İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

APPLICATION OF LIFTING LINE AND LIFTING SURFACE METHODS FOR OPTIMUM MARINE PROPELLER DESIGN

M.Sc. Thesis by
Barış BİÇER

Department : Naval Architecture and Naval Machinery Engineering

Programme : Naval Architecture and Marine Engineering

AUGUST 2010

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

APPLICATION OF LIFTING LINE AND LIFTING SURFACE METHODS FOR OPTIMUM MARINE PROPELLER DESIGN

M.Sc. Thesis by
Barış BİÇER
(508071018)

Date of submission : **23 August 2010**
Date of defence examination: **31 August 2010**

Supervisor (Chairman) : **Prof. Dr. Şakir BAL (ITU)**
Members of the Examining Committee : **Prof. Dr. Ahmet ERGİN (ITU)**
Prof. Dr. Mesut GÜNER (YTU)

AUGUST 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

OPTİMUM GEMİ PERVANE DİZAYNI İÇİN BİR KALDIRICI HAT VE KALDIRICI YÜZEY TEORİSİNİN UYGULANMASI

YÜKSEK LİSANS TEZİ

Barış BİÇER
(508071018)

Tezin Enstitüye Verildiği Tarih : 23 Ağustos 2010
Tezin Savunulduğu Tarih : 31 Ağustos 2010

Tez Danışmanı : Prof. Dr. Şakir BAL (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Ahmet ERGİN (İTÜ)
Prof. Dr. Mesut GÜNER (YTÜ)

AĞUSTOS 2010

DEDICATED TO RAHMAN, HE IS MOST MERCIFUL…

FOREWORD

I would like to thank my supervisor Prof. Dr. Şakir BAL for his helpful comments, suggestions, improvements, and corrections to my master thesis.

I would also like to thank my working company Besiktaş Shipyard, especially REPAIR DEPARTMENT members due to their supports, patiences and understanding during my master's education.

The last but not least, I would like to express my deep appreciation for my wife and family for their endless support.

August 2010

Barış BİÇER

Naval Architecture and Marine Engineering

TABLE OF CONTENTS

Page

ABBREVIATIONS

BEM : Boundary Element Method
CFD : Computational Fluid Dynamics
CPP : Controllable Pitch Propeller
FEM : Finite Element Method
FORTRAN : FORmula TRANslating System
ITTC : International Towing Tank Conference
NACA : National Advisory Committee for Aeronautics
PVL : Propeller Vortex Lattice
PUF : Propeller Unsteady Flow

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOL

Roman Symbols

A	:	Cross-sectional Area
A_e/A_0	:	Expanded Blade Area
A_E	:	Expanded area
A_D	:	Developed area
A_P	:	Projected area
AR	:	Aspect ratio
BAR	:	Blade area ratio
c	:	Chord Length
c/D	:	Chord / diameter ratio
c_d	:	Viscous drag coefficient
C_D	:	Drag Coefficient
C_F	:	Frictional resistance coefficient
C_f	:	Frictional drag coefficient
C_L	:	Lift Coefficient
C_M	:	Moment coefficient
C_P	:	Pressure coefficient
C_T	:	Thrust loading coefficient
D	:	Propeller Diameter
D	:	Drag Force
Fn	:	Froude number
G	:	Non-dimensional circulation distribution coefficient
J	:	Advance coefficient $J = V_\infty/(nD)$
K_Q	:	Propeller torque coefficient $= Q / \rho n^2 D^5$
K_T	:	Thrust coefficient $= T / \rho n^2 D^4$
$\vec{L}$	:	Vortex length element
M	:	Moment of force
N	:	Rotational speed (RPM)
P	:	Propeller pitch
P	:	Break Power

P_A	:	Available power from the propeller
r	:	The distance from the root to a given section
r_c	:	Control Point
r/R	:	Non-dimensional radii
$\vec{R}$	:	Distance between the element and field point
$u_a(n,m)$	:	Axial horseshoe influence function
$u_t(n,m)$	:	Tangential horseshoe influence function
V	:	Local relative inflow
V_a	:	Axial inflow velocity
V_r	:	Induced velocity
V_t	:	Tangential inflow velocity
V_S	:	Ship speed
V_∞	:	Speed of inflow
Z	:	Number of Blade

Greek Symbols

α	:	Angle of attack
α_o	:	Zero lift angle
β	:	Pitch angle
β_i, β_w	:	Hydrodynamic pitch angle
θ_0	:	Effective pitch angle
θ_{nt}, θ	:	Geometric pitch angle
Γ	:	Circulation
ω	:	The angular velocity of the propeller in radians per second
η	:	Propeller efficiency = $(JK_T)/(2\pi K_Q)$

APPLICATION OF A LIFTING LINE AND LIFTING SURFACE METHODS FOR OPTIMUM MARINE PROPELLER DESIGN

SUMMARY

In the present thesis, it is designed a marine propeller by coupling a lifting line and a lifting surface methods. First of all, it is computed optimum (max lift torque ratio) main dimensions of the propeller by a lifting line theory. Then, by using a lifting surface method, the section details of the blades such as pitch diameter ratio, and camber ratio have been found and analyzed. In order to do this, the span of the key blade is divided into a number of panels extending from hub to tip. The radial distribution of bound circulation has been computed by a set of vortex elements of constant strength. A discrete trailing free vortex line has been shed at each of the panel boundaries with strength equal to the difference in strengths of the adjacent bound vortices. It is considered that the vortex system is built from a set of horseshoe vortex element, each consisting of a bound vortex segment of constant strength and two free vortex lines of constant strengths. In addition, each horseshoe vortex element actually represents a set of number of blades identical elements of equal strength, one originating from each blade. Each set of horseshoe vortex elements induces an axial and tangential velocity at a specified control point on the key blade. The contributions of these two free vortices have been also found. An algebraic equation system is formed by using these influence coefficients. Once this equation system is solved with a prescribed hydrodynamic pitch angle, the circulation distribution has been computed. Then, with Betz and Lerb's methods the optimum circulation distribution is computed. In order to get optimum circulation distribution, the section details of blades have been analyzed and modified by a lifting surface method that is very similar to the lifting line explained above.

OPTİMUM GEMİ PERVANE DİZAYNI İÇİN BİR KALDIRICI HAT VE KALDIRICI YÜZEY TEORİSİNİN UYGULANMASI

ÖZET

Bu tezde, bir kaldırıcı hat ve bir kaldırıcı yüzey yöntemi kullanılarak gemi pervanesi dizayn edilmiştir. İlk olarak, kaldırıcı hat teorisi kullanılarak pervaneye ait optimum (maksimum lift tork oranı) ana ölçüler hesaplandı. Sonra, kaldırıcı yüzey yöntemi kullanılarak, kanat kesitlerine ait piç ve sehim oranları bulunarak analiz edildi. Bu analizi yapabilmek için kanat açıklığı, göbekten uca kadar devam eden panellere bölündü. Sirkülasyon radyal dağılımı, sabit şiddetteki bir dizi girdap elemanı olarak hesaplandı. Her bir panel sınırından, şiddeti bitişiğindeki sınır girdaplarının farkına eşit olan ayrık bir serbest girdap hattı çıkarıldı. Bu girdap sisteminin, her biri kendi içinde sabit şiddete sahip bağlı bir girdap katmanı ile yine sabit şiddetteki iki adet serbest girdap hattının oluşturduğu bir dizi at nalı girdap elemanından meydana geldiği kabul edilmiştir. Ayrıca, her bir at nalı girdap elemanı, aslında her biri bir kanattan kaynaklanan, eşit şiddetteki özdeş elemanların oluşturmuş olduğu bir dizi kanat sayısını temsil etmektedir. Her bir at nalı girdap elemanı kanat üzerinde belli bir kontrol noktasında bir teğetsel ve eksenel hız indükler. Ek olarak, bahsedilen iki serbest girdabın katkısı da bulunmuştur. Matematiksel bir denklem sistemi bu etki katsayıları kullanılarak oluşturulmuştur. Öncelikle, bu denklem sistemi belirtilen hidrodinamik piç açısı ile çözülmüş ve sirkülasyon dağılımı hesaplanmıştır. Sonrasında, Betz ve Lerbs yöntemleri kullanılarak optimum sirkülasyon dağılımı hesaplanmıştır. Optimum sirkülasyon dağılımını elde etmek için kanat kesit detayları analiz edilmiş ve yukarıda açıklanan kaldırıcı hat teorisi ile benzerlik gösteren bir kaldırıcı yüzey yöntemi ile modifiye edilmiştir.

1. INTRODUCTION

Even though marine propellers have been used to drive ships for over a century and lots of innovations made on propellers, scientifical and technological research and the design investigations of the propellers are as important as ever and still continuing (Ekinci, 2007). Both because the size of ships continues to increase and because increased focus on the limited fuel resources and the impact on the environment from the burning of fuel (emission rate decrase). This force the propeller designers and manufacturers to think of new alternative solutions of ship propulsion system which are capable of fulfilling the requirements of developing higher thrust with higher efficiency (Gerr, 1989).

In the present thesis, the main objective is to apply classical lifting line theory for determination of the optimum distribution of circulation along a propeller blade for the purpose of achieving the highest efficiency for a given thrust and to apply accordingly a vortex lattice method to get the optimum pitch and camber values.

The optimum distribution of circulation can be found by solving a variational problem where the propeller torque is minimised for a given propeller thrust or the propeller thrust is maximised for a given propeller torque (Olsen, 2001). In classic theory this problem is solved in an integral formulation where the propeller is modelled as a lifting line with a continuous distribution of circulation (Olsen, 2001). First of all, in 1927, Betz solved this problem for a propeller in open water, which is called optimum propeller criteria in the uniform flow [$\tan\beta(r)/\tan\beta_i(r)=const.=\eta$]. In 1952, Lerbs solved this problem for a propeller in a radially varying wake, which is called optimum propeller criteria in non-uniform flow. In this case, the pitch of the induced inflow on the lifting line is required to be proportional to the square root of the inflow velocity. In order to solve the problem it is necessary to use Munk's displacement theorem and linearise the problem (Olsen, 2001).

On the other hand, it is introduced a vortex lattice method (VLM) for the analysis of unsteady flow around marine propellers subject to non-uniform inflow in Kerwin and Lee (1978). In the vortex lattice method the propeller blade is replaced by a lattice of quadrilateral panels with constant circulation and the shed horseshoe vortices follow regular helices. It is improved the earlier vortex lattice method by taking account of viscous effect bear the leading edge and that cavitations inception based on a semi-empirical method in Kerwin and Greeley (1982). Coney (1992) developed a vortex lattice lifting line method for the determination of the optimum radial circulation distribution. This method is also applicable to multi-component propulsors, such as ducted propellers and propeller-stator combination. In 1997, Mishima and Kinnas noted the results from five propellers from the David W.Taylor Naval Ship Research and Development Centre propeller series with systematic varying skew and skew-induced rake, which show that the radial distribution of thrust is almost identical for all the propellers, whereas the distribution of circulation and torque differ. If the skew-induced rake is removed, the efficiency is further increased. Moreover, Performance analysis of podded propulsors, has been made with a vortex lattice method in Bal and Güner (2009). In this present thesis, a very similar method is applied to analyze the propeller blades

The present thesis covers the following four chapters. In the section of "A Brief Description of Propeller Characteristics", it is explained a geometric and hydrodynamic characteristic of a marine propeller. It is also explained the propeller vortex lattice method and propeller unsteady flow analysis programs.

In the next section of "Numerical Results", an application of PVL (Propeller Vortex Lattice) and PUF (Propeller Unsteady Flow) programs has been given for an optimum marine propeller. Furthermore, these results have been compared with the conditions which are calculated by changing blade numbers, input radii, and with/without hub affect. Meanwhile, it is also obtained optimum circulation distribution by using cosine spacing and compared with uniform spacing. Optimum circulation distribution has been obtained from PVL and PUF in this case.

In the last section of "Conclusion and Discussion", it has been done general assessments about obtained results and given some comments for the future study.

2. A BRIEF DESCRIPTION OF PROPELLER CHARACTERISTICS

It is duty of the naval architect to design ships with hull forms having low resistance when they move through water. The propulsion system must be more efficient, that is, the amount of enegry necessary for the propulsion of the ship must be as small as possible (Harvald, 1991). The marine propeller is the most common form of marine propulsion device; in general, it is also the most efficient. Therefore, in this section a brief description of the basic principles of propeller has been examined.

2.1 Propeller

Hydrofoils and propellers have something in common that they are both made up of foil sections designed to generate a hydrodynamic force. The foil force provides lift to sustain the marine vessels in the water; the propeller force provides thrust to push the marine vessels through the water (Anderson, 2005). A sketch of a simple three-blade propeller is given in Figure 2.1, illustrating that a cross section is indeed a hydrofoil shape.

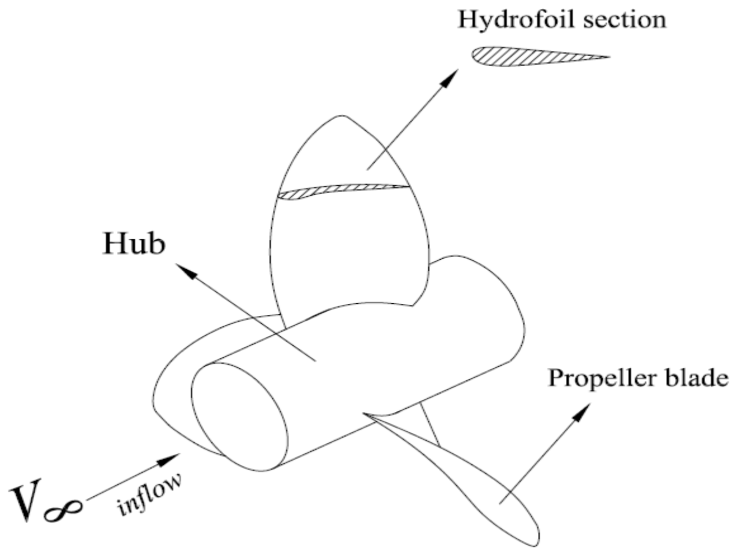


Figure 2.1 : The marine propeller, emphasizing that a propeller cross section is a hydrofoil shape.

However, unlike a hydrofoil, where the chord lines of the hydrofoil sections are essentially all in the same direction, a propeller is twisted such that the chord line changes from being almost parallel to V_∞ at the root, to almost perpendicular at the tip. This is showed in Figure 2.2, which displays a side of the propeller, as well as two sectional views, one at the tip and the other at the root. As it is seen in the figure 2.2, the angle between the chord line and the propeller's plane of rotation is defined as the *pitch angle* β. The distance from the root to a given section is *r*. It is noted that $\beta = \beta(r)$

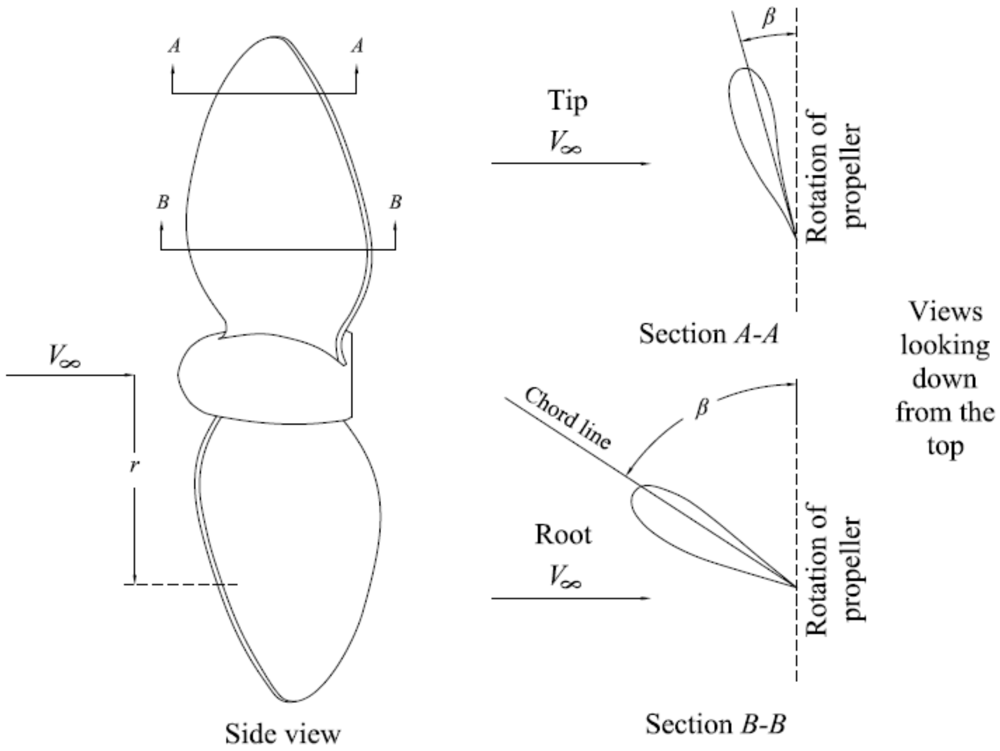


Figure 2.2 : Illustration of propeller, showing variation of pitch along the blade.

The hydro flow seen by a given propeller section is combination of the marine vessel's forward motion and the rotation of the propeller itself. This is sketched in Figure 2.3(a), where the marine vessel's relative inflow is V_∞ and the speed of the blade section due to rotation of the propeller is $r\omega$. The relative inflow seen by the propeller section is the vector sum of V_∞ and $r\omega$, as shown in Figure 2.3(b)

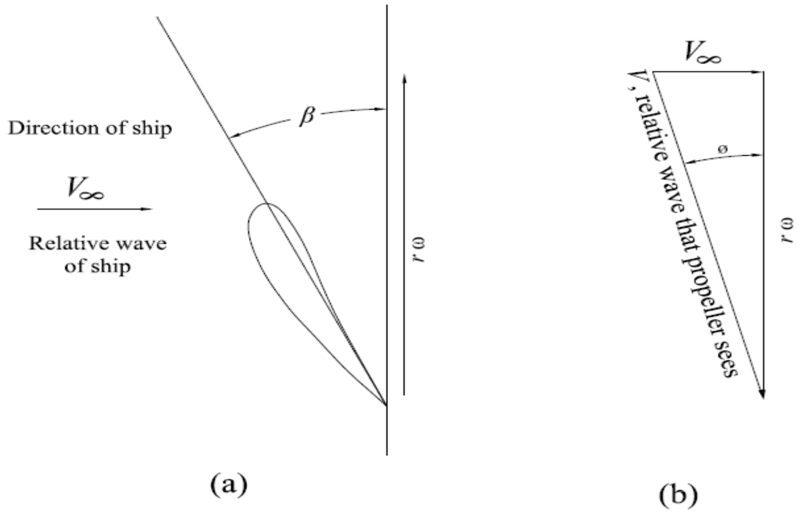


Figure 2.3 : Velocity diagram for the flow velocity relative to the propeller.

