Development of the QFEM Solver

The Development of Modal Analysis Code for Wind Turbine Blades in QBLADE

MATTHEW DALE LENNIE

Master of Science Thesis
EGI-2013-140MSC EKV988
KTH School of Industrial Engineering and Management
Energy Technology
Division of Heat and Power
SE100-44 Stockholm October 2013
The Development of Modal Analysis Code for Wind Turbine Blades in QBLADE

MATTHEW LENNIE

Master's Thesis undertaken at the Technical University of Berlin
Supervisor: Christian Navid Nayeri (TUB)
Examiner: Björn Laumert (KTH)
Master of Science Thesis EGI 2013:  
EGI-2013-140MSC EKV988  
Development of the QFEM Solver

Matthew Lennie

<table>
<thead>
<tr>
<th>Approved</th>
<th>Examiner</th>
<th>Supervisor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Björn Laumert</td>
<td>Nenad Glodic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commissioner</th>
<th>Contact person</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Master of Science Thesis  
KTH School of Industrial Engineering and Management  
Energy Technology EGI-2013-140MSC EKV988  
Division of Heat and Power Technology  
SE-100 44 STOCKHOLM
Abstract

The Wind Turbine industry continues to drive towards high market penetration and profitability. In order to keep Wind Turbines in field for as long as possible computational analysis tools are required. The open source tool QBlade[38] software was extended to now contain routines to analyse the structural properties of Wind Turbine blades. This was achieved using 2D integration methods and a Tapered Euler-Bernoulli beam element in order to find the mode shapes and 2D sectional properties. This was a key step towards integrating the National Renewable Energy Laboratories FAST package[32] which has the ability to analyse Aeroelastic Responses. The QFEM module performed well for the test cases including: hollow isotropic blade, rotating beam and tapered beam. Some improvements can be made to the torsion estimation of the 2D sections but this has no effect on the mode shapes required for the FAST simulations.
In the past two years I have had the fantastic opportunity to take part in the THRUST program. My unreserved thank you goes to the Thrust faculty who had the vision to make such a program possible. A particular thank you goes to Damian Vogt, Maria Majorca, Nenad Glodic and Karin Knutson who worked tireless to run the program. I also acknowledge and thank the people of Europe (represented by European Commission) who fund such an amazing opportunity.

On a personal level I have had the pleasure of making many new friends abroad and experiencing the strong camaraderie of my Thrust colleagues. Late nights and hard work are made fun when you share them with such amazing people. All while I have been supported by Svenja and also by my family and friends at home in Australia whom I dearly miss.
Contents

I Project Description 1

1 Introduction 2
  1.1 Project Scope ................................. 3

2 Literature Review 5
  2.1 What are wind turbines and how do they work? .... 5
  2.2 Wind Resources .................................. 9
  2.3 Wind Turbine Aeroelasticity ....................... 11
  2.4 The Simulation Chain of Wind Turbine Aeroelasticity . 12
  2.5 FAST .............................................. 21
  2.6 Summary .......................................... 24

II Program Design 25

3 Program Design 26
  3.1 Blade Geometry and Structural Properties ....... 27
  3.2 2D Section Structural Properties .................. 27
  3.3 Beam Elements .................................... 35
  3.4 Beam Elements Localisation ...................... 37
  3.5 Eigenvalue Problem ................................ 42
  3.6 Mode Shape Polynomial ........................... 42

III Testing 48

4 Test Cases 49
  4.1 Isotropic Cantilevered Beam Analysis .............. 49
  4.2 Isotropic Cantilevered Hollow Blade Analysis .... 54
  4.3 Non-Rotating Tapered Beam ....................... 56
  4.4 Rotating Beam .................................... 57
  4.5 Composite Blade Test Case ....................... 58

5 Conclusion 61
  5.1 Further Work ..................................... 61
Nomenclature

\[ [C] \quad \text{Damping Matrix} \]
\[ [EA] \quad \text{Longitudinal Stiffness} \]
\[ [ED_{xyR}] \quad \text{Moment of Centrifugal Stiffness} \]
\[ [EI_{xR}] \quad \text{Moment of Stiffness Inertia about the x Ref. Axis} \]
\[ [EI_{yR}] \quad \text{Moment of Stiffness Inertia about the y Ref. Axis} \]
\[ [ES_{xR}] \quad \text{Moment of Stiffness about the x Ref. Axis} \]
\[ [ES_{yR}] \quad \text{Moment of Stiffness about the y Ref. Axis} \]
\[ [K] \quad \text{Stiffness Matrix} \]
\[ [M] \quad \text{Mass Matrix} \]
\[ \alpha \quad \text{Angle between the reference axis and elastic axis} \]
\[ \omega_{\text{natural}}, \omega_n \quad \text{Natural/Eigen Frequency} \]
\[ \rho_f \quad \text{Fluid Density} \]
\[ \rho_m \quad \text{Material Density} \]
\[ \rho_{\text{linear}} \quad \text{Linear Density of the blade(i.e. kg/m)} \]
\[ \{\ddot{x}, \dot{x}, x\} \quad \text{Motion Vectors} \]
\[ A_{\text{disc}} \quad \text{Disc Area} \]
\[ c \quad \text{Chord Length} \]
\[ D \quad \text{Drag Force} \]
\[ EI_1, EI_2 \quad \text{Moments of Stiffness Inertia about the elastic axes} \]
\[ F(t) \quad \text{Forcing Terms} \]
\[ GJ \quad \text{Torsional Stiffness} \]
\[ H(\omega) \quad \text{Amplification Function} \]
\[ J' \quad \text{Polar Moment of Inertia about the Centroidal Axis} \]
\[ L \quad \text{Lift Force} \]
$M$ Twisting Moment

$P$ Power

$V_{\infty}$ Freestream Velocity

$x_R,y_R$ Foil Local Coordinates Compared to Reference Origin

$x_{\text{Centroid}},y_{\text{Centroid}}$ Section Centroid Coordinates

$x_{ei},y_{ei}$ Dimensions of the Integration Element

$X_e,Y_e$ Location of Elastic Axis Compared to the Reference

$x_{\text{Res}},y_{\text{Res}}$ Resolution of Integration in each direction. $x_{\text{Res}} = y_{\text{Res}}$ in the implementation.
Part I

Project Description
Chapter 1

Introduction

Wind Turbines work on the same principle as any gas turbine. A source of heat makes gases move and the turbine extracts energy from it. The big difference is that, in a normal gas turbine the gas is driven by fossil fuels. Fossil fuels are essentially concentrated sunlight. The Sun provides energy for plants and animals, which grow and eventually die. After many years buried beneath the earth the carbon matures into a more complex state. These fossil fuels are a fantastically dense source of energy and they have fuelled; modern medicine, clean running water and agriculture, all key factors for improving the quality of life around the world. However, these fuels have a downside, they release long trapped carbon into the atmosphere. This has been going on since the industrial revolution but David MacKay captured it best by saying: \textit{"This is a global geo-technical experiment that we have been well advised to stop"} [36][37].

We can not simply turn off 200 years of industrial revolution, so another energy source is required. The sun creates more than just fossil fuels, it also creates wind, again; something hot making gases move. A Wind Turbine is a turbo-machine built to extract energy from the wind. Unfortunately wind is a very low quality source because wind is relatively slow, variant and air isn’t very dense, however if we look globally there is a lot of it (figure 1.1). Every improvement to Wind Turbine technology enables more and more of this resource to be harvested.

Engineering analysis tools can be used in a number of ways to develop Wind Turbines. Structural Engineers can design better and cheaper structures that will last longer. Aerodynamicists can design wind turbines that will remain efficient all through their life. Students and innovators can experiment with ideas in order to help drive the next generation of technology. With computer simulation tools engineers can test slight optimisations through to new undeveloped concepts before having to fully commit to building and testing them. It is faster and cheaper which allows companies to drive innovation while maintaining profitability.

QBlade[38] is an open source simulation tool built at the Technical University of Berlin. It is built as a relatively light weight tool used to put quick answers into users hands. Currently the program contains functionality to build simulation Wind Turbines, generate their aerodynamic properties and then simulate their performance under a range of wind conditions. It is not built to answer highly detailed research or design questions, these require more robust and resource intensive methods. Essentially the program is a collection of methods and tools used to create the first stage early design of a Wind Turbine, in other words a good engineering first guess.

All Wind Turbines must be designed to operate safely under a range of conditions for
1.1. PROJECT SCOPE

Figure 1.1: Global Wind Resources. Reproduced from NASA’s Visible Earth Project [17]

a considerable lifetime. To achieve this one needs to fully understand the aerodynamic, structural and aeroelastic loads on the turbine. The National Renewable Energy Laboratory (NREL) - in Boulder, Colorado - have built a simulation tool called FAST[32]. This program is designed to take an input of: blade/tower geometry, blade aerodynamic properties and structural properties as well as a set of operating conditions. Thereafter, it is capable of simulating the dynamics of the turbine under the prescribed conditions. This gives users information required to check fatigue loads or even if it is possible for the blades to strike the tower. This code is freely available and will be integrated into QBlade[38] as the aeroelastic package. This project is concerned with taking the turbine designed in QBlade[38] and then translating it for input into FAST. This means finding and processing the structural properties of the blades in a way that is simple for the average user. This will be the key link to help users quickly generate an aeroelastic analysis, a key tool for understanding the life of Wind Turbines.

1.1 Project Scope

The aim of this project is enable QBlade to call FAST. The Graphical User Interface (GUI) to create the input files for FAST has already been created. This gives the user options to set up the wind profile, the turbine behaviour and the output. The information that is missing is the mode shapes and structural properties of the turbine blades.

An alternate compilation of FAST has been developed by the Larwood in his PhD
CHAPTER 1. INTRODUCTION

Thesis [35]. The CurveFAST[35] program was developed using the same principles as FAST. Larwood created the code in order to extend the capability of FAST to analyse swept blades as well as including twisting modes. The CurveFAST package would be a valuable addition to QBlade, however, FAST has already been certified by Germanischer Lloyd[4] for aeroelastic calculations. Therefore FAST has been chosen as the first package to be integrated into QBlade. Notwithstanding, care should be taken to ensure that the QBlade set up can interface with both packages, thus minimising future work.

NREL has developed codes - BModes and PreComp - to create the mode shapes of the blades. These codes will not be used for a number of reasons,

1. The interface assumes a user more advanced than the archetypical QBlade user.
2. The program only produces the modes as per required for FAST, meaning that the CurveFAST extension from Scott Larwood[35] would be difficult to integrate at a later stage.
3. The BModes/PreComp codes have a limited ability to model some mode shapes as highlighted by De Frias Lopez [12].

These reasons will become more clear during the review of literature. However, the result is that; a new modal analysis program was written and achieves the following:

1. Use Basic Formulae to find the 2D sectional Properties of a Turbine Blade which can be hollow and with a spar.
2. Create a Finite Element Model of the Wind Turbine Blade which will be solved as an Eigenvalue Problem.
3. The resulting mode shapes will be sorted into categories, with minimum user intervention.
4. The mode shapes will be processed into polynomial form for FAST Input
5. The program will have the ability to return the global coordinates of the un-deformed blade and individual mode shapes.

All of this will be achieved by maintaining the minimum complexity required for the user to produce sensible results. Furthermore, the code will be developed as independent from QBlade as possible such that the code is portable to other projects in the future. The code will also be designed with the intent that extra features and static solution methods can be added in the future.
Chapter 2

Literature Review

2.1 What are wind turbines and how do they work?

The most common form of modern wind turbine is a three bladed Horizontal Axis Wind Turbine (HAWT). The machine consists of four key component groups; the blades, the rotor, the nacelle and the tower. Figure 2.4 shows an example of an offshore wind turbine. To understand how a wind turbine works one must start with the blades and follow the process through the ground.

Wind Turbine Blades

The blades are responsible for turning incoming wind into torque. When wind blows one considers that on average it enters the Wind Turbine front on, in other words in line with the rotation axis. As the Wind Turbine spins the blade will have a velocity which is then subtracted from the Wind Speed to give the relative velocity between the wind and the blade. This flow moves across the blade creating both lift, drag and moment forces. We can take these forces and consider them in terms that are useful for the Wind Turbine. The resulting forces are tangential force which rotates the blade about its axis providing the torque and the perpendicular axial/thrust forces push in the same direction as the wind. This force does not produce energy in a HAWT.

When a Wind Turbine Blade spins simple geometry defines that tip will have a greater velocity than the hub. To consider a full blade one must realise that each part of the blade must be designed differently to account for the change in relative velocity. Figure 2.2 shows a comparison between the velocity components at the tip and hub. This is why on modern wind turbines the hub and tip region the geometry will vary dramatically. This is also why the tangential and thrust/axial forces will vary across the blades as shown in figure 2.3. This thesis extensively treats the structural dynamics caused by these forces and others.
CHAPTER 2. LITERATURE REVIEW

Absolute and Relative Flow

Blade Rotation
\[ \mathbf{U} = \omega_{\text{rotor}} \mathbf{r} \]

Relative Inflow
\[ C_1 = V_\infty \]

Absolute Axial Inflow

Forces Generated by the Aerofoil

Tangential Force

Thrust Force

Figure 2.1: How wind moves through a Wind Turbine

Flow at the Hub Region

Blade Rotation
\[ \mathbf{U}_{\text{hub}} = \omega_{\text{rotor}} \mathbf{r}_{\text{hub}} \]

Relative Inflow
\[ C_1 = V_\infty \]

Absolute Axial Inflow

Flow at the Tip Region

Blade Rotation
\[ \mathbf{U}_{\text{tip}} = \omega_{\text{rotor}} \mathbf{r}_{\text{tip}} \]

Relative Inflow
\[ C_1 = V_\infty \]

Absolute Axial Inflow

Figure 2.2: How relative flow changes at different parts of the blade
2.1. WHAT ARE WIND TURBINES AND HOW DO THEY WORK?

