Using Blade Element Momentum Theory for Propeller Analysis

Nicholas McCaw

March 21, 2017

Abstract

Chapter 1

Introduction

This report outlines the work completed during the 1st year of the EngD program. This chapter will give an outline of the content of the report as well as the motivation for doing the project.

The second chapter of the report gives a brief overview of the literature which has been reviewed, the third chapter details the background theory of blade element momentum theory, Timoshenko beam theory and finite element analysis. The fourth chapter will discuss the code developed using these techniques and some preliminary results obtained.

Propellers are the most widely used form of propulsion for marine vehicles. It is desired for the noise to be reduced. This is to increase efficiency and to reduce the environmental impact.

To lessen the noise of the propeller it is vital to reduce vibration. Vibration is caused by the fluid interacting with the structure. The geometry of the propeller blade causes a pressure difference between its two surfaces. This pressure difference causes a force in the direction toward the low pressure side, normal to the surface of the blade. This phenomena is used to produce thrust.

However the load on the propeller blade causes the blade to deform. The changed blade shape then causes the load produced to be changed which, in turn, causes the shape of the propeller blade to change. This is the fundamentals of hydroelastics.

A computationally inexpensive way to obtain the propeller blade loads is to use blade element momentum theory. To obtain blade deformations the blade is treated as a simple 1D Timoshenko Beam and Timoshenko beam theory is used along side 1D finite element analysis. Both these techniques will be discussed in more detail in chapter 3. Finally the code produced will be discussed in chapter 4

Chapter 2

Literature Review

Firstly the International Towing Tank Conference (ITTC) (CITATION) notes where reviewed. This was mainly to review the state of the art marine technology.

The ITTC propulsion committee was reviewed. This detailed the most up to date propulsion technologies and current research trends. Furthermore it detailed areas where more research was required. The first item of new technology was counter-rotating propellers which have the advantage over single propellers due to having two propellers sharing the propulsive force hence a reduction is rotation speed is obtained(INSERT CITATION). This is beneficial as the reduced reduction speed will result in reduced vibration and hence noise. Flexible blade propulsors are also of high interest to researchers. This is because having fully composite propeller blades offer significant weight saving (TTC Citation). Furthermore composites can be passively tailored during loading thus improving performance. This is of particular interest to researchers as the examples of propellers made from composite materials have been shown to have a higher efficiency (BLACK 2011) and cavitation speed (Kane 2001).

This area of hydroelastics is of interest to the project therefore the paper by Young (YOUNG 2006) was read further. This paper described the numerical model and solution procedure for modelling the hydroelastic response of composite marine propellers. The numerical model was a 3D boundary element method with the total flow velocity composed of inflow velocity and a propeller induced perturbation. The solution procedure involved computing the hydrodynamic added mass and damping. The pressure is calculated using the boundary element method. The deformations can then be calculated and applied to the new blade geometry. This process is repeated until the solution converges. A similar procedure is implemented in the code described in chapter 4.

The ITTC proceeding then went on to describe the need for research and development. The areas where:

1. Model and fullscale measurements of propulsors in off-design conditions

2. Full scale measurements of ship propulsive gain due to use of energy saving

devices.

3. Propulsive performances on composite propeller at full scale and model scale with measurement of blade deformation and torque.

4. full-scale measurement on hybrid contra-rotating shaft pod propulsors.

5. CFD simulation on the effect of varying Reynolds numnber on the performance of blade sections.

6. full scale measurement of waterjet inlet flow velocity fields.

Points 3. and 5. relate closely to the project undertaken this year.

The ITTC proceedings had a large section based on energy saving devices such as: wake equalizing ducts and pre-swirl stators. This section was read however it was deemed out of the scope of this project.

Other new technologies where described in the proceedings including: surface piercing propellers (HIMEI ET AL 2013), High speed vehicles and contracted and loaded tip propellers.

After this review it was decided to focus on the propeller blade deformation and CFD of Reynolds number effects. To model the Reynolds number effect using CFD a CFD package must be used. It was decided that open source software would be desirable hence OpenFOAM was used.

2.1   OpenFOAM Tutorials

I was decided that the OpenFOAM would be the CFD software of choice due to the open source nature. This means that the software can be easily changed to suit the needs of the user. Also the software is free making it cheaply used on many cores.

This sub-section will describe the tutorials undertaken using OpenFOAM.The first tutorials objective is to introduce the meshing and simulation of a simple test case. The test case used in this example is a lid-driven cavity flow. Firstly the mesh is created. This is done by defining the vertices of the block the user wishes to populate with cells. The number of cells are then defined in the x,y and z directions with any grading defined along each direction. For the initial case a uniform grid is created. Finally the wall boundary conditions are defined with the sides and bottom set to fixed and the top wall set to moving. The mesh is then created using the 'blockMesh' command. Once the mesh has been generated the fluid flow conditions can be analysed. There are two files for this test case: a pressure file and a velocity file. Within the pressure file the dimensions of the pressure field (i.e kinematic pressue), the kinematic pressure field value (zero for the initial condition) and the wall boundary conditions are all defined. The fixed walls and moving wall are set to have a normal gradient of pressure of zero. Similar conditions are set in the velocity field file. The kinematic viscosity ν was then set to fix the Reynolds number. The time-step must

then be selected such that the simulation is stable. When running 'icoFOAM' stability and accuracy are ensured by keeping the Courant number less than one throughout the domain. This is done using the equation:

$$Co = \frac{\delta t |U|}{\delta x} \tag{2.1}$$

When the grid is uniform i.e δx is constant this is a simple condition to meet. However in non-uniform grids this becomes slightly more difficult.

