UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Engineering

A Computationally Efficient Method of Determining Vibration Characteristics of Marine Propellers in Drift Conditions

by

Nicholas James McCaw

A thesis for the degree of
Engineering Doctorate

May 2021

University of Southampton

Faculty of Engineering and Physical Sciences
School of Engineering

Engineering Doctorate

A Computationally Efficient Method of Determining Vibration Characteristics of Marine Propellers in Drift Conditions

by Nicholas James McCaw

Abstract

The unsteady, non-uniform inflow to marine propellers causes a time dependent load and response of the propeller blades. Although techniques exist to model the fluid structure interaction of marine propellers operating behind a hull wake, these are often computationally expensive. The ability to predict the response of the propeller blades at the design phase is desirable as it will enable a propeller designer to obtain a more optimal design with a smaller time investment. This Thesis aims to develop a computationally efficient and validated numeric tool for computing the vibration characteristics of marine propeller designs. This can be achieved by use of a numerically efficient hydrodynamic model and coupling it to a reduced degree of freedom structural model.

The hydrodynamic model used to obtain the performance of the propeller and load distributions along and across the blades is Blade Element Momentum Theory. This includes a database of 2D foil CFD simulations to calculate pressure on each blade section. This hydrodynamic model is validated using high fidelity CFD simulations and found to agree well.

To obtain an unsteady, non-uniform inflow to the propeller CFD has been utilized to obtain the flow field at the propeller plane of the KVLCC2 hull form. This has been validated against experimental data available. To obtain the structural characteristics of the propeller blades a plate model has been implemented. This is shown to give reasonably good accuracy compared to a full 3-D model but at an order of magnitude computationally cheaper.

An algorithm has been developed to couple the hydrodynamic and structural models. This is compared to a high-fidelity CFD-FEA coupled simulation. The computationally efficient model compares reasonably well to the high fidelity model. However the Plate-BEMT model achieves the deflection in a fraction of the time of the high fidelity model. The method developed can assist the propeller designer generate geometries which have optimal vibration properties for the given hull form and can perform well in manoeuvring configurations.

Contents

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

Signed:..... Date:.....

Acknowledgements

I have many people to thank for the completion of this Thesis. Firstly, I would like to thank Dr. Ian Hawke and the NGCM directors for their excellent support during the 1st year of the EngD.

Thanks goes to Artur Lidke for all his excellent support for many years and the FSI group for putting up with me. Thank you also to everyone who supported me at QinetiQ, especially all of my proof readers.

Big thank you must go to Professor Stephen Turnock for all his wise words and support during the difficult times.

I couldn't of submitted this without the support of Lorraine and Andrew McCaw. Thanks for all the calls and the much needed relaxing trips to Turnberry!

Finally, I must reserve my biggest thanks to wee Emily Davies. Thanks for being with me and making me laugh through all the hard times.

List of Figures

List of Tables

Listings

Nomenclature

α	Angle of Attack
α_d	Alpha Damping Coefficeint
$\bar{u}$	Time Averaged Velocity
β	Ship Drift Angle
β_d	Beta Damping Coefficeint
δ_{ij}	Kroneker Delta
$\dot{}$	Time Derivative
ϵ	Turbulent Dissipation Rate
η	Propeller Efficiency
γ	Shear Strain
ν	Kinematic Velocity
ν	Poissons Ratio
Ω	Propeller angular velocity
ω_n	Natural Frequency
ϕ	Induced Flow Angle
ϕ(s)	Wagner Function
ρ	Fluid Density
τ	Turbulent Time Scale
τ_w	Wall Shear Stress
$\underline{\epsilon}$	Strain Vector
$\underline{\sigma}$	Stress Vector

$\underline{d_e}$ Displacement Vector

$\underline{F}$ Force Vector

ζ Viscous Damping Factor

a Axial momentum factor

a' Tangential momentum factor

a_{ij} Orthogonal Transformation Tensor

B_{wl} Beam at Water Level

BAR Blade Area Ratio

c Chord

c Damping Coefficient

C_d Drag Coefficient

C_F Friction Coefficeint

C_l Lift Coefficient

C_m Moment Coefficeint

C_p Pressure Coefficient

D Propeller Diameter

E Youngs Modulus

$epsilon_{ijk}$ Permutation Tensor

F_R Froude Number

g Acceleration Due to Gravity

J Advance Ratio

K Goldstien Correction Factor

k Stiffness Coefficient

k Turbulent Kinetic Energy

K_Q Torque Coefficient

K_T Thrust Coefficeint

l Turbulent Length Scale

L_{pp} Length Between Perpendiculars

L_{wl}	Length at Water Level
m	Blade Section Maximum Camber
N	Shape Function
n	Propeller Rotation Rate (rps)
P	Pressure
P/D	Pitch Diameter Ratio
Q	Q-Criterion
Q	Torque
r	Radial position
Re	Reynolds Number
T	Thrust
t	Blade Section Maximum Thickness
T_e	Kinetic Energy
u^*	Non-Dimensional Friction Velocity
U_e	Potential Energy
V_a	Local axial Velocity
V_s	Ship Velocity
w_T	Wake Fraction
y^+	Wall Units
Z	Number of Blades
$[C]$	Damping Matrix
$[K]$	Stiffness Matrix
$[M]$	Mass Matrix

Chapter 1

Introduction

1.1 Motivation

The requirement to reduce marine vehicle noise has been relevant for military applications for years Carlton (2012), however there is a growing concern to reduce the environmental impact of merchant shipping. Maritime noise can take the form of vibration of the drive train however, the main sources of noise and vibration originate from the propeller Merz et al. (2009). The vibration can be caused by flow features exciting the vibration modes of the marine propeller or by cavitation, where low pressures on the blade surface causes the water to boil and bubble. Although cavitation has a more prominent influence on noise, the modelling of this phenomena will not be performed in this thesis.

To design a propeller with low vibration properties, traditionally, a geometry would be developed to fulfil the design parameters. The operating conditions, noise requirements, thrust and torque requirements and the inflow conditions, characterised by the geometry of the hull form and appendages, will be considered. Using previous expertise and propeller geometries a designer can obtain a distribution of chord, pitch, thickness, camber, skew and rake. A prototype can then be created and tested in a physical experiment. This is an expensive procedure and will not yield effective results in an efficient manner. This is particularly prevalent in the design stage as changes to the hull form can occur in a short time frame which will require changes in the propeller design rapidly.

If a propeller geometry designed for a previous case is used then this will provide sub optimal performance for the design parameters. To decrease the cost in generating a geometry with desired characteristics computational techniques have been developed to analyse the performance of the propeller.

Propeller performance characteristics can be obtained using: experimental techniques, high fidelity simulations or low fidelity modelling. High fidelity simulations capture small details of the system which make up the output whereas modelling generates an output using simplified physics.

High fidelity solutions such as Computational Fluid Dynamics (CFD) can be used to obtain fluid loading on the propeller blades. To obtain accurate results using CFD a good quality mesh must be created. The mesh consists of a domain which encompasses the flow, the domain is then split into small grid cells therefore allowing the small details of the domain to be captured. To simulate a propeller design using CFD requires a significant computational cost where several hours on a high performance computing module are required to obtain characteristics of a single design.

A 3D Finite Element Analysis model can be used to obtain the structural properties of the propeller. The structural model will also require a mesh on the geometry itself as opposed to the surrounding domain. These simulations can be coupled to obtain a Fluid-Structure interaction model. This is where the fluid causes a force to act on the model which causes the structure to deform, this deformation causes the geometry to change therefore the fluid loading will change. These high fidelity simulations become especially computationally demanding when mesh motion is applied which is the case for fluid-structure interaction simulations.

The use of these high fidelity simulation techniques has grown due to the increase in computational power available in the form of High Performance Computing (HPC) Hawkes (2017) where many compute nodes are used in parallel to increase turn around time for large scale simulations. The use of HPC requires access to a cluster of compute nodes which can have a significant cost to maintain and operate. Although the cost for high fidelity simulation is reducing and HPC becoming more readily available this still requires a substantial overhead and expertise to use obtain accurate results.

Due to the high computational cost of fluid structure interaction simulations, it is impractical to use CFD-FEA coupling for the early design stages of a propeller. It is therefore desirable to obtain a fluid structure interaction model of an arbitrary propeller geometry using accurate models.

These approximations have been utilised in the wind turbine and tidal turbine environments Otero and Ponta (2010) , Otero and Ponta (2015) , Harrison and Batten (2010). These models uses a body-force model to generate the hydrodynamic forces and a 1 dimensional FEA model to model the structures. Although body-force models have been used to model the hydrodynamics of marine propellers Phillips (2009), Badoe (2015) there is little literature in coupling a body-force model to an FEA model

for marine propellers. This technique will capture the FSI properties of the marine propeller rapidly but will miss some details the high fidelity simulations can achieve.

1.2 Background

The design of propellers is often a balance of compromise of characteristics. A more efficient propeller can be generated by using a shorter chord distribution, this will cause cavitation inception, where the pressure falls and water turns to vapour, to occur more readily. Furthermore, a propeller blade with higher loading towards the tip will produce greater efficiency but this causes stronger tip vortex structures which increases noise. Additionally, to obtain weight saving a thin blade can be designed however this will cause structural fatigue issues. Therefore, to obtain an optimal solution the different demands must be considered and prioritised.

Noise can originate from many aspects of the propeller. The sources of noise are shown in Figure 1.1. This demonstrates that at low frequencies the inflow turbulence and displacement of the propeller blade dominates. The excitation of the displacement frequencies is known as propeller 'swathing'. The trailing edge noise occurring at high frequencies are due to vortices shedding from the trailing edge of the propeller blade. The frequency of the vortex shedding can 'lock-on' to the natural frequency of the blade causing a phenomena called 'singing' Ausoni (2009) Zobeiri (2012).

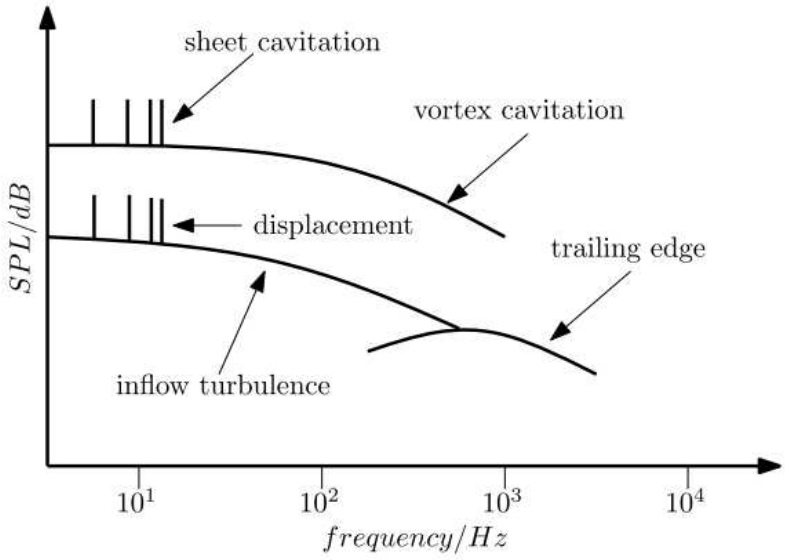


FIGURE 1.1: Marine Propeller Noise Sources, adapted from Lloyd (2013)

The inflow turbulence arises from the boundary layer of the ship which will be fully turbulent at the propeller plane. This causes high frequency changes in velocity which

cause changes rapid changes in fluid loading on the blade which will change the frequency of blade deformation.

Due to different appendages of the hull and operating conditions the wake inflow to the propeller plane and the flow conditions to the propeller can vary. The time dependent change in inflow conditions can excite vibration modes of the propeller blades. Changing the material or geometry of the propeller blade will cause the vibration modes to change. The blade material properties and geometry can then be tuned to minimize the vibration of the blades given the inflow they experience.

The use of high fidelity simulation techniques such as CFD can be used to obtain the performance and noise characteristics of the propeller design however, these are computationally expensive and take significant time to complete. The use of CFD-FEA coupling is particularly complex and expensive when the propeller is operating behind a hull. This requires a large time commitment both in human cost of setting up a complex computation and when running the simulation. This is impractical for use at the design stage of the propeller due to commercial and industrial demands on time. Hence, it is desirable to obtain the vibration properties of the propeller blades using an computationally efficient tool which retains a high level of accuracy. In this context, the efficiency of the methods are described by the ease of use for the propeller designer and the time taken for useful data to be generated in the computational environment.

Increased use of machine learning and optimization for marine propellers He et al. (2012) require accurate, low computational cost modelling. A parameter used in the optimization computation could be to reduce the blade deformation. Therefore a computational tool to efficiently calculate the deformation of an arbitrary propeller blade design would be used for these purposes.

Therefore, the development of a numeric tool to determine the vibration properties of a propeller blade with a non-uniform and transient velocity inlet is required. This tool is to be verified with respect to higher fidelity models whilst keeping the computational cost low.

1.3 Aim and objectives

The aim of the research is to develop a computationally efficient and validated numeric tool for computing the vibration characteristics of marine propeller designs due to ship manoeuvring.

This is achieved by fulfilling the following objectives:

1. Develop an accurate and time efficient hydrodynamic model to obtain the load distribution caused by the pressure on the propeller blades. This will be achieved for an unsteady and non-uniform inflow. The model should be computationally efficient enough to be run thousands of times without use of a HPC and the model must distinguish the blade position relative to the hull.
2. Using an easily modifiable description of propeller rake, skew, pitch, camber, chord, thickness and diameter be able to mesh an arbitrary blade geometry for use in FEA.
3. Develop and verify an FEA model for the structural properties of the propeller blade and be able to obtain the dynamic response from an external load. This model, must be efficient in using computational resources and time and accurately represent the propeller blade.
4. Produce unsteady, non-uniform flow at the propeller plane of a ship. This will be done using high fidelity CFD simulations of the KVLCC2 hull form and include efficient modelling of the turbulent flow features.
5. Establish techniques to accurately describe the physical properties of propeller blades and performance characteristics of marine propellers using high fidelity modelling techniques.
6. Verify the computationally efficient FEA model with high fidelity commercial FEA codes and compare the hydrodynamic model with high fidelity propeller CFD simulations to ensure the hydrodynamic model captures the load distribution accurately.

1.4 Novelty and contributions

Numerous studies of fluid structure interactions have taken place over the past 40 years. In recent years the use of high fidelity CFD-FEA coupling has been utilized due to their accurate results however these simulations require a large HPC cluster and a significant time commitment.

The use of low fidelity computational tools for marine propeller fluid structure interaction mainly consist of potential flow boundary element methods. These methods, however, do not account for Reynolds number effects.

Therefore the contribution this thesis provides is the use of a Blade Element Momentum theory code which includes: Reynolds number effect, hysteresis and accounts for the turbulence at the propeller plane of a ship, coupled with a reduced FEA model. The applicability of the code is discussed and compared with a high fidelity model.

Finally, due to the extremely high computational cost of Large Eddy Simulations coupled with FEA, little work has been done in analysing the blade deflection due to the turbulent fluctuations in the inflow to the propeller.

1.4.1 Papers

1. McCaw N, Turnock S.R, Batten,W.M.J The coupling of blade element momentum theory and a transient Timoshenko beam model to predict propeller blade vibration response *Proceedings of the Sixth International Symposium on Marine Propulsors: smp'19 26 - 30 May 2019, Rome, Italy.*
2. McCaw, N, Turnock, S R. Batten, W. M. J. The use of BEMT and a transient Timoshenko beam to obtain vibration properties of propeller blades under manoeuvring conditions *Numerical Towing Tank Symposium 2019*
3. McCaw, N, Turnock, S R. Batten, W. M. J An investigation of the vortex shedding frequency and natural frequency of a ship propeller *Numerical Towing Tank Symposium 2018*

1.5 NGCM/Industry Sponsor

The work undertaken is an Engineering Doctorate (EngD) with the industry sponsor QinetiQ and the Next Generation Computational Modelling Group (NGCM) at the University of Southampton.

QinetiQ are a large technology company working with the Ministry of Defence and other private companies. They have many sites across the UK and other countries each specialising in different areas of defence technology such as radar, control, marine and air.

The site at Haslar specialises in marine technologies with capabilities in model testing, simulations and propeller design and analysis. QinetiQ have a drive to increase their capability to design propellers and therefore require a design tool to assess the vibration properties of the blades to add to their propeller design suite under development.

The NGCM is a centre of doctoral training at the University of Southampton. The aim of the NGCM is to provide training for post-graduate students to:

- be trained in state-of-the-art best practice for computational modelling
- be exposed to industrial and real-world problems
- become cross-disciplinary communicators

- become cross-disciplinary computational problem solvers

1.6 Thesis Structure

Figure 1.2 outlines the structure of the thesis. All Chapters leading up to Chapter 7 form the necessary building blocks which are then brought together in Chapter 7. For instance Chapter 3 is used as the hydrodynamic model for the FSI model and is compared with high fidelity modelling of the propeller hydrodynamics described in Chapter 4. Also, the data generated in Chapter 5 used as the inflow to the propeller in Chapter 7 and Chapter 6 describes the structural modelling for the propeller.

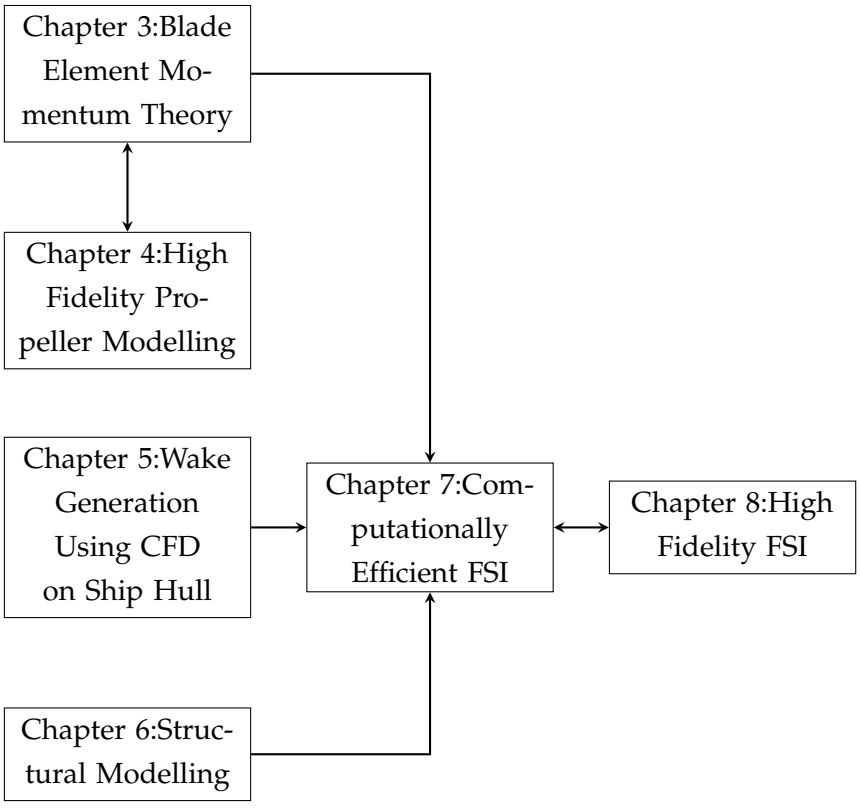


FIGURE 1.2: Flow Chart of Thesis

Chapter 2 outlines the relevant physics and modelling techniques relating to propeller vibration. A discussion of the pertinent literature is also included here, discussing current propeller modelling techniques. The advantages and disadvantages of different propeller performance modelling techniques is presented. In this chapter the principals of vibration and structural modelling are presented with state of the art techniques discussed. The methods for coupling the fluids and structures for simulations are also discussed.

Following Chapter 2, Chapter 3 introduces the propeller fluid loading modelling techniques and how it is improved for advanced computational efficiency and accuracy. Chapter 4 then discusses high fidelity modelling of marine propellers by using computational fluid dynamics to obtain the pressure loading along the propeller blade. The solutions from the high fidelity and computationally efficient methods are compared with advantages of the method from Chapter 3 outlined.

The hydrodynamic model described in Chapter 3 is capable of operating within a non-uniform and unsteady regime. Chapter 5 discusses the use of computational fluid dynamics to obtain the unsteady and non-uniform wakes of the KVLCC2 hull form at several drift angles. The non-uniform inlet at these drift angles will cause the propeller to vibrate. In addition, the turbulence contribution to the flow field is investigated by comparing the flow field of a large eddy simulation with a synthetic turbulence model.

Chapter 6 discusses the development of the structural model used to describe the motion of the propeller blades. This chapter investigates the modelling techniques used for optimal computational performance whilst maintaining good accuracy.

With each aspect of the numeric tool discussed Chapter 7 describes the algorithm used to provide vibration properties of the propeller blade using one-Way and two-Way coupling techniques. These results are then compared to a high fidelity one-way coupling computation using CFD-FEA coupling which is described in Chapter 8.

Chapter 2

Propeller Fluid Structure Interaction

2.1 Introduction

This chapter discusses marine propeller geometry definitions and design considerations. The techniques used to model propeller performance are then discussed, this includes a wide range of techniques with a brief description each method and its associated computational cost.

The structures and vibration properties of marine propellers are then discussed followed by the techniques for modelling Fluid-Structure interaction, not only for marine propeller applications but in the wider context of numeric modelling

Finally, the literature surrounding ship manoeuvring in the context of obtaining validated data for fluid velocity fields at the propeller plane is examined.

2.2 Propeller Design Process

Propellers predominantly operate in a behind condition in the wake of a ship hull. This causes the fluid velocity to be non-uniform at the position of the propeller operation. As the propeller blade rotates they experience a varying load due to the non-uniformity of the inflow which causes time varying blade deformation and hence vibration.

The propeller designer must therefore take the hull wake into consideration when designing a propeller geometry. This section discusses these aspects in more detail.

2.2.1 Geometry and Design considerations

Propellers can have many different configurations with many variables to be altered to change the performance. The design of the propeller is a trade off between all geometric variables. For example a larger propeller diameter will give greater thrust however, structural and weight considerations must be made.

Moreover, the inflow characteristics must be considered to design the propeller to operate in conditions of maximum efficiency. The ratio between the velocity inflow and the rotational speed is established such that the propeller can operate at maximum efficiency.

2.2.1.1 Geometry

Propellers are effectively rotating foils with changing chord, twist, thickness, rake and skew from the blade root to the blade tip. The pitch, skew and rake definitions are shown in Figure 2.1. The pitch is the distance the propeller would move after a full rotation due to the angle of the blade relative to the normal of the propeller hub. The skew is the angle the propeller bends when looking normal to the blade surface. The rake is the angle of the blade from normal to the shaft centreline. These geometry definitions have been standardized by the ITTC committee in 1978 Carlton (2012).

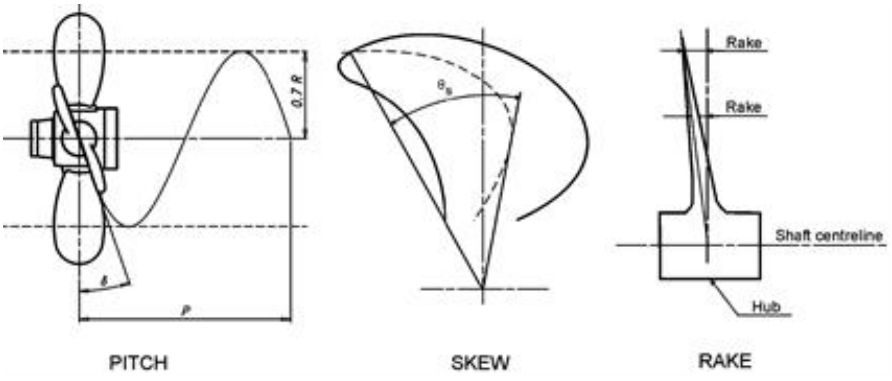


FIGURE 2.1: Propeller geometry definition Carlton (2012)

A propeller works by rotating about its centre with an inflow velocity. The combination of inflow velocity and rotational speed causes each section of the blade to experience an angle of attack. The pressure difference of the foil produces a lift and drag force, which are components of thrust and torque forces. Changing the geometry of the propeller and blade section will alter the local pressure distribution. For example, the blades section can have greater camber which will increase lift and therefore thrust however this promotes the onset of stall.

Changing the geometric properties of the propeller causes different performance characteristics. Allowing a greater skew angle causes the point at which cavitation is onset to change. Increasing pitch allows for greater power output at faster speeds but the propeller will perform poorly at low speeds Newman (2002). Rake has little effect on performance but can alter the pressure distribution on the blade Gorji et al. (2017).

The effective design of a propeller depends on many considerations such as: the operating condition, is the onset of cavitation a priority, is high efficiency a priority, structural considerations and weight. There is often a balance between these design consideration as one often counter acts another. A propeller designer therefore requires effective tools and experience to manage the contradicting characteristics.

2.2.1.2 Inflow Conditions

The inflow conditions in which the propeller is operating is a key aspect of propeller design. A propeller is designed to operate within the wake of a hull so the maximum efficiency occurs at the design condition. The propeller is often located at the rear of the ship and a complex wake develops due to change in flow field due to appendages on the hull, therefore the flow will be non-uniform and unsteady at the propeller.

The velocity profile of the wake at the propeller plane has been studied in detail with Hoekstra (1977) giving the circumferential wake field for a NSMB ship model No. 4984. The axial and tangential wakes are shown in Figure 2.2.

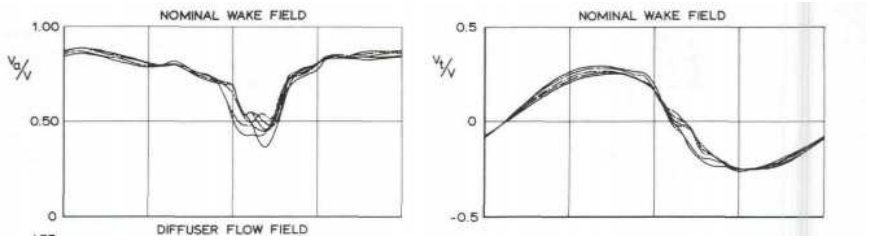


FIGURE 2.2: Wake field of NSMB ship model by Hoekstra (1977)

From Figure 2.2 it can be seen that the axial flow field is almost uniform with a large decrease in velocity at the 180^o position. The tangential wake follows a sinusoidal curve pattern.

It is noted that the flow velocity in the wake is slower than the ship velocity. This is due to the energy lost in the flow because of the presence of the ship. The ratio between the velocity of the fluid in the wake to the ship velocity is called the wake fraction as described in equation 2.1

$$w_T = \frac{(V_s - V_a)}{V_s} \tag{2.1}$$

Where V_s is the velocity of the ship and V_a is the local velocity at the propeller plane. The wake consists of 3 component in the x, y and z directions. To relate this to the propeller these are written as polar velocities as axial, radial and tangential velocity components. Different hulls will form different wakes therefore designing a propeller to account for the wake field is imperative.

2.2.1.3 Advance ratio

To compare propellers operating at different operating conditions a non-dimensional coefficient has been established known as advance ratio. This is the ratio between the axial inflow velocity V_a to the rotational velocity n shown in equation 2.2.

$$J = \frac{V_a}{nD} \tag{2.2}$$

Where V_a is the inflow velocity in m/s, n is the rotational velocity in rotations per second and D is the propeller diameter in meters. The advance ratio is useful as it non-dimensionalises the performance curves of the propeller. So, if V_a and n are changed together in such away that the advance ratio remains constant then the efficiency, thrust and torque coefficients will remain constant. Also, propellers of different size and geometry can easily be compared. A graph of thrust, torque and efficiency against advance ratio can be made for each propeller design as shown in Figure 2.3 Young (2008).

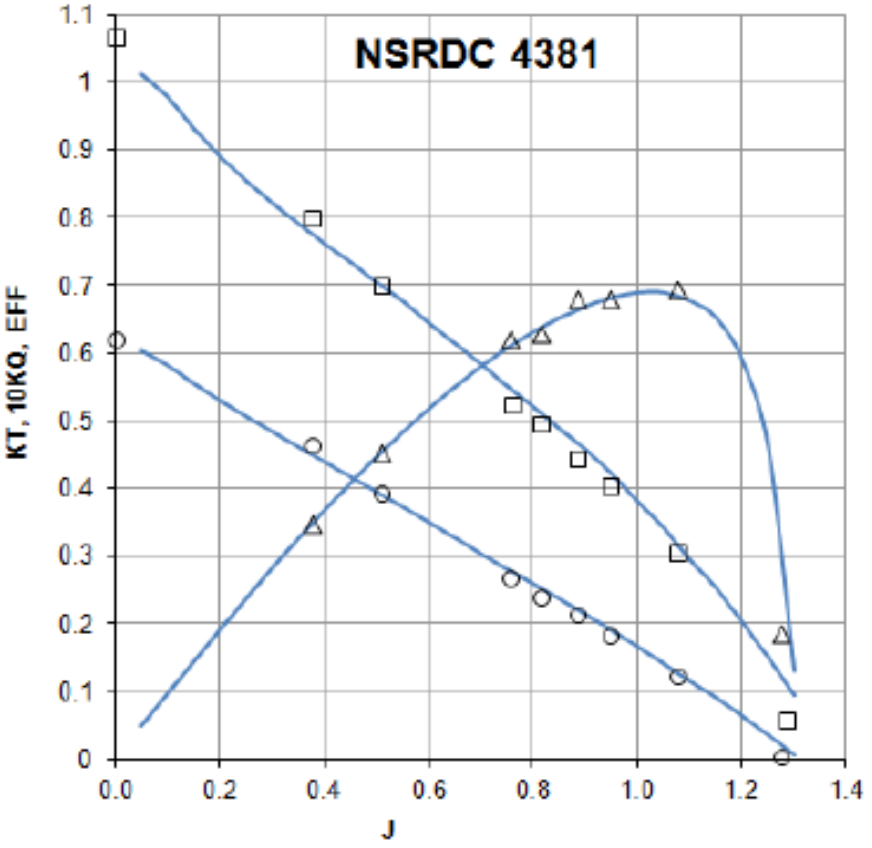


FIGURE 2.3: Example of K_t, K_q,η against advance ratio for the NSRDC 4381 propeller. Note the efficiency rises to a peak value before rapidly droppingYoung (2008) .

As seen from Figure 2.3 as the advance ratio increases the thrust and torque coefficients decrease however the efficiency increases to an optimal value then decreases rapidly. The efficiency is defined as:

$$\eta = \frac{JK_t}{2\pi K_Q} \tag{2.3}$$

Where K_Q is the propeller torque coefficient, K_t is the propeller thrust coefficient. The thrust and torque coefficients are non-dimensional coefficients described in equation 2.4. The propeller is normally operated at the position of maximum efficiency.

$$\begin{aligned} K_T &= \frac{T}{\rho n^2 D^4} \\ K_Q &= \frac{Q}{\rho n^2 D^5}. \end{aligned} \tag{2.4}$$

Where T and Q are the propeller thrust [N] and torque [Nm] respectively and ρ is the fluid density. Thrust and torque coefficients are useful as a comparison between different propeller performance and for scaling between model scale and full scale.

2.2.2 Optimization

In recent years the use of optimization and machine learning have become prominent. The use of optimization can lead to a great reduction in time spent by the propeller designer in trialling and testing designs. However, to effectively use optimization computational tools must be efficient enough to be able to run 100s of design cases in an efficient manner. As the presented work offers a novel method for obtaining the dynamic structural response of a propeller blade, it can be used within an optimization framework in future developments.

Optimization may be defined as a search for an input or set of inputs that minimize the output given the constraints Keane and Nair (2005). So, for propeller vibration reduction the inputs will be the propeller geometry and material with constraints being the operating wake and conditions and the output will be maximum deformation. The goal of the optimizer would be to minimize the unsteady deformation.

Although optimization techniques can be achieved using high fidelity simulations such as in the work by He et al. (2012) it is more practical to obtain an optimized solution using computationally cheaper tools such as the work by Jiang et al. (2018). Without the use of an efficient and accurate method for obtaining vibration properties it is impractical to use optimization techniques in a commercial setting.

2.3 Modelling Propeller Performance

This sections describes techniques to predict marine propeller performance for specific designs. The computational methods range from high fidelity simulations to quick models which are 1000s times less computationally demanding.

2.3.1 Blade Element Momentum Method

Blade element momentum theory is a method to model the performance of a propeller design. This method combines momentum theory, where the propeller is modelled as an infinitely thin annulus with a momentum change, and blade element theory where the propeller blades are modelled as a 2D lifting surface. This method is computationally cheap and has proven to be reasonably accurate Molland et al. (2016). Initially developed for high aspect ratio blades for aircraft this method was inaccurate

for marine propellers due to their low aspect ratio. Lerbs (1952) released a paper using an extension to previous corrections to ensure an accurate model for marine propellers can be used. This method is still used to this day for early design developments.

Blade Element momentum theory has been used extensively in literature both as an initial design tool and to verify larger CFD simulation results Badoe (2015), Winden (2014), Phillips (2009). Lee et al. (2017a) performed a CFD simulation of a wind turbine and a one-way structural coupling was produced, this was done by mapping the CFD pressures to a FEA model using the Arbitrary Lagrangian Eulerian (ALE) method. The CFD simulations were validated by using blade element momentum theory with 5 elements and 16 elements. 7 wind speeds were tested and the main finding was that the wind speed increasing did not cause greater deformation. The greater deformation is due to the blade foil aerodynamics causing greater pressure at the tips at stall on set. The thrust force was the dominant factor in deformation not the torque coefficient.

The use of blade element momentum theory for use in modelling the propeller performance characteristics dates back to Saunders (1961) where the theory is outlined and introducing the early corrections by Prandtl (1921) and Goldstein (1929). In the use of blade element momentum theory has been used for many applications including: rotorcraft Newman (2002), wind turbines Blackwell et al. (1977), tidal turbines Harrison and Batten (2010) and marine propellers Turnock (1993).

In recent years BEMT has been used to model the effect of the propeller on the flow field to study other aspects on the ship. For example Badoe (2015) used BEMT to study the flow features of the rudder- propeller interaction and the rudder propeller skeg interaction. In addition, Phillips (2009) used BEMT with computational fluid dynamics to determine the wake of a self propeller autonomous underwater vehicle. Although the original BEMT model could not account for a non-uniform wake inflow Phillips (2009) extended the code to allow for a non-uniform inflow by splitting the annulus into radial and circumferential sections as shown in Figure 2.4.

This can add accuracy to the performance characteristics and the loading on the blade can be extracted for each blade azimuthal section giving the designer a more detailed picture of the loading distribution. This can also be extended to an unsteady RANS-BEMT simulation which was achieved in Turnock et al. (2011) where a tidal turbine arrays were studied. The RANS-BEMT coupling was used to study the changes in turbine positioning without the need to simulate the propeller thus reducing computational expense.

2.3.2 Panel methods

Panel methods were first introduced by Hess and A.M (1964) who used the surface source method in 1964. Since this early development panel methods have been

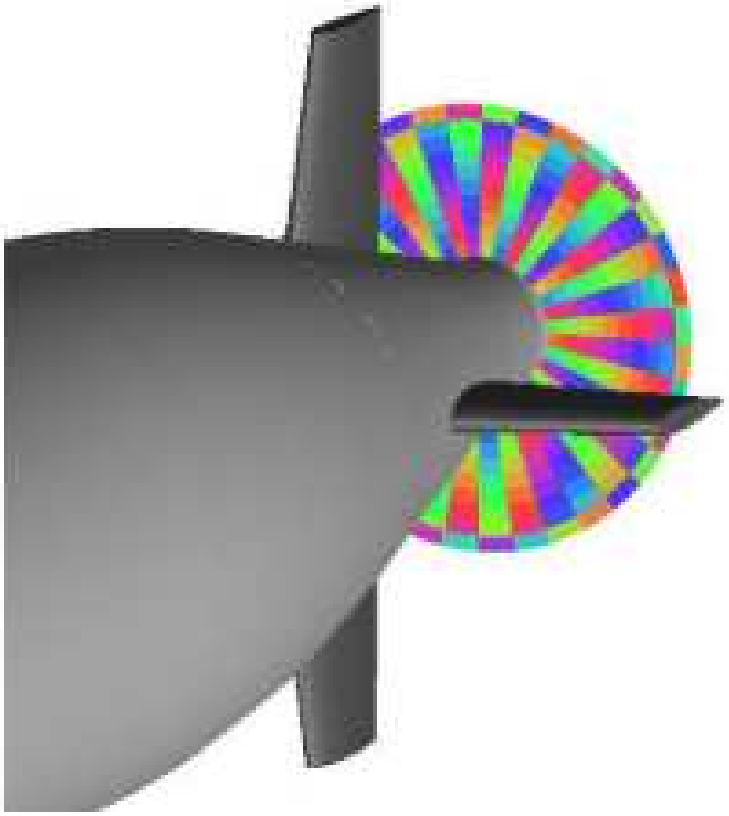


FIGURE 2.4: Circumferential division of BEMT from Phillips (2009)

adapted to model aircraft wings and propellers as well as marine propellers by including various corrections. The adaptability of these methods for many applications have made them a popular choice amongst engineers. The basic principle of panel methods is where the propeller is modelled by a mesh of panels whose source strength varies over the body surface Carlton (2012). This is a potential flow based solver and can give performance characteristics of a propeller design with good confidence Molland et al. (2011). Due to the potential flow nature of panel methods viscosity is accounted for by empirical corrections. Also, the wake of the propeller must also be modelled. This requires a mesh with accurate wake contraction which is not know prior to the simulation.

Despite the disadvantages panel methods they give accurate results at a reasonably low computational cost. They have also become very popular within literature due to the good accuracy and relative simplicity of the simulations. Brizzolara (2008) did a systematic study on the comparison between RANS and Panel methods. The author found the panel methods performed relatively poorly for low-advance ratios and off-design conditions where the dependence on Reynolds number was high. The use of empirical corrections used for viscous effects in panel methods were found to be too inaccurate to represent the propeller at off-design conditions. The paper found that using RANS to inform the cause of inaccuracies of Panel methods was a useful exercise.

Moulijn (2015) compared panel methods to a full CFD simulation and found the agreement in pressure distributions for an open water ducted propeller was excellent. Chen (2019) developed a method of improving the panel method by the inclusion of B-spline interpolation. This was shown to improve the hydrodynamic performance prediction of the propeller.

2.3.3 High Fidelity Flow Simulations Techniques

2.3.3.1 General Overview

A summary of the methods for modelling/simulating propeller performance are shown in Table 2.1.

A widely used method to model fluid flow is computational fluid dynamics. Computational Fluid Dynamics (CFD) is the study of fluid flow using numerical simulations. There are many levels of simulation involved in CFD from simple 2D laminar flow modelling to large scale 3D direct numerical simulations. Methods of computational fluid dynamics include modelling laminar flow, Reynolds Averaged Navier Stokes (RANS) Turnock et al. (2011), detached eddy simulations (DES) Squires (2003), large eddy simulations (LES) Abbas and Kornev (2016) and direct numerical simulations (DNS) Chang et al. (2002).

To effectively model the fluid flow a domain is firstly established, this is the region in which the fluid being studied will pass through. The domain contains the geometry of the object being studied and will need to be broken into several subdomains. Each subdomain contains cells which are used to discretize the domain such that a solution can be found. For computational fluid dynamics there are several numerical schemes to compute the solution of the partial differential equation.

To simplify a complex 3D problem a 2D solution is often considered first. This is useful as the grid size is greatly reduced thus reducing computational expense. Also, 2D grids are far easier to set up resulting in less human effort. The disadvantage with 2D simulations is they don't account for the inherent 3D effects of fluids and therefore are not accurate for higher fidelity simulations such as large eddy simulations Lloyd et al. (2014).

Turbulence is a phenomena which occurs in nature and engineering far more often than laminar flow, therefore great efforts have been made to model turbulence in computational fluid dynamics. A method for this is using Reynolds Averaged Navier Stokes equations (RANS). RANS modelling takes each variable and writes it in the form of the sum of a time averaged value and a fluctuating value.

$$\phi = \bar{\phi}(x_i) + \phi'(x_{i,t}) \tag{2.5}$$

TABLE 2.1: Summary of propeller modelling techniques from Molland and Turnock (2007)

Method	Description	Computational Cost
Blade Element Momentum Theory	Combination of blade element theory and momentum theory to compute the thrust and torque coefficients and load distribution from blade root to tip	1
Vortex Lattice Method (Surface panel/lifting line/ lifting surface)	The propeller is split into many panels and the induced velocity is computed using the circulation and the Biot-Savart law	$\sim 10^3$
RANS	Time averaged Navier stokes equations are solved using a finite volume method. An appropriate turbulence closure model is used to model the Reynolds stresses and, if applicable, the near wall functions.	$\sim 10^6$
LES	Computes the Navier Stokes equations by filtering out the smaller turbulent structures and a sub-grid model is used to compute the velocities caused by the small eddies.	$\sim 10^8$
DNS	Navier-Stokes equations are numerically solved. All spatial and temporial scales must be solved which causes small time steps and large grid size.	$\sim 10^{11}$

Where $\bar{\phi}(x_i)$ is the time averaged value and $\phi'(x_{i,t})$ is the fluctuating value. This is applied to the Navier Stokes equations to give the Reynolds averaged Navier Stokes equations as shown in index notation in equation 2.6.

$$\rho\bar{u_i}\frac{\partial\bar{u_i}}{\partial x_j} = \rho\bar{f_i} + \frac{\partial}{\partial x_j}[-\bar{p}\delta_{ij} + \mu(\frac{\partial\bar{u_i}}{\partial x_j} + \frac{\partial\bar{u_j}}{\partial x_i}) - \rho\overline{u_i'u_j'}] \tag{2.6}$$

It is noted that equation 2.6 is not closed, meaning there are more unknowns than equations. This is caused by the presence of the Reynolds Stress term $\overline{u_i'u_j'}$. To overcome this issue turbulence closure models have been developed such as the Spalart-Alamaras model, $k-\omega$ and $k-\epsilon$ models.

The Spalart-Alamaras turbulence model was developed by Spalart and Alamaras in 1992 Spalart and Allmaras (1992). This is a one-equation transport equation for the turbulent viscosity. The equation includes a non-viscous destruction term that depends on the distance from the wall. The model is local so it does not depend on the solution at other points. The study found that the model performs well in the near wake region.

The $k-\epsilon$ turbulence model is a two equation model solving the transport equation for turbulent kinetic energy k and turbulence dissipation rate ϵ. This model is the most widely used model for industry applications performing well in a variety of simulations however it performs poorly in flows of strong separation and pressure gradients Wilcox (1993). The turbulent kinetic energy is described by equation 2.7 and the turbulence dissipation rate is described in equation 2.8.

$$k = \overline{u_i' u_i'}/2 \tag{2.7}$$

$$\epsilon = \nu \overline{\frac{\partial u_i'}{\partial x_j}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)} \tag{2.8}$$

The $k-\epsilon$ model contains terms which cannot be calculated at the wall. Therefore wall functions must be introduced. Wall functions are used to describe the flow near the wall. For turbulent flow this is described in three layers; the viscous sub-layer, the logarithmic layer and the wake layer.

To effectively describe the turbulent boundary layer wall units must first be defined. Wall units, y^+ are the non-dimensional distance from the wall used to describe the law of the wall. The equation for wall units are shown in equation 2.9

$$y^+ = \frac{u^* y}{\nu} \tag{2.9}$$

Where u^* is the non-dimensional friction velocity $u^* = \sqrt{\frac{\tau_w}{\rho}}$, ν is the kinematic viscosity, y is the distance from the wall, τ_w is the wall shear stress and ρ is the fluid density. Using the non-dimensional functions the law of the wall can be established.

The law of the wall is a universal law that describes all turbulent boundary layers. The first layer is the viscous sub-layer. In this region there is a linear relationship between y^+ and u^* which occurs at $y^+ = 0-5$. The next region is the logarithmic region, in this region the relationship between friction velocity and wall units is shown in equation 2.10.

$$u^* = \frac{1}{\kappa} ln(y^+) + B \tag{2.10}$$

Where κ is the Von-Karman constant and B is an empirical constant dependent on the surface. The final region is the wake layer, in this region the flow no longer follows equation 2.10 as it reaches the free stream velocity. A standard graph of a turbulent boundary layer is shown in Figure 2.5.

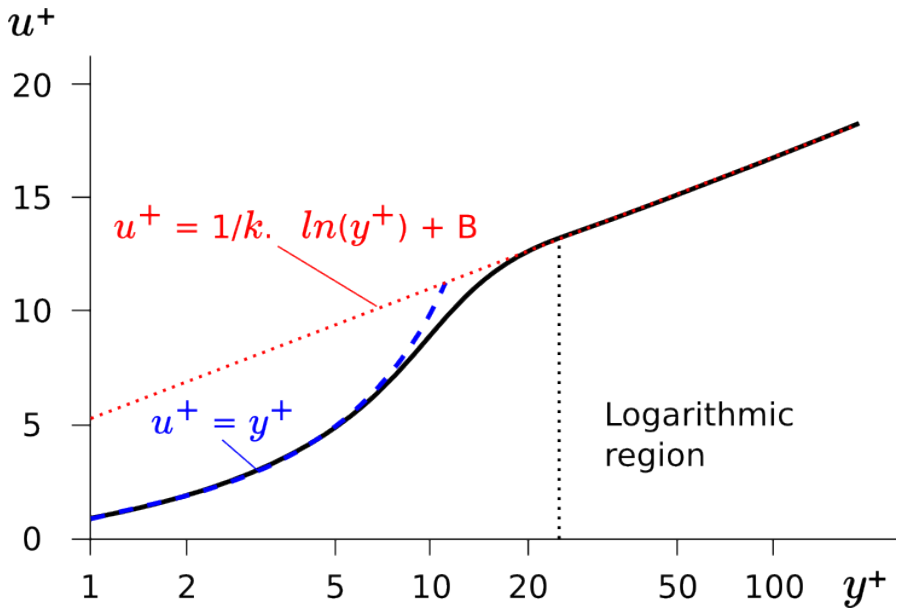


FIGURE 2.5: Law of the wall example

A turbulence model which does not require wall functions is the $k-\omega$ model. This is because all terms are defined at the wall Fluent (2006). This is a two equation model which accounts for turbulent kinetic energy and specific dissipation rate $\omega = \frac{\epsilon}{k}$. A popular variety of the $k-\omega$ model is the shear stress transport $k-\omega SST$. This model has a blending function such that the standard $k-\omega$ model is used in the near wall region and the model changes to the $k-\epsilon$ model as the distance from the wall increases to the outer portion of the boundary layer Wilcox (1993). The $k-\omega SST$ model has grown in popularity and has been used in many RANS based studies Lidtke et al. (2015), Shen and Korpus (2015),Viitanen et al. (2017).

Large Eddy Simulations (LES) is a more computationally expensive way of modelling turbulent flow. A key characteristic of turbulent flow is that has a variety of length and time scales in the fluctuations. There are both large vortical structures as well as small eddy vortices. The large scale vortices are far more effective at transportation of conserved properties, such as momentum. Therefore it is key to simulate the large scale turbulent structures over the small scale properties. LES is fully 3D and time dependent, it is more computationally expensive than RANS simulations but less expensive than full Direct Numerical Simulation (DNS). However, DNS is usually

preferred over LES unless the flow is of high Reynolds number or the geometry is complex Ferziger and Peric (2002).

LES works by firstly defining the quantities to be computed. The velocity field is filtered to only include the large scale components. The function for this in 1D is:

$$\bar{u_i}(x) = \int G(x, x')u_i(x')dx' \tag{2.11}$$

$G(x, x')$ is the filter function. This filtering can then be used to filter the incompressible Navier-Stokes equations resulting in equation 2.12

$$\frac{\partial \rho \bar{u_i}}{\partial t} + \frac{\partial(\rho \overline{u_i u_j})}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}[\mu(\frac{\partial \bar{u_i}}{\partial x_j} + \frac{\partial \bar{u_j}}{\partial x_i})] \tag{2.12}$$

The difficulty arises with the $\overline{u_i u_j}$ term as $\overline{u_i u_j} \neq \bar{u_i}\bar{u_j}$. This results in an additional term to equate these terms which results in further modelling.

The most accurate method of simulating turbulent flow is to use Direct Numerical Simulations (DNS). This method does not use any modelling or approximations but only numerically discretises the Navier Stokes equations. The considerations required for a DNS simulation is for the simulation domain to be large enough to capture the largest scale fluid structures and long enough to capture the lowest frequency simulations. The grid within the simulation domain must be fine enough to capture the smallest vortical structures. This is where the large numeric expense occurs with DNS as often the grid must be very fine resulting in high memory and processor time requirements. The benefit of DNS is that it produces large amounts of detailed data. It is therefore used as a research tool as opposed to a design tool.

2.3.3.2 Synthetic Turbulence Models

Synthetic turbulence models are used to generate a fluctuating velocity field from turbulence information. The can provide a fast and numerically efficient method of simulating random fields. Patruno and Ricci (2018) describes the methods used for generating synthetic turbulence models for spectral methods. The focus of the paper is to describe methods of determining the inflow conditions for Large Eddy Simulations in CFD.

Lloyd (2013) used synthetic turbulence models as an initial condition for large eddy simulation to simulate tidal turbines. Backaert et al. (2015) uses synthetic turbulence models to develop a efficient pre-computed turbulent flow field in the context of aerospace applications.

2.3.3.3 OpenFOAM

OpenFOAM (OPEN source Field Operation And Manipulation) is an open source simulation software which is used as a general numeric solver. The software is C++ orientated and has the benefit of being open source meaning the source code can be added to and changed to suit the user. The solver can be used in a wide variety of applications such as: fire modelling, turbulence, acoustics, solid mechanics and electromagnetism. The adaptability and low cost gives OpenFOAM and advantage over commercial codes .However, commercial codes have the advantage of official support and an easier to learn user interface.

OpenFOAM has been used in many engineering studies in both academia and industry, with good documentation and online support making it a useful open source option.

An OpenFOAM simulation is built using several files with a specific file structure. The case study must be stored in its own directory and consist of an initial conditions folder where the initial pressures and velocities are stored as well as boundary conditions. A system directory must also exist with meshing files, solution procedures and writing controls defined. The final directory required is the constant directory. This contains the geometry files of the case and the files defining the turbulence properties.

The OpenFoam CFD portion can simulate laminar, RANS,URANS, DES, LES and DNS Greenshields (2016). To simulate a steady flow simpleFoam is often used. This is using the SIMPLE algorithm which stands for Semi-Implicit Method for Pressure-Linked Equations. This works by:

1. Set the boundary conditions of the problem
2. Compute the pressure and velocity gradients
3. Solve the discretized momentum equation to compute the intermediate velocity field.
4. Compute the uncorrected mass fluxes at the faces
5. Solve the pressure correction equation to obtain the pressure field.
6. Update the pressure field using the relaxation factor.
7. Using the pressure field and the mass equation compute the velocity field.

The resulting velocity field is then fed back to the velocity gradients and boundary conditions and the process is repeated until convergence is reached.

To simulate unsteady flow the PIMPLE algorithm is often used, this is an extension of the SIMPLE algorithm where the SIMPLE algorithm is employed until convergence is achieved with several 'corrector loops' then the simulation moves to the next time step. It has been shown that the PIMPLE algorithm has good stability characteristics even when the Courant number is above unity Newman (2002). Where the Courant number is defined as the ratio between the distance the fluid particle will travel over the size of the cell.

2.4 Structural analysis of marine propellers

2.4.1 Vibration

Vibration is an important issue in fluid-structure interaction as small oscillatory motions can occur on the structure due to unsteady loading Carlton (1994). It is important to study as excessive vibration can effect the fluid flow field and therefore the performance. Furthermore, vibration can cause noise issues, fatigue, discomfort and potentially instabilities such as flutter or divergence which is catastrophic to the structure.

The vibration properties of a marine propeller depends the geometry of the blades and the material which the propeller is made from. Propellers made from more than one material, i.e composite materials, can have significantly different vibration properties compared to fully metallic propellers.

To model the vibration of a structure it can be simplified as a mass-spring damper system as shown in Figure 2.6.

Here the system has mass m, stiffness coefficient k, damping coefficient c and force F. The stiffness coefficient is the resistance to the force the system experiences defined as:

$$k = \frac{F}{x} \tag{2.13}$$

Where x is the extension of the spring, moreover the damping coefficient is the rate in which the damper dissipates the energy. The force produced by the damper is given by:

$$F_d = c\dot{x} \tag{2.14}$$

Given Newtons 2nd law $\sum F = ma$ and applying the spring and damper for a system under free vibration, this results in the equation of motion:

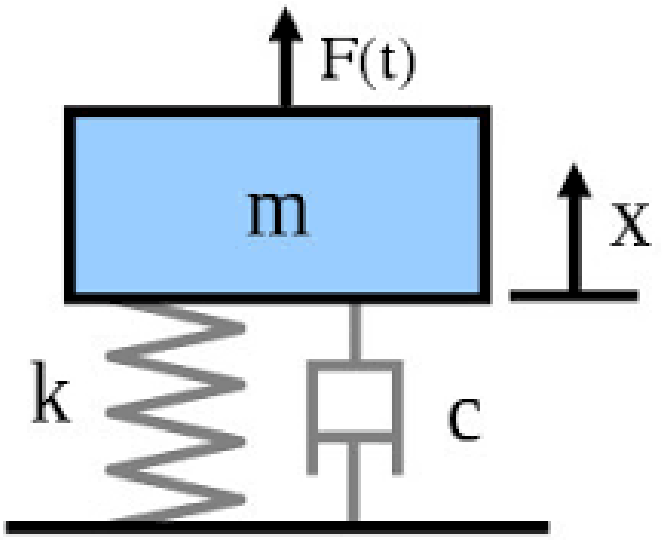


FIGURE 2.6: An example of a mass spring damper system

$$m\ddot{x} + c\dot{x} + kx = 0 \tag{2.15}$$

When dividing by mass it becomes:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = 0 \tag{2.16}$$

Where ω_n is the natural frequency and ζ is the viscous damping factor.

The natural frequency of the system will determine the period of the oscillation whereas the damping coefficient changes the rate of decay of the oscillation. A more applicable vibration system is when the system is forced. The force on the system can be a pulse force or a force with a frequency. The equation of motion then becomes:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = F(t) \tag{2.17}$$

Where $F(t)$ is a time dependent force input. If the frequency of the input force matches the natural frequency of the system the amplitude of the oscillation will be large, this is called resonance. When modelling a system with multiple degrees of freedom the equation of motion is represented by a system of equations such that:

$$\underline{\underline{M}}\,\underline{\ddot{x}} + \underline{\underline{C}}\,\underline{\dot{x}} + \underline{\underline{K}}\,\underline{x} = \underline{F} \tag{2.18}$$

Where $\underline{\underline{M}}$ is the mass matrix, $\underline{\underline{C}}$ is the damping matrix, $\underline{\underline{K}}$ is the stiffness matrix. The displacement vector $\underline{x}$ and its derivatives will have terms for each of the degrees of freedom and there will be as many natural frequencies and mode shapes as degrees of freedom. Mode shapes are the shapes of the oscillations which the mass will vibrate at when excited at the corresponding natural frequency. The mode shapes are determined by considering a single mode as:

$$[\underline{\underline{K}} - \omega^2 \underline{\underline{M}}]\underline{\phi} = \underline{0} \tag{2.19}$$

Where $\underline{\phi}$ is the mode shape. The system shown in eq 2.19 is an eigen-value problem and can be solved to determine the modal amplitudes at each degree of freedom to then determine the mode frequencies and shapes.

Propellers have natural frequencies ω_n which can be excited through normal rotation or by the vibration from the shaft or engine. As the propeller is producing a force and it is immersed in a fluid, additional sources of vibration are present known as flow induced vibration. A key aspect of flow induced vibration is vortex induced vibration where vortices shedding from the trailing edge of the propeller blades cause oscillating pressure on the trailing edge on the blade. The frequency of the oscillations can match with the natural frequency of the blade causing a phenomena called singing.

2.4.2 Modelling Propeller structures using Finite Element Analysis

The classic theories to determine the structural properties of solids have been well established Timoshenko (1930). The theories developed can be used to analytically determine the properties of a simple structure. However, as the structure becomes more complex efficient numeric techniques are required. For this reason Finite Element Analysis (FEA) has been established by Turner (1956) with a comprehensive overview of the finite element method found in Zienkiewicz (1977).

To obtain the Mass, Stiffness and Damping matrices of a system finite element analysis is often used. The matrices of each element can be easily approximated and collated to give an accurate solution for the whole structure. Several element types can be used for the mesh each with there own advantages and disadvantages depending on the application. The element types can range from a 1D beam element which models the structure as a series of nodes in one direction. The number of dimensions can be increased to a 2D plate model or a full 3D model.

Tannous et al. (2014) presents a method which enables the switch between a beam and a 3D model. The 3D model is better for local and non-linear effects, the simplified beam model is more suited for linear phenomena occurring for a long period of time.

The author uses explicit schemes for non-linear regions and for small time steps. Implicit schemes are unconditionally stable therefore larger time steps are used. The 3D portion is restricted in time and space and decomposition methods are used to deal with different subdomains, such methods are finite element particles or harmonic particles. With respect to restriction in time the model starts as a beam model then switches to a 3D model before a non-linearity occurs then switches back. A good way of checking the validity of the solution is to ensure the energy in the system remains constant. Two switch methods are presented: numerical damping to filter possible oscillations in accelerations and velocities, the second called triple static switch procedure which doesnt require numeric damping. The switch is shown to work for both static and dynamic cases and the solution to the 3D switch case is very close to the fully 3D reference model with no energy inserted or removed from the system.

Further exploration of numerical methods for FEA was undertaken by Noels et al. (2002) where the desire to study implicit and explicit algorithms was identified due to implicit algorithms being more suited to slower dynamics problems and explicit algorithms being more suited to fast dynamics. Three methods allow a combination: First integrates in time some forces with an implicit method and others with explicit one. The second method separates the mesh into subdomains with different integration techniques and time steps and the third integrates over some time intervals with an implicit method and others with an explicit. The scheme presented integrates some time intervals with an implicit scheme and others with an explicit. Automatic criteria that decide the shift is developed which is based on integration error ratio between CPU time of both schemes with stable initial conditions have also been presented.

The modelling of the forces on the propeller shaft caused by the propeller has been studied by Lee et al. (2017b) which presents the results of the estimation on the propeller shaft forces and their effects on the bearing loads during a straight run and a manoeuvre. Single and contra-rotating propellers are selected as the target vessels and the hydrodynamic loads and moments are computed using the RANS CFD technique. Bearing loads were investigated with the elastic alignment calculation. It was found that the propeller hydrodynamic forces do not affect the loads of the bearings supporting the shafts of the contra-rotating gearbox. Extreme loads occur at the early stages of turning which can cause greater amplitude of vibration making manoeuvring conditions a key aspect to the vibration analysis of the EngD.

Different load conditions are established when studying the structural response of a propeller. Lelong et al. (2016) developed an experimental procedure to analyse the FSI of a flexible foils operating in heavy fluid and cavitating flow. In non-cavitating case the foil experienced mainly bending and very little twist. Mode shapes were clearly seen in both air and water with a net decrease in modal frequencies in water compared to air. A large vibration peak occurred near the twisting mode frequency for

low angle of attack at high Reynolds number. This is believed to be caused by coupling of the structure and a laminar separation bubble inducing transition. In partial cavitating flow the mean value of the strains and stresses tended to slightly decrease in the early stages of cavity development. As the cavitation index increased the strains increased as well as the pressure fluctuations and the vibration frequencies spread to a large bandwidth and the global level of vibration increased. The author believed this was due to added mass and water vapour around the foil.

Different methods of finite element modal analysis are described in Vysoký (2017). The three cases discussed were the classic approach, cyclic symmetry modelling and the component mode synthesis. It was found the accuracy of all methods were equivalent, the only difference was the speed of computation with the Cyclic symmetry method being quickest whereas the component mode synthesis method was the slowest.

There have been many studies with the aim to assess the vibration properties of propellers and propeller blades. Lee et al. (2014) studied the free vibration of a composite propeller blade and assumed to be cantilever rigidly attached.This was an iterative procedure to calculate new blade geometry and then to compute the loading. It was found that there was an abundance of literature for computing vibration modes in air and a vacuum but not in water, therefore added mass is calculated from the surface pressure induced by the flow. It was found that the Manganese Aluminium Bronze blade had significantly lower natural frequencies in water compared to air. An analysis was performed for a composite blade. The effect of added mass is more significant in composite blades and mode shapes are very similar in air and in water. It was found in this study that the effects of rotational can be ignored as the rotation rate is not high enough in propellers.

Chen et al. (2019) studied the vibration of a marine propeller-shaft-hull system. Here the different subsystems are modelled using different techniques with the shaft modelled as beam elements, the propeller by cyclic elements and the hull by 3D elements. This was shown to be a numerically efficient method of obtaining the vibration characteristics of the whole system. The benefit of using this method as opposed to using full 3D FEA for the whole system is savings in computational cost. However, no figures outlining the benefit are provided in this paper.

2.5 Ship Manoeuvring

As the propeller is operating in a behind configuration the inflow to the propeller must be obtained. The wake causes a non-uniform, unsteady load to the propeller blade which causes a dynamic response. Studies involving the wakes formed by ships has been reviewed in literature. The early literature studied the wake fraction of a ship

operating under a steady condition whereas the later studies performed unsteady non-prescribed manoeuvring.

Schetz and Favin (1977) was one of the earliest papers on modelling the flow from a body with the influence of a propeller with the flow modelled as RANS and the propeller modelled using an actuator zone. Stern et al. (1986) developed a method to predict the propeller-hull interaction using RANS and a body force model. The developed numerics agreed well with the experimental data however the unsteady flow regime was identified as an area of required improvement.

Taylor et al. (1998) describes the computational capability for predicting the unsteady flow field around a manoeuvring submarine with a propulsor using URANS. The results of several manoeuvring conditions are discussed. Venkatesan and Clark (2007) used URANS and dynamic meshing to predict the flow field of a manoeuvring submarine with appendages and a propeller. The results were compared to experimental measurements and agree well. Subramanian (2010) coupled a vortex lattice method with a RANS solver, the vortex lattice method produces the propeller forces and the RANS solves the wider flow field.

Badoe (2015) produced a comprehensive CFD design method for studying the flow behind a ship with hull propeller interaction and included the interaction with a skeg also. Phillips (2009) described the use of computational fluid dynamics to obtain the manoeuvring conditions of an autonomous underwater vehicle. Here the use of BEMT to model the propeller within the RANS environment was established. Winden (2014) studied the operation of a self propeller ship in waves and outlines the numeric procedures to obtain this.