Pitch System and Rotor

Wind Turbine Blades are not a fixed part. As the wind speed changes so do the aerodynamic requirements. As a result many wind turbines will incorporate a pitch system so that the inflow angle or 'Angle of Attack' can be adjusted as required. Pitch systems also form a vital part of the fail safe mechanisms allowing the control system to 'feather' the blades into the wind during storms, or to help start the turbine after a shut down. The blade is held by a bearing in the pitch system. The pitch is actuated by a motor which positions the blades via a gear set. The torque generated from the blades is transmitted through to the rotor causing it to spin.

Nacelle

When the rotor spins it links the blades to the second component group, the Nacelle (see figure 2.5). A Nacelle contains the control and power generation equipment for the wind turbine. The rotor is connected to the Nacelle via the main shaft. In most modern turbines the main shaft is then connected to an up speed gearbox. The high speed output of the gearbox is connected to the electricity generator. Brakes to shut down the wind turbine are attached to the high speed shaft to enable shut downs for maintenance and in emergencies. All of these components are connected via the main structural frame of the Nacelle.

Tower

The last component group is perhaps the most simple, the tower. The tower simply acts as an anchor to transfer the reaction forces from the turbine through to the ground. Although it is a simple part, the tower can undergo immense forces especially during exceptional
CHAPTER 2. LITERATURE REVIEW

Figure 2.4: Wind Turbine Components and Pitch Actuation. Reproduced with Permission [45]

Figure 2.5: HAWT Nacelle Components. Reproduced with Permission [45]
2.2. WIND RESOURCES

events like turbine shut downs. The tower is the final group that make up a wind turbine. We can now understand the power generation chain. The turbine blades deviate incoming wind to create a torque. The torque is carried through the rotor to the Nacelle, through a gearbox and then onto the electricity generator. The resultant forces not converted into electricity are carried through the tower to the ground.

2.2 Wind Resources

In the introduction the vast wind resources of the world were used to argue the case for wind turbines. However, to use wind it is essential to first understand that it is a variable resource. Wind speed varies in space and in time which heavily effects both the economics and engineering of a wind turbine. In turbomachinery there is a cubic relationship between the flow speed and the power extracted by a device. Therefore, a site with high average wind speed would seem to be a good choice. Engineering is rarely that simple and this is no exception as the structural design of the turbine could become very complex if the wind at that site was susceptible to high turbulence. In order to design a wind turbine, one must first understand how the wind behaves.

Spatial variations of wind can be considered in three levels; Global, Regional, and Local. Global variations are responsible for large scale global circulation patterns. Certain features in these systems have been long since recognised by sailors pre-dating modern measurements. California provides an example of regional variations. In this region the cold ocean currents meet with the deserts in the region creating a strong differential temperature, a key driving force of wind. The ensuing wind is then funnelled through the mountainous terrain creating areas with a predictable high speed resource. Local variations of the wind are created as the wind passes local topology such as hills, trees, houses and even other wind turbines.[8]

Temporal variations also occur over a range of frequencies. Long term changes to wind patterns over years and decades are not particularly well defined but will heavily effect the economics of a project. Seasonal variations are well understood and well predicted ahead of time. Synoptic variations occur over the 1-4 day scale and are caused by the movement of high and low pressure systems. Diurnal patterns are caused by daily heat variations from sunlight and are usually well predicted. The predictability of these resources is important because wind turbines are attached to the electricity grid. These variations all help understand the full operational regime of a turbine.[8]

It is not likely that seasonal, and diurnal variations of mean wind speed will cause aeromechanical excitation of the wind turbine, as the frequency is far to low to be considered vibratory, instead shorter time scale variations are the enemy. Turbulence is caused by a combination of topology and thermal effects; and is described as any fluctuation that has a period less than 10 minutes. Local topology and asperities will cause disturbances in the flow field which create non-uniform flow. Temperature gradients between the hot earth surface and cool air can drive natural convection, which is experienced by the turbine as turbulent eddies. Although it is possible to mathematically describe turbulence, it is more practical to consider the system to be chaotic and then define turbulence by its statistical properties. Turbulence properties form a key input into determining the fatigue life of the turbine. [8]

Like all real fluid flows, wind over the earth’s surface will generate a boundary layer, with slow moving fluid at the surface and high velocities at altitude. The boundary layer of wind over the earth’s surface is controlled by the strength of the wind, the surface asperities, thermal effects as well as the Coriolis effect of the earth’s rotation. The presence of a boundary layer is responsible for asymmetric flow with higher flow speeds at higher heights
as visualised in figure 2.6. In some conditions this can even manifest as strong gusts at low heights. This kind of flow can be seen as analogous to inlet distortion in gas turbines or jet engines; therefore one must consider the boundary layer as a structural excitation source. [8]

![Figure 2.6: Wind Boundary Layer and Turbulence. Reproduced with Permission [45]](image)

The most practical method to describe wind conditions is to use statistical methods. These methods enable wind turbines to be simulated in a variety of steady and unsteady conditions. The longer period variations of mean wind speed can be described using a Weibull distribution [8]. Veers [56] gives examples of reconstruction of wind fields, where power spectrum models (Von Karman and Kaiman) give the variance of turbulence in time and coherence functions give the variance in space. While only briefly described here, the wind models will form a crucial into the aeroelastic model. In a separate effort the QBlade
2.3 Wind Turbine Aeroelasticity

Aeroelasticity is the field of engineering where a dynamic system is influenced by a combination of aerodynamic, inertial and elastic forces. A good example of aeroelasticity is on a regular plane wing. The wing is subjected to a flow which results in lift generated. The lift deflects the wing and elastic forces will try to restore with an opposing force. Unless the forces are always balanced a net force is created and the wing will accelerate. From D’Alemberts’ principle one can establish that inertial forces will work to oppose this acceleration. As the wing moves the lift it generates changes. Aeroelasticity explores when all three of these forces combine into a continuous feedback loop. The interaction of forces is visualised in the collar triangle (figure 2.7).

![Collar's Triangle](image)

Figure 2.7: Collar’s Triangle [46]

There are two interpretations of aeroelastic forces where one can consider them to be an external force or be linearised to become a damping term. The damping term gives a good physical interpretation of aeroelastic phenomena. When the aero-elastic damping is positive the aeroelastic effects act to dampen the blades vibrations; work is done by the blade on the fluid. When the aero-elastic damping is negative the aeroelastic effects will draw energy from the fluid onto the blade; the result is a rapid growth in the vibrational amplitude. This is the birth of flutter which can result in the sudden and spectacular death of the structure.

Classic (or Coalescence) flutter occurs when the negative damping of the aeroelastic effects overcome the natural damping in the system. For this to occur a Torsion and Flapwise modes have to be coupled, meaning both modes have to be excited. When this occurs there will be a phase difference between the flap motion and the torsion motion. In figure 2.8, Ashraf et al.[6] show the coherence of forces and motion with differing phase
angle. The phase difference will determine whether the structure dissipates energy (in the case of a -90 degree phase difference which isn’t shown) or gains energy from the flow. In this paper the authors are using flutter as a form of power generation, in short they seek the opposite goal of a Wind Turbine designer.

Currently Wind Turbines are relatively stiff in torsion and therefore operate at roughly 1/6 of the flutter speed [48]. As industry drives towards larger and larger rotors this flutter margin is being gradually decreased and so Wind Turbine designers are becoming more aware of flutter as a design concern. However even without the imminent threat of flutter, it is important to take into account the aeroelastic effects in a Wind Turbine.

2.4 The Simulation Chain of Wind Turbine Aeroelasticity

Aeroelasticity consists of Aerodynamic, Inertial and Structural forces. When making simulation of aeroelastic behaviour all three forces must be considered. These forces must be considered somewhat separately. With in the simulation structure this could mean linking separate programs, or including aerodynamic terms into a FEM solver. An example of the latter is demonstrated by Capponi[10]. Such a solver was intended to perform stability analysis whereas most of the simulation techniques that are being considered require a more "brute force" time marching approach.

Figure 2.9 shows a generic form of a simulation chain used to certify Wind Turbines for operations. In the Aeroelastic Simulation Loop one can see that two separate solvers will solve system in steps. This link can be made in a number of ways. A 'Fully Resolved' system will make multiple loops for each time step. In each loop the structural dynamics are simulated and then fed to the aerodynamics solver. The aerodynamics solver then take the structural dynamics as input to estimate the aerodynamic forces onto the structure. These forces are fed back to the structural dynamics solver and the process is repeated until a converged solution is found. A partially resolved system may avoid looping by simply feeding the result in one direction and use the previous time step as the input. Such
2.4. THE SIMULATION CHAIN OF WIND TURBINE AEROELASTICITY

a system maybe faster per time step but it is likely that a smaller time step is needed to maintain a stable solution.

This discussion highlights an important feature of aeroelastic simulations, the nested iterations. Let us take an example, it is not based on accurate numbers but is used just as a rough demonstration. Suppose that a high fidelity CFD package is used to solve the aerodynamics of the Wind Turbine. To converge each time may take in the order of 4-5 implicit steps, each taking 1 second each. The fully resolved package may have to do this 1-4 times each time step. If there is a large number of test cases each with 1000 timesteps, the problem becomes self evident that simulation times can blow out very quickly.

Because Aeroelastic simulation times can easily become large a number of approaches can be used. Let us take an example of a research project where one configuration will be run to isolate a certain design feature or phenomenon. In this hypothetical project, highly accurate results are needed and so money and time is invested to develop a high fidelity solution. In this solution the structural and aerodynamics solvers are both high fidelity, the computational and set up time is large but it is only run once to get a specific answer. The opposing example would be during a certification test. These cases will simulate times ranging from seconds up to 12 sets of 10 minute simulations [8]. A large number of test cases have to be run, and less detail is required than the previous example. In this example, a more simplistic dynamics and fluids solver would be coupled together resulting in a more modest run time per test case. In the following sections the preprocessor steps and aeroelastic loop steps will be explored. A range of methods will be explored through out the analysis chain in order to select the methods most appropriate for QBlade.
CHAPTER 2. LITERATURE REVIEW

Figure 2.9: Generic Aeroelastic Simulation Chain used for certifying Wind Turbines
Load Cases

Wind Turbine failures can be lumped into two categories, extreme and fatigue failures. An extreme failure is caused by off design occurrences, for example very strong winds in a storm. One can design for known extreme load cases but it is impossible to design for all possible circumstances. Fatigue failure is caused by a cyclic loading and unloading of the structure. Loading and geometry can localise stress and eventually the structure initiates and propagates a fracture through the materials causing failure.

To establish the loads on a Wind Turbine there must be a set of boundary conditions defining the inputs. Though a number of standards exist, the following examples are drawn from the IEC 61400-1 standard [26] as well as the Germanischer Lloyd Design Guidelines [62]. These standards outline a set of wind conditions and turbine operating modes which should be simulated to ensure the turbine will survive all that it is subjected to in its considerable life span.

The methodology is to account for all of the; Aerodynamic, Gravitational, Inertial and Operational loads. Keeping this in mind the IEC 61400-1 standard generates 22 load cases which can be broken down into 17 ultimate/extreme stress and 5 fatigue cases. These cases are generated by a combination of wind and operational states, for example, extreme wind speed during normal operation or normal wind speed with a parked turbine.[8]

Ultimate loads will arise from less common combinations of conditions such as extreme wind speeds or extreme wind shear. In such cases, tower strikes are a primary concern. However, the life of a turbine is more commonly determined by the fatigue loading. For example, a 2MW rotor will undergo $10^8$ gravity load reversals during its 20 year life [8]. Other out of plane fatigue loads include, wind shear, yaw error, shaft tilt, tower shadow and turbulence. Good response estimates have to be made for all of these load sources [8]. It would be ideal if QBlade could have the ability to address such cases.

Aerodynamics Modelling Methods

It should be fairly clear by this point that the aerodynamics of a Wind Turbine blade will be relatively complex. In addition, 'full' size wind tunnel testing is very difficult to achieve(fund) and has only been done in some special cases like the Ames Test by NREL [51]. It follows that; Wind turbines must be designed with empirical relations, numerical models and Computational Fluid Dynamics(CFD). As already discussed the wind inflow speed of a Wind Turbine is highly variable and the device is very large making even numerical simulations complex.

Blade Element Momentum Method

A common analysis technique is to split the swept area into annular sections in what is called Blade Element Momentum(BEM) theory. In this theory the flow is analysed in 2D sections which are assumed to be isolated from the other sections and from other blades. The method works by equating the momentum of a rotating annular stream tube with the lift and drag coefficients.

The BEM method relies upon a priori knowledge of the 2D aerofoil sections coefficients. In aerodynamics, polar diagrams - values for various angles of attack - are produced for the lift and drag coefficients of a 2D section. For passenger comfort (and many other) reasons it is unusual that planes will encounter very high angles of attack. As a result most aerofoil polars are only valid for small angles of attack(pre-stall). However, Wind Turbines are often stall regulated and have highly variable in-flow conditions. This means that full 360 degree angle of attack polar diagrams are required. Usually experimental tests and programs
such as XFOIL\[15\] provide data only valid for small angles of attack, so in Wind Turbine aerodynamics, extrapolation techniques are required as exemplified by Montgomery [40] and Viterna-Corriigan [57]. These polar diagrams can be used in computer simulations by calculating the angle of attack and then reading out the consequent coefficients. This is obviously much faster than resorting to CFD for every simulation step but fails to take into account dynamic effects.

There are two notable downfalls of the BEM method; failing to account for 3D effects and unsteady flow. 3D effects can somewhat be compensated for by using Prandlts’ Tip Loss formulas. Unsteady effects are especially important for an aeroelastic simulation where both the wind source and the blades’ vibrations will cause angle of attack variations. Oscillating wing research [6] provides excellent examples of unsteady effects like vortex shedding. Such phenomena can cause temporary changes in performance on the aerofoil, for example a temporary increase in lift. Without exploring the topic too deeply, it can be noted that BEM can be extended to unsteady simulations with empirical adjustments[24].