The appropriate timestep δt can therefore be calculated using equation 2.1. The finite volume discretisation solver can then be specified. The mesh is then viewed in ParaView to ensure it has been generated correctly. The simulation is then run using icoFOAM. The residuals and maximum Courant numbers are displayed as the simulation is running. Once the simulation is complete the pressures can be plotted as shown in 2.1
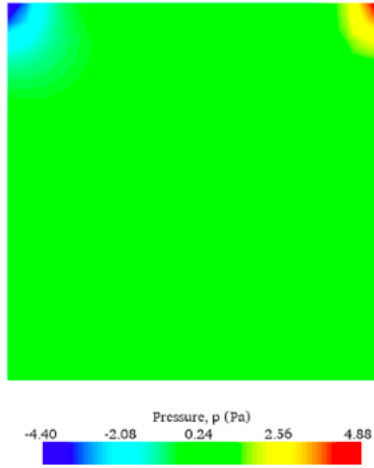

Figure 2.1: Pressures in Cavity test case

It can be seen from 2.1 that there is a low pressure at the top left corner of the cavity with higher pressure at the top right corner. This is as expected because the flow is moving from left to right within the cavity. The flow features can be seen in greater detail if the velocity vectors are shown as in figure 2.2.
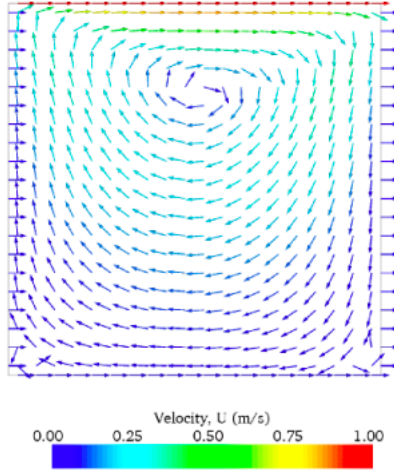
4

Figure 2.2: Vectors in Cavity test case

Here it is seen that a small vortex has been created within the cavity. However it is noted that the resolution of the vectors are not good. It is therefore desirable to increase the number of mesh cells. This is done by going to the mesh file and increasing the number of cells. As seen in figure 2.3 the resolution of vectors has increased.
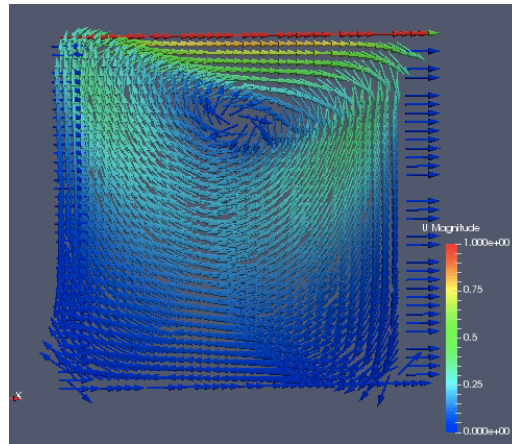
Figure 2.3: Vectors of cavity case with finer mesh

As the mesh has now been refined the details of the vortex can be seen far more clearly. This is, perhaps, not the most useful for this case however when using more complex geometry smaller flow features may be created and if the mesh is not sufficiently fine, the features may be lost. // When studying fluid flow, most of the interesting flow features occur at the wall with the flow

5

becoming less interesting as the distance from the wall increases. To conserve processing power it therefore beneficial to introduce a mesh grading. This is having a finer mesh at the wall with the mesh becoming more coarse as we move away from the wall. / To do this in OpenFOAM in this case the cavity is split into 4 blocks. Additional vertices are created in the mesh file to define the blocks. The mesh for each block is defined. For this case simple grading is introduced. For each block the mesh is defined to be twice as fine at the wall compared to the other end of the block. The number of cells has been reduced with this grading. Due to the change in the cell size the Courant number will change as seen from equation 2.1. The smallest cell size must then be computed then the timestep changed to ensure the Courant number remains below 1.

To further reduce computational expense the results of the previous simulation can be mapped to the new mesh and the simulation time can be started from this point. The graded mesh is shown in figure 2.4.
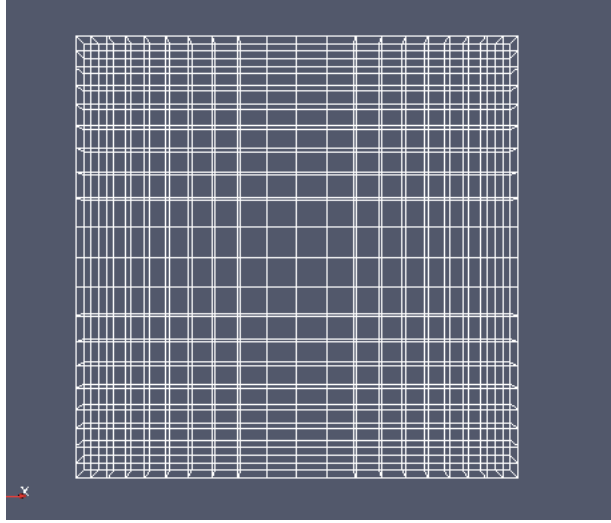
Figure 2.4: Graded Mesh For Cavity Case

It is clearly seen that the mesh is far finer at the walls compared to the centre of the cavity.