The use of LES for numerical simulation of ship hulls has become more prominent in recent years with increased computational power available. Abbas and Kornev (2016) validated the use of URANS and LES simulations on the KVLCC2 hull form.

The SIMMAN conferences have been created in an effort to improve methods for simulating ship manoeuvring. For this reason three test case geometries have been established; the KVLCC2 tanker, the KCS container ship and the DTMB 5415 frigate hull forms. This has generated strong experimental data Abbas and Kornev (2016) which can be used as validation data. Xing et al. (2012) used DES to determine the vortical and unsteady features of the KVLCC2 at a drift angle of 30^o, this paper showed several main features emerged; a shear-layer vortex, a 'Karman-like' vortex and three helical modes are observed. Abdel-Maksoud et al. (2015) verified the results using experimental and numerical techniques. The underside of the ship form with vortical structures can be seen in Figure 2.7.

FIGURE 2.7: Vortical Structures from KVLCC2 Abdel-Maksoud et al. (2015)

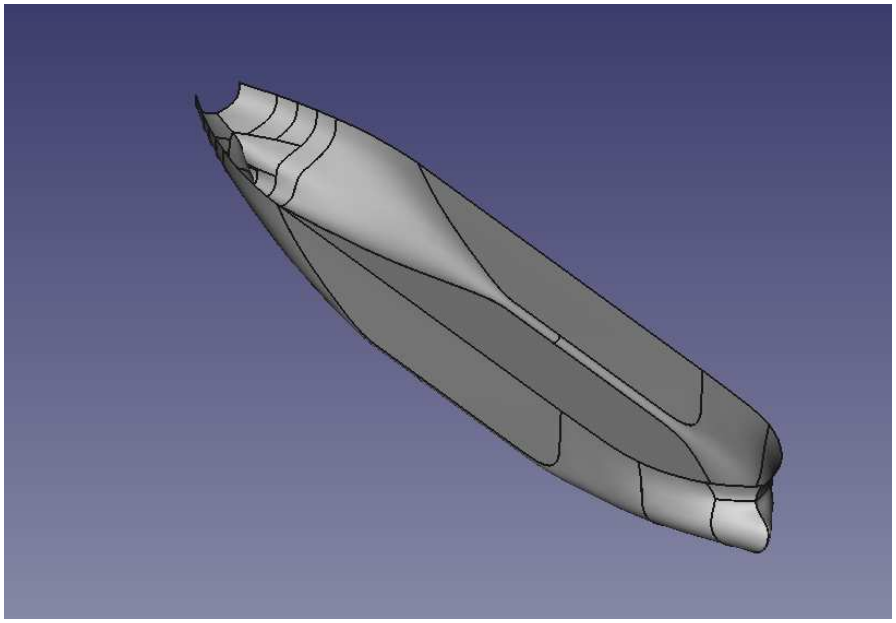


FIGURE 2.8: Isometric view of the KVLCC2 ship hull form

The highly unsteady nature of the flow field around the KVLCC2 in manoeuvring makes it an ideal candidate to study the vibration properties of the propeller blades.

2.6 Numeric Modelling of Fluid Structure Interaction of Propellers

With the increase use computational power fluid-structure interaction has been more readily studied in recent years. This section discusses the previous use of fluid structure interaction for both high fidelity simulations and design tool applications.

As Panel methods compute the pressures on the blade using a well defined mesh it is often used in Propeller FSI design algorithms. The application of FSI to propeller analysis was conducted by Young (2008) who uses BEM and FEM to explore the advantages of composite propellers. The aim of the study was to exploit the bend-twist coupling of composites. Previous studies suggest use of composites reduced vibration by 25% Ashkenazi et al. (1974). The focus of paper is to improve the understanding of hydroelastic behaviour of flexible composite propellers in sub

cavitating and cavitating flow. An interest in different propeller geometries have also been investigated with Ghassemi et al. (2012) investigating the effect of skew angle on the performance of a flexible marine propeller. This study uses boundary element method coupled with finite element method. As the skew angle increases the maximum blade deflection increases and the thrust coefficient and torque coefficient decrease. For all skew angles as the advance ratio increases the thrust and torque coefficients decrease as does the maximum deflection. As the skew angle decreases the maximum efficiency advance ratio increases.

Different wake inflows has been studied by He et al. (2012) where the aim of the paper is to optimise the hydroelastic performance of a composite marine propeller to reduce vibration and dynamic stress. FEA coupled with CFD for a propeller in a non-uniform wake is employed and the objective is to minimise the vibratory hub loads. Vibratory loads are decreased by nearly 49.6% by changing the blade composite blade stacking. The use of FSI analysis has also been extended to the optimization of flexible propellers. Blasques et al. (2010) studied the design and optimization of a flexible composite marine propeller with the aim to tailor the laminate to control the deformed shape of the blade. The laminate layup which minimizes the fuel consumption for the cruising and maximum speed is determined with a reduction of 1.25% in fuel consumption achieved with no failure under normal operating conditions after calculation of Tsai-Wu failure criteria.

FSI has been used to optimize the design of composite propellers. Composite materials have deformation coupling, this means that as the material bends it can also twist. Engineers have attempted to utilise this property by changing the geometry or laminate design such that, with the load applied, the material deforms to produce a more efficient shape for that loading condition.

In Akcabay and Young (2019) a study was performed to compute the static flutter characteristics of a composite plate in water and air. The change in fluid caused the natural frequency to lower and damping coefficients to increase. However, the main objective of the study was to understand the effects of laminate orientation on the plate loading.

Tian et al. (2017) performed an in depth study of the vibration response of marine propellers operating in cyclic flow, particularly at the propellers fundamental frequency. The vibration amplitude and frequencies were compared between experimental and numerical techniques. A wake screen was used to create a cyclic inflow with several advance ratios tested. Several mesh qualities were used for both CFD and FEA to act as a mesh sensitivity analysis. It was found that the fundamental frequencies computed and the experimental results matched closely. It was also found that the pressure fluctuates 7 times per revolution due to the cyclic inflow with larger fluctuations occurring at the leading edge compared to the trailing edge. The high

frequency peak occurs at 7APF (axial passing frequency) which is caused by the cyclic inflow. Experiments were carried out at J = 0 to obtain the basic vibration characteristics of the propeller. Finally the amplitude of the deflection of the prop was determined against the excitation frequency. The amplitude greatly increased as the excitation frequency approached the fundamental frequency of the prop.

Chen et al. (2017) numerically investigated the influence of elasticity of blades on the transmission of unsteady thrust. The aim is to establish an accurate method to describe characteristics of thrust transmission in the presence of elastic blades. Various flow conditions where analysed and the power spectra density of the thrust was produced with peaks occurring at the number of blades multiplied by the rotational velocity in hertz. The modal shapes were analysed and it was found that each mode consisted of 10 frequencies very close to each other (i.e the number of blades) so the modes where grouped. The main findings of this study are: both the bending and the torsional modes of the propeller blades are responsive to the random loads induced by the inflow turbulence, but only the bending modes (especially the first order) amplifies the transmitted unsteady thrust. The frequency coincidence between the first bending mode of the propeller blades and the longitudinal vibration mode of the shafting should be avoided strictly, otherwise resonant responses will be developed. A propeller with lower blade frequency can enhance the hydrodynamic damping at the first blade bending frequency but will experience more intense excitation as the turbulent energy is more concentrated at lower frequencies, especially when the turbulence integral scale is large. There are two methods of two way fluid structure interaction coupling. The first method is the monolithic approach. In this method the two simulations are run using the same spatial discretization and with using the same purpose designed code. The second is the partitioned approach treats both parts of the system as separate. This means the fluid simulation is dealt with by a specialized CFD or fluid modelling software and the structure is simulated using a dedicated FEA solver. This offers some flexibility when simulating however difficulties can arise when coupling. Different coupling software will need to be considered when developing a coupled simulation. As the simulations are run separately and data is transferred between the two, considerations therefore have to be made to ensure that the grids and mesh for each simulation line up or there is appropriate interpolation between data. Finally the convergence between the FEA and the CFD can differ causing large differences in step size.

Furthermore, different coupling techniques exist within partitioned coupling. The two techniques are weak and strong coupling. In weak coupling the fluid is computed until convergence is reached, the forces are then transferred to the structure. The structural simulation is then computed until convergence is reached, the displacement is then transferred to the fluid simulation and the grid is moved. Strong coupling differs from weak coupling by adding an extra iteration loop. In strong coupling the

fluid is computed and the loads transferred to the structure which is then computed. The geometry is then changed and the fluid mesh is updated. This process is repeated until convergence is reached. When convergence is reached the next time step is computed using the same procedure.

Fluid structure interaction has been studied in some depth in literature. Habchi et al. (2013) outlined the development of an OpenFOAM solver for fluid-structure interaction. It described the algorithm of the solver and defined three validation cases. The case involved a time dependent flow over a cavity with a flexible bottom. The displacement of the centre of the bottom was recorded and validated against other studies. The 2nd validation case was flow over a square bluff body with a flexible cantilever beam. The final validation case was two elastic flaps immersed in pulsatile flow. The flaps elastic modulus were changed and the tip displacement was recorded.

Fully coupled simulations are computationally expensive with most of the expense coming from large CFD simulations. For this reason attempts have been made to reduce the computational expense by firstly reducing the complexity of the fluid model or by improving the numerical methods. Friedmann (1980) produced a comprehensive review of the techniques available to model aeroelastic problems.

Skaar and Carlsen (1980) described the use of FEA for ship vibration analysis. Here 1D beam elements are used to model the vibration of a ship hull, whilst degree of freedom reduction was discussed. Ramberg and Levy (1938) developed a method to determine the natural frequencies and stresses of an aircraft propeller using Rayleigh's method.

Li et al. (2017) produces a 3D panel method in the frequency domain with a finite element method for structural analysis. The effect of excitation frequency, inflow velocity and material parameters are investigated as is the effect of advance ratio on added mass and damping. It was found that the added damping due to the fluid can significantly affect unsteady performance. It was also found that stationary flow is sufficient for analysing the wet modes of vibration. Seng et al. (2014) and Lee (2015) have performed similar studies with Seng et al. (2014) studying a flexible barge and Lee (2015) studying a composite marine propeller.

Computationally cheaper methods have been implemented to inform design decisions but can also be used as an optimization tool. Jiang et al. (2018) used a panel method along with the finite element method to optimize the design of a propeller used on a bulk freighter. Recently, Maljaars (2018) presented work based on the boundary element methods with FEA to produce a FSI model. Here added mass and added damping were modelled using closed expressions. It was also shown here that a reasonable levels of error from modelling the added mass and damping does not effect the FSI results significantly since the propeller blades are stiffness dominated. The modelling of these terms can greatly reduce the computational time.

Blade element momentum theory has been used with FEA to determine the performance of tidal turbines by Suzuki and Mahfuz (2018). This paper cited the use of BEMT-FEA coupled modelled for use in wind turbine applications as being fundamentally different when compared to applications in water due to changes in density ratios. This holds true for tidal turbine applications as well as for marine propellers. The significant difference between marine propellers and tidal turbines and marine propellers is geometry. Tidal turbines often have high aspect ratio blades which are modelled as beam elements LiuX et al. (2014). However marine propeller blades require different structural modelling due to high aspect ratio.

Despite the abundance of literature on the fluid structure interaction of propeller blades little work has been complete using Blade element momentum theory to determine the hydrodynamic loading. The vast majority of papers focus on panel methods coupled with FEA particularly using a steady state or URANS based approach for the inflow to the propeller.

In addition, due to the incredibly high computational cost of LES very little work has been presented in studying the blade fluid structure interaction when operating in a highly turbulent flow field. Feymark (2013) described the challenges of using high fidelity LES simulations in the context of fluid structure interaction. Here it is described that a main challenge is the scalability of a robust FSI algorithm required to run on a high performance computer.

The literature surrounding FSI of marine propellers clearly outlines a lack of knowledge when using Reynolds dependent hydrodynamic models for propellers operating in off-design conditions. The lack of highly fluctuating inflow to the propeller being studied is also apparent. This Thesis aims to full fill this gap.

2.7 Chapter Summary

This chapter firstly discusses the theory associated with propeller fluid-structures interaction. The numerical techniques used for high fidelity methods have been discussed as well as the historic and state-of the art computationally efficient models.

The literature surrounding marine propeller fluid structure analysis is predominately focused on the use of computationally expensive CFD-FEA coupling for detailed analysis of a marine propeller. However, panel methods tend to be used when obtaining the fluid-structural response of a marine propeller. These methods have the distinct limitation of not including viscous effects in the calculation. The viscous effects can make a significant difference in propeller performance and blade loading. This can be overcome by the use of blade element momentum theory which is a highly

computationally efficient method which can include viscous effects via a 2D section database.

Despite advances in finite element analysis capabilities the computational cost of a dynamic simulation can be high. The use of 1-D beam elements have been used for wind turbine applications to reduce the computational cost. However, the aspect ratio is far lower for propeller blades and a 1-D beam equation will not be applicable. Plate elements have a significant computational cost saving over 3D elements, however these have not been applied to propeller blades to solve for a dynamic response solution.

The presence of a ship hull and appendages causes a non-uniform inflow and potentially highly unsteady wake to the propeller blades. This is particularly an issue when the ship is operating in off-design conditions. The non-uniform inflow will cause the propeller blades to experience rapid load changes as it rotates thus causing a dynamic response by the propeller. With frequent loading and unloading the propeller can experience greater fatigue issues.

As shown in the literature little work has been done on fluid-structure interaction of marine propellers operating in an off design flow regime. This is due to the high computational cost of current techniques for FSI simulations operating in the behind condition. The need for a computationally efficient numeric tool for FSI at the early stages of design has therefore been identified.

Chapter 3

Propeller Performance Hydrodynamic Load Model

3.1 Introduction

Blade Element Momentum Theory is chosen to obtain the hydrodynamic loading on the propeller blades. It is a highly computationally efficient method capable of obtaining the propeller performance in open water and operating in a wake. The development and implementation of BEMT is described in this Chapter. The BEMT method has been chosen over other computationally efficient methods, such as panel methods, for two reasons. Firstly, panel methods require the wake of the propeller to be meshed. This means the description of the wake must be known before the simulation is run, furthermore, this will increase the computational expense of the method compared to BEMT. Secondly, panel methods use empirical corrections to account for Reynolds number effects, this can be particularly troublesome in off-design conditions. This can be improved upon by using accurate experimental or simulation data in BEMT.

Although the blade element momentum method can be shown to be accurate the theory can be improved by the inclusion of changes in Reynolds number. This can be achieved by the inclusion of a 2D foil database to obtain the lift-curve slopes of various foil sections. The pressure distribution on each section can also be obtained to be included for structural loading. The development of the database is discussed in this chapter.

The use of BEMT is to achieve the hydrodynamic loading in a computationally efficient manner, the computational efficiency can be improved using compiled languages whilst maintaining the ease of implementation of an interpreted language.

The objective of this chapter is to describe and verify a computationally efficient model to obtain the hydrodynamic properties of the propeller using Blade Element Momentum Theory in an unsteady flow regime.

3.2 Blade Element Momentum Theory

Blade element momentum theory combines two models, blade element theory and momentum theory. Momentum theory treats the propeller as an infinitesimally thin plate with a change in momentum occurring at the plate. The flow is modelled by having an axial velocity V occurring upstream of the plate. When the flow reaches the plate the flow velocity increases to $V_1 = V(1+a)$ where a is the axial inflow factor. Similarly, the flow has angular velocity relative to the plate of Ω upstream of the plate then at the plate the flow angular velocity is $\Omega(1-a')$ where a' is the angular inflow factor.

It can also be shown that in the far wake the axial flow velocity is $V_2 = V(1+2a)$. Therefore the thrust and torque of the disc can be given by the axial and angular change of momentum respectively. The mass flow rate through the disc is given by $2\pi r\delta r\rho V(1+a)$ and the thrust is expressed by the axial change in momentum.

$$\delta T = 2\pi r\delta r\rho V(1+a)(V_2 - V_1) \tag{3.1}$$

Similarly the torque is given by the change in angular momentum.

$$\delta Q = 2\pi r\delta r\rho V(1+a)r^2\omega_1 \tag{3.2}$$

Where ω_1 is the fluid angular velocity at the disc. So the trust and torque per unit span are given by:

$$\frac{dT}{dr} = 4\pi\rho r V^2 a(1+a) \tag{3.3}$$

and

$$\frac{dQ}{dr} = 4\pi\rho r^3\Omega V a'(1+a) \tag{3.4}$$

This will give the thrust and torque of the disc once integrated. However there are only a limited amount of blades on a real propeller. To correct for this the Goldstein correction factors are introduced Goldstein (1929) which continued from the early work by Prandtl (1921). The Goldstein factor has been calculated for 2-7 blades by

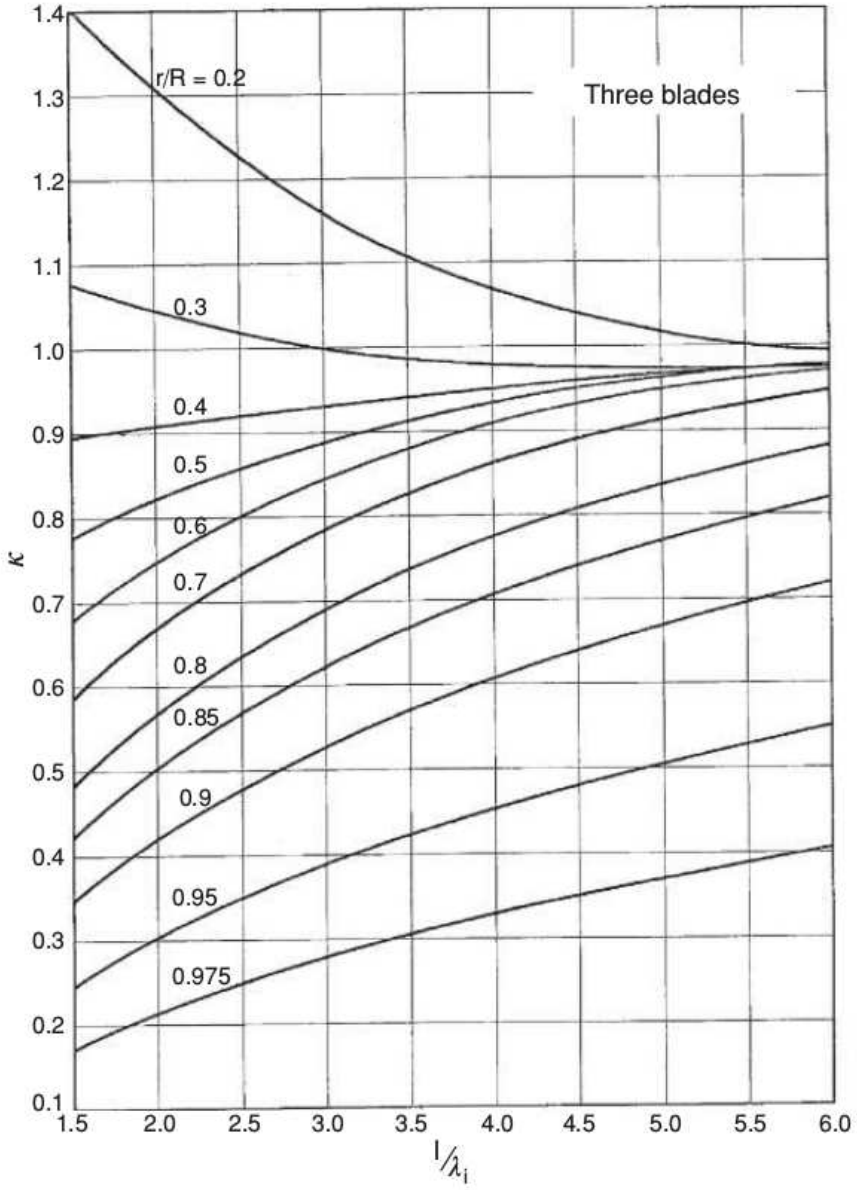


FIGURE 3.1: Goldstein Correction factors

several authors. It can be seen from Figure 3.1 that the value of the correction factor is highly dependent on radial position. Therefore a functional relationship is given by Wellicome.

$$K = \frac{2}{\pi} cos^{-1} \frac{cosh(xF)}{cosh(F)} \tag{3.5}$$

Where $F = \frac{Z}{2xtan\phi} - \frac{1}{2}$ and Z is the number of blades. This is then inserted into equations 3.3 and 3.4 such that:

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

and

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

This is the required equations for the momentum part of blade element momentum theory. To increase accuracy the blade is modelled as a 2D lifting surface at an angle of attack as shown in Figure 3.2.

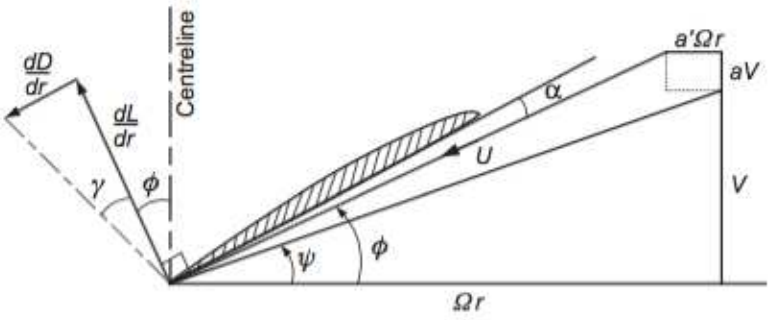


FIGURE 3.2: Blade Element theory schematic

The 2D lift and drag per radii on the lifting surface is given by:

$$\begin{aligned} \frac{dL}{dr} &= \frac{1}{2}\rho Z c U^2 C_L(\alpha) \\ \frac{dD}{dr} &= \frac{1}{2}\rho Z c U^2 C_D(\alpha) \end{aligned} \tag{3.8}$$

Where c is the blade chord, α is the angle of attack, Z is the number of blades and U is the inflow velocity. From Figure 3.2 it can also be seen that:

$$tan\phi = \frac{V(1+a)}{\Omega r(1-a')} = \frac{1+a}{1-a} tan\psi \tag{3.9}$$

The local section pitch P is the effective angle of attack plus the induced flow angle. $tan(\phi+\alpha) = \frac{P/D}{\pi x}$. Where ϕ is the hydrodynamic pitch angle. From Figure 3.2 it can be seen that the thrust and torque can be resolved into lift and drag contributions and $U = \pi n D x(1-a')sec\phi$. Elemental thrust and torque are then given by:

$$\frac{dK_T}{dx} = \frac{\pi^2}{4}(\frac{Z_c}{D})C_L x^2(1-a')^2 \sec\phi(1-\tan\phi\tan\gamma) \tag{3.10}$$

TABLE 3.1: HMRI geometry case used in BEMT

r/R	P/D	Rake	Skew	C/D	fo/C	to/D
0.155	0.576543	0.0000	-2.525	0.151495	0.031264	0.046806
0.250	0.613000	0.0000	-4.000	0.177166	0.034908	0.042166
0.300	0.631000	0.0000	-4.400	0.189166	0.035621	0.038516
0.400	0.663000	0.0000	-4.400	0.209334	0.033798	0.032016
0.500	0.691500	0.0000	-3.150	0.224666	0.029334	0.026016
0.600	0.712000	0.0000	-0.820	0.233500	0.024984	0.020550
0.700	0.721200	0.0000	2.490	0.233834	0.021895	0.015600
0.800	0.716000	0.0000	6.350	0.219166	0.019797	0.011050
0.900	0.692700	0.0000	10.760	0.180834	0.016147	0.007000
0.950	0.674800	0.0000	13.150	0.142166	0.012805	0.004717
1.000	0.651000	0.0000	16.750	0.000016	3.062500	0.003200

and

$$\frac{dK_Q}{dx} = \frac{\pi^2}{8}(\frac{Z_c}{D})C_L x^3(1-a')^2 \sec\phi(\tan\phi + \tan\gamma) \tag{3.11}$$

Local efficiency is described as $\eta = \frac{V\frac{dT}{dr}}{\Omega\frac{dQ}{dr}}$ which can then be expressed as $\eta = \frac{tan\psi}{tan(\phi+\gamma)}$ where γ is the tangent of the ratio of lift coefficient to drag coefficient.

The inflow factors are derived from local efficiency with axial inflow factor described as:

$$a = \frac{1-\eta_i}{\eta_i + \frac{1}{\eta}tan^2\psi} \tag{3.12}$$

and angular inflow factor described as:

$$a' = 1 - \eta_i(1+a) \tag{3.13}$$

The solution procedure is an iterative scheme following the procedure described in Figure 3.3.

The BEMT code was first implemented in MATLAB, this code was then adapted to be used in Python in Mccaw (2017) for ease of use. The code has the input of a geometry file such as the one shown in Table 3.1 and outputs the propeller thrust and torque coefficients as well as the propeller efficiency. The code works by following the BEMT algorithm as described.

The local thrust and torque coefficients at each blade section are computed and then numerically integrated. To ensure the propeller coefficients are correct a generic integrator is implemented. This takes the local thrust coefficients and the size of each

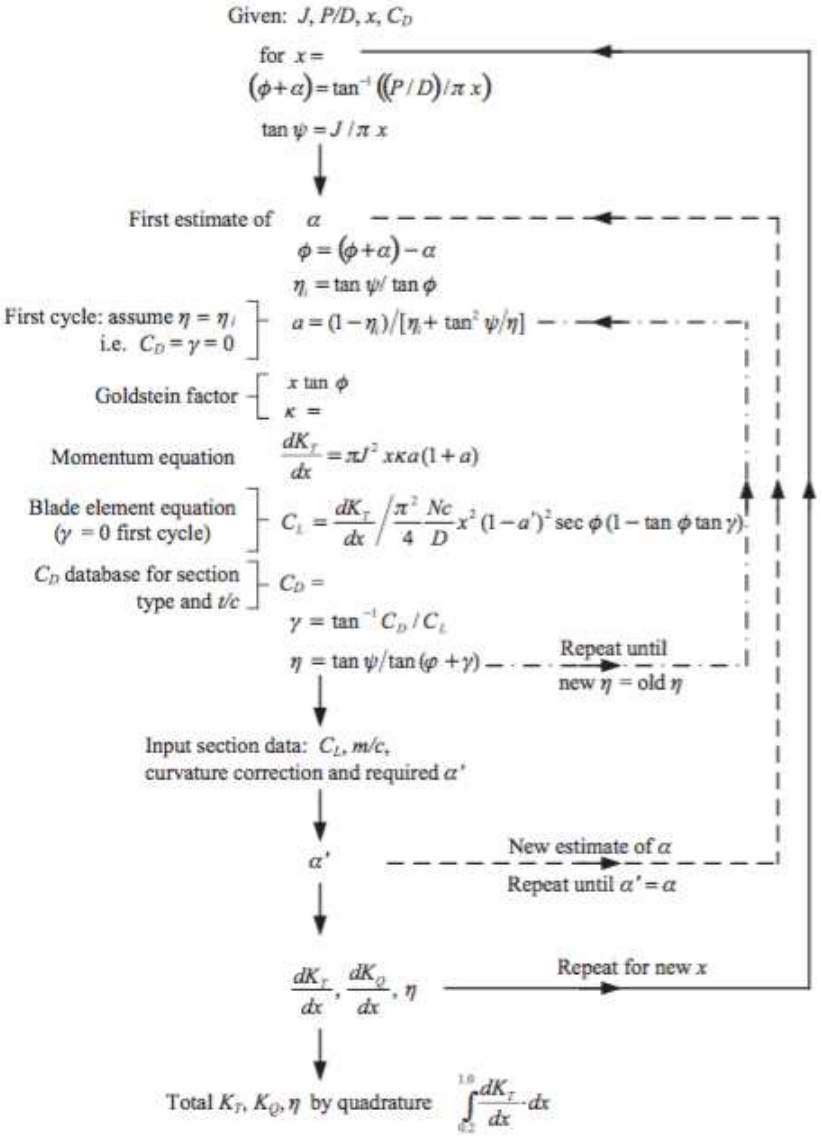


FIGURE 3.3: Blade Element solution procedure from Molland et al. (2016)

blade section and uses trapezoidal rule to approximate the integral. This code can be found in listing A.2.

```
def TrapRuleIntegration(KT_dx, KQ_dx,x_R_list):

# Integrate over blade to compute total coefficients.

f_x = np.zeros_like(x_R_list)
f_x2 = np.zeros_like(f_x)
j = 2

for i in range(len(x_R_list)-1):
h = x_R_list[i] - x_R_list[i-1]
f_x[i] = (h/2) * (KT_dx[i-1] + KT_dx[i])
f_x2[i] = h/2 * (KQ_dx[i-1] + KQ_dx[i])

Total_Thrust = np.sum(f_x)
Total_Torque = np.sum(f_x2)
```

```
return Total_Thrust, Total_Torque
```

LISTING 3.1: Generic integrator code segment

This model has initially been implemented using the interpreted programming language Python. Python is a highly usable programming language with an excellent support network. It is quickly coming popular due to its relatively simple syntax making it easy to use for beginners whilst maintaining an excellent use for scientific programming. It makes use of packages such as: SciPy, Numpy and Pandas to easily deal with numeric problems. The use of Python as the advantage of making software development times short and making code easily changeable without the need for memory management or compiling.

3.3 2D foil database

To improve the accuracy of BEMT a 2D airfoil database is used to obtain the drag coefficient and angle of attack for the required lift coefficient. This improves the accuracy as the lift curve slope is not assumed to be ideal at 2π per radian. Moreover, the influence of local Reynolds number can be included in the database which gives BEMT an advantage over potential flow solvers.

To generate the 2D foil database OpenFOAM has been used to run a series of 2D simulations with a NACA66 modified foil with several thickness' , cambers, Reynolds numbers and angle of attacks. The NACA66 modified foil shape was used for this case as it is commonly used profile for marine propellers. This database can be extended to include several profiles. A summary of the cases run is shown in Table 3.2

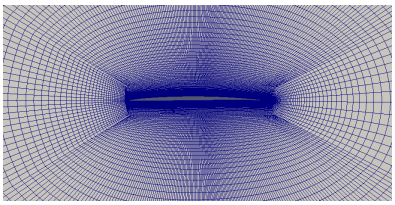
TABLE 3.2: Summary of cases for 2D foil database

$\frac{t}{c}$ / $\frac{m}{c}$	0	0.033	0.0165
0.1	Re = 50e3,150e3,250e3,500e3,1e6	...	...
0.03	⋮	⋮	⋮
0.16	⋮	⋮	⋮
0.22	⋮	⋮	⋮
0.28	⋮	⋮	⋮

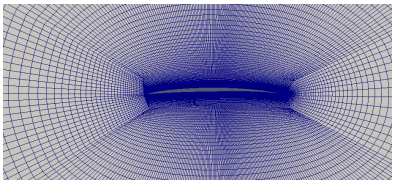
Each case was complete at angle of attacks ranging from -6^o to 16^o. This was done by generating a circular domain for each foil. The Reynolds number chosen is the likely range of Reynolds numbers the propeller with experience when at model scale. This is to coincide with the work completed by Lidtke et al. (2018).

The mesh for several foils are shown in Figure 3.4. The mesh was generated using blockMesh with each blockMesh dictionary generated using a Python script.

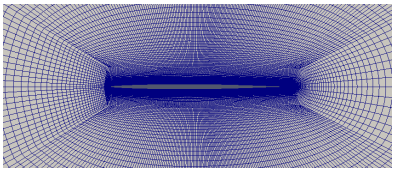
The meshes generated are of high enough quality to ensure the maximum y^+ is below 1 for all geometries and angles of attack and Reynolds numbers. Each foil has a chord of 1m and the Reynolds number is adjusted by changing the inflow velocity. The angle of attack is changed by rotating the simulation domain by the desired angle of attack.



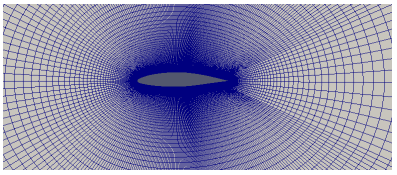
(A) Mesh for tc = 0.03, mc = 0.0165



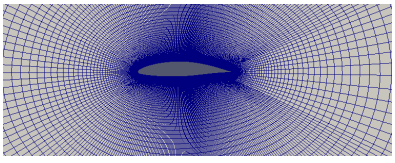
(B) Mesh for tc = 0.03, mc = 0.033



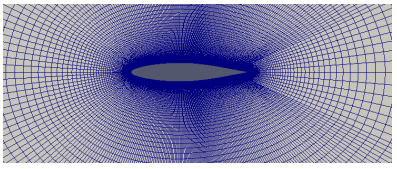
(C) Mesh for tc = 0.03, mc = 0.0



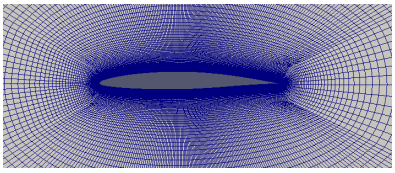
(D) Mesh for tc = 0.16, mc = 0.0165



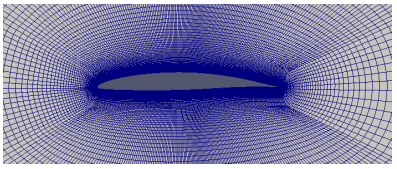
(E) Mesh for tc = 0.16, mc = 0.033



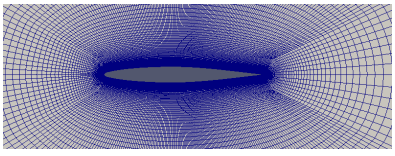
(F) Mesh for tc = 0.16, mc = 0.0



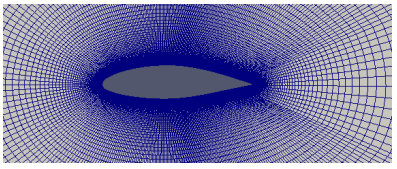
(G) Mesh for tc = 0.1, mc = 0.0165



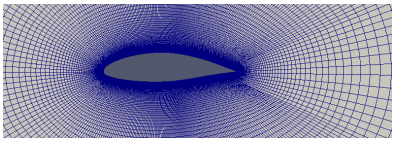
(H) Mesh for tc = 0.1, mc = 0.033



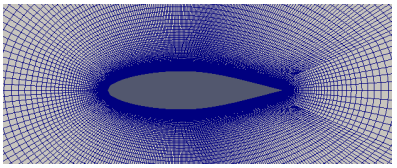
(I) Mesh for tc = 0.1, mc = 0.0



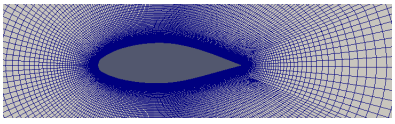
(J) Mesh for tc = 0.22, mc = 0.0165



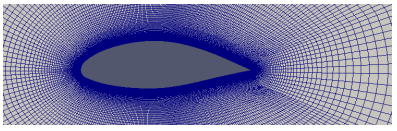
(K) Mesh for tc = 0.22, mc = 0.033



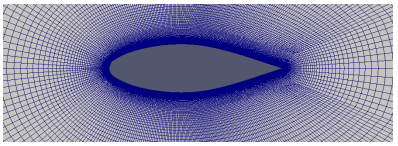
(L) Mesh for tc = 0.22, mc = 0.0



(M) Mesh for tc = 0.28, mc = 0.0165



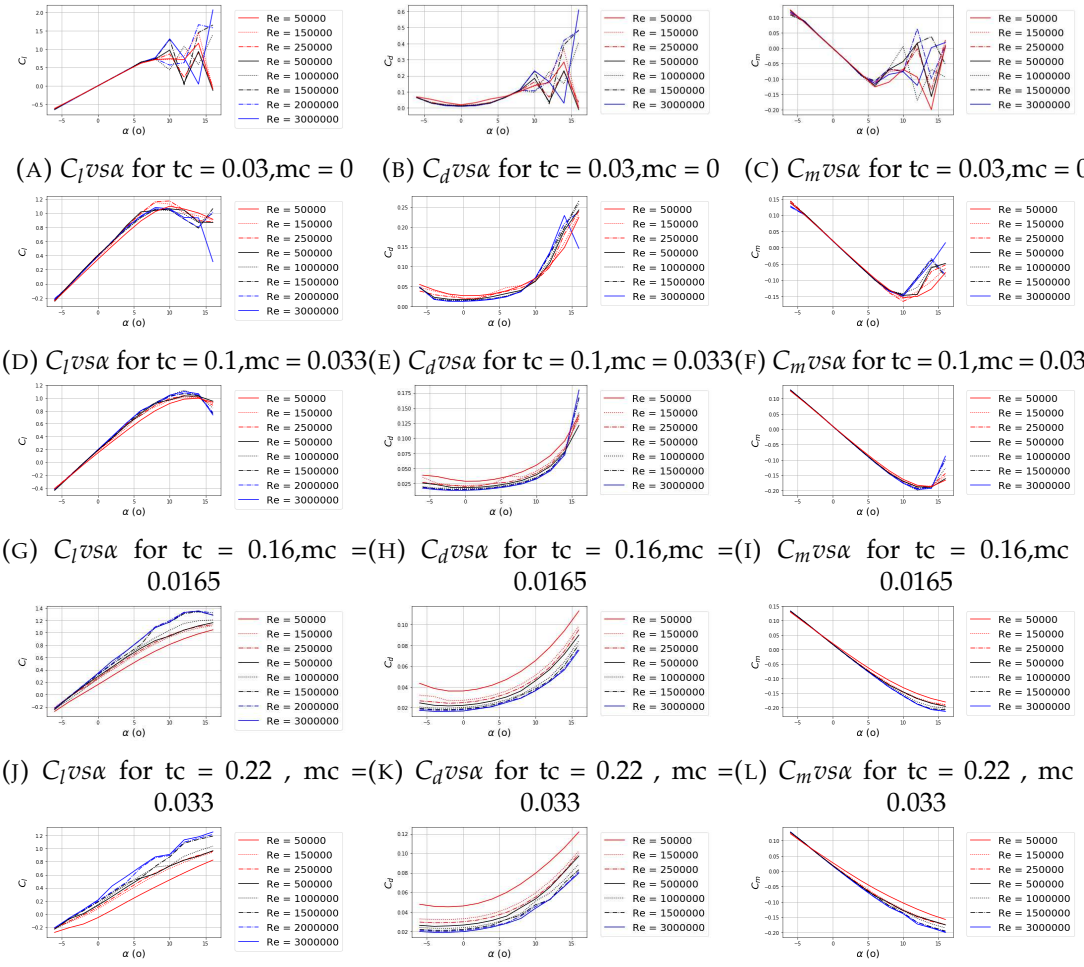
(N) Mesh for tc = 0.28, mc = 0.033



(O) Mesh for tc = 0.28, mc = 0.0

FIGURE 3.4: Mesh for several foil shapes used in CFD simulations to generate 2D foil database

These cases were completed using the simpleFOAM algorithm with a mesh size of approximately 62,000 cells for each 2D simulation. This mesh was found to be of sufficient quality to ensure the y^+ value was 1 for all simulations. The Spalart-Allmaras turbulence model was chosen for these simulations. This is a one-equation model which solves the turbulent eddy viscosity. It was chosen as it assumes the flow is fully developed turbulence. This is beneficial compared to transition models such as the $k-\omega SST$ as the flow at the propeller plane will be fully turbulent. In addition, there is greater stability when running the CFD code.



(A) $C_l vs\alpha$ for tc = 0.03,mc = 0 (B) $C_d vs\alpha$ for tc = 0.03,mc = 0 (C) $C_m vs\alpha$ for tc = 0.03,mc = 0

(D) $C_l vs\alpha$ for tc = 0.1,mc = 0.033(E) $C_d vs\alpha$ for tc = 0.1,mc = 0.033(F) $C_m vs\alpha$ for tc = 0.1,mc = 0.033

(G) $C_l vs\alpha$ for tc = 0.16,mc = 0.0165 (H) $C_d vs\alpha$ for tc = 0.16,mc = 0.0165 (I) $C_m vs\alpha$ for tc = 0.16,mc = 0.0165

(J) $C_l vs\alpha$ for tc = 0.22 , mc = 0.033 (K) $C_d vs\alpha$ for tc = 0.22 , mc = 0.033 (L) $C_m vs\alpha$ for tc = 0.22 , mc = 0.033

(M) $C_l vs\alpha$ for tc = 0.28 , mc = 0.033 (N) $C_d vs\alpha$ for tc = 0.28 , mc = 0.033 (O) $C_m vs\alpha$ for tc = 0.28 , mc = 0.033

FIGURE 3.5: Lift, drag and moment coefficients for various 2D foil sections. Several Reynolds numbers are plotted on the same graph to understand the influence of Reynolds number

The $C_l\alpha$, $C_d\alpha$ and $C_m\alpha$ curves are shown in Figure 3.5. Here it can be seen that the thin foil stalls at a lower angle of attack when compared to the the thicker foils. It should be noted that the within the stall region the coefficients are erratic and not to be taken at face value. Therefore, within the BEMT code if a 2D section is using data from this

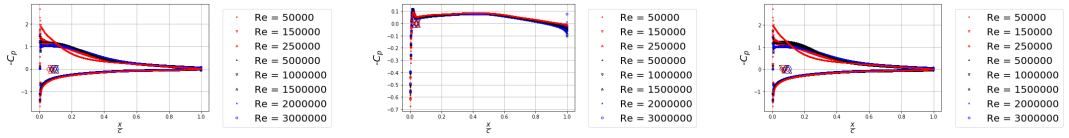
region then it should be flagged and instead use the linear portion of the curve and extend the gradient to the calculated angle of attack.

The dependence on Reynolds number can also be significant particularly for the thicker foils. The drag coefficient is significantly higher for low Reynolds number. These results compare well with the 2D foil database presented in Lidtke et al. (2018).

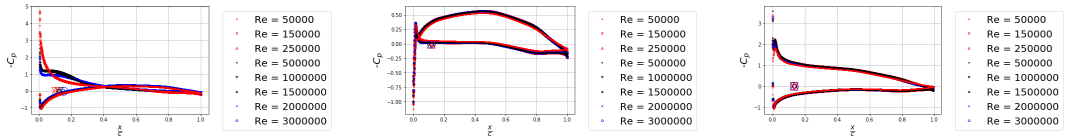
As well as the coefficient curves the pressure coefficients for each foil for each operating condition can be obtained and stored. This data will be used later to obtain the load distribution on the propeller blades.

The pressure coefficient data is shown in Figure 3.6. The coefficients for a few selected geometries is shown at angles of attack of -6^o 0^o and 6^o for illustrative purposes. The pressure differential from the upper and lower surfaces generate the lift and will therefore dictate the chord-wise load distribution in the structural model. It is also noted that changes in Reynolds number causes a change in load distribution which is captured with this model. The centre of pressure for each configuration is shown on Figure 3.6 also. It can be seen that as the Reynolds number is increased the centre of pressure shifts towards the trailing edge. The centre of pressure is calculated using equation 3.14.

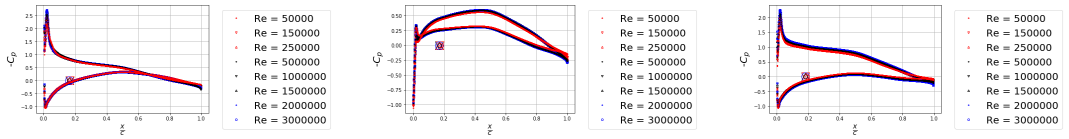
$$CoP = \frac{P(x)x}{P(x)} \tag{3.14}$$



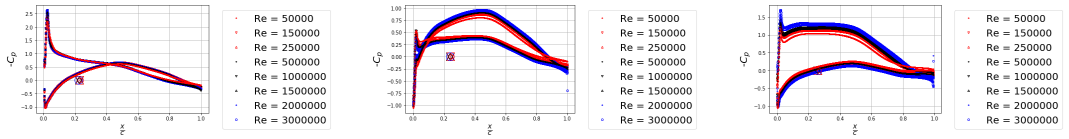
(A) Cp curves for tc = 0.03, mc = 0, $\alpha = -6^o$ (B) Cp curves for tc = 0.03, mc = 0, $\alpha = 0^o$ (C) Cp curves for tc = 0.03, mc = 0, $\alpha = 6^o$



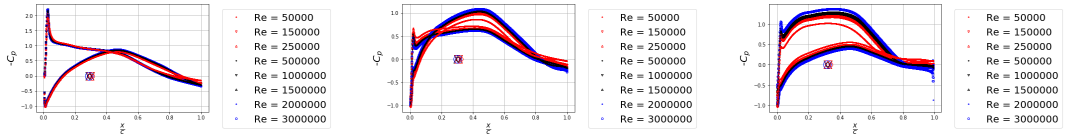
(D) Cp curves for tc = 0.1, mc = 0.033, $\alpha = -6^o$ (E) Cp curves for tc = 0.1, mc = 0.033, $\alpha = 0^o$ (F) Cp curves for tc = 0.1, mc = 0.033, $\alpha = 6^o$



(G) Cp curves for tc = 0.16, mc = 0.0165, $\alpha = -6^o$ (H) Cp curves for tc = 0.16, mc = 0.0165, $\alpha = 0^o$ (I) Cp curves for tc = 0.16, mc = 0.0165, $\alpha = 6^o$



(J) Cp curves for tc = 0.22, mc = 0.033, $\alpha = -6^o$ (K) Cp curves for tc = 0.22, mc = 0.033, $\alpha = 0^o$ (L) Cp curves for tc = 0.22, mc = 0.033, $\alpha = 6^o$



(M) Cp curves for tc = 0.28, mc = 0.033, $\alpha = -6^o$ (N) Cp curves for tc = 0.28, mc = 0.033, $\alpha = 0^o$ (O) Cp curves for tc = 0.28, mc = 0.033, $\alpha = 6^o$

FIGURE 3.6: Pressure Coefficient curves for NACA 66 section

3.3.1 Integration into BEMT code

Now the 2D foil database has been established it must be used within the BEMT loop. For the required lift coefficient obtained from equation 3.10 the angle of attack for which that lift coefficient occurs for the geometry must be found. This is done by using linear interpolation between the section thickness, camber and Reynolds number to obtain a $C_l\alpha$ curve for the blade section from which the angle of attack can be obtained. When using this option the iterative loop changes by, instead of the lift coefficient being calculated by the empirical formula the interpolation is used. However, during the first few iterations the angle of attack is often calculated as being very high. In this region the linear portion of the lift curve slope is used.

To generate the correct lift curve the thickness/chord of the blade section is compared to that of the database and the two thickness' in which the blade section lies between

is found. A weighting is then found to know the difference between the blade section thickness and the database thickness'. This is repeated for the camber and the Reynolds number. The lift curve is then calculated by multiplying the weightings by the corresponding bounding variables.

3.4 Cython

The computational cost for BEMT is small however this can become more significant when studying an unsteady, non-uniform inflow as the BEMT Python code will be required to run every time step.

To reduce the computational cost of the BEMT code it has been written in Cython. Cython is used to write software as fast as a compiled programming language but as easy to implement as an interpreted programming language. This is done by defining variables and arrays as static type declarations.

For example the Python code for defining a variable and defining it as a Cython static variable is shown in listing 3.2

```
#Python
phi_plus_alpha1 = np.zeros_like(chord)

#Cython
cdef double [:] phi_plus_alpha = phi_plus_alpha1
```

LISTING 3.2: Initializing Array for Python vs Cython

The use of Cython causes a significant decrease in computational time to perform each BEMT calculation. To quantify the computational speed up using Cython a test problem was performed using the HMRI propeller geometry. All variables are kept the same apart from increasing the number of circumferential sections used. This number of circumferential sections used is the main drive of computational cost in the BEMT solver.

3.4.1 Computational Cost

The time comparison between the Python implementation and the Cython implementation is shown in Figure 3.7 on a log-log plot for clarity. It can be seen the increase in computational cost increases exponentially with increasing circumferential sections for both Python and Cython implementations. There is however an outlier for the first circumferential section size for the Python code. This can clearly be seen when observing the relative speed up between Cython and Python in Figure 3.8

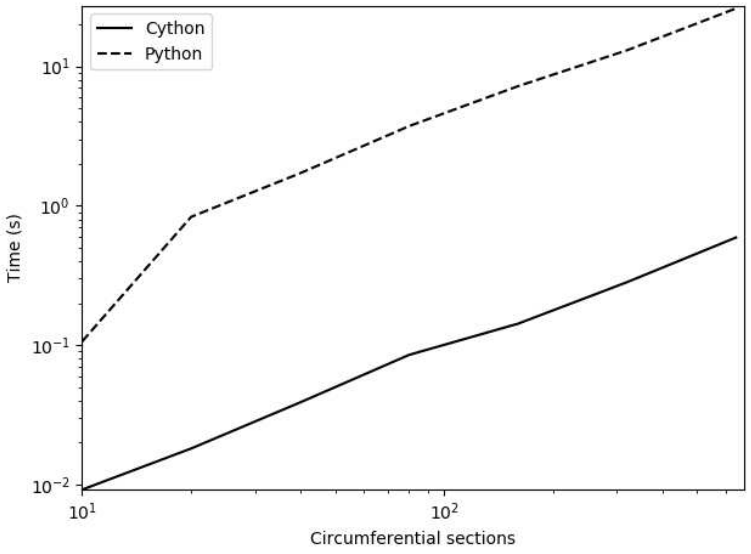


FIGURE 3.7: Cython vs Python Computational cost

The first Cython simulation speed up is approximately 10x faster compared to the Python implementation. This then increases to 45x to 50x speed up for subsequent simulations. This change in speed up seems to be driven by the Python implementation as the Python line in Figure 3.7 changes steeply for the 1st simulation. Interesting to note the initial speed up was actually a 30% slow down. After going through a line profiler $\approx$ 30% of the time was used up in the tangent function this was changed to $\frac{sin(\gamma)}{cos(\gamma)}$ which then saw the rapid speed up. Although a 50x speed up is desirable it does not fully utilize the potential of a compiled language such as C or Fortran. This is mainly due to the interaction with the foil data base which uses a Pythonic dictionary which is relatively slow.

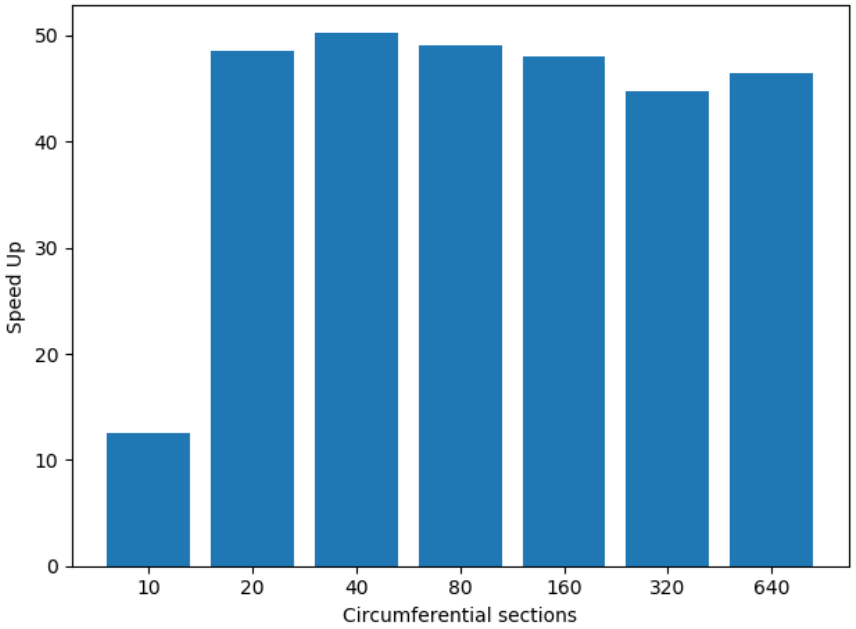


FIGURE 3.8: Relative speed up between Python and Cython models with changing circumferential sections

3.5 Theodorsen Unsteady Aerodynamics

As the propeller blade vibrates the hydrodynamic properties of each blade section will change. This is caused by changes in pitching moment and heave. If this change occurs slowly then the steady state hydrodynamic coefficients can be used. However if the change is rapid there is a time delay between the motion and the hydrodynamic effect.

Although more complex dynamic stall models are available for high angle of attack regions an arbitrary foil motion model has been implemented using the procedure outlined in Gülçat (2016).

The 2D foil section is modelled by two degrees of freedom, pitching around its mid-chord and vertical translation. The foil is simplified to a flat plate. The equation of the chord line will then be:

$$z_a(x,t) = -h - \alpha x \tag{3.15}$$

Where z_a is the vertical position, h is the distance travelled, α is the angle of attack and x is the distance from the foil mid-chord.

The downwash for the foil is then given as:

$$w(x,t) = \frac{\partial z_a}{\partial t} + U\frac{\partial z_a}{\partial x} = -(\dot{h} + \dot{\alpha}x + U\alpha) \tag{3.16}$$

Referencing the derivation in Gülçat (2016) the unsteady lift coefficient can be represented as functions of reduced time where the reduced time is given as $s = \frac{2Ut}{c}$. Here U is the inflow velocity, t is the time, and c is the foil chord. s represents the number of semi-chords the foil travels per time step.

$$C_l(s) = \frac{\pi b}{U^2}(\ddot{h} + U\dot{\alpha}) - \frac{2\pi}{U}[w(b/2,s)/2 + \int_0^s w(b/2,\sigma)\phi'(s-\sigma)d\sigma] \tag{3.17}$$

Here ϕ' is the derivative of the Wagner function. The Wagner function is a time dependent function which takes the form:

$$\phi(s) = \frac{2}{\pi}\int_0^{\inf} \frac{F(k)}{k} sin(ks)dk = 1 + \frac{2}{\pi}\int_0^{\inf} \frac{G(k)}{k} cos(ks)dk \tag{3.18}$$

Where $F(k)$ and $G(k)$ are the real and imaginary parts of the Theodorsen function. For practical uses the Wagner function is given by:

$$\phi(s) \approx 1 - 0.165e^{-0.0455s} - 0.335e^{-0.3s} \tag{3.19}$$

When plotted against the reduced time the Wagner function is shown in Figure 3.9

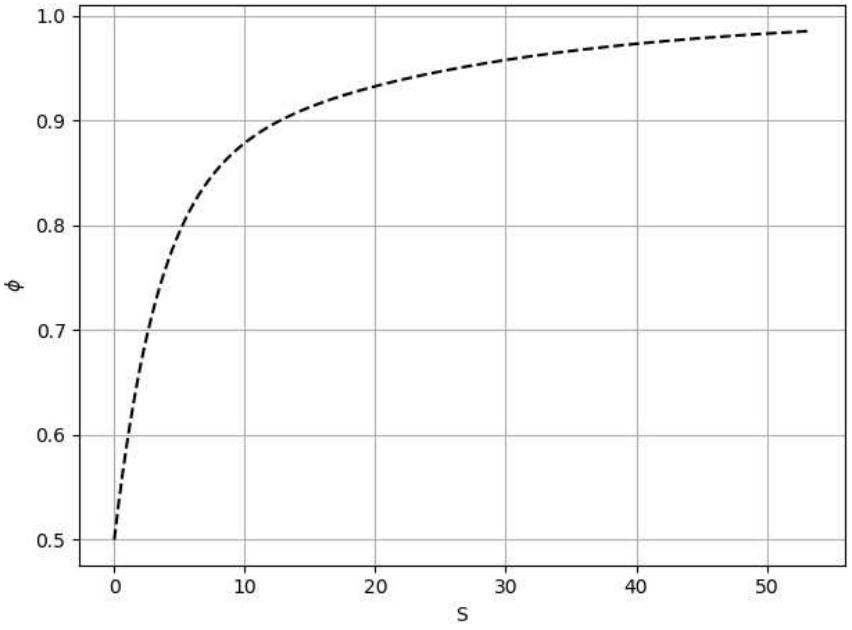
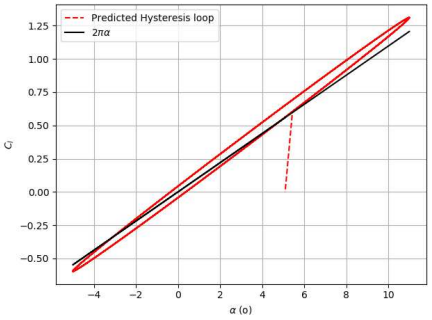


FIGURE 3.9: Wagner function

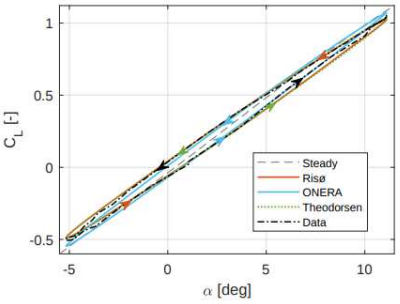
From 3.9 it can be seen that at $t = 0$ the Wagner function is 0.5. This means that for a rapid change in angle of attack the hydrodynamic forces will be half that of the steady state value.

The unsteady lift model from 3.17 was implemented in Python. To verify the model a foil was given an arbitrary motion and validated against models created in literature.

The model chosen was a foil operating at Reynolds number of 1.48 million, oscillating at a reduced frequency of 0.051 with an angle of attack of $3^o \pm 8^o$. The hysteric loop is compared to models given by Faber (2018) shown in Figure 3.10.



(A) Pure pitch oscillation from 3.17



(a) C_L vs α

(B) Pure pitch of foil from Faber (2018)

FIGURE 3.10: Hysteresis loop of a pitching foil using equation 3.17

This model has been included into the BEMT code to account for the blade motions computed by the structural code. This is done by replacing the 2π portion of equation 3.17 by the lift curve slope obtained from the 2D section data. The 2D section data is modified by equation 3.17 to give a new Cl vs α curve. The angle of attack is obtained through the blade element iterations based on the modified Cl vs α curve

Other more sophisticated models for unsteady aerodynamics exist such as the Leishman (1988), Oye (1991) and Snel (1997) models. These models are more complex than the implemented model and can account for flow separation and deep dynamic stall. However, the angle of attack rarely reaches regions of very high angle of attack and is therefore not within the deep stall region. Although for particular flow regimes this could be the case and should be considered for future developments. The changes in moment coefficient follow that of the lift coefficient. The drag coefficient is far more dependent on separation so the static value is considered satisfactory for this study.

3.6 Verification

To ensure the BEMT is calculating the force coefficients correctly the Hyundai Maritime Research Institute (HMRI) propeller geometry has been calculated and

compared with open water experimental results. The propeller geometry can be seen in Table 3.3.

TABLE 3.3: HMRI Propeller Geometry

r/R	P/D	Rake	Skew(o)	C/D	fo/C	to/D
0.155	0.576543	0.0000	-2.525	0.151495	0.031264	0.046806
0.250	0.613000	0.0000	-4.000	0.177166	0.034908	0.042166
0.300	0.631000	0.0000	-4.400	0.189166	0.035621	0.038516
0.400	0.663000	0.0000	-4.400	0.209334	0.033798	0.032016
0.500	0.691500	0.0000	-3.150	0.224666	0.029334	0.026016
0.600	0.712000	0.0000	-0.820	0.233500	0.024984	0.020550
0.700	0.721200	0.0000	2.490	0.233834	0.021895	0.015600
0.800	0.716000	0.0000	6.350	0.219166	0.019797	0.011050
0.900	0.692700	0.0000	10.760	0.180834	0.016147	0.007000
0.950	0.674800	0.0000	13.150	0.142166	0.012805	0.004717
1.000	0.651000	0.0000	16.750	0.000016	3.062500	0.003200

The diameter of the propeller is 0.212m which corresponds to a scaling factor of 1:46.72 compared to full scale. The propeller operating conditions are shown in Table 3.4.

TABLE 3.4: HMRI Propeller Conditions

Parameter	Value
Z	4
BAR	0.431
D	0.212m
Va	1m/s
n	90.9 ,22.72,15.15,9.08,6.5,5.67 rps
Section Profile	NACA66 a=0.8 mean line camber

Where Z is the number of blades, BAR is the blade area ratio, D is the diameter, Va is the inlet velocity for the open water simulations and n is the rotation rate of the propeller.

The open water experiment was performed as part of the SIMMAN 2013 ship manoeuvring workshop performed by HMRI.

Figure 3.11 shows the open water performance curve of the propeller when compared to the experimental values. The efficiency of of the BEMT solution compares very well with that of the experiment. The thrust coefficient matches reasonably closely to the experiment with slight discrepancies occurring at J = 0.4 - 0.6 however, more significant errors occur for the torque coefficient in this range of J.

The thrust coefficient is predominantly dictated by the lift coefficient and the torque coefficient is dominated by the drag coefficient. The result shown in Figure 3.11 show that the use of the 2D foil section data gives good approximations for lift but overestimates the drag which is a know issue for CFD.

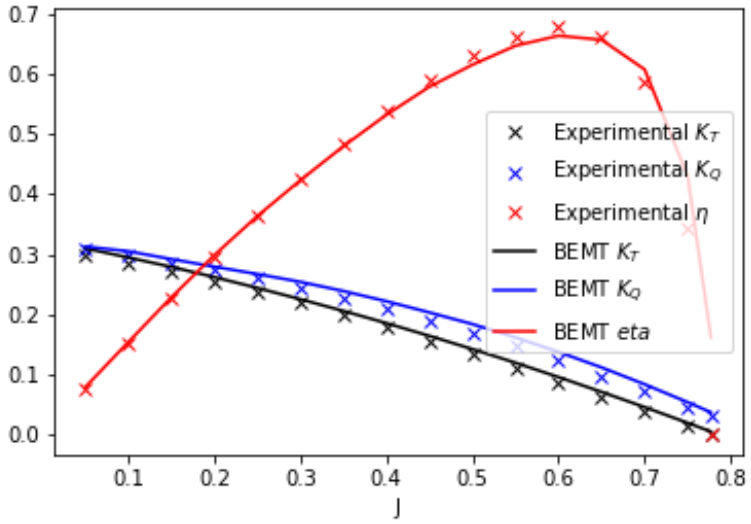


FIGURE 3.11: Performance Curves from BEMT compared with Experimental values.

3.7 Chapter Summary

Blade element momentum theory has been used to provide hydrodynamic loads on the propeller blades. This method has been improved by using Reynolds number dependent 2D section data to be able to match the performance curves of the propeller well. Although the NACA66 mod section profile has been used for the data base this can be altered to include other foil sections and the database extended.