Applications of CFD on Wind Turbines

In their paper Guo et al.[22] use an advanced CFD model in order to conduct an aeroelastic simulation of a Wind Turbine. The authors point out that while BEM and vortex wake methods are commonly used they heavily rely on accurate aerofoil polars and cannot account for 3D effects. Furthermore, the authors also comment on the limitations of most common CFD approach which is the Reynolds-averaged N-S (RANS) approach. In their paper the authors employ a Detached eddy simulation(DES) in an attempt to account for the 'massively separated' flow around the Wind Turbine. In such a simulation a Large Scale Eddy(LES) model is used in the separated region and RANS model is used in the attached boundary layer region. Such a method is driving towards more accurate results in the CFD part of the aeroelasticity simulation chain.

In a recent paper, Eisele et al. made an investigation comparing CFD results with various wind tunnel tests [18]. The results showed that inflow turbulence and surface roughness would have a distinct effect on the resulting polars. Gilling et al. showed that LES simulations could be effective at accounting for such effects[20]. Gilling et al. also made comments on how DES simulations can fail to account for in flow turbulence due to meshing configurations [20]. Eisele et al.[18] highlight these results in their own report but point out that LES simulations are rather computationally intensive. Instead the authors compare RANS simulations using various turbulence models. The results showed that with judicious and realistic choice of turbulence models, RANS simulations can be a good tool[18]. Furthermore, it is more lightweight than the LES model.

In this section we looked at some of the methods that can be used to model Wind Turbine aerodynamics. BEM methods are a computationally cheap and effective tool that is heavily reliant on 2D polar data. CFD methods (if applied properly) will be more accurate than the BEM method but will also be more computationally expensive too. LES models appear to be able to accurately capture effects of inflow turbulence. However RANS and DES solutions can be used judiciously as a more computationally efficient method than LES. AeroDyn[41] - which is integrated into FAST - uses the BEM method to obtain the aerodynamic forces. a comparison was made between this method and the more intense CFD codes to make a distinction to readers between high and low fidelity analysis techniques. Other methods certainly do exist but aren’t important for the discussion here. The important conclusion is an understanding that the aerodynamics part of the FAST simulation will have its limitations.
Blade Structural Modelling

Analysing responses to dynamic forces can be difficult. For many turbomachinery analysis applications, much of the complexity can be skipped by linearising the system. In the most simple form a system can be reduced to a single degree of freedom 'spring-mass-damper' model such as shown in Figure 2.10. Such a simple representation is useful to understand concepts such as resonance and damping but will rarely represent a physical system well. Multi-degree of freedom systems like the one shown in Figure 2.10 can be expanded to explore problems such as rotor mistuning in gas or aero turbines but again have limited scope. The problem lies in that, these two systems can be solved analytically but higher order systems do not have a closed form solution.

![Lumped Spring Mass Damper Models](image_url)

To analyse a Wind Turbine Blade one can employ more general Finite Element Solutions. 1D elements such as rods can only transmit force along its own axis. Euler-Bernoulli and Timoshenko Beam elements are capable of transmitting axial, bending and torsional forces. For plain stress or plain strain approximations, triangle and quad elements can be used to form a 2D solution. In a full 3D solution the geometry is discretized into brick or tetrahedral shapes. Simple elements such as beam elements and rods only have 2 nodes, a tetrahedral element will have 4 nodes, each with 6 degrees of freedom. Furthermore, 3D elements are usually applied more densely than a beam or rod element resulting in very large sets of equations. For most beam element solutions a simple matrix inverse can be used to solve the system with minimal computational effort. A normal more robust finite element solver will apply a guess and check type solver iteratively to find the configuration with the lowest residual energy (technically described as the minimization of a functional). With in these examples exists many other examples and sub-categories, the point is that the problem can be approached with different techniques, some are resource intensive some are not.

More advanced solution methods apply these types of elements to the geometry and simulate the dynamics in a large deformation framework i.e. like in Metaphor [9]. Large deformation software has the distinct advantage that no assumptions of small deformations are retained unlike the conventional FEM small deformation methods. Such methods implement time marching schemes to solve the problem in sufficiently small time steps to solve the dynamics. The time integration schemes can be achieved explicitly or implicitly with
numerical damping being tuned to retain a stable solution [47]. Setting up such a simulation is a significant task and requires advanced knowledge of the boundary conditions such as material behaviour laws. Such a solution will produce (with the right user and information) highly detailed and thorough results but is far too heavy to be part of a quick design and iterate cycle, especially if one considers a high number of load cases.

The problem arises that small deformation assumptions can have effects on the solutions. It is possible in some cases to add secondary terms to account for these effects. This is demonstrated by Larwood[35] and Munteanu et al[19] where spin stiffness terms are added to the beam element formulation. Such secondary terms are still linear and are still only valid for a certain extended domain.

Modal Analysis Methods

For most analyses the dynamics are modelled using modal analysis. A modal transformation separates the temporal and spatial terms. In essence the method is analogous to a Fourier transform, where one can break down the structural response into different eigenfrequencies (natural frequencies). The result is that one can extract distinct modes of vibration. Three common classes of blade modes are shown in figure 2.11, these are; First Flap (First Bending), First Edgewise and First Torsion. The total response of the structure would be the linear addition of these modes.

![Example Mode Shapes of a turbomachinery blade](image)

Figure 2.11: Example Mode Shapes of a turbomachinery blade

Returning to the spring damper model, one can inspect how a modal transformation is achieved. From D’Alemberts principle one arrives at a set of equations of motion in the following form:

\[
[M]{\ddot{\mathbf{x}}} + [C]{\dot{\mathbf{x}}} + [K]{\mathbf{x}} = F(t) \tag{2.1}
\]

Where,

\[
[M] = \text{Mass Matrix} \\
[C] = \text{Damping Matrix} \\
[K] = \text{Elastic Matrix} \\
F(t) = \text{External Forcing} \\
{\ddot{\mathbf{x}}}, {\dot{\mathbf{x}}}, {\mathbf{x}} = \text{Motion Vectors}
\]

By assuming that the motion is undamped and sinusoidal one can reduce the system to;

\[
-\omega^2[M]{\mathbf{x}} + [K]{\mathbf{x}} = F(t) \tag{2.2}
\]
Then further simplification and removal of the forcing term yields the homogeneous equation:

\[ \omega^2 \{ x \} = [M]^{-1} [K] \{ x \} \] (2.3)

Thus can be solved as an eigenvalue problem yielding the mode shapes and natural frequencies for the system.

Alternatively, one can solve the equations using the Duncan Method[16]. This is used when damping terms can not be neglected and/or there are gyroscopic terms present. As this paper is based on the analysis of a rotating blade some treatment should be given to this method. The basis equations (without gyroscopic terms here) are;

\[ [M] \{ \ddot{x} \} - [M] \{ \dot{x} \} = 0 \] (2.4)

\[ [M] \{ \ddot{x} \} + [C] \{ \dot{x} \} + [K] \{ x \} = F(t) \] (2.5)

Which reduces into matrix form as;

\[
\begin{pmatrix}
A & B \\
C & D
\end{pmatrix}
\begin{pmatrix}
\ddot{x} \\
\dot{x}
\end{pmatrix}
+
\begin{pmatrix}
\ddot{x} \\
\dot{x}
\end{pmatrix} = \begin{pmatrix} 0 \\ F(t) \end{pmatrix}
\] (2.6)

Which can again can be solved as an eigenvalue problem;

\[ |A + B\lambda| = 0 \] (2.7)

Where the resulting modal vector will yield paired complex conjugate mode shapes with associated eigenvalues. Gyroscopic terms can be included into this method as well but the solution method does not change. Therefore there are two choices for performing the modal transformations. The derivations listed here serve only as a short hand reminder to readers that are already familiar with these solution methods. The complete derivation of Modal Transformations can be found in the classic book from Meirovitch [39].

Having now built a modal transform it is possible to see how the system will react to loads. A single degree of freedom undamped oscillator(analogous to figure 2.10 without damping) has a natural frequency defined by;

\[ \omega_{\text{natural}} = \sqrt{\frac{k}{m}} \] (2.8)

In an undamped system the natural frequency determines when the response of a system tends to infinity. The source of this singularity is the amplification between the forcing and the response which is described by[39];

\[ H(\omega) = \frac{\frac{\omega}{\omega_n}}{\sqrt{1 - \left(\frac{\omega}{\omega_n}\right)^2 + \left(\frac{2\xi}{\omega_n}\right)^2}} \] (2.9)

With zero damping (\( \xi = 0 \)) the equation yields;

\[ H(\omega) = \frac{\frac{\omega}{\omega_n}}{\sqrt{1 - \left(\frac{\omega}{\omega_n}\right)^2}} \] (2.10)

From this formula, it is elemental to see that having \( \frac{\omega}{\omega_n} \approx 1 \) would result in large amplifications. In a real system damping occurs and magnitudes do not reach infinity. In most
physical systems one can expect damping to be under 10% so the amplitude of vibration at resonance is still a major concern. We could ensure low amplitudes in the oscillator by keeping excitation frequencies away from the natural frequency or by modifying the stiffness or mass of the blade. Such changes are usually achieved by modifying the geometry in the case of a turbomachinery blade.

Modal analysis makes several key assumptions at different stages to enforce the linearity of the system. The key result of this is that each of the modes are treated to be an orthogonal, essentially; an independent single degree of freedom oscillator. This means the external forcing terms can be split into frequencies and then projected onto the modes giving: the modal forces, the amplifications and each modes response.

A forced response analysis ensures that the input forces on the dynamical system will not create large amplitude responses. In the case of a Wind Turbine this means considering; gravitational terms, flow interferences such as the tower or local topology, as well as the changes in wind as described in section 2.2. Further to the wind speed one must consider; changes in wind direction as well as pitching and yawing of the turbine. Aeroelastic damping will influence the response to all of these stimuli and should be accounted for. A good test and verification method would analyse all of these possible cases with all parts of the Collar Triangle with the minimum computational power.

Applications of Small and Large Deformation analyses

Earlier in this thesis, the work of Guo et al. [22] was discussed in the context of comparing aerodynamics analysis methods. However Guo et al were considering an aeroelastic simulation not aerodynamics in isolation. In order to simulate the blade dynamics the modal superposition method was used. The first 14 modes were considered, therefore disregarding higher order modes as well as torsion modes. The mode shapes were found by creating a finite element mode and then analysing the results at different rotational speeds. The aerodynamic forces were then transformed and then applied to each of the mode shapes. The mode responses were then added together to form the beams full response. This process was repeated at each time step to give a time domain solution.

In his PhD thesis Ahlstrom [1] built a FEM solution for a Wind Turbine. Importantly, Ahlstrom considered that as Wind Turbine blades become more flexible; small deformation assumptions may no longer be valid. As such a simulation was built in the commercial software MSC Marc[43] a software similar to the earlier mentioned Metafor package[9]. Ahlstrom notes that instead of shell elements, beam elements are chosen for the simulation due to computational cost. Both Timoshenko and Euler-Bernoulli beams are available but little difference is found between the two in a comparison of results. Furthermore, he assumed that the aerofoils would be structurally shaped as rectangles thus simplifying the calculation of the torsional stiffness. Fortunately Wind Turbines vibrate with relatively low frequencies so with out any significant loss of accuracy the high frequency content could be damped out numerically. This increases the numerical stability of the simulation but even then it was necessary to choose a time step of 6ms(even 2ms in some configurations). The structural model then connected to a BEM model for the aerodynamic forces and time marching solutions were produced.

With this set up Allistom[1] was able to produce a number of interesting test cases. One such case was the trajectory of a blade that had separated from the Hub. Another was the dynamics of the turbine during a dynamic shut down. Such results are extremely useful for research applications and would help industry improve designs. Unfortunately, on a DELL Xeon 2.4 GHz with 1 GB ram 600 second of simulation took 8 hrs to solve. One could also draw on personal experience to add the phase 'when it actually converges'. Even
though the computer used was by present standards very slow, it still demonstrates that for multiple test cases the run time would be intensive.

In this section a number of structural mechanics approaches were discussed. The linear modal analysis method provides a simplified method of analysis. Being linear this method is suitable for small deformations only but can be extended somewhat with secondary terms. An alternate case was discussed where a large deformation solution was obtained. This solution was able to provide very interesting results but required intensive set up and simulation time. In FAST the blades are reduced to 3 degrees of freedom, 2 Flap and 1 Edgewise (note there is no Torsion) and then simulated using modal superposition [11]. The modal analysis method is therefore the most relevant for this thesis but care should be taken to understand its limitations.

2.5 FAST

Within the wind turbine industry several codes have emerged for aeroelastic analysis of wind turbines, to list a few examples; GL Garrad Hassan has the commercial code called Bladed[25], RISO in Denmark produce the HAWC2 software[49] and The National Renewable Energy Laboratory in Colorado produce FAST [32]. In this project FAST was chosen to integrate into QBlade due to the availability of source code and the analysis capabilities. To borrow the exact description from the creators 'The FAST_ AD design code is a medium-complexity code used to (1) model a wind turbine structurally given the turbine layout and aerodynamic and mechanical properties of its members and (2) simulate the wind turbine’s aerodynamic and structural response by imposing complex virtual wind-inflow conditions.' [30, p. 59]. This combination of capabilities renders FAST highly suitable to be included as an aero-elastic simulation tool. Such code is capable of creating the wide range of test conditions required to certify turbines for use (though certification of any kind is not included in this project). Colloquial industry evidence suggests that the full run of tests takes in the order of 1-3 days.

Program Structure

The FAST program can operate in a few different modes (see figure 2.5). This thesis is focussed on the ability to produce time series data of the various forces in the blades. The program can also act as a preprocessor to ADAMS (a multi body physics program, from MSC Software)[42], such features will be ignored for now.

One can see in figure 2.5 that the FAST solver is linked to the AeroDyn code in a circular manner as was described in the more generic figure 2.9. The FAST routines set up the dynamics of the wind turbine using Kane’s equations. The position and velocity of each of the 2D blade sections are then passed to the AeroDyn routines. The AeroDyn routines reads from the 2D section lift (and drag) polar diagrams to implement BEM theory. This in turn generates the blade forces at each section. This is fed back to the structural routine to balance the forces. This process is iterated, converged and then time marched.