The Reynolds number is then increased an order of magnitude. This is achieved by reducing the kinematic viscosity ν by a factor of 10. This is clear by the equation for Reynolds number.

$$Re = \frac{UL}{\nu} \tag{2.2}$$

Where U is the flow speed and L is the characteristic length. Due to the increase in Reynolds number the overall simulation time is increased. As the Reynolds number is increased the flow changes regime from laminar to turbulent. This means that the variations in flow velocity will be very high hence the mesh will

need to be very fine. This is impractical hence Reynolds averaged turbulence models will need to be used. The turbulence models calculate the mean flow behaviour and the random fluctuations. For this case the $k - \epsilon$ model will be used for a Reynolds number of 10^4.

The second OpenFOAM tutorial looked at the stress analysis of a plate with a hole. This was thought to be useful as the project will be looking at the structural effects as well as fluid flow.

This case was a simple linear-elastic steady state case with the geometry shown in figure 2.5
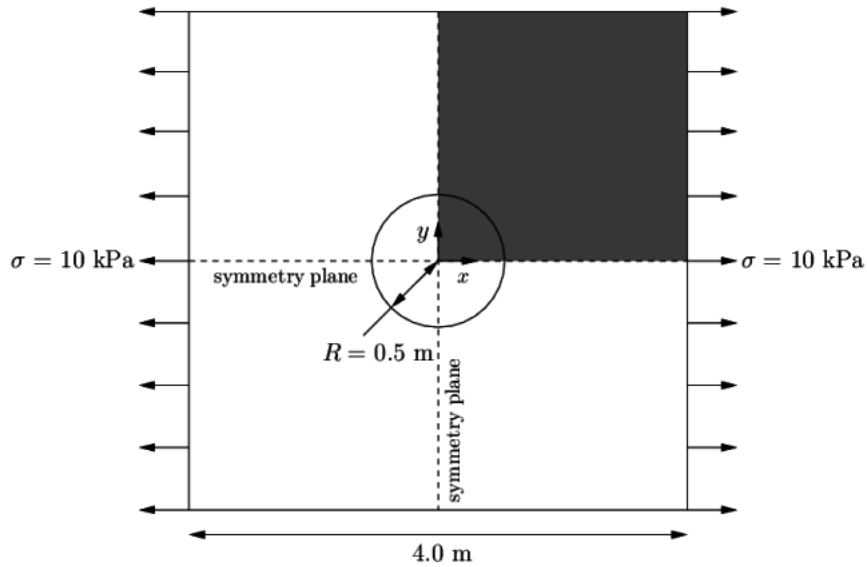
Figure 2.5: Geometry for stress analysis case

Note there are two lines of symmetry here hence only one quarter of the object need be modelled. The mesh for this is created in a similar way to the cavity case. However for this case there are curved edges. This is accounted for by having 5 blocks with vertices at corners between blocks and the midpoint of the curved edges as shown in figure 2.6

The curved cells are created using the keyword *edges* in the mesh generation file. The edges between the vertices which define the curve set to *arc*. In this file the down and left faces are set to symmetry planes to reflect the geometry of the entire plate. This creates the mesh as shown in figure 2.7

The mechanical properties of the system can then be defined such as density ρ, Youngs modulus E and Poissons ratio ν

The code is then run using solidDisplacementFoam. The stress field over the
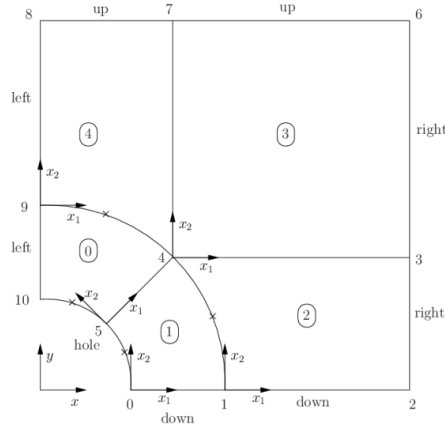
Figure 2.6: Block and Vertex definitions.
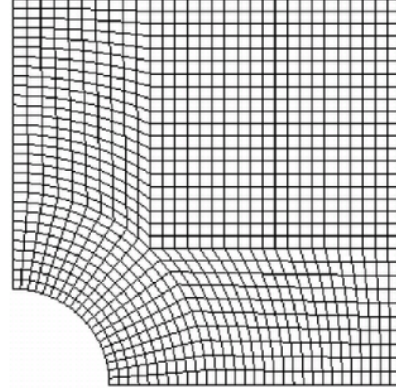
Figure 2.7: Mesh for stress case

plate can then be viewed using paraview this can be seen in figure 2.8

It can be seen that there is a stress concentration at the top of the hole with decreased stress at the side. This is because the stress through the plate acts like a fluid and is deflected around the hole.