Clearly, if the chord line of the hydrofoil section is at an angle attack α with respect to the local relative inflow *V*, then lift and drag (perpendicular and parallel to *V*, respectively) are generated. In turn, as shown in Figure 2.4, the components of *L* and *D* in the direction of V_∞ produce net thrust *T*:

$$T = L\cos\phi - D\sin\phi \tag{2.1}$$

where $\phi = \beta - \alpha$. This thrust, when summed over the entire length of the propeller blades, yields are the net thrust available, which drives the ships forward.

This simple picture is the essence of how a propeller works. However, the actual prediction of propeller performance is more complex. The propeller is analogous to a finite foil that has been twisted. Therefore, the hydrodynamics of the propeller are influenced by the same induced flow due to tip vortices (Anderson, 2005).

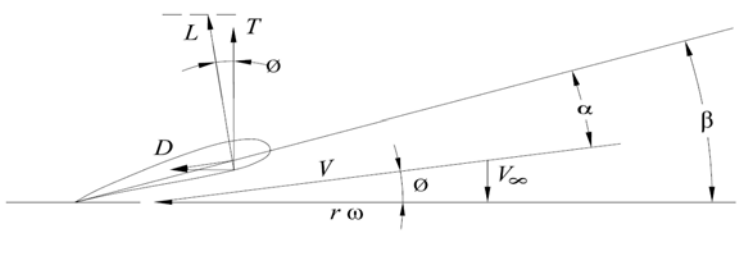


Figure 2.4 : Generation of propeller thrust.

On the other hand, on the understanding of propeller efficiency η is defined in the following Eq. (2.2)

$$\eta = \frac{P_A}{P} \tag{2.2}$$

where P is the shaft brake power (the power delivered to the propeller by the shaft of the engine) and P_A is the power available from the propeller as given in Eq. (2.3) follow,

$$P_A = T_A . V_\infty \tag{2.3}$$

Hence Eq. (2.2) becomes

$$\eta = \frac{T_A V_\infty}{P} \tag{2.4}$$

where T_A is basically an hydrodynamic phenomenon that is dependent on the angle of attack α which is showed in Figure 2.4. In turn, α is dictated by the pitch angle β and ϕ, where ϕ itself depends on the magnitudes of V_∞ and $r\omega$. The angular velocity is defined,

$$\omega = 2\pi n \tag{2.5}$$

where n is number of propeller revolutions per second. Consequently, T_A must be a function of at least β, V_∞ and n. Finally, the thrust must also depend on the size of the propeller, characterized by the propeller diameter D. Indeed, theory and experiment both show that for a fixed pitch angle β, η is a function of the dimensionless quantity advance ratio which is given below,

$$J = \frac{V_\infty}{nD} \tag{2.6}$$

A typical variation of η with J for a fixed β is sketched in Figure 2.5; three curves are shown corresponding to three different value of pitch. This figure is also important it is from such curves that η is obtained for a ship performance analysis.

In this figure note that η<1; this is because some of the power delivered by shaft to propeller is always lost, and hence P_A<P.

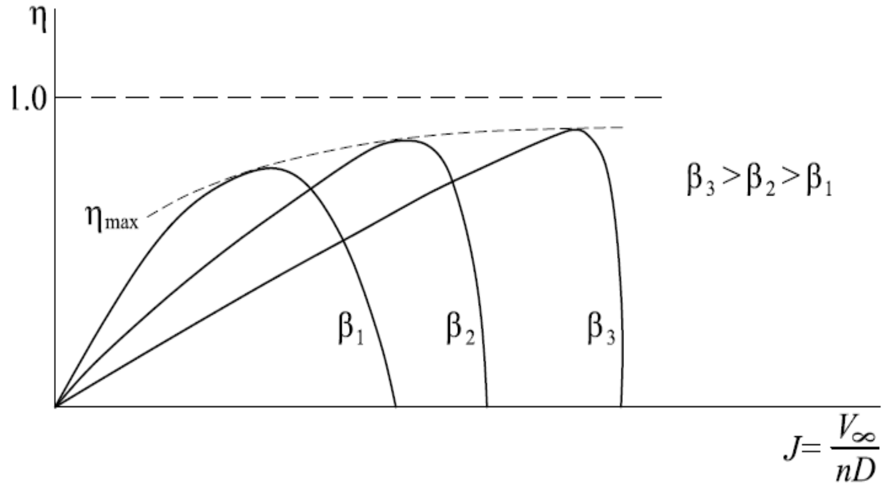


Figure 2.5 : Propeller efficiency versus advance ratio.

In Fig. 2.5, it is also showed that for a fixed β, the efficiency is zero at J = 0, increases as J increases, goes through a maximum, and then rapidly decreases at higher J, finally again going to zero at the some large finite value of J.

A consideration of the relative foil also explains why a propeller blade is twisted, with the large β at the root and a small β at the tip. Near the root, r, and hence $r\omega$, is small. Thus, as shown in Figure 2.6(a), β must be large to have reasonable α. In contrast, near the tip, r, and hence $r\omega$, is large. Therefore, as shown in Figure 2.6(b), β must be smaller in order to have reasonable α.

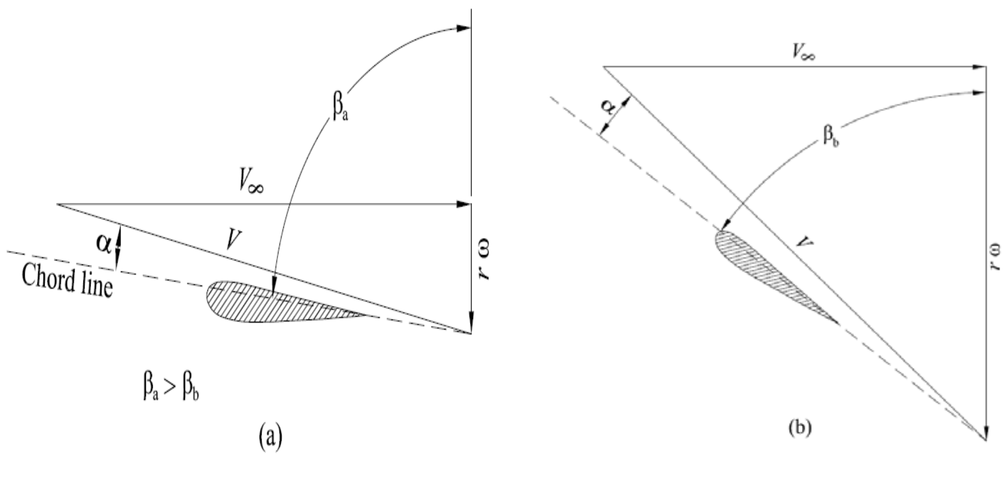


Figure 2.6 : Difference in the relative hydrofoil along the propeller blade. (a) Near the root; (b) near the tip.

2.1.1 Different pitch definitions

There are several pitch definitions which are defined below;

***Nose-tail pitch*:** The straight line connecting the extremities of the mean line or nose and tail of a propeller blade is called nose-tail pitch line the section angles of attack are defined to the nose-tail line.

***Face pitch*:** The face pitch line is basically a tangent to section's pressure side surface and you can draw so many lines to the pressure side. Therefore its definition is not clear. It is rarely used but it can be seen in older drawings like Wageningen-B series.

Effective or no-lift pitch: It is the pitch line of the section corresponding to aerodynamic no-lift line which results zero lift.

Hydrodynamic pitch: The hydrodynamic pitch angle (β_i) is the pitch angle at which the incident flow encounters the blade section.

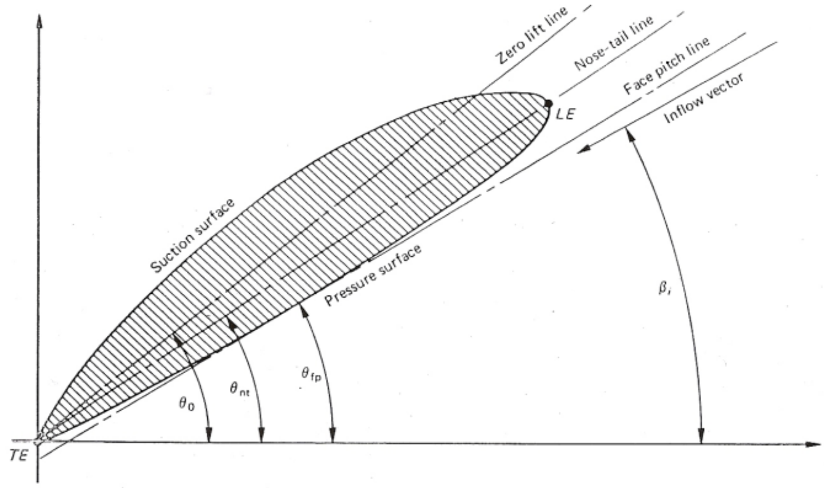


Figure 2.7 : Pitch lines (Carlton, 2007)

where

- θ_0 is the effective pitch angle of the propeller
- θ_{nt} or θ is the geometric pitch angle of the propeller
- $\beta_i = \beta_w$ is the hydrodynamic pitch angle
- α is the angle of attack of section

2.1.2 Blade section geometry and definitions

Below figure 2.8 shows the general definitions of the aerofoil.

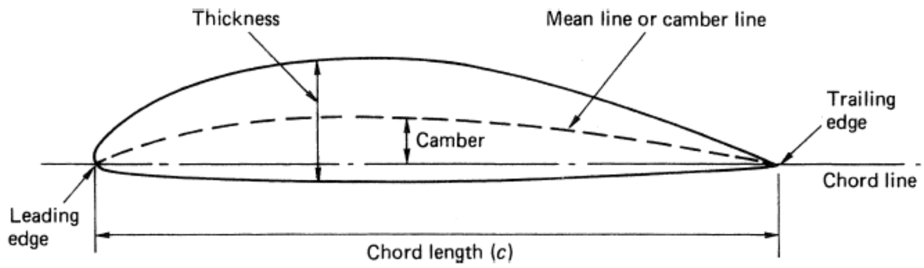


Figure 2.8 : General definition of an aerofoil section (Carlton, 2007)

Mean line or chamber line: The mean line or camber line is the locus of the mid-points between the upper and lower surfaces when measured perpendicular to the camber line. The extremities of the camber line are termed the leading and trailing edges of the aerofoil and the straight line joining these two points is termed the chord line.

Chord length (c): The distance between the leading edge and trailing edges when measured along the chord line is termed as chord length of the section.

***Chamber**:* The camber of the section is the maximum distance between the mean camber line and the chord line, measured perpendicular to the chord line.

Thickness: The aerofoil thickness is the distance between the upper and lower surfaces of the section, usually measured perpendicularly to the chord line although strictly this should be to the camber line.

Leading Edge: When the propeller rotating the edge piercing water is called leading edge.

Trailing Edge: When the propeller rotating the edge trailing the leading edge is called trailing edge.

Leading edges are usually circular having a leading edge radius defined about a point on the camber line. Typical section used for ship propeller is NACA66 series with the mean line a=0,8

2.2 PVL (Propeller Vortex Lattice)

A vortex lattice solution to the lifting line problem is conceptually very similar to the solution of the planar lifting line problem. The span of the key blade is divided into M panels extending from $r = r_h$ to $r = R$. The radial distribution of (bound) circulation, $\Gamma(r)$, is approximated by a set of M vortex elements of constant strength Γ_m extending from $r_v(m)$ to $r_v(m+1)$. A discrete trailing (free) vortex line is shed at each of the panel boundaries, with a strength equal to the difference in strengths of the adjacent bound vortices. However, as with planar lifting line theory, it is more convenient to consider that the vortex system is built from a set of M horseshoe elements, each consisting of a bound vortex segment of strength Γ_m and two free vortex lines of strength $\pm\Gamma_m$. But in addition, each horseshoe element actually represents a set of Z identical elements of equal strength, one originating from each blade.

Each horseshoe vortex elements results in an axial and tangential velocity at a specified control point, $r_c(n)$ on the key blade. The contribution of the two free vortices can be calculated as follows;

For $r_c < r_v$:

$$\bar{u}_a(r_c) = \frac{Z}{4\pi r_c}(y - 2Zr_vF_1) \qquad \textbf{(2.7a)}$$

$$\bar{u}_t(r_c) = \frac{Z^2}{2\pi r_c}\ y_0F_1 \qquad \textbf{(2.7b)}$$

For $r_c > r_v$:

$$\bar{u}_a(r_c) = -\frac{Z^2}{2\pi r_c}yy_0F_2 \qquad \textbf{(2.8a)}$$

$$\bar{u}_t(r_c) = \frac{Z}{4\pi r_c}(1 + Z2y_0F_2) \qquad \textbf{(2.8b)}$$

where;

$$F_1 \approx -\frac{1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{0.25}\left\{\frac{1}{U^{-1}-1}+\frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}}+\frac{3y^2-2}{(1+y^2)^{1.5}}\right]\ln\left(1+\frac{1}{U^{-1}-1}\right)\right\} \quad \textbf{(2.9a)}$$

$$F_2 \approx -\frac{1}{2Zy_0}\left(\frac{1+y_0^2}{1+y^2}\right)^{0.25}\left\{\frac{1}{U-1}+\frac{1}{24Z}\left[\frac{9y_0^2+2}{(1+y_0^2)^{1.5}}+\frac{3y^2-2}{(1+y^2)^{1.5}}\right]\ln\left(1+\frac{1}{U-1}\right)\right\} \quad \textbf{(2.9b)}$$

and,

$$U = \left\{\frac{y_0(\sqrt{1+y^2}-1)}{y(\sqrt{1+y_0^2}-1)}\exp(\sqrt{1+y^2}-\sqrt{1+y_0^2}\right\}^Z \quad \textbf{(2.10a)}$$

$$y = \frac{r_c}{r_v \tan\beta_w} \quad \textbf{(2.10b)}$$

$$y_0 = \frac{1}{\tan\beta_w} \quad \textbf{(2.10c)}$$

The contribution of the bound vortex element of the set of horseshoe vortices is zero, provided that the lifting line is radial and that the blades have uniform angular spacing. Clearly the bound vortex on the key lifting line induces zero velocity anywhere along that line. Bound vortex elements on another blade may induce a velocity on the key blade, but their summed effect will cancel due to symmetry. The total induced velocity at control point $r_c(n)$ is therefore,

$$u_a^*(r_c(n)) = \sum_{m=1}^{M}\Gamma_m \bar{u}_a(n,m) \quad \textbf{(2.11a)}$$

$$u_t^*(r_c(n)) = \sum_{m=1}^{M}\Gamma_m \bar{u}_t(n,m) \quad \textbf{(2.11b)}$$

where $u_a(n,m)$ and $u_t(n,m)$ are the horseshoe influence functions.

As with planar foil lifting line theory, it is best to use cosine spacing for the vortex and control points. Defining $h = 0.5(R - r_h)$ and $\delta = \pi/(2M)$, their coordinates are,

$$r_v(m) = r_h + h\left[1 - \cos\left(2\left(m-1\right)\delta\right)\right] \tag{2.12a}$$

$$r_c(n) = r_h + h\left[1 - \cos\left(2n-1\right)\delta\right] \tag{2.12b}$$

There are a number of possible approaches to solving for the circulation. The method used in PVL is similar to Lerbs method. In addition, instead of Lerbs extension of Glauert's sine series method, a vortex lattice is used. The intension will be to find the circulation distribution for a propeller with a specified thrust coefficient, C_T.

PVL program has been run by using following input data file and the following results have been obtained (Kerwin, 2001).

```
32  : NUMBER OF VORTEX PANELS OVER THE RADIUS
10  : MAXIMUM ITERATIONS IN WAKE ALIGNMENT
0   : HUB IMAGE FLAG: 1=YES, 0=NO
0.25 : HUB VORTEX RADIUS/HUB RADIUS
11  : NUMBER OF INPUT RADII
5   : NUMBER OF BLADES
0.8 : ADVANCE COEFFICIENT, J, BASED ON SHIP SPEED
1.000 : DESIRED THRUST COEFFICIENT, CT
0.000 : HUB UNLOADING FACTOR: 0.0=OPTIMUM (NO UNLOADING)
0.000 : TIP UNLOADING FACTOR 1.0=REDUCED LOADING
1.000 : CRP SWIRL CANCELLATION FACTOR: 1.0=NO CANCELATION
r/R c/D Cd Va/Vs Vt/Vs
0.20000 0.17400 0.00800 0.71969 0.00000
0.25000 0.19700 0.00800 0.74300 0.00000
0.30000 0.22900 0.00800 0.76260 0.00000
0.40000 0.27500 0.00800 0.79460 0.00000
0.50000 0.31200 0.00800 0.82034 0.00000
0.60000 0.33700 0.00800 0.84198 0.00000
0.70000 0.34700 0.00800 0.86073 0.00000
0.80000 0.33400 0.00800 0.87731 0.00000
0.90000 0.28000 0.00800 0.89218 0.00000
0.95000 0.24000 0.00800 0.89911 0.00000
1.00000 0.00200 0.00800 0.90572 0.00000
```

Figure 2.9 : Sample input data for PVL.

The first entry is number of panels, which in this case is M = 32. The last part of the data file consists of a tabulation of the chord/diameter ratio, c/D, the viscous drag coefficient, Cd, the axial inflow velocity, Va, and the tangential inflow velocity, Vt, at a user-specified set of non-dimensional radii, r/R starting with the hub and ending with the tip. The number of input radii is arbitrary, and is given in the 5'th entry in the table— in this case 11.

First Result: By using above input data, following circulation – non-dimensional Radii graph has been obtained with PVL code in non-uniform flow. In this Example, there is no hub affect, and a Lerbs optimum circulation distribution has been selected. Found results have been showed in the following table 2.1.

