ignored, a more realistic representation of the wake field could be obtained by considering that it is a function of the radius and the angular position in the plane of the propeller disk. The blades

sections at any given radius would then pass through regions of very different wake concentrations giving rise to vibratory forces and unsteady cavitation even when trust loadings are relatively low.

Analysis of the cavitation performance of wake-adapted propellers operating in a circumferentially varying axial wake field is performed in this section. For this purpose an axial nominal wake for a notional single-screw ship has been developed (Fig. 6-3). The iso-velocity contours indicate that the wake is symmetrical about the centerline, most intense over the upper part of the disk and much smaller over the outer lower quadrants. Of course the effective wake is expected to be different than the nominal one, but assuming that the interaction component between the hull and the propeller is small, use of the nominal wake will not affect the results considerably.



**Figure 6-3: Circumferential axial wake distribution**

The variation in inflow velocity, described by the iso-velocity contours, results in a periodic change in the angle of attack. Alternatively, we can suppose that we have a ‘standing wave’



through which the blade sections cyclically ‘orbit’. The mean wake at each radial location along the propeller radius will then be given by:

$$(1 - w)_r = \left( \frac{V_a}{V_s} \right)_r = \int_0^{2\pi} \frac{(1 - w)_\theta d\theta}{2\pi} \quad (52)$$

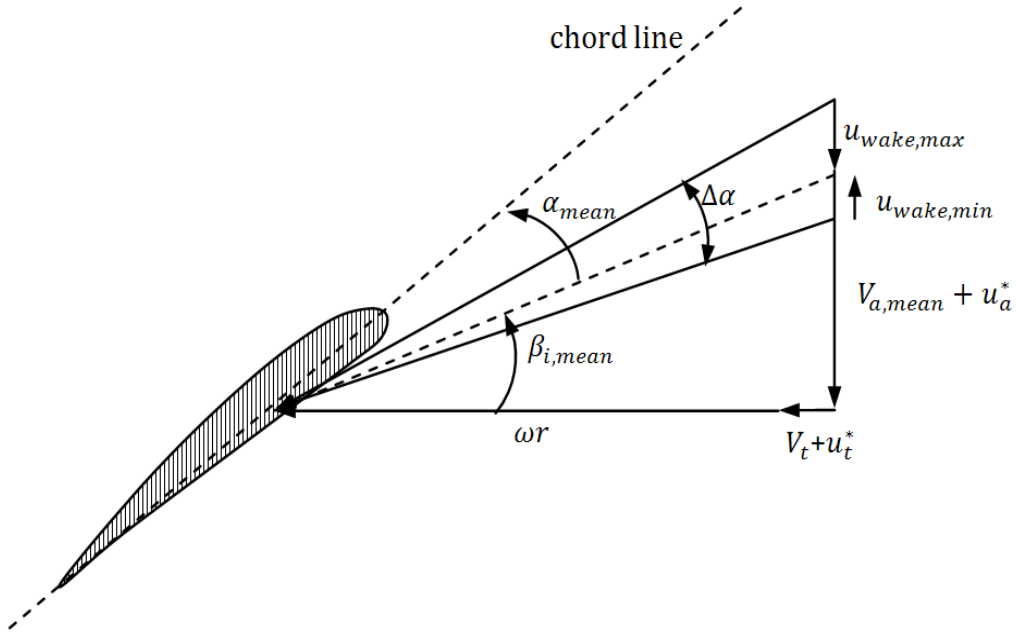
where index  $r$  ranges between the hub radius  $r_h$  and the tip radius  $R$ .

Since the propeller is designed to perform in the mean wake at each radial location (wake-adapted propeller), calculated using eq. 52, we expect that the hydrodynamic pitch angles  $\beta_i$  and the corresponding angles of attack  $\alpha$  for each blade section will vary around mean values  $\beta_{i,mean}$  and  $\alpha_{mean}$ . Fig. 6-4 shows the variation  $\Delta\alpha$  in the angle of attack of a blade section when the axial velocity component reaches its extreme values  $V_{a,min}$ , and  $V_{a,max}$ . In the figure,  $u_{wake,min}$  and  $u_{wake,max}$  correspond to the varying part of the axial velocity relative to the mean value, given by:

$$u_{wake}(r, \theta) = V_{a,mean}(r) - V_a(r, \theta) \quad (53)$$

The induced velocities  $u_a^*$  and  $u_t^*$  are included in the calculation for the angle of attack variation, however they are considered to be independent of the angular position. Thus, the self induced velocities (and the interaction velocities in the case of CRPs) are not expected to adapt as fast as the angles of attack change. This assumption is reasonable considering the level of accuracy we expect from a lifting line method when it comes to predicting unsteady effects.

Apart from the axial, the tangential wake distribution  $V_t(r, \theta)$  is also expected to be non-uniform. However, its contribution to the circumferential variation of the angle of attack is generally less than the axial’s component, the reason being that it is small compared to the rotational speed  $\omega r$ . Therefore it has not been taken into account in the following cavitation analysis.



**Figure 6-4: Variation in the angle of attack due to non-uniform axial wake**

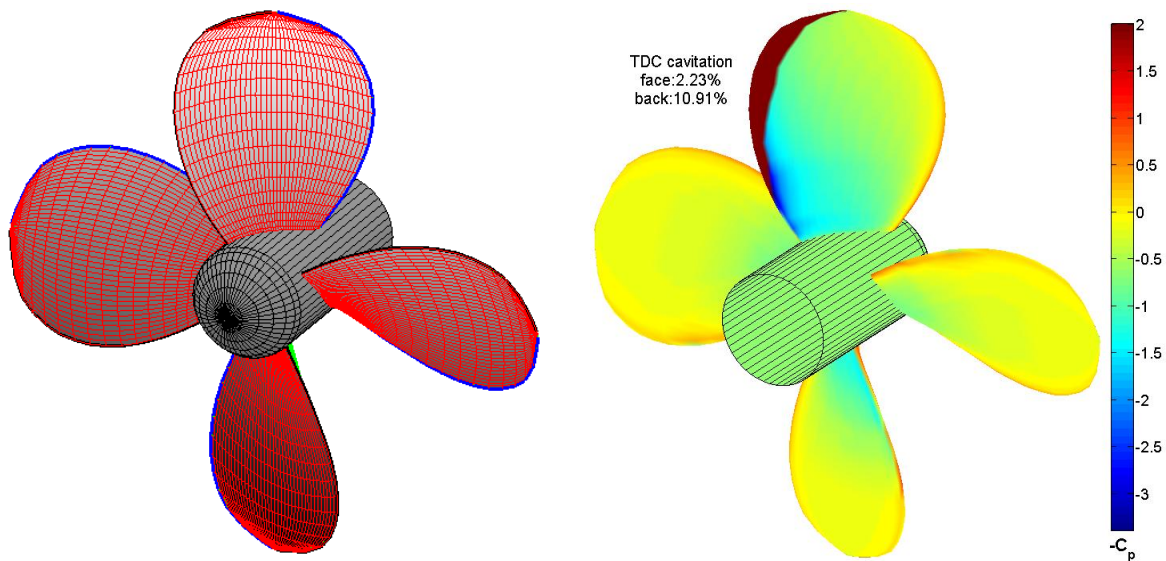
Fig. 6-5 shows the geometry and the cavitation analysis results for a conventional single propeller operating in the non-uniform axial wake presented earlier. The particulars of the propeller are given in Table 6-2. The advance coefficient is the one yielding the maximum propeller efficiency for the specific thrust loading.

**Table 6-2: SR propeller characteristics used in unsteady cavitation analysis**

Number of blades	Z=4
Blade Diameter	2 m
Hub Diameter	0.4 m
Thrust Loading	Ct=0.69
Advance coefficient	Js=0.89
Shaft Centerline Depth	H=2 m

It is evident from Fig. 6-5 that cavitation occurs only on the blade oriented at the 12 o'clock position. The axial velocity in that region is lower than the mean axial velocity value  $U_{a,mean}$ , for all radial locations, and the blade sections operate at extremely high (and positive) angles of

attack. Moreover, the cavitation numbers in that sector are lower due to the decreased submergence depths. Thus, areas where back sheet cavitation occurs near the leading edge on the suction side are formed. Comparing this pattern to the one developed due to steady cavitation (Fig. 6-1), we conclude that not only highly loaded propellers are prone to cavitation inception, but also adverse flow conditions, such as operation in a highly non-uniform wake field, can lead to the onset of cavitation. In order to decrease the severity of the phenomenon, thicker blade sections should be adopted, given the cavitation sensitivity of thin sections.



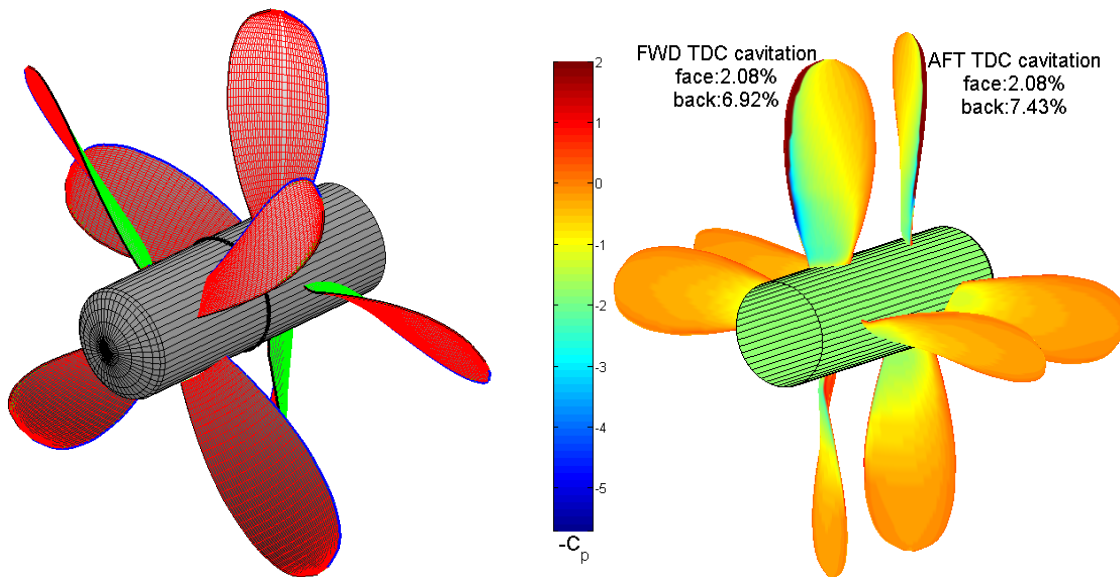
**Figure 6-5: Geometry and unsteady cavitation patterns of a conventional propeller using VLM**

Studying the cavitation performance of the propeller blades involved the following steps; after calculating the radial circulation distribution of the wake-adapted propeller and determining the required geometric pitch angles of its blades, the variation in the operating angles of attack for each section along the blade span was determined in a manner similar to the one presented in Fig. 9. The pressure distributions over the surface of the blades given the local values of the ideal lift coefficient and the angle of attack,  $C_{li}(r)$  and  $a(r, \theta)$  respectively (varying along the blade span and around the disk periphery) were then obtained using VLM. Additionally, the cavitation index variation in one revolution due to the variation of the water depth was calculated. In the final step, the areas where cavitation was expected to occur were designated by comparing the pressure coefficient to the cavitation index. A color plot of the local pressure coefficients and the

This page intentionally left blank

cavitating areas was the final output. The numerical implementation of the procedure described above can be found in *Cavitation.m* function in Appendix B.

A contra-rotating propeller set was subjected into the same non-axisymmetric wake field. The underlying assumption was that the wake is the same for the forward and the aft components of the CRP, even though in reality, at the aft propeller plane the axial velocity field will probably be more uniform. The geometry and the cavitation patterns on the blades of the forward and aft screws are shown in Fig. 6-6.



**Figure 6-6: Geometry and Unsteady Cavitation patterns of a CRP set using VLM**

The CRP set was required to develop the same thrust and had the same diameter and expanded area ratio as the single propeller. However, a higher advance coefficient was assumed which effectively means that each component of the set rotated slower. This selection is supported by the results of the parametric design study presented in the first part of this thesis. According to those, the optimal advance coefficient for CRPs in terms of efficiency is lower than the one for single rotation propellers. Thus, the small improvement in the cavitation behavior of the CRP set observed in Fig. 6-6 can be attributed to the higher values of the cavitation index, as a consequence of the lower rotational speeds. Nevertheless, this improvement alone could not justify the selection of a contra-rotating propeller over a conventional one, given the increased mechanical complexity and cost of the former. It is the combined improvement in hydrodynamic efficiency and cavitation performance that makes the CRP set a very attractive choice. Table 6-3

presents the results for the two propulsor variants designed for operation in the non-axisymmetric axial wake field.

**Table 6-3: Comparison between SRP – CRP design for operation in a non-uniform axial wake**

	<b>SRP</b>	<b>CRP</b>
Advance coefficient (Js)	0.89	1.36
Rotational speed (N)	168.5 RPM	110 RPM
Suction Side(s) cavitation	10.9 %	7.2 %
Efficiency ( $\eta$ )	66.2 %	74.4 %

Apart from the smaller cavitation patterns, the CRP pair achieves an improvement in the hydrodynamic efficiency of 12.4% over the conventional single propeller.

### **6.3 Remarks on Cavitation Performance of CRP**

From the preceding sections it became obvious that contra-rotating propellers are potentially characterized in general by improved cavitation performance. The reasons for this improvement can be summarized as follows:

1. Different optimal working conditions (lower rotational speeds)

It is deceptive to design a CRP set for the same working condition that corresponds to the optimum efficiency for an equivalent single propeller. The major advantage of contra-rotating propellers, the cancellation of rotational losses, can only be effective when there is room for reduction of those losses. As explained and shown in chapter 3, as the advance coefficient increases, rotational losses dominate over viscous losses and the efficiency benefit of CRPs over conventional propellers increases.

Thus, contra-rotating propellers are designed to rotate, in principle, at slower rates than conventional single ones (see Fig. 6-7). The cavitation numbers are therefore expected to be larger which means that CRPs are less prone to the occurrence of cavitation.

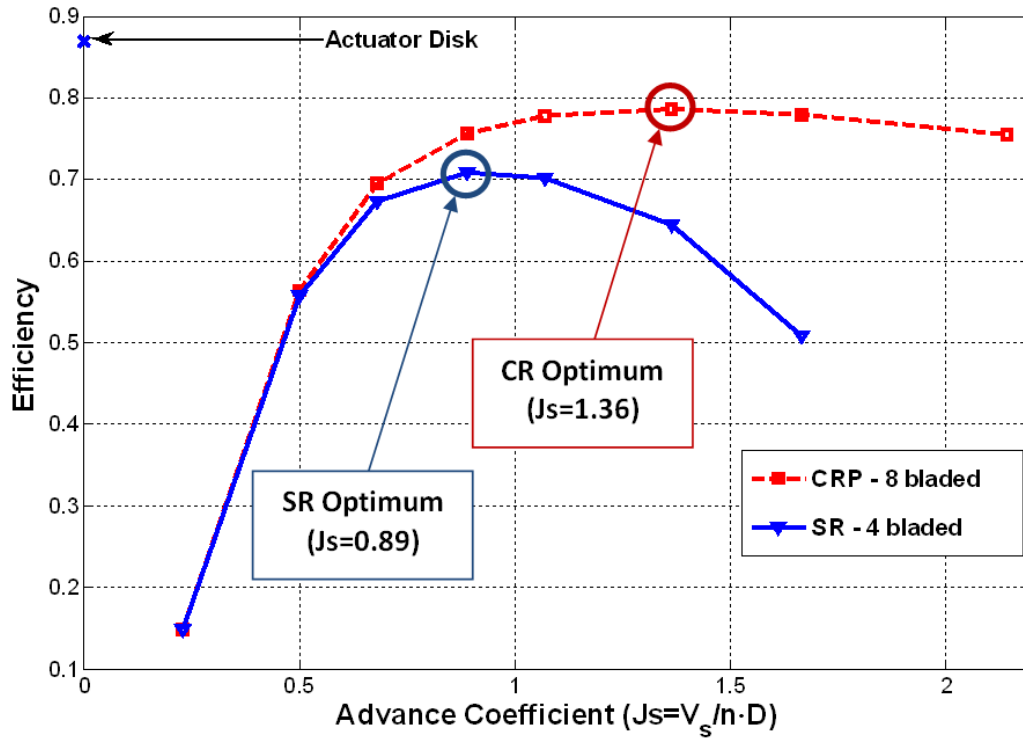


Figure 6-7: Comparison between SRP and CRP efficiency curves at the design point for a thrust loading of 0.69

## 2. Tip-Region Unloading

Optimum circulation distributions for CRPs (particularly for the aft screw of the set) indicate an inward shift towards the blade root. Consequently, the required lift coefficients for the outer blade sections will be lower compared to single propellers, provided that the respective expanded area ratios will be comparable. As a result, the minimum values of the pressure coefficients ( $-C_{Pmin}$ ) will be lower, thus decreasing or even eliminating cavitation. A comparison between SRP and CRP with respect to the blade loading is shown in Fig. 6-8. Principal dimensions, expanded area ratios and required thrusts are the same for both propulsors.

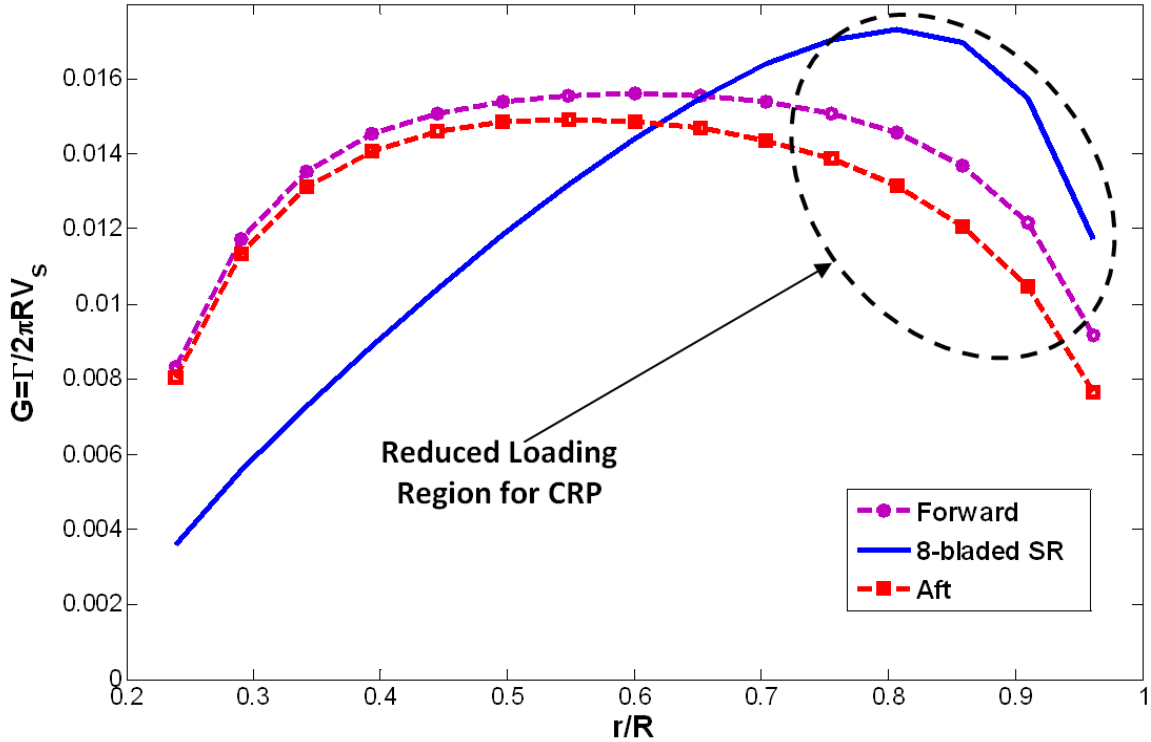


Figure 6-8: Comparison of tip loading between SRP - CRP for a thrust loading of 0.69

### 3. Greater flexibility in ‘controlling’ cavitation

The only way to reduce the effects of cavitation, assuming that the blades’ thickness and camber distribution remain the same, is to increase the blade area ratio and reduce the loading per blade area. The direct disadvantage would be the increase of frictional losses due to the larger blade area. When contra-rotating propellers are used, we can afford to sacrifice some of the propeller’s efficiency towards improving its cavitation performance.