The use of a C based Python compiler called Cython has been used to greatly increase performance of the BEMT solution which will be required to run at every time step for the FSI algorithm. The use of Cython has been shown to give an $\approx$ 80x speed up compared to the pure Python version.

Finally, an attempt to capture the unsteady motion of the propeller blade using Theodorsen aerodynamic functions have been utilised and implemented within the BEMT code. This alters the lift curve slope of the 2D section based on its pitching and heaving motions.

Chapter 4

High Fidelity Propeller Simulations

4.1 Introduction

To ensure the accuracy of the computationally efficient methods high fidelity simulation methods have been developed.

This chapter describes the methods used for high fidelity computational fluid dynamics to obtain propeller performance in open water. The blade element momentum theory can be compared against the high fidelity solution. Both steady state simulations and unsteady simulations are discussed in this chapter.

The use of high fidelity simulations can give a great deal of information with regards to the flow features and forces. The focus on this chapter is to obtain force distribution on the propeller blades and to compare the high fidelity methods to blade element momentum theory. This is to determine if the use of Blade Element Momentum Theory is of sufficient accuracy to use as a model for the propeller forces.

4.2 High Fidelity simulations of HMRI propeller

CFD was used to obtain the propeller performance of the HMRI Propeller. The open water performance was obtained using a fixed inlet velocity and changing the rotational velocity to change the advance ratio.

4.2.1 Meshing Strategy

The point of interest is for the force loading on the propeller blade. Therefore, the mesh will be changed by the cell count on the blades. This will ensure the geometry is captured correctly and the distribution of pressure is accurate.

The meshes were generated using snappyHexMesh, Table 4.1. The propeller blades used a wall boundary condition and the outer domain used a symmetry boundary condition to avoid any wall blockage effects.

TABLE 4.1: Steady HMRI open water mesh definitions

Mesh	Cell Count
Coarse	2.2 million
Medium	2.8 million
Fine	5.4 million

The domain can be seen in Figure 4.1 with cross sections shown in Figure 4.2. The inlet is located 6 propeller diameters upstream from the propeller and the diameter of the domain is 3 propeller diameters.

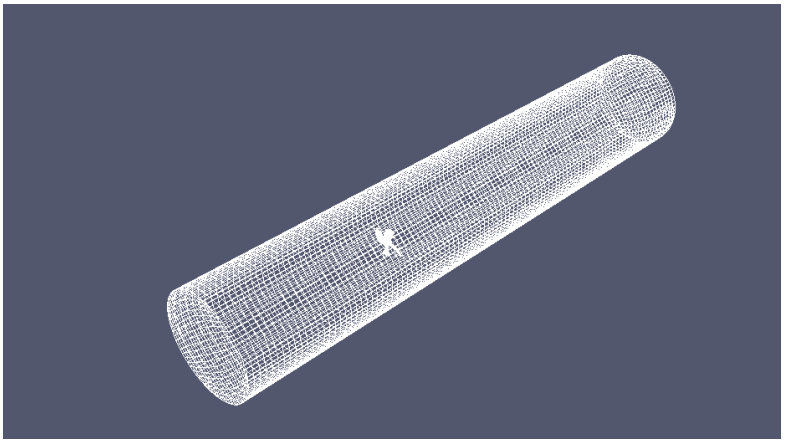


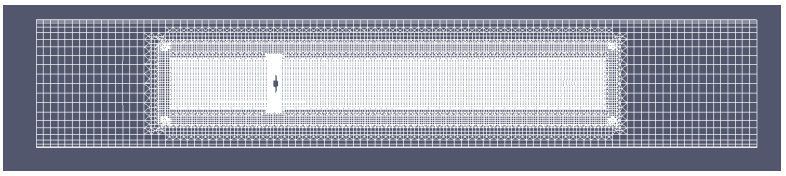
FIGURE 4.1: CFD domain for HMRI propeller.

From Figure 4.2 it is seen mesh refinements occur upstream and downstream of the propeller to ensure good capture of any shed features which may have an impact on the blade loading.

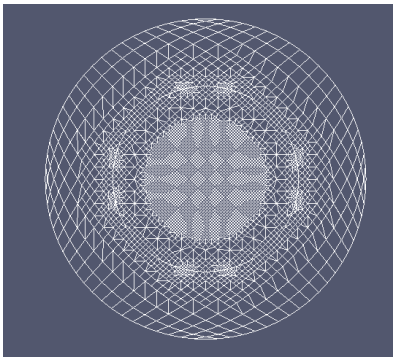
To model the rotation of the propeller two options are available in OpenFOAM: Multiple Reference Frame (MRF) or Arbitrary Mesh Interface (AMI). Initially the MRF model is used for the steady state simulation as it is significantly computationally cheaper.

The MRF zone is selected using OpenFOAMs topoSet functionality where cell zones can be created. Within the MRF zone the governing equations change to that shown in equation 4.1 from Greenshields (2016).

$$\nabla \cdot (u_R u_R) + 2\Omega \times u_R + \Omega \times (\Omega \times r) = -\nabla p + (\nu_{eff}(\nabla u_R + (\nabla u_R)^T)) \tag{4.1}$$



(A) Side View of HMRI Propeller fine mesh



(B) Face View of HMRI Propeller fine mesh

FIGURE 4.2: Slice view of HMRI domain slices.

Where Ω is the rotational vector and r is the positional vector from the centre of rotation to the cell and the rotational vector is defined by the user. The Coriolis force is defined by $2\Omega \times u_R$ and the centrifugal force is defined by $\Omega \times (\Omega \times r)$ in equation 4.1.

Table 4.2 gives an overview of the HMRI open water simulation.

TABLE 4.2: Steady HMRI open water mesh definitions

CFD software	OpenFOAM v1712
Turbulence Model	$K - \omega - SST$
Mesh type	Unstructured
Cores	48
Reynolds Number at 0.7r	1.5e5- 2e6

4.2.2 Results

The conditions tested are shown with the thrust and moment conditions are shown in Table 4.3. The inlet velocity is kept fixed and the advance ratio is altered by changing the rotational velocity. A range of advance ratios are tested to account for the range tested in experimentation.

TABLE 4.3: HMRI inlet and rotation conditions

inlet velocity (m/s)	Rotational velocity (rps)	J
1	22.72	0.2
1	15.15	0.3
1	9.08	0.5
1	6.5	0.7

The thrust, torque and efficiency curves are compared to the experimental results as shown in figure 4.3. The performance curves for the coarse, medium and fine meshes are presented. The CFD results compare reasonably well for thrust and torque coefficient however the efficiency is not well captured. Although the fine mesh tends to match more closely to the experimental values over the coarse and medium mesh. There are two primary reasons for this. The CFD simulation can overestimate the drag impact therefore increasing the torque coefficient hence efficiency is in error despite the thrust prediction performing well. In addition, there is little detail in the experimental technique to obtain the range of advance ratios. This means that the change of advance ratio can be due to either advance speed or rotation rate. The impact will be a large change in Reynolds number which can have a significant impact on efficiency as shown in Rijpkema (2015).

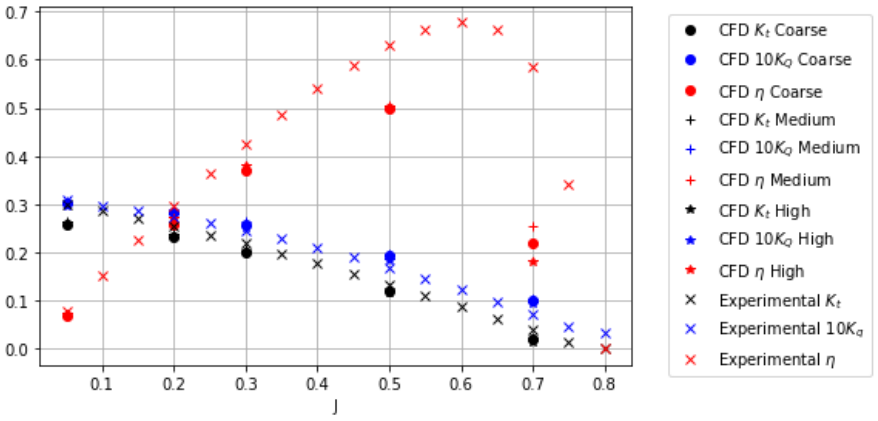


FIGURE 4.3: Performance curves for HMRI propeller.

4.2.2.1 Load Distribution Comparison with BEMT

Although the BEMT code gives good comparison for overall thrust and torque coefficients confidence must be gained that the load distribution along and across the blade is correct. This is done by comparing the BEMT solution to high fidelity CFD simulations. A reference blade is radially sectioned at the same positions as the BEMT input geometry file such that OpenFOAM can calculate the force at each blade section. The geometry is split altering the Standard Triangle Language (STL) file. An STL file

defines the geometry by building up a series of triangles which form the larger structure. The centre of each triangle in the STL is found, each point is translated to polar coordinates and is sorted into the radial portion and angular positions. Each triangle can then be flagged of belonging to a certain radial and angular section. The split blade geometry in the CFD simulation is shown in Figure 4.4. Slight discrepancies can be seen in splitting the blade into sections however, this should average out over the entire blade surface.

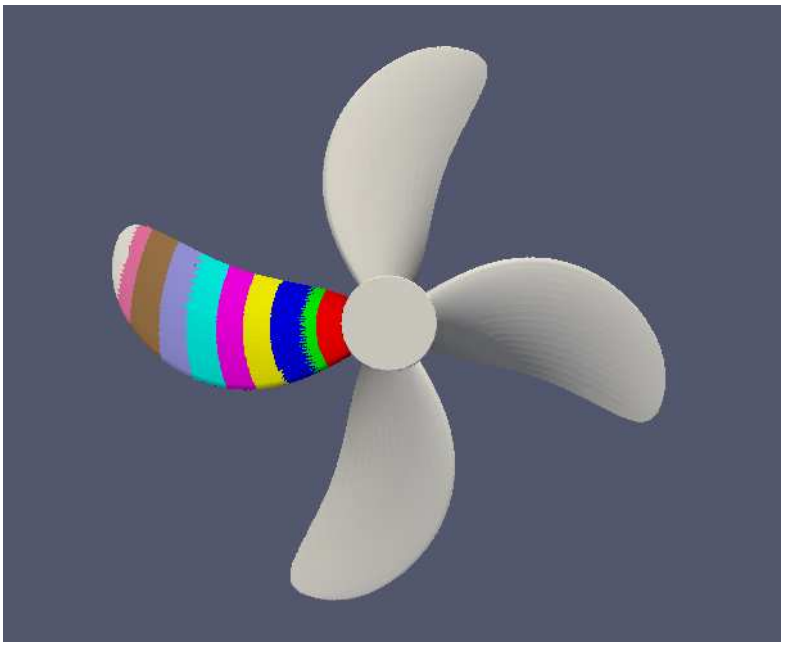


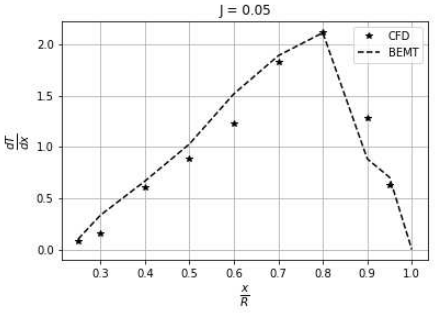
FIGURE 4.4: HMRI with blade sectioned

The load distribution of the blade compared with the CFD load distribution is shown in Figure 4.5. As the BEMT code outputs the thrust coefficient curve as opposed to thrust, the coefficient is changed to thrust using equation 4.2.

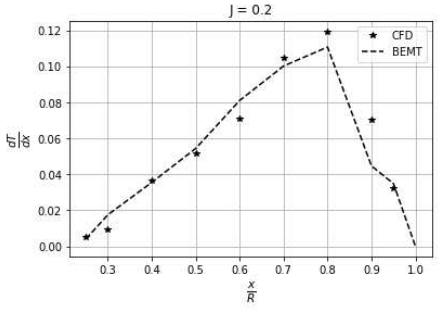
$$\frac{dT_{BEMT}}{dx_{BEMT}} = \rho n^2 D^4 \frac{dK_T}{dx} dx d\theta \tag{4.2}$$

Here ρ is the density of the fluid, D is the diameter of the propeller, dx is the length of the element in the radial direction and $d\theta$ is the angular portion i.e2π / the number of circumferential sections. dK_T is the local thrust coefficient of that section.

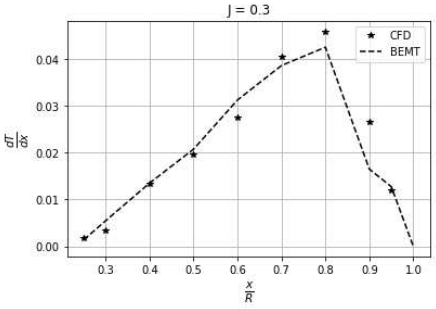
To accurately compare the load distributions the CFD thrust distribution is normalised by the radial length to obtain $\frac{dT_{CFD}}{dx_{CFD}}$.



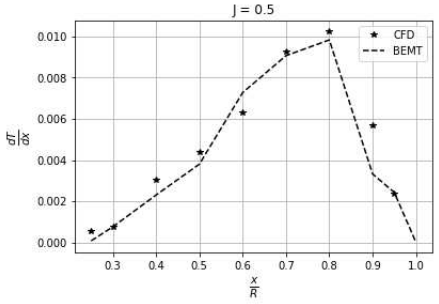
(A) Thrust distribution of HMRI propeller at J = 0.05



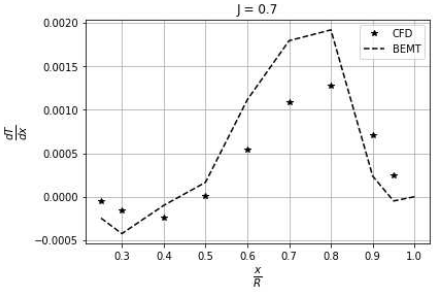
(B) Thrust distribution of HMRI propeller at J = 0.2



(C) Thrust distribution of HMRI propeller at J = 0.3



(D) Thrust distribution of HMRI propeller at J = 0.5



(E) Thrust distribution of HMRI propeller at J = 0.7

FIGURE 4.5: Load distribution for various advance ratios

It can be seen that the load distributions match well except at J = 0.7 . The position of maximum load occurs at the same position and the gradient of the curve slope matches well. However, for J = 0.7 the upper load contribution towards the tip is well over estimated although the initial negative thrust at the root of the blade is captured. There is also a small kink in the load distribution towards the tip. This is due to the resolution of points and can be removed by increasing the number of points in the x/R plane.

It is therefore shown that the use of BEMT gives a reasonable estimate of load distribution along the propeller blade.

4.3 Unsteady Propeller performance

4.3.1 Introduction

The use of high fidelity propeller modelling using a steady MRF region has been discussed however, this cannot be used when the propeller is performing in a non-uniform flow field. The load distribution on the blade will change as the propeller rotates and experiences different flow regimes.

Therefore, an accurate and reliable method for obtaining the blade loading in an unsteady state is required. This section discusses the development of the method and compares the performance curves between the experimental and unsteady simulations.

4.3.2 Meshing Strategy

To calculate the propeller performance in an unsteady flow regime a cyclic AMI is used in OpenFoam. This is a dynamic mesh calculation where the the propeller is spinning at the user defined rate to simulate the rotation of the propeller. The use of a dynamic mesh will require an interface between the stationary portion of the domain and the rotating portion. The addition of the interface and mesh movements means the increase in computational cost for an unsteady, dynamic mesh simulation is significantly higher when compared to a steady state MRF simulation.

To obtain a good quality mesh, the interface between the rotating portion and stationary portion of the mesh is vital. This is because if the interface is poor, gaps can appear between the rotating region and stationary region which OpenFOAM cannot deal with causing simulations to crash.

Therefore, to ensure a good quality interface is kept throughout the simulation Pointwise is used as the meshing tool as snappyHexMesh is poor at generating a good interface between the rotating region and stationary region. When using snappyHexMesh the initial interface can be of good quality however, as the rotation occurs the communication between cells can change and eventually become zero. This is a well documented problem when using snappyHexMesh as a mesher.

This is done by creating two regions in Pointwise. The first region is the rotating region which contains the Propeller as shown in Figure 4.6

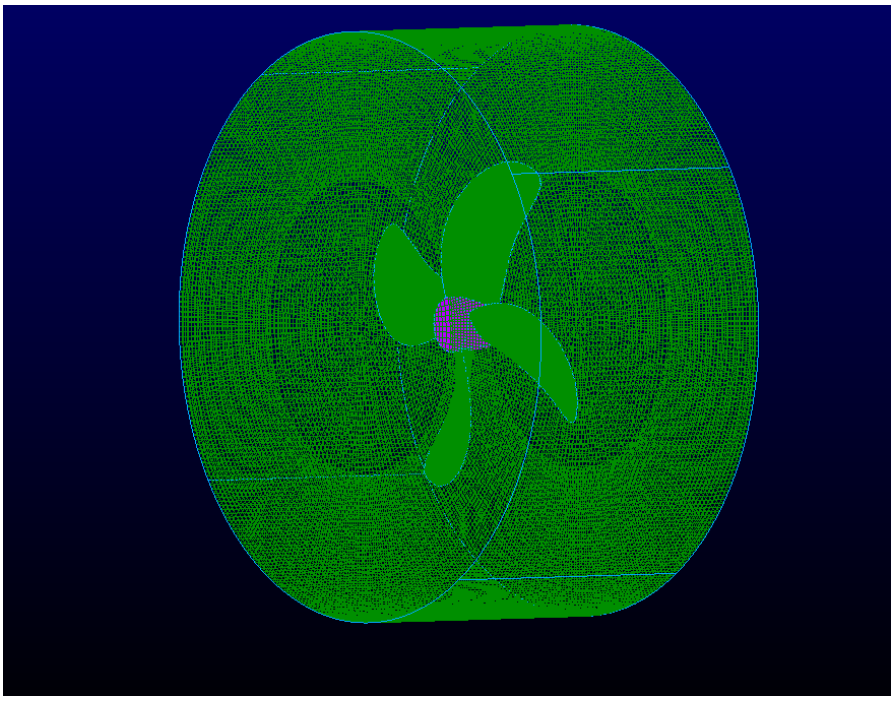


FIGURE 4.6: Rotating region using Pointwise meshing.

The outer, stationary region is created with the rotating region boundaries to create the domain as shown in Figure 4.7. The boundary conditions for the rotating region are set to wall for the propeller and the interface regions are defined as AMI patches. In the Pointwise software the connectors are selected and defined for each mesh region to ensure the the boundary conditions are two-sided and can share data.

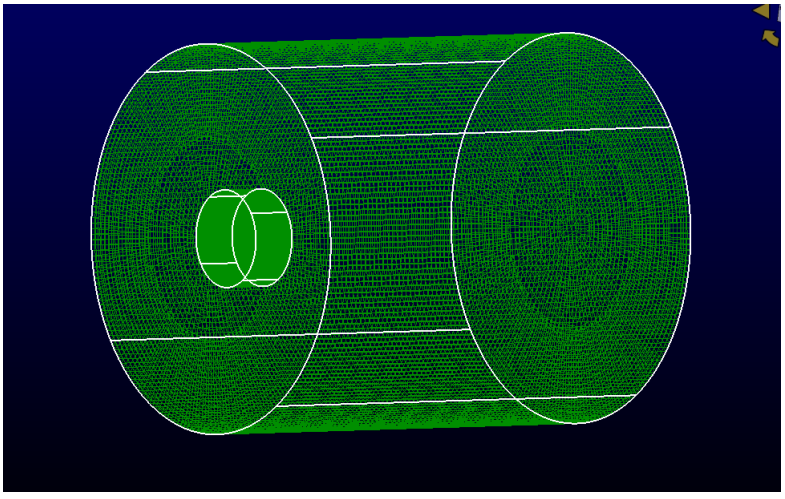


FIGURE 4.7: Domain using Pointwise meshing.

The outer stationary domain is then created and the two regions are merged using the mergeMesh utility in OpenFOAM. When using Pointwise as a meshing tool it is vital

to renumber the mesh before running any simulations as OpenFOAM crashes otherwise.

The boundary and initial conditions are shown in Table 4.4.

TABLE 4.4: Unsteady Propeller simulation boundary conditions

Patch	U	p
Propeller	movingWallVelocity	zeroGradient
inlet	fixed Value	zeroGradient
outlet	zeroGradient	fixedValue
domain	symmetry	-
AMIRotor/Stator	cyclicAMI	-

4.3.3 Results

4.3.3.1 Performance Curve

The unsteady simulation is run using the pimpleFoam algorithm, which is the transient version of the SIMPLE algorithm. The pimple algorithm effectively uses the SIMPLE algorithm to find a converged solution at each timestep. It is a stable solution for a larger timestep. This simulation is complete using 48 cores on the iridis5 cluster. These are completed for the same range of operating advance ratios as the MRF simulations as a comparison using the same range of Reynolds number.

The force coefficients for advance ratios J = 0.05 , J = 0.2, J = 0.3, 0.5 and 0.7 are shown in Figure 4.8.

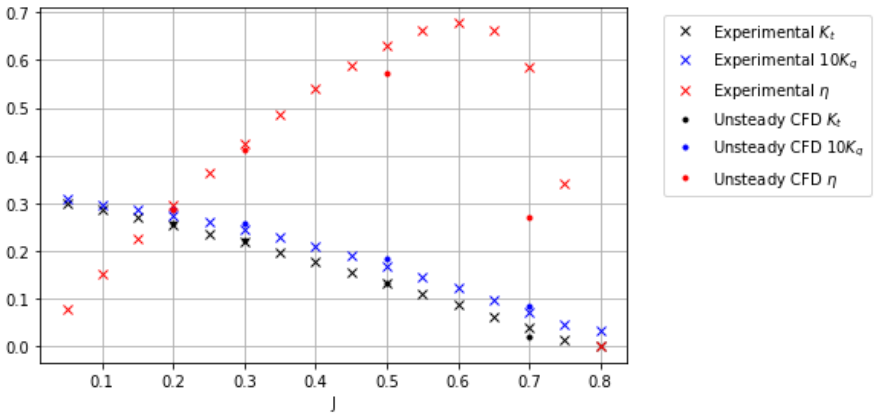


FIGURE 4.8: Performance curves for Unsteady Propeller

It is shown in Figure 4.8 that the unsteady performance curves match the experimental performance well except at J = 0.7. The thrust coefficient in particular matches closely to that of the experiment.

4.3.4 Computational Cost

The steady and unsteady simulations were complete on the Iridis 5 supercomputer at the University of Southampton. Each simulation was complete on 48 2.0GHz processors. The comparison between computation time per time step is shown in Table 4.5

TABLE 4.5: Computational Cost of High Fidelity CFD propeller simulations

Simulation	Execution time per time step
MRF	≈0.5s
AMI	≈100s

The AMI simulation is significantly more expensive compared to the MRF simulation. This is due to the significant cost of manipulating the mesh at each timestep. This further demonstrates the use of BEMT as a computationally effective way of calculating the blade forces as it is a fraction of the computational cost using a single core compared to using multiple cores for a CFD simulation.

4.4 Chapter Summary

In this chapter the high fidelity computational techniques are compared to BEMT. It is shown that the overall performance curves achieved by the blade element momentum solution compares very well with the experimental values as do the high fidelity techniques.

Moreover, the force distribution along the blade obtained from the high fidelity solution compare well to that of the blade element solution. This is useful as the load on the blade can be modelled to a reasonably good accuracy compared to high fidelity simulations at a fraction of the computational time. Each open water CFD simulation takes several hours over several cores to complete whereas the BEMT solution takes less than a second.

Despite the advantages of BEMT over CFD in terms of computational cost, CFD does give a more detailed flow features which BEMT cannot account for such as blade tip vortex shedding.

The good agreement between the force distribution of the blade when comparing CFD and blade element momentum theory gives good confidence that BEMT can be used as a hydrodynamic model for the propeller blade loading.

Chapter 5

Unsteady Wake Generation

5.1 Introduction

The open water characteristics of the propeller have been captured and the BEMT solution has shown to agree with experimental values. However, the propeller operates behind the hull. The presence of the hull causes a non-uniform and unsteady inflow to the propeller.

This wake inflow to the propeller will cause it to experience different loading and angles of attack as the propeller rotates and as any flow features interact with the propeller at the propeller plane. The unsteady inflow can match closely to the natural frequency of the blade and cause severe vibration and deformation. This will have a large impact on the noise and fatigue of the propeller. Moreover, the change of drift angle can cause large velocity gradients at the propeller plane therefore causing rapid loading and unloading.

To obtain the non-uniform and unsteady velocities at the propeller plane computational fluid dynamics have been utilised to model the KVLCC2 hull form. This hull form has been chosen due to the abundance of validation data from the SIMMAN conferences Van Hoydonck et al. (2014).

The hull form is rotated to several drift angles which match those studied for the SIMMAN 2019 conference. This is to obtain a series of wakes with different flow fields and velocity gradients. The different flow fields will cause the propeller blade to respond in different ways which will be studied in Chapter 7

5.2 Meshing strategy

The focus of the simulations is to obtain a wake at the propeller plane therefore different mesh refinements at the propeller plane were studied.

The mesh is generated using the commercial software Pointwise. This was chosen as it was found that using the meshing software provided by snappyHexMesh provided poor cell quality in the boundary layer. Good quality boundary layer cells were found to be highly important when capturing key features such as the bilge vortices.

The CFD domain is shown in Figure 5.1. With boundary conditions shown in Table 5.1

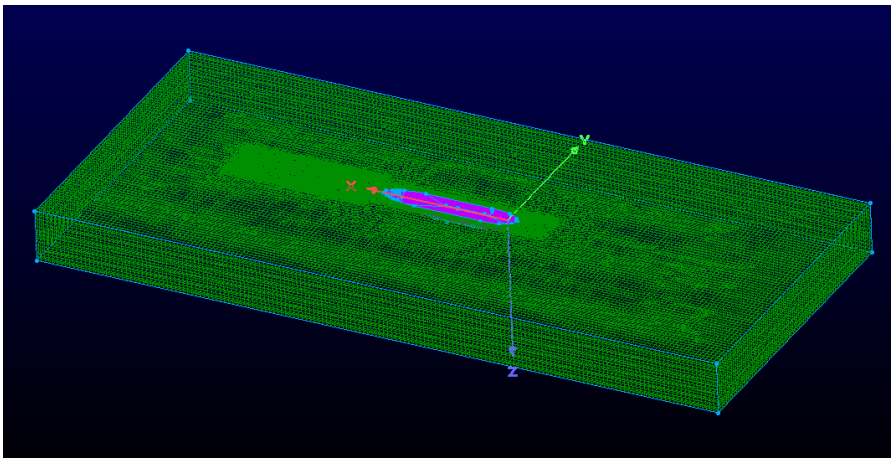


FIGURE 5.1: CFD domain used to model KVLCC2 hull form

TABLE 5.1: Boundary conditions

Inlet	fixed value
KVLCC2 walls	noSlip
sides, top and bottom	symmetry plane

In this study, the KVLCC2 is scaled to the model scale described in Van Hoydonck et al. (2014) with particulars shown in Table 5.2. In this CFD study the hull form is without any appendages, propeller or rudder.

TABLE 5.2: KVLCC2 Geometry and Operating Conditions

Ship Speed V_s (m/s)	1.1702
Froude Number F_R	0.142
L_{pp} (m)	6.893
L_{wl} (m)	7.0112
B_{wl} (m)	1.249
Scale	46.426

Where the Froude number is defined by equation 5.1.

$$F_R = \frac{V_s}{\sqrt{gL_{wl}}} \tag{5.1}$$

Where g is the gravitational constant, L_{wl} is the length of the hull at the water line, L_{pp} is the length between particulars and B_{wl} is the beam at the water line.

The symmetry plane at the top was chosen to act as a 'double body' simulation. This is acceptable due to the little impact the free surface has on the motion of the ship and the velocities at the propeller plane. This saves a great deal in complexity and computational expense as it has been shown that free surface simulations take 1000% longer than double body simulations Turnock et al. (2008)

An unstructured grid has been generated with good quality T-Rex layers. T-Rex meshing is anisotropic tetrahedral extrusion where a hybrid between structured and unstructured meshing occurs. The mesh is structured within the layers at the wall hull up to the user defined thickness. Then the domain is filled using a Delaunay-based mesher. A side view of the mesh is shown in Figure 5.2 with the rear of the geometry shown in Figure 5.3. It can be seen that the Pointwise mesher creates high quality isotropic layers around the geometry.

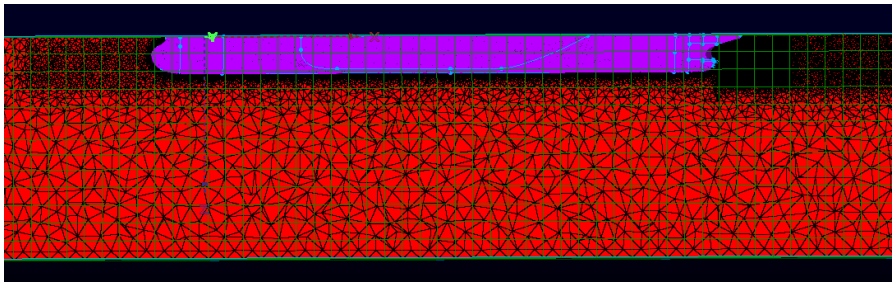


FIGURE 5.2: KVLCC2 Mesh

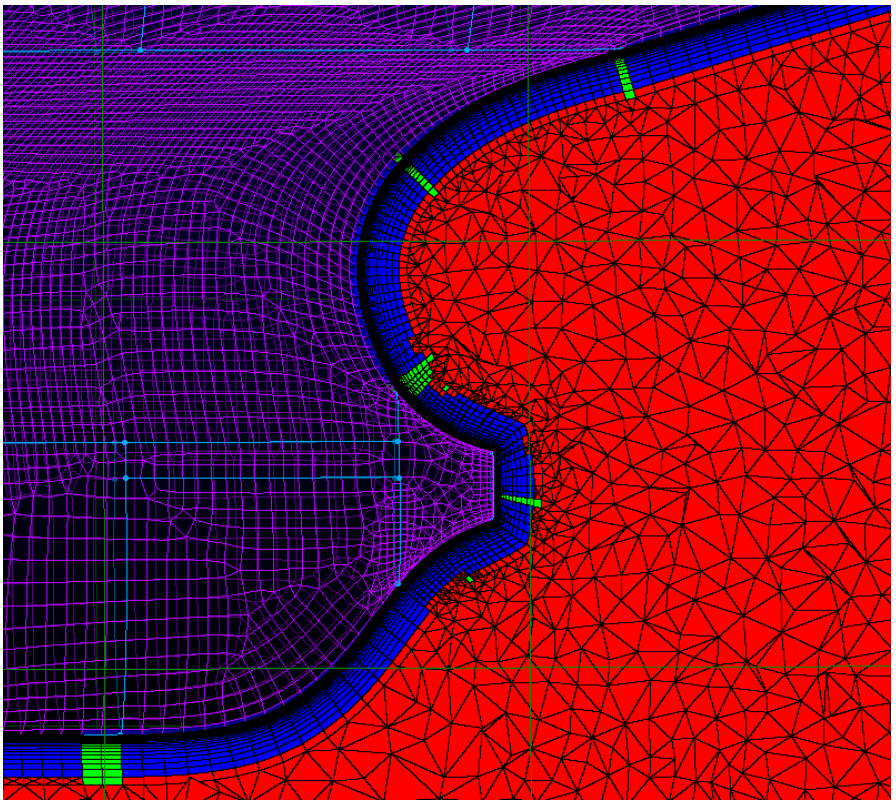


FIGURE 5.3: The rear of KVLCC2 with layers

Three meshes are developed to ensure the velocities at the propeller plane are converged. These are compared to experimental results given in detail by Larsson and Stern (2014). The mesh changes by increasing the mesh refinement at the propeller plane.

A summary of the meshes is shown in Table 5.3.

TABLE 5.3: KVLCC2 mesh summary

Mesh name	Number of cells
G1	6.8 million
G2	10 million
G3	11 million

5.2.1 Results

The CFD simulation was run using OpenFoam v1906 using the simpleFoam algorithm to obtain a steady state solution. This is to compare with previous simulations and experimentation of the KVLCC2 hull form at the propeller plane.

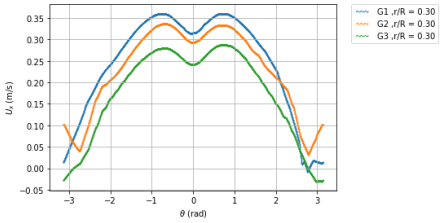
TABLE 5.4: Solver Conditions

CFD code	OpenFoam v1906
Turbulence model	$k-\omega$-SST
ddt Scheme	steady state
algorithm	simpleFoam

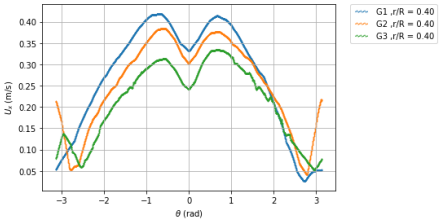
The velocities at the propeller plane for each mesh is shown in Figure 5.4. This Figure shows the axial velocities at the propeller plane against the angular position where the point where $\theta = 0$ is top dead centre of the propeller plane.

The mesh dependence is the most significant at the near hub region. This is because this is where the velocity gradients are more extreme, compared to the change outer regions where the flow speed is closer to that of free stream.

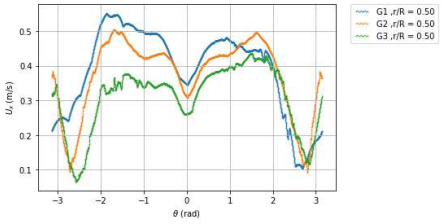
As the hull is operating in the straight ahead configuration the steady state wake should be symmetric. This is not the case for grid G3 as clear asymmetry is observed at the $\frac{r}{R} = 0.4 - 0.6$ region. The reason for this asymmetry is due to potentially poor quality cells in the boundary layer as the aspect ratio would be increased when refinement is achieved.



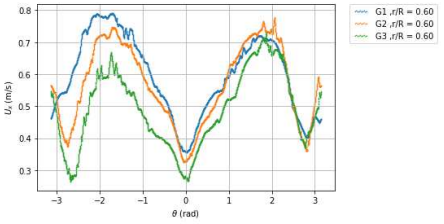
(A) Axial velocity at radial position $\frac{r}{R} = 0.3$



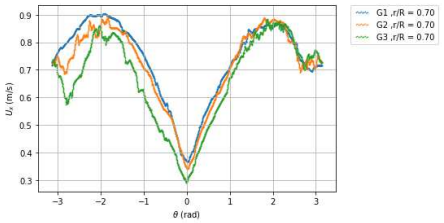
(B) Axial velocity at radial position $\frac{r}{R} = 0.4$



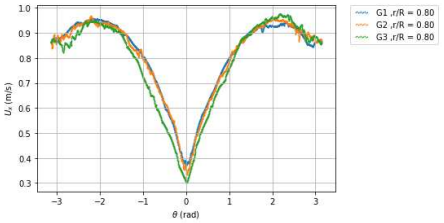
(C) Axial velocity at radial position $\frac{r}{R} = 0.5$



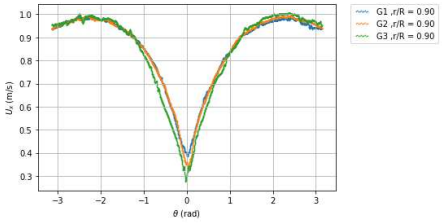
(D) Axial velocity at radial position $\frac{r}{R} = 0.6$



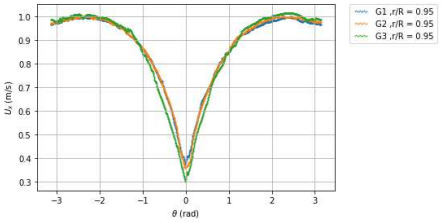
(E) Axial velocity at radial position $\frac{r}{R} = 0.7$



(F) Axial velocity at radial position $\frac{r}{R} = 0.8$



(G) Axial velocity at radial position $\frac{r}{R} = 0.9$



(H) Axial velocity at radial position $\frac{r}{R} = 0.95$

FIGURE 5.4: Axial velocity at various radial positions at the propeller plane for three meshes where R is the propeller radius

It is also important to note that the maximum Reynolds stress at the propeller plane matches well for grid G2 when compared to literature Larsson and Stern (2014). The maximum Reynolds stress value at the propeller plane is shown in Table 5.5.

TABLE 5.5: Reynolds Stress components at propeller plane behind KVLCC2

Reynolds Stress	Value EFD Lee and Kim (2003)	CFD
$max(\bar{u}\bar{u})$	0.016	0.01625
$max(\bar{v}\bar{v})$	0.008	0.0082
$max(\bar{w}\bar{w})$	0.002	0.0018
$max(k)$	0.022	0.027

For these reasons Grid G2 has been deemed suitable to continue with the numeric experiments.

5.3 Drift Angles

The flow field at the propeller plane has been determined for various drift angles. The drift angles chosen coincide with the angles requested by the SIMMAN 2020 conference which are -2^o , 0^o, 2^o ,4^o,6^o, 8^o ,12^o and 16^o. The different drift angles will have different flow features at the propeller plane.

5.3.1 Steady Wakes

Using mesh G2 the CFD has been run using the steady state simulation technique as described in Section 5.2.1.

The resulting velocity field at the propeller plane for each drift angle is shown in Figure 5.5. Each plot covers the diameter of the propeller. Here the velocities are normalised by the inlet velocity of 1.1702 m/s. The contours of each plot represent the axial velocity with the arrows representing the radial and tangential velocities where the positive axial velocity is coming out of the page. For the 0^o drift angle shown in Figure 5.5b the velocity profile is mostly symmetric with radial and tangential velocities heading towards the centre. Here the velocity profile is the classic U-shaped profile with lower velocity near the ship hull.

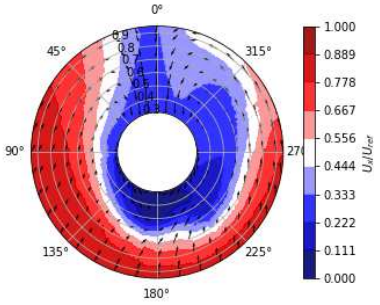
As the drift angle increases the velocity gradients become higher particularly at the 4^o drift angle where the velocity is lower on port side compared to the starboard and there is a secondary region of low velocity fluid at $\frac{r}{R} = 0.4$, $\theta = 90^0$. At the high drift angle cases there is a strong asymmetry in the flow particularly in the radial and tangential velocities.

To ensure flow structures have been captured the Q-criterion Hunt et al. (1988) has been calculated to identify the vortical structures. The Q-criterion is the second invariant of the velocity gradient tensor.

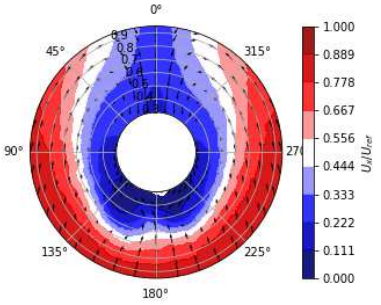
$$Q = \frac{1}{2}[(tr(\nabla u))^2 - tr(\nabla u \cdot \nabla u)] \tag{5.2}$$

Where u is the velocity field and tr is the trace of the matrix. The iso-contour plots, coloured by vorticity, of the flow features around the KVLCC2 at several drift angles are shown in Figure 5.6. It can be seen here that two strong counter rotating vortices occur at the rear of the hull. As the drift angle increases the further flow features are captured as defined by Xing, T., Bhushan, S., and Stern (2012). This is well defined at a

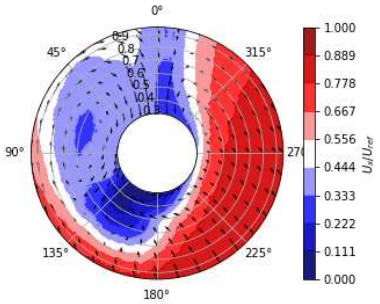
12^o drift angle where a fore-body bilge vortex, a fore-body side vortex and aft body hair pin vortex and an aft-body bilge vortex occur. This is shown to emphasise the correct application of the CFD software and that correct fluid structures have been captured. This can then be used to assess the impact these fluid structures can have on the propeller vibration properties.



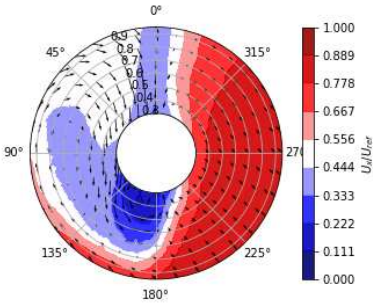
(A) Velocity contour at propeller plane for KVLCC2 at drift angle -2^o



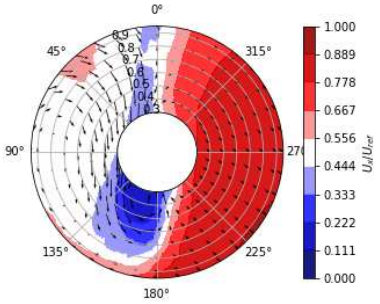
(B) Velocity contour at propeller plane for KVLCC2 at drift angle 0^o



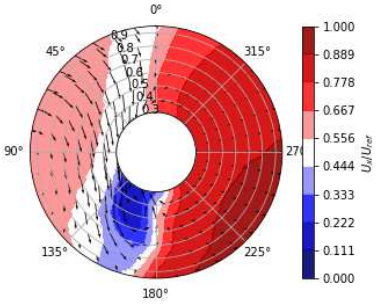
(C) Velocity contour at propeller plane for KVLCC2 at drift angle 4^o



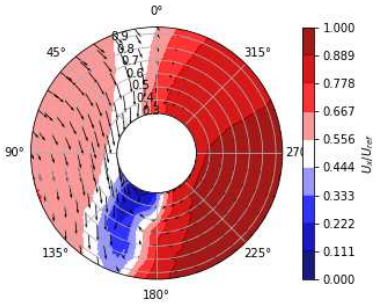
(D) Velocity contour at propeller plane for KVLCC2 at drift angle 6^o



(E) Velocity contour at propeller plane for KVLCC2 at drift angle 8^o

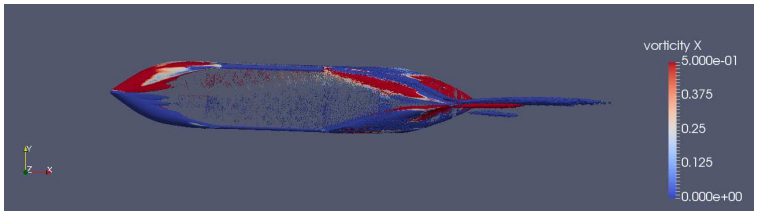


(F) Velocity contour at propeller plane for KVLCC2 at drift angle 12^o

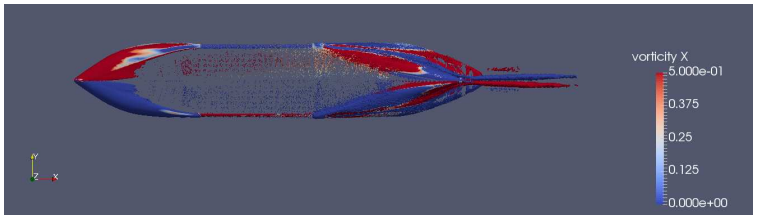


(G) Velocity contour at propeller plane for KVLCC2 at drift angle 16^o

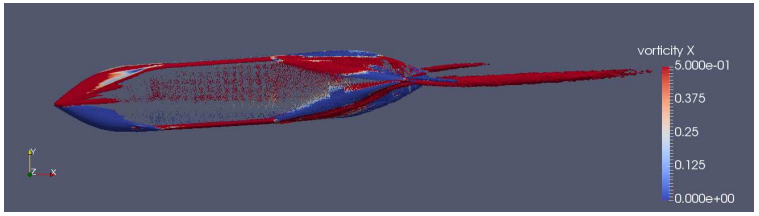
FIGURE 5.5: Velocity contour plots at the propeller plane



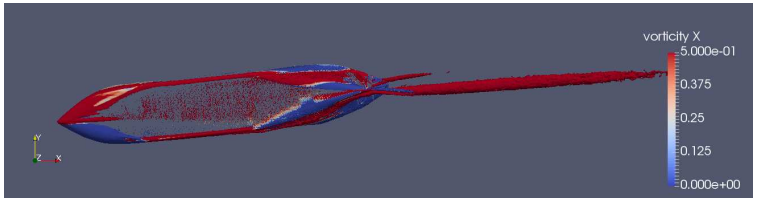
(A) Q criterion for KVLCC2 at drift angle -2^o



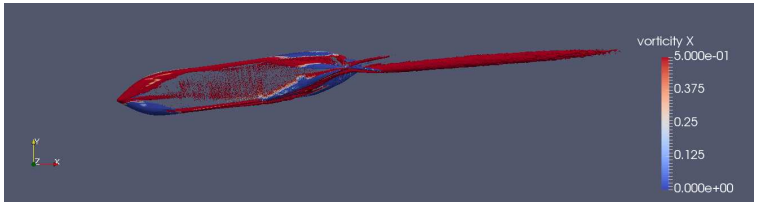
(B) Q criterion for KVLCC2 at drift angle 0^o



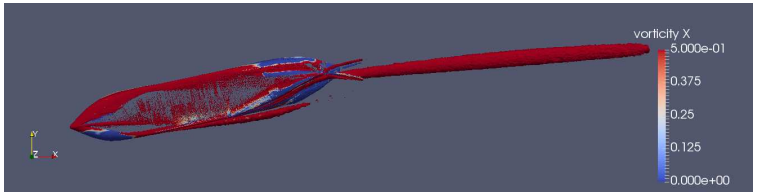
(C) Q criterion for KVLCC2 at drift angle 4^o



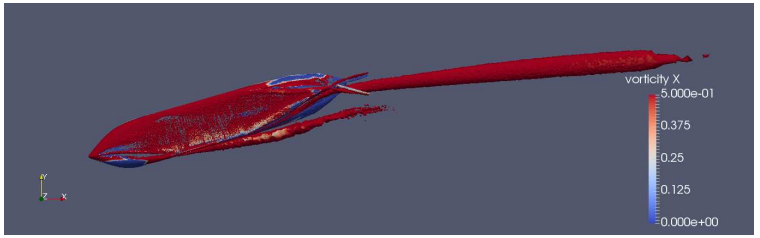
(D) Q criterion for KVLCC2 at drift angle 6^o



(E) Q criterion for KVLCC2 at drift angle 8^o

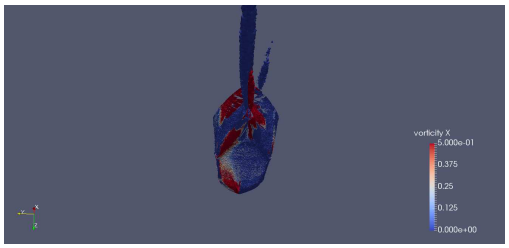


(F) Q criterion for KVLCC2 at drift angle 12^o

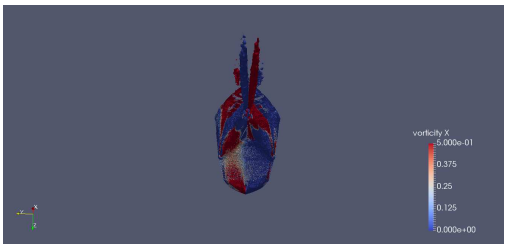


(G) Q criterion for KVLCC2 at drift angle 16^o

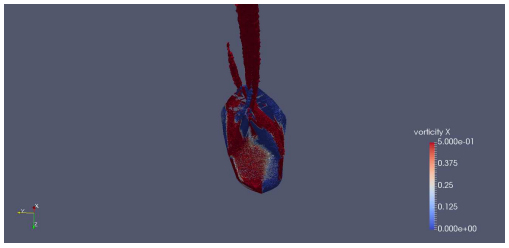
FIGURE 5.6: Q-criterion iso-contour plots for KVLCC2 at several drift angles.



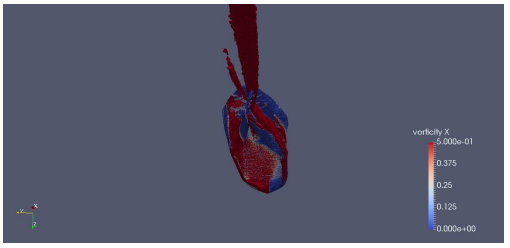
(A) Q criterion rear view for KVLCC2 at drift angle -2^o



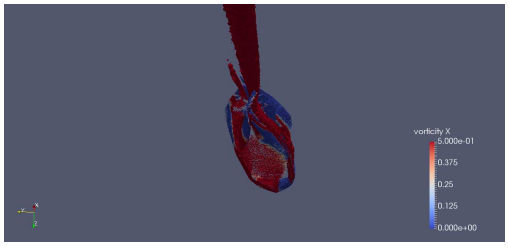
(B) Q criterion rear view for KVLCC2 at drift angle 0^o



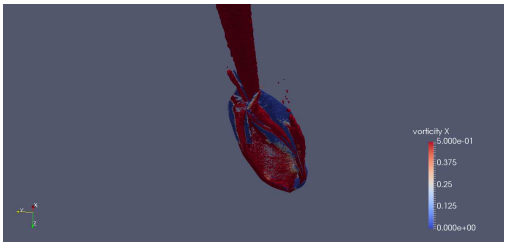
(C) Q criterion rear view for KVLCC2 at drift angle 4^o



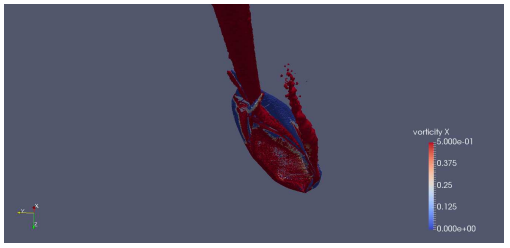
(D) Q criterion rear view for KVLCC2 at drift angle 6^o



(E) Q criterion rear view for KVLCC2 at drift angle 8^o



(F) Q criterion rear view for KVLCC2 at drift angle 12^o



(G) Q criterion rear view for KVLCC2 at drift angle 16^o

FIGURE 5.7: Q-criterion iso-contour plots for KVLCC2 at several drift angles rear view.

5.3.2 Unsteady Wakes

The wake of the ship hull is inherently unsteady causing a variation in the velocity field at the propeller plane with time. This results in time varying loading on the propeller which can result in additional excitation of the propeller blade modes.

To obtain an unsteady solution of the velocity field unsteady Reynolds averaged Navier Stokes (URANS) has been used. The 1st Order Euler scheme has been used for the first and second time derivatives. This has the advantage of being bounded so a

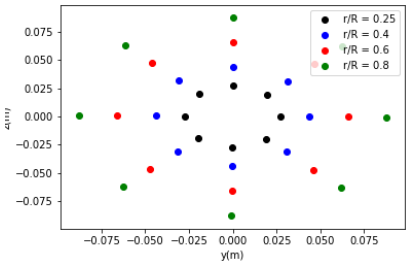
solution is more likely to be found, however the convergence to a solution is not as quick as a 2nd order backward solution.

To achieve the unsteady solution the steady state solution has been used as an initial condition. The time step is fixed to 5e-3 seconds to ensure the Courant number remains below 1.

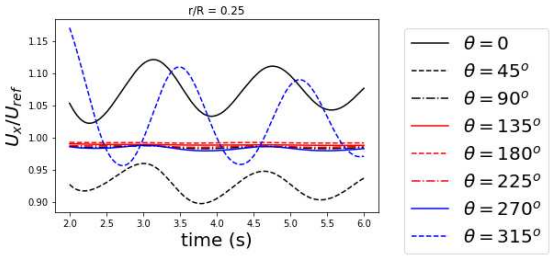
TABLE 5.6: Solver Conditions

CFD code	OpenFoam v1906
Turbulence model	$k-\omega$-SST
ddt Scheme	Euler
algorithm	pimpleFoam
Time Step (s)	0.0005s
Mesh	G2

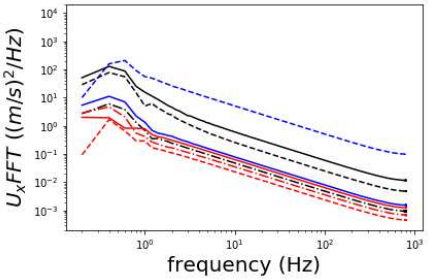
For illustrative purposes the plots shown in Figures 5.8b to 5.13h are taken at every 2nd radial position and every 45^o as shown in Figure A.7a. The velocities have been normalised by the initial velocity, this is to be able to compare the changes in velocity with respect to time for each point. The axial velocities are shown here however the radial and tangential velocities are available in the appendix.



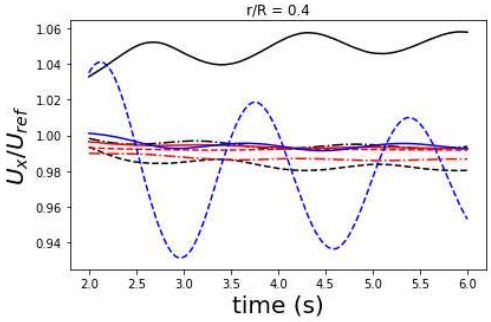
(A) Plotting positions for unsteady wake



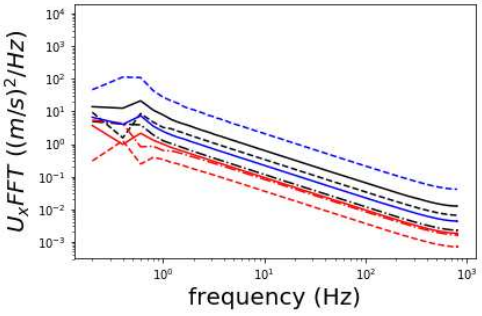
(B) r/R = 0.25 unsteady velocity for drift angle = 0^o



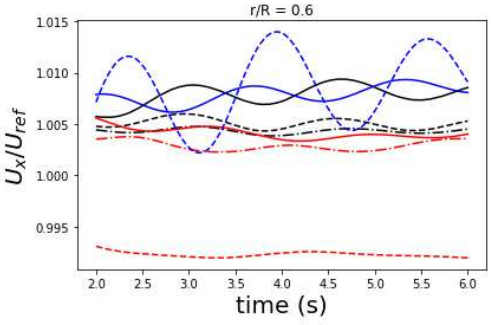
(C) r/R = 0.25 unsteady velocity FFT for drift angle = 0^o



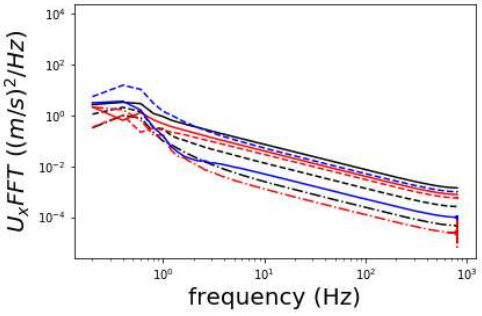
(D) r/R = 0.4 unsteady velocity for drift angle = 0^o



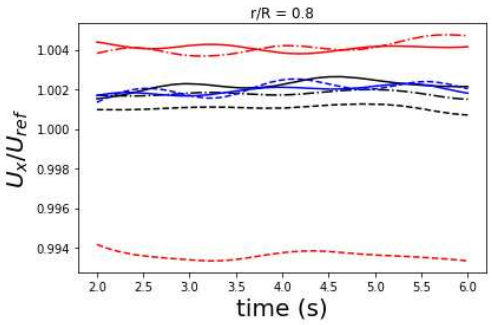
(E) rR = 0.4 unsteady velocity FFT for drift angle = 0^o



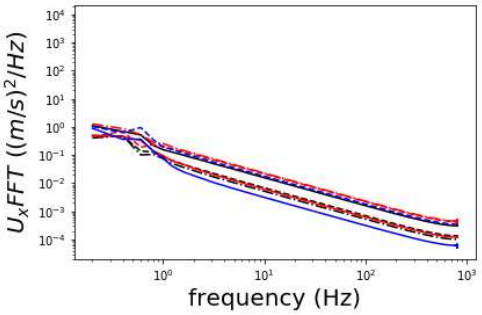
(F) r/R = 0.6 unsteady velocity for drift angle = 0^o



(G) r/R = 0.6 unsteady velocity FFT for drift angle = 0^o

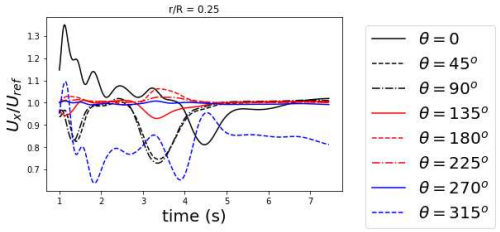


(H) r/R = 0.8 unsteady velocity for drift angle = 0^o

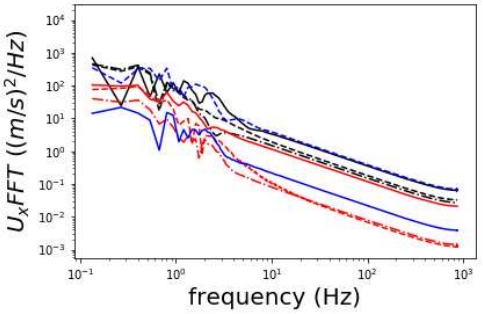


(I) r/R = 0.8 unsteady velocity FFT for drift angle = 0^o

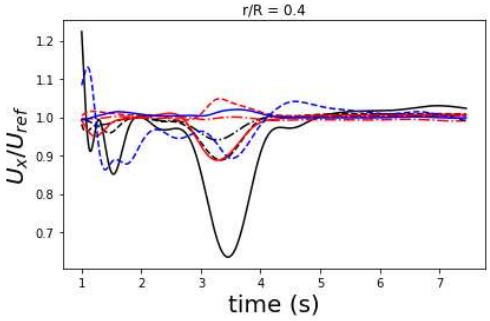
FIGURE 5.8: Unsteady Velocity plots for 0^o drift angle



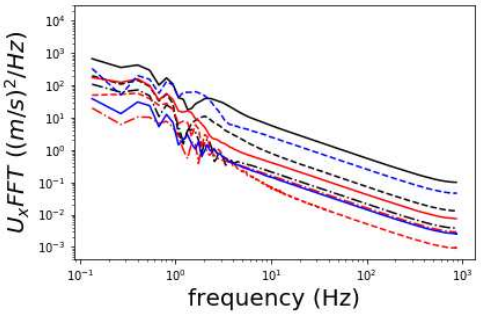
(A) r/R = 0.25 unsteady velocity for drift angle = 4^o



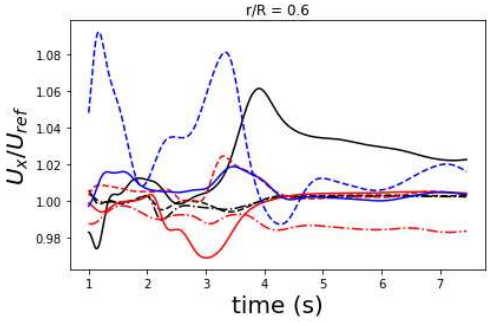
(B) r/R = 0.25 unsteady velocity FFT for drift angle = 4^o



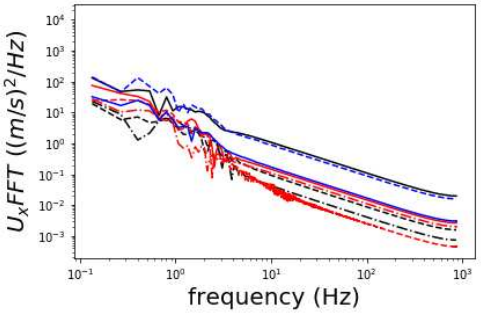
(C) r/R = 0.4 unsteady velocity for drift angle = 4^o



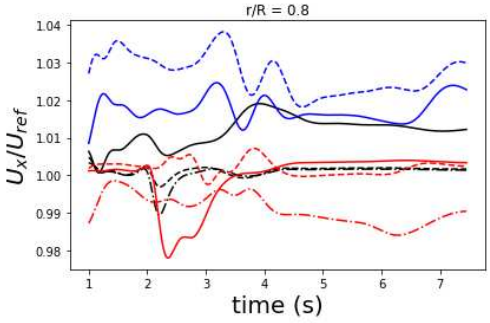
(D) r/R = 0.4 unsteady velocity FFT for drift angle = 4^o



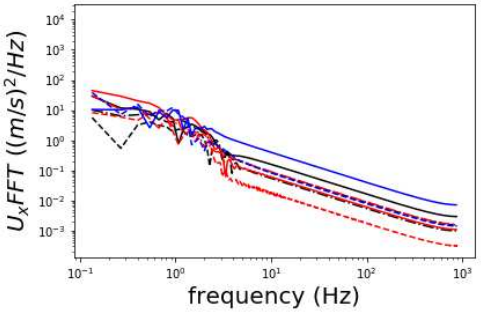
(E) r/R = 0.6 unsteady velocity for drift angle = 4^o



(F) r/R = 0.6 unsteady velocity FFT for drift angle = 4^o

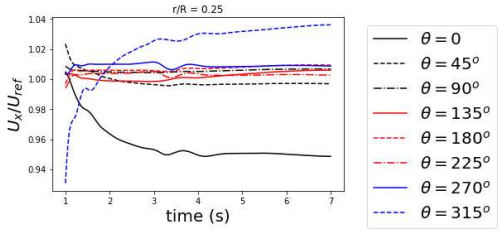


(G) r/R = 0.8 unsteady velocity for drift angle = 4^o

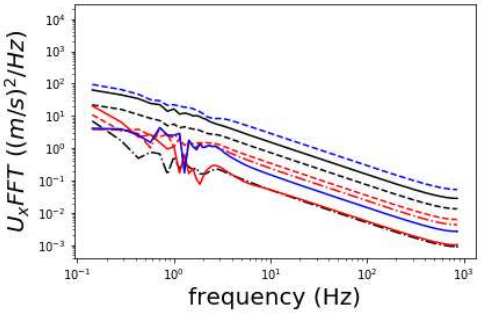


(H) r/R = 0.8 unsteady velocity FFT for drift angle = 4^o

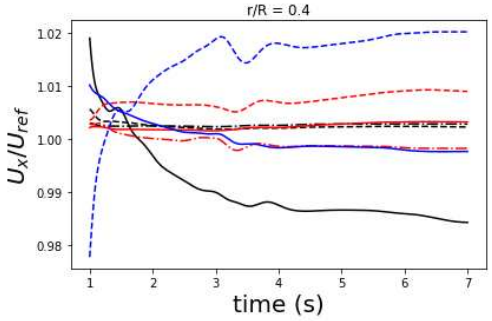
FIGURE 5.9: Unsteady Velocity plots for 4^o drift angle



