Effect of structural elasticity on simulation of wind turbines and its verification

OpenFOAM based CFD simulations of HAWT coupled with FEM analysis based on spinning element

Kishan Gajjar

Delft University of Technology

Effect of structural elasticity on simulation of wind turbines and its verification

OpenFOAM based CFD simulations of HAWT coupled with FEM analysis based on spinning element

by

Kishan Gajjar

Student Name	Student Number
Kishan Gajjar	5471982

To obtain the degree of Master of Science in Sustainable Energy Technology at the Delft University of Technology,

Supervisor:	Prof. ir. Axelle Viré , Dr.ir. Delphine De Tavernier
Project Duration:	December 2022 - September 2023
Faculty:	Faculty of Aerospace Engineering, Delft

Acknowledgement

My incredible two-year journey at TU Delft has come to a close upon the completion of this thesis. During this time, I have acquired a wealth of knowledge and developed numerous skills that will enable me to make meaningful contributions to the ongoing energy transition. I am forever indebted to those who have played a pivotal role in shaping the person I have become.

To begin with, I want to express my gratitude to my two proficient mentors, Axelle and Delphine, who consistently endeavoured to maintain my motivation and extract the utmost potential from me. A special acknowledgement goes to Delphine for consistently highlighting areas for improvement in my sometimes mundane weekly meeting slides and reports. I want to extend my heartfelt gratitude to my Phoenixstraat family in Delft - Devansh, Janki, Shloka, Ritik, Gaurav, and Shanti, I appreciate you all for making my weekends enjoyable throughout my two years of master's studies.

Last but certainly not least, my gratitude to my parents knows no bounds. Their lifelong encouragement and unwavering emotional support have been the cornerstones of my journey, and I will carry their influence with me for the rest of my life.

-Kishan Gajjar
Rotterdam, Sep 2023

Abstract

As wind turbines get bigger and bigger, the simulation of wind turbines becomes more complex. The increase in size brings about a multitude of intricate challenges that must be addressed in the simulation process. These challenges lie in the different aspects of the simulations such as aerodynamics, structural dynamics, power electronics, hydrodynamics, turbine control etc.

The central theme of this thesis is to explore the effect of one of the aspects- structural dynamics, on the wind turbine simulations. Moreover, the thesis also emphasises the development of a structural module and it's integration in an OpenFOAM-based wind turbine simulation library called *TurbinesFoam*. *TurbinesFoam* is an actuator line method-based simulation tool, which enables the study of turbine performance as well as wake dynamics using Computational Fluid Dynamics. The key motivation is also to contribute towards the accurate simulation of wind turbines by performing a successful integration as said above.

To fulfil the goals, a structural module was developed in Matlab to simulate the structural dynamics of wind turbines. The module is developed from the theory of spinning elements assuming the blade and tower as Euler-Bernoulli beam. The developed module code is called while running the CFD simulations in OpenFOAM to add deflection and rotation of the blades and towers, using a MatLab pipe class. The accuracy of the developed code was measured using BModes and OpenFAST. The results exhibited satisfactory agreement between the outputs.

The NREL 5MW turbine was used to study the effects. The results revealed that, at slightly above the rated condition, the turbine showed 1% decrease in power production due to elasticity when compared with a rigid turbine. The thrust and torque coefficients showed similar trends of reduction in value. Moreover, as the inflow wind velocity increased, the differences in performance got broader, due to increased deflection value. The wake region of the elastic turbine showed a mixed region of both increased and decreased wind velocity when compared with the rigid turbine.