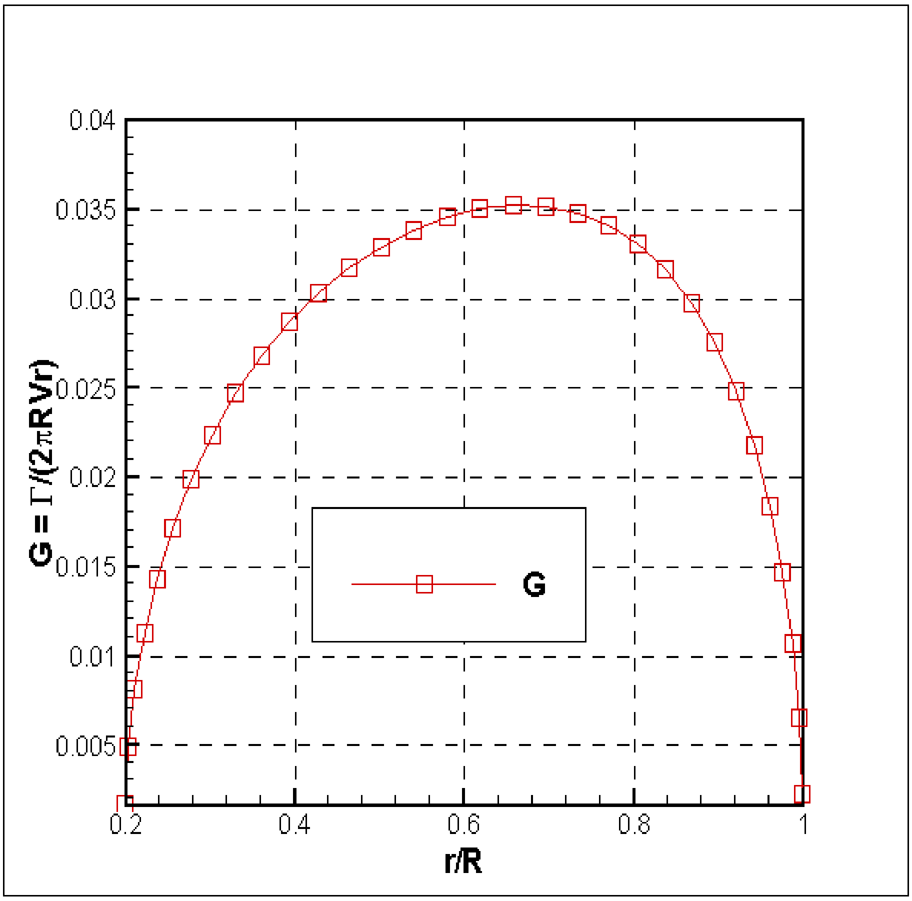


Figure 2.10 : Lifting Line results for a 5 bladed propeller obtained with PVL code.

Table 2.1 : Lifting Line results for a 5 bladed propeller obtained with PVL program without hub effect

J	K_T	K_Q	η
0.80	0.2513	0.0430	0.6347

<u>Second Result:</u> By using again above input data, following circulation–Radius graph obtained by using the PVL code in non-uniform flow with an image hub affect. In this Example, there is an image hub affect, and a Lerbs optimum circulation distribution has been selected.

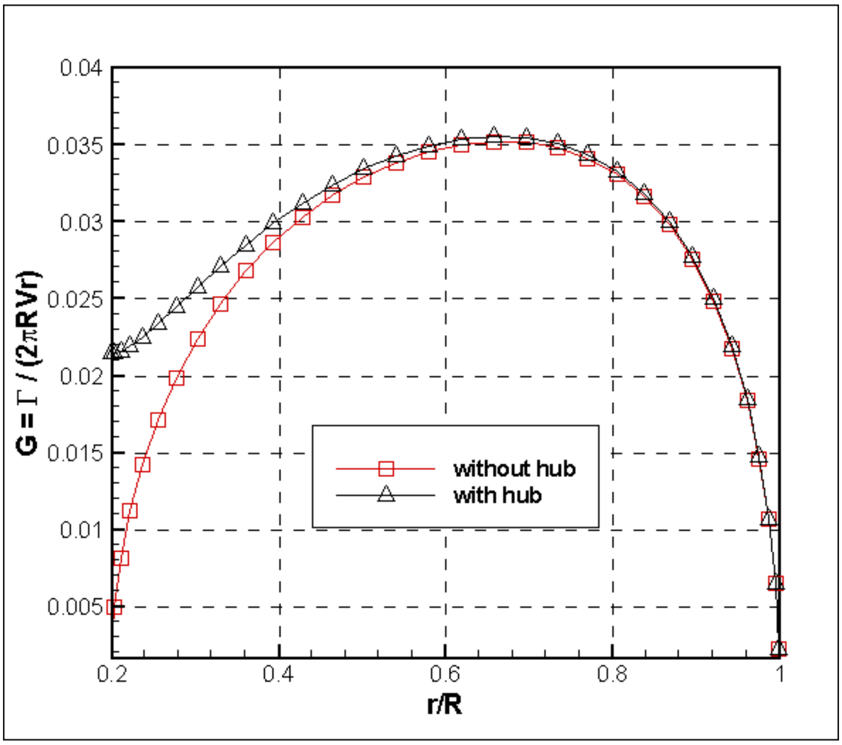


Figure 2.11 : Lifting Line results for without hub / with hub effect.

Table 2.2 : Lifting Line results for a 5 bladed propeller obtained without hub/with hub effect

	Without hub	With hub
J	0.8	0.8
K_T	0.2513	0.2513
K_Q	0.0430	0.0440
η	0.6347	0.6203

As it is seen above table 2.2, the efficiency has been reduced slightly due to hub vortex drug.

Third Result: By using again above input data, following circulation – Radius graph obtained with the PVL code in non-uniform flow with an image hub affect and by modifying distribution to unload the tip and to unload the hub. In this Example, there is an image hub affect, and a Lerbs optimum circulation distribution has been modified to unload the tip and to unload the hub, using HT = 1.0 and HR = 1.0, respectively.

The unloading factors, HR and HT are defined as fractional amount that difference between the optimum values of tanβi and tanβ are reduced. For example, if HR=0, tanβi – tanβ at the hub is retained at it's optimum value from Betz/Lerbs Criteria. If HR=1.0, tanβi- tanβ at the hub zero and the values up to mid span of the blade are blended parabolically to the optimum value. The same procedure applies to the tip.

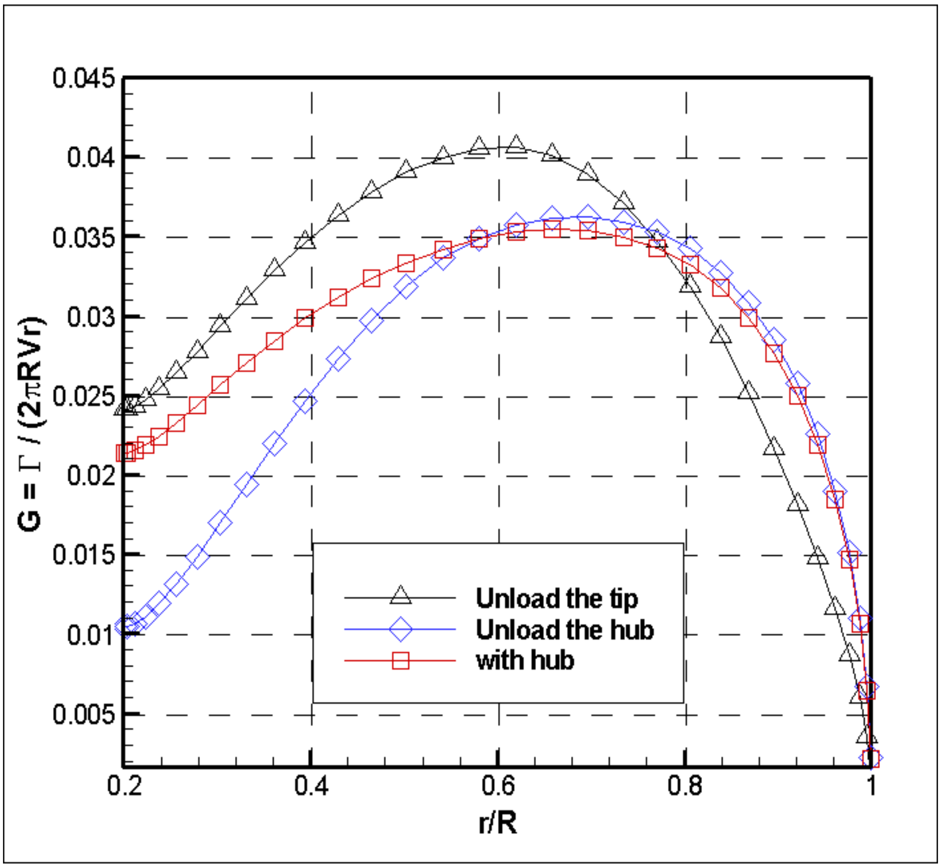


Figure 2.12 : Lifting Line results for unload tip and unload hub effects.

Table 2.3 : Lifting Line results for for a 5 bladed propeller with unload the hub / unload tip effects

	With hub	Unload the hub	Unload the tip
J	0.8	0.8	0.8
K_T	0.2513	0.2513	0.2513
K_Q	0.0440	0.0430	0.0461
η	0.6203	0.6343	0.5924

As it is seen above table 2.3, the efficiency has been further reduced due to tip unloading. On the other hand, the efficiency has actually improved, since the reduced hub loading reduced the hub vortex drag. Further studies on PVL can be found in Kerwin 2001.

2.3 PUF (Propeller Unsteady Flow)

A lifting surface method is applied to calculate the propulsive performance and induced velocities, due to propeller as very similar to one given in Bal and Guner (2009). This model is based on appropriate vortex and source-sink distribution. The singularities are distributed on the mean lines of the propeller blade sections. Those vortices are divided into two parts; bound and trailing vortices. The bound vortices, located in a radial direction, are to simulate the load distribution on the propeller blade. The trailing vortices are placed in the direction of the flow, obtained from the different intensities of adjacent bound vortex elements. A number of source elements are taken, at adjacent bound vortex, to simulate the thickness of the blade. The vortex strengths are calculated by solving a set of simultaneous equations, which satisfy the flow tangency condition at the blade control points. Induced velocities due to vortex elements of the lifting surface are calculated using Biot-Savart Law showed below given in equation (2.13). Induced velocities due to sources/sinks are computed based on given source/sink intensity.

$$\vec{V}_r = \left(\frac{\Gamma}{4\pi}\right) \cdot \left(\frac{\vec{L}}{\vec{R}}\right) \cdot \left(\frac{1}{\vec{R}^3}\right) \qquad \textbf{(2.13)}$$

where $\overrightarrow{Vr}$ is induced velocity, Γ is the circulation, $\vec{L}$ is the vortex length element, $\vec{R}$ is the distance between the element and field point. Once the bound vortex elements intensity is solved, then the velocity induced by the propeller in any point in space can be computed, using five angular position of the propeller blade. Finally, the arithmetic average of the values becomes induced velocity at the corresponding point.

3. NUMERICAL RESULTS

It is chosen a sample propeller to compare the results of the present application (Coupling of PVL and PUF programs) with those of given in Mishima and Kinnas (1997). By changing the blade numbers, number of radii input and with/without hub effects new circulation distributions have been obtained with PVL and PUF programs and compared with the optimum circulation distribution given in Mishima and Kinnas (1997). Then, by using cosine spacing instead of uniform spacing along propeller radius, new circulation distribution has been found with PUF and the results compared with chosen sample. In addition, in these calculations, the pitch distribution and camber ratios have been only taken into consideration. The rake and skew effects have been ignored.

3.1 Optimum Sample Propeller in Uniform Flow

Design Conditions of the sample propeller taken from Mishima and Kinnas (1997) are as follows;

- ✓ $J_S = 1.0$
- ✓ *The hub/diameter ratio* = 0.2
- ✓ $K_T = 0.1500$
- ✓ $K_Q = 0.0304$
- ✓ *The frictional drag coefficient,* $c_f = 0.004$
- ✓ *The number of blades,* $Z = 3$
- ✓ *Without hub affect*

It is shown in the following figure 3.1 that by using above design conditions, sample values and graphs have been obtained for optimum circulation distribution and optimum blade geometry taken from Mishima and Kinnas (1997).

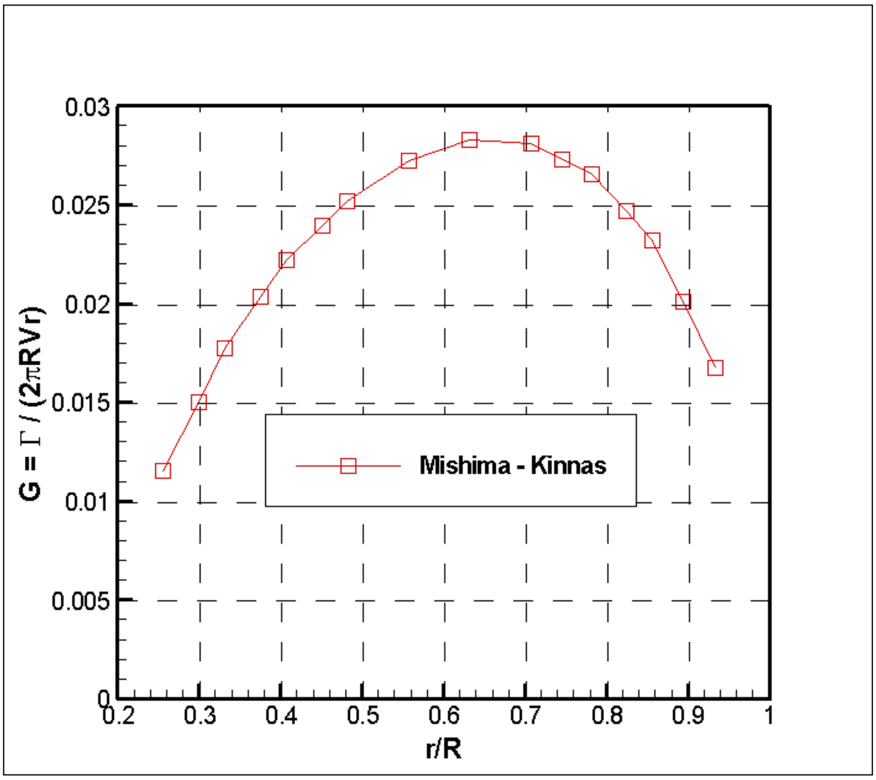


Figure 3.1 : Optimum circulation distribution from Mishima and Kinnas (1997)

Table 3.1 : Optimum Circulation distribution values

r/R	G
0.256131	0.011558
0.298984	0.014987
0.332039	0.017716
0.376139	0.020374
0.407992	0.022192
0.45089	0.023938
0.48153	0.025196
0.557549	0.027218
0.632368	0.02826
0.707218	0.02811
0.745277	0.027263
0.782106	0.026558
0.823874	0.02466
0.857042	0.023183
0.893935	0.020094
0.933289	0.016724

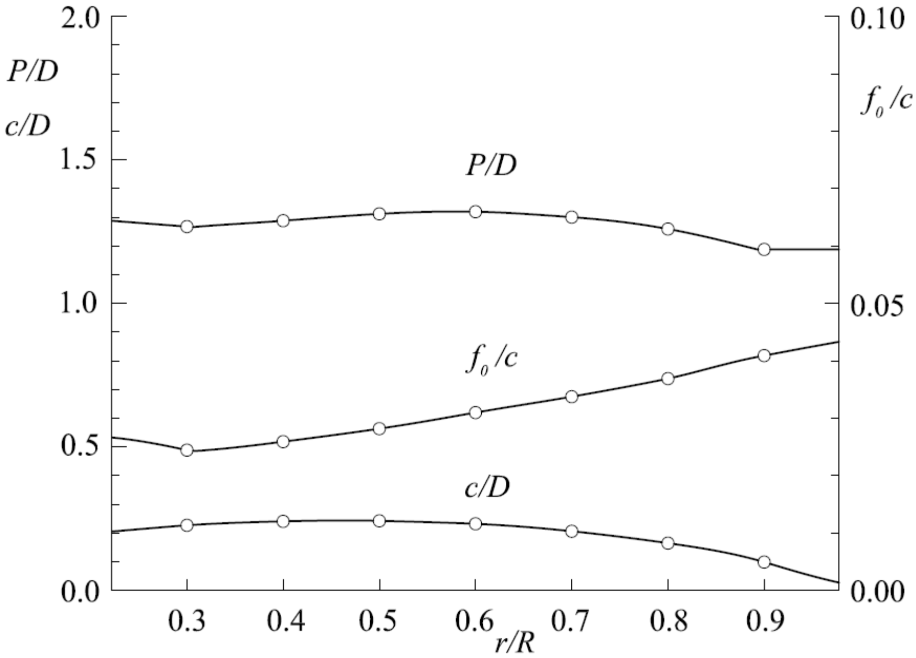


Figure 3.2 : Optimum blade geometry from Mishima and Kinnas (1997).

In the table 3.2, it is displayed the optimum blade geometry values from taken above graph.

Table 3.2 : Optimum Blade Geometry Values

r/R	**P/D**	**f_0/c**	**c/D**
0.22000	1.28500	0.02500	0.20000
0.30000	1.27000	0.02250	0.22500
0.40000	1.29500	0.02500	0.23000
0.50000	1.32000	0.02750	0.24000
0.60000	1.33000	0.03000	0.23500
0.70000	1.31000	0.03350	0.22500
0.80000	1.30000	0.03650	0.19000
0.90000	1.22000	0.41000	0.15000
0.98000	1.22000	0.04350	0.05000

3.2 Optimum Circulation Distribution with PVL

The above sample optimum circulation distribution has been obtained by PVL without hub effect. In this section, calculation of optimum circulation distribution has been done by using same c/D ratio nd input values of PVL program have been adjusted as follows (according to sample propeller given in Mishima and Kinnas, 1997);

```
32 : NUMBER OF VORTEX PANELS OVER THE RADIUS
10 : MAXIMUM ITERATIONS IN WAKE ALIGNMENT
0 : HUB IMAGE FLAG: 1=YES, 0=NO
0.25 : HUB VORTEX RADIUS/HUB RADIUS
11 : NUMBER OF INPUT RADII
3 : NUMBER OF BLADES
1.0 : ADVANCE COEFFICIENT, J, BASED ON SHIP SPEED
0.3820 : DESIRED THRUST COEFFICIENT, CT
1.000 : HUB UNLOADING FACTOR: 0.0=OPTIMUM (NO UNLOADING)
0.000 : TIP UNLOADING FACTOR 1.0=REDUCED LOADING
1.000 : CRP SWIRL CANCELLATION FACTOR:1.0=NO CANCELATION
r/R c/D Cd Va/Vs Vt/Vs
0.20000 0.17600 0.00400 1.00000 0.00000
0.25000 0.22000 0.00400 1.00000 0.00000
0.30000 0.22500 0.00400 1.00000 0.00000
0.40000 0.23000 0.00400 1.00000 0.00000
0.50000 0.24000 0.00400 1.00000 0.00000
0.60000 0.23500 0.00400 1.00000 0.00000
0.70000 0.22500 0.00400 1.00000 0.00000
0.80000 0.19000 0.00400 1.00000 0.00000
0.90000 0.15000 0.00400 1.00000 0.00000
0.95000 0.14211 0.00400 1.00000 0.00000
1.00000 0.00200 0.00400 1.00000 0.00000
```

Figure 3.3 : Input values for PVL

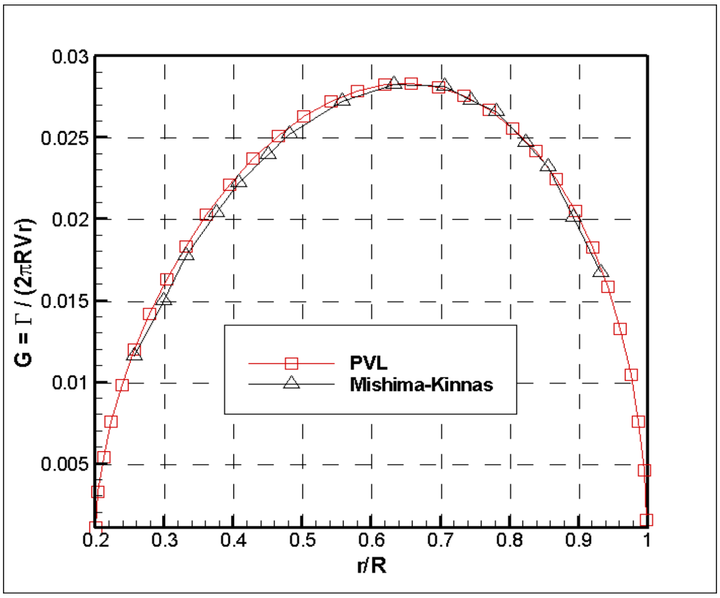


Figure 3.4 : Optimum circulation distribution obtained by PVL compared with those of Mishima and Kinnas (1997).