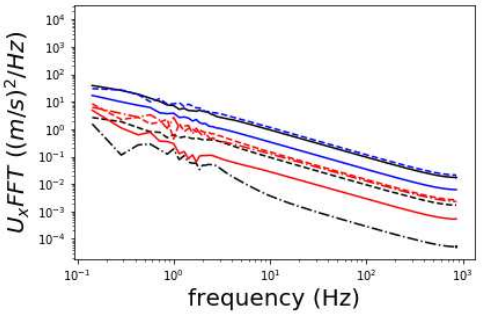
(A) r/R = 0.25 unsteady velocity for drift angle = 6^o



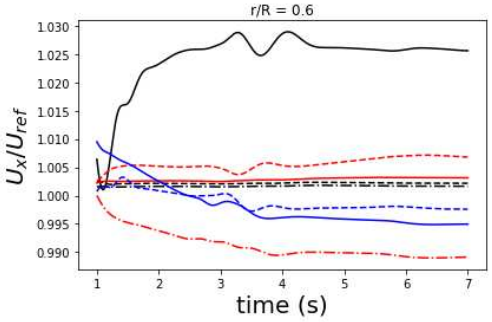
(B) r/R = 0.25 unsteady velocity FFT for drift angle = 6^o



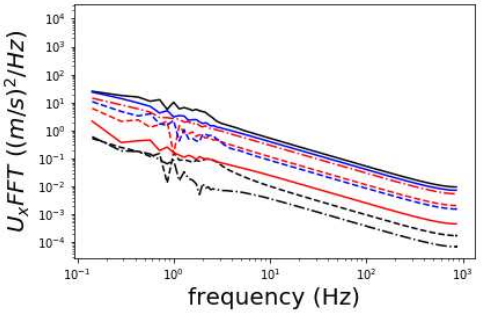
(C) r/R = 0.4 unsteady velocity for drift angle = 6^o



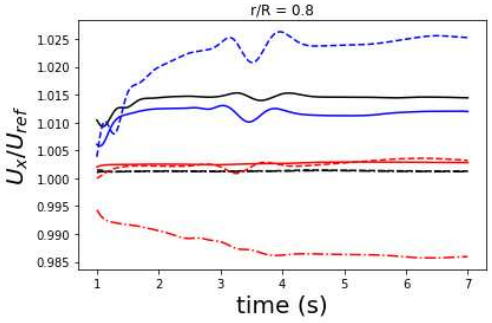
(D) r/R = 0.4 unsteady velocity FFT for drift angle = 6^o



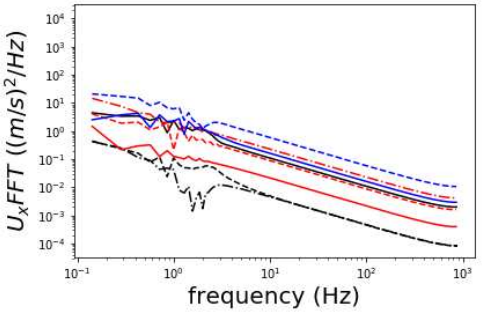
(E) r/R = 0.6 unsteady velocity for drift angle = 6^o



(F) r/R = 0.6 unsteady velocity FFT for drift angle = 6^o

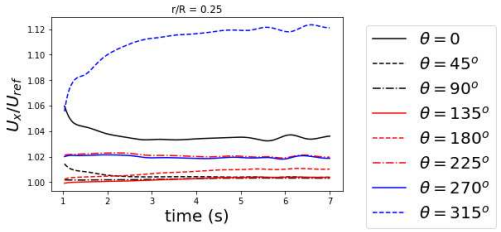


(G) r/R = 0.8 unsteady velocity for drift angle = 6^o

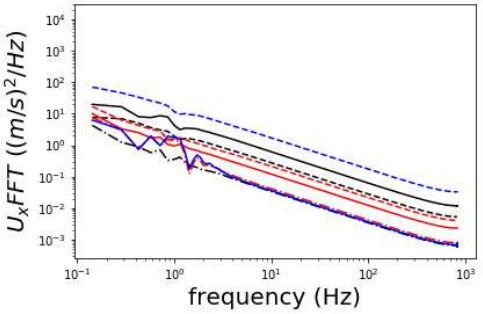


(H) r/R = 0.8 unsteady velocity FFT for drift angle = 6^o

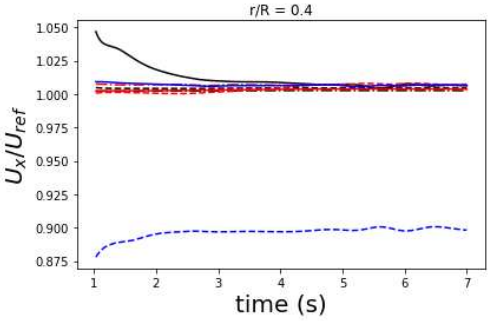
FIGURE 5.10: Unsteady Velocity plots for 6^o drift angle



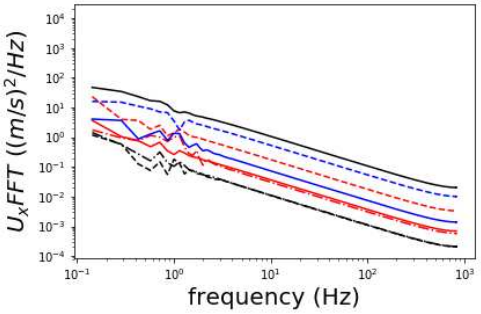
(A) r/R = 0.25 unsteady velocity for drift angle = 8^o



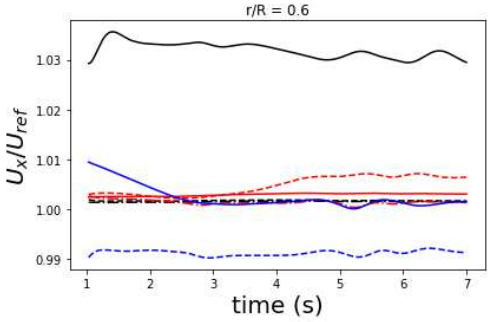
(B) r/R = 0.25 unsteady velocity FFT for drift angle = 8^o



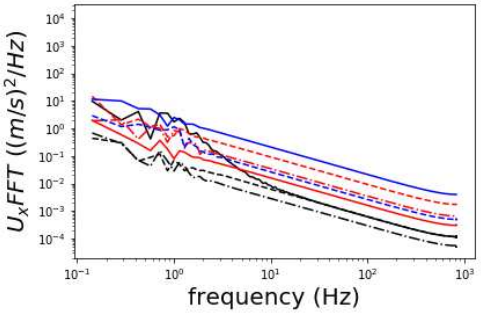
(C) r/R = 0.25 unsteady velocity for drift angle = 8^o



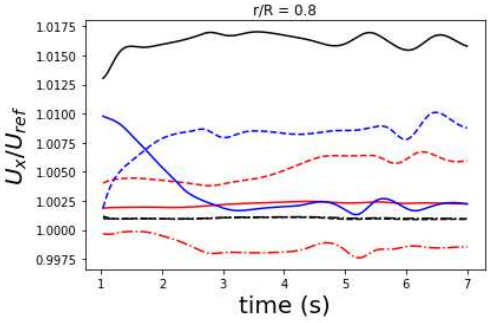
(D) r/R = 0.25 unsteady velocity FFT for drift angle = 8^o



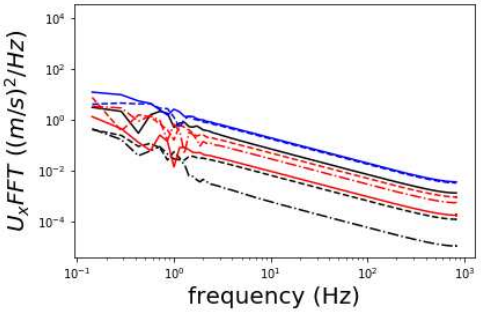
(E) r/R = 0.25 unsteady velocity for drift angle = 8^o



(F) r/R = 0.25 unsteady velocity FFT for drift angle = 8^o

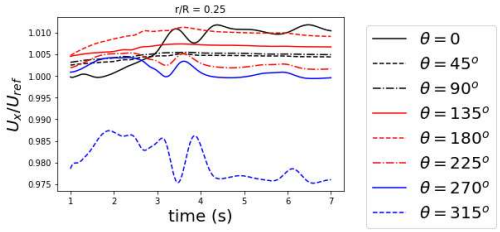


(G) r/R = 0.25 unsteady velocity for drift angle = 8^o

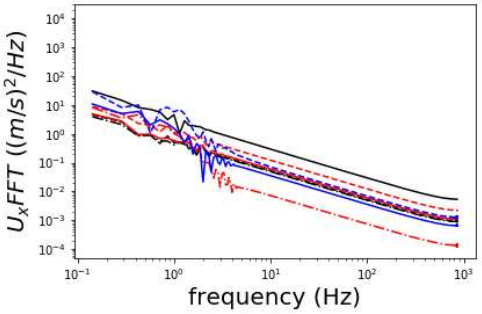


(H) r/R = 0.25 unsteady velocity FFT for drift angle = 8^o

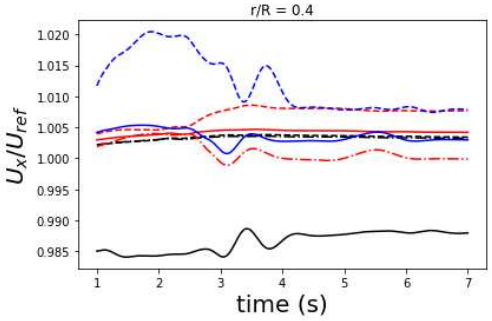
FIGURE 5.11: Unsteady Velocity plots for 8^o drift angle



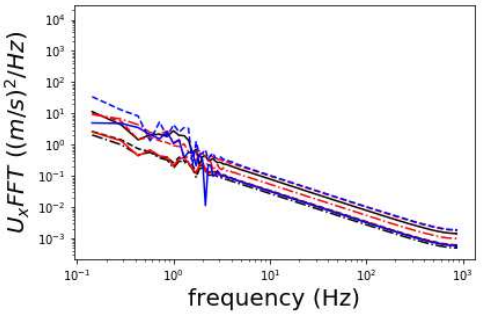
(A) r/R = 0.25 unsteady velocity for drift angle = 12^o



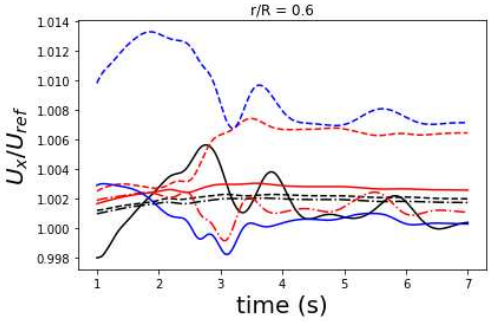
(B) r/R = 0.25 unsteady velocity FFT for drift angle = 12^o



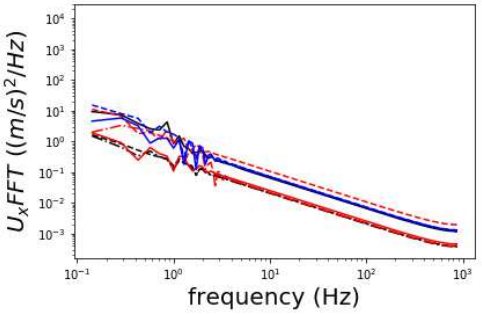
(C) r/R = 0.4 unsteady velocity for drift angle = 12^o



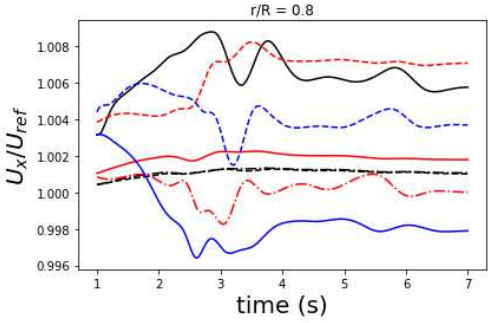
(D) r/R = 0.4 unsteady velocity FFT for drift angle = 12^o



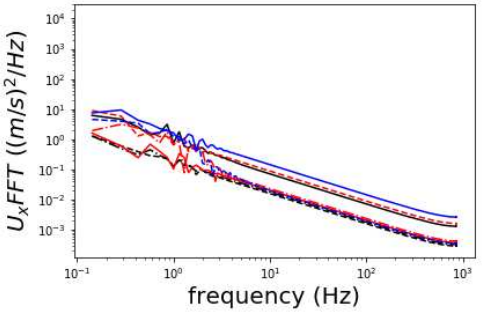
(E) r/R = 0.6 unsteady velocity for drift angle = 12^o



(F) r/R = 0.6 unsteady velocity FFT for drift angle = 12^o

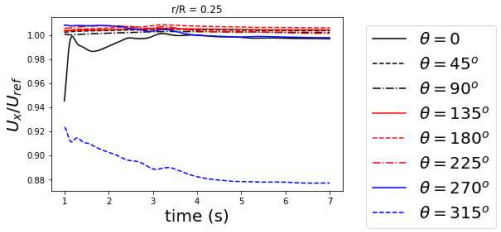


(G) r/R = 0.8 unsteady velocity for drift angle = 12^o

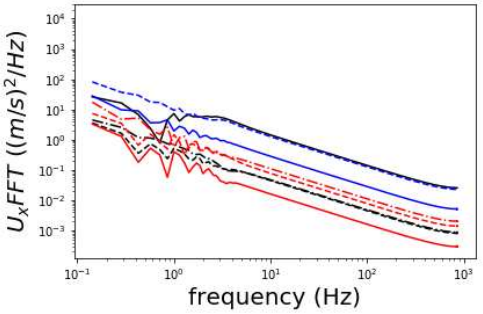


(H) r/R = 0.8 unsteady velocity FFT for drift angle = 12^o

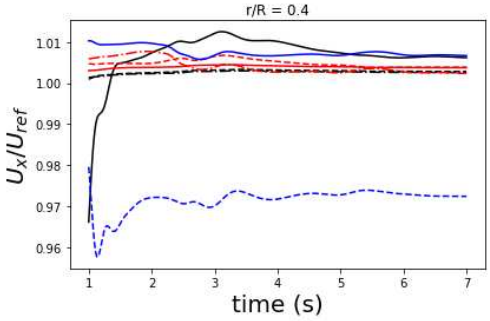
FIGURE 5.12: Unsteady Velocity plots for 12^o drift angle



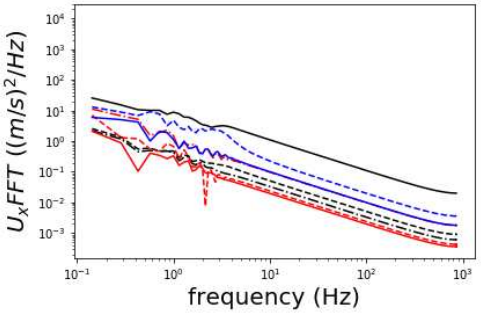
(A) r/R = 0.25 unsteady velocity for drift angle = 16^o



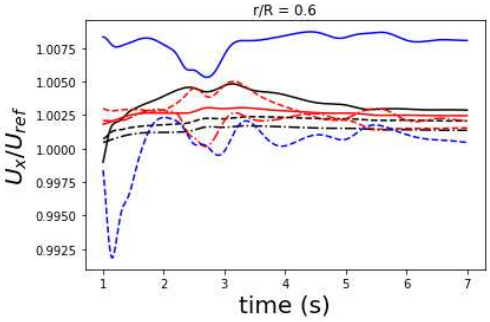
(B) r/R = 0.25 unsteady velocity FFT for drift angle = 16^o



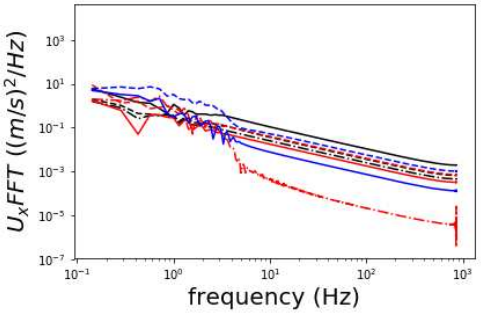
(C) r/R = 0.4 unsteady velocity for drift angle = 16^o



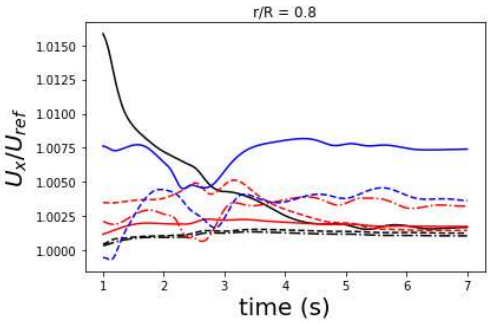
(D) r/R = 0.4 unsteady velocity FFT for drift angle = 16^o



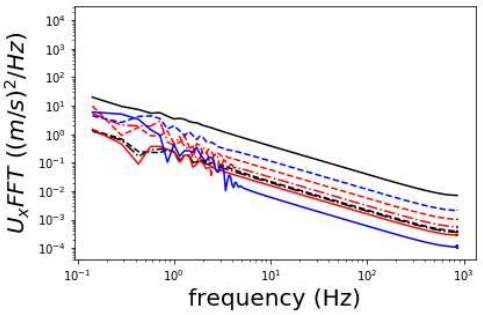
(E) r/R = 0.6 unsteady velocity for drift angle = 16^o



(F) r/R = 0.6 unsteady velocity FFT for drift angle = 16^o



(G) r/R = 0.8 unsteady velocity for drift angle = 16^o



(H) r/R = 0.8 unsteady velocity FFT for drift angle = 16^o

FIGURE 5.13: Unsteady Velocity plots for 16^o drift angle

For the 0^o drift angle case shown in Figure 5.5b, the maximum change in velocity

occurs at the low radial positions, with the largest change of velocity occurring at the low velocity positions. As the radial position moves further out the change in relative velocity gets lower. It is clear that the axial velocity oscillates around a value. To understand the frequency of these oscillations the fast Fourier transform (FFT) of the signals has been taken and shown on the right hand sides of Figures 5.8 to 5.13. This allows the frequency content to be analysed and understand what frequencies are dominant. From 5.8c it can be seen that the frequency content is made up of low frequencies in the order of 1 Hz or less. This can be compared to the blade passage frequency and natural frequency of the blade when analysing the potential response frequency of the blade.

As the drift angle increases to 4^o the frequency content increases also. This is due to the increase in unsteadiness and vortical structures intercepting the propeller plane as shown in Figure 5.7c. The increase from the reference velocity also changes greatly at the $\frac{r}{R} = 0.25$ value. The FFT plots show that there is a larger range of frequencies in the velocity signal when compared to the 0^o drift case. The higher range of frequencies can contribute to a greater vibratory response from the propeller.

The drift angle is increased to 6^o and shown in Figures 5.10. The magnitude of the axial velocity fluctuations tend to decrease whilst retaining a wide range of frequencies. Despite the relatively low fluctuations in axial velocity the radial and tangential fluctuations remain high as seen in Figures A.3 and A.9.

For a drift angle of 8^o the velocity changes are quite high at the $\frac{r}{R} = 0.25$. However as the radial position increases the magnitude of the velocity changes becomes less but the frequency content becomes greater. This is similar to the tangential and radial velocities.

As the drift angle becomes larger the amplitude of velocity changes do not change hugely but the frequency content increases although the changes in radial and tangential velocity can be large.

5.4 Large Eddy Simulation

The unsteady RANS simulations capture the large scale velocity fluctuations however this misses out the turbulent eddys at scales equivalent to the size of the mesh. These turbulent fluctuations cause small velocity variations occurring at short time scales. These high frequency velocity changes can have an impact on the vibratory response of the propeller blade.

To capture large scale velocity fluctuations Large Eddy Simulations (LES) have been used.This simulation type simulates the large scale turbulence fluctuations and

models the small scale turbulence to make the calculation computationally cheaper when compared to Direct Numeric Simulation (DNS).

LES works by first defining the quantities to be computed, for this case a velocity field which only contains the large scale components. This is done using a filtering function, so for a 1-D velocity field the filtered velocity is shown in equation 5.3.

$$\bar{u_i}(x) = \int G(x, x')u_i(x')dx' \tag{5.3}$$

Here $G(x, x')$ is the filter kernel. The filter has a local length scale associated with it, Δ Ferziger and Peric (2002). The eddies larger than Δ are defined as large eddies and the eddies smaller than Δ are small eddies and are modelled. This can be applied to the incompressible Navier Stoked equations to give.

$$\frac{\partial(\bar{u_i})}{\partial t} + \frac{\partial(\rho u_i \bar{u}_j)}{\partial x_j} = \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}[\mu(\frac{\partial \bar{u_i}}{\partial u_j} + \frac{\partial \bar{u_j}}{\partial x_i})] \tag{5.4}$$

The modelling of the small eddies take the form:

$$\tau_{ij} = -\rho(u_i \bar{u}_j - \bar{u_i}\bar{u_j}) \tag{5.5}$$

Where $\tau_{i,j}$ is called the subgrid-scale Reynolds Stress.This contains the local averages of the small scale field, therefore the model should be based on either the local velocity or the previous properties of the fluid. There are several options to model the subgrid-scale Reynolds Stress, one of which is the Smagorinsky model.

5.4.1 Simulation Set-up

The mesh for the LES simulation was produced using Pointwise. The mesh used is shown in Figure 5.14.

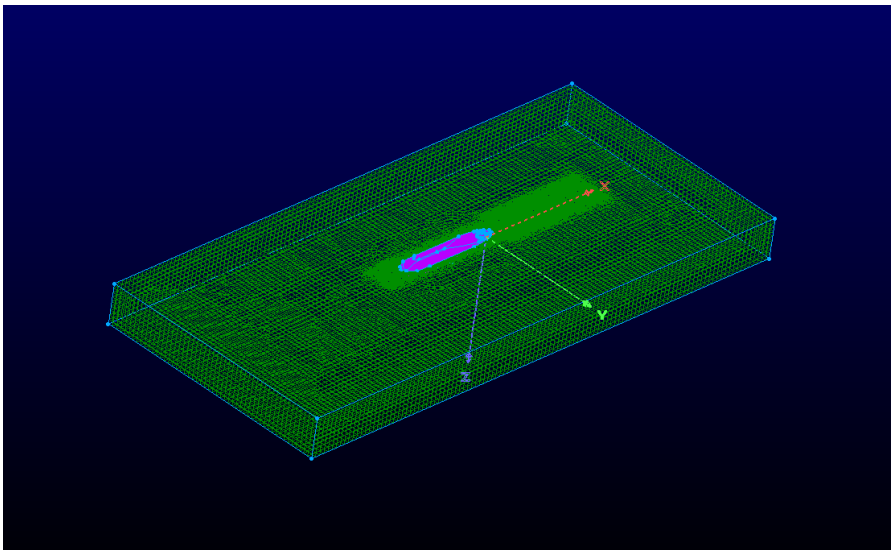


FIGURE 5.14: Overview of LES grid

The side view of the KVLCC2 using the LES grid is shown in Figure 5.15. It is shown that mesh refinement is used at the walls of the hull as well as in the wake region.

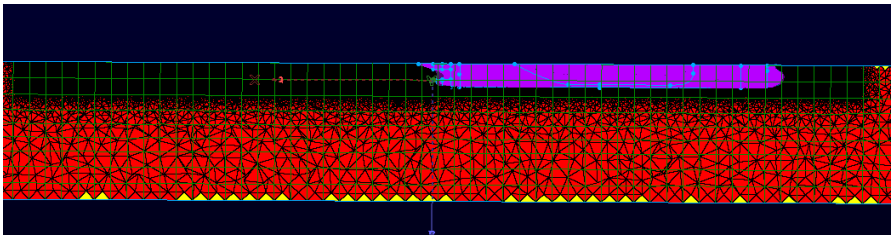


FIGURE 5.15: Side View of LES grid

High levels of refinement are used at the propeller plane as shown in Figure 5.16. This is used in order to appropriately capture the large scale eddies. The size of the eddies simulated as opposed to modelled is dependent on the local grid size.

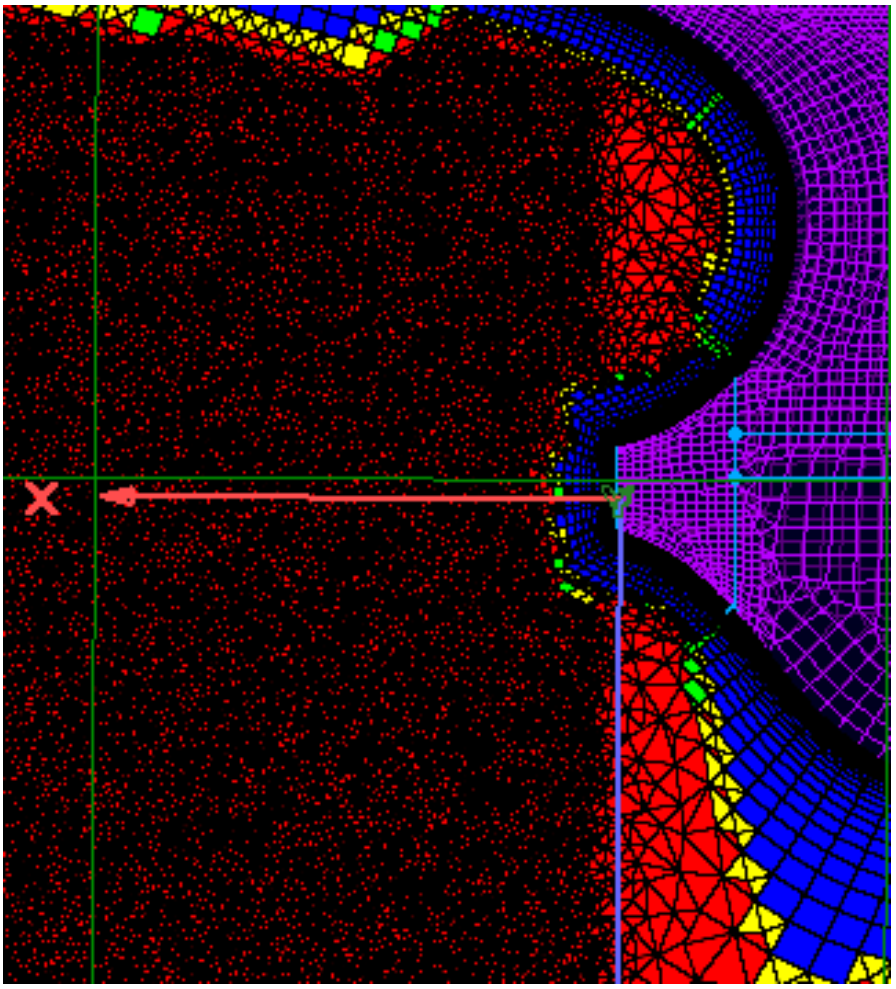


FIGURE 5.16: LES grid at Propeller Plane

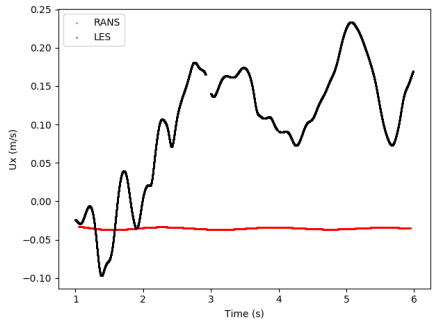
TABLE 5.7: LES Solver Conditions

CFD code	OpenFoam v1906
Simulation type	LES
Turbulence model	Smagorinsky
Δ	Van Driest
ddt Scheme	backward
algorithm	pimpleFoam
Time Step	1.67e-4
Grid Size	$\approx$ 100 million cells

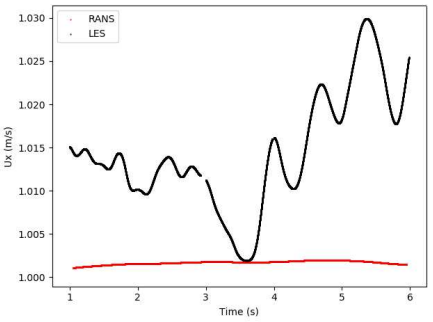
5.4.2 Results

The comparison between the RANS results and LES results can be seen in Figures 5.17a to 5.17d. The change in axial velocity with time is shown at a few radial and tangential positions. The LES simulation creates a significantly more fluctuating flow

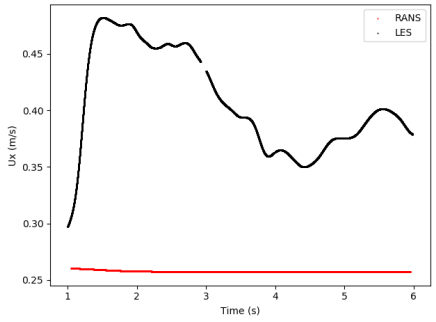
field compared to RANS simulations. The random fluctuations in the simulations have clearly been captured using LES.



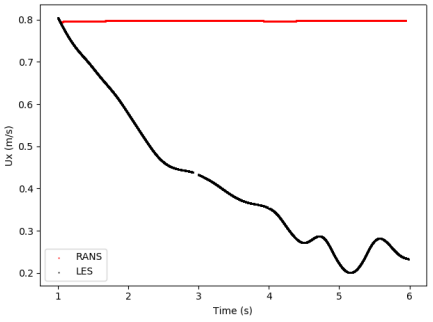
(A) r/R = 0.95 RANS vs LES at propeller plane position 1



(B) r/R = 0.95 RANS vs LES at propeller plane position 1



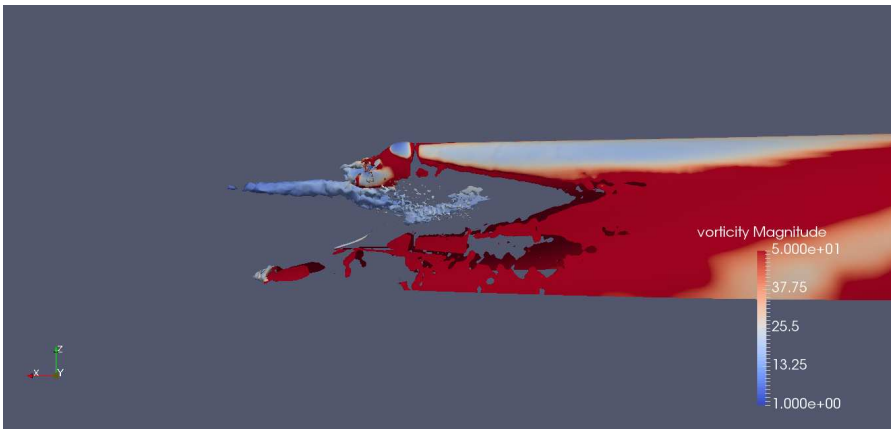
(C) r/R = 0.95 RANS vs LES at propeller plane position 2



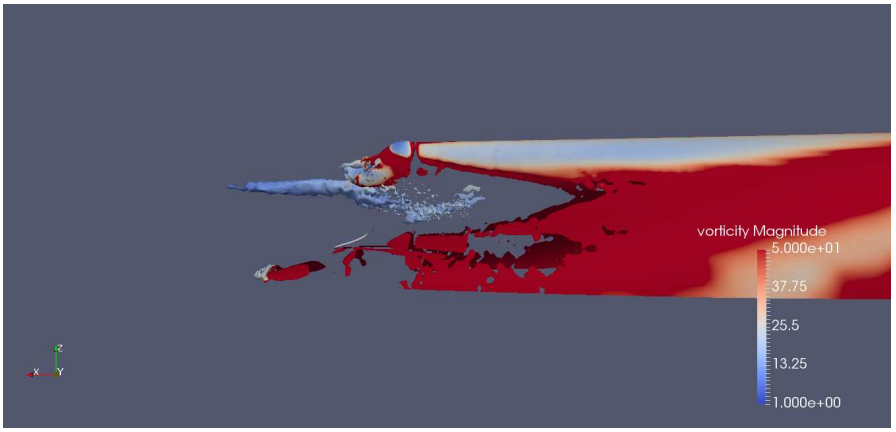
(D) r/R = 0.95 RANS vs LES at propeller plane position 2

FIGURE 5.17: RANS vs LES simulation results

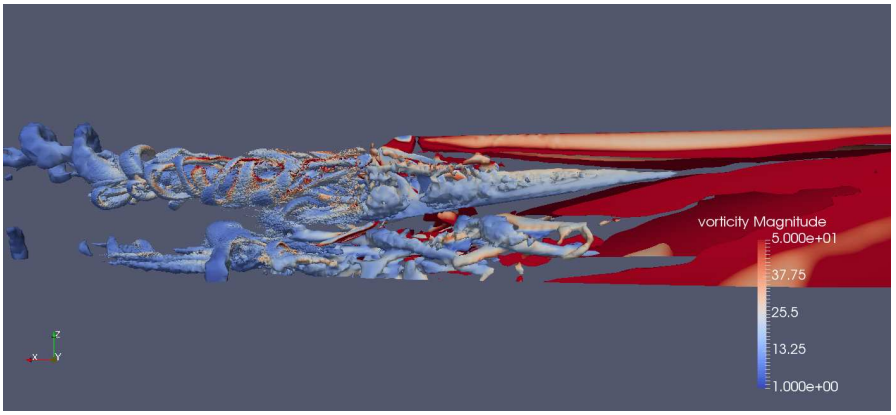
This rapid change in velocity is further exemplified by considering the Q-criterion shown in Figure 5.18. The Q-Criterion at Q=10, is shown for the RANS, URANS and LES simulations at a snapshot in time for comparison. This is for the 0^o drift condition. Here the unsteady separation in the wake is clearly seen creating a compound wake which consists of various vortical structures and is highly complex. This is in agreement with previous simulations produced in literature Fureby et al. (2016).



(A) Q criterion for RANS



(B) Q criterion for URANS



(C) Q criterion for LES

FIGURE 5.18: Q criterion for RANS, URANS and LES simulations, coloured by vorticity

5.5 Turbulent flow field modelling

From the LES results it can be seen that the large scale turbulent fluctuations can have a strong impact on the velocity fluctuations. However the computational cost for LES simulations are substantially larger than RANS simulations. Therefore it is

unreasonable to generate LES simulations for various ship manoeuvring conditions hence RANS is used. It is therefore desirable to model the turbulent fluctuations in a computationally cheap way.

This has been done for wind turbine applications using the Veers turbulence model Veers (1984). This model was also used for Tidal turbine applications in Milne et al. (2010). This model approximates the low frequency velocity fluctuations based on the atmospheric conditions.

Other synthetic turbulence models exist such as Mann (1998) which generates a 3D turbulent wind field for wind turbine applications. Like the method by Veers (1984) it uses the spectral tensor to model the fluctuations.

Synthetic turbulence models are used as an input for LES simulations. Patruno and Ricci (2018) provides a good summary of the available methods for generating turbulent fluctuations. An approach detailed consists of building the fluctuations by superposition of circular functions, usually defined as spectral functions and based on the work by Kraichnan (1970). This method has been used to easily generate homogeneous, isotropic divergence-free velocity fields Patruno and Ricci (2018).

The model implemented in the propeller vibration tool set is described by Smirnov et al. (2001). This was chosen due to its computational efficiency and ease of use when combined with RANS simulations.

Firstly the anisotropic velocity tensor of a turbulent flow field is given by

$$r_{ij} = \overline{\tilde{u_i}\tilde{u_j}} \tag{5.6}$$

This is then diagonalized using an orthogonal transformation tensor a_{ij} such that

$$a_{mi}a_{nj}r_{ij} = \delta_{mn}c^2_{(n)} \tag{5.7}$$

$$a_{ik}a_{kj} = \delta_{ij} \tag{5.8}$$

Where δ is the Kronecker delta, c_n is a vector that defines the fluctuations in the 3 directions $(u'v'w')$ and a_{ij} is the transformation tensor. The transient flow-field in 3 dimensions is given by the Kraichnan method.

$$v_i(x,t) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} [p_i^n cost(\tilde{k_j^n}\tilde{x}_j + \omega_n\tilde{t}) + q_i^n sin(\tilde{k_j^n}\tilde{x}_j + \omega_n\tilde{t})] \tag{5.9}$$

$$\tilde{x_j} = \frac{x_j}{l}, \tilde{t} = \frac{t}{\tau}, c = \frac{l}{\tau}, \tilde{k_j^n} = k_j^n \frac{c}{c_{(j)}} \tag{5.10}$$

$$p_i^n = \epsilon_{ijm} \zeta_j^n k_m^n, q_i^n = \epsilon_{ijn} \eta_j^n k_m^n \tag{5.11}$$

$$\eta_j^n, \zeta_j^n, \omega_n \subset N(0,1), k_i^n \subset N(0, \frac{1}{2}) \tag{5.12}$$

Where $v_i(x,t)$ is the transient flow field which varies in time and space, l and τ are the length and time-scales of turbulence, ϵ_{ijk} is the permutation tensor, $N(M,\sigma)$ is a normal distribution of random points with mean M and standard deviation σ. k_j^n and ω_n represent a sample of n wave number vectors and frequencies. The number of N points can be changed depending on the number of random distributions although this will affect the computational cost.

The permutation tensor in 3 dimensions is given as:

$$\epsilon_{ijk} = \begin{cases} +1, & \text{if}(i,j,k) = (1,2,3) \text{ or } (2,3,1) \text{ or } (3,1,2) \\ -1, & \text{if}(i,j,k) = (3,2,1) \text{ or } (2,1,3) \text{ or } (1,3,2) \\ 0, & \text{if} i = j \text{ or } j = k \text{ or } k = i \end{cases} \tag{5.13}$$

The next step is to apply the scaling and transformations from equations 5.8 to equation 5.9.

$$w_i c_{(i)} v_{(i)} \tag{5.14}$$

$$u_i = a_{ik} w_k \tag{5.15}$$

The correlation tensor, turbulent length and time scales are taken from the RANS simulation. The scaling factors in c_i represent the scales of the turbulent fluctuations along each axis.

It is important to note here that the velocity field generated by this procedure do not represent solutions to the Navies Stokes equations but generates an approximation to the turbulent fluctuation spectrum. This is sufficient as this is what is required for propeller load response.

5.5.1 Numerical procedure

To obtain the time dependent flow field as defined in equation 5.9 the turbulence properties at the propeller plane are taken. The velocity correlation tensor is taken from the Reynolds Stress tensor. This is then diagonalized obtaining the eigenvalues and vectors such that the conditions defined in equation 5.8 are met.

The turbulent kinetic energy k and specific turbulent dissipation rate ω are read from the CFD data whilst the turbulent dissipation rate ϵ is calculated using $\epsilon = 0.09k\omega$.

The length and time scales of turbulence l and τ are also calculated using the turbulent fields. The turbulent length and time scale are defined in equations 5.17.

$$l = C_\mu \frac{k^{\frac{3}{2}}}{\epsilon} \tag{5.16}$$

$$\tau = l/k^{0.5} \tag{5.17}$$

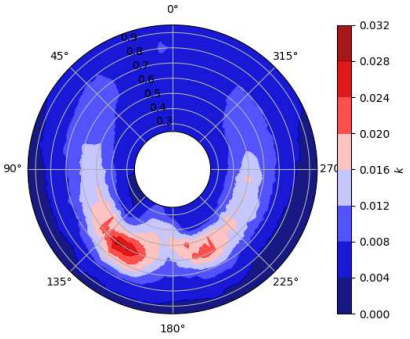
The number of random points in the normal distributions N is then defined and the ω, ζ , ξ, κ ,p and q matrices are populated using equations 5.10 to 5.12.

The majority of this implementation has been done in Python. The database of velocities at points around the propeller plane has been generated using the Numpy making the access of the velocity time history easy to manipulate. Depending on the number of random points N the population of the transient flow field described in section 5.9 can be computationally expensive. For this reason the population of equation 5.9 has been developed using Cython using a similar procedure described in section 3.4.

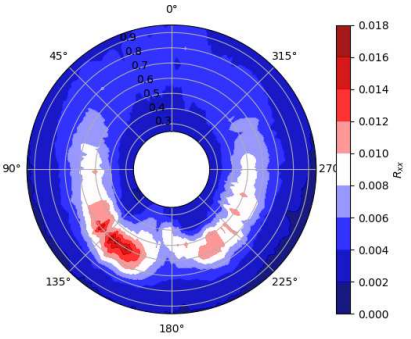
5.5.2 Turbulent inflow results

The procedure defined in 5.5.1 is used for the 0^o drift angle case and compared to the LES simulation. The input to the Krainchan method is shown in Figure 5.19. Here the turbulent kinetic energy, the xx-component of the Reynolds stress tensor and the turbulent dissipation rate at the propeller plane are shown.

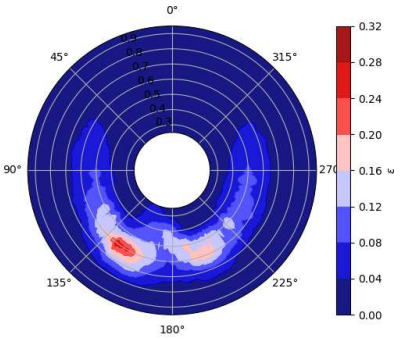
Figure 5.20 shows the turbulent overlay in comparison to the RANS and LES results.



(A) Turbulent kinetic energy at propeller plane

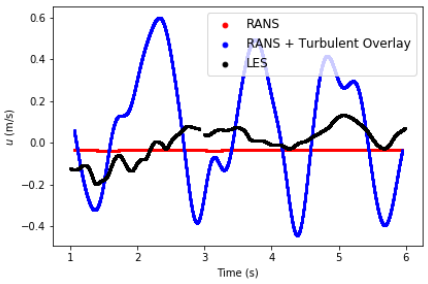


(B) R_{xx} at propeller plane

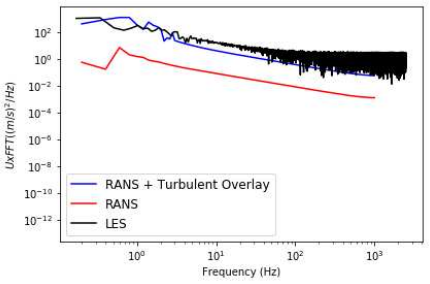


(C) ϵ at the propeller plane of KVLCC2

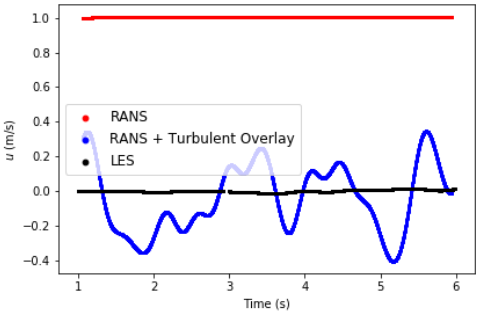
FIGURE 5.19: Turbulence properties at propeller plane for 0^o drift



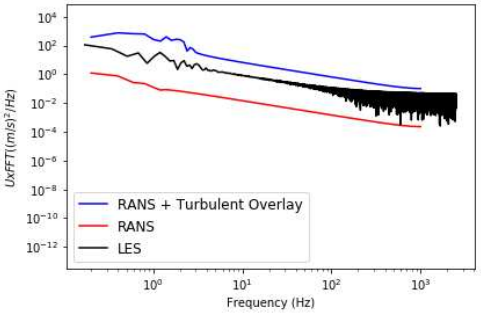
(A) r/R = 0.25 Turbulent Overlay vs RANS vs LES at propeller plane position 1



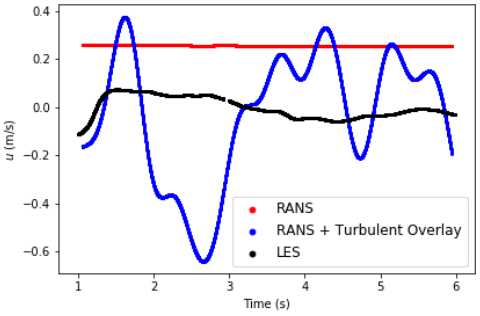
(B) r/R = 0.25 Turbulent Overlay vs RANS vs LES at propeller plane position 1 FFT



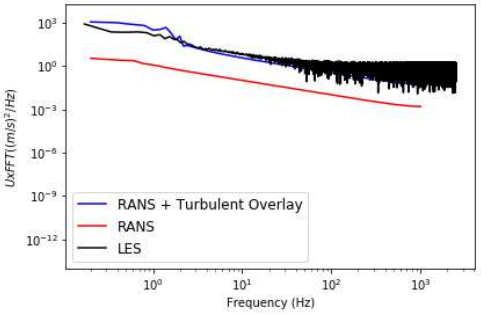
(C) r/R = 0.95 Turbulent Overlay vs RANS vs LES at propeller plane position 1



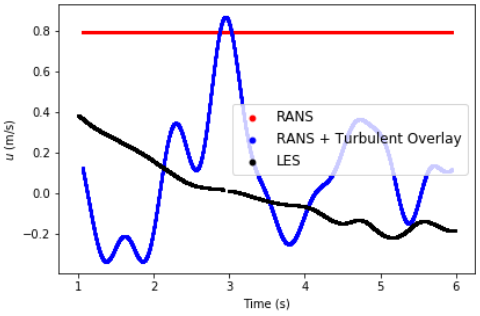
(D) r/R = 0.95 Turbulent Overlay vs RANS vs LES at propeller plane position 1 FFT



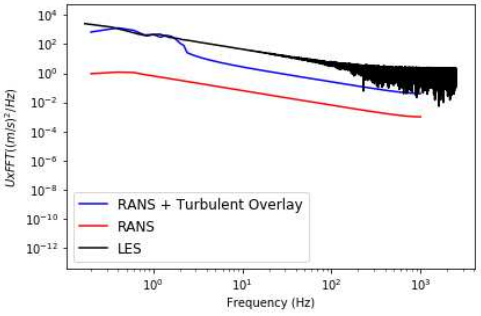
(E) r/R = 0.25 Turbulent Overlay vs RANS vs LES at propeller plane position 2



(F) r/R = 0.25 Turbulent Overlay vs RANS vs LES at propeller plane position 2 FFT



(G) r/R = 0.95 Turbulent Overlay vs RANS vs LES at propeller plane position 2



(H) r/R = 0.95 Turbulent Overlay vs RANS vs LES at propeller plane position 2 FFT

FIGURE 5.20: Turbulent Overlay vs RANS vs LES simulation results

The left hand side of Figure 5.20 is the time history of the velocity at the propeller plane for the RANS, LES and turbulent overlay. The right hand side is the FFT of the corresponding velocities. It is shown that the use of the turbulence model certainly increases the velocity fluctuations compared to the RANS case. The increase in frequency content increases to values reasonably close to the LES values.

However, at some points the turbulent overlay model performs poorly. This is because the turbulent overlay model is not designed to solve for the Navier-Stokes equation. In addition, the turbulent overlay model is based on the RANS solution at the propeller plane which does not predict the unsteady wake to the same extent as the LES. LES is capable of capturing the shed vortical structures shown in Figure 5.18.

The turbulent overlay model is, however, a useful addition as it approximates the turbulent fluctuations of the wake based on the turbulence properties well. Despite over-estimating some of the velocity fluctuations it can give a useful 'worse-case scenario' to the designer.

Although LES could be used instead of using this turbulent flow overlay the computational cost of obtaining an LES solution is significantly higher. The aim of the turbulent overlay is to use information from RANS data to generate a good approximation to the turbulent fluctuations. This is of particular use when simulating the hull in unsteady manoeuvring conditions when the computational cost for LES is particularly high.

5.6 Model Scale vs Full Scale

The model ship wake does not completely represent the wake of the full scale ship. This is due to the Reynolds number difference causing a relative difference in boundary layer thickness between the model ship and the full scale ship Carlton (2012). It is therefore desirable to scale the wake to the full scale wake for propeller design purposes. The change in wake from model scale to full scale is a wake contraction.

Methods have been developed to numerically obtain the full scale wake from the model scale, namely the method by Sasajima (1966) that assumes the displacement wake is purely potential. Hoekstra (1975) extended this method to three dimensions and modelled three forms of contraction: centre-plane contraction, concentric contraction and a contraction to a horizontal plane above the propeller.

Wang and Bobo (2011) outlines the various models for the assessment of hull aft end performance and compares each model against a series of criterion as described by Odabasi and Fitzsimmons (1978):

1. The maximum wake around the top dead center position of the propeller disc should satisfy: $w_{max} < MIN0.75; CB$.
2. The maximum acceptable wake peak should satisfy: $w_{max} < 1.7\bar{w}_{0.7}$
3. The width of the wake peak should not be less than $\theta_B = 360/Z + 10deg$.
4. The cavitation number for the propeller tip, defined as:

$$\sigma_n = \frac{9.903 - D/2 - Z_p + T_a}{(0.051\pi nD)^2} \tag{5.18}$$

should lay in the safe region defined with reference to non-dimensional wake gradient.

Wang and Bobo (2011) found that none of the models filled every criterion and none of the models were ideal with changes in hull form causing each model to behave differently. Therefore, due to its ease of implementation the Hoekstra wake model has been implemented to account for wake scaling.

5.6.1 Hoekstra Wake model

This sub section describes the methodology of the Hoekstra Wake model and its implementation to the model wakes defined. The comparison between the full scale wake using the wake model and CFD are then compared.

5.6.1.1 Methodology

Firstly the model-ship boundary layer thickness ratio is determined as the contraction factor c. This is determined from the coefficients of frictional resistance of the full scale ship and model given in equation 5.21. This is from ITTC '58

$$c = \frac{C_{Fs} + \Delta C_{Fs}}{C_{Fm}} \tag{5.19}$$

$$C_{Fs} = \frac{0.075}{(log_{10}Re_s - 2)^2} \tag{5.20}$$

$$C_{Fm} = \frac{0.075}{(log_{10}Re_m - 2)^2} \tag{5.21}$$

Where Re_s is the full scale ship Reynolds number and Re_m is the model scale Reynolds number. ΔC_{Fs} is taken as the roughness allowance of hull resistance given by Townsin (1985) as:

$$\Delta C_{Fs} = 0.044((\frac{k_s}{L_{wl}})^{0.33} - 10Re_S^{-0.333}) + 0.000125 \tag{5.22}$$

Where L_{wl} is the length of the ship at the water line with $k_s = 150 \times 10^{-6}m$ being recommended.

For the Hoekstra wake model the contraction factor is split into three components:, the contraction factor to the centre of the propeller-shaft, j the contraction to the centre plane and k the contraction to the hull above the propeller.

The ship wake is defined in terms of radial and tangential coordinates (r, θ), the point of application of the velocity vector is then changed due to the three contraction factors given as:

$$r_{new} = i[c(r - r_{hub}) + r_{hub}] + \tag{5.23}$$
$$j[r\sqrt{cos^2\theta + c^2 sin^2\theta}] + \tag{5.24}$$
$$k[r\sqrt{c^2 cos^2\theta + sin^2\theta}] \tag{5.25}$$
$$\tag{5.26}$$

$$\theta_{new} = i\theta + \tag{5.27}$$
$$j[tan^{-1}(ctan\theta)] + \tag{5.28}$$
$$k[tan^{-1}(\frac{1}{c}tan\theta)] \tag{5.29}$$

If i, j or k is negative then the absolute value is taken and c is replaced by 2-c to obtain expansion over contraction. Hoekstra (1975) have conducted preliminary calculations which have indicated that it is more accurate to not account for the k contraction in the lower half of the disc. Modifications are made to r_{new} to account for the presence of the hub. This modification only applies to concentric contraction.

If $|rcos\theta| < 2r_{hub}$ then the j component of r_{new} becomes:

$$j[\sqrt{r^2 cos^2\theta + c(r|sin\theta| - A) + A^2} \tag{5.30}$$

where $A = r_{hub}^2 - 0.25r^2 cos^2\theta$. Also if $|rsin\theta| < 2r_{hub}$ the j component of r_{new} becomes:

$$j[\sqrt{c(r|cos\theta| - B) + B^2 + r^2 sin^2\theta} \tag{5.31}$$

Where $B = r_{hub}^2 - 0.25r^2 sin^2\theta$. The contraction factors i, j and k are obtained by taking the harmonic content of the model wake field. It is assumed that the 1st six harmonics is sufficiently representative. The coefficients obtained are :

$$
\begin{aligned}
S_i &= 1 - A_o + \begin{cases} A_2 + A_4 + A_6 - 0.5S_K \text{ If} S_k \geq S_j \\ -S_j + (A_2 + A_4 + A_6) \text{ If} S_k < S_j \end{cases} \\
S_j &= -[A_2 + A_4 + A_6 + |max(A_2 cos2\theta + A_4 cos4\theta + A_5 cos6\theta)|] \\
S_k &= 2(A_1 + A_3 + A_5)
\end{aligned}
$$

Where A_n are the fourier coefficients. These are obtained using the scientific python FFT module. These coefficients S_i,$S_j$and S_k are then integrated over the radius of the wake disc between r_{hub} and D. The contraction factors are then given by:

$$i = \frac{F_i}{|F_i| + |F_j| + |F_k|} \tag{5.32}$$

$$j = \frac{F_j}{|F_i| + |F_j| + |F_k|} \tag{5.33}$$

$$k = \frac{F_k}{|F_i| + |F_j| + |F_k|} \tag{5.34}$$

in which

$$F_i = \int_{r_{hub}}^{2R} S_i(r)dr \tag{5.35}$$

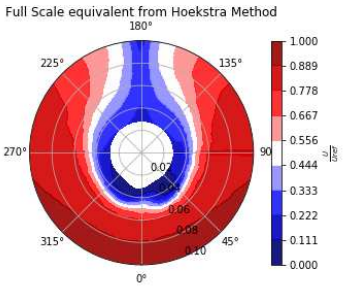
$$F_j = \int_{r_{hub}}^{2R} S_j(r)dr \tag{5.36}$$

$$F_k = \int_{r_{hub}}^{2R} S_k(r)dr \tag{5.37}$$

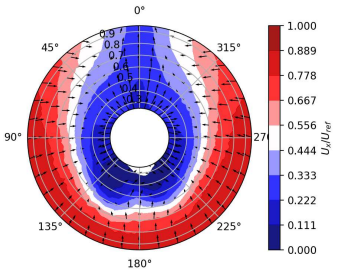
5.6.1.2 Results

This implementation has been tested by comparing the wake generated using the Hoeskstra method on the CFD simulation at model scale against a full scale CFD simulation presented in Pereira et al. (2017).

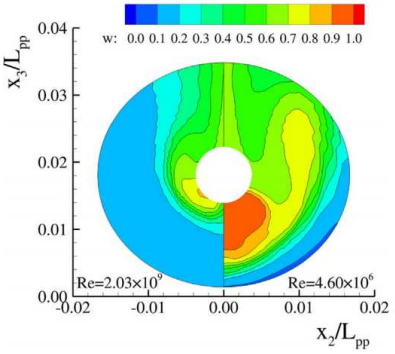
Figure 5.21 shows the axial velocities at the propeller plane. Figure 5.21a shows the wake generated from the Hoekstra model. It can be seen that the wake contraction is well captured and overall compares well to the full scale wake shown in Figure 5.21c.



(A) Wake obtained using model scale CFD and Hoekstra method



(B) Model scale wake obtained from CFD



(C) Full scale wake and Model scale wake from Pereira et al. (2017)

FIGURE 5.21: Comparison between Full scale wake from Hoekstra model and wake from Pereira et al. (2017)

This is a useful result as the computational effort to model the full scale hull is significantly higher than the effort to model the hull at model scale. The change in velocity field is significantly altered when operating at full scale. The velocity at the outer regions of the propeller operating region is higher, this will change the loading on the propeller significantly.

5.7 Chapter Summary

In this chapter a computational fluid dynamics method has been developed to capture the wake of the KVLCC2 hull form at a drift angle of 0^o. The wake at the propeller plane was validated by previous numerics and experimentation on the hull form from literature.

The RANS simulations have been extended to obtain the unsteady flow field at the wake field. The steady state solution has been verified at the propeller plane giving reasonable confidence that the URANS simulations generate a reliable unsteady wake.

This has been done for drift angles of -2^o,0^o,4^o,6^o,8^o,12^o,16^o to create a series of unsteady wakes which can be used as inputs to the computationally efficient FSI tool.

The unsteady flow field will cause time dependent loading on the propeller which will cause the propeller to respond in an unsteady manner.

By using the Kraichan turbulence model a synthetic turbulent field can be generated to capture the effects of the turbulent fluctuations on the structural response of the propeller blade. This is useful as it can reduce the need for computationally expensive LES simulations to obtain a unsteady wake which the propeller experiences.

Finally, the Hoekstra wake model has been used to obtain the wake of a full scale ship using the output of a model scale simulation. This can reduce the need for a full scale hull form simulation which can be significantly more computationally expensive compared to model scale simulations.

The work complete in this chapter can be used as the input velocities to the propeller hydrodynamic model described in Chapter 3. In future work the CFD simulation can be extended to include the influence of a free surface and more complex manoeuvres such as zig zagging or turning circles.

Chapter 6

Efficient Structural Modelling of a Propeller Blade

6.1 Introduction

This chapter discusses the modelling techniques available to model the structural response of the propeller blade. Finite element Analysis (FEA) is a well established technique to obtain the structural properties of an object. This includes obtaining the mass and stiffness matrices of the structure which dictate the response of the object to an external load.

An FEA solution splits the structure into many small parts called elements. These are usually simple shapes such as triangles, rectangles, or the 3D equivalent where obtaining the mass and stiffness matrices are relatively simple. These can then be joined to obtain the mass and stiffness matrices of a complex geometry.

There are several element types which capture different physics and have different computational costs. The elements studied in this chapter are: Plate, Shell and 3D elements. The implementation of these elements, as well as their computational cost and ability to capture the blade properties are studied and compared to industry standard methods.

The development of a robust and accurate structural model will allow the dynamic pressure loads obtained from the hydrodynamic model to be applied and the structural response can be computed. Although this is possible using commercial codes such as Ansys, these can have a significant computational cost particularly when the dynamic response is required.

6.2 Plate Model Theory

A plate element is a structural model to approximate the stress and deformation. There are two main plate models: the Kirchoff-love plate theory A. E. H. Love (1888), or the Reissner-Mindlin plate theory R. D. Mindlin (1951). The Kirchoff-love plate theory is commonly referred to as thin plate theory and Reissner-Mindlin is thick plate theory. The thin plate theory has limitations in that shear deformations are ignored. Thick plate theory, however, does not make this assumption and is therefore seen as a more applicable theory for propeller blades. This is due to propeller blades having a relatively low aspect ratio. Although plate theory will not be a completely accurate representation of the blades it will give a good approximation as blades tend to be reasonably flat and will exhibit small deflections.

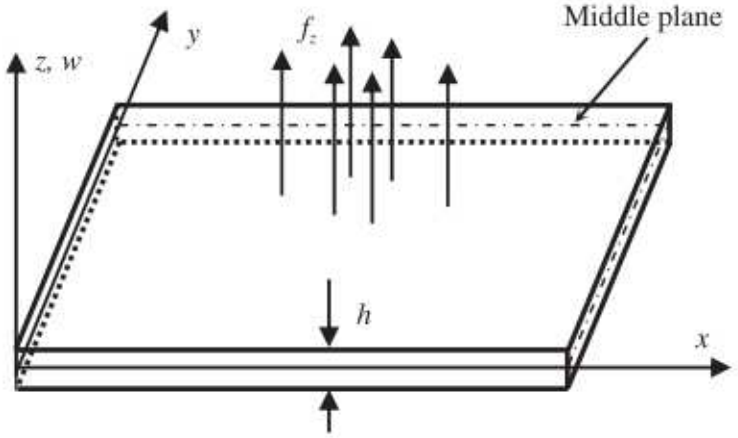


FIGURE 6.1: Schematic of Thin Plate Theory

The model is split into several elements each with a degree of freedom in the z-direction and rotation about the x and y-directions as seen in figure 6.1. The plate is usually modelled with constant thickness h. The assumptions of thin plates are described by Ventsel (2001):

- The material is elastic, homogeneous and isotropic.
- The plate is initially flat.
- The deflection is small compared to the thickness of the plate
- The length of the elements at the mid-plane are not altered. This means that the vertical shear strains γ_{xy} and γ_{yz} are negligible, as is the normal strain ϵ_x
- The stress normal to the middle plane σ_z is small compared to the other stress components and is therefore neglected.

- The displacements of the plate are small.

These assumptions are known as the Kirchhoff hypotheses. This aids in the development of the Kirchhoff plate which is valid for thin plates. However, an extension to this theory is the Reissner Mindlin plate theory which is more practical for 'thick' plates Liu (2014). The practical differences between thick plate theory and thin plate theory is the ratio between the reference length a and the thickness h. A thick plate has a ratio $\frac{a}{h}$ of $\leq \approx 8$, whereas a thin plate has $\approx 8 \leq \frac{a}{h} \leq \approx 80$ Liu (2014). Typically a marine propeller has an length to thickness ratio within the 'thick' region albeit close to the upper limits.

The three dimensional structural problem can be simplified by removing a degree of freedom and making the system two-dimensional.

For 2D stress, the components are:

$$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} \tag{6.1}$$

And the strain components are:

$$\epsilon = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \tag{6.2}$$

The strain-displacement relationships are:

$$\epsilon_{xx} = \frac{\partial u}{\partial x}, \epsilon_{xx} = \frac{\partial v}{\partial y}, \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{6.3}$$

Where u and v are the displacements in the x and y directions respectively. The stress and strain can be related using Hooke's law such that:

$$\sigma = c\epsilon \tag{6.4}$$

Where

$$c = \frac{E}{1-\nu^2}\begin{Bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{Bmatrix} \tag{6.5}$$

For application to the Reissner-Mindlin plate theory the displacement components can be expressed as:

$$\begin{aligned} u &= z\theta_y \\ v &= -z\theta_x \end{aligned} \tag{6.6}$$

where θ_y and θ_x are the rotations about the y and x axis respectively.