The structural code is built on the assumption that only 3 modes of vibration are important; 1st Flap, 2nd Flap and 1st Edgewise. Notable is the absence of any torsion modes, a predicate to coalescence flutter. As already discussed this is not a monumental problem due to the large flutter margin. Nonetheless, in his PhD thesis Larwood reformulated FAST to include torsion and more advanced geometry consideration [35]. At this point CurveFAST -the Larwood reformulation - will not be integrated into QBlade, however parts of his formulation will play a pivotal role in later sections.
Preprocessing of selected inputs

Figure 2.5 highlights that several inputs are necessary to run a FAST simulation these are shown in figure 2.9 as the preprocessor steps. These can be split into four categories, aerodynamic data, structural data, machine data and load case data. The aerodynamic data will be generated by QBlade in two steps, first QBlade will call the integrated Xfoil[15] routines and the simulation will generate the 2D polars. Xfoil cannot converge beyond stall and so in the second step the polars are extended to the full \(360^\circ\) internally by QBlade, with the algorithms described by Montgomerie [40] or and Viterna-Corrigan [57]. The aerodynamic data is already developed within QBlade and just need formatting for FAST.

Load Case and Machine Data will constitute user input within QBlade. A random wind profile can be generated by QBlade - from power spectrum and coherence statistics - and then be set as the FAST test case. QBlade has the ability to generate a 2D wind profile which varies with time. The machine data will be completely determined by the user. Therefore these inputs can be considered to be existing data.

At this stage in the development QBlade has not integrated any kind of structural analysis. The user is able to generate the blades but conduct purely aerodynamic analysis. For this reason it is important to closely inspect the structural inputs to FAST.

FAST uses simplified variables to describe the structural properties of the wind turbines. The blades for example are described with input parameters as listed here:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BldFIDmp</td>
<td>Modal Structural Damping (%)</td>
</tr>
<tr>
<td>StrcTwst</td>
<td>Angle between the Elastic axis of each section compared to the tip.</td>
</tr>
<tr>
<td>BMassDen</td>
<td>Density of the Blade per Length at that particular span.</td>
</tr>
<tr>
<td>FlapStff</td>
<td>Stiffness in the Flapwise Direction</td>
</tr>
<tr>
<td>EdgStff</td>
<td>Stiffness in the Edgewise Direction</td>
</tr>
<tr>
<td>GJStiff</td>
<td>Polar Stiffness</td>
</tr>
<tr>
<td>EAStff</td>
<td>Axial Stiffness</td>
</tr>
</tbody>
</table>
The mode shapes are input into FAST as a polynomial in the form:

$$\phi = C_0 + C_1 x + C_2 x^2 + C_3 x^3 + C_4 x^4 + C_5 x^5 + C_6 x^6$$  \(2.11\)

Where the coefficients \(C_n\) have to be determined by fitting the polynomial to the mode shape. This simplification raises some issues as it assumes the direction of the vibration in each mode to be aligned with the structural twist of the blade\[32\]. This is an simplification made with the assumption that only light changes in twist and chord will occur between span stations. There are other values in the input file but these are only required for running the ADAMS preprocessor option, not a primary objective. The variables above represent the values that can not be created with existing QBlade functionality.

**Calculating Mode Shapes for FAST input**

In the previous section a brief investigation showed that a range of structural properties are required in order to run the FAST simulation. For this section the emphasis will be placed on methods used to find the 2D structural properties and mode shapes of blades keeping in mind they are; rotating, made of composites and complex geometry. We can split these methods into three categories; experimental, beam element modal analysis codes and full 3D element codes (ie the codes that are far too difficult to develop individually).

Bennett \[7\] undertook finite element simulations with Abaqus/CAE \[53\] to find the natural frequencies and mode shapes of blades. In such an analysis the author employed shell elements with roughly 70000 Elements. Each element has 8 nodes, each with 6 degrees of freedom each. One can see already that this is not a lightweight analysis. However, a baseline study using this method compares excellently with the NREL 5MW experimental data. Such an approach is heavy but accurate.

Lopez \[12\] took a slightly different approach to finding the mode shapes of turbine blades. Seeing that current beam elements failed to account for cross stiffness terms found in composites, the author created a new 3D finite beam element. The author also highlights research from Kollar et al \[33\] that thin-walled composite beam structures can be prone to buckling in pure bending. Results showed that element performed well against an Abaqus Shell Element simulation. Implementing such an element requires a good knowledge of the composite materials which is considered to be too complex for the first available structural model in QBlade.

The NREL code PreComp is designed to find the structural properties of a composite wind turbine blade. While this package would be ideal to use - and may be at a later stage - two problems were identified. Producing the input files requires a strong knowledge of composites. The most basic structural model in QBlade should not be so complicated. More advanced models -like from Lopez \[12\] - can be integrated later. The second problem is that comparisons - conducted by Lopez \[12\]- show that PreComp is not particularly accurate at predicting Torsion Modes. It seems that PreComp is not the desired software for now.

In the thesis "Dynamic Analysis Tool Development for Advanced Geometry Wind Turbine Blades"\[35\], Larwood derives a module of code called CurveFEM. CurveFEM uses Euler-Bernoulli Beam elements (just as Ahlstrom \[1\]) with tapered properties to find the mode shapes. The rotational and non-linear effects are accounted for by including Spin Stiffness, Axial Force, Axial Reduction and Gyroscopic terms. In the final implementation the Gyroscopic terms were ignored so that the eigenvalue could be solved without using the Duncan Method. The algorithm only implements isotropic material properties (i.e. not cross stiffness terms) making it a good candidate for a first implementation into QBlade.
2.6 Summary

In this chapter we have explored the fundamentals of aeroelasticity and how it is modelled in Wind Turbines. An emphasis was placed on comparing high and low fidelity methods and where they are used appropriately. The result of the discussion that the selection of various tools were made pragmatically to ensure simple interface without completely compromising the quality of results. Other more intensive methods were discussed and it was especially highlighted where these tools will deliver more accurate results. The summary of this analysis is that FAST is a good option to integrate into QBlade but first the 2D structural properties and mode shapes are required as input. In order to get these properties a FEM solver is required and it was decided to create a module called QFEM. This code should be capable of creating an Euler-Bernoulli Beam element FEM model which is solvable by modal analysis.
Part II

Program Design
Chapter 3
Program Design

The QFEM module has been designed to be an essentially standalone module from QBlade. This is achieved by setting the interfaces of the program to be only at the start and end. The program accepts input blade data from QBlade at the beginning of the code via a number of calls to QBlade. For instance, the normalized coordinates of the blade sections are pulled from the QBlade data base along with the appropriate geometry information. The variables discussed so far are all readily available within QBlade, all other inputs are user defined.

Once the inputs have been processed, each of the 2D sections are analysed. The analysis works to apply analytical formulae to the sections to obtain 2D structural data. The 2D structural data can then be applied to form a number of beam elements along the blade. The beam elements are localised into the global system of equations which are then solved by the Eigenvalue problem code. The modes are sorted, fitted to polynomials and then fed back to QBlade along with other interesting outputs like the natural frequencies. Figure 3.1 demonstrates the flow of the program. The following section will discuss each of these steps in further detail.

This part of the thesis focusses on describing algorithms rather than code. References will be made to UML diagrams to help discussion but the full technical documentation is included in the QBlade Documentation. Here users can find specific descriptions of types, inputs, outputs and the different functions. This was done to focus discussion onto the engineering considerations rather than the programming structures.
3.1 Blade Geometry and Structural Properties

In order to run a Modal Analysis the input geometry and material data has to be handed from QBlade. This information falls into two categories; existing QBlade information and extra user inputs. During the QBlade aerodynamic design the blades are designed with all relevant parameters such as; chord length, angle of twist, location of pitch axis, and the normalized foil coordinates. However, extra inputs are required for the Elastic Modulo and Density of all materials as well as the parameters defining the shell thickness, spar thickness, spar location and spar angle where required. Little more detail is required than to state that in the current state the interface is contained in RunModalTest.h where the QBlade information is loaded into the QFEM data types.

3.2 2D Section Structural Properties
In this chapter the first part of the structural model is presented. In order to create the beam elements one must first formulate the structural properties at each of the connecting nodes. Therefore the structure sections need to be reduced into a set of useful values which can be then transferred to the beam elements.

**Algorithm**

Each wind turbine blade section can be reduced into three broad categories; Solid, Hollow or Hollow with an internal strengthening structure as shown in figures 3.4 & 3.5. Wind turbine blades can be constructed in many different ways, for example; both isotropic materials or complex composite layups can be used. However as already discussed in the introduction of this thesis; only isotropic materials will be considered. The spar material can be considered to be different to the outer shell material. Any estimation method must take into account these attributes.

Many different configurations exist for the internal spars. Two different examples are given in figures 3.4 & 3.5. Figure 3.4 shows a modern structure with two spars(note the thick spar caps forming a box beam). Figure 3.5 shows an older blade with an internal beam. For the first implementation of QFEM only a single case will be considered, that is: a single spar which can have customized thickness, chord-wise location and angle. These variables constitute the input during the construction of the blade sections. In the book Aerodynamics of Wind Turbines 2ed by Martin Hansen [23, p109] a set of integral formulae are presented. These formulae are used to find the structural properties of a wind turbine blade. The process starts by finding the values about a reference axis \((x_{R}, y_{R})\).

\[
\text{Longitudinal Stiffness}: [EA] = \int_A E \text{d}A \quad (3.1)
\]

\[
\text{Moment of Stiffness about the x Ref. Axis}: [ES_{xR}] = \int_A E_y y_{R} \text{d}A \quad (3.2)
\]

\[
\text{Moment of Stiffness about the y Ref. Axis}: [ES_{yR}] = \int_A E_x x_{R} \text{d}A \quad (3.3)
\]

\[
\text{Moment of Stiffness Inertia about the x Ref. Axis}: [EI_{xR}] = \int_A E_y y_{R}^2 \text{d}A \quad (3.4)
\]

\[
\text{Moment of Stiffness Inertia about the y Ref. Axis}: [EI_{yR}] = \int_A E_x x_{R}^2 \text{d}A \quad (3.5)
\]

\[
\text{Moment of Centrifugal Stiffness}: [ED_{xyR}] = \int_A E_x y_{R} \text{d}A \quad (3.6)
\]
3.2. 2D SECTION STRUCTURAL PROPERTIES

Figure 3.4: Example of a Wind Turbine Internal Structure. Picture taken with permission at Deutches Technikmuseum - Berlin[13]

Figure 3.5: Second Example of an older Wind Turbine Internal Structure - Picture taken at Blade Care Academy [64]
Where,

$$\int_A dA = \int_x \int_y dy \, dx = \text{Integration over the Foil Area}$$

In order to apply the formulae, the integrals were converted to discrete summations with each discrete element having the length $(x_{El}, y_{El})$:

- $EA = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} x_{El} y_{El}$  \hspace{1cm} (3.7)
- $ES_xR = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} x_{El} y_{El} y_m$  \hspace{1cm} (3.8)
- $ES_yR = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} x_{El} y_{El} x_n$  \hspace{1cm} (3.9)
- $EI_{xR} = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} \frac{E}{3} x_{El} \left( (y_m + \frac{y_{El}}{2})^3 - (y_m - \frac{y_{El}}{2})^3 \right)$  \hspace{1cm} (3.10)
- $EI_{yR} = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} \frac{E}{3} y_{El} \left( (x_n + \frac{x_{El}}{2})^3 - (x_n - \frac{x_{El}}{2})^3 \right)$  \hspace{1cm} (3.11)
- $ED_{xyR} = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} \frac{(x_n + \frac{x_{El}}{2})^2 - (x_n - \frac{x_{El}}{2})^2}{4} \left( (y_m + \frac{y_{El}}{2})^2 - (y_m - \frac{y_{El}}{2})^2 \right)$  \hspace{1cm} (3.12)

Where,

$x_{Res}, y_{Res} = \text{Summation Density}$

In practice, splitting the foil into a mesh or similar is a difficult task simply because it requires a more advanced definition of the geometry. Furthermore, manual intervention is commonly required to build a high quality mesh. To avoid this the algorithm is implemented with the help of the Clipper Library [28]. By using the Clipper Library it is possible to construct a polygon of the aerofoil coordinates and then determine whether a point lies in or outside. Therefore to implement the above formula an imaginary box is drawn around the foil, split into sections and then each section is integrated if it falls inside the polygon. Figure 3.6 shows a coarse representation of the integration pattern.

![Integration element (x_n,y_m)](image)

Figure 3.6: Discretization of the Foil Structure[28]

Now having established the method of integrating the foil shape, one must again consider how the blades are constructed. As already discussed it is possible that the blades will be hollow, with or without an extra structural spar. To account for all possible configurations the code implements a boolean addition of the elements as shown in figure 3.7. This means in a structure that is hollow with a spar, the integration is completed for the three elements (the hollow being taken as a negative element) and then summed. If the elements don’t exist, they simply have values of zero. An example of the programs output is displayed for a single blade and blade stack in Figures 3.8 & 3.9.