It is also desirable to compare the numerical results to the analytical solution. This done through OpenFOAM by using a *singleGraph* file. The results are then compared and shown on figure 2.9

As seen in figure 2.9 the numerical solution slightly underestimates the stress at the edge of the plate and overestimated the stress at the hole. This can be improved by introducing a finer grid or having mesh grading as described earlier in this chapter
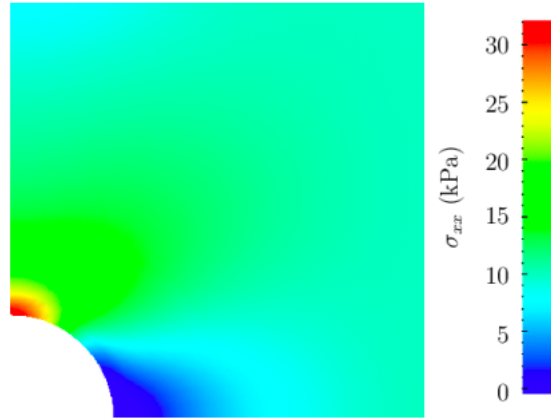
Figure 2.8: Stress field on plate
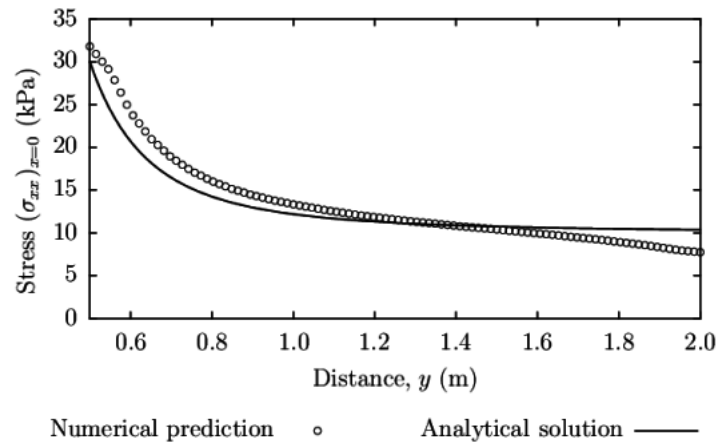
Figure 2.9: Analytical Solution compared to numerical solution

9

Chapter 3

Background Theory

This chapter describes the background theory used to describe the performance of the propeller. *MORE STUFF

3.1   Blade Element Momentum Theory

Blade Element Momentum Theory (BEMT) is a combination of two theories: axial momentum theory and blade element theory. BEMT takes in various geometric parameters such as pitch, chord and camber to produce a body force model of thrust and torque. BEMT also uses various empirical correction factors to correct for tip losses, curvature and wake losses. This section will discuss the theory and solution procedure of BEMT.

3.1.1   Momentum Theory

Momentum theory is based on the a streamtube with an actuator disc. The actuator disc is an infinitesimally thin disc with an increase in axial velocity occurring at the disc. The axial velocity at the disc is therefore:

$$V_1 = V(1 + a) \tag{3.1}$$

Where a is the axial inflow factor and V is the inflow velocity. The axial velocity far downstream is $V_2 = V(1 + 2a)$. Similar results can be shown for the
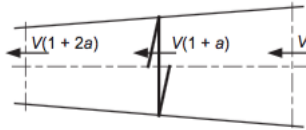
Figure 3.1: Momentum theory

angular velocity where the angular velocity relative to the blades at the disc is $\Omega(1-a')$ where Ω is the angular velocity of the disc and a' is the circumferential inflow factor.

The actuator disc as radius r and thickness δr. The thrust on the disc is therefore:

$$\delta T = 2\pi r \delta r \rho V (1+a)(V_2 - V) \tag{3.2}$$

This is the rate of momentum change. Similarly the torque is given by:

$$\delta Q = 2\pi r \delta r \rho V (1+a) r^2 2a' \Omega \tag{3.3}$$

Hence the thrust per unit span is given by:

$$\frac{dT}{dr} = 4\pi r V^2 a(1+a) \tag{3.4}$$

With Torque given as:

$$\frac{dQ}{dr} = 4\pi r^3 V \Omega a'(1+a) \tag{3.5}$$

A real propeller is not a disc but a finite number of blades.There must be a correction for this as the flow conditions are not circumferentially uniform (TURNOCK). The Goldstein factor, K (GOLDSTIEN) is introduced into equations 3.4 and 3.5. The thrust and torque per unit span are now defined in equations 3.6 and 3.7:

$$\frac{dT}{dr} = 4\pi r V^2 K a(1+a) \tag{3.6}$$

$$\frac{dQ}{dr} = 4\pi r^3 V \Omega K a'(1+a) \tag{3.7}$$

where K is the Goldstein factor. Charts have been created for propellers with 2-7 blades. However a functional relationship for the correction factor is given by:

$$K = \frac{2}{\pi} \cos^{-1}\left(\frac{\cosh(xF)}{\cosh(F)}\right) \tag{3.8}$$

where $F = \frac{Z}{2x\tan\phi} - \frac{1}{2}$ Z is the number of blades and ϕ is the hydrodynamic pitch angle.
The blade element equations are based on figure 3.2