6.3 FEA development

To effectively model the deformation of the plate model the geometry must be split into several elements. The element shapes can be: triangular, rectangular, or quadrilateral. For the propeller geometry, quadrilateral elements have been chosen due to the flexibility in geometry. Although triangular elements can be used for complex geometry, they have been shown to be less accurate compared to quadrilateral elements.

For FEA two coordinate systems exist a global coordinate system and a local coordinate system. These can also be called physical and natural coordinate systems respectively. The global coordinates contain information of the geometry, the local coordinates are always numbered anti-clockwise. The local coordinates are η and ξ and the at node 1η = -1 ,ξ = -1 at node 2 η = -1 ξ = 1, etc as shown in figure 6.2.

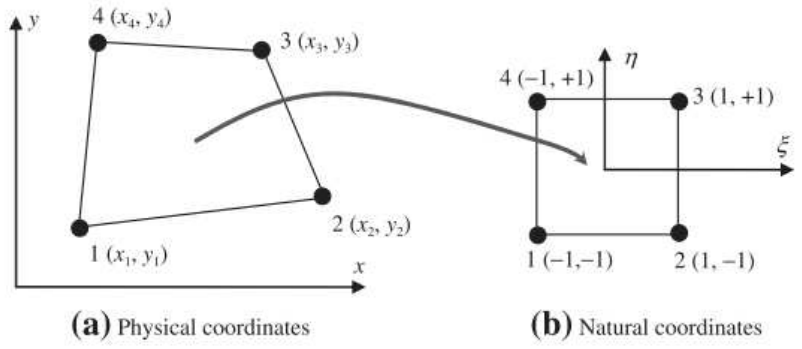


FIGURE 6.2: Coordinate mapping from Global to local coordinates

6.3.1 Shape Functions

To interpolate the deflections and stresses from nodal positions to give a field variable, shape functions are used. There is one shape function for each local node with the value of the shape function equal to one at its corresponding nodal position and equal to zero everywhere else. For the quadrilateral elements the shape functions are:

$$\begin{aligned} N_1 &= \frac{1}{4}(1-\xi)(1-\eta) \\ N_2 &= \frac{1}{4}(1+\xi)(1-\eta) \\ N_3 &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_4 &= \frac{1}{4}(1-\xi)(1+\eta) \end{aligned} \tag{6.7}$$

For example, at node 1 $\xi = -1$ and $\eta = -1 \therefore N_1 = 1$,$N_2 = 0$,$N_3 = 0$,$N_4 = 0$

The shape functions can be stored in a matrix such that:

$$N = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & N_3 & 0 & 0 & N_4 \end{bmatrix} \tag{6.8}$$

This can be used to obtain the global coordinates from the local coordinates such that:

$$\mathbf{X} = \mathbf{N}x_e \tag{6.9}$$

Where $\mathbf{X}$ is the vector of the physical coordinates and x_e is a vector of global coordinates at the nodes. The deflections and rotations can then be expressed by:

$$\begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} = \mathbf{N}d_e \tag{6.10}$$

Where d_e is the displacement vector for all nodes.

$$d_e = \begin{Bmatrix} w_1 \\ \theta_{x1} \\ \theta_{y1} \\ w_2 \\ \theta_{x2} \\ \theta_{y2} \\ w_3 \\ \theta_{x3} \\ \theta_{y3} \\ w_4 \\ \theta_{x4} \\ \theta_{y4} \end{Bmatrix} \tag{6.11}$$

6.3.2 Element matrices

The deflection, velocity and acceleration of the plate are dictated by the equation of motion:

$$[M]\ddot{x} + [C]\dot{x} + [K]x = f \tag{6.12}$$

Where $[M]$ is the mass matrix,$[K]$ is the stiffness matrix and $[C]$ is the damping matrix. Each finite element has its own mass, damping and stiffness matrices which are combined to make the global matrices.

To populate the local matrices the kinetic energy of the plate is:

$$T_e = \frac{1}{2}\int_{Ve} \rho(\dot{u}^2 + \dot{v}^2 + \dot{w}^2)dV \tag{6.13}$$

Where u , v and w are the displacement components of the plate and the $\dot{}$ notates time derivatives.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} z\theta_y \\ -y\theta_x \\ w \end{bmatrix} \tag{6.14}$$

Where the left hand side is the displacement matrix. The kinetic energy equation then becomes:

$$T_e = \frac{1}{2}\int_{Ae} \rho(h\dot{w}^2 + \frac{h^3}{12}\dot{\theta_x}^2 + \frac{h^3}{12}\dot{\theta_y}^2)dA \tag{6.15}$$

Where h is the thickness of the plate element. In matrix form this gives:

$$T_e = \frac{1}{2}\int_{Ae} \rho(\dot{d}^T I \dot{d})dA \tag{6.16}$$

Where **I** is the inertial matrix:

$$\mathbf{I} = \begin{bmatrix} \rho h & 0 & 0 \\ 0 & \rho h^3/12 & 0 \\ 0 & 0 & \rho h^3/12 \end{bmatrix} \tag{6.17}$$

Using equation 6.10 the mass matrix for the element can be defined as:

$$m_e = \int_{A_e} N^T I N dA \tag{6.18}$$

The stiffness matrices are defined from the potential energy for the plate:

$$U_e = \frac{1}{2}\int_{A_e}\int_{-h/2}^{h/2} \epsilon^T \sigma dA dz + \frac{1}{2}\int_{A_e}\int_{-h/2}^{h/2} \tau^T \gamma dA dz \tag{6.19}$$

The 1st term on the right hand side is the in-plane stresses and strains and the 2nd term on the right hand side is the transverse stresses and strains. γ is the off plane shear strain components given as:

$$\gamma = \begin{Bmatrix} \theta_y + \frac{\partial w}{\partial x} \\ -\theta_x + \frac{\partial w}{\partial y} \end{Bmatrix} \tag{6.20}$$

and τ is the average shear stresses given as:

$$\tau = \kappa \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} = \kappa c_s \tag{6.21}$$

Where G is the shear modulus and κ is the shear correction factor. The shear correction factor is to account for the non uniformity of the shear stress across the thickness of the plate. The shear stress is non-uniform as the shear stress is zero at the surfaces Liu (2014). Substituting 6.1, 6.2, 6.20, 6.21 into 6.19 gives the elemental stiffness matrix.

$$k_e = \int_{A_e} \frac{h^3}{12} [B^I]^T c B^I dA + \int_{A_e} \kappa h [B^o]^T c_s B^o dA \tag{6.22}$$

Where $[B^I]$ is the strain matrix associated with the in-plane stress taking the form $B^I = [B_1^I B_2^I B_3^I B_4^I]$ and $[B^o]$ is the strain matrix associated with the off-plane shear stress and strain.

$$B_j^I = \begin{bmatrix} 0 & \frac{\partial N_j}{\partial x} & 0 \\ 0 & 0 & \frac{\partial N_j}{\partial y} \\ 0 & \frac{\partial N_j}{\partial y} & \frac{\partial N_j}{\partial x} \end{bmatrix} \tag{6.23}$$

and

$$B_j^o = \begin{bmatrix} \frac{\partial N_j}{\partial x} & -N_j & 0 \\ \frac{\partial N_j}{\partial y} & 0 & -N_j \end{bmatrix} \tag{6.24}$$

6.3.3 Global Matrix Formation

To solve the matrix problem a global matrix is formed. This is done by first creating the mesh. The mesh consists of a series of quadrilateral elements made up of 4 nodes. Each node has an ID number, an IEN and a node list. The ID is an array with each array element corresponding to a node. The IEN corresponds to the element and consists of 4 array elements corresponding to the node number. The node list contains information on the global coordinates of the element. An example for a 2x2 square matrix is shown in figure 6.3.

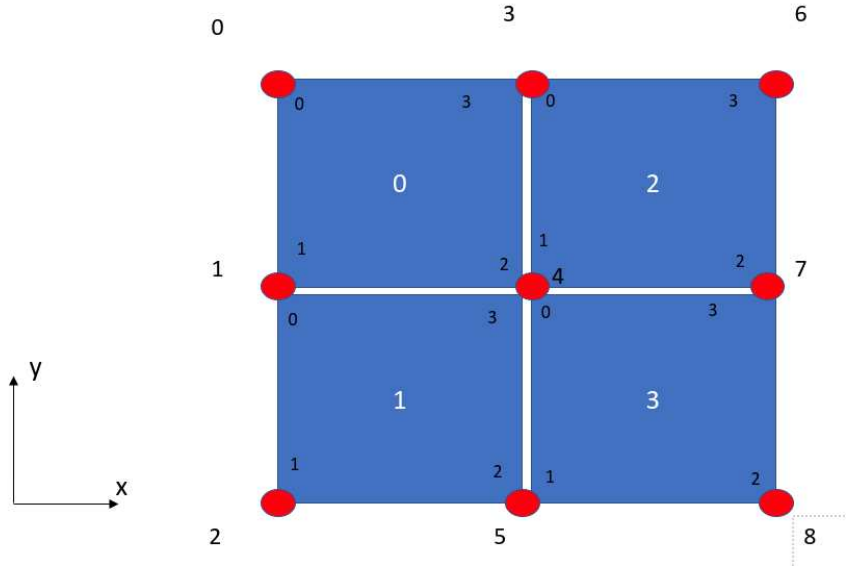


FIGURE 6.3: 2 by 2 finite element grid

Here the element numbers are in the centre of the elements. The global node numbers range from 0 to 8 and the local node numbers are inside the elements and range from 0 to 3 . Local node numbers are inside the elements and are arranged in an anti clockwise direction. The IEN for this case would be:

$$IEN = \begin{bmatrix} 0 & 1 & 4 & 3 \\ 1 & 2 & 5 & 4 \\ 3 & 4 & 7 & 6 \\ 4 & 5 & 8 & 7 \end{bmatrix} \tag{6.25}$$

The 1st row of matrix 6.25 refers to element 1, the 2nd row refers to element 2 etc. The ID matrix stores the boundary conditions. For the purposes of the thesis this will be a cantilever boundary condition at the y = 0 points. As each node has 3 degrees of freedom, the condition for each of these three conditions are defined. For a cantilever plate the ID matrix is shown in equation 6.26.

$$ID = \begin{bmatrix} -3 \\ -2 \\ -1 \\ -3 \\ -2 \\ -1 \\ -3 \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ \vdots \\ 15 \\ 16 \\ 17 \end{bmatrix} \tag{6.26}$$

The node matrix defines the global coordinates of each node to capture the physical geometry. This consists of the x and y coordinates for each node.

The global mass and stiffness matrices can now be populated. This is done by applying equations 6.18 and 6.22 to each element. The local mass and stiffness matrix are then added to the global matrices only if the ID corresponding to the nodes of the element is above or equal to 0, hence applying the boundary conditions. Using the ID matrix and IEN matrix the correct global matrix position can be added to such that

TABLE 6.1: Material Properties for Square Plate

Youngs Modulus	2E11
Poissons ratio ν	0.3
Plate thickness	0.1m
Plate Length	1m
Density	7850 kg/m^3

elements with sharing nodes can be added together. For example the ID value of element 0, node 2 is equal to the ID value of element 1, node 3. i.e ID[IEN[0,2]] == ID[IEN[1,3]].

To compute the integral in equations 6.18 and 6.22 gauss quadrature is used. It is important to note here that 2x2 Gauss points are used for the quadrature, however this is not done for the 2nd integral of 6.22. This is due to the phenomena called 'Shear locking' where the plate becomes over stiff when the thickness of the plate is reduced, this is caused by the fact that shape functions are used to approximate the deformation of the displacements and thus constraining the deformation to that shape function Liu (2014). To overcome this, only one Gauss point is used for the 2nd term in 6.22.

6.3.4 Application to Square Plate

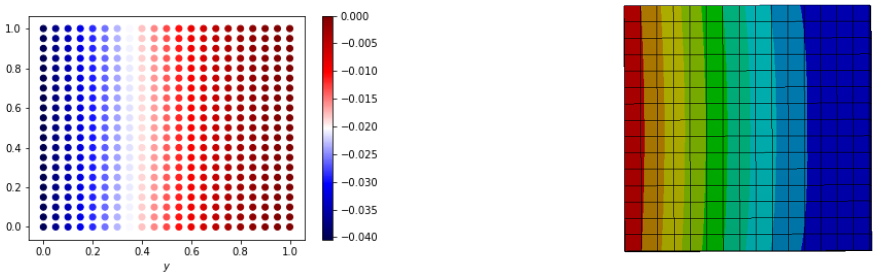
The FEA plate model was applied to a square plate to ensure correct physics and implementation. A simple square geometry was chosen with side length 1m and cantilevered on the y = 1 position. The material properties are shown in table 6.1.

To test the mass and stiffness matrices have been populated correctly, the natural frequencies were calculated and compared to Ansys. The natural frequency was chosen as it only checks for mass and stiffness as opposed to a deflection test which would mean ensuring the force is applied correctly also. The 1st 6 natural frequencies of the Plate model and Ansys model is compared as shown in table 6.2.

TABLE 6.2: Comparison between Plate model and full 3D FEA model from Ansys

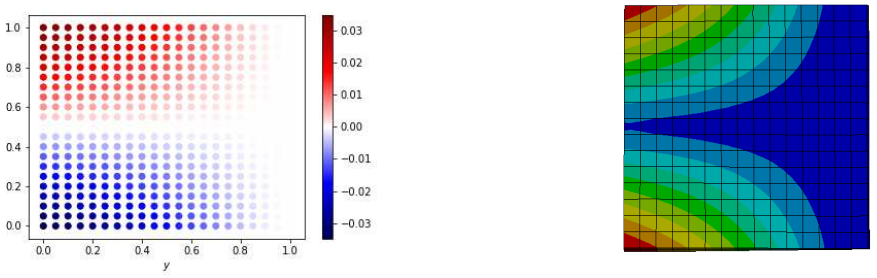
Mode	Ansys Frequency, f_{ref} (Hz)	Python Frequency, f (Hz)
1	83.786	83.4
2	196.65	196.11
3	491.18	491.05
4	530.41	-
5	621.43	622.2
6	690.2	689.64
7	1161.2	1160.59

Moreover, the mode shapes have been compared as shown in figures 6.4a to 6.10b. Here the figures marked a) are from the Plate model and the figures marked b) are from Ansys. The colours in the figures marked a) indicate the level of deflection in that mode. It can be seen that the mode shapes match well. It is, however, important to note that mode 4 of the Ansys model shown in figure 6.7a occurs on the x-y plane which is not included as a degree of freedom in the Plate model.



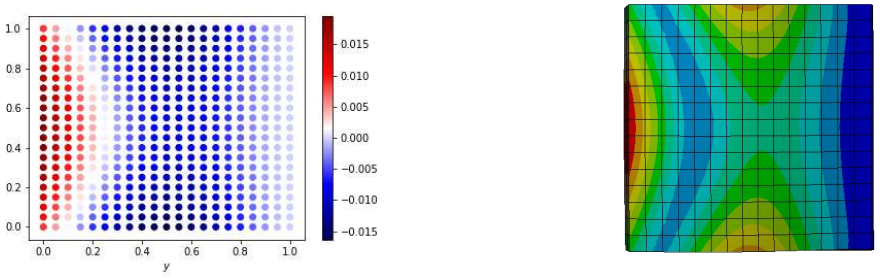
(A) (B)

FIGURE 6.4: Comparison of mode shapes for mode 1



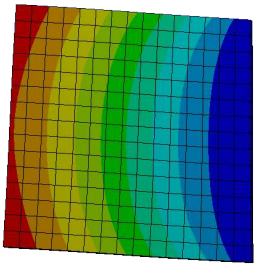
(A) (B)

FIGURE 6.5: Comparison of mode shapes for mode 2



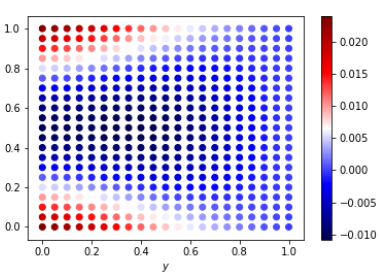
(A) (B)

FIGURE 6.6: Comparison of mode shapes for mode 3

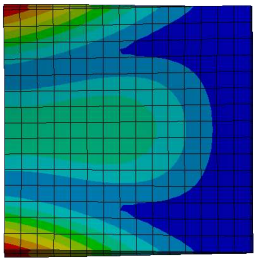


(A)

FIGURE 6.7: Comparison of mode shapes for mode 4

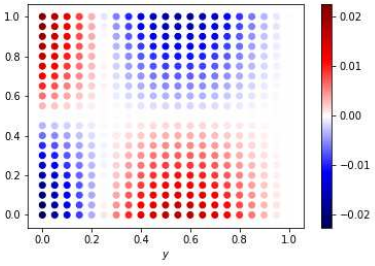


(A)

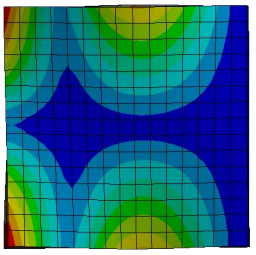


(B)

FIGURE 6.8: Comparison of mode shapes for mode 5

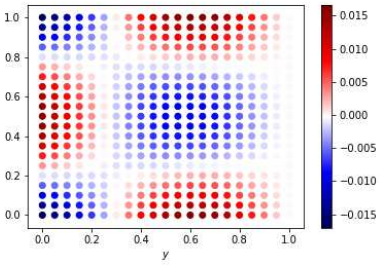


(A)

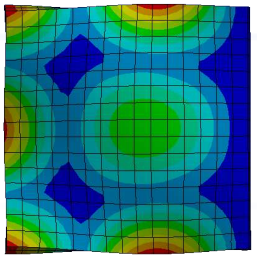


(B)

FIGURE 6.9: Comparison of mode shapes for mode 6



(A)



(B)

FIGURE 6.10: Comparison of mode shapes for mode 7

The 1st mode shape is a clear bending mode with the second mode clear torsion which are standard for cantilevered geometry.

The mesh consisted of a 10x10 grid. The grid dependence can be clearly demonstrated in figure 6.11

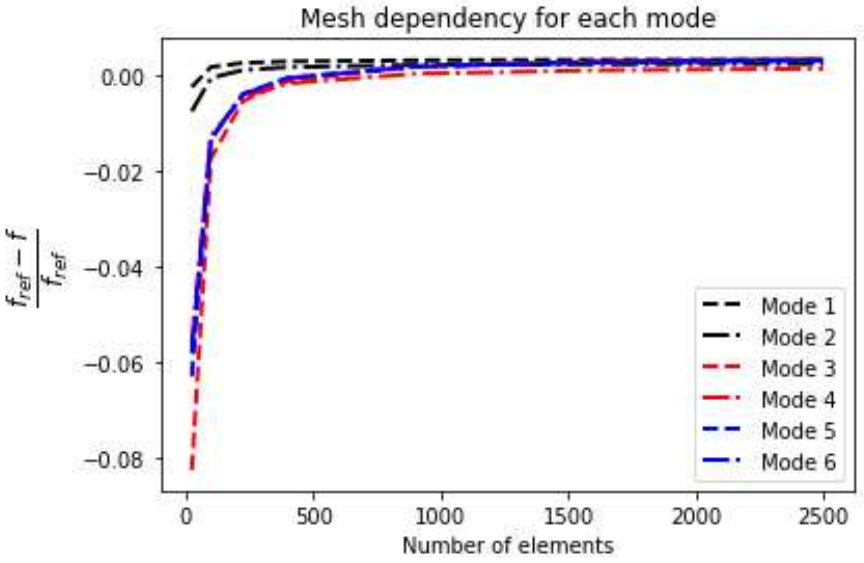


FIGURE 6.11: Mesh Convergence of Square Plate. Difference between computed natural frequency and reference frequency

Where f are the frequencies of each mode depending on the mode and f_{ref} are the frequencies taken from Ansys. The mesh sizes are a 5x5 , 10x10, 15x15,20x20,30x30,40x40 and 50x50 grids. At low grid resolution the maximum error is 8%, as the grid increases the computational cost increases as shown in figure 6.12. This was run on an Intel Core i7-6700 CPU @ 3.40GHz processor.

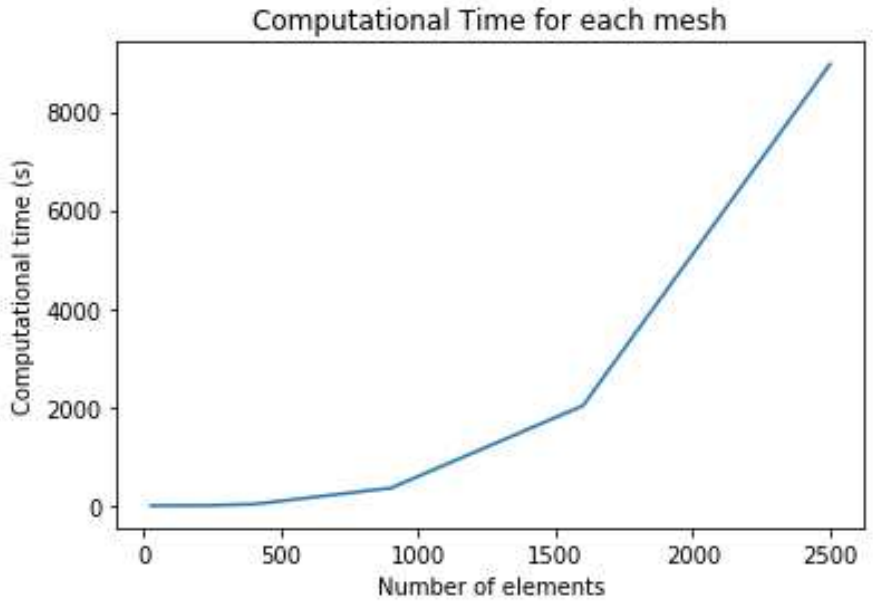


FIGURE 6.12: Change in computational cost due to increasing mesh size

The computational time increases rapidly as the number of elements increases. Therefore, the minimum number of elements required for an accurate result is to be used. From Figure 6.11 it can be seen that an accurate result can be achieved using ≈1000 elements

6.3.5 Application to Propeller Geometry

The objective of the plate model is to apply it to a Propeller blade, therefore the model is applied to a propeller geometry such as the work completed by Khan (1997). To do this effective meshing must take place and variable thickness to account for the variable thickness in the propeller.

6.3.5.1 Meshing

To mesh the Propeller blade the propeller geometry file is used as an input. Here each radial position is defined as well as the chord, skew and thickness. The nodal points of the blade are then defined using the equations from Carlton (2007).

$$x_p = -[i_G + r\theta_s tan(\theta_{nt})] + (0.5c - \frac{x}{c})sin(\theta_{nt} + y_{u,l}cos(\theta_{nt}) \tag{6.27}$$

$$y_p = rsin[\theta_+ \frac{(0.5c - \frac{x}{c})cos(\theta_{nt}) - y_{u,l}sin(\theta_{nt})}{2r}] \tag{6.28}$$

$$z_p = rcos[\theta_+ \frac{(0.5c - \frac{x}{c})cos(\theta_{nt}) - y_{u,l}sin(\theta_{nt})}{2r}] \tag{6.29}$$

Where, i_G is the generator line rake, r is the radial position, θ_{nt} is the pitch angle, θ_s is the skew angle, x is the x coordinate, c is the chord and $y_{u,l}$ is the upper or lower coordinate of the foil section. The foil geometry can be changed for each propeller, the foil geometry for this case is the NACA66 mod geometry. The x_c and y_c is shown in equation 6.31.

$$\begin{aligned}
\frac{x_t}{T} &= [0 \quad 0.2368 \quad 0.3249 \quad 0.4368 \quad 0.6086 \quad 0.8262 \quad 0.9493 \quad 1 \quad 0.9736 \quad 0.8617 \\
&\qquad \ldots \quad 0.6871 \quad 0.4709 \quad 0.2378 \quad 0.1209 \quad 0.0042] \\
\frac{x}{c} &= \begin{bmatrix} 0 & 0.0125 & 0.025 & 0.05 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 0.95 & 1 \end{bmatrix} \\
\frac{x_m}{M} &= [0 \quad 0.0906 \quad 0.1585 \quad 0.2709 \quad 0.4479 \quad 0.6988 \quad 0.8627 \quad 0.9606 \quad 0.9992 \quad 0.9778 \quad 0.8884 \\
&\qquad \ldots \quad 0.7021 \quad 0.3684 \quad 0.1842 \quad 0] \qquad (6.30)
\end{aligned}$$

and

$$y_{u,(l)} = \frac{x_m}{M} M + (-) \frac{x_t}{T} T \qquad (6.31)$$

where, M is the maximum camber for that section and T is the maximum thickness.

There is now points defined in Cartesian coordinates which describe the geometry of the propeller blade.

The next step is to apply the node ID values for each node point, the propeller blade is to be cantilevered so the points at the root of the blade are set such that the deflection and rotation are zero, the IEN is then populated. As the shape of the propeller blade is not a square or rectangle the finite elements will not be square, this is an advantage of using quadrilateral elements as they can be formed in an arbitrary shape.

The square plate validation case is of uniform thickness however, the propeller blade is not. To account for thickness, each element is prescribed a thickness dictated by the thickness distribution, described by the input file. To account for the chordwise thickness distribution equation 6.30 is used. However, this results in zero thickness at the trailing edge, this is corrected by adding a trailing edge thickness in the order of $\approx$ 1mm to the section profile.

For the HMRI propeller this results in the mesh shown in figure 6.13.

TABLE 6.3: Material Properties for HMRI propeller

Material Name	Stainless Steel
Youngs Modulus	2E11
Poissons ratio ν	0.3
Propeller Diameter	0.212m
Density	7850 kg/m^3
Yield Stress	260 MPa

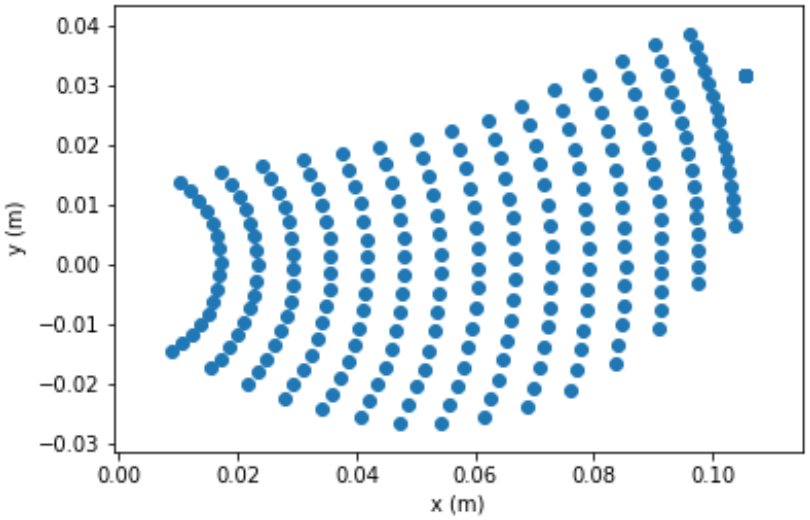
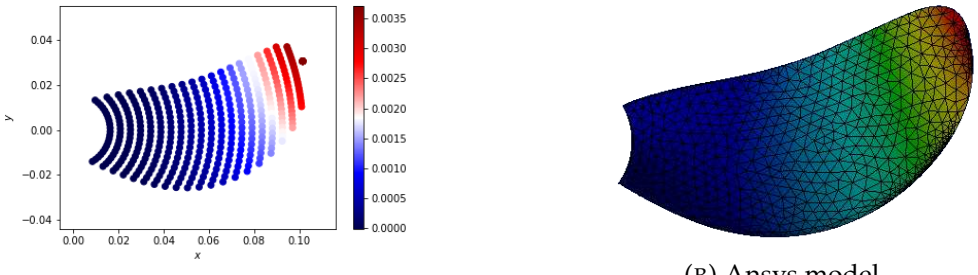


FIGURE 6.13: FEA mesh for HMRI propeller with a 15x15 element mesh

The material properties and propeller geometry are shown in table 6.3.

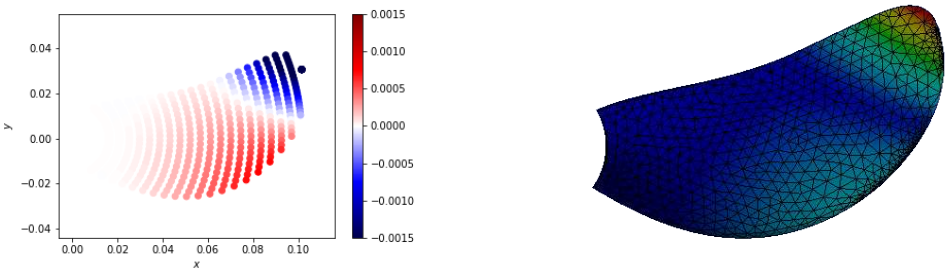
An Ansys model has been generated to verify the applicability of the Plate model. Again, the modal frequencies and shapes have been used as a verification case. The results of the Ansys model and Plate model are shown in figures 6.29a to 6.29n



(A) Python model

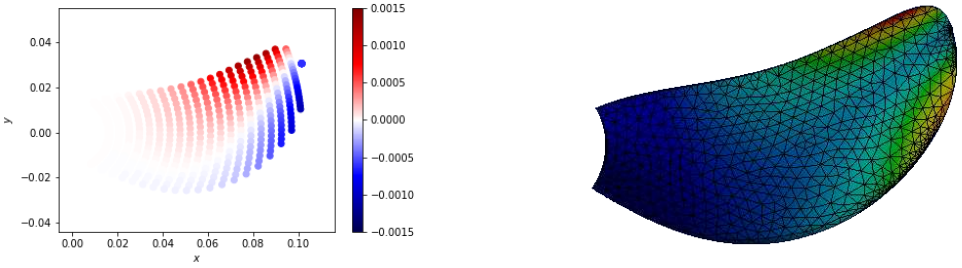
(B) Ansys model

FIGURE 6.14: HMRI propeller vibration mode 1



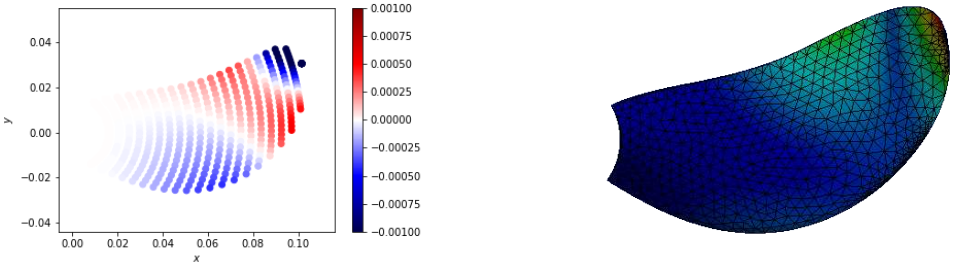
(A) Python model (B) Ansys model

FIGURE 6.15: HMRI propeller vibration mode 2



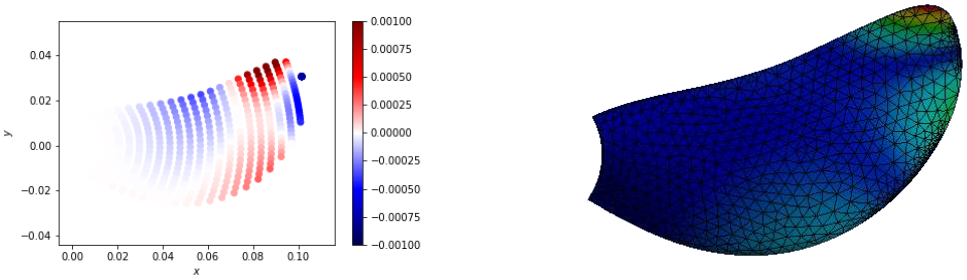
(A) Python model (B) Ansys model

FIGURE 6.16: HMRI propeller vibration mode 3



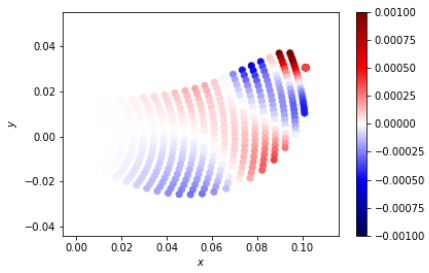
(A) Python model (B) Ansys model

FIGURE 6.17: HMRI propeller vibration mode 4

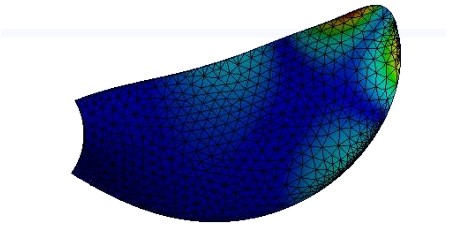


(A) Python model (B) Ansys model

FIGURE 6.18: HMRI propeller vibration mode 5



(A) Python model



(B) Ansys model

FIGURE 6.19: HMRI propeller vibration mode 6

It can be seen that the mode shapes match closely between the plate model and the Ansys Model. As seen in Table 6.4 the frequencies of the 1st two modes match quite well however the higher frequency do not match as well. This seems to be an issue with the chord of the blade as the higher modes tend to be rotational modes around the centre line.

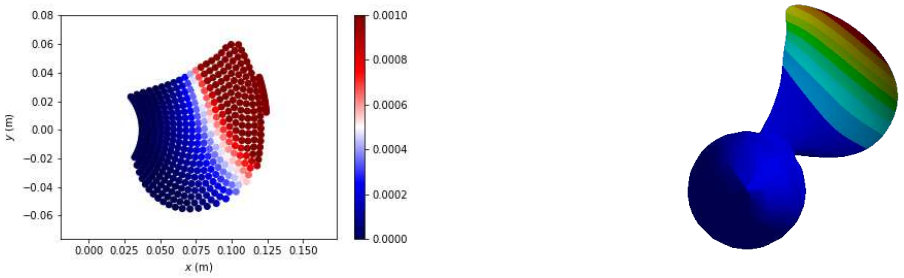
TABLE 6.4: Comparison between Plate model and full 3D FEA model from Ansys for HMRI propeller

Mode	Ansys Frequency, f_{ref} (Hz)	Python Frequency, f (Hz)	Error
1	1132.6	1094.2	3%
2	2435.2	2541.76	4%
3	3637.7	4130.24	11%
4	4520.4	5002.45	9.6%
5	4935.	7084.78	30%
6	6809.5	8424.8	20%

The same analysis has been performed using the Potsdam Propeller test case which has a higher chord thoughout the blade with results shown in table 6.5. Here the frequencies match closely throughout the first six modes.

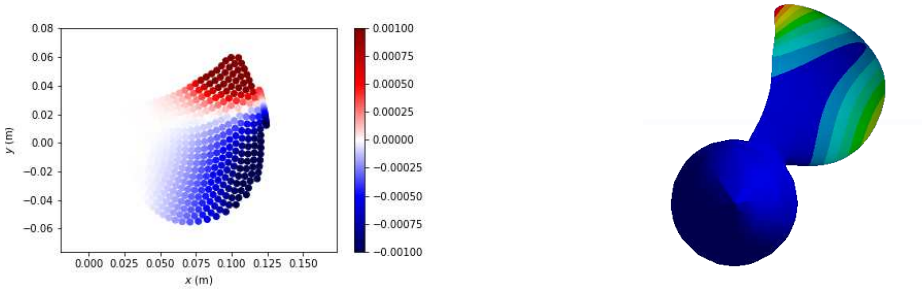
TABLE 6.5: Potsdam Propeller Test Case Modal Frequencies

Mode	Ansys Frequency (Hz)	Python Frequency (Hz)	Error
1	1130	1122.18	0.7%
2	1885.5	1693.7	13%
3	2726.1	2635.33	3%
4	3290	3844.9	16%
5	4046.3	3911.9	3%
6	4559.6	4764.77	4%



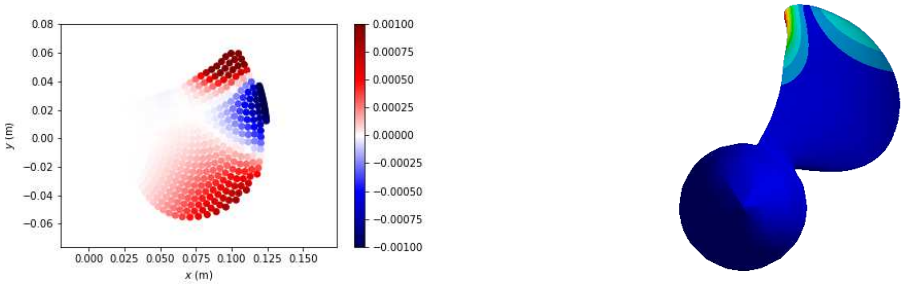
(A) Python model (B) Ansys model

FIGURE 6.20: PPTC propeller vibration mode 1



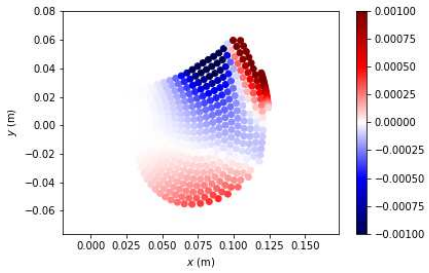
(A) Python model (B) Ansys model

FIGURE 6.21: PPTC propeller vibration mode 2

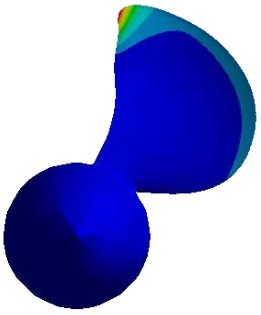


(A) Python model (B) Ansys model

FIGURE 6.22: PPTC propeller vibration mode 3

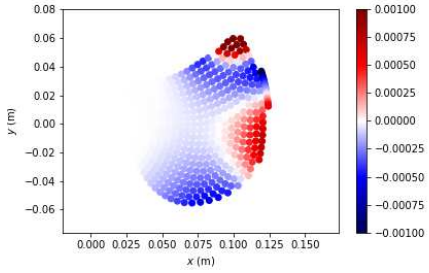


(A) Python model

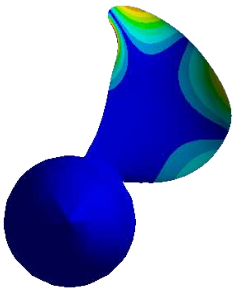


(B) Ansys model

FIGURE 6.23: PPTC propeller vibration mode 4

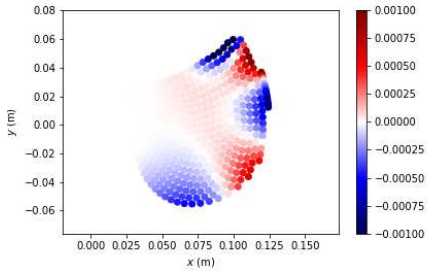


(A) Python model

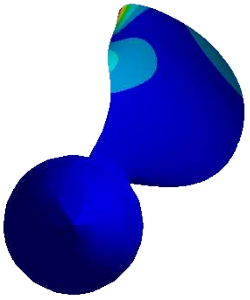


(B) Ansys model

FIGURE 6.24: PPTC propeller vibration mode 5



(A) Python model



(B) Ansys model

FIGURE 6.25: PPTC propeller vibration mode 6

From a modal analysis perspective, using the plate model gives a reasonably good approximation when compared to the full 3D Ansys model. This gives good confidence that the mass and stiffness matrices are somewhat correct. Since the modal frequencies are well captured, any excitation of these modes by an external force will be captured by the plate model in the same manner as the 3D model.

6.3.5.2 Stress Modelling

The stress distribution across the blade is informative to the propeller designer as regions of high stress can cause damage and fatigue issues to the blade. High stresses should be designed against by choice of material and geometry properties such as thickness distribution and chord.

To model stress using the plate formulation, the deflection must be obtained either from static deflections using equation 6.12 or dynamic deflections from the $HHT - \alpha$ method described in section 6.5.

The stress is computed using equation 6.32.

$$\sigma = -zcBw \tag{6.32}$$

Where z is the distance from the plate mid plane to the face i.e 0.5 thickness, w is the deformation vector for the 3 degrees of freedom for each node in the element. B is the stress matrix described in 6.23 and c is the material matrix.

σ consists of the stress in the x-direction σ_{xx}, y-direction σ_{yy} and the shear stress σ_{xy}

This gives the elemental stress, the global stress matrix is obtained in the same manner as the global stiffness matrix.

When applied to the 1m square plate example this gives the stress distribution shown in figure 6.26.

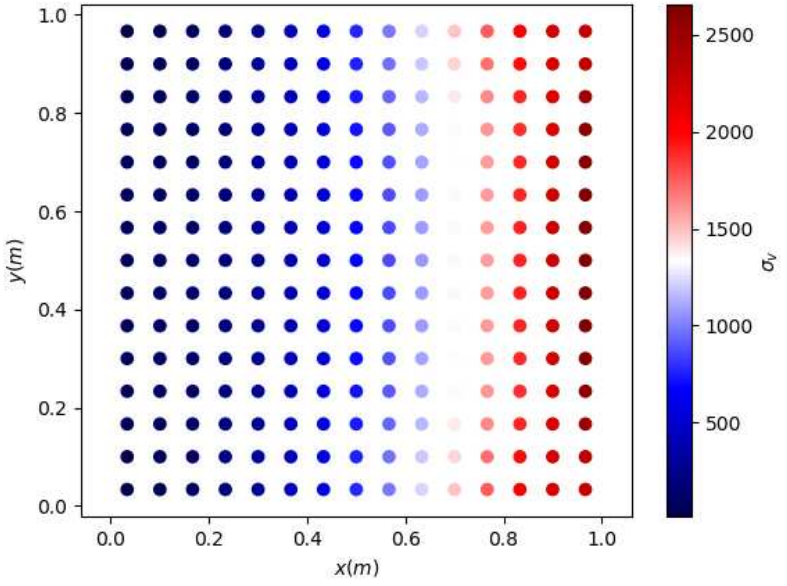


FIGURE 6.26: Von Mises Stress Distribution of 1m x 1m plate

The stress distribution shown is the Von Mises Stress distribution. This is often used as the Von Mises yield criterion are stated for the material. The Von Mises Stress is given by equation 6.33.

$$\sigma_v = \sqrt{\sigma_{xx}^2 - \sigma_{xx}\sigma_{yy} + \sigma_{yy}^2 + 3\sigma_{xy}^2} \tag{6.33}$$

The stress distribution on the plate can be compared to distribution given using Ansys shown in figure 6.27.

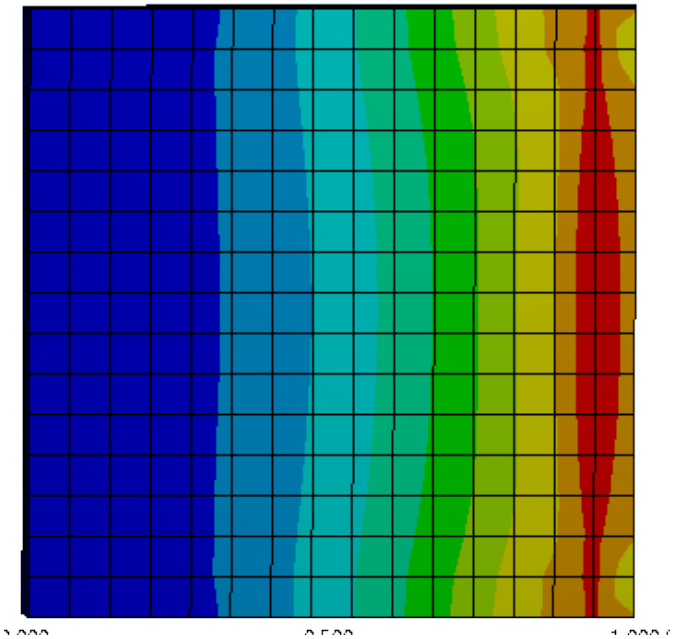


FIGURE 6.27: Von Mises Stress Distribution of 1m x 1m plate using Ansys

From these figures it can be seen that the stress distribution compares well between the two models and the maximum stress is equal between the two models. This gives a good indication that the stress is being analysed correctly within the plate model.

6.3.5.3 Wet modal Analysis

The modal frequencies change when the solid is submerged in a fluid. This is due to the additional mass on the blade due to the fluid. The additional mass adds to the global mass matrix thus changing the eigen frequencies shown in equation 2.19.

To compute the wet modal frequencies and shapes, the blade is submerged in a domain of water as shown in figure 6.28. The domain is set spherical and as close to the radius of the blade. This is to ensure the numerical stability as larger domains are more likely to obtain spurious modes.

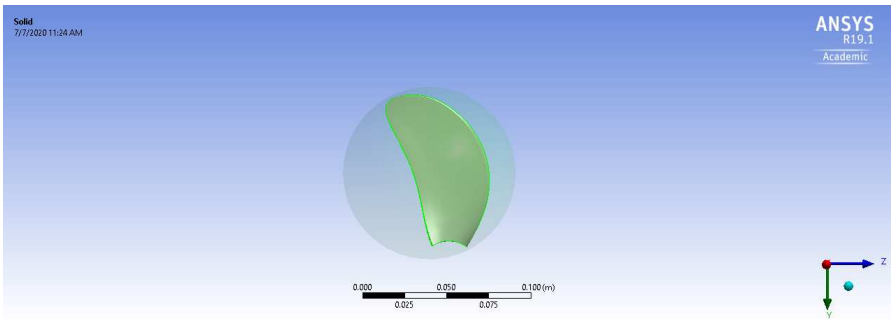


FIGURE 6.28: HMRI propeller in sphere of water for wet modal analysis.

To effectively compute the wet modes, additional boundary conditions must be defined. These are: defining the material of the sphere as water, defining the sphere as an acoustic region, the propeller is then defined as a structural region. These two conditions tell Ansys that the structural region is the solid and to perform the modal analysis on this region. The sphere acts as a fluid which the energy can transfer to. A fluid-solid interface surface is defined on the surface of the propeller, this is to define the surface which connects the fluid region and solid region. The root of the blade is set to have zero displacement. The outer circumference of the sphere has a defined boundary condition. This can either be radiation or absorption boundary conditions. The radiation boundary condition treats the boundary as if it goes to infinite and the ratio of the pressure and outward normal velocity is equal to $Z = \rho_0 C_0$, where ρ_0 is the fluid density and C_0 is the speed of sound. The absorption boundary condition absorbs the pressure wave to ensure no pressure wave are reflected back.

The wet vibration frequencies for a series of configurations are shown in table 6.6.

TABLE 6.6: Wet Modal frequencies for the HMRI propeller

Mode number	ConfigA (Hz)	Config B	C	D	E	F	G
1	1150.59	671.282	671.28	667.8	667.02	705.4	671.2
2	2512.41	1716.71	1716.7	1707.7	1704.8	837.7*	1716
3	3693.32	2686.6	2686.6	2667.75	2663.86	1132.7*	2686.6
4	4590	3497.1	3497.1	3476.5	3474.1	1132.7*	3497.14
5	5203	4210.8	4210.87	4186	4181.9	1132.7*	4210.88
6	6909	5208.03	5208.02	5170.8	5136	1398*	5208.03

* erroneous mode The 1st 6 modes for each configuration are shown in table 6.6. Configuration A are the dry modes i.e the modes in a vacuum. Configuration B used an absorption surface as an outer boundary condition with using mesh 1. Configuration C uses a radiation boundary condition using mesh 1. Configuration D uses an absorption boundary condition with mesh 2 with configuration G using a radial boundary using mesh 2. Configuration E uses an absorption boundary

condition with a low absorption coefficient. Finally Configuration F uses a large domain with an absorption configuration. A summary of the configurations is shown in table 6.7. Where a is the absorption coefficient.

TABLE 6.7: Wet Modal frequencies for the HMRI propeller

Configuration	Mesh size (No. elements)	Boundary Condition	Domain radius (m)
A	5000	N/A	N/A
B	18000	absorption a = 1	0.1
C	18000	radiation	0.1
D	100000	absorption a = 1	0.1
E	18000	absorption a = 0.1	0.1
F	104000	absorption a = 1	1
G	100000	radiation	0.1

It is clear that the wet modal frequencies are much reduced when compared to the dry frequencies which is in agreement with literature Kim et al. (2018), Chen et al. (2017).

It is important to note that only the frequencies change with the addition of the fluid domain. The mode shapes are the same as the mode shapes shown in figures 6.29a to 6.29n. This is because the local additional mass of the fluid is distributed uniformly across the blade. The potential and kinetic energy within the model do not change when moving from a vacuum to water Amabili (1996).

6.3.6 Added Mass Modelling

The added mass due to the fluid is modelled by treating each element as a rectangular plate. Analytical approximations are available for simple 3D bodies in infinite fluids as shown in Paidoussis (2014) Blevins (1990).

The added mass of a ellipse plate in heave is given by equation 6.34.

$$m_{added} = \rho_{fluid} a b^2 K1 \tag{6.34}$$

Where ρ_{fluid} is the density of the fluid, a is half the length of the plate, b is half the breadth of the plate and $K1$ is a constant based on the ratio of $\frac{b}{a}$. The relationship of b, a and K1 is found in Table 6.8.

TABLE 6.8

$\frac{b}{a}$	K1	K
0	4.1846	-
0.1	4.1228	0.8033
0.2	3.9874	0.7398
0.3	3.8202	0.6713
0.4	3.6404	0.6067
0.5	3.4588	0.5489
0.6	3.2819	-
0.7	3.113	-
0.8	2.9538	-
0.9	2.8051	-
1.0	2.6667	0.3556

Furthermore, the added mass moment of inertia of a 3D body must be accounted for. This again uses an elliptical plate to model the added mass at each element. The added mass moment of inertia is given by equation 6.35

$$m_{addedI} = \rho_{fluid} a^3 b^2 K \tag{6.35}$$

Where K is given by the values in table 6.8. It is important to note that the values of $K1$ and K are bounded in the analysis tool. That is to say if the ratio of b and a is above the maximum, then the maximum value of $K1$ or K is used.

Each component of the added mass is applied to a local added mass matrix which takes the form:

$$\begin{bmatrix} m_{added} & 0 & 0 & m_{added} & 0 & 0 & m_{added} & 0 & 0 & m_{added} & 0 & 0 \\ 0 & m_{addedI} & 0 & 0 & m_{addedI} & 0 & 0 & m_{addedI} & 0 & 0 & m_{addedI} & 0 \\ 0 & 0 & m_{addedI} & 0 & 0 & m_{addedI} & 0 & 0 & m_{addedI} & 0 & 0 & m_{adde} \end{bmatrix} \tag{6.36}$$

On the third row the added mass inertial term is changed to represent the change in inertial direction so the major and minor radii have been switched.

The new global mass matrix now takes the form $[M] = [M] + [M_{added}]$. This is now applied to the HMRI propeller geometry using the same structural properties as shown in table 6.3. The wet modal frequencies are shown in table 6.9 and compared with the modal frequencies of configuration B.

TABLE 6.9: Wet Modal frequencies for the HMRI propeller comparison between Ansys model and Plate model

Mode number	Plate Dry Modes (Hz)	Config B	Plate Wet Modes	Error
1	1094.2	671.282	704.04	5%
2	2541.76	1716.71	1596.1	7%
3	4130.24	2686.6	2494.25	7%
4	5002.45	3497.1	3177.33	10%
5	7084.78	4210.8	4126	2%
6	8424.8	5208.03	5228.64	0.3%

It can be seen that the modal frequencies from the Ansys model and the Python model are very close with a maximum error of 10 percent. This is achieved using a simple approximation for added mass with very little additional computational expense.

This level of accuracy can also be achieved when studying the PPTC propeller. The Ansys model is compared to the Plate model in table 6.10

TABLE 6.10: Wet Modal frequencies for the PPTC propeller comparison between Ansys model and Plate model

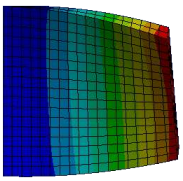
Mode number	PPTC Ansys Mode Frequencies	Plate Wet Modes	Error
1	679.49	702.04	3%
2	1260	1129.06	11.5%
3	1773.6	1944.7	11%
4	2568	2589.9	0.8%
5	2876.4	3061.8	7%
6	3837.3	4171.0	8%

6.3.7 Effect of Twist

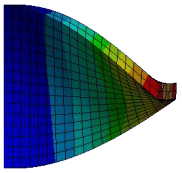
A key component of the design of a propeller blade is the pitch distribution. Using a Plate to model the structures of a propeller, by definition, removes all twist. To understand the impact of twist on the modal frequencies, the rectangular plate has been used with various levels of twist as shown in figure 6.29. The angles chosen are 45^o and 90^o as the maximum twist on a propeller blade is approximately 50^o-60^o root to tip, the chosen angles are slightly high but exemplify the issue of using plate theory.

(A) 45^o twist plate

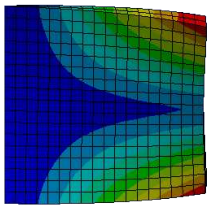
(B) 90^o twist plate



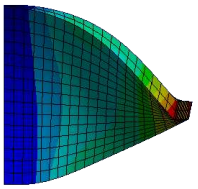
(C) 45^o twist plate Mode 1



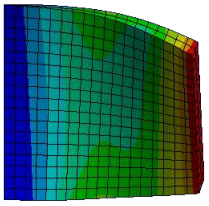
(D) 90^o twist plate Mode 1



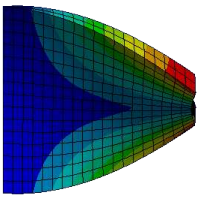
(E) 45^o twist plate Mode 2



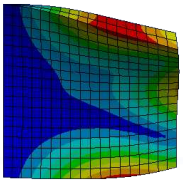
(F) 90^o twist plate Mode 2



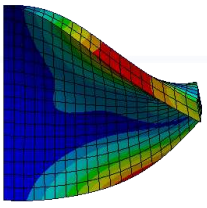
(G) 45^o twist plate Mode 3



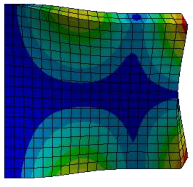
(H) 90^o twist plate Mode 3



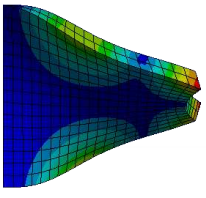
(I) 45^o twist plate Mode 4



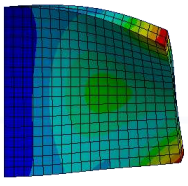
(J) 90^o twist plate Mode 4



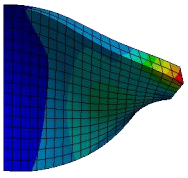
(K) 45^o twist plate Mode 5



(L) 90^o twist plate Mode 5



(M) 45^o twist plate Mode 6



(N) 90^o twist plate Mode 6

FIGURE 6.29: Twisted Plate mode shapes

It is shown that the levels of twist, albeit quite extreme twist, can have an impact on the mode shapes and frequencies. Table 6.11 shows the frequencies when the plate is twisted. With the addition of twist, the 1st bending mode reduces significantly, indicating that the stiffness is reduced. This change is because the 1st mode shape is still acting in the pure bending direction but there is more mass concentrated to encourage this bending mode.

TABLE 6.11: Modal frequencies of twisted Plate from Ansys

Mode number	45^o twist	90^o twist
1	78.767	68.915
2	272.14	230.03
3	350.51	357.8
4	560.42	529.67
5	698.62	704.21
6	700.51	709.83

Plate elements do not account for degrees of freedom in the x and y directions as a way to reduce the problem. In the case of a highly twisted blade where the plate model may not be sufficient to capture the motion correctly 3D elements can be used.

6.4 3D Solid Elements

To better account for twist, 3D solid elements have also been used. This can be done by including a Z direction to the mesh and changing the element type from a 2D plate element to a 8 node 3D brick element. The same principals for construction of the plate elements apply to the construction of the 3D elements.

The shape functions for the 8 node 3D element are:

$$
\begin{aligned}
N_1 &= \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta) \\
N_2 &= \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta) \\
N_3 &= \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta) \\
N_4 &= \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta) \\
N_5 &= \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta) \\
N_6 &= \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta) \\
N_7 &= \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta) \\
N_8 &= \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)
\end{aligned}
$$

Where ξ , η and ζ are local coordinates. Each node has degrees of freedom in the u, v and w directions. Therefore every element has 24 degrees of freedom.

The strain matrix is now defined for each node as:

$$
B_i = \begin{bmatrix}
\frac{\partial N_i}{\partial x} & 0 & 0 \\
0 & \frac{\partial N_i}{\partial y} & 0 \\
0 & 0 & \frac{\partial N_i}{\partial z} \\
0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\
0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\
\frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\
\frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0
\end{bmatrix} \tag{6.37}
$$

The elemental stiffness matrix is then given as:

$$
k_e = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} B^T c B det|J| d\xi d\eta d\zeta \tag{6.38}
$$

and the elemental mass matrix is given by:

$$
k_e = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \rho N^T N det|J| d\xi d\eta d\zeta \tag{6.39}
$$

Where **N** is the matrix of shape functions, J is the Jacobian extended to include the 3rd dimension and **c** is the material constant for 3D solids as shown in 6.40.

$$c = \begin{bmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{12} & c_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \tag{6.40}$$

Where $c_1 1 = \frac{E(1-\nu}{(1-2\nu)(1+\nu}$ $c_1 2 = \frac{E\nu}{(1-2\nu)(1+\nu}$ and $G = \frac{E}{2(1+\nu}$.

The elemental mass and stiffness matrices are again integrated numerically using the Gaussian integration scheme, however this is extended to 8 nodes with there being 2 nodes in each direction.

This 8 node element has been applied to the square plate using a 15x15x2 elements grid, giving the modal frequencies shown in table 6.12. Here it can be seen that the 8 node element frequencies are higher when compared to the base case. This is a well documented case where this element type is over-stiff in bending which leads to the higher frequencies Bathe (2016).

To correct this problem a 20 node 3D element can be used. Here, additional nodes are created between the original 8 nodes as shown in figure 6.31. The main difference here is the Shape functions are no longer linear but are quadratic.

The shape functions are different depending on the node. The shape functions are defined such that they are equal to 1 at the corresponding node and 0 at every other node. The shape functions are quadratic meaning they can obtain a greater level of accuracy for a given mesh size when compared to the linear shape functions.

$$\begin{aligned} N_{1,2,3,4,5,6,7,8} &= \frac{1}{8}(1+\xi_j\xi)(1+\eta_j\eta)(1+\zeta_j\zeta)(\xi_j\xi+\eta_j\eta+\zeta_j\zeta-2) \\ N_{10,12,14,16} &= \frac{1}{4}(1-\xi^2)(1+\eta_j\eta)(1+\zeta_j\zeta) \\ N_{9,11,13,15} &= \frac{1}{4}(1-\eta^2)(1+\xi_j\xi)(1+\zeta_j\zeta) \\ N_{17,18,19,20} &= \frac{1}{4}(1-\zeta^2)(1+\xi_j\xi)(1+\eta_j\eta) \end{aligned} \tag{6.41}$$

The formulation of the elemental mass and stiffness matrices can be obtained in the same manner as previous however, as there are now 20 nodes compared to 8 the computational cost increases considerably for the same mesh size. For a mesh of size 1X1X2 an example grid is shown in figure 6.30

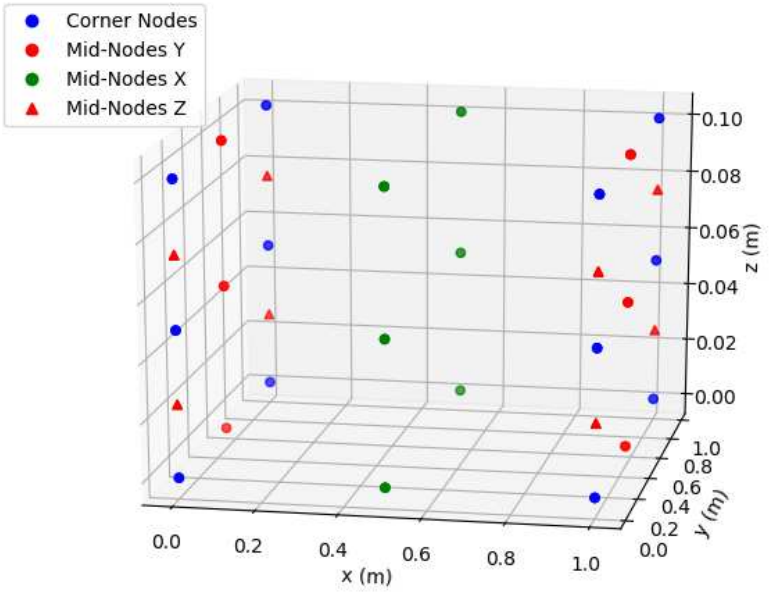


FIGURE 6.30: Example of a 1X1X2 grid using 20 node 3D elements

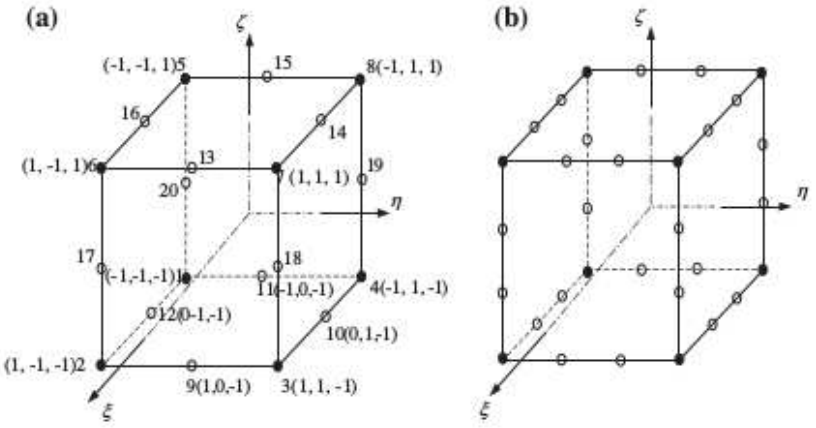


FIGURE 6.31: Schematic of 20 node 3D element

The modal frequencies for the 20 node element are compared to the 8 node elements for the same mesh size. It can be seen that the 20 node elements perform far better when compared to the 8 node elements.