Table 3.3 : Comparision results of PVL and Mishima-Kinnas methods

	Present Method PVL	Mishima and Kinnas method
J	1.00	1.00
K_T	0.1500	0.1500
K_Q	0.0300	0.0304
η	0.7957	0.7853

As it is seen from above table 3.2 and figure 3.4, optimum circulation distribution found by Mishima and Kinnas has been obtained by using PVL program.

3.2.1 Changing of strip numbers

During calculating of above found results with PVL, 11 pieces input radii have been used and in this section number of input radii has been chosen as 9 and 15, and results have been compared with the result of number of input radii 11.

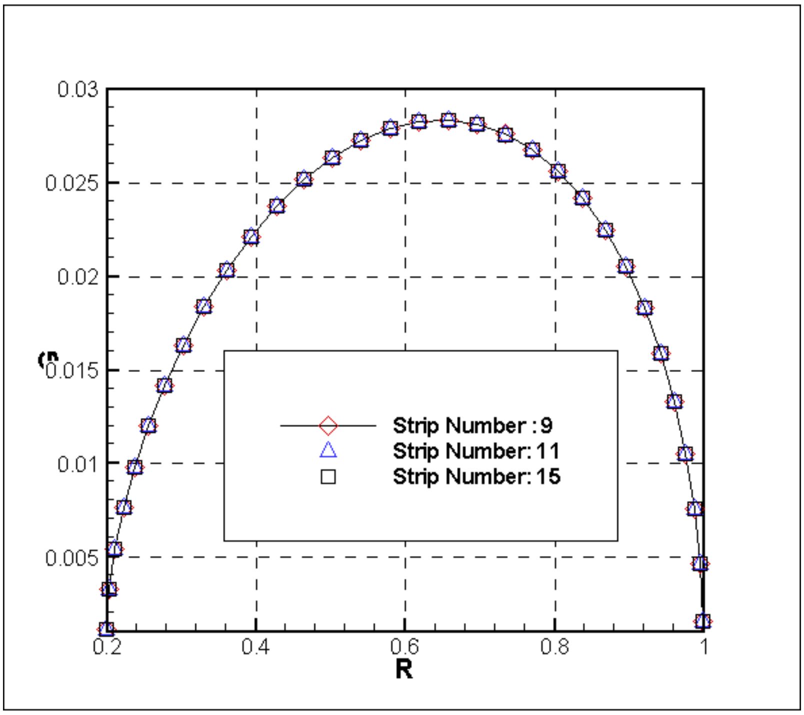


Figure 3.5 : Optimum circulation distribution obtained by PVL by using different strip numbers

Table 3.4 : Comparision of different strip numbers

	Number of input radii : 9	Number of input radii: 11	Number of input radii: 15
J	1.00	1.00	1.00
K_T	0.1500	0.1500	0.1500
K_Q	0.0300	0.0300	0.0300
η	0.7971	0.7957	0.7963

As it is clearly seen from above the table 3.4, changing of the input radii do not affect the results significantly. Therefore, in the calculations with PVL number of input radii have been taken as 11.

3.3 Optimum Circulation Distribution with PUF

To calculate the same optimum circulation distribution for chosen sample propeller with PUF, the same pitch and camber ratios have been taken from table 3.1 and then following circulation distribution and the corresponding results have been obtained.

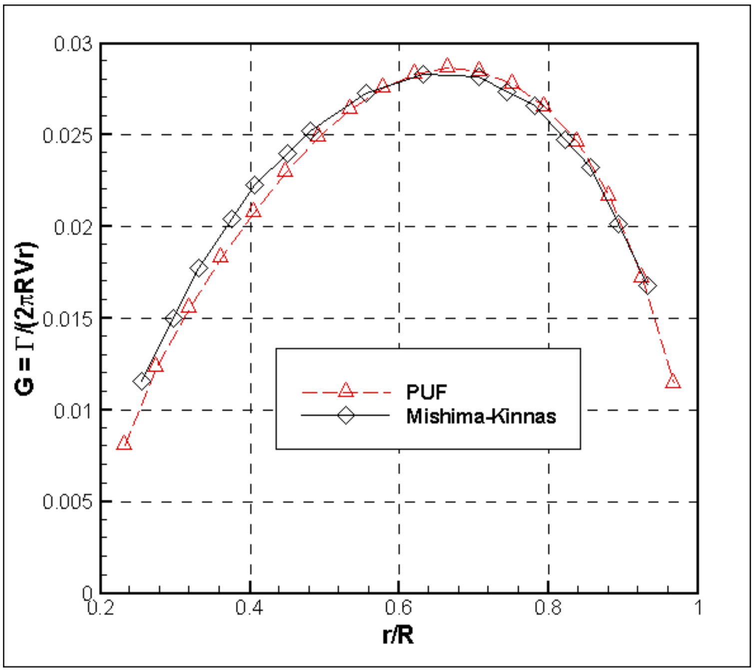


Figure 3.6 : Optimum circulation distribution obtained by PUF

Table 3.5 : Comparision of PUF results with those of Mishima and with Kinnas

	Present Method PUF	Mishima and Kinnas method
J	1.00	1.00
K_T	0.1453	0.1500
K_Q	0.0292	0.0304
η	0.7919	0.7853

It is shown in figure 3.7 optimum circulation distribution obtained by PVL and PUF programs together compared with those of Mishima and Kinnas. Also, as it is seen below figure 3.7 and table 3.5, the chosen optimum circulation distribution found by Mishima and Kinnas method has been obtained by using PVL and PUF programs approximately.

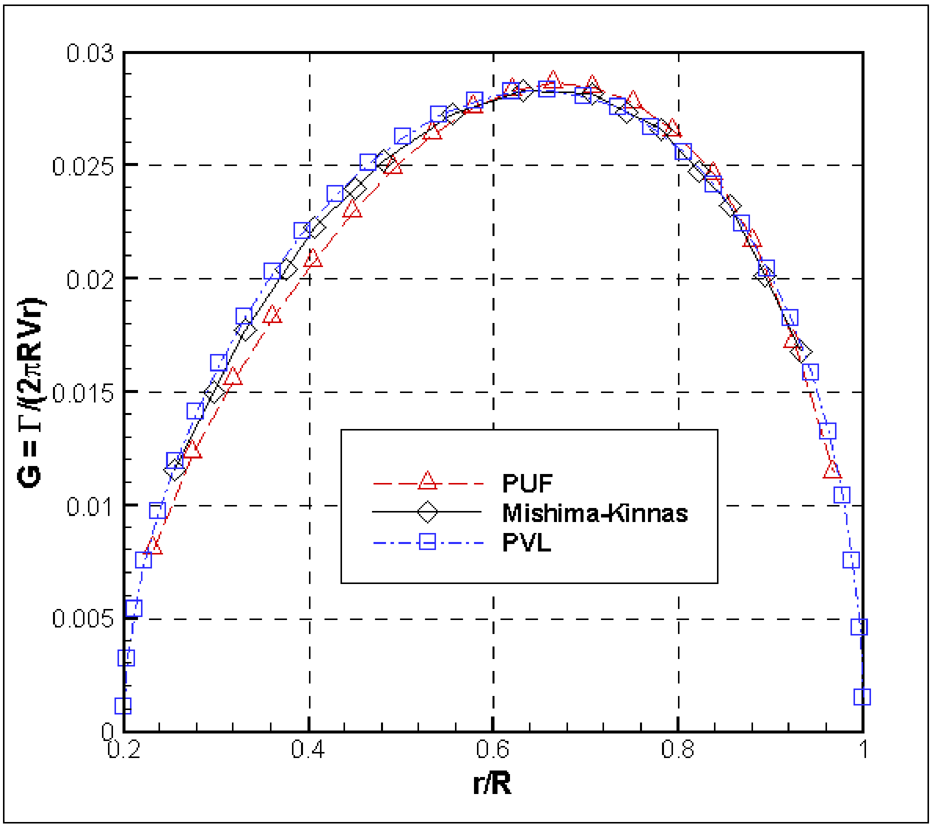


Figure 3.7 : Comparision of optimum circulation distribution obtained by PUF, PVL and Mishima and Kinnas

Table 3.6 : Comparision of PUF and PVL results with those of Mishima and Kinnas

	Present Method PUF	Present Method PVL	Mishima and Kinnas method
J	1.00	1.00	1.00
K_T	0.1453	0.1500	0.1500
K_Q	0.0292	0.0300	0.0304
η	0.7919	0.7957	0.7853

In the following figures it is shown different views of the optimum propeller blade obtained by PUF without hub effect.

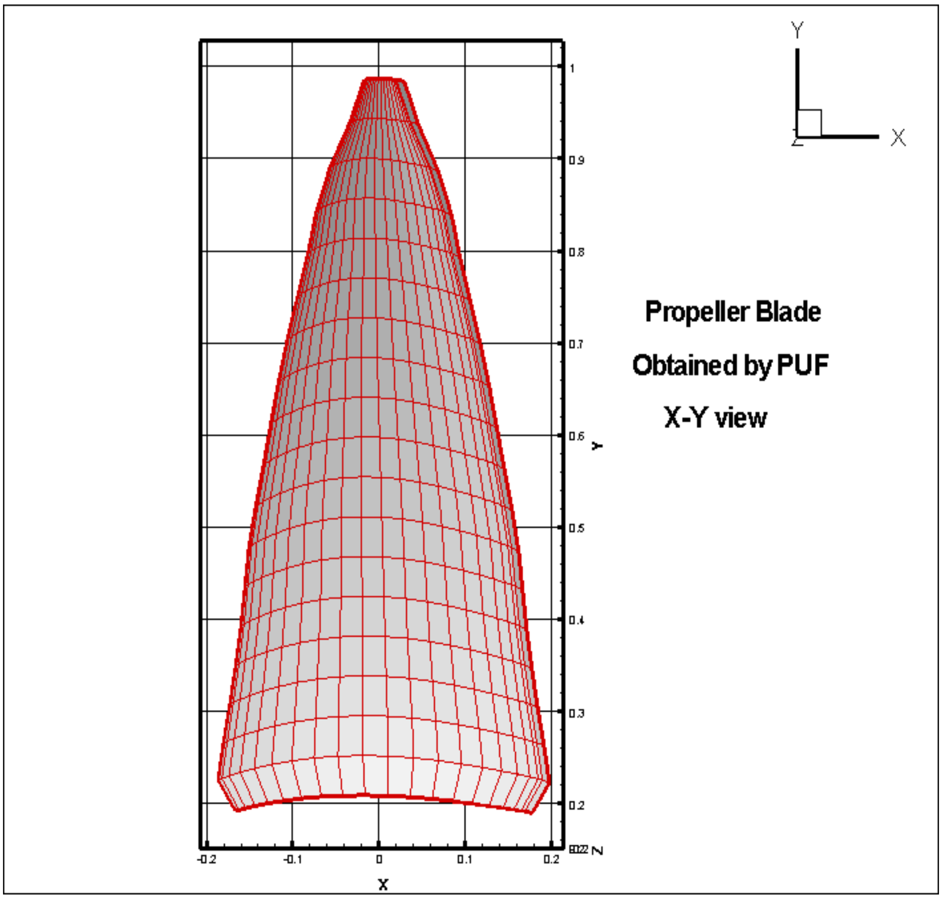


Figure 3.8 : X-Y view of optimum propeller blade

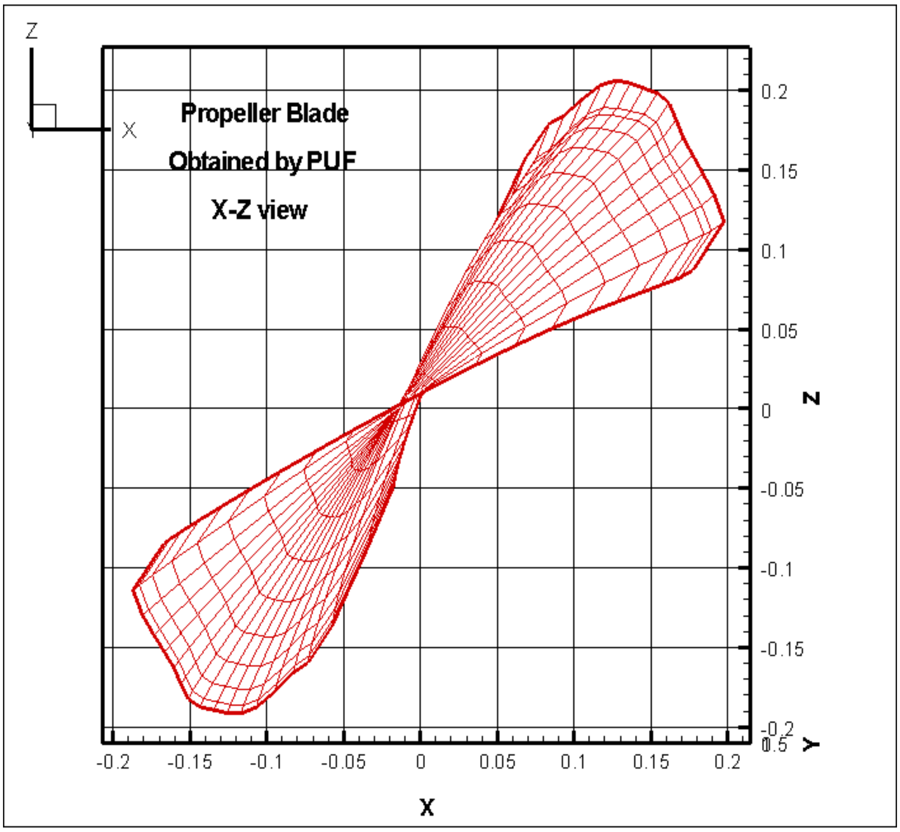


Figure 3.9 : X-Z view of optimum propeller blade

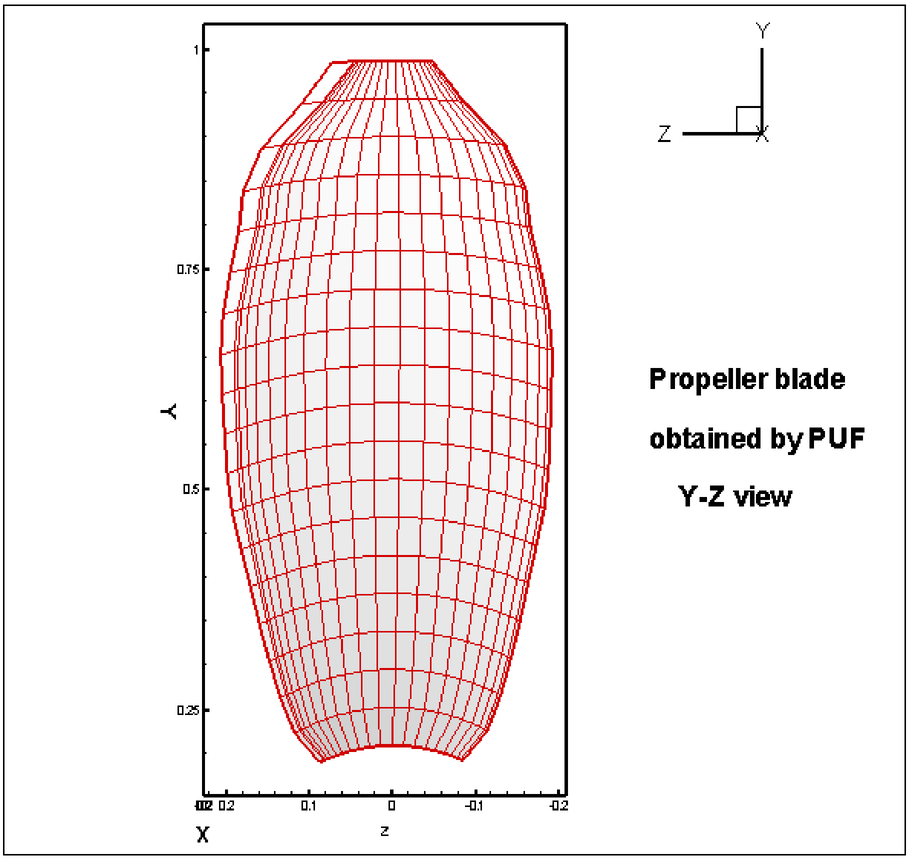


Figure 3.10 : Y-Z view of optimum propeller blade

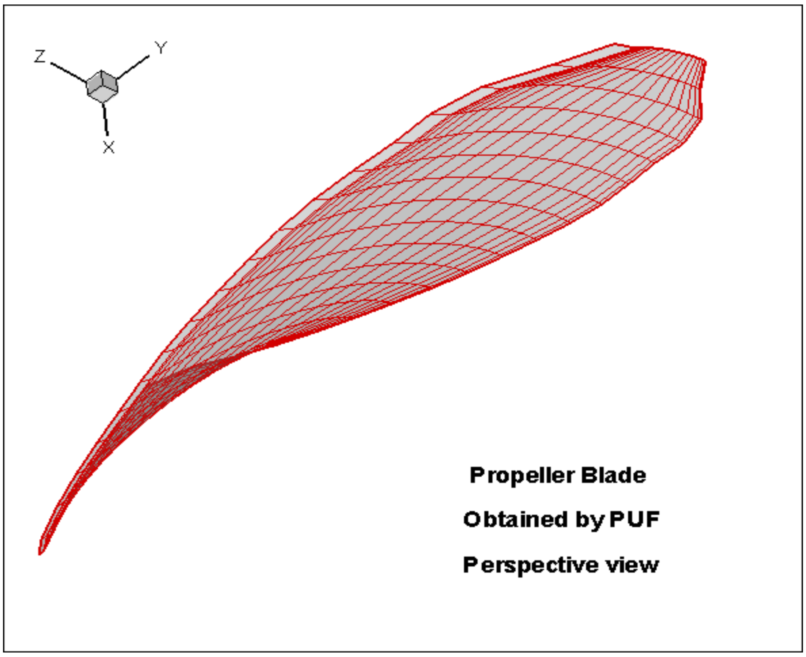


Figure 3.11 : Perspective view of optimum propeller blade

3.3.1 Calculations of circulation distribution with hub effect

As it is given at the beginning of the section 3.1 optimum circulation distribution has been calculated for sample propeller without hub effect. In this section, circulation distribution has been calculated by PVL and PUF programs with hub effect by taken same input value of optimum propeller geometry from the table 3.2 and comparison of the results are given following tables 3.7 and 3.8.