Where C_l is the lift coefficient and C_d is the drag coefficient which are both dependent on angle of attack α. For a two dimensional blade section the lift and drag forces are defined by:

$$\frac{dL}{dr} = \frac{1}{2}\rho Z c U^2 C_l \tag{3.9}$$

$$\frac{dD}{dr} = \frac{1}{2}\rho Z c U^2 C_d \tag{3.10}$$

11

Figure 3.2: Blade element diagram

The lift and drag can then be resolved in terms of the hydrodynamic pitch angle to give the section thrust and torque. These can then combined with the equations 3.9 and 3.10 to give the thrust and torque coefficients:

$$\frac{dK_T}{dx} = \frac{\pi^2}{4}(\frac{Z_c}{D})C_L x^2 (1 - a')^2 \sec\phi(1 - \tan\phi\tan\gamma) \qquad (3.11)$$

$$\frac{dK_Q}{dx} = \frac{\pi^2}{8}(\frac{Z_c}{D})C_L x^3 (1 - a')^2 \sec\phi(\tan\phi + \tan\gamma) \qquad (3.12)$$

where $\tan\gamma = \frac{C_D}{C_L}$. Local efficiency η can be defined by $\eta = \frac{\tan\psi}{\tan(\phi+\gamma)}$. Which can then be used to compute the inflow factors a and a' through the ideal efficiency $\eta_i = \frac{\tan\psi}{\tan\phi}$. The required lift and drag coefficient can then be computed using equations 3.11 and 3.12, with the torque coefficient calculated from equation 3.6.

The flow diagram for the solution procedure is shown in 3.3. In this procedure the advance ratio, pitch/diameter, position along blade and drag coefficient are given. The hydrodynamic pitch angle ϕ plus angle of attack α are computed. A first guess of α is then made to calculate ϕ which is used to calculate the ideal efficiency. The axial inflow factor a is then computed along with the Goldstein correction factor. In the first case the efficiency η is assumed to be equal to the ideal efficiency. The torque coefficient is then calculated as is the lift and drag coefficients which are used to calculate the local efficiency. This value of local efficiency is then used to compute the new axial inflow factor. The procedure is repeated until the local efficiency converges to a value.

Once the local efficiency value has converged the required lift coefficient is computed along with the curvature correction. The angle of attack is then calculated from the lift curve slope $\frac{dC_l}{d\alpha}$ and the required lift coefficient. The entire process is repeated until the angle of attack converges. Once α converges the process is repeated at the next position on the blade. This procedure is used in the code described in CHAPTER BLAH!

As the flow moves along the blade it experiences a change in angular inflow factor. This is taken into account by the addition of curvature correction. The curvature correction is calculated using the Ludweig-Ginzel corrections (REFERENCES). The curvature correction works by firstly approximating $\frac{dC_L}{d(m/c)}$.

Given: $J, P/D, x, C_D$

for $x =$
$$(\phi + \alpha) = \tan^{-1}\left((P/D)/\pi x\right)$$
$$\tan \psi = J/\pi x$$

First estimate of α

$$\phi = (\phi + \alpha) - \alpha$$
$$\eta_i = \tan \psi / \tan \phi$$

First cycle: assume $\eta = \eta_i$
i.e. $C_D = \gamma = 0$
$$a = (1 - \eta_i)/[\eta_i + \tan^2 \psi/\eta]$$

Goldstein factor
$$x \tan \phi$$
$$\kappa =$$

Momentum equation
$$\frac{dK_T}{dx} = \pi J^2 x \kappa a (1 + a)$$

Blade element equation
($\gamma = 0$ first cycle)
$$C_L = \frac{dK_T}{dx} \bigg/ \frac{\pi^2}{4} \frac{Nc}{D} x^2 (1 - a')^2 \sec \phi (1 - \tan \phi \tan \gamma)$$

C_D database for section
type and t/c
$$C_D =$$
$$\gamma = \tan^{-1} C_D/C_L$$
$$\eta = \tan \psi / \tan (\phi + \gamma)$$

Repeat until
new η = old η

Input section data: $C_L, m/c,$
curvature correction and required α'

α'

New estimate of α
Repeat until $\alpha' = \alpha$

$$\frac{dK_T}{dx}, \frac{dK_Q}{dx}, \eta$$

Repeat for new x

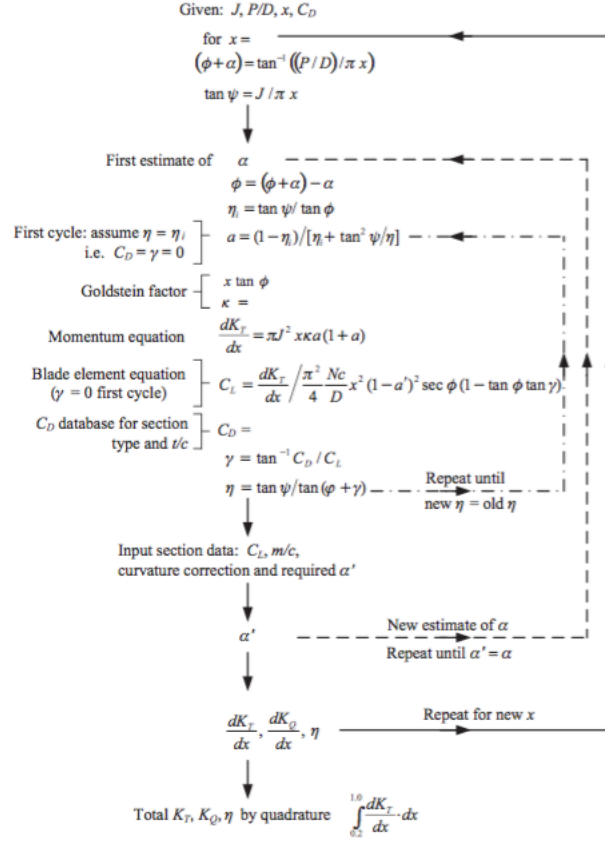Total K_T, K_Q, η by quadrature $\int_{0.2}^{1.0} \frac{dK_T}{dx} \cdot dx$