Another parameter that delays the occurrence of cavitation on the aft propeller of a CRP set is its smaller diameter compared to the forward one, which leads to larger cavitation numbers. The difference in propeller diameters is necessary in order to prevent the forward propeller’s cavitating tip vortex from hitting the aft propeller blades. Moreover, in CRP configurations, erosion on the rudder’s surface, caused by the cavitating hub vortex, can be avoided.



# **Chapter 7**

## **Conclusions and Future Work**

With regard to CRP design, two methods for calculating optimum loading distributions have been implemented numerically within the assumptions of lifting line theory. It is proposed that the ‘Coupled’ method should be adopted since it yields slightly more efficient ‘products’ and can handle ‘hub-loaded’ cases. Additionally, two methods for obtaining pressure distributions and predicting the cavitation patterns on the propeller surface have been studied. The selection of the specific one depends on the needs of the propeller designer. The major advantage of XFOIL over the VLM code is that unconventional foil geometries can be analyzed by the former.

The numerical implementation of the above methods and their integration into the ‘OpenProp’ propeller design suite will significantly extend its range of applications. This will be the first step towards the development of a preliminary design tool with multiple blade-row design capabilities. A summary of the findings and the conclusions of this work with respect to the performance and the cavitation behavior of CRP is given next.

### **7.1 Conclusions**

First, the superiority of contra-rotating propellers over conventional single propulsors in terms of open-water efficiency is confirmed. Reduction of rotational losses, even though these are small compared to the axial losses, results in significant improvement in efficiency, particularly for high thrust loadings.

Second, it is expected that CRP will be less prone to the inception of cavitation, something that further fosters their use in highly loaded propeller applications. Furthermore, reduction in pressure pulses can be achieved by adapting CRP sets which inherently feature smaller blade diameters (larger hull-tip clearances) and lower rotational speeds.

In spite of the forementioned attractive hydrodynamic characteristics of CRP, these propulsor configurations have not been widely adopted to date, the main reason being the increased mechanical complexity of the supporting drive train and the associated high installation and maintenance costs. The advent of electric propulsion and the improvement in the power density of propulsion motors will most certainly increase the application of CRP by removing the need for complex gear boxes, bearing and sealing systems. Therefore, propeller design codes capable of handling CRP pairs, as the one presented in this thesis, will receive increased attention during the ship design process.

## **7.2 Recommendations for future work**

Throughout this thesis contraction of the slipstream was not considered. Even though this might not have a significant effect on the design of single propellers, it plays an important role for CRR where accurate predictions for the interaction velocity fields are required. The exclusion of the wake contraction makes the computation of the interaction velocities much easier. If this effect were to be taken into account, tracking of contracting streamlines would be necessary, something that could be approached in the following ways:

1. Assuming that the actual wake contraction occurs over a short distance, compared to the propeller diameter, a better approximation to the wake geometry would be to model it as purely helical, but contracted to some new radius. Empirical contraction ratios based on water tunnel measurements indicate that this ratio is close to 0.83(J.E. Kerwin & Lee 1978)
2. Instead of using empirical corrections, analytical solutions for the contraction ratios and the trajectories of the contracting streamlines could be implemented. Thus, the effects of the actual clearance between the propulsor components could be predicted accurately. Closed form solutions for the velocities and the streamfunctions for a wide range of thrust loadings have been given by Conway(1995; 1998).

Moreover, in the current methodology the selection of the chord lengths is not part of the propeller optimization process, but it is a direct user input. The optimization procedure could be greatly improved by taking into account the effects of the chord length on the viscous drag, the

cavitation performance and the structural strength of the blades, such that the selection of chord lengths could be made part of the circulation optimization scheme.

All of the above points requiring further investigation refer to the design problem, where the geometric pitch and the loading of the blades are of interest. The inverse design problem, in which the thrust and the torque coefficients at off design points are sought, given the blade geometry, should also be addressed. A procedure for obtaining the performance curves for single propellers using lifting line theory has been developed by Epps (2010) and could be extended to treat CRP as well. Validation using experimental data and inclusion of lifting surface corrections would result in a complete and reliable preliminary design and analysis tool.

This page intentionally left blank

## References

- Abbott, I. & vonDoenhoff, A., 1959. *Theory of wing sections, including a summary of airfoil data*, New York: Dover Publications.
- Betz, A., 1919. Schraubenpropeller mit geringstem Energieverlust. *Nachrichten der K. Gesellschaft der Wissenschaften zu Göttingen, Math-Phys.*, 193-217.
- Breslin, J.P. & Andersen, P., 1994. *Hydrodynamics of Ship Propellers*, Cambridge University Press.
- Brockett, T., 1966. *Minimum pressure envelopes for modified NACA-66 sections with NACA  $a=0.8$  camber and buships type I and type II sections*, Washington, DC, United States: David Taylor Model Basin, Navy Department.
- Caponnetto, M., Rolla, P. & Porro, M., 1997. A New Propeller Design Method for Fast Planing Boat Application. In FAST '97:Fourth International Conference on Fast Sea Transportation, July 21-23, 1997. Sydney, Australia: Baired Publications, pp. 175-185.
- Caponnetto, M., 2000. Optimisation and Design of Contra-Rotating Propellers. In SNAME Propellers/Shafting Symposium. Virginia Beach, Virginia, pp. 3-1 to 3-9.
- Caster, E.B. & LaFone, T.A., 1975. *A Computer Program for the Preliminary Design of Contrarotating Propellers.*, Washington, D.C.: David W. Taylor Naval Ship Research and Development Center.
- Chung, H., 2007. *An Enhanced Propeller Design Program Based on Propeller Vortex Lattice Lifting Line Theory*. S.M. Thesis. Massachusetts Institute of Technology.
- Coney, W., 1989. *A method for the design of a class of optimum marine propulsors*. Ph.D. Thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Conway, J., 1995. Analytical solutions for the actuator disk with variable radial distribution of load. *Journal of Fluid Mechanics*, 297, 327-55.
- Conway, J., 1998. Exact actuator disk solutions for non-uniform heavy loading and slipstream contraction. *Journal of Fluid Mechanics*, 365, 235-67.
- Cox, B. & Reed, A., 1988. Contrarotating Propellers-Design Theory and Application. In *Propellers '88 Symposium, September 20 and 21, 1988*. Virginia Beach, Virginia: Jersey City N.J.:Society of Naval Architects and Marine Engineers, pp. 15.1-29.
- Drela, M., 1989. XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils. In *Proceedings of the Conference, Notre Dame, Indiana, USA, 5-7 June 1989*. Low

- Reynolds number aerodynamics. Springer-Verlag.
- Epps, B. et al., 2009. OpenProp: An open-source parametric design and analysis tool for propellers. In *Grand Challenges in Modeling & Simulation Conference (GCMS'09)*. Instabul, Turkey: Society for Modeling and Simulation International.
- Epps, B.P., 2010. *An impulse framework for hydrodynamic force analysis: fish propulsion, water entry of spheres, and marine propellers*. Ph.D. Thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Glover, E., 1987. Propulsive Devices for Improved Propulsive Efficiency. In *Trans. Institute of Marine Engineers*. London: Published for the Institute of Marine Engineers by Marine Management (Holdings) Ltd., pp. 23-31.
- Hough, G. & Ordway, D., 1965. Generalized actuator disk. *Developments in Theoretical and Applied Mechanics*, 2, 317-336.
- Hsin, C., 1987. *Efficient Computational Methods for Multi-Component Lifting Line Calculations*. S.M. Thesis.
- Kerwin, J., 2001. "Hydrofoils and Propellers". In *Course 2.23 Lecture Notes*. Cambridge, MA: Massachusetts Institute of Technology.
- Kerwin, J., Coney, W. & Hsin, C., 1986. Optimum Circulation Distributions for Single and Multi-Component Propulsors. In *Twenty-First American Towing Tank Conference (ATTC)*. Washington, D.C.: National Academy Press, pp. 53-62.
- Kerwin, J. & Lee, C., 1978. Prediction of Steady and Unsteady Marine Propeller Performance by Numerical Lifting-Surface Theory. In *Trans. SNAME, Vol.86*. pp. 218-253.
- Kerwin, J. & Leopold, R., 1964. A Design Theory for Subcavitating Propellers. In *Trans. SNAME, Vol.72*. pp. 294-335.
- Kerwin, J.E., 1986. MARINE PROPELLERS. In *Annual Review of Fluid Mechanics*. Palo Alto, CA, USA: Annual Reviews Inc, pp. 367-403.
- Kimball, R., Epps, B. & Stanway, J., 2008. OpenProp MATLAB code. <http://openprop.mit.edu>.
- Lerbs, E., 1955. *Contra-Rotating Optimum Propellers Operating in a Radially Non-Uniform Wake*, Ft. Belvoir: Defense Technical Information Center.
- Lerbs, E., 1952. Moderately loaded propellers with finite number of blades and arbitrary distribution of circulation. In *Trans. SNAME, Vol.60*. New York, N.Y.: SNAME, pp. 73-123.
- Morgan, W., 1960. The Design of Counterrotating Propellers Using Lerbs' Theory. In *Trans.*

- SNAME, Vol.68*. New York, NY: SNAME, pp. 6-38.
- Morgan, W., Silovic, V. & Denny, S., 1968. Propeller lifting-surface corrections. In *Trans. SNAME, Vol.76*. New York, N.Y.: SNAME, pp. 309-347.
- Morgan, W. & Wrench, J., 1965. Some Computational Aspects of Propeller Design. In *Methods in Computational Physics*. New York: Academic Press, pp. 301-31.
- Peterson, J., 2008. *Minimum pressure envelope cavitation analysis using two-dimensional panel method*. S.M. Thesis. Massachusetts Institute of Technology.
- Sasaki, N. et al., 1998. Design system for optimum contra-rotating propellers. *Journal of Marine Science and Technology*, 3(1), 3-21.
- Wang, M., 1985. *Hub effects in propeller design and analysis*. Ph.D. Thesis. Cambridge, MA: Massachusetts Institute of Technology.
- Wrench, J., 1957. *The Calculation of Propeller Induction Factors*, Washington, DC, United States: David Taylor Model Basin, Navy Department.

This page intentionally left blank



## Appendices

### Appendix A: CRP Variational Optimization Equations

Upon expansion of eq.12 with eqs 15~20, we obtain the following two equations:

$$\begin{aligned}
\frac{\partial H}{\partial \Gamma_1(i)} = 0 \Rightarrow & \sum_{m=1}^{M_1} \Gamma_1(m) \{ (\omega_1 + q\tilde{\lambda}_Q) \rho Z_1 [\bar{u}_{a_{1,1}}^*(i, m) r_1(i) \Delta r_1(i) \\
& + \bar{u}_{a_{1,1}}^*(m, i) r_1(m) \Delta r_1(m)] \\
& + \tilde{\lambda}_T \rho Z_1 [\bar{u}_{t_{1,1}}^*(i, m) \Delta r_1(i) + \bar{u}_{t_{1,1}}^*(m, i) \Delta r_1(m)] \} \\
& + \sum_{m=1}^{M_2} \Gamma_2(m) \{ [\omega_1 + q\tilde{\lambda}_Q] \rho Z_1 \bar{u}_{a_{1,2}}^*(i, m) r_1(i) \Delta r_1(i) \\
& + [\omega_2 - \tilde{\lambda}_Q] \rho Z_2 \bar{u}_{a_{2,1}}^*(m, i) r_1(m) \Delta r_1(m) + \tilde{\lambda}_T \rho Z_1 \bar{u}_{t_{1,2}}^*(i, m) \Delta r_1(i) \\
& + \tilde{\lambda}_T \rho Z_2 \bar{u}_{t_{2,1}}^*(m, i) \Delta r_2(m) \} \\
& = -\rho Z_1 \{ [\omega_1 V_{a_1}(i) + \lambda_Q q V_{a_1}(i)] r_1(i) \\
& + \lambda_T [V_{t_1}(i) + \omega_1 r_1(i)] \} \Delta r_1(i),
\end{aligned} \tag{A-1}$$

for  $i = 1, \dots, M_1$

and

$$\begin{aligned}
\frac{\partial H}{\partial \Gamma_2(i)} = 0 \Rightarrow & \sum_{m=1}^{M_1} \Gamma_1(m) \{ [\omega_1 + q\tilde{\lambda}_Q] \rho Z_1 \bar{u}_{a_{1,2}}^*(m, i) r_1(m) \Delta r_1(m) \\
& + [\omega_2 - \tilde{\lambda}_Q] \rho Z_2 \bar{u}_{a_{2,1}}^*(i, m) r_2(i) \Delta r_2(i) + \tilde{\lambda}_T \rho Z_1 \bar{u}_{t_{1,2}}^*(m, i) \Delta r_1(m) \\
& + \tilde{\lambda}_T \rho Z_2 \bar{u}_{t_{2,1}}^*(i, m) \Delta r_2(i) \} \\
& + \sum_{m=1}^{M_2} \Gamma_2(m) \{ (\omega_2 - \tilde{\lambda}_Q) \rho Z_2 [\bar{u}_{a_{2,2}}^*(i, m) r_2(i) \Delta r_2(i) \\
& + \bar{u}_{a_{2,2}}^*(m, i) r_2(m) \Delta r_2(m)] \\
& + \tilde{\lambda}_T \rho Z_2 [\bar{u}_{t_{2,2}}^*(i, m) \Delta r_2(i) + \bar{u}_{t_{2,2}}^*(m, i) \Delta r_2(m)] \} \\
& = -\rho Z_2 \{ [\omega_2 V_{a_2}(i) - \lambda_Q V_{a_2}(i)] r_2(i) \\
& + \lambda_T [V_{t_2}(i) + \omega_2 r_2(i)] \} \Delta r_2(i),
\end{aligned} \tag{A-2}$$

for  $i = 1, \dots, M_2$

Here,  $\widetilde{\lambda}_Q$  and  $\widetilde{\lambda}_T$  represent the Lagrange multipliers which were frozen during each iteration for the solution of the inner problem (convergence of the circulation values). On the other hand,  $\lambda_Q$  and  $\lambda_T$  represent the values contributing to the system of equations.

Eqs 13~14 are also part of the linear system and expansion with eqs 15~20 yields :

$$\begin{aligned}
T_r + T_{V_1} + T_{V_2} = & \rho Z_1 \sum_{m=1}^{M_1} [V_{t_1}(m) + \omega_1 r_1(m) + u_{t_1}^*(m)] \Gamma_1(m) \Delta r_1 \\
& + \rho Z_2 \sum_{m=1}^{M_2} [V_{t_2}(m) + \omega_2 r_2(m) + u_{t_2}^*(m)] \Gamma_2(m) \Delta r_2
\end{aligned} \tag{A-3}$$

and

$$\begin{aligned}
Q_{V_2} - qQ_{V_1} = & \rho Z_1 \sum_{m=1}^{M_1} [V_{a_1}(m) + u_{a_1}^*(m)] \Gamma_1(m) r_1(m) \Delta r_1 \\
& - \rho Z_2 \sum_{m=1}^{M_2} [V_{a_2}(m) + u_{a_2}^*(m)] \Gamma_2(m) r_2(m) \Delta r_2
\end{aligned} \tag{A-4}$$

The complete linear system of equations in matrix form is given below:

$$\begin{bmatrix}
A(1,1) & \cdots & A(1, M_1 + M_2 + 2) \\
\vdots & \ddots & \vdots \\
A(M_1 + M_2 + 2, 1) & \cdots & A(M_1 + M_2 + 2, M_1 + M_2 + 2)
\end{bmatrix} \times \begin{bmatrix}
G_1(1) \\
\vdots \\
G_1(M_1) \\
G_2(1) \\
\vdots \\
G_2(M_2) \\
\lambda_T \\
\lambda_Q
\end{bmatrix} = \begin{bmatrix}
B(1) \\
\vdots \\
B(M_1 + M_2 + 2)
\end{bmatrix},$$

where the elements in the first  $(M_1 + M_2)$  lines of matrix A correspond to the coefficients of the unknown non-dimensional circulation values  $G_1(1) \dots G_2(M_2)$  and the Lagrange multipliers in eqs A-1~A-2 and, the elements of the last two lines of A are the coefficients of the circulation values in the constraint equations A-3~A-4.