TABLE 6.12: Modal frequencies of Plate when using 8 node elements and 20 node elements

Mode number	3D 8 node elements	3D 20 node elements
1	92.39	83.78
2	203.95	196.64
3	533.1374	491.18
4	544.617	530.4
5	676.822	621.43
6	738.33	690.2

Despite the 20 node 3D element type performing well in terms of accuracy, the computational cost can be significant. The computational time to calculate the modal frequencies for the plate elements, the 8 node 3D elements and 20 node 3D elements were computed for the same number of elements for each. The computational comparison is shown in figure 6.32 with the y-axis presented in a logarithmic plot.

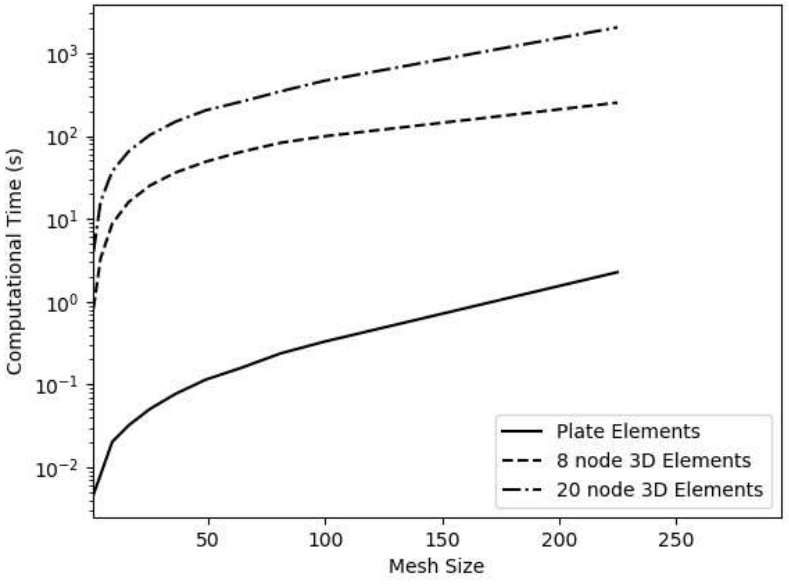


FIGURE 6.32: Computational cost comparison between plate elements, 8 node 3D elements and 20 node 3D elements

From 6.32 it can be seen that the Plate elements have a significant computational saving of 2 orders of magnitude for the 8 node element and approximately 3 orders of magnitude over the 20 node 3D element. This gives a strong indication that the plate elements are a good compromise between accuracy and computational cost. It is also interesting to note that the change in computational cost differs between the different element types. This can be explained when the code is analysed line by line using a

line profiler. The line profiler analyses the time taken for each line of a function to be run. This is done for the main FEA function with the results shown in figure 6.33.

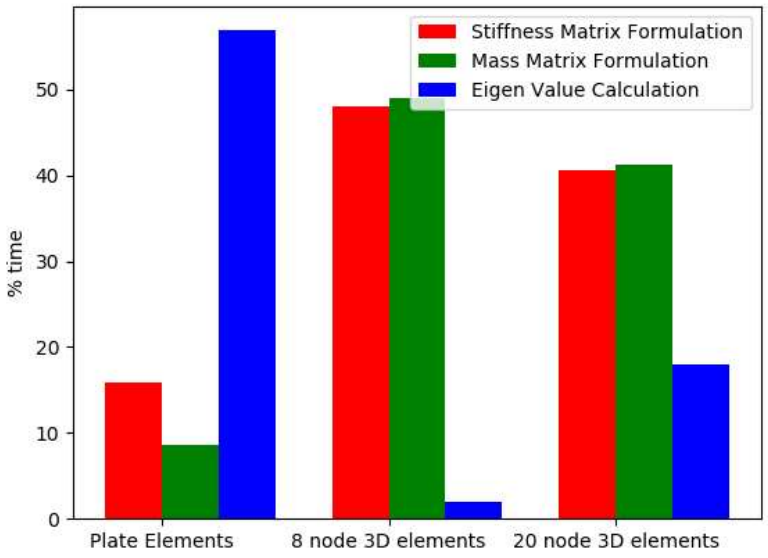


FIGURE 6.33: Percentage time each part of the FEA formulation takes

The results of the line profiler indicate that for the Plate elements the majority of the time is taken in the eigenvalue calculation. This is because there are only 4 nodes per element therefore the shape function matrices will be small, moreover only 4 Gauss integration points are used therefore saving computation.

This changes when the 3D elements are used. For the 8 node elements the Eigenvalue calculation only requires 2% of the overall compute time. This is because the number of Gauss integration points has increased from 4 points for the 2D case to 8 points for the 3D case. The addition of the 3rd dimension also increases the shape function matrices for each element. The 20 node element requires more time using the Eigenvalue solver compared to the 8 node elements. This is due to the fact that each element has 20 nodes, hence the global mass and stiffness matrices will be much larger for the same element size causing greater computation to solve the eigenvalues. Despite this, the number of Gauss integration points remains 8 for the 20 node element hence why the relative time solving the local mass and stiffness matrices are lower.

6.5 HHT - α

To calculate the dynamic response to a load, a numeric integration scheme must be applied. For this case the Hilber-Hughes-Taylor-α scheme is used. The

Hilber-Hughes-Taylor-α ($HHT-\alpha$) is a generalized method of the $Newark-\beta$ numerical integration method which is an unconditionally stable, implicit scheme. Full explanation of the $HHT-\alpha$ method can be found in Hughes (1983). This method is chosen due to its wide use in FSI literature Bathe (2016), Liaghat et al. (2014).

The dynamic system of the blade is modelled by:

$$[M]\ddot{x} + [C]\dot{x} + [K]x = f(t) \tag{6.42}$$

Where M is the mass matrix C is the damping matrix K is the stiffness matrix, x is the nodal points and $\dot{}$ represents the derivative with time.

For the HHT-α method the equation of motion 6.42 is modified such that.

$$M\ddot{x}_{i+1} + (1-\alpha)C\dot{x}_{i+1} + \alpha C\dot{x}_i + (1-\alpha)Kx_{i+1} + \alpha Kx_i = (1-\alpha)f_{i+1} + \alpha f_i \tag{6.43}$$

Here, each node's position is defined at a time step i with time moving to the next time by time step h such that $t_{i+1} = t_i + h$. Therefore the subscript i represents the relative time position. The parameter α represents numerical lag in the damping, stiffness and forces.

The HHT-α method is an extension of the Newmark-β method which used the finite difference approximation:

$$\begin{aligned} x_{i+1} &= x_i + h\dot{x}_i + h^2[(\frac{1}{2}-\beta)\ddot{x}_i + \beta\ddot{x}_{i+1} \\ \dot{x}_{i+1} &= \dot{x}_i + h[(1-\gamma)\ddot{i} + \gamma\ddot{x}_{i+1} \end{aligned} \tag{6.44}$$

Where γ and β are constants which control the stability of the solution. Using conditions:

$$\begin{aligned} 0 \leq \alpha &\leq \frac{1}{3} \\ \beta &= \frac{(1+\alpha)^2}{4} \\ \gamma &= \frac{1}{2} + \alpha \end{aligned} \tag{6.45}$$

The solution will be 2nd order accurate and unconditionally stable Gavin (2018).

Substituting 6.44 into 6.45 gives:

$$\begin{aligned}[M + h(1-\alpha)\gamma C + h^2(1-\alpha)\beta K]\ddot{x}_{i+1} = [h(1-\alpha)(1-\gamma)C + h^2(1-\alpha)(\frac{1}{2} - \beta)K]\ddot{x}_i \\ + [C + h(1-\alpha)K]\dot{x}_i \\ + Kx_i + (1-\alpha)f_{i+1} + \alpha f_i\end{aligned} \tag{6.46}$$

Which can then be numerically solved to computed the dynamic response of the structure to a load.

6.5.1 Verification

To ensure the HHT-α is correctly implemented, a small problem implemented and verified against the solution given in Gavin (2018).

This problem was a simple linear structural model shown in Figure 6.34. This will have a simple 3X3 mass, stiffness and damping matrix.

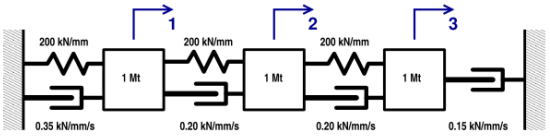
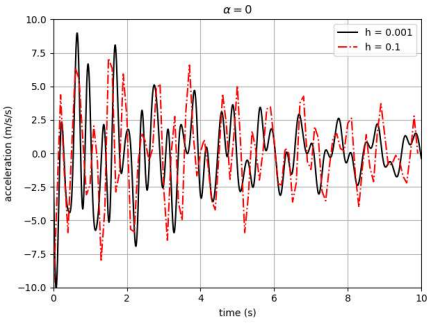


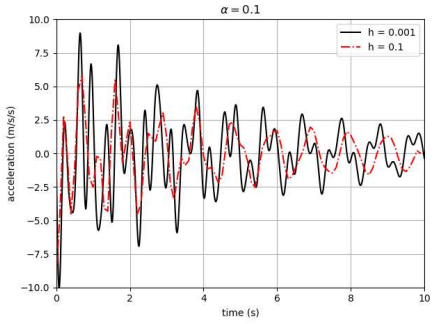
FIGURE 6.34: Simple linear model for HHT verification

The initial conditions were given as: $x = [0, 0, 0]$,$\dot{x}(0) = [1, 1, 1]$ and $\ddot{x}(0) = [0, 0, 0]$. A time step of 0.1s and 0.001s were used as well as α values of 0 and 0.1.

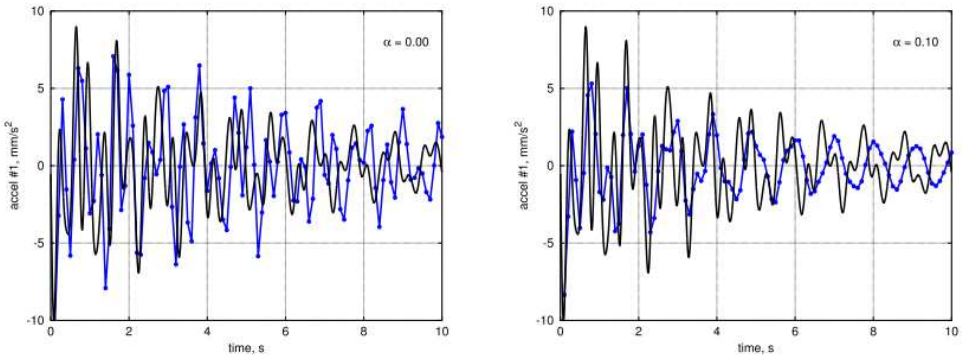
The results computed by Gavin (2018) are shown in figure 6.35 and compared with the implemented method.



(A) $\alpha = 0$



(B) $\alpha = 0.1$



(C) Results from Gavin (2018)

FIGURE 6.35: acceleration of 1st mass with time using HHT-α method

Here it can be seen that the time history of the acceleration of the 1st mode match well when comparing the implemented method and the method by Gavin (2018). This gives confidence that the $HHT - \alpha$ method has been implemented correctly.

6.6 Chapter Summary

In this chapter the use of a Plate model to represent the propeller blades has been developed and the implementation procedure has been discussed.

The code has been verified against the industry standard code Ansys and the code gives excellent agreement when comparing the results of a flat plate. This is capable of obtaining the deformation, stress and modal frequencies well.

The use of the plate model has also been used to model propeller blades. This has been shown to give reasonably close results to the full 3D structural model. However, issues can arise when a highly twisted blade is used.

The computational cost between using 3D elements and plate elements has been discussed with the plate elements displaying a significant computational saving.

By approximating the added mass effect on each plate element, the modal frequencies of a wet blade can be achieved. This gives a reasonably good approximation of lowering the modal frequencies.

The use of plate model elements have been shown to be a computationally efficient method of obtaining the structural properties of the blade and will be used in the fluid-structure interaction model.

Chapter 7

Fluid-Structure Interaction Model

7.1 Introduction

Changes in inflow to the propeller will change the imposed time varying load on the blades. Moreover, different propeller geometries and material properties will cause the blade to deform differently. The change in deformation behaviour is required using a fluid-structure interaction model which is to be computationally efficient.

This chapter introduces the implementation of the fluid-structure interaction tool. The algorithm is discussed for one-way and two-way coupling. Results from using the tool are shown for various numeric and physical parameters changes. Finally, some downfalls and improvements to be made are discussed.

7.2 Coupling Algorithm

To couple the BEMT hydrodynamic model and the plate FEA model, the hydrodynamic and structural conditions must be defined. Firstly, the unsteady inflow to the propeller must be defined along with the geometry of the propeller. The operating conditions can also be selected.

The structural properties can then be chosen to define the elastic modulus and the density of the material. A flow chart of the coupling algorithm is shown in Figure 7.1.

Secondly the propeller geometry is defined as well as the wake data. In this case the wake data is taken from CFD simulations of the hull. The user can then check if turbulent fluctuations are to be generated using the technique shown in Section 5.5.1. If this is selected then the wake with velocity fluctuations are used as an input to the BEMT code, otherwise the RANS velocity data is used. The C_l v α data base is called here to be used within the BEMT iterations.

The BEMT code outputs the thrust distribution along the blade, however this is only the distribution along the blade line and not the chordwise distribution. This is generated using the method described in Section 3.3 were the C_p curves from the 2D foil simulations are used.

The load distribution is then applied to the plate structural model to obtain the blade nodal deformations, velocities and accelerations. The load and nodal deformations are then stored for future analysis. If the time step is the final time step, the simulation is complete.

Otherwise, the user has previously stated if the coupling is one-way or two-way. If the coupling is one-way the BEMT is called again at the next time step and the procedure repeated until the final time step. If two-way coupling is used the blade geometry is updated then the BEMT called again and the procedure repeated until completion.

The blade geometry is updated by calculating the additional pitch of the blade due to the deformation. Although the rake will be changed due to the blade bending this is not accounted for in the BEMT formulation.

7.2.1 Damping

Damping causes the dynamic motion of the system to reduce with time. There are two main contributors to damping: hydrodynamic and material damping. The hydrodynamic damping has not been investigated in this study and is an aspect that must be considered and efficiently modelled in future work, particularly within a moving fluid. The damping of the material has in fact been accounted for and can be changed within the design environment.

The damping matrix is defined in equation 7.1.

$$C = \alpha_d[M] + \beta_d[K] \tag{7.1}$$

Where α_d is defined as $\frac{\omega_0}{\omega_1}$ where ω_n is the blade natural frequency at the nth mode. β_d is defined in equation 7.2.

$$\beta_d = \frac{(1-\alpha_d)\zeta}{\omega_1 - \alpha_d\omega_0} \tag{7.2}$$

Here, ζ is the damping ratio. These are known as Rayleigh damping coefficients and have been used for structural applications for many years Song and Su (2017).

The damping ratio term ζ can be changed to obtain a more accurate structural response. Although this may be acceptable for a design tool application far more work

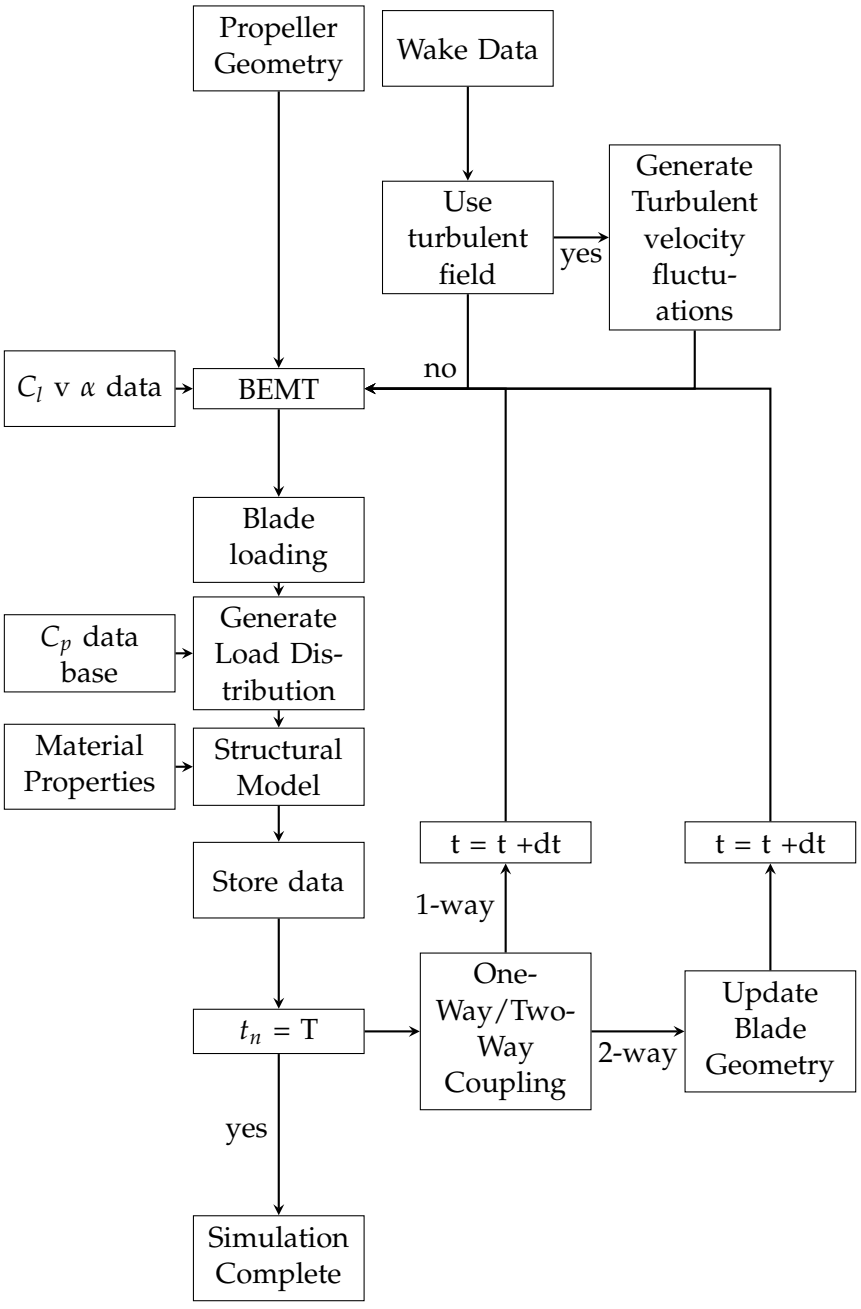


FIGURE 7.1: Flow Chart of FSI algorithm

is required in identifying proper damping behaviour for the structure in moving water. Furthermore, the damping properties of non-homogeneous composite materials often differ from metallic materials . Therefore, a more robust and comprehensive damping model will be required.

7.2.2 Force Modelling

To model the force distribution of the blade, the Cp curves developed from Section 3.3 have been used. Given the bounds of the camber, thickness, Reynolds number, angles

of attack and the weighting for each blade section, a pressure coefficient distribution is obtained.

The pressure distribution on the blade is obtained from the Cp curves and is shown in Figure 7.2. The pressure difference between the top surface and bottom surface is very small hence the small pressure at the leading edge.

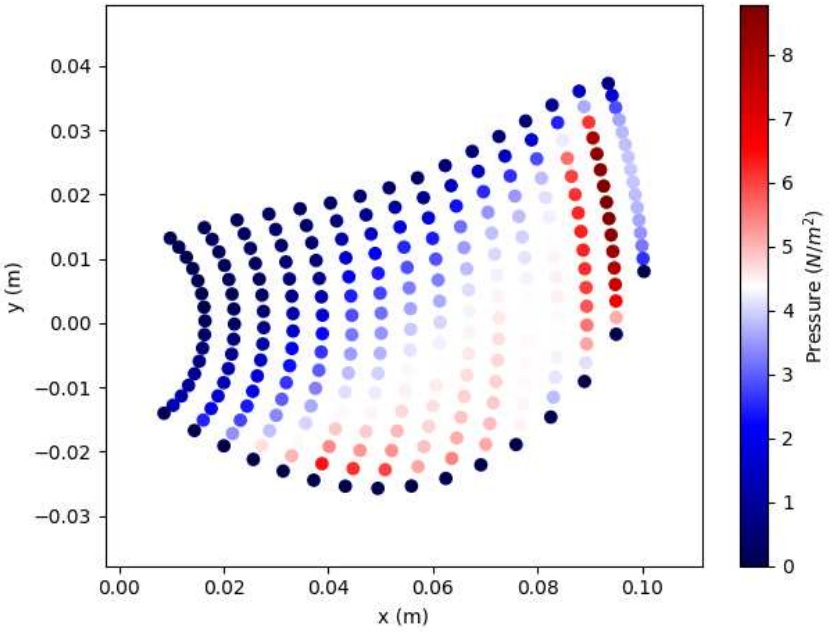


FIGURE 7.2: Pressure Distribution on HMRI propeller from BEMT

However, to use the $HHT-\alpha$ method the force must distributed on the nodes. Here it is assumed the load on each element is uniformly distributed. To obtain the nodal loads the force vector from the plate FEA formulation from section 6.3 is generated.

This gives a load distribution on the blade shown in Figure 7.3.

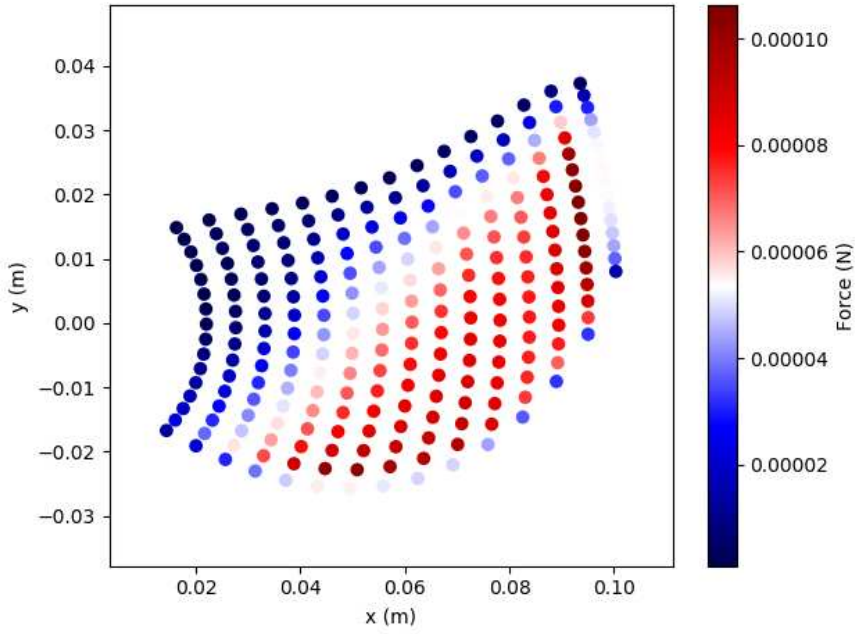


FIGURE 7.3: Load Distribution from FEA

As discussed in Chapter 6.3 the force distribution requires the calculation of element areas at every timestep, this causes the computational cost to increase. Also, the interaction with the foil database within the BEMT loop can have a considerable computational cost, especially when a high number of circumferential sections are used. This can be improved by good optimization and writing the code in a fully compiled language such as C or FORTRAN.

7.3 Numerical Investigations

To determine the applicability of the BEMT-Plate model coupling algorithm a series of numeric tests have been run to obtain the deflection of the blade and the frequency content of the deflection.

These test include the influence of one-way coupling and two way coupling, changes in physical parameters and changes in numerics.

The changes in physical parameters involves changing the drift angle the propeller is operating at, influence of turbulent fluctuations, applying to a full scale propeller and changing the material property.

Changing the numerics will also change the resolution of the results obtained. For instance changing the number of circumferential sections will change the detail of the wake the propeller will interact with, changing the timestep will change the frequency

range the blade will vibrate and changing the FEA mesh size will influence the detail of the mass and stiffness matrices.

This section will investigate the influence of these changes using the BEMT-Plate coupling tool.

7.4 Application of Tool

Firstly, the tool is applied to the several drift angles obtained in Chapter 5. To obtain the difference between the different cases, a reference case has been run. This reference case is using steel as a material with a setting of 1^o per circumferential section with no hysteresis and one-way coupling. The deflection of the blade tip is shown in Figure 7.4 with each drift angle shown also.

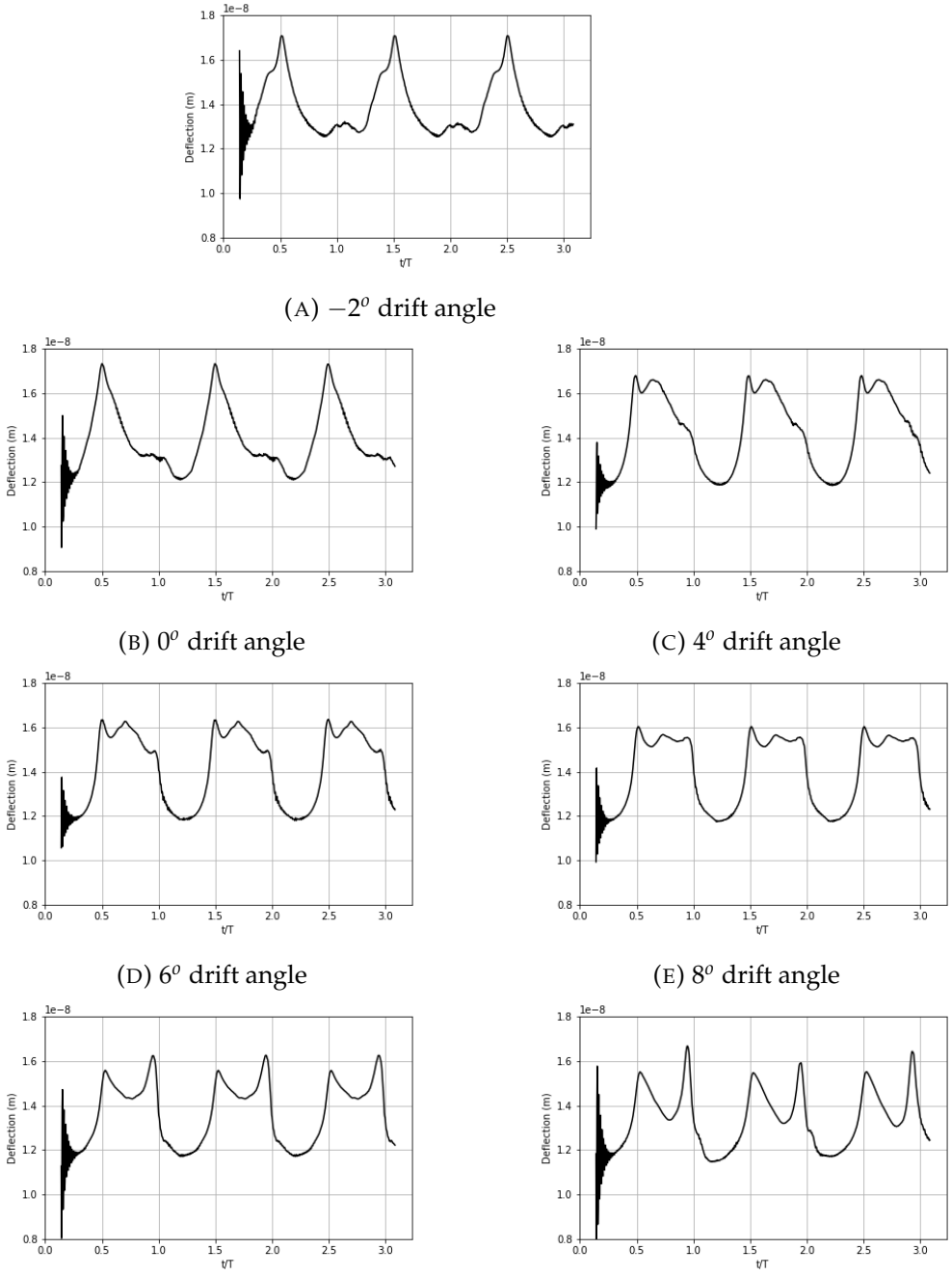
Figure 7.4 shows the blade tip deflection with time as the propeller rotates. For the first few time steps the change in deflection is quite rapid due to the propeller going from zero loading to loaded. This causes an impulse load which causes the propeller blade to vibrate at its natural frequency.

Figure 7.4b is the deflection of the blade at 0^o drift angle. It can be seen here that the blade goes from a high loading region of the wake to a low loading of the wake per cycle. Furthermore, the effect of the tangential wake is captured quite well with the rapid increase in loading occurring as the blade is on the up part of the cycle. The loading is reduced slowly as the blade cycles down with the tangential wake causing the reduction in loading.

It is also noted that there are a few discrepancies on the blade deformation where the deflection rapidly changes. This is due to convergence not being met in the BEMT solution therefore causing a sudden change in loading. In an attempt to ensure convergence is met the angle of attack from the previous time step is used as the initial guess for the current time step angle of attack. If convergence is still not met, a flag is initialised and simple finite difference solution is used to obtain the angle of attack of the current time step using the previous angle of attack and the gradient.

As the drift angle increases the shape of the deflection changes. For the 4^o drift angle a 2nd uptick in deflection can be seen with a 3rd scene in the 6^o and 8^o drift angle simulations. The change in deflection for drift angles 12^o and 16^o are similar with a rapid reduction on deflection in as the blade rotates down.

The differences between the deflections are very distinct therefore the changes caused by the velocity profiles shown in Figures 5.5 are well captured. The time history is normalised by the time for one propeller rotation,T.



(A) -2^o drift angle

(B) 0^o drift angle

(C) 4^o drift angle

(D) 6^o drift angle

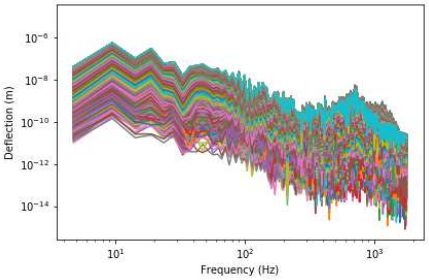
(E) 8^o drift angle

(F) 12^o drift angle

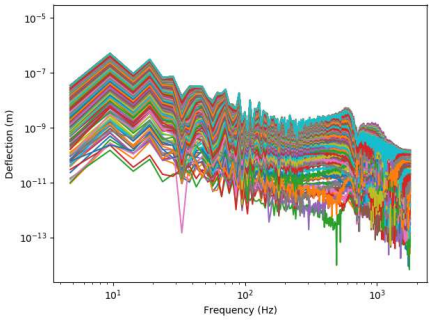
(G) 16^o drift angle

FIGURE 7.4: 1 way coupling of BEMT-Plate using steel as the material

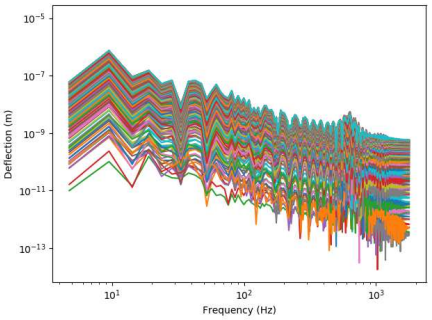
To obtain the frequency content of the deflection, the fast Fourier transformation of the deflection of all points are taken. This is shown in Figure 7.5. It can be seen that the dominant frequency occurs at approximately 10Hz for each case. This is expected as it is the blade passing frequency. Also, there are increases in energy at multiples of the passing frequency. This is in agreement with other propeller FSI studies in literature Tian et al. (2017).



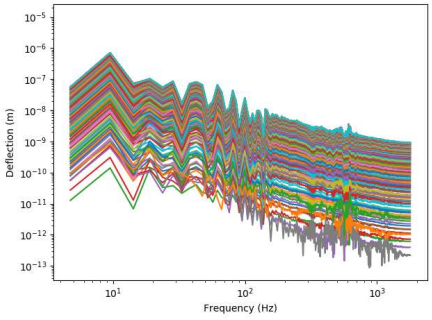
(A) -2^o drift angle



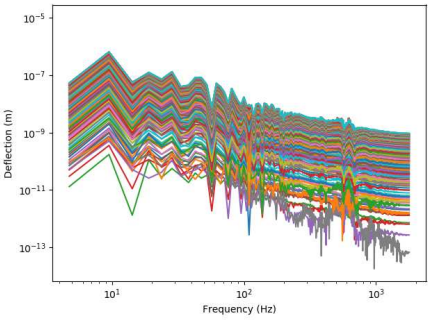
(B) 0^o drift angle



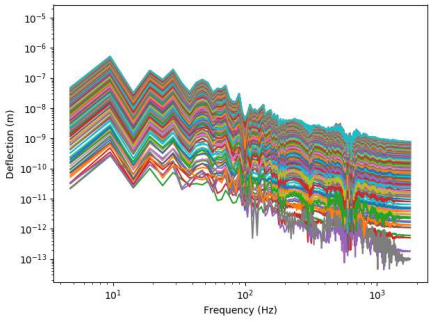
(C) 4^o drift angle



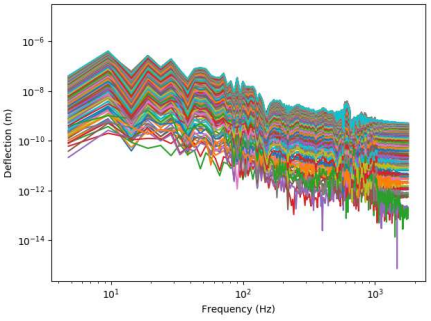
(D) 6^o drift angle



(E) 8^o drift angle



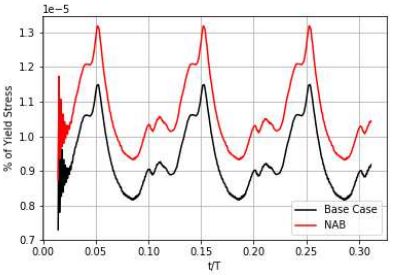
(F) 12^o drift angle



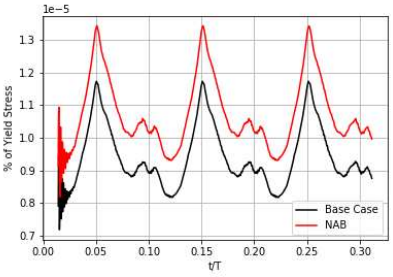
(G) 16^o drift angle

FIGURE 7.5: Frequency content of blade elements for each drift angle.

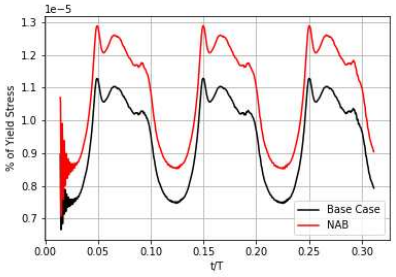
As well as the blade deformations, the designer can design against stress requirements. The stress through the blade at each time step is calculated and the maximum Von Mises stress is plotted as shown in Figure 7.6.



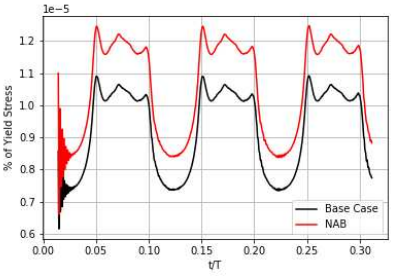
(A) -2^o drift angle



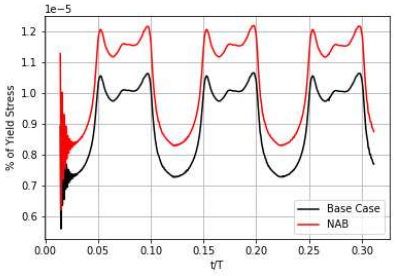
(B) 0^o drift angle



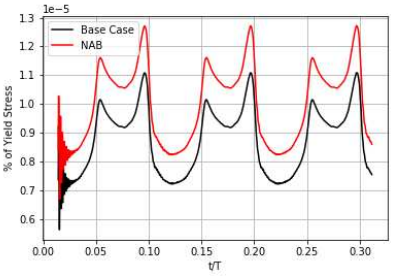
(C) 4^o drift angle



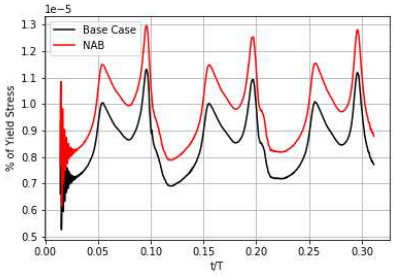
(D) 6^o drift angle



(E) 8^o drift angle



(F) 12^o drift angle



(G) 16^o drift angle

FIGURE 7.6: Steel Two-way coupling

It can be seen here that as the blade rotates the stress increases and decreases according to the deflection. The stress variation can cause fatigue issues. It is interesting to note the rapid change in maximum stress, particularly for the drift angle of 6^o. If the designer finds the stress is too large for the given blade, design changes in blade thickness can be made to ensure stresses remain below the desired value for all manoeuvres and hull appendages.

7.4.1 Numeric Changes

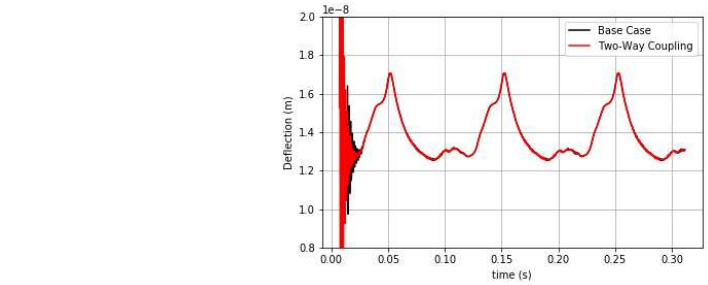
To ensure the computationally efficient tool is providing accurate and precise results a series of numeric tests have been run. These include: using one-way vs two-way coupling, changing the structural mesh size, changing the time step, and the inclusion of hysteresis.

7.4.1.1 One-way vs Two-way Coupling

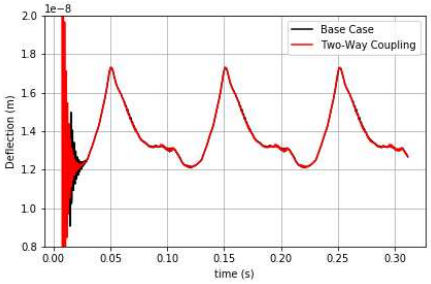
The use of the one-way and two-way coupling techniques have been studied for each drift angle. These are shown in Figure 7.7.

The change in maximum deflection going from one-way coupling to two-way coupling is shown to be very small. This is because the material is very stiff and the deformation is predominantly bending with very little twist occuring. The only geometric parameter changing in the BEMT algorithm is the local pitch as the rake and skew is not accounted for. A change in skew or rake will change the pressure distribution on the blade and therefore change the structural response.

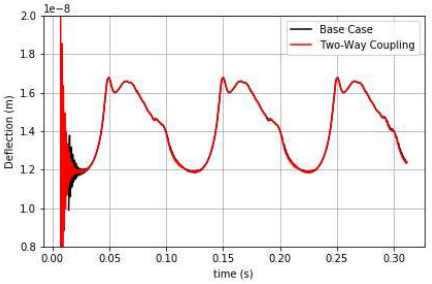
The plate model can, currently, only model metallic materials. However, composite propellers can be used where there can be strong bend-twist coupling. The use of these materials will show a more distinct change between one-way and two-way coupling techniques.



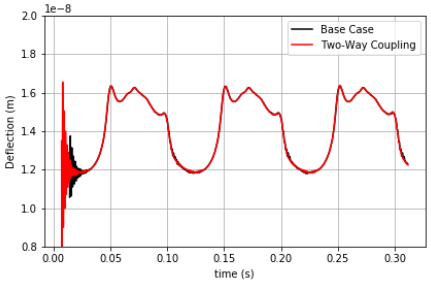
(A) -2^o drift angle



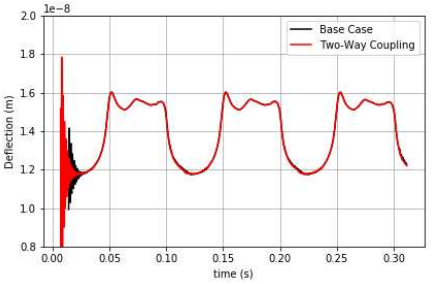
(B) 0^o drift angle



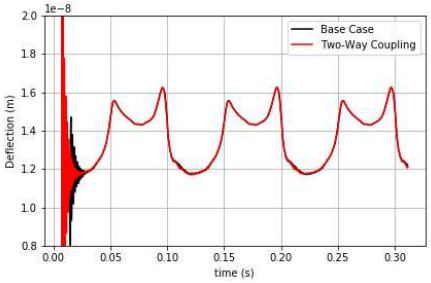
(C) 4^o drift angle



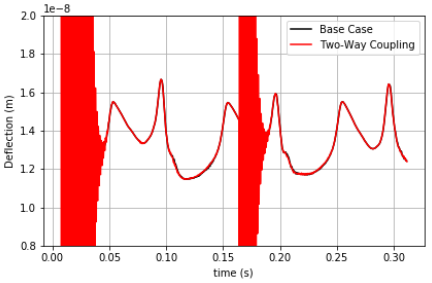
(D) 6^o drift angle



(E) 8^o drift angle



(F) 12^o drift angle

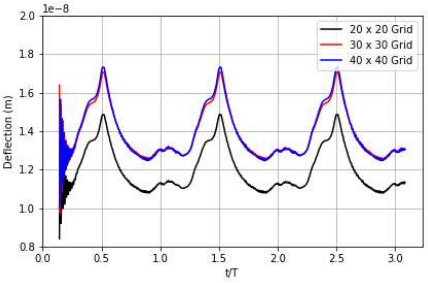


(G) 16^o drift angle

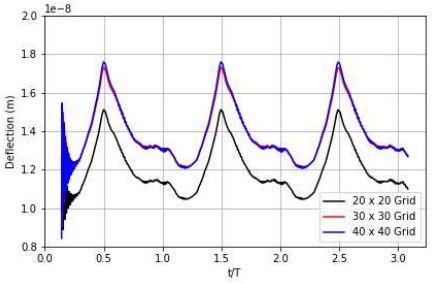
FIGURE 7.7: Deflection of steel blade comparing one-way and two-way coupling

7.4.1.2 Influence of Mesh size

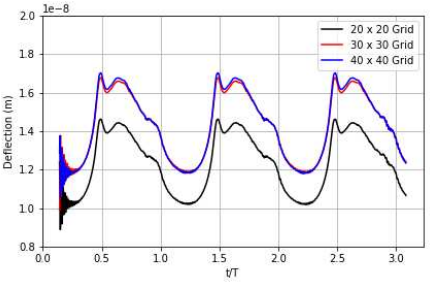
To ensure an accurate solution, the mesh convergence for the structural elements is shown in Figure 7.8. Here a 20 x 20 grid, a 30 x 30 grid and a 40 x 40 grid have been tested. It is clear that convergence is met when a 30 x 30 grid is used and the use of a 40 x 40 grid changes the deformation an insignificant amount. The 20 x 20 grid is too low resolution to give accurate deformation, although significant computational expense is saved when using the 20 x 20 grid.



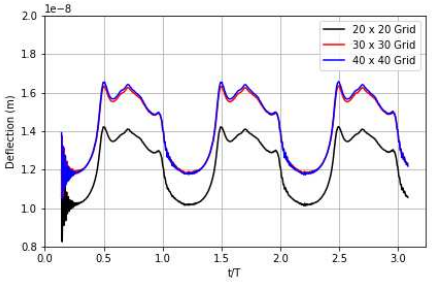
(A) -2^o drift angle



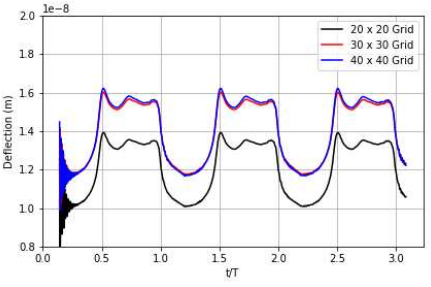
(B) 0^o drift angle



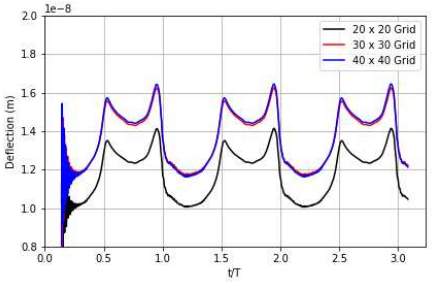
(C) 4^o drift angle



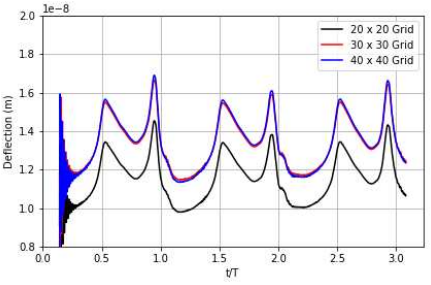
(D) 6^o drift angle



(E) 8^o drift angle



(F) 12^o drift angle

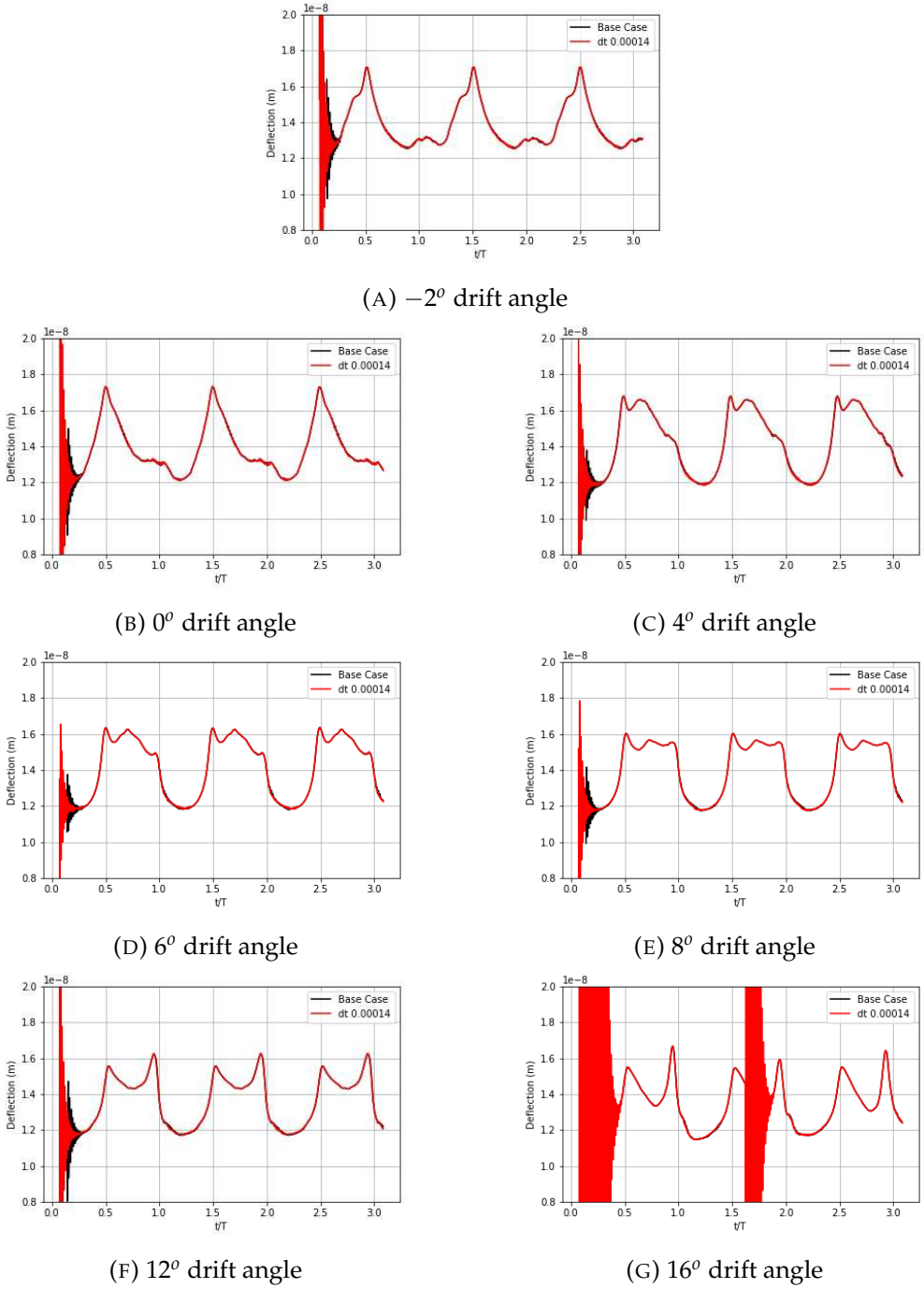


(G) 16^o drift angle

FIGURE 7.8: Deflection of steel blade with various mesh densities

7.4.1.3 Time Step

Figure 7.9 shows the influence in changing the time step. In the base case, a time step of 0.00028s was used corresponding to 1^o of propeller rotation per time step. To check convergence the time step has been halved to correspond to 0.5^o per time step.



(A) -2^o drift angle

(B) 0^o drift angle

(C) 4^o drift angle

(D) 6^o drift angle

(E) 8^o drift angle

(F) 12^o drift angle

(G) 16^o drift angle

FIGURE 7.9: Deflection of steel blade comparing time steps

The changing of the time step is shown to have no influence on the maximum deflection. Therefore, a time step of 1^o of propeller rotation is sufficient for the studied

wake. However, if the wake had particularly large circumferential velocity gradients, then a higher time step would be recommended. Again, it can be seen that large fluctuations in deformation occur at the 16^o drift angle due to the sensitivity of the BEMT convergence.

7.4.1.4 Hysteresis

The inclusion of the hysteresis model, described in Section 3.5, is studied here with the maximum deflection for each drift angle shown in Figure 7.10.

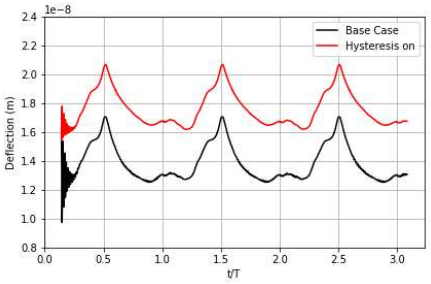
The inclusion of the hysteresis model shifts the deformation up by approximately 20%. As the blade deforms, the change in pitch and heave cause the lift curve to shift. The pitch change due to the deformation is very small with the heave being far more significant.

As the blade heaves, up the lift coefficient for a given angle of attack is lowered. Therefore, within the blade element momentum theory algorithm, a higher angle of attack is required to obtain the same lift coefficient as the run without hysteresis. This causes higher loading on the blade when the pressures from the 2D foil database are applied to the structural model causing higher deformations and stress. Although, work has been done including hysteresis with BEMT Yu et al. (2017), little work has been done when analysing the deflection of the blades.

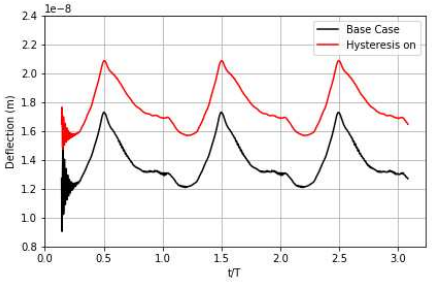
What is not included in the BEMT-Plate FSI model is a correction to the pressure distribution due to the blade section deformation. This should alter the angle of attack the blade experiences and therefore the load distribution on the blade section whilst keeping the same blade loading as the non-hysteresis model. This can be investigated and implemented in future developments.

7.4.2 Physical Parameters Changes

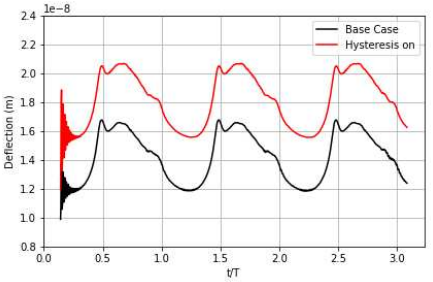
To showcase how the tool reflects various changes in the physical parameters, simulations have been performed to include: changes in material, changes in geometry, and the application to a full scale propeller. The latter includes adjustments to reflect the full scale change using the method presented in section 5.6.1. In addition, the effectiveness of the turbulence model, described in section 5.5.1, in accurately describing the blade deformation is discussed and compared to the LES input. Finally, the impact on one-way vs two-way coupling is also shown.



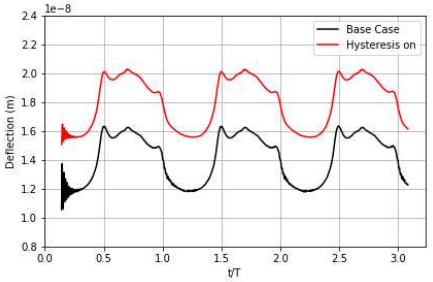
(A) -2^o drift angle



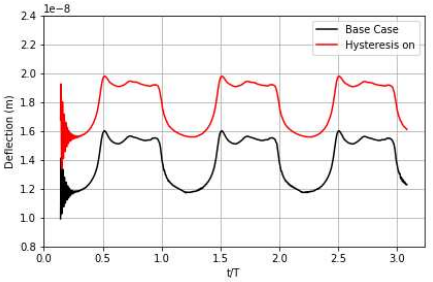
(B) 0^o drift angle



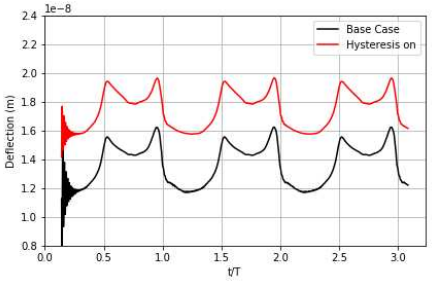
(C) 4^o drift angle



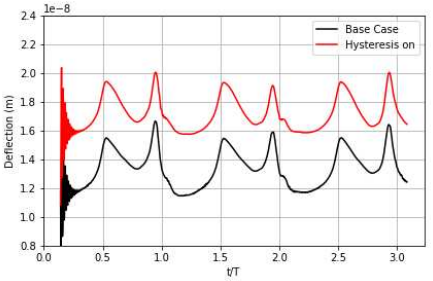
(D) 6^o drift angle



(E) 8^o drift angle



(F) 12^o drift angle



(G) 16^o drift angle

FIGURE 7.10: Deflection of steel blade using Hystersis model.

7.4.2.1 Change in Material

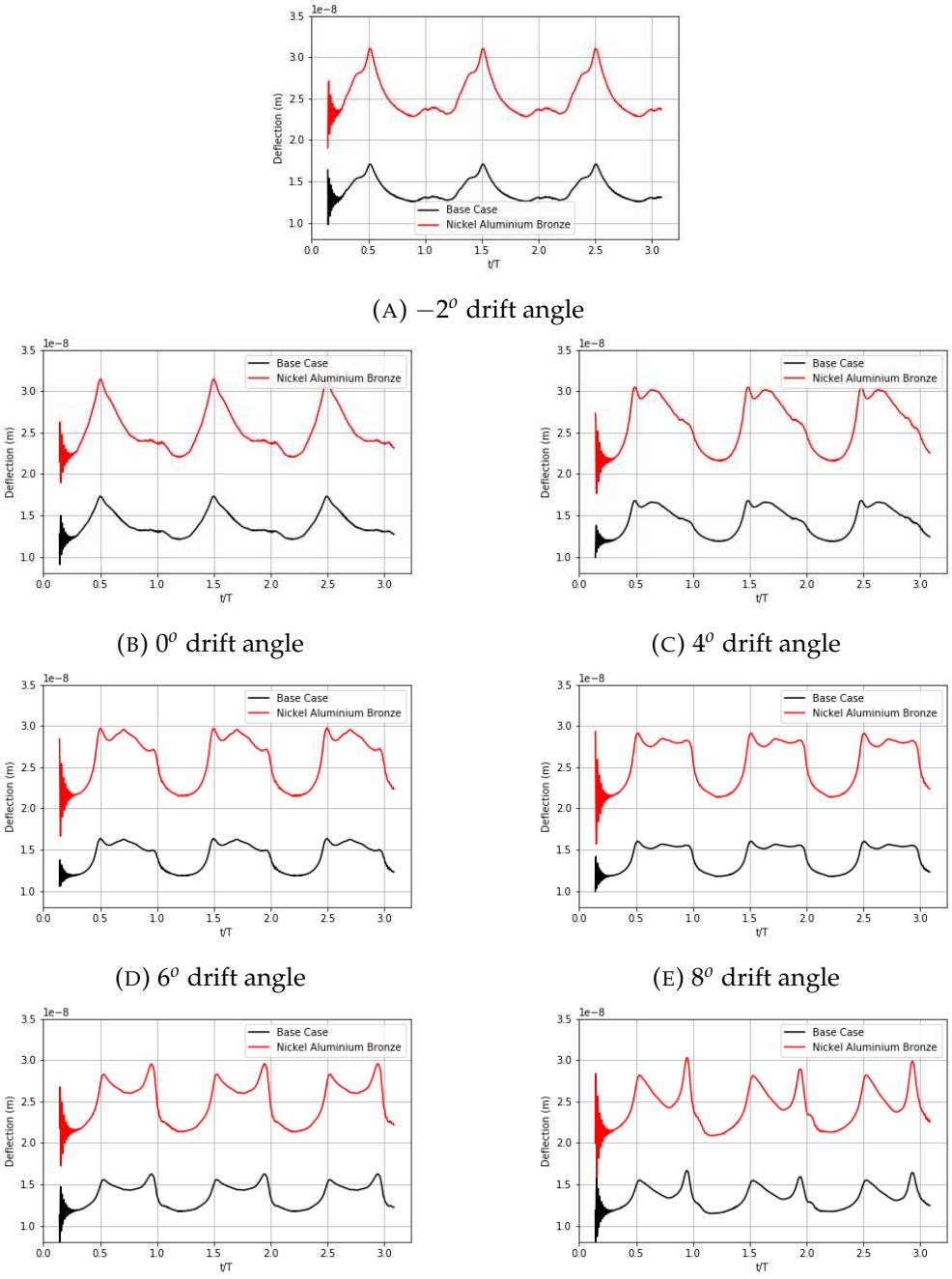
The reference case used steel as a material, however, traditionally propeller blades are made from Nickel Aluminium Bronze (NAB). This material has a lower Youngs modulus and density when compared to steel. The material properties of NAB are shown in Table 7.1

TABLE 7.1: Nickel Aluminium Bronze Material Properties

Property	Value
Youngs Modulus	110000 10^6
Density	7530
Poissons ratio	0.3
Yield Stress	240 MPa

The deflection and stress for each drift angle are shown in Figures 7.11 and 7.12 respectively.

Here it can be seen that the deflection is much larger for the NAB when compared to the steel blade, although the shape of the deflection is the same as the steel blade. This is expected given the blades Youngs modulus is approximately half of that of steel, ergo the deflection is twice as large. Although the mode shapes remain constant when changing materials, the modal frequencies change significantly. With less stiffness the frequencies are reduced by $\approx 30\%$.



(A) -2^o drift angle

(B) 0^o drift angle

(C) 4^o drift angle

(D) 6^o drift angle

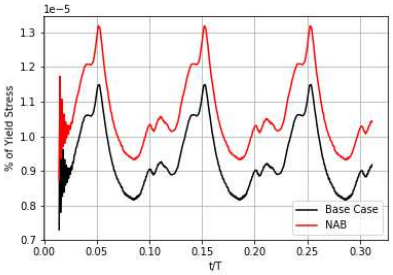
(E) 8^o drift angle

(F) 12^o drift angle

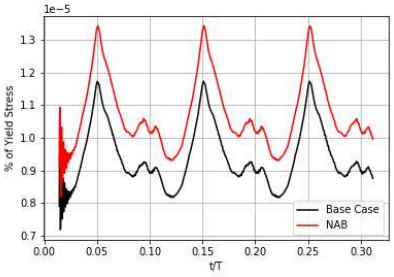
(G) 16^o drift angle

FIGURE 7.11: Deflection of NAB blade compared to Steel blade.

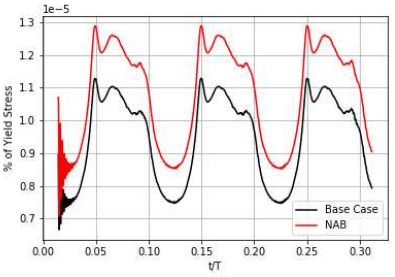
Despite the deflection being twice as large, the maximum stress of the blade remains constant next to the steel blade. This is because the stress is computed based on the local deformation and structural properties. Given the Youngs modulus has been approximately halved and the deformation has been doubled it should follow that the stress remains constant. However, the yield stress of NAB is less than steel, therefore, the NAB blade will experience higher levels of fatigue per revolution.



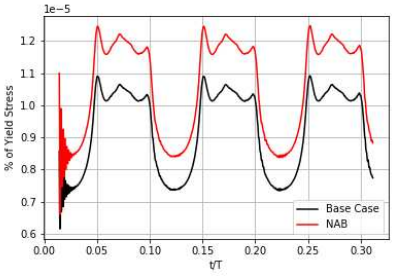
(A) 0^o drift angle



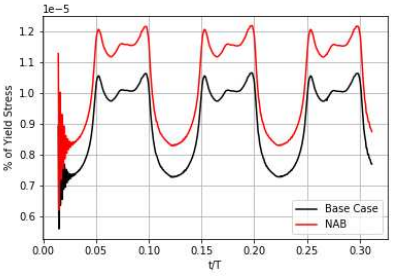
(B) 4^o drift angle



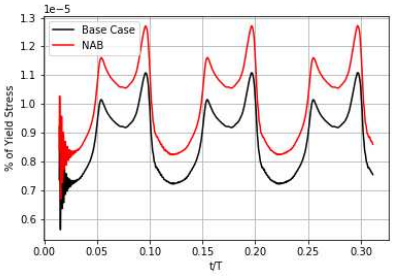
(C) 6^o drift angle



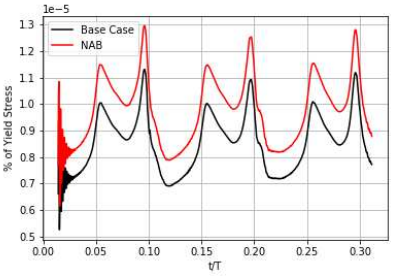
(D) 8^o drift angle



(E) 12^o drift angle



(F) 16^o drift angle

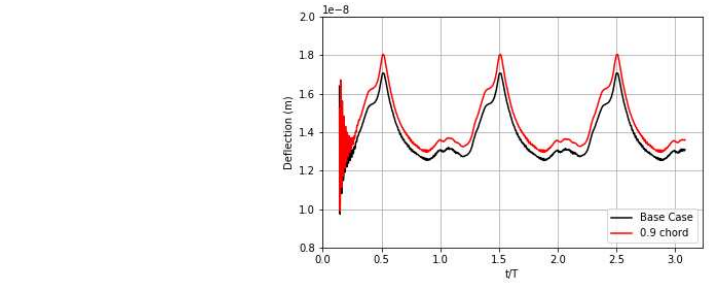


(G) 16^o drift angle

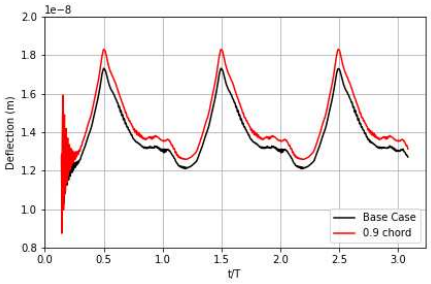
FIGURE 7.12: Maximum Von Mises Stress for NAB blade and Steel Blade.