Figure 3.3: Blade element momentum theory solution procedure flow diagram (TURNOCK)

The camber required to produce the same C_L with 2D flow $\frac{m_0}{c}$ is given by $\frac{m_0}{c} = \frac{C_L}{\frac{dC_L}{d(m/c)}}$. Therefore:

$$\frac{m}{c} = \frac{m_0}{c} k_1 k_2 \tag{3.13}$$

Where k_1 and k_2 are given by the curves shown in figure 3.4. From this data and the angle of attack a change of lift coefficient can be computed which is used to compute the change camber.
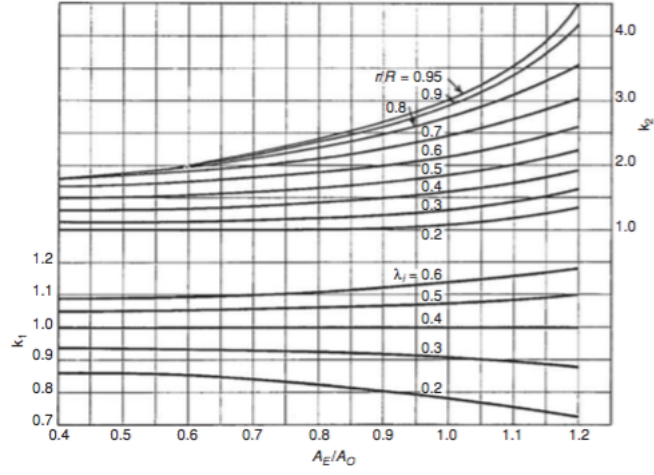
Figure 3.4: Ludweig-Ginzel camber correction coefficients (REF)

Figure 3.5: Forces on beam

3.2 Timeshenko Beam Theory and Finite Element Analysis

This section will discuss the theory behind Timoshenko beam theory and finite element analysis.

3.2.1 Timoshenko Beam Theory

Timoshenko beam theory was developed by Stephan Timoshenko in the early 20th century(CITATION). It is used as a model of structural deformation and accounts for shear deformation and rotational effects (BROWN CITE). Because of this it can be used for short thick beams which is useful when studying marine propellers.

Figure 3.5 shows the forces acting on a beam segment. Here V is the shear force, M is the bending moment acting on the beam, w is the load and dx is the
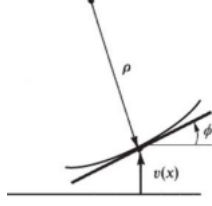
Figure 3.6: Curvature on beam at position x

section length. If the equilibrium state is now considered it can be shown:

$$\sum M_z = 0 = (M + dM) - M - V dx + W dx \frac{dx}{2} \tag{3.14}$$

$$\sum F_y = 0 = V - (V + dV) - W dx \tag{3.15}$$

It can therefore be shown:

$$w = -\frac{dV}{dx} \tag{3.16}$$

and

$$V = \frac{dM}{dx} \tag{3.17}$$

When under a load the beam bends with a radius of deflected curve ρ. The curvature is given by $\kappa = \frac{1}{\rho} = \frac{M}{EI}$ where E is the modulus of elasticity and I is the moment of inertia. The beam curves by angle ϕ with translational movement v at position x however when the angles are small $\phi = \frac{dV}{dx}$. From figure 3.6 it can be seen that $\kappa = \frac{d^2 v}{dx^2}$. Hence equating κ gives:

$$M = EI \frac{d^2 v}{dx^2} \tag{3.18}$$

Substituting equation 3.17 into equation 3.16 and then substituting equation 3.18 into equation the resulting equation gives:

$$w = -EI \frac{d^4 v}{dx^4} \tag{3.19}$$

The shears can now be related to the translational displacement by:

$$V = EI \frac{d^3 v}{dx^3} \tag{3.20}$$

The stress σ and strain ϵ can now be introduced. The displacement in the x-direction is given by $u = -y\theta$

The strain components are therefore:

$$\epsilon_{xx} = -y\theta' \qquad \epsilon_{yy} = 0 \qquad \gamma_{xy} = -\theta + v' \tag{3.21}$$

15

and the stress is:

$$\sigma_{xx} = -Ey\theta' \qquad \sigma_{yy} = -E\frac{\nu}{1-\nu^2}y\theta \qquad \sigma_{xy} = \kappa G\gamma_{xy} \qquad (3.22)$$

The potential energy of the beam is the stored elastic energy plus the energy applied to the beam:

$$\pi = \frac{1}{2}\int \sigma_{xx}\epsilon_{xx}dV + \frac{1}{2}\int \sigma_{xy}\epsilon_{xy}dV - \int wvdx \qquad (3.23)$$