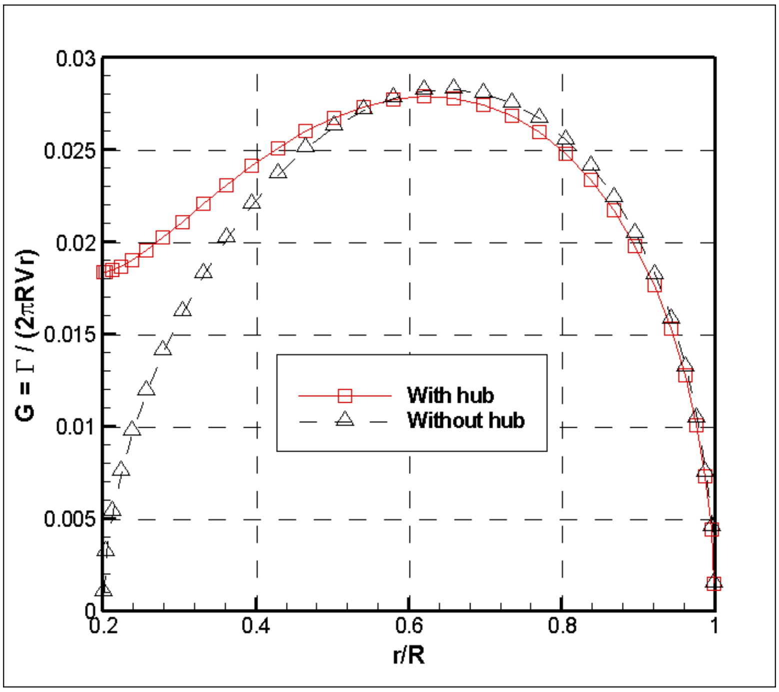


Figure 3.12 : Circulation distribution obtained from PVL without hub / with hub

Table 3.7 : Comparision of PVL results with with hub / without hub

	Without hub	With hub
J	1.00	1.00
K_T	0.1500	0.1500
K_Q	0.0300	0.0303
η	0.7957	0.7882

As it is clearly seen from above graph 3.12 and table 3.7 which are obtained by PVL program, the efficiency has been reduced slightly due to hub vortex drug.

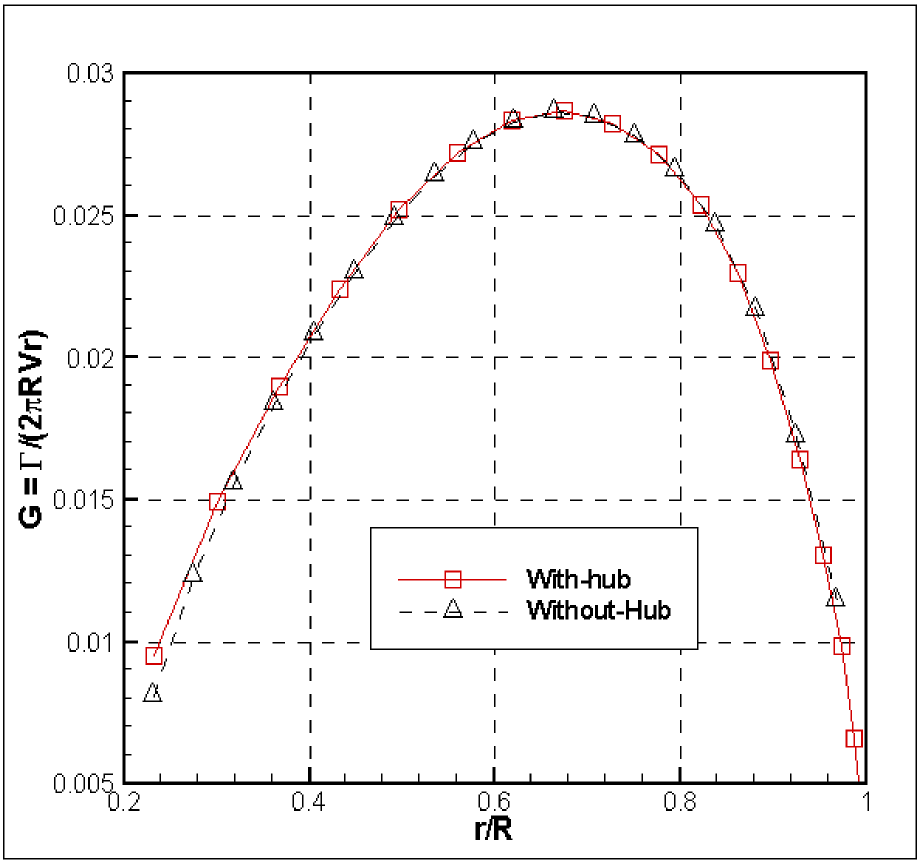


Figure 3.13 : Circulation distribution obtained from PUF without hub / with hub

Table 3.8 : Comparision of PUF results with with hub / without hub

	Without hub	With hub
J	1.00	1.00
K_T	0.1453	0.1457
K_Q	0.0292	0.0293
η	0.7919	0.7904

The results given in the table 3.8 indicate that the efficiency of the propeller has been reduced slightly due to hub vortex drug. Also, below it is displayed some perspective views of the propeller blades which have been obtained from PUF with hub and without hub effects.

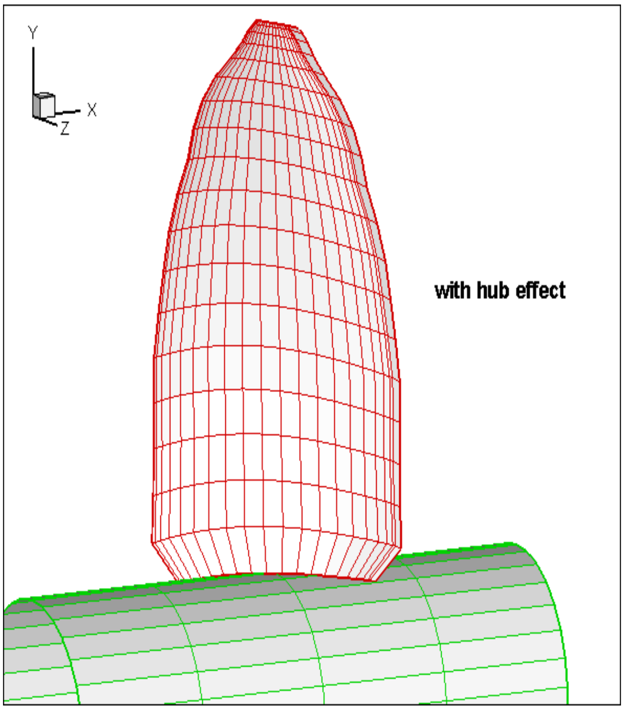


Figure 3.14 : Perspective views of propeller blades obtained by PUF with hub effect

3.3.2 Effects of blade numbers

In this section, it is shown effects of changing of blade numbers in the calculations with PVL and PUF programs by comparing with the first obtained optimum condition results.

3.3.2.1 Calculation with PVL

As it is clearly seen below table 3.11, the increasing of the blade number until 4 is result in increasing value of the efficiency slightly, but when blade number chose as 5, the efficiency has been slightly reduced for this optimum propeller design.

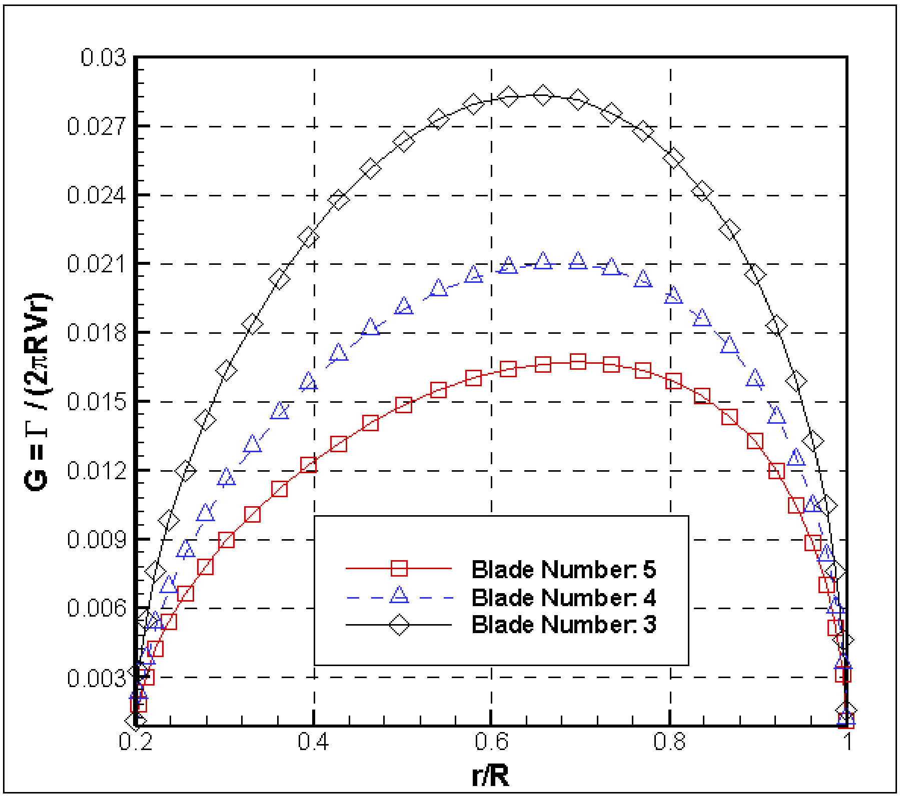


Figure 3.15 : Circulation distributions obtained by PVL with different blade numbers

Table 3.9 : Comparision of PVL results with different blade numbers

	3 Blade	4 Blade	5 Blade
J	1.00	1.00	1.00
K_T	0.1500	0.1500	0.1500
K_Q	0.0300	0.0299	0.0301
η	0.7957	0.7982	0.7932

3.3.2.2 Calculation with PUF

As is it understood from below shown result, the efficiency is decreasing with the increasing of the blade number in the calculations with PUF program.

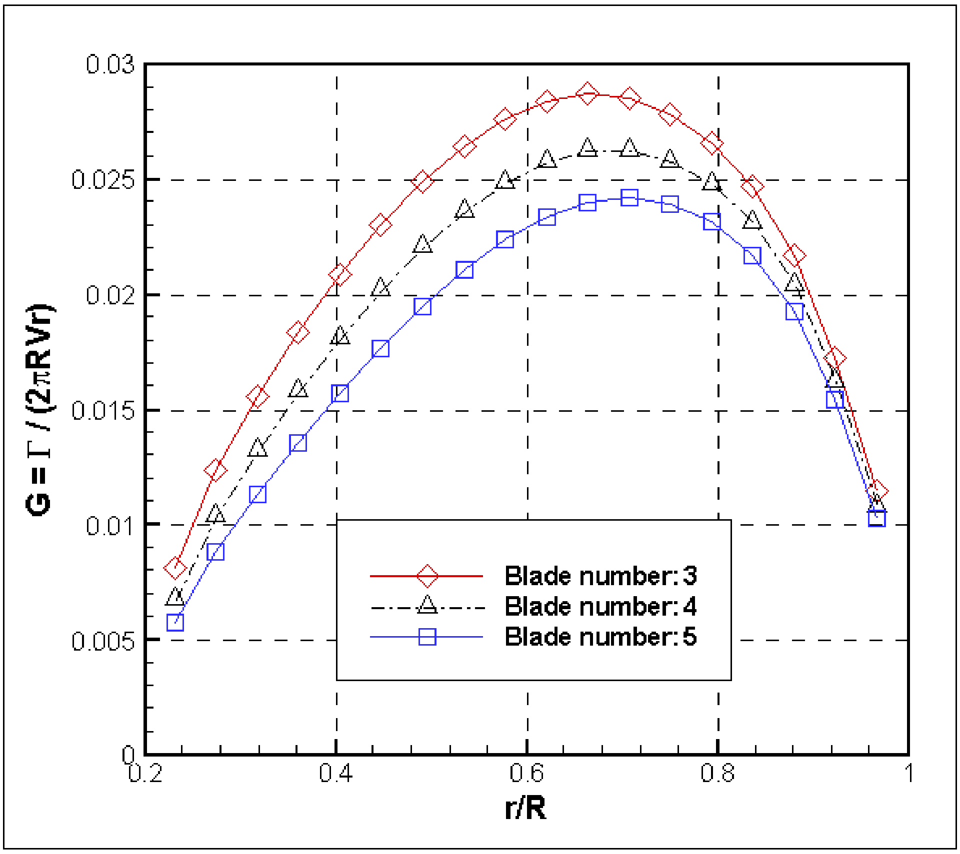


Figure 3.16 : Circulation distributions obtained by PUF with different blade numbers

Table 3.10 : Comparision of PUF results with different blade numbers

	3 Blade	4 Blade	5 Blade
J	1.00	1.00	1.00
K_T	0.1453	0.1832	0.2124
K_Q	0.0292	0.0377	0.0447
η	0.7919	0.7734	0.7563

Morover, it is also given comparisions of circulation distribution obtained by PVL and PUF programs for the different blade numbers in the following tables and figures.

- ✓ ***<u>Comparision for 3 Blade:</u>***

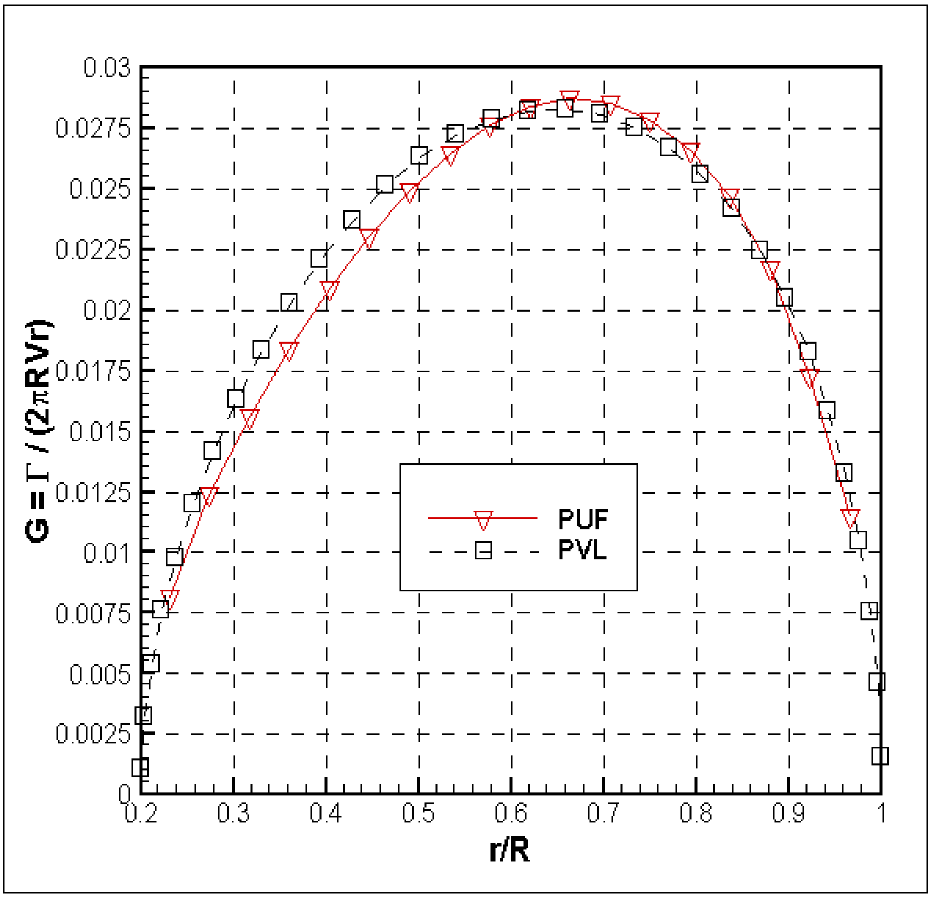


Figure 3.17 : Comparision of circulation distribution between PVL / PUF for 3 blade

Table 3.11 : Comparision of PUF and PVL results for 3 blade.