7.4.2.2 Change in Geometry

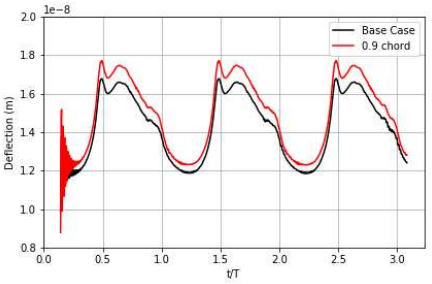
When designing a propeller blade there are many changes to make which fullfill different requirements. For instance, the efficiency of the propeller can be increased by reducing the chord. This, however, will have a detrimental effect on the structural properties of the blade. Figure 7.13 shows the deflection of the blade using 90% of the original blade chord.



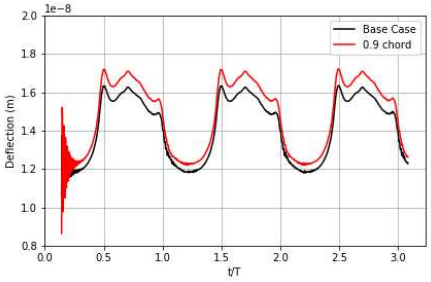
(A) -2^o drift angle



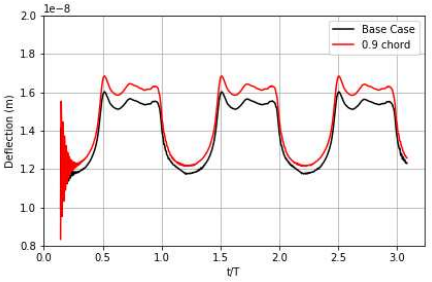
(B) 0^o drift angle



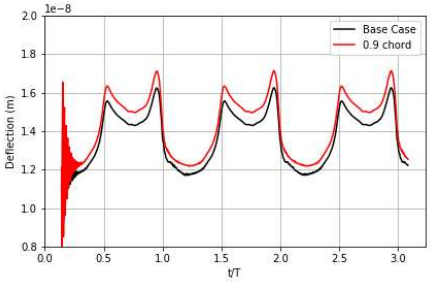
(C) 4^o drift angle



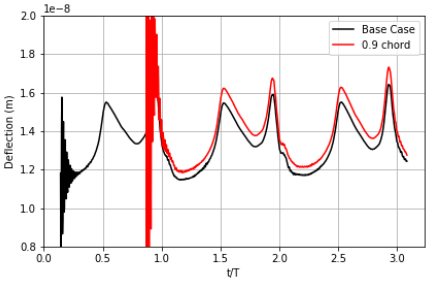
(D) 6^o drift angle



(E) 8^o drift angle



(F) 12^o drift angle



(G) 16^o drift angle

FIGURE 7.13: Deflection of steel blade with 10% reduction of the chord.

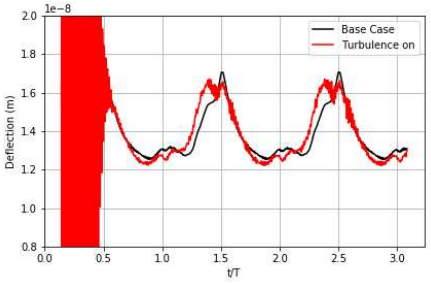
It can be seen that the reduction in chord does in fact increase the deflection for all drift angles. The reduction in chord causes the changes in the convergence capability of the BEMT solution causing a rapid change in loading on the blade. This is the reason for the large deformations at the 16^o drift angle case. The discrepancy only occurs at 16^o drift angle due to the larger velocity gradient causing the BEMT solution to fail to converge. This demonstrates a weakness in the capabilities of the BEMT-Plate

model as the lack of successful convergence in the BEMT solution can cause non-physical blade excitation in more extreme wakes.

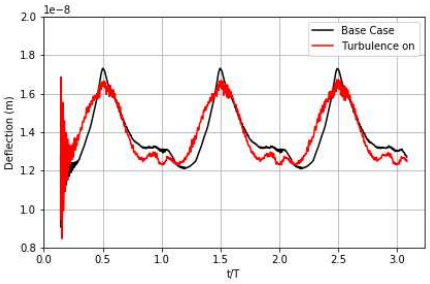
The effect of changes in chord is shown in this case however the designer can also change geometric parameters such as: thickness, skew, camber distribution and pitch which can all be captured by the model.

7.4.2.3 Turbulence Response

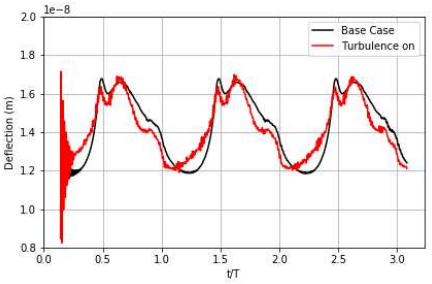
In this sub-section the inclusion of the turbulence model described in section 5.5.1 is included. The maximum blade deflection for each drift angle is shown in Figure 7.14.



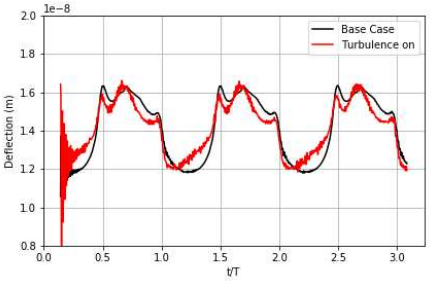
(A) -2^o drift angle



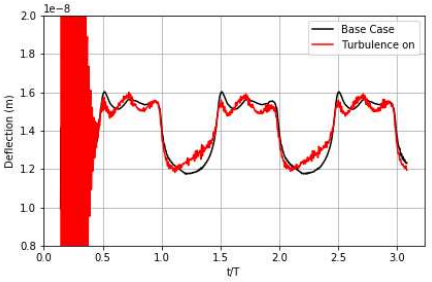
(B) 0^o drift angle



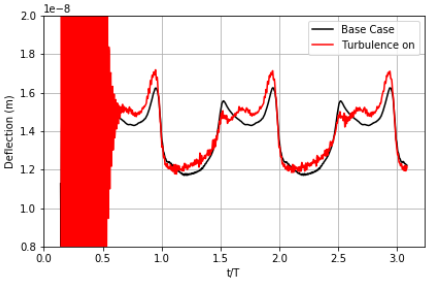
(C) 4^o drift angle



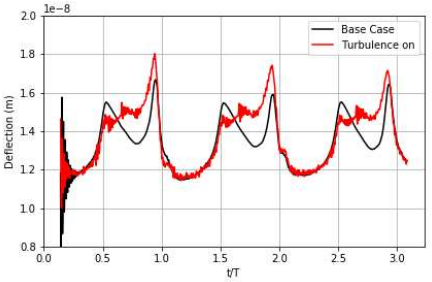
(D) 6^o drift angle



(E) 8^o drift angle



(F) 12^o drift angle



(G) 16^o drift angle

FIGURE 7.14: Deflection of steel blade with turbulent overlay

The effect of turbulence has a significant effect on the blade deformation. Firstly, the solution takes far longer to converge due to the highly fluctuating velocity field, hence the large spikes at the beginning of the simulation.

Secondly, the shape and peak of the blade deformations are altered considerably. For the -2^o drift angle case, the peak in the rotation is considerably reduced and a secondary peak occurs. This can have a significant effect on the life cycle of the blade

as a peak in blade deformation occurs twice in one cycle which will be detrimental to the fatigue performance of the blade.

For the 0^o, 4^o, 6^o, 8^o and 12^o cases the shapes of deformation have been altered by reducing the peaks and troughs of deformation. However, the 16^o case shows a definite increase in deformation at the 2nd peak of its rotation.

It is noted that there is a consistent high frequency deformation for all cases. These fluctuations are too small to be caused by the BEMT non-convergence. They are a result of the turbulent fluctuations causing different parts of the blade to experience different loading which in turn gives rise to the first natural frequency to be excited. The damping of the structure is well captured in the model to ensure the larger fluctuations are damped. As described in section 5.5.1, the turbulent flow field generated by the synthetic turbulence model overestimate the magnitude of the fluctuations therefore the high frequency deformations can be treated as a worse case scenario.

7.4.2.4 Application to a Full Scale Propeller

Although model propellers are often used for testing and validation, the propeller will ultimately be used at full scale. Using the Hoekstra contraction method, described in section 5.6.1, the input velocities have been modified and the HMRI propeller has been resized to full scale. The full scale parameters are shown in Table 7.2.

TABLE 7.2: HMRI Full scale propeller Geometry

Diameter	9.86m
n	1.45m/s
Vs	7.987m/s

Figure 7.15 shows the maximum blade deflection for the full scale propeller. It is shown that the deflection is significantly larger when compared to model scale. In addition, the contraction of the wake causes more rapid changes in loading to occur at the root of the blade. This gives a far more smooth deflection when compared to the model scale deflections which are more sensitive to circumferential changes in the wake. This is because in model scale the slower moving wake regions extend to larger radial sections causing higher loading at the mid-blade compared to full scale deflections.

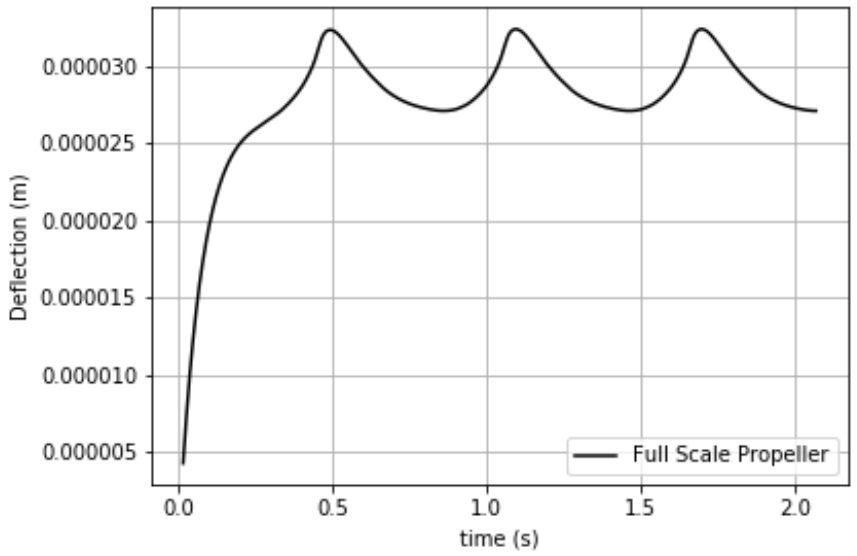


FIGURE 7.15: Maximum deflection of blade of full scale propeller using Hoekstra wake contraction at 0^o drift angle.

7.5 Chapter Summary

This chapter introduced the methodology of coupling the Blade Element Momentum Theory hydrodynamic loading with a 2D plate structural model. The method for computing damping has been discussed and numerical tests have been performed.

It has been found that changes in material and geometry of the propeller blade can be altered and are well captured in the model. Scaling to a full scale propeller using wake contraction methods have been discussed, although the deflection results require verification. Furthermore, the changing of mesh size and time stepping can be altered to the users requirements of computational time and detail of the studied wake.

The use of one-way coupling vs two-way coupling gives a negligible difference in deflection results. The extension of the plate model to include composite materials with bend-twist coupling would be required to have a significant impact for two way coupling.

However, the use of the hysteresis model has showcased the need for further development requirements as the deflections have been significantly increased by the inclusion. The need for the pressure distribution modification due to the blade motion should be investigated and a model to capture this should be implemented.

Overall, the sensitivities of the modelling techniques have been captured and discussed with the changes available to the propeller designer being shown to be captured by the stresses and deformations produced by the model.

Chapter 8

High Fidelity Fluid-Structure Interaction Simulation

8.1 Introduction

To test the accuracy of the fluid-structure interaction model a high fidelity model has been used using one-way finite element analysis with CFD coupling to obtain the vibration properties of the propeller. This has been achieved running a CFD simulation of the propeller with a non-uniform inlet in OpenFoam and mapping the pressures on a reference blade to a transient model in Ansys.

8.2 Non-uniform inlet

8.2.1 CFD Setup conditions

To obtain a non-uniform inlet to the propeller the wake of the KVLCC2 at the propeller plane is mapped to the inlet of the propeller simulation as shown in Figure 8.1. The inlet profile is coloured by velocity with blue for low velocity and red indicating high velocity.

To obtain a non-uniform inlet in Openfoam the inlet boundary condition is set to a *timeVaryingMappedFixedValue* condition. This allows an inlet which varies in space and time for velocity and pressure. An additional folder must also be introduced to allow the definition of the data. The data is stored in the directory tree as shown:

```
case directory
└── constant
    └── boundary data
        └── inlet
```

The inlet folder contains a *points* file which defines the coordinates of the points in which the fields are mapped to. The *0* directory contains the values of the field, which are mapped to the required boundary. To ensure the simulation is representative of the model described in Chapter 7, only the velocity has been mapped to the inlet and not pressures or turbulence properties.

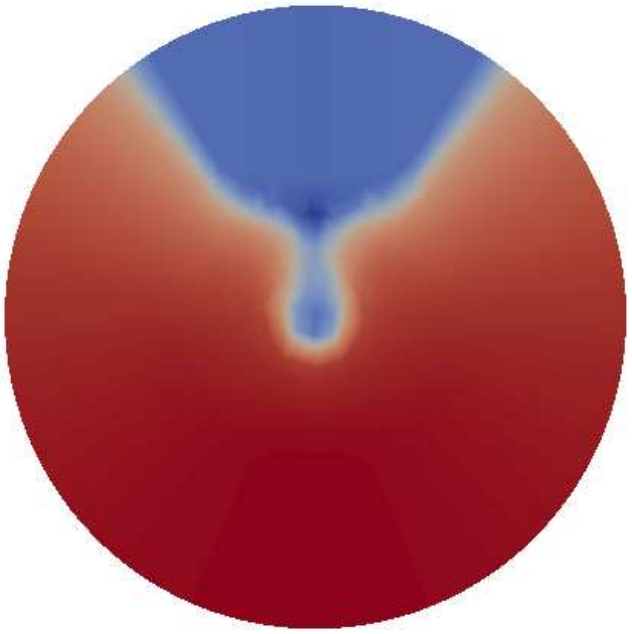


FIGURE 8.1: Velocity inlet to Propeller CFD simulation

The same mesh as described in Section 4.3 has been used with the only change being the inlet conditions. The propeller is located 0.7D from the inlet, this was found to be optimal in previous studies Ji et al. (2012). The propeller is rotating at a rate of 9.9 rps as this is the rotation rate of the propeller at a ship operating speed of 1.1702 m/s according to Sung et al. (2014).

To obtain the pressure distribution over the blade the sampleDict utility has been used which stores the pressure of the blade at each time step. A time step of $2.8e^{-04}$ has been used to represent 1^o of propeller rotation per timestep. Moreover, a reference blade is defined to map the pressures of one blade to the structural model.

8.2.2 Results

The thrust force produced by the whole propeller and the reference blade was recorded, the time history of these forces are shown in Figure 8.2. Here the propeller

thrust force oscillates about a mean value due to shed vorticity. A larger fluctuation of thrust force can be seen on a single propeller blade due to the non-uniform inlet. The frequency of this oscillation is the blade passing frequency. As the blade rotates it will experience high loading at the region of low velocity, this is indicated by the shape of the propeller performance curve as at low advance ratio the thrust coefficient is high. The blade rotates and experiences a reduction in loading as the fluid velocity increases again.

The asymmetry between the unloading and loading region is due to the tangential wake. As shown in Figure 5.5b the z-component of velocity is directed towards the ship hull. This causes the tangential wake to increase the effective incidence when the tangential wake is positive. This increases the loading thus slowing the unloading. The opposite is applicable in the opposite half of the propeller rotation where the tangential wake reduces the incidence hence increasing the loading causing the rapid rise in loading.

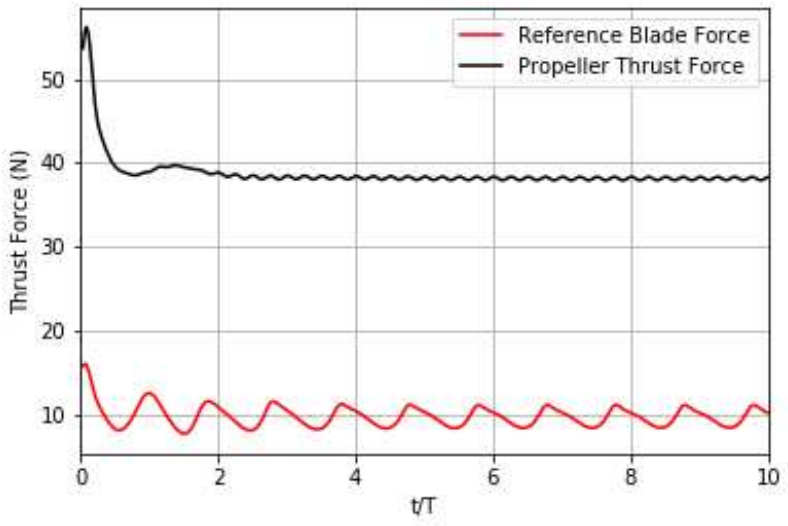


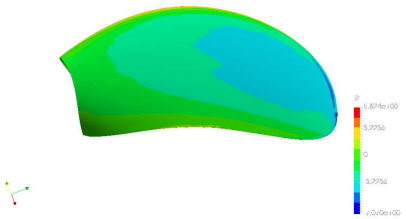
FIGURE 8.2: Thrust force time history of Reference Blade and Overall propeller.

8.2.3 Transient Structural simulation

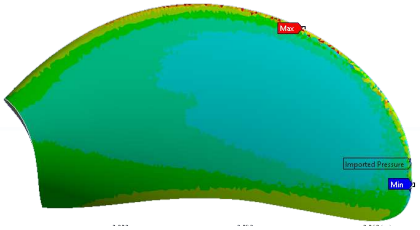
To obtain the transient response to the propeller loading a high fidelity Ansys model has been used. This is done by mapping the pressure loads of the propeller blade from the CFD solution to the model. The base of the blade is restricted in motion, making it a cantilevered blade. The pressure distribution of the propeller blade at each timestep of the CFD simulation are applied at the same corresponding timestep of the FEA calculation.

8.2.3.1 Set up

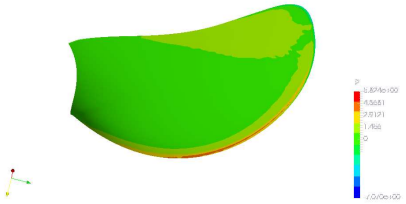
Firstly the geometry was imported and the propeller loading imported in the correct format. Care has been taken to correctly match the time step of the CFD simulation to the FEA simulation. The pressure loading on the Ansys model and the CFD model are shown on Figures 8.3 .



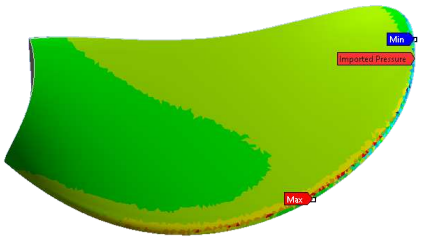
(A) Face of propeller blade pressure distribution from CFD.



(B) Pressure loading on Face of propeller blade for Ansys simulation



(C) Back of propeller blade pressure distribution from CFD.



(D) Pressure loading on Back of propeller blade for Ansys simulation

FIGURE 8.3: Pressure Distribution on HMRI Blade

As shown the pressure distribution is mapped well between the CFD simulation and FEA simulation. The FEA mesh consists of C elements and is applied in a Vacuum as opposed to water. This is in keeping with procedures within the industry sponsor. The main difference between operating in a vacuum and water is the frequency in which the blade will be excited due to added mass effects. Also the added damping will not be considered in a simulation in a vacuum. The blade remains stationary throughout the simulation, the coordinates of each load application point is applied at a stationary point and the pressure loading changed with time.

8.2.3.2 Results

The deformation of the blade at the 1st loading time step is shown in Figure 8.4. Here the blade deformation shape is predominantly bending with peak bending at the blade tip.

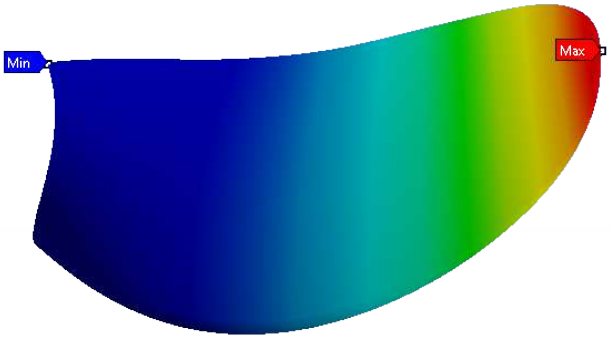


FIGURE 8.4: Blade deformation shape at 1st time step

Figure 8.5 shows the deformation of the blade at several probe points on the blade surface. The shape of bending with time matches the shape of propeller loading. The maximum loading is very small at approximately 1 nanometer. It should be noted that the 1st few time steps up to ≈ 0.15s is the reaction to an impulse force. The frequency of this reaction is the 1st natural frequency and should be discounted from the analysis of determining the vibration frequencies.

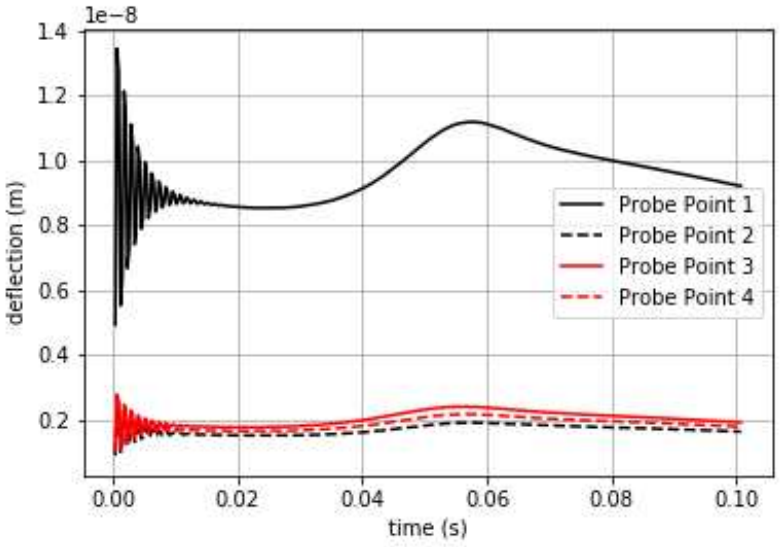


FIGURE 8.5: Time dependent deformation at several probe positions.

8.3 Comparison with Computationally Efficient model

The computationally expensive model has been compared against the computationally efficient model described in Chapter 6 . The deflection at the tip for the BEMT-Plate coupling and CFD-FEA coupling is shown in Figure 8.6.

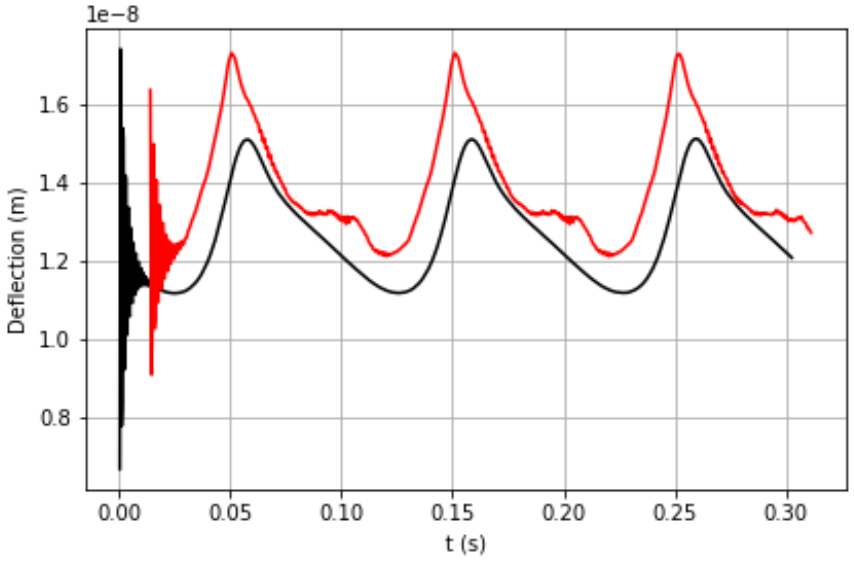


FIGURE 8.6: Comparison between the tip deflection of CFD-FEA coupling vs BEMT-Plate model coupling

It can be seen that the overall trend of the deflection compares well between the two simulations. The tip deflection magnitude is slightly higher for the plate coupling tool. Additional features are seen at the peak loading and as the blade loading decreases.

These additional features can be attributed to the tangential wake. At these positions the tangential wake is not precisely zero which will add a slight force at this position. The BEMT solution does not account for the effect of the presence of the propeller on the upstream velocity which the CFD simulation will account for. The inclusion of the wake contraction for the full scale propeller described in Section 5.6.1 reduced the rapidly changing deflections and gave results in a similar shape to the CFD-FEA coupled simulation. Therefore, the inclusion of the wake contraction for the BEMT-Plate model will be required.

Despite the errors associated with the BEMT-Plate coupling model, considering the computational cost, it gives a good approximation of the deflection of the blade due to unsteady loading. The shape of deflection, shown in Figure 8.7 matches well to the CFD-FEA coupled simulation shown in 8.4.

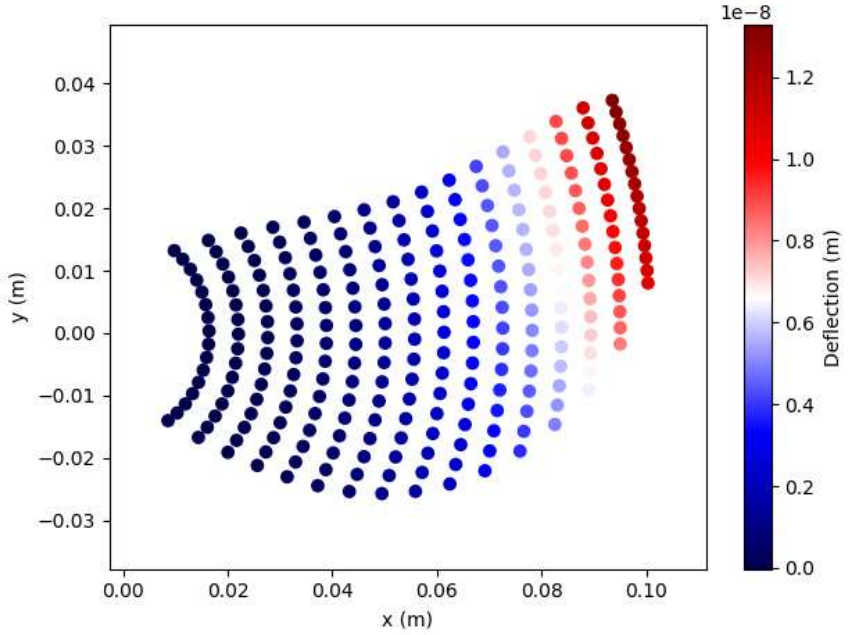
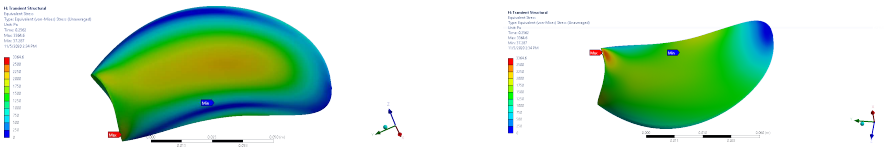


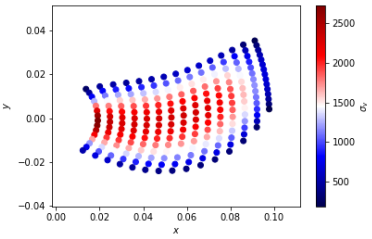
FIGURE 8.7: Deflection shape due to pressure loading

The Von Mises stress distribution of the blade occurring at a point of maximum deflection is shown in Figure 8.8. It is shown that the stress distribution of the Python model matches the Ansys model reasonably well with the central spine of the blade, where the majority of the thickness is held, experiencing the most stress. Also, the region at the root of the blade experiences high stress. When compared to experimental data presented by Savio (2015) the shape of deflection matches well to the presented model. Although this study used a material with a lower elastic modulus the shape of deformation is inline with the literature albeit the deflection shown is far smaller in this case due to the higher elastic modulus.



(A) Stress distribution on Back of blade from Ansys.

(B) Stress distribution on Back of blade from Ansys



(C) Stress Distribution from Python Model

FIGURE 8.8: Stress Distribution Comparison between Ansys model and Python Model.

8.4 Discussion

8.4.1 Computational Cost

The main benefit of utilizing the BEMT-Plate model is the computational cost. The computational time per time step of both the CFD-FEA coupled simulation and BEMT-Plate coupled simulation is discussed in this section.

The CFD-FEA simulation is a one-way coupled simulation so the pressure loads from CFD are mapped directly to the structural simulation. Therefore, the computational cost of each simulation can be easily obtained.

The CFD simulation was run on 48 cores on the Iridis 5 cluster. Each time step takes $\approx$ 68s. Although the computational cost is highly dependent on mesh size and convergence criteria an estimate of cost can be provided. The FEA simulation was run on 4 cores using Ansys. Each time step takes $\approx$ 13s.

Table 8.1 shows the break down of computational cost for the CFD and FEA simulations. For both simulation methods the parallel efficiency was close to 99% meaning that there was only a small overhead in transferring data between cores. Therefore, it is a reasonable assumption to linearly scale the computational time to obtain the cost as if it where run in serial.

TABLE 8.1: Computational Cost of CFD-FEA simulation

Simulation	Cost (parallel)	Cost (serial)
CFD	$\approx$ 68s	$\approx$3264s
FEA	$\approx$ 13s	$\approx$52s
Total	$\approx$ 81s	$\approx$3316s

In comparison, the BEMT-Plate modelling technique, using 30 structural elements, takes $\approx$ 1.8s - 2s per time step. For the same computational cost the BEMT-Plate modelling technique can analyse 1000s of designs as one coupled CFD-FEA simulation. This is, perhaps, a somewhat unfair comparison given the far more detailed data that the high fidelity model can output however, the applicability of the BEMT-Plate coupled modelling technique as a design tool has been demonstrated. Finally, a good speed up has been shown to decrease the computational cost when going from an interpreted programming language to a compiled language. This can be used throughout each part of the design code to decrease the computational cost further. The use of Python allows more rapid development due to the lack of need to deal with memory management and compliers.

In this Thesis only one case has been presented for the given wake fields however, the computational cost of the algorithm is low enough to run many thousands of propeller designs and operating conditions. Although the deformation results may not be as accurate as the high fidelity simulation, it is possible to compare propeller designs against each other.

8.4.2 Design applications

It is desirable to obtain the blade deformation characteristics at the early stage of propeller design, as significant structural damage and fatigue can be mitigated by ensuring the stress and deformations that the blade experiences when operating in the unsteady wake, are within the found criteria. Designs can be tuned in a more efficient manner as opposed to being overly cautious with blade thickness and chord to ensure the structural requirements are met.

With the increased use of optimization and machine learning in recent years a computationally cheap tool is required to run many designs and obtain the optimal design based on user defined parameters. This is not realistically feasible using high fidelity methods within the design stage.

The use of the Plate-BEMT method can be applied to several applications of rotating systems such as: fan design, wind and tidal turbines and aircraft propellers, although these should be validated.

Although this chapter has demonstrated the applicability of BEMT-Plate coupling there are some disadvantages. Firstly, the BEMT can achieve poor convergence when experiencing a unsteady wake with extreme velocity gradients. The non-convergence can cause a sudden change in load on the blade which causes the blade to vibrate at the natural frequency. Although care must be taken to determine if this is an numeric caused vibration or an impulse due to the fluid loading and structural response.

The predominant deformation which occurs on the propeller is in the rake direction. This is not, however, accounted for in the BEMT solution. Therefore the change in rake caused by the deformation will not be considered when completing a 2-way coupled simulation. This has been considered by the inclusion of hyteresis changing the pressure distribution on the blade. Further tests will be required to ensure this is an accurate model or if more sophisticate models are required.

With increased use of composite materials the plate FEA should be extended to include this choice of material. This will allow the user to define materials with strong bend-twist coupling and design a propeller blade which utilises this material property. The main, practical benefit of obtaining the vibration properties of the propeller blades at the design stage is to design against the high vibration before models are built which will greatly improve the speed in which low vibration propellers are developed. Further to this, the tool can take into account various operating conditions which can cause greater vibration to the propeller blades. The use of the developed tool can minimize the need for an expensive re-design if vibration issues arise during certain drifting cases.

8.5 Chapter Summary

In this chapter an unsteady marine propeller simulation has been run to obtain pressure loading on a blade as it rotates in a non-uniform wake using CFD. The pressure loading from the CFD simulation has been mapped on to a structural model of the propeller blade.

The blade deformation has been compared to the BEMT-Plate coupled simulation to obtain the difference in blade tip deflection. It was shown that the BEMT-Plate model compares reasonably well with the high fidelity model with a tip deflection error of $\approx$ 10 %. The blade deformation shape matches well to the high fidelity model.

The difference in maximum deflection at the tip is consistent throughout the rotation with the shape of tip deflection matching well through time. The need for good wake contraction has been identified, as the inflow velocity to the propeller contracts in the CFD simulation which is not captured in the BEMT solution. Without this contraction

an additional perturbation in deformation is seen as the propeller is rotating upward towards the dead centre position.

The stress distribution of the plate model agrees very well with that of the 3D FEA model as does the maximum Von Mises stress. However, the 2D nature of the plate model is limiting to only computing the stress due to tension.

Finally, the impact of the BEMT-Plate model has been discussed, citing that the computational cost has the main advantage over high fidelity models. The BEMT-Plate technique is therefore, a good candidate for design stage FSI modelling of propeller blades. Although some aspects of the modelling techniques have found to be lacking in robustness and accuracy, the capability for implementation has been demonstrated.

Chapter 9

Conclusions

A computationally efficient method has been developed that allows the vibration characteristics of alternative propeller design at the concept to be evaluated. The one-way coupled method using a blade element momentum theory and two dimensional FEA plate structure is evaluated to be at least three orders of magnitudes faster than a full high fidelity two way coupling approach. As such it can provide an excellent means of evaluating many designs, as for example in genetic algorithm based optimisation process prior to the eventual confirmation of behaviour using a high fidelity simulation on the final design.

The approach discussed can provide a designer with rapid insight into the vibration properties of the propeller blades for a given geometry. Although fine details, such as trailing edge vortex structures, will not be captured by this model the analysis of such features can be reserved for high fidelity simulations.

Furthermore, the approach can rapidly examine the ship speed in which vibration inducing cases can be developed. This has been developed within a framework of standard wakes that have a higher probability of having mechanisms which cause severe vibration issues. In this study, various drift angles have been used in the initial framework. However, this framework can be built upon to include far more conditions such as: zig zag manoeuvre, turning circles or crash stops. This is a useful addition as, currently, propellers are designed using the wake field when the ship is operating normally. This can result in excessive vibration and fatigue during manoeuvres which are not accounted for in the propeller design. This has the potential to be costly as it could not be found to be an issue until the propeller is in service, resulting in an expensive redesign.

To obtain the hydrodynamic loading the use of BEMT has been shown to give good agreement with the experimental performance curves. BEMT has been shown to be computationally efficient and can include Reynolds number effects which has the advantage over other computationally efficient models such as potential flow based

panel methods. The use of the Reynolds number dependent hydrofoil database has been shown to give good $C_l vs \alpha$ which change due to Reynolds numbers. In this thesis only the NACA66 mod foil section has been used as it is a commonly used section shape for propeller blades however other section shapes could be included in the database giving the designer more control over the design.

A hysteresis model has been included in the hydrodynamic model to account for the changes in lift coefficient of the blade section caused by the pitching and heaving of the blade. This has been verified against previous work using the same model. The stall region of the blade section is accounted for by the 2D foil database however, the dynamic deep stall properties of the foil have not been modelled. For the present study this is acceptable due to the reasonably low angles of attack the blade sections experience however, further work could be done to effectively model these regions. Cython is utilized at this stage to significantly decrease the computational cost of each BEMT cycle. It is shown that the use of this compiled version of Python decreases the computational time by $\approx$ 80x. This fulfils objective 1 as, due to the low computational cost of running a BEMT routine in Cython is small it can be run on a local machine hundreds of times with ease.

The BEMT solution is compared against a high fidelity CFD simulation. It is found that using BEMT gives a load distribution which compares well to high fidelity simulations. The high fidelity simulation techniques have been established for both a steady state simulation and unsteady simulations.

Objective 4 has been completed as a series of unsteady, non-uniform wakes have been generated using CFD simulations of the KVLCC2 hull form at various drift angles. These simulations have been shown to agree well with various studies in literature. In an attempt to capture turbulent fluctuations in the wake a synthetic wake model has been implemented which can generate a turbulent flow field based on the RANS results. The synthetic turbulent flow field was compared to an LES simulation. The synthetic model compared reasonably well in some regions of the wake but very poorly in others. The LES simulation showed a significant difference in flow field at the propeller plane compared to URANS simulations so should be accounted for when studying propeller vibration. The synthetic turbulence model did not capture this well however, the unsteady flow field was captured. This drives the fluid-structure interaction that cause a propeller to vibrate. The Hoekstra wake contraction model has been implemented to obtain the equivalent full scale wake. This is shown to give a reasonably accurate description of the effects of scale when compared to a high fidelity CFD simulation with a small computational cost.

To obtain the structural properties the propeller blade has been modelled using plate elements. The uses of these elements has been shown to give good agreement with the mode shapes and modal frequencies when compared to a high fidelity full 3D FEA

analysis using Ansys software. It has been shown that using plate elements significantly reduces the computational cost of the structural analysis and are therefore determined as a good choice for obtaining the structural properties in an efficient manner. The stress of distribution of a 2D plate compares well with the high fidelity simulation. The $HHT - \alpha$ method is discussed and implemented to capture the dynamic motion of the structure and to capture the response to a time dependent load.

As the propeller is submerged in water the added mass effects are modelled using empirical corrections. These use of the empirical corrections are shown to give a good reduction in modal frequencies without changing the mode shapes which agrees well with literature. Moreover, the wet modal frequencies agree well with the high fidelity modelling technique for submerged propeller blades.

Although the plate model gives good agreement with 3D models there is a limitation due to the lack of degrees of freedom. This causes errors when a highly twisted blade is being studied. The lack of modelling of composite materials is also an issue but this can be developed in future work. Although the wet blade has been modelled reasonably well the change in structural response due to moving water has not been included nor has the inclusion of added damping which should be complete in future studies. Objectives 2 3 and 5 have therefore been realized with the structural properties well established and compared well against high fidelity methods.

Chapter 7 discusses the use of the BEMT-Plate modelling technique. A series of numeric and physical parameter changes were performed to test for the sensitivity of changes to the model. It was shown that the modelling technique captures the deformation of the blade due to changes in wake field well. It is also shown that the designer can change physical parameters such as: the geometry and material when designing the propeller and these changes are well captured by the model.

The weakness of the BEMT-Plate model are shown to be the lack of robust convergence in the BEMT solution causing non-physical rapid deformations of the blade. Due to metallic materials being used the rotational deformation is very small. This causes the difference between the one-way and two-way coupling algorithms to be negligible. This further demonstrates the need for implementation of a composite material option within the plate model FEA as this can enable the designer to design for the coupled deformation characteristics of these materials. The hysteresis model is shown to significantly increase the deformation. This is due to the change in pressure distribution on the blade section due to blade motion not being accounted for, which should be implemented in future.

Chapter 8 compares the computationally efficient model to a high fidelity CFD-FEA coupled simulation. Both simulation methods match in terms of initial conditions and boundary conditions and both are coupled using the one-way algorithm. It is shown that the BEMT-Plate model overestimates the deflections when compared to the high

fidelity model however the shape of deflection and stress distribution on the blade compare very well. The computational cost of each method is also discussed with the BEMT-Plate model showing significant computational cost savings such that 1000s of designs can be analysed at the same cost as one high fidelity model.

The contribution this thesis provides is the use of a Blade Element Momentum theory code, coupled with a reduced FEA model to provide the vibration properties of marine propeller blades in a computationally efficient method. The applicability of the code has been discussed and compared with a high fidelity model and is shown to agree well with high fidelity methods.

9.1 Recommendations For Future Work

Following from the work completed in this thesis it is recommended to further investigate the effects of hysteresis on the blade deformation and how it is implemented into the blade element momentum theory loops. Moreover, particularly high angles of attack can be experienced by the blade. A more complete dynamic stall model for these regions will be of interest. Moreover, the incorporation of cavitation modelling would be highly beneficial due to its dominance of noise contribution, this should be done in a computationally efficient way to ensure the code is not slowed to be unpractical for use.

Further investigations into the applicability of using the plate FEA model to capture the properties of composite materials will also add value to the project. The ability to model composite coupling at the design stage would be of benefit for future applications.

Furthermore, due to the quick nature of the proposed method it would be of interest to apply the tool in an optimization environment to design a propeller with low vibration properties within various non-uniform, unsteady wakes.

Although the synthetic turbulence model has been used to generate a turbulent overlay has been implemented a more robust study of available models should be carried out in an attempt to better produce a turbulent flow field.

A larger database of wakes should be generated to include different manoeuvres and operating conditions. This will aid designers to develop a low vibration blade for a wider range of flow regimes.

Finally, full validation of the numeric tool is required. Although the tool has been verified using computationally expensive techniques there is still the requirement to validate against both physical model scale data or full-scale data.

References

A. E. H. Love. On the small free vibrations and deformations of elastic shells. *Philosophical trans. of the Royal Society (London)*, 1888.

N. Abbas and N. Kornev. Validation of hybrid URANS/LES methods for determination of forces and wake parameters of KVLCC2 tanker at manoeuvring conditions. *Ship Technology Research*, 63(2):96–109, may 2016. ISSN 0937-7255. . URL https://doi.org/10.1080/09377255.2016.1157275.

M. Abdel-Maksoud, V. Müller, T. Xing, S. Toxopeus, F. Stern, K. Petterson, M. Tormalm, S. Kim, S. Aram, U. Gietz, P. Schiller, and T. Rung. Experimental and Numerical Investigations on Flow Characteristics of the KVLCC2 at 30 Drift Angle. *Wmtc15*, page 26, 2015.

D. T. Akcabay and Y. L. Young. Steady and dynamic hydroelastic behavior of composite lifting surfaces. *Composite Structures*, 227(December 2018):111240, 2019. ISSN 0263-8223. . URL https://doi.org/10.1016/j.compstruct.2019.111240.

Amabili. Free Vibrations Of Annular Plates Coupled With Fluids. 191:825–846, 1996.

Y. Ashkenazi, I. Gol'fman, K. Rezhkov, and N. Sidorov. Glass -Fiber-Reinforced Plastic Parts in Ship Machinery. *Sudostroyenniye Publishing House, Leningard*, 1974.

P. Ausoni. Turbulent Vortex Shedding from a Blunt Trailing Edge Hydrofoil. 4475, 2009.

S. Backaert, P. Chatelain, and G. Winckelmans. Vortex Particle-Mesh with Immersed Lifting Lines for Aerospace and Wind Engineering. *Procedia IUTAM*, 18:1–7, 2015. ISSN 22109838. . URL http://dx.doi.org/10.1016/j.piutam.2015.11.001.

C. Badoe. *Design practice for the stern hull of a future twin-skeg ship using a high fidelity numerical approach*. PhD thesis, 2015.

K.-J. Bathe. *Finite Element Procedures*. 2016. ISBN 9780979004957.

B. Blackwell, W. Sullivan, R. Reuter, and J. Banas. Engineering development status of the Darrieus wind turbine. *Journal of Energy*, 1(1):50–64, 1977.

J. P. Blasques, C. Berggreen, and P. Andersen. Hydro-elastic analysis and optimization of a composite marine propeller. *Marine Structures*, 23(1):22–38, 2010. ISSN 09518339. . URL http://dx.doi.org/10.1016/j.marstruc.2009.10.002.

R. Blevins. *Flow-induced vibration.* New York, NY (USA); Van Nostrand Reinhold Co., Inc., 1990.

S. Brizzolara. A systematic comparison between RANS and Panel Methods for Propeller A systematic comparison between RANS and Panel Methods for Propeller Analysis. (May 2014), 2008.

J. Carlton. Marine Propellers and Propulsion. *Marine Propellers and Propulsion (Second Edition)*, pages 421–429, 2012. .

J. S. Carlton. Propeller blade vibration. In *Marine Propellers and Propulsion*. 1994. .

J. S. Carlton. Propeller geometry. In *Marine Propellers and Propulsion (Second Edition)*, chapter 3, pages 33–49. 2007. .

Y. Chang, S. S. Collis, and S. Ramakrishnan. Viscous effects in control of near-wall turbulence. *Physics of Fluids*, 14(11):4069–4080, 2002. ISSN 10706631. .

C.-w. Chen. SS symmetry Improved Hydrodynamic Analysis of 3-D Hydrofoil and Marine Propeller Using the Potential Panel Method Based on B-Spline Scheme. 2019. .

F. Chen, Y. Chen, and H. Hua. Vibration analysis of a submarine elastic propeller-shaft-hull system using FRF-based substructuring method. *Journal of Sound and Vibration*, 443:460–482, 2019. ISSN 0022-460X. . URL https://doi.org/10.1016/j.jsv.2018.11.053.

Y. Chen, L. Wang, and H. X. Hua. Longitudinal vibration of marine propeller-shafting system induced by inflow turbulence. *Journal of Fluids and Structures*, 68(April 2016): 264–278, 2017. ISSN 10958622. . URL http://dx.doi.org/10.1016/j.jfluidstructs.2016.11.002.

M. Faber. *A comparison of dynamic stall models and their effect on instabilities*. PhD thesis, 2018.

J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. 2002. ISBN 3540420746. .

A. Feymark. *A Large Eddy Simulation Based Fluid-Structure Interaction Methodology with Application in Hydroelasticity*. 2013. ISBN 9789173858403.

I. Fluent. Modeling Turbulent Flows. *ANSYS.Inc.*, pages 6–2, 6–49, 2006. URL http://aerojet.engr.ucdavis.edu/fluenthelp/html/ug/node988.htm.

P. P. Friedmann. AEROELASTIC STABILITY AND RESPONSE ANALYSIS OF LARGE HORIZONTAL-AXIS WIND TURBINES. 5:373–401, 1980.

C. Fureby, S. L. Toxopeus, M. Johansson, M. Tormalm, and K. Petterson. A computational study of the fl ow around the KVLCC2 model hull at straight ahead conditions and at drift. *Ocean Engineering*, 118:1–16, 2016. ISSN 0029-8018. . URL http://dx.doi.org/10.1016/j.oceaneng.2016.03.029.

H. P. Gavin. Numerical Integration in Structural Dynamics. Technical report, 2018.

H. Ghassemi, M. G. Saryazdi, and M. Ghassabzadeh. Influence of the skew angle on the hydroelastic behaviour of a composite marine propeller. *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, 226(4):346–359, 2012. ISSN 14750902. .

S. Goldstein. On the vortex theory of screw propellers. *Proc. R. Soc. London Ser. A 123:440-465*, 1929.

M. Gorji, H. Ghassemi, and J. Mohamadi. Effect of Rake and Skew on the Hydrodynamic Characteristics and Noise Level of the Marine Propeller. *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering*, (September), 2017. ISSN 2364-1835. .

C. J. Greenshields. OpenFOAM. *OpenFOAM Foundation Ltd*, (June), 2016.

Ü. Gülçat. *Fundamentals of Modern Unsteady Aerodynamics*. 2016. ISBN 9789811000164.

C. Habchi, S. Russeil, D. Bougeard, J. L. Harion, T. Lemenand, A. Ghanem, D. D. Valle, and H. Peerhossaini. Partitioned solver for strongly coupled fluid-structure interaction. *Computers and Fluids*, 71:306–319, 2013. ISSN 00457930. . URL http://dx.doi.org/10.1016/j.compfluid.2012.11.004.

M. Harrison and W. M. J. Batten. A blade element actuator disc approach applied to tidal stream turbines. In *Oceans 2010*, 2010.

J. Hawkes. *Chaotic Methods for the Strong Scalability of CFD*. PhD thesis, 2017.

X. D. He, Y. Hong, and R. G. Wang. Hydroelastic optimisation of a composite marine propeller in a non-uniform wake. *Ocean Engineering*, 39:14–23, 2012. ISSN 00298018. .

J. L. Hess and S. A.M. Calculation of Nonlifting Potential Flow About Arbitrary Three-Dimensional Bodies. *Journal of Ship Research*, 8(2), 1964.

M. Hoekstra. PREDICTION OF FULL SCALE WAKE CHARACTERISTICS BASED O N MODEL WAKE SURVEY. *International Shipbuilding Progress,*, vol. 22(no. 250): 204–219, 1975.

M. Hoekstra. An investigation into the difference between nominal and effective wakes for two twin-screw ships. *Educational and Psychological Measurement*, (October), 1977. .

J. Hunt, A. Wray, and P. Moin. Eddies, stream, and convergence zones in turbulent flows. Technical report, Center for Turbulence Research Report, 1988.

B. Ji, X. Luo, X. Peng, Y. Wu, and H. Xu. Numerical analysis of cavitation evolution and excited pressure fluctuation around a propeller in non-uniform wake. *International Journal of Multiphase Flow*, 43:13–21, 2012. ISSN 0301-9322. . URL http://dx.doi.org/10.1016/j.ijmultiphaseflow.2012.02.006.

J. Jiang, H. Cai, C. Ma, Z. Qian, K. Chen, and P. Wu. Mechanics A ship propeller design methodology of multi- objective optimization considering fluid structure interaction fluid structure interaction. 2060, 2018. . URL https://doi.org/10.1080/19942060.2017.1335653.

A. J. Keane and P. B. Nair. *Computational Approaches for Aerospace Design Computational Approaches for Aerospace Design The Pursuit of Excellence*. 2005. ISBN 9780470855409.

A. M. Khan. *Constrained optimization techniques*. PhD thesis, 1997.

T. Kim, J. Hur, and H. L. Hyundai. Cavitation Influence on Singing Propellers. (August):5–10, 2018.

R. Kraichnan. Diffusion by a random velocity field. *Phys. Fluids*, 13:22–31, 1970.

L. Larsson and F. Stern. *Numerical Ship Hydrodynamics An assessment of the Gothenburg 2010 Workshop*. 2014. ISBN 9789400771888.

H. Lee. A Numerical Study on the Hydro-elastic Behavior of Composite Marine Propeller. *4th International Symposium on Marine Propulsors*, 2015.

H. Lee, M. C. Song, J. C. Suh, and B. J. Chang. Hydro-elastic analysis of marine propellers based on a BEM-FEM coupled FSI algorithm. *International Journal of Naval Architecture and Ocean Engineering*, 6(3):562–577, 2014. ISSN 20926790. . URL http://dx.doi.org/10.2478/IJNAOE-2013-0198.

K. Lee, Z. Huque, R. Kommalapati, and S. E. Han. Fluid-structure interaction analysis of NREL phase VI wind turbine: Aerodynamic force evaluation and structural analysis using FSI analysis. *Renewable Energy*, 113:512–531, 2017a. ISSN 18790682. . URL http://dx.doi.org/10.1016/j.renene.2017.02.071.

S. Lee and H. Kim. Wind tunnel tests on flow characteristics of the KRISO 3,600 TEU containership and 300K VLCC double-deck ship models. *Journal of Ship Research*, 2003.

T.-g. Lee, G.-s. Song, J.-n. Kim, J.-s. Lee, and H.-g. Park. Effect of Propeller Eccentric Forces on the Bearing Loads of the Complicated Shafting System for Large Container Ships. *Proceedings of the Fifth International Symposium on Marine Propulsors*, 1(June), 2017b.

J. G. Leishman. Validation of Approximate Indicial Aerodynamic Functions for Two-Dimensional Subsonic Flow. *Journal of Aircraft*, 25(10):914–922, 1988. .

A. Lelong, P. Guiffant, and J. Andr. An Experimental Analysis of the Structural Response of Flexible Lightweight Hydrofoils in Various Flow Conditions. *International symposium on transport phenomena and dynamics of rotating machinery*, 140 (February):1–9, 2016. ISSN 0098-2202. .

H. Lerbs. Moderately loaded propellers with a finite number of blades and an arbitrary distribution of circulation. *Transactions of the Society of Naval Architects and Marine Engineers*, Vol. 60, 1952.

J. Li, Y. Qu, and H. Hua. Hydroelastic analysis of underwater rotating elastic marine propellers by using a coupled BEM-FEM algorithm. *Ocean Engineering*, 146 (January):178–191, 2017. ISSN 00298018. . URL https://doi.org/10.1016/j.oceaneng.2017.09.028.

T. Liaghat, F. Guibault, L. Allenbach, and B. Nennemann. Two-Way Fluid-Structure Coupling in Vibration and Damping Analysis of an Oscillating Hydrofoil. *Volume 4A: Dynamics, Vibration, and Control*, page V04AT04A073, 2014. . URL http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/IMECE2014-38441.

A. K. Lidtke, S. R. Turnock, and V. F. Humphrey. Use of acoustic analogy for marine propeller noise characterisation. *Proceedings of the 4th Symposium on Marine Propulsors*, (June):231–239, 2015.

A. K. Lidtke, S. R. Turnock, and Downes J. BRIDGES-UoS-03 Detailed Design of Deep and Ultra Deep Explorer Propelle. Technical report, 2018.

G. R. Liu. FEM for Plates and Shells. In *FEA- A practical Guide*, volume 7, pages 219–247. 2014. ISBN 9780080983561. .

W. LiuX, R. N, S. M, and H. GM. Nonlinear aeroelastic modelling for wind turbine blades based on blade element momentum theory and geometrically exact beam theory. *Energy*, 76(1):487–501, 2014.

T. P. Lloyd. *Large eddy simulations of inflow turbulence noise : application to tidal turbines*. PhD thesis, 2013.

T. P. Lloyd, S. R. Turnock, and V. F. Humphrey. Assessing the influence of inflow turbulence on noise and performance of a tidal turbine using large eddy

simulations. *Renewable Energy*, 71:742–754, 2014. ISSN 09601481. . URL http://dx.doi.org/10.1016/j.renene.2014.06.011.

P. Maljaars. Boundary Element Modelling Aspects for the Hydro-Elastic Analysis of Flexible Marine Propellers. pages 1–32, 2018. .

J. Mann. Wind field simulation. 13(4):269–282, 1998.

N. Mccaw. The Development of a Preliminary Tool for Investigating Fluid Structure Interaction of Propellers. 2017.

S. Merz, R. Kinns, and N. Kessissoglou. Structural and acoustic responses of a submarine hull due to propeller forces. *Journal of Sound and Vibration,*, 2009.

I. A. Milne, R. N. Sharma, R. G. J. Flay, and S. Bickerton. The Role of Onset Turbulence on Tidal Turbine Blade Loads. (December), 2010.

A. F. Molland and S. R. Turnock. *Marine Rudders and Control Surfaces*. 2007.

A. F. Molland, S. R. Turnock, and D. A. Hudson. *Ship Resistance and Propulsion*. 2011. ISBN 9780511974113. . URL http://ebooks.cambridge.org/ref/id/CBO9780511974113.

A. F. Molland, S. R. Turnock, and D. A. Hudson. Wake and Thrust Deduction. pages 144–165, 2016.

J. Moulijn. Application of Various Computational Methods to Predict the Performance and Cavitation of Ducted Propellers. (June), 2015.

S. Newman. *Basic Helicopter Aerodynamics*. 2002. ISBN 063205283X.

L. Noels, L. Stainier, J.-P. Ponthot, and J. Bonini. Combined implicit-explicit algorithms for non-linear structural dynamics. *Revue Europeenne des Elements*, 11(5), 2002. ISSN 12506559. .

A. D. Otero and F. L. Ponta. Structural Analysis of Wind-Turbine Blades by a Generalized Timoshenko Beam Model. *Journal of Solar Energy Engineering*, 132(1): 011015, 2010. ISSN 01996231. . URL http://solarenergyengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1473753.

A. D. Otero and F. L. Ponta. Structural Analysis of Wind-Turbine Blades by a Generalized Timoshenko Beam. (October), 2015. .

S. Oye. Dynamic stall simulated as time lag of separation. In *In: Proceedings of the 4th IEA Symposium on the Aerodynamics of Wind Turbines. Rome, Italy.*, 1991.

M. P. Paidoussis. Fluid-Structure Interactions (Second Edition). chapter Vibrations. 2014. .

L. Patruno and M. Ricci. A systematic approach to the generation of synthetic turbulence using spectral methods. *Comput. Methods Appl. Mech. Engrg.*, 340: 881–904, 2018. ISSN 0045-7825. . URL https://doi.org/10.1016/j.cma.2018.06.028.

F. S. Pereira, L. Eça, and G. Vaz. Verification and Validation Exercises for the Flow Around the KVLCC2 Tanker at Model and Full-Scale Reynolds Numbers Veri fi cation and Validation exercises for the fl ow around the KVLCC2 tanker at model and full-scale Reynolds numbers. *Ocean Engineering*, 129(March):133–148, 2017. ISSN 0029-8018. . URL http://dx.doi.org/10.1016/j.oceaneng.2016.11.005.

A. B. Phillips. *Simulations of a Self Propelled Autonomous Underwater Vehicle*. PhD thesis, 2009. URL http://eprints.soton.ac.uk/158357/.

L. Prandtl. Application of modern hydrodynamics to aeronautics. *NACA Annual Report*, 7:157–215, 1921.

R. D. Mindlin. Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates. *ASME Journal of Applied Mechanics,*, Vol. 18:pp. 31–38., 1951.

B. W. Ramberg and S. Levy. Calculation of Stresses and Natural Frequencies for a Rotating Propeller Blade Vibrating. *Journal of Research of the National Bureau of Standards*, 21, 1938.

D. Rijpkema. Viscous Flow Simulations of Propellers in Different Reynolds Number Regimes. (April 2016), 2015.

H. Sasajima. On the estimation of wake of ships. *Proceedings of the Eleventh ITTC*, 1966.

H. Saunders. *Hydrodynamics in Ship Design*. 1961.

L. Savio. Measurements of the deflection of a flexible propeller blade by means of stereo imaging. *Fourth International Symposium on Marine Propulsors*, (June), 2015.

J. A. Schetz and S. Favin. Numerical Solution for the Near Wake of a Body with Propeller. *Journal of Hydronautics*, 11(4):136–141, 1977.

S. Seng, J. J. Jensen, and Š. Malenica. Global hydroelastic model for springing and whipping based on a free-surface CFD code (OpenFOAM). *International Journal of Naval Architecture and Ocean Engineering*, 6(4):1024–1040, 2014. ISSN 20926790. .

Z. Shen and R. Korpus. Numerical Simulations of Ship Self-Propulsion and Maneuvering Using Dynamic Overset Grids in OpenFOAM. (August), 2015.

K. T. Skaar and C. A. Carlsen. ASPECTS FOR FINITE ELEMENT ANALYSIS OF SHIP VIBRATION. 1980.

A. Smirnov, S. Shi, and I. Celik. Technique for Large Eddy Simulations and Particle-Dynamics Modeling. *Journal of Fluids Engineering*, 123(June), 2001. .

H. Snel. Heuristic modelling of dynamic stall characteristics. In *EWEC Conference Dublin Castle, Ireland*, 1997.

Z. Song and C. Su. Computation of Rayleigh Damping Coefficients for the Seismic Analysis of a Hydro-Powerhouse. 2017(2), 2017.

P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *30th Aerospace Sciences Meeting and Exhibit*, (April 2015), 1992. ISSN 00341223. . URL http://arc.aiaa.org/doi/10.2514/6.1992-439.

K. D. Squires. Detached Eddy Simulation : Current Status And Perspectives. 2003.

F. Stern, T. Kim, V. Patel, and C. H. Computation of Viscous Flow around rpropeller-shaft configurations. *ONR Report*, (305), 1986.

A. Subramanian. Simulation of Propeller Hull Interaction Using Ranse Solver. 1(3): 189–208, 2010.

Y. Sung, S.-H. Park, K.-S. Ahn, S.-H. Chung, S. Shin, and Jae-Hyoung. Evaluation on Deep Water Manoeuvring Performances of KVLCC2 Based on PMM Test and RANS Simulation. In *Proceedings of SIMMAN 2014*, 2014.

T. Suzuki and H. Mahfuz. Analysis of large-scale ocean current turbine blades using Fluid Structure Interaction and blade element momentum theory. *Ships and Offshore Structures*, 13(5):451–458, 2018. ISSN 1744-5302. . URL https://doi.org/10.1080/17445302.2017.1418621.

M. Tannous, P. Cartraud, D. Dureisseix, and M. Torkhani. A beam to 3D model switch in transient dynamic analysis. *Finite Elements in Analysis and Design*, 91:95–107, 2014. ISSN 0168874X. . URL http://dx.doi.org/10.1016/j.finel.2014.07.003.

L. Taylor, R. Pankajakshan, M. Jiang, C. Sheng, W. Briley, D. Whitfield, F. Davoudzadeh, D. Boger, H. Gibeling, J. Gorski, and A. E. Large-scale simulations for maneuvering submarines and propulsors. In *29th AIAA, Plasmadynamics and Lasers Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, jun 1998. . URL https://doi.org/10.2514/6.1998-2930.