## Appendix B: MATLAB® functions

### B1. 'Uncoupled' Method

#### UncoupledCRP.m

```
% ===== Written by Dimitrios Laskos =====
% Contra-Rotating Propeller Design Code based on lifting line theory
% Numerical Implementation of the iterative process for CRP Design
% by coupling Single Propellers Design codes. The Variational Optimization
% Method developed by Kerwin, et al. (1986) is used in order to
% determine optimum circulation distributions for the CRP set.
% ----- References
% 1) E.B. Caster & T.A. Lafone, "A Computer Program for the Preliminary
%     Design of Contrarotating Propellers", DTNSRDC Report SPD-596-01,
%     1975.
% 2) J. Kerwin, W. Coney & C. Hsin, "Optimum Circulation Distributions for
%     Single and Multi-Component Propulsors", 21st American Towing
%     Tank Conference (ATTC), 1986.
% 3) W. Coney, "A Method for the Design of a Class of Optimum Marine
%     Propulsors", Ph.D. thesis, MIT, 1989.
% 4) B.D. Cox & A.M. Reed, "Contrarotating Propellers-Design Theory and
%     Application", Propellers '88 Symposium, 1988.
% 5) J. Kerwin, "Hydrofoils and Propellers", MIT Course 2.23 notes, 2007.
% 6) J.W. Wrench, "The Calculation of Propeller Induction Factors", David
%     Taylor Model Basin (Technical Report 1116), Feb. 1957.
% 7) G. Hough & D. Ordway, "Generalized Actuator Disk", Developements in
%     Theoretical and Applied Mechanics, Vol.2, pp.23-31, 1965.
% 8) B. Epps et al., "OpenProp: An open-source parametric design tool for
%     propellers", Grand Challenges in Modeling & Simulation
%     Conference (GCMC '09), 2009.
% -----
% The function 'Coney' is a modified version of the one used in MIT
% OpenProp_v2 Version 1.0. For the calculation of the axial and tangential
% interaction velocities a modified version of the 'CMV' function is used.
% =====
% indices 1,2 refer to the forward and the aft propellers respectively
% Input Variables:
%
% q           [ ],      torque ratio Q2/Q1
% Js          [ ],      Advance coefficient (same for both propellers)
% Rhub        [m],      Hub radius (common for both propellers)
% R           [m],      Propeller radius
% Z1,Z2       [ ],      Number of blades
% Mp          [ ],      Number of vortex panels over the radius
% CTPDES      [ ],      Design thrust loading
% Hub_Flag    [ ],      Inclusion of hub effects (1=YES, 0=NO)
% XR          [ ],      Radial locations for defining inflow velocities
%              and geometric properties
% XVA         [ ],      Va/Vs, axial inflow vel. / ship vel.
% XVT         [ ],      Vt/Vs, tangential inflow vel. / ship vel.
% XCoD        [ ],      chord / propeller diameter
% XCD         [ ],      section drag coefficient
% spacing     [ ],      Type of radial spacing ('cosine' or 'constant')
```

```

% Xs          [m],      Axial separation between propellers

% Output variables:
% n_total     [ ],      total efficiency
% RC,RC2      [ ],      Control Points for forward and aft propellers
% G1,G2       [ ],      Non-dimensional circulation

% -----
function [RC,RC2,G1,G2,n_total]=CRPUncoupled(q,Js,Rhub,R,Z1,Z2,Mp,...
      CTPDES,Hub_Flag,XR,XVA,XVT,XCoD,XCD,spacing,Xs)

% ===== New Inputs for Variational Optimization =====
Rhv=1;
SCF=1;
% =====
Xf=Xs/(R);
ITER=10;
q_iter=1;
q_res=1;
q_last2=0;
q_last1=0;
CTPDES1MF_last2=0;
CTPDES1MF_last1=0;

% Application of Newton method for finding the specific thrust ratio which
% yields the required torque ratio for a given thrust loading
while q_iter<ITER & q_res>1e-5 % (WHILE LOOP A)
    if q_iter==1
        CTPDES1MF=1;
    elseif q_iter==2
        CTPDES1MF=1+(q-(Kq2/Kq1))/(5*q);
    elseif q_iter>2
        CTPDES1MF=CTPDES1MF_last1+(CTPDES1MF_last1-CTPDES1MF_last2)*...
            (q-q_last1)/(q_last1-q_last2);
    end
    CTPDES1=CTPDES1MF*(CTPDES/2);
    CTPDES2=CTPDES-CTPDES1; %thrust coefficient required by aft propeller
    iter_flag=1;
    if iter_flag==1 % (IF CONDITION B)
% =====
% iterative procedure for determining circulation distributions
% for the forward and the aft propellers of the CRP set.
G1_last=0;
G2_last=0;
G_iter=1;
G1_res=1;
G2_res=1;

while G_iter<ITER & (G1_res>1e-5 | G2_res>1e-5) % (WHILE LOOP B)
    %solve for G1,G2 and update respective onset flows

    % Variational Optimization for forward prop
    if G_iter==1
        [RV,G1,TANBIV,TANBIC,VAC,VTC,UASTAR,UTSTAR,RC,CD,CoD]...

```

```

        = Coney (Rhub, R, Z1, Mp, ITER, Rhv, SCF, Js, CTPDES1, Hub_Flag, ...
        XR, XCoD, XCD, XVA, XVT, spacing, 'normal', 0);
elseif G_iter~=1
    [RV, G1, TANBIV, TANBIC, VAC1, VTC1, UASTAR, ...
    UTSTAR, RC, CD, CoD, Kq1, Kt1, CT1, CP1] ...
    = Coney (Rhub, R, Z1, Mp, ITER, Rhv, SCF, Js, CTPDES1, Hub_Flag, ...
            RC, CoD, CD, VA1, VT1, spacing, 'none', RV);
end

% Calculate interaction velocities at aft propeller plane
Vinter2=zeros (2, length(G1));
for i=1:length(G1)
    [Vinter2(:, i)] = CMV (Xf, RC (i), RV, G1, TANBIV, Z1);
end
UA2_INT=Vinter2 (1, :);
UT2_INT=Vinter2 (2, :);
VA2=VAC-UA2_INT;
VT2=VTC-UT2_INT;

% Variational Optimization for aft prop
[RV2, G2, TANBIV2, TANBIC2, VAC2, VTC2, UASTAR2, UTSTAR2, RC2...
CD, CoD, Kq2, Kt2, CT2, CP2] ...
= Coney (Rhub, R, Z2, Mp, ITER, Rhv, SCF, Js, CTPDES2, Hub_Flag, ...
        RC, CoD, CD, VA2, VT2, spacing, 'none', RV);
% calculate interaction velocities at forward propeller plane
Vinter1=zeros (2, length(G2));
for i=1:length(G2)
    [Vinter1(:, i)] = CMV (-Xf, RC (i), RV2, G2, TANBIV2, Z2);
end
UA1_INT=Vinter1 (1, :);
UT1_INT=Vinter1 (2, :);
VA1=VAC-UA1_INT;
VT1=VTC-UT1_INT;

G_iter=G_iter+1
G1_res=abs (G1-G1_last);
G2_res=abs (G2-G2_last);
G1_last=G1;
G2_last=G2;
end % (END OF WHILE LOOP B)
=====
elseif iter_flag~=1

% Variational Opt for forward prop
[RV, G, TANBIV, TANBIC, VAC, VTC, UASTAR, UTSTAR, RC, CD, CoD] ...
= Coney (Rhub, R, Z1, Mp, ITER, Rhv, SCF, Js, CTPDES1, Hub_Flag, ...
        XR, XCoD, XCD, XVA, XVT, spacing, 'normal', 0);

% First calculate interaction velocities at aft propeller plane
Vinter2=zeros (2, length(G));
for i=1:length(G)
    [Vinter2(:, i)] = CMV (Xf, RC (i), RV, G, TANBIV, Z1);
end
VA2=VAC-Vinter2 (1, :);
VT2=VTC-Vinter2 (2, :);

```

```

CTPDES2=CTPDES-CTPDES1;    %thrust coefficient required by aft propeller

[RV2,G2,TANBIV2,TANBIC2,VAC2,VTC2,UASTAR2,UTSTAR2,RC2]...
=Coney(Rhub,R,Z2,Mp,ITER,Rhv,SCF,Js,CTPDES2,Hub_Flag,...
      RC,CoD,CD,VA2,VT2,spacing,'none',RV);

% calculate interaction velocities at forward propeller plane
Vinter1=zeros(2,length(G2));
for i=1:length(G2)
    [Vinter1(:,i)] = CMV(-Xf,RC(i),RV2,G2,TANBIV2,Z2);
end
VA1=VAC-Vinter1(1,:);
VT1=VTC-Vinter1(2,:);

% again run Coney.m for forward prop
[RV_new,G1_new,TANBIV_new,TANBIC_new,VAC1,VTC1,UASTAR,UTSTAR,RC_new,...
 CD1,CoD1,Kq1,Kt1,CT1,CP1,EFFY1,VMIV1]=Coney(Rhub,R,Z1,Mp,ITER,Rhv,...
 SCF,Js,CTPDES1,Hub_Flag,RC,CoD,CD,VA1,VT1,spacing,'none',RV);

% calculate new interaction velocities at aft propeller plane
Vinter2b=zeros(2,length(G));
for i=1:length(G)
    [Vinter2b(:,i)] = CMV(Xf,RC2(i),RV_new,G1_new,TANBIV_new,Z1);
end
VA2=VAC-Vinter2b(1,:);
VT2=VTC-Vinter2b(2,:);

% run Coney.m for aft propeller
[RV2_new,G2_new,TANBIV2_new,TANBIC2_new,VAC2,VTC2,UASTAR2,UTSTAR2,...
 RC2_new,CD2,CoD2,Kq2,Kt2,CT2,CP2,EFFY2,VMIV2]=Coney(Rhub,R,Z2,Mp,...
 ITER,Rhv,SCF,Js,CTPDES2,Hub_Flag,RC2,CoD,CD,VA2,VT2,spacing,'none',RV2);

end % (END OF IF CONDITION B)
q_iter=q_iter+1;
q_res=abs((Kq2/Kq1)-q);
q_last2=q_last1;
q_last1=Kq2/Kq1;
CTPDES1MF_last2=CTPDES1MF_last1;
CTPDES1MF_last1=CTPDES1MF;
end % (END OF WHILE LOOP A)

% ----- Compute total efficiency -----
% This expression however applies only to the case for which the diameters,
% and the speeds of the two propellers are equal. The effect of the hub
% drag is also neglected
n_total=(CT1+CT2)/(CP1+CP2);
% -----
figure;
subplot(3,1,1)
plot(RC,G1,'-*',RC2,G2,'-*r');grid on;xlabel('Control points radii (RC)');
ylabel('Non-dimensional Circulation (G)');subplot(3,1,2)
plot(RC,atand(TANBIC),RC2,atand(TANBIC2),'-r');
xlabel('Control points radii (RC)');
ylabel('Hydrodynamic Pitch Angle \beta_i (deg)');grid on
% =====

```

```

% Plotting total inflow velocities
VASTAR= VAC          + UASTAR; % total axial inflow vel. / ship vel.
VTSTAR= VTC + pi*RC/Js + UTSTAR; % total tangential inflow vel. / ship vel.
VSTAR = sqrt(VTSTAR.^2 + VASTAR.^2); %magnitude of the inflow vel./ship vel.

VASTAR2= VAC2          + UASTAR2; %total axial inflow vel./ship vel.
VTSTAR2= VTC2 + pi*RC2/Js + UTSTAR2; %total tangential inflow vel./ship vel.
VSTAR2 =sqrt(VTSTAR2.^2+VASTAR2.^2); %magnitude of the inflow vel./ship vel.
subplot(3,1,3);plot(RC,VSTAR, '-*', RC2,VSTAR2, '-*r');grid on;
xlabel('Control points radii (RC)')
ylabel('total inflow velocity (V_s_t_a_r/V_s)')

% ===== Induced Velocities far downstream =====
M1=Mp;M2=Mp;Z1=Z;Z2=Z;
[UAHIFinf_1,UTHIFinf_1]=Horseshoe_int(M2,M1,Z1,TANBIV,RC2,RV,20*Xf,...
                                     Hub_Flag,Rhub_or);
[UAHIFinf_2,UTHIFinf_2]=Horseshoe_int(M1,M2,Z2,TANBIV2,RC,RV2,19*Xf,...
                                     Hub_Flag,Rhub_or);
[UA_INTinf_1,UT_INTinf_1]=Induced_Velocity_int(M2,M1,G1,UAHIFinf_1,...
                                               UTHIFinf_1);
[UA_INTinf_2,UT_INTinf_2]=Induced_Velocity_int(M1,M2,G2,UAHIFinf_2,...
                                               UTHIFinf_2);

% Induced velocities far downstream for Single propeller having double the
% the same number of blades as the CRP set
Z_SR=Z1+Z2;
[RVs,Gs,TANBIVs,TANBICs,VACs,VTCs,UASTARs,UTSTARs,RCs,CDs,CoDs]= ...
    Coney(Rhub,R,Z_SR,Mp,ITER,Rhv,SCF,Js,CTPDES,Hub_Flag,XR,XCoD,XCD,XVA,...
        XVT,spacing,'normal',0);
[UAHIFinf_SR,UTHIFinf_SR]=Horseshoe_int(Mp,Mp,Z_SR,TANBIVs,RCs,RVs,...
                                       20*Xf,Hub_Flag,Rhub_or);
[UA_INTinf_SR,UT_INTinf_SR]=Induced_Velocity_int(Mp,Mp,Gs,UAHIFinf_SR,...
                                               UTHIFinf_SR);

% -----
figure;
plot(RC2,UA_INTinf_1,RC2,UA_INTinf_2,'-r',...
     RC2,UA_INTinf_1+UA_INTinf_2,'--g*',RCs,UA_INTinf_SR,'--b*');grid on
legend('forward propeller','aft propeller','total CRP','single propeller')
title('Axial induced velocities far downstream / Uncoupled method');

figure;
plot(RC2,UT_INTinf_1,RC2,-UT_INTinf_2,'-r',...
     RC2,UT_INTinf_1-UT_INTinf_2,'--g*',RCs,UT_INTinf_SR,'--b*');grid on
legend('forward propeller','aft propeller','total CRP','single propeller')
title('Tangential induced velocities far downstream / Uncoupled method');
% =====
% Plot interaction and self induced velocities
figure;
subplot(2,1,1);
plot(UASTAR,RC, '-*', Xs+UASTAR2,RC2, '-*r');grid on
hold on
plot(zeros(1,length(RC)),RC, '--', 'Linewidth',2)
plot(Xs*ones(1,length(RC2)),RC2, '--r', 'Linewidth',2)
hold off
title('axial self-induced velocities')
subplot(2,1,2)
plot(UTSTAR,RC, '-*', Xs+UTSTAR2,RC2, '-*r');grid on

```

```

hold on
plot(zeros(1,length(RC)),RC,'--','Linewidth',2)
plot(Xs*ones(1,length(RC2)),RC2,'--r','Linewidth',2)
title('tangential self-induced velocities')
% ----- Interaction Velocities-----
figure;
subplot(2,1,1);
plot(-UA1_INT,RC,'-*',Xs-UA2_INT,RC2,'-*r');grid on
hold on
plot(zeros(1,length(RC)),RC,'--','Linewidth',2)
plot(Xs*ones(1,length(RC2)),RC2,'--r','Linewidth',2)
title('axial interaction velocities')
subplot(2,1,2)
plot(-UT1_INT,RC,'-*',Xs-UT2_INT,RC2,'-*r');grid on
hold on
plot(zeros(1,length(RC)),RC,'--','Linewidth',2)
plot(Xs*ones(1,length(RC2)),RC2,'--r','Linewidth',2)
title('tangential interaction velocities')
end

```

## B2. ‘Coupled’ Method

### CoupledCRP.m

```

% =====
% Contra-Rotating Propeller Design Code based on lifting line theory
% Numerical Implementation of the Variational Optimization Method
% for Two-Component Propulsors developed by Kerwin, et al. (1986)
%
% ----- Copyright 2010 Dimitrios Laskos -----
% This program is free software. You can redistribute it and/or modify it
% under the terms of the GNU General Public License version 2, as published
% by the Free Software Foundation. This program is distributed in the hope
% that it will be useful, but WITHOUT ANY WARRANTY; without even the
% implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
% See the GNU General Public License for more details.
%
% ----- References -----
% 1) J.S. Carlton, "Marine Propellers & Propulsion", chapter 3, 1994.
% 2) J. Kerwin, W. Coney & C. Hsin, "Optimum Circulation Distributions for
%     Single and Multi-Component Propulsors", 21st American Towing
%     Tank Conference (ATTC), 1986.
% 3) W. Coney, "A Method for the Design of a Class of Optimum Marine
%     Propulsors", Ph.D. thesis, MIT, 1989.
% 4) J. Kerwin, "Hydrofoils and Propellers", MIT Course 2.23 notes, 2007.
% 5) J.W. Wrench, "The Calculation of Propeller Induction Factors", David
%     Taylor Model Basin (Technical Report 1116), Feb. 1957.
% 6) M. Wang, "Hub Effects in Propeller Design and Analysis", Ph.D. thesis,
%     MIT, 1985.
% 7) G. Hough & D. Ordway, "Generalized Actuator Disk", Developements in
%     Theoretical and Applied Mechanics, Vol.2, pp.23-31, 1965.
% 8) B. Epps et al., "OpenProp: An open-source parametric design tool for
%     propellers", Grand Challenges in Modeling & Simulation

```



```

% Conference (GCMC '09), 2009.
% =====
% Last modified: 05/03/2010 by Dimitrios Laskos
% -----
% indices 1,2 refer to the forward and the aft propellers respectively
% Input Variables:
%
% Rhub      [m],      Hub radius (common for both propellers)
% R1,R2     [m],      Propeller radius
% M1,M2     [ ],      Number of vortex panels over the radius
% Z1,Z2     [ ],      Number of blades
% Tr        [N],      Required total thrust
% q         [ ],      torque ratio Q2/Q1
% N1,N2     [RPM],    Propeller speed
% XR1,XR2   [ ],      Radial locations for defining inflow velocities
%            and geometric properties
% XCoD1,XCoD2 [ ],    chord / propeller diameter
% CD        [ ],      section drag coefficient
% XVA1,XVA2 [ ],      Va/Vs, axial inflow vel. / ship vel.
% XVT1,XVT2 [ ],      Vt/Vs, tangential inflow vel. / ship vel.
% Vs        [m/s],    Ship speed
% Xf        [m],      Axial separation between propellers
% ITER      [ ],      Max. Iterations for Circulation Convergence and
%            Wake Alignment
% spacing   Type of radial spacing ('cosine' or 'constant')
% Hub_Flag  Inclusion of hub effects (1=YES, 0=NO)
% Rhv      Hub Vortex Radius/Hub Radius

% Output variables:
% EFFY      [ ],      total efficiency
% CT        [ ],      thrust coefficient, eqn (161) p.138
% CQ        [ ],      torque coefficient, eqn (161) p.138
% KT        [ ],      thrust coefficient, eqn (162) p.138
% KQ        [ ],      torque coefficient, eqn (162) p.138
% G         [ ],      Non-dimensional Circulation
% UA_SELF   [ ],      Axial self-induced velocity vector / Vs
% UT_SELF   [ ],      Tangential self-induced velocity vector / Vs
% UA_INT1_2 [ ],      Axial interaction velocity vector on component 1
%            induced by component 2 / Vs
% UT_INT1_2 [ ],      Tangential interaction velocity vector on
%            component 1 induced by component 2 / Vs
% TANBIC    [ ],      Tangent of hydrodynamic pitch angle
% VSTAR     [ ],      Total inflow velocity / Vs
% C1        [ ],      Required lift coefficient

% -----
function [EFFY,CT1,CT2,CQ1,CQ2,KT1,KT2,KQ1,KQ2,RC1,RC2,G1,...
         G2,UA_SELF1,UT_SELF1,UA_INT1_2,UT_INT1_2,UA_SELF2,UT_SELF2,...
         UA_INT2_1,UT_INT2_1,TANBIC1,TANBIC2,VSTAR1,VSTAR2,C11,C12]=...
CoupledCRP(Rhub,R1,R2,M1,M2,Z1,Z2,Tr,q,N1,N2,XR1,XR2,XCoD1,...
           XCoD2,CD,XVA1,XVA2,XVT1,XVT2,Vs,Xf,ITER,spacing,Hub_Flag,Rhv)

% ----- Apply spacing -----
RV1=zeros(1,M1+1);RC1=zeros(1,M1); % initialize RC1 and RV1