Contents

List of Figures

List of Tables

List of symbols and acronyms

$HAWT$	Horizontal Axis Wind Turbine
ALM	Actuator Line Method
FEM	Finite Element Method
$FOWT$	Floating Offshore Wind Turbine
AE	Actuator Element
CFD	Computational Fluid Dynamics
$BEMT$	Blade Elemental Momentum Theory
$RANS$	Reynolds's Averaged Navier Stokes
LES	Large Eddy Simulation
Re	Reynolds number
DES	Detached Eddy Simulation
$[K]$	stiffness matrix
k	Elemental stiffness in a direction
T, R	Transformation matrix
$[d]$	displacement matrix
$[C]$	Damping matrix
$[M]$	Mass matrix
TSR	Tip Speed Ratio
$[\square_e], [\square_l]$	Elemental, local matrix
$[\square_g]$	Global matrix
CFL number	Courant-Friedrichs-Lewy number
OOP	Out-Of-Plane
IP	In-Plane
$[G_{\Omega_e}]$	Gyroscopic damping matrix
$[C_s]$	Structural damping matrix
$[K_{\Omega_e}]$	Spinning stiffness matrix
$[K_{C_e}]$	Centrifugal stiffness matrix
$[K_{g_e}]$	Gravitational stiffness matrix
$EALM$	Elastic Actuator Line Method
$RALM$	Rigid Actuator Line Method

1
Introduction

Wind turbines have become an increasingly important source of renewable energy worldwide, with installed wind capacity reaching 743 GW in 2020 (IRENA, 2021). As wind turbines continue to grow in size and capacity, the need for accurate and efficient simulations becomes increasingly critical to ensure the safety and reliability of these structures. The simulation of wind turbines involves the consideration of multiple physical phenomena, including aerodynamics, hydrodynamics, and structural mechanics.

One of the key challenges in wind turbine simulation is the accurate modelling of structural elasticity, which can significantly affect the behaviour and performance of wind turbines under different loading conditions. Wind turbines are subject to a range of dynamic loads, including wind and wave-induced loads, internal loads, as well as gravity loads. These loads can cause complex and nonlinear structural responses, including deflections, vibrations, and stress concentrations.

The accurate modelling of structural elasticity is crucial for predicting the response of wind turbines to these loads and ensuring their safety and reliability. However, the use of fully detailed finite element models can be computationally expensive and time-consuming, limiting their practical application. Researchers have developed simplified structural models, such as beam models, that can incorporate the effects of structural elasticity while still being computationally efficient.

Beam models are one of the most commonly used simplified models for wind turbine simulation. These models represent the wind turbine structure as a series of interconnected beams, allowing for the consideration of structural flexibility and deformation under different loading conditions. Beam models have been shown to accurately predict the structural response of wind turbines under various loading scenarios.(Griffith, 2009)

In recent years, researchers have also explored the use of reduced-order models for wind turbine simulation. These models involve the use of simplified mathematical representations of the wind turbine structure and can provide significant computational savings compared to fully detailed finite element models. Reduced-order models have shown promise in accurately predicting the structural response of wind turbines under different loading conditions. (Naunidh, 2019)

The simulation of wind turbines becomes more complex for floating offshore wind turbines (FOWT). The floater of the turbine is subjected to a huge variety of motions in all 6 degrees of freedom (DOF). Those motions are transferred to the wind turbine and their effect, in combination with the turbine elasticity, can be seen in the variety of turbine performance criteria, from turbine power output, turbine loads to wake region and also the performance of adjacent turbines.(Ma et al., 2020)

1.1. Motivation and research goals

In this thesis, we will investigate the effect of structural elasticity on wind turbine simulations and verify the accuracy of a simplified structural beam model. The motivation behind this research is to improve the accuracy and efficiency of wind turbine simulations and ultimately contribute to the development of a more effective and accurate open-source simulation tool (Pericas, 2022). The ultimate research goal of this thesis is as follows

"How does structural elasticity affect the ALM-based wind turbine simulation? "

To further simplify, the main research question is divided into the following sub-questions.

- How can structural dynamics of wind turbines be modelled using a simple yet accurate manner?
- How to verify the developed structural model?
- How to strongly couple the ALM-based wind turbine simulation with the dynamic structural simulations?
- What is the effect of structural elasticity on the simulation?