	PUF	PVL
J	1.00	1.00
K_T	0.1453	0.1500
K_Q	0.0292	0.0300
η	0.7919	0.7957

✓ ***Comparision for 4 Blade:***

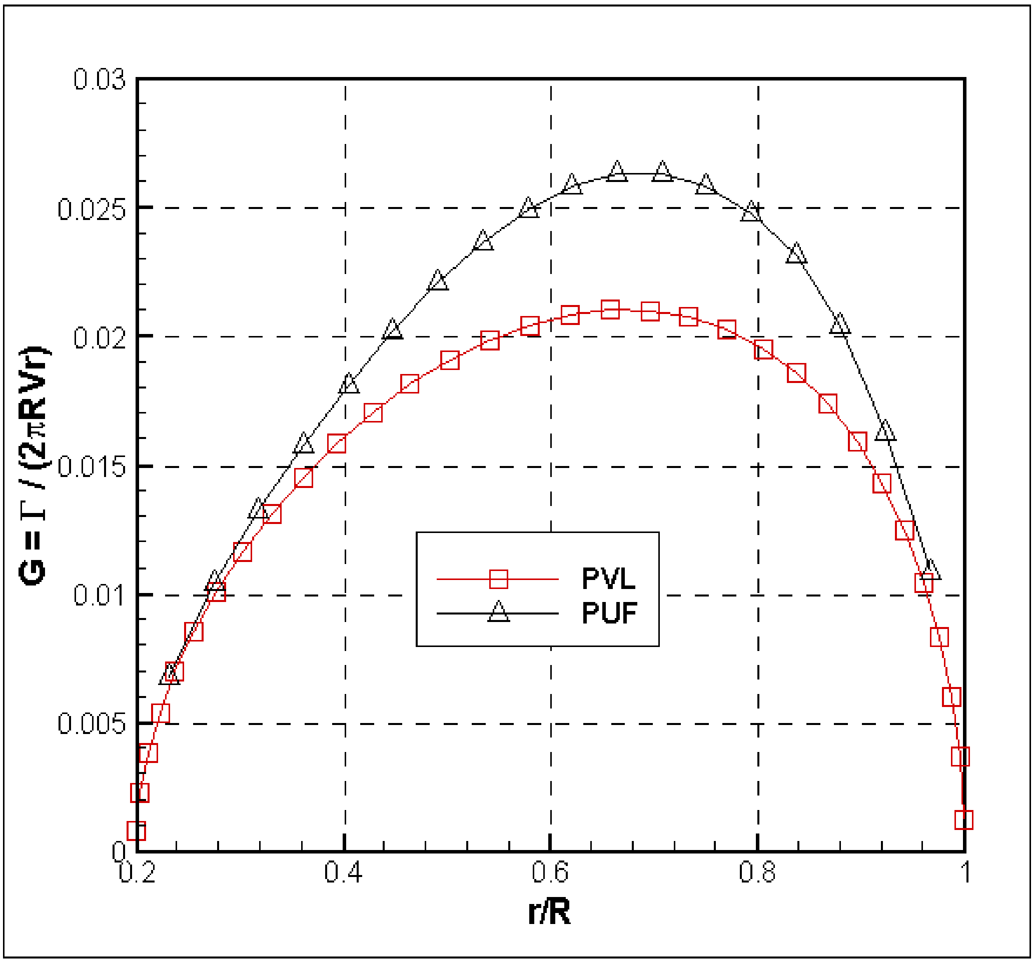


Figure 3.18 : Comparision of circulation distribution between PVL / PUF for 4 blade

Table 3.12 : Comparision of PUF and PVL results for 4 blade.

	PUF	PVL
J	1.00	1.00
K_T	0.1832	0.1500
K_Q	0.0377	0.0299
η	0.7734	0.7982

✓ ***Comparision for 5 Blade:***

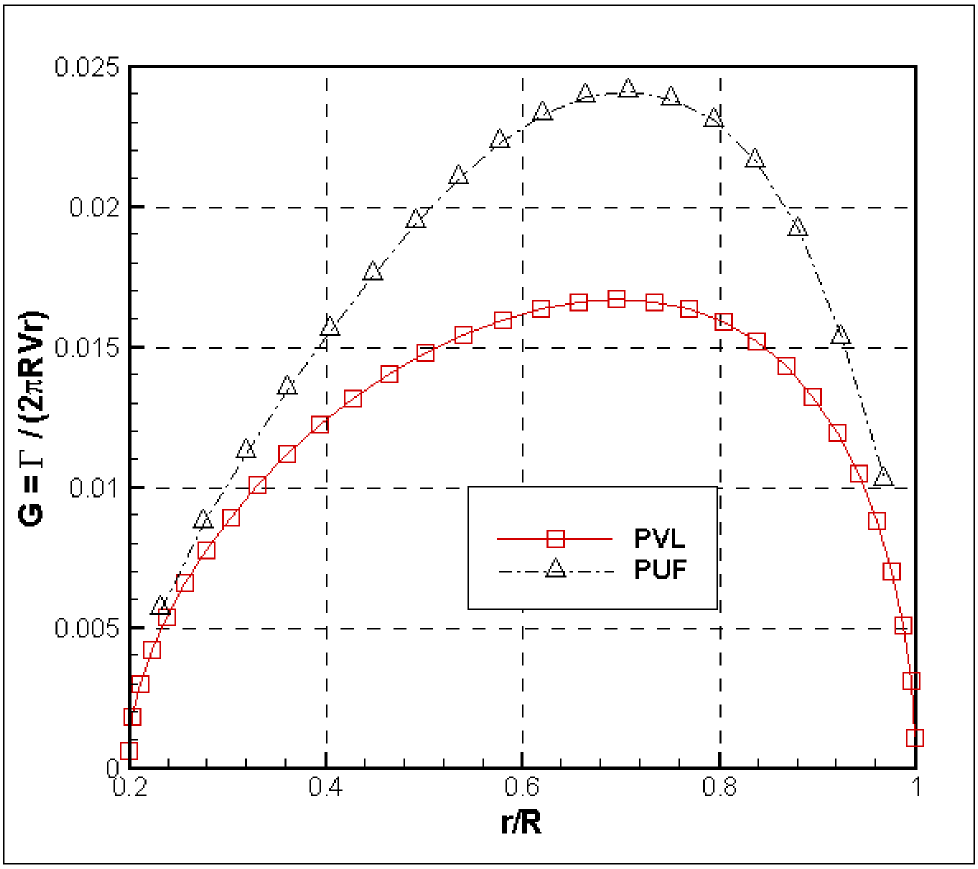


Figure 3.19 : Comparision of circulation distribution between PVL / PUF for 5 blade

Table 3.13 : Comparision of PUF and PVL results for 5 blade.

	PUF	PVL
J	1.00	1.00
K_T	0.2124	0.1500
K_Q	0.0447	0.0301
η	0.7563	0.7932

3.3.3 Changing of Spanwise Spacing

In this section, circulation distribution has been calculated with PUF program by changing spanwise spacing from uniform to cosine and it has been obtained results compared with each other.

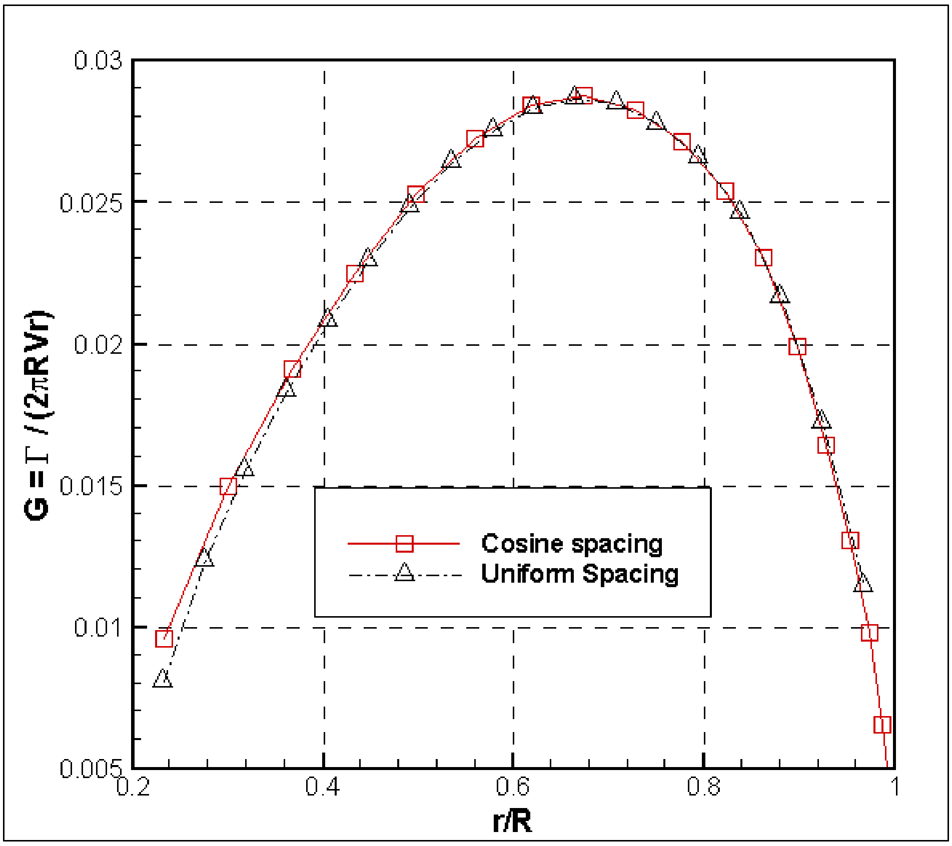


Figure 3.20 : Comparision of circulation distribution with uniform and cosine spacing

Table 3.14 : Comparision results of uniform / cosine spacing.

	uniform	cosine
J	1.00	1.00
K_T	0.1453	0.1459
K_Q	0.0292	0.0293
η	0.7919	0.7925

The efficiency has been slightly increased due to changing of spanwise spacing from uniform to cosine.

In the following figure 3.22, it is displayed X-Y views of propeller blades for uniform and cosine spacings.

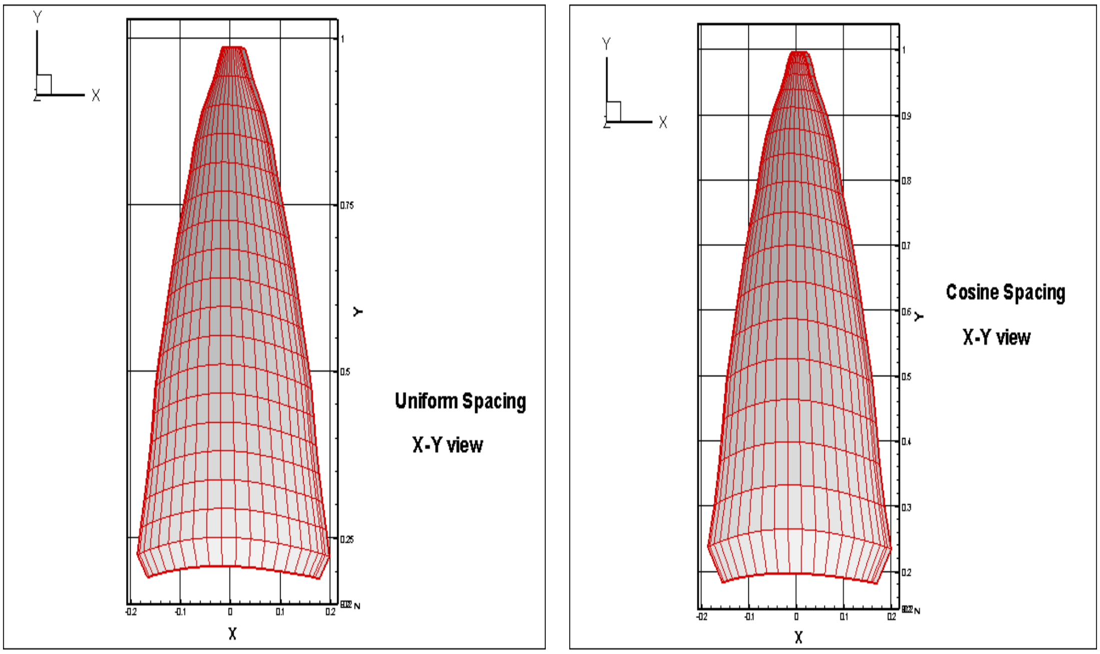


Figure 3.21 : X-Y views of propeller blades for uniform and cosine spacings.

4. CONCLUSION AND DISCUSSION

In the present thesis, it has been obtained an optimum circulation distribution around marine propellers by applying PVL and PUF programs and the results are compared with those of taken from Mishima and Kinnas (1997). The satisfaction between the results is good. In addition, different cases such as effect of number of spacing, effect of hub, effect of number of blades etc. have been examined with PVL and PUF programs, and compared with those of Mishima and Kinnas.

First, circulation distribution has been calculated by PVL / PUF programs and the results are compared with and without hub effects. Later, the effect of strip numbers has been analyzed with PVL program by changing the number of radius. Moreover, the effects of different blade numbers have been examined by using PVL / PUF and circulation distributions have been compared with each other. Finally, it is obtained optimum circulation distribution with PUF by using cosine spacing and compared with uniform spacing. It is found that if the number of blades is increased while keeping the thrust coefficient fixed, the circulation per blade is decreased for PVL running. On the other hand, the efficieny is first increased (for four blades) later decreased (for five blades) slightly. Additionally, for the PUF running the thrust coefficient is increased for increasing number of blades while keeping the blade geometric characteristics (pitch ratio and camber ratio) fixed.

As a future extension study, it is planned to take into consideration the non-uniform flow and cavitation effects for an optimum marine propeller design by using PVL and PUF programs.

REFERENCES

Anderson, J. D., 2005: Introduction to Flight. *Fifth Edition,* pp. 642 – 647, *McGraw - Hill Companies, Inc., Boston*.

Bal, S., 2009: Ship Propeller Hydrodynamics. *Lecture Notes.*

Bal, S., and Guner, M., 2009: Performance Analysis of Podded Propulsors, *Ocean Engineering, 36, 556-563*.

Carlton, J. S., 2007: Marine Propellers and Propulsion. *Second Edition,* pp. 31- 50, *Elsevier Ltd.*

Coney, W. B., 1992: Optimum Circulation Distribution for a Class of Marine Propulsors, *Journal of Ship Research,* 36(3):210-222

Ekinci, S., 2007: Examination of the Sheet Cavitation on the Marine Propeller Surface. *PhD Thesis,* Yildiz Technical University, Department of Naval Architecture and Naval Machinery Engineering.

Gerr, D., 1989: Propeller Handbook. *The Complete Reference for Choosing, Installing, and Understading Boat Propellers,* International Marine Publishing Company.

Harvald, A. S., 1991: Resistance and Propulsion of Ships. *Second Edition,* pp. 135 – 145, *Krieger Publishing Company, Malabar, Florida.*

Kerwin, E. J., and Greeley D. S., 1982: Numerical Methods for Propeller Design and Analysis in Steady Flow, *Trans*, SNAME, 90-415-53

Kerwin, E. J., 2001: Hydrofoils and Propellers. Massachusetts Institute of Technology, Mechanical Engineering, MIT Open Course Ware, *13.04 Lecture Notes.*

Mishima, S., and Kinnas, A. S., 1997: Application of Numerical Optimization Technique to the Design of Cavitating Propellers in Non-uniform Flow, *Journal of Ship Research, Vol. 41, No. 2.*

Olsen, S. A., 2001: Optimization of Propeller Using the Vortex Lattice Method, *PhD Thesis,* Technical University of Denmark, Department of Mechanical Engineering.

Url-1 <*http://www.propellerpages.com*>, accessed at 29.07.2010.