```

```

RV2=zeros(1,M2+1);RC2=zeros(1,M2); % initialize RC2 and RV2
RoR=1;
Rhub_oR1=Rhub/R1;
Rhub_oR2=Rhub/R2;
if strcmp(spacing,'constant')==1 %Constant spacing
    if Hub_Flag==0
        DRR1 = (RoR-Rhub_oR1)/(M1+.5); % panel size
        DRR2 = (RoR-Rhub_oR2)/(M2+.5);
        RV1(M1+1)=RoR-.25*DRR1; % 25% tip inset
        RV2(M2+1)=RoR-.25*DRR2;
        RV1(1)=Rhub_oR1+.25*DRR1; % 25% hub inset (NO IMAGE HUB)
        RV2(1)=Rhub_oR2+.25*DRR2;
    elseif Hub_Flag==1
        DRR1 = (RoR-Rhub_oR1)/(M1+.25); % panel size
        DRR2 = (RoR-Rhub_oR2)/(M2+.25);
        RV1(M1+1)=RoR-.25*DRR1; % 25% tip inset
        RV2(M2+1)=RoR-.25*DRR2;
        RV1(1)=Rhub_oR1; % 25% hub inset (NO IMAGE HUB)
        RV2(1)=Rhub_oR2;
    end

    RC1(1)=RV1(1)+.5*DRR1; % ctrl pt at mid-panel
    for m=2:M1
        RV1(m)=RV1(m-1)+DRR1;
        RC1(m)=RC1(m-1)+DRR1;
    end
    RC2(1)=RV2(1)+.5*DRR2; % ctrl pt at mid-panel
    for m=2:M2
        RV2(m)=RV2(m-1)+DRR2;
        RC2(m)=RC2(m-1)+DRR2;
    end
end

elseif strcmp(spacing,'cosine')==1 %Cosine spacing

    DEL1 = pi/(2*M1);
    Rdif1 = 0.5*(RoR - Rhub_oR1);
    for m = 1:M1+1
        RV1(m) = Rhub_oR1 + Rdif1*(1-cos(2*(m-1)*DEL1));
    end
    for n = 1:M1
        RC1(n) = Rhub_oR1 + Rdif1*(1-cos((2*n-1)*DEL1));
    end
    DEL2 = pi/(2*M2);
    Rdif2 = 0.5*(RoR - Rhub_oR2);
    for m = 1:M2+1
        RV2(m) = Rhub_oR2 + Rdif2*(1-cos(2*(m-1)*DEL2));
    end
    for n = 1:M2
        RC2(n) = Rhub_oR2 + Rdif2*(1-cos((2*n-1)*DEL2));
    end
end

end
% -----
DR1=diff(RV1);DR2=diff(RV2);
% -----Interpolate Va,Vt and CoD at vortex and control points-----
VAC1 = pchip(XR1,XVA1,RC1); % axial inflow vel. / ship vel. at ctrl pts
VTC1 = pchip(XR1,XVT1,RC1); % tang. inflow vel. / ship vel. at ctrl pts

```

```

CoD1 = pchip(XR1,XCoD1,RC1);      % chord / propeller diameter at ctrl pts
VAC2 = pchip(XR2,XVA2,RC2);
VTC2 = pchip(XR2,XVT2,RC2);
CoD2 = pchip(XR2,XCoD2,RC2);
% -----
Js1=Vs/((N1/60)*2*R1); %n=N/60 [rev/sec]
Js2=Vs/((N2/60)*2*R2);
om1=pi/Js1;          %tip speed ratio
om2=pi/Js2;

%----- Initialize induced velocity vectors -----
UTSTAR1(1:M1)=0;
UTSTAR2(1:M2)=0;
UASTAR1(1:M1)=0;
UASTAR2(1:M2)=0;

% ----- Assign Initial Values -----
G1_last=0;
G2_last=0;
LT_last=-1;
LQ_last=0;
A=zeros(M1+M2+2);
B=zeros(M1+M2+2,1);
% ----- Initial estimates for hydrodynamic pitch angles -----
[TANBIC1,TANBIV1] = find_tan_BetaI(VAC1,VTC1,UASTAR1,UTSTAR1,RC1,RV1,Js1);
[TANBIC2,TANBIV2] = find_tan_BetaI(VAC2,VTC2,UASTAR2,UTSTAR2,RC2,RV2,Js2);
TANBC1=TANBIC1;
TANBC2=TANBIC2;
% -----
% Iteration for betaI's. BetaI's are fixed.
B_iter=1;
B1_res=1;
B2_res=1;
B_res=[B1_res B2_res];
TANBIC1_last=TANBIC1;
TANBIC2_last=TANBIC2;
% ----- Compute Horseshoe Influence Functions -----
% UAHIF1_2 is the horseshoe influence matrix for the axial interaction
% velocities induced by component 2 (aft) on component 1 (forward)
[UAHIF1,UTHIF1]=Horseshoe_self(M1,Z1,TANBIV1,RC1,RV1,Hub_Flag,Rhub_oR1);
[UAHIF1_2,UTHIF1_2]=Horseshoe_int(M1,M2,Z2,TANBIV2,RC1,RV2,-Xf,Hub_Flag,...
                                   Rhub_oR2);
[UAHIF2,UTHIF2]=Horseshoe_self(M2,Z2,TANBIV2,RC2,RV2,Hub_Flag,Rhub_oR2);
[UAHIF2_1,UTHIF2_1]=Horseshoe_int(M2,M1,Z1,TANBIV1,RC2,RV1,Xf,Hub_Flag,...
                                   Rhub_oR1);
% -----
figure;
hold on
G1(1)=0;
G2(1)=0;
while B_iter<ITER & any(B_res)==1 % (WHILE LOOP B1)
    G_iter=1;
    G1_res=1;
    G2_res=1;
    LT_res=1;
    LQ_res=1;

```

```

rho=1025;
while G_iter<ITER &(G1_res>1e-5 | G2_res>1e-5 | LT_res>1e-5 | ...
                                LQ_res>1e-5)% (WHILE LOOP G1)
% Solve simultaneous equations for G1,G2,LT and LQ.
% Setting up the linear system of M1+M2+2 equations
% There are omissions in equations 2.61 and 2.62 which affect the
% coefficients of LQ and LT, A(:,M1+M2+1) and A(:,M1+M2+2), as well as
% the constant values in matrix B.
% First eq. 2.61 (Coney, p.42)
    for i=1:M1
        for m=1:M1
            A(i,m)=(om1+q*LQ_last)*Z1*(UAHIF1(i,m)*RC1(i)*DR1(i)+...
                                       UAHIF1(m,i)*RC1(m)*DR1(m))+...
                LT_last*Z1*( UTHIF1(i,m)*          DR1(i)+...
                            UTHIF1(m,i)*          DR1(m));
        end
        for m=1:M2
            A(i,m+M1)=(om1+q*LQ_last)*Z1*(UAHIF1_2(i,m)*RC1(i)*DR1(i))+...
                (om2-LQ_last)* Z2*(UAHIF2_1(m,i)*RC2(m)*DR2(m))+...
                LT_last*      Z1* UTHIF1_2(i,m)*      DR1(i)+...
                LT_last*      Z2* UTHIF2_1(m,i)*      DR2(m);
        end
        A(i,M1+M2+1)=Z1*(VTC1(i)+om1*RC1(i))*DR1(i);
        A(i,M1+M2+2)=Z1*q*VAC1(i)*RC1(i)*DR1(i);

% The circulation coefficients in the thrust constrain equation must be
% multiplied by (2*rho*Vs^2*pi*R^2) for dimensional consistency since Tr
% has dimensions [N]
        A(M1+M2+1,i)=(2*rho*Vs^2*pi*R1^2)*...
            Z1*(VTC1(i)+om1*RC1(i)+UTSTAR1(i))*DR1(i);%thrust terms
        A(M1+M2+2,i)=q*Z1*(VAC1(i)+UASTAR1(i))*RC1(i)*DR1(i);%torque t.
        B(i)=-Z1*om1*VAC1(i)*RC1(i)*DR1(i);
    end

% Then eq. 2.62 (Coney, p.43)
    for i=1:M2
        for m=1:M1
            A(i+M1,m)=(om1+q*LQ_last)*Z1*UAHIF1_2(m,i)*RC1(m)*DR1(m)+...
                (om2-LQ_last)*Z2* UAHIF2_1(i,m)*RC2(i)*DR2(i)+...
                LT_last*      (Z1*UTHIF1_2(m,i)*      DR1(m)+...
                            Z2*UTHIF2_1(i,m)*      DR2(i));
        end
        for m=1:M2
            A(i+M1,m+M1)=(om2-LQ_last)*Z2*(UAHIF2(i,m)*RC2(i)*DR2(i)+...
                                       UAHIF2(m,i)*RC2(m)*DR2(m))+...
                LT_last*Z2*      (UTHIF2(i,m)*      DR2(i)+...
                            UTHIF2(m,i)*      DR2(m));
        end
        A(i+M1,M1+M2+1)=Z2*(VTC2(i)+om2*RC2(i))*DR2(i);
        A(i+M1,M1+M2+2)=-Z2*VAC2(i)*RC2(i)*DR2(i);

% The circulation coefficients in the thrust constrain equation must be
% multiplied by (2*rho*Vs^2*pi*R^2) for dimensional consistency since Tr
% has dimensions [N]
        A(M1+M2+1,i+M1)=(2*rho*Vs^2*pi*R2^2)*...
            Z2*(VTC2(i)+om2*RC2(i)+UTSTAR2(i))*DR2(i); %thrust terms

```

```

    A(M1+M2+2,i+M1)=-Z2*(VAC2(i)+UASTAR2(i))*RC2(i)*DR2(i);%torque
    B(M1+i)=-Z2*om2*VAC2(i)*RC2(i)*DR2(i);
end
%
%   Modify terms related to circulation at the hub (innermost radial
%   distance). The difference in the results is very small though.
if Hub_Flag==1
    a=LT_last/(16*pi)*(log(1/Rhv)+3);
    A(1,1)=A(1,1)-a*2*Z1^2;
    A(1,M1+1)=A(1,M1+1)+a*2*Z1*Z2;
    A(M1+1,1)=A(M1+1,1)+a*2*Z1*Z2;
    A(M1+1,M1+1)=A(M1+1,M1+1)-a*2*Z2^2;
end
%---Compute total velocities used in viscous force calculations---
VASTAR1=VAC1+UASTAR1;VASTAR2=VAC2+UASTAR2;
VTSTAR1=VTC1+om1*RC1+UTSTAR1;VTSTAR2=VTC2+om2*RC2+UTSTAR2;
VSTAR1=sqrt(VASTAR1.^2+VTSTAR1.^2);
VSTAR2=sqrt(VASTAR2.^2+VTSTAR2.^2);

Tv1=0;
Qv1=0;

for i=1:M1
    Tv1=Tv1-(1/2)*Z1*VSTAR1(i)*(VAC1(i)+UASTAR1(i))*...
        *CoD1(i)*CD*DR1(i); %viscous thrust

    Qv1=Qv1+(1/2)*Z1*VSTAR1(i)*(VTC1(i)+om1*RC1(i)+UTSTAR1(i))*...
        *RC1(i)*CoD1(i)*CD*DR1(i);%v.torque
end
%
%   or Tv1=Z1*sum(VSTAR1.*(VAC1+UASTAR1).*CoD1.*CD.*DR1)
%   and similarly for Qv1.

Tv2=0;
Qv2=0;
for i=1:M2
    Tv2=Tv2-(1/2)*Z2*VSTAR2(i)*(VAC2(i)+UASTAR2(i))*...
        *CoD2(i)*CD*DR2(i); %viscous thrust

    Qv2=Qv2+(1/2)*Z2*VSTAR2(i)*(VTC2(i)+om2*RC2(i)+UTSTAR2(i))*...
        *RC2(i)*CoD2(i)*CD*DR2(i);%v. torque
end
% The viscous thrust terms above must be multiplied by 2*rho*Vs^2*R^2 in
% order to represent dimensional values [N]since Tr is dimensional.
    B(M1+M2+1)=Tr-(2*rho*Vs^2)*(R1^2*Tv1+R2^2*Tv2);
% =====
% Account for hub drag term if a hub image is present
% Fh is expressed in dimensional form in order to be consistent
% with the dimensional value of Tr.
    if Hub_Flag==1;
        Fh=rho/(16*pi)*(log(1/Rhv)+3)*(Z1*G1(1)*sqrt(0.5*rho*...
            Vs^2*pi*R1^2)-Z2*G2(1)*sqrt(0.5*rho*Vs^2*pi*R2^2))^2;
    elseif Hub_Flag==0
        Fh=0;
    end
    B(M1+M2+1)=B(M1+M2+1)+Fh;
% =====
% Divide viscous torque terms by pi since the non-dimensionalizing param-

```

```

% ters for Qi and Qv differ by the multiplication parameter pi (Qi=1/pi*Qv)
B(M1+M2+2)=(Qv2-q*Qv1)/pi;
GL=linsolve(A,B);
G1=GL(1:M1);
G2=GL(M1+1:M1+M2);
LT=GL(M1+M2+1);
LQ=GL(M1+M2+2);

% -----Compute induced velocities-----
[UA_SELF1,UT_SELF1,UA_INT1_2,UT_INT1_2]=Induced_Velocity(M1,M2,...
G1,G2,UAHIF1,UTHIF1,UAHIF1_2,UTHIF1_2);
UASTAR1=UA_SELF1+UA_INT1_2; %total axial induced velocity/Vs
UTSTAR1=UT_SELF1+UT_INT1_2; %total tangential induced velocity/Vs

[UA_SELF2,UT_SELF2,UA_INT2_1,UT_INT2_1]=Induced_Velocity(M2,M1,...
G2,G1,UAHIF2,UTHIF2,UAHIF2_1,UTHIF2_1);
UASTAR2=UA_SELF2+UA_INT2_1; %total axial induced velocity/Vs
UTSTAR2=UT_SELF2+UT_INT2_1; %total tangential induced velocity/Vs
%-----

% -----Prepare for next iteration-----
G_iter=G_iter+1;
G1_res=abs(G1-G1_last);
G2_res=abs(G2-G2_last);
LT_res=abs(LT-LT_last);
LQ_res=abs(LQ-LQ_last);
G1_last=G1;
G2_last=G2;
LT_last=LT;
LQ_last=LQ;
% WARNING IF LOOP G1 DOESN'T CONVERGE
% check for G1,G2 and LM convergence
if G_iter > ITER
    warning('on'),
    warning('WARNING: While loop G1 did NOT converge.'),
    warning('off'),
end
end % (END WHILE LOOP G1)

% -----Align wake to new circulation distributions-----
[UAHIF1,UTHIF1,UAHIF2,UTHIF2,UAHIF1_2,UTHIF1_2,UAHIF2_1,UTHIF2_1,...
UASTAR1,UTSTAR1,UASTAR2,UTSTAR2,TANBIC1,TANBIV1,TANBIC2,TANBIV2] = ...
Align_wake(TANBIC1,TANBIV1,TANBIC2,TANBIV2,ITER,M1,M2,Z1,Z2,RC1,...
RV1,RC2,RV2,G1,G2,VAC1,VTC1,VAC2,VTC2,Js1,Js2,Xf,Hub_Flag,Rhub_or1,...
Rhub_or2);
% -----End of wake alignment-----

B_iter=B_iter+1;
B1_res=abs(TANBIC1-TANBIC1_last)
B2_res=abs(TANBIC2-TANBIC2_last)
B_res=[B1_res>1e-2 B2_res>1e-2]; %convergence limit for TANBI
TANBIC1_last=TANBIC1;
TANBIC2_last=TANBIC2;

% -----Plot Circulation Distributions-----

plot(RC1,G1,RC2,G2,'r')

```

```

% -----
if B_iter > ITER
    warning('on'),
    warning('WARNING: While loop B1 did NOT converge.'),
    warning('off'),
end
end                                     %(END WHILE LOOP B1)
grid on;
hold off

% ===== Plotting self-induced and interaction velocities =====

figure;
subplot(2,1,1);
plot(RC1,UA_SELF1,'-*',RC2,UA_SELF2,'-*r');grid on
title('axial self-induced velocities')
subplot(2,1,2)
plot(RC1,UT_SELF1,'-*',RC2,UT_SELF2,'-*r');grid on
title('tangential self-induced velocities')

figure;
subplot(2,1,1);
plot(RC1,UA_INT1_2,'-*',RC2,UA_INT2_1,'-*r');grid on
title('axial interaction velocities')
subplot(2,1,2)
plot(RC1,UT_INT1_2,'-*',RC2,UT_INT2_1,'-*r');grid on
title('tangential interaction velocities')

% ===== Induced Velocities Far Downstream =====
[UAHIFinf_1,UTHIFinf_1]=Horseshoe_int(M2,M1,Z1,TANBIV1,RC2,RV1,20*Xf,...
                                     Hub_Flag,Rhub_or1);
[UAHIFinf_2,UTHIFinf_2]=Horseshoe_int(M1,M2,Z2,TANBIV2,RC1,RV2,19*Xf,...
                                     Hub_Flag,Rhub_or2);

[UA_SELFinf2,UT_SELFinf2,UA_INTinf_1,UT_INTinf_1]=Induced_Velocity(M2,...
                           M1,G2,G1,UAHIF2,UTHIF2,UAHIFinf_1,UTHIFinf_1);
[UA_SELFinf1,UT_SELFinf1,UA_INTinf_2,UT_INTinf_2]=Induced_Velocity(M1,...
                           M2,G1,G2,UAHIF1,UTHIF1,UAHIFinf_2,UTHIFinf_2);

figure;
plot(RC2,UA_INTinf_1,RC2,UA_INTinf_2,'-r',RC2,...
     UA_INTinf_1+UA_INTinf_2,'--');grid on
legend('forward propeller','aft propeller','total CRP')
title('Axial induced velocities far downstream');
figure;
plot(RC2,UT_INTinf_1,RC2,-UT_INTinf_2,'-r',RC2,...
     UT_INTinf_1-UT_INTinf_2,'--');grid on
legend('forward propeller','aft propeller','total CRP')
title('Tangential induced velocities far downstream');
% ===== Forces Function=====
VMIV1 = 2*trapz(XR1,XR1.*XVA1)/(RoR^2-Rhub_or1^2);% [ ], VMIV/ship velocity
VMIV2 = 2*trapz(XR2,XR2.*XVA2)/(RoR^2-Rhub_or2^2);% [ ], VMIV/ship velocity

[CT1,CQ1,KT1,KQ1,CT2,CQ2,KT2,KQ2,EFFY,VSTAR1,VSTAR2] =...
Forces(CD,DR1,DR2,VAC1,VAC2,TANBC1,TANBC2,...
      UASTAR1,UASTAR2,UTSTAR1,UTSTAR2,CoD1,CoD2,G1,G2,M1,M2,RC1,RC2,...
      Fh,Z1,Z2,Js1,Js2,VMIV1,VMIV2,N1,N2,Vs,R1,R2);