When substituting values for strain and stress into equation 3.23 the potential energy becomes:

$$\pi = \frac{1}{2}EI\int_0^L \theta'^2 dx + \frac{1}{2}\kappa GA\int_0^L (-\theta + v')^2 dx - \int_0^L wvdx \qquad (3.24)$$

and the change in potential energy is:

$$\delta\pi = EI\int_0^L \frac{d\theta}{dx}\frac{d\delta\theta}{dx}dx + \kappa GA\int_0^L (-\theta + \frac{dv}{dx})(\delta\theta - \frac{d\delta v}{dx})dx - \int_0^L w\delta vdx = 0$$
$$(3.25)$$

We now have a system of equations of the form:

$$[K]\vec{u} = \vec{F} \qquad (3.26)$$

Where K is the stiffness matrix, $\vec{u}$ is the displacement vector and $\vec{F}$ is the force vector. There is now a system of equations which describe the beam deformation for a given load. However not all terms in equation 3.25 are known. Approximations must be made for the interpolation of bending v and twist θ. This is done using shape functions. This process is described in section 3.2.2.

3.2.2 Finite Element Analysis

Finite element analysis is used to obtain an approximation to properties of structures. The domain is split into many intervals or elements. The governing equation of the property is known however there are terms in the equation which are unknown. These terms are represented using shape functions such that:

$$v = \sum_a N^a v^a \qquad \theta = \sum_a N^a \theta^a$$
$$\delta v = \sum_b N^b \delta v^a \qquad \delta\theta = \sum_b N^b \delta\theta^b \qquad (3.27)$$

The shape functions are chosen such that $N^a = 1$ at the associated node and $N^a = 0$ at all other nodes. For simplicity the shape functions have been chosen to be linear. For this case the shape functions are:

$$N^a = 0.5(1 - x_i) \qquad N^b = 0.5(1 + x_i) \qquad (3.28)$$

Where x_i is the local coordinate. From convention x_i is in the interval $[-1, 1]$. Equations 3.27 can now be substituted into equation 3.25 to give:

$$\left\{ EI \int_0^L \frac{dN^a}{dx} \frac{dN^b}{dx} dx \right\} w^a + \left\{ -\kappa GA \int_0^L N^a \frac{dN^b}{dx} dx \right\} \theta^a = \int_0^L w N^b dx$$

$$(3.29)$$

This generates the local stiffness matrix and force vector. The local values must then be mapped on to the global stiffness matrix and global force vector. This is relatively simple for a 1D beam.

Equation 3.29 is now in a form which is solvable via $\vec{u} = \vec{F}[K]^{-1}$. This is done by the code described in section (CODE SECTION HERE)

Chapter 4

The Code

This chapter will describe the code used for the blade element momentum theory and finite element analysis. Furthermore the chapter will describe how the BEMT and FEA are coupled. Full code can be found in appendix (INSERT APPENDIX REF)

4.1 Blade Element momentum theory code

The blade element momentum theory code uses the same algorithm as described in figure 3.3. However there are portions which require user input. This section will describe the code and the required user input.

The 1st part of the user needs to interact is the initializing data

```
1   if __name__ == '__main__':
2           Re = 500000 # Reynolds Number
3           foil_filename = 'naca2412.dat'
4           prop_data_file = 'propgeom.txt'
5           P_D = 1.2        # Pitch/Diamter Ratio
6           BA_ratio = 0.95 #  Blade Area Ratio
7           N_blades = 3      # number of blades
8           xfoil_flag = 0 # use xfoil for Cl and Cd approximations 1 yes, 0 no
9           rho = 1029 # density of seawater
10          n = 10 # rotations per sec
11          Va = 10 # prop inflow velocity
12          chord, x_R_list, local_P_D, chord_diameter, thickness_distibution ,MC, file
13           ,D = generate_foil_data(prop_data_file)
14          J = Va/(n*D)      #Advance Ratio
15          E = 5e6 # Elastic Modulus
16          nu = 0.3 # Poissons Ratio
17          kappa = 5/6
18          h = 2
```

```
Prop: KP458 for KVLCC and KVLCC2
Number of Blade: 4
Section Profile: KH40( Modified NACA66 + a=0.8 mean line Camber)
     (note) Details of KH40 section can not be disclosed.
          However, it can be replaced by NACA66 section for CFD purpose.
Scale ratio: 58.0
Propeller Diameter: 1.25 m
Hub Ratio: 0.2

  r/R      P/D       Rake     Skew(°£)     C/D       fo/C        to/D
0.200   0.843600   0.0000     -4.000    0.166400   0.000000    0.036585
0.300   0.957500   0.0000     -4.400    0.192800   0.000000    0.032378
0.400   1.002250   0.0000     -4.400    0.210400   0.000000    0.028170
0.500   1.007000   0.0000     -3.150    0.220800   0.000000    0.023963
0.600   0.987050   0.0000     -0.820    0.223200   0.000000    0.019755
0.700   0.950000   0.0000      2.490    0.215200   0.000000    0.015545
0.800   0.900600   0.0000      6.350    0.192800   0.000000    0.011340
0.900   0.843600   0.0000     10.760    0.147200   0.000000    0.007135
```

Figure 4.1: Example of a propeller data file

Here the basic conditions are defined such as Reynolds number, Elastic modulus and Poissons ratio. Furthermore on line 3 the foil filename is defined. This is the data file defining the foil coordinates. This file is used only if Xfoil is being used to generate the foils lift and drag curves. The option to use Xfoil is specified in line 8, if 1 is selected the Xfoil will be used, if 0 is selected then the lift and drag curves will be defined by a data from a lookup table. The propeller data file is defined in line 4. An example of this data file is shown in figure 4.1. This file gives information on the number of blades, section profile,and propeller diameter. The radial positions to be analysed are given as well as: the pitch, chord, camber and thickness at these positions. All this data is stored and used in blade element momentum theory.