Url-2 <*http://www.eng-tips.com*>, accessed at 29.07.2010

APPENDICES

APPENDIX A.1: The Code of PVL Program (Kerwin, 2001).

```
PROGRAM PVL
!--------------------------------------------------------------------------------
! Propeller Vortex Lattice Lifting Line Code
! COPYRIGHT (C) 2001 JUSTIN E. KERWIN -------------------------------------------
      USE PVLMOD
      USE DUCKMOD
!------------------------- Declare the Variables ------------------------------
      IMPLICIT NONE
      CHARACTER*36 :: FNAME,LABEL
      CHARACTER*72 :: TITLE
      INTEGER :: MT,NX,ITER,NBLADE,N,M,KTRY,IERR,IHUB
      REAL :: KT,KQ,DEL,HRR,RCWG,RM,DTANB,EDISK,ADVCO,CTDES,HR,HT,CRP,
     1WAKE,CQ,CP,EFFY,HRF,CTH,RHV
      REAL, PARAMETER :: PI=3.1415927E00, TOL=0.000005, R2D=57.29578E00
      DOUBLE PRECISION :: TANBIW,RCW,RVW,UAIF,UTIF
      REAL, DIMENSION(:), ALLOCATABLE :: XR,XCHD,XCD,XVA,XVT,XRC,RV,
     1TANBV,RC,TANBC,VAV,VTV,VAC,VTC,TANBIV,TANBIC,UAW,UTW,B,G,UASTAR,
     2UTSTAR,T,CT,TANBXV,TANBXC,VBAV,VBAC,CD,CDC
      REAL, DIMENSION(:,:), ALLOCATABLE :: CHCUB,CDCUB,VACUB,VTCUB,UAHIF
     1,UTHIF,A
!------------------ Start reading the input data ------------------------------
!      WRITE(*,'(A)') 'ENTER INPUT FILE NAME..... '
!      READ(*,'(A)') FNAME
!      OPEN(2,FILE=FNAME,STATUS='OLD',FORM='FORMATTED')
      OPEN(2,FILE='inp.dat',STATUS='OLD',FORM='FORMATTED')
!      READ(2,'(A)') TITLE ! Title describing data file
       title='output'
      READ(2,*) MT ! Number of vortex lattice panels
      READ(2,*) ITER ! Number of iterations to align wake
      READ(2,*) IHUB ! Hub image flag. IHUB=0 : No hub image, IHUB=1 : Image hub
      READ(2,*) RHV ! Hub vortex radius/Hub radius. Only used if IHUB=1
      READ(2,*) NX ! Number of radii used to specify the input data
!------------- Allocate all the arrays before reading rest of input-------------
      ALLOCATE ( XR(NX),XCHD(NX),XCD(NX),XVA(NX),XVT(NX),XRC(NX) )
      ALLOCATE ( CHCUB(NX-1,5),CDCUB(NX-1,5),VACUB(NX-1,5),
       1VTCUB(NX-1,5))
      ALLOCATE ( RV(MT+1),TANBV(MT+1),RC(MT),TANBC(MT),VAV(MT+1),
       1VTV(MT+1),VAC(MT),VTC(MT),TANBIV(MT+1),TANBIC(MT),UAW(MT+1),
     2UTW(MT+1),UAHIF(MT,MT),UTHIF(MT,MT),A(MT,MT),B(MT),G(MT),
     3UASTAR(MT),UTSTAR(MT),T(ITER),CT(ITER),TANBXV(MT+1),TANBXC(MT),
       4VBAV(MT+1),VBAC(MT),CD(MT),CDC(MT) )
!----------------All arrays allocated. read in rest of input data --------------
      READ(2,*) NBLADE ! Number of blades
      READ(2,*) ADVCO ! Advance coefficient based on ship speed
      READ(2,*) CTDES ! Desires thrust coefficient CT (based on ship speed)
      READ(2,*) HR ! Unloading ratio at hub
      READ(2,*) HT ! Unloading ratio at tip
      READ(2,*) CRP ! Tangential velocity cancellation factor
      READ(2,'(A)') LABEL ! Alphanumeric label for output

!--------------------------------------------------------------------------------
! XR=Input radii r/R, XCHD=Input chord length c/D, XCD=Input viscous drag
! coefficient, Cd or Lift/Drag ratio, XVA,XVT=Input axial and tangential
! velocities, Va/Vs, V_t/Vs
!--------------------------------------------------------------------------------
      READ(2,*) (XR(N),XCHD(N),XCD(N),XVA(N),XVT(N),N=1,NX)
      CLOSE(2)
!-----Compute volumetric mean inflow velocity ratio VA/VS ----------------------
      WAKE=VOLWK(XR,XVA)
!-----Spline chord over radius using square root stretched coordinates----------
      XRC(:)=1.0-SQRT(1.0-XR(:))
      CALL UGLYDK(0,0,XRC,XCHD,0.0,0.0,CHCUB)
!-----Spline Drag Coefficient Cd, Inflow Vx, Vt using radial coordinate directly
      CALL UGLYDK(0,0,XR,XCD,0.0,0.0,CDCUB)
      CALL UGLYDK(0,0,XR,XVA,0.0,0.0,VACUB)
      CALL UGLYDK(0,0,XR,XVT,0.0,0.0,VTCUB)
!-----Compute cosine spaced vortex radii and get Va,Vt,tanB,Vt*tanB/Va----------
      DEL=PI/(2.0*REAL(MT))
      HRR=0.5*(XR(NX)-XR(1))
      DO M=1,MT+1
      RV(M)=XR(1)+HRR*(1.0-COS(REAL(2*(M-1))*DEL))
      CALL EVALDK(RV(M),VAV(M),VACUB)
      CALL EVALDK(RV(M),VTV(M),VTCUB)
      TANBV(M)=VAV(M)/((PI*RV(M)/ADVCO)+VTV(M))
      VBAV(M)=VTV(M)*TANBV(M)/VAV(M)
      END DO
!-----Cosine spaced control point radii: Evaluate c/D,Va,Vt,tanB,Cd,Vt*tanB/Va -
```

```
      DO M=1,MT
      RC(M)=XR(1)+HRR*(1.0-COS(REAL(2*M-1)*DEL))
      RCWG=1.0-SQRT(1.0-RC(M))
      CALL EVALDK(RCWG,CDC(M),CHCUB)
      CALL EVALDK(RC(M),VAC(M),VACUB)
      CALL EVALDK(RC(M),VTC(M),VTCUB)
      TANBC(M)=VAC(M)/((PI*RC(M)/ADVCO)+VTC(M))
      CALL EVALDK(RC(M),CD(M),CDCUB)
      VBAC(M)=VTC(M)*TANBC(M)/VAC(M)
      END DO
!-----First estimate of tanBi based on 90 percent of actuator disk efficiency --
      EDISK=1.8/(1.0+SQRT(1.0+CTDES/WAKE**2))
      TANBXV(:)=TANBV(:)*SQRT(WAKE/(VAV(:)-VBAV(:)))/EDISK ! Lerbs optimum-----
      TANBXC(:)=TANBC(:)*SQRT(WAKE/(VAC(:)-VBAC(:)))/EDISK
!-----Unload hub and tip as specified by input HR and HT -----------------------
      RM=0.5*(XR(1)+XR(NX)) ! Mid-radius. Unloading is quadratic, starting here
      DO M=1,MT+1
      IF(RV(M).LT.RM) THEN
      HRF=HR
      ELSE
      HRF=HT
      END IF
      DTANB=HRF*(TANBXV(M)-TANBV(M))*((RV(M)-RM)/(XR(1)-RM))**2
      TANBXV(M)=TANBXV(M)-DTANB
      END DO
      DO M=1,MT
      IF(RC(M).LT.RM) THEN
      HRF=HR
      ELSE
      HRF=HT
      END IF
      DTANB=HRF*(TANBXC(M)-TANBC(M))*((RC(M)-RM)/(XR(1)-RM))**2
      TANBXC(M)=TANBXC(M)-DTANB
      END DO
!-------------------------------------------------------------------------------
! Iterations to scale tanBi to get desired value of thrust coefficient
!-------------------------------------------------------------------------------
       DO KTRY=1,ITER
       IF(KTRY.EQ.1) THEN
       T(KTRY)=1.0 ! T(KTRY) is the scale factor to apply to tanBi
      ELSE IF(KTRY.EQ.2) THEN
      T(KTRY)=1.0+(CTDES-CT(1))/(5.0*CTDES) ! Guess for second iteration
      ELSE IF(KTRY.GT.2) THEN
      T(KTRY)=T(KTRY-1)+(T(KTRY-1)-T(KTRY-2))*(CTDES-CT(KTRY-1))/
       1(CT(KTRY-1)-CT(KTRY-2)) ! Secant method for remaining iters
      END IF
       TANBIV(:)=T(KTRY)*TANBXV(:) ! Scale tanBi at the vortex radii
       TANBIC(:)=T(KTRY)*TANBXC(:) ! Scale tanbi at the control points
!-------------------------------------------------------------------------------
! Compute axial and tangential horseshoe influence coefficients !
!-------------------------------------------------------------------------------
       DO M=1,MT
       RCW=RC(M)
       DO N=1,MT+1
!--------------Induction of trailing vortices shed at RV(N)---------------------
       TANBIW=TANBIV(N)
       RVW=RV(N)
       CALL WRENCH(NBLADE,TANBIW,RCW,RVW,UAIF,UTIF)
       UAW(N)=-UAIF/(2.0*(RC(M)-RV(N)))
       UTIF=UTIF*CRP ! Note if CRP=0, the tangential velocity is zero--
       UTW(N)=UTIF/(2.0*(RC(M)-RV(N)))
!--------------Induction of corresponding hub-image trailing vortices (if any)--
       IF(IHUB/=0) THEN
       RVW=XR(1)**2/RV(N)
       TANBIW=TANBIV(1)*RV(1)/RVW
       CALL WRENCH(NBLADE,TANBIW,RCW,RVW,UAIF,UTIF)
       UAW(N)=UAW(N)+UAIF/(2.0*(RC(M)-RVW))
       UTIF=UTIF*CRP
       UTW(N)=UTW(N)-UTIF/(2.0*(RC(M)-RVW))
       END IF
       END DO
!-----------Final step in building influence functions--------------------------
       DO N=1,MT
       UAHIF(M,N)=UAW(N+1)-UAW(N)
       UTHIF(M,N)=UTW(N+1)-UTW(N)
       END DO
       END DO
```

```
!-------------------------------------------------------------------------------
! Solve simultaneous equations for circulation strengths G(M) !
!-------------------------------------------------------------------------------
       DO M=1,MT
       B(M)=VAC(M)*((TANBIC(M)/TANBC(M))-1.0) ! Right-hand side
       DO N=1,MT
       A(M,N)=UAHIF(M,N)-UTHIF(M,N)*TANBIC(M) ! Coefficient matrix
       END DO
       END DO
       CALL SIMEQN(A,B,G,IERR) ! Simultaneous equation solver
       IF(IERR/=0) EXIT ! Error return for singular matrix
!-------------------------------------------------------------------------------
! Evaluate the induced velocities from the circulation GM) !
!-------------------------------------------------------------------------------
       DO M=1,MT
       UASTAR(M)=0.0
       UTSTAR(M)=0.0
       DO N=1,MT
       UASTAR(M)=UASTAR(M)+G(N)*UAHIF(M,N)
       UTSTAR(M)=UTSTAR(M)+G(N)*UTHIF(M,N)
       END DO
       END DO
!-------------------------------------------------------------------------------
! Compute the forces and test if Ct has converged to desired value !
!-------------------------------------------------------------------------------
       CALL FORCES(NBLADE,MT,ADVCO,WAKE,RV,RC,TANBC,UASTAR,UTSTAR,VAC,
       1CDC,CD,G,CT(KTRY),CQ,CP,KT,KQ,EFFY,RHV,CTH,IHUB)
       WRITE(*,'(I5,'' CT='',F10.5,'' DESIRED VALUE='',F10.5)') KTRY,
       1CT(KTRY),CTDES
       IF(ABS(CT(KTRY)-CTDES)<TOL) EXIT
       END DO
!-----Stop run if matrix is sigular-------------------------------------------
       IF(IERR/=0) THEN
       WRITE(*,'(A)') ' MATRIX SINGULAR. RUN TERMINATED..... '
       STOP
       ELSE
!-------------------------------------------------------------------------------
! Output results to Tecplot file !
!-------------------------------------------------------------------------------
       WRITE(*,'(//'' EFFICIENCY ='',F8.4)') EFFY
       WRITE(*,'('' Kt, Kq'',F8.4,F8.5)') KT,KQ
       WRITE(*,'('' HUB DRAG COEFFICIENT Cth='',F8.4)') CTH
       OPEN(1,FILE='APLOT.PLT',STATUS='UNKNOWN',FORM='FORMATTED')
       WRITE(1,'(A)') ' VARIABLES="R","G","VA","VT","UA","UT","BETA",
       1"BETAI","CDC","CD" '
       WRITE(1,'('' TEXT X=0.5, Y=0.50, T=" Ct='',F8.4,'' "'')')
       1CT(KTRY)
       WRITE(1,'('' TEXT X=0.5, Y=0.46, T=" Cp='',F8.4,'' "'')')
       1CP
       WRITE(1,'('' TEXT X=0.5, Y=0.42, T=" Kt='',F8.4,'' "'')')
       1KT
       WRITE(1,'('' TEXT X=0.5, Y=0.38, T=" Kq='',F8.4,'' "'')')KQ
       WRITE(1,'('' TEXT X=0.5, Y=0.34, T=" Va/Vs='',F8.4,'' "'')')WAKE
       WRITE(1,'('' TEXT X=0.5, Y=0.30, T=" E='',F8.4,'' "'')')EFFY
       WRITE(1,'('' TEXT X=0.5, Y=0.26, T="'',A,'' " '')')TITLE
       WRITE(1,'(F10.5,F10.6,4F10.5,2F10.3,2F10.5)') (RC(M),G(M),VAC(M),
       1VTC(M),UASTAR(M),UTSTAR(M),R2D*ATAN(TANBC(M)),R2D*ATAN(TANBIC(M))
     2,CDC(M),CD(M),M=1,MT)
       CLOSE(1)
       END IF
!      STOP
       END PROGRAM PVL

      SUBROUTINE UGLYDK(NCL,NCR,XIN,YIN,ESL,ESR,CUBIC)
!-----------------------------------------------------------------------
!     Last change:  JEK   5 Mar 98    2:25 pm
!     Fortran 90 version of original Duck Series written by J.E.Kerwin
!    Arguments:
!    NCL  - Integer describing the left end condition for the spline
!           0-Second derivative specified in ESL
!           1-Rate of change of curvature is continuous at second input
!             point(best option if you do not know what NCL should be)
!           2-First derivative specified in ESL
!   NCR   - Same as for NCL but for right end of spline
!   XIN,YIN - Arrays of input point pairs(NIN in each array)
!   ESL   - First or second derivative at left end of curve (if NCL=0,2)
!   ESR   - Same as ESL but for right end of spline
```

```
!   NOTE!!- Positive slope at left end is the spline angling UP from
!            left to right.  Negative slope at the right end is the
!            spline angling DOWN from left to right.
!   CUBIC - Array of dimension (5,(NIN-1)) which will contain the
!           cubic coefficients on completion of the subroutine,
!           as well as the first NIN-1 x coordinates of the base points.
!------------------------------------------------------------------------

!------------------ Declare the variables ------------------------------
      IMPLICIT NONE
      REAL, DIMENSION(:), INTENT(IN) :: XIN,YIN
      REAL, DIMENSION(:,:), INTENT(OUT) :: CUBIC
      REAL, INTENT(IN) :: ESL,ESR
      INTEGER, INTENT(IN) :: NCL,NCR
      REAL, DIMENSION(:), ALLOCATABLE :: H,D,AU,AM,AL,X,S
      REAL, PARAMETER :: HALF=0.5E00, TWO=2.0E00, THREE=3.0E00
      REAL, PARAMETER :: SIX=6.0E00
      REAL HFACT
      INTEGER NIN,K,L,N

!-----Allocate the local arrays for the coeffcinet matrix, RHS, solution
      NIN=SIZE(XIN)
      ALLOCATE (H(NIN-1),D(NIN-1),AU(NIN-3),AM(NIN-2),AL(NIN-3),
     &          X(NIN-2),S(NIN))

!-----Compute the intervals, H, and the divided differences, D ---------
      DO N=1,NIN-1
         H(N)=XIN(N+1)-XIN(N)
         D(N)=(YIN(N+1)-YIN(N))/H(N)
      END DO

!-----Set up the principal diagonal (AM) and right hand side (S)-----
      DO N=1,NIN-2
         AM(N)=TWO*(H(N)+H(N+1))
         S(N)=SIX*(D(N+1)-D(N))
      END DO
!-----Set up the upper (AU) and lower (AL) diagonals--------------------
      DO N=1,NIN-3
         AL(N)=H(N+1)
         AU(N)=H(N+1)
      END DO

!-----Modify the first equation based on the left end condition---------
      IF(NCL.EQ.0) THEN         ! Second derivative specified as ESL
         S(1)=S(1)-ESL*H(1)
      ELSE IF(NCL.EQ.1) THEN    ! Extrapolated curvature end condition--
         AM(1)=AM(1)+H(1)*(H(1)+H(2))/H(2)
         AU(1)=AU(1)-H(1)**2/H(2)
      ELSE IF(NCL.EQ.2) THEN    ! First derivative specified as ESL
         AM(1)=AM(1)-HALF*H(1)
         S(1)=S(1)-THREE*(D(1)-ESL)
      END IF

!-----Modify the last equation based on the right end condition--------
      IF(NCR.EQ.0) THEN         ! Second derivative specified as ESR
         S(NIN-2)=S(NIN-2)-ESR*H(NIN-1)
      ELSE IF(NCR.EQ.1) THEN    ! Extrapolated curvature end condition--
         AM(NIN-2)=AM(NIN-2)+H(NIN-1)*(H(NIN-2)+H(NIN-1))/H(NIN-2)
         AL(NIN-3)=AL(NIN-3)-H(NIN-1)**2/H(NIN-2)
      ELSE IF(NCR.EQ.2) THEN    ! First derivative specified as ESR
         AM(NIN-2)=AM(NIN-2)-HALF*H(NIN-1)
         S(NIN-2)=S(NIN-2)+THREE*(D(NIN-1)-ESR)
      END IF

!-----Solve the tri-diagonal system: First pass eliminates lower diag---
      DO K=2,NIN-2
         AL(K-1)=AL(K-1)/AM(K-1)
         AM(K)=AM(K)-AL(K-1)*AU(K-1)
         S(K)=S(K)-AL(K-1)*S(K-1)
      END DO
!-----Second pass back substitutes along principal diagonal-------------
      X(NIN-2)=S(NIN-2)/AM(NIN-2)
      DO L=2,NIN-2
         K=NIN-L-1
         X(K)=(S(K)-AU(K)*X(K+1))/AM(K)
      END DO
```

```
!-----Generate array of second derivatives at base points S(N)----------
!-----First get S(1) from the left end condition------------------------
      IF(NCL.EQ.0) THEN
         S(1)=ESL
      ELSE IF(NCL.EQ.1) THEN
         HFACT=H(1)/H(2)
         S(1)=(1.0+HFACT)*X(1)-HFACT*X(2)
      ELSE IF(NCL.EQ.2) THEN
         S(1)=-HALF*X(1)+THREE*(D(1)-ESL)/H(1)
      END IF

!-----Copy the interior values from the solution X----------------------
      DO N=2,NIN-1
         S(N)=X(N-1)
      END DO

!-----Finally, get S(NIN) from the right end condition------------------
      IF(NCR.EQ.0) THEN
         S(NIN)=ESR
      ELSE IF(NCR.EQ.1) THEN
         HFACT=H(NIN-1)/H(NIN-2)
         S(NIN)=(1.0+HFACT)*S(NIN-1)-HFACT*S(NIN-2)
      ELSE IF(NCR.EQ.2) THEN
         S(NIN)=-HALF*S(NIN-1)-THREE*(D(NIN-1)-ESR)/H(NIN-1)
      END IF

!-----Form the output CUBIC array---------------------------------------
      DO N=1,NIN-1
         CUBIC(N,1)=(S(N+1)-S(N))/(SIX*H(N))
         CUBIC(N,2)=HALF*S(N)
         CUBIC(N,3)=D(N)-H(N)*(TWO*S(N)+S(N+1))/SIX
         CUBIC(N,4)=YIN(N)
         CUBIC(N,5)=XIN(N)
      END DO

      DEALLOCATE ( H,D,AU,AM,AL,X,S )

      RETURN
      END SUBROUTINE UGLYDK

      SUBROUTINE EVALDKA(X,Y,C)
!-----------------------------------------------------------------------
!     Last change:  JEK  25 Feb 98    8:39 am
!     Fortran 90 version of original Duck series. Evaluates a spline
!     Arguments:
!     X -  Array of length NOUT containing desired x coordinates
!     Y -  Array of length NOUT : EVALDK will return values of spline
!     C -  Array of size (NIN-1,5) containing spline cubic from UGLYDK
!          Note that the 5th column of C contains the x coordinates of
!          the original base points passed to UGLYDK.
!-----------------------------------------------------------------------
      IMPLICIT NONE
      REAL, DIMENSION(:), INTENT(IN) :: X
      REAL, DIMENSION(:), INTENT(OUT) :: Y
      REAL, DIMENSION(:,:), INTENT(IN) :: C
      REAL P
      INTEGER NIN,NOUT,N,J,JI
      NIN=SIZE(C,1)+1
      NOUT=SIZE(X)

      DO  N=1,NOUT
         IF(X(N).LE.C(2,5)) THEN
            JI=1
         ELSE IF(X(N).GE.C(NIN-1,5)) THEN
            JI=NIN-1
         ELSE
            DO J=2,NIN-1
               JI=J
               IF(X(N).GE.C(J,5).AND.X(N).LT.C(J+1,5))   EXIT
            END DO
         END IF
         P=X(N)-C(JI,5)
         Y(N)=C(JI,4)+P*(C(JI,3)+P*(C(JI,2)+P*C(JI,1)))
      END DO

      RETURN
      END SUBROUTINE EVALDKA
```

```
      SUBROUTINE EVALDKP(X,Y,C)
!-----------------------------------------------------------------------
!     Las
t change:  CLW made into one point version 4/21/98
!     Fortran 90 version of original Duck series. Evaluates a spline
!     Arguments:
!     X -  Point containing desired x coordinates
!     Y -  Point  : EVALDK will return the one value of spline
!     C -  Array of size (NIN-1,5) containing spline cubic from UGLYDK
!          Note that the 5th column of C contains the x coordinates of
!          the original base points passed to UGLYDK.
!-----------------------------------------------------------------------
      IMPLICIT NONE
      REAL,  INTENT(IN) :: X
      REAL, INTENT(OUT) :: Y
      REAL, DIMENSION(:,:), INTENT(IN) :: C
      REAL P
      INTEGER NIN,J,JI
      NIN=SIZE(C,1)+1

      IF(X.LE.C(2,5)) THEN
         JI=1
      ELSE IF(X.GE.C(NIN-1,5)) THEN
         JI=NIN-1
      ELSE
         DO J=2,NIN-1
            JI=J
            IF(X.GE.C(J,5).AND.X.LT.C(J+1,5))  EXIT
         END DO
      END IF
      P=X-C(JI,5)
      Y=C(JI,4)+P*(C(JI,3)+P*(C(JI,2)+P*C(JI,1)))

      RETURN
      END SUBROUTINE EVALDKP

      SUBROUTINE INTDK1(XL,XU,YDX,CUBIC)
      USE DUCKMOD, ONLY : ISPAN
      IMPLICIT NONE
!------------------- Declare the arguments ------------------------------------
      REAL,  INTENT(IN) :: XL,XU  ! Upper and lower limits of integral
      REAL, DIMENSION(:,:), INTENT(INOUT) :: CUBIC  ! Spline cubic array
      REAL, INTENT(OUT) :: YDX   ! Integral frol XL to XU
!------------------- Declare the local variables ------------------------------
      INTEGER :: JU,JL,J
      REAL H1,H2,H3,H4

      JL=ISPAN(XL,CUBIC)
      JU=ISPAN(XU,CUBIC)
!--------------- Evaluate integral at the lower limit --------------------------
      H1=XL-CUBIC(JL,5)
      H2=H1**2
      H3=H1*H2
      H4=H2**2
      YDX=-CUBIC(JL,1)/4.0*H4-CUBIC(JL,2)/3.0*H3-CUBIC(JL,3)/2.0*H2
     1-CUBIC(JL,4)*H1
!--------------- Evaluate integral at the upper limit --------------------------
      H1=XU-CUBIC(JU,5)
      H2=H1**2
      H3=H1*H2
      H4=H2**2
      YDX=YDX+CUBIC(JU,1)/4.0*H4+CUBIC(JU,2)/3.0*H3+CUBIC(JU,3)/2.0*H2
     1+CUBIC(JU,4)*H1
!--------------- Evaluate integral over intermediate spans, if any -------------
      IF(JU>JL) THEN
         DO J=JL,JU-1
            H1=CUBIC(J+1,5)-CUBIC(J,5)
            H2=H1**2
            H3=H1*H2
            H4=H2**2
            YDX=YDX+CUBIC(J,1)/4.0*H4+CUBIC(J,2)/3.0*H3+CUBIC(J,3)/2.0
       1*H2+CUBIC(J,4)*H1
         END DO
      END IF
      RETURN
```

```
END SUBROUTINE INTDK1

INTEGER FUNCTION ISPAN(X,CUBIC)
IMPLICIT NONE
REAL, INTENT(IN) :: X
REAL, DIMENSION(:,:), INTENT(IN) :: CUBIC
INTEGER NM,NLOW,NHIGH,MID
NM=SIZE(CUBIC,1)
IF(X<CUBIC(2,5)) THEN
   MID=1     ! X is in the first span, or out of range to the left.
ELSE IF(X>=CUBIC(NM,5)) THEN
   MID=NM
ELSE
   NLOW=2    ! Do binary search for the span index
   NHIGH=NM
   MID=(NLOW+NHIGH)/2
   DO WHILE (X<CUBIC(MID,5).OR.X>=CUBIC(MID+1,5))
      IF(X<CUBIC(MID,5)) THEN
         NHIGH=MID
      ELSE
         NLOW=MID
      END IF
      MID=(NLOW+NHIGH)/2
   END DO
END IF
ISPAN=MID
RETURN
END FUNCTION ISPAN

SUBROUTINE FACTOR(W,IPIVOT,IERR)
IMPLICIT NONE
REAL, DIMENSION(:,:), INTENT(INOUT) :: W
INTEGER, DIMENSION(:), INTENT(OUT) :: IPIVOT
INTEGER, INTENT(OUT) :: IERR

REAL, ALLOCATABLE, DIMENSION(:) :: D
REAL :: ROWMAX,AWIKOV,COLMAX,RATIO
INTEGER :: N,NM1,I,J,K,KP1,IP,IPK

N=SIZE(W,1)
ALLOCATE(D(N))
IERR=1
DO I=1,N
   IPIVOT(I)=I
   ROWMAX=0.
   DO J=1,N
      ROWMAX=MAX(ROWMAX,ABS(W(I,J)))
   END DO
   IF(ROWMAX==0.0) THEN
      IERR=2
      EXIT
   END IF
   D(I)=ROWMAX
END DO

IF(IERR==1) THEN
   NM1=N-1
   IF(NM1==0.0) RETURN
   DO K=1,NM1
      J=K
      KP1=K+1
      IP=IPIVOT(K)
      COLMAX=ABS(W(IP,K))/D(IP)
      DO I=KP1,N
         IP=IPIVOT(I)
         AWIKOV=ABS(W(IP,K))/D(IP)
         IF(AWIKOV>COLMAX) THEN
            COLMAX=AWIKOV
            J=I
         END IF
      END DO
      IF(COLMAX==0.0) THEN
         IERR=2
         EXIT
      END IF
      IPK=IPIVOT(J)
      IPIVOT(J)=IPIVOT(K)
```

```
            IPIVOT(K)=IPK
            DO I=KP1,N
               IP=IPIVOT(I)
               W(IP,K)=W(IP,K)/W(IPK,K)
               RATIO=-W(IP,K)
               DO J=KP1,N
                  W(IP,J)=RATIO*W(IPK,J)+W(IP,J)
               END DO
            END DO
         END DO
      END IF
      IF(W(IP,N)==0.0) IERR=2
      RETURN
      END SUBROUTINE FACTOR

      SUBROUTINE SUBST(W,B,X,IPIVOT)
      IMPLICIT NONE
      REAL, DIMENSION(:,:), INTENT(IN) :: W
      REAL,DIMENSION(:), INTENT(IN) :: B
      REAL, DIMENSION(:), INTENT(OUT) :: X
      INTEGER, DIMENSION(:), INTENT(IN) :: IPIVOT

      INTEGER :: N,IP,K,KM1,J,NP1MK,KP1
      REAL :: SUMT
      N=SIZE(W,1)
      IF(N==1) THEN
         X(1)=B(1)/W(1,1)
      ELSE
         IP=IPIVOT(1)
         X(1)=B(IP)
         DO K=2,N
            IP=IPIVOT(K)
            KM1=K-1
            SUMT=0.
            DO J=1,KM1
               SUMT=W(IP,J)*X(J)+SUMT
            END DO
            X(K)=B(IP)-SUMT
         END DO
         X(N)=X(N)/W(IP,N)
         K=N
         DO NP1MK=2,N
            KP1=K
            K=K-1
            IP=IPIVOT(K)
            SUMT=0.0
            DO J=KP1,N
               SUMT=W(IP,J)*X(J)+SUMT
            END DO
            X(K)=(X(K)-SUMT)/W(IP,K)
         END DO
      END IF
      RETURN
      END SUBROUTINE SUBST

       SUBROUTINE FORCES(NBLADE,MCP,ADVCO,WAKE,RV,RC,TANBC,UASTAR,UTSTAR
       1,VA,CHORD,CD,G,CT,CQ,CP,KT,KQ,EFFY,RHV,CTH,IHUB)
       IMPLICIT NONE
!--------------------- Declare the arguments ----------------------------------
       INTEGER, INTENT(IN) :: NBLADE,MCP,IHUB
       REAL, INTENT(IN) :: ADVCO,WAKE,RHV
       REAL, DIMENSION(:), INTENT(IN) :: RV,RC,TANBC,UASTAR,UTSTAR,VA,
       1CHORD,CD,G
       REAL, INTENT(OUT) :: CT,CQ,CP,KT,KQ,EFFY,CTH
!--------------------- Declare the local variables ----------------------------
       REAL, PARAMETER :: PI=3.1415927E00, TWO=2.0E00, FOUR=4.0E00,
       1EIGHT=8.0E00
       REAL :: DR,VSTAR,VTSTAR,VASTAR,VSTRSQ,DVISC,FKJ
       INTEGER :: M
       LOGICAL :: CD_LD
       CD_LD=.TRUE. ! Default: Input CD interpreted as viscous drag coefficient
       IF(CD(1)>1.0) CD_LD=.FALSE. ! CD(1)>1 signals that input is L/D ----------
       CT=0.0
       CQ=0.0
       DO M=1,MCP
       DR=RV(M+1)-RV(M)
       VTSTAR=VA(M)/TANBC(M)+UTSTAR(M)
```