1.2. Intended methodology

To satisfy the main goal of this thesis, it is required to first have an accurate structural dynamic model for the wind turbine. After a literature review, a code will be developed in MATLAB for structural simulation. The ultimate goal is to replicate MATLAB code in OpenFOAM's own C++ environment. At first, MATLAB's ease of use and availability of a variety of mathematical operations will make the development of the structural module much more effortless. The developed model will then be validated against the widely used model in the industry for structural simulation

After a validated code is developed, that code can further be linked with the OpenFOAM with the use of a MATLAB pipe class. The MATLAB pipe class can effectively communicate with OpenFOAM to add structural deformation in the actuator elements.

The following figure 1.1 shows the already existing simulation flow (other than purple) and probable space for the inclusion of the structural dynamic module (in purple).

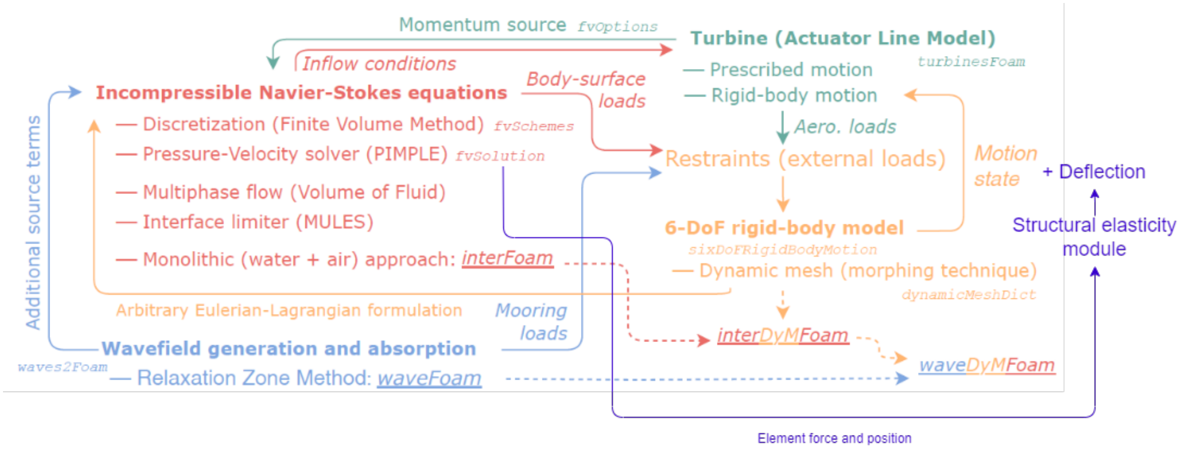

Momentum source fvOptions
Turbine (Actuator Line Model)
Inflow conditions
turbinesFoam
— Prescribed motion
— Rigid-body motion
Body-surface loads
Incompressible Navier-Stokes equations
— Discretization (Finite Volume Method) fvSchemes
Aero. loads
— Pressure-Velocity solver (PIMPLE) fvSolution
Restraints (external loads)
Motion state
+ Deflection
— Multiphase flow (Volume of Fluid)
— Interface limiter (MULES)
6-DoF rigid-body model
Structural elasticity module
sixDoFRigidBodyMotion
— Monolithic (water + air) approach: interFoam
— Dynamic mesh (morphing technique)
dynamicMeshDict
Additional source terms
Arbitrary Eulerian-Lagrangian formulation
Mooring loads
interDyMFoam
Wavefield generation and absorption
waves2Foam
— Relaxation Zone Method: waveFoam
waveDyMFoam
Element force and position

Figure 1.1: Strategy for CFD simulation of FOWT (Pericas, 2022) updated with structural deflection

2 Literature review

This section provides an extensive overview of the fundamental concepts employed in the simulation of wind turbines and structural modelling of wind turbines. It serves as the foundation for the subsequent development of the simulation models. The literature review examines the simulation strategy used in current work for wind turbines, the different models to simulate the structural elasticity of wind turbines, an evaluation of various Finite Element Method (FEM) models, followed by the effect of aeroelasticity on simulations. The selected model is then used to establish a framework for simulating the structural behaviour of the combined blade and tower model in the next chapter.