```

```

% =====
geometry_flag=1;      %flag for geometry generation
if geometry_flag==1
% ----- Compute required lift coefficients
Gamma1=G1*2*pi*R1*Vs;
Gamma2=G2*2*pi*R2*Vs;
Cl1= 2*Gamma1'./(VSTAR1*Vs.*CoD1.*2*R1);
Cl2= 2*Gamma2'./(VSTAR2*Vs.*CoD2.*2*R2);
% ===== Inputs necessary for geometry generation =====
skew01      = zeros(1,11);          % Skew [deg]
skew02      = zeros(1,11);
rake01      = zeros(1,11);          % Xs/D, Rake
rake02      = zeros(1,11);
% ----- t0/c, thickness / chord -----
t0oc01      = [.2056 .1551 .1181 .0902 .0694 .0541 .0419 .0332 .0324...
               .0204 .005];
               % [0.0815 0.0771 0.0731 0.0664 0.0608 0.0561 0.0522...
               % 0.0489 0.0457 0.0457 0.005];
t0oc02      = t0oc01;
% -----
BetaI_c1=atand(TANBIC1);
BetaI_c2=atand(TANBIC2);
Np=40;      % Number of points over the chord
% ===== Generate Propeller Geometry =====
Geometry(XR1,XR2,t0oc01,t0oc02,skew01,skew02,rake01,rake02,RC1,RC2,Cl1,...
         Cl2,BetaI_c1,BetaI_c2,Xf,Z1,Z2,Rhub,CoD1,CoD2,R1,R2,M1,M2,Np)

% ===== Increase the number of sections and the respective values
% (t0oc,f0oc,AlphaI,Cl,Z3D,Vstar,RC,c) such that the computation of the
% cavitating area is more accurate =====
% Cosine spacing is used with the end values remaining the same. The new
% number of sections are given by M1_int, M2_int.
M1_int=40;M2_int=40;
DEL1=(RC1(end)-RC1(1))/2;
DEL2=(RC2(end)-RC2(1))/2;
for n=1:M1_int
    RC1_int(n)=RC1(1)+DEL1*(1-cos(n*pi/M1_int));
end

for n=1:M2_int
    RC2_int(n)=RC2(1)+DEL2*(1-cos(n*pi/M2_int));
end
% ===== Now interpolate to find new values at RC1,2_int locations =====
Gamma1_int=pchip(RC1,Gamma1,RC1_int);
Gamma2_int=pchip(RC2,Gamma2,RC2_int);
VSTAR1_int=pchip(RC1,VSTAR1*Vs,RC1_int);    %dimensional velocity!
VSTAR2_int=pchip(RC2,VSTAR2*Vs,RC2_int);    %dimensional velocity!
UASTAR1_int=pchip(RC1,UASTAR1,RC1_int);
UASTAR2_int=pchip(RC2,UASTAR2,RC2_int);
CoD1_int=pchip(RC1,CoD1,RC1_int);
CoD2_int=pchip(RC2,CoD2,RC2_int);
Cl1_int= 2*Gamma1_int./(VSTAR1_int.*CoD1_int.*2*R1); %new Cl1
Cl2_int= 2*Gamma2_int./(VSTAR2_int.*CoD2_int.*2*R2); %new Cl2
BetaI_c1_int=pchip(RC1,BetaI_c1,RC1_int);
BetaI_c2_int=pchip(RC2,BetaI_c2,RC2_int);

```



```

% ----- Alternatively could extrapolate -----
% RC1_int=0.9*Rhub_oR1+(1-0.9*Rhub_oR1)*(sin((0:40)*pi/(2*40)));
% RC2_int=0.9*Rhub_oR1+(1-0.9*Rhub_oR1)*(sin((0:40)*pi/(2*40)));
% Gammal_int=interp1(RC1,Gammal,RC1_int,'pchip','extrap');
%...
% -----

% ===== Run Geometry module again =====
[f0oc1,f0oc2,t0oc1,t0oc2,AlphaI1,AlphaI2,X3D,Y3D,Z3D,X3D_aft,Y3D_aft,...
Z3D_aft,c1,c2,x0_1,x0_2,theta_Z1,theta_Z2] = Geometry(XR1,XR2,t0oc01,...
t0oc02,skew01,skew02,rake01,rake02,RC1_int,RC2_int,C11_int,C12_int,...
BetaI_c1_int,BetaI_c2_int,Xf,Z1,Z2,Rhub,CoD1_int,CoD2_int,R1,R2,M1_int,...
M2_int,Np);

% ===== Run Cavitation module for both propellers =====
Cp_mode='VLM';
H=2; % shaft centerline depth
[Color_matrix_upper,Color_matrix_lower,cav_mess1]=Cavitation(Cp_mode,...
M1_int,t0oc1,f0oc1,AlphaI1,C11_int,H,Z3D,VSTAR1_int,Np,Z1,x0_1,RC1_int,...
c1,R1,theta_Z1,BetaI_c1_int,UASTAR1_int);

[Color_matrix_upper_aft,Color_matrix_lower_aft,cav_mess2]=...
Cavitation(Cp_mode,M2_int,t0oc2,f0oc2,AlphaI2,C12_int,H,Z3D_aft,...
VSTAR2_int,Np,Z2,x0_2,RC2_int,c2,R2,theta_Z2,BetaI_c2_int,UASTAR2_int);

% ===== Plot Propeller Cavitation Image =====
figure;
grid on;
axis equal;
axis([-2*Xf*R1 R1 -1.1*R1 1.1*R1 -1.1*R1 1.1*R1]);
xlabel('X (3D) [m]','FontSize',12);
ylabel('Y (3D) [m]','FontSize',12);
zlabel('Z (3D) [m]','FontSize',12);
title(['3D Cavitation Image using ',Cp_mode],'FontSize',16);
hold on
% ===== Plot forward propeller blade surfaces =====
for k=1:Z1

    surf(X3D(:,1:Np,1),Y3D(:,1:Np,k),Z3D(:,1:Np,k),...
        Color_matrix_upper(:, :, k));

end

for k=1:Z1

    surf(X3D(:,2*Np:-1:Np+1,1),Y3D(:,2*Np:-1:Np+1,k),...
        Z3D(:,2*Np:-1:Np+1,k),Color_matrix_lower(:, :, k));

end

% ===== Plot aft propeller blade surfaces =====
for k=1:Z2

    surf(X3D_aft(:,1:Np,1),Y3D_aft(:,1:Np,k),Z3D_aft(:,1:Np,k),...
        Color_matrix_upper_aft(:, :, k));

end

```

```

for k=1:Z2

    surf(X3D_aft(:,2*Np:-1:Np+1,1),Y3D_aft(:,2*Np:-1:Np+1,k),...
         Z3D_aft(:,2*Np:-1:Np+1,k),Color_matrix_lower_aft(:, :,k));
end
shading interp;

% Plot the hub using only one color
% =====
hub_clr=mean(caxis);
tick = 90:-15:0;
[yh0,zh0,xh0] = cylinder(Rhub*sind(tick),50);
xh0 = -0.5*c1(1)*xh0 - 0.75*R1;
surf(xh0,yh0,zh0,hub_clr*ones(7,51));

[yh1,zh1,xh1] = cylinder(Rhub,50);
xh1 = 6*c1(1)*xh1 - 0.75*R1;
surf(xh1,yh1,zh1,hub_clr*ones(2,51));
colorbar
text(0,0,R1,cav_mess1,'FontSize',15,'HorizontalAlignment','center')
text(0,0,-R1,'-C_p','FontSize',15,'HorizontalAlignment','center')
text(-Xf*R1,0,R2,cav_mess2,'FontSize',15,'HorizontalAlignment','center')
end

```

## **Cavitation.m**

%Written by Dimitrios Laskos

% Function Cavitation calculates pressure coefficients on blades' surfaces  
% and assigns colors depending on whether the values exceed cavitation  
% numbers (indicating cavitation inception) or not.

```

% =====Inputs=====
% Cp_mode          'VLM' or 'XFOIL' depending on which method is
%                  implemented for calculating pressure coefficients
% Mp              [ ],      Number of points over the span
% t0oc            [ ],      Maximum thickness / chord at each radius
% f0oc            [ ],      maximum camber ratio (f0/c=0.0679*C1)
% AlphaI          [deg],    Ideal angle of attack (AlphaI=1.54*C1)
% C1              [ ],      Lift coefficient (C1=Clideal)
% H               [m],      Shaft centerline depth
% Z3D             [m],      vertical location
% Vstar           [ ],      Total inflow velocity
% Np              [ ],      Number of points over the chord
% Z               [ ],      Blade number
% x0              [ ],      chordwise location [0:1]
% RC              [ ],      Non-dimensional radius for control points
% c               [m],      Chordlengths of blade sections along span
% R               [m],      Propeller radius
% theta_Z         [deg],    angle between blades
% BetaI_c         [deg],    Hydrodynamic pitch angle
% UASTAR          [ ],      Total axial induced velocity (self- and interaction-)
% =====

```

```

function[Color_matrix_upper,Color_matrix_lower,cav_mess]=...
    Cavitation(Cp_mode,Mp,t0oc,f0oc,AlphaI,C1,H,Z3D,Vstar,...

```

Np, Z, x0, RC, c, R, theta\_Z, BetaI\_c, UASTAR)

```
% ===== Execution of XFOIL =====
if strcmp(Cp_mode, 'XFOIL')
    xdir='.\Xfoil\';
    foil_type='LOAD'; % or 'NACA'
    foil_name='foildata';
    for i=1:Mp % for each section along the span
        makefoil(t0oc(i), f0oc(i), 'NACAa=08.txt', '65A010.txt', foil_name);
        cmd=[xdir, 'xfoil.exe', ' ', foil_type, ...
            ' ', foil_name, ' NORM ', ' GDES TSET ', num2str(t0oc(i)), ' ', ...
            num2str(f0oc(i)), ' ', 'GDES EXEC ', ' PANE', ' OPER ALFA ', ...
            num2str(AlphaI(i)), ' OPER CPWR ', ' ', 'CParray'];
        system(cmd)

        fid=fopen('CParray');
        datain=textscan(fid, '%f64 %f64', 'headerlines', 1);
        fclose(fid);
        Length=length(datain{1,2})
        if length(datain{1,2})>=160
            cpi{1,i}=datain{1,2};
            xcpi{1,i}=datain{1,1};
        else
            cmd=[xdir, 'xfoil.exe', ' ', foil_type, ...
                ' ', foil_name, 'NORM ', ' GDES TSET ', num2str(t0oc(i)), ' ', ...
                num2str(f0oc(i)), ' ', 'GDES EXEC ', ' OPER ALFA ', ...
                num2str(AlphaI(i)), ' OPER CPWR ', ' ', 'CParray'];
            system(cmd)

        end

        fid=fopen('CParray');
        datain=textscan(fid, '%f64 %f64', 'headerlines', 1);
        fclose(fid);
        cpi{1,i}=datain{1,2};
        xcpi{1,i}=datain{1,1};

    % Remove double values from xcpi arrays and keep only those appearing
    % first such that the interpolation routine doesn't crash
    n=length(xcpi{1,i})-1;
    Bpos=[];
    counter=0;
    for l=1:n
        if xcpi{1,i}(l)==xcpi{1,i}(l+1)
            counter=counter+1;
            Bpos(counter)=l+1;
        end
    end

    if isempty(Bpos)~=1
        counter1=0;
        ind_matrix=[];
        for l=1:n+1
```

```

        if (l~=Bpos)==1
            counterl=counterl+1;
            ind_matrix(counterl)=1;
        end
    end
    xcpi{1,i}=xcpi{1,i}(ind_matrix);
    cpi{1,i}=cpi{1,i}(ind_matrix);
end

%   plot Cp distribution for each section
%   figure;grid on;
%   plot(xcpi{1,i},-(cpi{1,i}));
%   title(['section # ',num2str(i)]);

end
for i=1:Mp;
for j=1:length(xcpi{1,i})-1
    xcpi_compare{1,i}(j,1)=xcpi{1,i}(j)-xcpi{1,i}(j+1);
end
end
% Indexing begins from TE (xcpi=1), goes to LE (xcpi=0)along upper side
% and returns to TE (xcpi=1) again along the lower foil side
for i=1:Mp
ind_upper{1,i}=find(xcpi_compare{1,i}>=0);
ind_upper{1,i}=[ind_upper{1,i};ind_upper{1,i}(end)+1];
ind_lower{1,i}=find(xcpi_compare{1,i}<0);
ind_lower{1,i}=ind_lower{1,i}+ones(length(ind_lower{1,i}),1);
xcpi_upper{1,i}=xcpi{1,i}(ind_upper{1,i});
xcpi_lower{1,i}=xcpi{1,i}(ind_lower{1,i});
cpi_upper{1,i}=cpi{1,i}(ind_upper{1,i});
cpi_lower{1,i}=cpi{1,i}(ind_lower{1,i});

end
% Interpolate to find Cp values at Np positions along the chord
for i=1:Mp
Cpi_upper(i,:)=pchip(xcpi_upper{1,i},cpi_upper{1,i},x0);
Cpi_lower(i,:)=pchip(xcpi_lower{1,i},cpi_lower{1,i},x0);
end
%===== End of XFOIL Execution =====
elseif strcmp(Cp_mode,'VLM')
    unsteady_flag=0;
    % =====Cp calculation using VLM code =====
    if unsteady_flag==0
        for i=1:Mp
            [xt, CPU(i,:), CPL(i,.)]=VLMcav(40, Cl(i),0,t0oc(i));
            Cpi_upper(i,:)=pchip(xt,-CPU(i,:),x0);
            Cpi_lower(i,:)=pchip(xt,-CPL(i,:),x0);
        end
    end
    % ===== Unsteady Cavitation Calculation =====
    if unsteady_flag==1
        load wake_030910 wake_full
        theta=[0:5:360];
        roR_wake=[0.2:0.05:1];
        [THETA,ROR_WAKE]=meshgrid(theta,roR_wake);
        [THETA_Z,R_C]=meshgrid(theta_Z(1:end-1),RC);

```

```

wake_int=interp2(THETA,ROR_WAKE,wake_full',THETA_Z,R_C);
VAC_wake=wake_int';
% ===== calculate VTSTAR =====
Vs=5; % change manually (could be added as a function input)
VTSTAR=(Vstar/Vs).*cosd(BetaI_c);
beta_wake=atand((ones(Z,1)*UASTAR+VAC_wake)./(ones(Z,1)*VTSTAR));

% ===== Now calculate CP using VLM =====
for k=1:Z
    for i=1:Mp
        delta_alpha(k,i)=beta_wake(k,i)-BetaI_c(i);% (a_ideal-alpha)
    end
end

    for k=1:Z
        for i=1:Mp
            [xt, CPU(i,:,k), CPL(i,:,k)]=VLMcav(40, Cl(i),...
                -delta_alpha(k,i),t0oc(i));
            Cpi_upper(i,:,k)=pchip(xt,-CPU(i,:,k),x0);
            Cpi_lower(i,:,k)=pchip(xt,-CPL(i,:,k),x0);
        end
    end
end
end
% =====End of Cp Calculation=====

%Accurate calculation of sigma for all blades
% variation of Z3D along section is taken into account
rho=1025;
for k=1:Z
    for i=1:Mp
        for j=1:Np
            SIGMA2(i,j,k)=(101000+rho*9.81*(H-Z3D(i,j,k))-2500)./...
                (rho*Vstar(i)^2/2); % cavitation matrix
        end
    end
end

% Check for cavitation on suction side
% =====
Cpi_upper_cmp=zeros(Mp,Np,Z);
if unsteady_flag==1
    for k=1:Z
        Cpi_upper_cmp(:,:,k)=Cpi_upper(:,:,k);
    end
elseif unsteady_flag==0
    for k=1:Z
        Cpi_upper_cmp(:,:,k)=Cpi_upper;
    end
end
Cav_matrix_upper=+(SIGMA2<-Cpi_upper_cmp);

% Check for cavitation on pressure side

```

```

% =====
Cpi_lower_cmp=zeros(Mp,Np,Z);
if unsteady_flag==1
    for k=1:Z
        Cpi_lower_cmp(:,:,k)=Cpi_lower(:,:,k);
    end
elseif unsteady_flag==0
    for k=1:Z
        Cpi_lower_cmp(:,:,k)=Cpi_lower;
    end
end
Cav_matrix_lower=+(SIGMA2<-Cpi_lower_cmp);

% figure;
% grid on;
% axis equal;
% axis([-R/2 R -1.1*R 1.1*R -1.1*R 1.1*R]);
% xlabel('X (3D) [m]','FontSize',12);
% ylabel('Y (3D) [m]','FontSize',12);
% zlabel('Z (3D) [m]','FontSize',12);
% title(['3D Cavitation Image using ',Cav_module],'FontSize',16);
% hold on

% Suction sides
% =====
for k=1:Z
    for i=1:Mp
        for j=1:Np
            if Cav_matrix_upper(i,j,k)==1
                Color_matrix_upper(i,j,k)=2*Cav_matrix_upper(i,j,k);
            else
                Color_matrix_upper(i,j,k)=Cpi_upper_cmp(i,j,k);
            end
        end
    end
end

% Pressure sides
% =====
for k=1:Z
    for i=1:Mp
        for j=1:Np
            if Cav_matrix_lower(i,j,k)==1
                Color_matrix_lower(i,j,k)=2*Cav_matrix_lower(i,j,k);
            else
                Color_matrix_lower(i,j,k)=Cpi_lower_cmp(i,j,k);
            end
        end
    end
end

% Print message
% =====
B_lower=+any(any(any(Cav_matrix_lower)));
B_upper=+any(any(any(Cav_matrix_upper)));

```

```

% B_cmp=+(B_lower==1)|(B_upper==1);
if (B_lower==1)|(B_upper==1)==1
    message='Cavitation present';
else
    message='No cavitation present'
end
% text(0,0,R,message,'FontSize',15);

Area=0;
for i=1:Mp-1
    RCdif(i)=RC(i+1)-RC(i);
    Rdif(i)=RCdif(i)*R;
    Area=Area+(c(i)+c(i+1))*Rdif(i)/2;
end

% =====Cavitating area=====
% All trapezoids forming the blade area of a specific section
% ((Mp-1) sections in total), have the same area since the length
% of the bases is the same and their height (Rdif) is common
cav_area_lower=0;
cav_area_upper=0;
for i=1:Mp %or from 2:Mp
    num_lower(i)=length(find(Cav_matrix_lower(i,:),1));
    num_upper(i)=length(find(Cav_matrix_upper(i,:),1));
    if num_lower(i)~=0
        cav_area_lower=cav_area_lower+num_lower(i)*((c(i)+c(i-1))/...
            (Np-1))*Rdif(i-1)/2;
    end
    if num_upper(i)~=0
        cav_area_upper=cav_area_upper+num_upper(i)*((c(i)+c(i-1))/...
            (Np-1))*Rdif(i-1)/2;
    end
end

perc_lower=100*cav_area_lower/Area;
perc_upper=100*cav_area_upper/Area;
message1=strcat('TDC face cavitation:',num2str(perc_lower),'%');
message2=strcat('TDC back cavitation:',num2str(perc_upper),'%');
cav_mess={message1;message2};