In so far, all values have been evaluated in terms of the Reference Axis. In order to use the values properly the foil has to be reoriented about the Elastic Axis of the foil(see
3.2. 2D SECTION STRUCTURAL PROPERTIES

Figure 3.7: Boolean Operations used to Create the 2D Section Models

Figure 3.8: Example Blade Section

Figure 3.9: Example Blade Stack
Again Hansen provides formulae to reorient the values in three steps. The first step finds Point of the Elasticity with respect to the Reference Axis.

\[
X_e = \frac{ES_eR}{EA} \quad (3.13) \\
Y_e = \frac{ES_yR}{EA} \quad (3.14)
\]

The second step is to shift the values to the Point of Elasticity.

\[
EI_{xe} = EI_{xR} - Y_e^2EA \quad (3.15) \\
EI_{ye} = EI_{yR} - X_e^2EA \quad (3.16) \\
ED_{xye} = ED_{xyR} - X_eY_eEA \quad (3.17)
\]

Finally the values are rotated onto the Principle Axes.

\[
\alpha = \frac{1}{2} \tan^{-1}\left(\frac{2ED_{xye}}{EI_{ye} - EI_{xe}}\right) \quad (3.18) \\
EI_1 = EI_{xe} - ED_{xye} \tan \alpha \quad (3.19) \\
EI_2 = EI_{ye} + ED_{xye} \tan \alpha \quad (3.20)
\]

Where,

\[
EI_1, EI_2 = \text{Moments of Stiffness Inertia about the elastic axes} \\
\alpha = \text{Angle between the reference axis and elastic axis}
\]

In the previous integration step the centroid of the foil was calculated. This serves as a reference point for the second integration pattern required to get the Polar Moment about the Centroidal Axis. The same integration pattern is used to generate this value using the formula;

\[
J' = \sum_{n=0}^{x_{Res}} \sum_{m=0}^{y_{Res}} \rho_m x E_I Y_EI \times \{(x_n - x_{Centroid})^2 + (y_m - y_{Centroid})^2\} \quad (3.21)
\]

Again the integration is applied for the Inner, Outer and Spar Sections. The boolean addition gives the final value.
3.2. **2D SECTION STRUCTURAL PROPERTIES**

**Program Implementation**

The 2D structural properties are created and stored by the StructElem Class. Each instance contains the definition of a node as well as the definition of an Inner, Outer and Spar Structural Integrator("StructuralIntegrator") object. The Structural Integrator objects are responsible for integrating over the area of the blade parts to obtain the physical properties. These are then combined in the StructElem Class (see figure 4.1).

The constructor method of StructElem is responsible for initializing the Blade Outer Integrator which must be present in all configurations. This is achieved by inputting the coordinates and some material properties into the 'Outer' Structural Integrator instance. The CalculateRef() method then generates the structural properties about the reference coordinate system. The mathematics behind this calculation have already been discussed in detail in the previous section.

The 'Outer' properties are naturally compulsory, however, the blade inner and spar sections can be optionally instantiated. StrucElem contains a method called 'CreateInner'. This method accepts an input of a double which determines the shell thickness in percentage of the chord. This is achieved by taking the 'Outer' polygon and then offsetting using the Clipper Library. The reference structural properties are then calculated for the 'Inner' Section.

If the foil is hollow- i.e.. has an 'Inner'- a spar can be added to the foil. The CreateSpar method will create such a spar using some geometry and material property definitions. The Spar is created by first drawing a rectangle polygon with the specified thickness at the chordwise location. The spar is then rotated as per the user input. The rectangle is defined to be much longer than the foil so when the rotation is complete the ends of the spar still protrude the foil. The Clipper Library is then used to create a copy of the 'Inner' foil polygon which is then used to clip the rectangle in an 'Union' boolean operation. Again the CalculateRef method is called to find the reference structural values.

The StructElem then takes the structural properties of all three parts and combines them. The Inner and Spar Elements are by default: initialized to zero, and will remain such unless redefined. The StructElem code takes the combined values and realigns them along the principle elastic axis. The process is then repeated to get the JPrime values which are then combined. The structural values for each 2D section are then stored in a StructElem instance.
Figure 3.11: UML Structural Element
3.3 Beam Elements

This chapter looks at how the finite elements are created from the formulae described by Larwood [35]. The elements are a Euler-Bernoulli beam with the properties linearly tapered from one node to the other. These elements include extra terms such that rotational effects are accounted for.

![Figure 3.12: Module 3](image)

**Algorithm**

An element capable of modelling a wind turbine blade must account for several phenomena. Firstly, the blade acts in the same manner as a beam with bending, axial elongation and twisting modes. Introducing the rotation creates extra inertial and gyroscopic forces. When the turbine blade bends, rotational forces will act to restore the blade, this acts analogous to an elastic force and thus changes the natural frequencies of the blade. On top of this the blades have highly complex geometry.

In order to account for all of these phenomena, Larwood [35] derived a set of matrices to represent the blades. The Tapered Beam Element has two nodes, each with 6 degrees of freedom, that is 3 translational and 3 rotational (see figure 3.14). The element takes two sets of sectional properties and interpolates between them to form the element (see figure 3.13). In the derivation, these linear interpolation terms are retained, such that for the purposes of this report, one simply needs to put the right values into the right variables.

The element matrices are lumped into two groups, the stiffness terms and the mass terms. The Mass Matrix consistently lumps mass and rotary inertia onto the degrees of freedom. The Spring Matrix is built up from Elastic, Axial Force, Spin Stiffness, Axial Reduction and Gyroscopic Matrices. With the exception of the gyroscopic matrix, all of these terms are symmetric. In line with Larwood’s[35] reasoning, the Gyroscopic terms will be ignored for two reasons; the Gyroscopic terms require the more complicated Duncan method as a solution and Larwood demonstrates that the Gyroscopic terms do not significantly influence the solution. Therefore, it is a sensible choice to disregard the Gyroscopic terms. The remaining terms are combined linearly.

The exact matrix formulae will not be repeated here as they are a full set of $12\times12$ matrices. The implementation of these formulae can be found in the source code. However,
a number of important input variables for each of the element matrices are highlighted here.

\[
EI_1, EI_2 = \text{Moment of Stiffness Inertia about the elastic axes} \\
EA = \text{Axial Stiffness} \\
\alpha = \text{Angle between the reference axis and elastic axis} \\
\rho_{\text{linear}} = \text{Linear Density of the blade (i.e. kg/m)} \\
GJ = \text{Torsional Stiffness} \\
J' = \text{Polar Moment of Inertia about the Centroidal Axis}
\]

All of these values are subscripted by '0' and 'l' depending on whether the properties are drawn from the node which is closer to the hub or tip respectively. This convention is demonstrated in figure 3.14.

Following Larwoods’ Convention the DOF Vector is defined as follows.

\[
\{ x_0, y_0, z_0, \alpha_{x0}, \alpha_{y0}, \alpha_{z0}, x_l, y_l, z_l, \alpha_{xl}, \alpha_{yl}, \alpha_{zl} \}^T
\]

(3.22)

Where \( \alpha \) represents the angle at the node respective to the subscripted axis. The local element matrices are all set up using this convention.
In so far, significant discussion of damping has been neglected. Material damping can be included in a number of ways. With very specific material knowledge one can include viscous damping into the first formulation and then solve using the Duncan Method. This method is not commonly used as such specific material knowledge is rarely known a priori. A more simple method is needed to account for the damping.

Modal Damping or Rayleigh Damping is included in a different manner. The mode shapes and frequencies are derived without damping and then the damping is post applied. This is achieved by giving individual mode shapes damping rather than the degrees of freedom. This form of damping is readily measured using techniques such as ping testing. Developing the modal damping analytically is well beyond the scope of this project and is left to the user to input. Colloquial evidence suggests that $\zeta = 2 - 3\%$ is a reasonable estimation for composite blades in absence of better information [31].

In the Fortran implementation of the code by Larwood, a set of matrix transformations were created. This convention was retained, and so, the transformation is made in two steps. The first accounts for the slopes of the elements in 3D space and then the structural twist is accounted for. This is reduced into a $3 \times 3$ Matrix which is then copied 4 times to create the $12 \times 12$ transformation matrix. Larwood fully derives these conventions [35] so no in depth repetition will be made here.

Program Implementation

The Beam Elements are created and stored in an instance of the TaperedElem Class (as shown in figure 3.15). In this element all of the stiffness and mass terms are created from two StructElem instances. These are then transformed to the local axis and reduced into a Stiffness and Mass Matrix.

A TaperedElem instance is constructed with two StructElem instances and a nominal value for the rotational speed (user input). The two StructElem instances contain the structural properties at the hub and tip side of the beam element. As discussed above the structural properties are interpolated linearly across the element [35]. The formulae have already been derived so it is only necessary to insert the structural values from each of the nodes, that is, from each StructElem. This will produce all of the matrices but one.

The Axial Stiffness matrix is not an independent term, it relies on all elements further towards the tip than itself. This is logical as the axial stiffness term results from centripetal forces. In order to generate these terms a second method InitAxialStiff is created. All of the TaperedElements (including the one being operated on) are inserted into the method with an operator indicating the current elements’ position. The effects of the outer elements are then summed. The InitAxialStiff method is then responsible for performing the addition of all stiffness terms, thus creating the stiffness matrix. The final Mass and Stiffness matrices are transformed as the final step.

3.4 Beam Elements Localisation

A standard process of a Finite Element Code is to localise the element matrices. This step is essentially where all of the individual elements are glued together forming one large system of equations.

Algorithm

Each element matrix has 12 degrees of freedom, when the element does not have a free end, these 12 degrees of freedom are connected to another set of degrees of freedom. The
Figure 3.15: UML Tapered Element
connection between the elements is made by simple addition. The logic behind this is as follows; a single mass element matrix gives the contribution of mass on each degree of freedom due to that element. When the nodes are connected there are now two elements contributing mass to the degrees of freedom. In short, this is achieved through simple summation of the contribution of each element onto each degree of freedom.

To create a global system of equations the following steps are taken.

1. When each Node is created, it is assigned 6 numbers representing the degrees of freedom i.e.. '0,1,2,3,4,5'.

2. The total number of Degrees of Freedom is calculated (N).

3. Global Mass and Stiffness Matrices($N \times N$) are created.

4. Local Beam Matrix is added to the global matrix onto the coordinates given by the corresponding node number.

5. Process 4 repeated for all local elements for both the mass and stiffness matrices.

When this process is completed the beam is floating in free space with connected elements i.e.. x DOF of element 1 is connected to x DOF of element 2 ...

Currently, the mode shapes of the beam are from a free-free modal analysis i.e.. no DOFs are constrained. In a normal FEM solver, this results in Rigid Body Modes where the test object will mathematically fly off to infinity. For a modal analysis Rigid Body Modes can be isolated from the main solution leaving the real results. Jonkman describes -in the FAST documentation [32]- that the mode shapes input into FAST must be cantilevered. This means in practice that; the translational and rotational DOFs have to be constrained at the hub node, resulting in zero displacement and slope at that node. In order to achieve this, degrees of freedom must be constrained.

Constraining a FEM model means simply prescribing the displacements as zero and then deleting the corresponding row and column from the matrix. This achieved in two steps in QFEM, such that easy indexing within the program is maintained. For example; if DOF 2 was to be constrained, the 3rd row and column would be deleted(note +1 due to C++ indexing convention(figure 3.17)) leaving a N-1×N-1 set of matrices(figure 3.18).
After the reduced matrices are used to find the solution, these rows are restored with rows of zeros therefore there is no need to modify indices (figure 3.19). The result is a cantilevered solution while the program at large remains naïve to modifications to indexing. Now the matrices are ready to be solved.

Figure 3.17: Bogus Matrix boxed elements to be deleted to constrain the DOF # 2

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 2 & 0 & \cdots & 0 \\
0 & 0 & 0 & 3 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & 0 & 0 & N
\end{pmatrix}
\]

Figure 3.18: Bogus Matrix with DOF # 2 constrained
Program Implementation

The Modal Analysis solution is handled by an instantiation of the EQNMotion class. The EQNMotion class is responsible for constructing the full system of equations, solving the Eigenvalue problem and then post processing the mode shapes. The class also stores the equation matrices and solution data. The EQNMotion Class is instantiated with an input of the total DOFs in the system. This is used to initialize the global stiffness and mass matrices to the correct size of the equations.

Now that the class has been instantiated, it is time to fill it with elements. Elements are localised into the global matrix with the method: "AddTaperedElem". This method takes an input Tapered Element (TaperedElem) and then reads the 'Connectors' property within the object to create the localisation link. Each element- of course - has two coordinates (row and column), these are produced by the permutations of the Connectors Vector on itself. Let us take a random example that the first and second numbers in the "Connectors" Vector are 10 and 24. This means that the Local Matrix coordinate (1,1) would be added into Global Coordinate (10,1) Global Element (10,24) and so on. While all of this sounds awfully complicated, the following code snippet demonstrates the concept and just how simple it is. This process is repeated for Mass and Stiffness Matrices of all of the Tapered Elements.

```
for(int i = 0; i<Connectors.size(); i++)
{
    for(int j = 0; j<Connectors.size(); j++)
    {
        GlobalMatrix(Connectors.at(i), Connectors.at(j)) +=
        +ElemMatrix(i,j);
    }
}
```

Having now established the global system of equations it is necessary to constrain the degrees of freedom. As discussed this is achieved by removing rows and columns of the matrix. The method DeleteDOF takes the columns to the right of the target column and shifts them all to the left, thus deleting the target column. This is then repeated for the rows where the rows are moved up. The final row and column are then removed from the
matrix using the conservative resize method within Eigen.

When there is more than one row to be delete, special care is taken to retain the correct indexing. This is achieved by first sorting the DOFs that are to be deleted. The highest numbered DOFs are deleted first so that the lower values are not affected. This process is performed in reverse by the ReplaceDOF method such that in the final solution the constrained degrees of freedom will simply contain rows of zeros and all previous indexing conventions are retained.

### 3.5 Eigenvalue Problem

In the previous chapter the element matrices were localised into the global mass and stiffness matrices. A number of DOFs were constrained so that Rigid Body Modes won’t be present in the solution. Having done this, the equations can be assembled into the Eigenvalue Problem. In section 2 we explored the two possible solution methods available for solving the Eigenvalue problem. As discussed a number of times the Duncan method will not be employed so the solution will take the form;

$$\omega^2 \{x\} = [M]^{-1}[K]\{x\}$$

(3.23)

and as such the matrix can be placed into a standard Eigenvalue solver. For this implementation the 'Eigen' library was included in the code. The output from this code is a Vector and a matrix of complex numbers, these are Eigenvalues and EigenVectors respectively. Persons seasoned in modal analysis may have concern seeing the output as a complex number, however the complex terms are without exception zero. This inclusion is simply a result of using the external library and the complex numbers are thrown away from the result. As this code is largely externally sourced there is little need to go into further explanation[54].