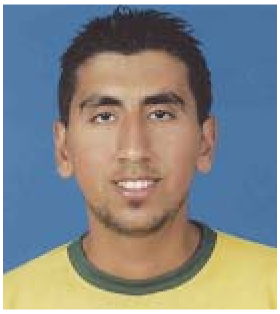
```
        VASTAR=VA(M)+UASTAR(M)
        VSTRSQ=VTSTAR**2+VASTAR**2
        VSTAR=SQRT(VSTRSQ)
        IF(CD_LD) THEN ! Interpret CD as viscous drag coefficient, Cd---------
        DVISC=(VSTRSQ*CHORD(M)*CD(M))/(TWO*PI)
        ELSE ! Interpret CD as the lift/drag ratio L/D -------------
        FKJ=VSTAR*G(M)
        DVISC=FKJ/CD(M)
        END IF
        CT=CT+(VTSTAR*G(M)-DVISC*VASTAR/VSTAR)*DR
        CQ=CQ+(VASTAR*G(M)+DVISC*VTSTAR/VSTAR)*RC(M)*DR
        END DO
        IF(IHUB/=0) THEN ! Add hub vortex drag if hub image is present ----------
        CTH=0.5*(LOG(1.0/RHV)+3.0)*(REAL(NBLADE)*G(1))**2
        ELSE
        CTH=0.0
        END IF
        CT=CT*FOUR*REAL(NBLADE)-CTH
        CQ=CQ*TWO*REAL(NBLADE)
        CP=CQ*TWO*PI/ADVCO
        KT=CT*ADVCO**2*PI/EIGHT
        KQ=CQ*ADVCO**2*PI/EIGHT
        EFFY=CT*WAKE/CP
        RETURN

        END

!     Last change:  JEK  25 Apr 2001    8:48 pm
      SUBROUTINE WRENCH(NB,TANB,RC,RV,UA,UT)
      IMPLICIT NONE
!------------------Declare variables in argument list -------------------------
      INTEGER, INTENT(IN) :: NB
      DOUBLE PRECISION, INTENT(IN) :: TANB,RC,RV
      DOUBLE PRECISION, INTENT(OUT) :: UA,UT

!------------------Declare local variables ------------------------------------
      DOUBLE PRECISION :: C25=0.25D00, ONE=1.0D00, C15=1.5D00,
       1TWO=2.0D00,THREE=3.0D00, NINE=9.0D00, C24=24.0D00
      DOUBLE PRECISION :: BL,XG,ETA,H,XS,T,V,W,AE,U,R,XX,Y,Z,AF,AA,
       1RATIO,AG,AB
      BL=DBLE(NB)

!-----Return infinite blade result if NB>20 JEK 9/19/98 ----------------
      IF(NB.GT.20) THEN
         IF(RC.GT.RV) THEN
            UA=0.0
            UT=BL*(RC-RV)/RC
         ELSE
            UA=-BL*(RC-RV)/(RV*TANB)
            UT=0.0
         END IF
         RETURN
      END IF
!-----End of infinite blade patch -------------------------------------

      XG=ONE/TANB
      ETA=RV/RC
      H=XG/ETA
      XS=ONE+H**2
      T=SQRT(XS)
      V=ONE+XG**2
      W=SQRT(V)
      AE=T-W
      U=EXP(AE)
      R=(((T-ONE)/H*(XG/(W-ONE)))*U)**BL
      XX=(ONE/(TWO*BL*XG))*((V/XS)**C25)
      Y=((NINE*XG**2)+TWO)/(V**C15)+((THREE*H**2-TWO)/(XS**C15))
      Z=ONE/(C24*BL)*Y
      IF(H.GE.XG) THEN
         AF=ONE+ONE/(R-ONE)
         AA=XX*(ONE/(R-ONE)-Z*LOG(AF))
         UA=TWO*BL**2*XG*H*(ONE-ETA)*AA
         UT=BL*(ONE-ETA)*(ONE+TWO*BL*XG*AA)
      ELSE
         IF(R.GT.1.0D-12) THEN
            RATIO=ONE/(ONE/R-ONE)
         ELSE
```

```
            RATIO=0.0
         END IF
         AG=ONE+RATIO
         AB=-XX*(RATIO+Z*LOG(AG))
         UA=BL*XG*(ONE-ONE/ETA)*(ONE-TWO*BL*XG*AB)
         UT=TWO*BL**2*XG*(ONE-ETA)*AB
      END IF
      RETURN
      END SUBROUTINE WRENCH

      REAL FUNCTION VOLWK(XR,XVA)
      USE DUCKMOD
      IMPLICIT NONE
      REAL :: YDX
      INTEGER :: NX,N
      REAL, DIMENSION(:), INTENT(IN) :: XR,XVA
      REAL, DIMENSION(:), ALLOCATABLE :: Y
      REAL, DIMENSION(:,:), ALLOCATABLE :: VWCUB
      NX=SIZE(XR)
      ALLOCATE ( Y(NX),VWCUB(NX-1,5) )
      Y(:)=XR(:)*XVA(:)
      CALL UGLYDK(0,0,XR,Y,0.0,0.0,VWCUB)
      CALL INTDK1(XR(1),XR(NX),YDX,VWCUB)
      VOLWK=2.0*YDX/(1.0-XR(1)**2)
      DEALLOCATE (Y,VWCUB)
      RETURN
      END FUNCTION VOLWK

!     Last change:  JEK   2 Mar 99   11:44 am
      SUBROUTINE SIMEQN(A,B,X,IERR)
!-----This is a Fortran 90 version of Dave Greeley's FACTOR & SUBST (combined)--
      IMPLICIT NONE

!------------------- Declare variables in argument list ------------------------
      REAL, DIMENSION(:,:), INTENT(INOUT) :: A    ! Coefficient matrix
      REAL, DIMENSION(:), INTENT(INOUT)   :: B    ! Right hand side vector
      REAL, DIMENSION(:), INTENT(OUT)     :: X    ! Solution vector
      INTEGER, INTENT(OUT)                :: IERR ! Error flag

!-------------------- Declare local variables ---------------------------------
      REAL, DIMENSION(:), ALLOCATABLE    :: D       ! Row swapping storage
      INTEGER, DIMENSION(:), ALLOCATABLE :: IPIVOT  ! Row swapping index
      INTEGER :: NEQ,I,J,NM1,K,KM1,KP1,IP,IPK,NP1MK
      REAL ::  ROWMAX,COLMAX,AWIKOV,SUMM,RATIO

!--------------------Allocate local arrays--------------------------------------
      NEQ=SIZE(B)
      ALLOCATE ( D(NEQ),IPIVOT(NEQ) )
      IERR=1

!-----Find |maximum| element in each row, and exit if a zero row is detected----
      IERR=1            ! Initialize error flag to 1 (denotes bad matrix)-------
      DO I=1,NEQ
         IPIVOT(I)=I
         ROWMAX=0.0
         DO J=1,NEQ
            ROWMAX=MAX(ROWMAX,ABS(A(I,J)))
         END DO
         IF(ROWMAX==0.0) RETURN     ! IERR=1 Matrix is singular ----------------
         D(I)=ROWMAX
      END DO

      NM1=NEQ-1
      IF(NM1>0) THEN                ! Otherwise special case of one equation----
         DO K=1,NM1
            J=K
            KP1=K+1
            IP=IPIVOT(K)
            COLMAX=ABS(A(IP,K))/D(IP)
            DO I=KP1,NEQ
               IP=IPIVOT(I)
               AWIKOV=ABS(A(IP,K))/D(IP)
               IF(AWIKOV>COLMAX) THEN
                  COLMAX=AWIKOV
                  J=I
               END IF
            END DO
```

```
         IF(COLMAX==0.0) RETURN  ! IERR=1 Matrix is singular ----------------
         IPK=IPIVOT(J)
         IPIVOT(J)=IPIVOT(K)
         IPIVOT(K)=IPK
         DO I=KP1,NEQ
            IP=IPIVOT(I)
            A(IP,K)=A(IP,K)/A(IPK,K)
            RATIO=-A(IP,K)
            DO J=KP1,NEQ
               A(IP,J)=RATIO*A(IPK,J)+A(IP,J)
            END DO
         END DO
      END DO
      IF(A(IP,NEQ).EQ.0.) RETURN   ! IERR=1 Matrix is singular --------------
   END IF
   IERR=0                          ! Matrix survived singular tests ---------

!------------------Back substitute to obtain solution (X) ----------------------

   IF(NEQ==1) THEN         ! Special case of one equation again--------------
      X(1)=B(1)/A(1,1)
   ELSE

      IP=IPIVOT(1)
      X(1)=B(IP)
      DO K=2,NEQ
         IP=IPIVOT(K)
         KM1=K-1
         SUMM=0.0
         DO J=1,KM1
            SUMM=A(IP,J)*X(J)+SUMM
         END DO
         X(K)=B(IP)-SUMM
      END DO
      X(NEQ)=X(NEQ)/A(IP,NEQ)
      K=NEQ
      DO NP1MK=2,NEQ
         KP1=K
         K=K-1
         IP=IPIVOT(K)
         SUMM=0.0
         DO J=KP1,NEQ
            SUMM=A(IP,J)*X(J)+SUMM
         END DO
         X(K)=(X(K)-SUMM)/A(IP,K)
      END DO
   END IF

   DEALLOCATE (D,IPIVOT)
   RETURN
   END SUBROUTINE SIMEQN
```

Figure A.1 : Code of PVL program

CURRICULUM VITAE



Candidate's full name: **Barış BİÇER**

Place and date of birth: **İzmir, 27.06.1984**

Universities attended: **İstanbul Technical University**