```

## **Geometry.m**

```

% =====
% ===== Determine Propeller Geometry Function
%
% This function determines the geometry of the CRP set. It outputs the
% geometry as a 3D image.
%
% Reference: J.S. Carlton, "Marine Propellers & Propulsion", ch. 3, 1994.
%
% -----
% Input Variables:
%
% filename          file name prefix for all output files
% Date_string       time and date to print on reports

```

```

% Make2Dplot_flag      flag for whether to make 2D geometry plot
% Make3Dplot_flag      flag for whether to make 3D geometry plot
% Make_Rhino_flag      flag for whetehr to make a Rhino output file
% Meanline             flag for choice of meanline form
% Thickness            flag for choice of thickness form
%
% XR                   [ ],      input radii / propeller radius
% t0oc0                [ ],      input thickness / chord at each radius
% skew0                [deg],    input skew          at each radius
% rake0                [ ],      input rake / diameter   at each radius
%
% RC                   [ ],      control point radii / propeller radius
% C1                   [ ],      section lift coefficients
% BetaI_c              [deg],    BetaI at the control points
% Xf                   [m]       Axial separation between propellers
% AlphaI               [deg],    ideal angle of attack
% Z                    [ ],      number of blades
% Rhub                 [m],      hub radius
% CoD                  [ ],      chord / diameter at each control point radius
% R                    [m],      propeller radius
% M                    [ ],      number of radial 2D cross-sections
% Np                   [ ],      number of points in each 2D section
%
% -----
function [f0oc1,f0oc2,t0oc1,t0oc2,AlphaI1,AlphaI2,X3D,Y3D,Z3D,...
        X3D_aft,Y3D_aft,Z3D_aft,c1,c2,x0_1,x0_2,theta_Z1,theta_Z2] = ...
        Geometry(XR1,XR2,t0oc01,t0oc02,skew01,skew02,rake01,...
                rake02,RC1,RC2,C11,C12,BetaI_c1,BetaI_c2,Xf,Z1,Z2,...
                Rhub,CoD1,CoD2,R1,R2,M1,M2,Np)
% ----- Interpolate input geometry at control points
f0oc1=0.0679*C11;           %max camber ratio (NACA a=0.8 meanline)
f0oc2=0.0679*C12;
t0oc1 = pchip(XR1,t0oc01,RC1);      % [ ],      thickness ratio
t0oc2 = pchip(XR2,t0oc02,RC2);
skew1 = pchip(XR1,skew01,RC1);      % [deg],    angular translation along
%                                     mid-chord helix
skew2 = pchip(XR2,skew02,RC2);
D1=2*R1;
D2=2*R2;
rake1 = pchip(XR1,rake01,RC1)*D1;   % [m],      translation along propeller
%                                     axis (3D X-axis)
rake2 = pchip(XR2,rake02,RC2)*D2;
AlphaI1=1.54*C11;
AlphaI2=1.54*C12;

% ----- Find basic geometry parameters chord, radius, pitch, etc.
theta_nt1 = BetaI_c1 + AlphaI1;     % Nose-tail pitch angle, [deg]
theta_nt2 = BetaI_c2 + AlphaI2;

PoD1      = tand(theta_nt1).*pi.*RC1; % Pitch / propeller diameter, [ ]
PoD2      = tand(theta_nt2).*pi.*RC2;
c1        = CoD1.*D1;                % section chord at the c. points [m]
c2        = CoD2.*D2;
r1        = RC1.*R1;                 % radius of the c. points [m]

```



```

r2      = RC2.*R2;
theta_z1 = 0:360/Z1:360;          % angle between blades [deg]
theta_z2 = 360/(2*Z2):360/Z2:360+360/(2*Z2); % angle between blades [deg]
%       or 0:360/Z2:360
% ----- Lay out the 2D coordinate system
%
% xN    [ ], x/c coordinate in 2D NACA foil tables
%       At the Leading Edge: xN = 0, x1 = c/2, x0 = 0
%       At the Trailing Edge: xN = 1, x1 = -c/2, x0 = 1
% x0    [ ], x/c distance along mid-chord line to interpolate NACA data.
% x1    [m], x distance along mid-chord line to evaluate elliptical or
%       parabolic formulae. By definition, x1 == c/2 - c*x0.
% x2D   [m], x position in 2D space on upper and lower foil surfaces
% y2D   [m], y position in 2D space on upper and lower foil surfaces
% x2Dr  [m], x position in 2D space after rotation for pitch angle
% y2Dr  [m], y position in 2D space after rotation for pitch angle
%
xN = [0 .5 .75 1.25 2.5 5 7.5 10 15 20 25 30 35 40 45 50 ...
      55 60 65 70 75 80 85 90 95 100]./100;

for i = 1:M1          % for each radial section along the span
    for j = 1:Np      % for each point along the chord
        x0_1(1,j)    = (j-1)/(Np-1); % [0 : 1]
        x1_1(i,j)    = c1(i)/2 - c1(i)*(j-1)/(Np-1); % [c/2 : -c/2]
    end
end

for i = 1:M2          % for each radial section along the span
    for j = 1:Np      % for each point along the chord
        x0_2(1,j)    = (j-1)/(Np-1); % [0 : 1]
        x1_2(i,j)    = c2(i)/2 - c2(i)*(j-1)/(Np-1); % [c/2 : -c/2]
    end
end

% ----- Find meanline and thickness profiles (at x1 positions)
%
% foc    = camber / chord ratio (NACA data at xN positions)
% dfdxN  = slope of camber line (NACA data at xN positions)
% fscale = scale to set max camber ratio to f0oc for each section
% tscale = scale to set max thickness ratio to t0oc for each section
% f       = camber at x1 positions
% dfdx   = slope of camber line at x1 positions
% t       = thickness at x1 positions

% ----- Use NACA a=0.8 meanline
foc = [0 .287 .404 .616 1.077 1.841 2.483 3.043 3.985 4.748 ...
      5.367 5.863 6.248 6.528 6.709 6.79 6.77 6.644 6.405 ...
      6.037 5.514 4.771 3.683 2.435 1.163 0]./100;

dfdxN = [.48535 .44925 .40359 .34104 .27718 .23868 .21050 ...
        .16892 .13734 .11101 .08775 .06634 .04601 .02613 ...
        .00620 -.01433 -.03611 -.06010 -.08790 -.12311 ...
        -.18412 -.23921 -.25583 -.24904 -.20385];

```

```

fscale1 = f0oc1 / max(foc);
fscale2 = f0oc2 / max(foc);

for i = 1:M1
    for j = 1:Np
        f1(i,:) = pchip(xN ,foc .*fscale1(i).*c1(i),x0_1);
        dfdx1(i,:) = pchip(xN(2:end),dfdxN.*fscale1(i) ,x0_1);
    end
end

for i = 1:M2
    for j = 1:Np
        f2(i,:) = pchip(xN ,foc .*fscale2(i).*c2(i),x0_2);
        dfdx2(i,:) = pchip(xN(2:end),dfdxN.*fscale2(i) ,x0_2);
    end
end

%this is for NACA 66mod with t0/c=0.1
toc_66 = [0 .665 .812 1.044 1.466 2.066 2.525 2.907 3.521 4 ...
          4.363 4.637 4.832 4.952 5 4.962 4.846 4.653 ...
          4.383 4.035 3.612 3.11 2.532 1.877 1.433 .333]./100;
%this is for the NACA 65A010
%   toc_65 = [0 .765 .928 1.183 1.623 2.182 2.65 3.04 3.658 4.127 ...
%             4.483 4.742 4.912 4.995 4.983 4.863 4.632 4.304 ...
%             3.899 3.432 2.912 2.352 1.771 1.188 .604 .021]./100;

tscale1 = t0oc1 / max(toc_66);
tscale2 = t0oc2 / max(toc_66);

for i = 1:M1
    for j = 1:Np
        t1(i,:) = pchip(xN,toc_66.*tscale1(i).*c1(i),x0_1);
    end
end

for i = 1:M2
    for j = 1:Np
        t2(i,:) = pchip(xN,toc_66.*tscale2(i).*c2(i),x0_2);
    end
end

% ----- Find 2D unroated section profiles
% x2D [m], x position in 2D space on upper (x2D_u) and lower (x2D_l) surf.
% y2D [m], y position in 2D space on upper (y2D_u) and lower (y2D_l) surf.
for i = 1:M1 % for each section along the span
    for j = 1:Np % for each point along the chord
        x2D_u(i,j) = x1_1(i,j) + (t1(i,j)/2)*sin(atan(dfdx1(i,j)));
        x2D_l(i,j) = x1_1(i,j) - (t1(i,j)/2)*sin(atan(dfdx1(i,j)));
        y2D_u(i,j) = f1(i,j) + (t1(i,j)/2)*cos(atan(dfdx1(i,j)));
        y2D_l(i,j) = f1(i,j) - (t1(i,j)/2)*cos(atan(dfdx1(i,j)));
    end
end

for i = 1:M2 % for each section along the span
    for j = 1:Np % for each point along the chord

```

```

x2D_u_aft(i,j) = x1_2(i,j) + (t2(i,j)/2)*sin(atan(dfdx2(i,j)));
x2D_l_aft(i,j) = x1_2(i,j) - (t2(i,j)/2)*sin(atan(dfdx2(i,j)));
% ----- For aft propeller signs are reversed-----
y2D_u_aft(i,j) = -f2(i,j) - (t2(i,j)/2)*cos(atan(dfdx2(i,j)));
y2D_l_aft(i,j) = -f2(i,j) + (t2(i,j)/2)*cos(atan(dfdx2(i,j)));
end
end

% -----Put all the numbers in one list-----
% First Np values are the upper surface (suction side),and the second Np
% values are the lower surface (pressure side).
x2D(:, 1:Np) = x2D_u(:,1:Np);
x2D(:,1+Np:Np+Np) = x2D_l(:,Np:-1:1);
y2D(:, 1:Np) = y2D_u(:,1:Np);
y2D(:,1+Np:Np+Np) = y2D_l(:,Np:-1:1);

% ----- Put all the numbers in one list for aft prop
x2D_aft(:, 1:Np) = x2D_u_aft(:,1:Np);
x2D_aft(:,1+Np:Np+Np) = x2D_l_aft(:,Np:-1:1);
y2D_aft(:, 1:Np) = y2D_u_aft(:,1:Np);
y2D_aft(:,1+Np:Np+Np) = y2D_l_aft(:,Np:-1:1);

% ----- Find 2D rotated section profiles
% x2Dr [m], x position in 2D space after rotation for pitch angle
% y2Dr [m], y position in 2D space after rotation for pitch angle
for i = 1:M1 % for each section along the span
    for j = 1:2*Np % for each point along the upper and lower surfaces
        x2Dr(i,j) = x2D(i,j)*cosd(theta_nt1(i))...
            - y2D(i,j)*sind(theta_nt1(i)); % rotated 2D upper surface x
        y2Dr(i,j) = x2D(i,j)*sind(theta_nt1(i))...
            + y2D(i,j)*cosd(theta_nt1(i)); % rotated 2D upper surface y
    end
end

% ----- Find 2D rotated section profiles for aft prop
theta_nt_aft=180-theta_nt2;
for i = 1:M2 % for each section along the span
    for j = 1:2*Np % for each point along the upper and lower surfaces
        x2Dr_aft(i,j) = x2D_aft(i,j)*cosd(theta_nt_aft(i))...
            - y2D_aft(i,j)*sind(theta_nt_aft(i)); % rotated upper surface x
        y2Dr_aft(i,j) = x2D_aft(i,j)*sind(theta_nt_aft(i))...
            + y2D_aft(i,j)*cosd(theta_nt_aft(i)); % rotated upper surface y
    end
end

% ----- Invoke skew and rake, and find 3D coordinates
% X3D [m], X position in 3D space (corresponds to y position in 2D space)
% Y2D [m], Y position in 3D space
% Z3D [m], Z position in 3D space

for i = 1:M1 % for each section along the span
    for j = 1:2*Np % for each point along the upper and lower surfaces
        X3D(i,j,1) = - raket1(i) ...
            - r1(i)*(pi*skew1(i)/180)*tand(theta_nt1(i)) + y2Dr(i,j);
    end
end

```

```

    for k = 1:Z1 % for each blade
        Y3D(i,j,k) = r1(i)*sind(skew1(i)...
            - (180/pi)*x2Dr(i,j)/r1(i) - theta_Z1(k));
        Z3D(i,j,k) = r1(i)*cosd(skew1(i)...
            - (180/pi)*x2Dr(i,j)/r1(i) - theta_Z1(k));
    end
end
end
% ----- Invoke skew and rake, and find 3D coordinates for aft prop

for i = 1:M2 % for each section along the span
    for j = 1:2*Np % for each point along the upper and lower surfaces
        X3D_aft(i,j,1) = - rake2(i) - ...
            r2(i)*(pi*skew2(i)/180)*tand(theta_nt_aft(i)) + y2Dr_aft(i,j);
        X3D_aft(i,j,1) = X3D_aft(i,j,1)-Xf*R1;
        for k = 1:Z2 % for each blade
            Y3D_aft(i,j,k) = r2(i)*sind(skew2(i) ...
                - (180/pi)*x2Dr_aft(i,j)/r2(i) - theta_Z2(k));
            Z3D_aft(i,j,k) = r2(i)*cosd(skew2(i)...
                - (180/pi)*x2Dr_aft(i,j)/r2(i) - theta_Z2(k));
        end
    end
end
end
% ----- Create 3D Propeller Image

Fig3_S = figure('units','normalized','position',[.61 .06 .4 .3],...
    'name','Propeller Image','numbertitle','off');
hold on;

% ----- Plot the propeller surface
for k = 1:Z1
    surf(X3D(:,:,1),Y3D(:,:,k),Z3D(:,:,k));
end

for k = 1:Z2
    surf(X3D_aft(:,:,1),Y3D_aft(:,:,k),Z3D_aft(:,:,k));
end

colormap gray;
shading interp;
% shading faceted;
grid on;
axis equal;
axis([-2*Xf*R1 R1 -1.1*R1 1.1*R1 -1.1*R1 1.1*R1]);
xlabel('X (3D) [m]','FontSize',12);
ylabel('Y (3D) [m]','FontSize',12);
zlabel('Z (3D) [m]','FontSize',12);
title('3D Propeller Image','FontSize',16);

% ----- Plot the hub
tick = 90:-15:0;
[yh0,zh0,xh0] = cylinder(Rhub*sind(tick),50);
xh0 = -0.5*c1(1)*xh0 - 0.75*R1;
surf(xh0,yh0,zh0);

```

```

[yh1,zh1,xh1] = cylinder(Rhub,50);
xh1 = 6*c1(1)*xh1 - 0.75*R1;
surf(xh1,yh1,zh1);

% ----- Plot the suction side (green) & pressure side (red) of the prop
for i = 1:M1 % for each section along the span
    for k = 1:Z1 % for each blade
        plot3(X3D(i,1:Np,1),Y3D(i,1:Np,k),Z3D(i,1:Np,k),...
            'g','Linewidth',1); % suction surface
        plot3(X3D(i,Np+1:2*Np,1),Y3D(i,Np+1:2*Np,k),...
            Z3D(i,Np+1:2*Np,k),'r','Linewidth',1); % pressure surface
    end
end

for i = 1:M2 % for each section along the span
    for k = 1:Z2 % for each blade
        % Now for aft prop
        plot3(X3D_aft(i,1:Np,1),Y3D_aft(i,1:Np,k),Z3D_aft(i,1:Np,k),...
            'g','Linewidth',1); % suction surface
        plot3(X3D_aft(i,Np+1:2*Np,1),Y3D_aft(i,Np+1:2*Np,k),...
            Z3D_aft(i,Np+1:2*Np,k),'r','Linewidth',1);%pressure surface
    end
end

for j = 1:Np % for each point along the chord
    for k = 1:Z1 % for each blade
        plot3(X3D(:,j,1),Y3D(:,j,k),Z3D(:,j,k),...
            'g','Linewidth',1); % suction surface
        plot3(X3D(:,j+Np,1),Y3D(:,j+Np,k),Z3D(:,j+Np,k),...
            'r','Linewidth',1); % pressure surface
    end
end

for j = 1:Np % for each point along the chord
    for k = 1:Z2 % for each blade
        % Now for aft prop
        plot3(X3D_aft(:,j,1),Y3D_aft(:,j,k),Z3D_aft(:,j,k),...
            'g','Linewidth',1); % suction surface
        plot3(X3D_aft(:,j+Np,1),Y3D_aft(:,j+Np,k),...
            Z3D_aft(:,j+Np,k),'r','Linewidth',1); % pressure surface
    end
end

% ----- Plot the leading and trailing edges
for k = 1:Z1 % for each blade
    plot3(X3D(:,1,1), Y3D(:,1,k), Z3D(:,1,k), 'b','Linewidth',2); %L.E.
    plot3(X3D(:,Np,1),Y3D(:,Np,k),Z3D(:,Np,k),'k','Linewidth',2); %T.E.
end

for k = 1:Z2 % for each blade
    plot3(X3D_aft(:,1,1),Y3D_aft(:,1,k),Z3D_aft(:,1,k),...
        'b','Linewidth',2); %L.E.
    plot3(X3D_aft(:,Np,1),Y3D_aft(:,Np,k),Z3D_aft(:,Np,k),...
        'k','Linewidth',2); %T.E.
end

```

```

end

% ----- Plot the coordinate system

% Axes
plot3([0 R1],[0 0],[0 0],'y','LineWidth',2),
plot3([0 0],[0 R1],[0 0],'y','LineWidth',2),
plot3([0 0],[0 0],[0 R1],'y','LineWidth',2),

% Circle at the X = 0 location on the hub
phi = 0:0.01:2*pi;
Xhc = zeros(size(phi));
Yhc = - Rhub * sin(phi);
Zhc = Rhub * cos(phi);
plot3(Xhc,Yhc,Zhc,'y','LineWidth',2),
% Circle at the X = -Xf*R1/2 location on the hub
Xhc_mid = -(Xf/2)*R1*ones(size(phi));
plot3(Xhc_mid,Yhc,Zhc,'k','LineWidth',4)