---

![Diagram](module5.png)

**Figure 3.20: Module 5**

### 3.6 Mode Shape Polynomial

The final step in the analysis chain is to prepare the mode shapes for insertion into the FAST software. Jonkman and Buhl [32] explain that the mode shapes should be inserted
in the form of a polynomial normalised about the tip deflection. In order to do this a polynomial fitting algorithm has to be implemented.

![Figure 3.21: Module 6](image)

**Algorithm**

The input blade geometry can be defined by any number of nodes and therefore structural sections. However, FAST specifically requires a 6th order polynomial for each of; the first two Flapwise modes and the first Edgewise mode. This rules out a simple matrix coefficient inversion solution where the polynomial order is determined by the number of input points. This method could be used but this would then require some sort of complicated spline algorithm to reduce the degree of the equation. A better method is to use a Least Squares solver.

Least Squares solvers all work on the same basic principle as all optimisation problems. In this case the penalty function is the sum of the differences between the guessed function and the actual data points, this is the simple part. The more complicated part is creating a reliable "next guess". Luckily such codes have already been implemented in the GNU GSL Library [21].

Using code from the Rosetta Code Website [50] it was possible to construct a solver by feeding the built in GSL linear solver, an x Coefficient Matrix(like that shown in figure 3.22) as well as responses at each node. When the order is set to 7; a full 6th order solution is produced by the code, raising a slight issue. FAST requires that the mode shapes are cantilevered, thus the first and second coefficients of the polynomial have to be zero. This means that the coefficient matrix is slightly altered with two columns to zeros to give the reduced input matrix as shown in figure 3.23. The result of the algorithm is a sixth order polynomial with the first two coefficients set to zero (see examples in figures 3.24 &3.25). These figures show that the polynomial fitting algorithm is functioning very well.
Figure 3.22: Input Full Coefficient Matrix

\[
\begin{pmatrix}
  x_0^0 & x_1^1 & x_2^1 & x_1^2 & x_0^3 & x_1^4 & x_0^5 & x_1^6 \\
  x_2^0 & x_2^1 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & x_2^6 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  x_n^0 & x_n^1 & x_n^2 & x_n^3 & x_n^4 & x_n^5 & x_n^6 & x_n^7
\end{pmatrix}
\]

Figure 3.23: Input Reduced Coefficient Matrix

\[
\begin{pmatrix}
  0 & 0 & x_1^2 & x_1^3 & x_1^4 & x_1^5 & x_1^6 \\
  0 & 0 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & x_2^6 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & 0 & x_n^2 & x_n^3 & x_n^4 & x_n^5 & x_n^6
\end{pmatrix}
\]
3.6. MODE SHAPE POLYNOMIAL

Figure 3.24: Example Polynomial fitted to the First Flap Mode

Figure 3.25: Example Polynomial Fitted to the Second Flap Mode
Program Implementation

This section of the program fits polynomials to the modes, sorts them into mode families (i.e. Flap or Torsion) and then sorts them by natural frequency. Again these tasks are handled within the EQNMotion class where the FitPolynomial method processes the modes and a Mode object is instantiated for each individual mode shape. This 'Mode' class contains the FAST polynomial, a vector indicating the deformation at each node as well as the frequency. Importantly the Mode class also contains an enumeration indicating what type of mode shape it is.

The sorting into mode groups is achieved at the same time as creating the polynomials. For each mode shape the deformation at each node is considered. For each node 6 values are assigned, one for each of the degrees of freedom. In order to automatically sort the modes into categories QFEM assumes that the Flap and Edgewise modes will vibrate about the principle axes (an assumption also made in FAST[32]). This method is used so that twisted blades can be correctly identified.

In order to test whether a node is deforming in the Flap or Edgewise direction, a Dot Product is taken between the unit vector of the deformation and the principle axes unit vector. In figures 3.26 & 3.27 there are two cases: one where the Dot Product will strongly project onto one axis and one where no conclusion can be reached. The Dot Product at each node is weighted by the magnitude of deformation and then summed (the axial and Z Torsion Terms are also summed at this point). For low order modes and uncomplicated geometry this method works well. In cases where the summation doesn’t strongly indicate a mode type it is declared as Unsorted. In most cases the algorithm will then label the mode as either: Flap, Edge, Axial or Torsion. This algorithm accounts for Torsion along the longitudinal axis of the blade but ignores the other two angular degrees of freedom.

![Figure 3.26: Case with strong projection onto one axis](image)
Figure 3.27: Case where one axis doesn’t dominate
Part III

Testing
Chapter 4

Test Cases

In a program such as QFEM it is possible to input an infinite choice of geometry and operating conditions. Hopefully by now readers are well acquainted with the methods implemented in QFEM. As extensively discussed in the Literature Review modal analysis has limitations, it is true that highly unusual geometry will probably move beyond these limitations. As with any simulation software the results are a reflection of the quality of the program and the quality of input data. In this chapter of the thesis a number of test cases are made in order to verify key features of the program against known cases. The methodology was to seek test cases that could isolate particular parts of the program to test their performance. Some of the cases seek to test the overall performance of the program, others seek to test the 2D integration routines, and some test particular changes to geometry. These cases will constitute the beginning of the testing of QFEM. As the integration into the QBlade GUI is complete more test cases can be added to the documentation of QBlade.

4.1 Isotropic Cantilevered Beam Analysis

A rotating beam is a rather complicated example of modal analysis. It would not be sensible to begin base lining the code with a case that involves so many variables and terms. Therefore the first test case will consider a simple cantilevered beam as described in Reference [5]. These lecture notes give an excellent baseline case which will be used to test the performance of the QFEM code. The beam is defined with the following dimensions;

With the following properties;

\[ E = 70 \text{GPa} \] (4.1)
\[ \nu = 0.35 \] (4.2)
\[ \rho_m = 2700 \text{kg/m}^3 \] (4.3)
\[ I = \frac{1}{192} m^4 \] (4.4)

Reference [5] uses Euler-Bernoulli Beam Theory to find the natural frequencies. The theory states that;

\[ EI \frac{d^4 y}{dx^4} + \rho A \frac{d^2 y}{dx^2} = 0 \] (4.5)

Splitting this equation into Spatial and Temporal Terms with;

\[ Y(x,t) = \overbrace{X(x)}^{\text{Spatial Temporal}} \overbrace{\tau(t)}^{\text{Temporal}} \] (4.6)
CHAPTER 4. TEST CASES

Figure 4.1: Cantilevered Beam Example Dimensions

Gives;

\[ \frac{EI}{\rho A} \frac{d^4y}{dx^4} \tau(t) + \frac{d^2y}{dx^2} X(x) = 0 \] (4.7)

The spatial solution is assumed to be of the form;

\[ X(x) = C_1 \cos x \lambda + C_2 \sin x \lambda + C_3 \cosh x \lambda + C_4 \sinh x \lambda \] (4.8)

Where \( \lambda \) is defined by;

\[ \lambda = \left( \frac{EI}{\rho A} \omega^2 \right)^{\frac{1}{4}} \] (4.9)

The beam is considered to be cantilevered exactly as assumed in the FAST documentation. This results zero displacement and slope at the fixed end giving;

\[ y(0) = 0 \] (4.10)
\[ \frac{d^2}{dx^2} y(0) = 0 \] (4.11)

The free end is assumed to have zero shear and moment giving;

\[ \frac{d^2}{dx^2} y(l) = 0 \] (4.12)
\[ \frac{d^3}{dx^3} y(l) = 0 \] (4.13)

These assumptions provide the results that;

\[ C_2 = -C_4 \] (4.14)
\[ C_1 = -C_3 \] (4.15)

Solving for \( C_1 \) and \( C_2 \) gives;

\[ \cos \lambda L \cosh \lambda L = -1 \] (4.16)
4.1. ISOTROPIC CANTILEVERED BEAM ANALYSIS

The roots of this equation are:

\[ \lambda_nL = \frac{(2n - 1)\pi}{2} \] (4.17)

The temporal equation is reduced to:

\[ \tau(t) = b_1 \sin \left( \lambda_n^2 \sqrt{\frac{EI}{\rho A}} t \right) + b_2 \cos \left( \lambda_n^2 \sqrt{\frac{EI}{\rho A}} t \right) \] (4.18)

The frequency can then be extracted the harmonic equation’s general form giving:

\[ \omega_n = \frac{\lambda_n^2}{L^2} \sqrt{\frac{EI}{\rho A}} \quad \text{[rad/s]} = \frac{\lambda_n^2}{2\pi L^2} \sqrt{\frac{EI}{\rho A}} \quad \text{[Hz]} \] (4.19)

It is now possible to extract the natural frequencies of the beam. In this case the first six natural frequencies were compared to a baseline execution of QFEM with an integration density of 400 Elements per axis and 20 elements. The results (in Table 4.1) show an excellent comparison with this set up.

| Mode Number | Theoretical Frequency [Hz] | QFEM Solution [Hz] (Baseline with 20 Elements) | Deviation (

| 1 | 16.45 | 16.42 | <0.2% |
| 2 | 103.09 | 102.89 | <0.2% |
| 3 | 288.60 | 288.09 | <0.2% |
| 4 | 565.67 | 564.57 | <0.2% |
| 5 | 935.09 | 933.38 | <0.2% |
| 6 | 1396.89 | 1394.59 | <0.2% |

Table 4.1: First Six Mode Frequencies - Baseline QFEM study and Theoretical Solution

Now having established that the baseline agrees with the theoretical solution the effect of the main set up parameters will be explored. In the Structural Integrator Code (described in Section 3.2) an integration density is chosen. This value determines how finely the integration will be performed, a value of 100 means that the integration grid will be 100×100. For the baseline set up above this was run with a value of 400 elements.

To check the sensitivity of this value it was varied in increments from 10 to 1500. The results predictably show that at very low density the results fluctuate. At a density of 100 the low order modes have converged, at 500 the first 6 modes have converged (see Figure 4.2). The computational time - graphed in Figure 4.3 - does not critically increase in this range. A density of 400 seems to be a sensible trade off remembering that FAST only requires the first two Flap and first Edgewise Modes.

The second important variable is the number of Beam Elements used to model the Test Beam. The number of elements was varied between 6 (the minimum accepted by the program) and 100. Figure 4.4 shows the variance in both absolute terms and normalized by the converged value. The results show that higher order modes are more prone to error with low numbers of elements, this is to be expected. However even though the error exists, it is still acceptably small. For a Simple Beam the user can use a relatively low number of elements to achieve an accurate result.

The structure of the program has a 1:1 relationship between 2D and 3D data. That is, every time a Tapered Beam Element is created, the Structural Integrator is called a number
Figure 4.2: First Six Mode Frequencies with Increasing Integration Density

Figure 4.3: Computational Time with Increasing Integration Density (see Chapter 6 for Computer Specs)
of times. The Structural Integrator is one of the two intensive processes. In future improved versions, multiple sections with the same properties may draw from a '1:many' database arrangement. The second computational burden is the eigenvalue solution, increasing the number of elements will naturally increase this burden. Figure 4.5 shows the compound of both of these effects, though most users will be unhindered within reasonable operation.

Figure 4.4: First Six Frequencies with Increasing Beam Element Density
4.2 Isotropic Cantilevered Hollow Blade Analysis

In his thesis, De Frias Lopez\cite{12} presents a test case where he conducts a modal analysis on a hollow aerofoil using, Abaqus Shell Elements, PreComp from NREL and his own newly implemented anisotropic beam element. This case provides a suitable baseline to compare the performance on slightly more complicated geometry (as defined by the properties in 4.3). This is achieved by inputting the geometry described in Table 4.3. In QFEM the rotational effects were neglected by putting $\omega = 0$.

<table>
<thead>
<tr>
<th>Blade Length [m]</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blade Chord [mm]</td>
<td>18.4</td>
</tr>
<tr>
<td>Chord Length [m]</td>
<td>1.68</td>
</tr>
<tr>
<td>Number of Sections [-]</td>
<td>7</td>
</tr>
<tr>
<td>Foil Type</td>
<td>a8-9 (See ch:Appendix B)</td>
</tr>
<tr>
<td>Support</td>
<td>Cantilevered</td>
</tr>
<tr>
<td>E [GPa]</td>
<td>10.30</td>
</tr>
<tr>
<td>G [GPa]</td>
<td>4.12</td>
</tr>
<tr>
<td>Density $\rho_m$ [kg/m$^3$]</td>
<td>1830.0</td>
</tr>
</tbody>
</table>

Table 4.2: Hollow Isotropic Blade Parameters \cite{12}

QFEM showed good agreement in the 2D sectional parameters for most values. But highlighted an assumption error with the torsional stiffness. In so far the code has erroneously assumed the section properties to be close to circular and therefore used the formulae stated in The Machinery’s Handbook \cite[p.143]{3}.

$$J_z = I_x + I_y$$ \hspace{1cm} (4.20)
4.2. ISOTROPIC CANTILEVERED HOLLOW BLADE ANALYSIS

Unfortunately this formula doesn’t hold and more rigorous investigation is required as detailed by Zehn [63]. The PreComp documentation indicates that an alternate method is used to find the torsional stiffness and De Frias Lopez highlights in later sections significant differences from PreComp and the Abaqus Shell element solution in torsion.

There are two methods to estimate the torsional properties. The thin shell section constant shear flow method as detailed by Zehn [63]. Such a method would only be valid for hollow blades with no spar. The second method would be to integrate a 2D FEM package into QFEM to analyse the section properties. It would be a more complicated property but would provide a far higher quality solution.

The analysis was continued to find the mode shapes and frequencies disregarding the torsional stiffness error. The results show an excellent comparison between the natural frequencies compared to both of the Abaqus De Frias Lopez Test Cases[12]. A comparison between the QFEM results and the De Frias Baseline is shown in table 4.4.