J. Tian, Z. Zhang, Z. Ni, and H. Hua. Flow-induced vibration analysis of elastic propellers in a cyclic inflow: An experimental and numerical study. *Applied Ocean Research*, 65:47–59, 2017. ISSN 01411187. . URL http://dx.doi.org/10.1016/j.apor.2017.03.014.

S. Timoshenko. *Strength_Of_Materials_parts_IandII-Timoshenko.pdf*. 1930.

R. Townsin. The ITTC line- its genesis and correlation allowance. *The Naval Architect*, 1985.

M. Turner. Stiffness and Deflection of analysis complex structures. *Journal of the Aeronautical Sciences*, 1956.

S. R. Turnock. *Prediction of ship rudder-propeller interaction using parallel computations and wind tunnel measurements*. PhD thesis, University of Southampton, 1993.

S. R. Turnock, A. Phillips, and M. Furlong. URANS SIMULATIONS OF STATIC DRIFT AND DYNAMIC MANOUVERES OF THE KVLCC2 TANKER. *In: Proceedings of the SIMMAN 2008: Workshop on Verification and Validation of Ship Manoeuvring Simulation Methods*, pages 4–9, 2008.

S. R. Turnock, A. B. Phillips, J. Banks, and R. Nicholls-Lee. Modelling tidal current turbine wakes using a coupled RANS-BEMT approach as a tool for analysing power capture of arrays of turbines. *Ocean Engineering*, 38(11-12):1300–1307, 2011. ISSN 00298018. . URL http://dx.doi.org/10.1016/j.oceaneng.2011.05.018.

W. Van Hoydonck, G. Delefortrie, K. Eloot, P. Peeters, and F. Mostaert. Simman 2014. Technical report, 2014.

P. Veers. Modeling Stochastic Wind Loads , on Vertical Axis Wind Turbines. Technical Report September, 1984.

G. Venkatesan and W. Clark. Submarine Maneuvering Simulations of ONR Body 1, 2007. URL http://dx.doi.org/10.1115/OMAE2007-29516.

E. Ventsel. *Thin Plates and Shells*. 2001. ISBN 0824705750.

V. M. Viitanen, A. Hynninen, L. Lars, R. Klose, T. Siikonen, and J. Tanttari. CFD and CHA simulation of underwater noise induced by a marine propeller in two-phase flows. *Proceedings of the Fifth International Symposium on Marine Propulsors*, 1(June), 2017.

Vysoký. Current Capabilities of Modal Analysis of Aircraft Propeller in ANSYS Mechanical Environment. *Advances in Military Technology*, 12(1):33–47, 2017. ISSN 18022308. . URL https://apl.unob.cz/dam/93.

J.-b. Wang and M. J. Bobo. The Specialist Committee on Scaling of Wake Field. In *Proceedings of the 26th ITTC*, volume II, pages 379–417, 2011.

D. Wilcox. *Turbulence Modelling for CFD*. 1993.

B. Winden. *Powering Performance of a Self Propelled Ship in Waves*. PhD thesis, 2014.

T. Xing, S. Bhushan, and F. Stern. Vortical and turbulent structures for KVLCC2 at drift angle 0, 12, and 30 degrees. *Ocean Engineering*, 55:23–43, 2012. ISSN 00298018. . URL http://dx.doi.org/10.1016/j.oceaneng.2012.07.026.

F. Xing, T., Bhushan, S., and Stern. Vortical and turbulent structures for KVLCC2 at drift angle 0, 12, and 30 degrees. *Ocean Engineering*, 55:2343, 2012.

Y. L. Young. Fluid-structure interaction analysis of flexible composite marine propellers. *Journal of Fluids and Structures*, 24(6):799–818, 2008. ISSN 08899746. .

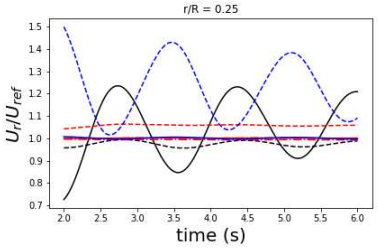
W. Yu, C. S. Ferreira, G. V. Kuik, and D. Baldacchino. Verifying the Blade Element Momentum Method in unsteady , radially varied , axisymmetric loading using a vortex ring model. (June 2016):269–288, 2017. .

O. Zienkiewicz. *The Finite Element Method*. 1977.

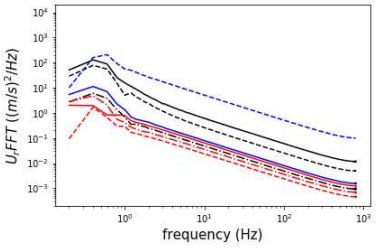
A. Zobeiri. Effect of Hydrofoil Trailing Edge Geometry on the Wake Dynamics. 5218: 132, 2012. . URL http://library.epfl.ch/theses/?nr=5218.

Appendix A

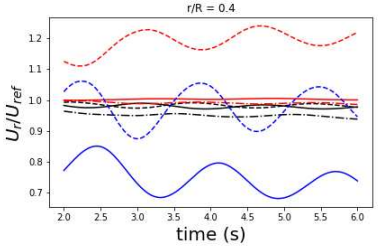
KVLCC2 wake plots



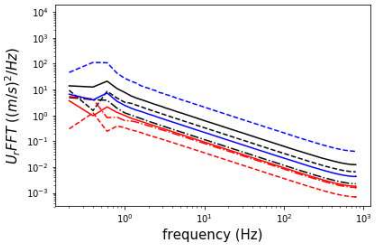
(A) rR = 0.25 unsteady velocity for drift angle = 0^o



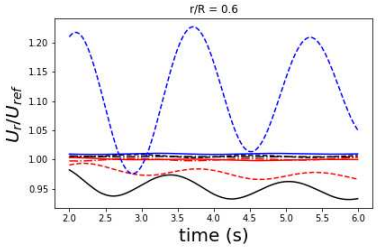
(B) rR = 0.25 unsteady velocity FFT for drift angle = 0^o



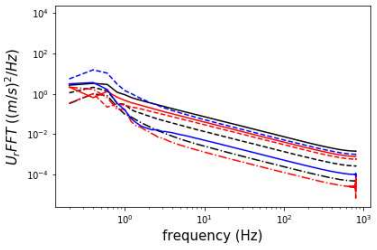
(C) rR = 0.4 unsteady velocity for drift angle = 0^o



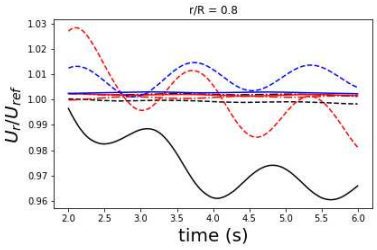
(D) rR = 0.4 unsteady velocity FFT for drift angle = 0^o



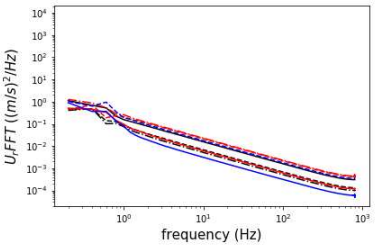
(E) rR = 0.6 unsteady velocity for drift angle = 0^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 0^o

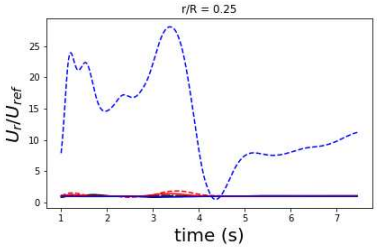


(G) rR = 0.8 unsteady velocity for drift angle = 0^o

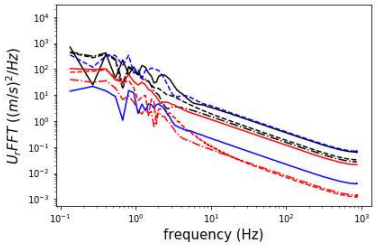


(H) rR = 0.8 unsteady velocity FFT for drift angle = 0^o

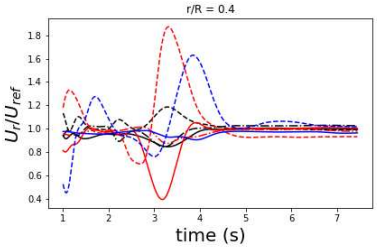
FIGURE A.1: Unsteady Velocity plots for 0^o drift angle



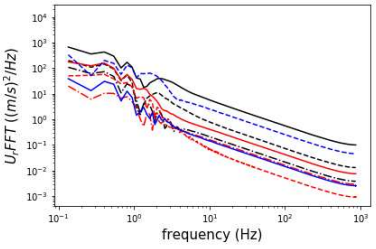
(A) rR = 0.25 unsteady velocity for drift angle = 4^o



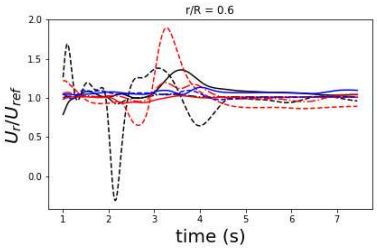
(B) rR = 0.25 unsteady velocity FFT for drift angle = 4^o



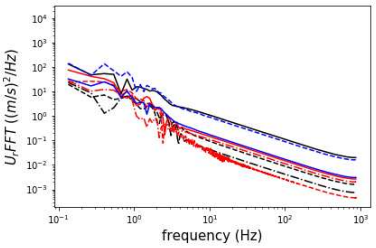
(C) rR = 0.4 unsteady velocity for drift angle = 4^o



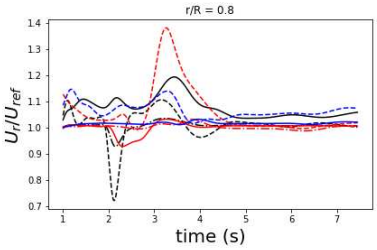
(D) rR = 0.4 unsteady velocity FFT for drift angle = 4^o



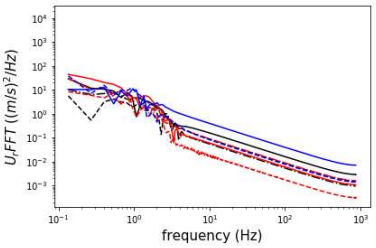
(E) rR = 0.6 unsteady velocity for drift angle = 4^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 4^o

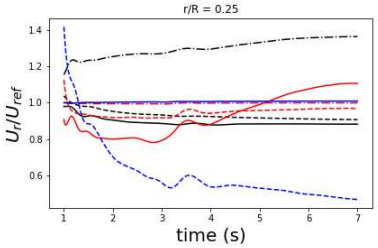


(G) rR = 0.8 unsteady velocity for drift angle = 4^o

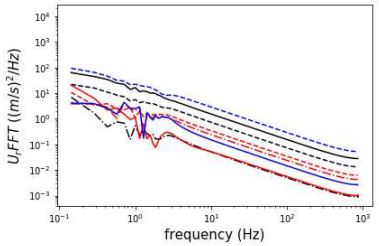


(H) rR = 0.8 unsteady velocity FFT for drift angle = 4^o

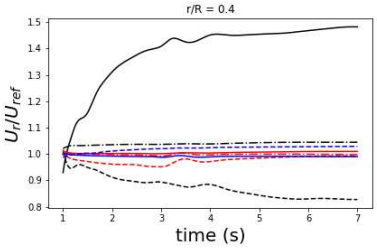
FIGURE A.2: Unsteady Velocity plots for 4^o drift angle



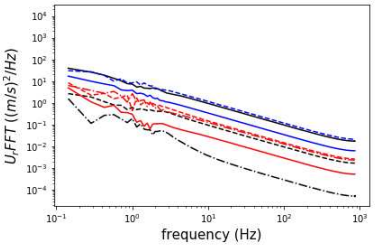
(A) rR = 0.25 unsteady velocity for drift angle = 6^o



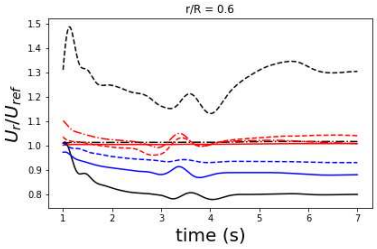
(B) rR = 0.25 unsteady velocity FFT for drift angle = 6^o



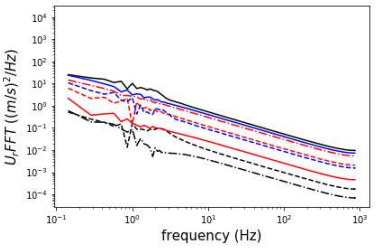
(C) rR = 0.4 unsteady velocity for drift angle = 6^o



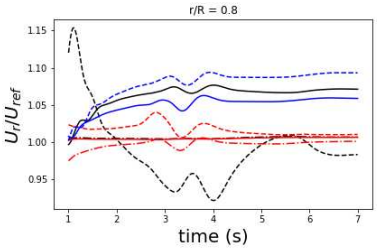
(D) rR = 0.4 unsteady velocity FFT for drift angle = 6^o



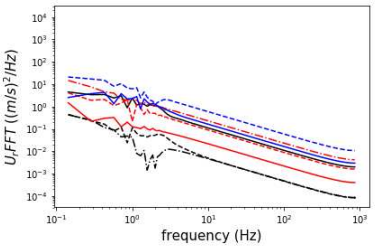
(E) rR = 0.6 unsteady velocity for drift angle = 6^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 6^o

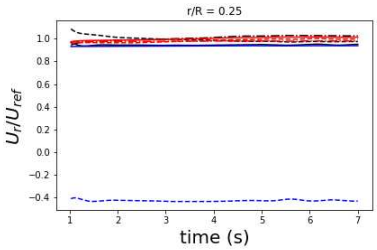


(G) rR = 0.8 unsteady velocity for drift angle = 6^o

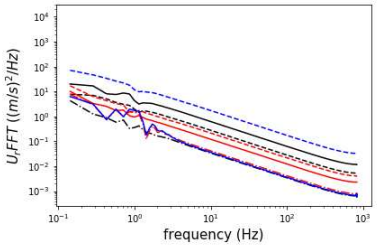


(H) rR = 0.8 unsteady velocity FFT for drift angle = 6^o

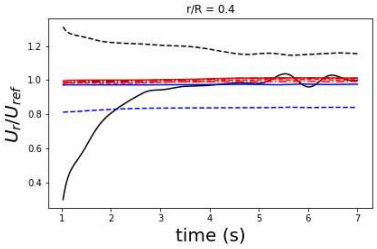
FIGURE A.3: Unsteady Velocity plots for 6^o drift angle



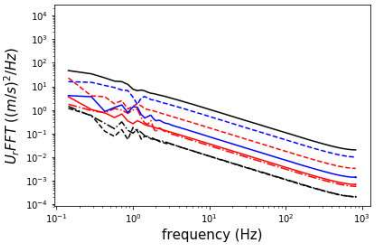
(A) rR = 0.25 unsteady velocity for drift angle = 8^o



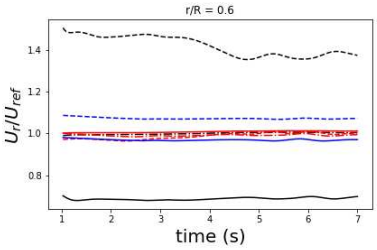
(B) rR = 0.25 unsteady velocity FFT for drift angle = 8^o



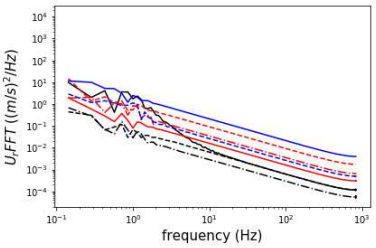
(C) rR = 0.25 unsteady velocity for drift angle = 8^o



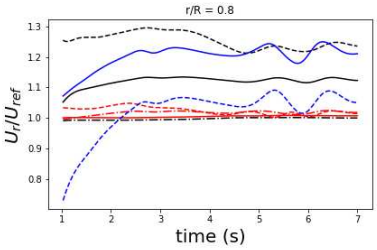
(D) rR = 0.25 unsteady velocity FFT for drift angle = 8^o



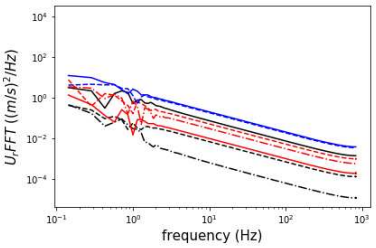
(E) rR = 0.25 unsteady velocity for drift angle = 8^o



(F) rR = 0.25 unsteady velocity FFT for drift angle = 8^o

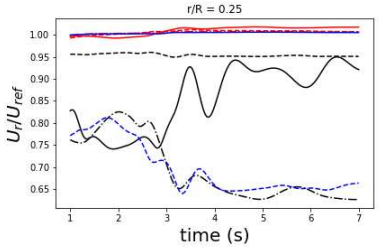


(G) rR = 0.25 unsteady velocity for drift angle = 8^o

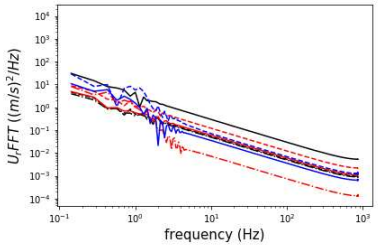


(H) rR = 0.25 unsteady velocity FFT for drift angle = 8^o

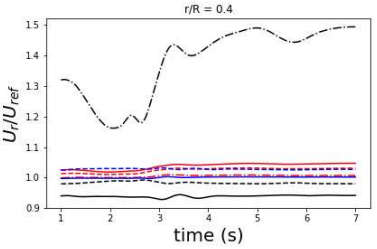
FIGURE A.4: Unsteady Velocity plots for 8^o drift angle



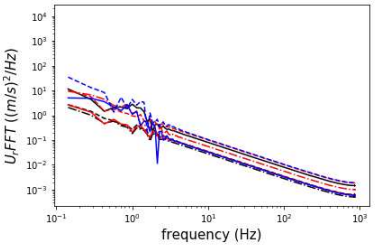
(A) rR = 0.25 unsteady velocity for drift angle = 12^o



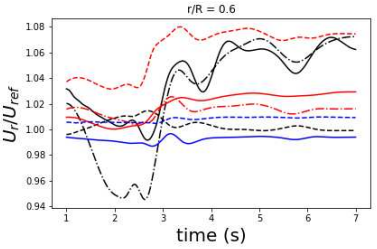
(B) rR = 0.25 unsteady velocity FFT for drift angle = 12^o



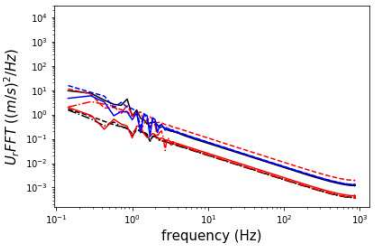
(C) rR = 0.4 unsteady velocity for drift angle = 12^o



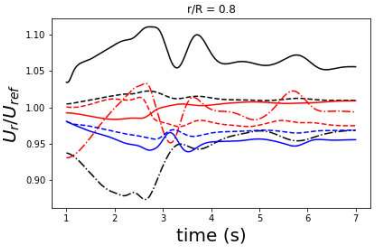
(D) rR = 0.4 unsteady velocity FFT for drift angle = 12^o



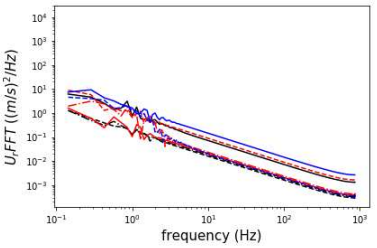
(E) rR = 0.6 unsteady velocity for drift angle = 12^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 12^o

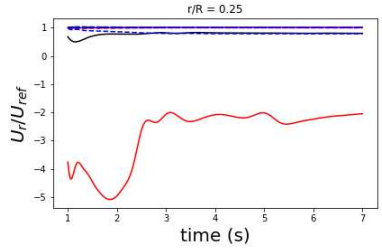


(G) rR = 0.8 unsteady velocity for drift angle = 12^o

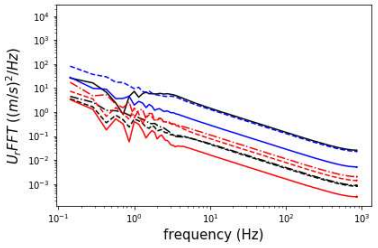


(H) rR = 0.8 unsteady velocity FFT for drift angle = 12^o

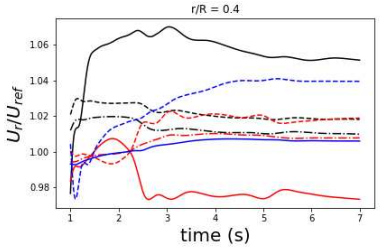
FIGURE A.5: Unsteady Velocity plots for 12^o drift angle



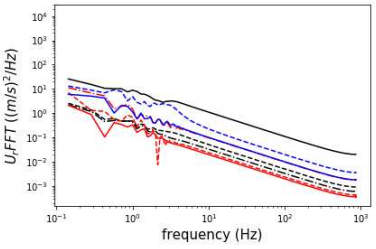
(A) rR = 0.25 unsteady velocity for drift angle = 16^o



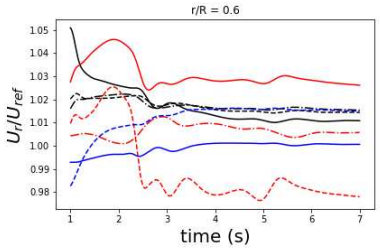
(B) rR = 0.25 unsteady velocity FFT for drift angle = 16^o



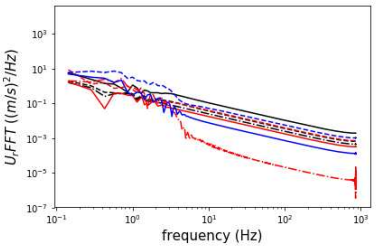
(C) rR = 0.4 unsteady velocity for drift angle = 16^o



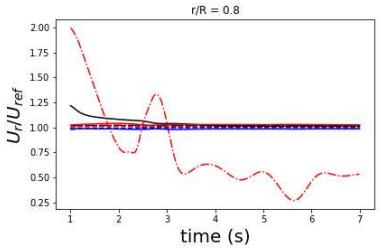
(D) rR = 0.4 unsteady velocity FFT for drift angle = 16^o



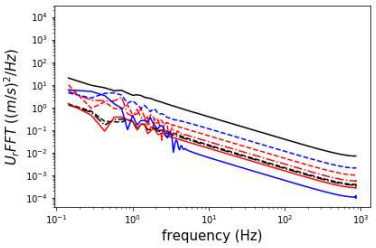
(E) rR = 0.6 unsteady velocity for drift angle = 16^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 16^o

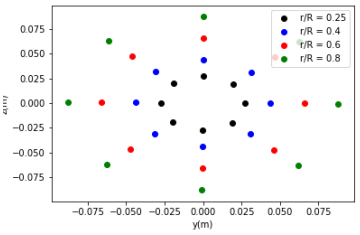


(G) rR = 0.8 unsteady velocity for drift angle = 16^o

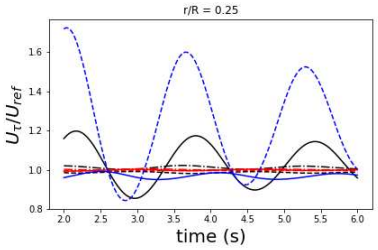


(H) rR = 0.8 unsteady velocity FFT for drift angle = 16^o

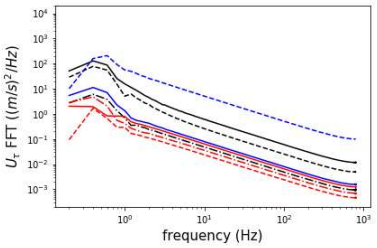
FIGURE A.6: Unsteady Velocity plots for 16^o drift angle



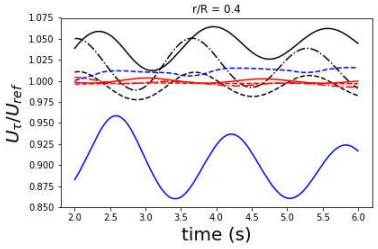
(A) Plotting positions for unsteady wake



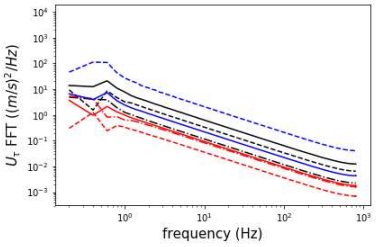
(B) rR = 0.25 unsteady velocity for drift angle = 0^o



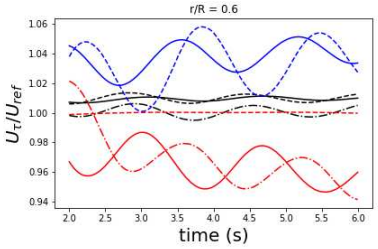
(C) rR = 0.25 unsteady velocity FFT for drift angle = 0^o



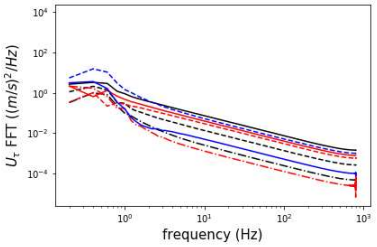
(D) rR = 0.4 unsteady velocity for drift angle = 0^o



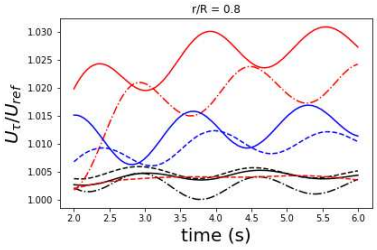
(E) rR = 0.4 unsteady velocity FFT for drift angle = 0^o



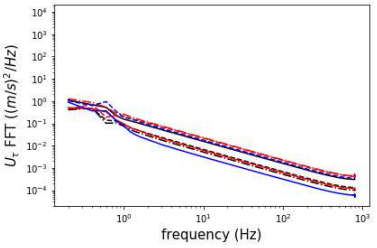
(F) rR = 0.6 unsteady velocity for drift angle = 0^o



(G) rR = 0.6 unsteady velocity FFT for drift angle = 0^o

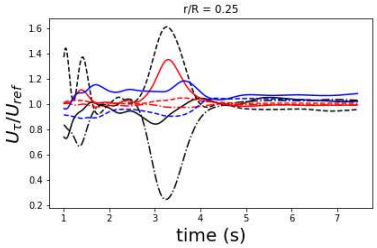


(H) rR = 0.8 unsteady velocity for drift angle = 0^o

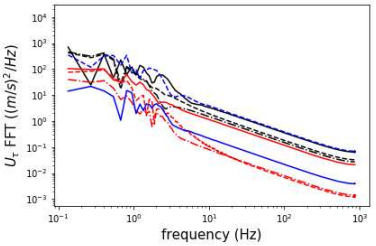


(I) rR = 0.8 unsteady velocity FFT for drift angle = 0^o

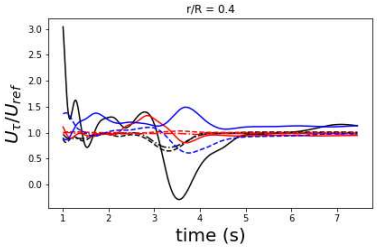
FIGURE A.7: Unsteady Velocity plots for 0^o drift angle



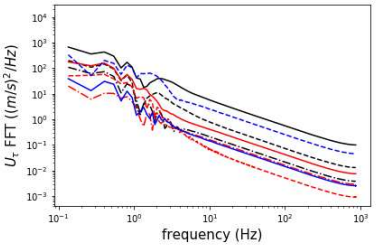
(A) rR = 0.25 unsteady velocity for drift angle = 4^o



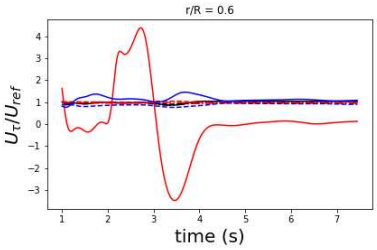
(B) rR = 0.25 unsteady velocity FFT for drift angle = 4^o



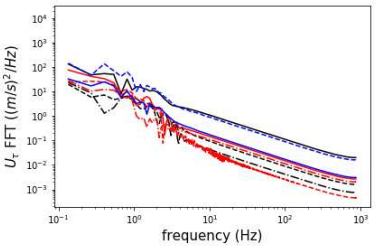
(C) rR = 0.4 unsteady velocity for drift angle = 4^o



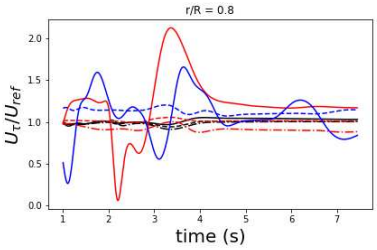
(D) rR = 0.4 unsteady velocity FFT for drift angle = 4^o



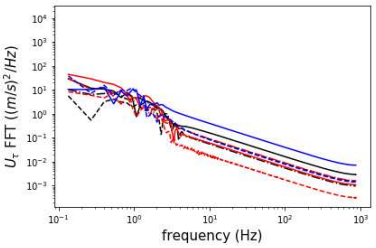
(E) rR = 0.6 unsteady velocity for drift angle = 4^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 4^o

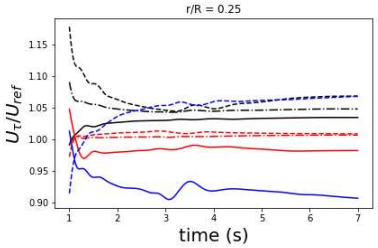


(G) rR = 0.8 unsteady velocity for drift angle = 4^o

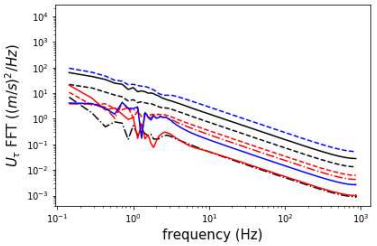


(H) rR = 0.8 unsteady velocity FFT for drift angle = 4^o

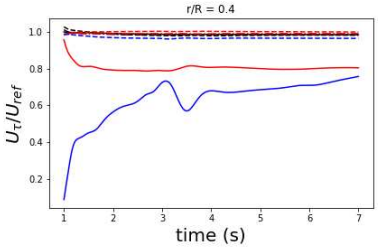
FIGURE A.8: Unsteady Velocity plots for 4^o drift angle



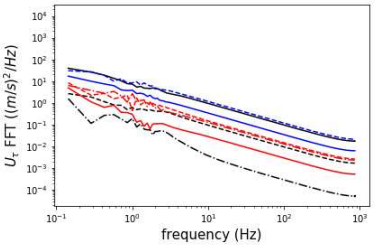
(A) rR = 0.25 unsteady velocity for drift angle = 6^o



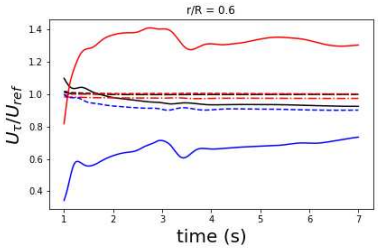
(B) rR = 0.25 unsteady velocity FFT for drift angle = 6^o



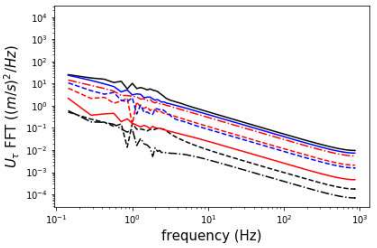
(C) rR = 0.4 unsteady velocity for drift angle = 6^o



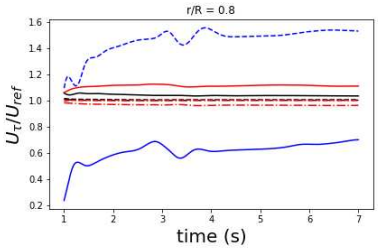
(D) rR = 0.4 unsteady velocity FFT for drift angle = 6^o



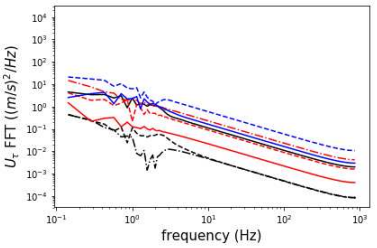
(E) rR = 0.6 unsteady velocity for drift angle = 6^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 6^o

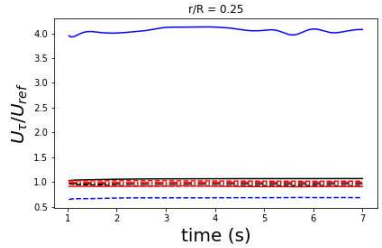


(G) rR = 0.8 unsteady velocity for drift angle = 6^o

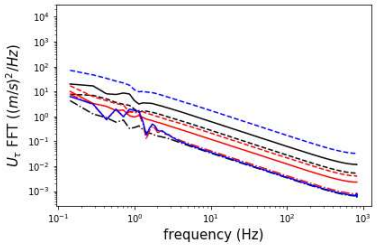


(H) rR = 0.8 unsteady velocity FFT for drift angle = 6^o

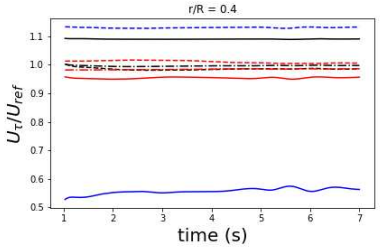
FIGURE A.9: Unsteady Velocity plots for 6^o drift angle



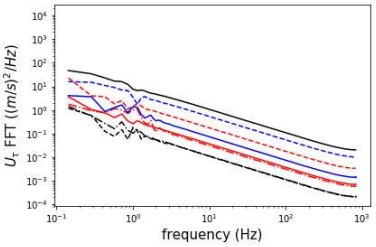
(A) rR = 0.25 unsteady velocity for drift angle = 8^o



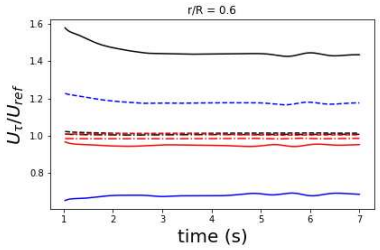
(B) rR = 0.25 unsteady velocity FFT for drift angle = 8^o



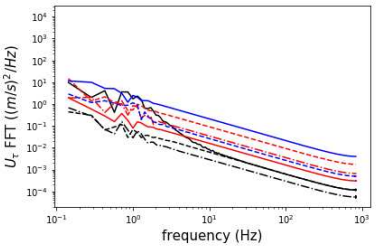
(C) rR = 0.25 unsteady velocity for drift angle = 8^o



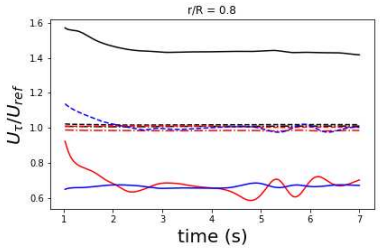
(D) rR = 0.25 unsteady velocity FFT for drift angle = 8^o



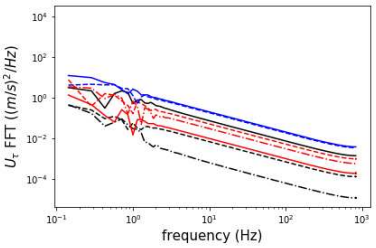
(E) rR = 0.25 unsteady velocity for drift angle = 8^o



(F) rR = 0.25 unsteady velocity FFT for drift angle = 8^o

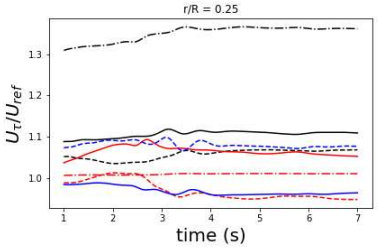


(G) rR = 0.25 unsteady velocity for drift angle = 8^o

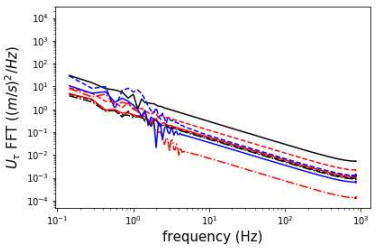


(H) rR = 0.25 unsteady velocity FFT for drift angle = 8^o

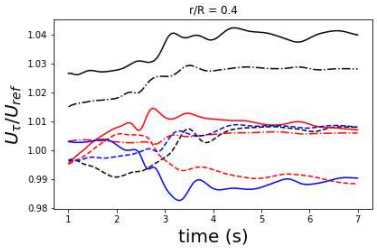
FIGURE A.10: Unsteady Velocity plots for 8^o drift angle



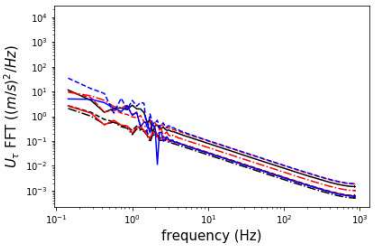
(A) rR = 0.25 unsteady velocity for drift angle = 12^o



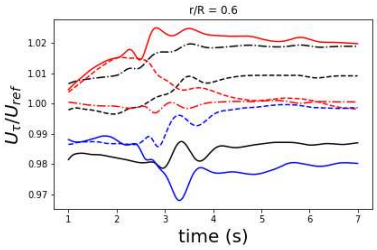
(B) rR = 0.25 unsteady velocity FFT for drift angle = 12^o



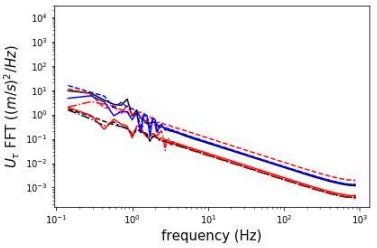
(C) rR = 0.4 unsteady velocity for drift angle = 12^o



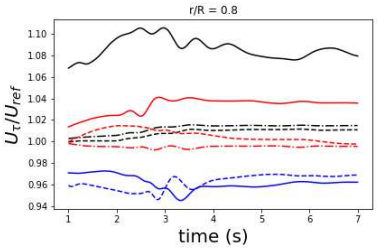
(D) rR = 0.4 unsteady velocity FFT for drift angle = 12^o



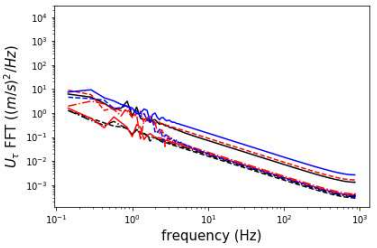
(E) rR = 0.6 unsteady velocity for drift angle = 12^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 12^o

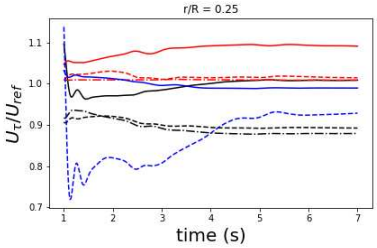


(G) rR = 0.8 unsteady velocity for drift angle = 12^o

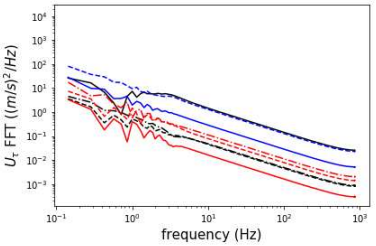


(H) rR = 0.8 unsteady velocity FFT for drift angle = 12^o

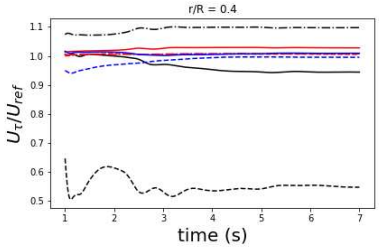
FIGURE A.11: Unsteady Velocity plots for 12^o drift angle



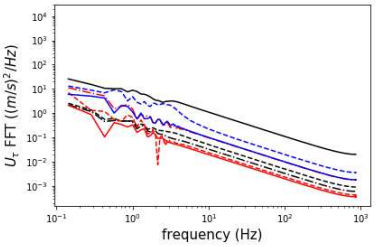
(A) rR = 0.25 unsteady velocity for drift angle = 16^o



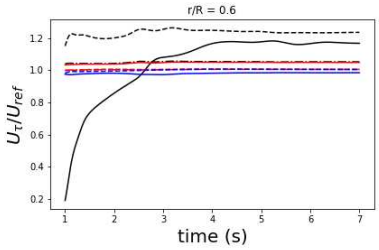
(B) rR = 0.25 unsteady velocity FFT for drift angle = 16^o



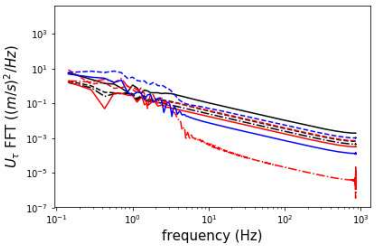
(C) rR = 0.4 unsteady velocity for drift angle = 16^o



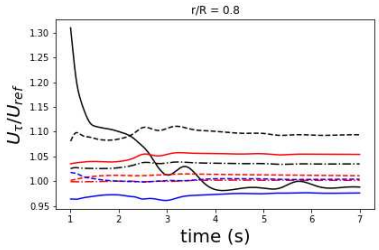
(D) rR = 0.4 unsteady velocity FFT for drift angle = 16^o



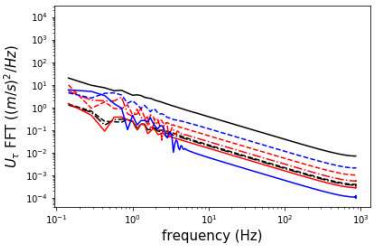
(E) rR = 0.6 unsteady velocity for drift angle = 16^o



(F) rR = 0.6 unsteady velocity FFT for drift angle = 16^o



(G) rR = 0.8 unsteady velocity for drift angle = 16^o



(H) rR = 0.8 unsteady velocity FFT for drift angle = 16^o

FIGURE A.12: Unsteady Velocity plots for 16^o drift angle

```
def BEMT(J,N_blades,chord,P_D,BA_ratio,
lift_curve_slope,root_thickness,x_R_list,local_P_D,chord_diameter,
thickness_distribution,MC_0,n,D,N_circum,Va,db,AllUx,Allr,AllUtau,data,Cl_list = [],Cd_list = []
'''
Main Blade Element Momentum Theory Function. Uses algorithm form referenced
book.

Parameters
----------

J: float
```

```
advance ratio
P_D: float
pitch diameter ratio at 0.7 radius
BA_ratio: float
blade area ratio
N_blades: int
number of blades
chord: array of floats
chord of blade at different radial positions
relative_pitch_distribution: array of float
pitch distribution at radial positions relative to pitch distibution at
x_R = 0.7
lift_curve_slope: float
lift curve slope of blade
root_thickness: float
thickness at blade root
x_R_list
local_P_D: array
local pitch/Diameter distribution

chord_diameter: array
chord/diameter distribution
thickness_distribution: array
thickness distribution
MC_0: array
camber distribution
n: float
rps
D: float
diameter
N_circum: int
number of circumferntial sections
Va: float
ship axial velocity
db: database
Cl alpha database
AllUx: array
local axial velocity
Allr: array
local radial velocity
AllUtau: array
local tangential velocity
data: database
Cl Cd database
Cl_list: list
list of Cl time history
Cd_list: list
list of Cd time history
alpha_list: list
list of alpha time history
hysteresis:bool
Hysteresis flag on or off
Clhyst:float
Cl hysteresis loop
hdotdot: float
heave acceleration
alphadot: float
angle of attack acceleration
hdot:float
```

```
heave velocity
dt: float
timestep
Returns
-------
Total_Thrust: float
total thrust coefficient along blade
Total_Torque: float
total torque coefficient along blade
Open_eta: float
open water efficiency

Cl: array
Cl distribution
U: array
Velocity distribution
KT_array: array
thrust coefficient distribution
alpha: array
angle of attack distribution
'''

assert N_blades >= 3 and N_blades <=5,'Number of blades must be between 3 \
and 5 but is of value {}'.format(N_blades)
assert type(J) == np.float or np.float64,'J is not of type float but is of type {}'.format(type(
assert type(P_D) == np.float or np.float64,'P_D is not of type float but is of type {}'.format(t
assert type(BA_ratio) == np.float or np.float64,'BA_ratio is not of type float but is of type {}
assert type(N_blades) == int,'N_blades is not of type int but is of type {}'.format(type(N_blade
assert type(chord) == np.array or np.ndarray,'chord is not of type array but is of type {}'.form
assert type(lift_curve_slope) == np.float or np.float64,'lift_curve_slope is not of type float b
assert type(root_thickness) == np.float or np.float64,'root_thickness is not of type float but i
# create arrays

phi_plus_alpha = np.zeros_like(chord)
tan_psi = np.zeros_like(chord)
K_series = np.zeros_like(chord)
thickness_chord = np.zeros_like(chord)
KT_dx = np.zeros_like(chord)
Cl = np.zeros_like(chord)
Cd = np.zeros_like(chord)
KQ_dx = np.zeros_like(chord)
phi = np.zeros_like(chord)
alpha = np.zeros_like(chord)

CC= np.zeros_like(chord)
U = np.zeros_like(chord)

circumferences = np.arange(0,1,1/N_circum)
KT_array = np.zeros((len(chord),N_circum))
KQ_array = np.zeros_like(KT_array)
inflowfunc = False
wt_axial = wakeinflow(x_R_list,N_circum, inflowfunc)
wt_axial = (Va - AllUx)/Va

U_tau = tangentialInflowfunction(x_R_list,N_circum,inflowfunc)
U_tau = np.zeros_like(wt_axial)
U_tau = AllUtau

oldmethod =  False
```

```
test = True

aList = []
apList = []



for j,circumference in enumerate(circumferences):

for i,x_R in enumerate(x_R_list):
# compute local advance ratio
J_local = J*(1-wt_axial[j,i])
# local tangential wake
tangential_wake = U_tau[j,i]*(1-wt_axial[j,i])

#
# thickness/chord
thickness_chord[i] = thickness_distribution[i] / chord_diameter[i]
# set initial angle of attack to 0
alpha[i] = 0.0
# set number of alpha iterations to 0
alpha_iterations = 0
# set convergence flag to 0
alpha_converge = 0
# start of alpha convergence loop
tc = thickness_chord[i]
# section camber
mc = MC_0[i]
# find thickness and camber bounds
tcBounds,mcBounds,wtc,wmc = interpolatedtcmc(tc,mc,data)


while alpha_iterations < 200 and alpha_converge == 0:
alpha_iterations += 1
# calculate tan of inflow angle
tan_psi[i] = J_local / (np.pi*x_R)
# induced flow angle plus angle of attack
phi_plus_alpha[i] = np.arctan2(local_P_D[i],(np.pi*x_R))

# inflow angle
phi[i] = phi_plus_alpha[i] - alpha[i]

# ideal efficieny
eta_ideal = tan_psi[i] / np.tan(phi[i])
# initially set efficiency to ideal
eta  = 0.9 * eta_ideal

gamma = 0.0 # zero for ideal efficiency
a_pp = tangential_wake*tan_psi[i] #wake rotation factor a''

# Goldstein Correction

SF=(N_blades/(2.0*np.tan(phi[i])*x_R))-0.5
F1=np.cosh(SF)
F2=np.cosh(SF*x_R)
F3=F2/F1
F4=np.arccos(F3)
K = (2.0*F4/np.pi)
```

```
# store factor in array
K_series[i] = K

# set number of efficiency iterations to 0
eta_iterations = 0
eta_converge = 0
while eta_iterations < 200 and eta_converge == 0:
# axial inflow factor
a = (1 - eta_ideal+a_pp)/(eta_ideal + ((tan_psi[i]**2)/eta))

# local thrust coefficient
R = D/2

# local thrust coefficient
KT_dx[i] = np.pi*(J_local**2) * x_R * K * a*(1+a)


# circumferential inflow factor (a')
a_p = 1-eta_ideal*(1+a+a_pp)

Cl[i] = KT_dx[i]/(((np.pi**2)/4)*(N_blades*chord_diameter[i])*(x_R**2)*((1-a_p+a_pp)**2) \
* (1/np.cos(phi[i]))*(1-np.tan(phi[i])*np.tan(gamma)))



gamma = np.arctan2(Cd[i],Cl[i])
# compute efficiency
eta_calculated = tan_psi[i]/(np.tan((phi[i]+gamma)))

eta_iterations += 1
if np.abs(eta_calculated - eta) < 0.01:
# if old efficiency = new efficiency
# set convergence flag to 1 to stop iteration
eta_converge = 1

# KF is lambda from goldstein correction
KF = np.tan(phi[i])*x_R
U1 = -0.65*KF*KF+1.1*KF+0.664
U2 = (0.85+(KF-0.3)*(-4.0+(KF-0.4)*(15.42-47.95*(KF-0.5))))
U2 = -0.09+(KF-0.2)*U2
U3 = (1.375+(KF-0.3)*(-3.75+(KF-0.4)*(20.85-75.7875*(KF-0.5))))
U3 = -0.2+(KF-0.2)*U3;
K1 = U1+(BA_ratio-0.4)*(U2+U3*(BA_ratio-0.8))

if oldmethod:
#CC[i] = 1


db2 = interpolator(db,tc,mc)
db2.prepareReWeights(Re)
dCl_dMC = db2.getdCldmc(db.ReValues)


MC_alpha_zero = ((Cl[i]/dCl_dMC) * CC[i])
#lift_curve_slope = 5.7
camber_deficit = (MC_alpha_zero) + MC_0[i]

alpha_calculated = (camber_deficit*dCl_dMC)/lift_curve_slope
```

```
#alpha_calculated = alpha_calculated *np.pi/180
continue
if test:



tc = thickness_chord[i]
mc = MC_0[i]
nu = 1.004e-6 #kinematic viscosity of water at 20C

U[i] = ((1+a)* AllUx[j,i]) /np.sin(phi[i])

Re = chord[i] * U[i] /  nu

db2 = interpolator(db,tc,mc)

ArturTestMethod = False
if ArturTestMethod:
db2.getPerformanceCurve(Re)



dCl_dMC = db2.getdCldmc(db.ReValues)
MC_alpha_zero = ((Cl[i]/dCl_dMC) * CC[i])


alpha_calculated,Cd[i],lift_curve_slope = db2.getParamsForCl(Cl[i],alpha[i])
alpha_calculated = hysteresisAlpha(Cl[i],alpha_calculated,Re,U[i],chord[i],hdotdot,alphadot,hdot,lift_c
else:
alpha_calculated,Cd[i],lift_curve_slope = GetAlpha(Re,alpha[i],tcBounds,mcBounds,wtc,wmc,data,Cl[i])



Cd[i] *= 0.75

alpha_calculated = np.deg2rad(alpha_calculated)
alphaPrime = alpha_calculated


continue

else:
# set the new efficiency to calculated efficiency
eta = eta_calculated
if np.abs((alpha[i] - alphaPrime)/alpha[i])<0.001:

# compute local torque coefficient
KQ_dx[i]=4.935*J_local*(x_R**3)*K*a_p*(1+a)

KQ_array[i,j] = KQ_dx[i]
#KT_dx[i] *= (1-np.tan(phi[i])*np.tan(gamma) )
KT_array[i,j] = KT_dx[i]
#U[i] = np.pi*n*D*x_R*(1-a_p)*(1/np.cos(phi[i]))

# set convergence flag to 1 to stop iteration
alpha_converge = 1
aList.append(a)
apList.append(a_p)
```

```
else:

alpha[i] = (alphaPrime + alpha[i]) / 2.0


if x_R_list[-1] != 1:
# set tip coefficients to 0
np.append(K_series,0)
np.append(local_P_D,local_P_D[-1])
np.append(alpha,alpha[-1])
#KT_array[-1] = 0
#KQ_array[-1] = 0
elif x_R_list[-1] == 1:
K_series[-1] = 0
local_P_D[-1] = local_P_D[-2]
alpha[-2] = alpha[-1]
KT_array[-1] = 0
KQ_array[-1] = 0




Total_Thrust = TrapIntegration2D(KT_array,x_R_list,circumferences)
Total_Torque = TrapIntegration2D(KQ_array,x_R_list,circumferences)
# open water efficiency
Open_eta=(J*Total_Thrust)/(2.0*3.1416*Total_Torque)


return Total_Thrust , Total_Torque, Open_eta, Cl,U,KT_array,alpha
```

LISTING A.1: Blade Element Momentum Theory Code

```
def finite_element_2d(nodes, IEN, ID, f,E,nu,h,rho,kappa,printMode,rhoFluid,Elementforce,onlyFor

# Numbers

N_equations = np.max(ID)+1


N_elements = IEN.shape[0]

N_nodes = nodes.shape[0]

N_dim = nodes.shape[1]

# Location matrix

LM = np.zeros((IEN.shape[1]*3,IEN.shape[0]),dtype = int)

for e in range(N_elements):

for a in range(4):

b = 3*a

LM[b,e] =   ID[IEN[e,a]]
```

```
LM[b+1,e] = ID[IEN[e,a]] + 1

LM[b+2,e] = ID[IEN[e,a]] + 2


# Global stiffness matrix and force vector


K = np.zeros((N_equations, N_equations))

M = np.zeros((N_equations, N_equations))

F = np.zeros((N_equations,))

S = np.zeros((N_elements,3))

# Loop over elements

if onlyForce:

for e in range(N_elements):


f_e = force(nodes[IEN[e,:],:], f,Elementforce[e])


for a in range(12):

A = LM[a, e]


if (A >= 0):

F[A] += f_e[a]



return F

else:


for e in range(N_elements):

k_e = stiffness_inplane(nodes[IEN[e,:],:],E,nu,h[e]) + stiffness_shear(nodes[IEN[e,:],:],E,nu,h[e],kapp

f_e = force(nodes[IEN[e,:],:], f,Elementforce[e])


m_e = mass(nodes[IEN[e,:],:],h[e],rho) + addedMassMatrix(nodes[IEN[e,:],:],rhoFluid,h[e])


for a in range(12):

A = LM[a, e]
```

```
for b in range(12):

B = LM[b, e]


if (A >= 0) and (B >= 0):

K[A, B] += k_e[a, b]

M[A ,B] += m_e[a, b]


if (A >= 0):

F[A] += f_e[a]

# Solve


D_A = np.linalg.solve(K, F)

w_A,B_A = scipy.linalg.eig(K,M)


D = np.zeros(N_nodes)

FF = np.zeros_like(D)

B = np.zeros(N_nodes)

# convert to Hz (from rads-1)

ef = np.sqrt(w_A)

ef /= (np.pi*2)

ef,B_A = sorting(ef,B_A)


for n in range(N_nodes):

if ID[n] >= 0: # Otherwise T should be zero, and we've initialized that already.

D[n] = D_A[ID[n]]

B[n] = B_A[ID[n],printMode]

FF[n] = F[ID[n]]

computeStress = True

if computeStress == True:


S = computeStresses(nodes, IEN, ID, f,E,nu,h,rho,D)
```

```
return D,ef,B,M,K,F,S


def stiffness_inplane(nodes,E,nu,h):
'''
Calculate local inplane Stiffness matrix

Parameters
-----------
nodes: array
node numbers
E: float
Youngs modulus
nu: float
poissons ratio
h: float
section thickness

Output
--------
k_ab: float
local inplane stiffness matrix

'''
def psi_k( N, dN):
B = BI_Matrix(dN)
c = materialMatrix(E,nu)

return  ((h**3)/12)*np.dot(np.dot(B.T,c),B)
k_ab = volume_quad_element(nodes, psi_k)
return k_ab
#@profile
def stiffness_shear(nodes,E,nu,h,kappa):
```

```
'''
Calculate local shear Stiffness matrix

Parameters
-----------
nodes: array
node numbers
E: float
Youngs modulus
nu: float
poissons ratio
h: float
section thickness
kappa: float
constant
Output
--------
k_ab: float
local shear stiffness matrix

'''
def psi_k( N, dN):
B = BO_Matrix(dN,N)
cs = materialMatrixShear(E,nu)
return h*kappa*np.dot(np.dot(B.T,cs),B)
k_ab = volume_quad_element2(nodes, psi_k)

return k_ab
def materialMatrixShear(E,nu):
'''
Calculate local  material matrix
```

```
Parameters

-----------

E: float

Youngs modulus

nu: float

poissons ratio


Output

--------

Material Matrix in Shear


'''

G =  E/(2*(1+nu))

cs = np.array([[G,0],

[0,G]])

return cs

def materialMatrix(E,nu):


'''

Calculate local  material matrix


Parameters

-----------

E: float

Youngs modulus

nu: float

poissons ratio


Output

--------

Material Matrix
```

```
'''

c1 = E/(1-nu**2)

c2 = np.array([[1, nu,0],

[nu,1,0],

[0,0,(1-nu)/2]])


return c1*c2


def volume_quad_reference(psi):

'''

Calculate integral using quadrature


Parameters

-----------

psi: function

funtion to be integrated


Output

--------

I: float

Integral


'''

eta = 1/np.sqrt(3)

xi1 = np.array([-eta, -eta])

xi2 = np.array([eta, -eta])

xi3 = np.array([eta, eta])

xi4 = np.array([-eta, eta])


I = (psi(xi1) + psi(xi2) + psi(xi3) + psi(xi4))


return I
```

```
def volume_quad_reference2(psi):
'''
Calculate integral using quadrature

Parameters
----------
psi: function
funtion to be integrated

Output
--------
I: float
Integral

'''
xi1 = np.array([0, 0],dtype = np.float64)

I = (psi(xi1)) * 4

return I

def volume_quad_element2(nodes, phi):
psi = lambda xi: jacobian_determinant(nodes, xi) * phi(shape_functions(xi),
global_derivs_from_local(nodes, xi))
return volume_quad_reference2(psi)

def volume_quad_element(nodes, phi):
psi = lambda xi: jacobian_determinant(nodes, xi) * phi(shape_functions(xi),
global_derivs_from_local(nodes, xi))
return volume_quad_reference(psi)

def BI_Matrix(dN):
'''
Calculate local B matrix
```

```
Parameters

-----------

dN: array

shape function derivatives


Output

--------

B: array

B matrix


'''


B0 = np.array([[0,dN[0,0],0],

[0,0,dN[0,1]],

[0,dN[0,1],dN[0,0]]])


B1 = np.array([[0,dN[1,0],0],

[0,0,dN[1,1]],

[0,dN[1,1],dN[1,0]]])


B2 = np.array([[0,dN[2,0],0],

[0,0,dN[2,1]],

[0,dN[2,1],dN[2,0]]])


B3 = np.array([[0,dN[3,0],0],

[0,0,dN[3,1]],

[0,dN[3,1],dN[3,0]]])




B = np.hstack((B0,B1,B2,B3))
```

```
return B

def B0_Matrix(dN,N):

'''

Calculate local B matrix

Parameters

-----------

dN: array

shape function derivatives

Output

--------

B: array

B matrix

'''


B0 = np.array([[dN[0,0],-N[0,0],0],

[dN[0,1],0,-N[0,0]]])

B1 = np.array([[dN[1,0],-N[0,3],0],

[dN[1,1],0,-N[0,3]]])

B2 = np.array([[dN[2,0],-N[0,6],0],

[dN[2,1],0,-N[0,6]]])

B3 = np.array([[dN[3,0],-N[0,9],0],

[dN[3,1],0,-N[0,9]]])

B = np.hstack((B0,B1,B2,B3))

return B
```

```
def sorting(ef,B):
'''
Sort matrices or arrays in order of acending frequency

parameters
----------
ef: array of floats
modal frequencies (Hz)
B: array of floats
array of mode shapes

returns
--------
f: array of floats
sorted modal frequencies (Hz)

BSorted: array of floats
sorted array of mode shapes
'''
indices = np.argsort((ef.real))
f = np.sort((ef.real))

BSorted = np.zeros_like(B,dtype=complex)
for i, index in enumerate(indices):
BSorted[:,i] = B[:,index]

return f,BSorted


def shape_functions(xi):
'''
Linear Shape Functions

Parameters
```

```
-----------

xi: array

local coordinates


Output

--------

N: array

Shape Functions


'''

N1 = 0.25*(1-xi[0])*(1-xi[1])

N2 = 0.25*(1+xi[0])*(1-xi[1])

N3 = 0.25*(1+xi[0])*(1+xi[1])

N4 = 0.25*(1-xi[0])*(1+xi[1])



N = np.array([[N1,0,0,N2,0,0,N3,0,0,N4,0,0],

[0,N1,0,0,N2,0,0,N3,0,0,N4,0],

[0,0,N1,0,0,N2,0,0,N3,0,0,N4]])


return N


def derivs_shape_functions(xi):

'''

Linear Shape Function derivitives


Parameters

-----------

xi: array

local coordinates


Output
```

```
--------

dN: array

Shape Functions derivitives

'''

dN = np.array([[0.25*(-1+xi[1]), 0.25*(-1+xi[0])],

[0.25*(1-xi[1]), 0.25*(-1-xi[0])],

[0.25*(1+xi[1]), 0.25*(1+xi[0])],

[0.25*(-1-xi[1]), 0.25*(1-xi[0])]])

return dN

def global_from_local(nodes, xi):

'''

Get global coordinates from local coordinates

Parameters

-----------

nodes: array

node numbers

xi: array

local coordinates

Output

--------

global coordinates

'''

N = shape_functions(xi)

return np.dot(N, nodes)

def jacobian(nodes, xi):
```

```
    '''
    Calculate local Jacobian matrix

    Parameters
    ----------
    nodes: array
    node numbers

    xi: array
    local coordinates

    Output
    --------
    Jacobian array

    '''
    dN = derivs_shape_functions(xi)

    return np.dot(dN.T, nodes)

def jacobian_determinant(nodes, xi):
    '''
    Calculate the determinant local Jacobian matrix

    Parameters
    ----------
    nodes: array
    node numbers

    xi: array
    local coordinates

    Output
    --------
```

```
    Jacobian determinant

    '''
    return abs(np.linalg.det(jacobian(nodes, xi)))

def global_derivs_from_local(nodes, xi):
    '''
    Calculate the determinant local Jacobian matrix

    Parameters
    ----------
    nodes: array
    node numbers

    xi: array
    local coordinates

    Output
    --------
    Calculate global derivatives from local derivatives

    '''
    J = jacobian(nodes, xi)
    dN = derivs_shape_functions(xi)
    dN_global = np.zeros_like(dN)
    for i in range(4):
    dN_global[i,:] = np.linalg.solve(J, dN[i,:])
    return dN_global
```

LISTING A.2: Plate Theory FEA Code