```

## **Forces.m**

```

% This function computes the thrust - torque coefficients, and it computes
% the total efficiency of the CRP set, Kerwin eqns 161-162, p.138, and
% eqns 196-197, p. 152, Coney eq. 2-65, p.45 with the inclusion of hub drag
%
% -----
% indices 1,2 refer to the forward and the aft propellers respectively
% Input Variables:
% CD      [ ],      section drag coefficient
% RV      [ ],      radius of vortex point / propeller radius
% VAC     [ ],      axial inflow velocity at c. points / ship velocity
% TANBC   [ ],      tangent of beta at the control points
% UASTAR  [ ],      axial induced velocity / ship velocity
% UTSTAR  [ ],      tangential induced velocity / ship velocity
% CoD     [ ],      section chord length / propeller diameter
% G       [ ],      circulation / (2*pi * prop radius * ship velocity)
% RC      [ ],      radius of control point / propeller radius
% Fh      [N],      Hub drag
% Z       [ ],      number of blades
% Js      [ ],      advance coefficient
% VMIV    [ ],      Volumetric Mean Inflow Velocity / ship velocity
% N       [RPM],   Propeller speed
% Vs      [m/s],   Ship speed
% R       [m],     Propeller radius

%
% Output variables:
% CT      [ ],     thrust coefficient, eqn (161) p.138
% CQ      [ ],     torque coefficient, eqn (161) p.138
% CP      [ ],     power coefficient based on torque
% KT      [ ],     thrust coefficient, eqn (162) p.138
% KQ      [ ],     torque coefficient, eqn (162) p.138

```

```

% EFFY      [ ],      total efficiency of the CRP set
% VSTAR     [ ],      total inflow velocity / ship velocity
%
% -----

function [CT1,CQ1,KT1,KQ1,CT2,CQ2,KT2,KQ2,EFFY,VSTAR1,VSTAR2] =...
    Forces (CD,DR1,DR2,VAC1,VAC2,TANBC1,TANBC2,...
        UASTAR1,UASTAR2,UTSTAR1,UTSTAR2,CoD1,CoD2,G1,G2,M1,M2,RC1,RC2,...
        Fh,Z1,Z2,Js1,Js2,VMIV1,VMIV2,N1,N2,Vs,R1,R2)

VASTAR1     = VAC1          + UASTAR1;          % total axial vel. / ship vel.
VASTAR2     = VAC2          + UASTAR2;
VTSTAR1     = VAC1./TANBC1 + UTSTAR1;          % total tang. vel. / ship vel.
VTSTAR2     = VAC2./TANBC2 + UTSTAR2;
VSTAR1      = sqrt (VTSTAR1.^2 + VASTAR1.^2); % total inflow vel. / ship vel.
VSTAR2      = sqrt (VTSTAR2.^2 + VASTAR2.^2);

sin_BetaI1  = VASTAR1./VSTAR1;
sin_BetaI2  = VASTAR2./VSTAR2;
cos_BetaI1  = VTSTAR1./VSTAR1;
cos_BetaI2  = VTSTAR2./VSTAR2;
if CD < 1
    DVISC1   = VSTAR1.^2.*CoD1.*CD/(2*pi);      % normalized viscous drag force
    DVISC2   = VSTAR2.^2.*CoD2.*CD/(2*pi);
else
    DVISC1   = VSTAR1.*G1./CD;                  % CD > 1 means the input is L/D (legacy code)
    DVISC2   = VSTAR2.*G2./CD;
end

% ----- Compute CT and CQ, Kerwin eqns. (196-197), p. 152
CT1 = 0;
CQ1 = 0;
CT2 = 0;
CQ2 = 0;

for m=1:M1
    CT1 = CT1 + ...
        (VSTAR1(m)*G1(m)*cos_BetaI1(m)-DVISC1(m)*sin_BetaI1(m))*DR1(m);
    CQ1 = CQ1 + ...
        (VSTAR1(m)*G1(m)*sin_BetaI1(m)+DVISC1(m)*cos_BetaI1(m))*RC1(m)*DR1(m);
end
for m=1:M2
    CT2 = CT2 + ...
        (VSTAR2(m)*G2(m)*cos_BetaI2(m)-DVISC2(m)*sin_BetaI2(m))*DR2(m);
    CQ2 = CQ2 + ...
        (VSTAR2(m)*G2(m)*sin_BetaI2(m)+DVISC2(m)*cos_BetaI2(m))*RC2(m)*DR2(m);
end

CT1 = CT1*4*Z1;          % eqn 196, p.152 (w/ addition for CTD)
CQ1 = CQ1*4*Z1;          % eqn 197, p.152
%CP1 = CQ1*pi/Js1;       % power coefficient based on torque
KT1 = CT1*Js1^2*pi/8;    % eqn 167, p.139
KQ1 = CQ1*Js1^2*pi/16;   % eqn 167, p.139
%EFFY1 = CT1*VMIV1/CP1;  % efficiency

```

```

CT2    = CT2*4*Z2;           % eqn 196, p.152 (w/ addition for CTD)
CQ2    = CQ2*4*Z2;           % eqn 197, p.152
%CP2   = CQ2*pi/Js2;         % power coefficient based on torque
KT2    = CT2*Js2^2*pi/8;     % eqn 167, p.139
KQ2    = CQ2*Js2^2*pi/16;    % eqn 167, p.139
%EFFY2 = CT2*VMIV2/CP2;      % efficiency

%----- Include effect of hub drag on efficiency -----
T1= CT1*(.5*1025*Vs^2*pi*R1^2);
T2= CT2*(.5*1025*Vs^2*pi*R2^2);
Q1= CQ1*(.5*1025*Vs^2*pi*R1^3);
Q2= CQ2*(.5*1025*Vs^2*pi*R2^3);
EFFY=Vs*(VMIV1*T1+VMIV2*T2-Fh)/((2*pi/60)*(N1*Q1+N2*Q2)); %total efficiency
% Coney, eq. 2-65, p.45

%
% ===== END Forces Function
% =====

```

## Align Wake.m

```

% =====
% ===== Align_wake Function
%
% This function aligns the wake to the given circulation distribution by
% iteratively computing:
%   UAHIF1,UTHIF1,UAHIF2,UTHIF2           = the horseshoe influence functions
%                                           for the self-induced velocities
%   UAHIF1_2,UTHIF1_2,UAHIF2_1,UTHIF2_2 = the horseshoe influence functions
%                                           for the interaction velocities
%   UASTAR1,UTSTAR1,UASTAR2,UTSTAR2      = the induced velocities
%   TANBIC1,TANBIV1,TANBIC2,TANBIV2      = the velocity angles
%
% -----
function [UAHIF1,UTHIF1,UAHIF2,UTHIF2,UAHIF1_2,UTHIF1_2,UAHIF2_1,...
    UTHIF2_1,UASTAR1,UTSTAR1,UASTAR2,UTSTAR2,TANBIC1,TANBIV1,TANBIC2,...
    TANBIV2] = Align_wake(TANBIC1,TANBIV1,TANBIC2,TANBIV2,ITER,M1,M2,...
    Z1,Z2,RC1,RV1,RC2,RV2,G1,G2,VAC1,VTC1,VAC2,VTC2,Js1,Js2,Xf,Hub_Flag,...
    Rhub_or1,Rhub_or2)

% ----- Iterate to ALIGN WAKE to the new circulation distribution
W_iter = 1;           % iteration in the wake alignment loop
W_res1  = 1;          % residual BetaI between interations
W_res2  = 1;
TANBIW1_last = TANBIC1; % the last value of TANBIC
TANBIW2_last = TANBIC2;

while W_iter < ITER & (W_res1 > 1e-5 | W_res2 > 1e-5 )%(WHILE LOOP WA1)

    % ----- Compute the vortex Horseshoe Influence Functions -----

```



```

[UAHIF1,UTHIF1]=Horseshoe_self(M1,Z1,TANBIV1,RC1,RV1,Hub_Flag,...
                                Rhub_or1);
[UAHIF1_2,UTHIF1_2]=Horseshoe_int(M1,M2,Z2,TANBIV2,RC1,RV2,-Xf,...
                                Hub_Flag,Rhub_or2);

[UAHIF2,UTHIF2]=Horseshoe_self(M2,Z2,TANBIV2,RC2,RV2,Hub_Flag,...
                                Rhub_or2);
[UAHIF2_1,UTHIF2_1]=Horseshoe_int(M2,M1,Z1,TANBIV1,RC2,RV1,Xf,...
                                Hub_Flag,Rhub_or1);

% ---- Compute induced velocities at control points. eqn 254, p.179
%[UASTAR,UTSTAR] = Induced_Velocity(Mp,G,UAHIF,UTHIF,UADUCT,dCirc);

[UA_SELF1,UT_SELF1,UA_INT1_2,UT_INT1_2]=Induced_Velocity(M1,M2,...
                                G1,G2,UAHIF1,UTHIF1,UAHIF1_2,UTHIF1_2);
UASTAR1=UA_SELF1+UA_INT1_2; %total axial induced velocity/Vs
UTSTAR1=UT_SELF1+UT_INT1_2; %total tangential induced velocity/Vs

[UA_SELF2,UT_SELF2,UA_INT2_1,UT_INT2_1]=Induced_Velocity(M2,M1,...
                                G2,G1,UAHIF2,UTHIF2,UAHIF2_1,UTHIF2_1);
UASTAR2=UA_SELF2+UA_INT2_1; %total axial induced velocity/Vs
UTSTAR2=UT_SELF2+UT_INT2_1; %total tangential induced velocity/Vs

% ----- Compute tan(BetaI) for the new induced velocities
[TANBIC1,TANBIV1] = find_tan_BetaI(VAC1,VTC1,UASTAR1,UTSTAR1,...
                                RC1,RV1,Js1);
[TANBIC2,TANBIV2] = find_tan_BetaI(VAC2,VTC2,UASTAR2,UTSTAR2,...
                                RC2,RV2,Js2);

% ----- Prepare for the next iteration
W_iter = W_iter + 1 % iteration in the BetaI loop
W_res1 = abs(TANBIC1 - TANBIW1_last); % residual BetaI
W_res2 = abs(TANBIC2 - TANBIW2_last);
TANBIW1_last = TANBIC1; % the last value of TANBIC
TANBIW2_last = TANBIC2;

if W_iter > ITER
    warning('on'),
    warning('WARNING: While loop WA1 did NOT converge.'),
    warning('off'),
end
end % (END WHILE LOOP WA1)

% ===== END Align_wake Function
% =====

```

## Horseshoe\_int.m

```

% This function computes the vortex horseshoe axial and tangential
% interaction influence functions UAHIF_int and UTHIF_int respectively
% UAHIF_int(n,m)=influence of mth horseshoe vortex shed from one propulsor
% component (Mact panels) on nth control point of the other component
% (Mpas panels).

function [UAHIF_int,UTHIF_int]=Horseshoe_int(Mpas,Mact,Zact,TANBIVact,...
                                             RCpas,RVact,Xf,Hub_Flag,Rhub_oRact)
UAHIF_int=zeros(Mpas,Mact);
UTHIF_int=zeros(Mpas,Mact);
for n=1:Mpas
    for m=1:Mact+1
        [UAHough(m)]=Hough(Zact,Xf,TANBIVact(m),RCpas(n),RVact(m));
        if Hub_Flag == 1
            RCW      = RCpas(n);
            RVW      = Rhub_oRact^2/RVact(m);
            TANBIW   = TANBIVact(m)*RVact(m)/RVW;

            [UAHough_h] = Hough(Zact,Xf,TANBIW,RCW,RVW);

            UAHough(m) = UAHough(m)+UAHough_h;
        end
    end

    for m=1:Mact
        UAHIF_int(n,m)=UAHough(m+1)-UAHough(m);
        S=(RVact(m)-RCpas(n))*(RVact(m+1)-RCpas(n));
        if S<0 && Xf>0
            UTHIF_int(n,m)=Zact/RCpas(n);
        else
            UTHIF_int(n,m)=0;
        end
    end
end
end

```

## **Hough.m**

```

% Hough function
% Returns circumferential mean axial 'induction factor'
% Xf:axial distance between propulsors in terms of R (Xf=distance/R)
% Xf:positive for downstream, negative for upstream.

function [UA_Hough]=Hough(Z,Xf,tanbi,rc,rv)
q=1+(Xf^2+(rc-rv)^2)/(2*rc*rv);
s=asin(Xf/sqrt(Xf^2+(rc-rv)^2));
t=sqrt(4*rc*rv/(Xf^2+(rc+rv)^2));
%amplitude wrt elliptical integrals
%t=k (modulus wrt elliptical integrals)
if rc>rv
    C1= Xf/(2*sqrt(rc*rv))*Q2Mhalf(q)-pi/2*Heuman(s,asin(t));
else
    C1=pi+Xf/(2*sqrt(rc*rv))*Q2Mhalf(q)+pi/2*Heuman(s,asin(t));
end

```

```
UA_Hough=Z*C1/(2*pi*rv*tanbi);
```

## Heuman.m

```
function [H] = Heuman(phi,alpha)
[K,E]=ellipke(sin(alpha)^2);
F=mfun('EllipticF',sin(phi),sin(pi/2-alpha)); %Incomplete elliptic
% integral, 1st kind
EE=mfun('EllipticE',sin(phi),sin(pi/2-alpha)); %Incomplete elliptic
% Integral, 2nd kind
H=2/pi*(K*EE-(K-E)*F);
```

## Horseshoe\_self.m

```
function [UAHIF,UTHIF] = Horseshoe_self(Mp,Z,TANBIV,RC,RV,Hub_Flag,...
                                         Rhub_oR)

for n = 1:Mp % for each control point, n (FOR LOOP MF2)
  for m = 1:Mp+1 % for each vortex point, m (FOR LOOP MF3)

    % -- Find induction factors for a unit vortex shed at RV(m)
    % -- Wrench returns 2*pi*R*u_bar
    [UAW(m),UTW(m)] = Wrench(Z,TANBIV(m),RC(n),RV(m));

    % ----- Find hub-image effects, Kerwin p.181
    if Hub_Flag == 1
      RCW = RC(n);
      RVW = Rhub_oR^2/RV(m);
      TANBIW = TANBIV(1)*RV(1)/RVW;

      [UAWH,UTWH] = Wrench(Z,TANBIW,RCW,RVW);

      UAW(m) = UAW(m)-UAWH;
      UTW(m) = UTW(m)-UTWH;
    end
  end % (END FOR LOOP MF3)

  % The Horseshoe Influence Function for vortex panel m is the
  % effect of the induction by a helical trailing vortex at
  % vortex point m with circulation -Gamma(m) and another at
  % vortex point m+1 with circulation +Gamma(m).
  % UAHIF(n,m) = u_barA horseshoe influence function in eqn 254.
  % UAW(m) = u_barA Wrench velocity given in eqn 202-203.
  for m = 1:Mp % for each vortex panel, m
    UAHIF(n,m) = UAW(m+1)-UAW(m); % 2*pi*R*(HIF)
    UTHIF(n,m) = UTW(m+1)-UTW(m); % 2*pi*R*(HIF)
  end

end % (END FOR LOOP MF2)
end
```

```

% ===== END Horseshoe Function
% =====

% ===== Wrench Function
%
% This function evaluates the Wrench  $\bar{u}$  velocity induced on a point on
% a lifting line due to a helical trailing vortex. These formulae were
% derived in 1957 by J.W. Wrench. This function returns  $\bar{u}$  given in
% Kerwin eqns 202-205, p.154.
%
% NOTE: There are TWO ERRORS in Kerwin, as of the Spring 2007 printing.
%       These have been corrected in the present implementation.
%
% 1. Eqn 202,  $\bar{u}_a$ . Should be  $(y-2*Z*y*y0*F1)$ , not  $(y-2*Z*rv*F1)$  to
%    agree with Wrench, eqn 31.
% 2. Eqn 204,  $F2$ . Need to kill the leading "-" sign to make  $F2$  agree
%    with Wrench equation 29.
%
% -----
% Variables:
%   Z          [ ],      number of blades
%   tan_betaW [ ],      tangent of the pitch angle of helical wake trail
%   rc         [ ],      radius of control point / propeller radius
%   rv         [ ],      radius of helical vortex / propeller radius
%
%   u_barA     [ ],      Wrench  $\bar{u}$  velocity in the axial direction
%   u_barT     [ ],      Wrench  $\bar{u}$  velocity in the tangential direction
%   y,y0,U,F1,F2,      auxilary variables. See Kerwin eqns. 202-205.
%
% -----

function [u_barA, u_barT] = Wrench(Z,tan_betaW,rc,rv)

%   % ----- Enable this to check for infinite bladed propellers
%   if Z > 20 % Return infinite blade result if Z > 20.
%       if rc = rv
%           IF_A = 0;
%           IF_T = 0;
%
%       elseif rc < rv
%           IF_A = Z/(2*rv*tan_betaW); % 2*pi*R*(eqn 206)
%           IF_T = 0; % 2*pi*R*(eqn 206)
%
%       else % rc > rv
%           IF_A = 0; % 2*pi*R*(eqn 207)
%           IF_T = Z/(2*rc); % 2*pi*R*(eqn 207)
%       end
%       return;
%   end

y = rc/(rv*tan_betaW);
y0 = 1/tan_betaW;

```

```

U = ((y0*(sqrt(1+y ^2)-1))*exp(sqrt(1+y^2)-sqrt(1+y0^2)))/...
      (y *(sqrt(1+y0^2)-1))^Z;

if rc == rv
    IF_A = 0;
    IF_T = 0;

elseif rc < rv

    F1      = -(1/(2*Z*y0)) * ((1+y0^2)/(1+y^2))^0.25 * ...
              ((1/(U^-1-1)) + (1/(24*Z)))*...
              (((9*y0^2+2)/(1+y0^2)^1.5)+((3*y^2-2)/(1+y^2)^1.5))*...
              log(1+(1/(U^-1-1))) );

    u_barA = (Z / (2*rc)) * (y-2*Z*y*y0*F1);          % 2*pi*R*(eqn 202)
    u_barT = (Z^2/rc)      * (y0*F1);                % 2*pi*R*(eqn 202)

else % rc > rv

    F2      = (1/(2*Z*y0)) * ((1+y0^2)/(1+y^2))^0.25 * ...
              ((1/(U-1))      - (1/(24*Z)))*...
              (((9*y0^2+2)/(1+y0^2)^1.5)+((3*y^2-2)/(1+y^2)^1.5))*...
              log(1+(1/(U-1))) );

    u_barA = -(Z^2/rc)      * (y*y0*F2);          % 2*pi*R*(eqn 203)
    u_barT = (Z / (2*rc)) * (1+2*Z*y0*F2);        % 2*pi*R*(eqn 203)

end
end
% ===== END Wrench Function
% =====

```

## **Induced Velocity.m**

```

% =====
% ===== Induced_Velocity Function
%
% This function computes induced velocities at control points, Kerwin
% eqn 254, p.179, normalized by the ship speed.
% The self-induced velocities are assumed to be those having an index of 1,
% while the interaction are those induced by component 2 on component 1
% having an index of 1_2.
% -----

function [UA_SELF,UT_SELF,UA_INT,UT_INT] = Induced_Velocity(M1,M2,G1,G2,...
    UAHIF1,UTHIF1,UAHIF1_2,UTHIF1_2)

UA_SELF(1:M1) = 0;
UT_SELF(1:M1) = 0;
UA_INT(1:M1)  = 0;
UT_INT(1:M1)  = 0;