Once the blade parameters have been defined the blade element momentum theory is then called.

Firstly the angle of attack is set to 0 for the 1st blade element.

```python
1  alpha[i] = 0.0
2  # set number of alpha iterations to 0
3  alpha_iterations = 0
4  # set convergence flag to 0
5  alpha_converge = 0
6  # start of alpha convergenc loop
7  while alpha_iterations < 2000 and alpha_converge == 0:
8          alpha_iterations += 1
9          # calculate tan of inflow angle
10         tan_psi[i] = J / (np.pi*x_R)
11         # induced flow angle plus angle of attack
12         phi_plus_alhpa[i] = np.arctan2(local_P_D[i],(np.pi*x_R))
13         # inflow angle
14         phi[i] = phi_plus_alhpa[i] - alpha[i]
15         # ideal efficieny
```

```
16            eta_ideal = tan_psi[i] / np.tan(phi[i])
17            # initially set efficiency to ideal
18            eta   = 0.9 * eta_ideal
19            gamma = 0.0 # zero for ideal efficiency
```

The alpha loop then begins. Firstly the inflow angle is calculated from the advance ratio. Then the induced flow angle plus angle of attack is computed from the local pitch (line 12). The angle of attack is then subtracted from this to compute the induced flow angle. The ideal efficiency is then calculated from the inflow angle and induced flow angle and the actual efficiency is set to 90% of the ideal efficiency.

The efficiency loop now begins.

```
1  while eta_iterations < 2000 and eta_converge == 0:
2            # axial inflow factor
3            a = (1 - eta_ideal)/(eta_ideal + (tan_psi[i]**2)/eta)
4            # local thrust coefficient
5            KT_dx[i] = np.pi*(J**2) * x_R * K * a*(1+a)
6            # circumferential inflow factor (a')
7            a_p =1-eta_ideal*(1+a)
```

Firstly the axial and circumferential inflow factors are computed as shown in lines 3 and 7. The thrust coefficient is also computed. The lift and drag coefficients are then computed. This is done using a table either produced by Xfoil or using the look up table. The table has values for lift and drag coefficient for many angles of attack. If the angle of attack of the blade section does not lie on a defined value simple linear interpolation is used. A new γ angle is calculated from the lift and drag coefficients. From the new γ the efficiency is computed. This efficiency loop is then repeated until $\eta_{old} = \eta_{new}$ within a small tolerance. Once the efficiency has converged the curvature correction is applied. This involves generating correct coefficients k_1 and k_2 using the graph shown on figure 3.4. The rate of change of lift coefficient to camber is assumed to be 12 (TURNOCK). The camber deficit is then computed by subtracting the required camber at $\alpha = 0$ from the actual camber. This is then multiplied by the assumed lift coefficient camber slope and divided by the lift curve slope to produce the new angle of attack. This process is repeated until the angle of attack converges on a single value. When the angle of attack converges the torque coefficient is calculated as is the velocity of the fluid into the blade element. The entire process is repeated for the next blade element.

Once the process is complete over all blade elements the total thrust and torque coefficients are computed by integrating the thrust and torque over all elements using Simpsons rule. The results of this process are shown in chapter 5.

4.2   Finite Element Analysis Code

The model of the blade for structural purposes is that of a simple 1D timoshenko beam. This makes the computational cost of computing the bending and twist

cheap whilst also having good accuracy.

The code starts by creating a grid of nodes evenly spaced along the beam. The number of nodes is defined by the number of elements. A location matrix is then set up to define the global node numbers to define each element. In other words, the nodes defining each element 1 will be stored in column 1 in the location matrix. An example of the location matrix is shown in matrix 4.1

$$LM = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix} \tag{4.1}$$

The integration points on each element are then defined. The integration will be an Gaussian quadrature. Therefore, for ease of calculation the weighting will be set to 1 hence the integration points chosen are $\pm\sqrt{\frac{1}{3}}$. For more on Gaussian quadrature see Appendix B. So for each element and each integration point of each element the value of the shape functions described in equations 3.28 are computed. The shape function derivatives are with respect to local coordinates so therefore must be converted to global coordinates this is done using the chain rule. The stiffness matrix contributions due to bending are computed 1st with the global stiffness matrix and global force vector being populated appropriately as consistent with equation 3.29. Note that np.meshgrid is being used here. This is to ensure that the correct elements of the stiffness matrix are being populated. np.meshgrid creates matrix from the two arrays it is given. This matrix allows a every combination of element pairs in the row and columns. It is therefore used in the stiffness matrix population to populate every element in the matrix which corresponds to the bending and twist of the finite element.

The shear contribution is then added to the global stiffness matrix. This process is repeated for every element over the entire beam. The boundary conditions are then enforced. For this case the propeller blade is treated as a cantilever beam. This means that the bending and rotation at base are strictly zero. This boundary condition is set and the system of equations $\vec{u} = \vec{F}[K]^{-1}$ is solved to give the bending and twist along the beam.

4.3 Interface between blade element momentum theory and finite element analysis

Chapter 5

Results and Discussion