<table>
<thead>
<tr>
<th>Value</th>
<th>QFEM</th>
<th>Abaqus[12]</th>
<th>QFEM</th>
<th>Abaqus</th>
<th>QFEM</th>
<th>Abaqus</th>
<th>QFEM</th>
<th>Abaqus</th>
</tr>
</thead>
<tbody>
<tr>
<td>$EA$ [N]</td>
<td>6.29621e+08</td>
<td>6.3148e+08</td>
<td>-0.3%</td>
<td>6.2380e+08</td>
<td>0.93%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EI_1$ [Nm$^2$]</td>
<td>1.00409e+07</td>
<td>1.0533e+07</td>
<td>-4.62%</td>
<td>1.0620e+07</td>
<td>-5.40%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$EI_2$ [Nm$^2$]</td>
<td>1.38786e+08</td>
<td>1.3891e+08</td>
<td>0.09%</td>
<td>1.3370e+08</td>
<td>3.80%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$GJ$ [Nm$^2$]</td>
<td>6.59524e+07</td>
<td>1.1797e+07</td>
<td>474%</td>
<td>1.1710e+07</td>
<td>477%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{linear}$ [kg/m]</td>
<td>111.86</td>
<td>112.19</td>
<td>-0.29%</td>
<td>110.80</td>
<td>0.96%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha_{elasticity}$ [deg.]</td>
<td>3.14</td>
<td>3.21</td>
<td>-2.18%</td>
<td>3.288</td>
<td>-4.5%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{elasticity}$ [m]</td>
<td>0.776533</td>
<td>0.7680</td>
<td>1.11%</td>
<td>0.7458</td>
<td>4.12%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$y_{elasticity}$ [m]</td>
<td>0.016079</td>
<td>0.0166</td>
<td>-3.13%</td>
<td>0.160</td>
<td>0.49%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3: Hollow Performance against De Frias Lopez Benchmark [12]

Some important information can be extracted from this table. Firstly it appears that the CurveFast Beam element formulation matches well with the De Frias Lopez formulation for isotropic beams. It also appears that a good comparison can be made between the shell element simulation and the QFEM Beam simulation, even in the presence of an incorrect torsional stiffness. Qualitatively the mode shapes appear to be the same though no exact analysis could be achieved due to lack of data from the baseline case.

The incorrect torsional stiffness was directly addressed with a second modal analysis. In the code, the torsional stiffness was temporarily divided by a factor of five to bring the
value closer to the Shell Element estimation. The results are shown in the far right column of table 4.4. For this particular geometry - simple straight aerofoil beam - it appears that as expected the torsional stiffness will not affect the low order frequencies. In summary, for the modes to be input into FAST, this is not a sensitive input value and QFEM is still performing well.

4.3 Non-Rotating Tapered Beam

Geometry variations can occur due to the foil profile itself or by a change of properties over the length of the element. The Tapered Beam Element was implemented by Larwood [35] so that properties could be linearly interpolated from properties at each of the two nodes. An obvious test case to show this functionality is by creating a tapered beam, such a case was provided in a benchmark by Alarcon [2].

In this test a square cross sectioned tapered beam (defined in table 4.5) was defined and then solved using a number of methods. Firstly, an analytical solution was obtained using a similar procedure to that described in Section 4.1. Two FEM solutions were then derived using SAMCEF Field as well as the package being tested: Aster. The results of these analyse are presented in table 4.6.

The results show again that there is an excellent correlation the Euler-Beam analytical solution. When one compares the results from Aster and Samcef it appears that a consistent error band under 5% exists. This variance is to be expected given that these solutions are full 3D FEM. It is possible to conclude that for this test case QFEM performs well.

<table>
<thead>
<tr>
<th>Beam Length [m]</th>
<th>Width at Support [m]</th>
<th>Height at Support [m]</th>
<th>Width at Tip [m]</th>
<th>Height at Tip [m]</th>
<th>Number of Sections [-]</th>
<th>Interpolation Type</th>
<th>E [GPa]</th>
<th>Density $\rho$ [kg/m$^3$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.04</td>
<td>0.04</td>
<td>0.01</td>
<td>0.01</td>
<td>20</td>
<td>Linear</td>
<td>2e11</td>
<td>7800</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.5: Tapered Beam Properties [2]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$ [Hz]</td>
<td>54.16</td>
<td>54.18</td>
<td>-0.03%</td>
<td>56.84</td>
<td>-1.71%</td>
<td>56.85</td>
<td>-1.73%</td>
</tr>
<tr>
<td>$f_2$ [Hz]</td>
<td>172.19</td>
<td>171.94</td>
<td>0.14%</td>
<td>180.0</td>
<td>-4.33%</td>
<td>180.08</td>
<td>-4.38%</td>
</tr>
<tr>
<td>$f_3$ [Hz]</td>
<td>385.11</td>
<td>384.40</td>
<td>0.18%</td>
<td>401.0</td>
<td>-3.96%</td>
<td>401.23</td>
<td>-4.01%</td>
</tr>
<tr>
<td>$f_4$ [Hz]</td>
<td>698.54</td>
<td>697.24</td>
<td>0.19%</td>
<td>723.2</td>
<td>-3.41%</td>
<td>724.02</td>
<td>-3.51%</td>
</tr>
<tr>
<td>$f_5$ [Hz]</td>
<td>1115.4</td>
<td>1112.28</td>
<td>0.28%</td>
<td>1145.41</td>
<td>-2.62%</td>
<td>1147.51</td>
<td>-2.80%</td>
</tr>
</tbody>
</table>

Table 4.6: Tapered Non-Rotating Beam against Aster Benchmark [12]
4.4 Rotating Beam

The testing so far has verified the accuracy of the natural frequencies with various geometries. None of these cases have so far considered a case where the beam is rotating as a real wind turbine blade would. The following test is aimed at verifying whether QFEM is valid for rotating beams.

Munteanu et al.[19] presented an excellent test case of a simple rotating beam. The beam had a constant square cross section along the length and was cantilevered to the rotating center. The beam had the following characteristics:

- Length = 0.3 m
- Cross Section Height = 0.005 m
- Cross Section Width = 0.005 m
- $E = 200\, GPa$
- $\rho_m = 7800\, kg/m^3$

A modal analysis was then conducted on the beam with varying rotational speeds.

A modal analysis is a linearisation of the system dynamics. It is highly useful but fails to capture effects such as centrifugal stiffening which change the modal frequencies at different rotational speeds. Both the QFEM and Munteanu et al[19] equations have terms to include the effects of rotational speed. The axial stiffness terms account for the centrifugal forces adding a steady pretension to the beam in the axial direction, this tends to increase frequency. However, if only axial stiffness terms are included the frequency will be overestimated. This is because the formulae so far fail to account for large deformations, this is accounted for by the spin stiffness terms. In table 4.7 and figure 4.6 results are presented with and without spin stiffness terms to demonstrate that the individual QFEM rotational matrices are functioning correctly. The QFEM and Munteanu et al. baseline match well but exact numbers are not compared for this case as the results were digitized from the baseline paper.

<table>
<thead>
<tr>
<th>Rotation Speed $\Omega [\text{Rad/s}]$</th>
<th>First Natural Frequency $\omega_n$ with Spin Stiffness Terms $[\text{Rad/s}]$</th>
<th>First Natural Frequency $\omega_n$ without Spin Stiffness Terms $[\text{Rad/s}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>285.996</td>
<td>285.996</td>
</tr>
<tr>
<td>100</td>
<td>289.322</td>
<td>306.117</td>
</tr>
<tr>
<td>200</td>
<td>298.745</td>
<td>359.511</td>
</tr>
<tr>
<td>300</td>
<td>312.876</td>
<td>433.464</td>
</tr>
<tr>
<td>400</td>
<td>330.097</td>
<td>518.618</td>
</tr>
<tr>
<td>500</td>
<td>349.024</td>
<td>609.769</td>
</tr>
</tbody>
</table>

Table 4.7: Spinning Beam Element Test Case QFEM Results
4.5 Composite Blade Test Case

Vasjaliya and Gangadharan conducted research to optimise Aero-Structural design of a Composite Wind Turbine Blade [55]. In this paper the authors choose three different configurations of composite layup and compared the modal frequencies using Ansys. This provides a good baseline to compare against.

This case is unlike the other cases, the material data given is for a composite lay up. Some assumptions had to be made in order to reduce the data into isotropic properties. This then constitutes a less rigorous test but demonstrates a useful approach of QBlade, the ability to remove some of the complexity to produce a first guess answer. The material properties were simplified from the fibre lay up to a guessed mean to give:

\[
E = 3.0 \times 10^5 [GPa]
\]

\[
\rho_{\text{material}} = 1900 [kg/m^3]
\]

Shell Thickness = 5%

The geometry listed in table 4.9 was input via QBlade. QFEM was then called from QBlade to generate the frequencies.

These results (table 4.9 gave a reasonably good estimation of the the mode shapes with the Flap modes. The Edge mode had a considerable margin of 10%. The key question is why such good matching in one direction and not the other? The reason that the direction of composite fibres in the turbine blade will affect the blade properties. In this case the material properties have been tuned for the Flapwise direction but fail to estimate the
4.5. COMPOSITE BLADE TEST CASE

<table>
<thead>
<tr>
<th>Station</th>
<th>Blade Location [m]</th>
<th>Rotor Radius [m]</th>
<th>Chord [m]</th>
<th>Twist Angle [degree]</th>
<th>Aerofoil</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3048</td>
<td>0.9398</td>
<td>0.4528</td>
<td>0</td>
<td>Circle</td>
</tr>
<tr>
<td>2</td>
<td>0.9144</td>
<td>1.5494</td>
<td>0.7475</td>
<td>0</td>
<td>Circle</td>
</tr>
<tr>
<td>3</td>
<td>1.524</td>
<td>2.159</td>
<td>1.1176</td>
<td>20</td>
<td>S808</td>
</tr>
<tr>
<td>4</td>
<td>2.1336</td>
<td>2.7686</td>
<td>1.0944</td>
<td>14.81</td>
<td>S807</td>
</tr>
<tr>
<td>5</td>
<td>2.7432</td>
<td>3.782</td>
<td>1.0520</td>
<td>10.61</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.3528</td>
<td>3.9878</td>
<td>0.9974</td>
<td>7.29</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.9624</td>
<td>4.5974</td>
<td>0.9324</td>
<td>4.74</td>
<td>S805A/S807</td>
</tr>
<tr>
<td>8</td>
<td>4.572</td>
<td>5.207</td>
<td>0.8587</td>
<td>2.87</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5.1816</td>
<td>5.8166</td>
<td>0.7775</td>
<td>1.57</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5.79</td>
<td>6.4262</td>
<td>0.6891</td>
<td>0.74</td>
<td>S805A</td>
</tr>
<tr>
<td>11</td>
<td>6.4008</td>
<td>7.0358</td>
<td>0.5938</td>
<td>0.27</td>
<td>S805A/S806A</td>
</tr>
<tr>
<td>12</td>
<td>7.0104</td>
<td>7.6454</td>
<td>0.4927</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>7.62</td>
<td>8.255</td>
<td>0.3958</td>
<td>0</td>
<td>S806A</td>
</tr>
</tbody>
</table>

Table 4.8: Dimensions of the the SERI-8 blade[55]

Edgewise properties well.

A second case was run using the average of the E2 directional material properties for all three materials used. This has little scientific basis but is used to demonstrate a lower limit. The results of this assumption are presented in table 4.10. We can see here that the results are not a good match at all demonstrating that a lower bound assumption is simply not sensible.

\[
E = 9.5\times10^9[GPa]
\]

\[
\rho_{\text{material}} = 1900[kg/m^3]
\]

Shell Thickness = 5%

For such cases QBlade users may simplify the composite material into an isotropic material. What is critically important for users to understand that it is difficult to predict exactly what effect this will have. Some modes may match, others not. Using such an approach should be considered a Black Art. The author therefore suggests that such a simple approach can be used but only as an engineering first guess.

<table>
<thead>
<tr>
<th>QFEM Mode Shapes</th>
<th>Vasjaliya Baseline Configuration</th>
<th>QFEM</th>
<th>Vasjaliya Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flap Mode 1</td>
<td>4.51225</td>
<td>4.54</td>
<td>-0.66%</td>
</tr>
<tr>
<td>Edge Mode 1</td>
<td>9.08494</td>
<td>8.19</td>
<td>10.1%</td>
</tr>
<tr>
<td>Flap Mode 2</td>
<td>13.2323</td>
<td>12.95</td>
<td>2.1%</td>
</tr>
</tbody>
</table>

Table 4.9: Comparison between QFEM and Vasjaliya[55]
### Table 4.10: Comparison between Minimum Bounds QFEM and Vasjaliya [55]

<table>
<thead>
<tr>
<th>Mode</th>
<th>QFEM Mode Shapes [Hz]</th>
<th>Vasjaliya Baseline Configuration [Hz]</th>
<th>QFEM Vasjaliya Baseline %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>2.41461</td>
<td>4.54</td>
<td>-46.9%</td>
</tr>
<tr>
<td>Mode 2</td>
<td>7.2133</td>
<td>8.19</td>
<td>-11.9%</td>
</tr>
<tr>
<td>Mode 3</td>
<td>14.7184</td>
<td>12.95</td>
<td>11.3%</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusion

In thesis a Finite Element Modal Analysis solver called QFEM was written. The intention of this solver was to create an easy and robust way for users to find the mode shapes of Wind Turbine blades. These mode shapes and additionally the 2D structural data form crucial input into the FAST Program which is capable of simulating aeroelastic responses of Wind Turbines under many different operating conditions.

The program was implemented and at the time of writing this thesis is partially integrated into QBlade. Testing shows that the solver is working well for a range of test cases though highlighted some issues regarding estimation of torsion. The mode sorting algorithm works but the heuristics will fall over for very complicated geometry. However, minor issues aside the program is working very well.

5.1 Further Work

At its current state QFEM is in the minimum functional state. What this means that there is opportunity to add many more useful features onto the package but the core solver is working. The author recommends that integrating an open source FEM package into the solver will have a number of benefits. Firstly, the 2D sectional data can be analysed more accurately especially for the Torsional Stiffness. Furthermore, this allows extra steps to be undertaken such as failure analysis.