```

```

for n = 1:M1 % for each control point, n
  for m = 1:M1 % for each vortex panel, m
    UA_SELF(n) = UA_SELF(n) + G1(m)*UAHIF1(n,m); % UASTAR / ship speed
    UT_SELF(n) = UT_SELF(n) + G1(m)*UTHIF1(n,m); % UASTAR / ship speed
  end
  for m=1:M2
    UA_INT(n) = UA_INT(n) + G2(m)*UAHIF1_2(n,m);
    UT_INT(n) = UT_INT(n) + G2(m)*UTHIF1_2(n,m);
  end
end
end
% ===== END Induced_Velocity Function
% =====

```

### **find tan BetaI.m**

```

% =====
% ===== find_tan_BetaI Function
%
% This function computes tan(BetaI), Kerwin eqn 193, p. 151.
% UASTAR, UTSTAR represent the total induced velocities
% (sum of self-induced and interaction velocities)
% -----
function [TANBIC,TANBIV] = find_tan_BetaI(VAC,VTC,UASTAR,UTSTAR,RC,RV,Js)

VASTAR = VAC + UASTAR; % total axial inflow vel. / ship vel.
VTSTAR = VTC + pi*RC/Js + UTSTAR; % total tangential inflow vel. / ship vel.

TANBIC = VASTAR./VTSTAR; % tan(BetaI) at control pts.
TANBIV = pchip(RC,TANBIC,RV); % tan(BetaI) at vortex pts.
end
% ===== END find_tan_BetaI Function
% =====

```

## **B3. Calculation of Pressure Distributions**

### **VLMcav.m**

```

% Last modified: MAY/01/10 by Dimitrios Laskos
% Original codes by Hsin-Lung Chung
% 2D Vortex/Source Lattice with Lighthill Correction Program (VLM)
% This file contains the algorithms for VLM.
function [xt, CPU, CPL, CLNum] = VLMcav(N,CL,Alpha,TOC);

```

```

% global N CL Alpha TOC;
% =====
U = 1;          c = 1;          % Free Stream Velocity and Chord Length
for i = 1:N
    xv(i) = c/2 * (1-cos((i-1/2)*pi/N));    % Vortex Position
    xc(i) = c/2 * (1-cos(i*pi/N));          % CP Position
    dx(i) = pi * sqrt(xv(i)*(c-xv(i))) / N; % Interval between vortices
end
for i = 1:N          % Influence Matrix A (i:CP; j:vortex)
    for j = 1:N
        A(i,j) = 1/(2*pi*(xv(j)-xc(i)));
    end
end
% ===== Camber Term
[B,F,Gexact] = MeanLine(xv,xc);    % Function for NACA a = 0.8 Mean Line
for i = 1:N
    B(i) = CL*B(i) - Alpha*pi/180;
    F(i) = CL*F(i);                % Camber F
end
Gamma = (B/A');                    % Point Vortex Strength
G = Gamma./dx;                      % Vortex Sheet Strength
CLNum = 2*sum(Gamma);               % Numerical Lift Coeff
% ===== Thickness Term
xt(1) = 0;                          % Thickness at the leading edge
for i = 1:length(xc)
    xt(i+1) = xc(i);
end

% =====
thick_toggle='NACA65A'; % or NACA66TMB
% =====

[RLE,yt,dydx] = Thickness(TOC, xt, thick_toggle);
for i = 1:N          % i for CP; j for Vortices
    for j = 1:N
        ut(i,j) = (yt(j+1)-yt(j))/(xc(i)-xv(j))/(2*pi);
    end
    UT(i) = sum(ut(i,:));          % UT @ Control Points
end
UTVP = spline(xc,UT,xv);          % UT @ Vortex Points
% ===== Leading Edge Surface Velocity
QU = Alpha*pi/180*sqrt(2*c/RLE);  % Surface Velocity
CPU(1) = QU^2-1;                  % Minus Cp on the upper surface at LE
CPL(1) = CPU(1);                  % Minus Cp on the lower surface at LE
% ===== Surface Velocity
for i = 1:N
    if dydx(i)>0
        FLH(i) = 1/sqrt(1+dydx(i)^2);
    else
        FLH(i) = 1;
    end
    QU(i) = (1+UT(i)+1/2*G(i))*FLH(i); % Velocity on Upper Surface
    CPU(i+1) = QU(i)^2-1;              % -Cp
    QL(i) = (1+UT(i)-1/2*G(i))*FLH(i); % Velocity on Lower Surface
    CPL(i+1) = QL(i)^2-1;              % -Cp
end

```

```

% ===== Plotting
plot_toggle='no'; %or 'yes'
if strcmp(plot_toggle,'yes')
    figure;
    plot(xt,CPU,'-r','LineWidth',2);        hold on;
    plot(xt,CPL,':b','LineWidth',2);        hold off;
    grid on;                                xlim([0 1]);    xlabel('X/C');    ylabel('-Cp');
    legend('Upper Surface','Lower Surface');
    % title(strcat('thickness form : ',thick_toggle))
    title(['thickness form ',thick_toggle])
end

% Former FORTRAN Subroutine "AEIGHT" ===== APR/27/07 by H.L. Chung
function [B,F,Gexact] = MeanLine(xv,xc)
a = 0.8; % For NACA a=0.8
MC = length(xv);
g = -1/(1-a) * (a^2*(log(a)/2-1/4)+1/4);
h = 1/(1-a) * ((1-a)^2*log(1-a)/2 - (1-a)^2/4) + g;
AlphaIdeal = -h / (2*pi*(a+1));
for i = 1:MC
    C1 = max(1- xv(i),1e-6);
    CA = a - xv(i);
    if (abs(CA)<1e-6)
        CA = CA+1e-5;
    end
    P = 1/2*CA^2*log(abs(CA)) - 1/2*C1^2*log(C1)+1/4*(C1^2-CA^2);
    F(i)=(P/(1-a)-xv(i)*log(xv(i))+g-h*xv(i))/(2*pi*(a+1))+C1*AlphaIdeal;
    if (xv(i)<=a)
        Gexact(i) = 1/(a+1);
    else
        Gexact(i) = 1/(a+1) * (1-xv(i))/(1-a);
    end
end
for j = 1:MC
    C1 = max(1-xc(j),1e-6);
    CA = a - xc(j);
    if (abs(CA)<1e-6)
        CA = CA+1e-5;
    end
    R = -(a-xc(j))*log(abs(CA)) - 1/2*CA+C1*log(C1)+1/2*C1;
    S = -1/2*C1+1/2*CA;
    T = -log(xc(j))-1-h;
    B(j) = ((R+S)/(1-a)+T)/(2*pi*(a+1)) - AlphaIdeal;
end

% Function for thickness =====
function[RLE,YT,DYDX] = Thickness(thk, xt, thick_toggle)
if strcmp(thick_toggle,'NACA66TMB')
    PC=[0.000, 0.010, 0.025, 0.050, 0.100, 0.200, 0.300, 0.400, 0.450,...
        0.500, 0.600, 0.700, 0.800, 0.900, 0.950, 0.975, 0.990, 1.000];
    THICK = [0.0000, 0.1870, 0.2932, 0.4132, 0.5814, 0.8000, 0.9274,...
            0.9904, 1.0000, 0.9917, 0.9256, 0.7934, 0.5950, 0.3306,...

```



```

        0.1736, 0.0888, 0.0360, 0.0000];
RLE_CONST = 0.448;
elseif strcmp(thick_toggle, 'NACA65A')
    PC = [0.000, 0.005, 0.0075, 0.0125, 0.0250, 0.05, 0.075, 0.1, 0.15, ...
        0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, ...
        0.75, 0.8, 0.85, 0.9, 0.95, 1];
    THICK = [0, 0.1556, 0.1879, 0.2387, 0.3265, 0.4379, 0.5311, 0.6094, ...
        0.7331, 0.8268, 0.8978, 0.9495, 0.9835, 1.0000, 0.9976, 0.9736, ...
        0.9272, 0.8612, 0.7803, 0.6868, 0.5828, 0.4709, 0.3548, 0.2382, ...
        0.1208, 0.0031];
    RLE_CONST = 0.654;
end
NT = length(xt);
RLE = RLE_CONST*thk^2;
PSQ = sqrt(PC);
TRLE = 2*sqrt(2*RLE_CONST);
XSQ = sqrt(xt);
YSPLN = spline(PSQ, THICK, XSQ);
YT = thk.*YSPLN;
for N=1:NT-1
    DYDX(N) = (YT(N+1)-YT(N))/(xt(N+1)-xt(N))/2;
end

```

## Required functions for execution of modified XFOIL code

### **makefoil.m**

```

%Code by Chris Peterson. Code will read in specified camber and thickness
% distributions and generate foil geometry file for XFOIL. Thickness and
% camber are scaled to t_set and f_set.
% Coordinates start at TE, go forward CCW along upper surfact to LE,
% and back to TE along lower surface.

```

```

function [] = makefoil(t_set, f_set, mean_type, thick_type, save_as)

%   clc; clear all; close all;
%   t_set      = 0.1;
%   f_set      = 0.08;
%   mean_type   = 'NACAa=08(Brockett).txt';
%   thick_type  = 'NACA66(Brockett).txt';
%   save_as    = 'brockett';

make_plot      = 'no'; %Generate plot toggle ('yes' or 'no')
N_parab_def    = 35;   %Number of points to make nose parabola. Fails at
                    %numbers < ~20
N_parab_eval   = 6;   %Number of points to include at the nose in
                    %data export;
N_surf_pts     = 80;  %Number of points along body to TE
                    %(not including LE)
                    %N_parab_pts + N_surf_pts must be < 150
fract          = 1-2/N_parab_eval; %Fraction of parabola to use from LE
                    %to 0.005.
                    %Max parabola point must be less than 0.005
                    %to prevent sharp cornder at 0.005.

```

```

conc_fact      = 2;      %Power for exponential distribution at LE.  This
                        %concentrates point near tip.

%Get meanline and dy/dx distributions from mean line data base
[x_f fc_o dydx_o] = getmeanline(mean_type);
[x_t tc_o RLE_o]  = getthickdist(thick_type);

%Scale appropriately
t_set  = t_set/2;      %uses 1/2 thickness
if max(fc_o) ~= 0
    f_scale = f_set/max(fc_o);
elseif max(fc_o) == 0
    f_scale = 0;
end
f_c    = fc_o * f_scale;
dydx   = dydx_o * f_scale;
t_scale = t_set/max(tc_o);
t_c    = tc_o * t_scale;
RLE    = RLE_o * (t_scale)^2;

%Find points along RLE nose parabola
x_RLE = fract*0.005*(0:1/(N_parab_def-1):1).^conc_fact;
t_RLE = sqrt(2*RLE*(x_RLE));

%Spline parabola and tabulated data for thickness function
x_locs = [x_RLE x_t(2:end)];      %New combined x/c values

%1e8 sets init slope = ~inf
t_fnct = csape(x_locs, [1e10 t_RLE t_c(2:end) 1],[1 0]);
    %Make x locations for generating data file
    %Cosine spacing from 0.005 to TE
x_cos_sp= 0.005 + 0.5*0.995*(1-cos(0:pi/(N_surf_pts-1):pi));
    %Exponential spacing for nose
x_eval_LE = fract*0.005*(0:1/(N_parab_eval-1):1).^conc_fact;
t_eval_LE = sqrt(2*RLE*(x_eval_LE));
x_eval_mb = [x_cos_sp]; %Establishes eval points
t_eval_mb = fnval(t_fnct, x_eval_mb); %Evaluates spline at eval points
x_eval = [x_eval_LE x_eval_mb];
t_eval = [t_eval_LE t_eval_mb];

%Spline tabulated data for camber at same x/c locations as thickness
f_fnct = csape(x_f, f_c);
f_eval = fnval(f_fnct, x_eval);
dydx_eval = fnval(fnder(f_fnct), x_eval);

%Plotting for unrotated parameters
if strcmp(make_plot,'yes')
    figure();
    hold on;
    axis equal;      %Set X:Y to unity
    title('Camber, Thickness, and LE Graphical Display')
    xlabel('X/C');
    xlim([-0.01 0.25]); %Set Initial Zoom
    %Plot thickness
    fnplt(t_fnct, 'y'); fnplt(f_fnct, 'g')

```

```

plot(x_t, t_c, 'co'); plot(x_f, f_c, 'ro')
plot(x_RLE, t_RLE, 'k.');
```

%Plot RLE Circle and parabola for viewing on plot

```

plot(RLE - RLE*cos(0:pi/100:pi), RLE*(sin(0:pi/100:pi)), 'b:');
plot((0:1/10000:0.2), sqrt(2*RLE*(0:1/10000:0.2)), 'r:');
%Plot camber

legend('Splined Thickness', 'Splined Camber',...
      'Tabulated Thickness (Scaled)', 'Tabulated Camber (Scaled)',...
      'Calcuated Parabola', 'Leading Edge Radius', 'LE Parabola',...
      'Location', 'southeast')
end

%Calculate upper and lower surface ordinates
x_u = x_eval - t_eval.*sin(atan(dydx_eval));
y_u = f_eval + t_eval.*cos(atan(dydx_eval));
x_l = x_eval + t_eval.*sin(atan(dydx_eval));
y_l = f_eval - t_eval.*cos(atan(dydx_eval));

%Solve for most forward point on foil
[x_fwd, min_i] = min(x_u);
y_fwd = y_u(min_i);

%New plot for actual upper and lower surfaces
if strcmp(make_plot, 'yes')
    figure();
    hold on;
    axis equal;           %Set X:Y to unity
    xlim([0 1]);         %Set Initial Zoom
    plot(x_u, y_u, 'b-', x_u, y_u, 'r.')
```

plot(x\_l, y\_l, 'b-', x\_l, y\_l, 'r.');

```

    plot(x_eval, f_eval, 'g-', x_eval, f_eval, 'r.')
```

plot(x\_fwd, y\_fwd, 'kp')

```

end

%Combine coordinates into a single array of points from TE along upper
%surface around LE back to TE along lower surface
x_comb = [fliplr(x_u) x_l];
y_comb = [fliplr(y_u) y_l];

%Rotate and scale such that max forward point is at 0,0, and TE is at 0,1.
%Assumes TE is already at 0,0 (Uses method in Brockett Report)
shift_ang = atan(y_fwd/(1-x_fwd));
%Scaled chord length back to 1 (accounts for portion forward of 0)
x_scaled = (x_comb-x_fwd)./(1-x_fwd);
y_scaled = (y_comb-y_fwd)./(1-x_fwd);
%Rotate so that most forward point is at 0,0
x_rot = (x_scaled.*cos(shift_ang) - y_scaled.*sin(shift_ang))/...
        sqrt(1+(y_fwd/(1-x_fwd))^2);
y_rot = (y_scaled.*cos(shift_ang) + x_scaled.*sin(shift_ang))/...
        sqrt(1+(y_fwd/(1-x_fwd))^2);

%New plot for final upper and lower surfaces
if strcmp(make_plot, 'yes')
```

```

figure();
hold on;
title('Final Points exported to Data File.');
```

axis equal;                   %Set X:Y to unity

xlim([0 1.1]);           %Set Initial Zoom

```

plot(x_rot, y_rot, x_rot, y_rot, 'r.');
```

legend('Connect the dots', 'Actual data points');

```

end

%Write to text file for use in XFOIL.
cmd = ['del ', save_as];       %save_as is file name to be written to
system(cmd);                   %Deletes previous file
fid = fopen(save_as, 'w');      %permission specifier changed from 'w' to 'wt'
for i = 1:length(x_rot)
    fprintf(fid, '%10.8f %10.8f\n', x_rot(i), y_rot(i));
end
fclose(fid);
```

### **getmeanline.m**

```

% Code by Chris Peterson
% Code developed to read meanline information from data file 'filename'.
% Data will be read in from file, and returned to function call. Data
% return is vectors containing x-locations, camber distribution, and
% camber line slope values. Function checks for 999 value specifying
% less data points than standard input format.

function [x_loc f_c dy_dx] = getmeanline(filename)

% cd('./Meanline');

input = dlmread(filename, '\t', 4, 0);
M = input';
x_loc_in= M(1,:)/100;
f_c_in = M(2,:)/100;
dy_dx_in= M(3,:);

for i=1:length(x_loc_in)
    if x_loc_in(i) == 9.99 %Checks to see if formatted with less points
        x_loc = x_loc_in(1:i-1);
        f_c = f_c_in(1:i-1);
        dy_dx = dy_dx_in(1:i-1);
        cd '..';
        return
    else
        x_loc = x_loc_in;
        f_c = f_c_in;
        dy_dx = dy_dx_in;
    end
end
end
```

### **getthickdist.m**

```

% Code by Chris Peterson
% Code developed to read thickness information from data file 'filename'.
% Data will be read in from file, and returned to function call. Data
% return is vectors containing x-locations, thickness distribution, and
% value of leading edge radius. Function checks for 999 value specifying
% less data points than standard input format.

function [x_loc t_c RLE] = getthickdist(filename)

% cd('./Thickness');

input = dlmread(filename, '\t', [4 0 29 2]);
M = input';
x_loc_in = M(1,:)/100;
t_c_in = M(2,:)/100;
fid = fopen(filename);
RLE = textscan(fid, '%s', 'headerlines', 29);
fclose all;
RLE = str2num(RLE{1}{7})/100;

for i=1:length(x_loc_in)
    if x_loc_in(i) == 9.99 %Checks to see if formatted with less points
        x_loc = x_loc_in(1:i-1);
        t_c = t_c_in(1:i-1);
        cd ..;
        return
    else
        x_loc = x_loc_in;
        t_c = t_c_in;
    end
end
end

```

### Appendix C: NACA $a=0.8$ and NACA 66(TMB mod) section

Note that the thickness distribution is not expressed in percent of the foil chord, as is customarily done with NACA families, but as a fraction of the maximum thickness.

Chordwise location $x/c$ (percent of chord)	Meanline NACA $\alpha=0.8$ , $f_0/c$ (percent of chord)	NACA 66 (TMB modified), $t/t_0$
0	0.0000	0.0000
0.5	0.2870	0.0665
0.75	0.4035	0.0812
1.25	0.6158	0.1044
2.5	1.0768	0.1466
5	1.8408	0.2066
7.5	2.4826	0.2525
10	3.0426	0.2907
15	3.9852	0.3521
20	4.7480	0.4000
25	5.3672	0.4363
30	5.8631	0.4637
35	6.2478	0.4832
40	6.5283	0.4952
45	6.7086	0.5000
50	6.7896	0.4962
55	6.7696	0.4846
60	6.6442	0.4653
65	6.4049	0.4383
70	6.0370	0.4035
75	5.5139	0.3612
80	4.7713	0.3110
85	3.6826	0.2532
90	2.4349	0.1877
95	1.1626	0.1433
100	0.0000	0.0333