Currently the QFEM software only allows the addition of a single spar to the 2D section properties. This is not the only possible configuration. The structural integrator class have been set up to be generic so that any number of parts can be added. Some modification would have to be made to StructElem but these would be mostly superficial. This could add the functionality to have Box Spars, I Beams and C Sections. So that more realistic configurations can be made.

Performance wise this package is currently acceptable. However, as more parts are added the calls to QFEM will become more heavy and may become a bottle neck for QBlade. QBlade already contains libraries for making C++ code run asynchronously thus attaining considerable performance gains is a modest goal.
Bibliography


Glossary

**Aerodynamic Centre** The point on the aerofoil used for aerodynamic analysis. This point is where the Moment does not change with lift/angle of attack. Do not confuse with the centre of pressure.[14]. 65

**Aerodynamic Centre** The point where all of the aerodynamic forces act with no moment. 65

**Angle of Attack - AOA** Angle between the foil and the incoming fluid. 65, 74

**Axial Induction Factor** Relation between the wind speed before and after the turbine (a). $u = (1 - a) \cdot V_0$ - Where $V_0$ = Wind Velocity, $u$ = Induced Velocity at the turbine. 65

**DES** Detached Eddy Simulation. 65

**FAST** Aeroelastic Analysis Package from NREL. 65

**Hard Stall** When separation of the flow begins at the leading edge when the angle of attack is increased, almost instantaneously.[24]. 65

**HAWT** Horizontal Axis Wind Turbine. 65

**LES** Large Eddy Simulation. 65

**NREL** is an acronym for National Renewable Energy Laboratory based in Boulder, Colorado. 65

**QBlade** Freeware code for Wind Turbine Analysis from the Technical University of Berlin. 65

**RANS** Reynolds-averaged N-S. 65

**Soft Stall** When separation of the flow begins at the trailing edge and the stalled region propagates gently towards the leading edge with increasing angle of attack [24]. 65
Part IV

Appendicies
Chapter 6

Appendix A - Baseline Computer

Figure 6.1: Baseline Computer set-up used for performance indications
Chapter 7

Appendix B - AF8-9 Coordinates
<table>
<thead>
<tr>
<th>x Coordinates</th>
<th>y Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.00116</td>
<td>0.00703</td>
</tr>
<tr>
<td>0.0083</td>
<td>0.00189</td>
</tr>
<tr>
<td>0.0206</td>
<td>0.0313</td>
</tr>
<tr>
<td>0.0377</td>
<td>0.0438</td>
</tr>
<tr>
<td>0.0592</td>
<td>0.0561</td>
</tr>
<tr>
<td>0.0847</td>
<td>0.0679</td>
</tr>
<tr>
<td>0.11409</td>
<td>0.079</td>
</tr>
<tr>
<td>0.14685</td>
<td>0.0892</td>
</tr>
<tr>
<td>0.18266</td>
<td>0.0982</td>
</tr>
<tr>
<td>0.22111</td>
<td>0.1058</td>
</tr>
<tr>
<td>0.26177</td>
<td>0.11175</td>
</tr>
<tr>
<td>0.30418</td>
<td>0.11564</td>
</tr>
<tr>
<td>0.34829</td>
<td>0.11696</td>
</tr>
<tr>
<td>0.39439</td>
<td>0.11573</td>
</tr>
<tr>
<td>0.59199</td>
<td>0.0947</td>
</tr>
<tr>
<td>0.64174</td>
<td>0.087</td>
</tr>
<tr>
<td>0.69037</td>
<td>0.0787</td>
</tr>
<tr>
<td>0.73723</td>
<td>0.072</td>
</tr>
<tr>
<td>0.78169</td>
<td>0.0615</td>
</tr>
<tr>
<td>0.82312</td>
<td>0.0528</td>
</tr>
<tr>
<td>0.8646</td>
<td>0.0357</td>
</tr>
<tr>
<td>0.9238</td>
<td>0.0271</td>
</tr>
<tr>
<td>0.94879</td>
<td>0.0185</td>
</tr>
<tr>
<td>0.98582</td>
<td>0.0047</td>
</tr>
<tr>
<td>0.99632</td>
<td>0.00112</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.99606</td>
<td>0.00102</td>
</tr>
<tr>
<td>0.98364</td>
<td>0.00377</td>
</tr>
<tr>
<td>0.96197</td>
<td>0.00701</td>
</tr>
<tr>
<td>0.93154</td>
<td>0.00910</td>
</tr>
<tr>
<td>0.89348</td>
<td>0.00917</td>
</tr>
<tr>
<td>0.79901</td>
<td>0.00134</td>
</tr>
<tr>
<td>0.74529</td>
<td>-0.00701</td>
</tr>
<tr>
<td>0.68912</td>
<td>-0.0182</td>
</tr>
<tr>
<td>0.63189</td>
<td>-0.0317</td>
</tr>
<tr>
<td>0.57485</td>
<td>-0.047</td>
</tr>
<tr>
<td>0.51999</td>
<td>-0.0633</td>
</tr>
<tr>
<td>0.46552</td>
<td>-0.0796</td>
</tr>
<tr>
<td>0.41483</td>
<td>-0.0951</td>
</tr>
<tr>
<td>0.36753</td>
<td>-0.10887</td>
</tr>
<tr>
<td>0.32394</td>
<td>-0.1199</td>
</tr>
<tr>
<td>0.28389</td>
<td>-0.12736</td>
</tr>
<tr>
<td>0.24712</td>
<td>-0.13079</td>
</tr>
<tr>
<td>0.21322</td>
<td>-0.12971</td>
</tr>
<tr>
<td>0.18122</td>
<td>-0.12425</td>
</tr>
<tr>
<td>0.12193</td>
<td>-0.10412</td>
</tr>
<tr>
<td>0.0952</td>
<td>-0.091</td>
</tr>
<tr>
<td>0.0708</td>
<td>-0.0766</td>
</tr>
<tr>
<td>0.0492</td>
<td>-0.0616</td>
</tr>
<tr>
<td>0.031</td>
<td>-0.0465</td>
</tr>
<tr>
<td>0.0164</td>
<td>-0.0316</td>
</tr>
<tr>
<td>0.00607</td>
<td>-0.0175</td>
</tr>
<tr>
<td>0.00048</td>
<td>-0.0047</td>
</tr>
</tbody>
</table>
Chapter 8

Appendix C - Wind Turbine Aerodynamics

8.1 1D Momentum Theory

Wind turbines are a highly complicated device operating in highly complicated conditions. Taking all the possible complexity into a design on the first pass of analysis is not a wise move in engineering, and as such we should first establish a basic model. One dimension momentum theory places an ideal disk into the wind flow in order to extract momentum. An ideal disk is assumed to be thin, frictionless, solid while permeable to fluids. The wind is considered to be steady, uniform, inviscid and isolated from any secondary interference such as; trees, houses or topology. One dimension momentum theory therefore, disregards many critical details.[24]

Why is it useful to develop such a model? By combining the described boundary conditions to the standard fluid equations - the Bernoulli Equation, the conservation of mass and the conservation of momentum equations - it is possible to create a number basic relationships. A typical result is to establish a relationship between $V_0$ (Free Stream Velocity) and $u$ (Flow Velocity at the exact point of the thin disk). The model also indicates that the wind velocity reduces half speed in front of the rotor and half behind and that the pressure is discontinuous at the rotor. By far the most important result derived from this model is the Betz Limit which states that;

$$C_{\text{power,max}} = \frac{16}{27}$$  \hspace{1cm} (8.1)

Where;

$$C_{\text{power}} = \frac{P}{\frac{1}{2} \rho f V_{\infty}^3 A_{\text{disc}}}$$  \hspace{1cm} (8.2)

The Betz Limit is the maximum achievable Coefficient of Power ($C_p$) with the theoretical ideal rotor, therefore a real rotor(un-shrouded) must have a lower $C_p$ than this value. Literature often references efficiencies in terms of the Betz Limit e.g. 'The turbine achieved 60% of the Betz Limit'.[24]

For certain conditions [Low Axial induction Ratio] 1D momentum theory is a useful first step design tool. However, 1D momentum theory is limited because it fails to account for viscous effects and exit swirl (i.e. assumes there is a stator) amongst the other simplifications.[24]
8.2 Turbine Blade Aerodynamics

The model established above assumes an 'ideal disk' is placed into the flow. In reality an ideal disk doesn’t exist, therefore we must look in more detail at the actual aerodynamics. Once again we do not jump straight into the aerodynamics of the rotating blades, but first establish an understanding of the 2D aerodynamics of a single cross section of the blade.

In 1858, Hermann von Helmholtz [59] published a paper describing that a vortex will be generated when two streams of different velocity meet at a sharp edge such as the trailing edge of an aerofoil. Ludwig Prandtl commented on the vortex shedding saying that "Nature does not like discontinues" [60]. Von Karman explains - in his book "Aerodynamics" [60] - that the vortex generated at the trailing edge of an aerofoil will have the reaction of creating circulation about the foil. The addition of circulation to the uniform flow field results in a difference in pressures and velocity on the top and bottom of the aerofoil. What these notable men had discovered is that the top side of the aerofoil will have a low pressure and the bottom side a high pressure, they had described Lift.

Lift is not the only force generated on an aerofoil. In viscous flow we are able to overcome d’Alembert’s paradox to explain a presence of a turbulent wake at the trailing edge of the foil. As fluids slide past the surface of the aerofoil a shear layer is generated and the result is skin friction. These two effects among others create forces parallel to the flow and are referred to as Drag.[60]

The 'centre of pressure' of a foil is described to be the point where all of the aerodynamic forces can be reduced to a single resultant force with no moment. However, as we change the pitch of an aerofoil the centre of pressure changes making mathematical analysis difficult. Therefore, a second definition was established for the aerodynamic centre of the aerofoil. The point chosen is that the moment will not vary with angle of attack over the linear range analysis. The result is that to fully describe the forces a moment must be included. The sign convention for the aerofoil forces centred about the aerodynamic centre are shown in Figure 8.2.[14]

In the field of fluid mechanics it is very common to describe properties of an element using non-dimensional values. The motivation is that constructing a full ship, racing car or wind turbine is an extremely expensive process. Instead of having to test every new design, one can establish non-dimensional relationships so that results from one test can be extrapolated onto another where similarities exist. The non-dimensional values used to

![Figure 8.1: Streamlines and Stagnation points on an Aerofoil](image-url)
describe an aerofoil are as follows [24]; The Coefficient of Lift

\[ C_l = \frac{L}{\frac{1}{2} \rho V_{\infty}^2 c} \]  

(8.3)

Coefficient of Drag

\[ C_d = \frac{D}{\frac{1}{2} \rho V_{\infty}^2 c} \]  

(8.4)

Moment Coefficient

\[ C_m = \frac{M}{\frac{1}{2} \rho V_{\infty}^2 c^2} \]  

(8.5)

An aerofoil section operates in more than one condition. Quite often the pitch is changed resulting in a different angle between the foil and the wind. The variable \( \alpha \) (Angle of Attack - AOA) in figure 8.2 defines this angle. When the angle of attack is increased the Lift and Drag Coefficients also increase. For small angles of attack the change of Lift is defined by:

\[ \frac{dC_l}{d\alpha} \approx 2\pi \]  

(8.6)

At higher angles of attack the linear behaviour no longer holds, and likewise the circulation theory of Lift no longer predicts the behaviour. The linear model for Lift breaks down because it assumes that the streamlines follow the shape of the foil. At higher angles of attack it has been observed this assumption no longer holds due to the stall phenomenon.[60]

The expression "separation of the boundary layer" describes the phenomenon where the flow streamlines stop following the geometry of the solid object in the flow and instead forms into a large boundary layer. Separation occurs because adverse pressure gradients cause a reversal of flow beginning at the boundary layer where there is little momentum. Adverse pressure gradients occur in areas where the flow is decelerated, a good example is the rear half of an aerofoil.[24]

Separation is influenced by the state of flow. In turbulent flow separation is resisted by a stronger velocity gradient, in laminar flow separation occurs readily. It would seem logical to simply trip the boundary layer from laminar to turbulent. However, skin friction generated by a turbulent boundary layer is greater than that of laminar. Advanced designs
will therefore promote laminar flow during the positive pressure gradient and then design the transition to turbulent flow to occur prior to the negative pressure gradients. This is an optimisation for the lowest skin and form drag while eliciting high Lift coefficients.[24]

The boundary layer can separate two ways, from the trailing and leading edge. When the separation occurs at the trailing edge it is described as soft stall. The stalled region will get larger and approach the leading edge with further increases of the angle of attack. When separation begins from the leading edge the foil is said to be in ‘Hard Stall’. This effect is more common in thin foil sections and results in a dramatic drop of Lift and increase in drag. Further effects such as dynamic stall can transiently when the angle of attack is changed.[24]

Having established that the 2D aerodynamics changes over the span let us now translate these concepts to the 3D case. 2D aerodynamics demonstrated that Lift is generated when there is low pressure on top of the foil and high pressure on the bottom. Now consider the tip of the blade or the wing of an aeroplane, at this point the two pressure differences meet and create flow around the tip and then further down the span. This effect can also be found in the hub and tip regions of a wind turbine blade and is termed cross flow (figure 8.2).

Figure 8.3: Cross Flow Contours. Reproduced with Permission [45]
In Prandtl’s Lifting theorem it is described that vortices are generated where there is a change in lift with span [24]. Therefore, vortices will develop all along the wing with the strongest vortices at the tip of the blades. These vortices are shed down stream as a sheet which then wraps up into the tip vortices down stream. The 3D vortex shedding pattern results in the angle of attack being altered slightly as well as a drop in the performance of a blade. For certain flow regimes it is not always necessary to fully calculate the 3D effects/tip losses. It is also possible to rely on empirical relations such as Prandtl’s Tip Loss correction[24]. Regardless of method, tip loses should be understood and accounted for.