2.1. Simulation of wind turbines

Various techniques are used for simulating wind turbines. These include the widely used Blade Element Momentum Theory (BEMT), which divides the rotor blades into sections and calculates aerodynamic forces based on local wind conditions. Simplified models such as Actuator Disc/Line/Point models approximate the rotor as a disc, line, or point to estimate the flow field. Computational Fluid Dynamics (CFD) offers high-fidelity simulations by solving the Navier-Stokes equations. The following literature focuses more on the techniques for simulation utilized in the thesis.

2.1.1. Aerodynamics of wind turbine

There are multiple ways to explain the aerodynamic phenomenon related to wind turbines. But on a higher level, a wind turbine can be described as an energy-consuming actuator disc. The incoming wind passes through the disc, and a portion of the kinetic energy of the incoming wind is absorbed by the disc. As described in Burton et al., 2021, the wind going in and coming out of the disc creates a stream tube. The mass flow rate through the tube is assumed to be constant. Because the mass flow rate is constant, the cross-section of this tube increases as wind velocity drops.

The fall in velocity is quantified by the induction factor a, such that at the disk and at far wake, the wind velocity is given by

$$U_D = (1-a)U_\infty, \; U_W = (1-2a)U_\infty \tag{2.1}$$

Where, U_D, U_W and U_∞ are the wind velocities at the turbine disc, at far wake and of free-stream respectively.

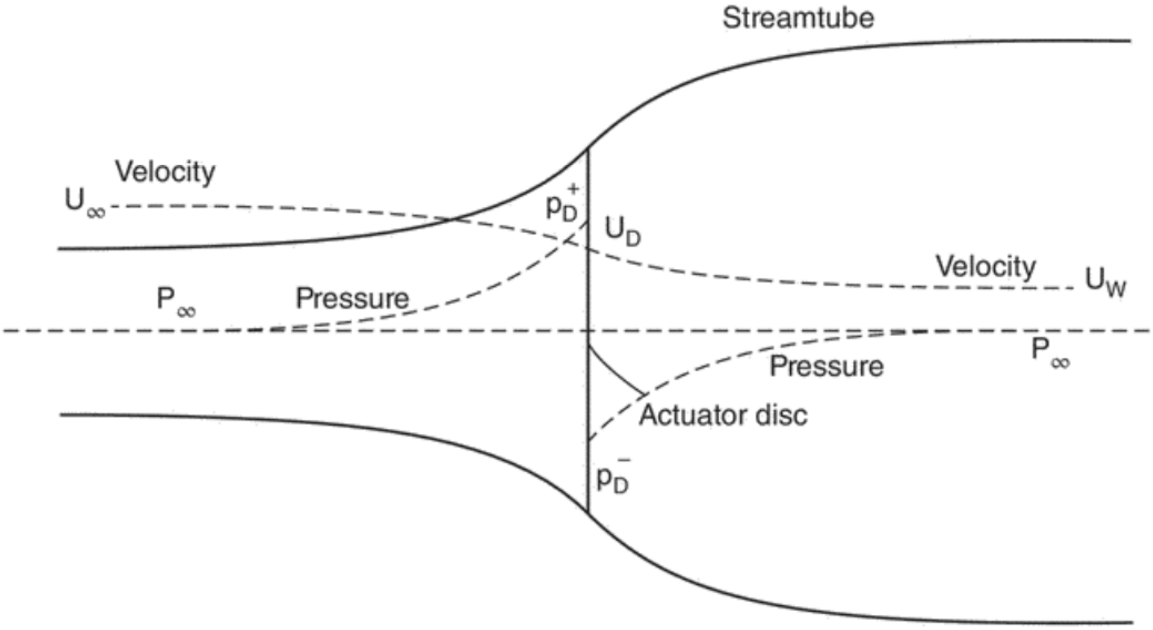


Figure 2.1: Actuator disc, extracting energy from incoming wind (Burton et al., 2021)

By using the difference in momentum of wind before and after the disc (using wind speed), the energy extracted by the disc can be calculated. Finally, the turbine performance can be obtained from energy as,

$$C_T = \frac{\text{Thrust}}{0.5\rho {U_\infty}^2 A_D} = 4a(1-a), C_P = \frac{\text{Power}}{0.5\rho {U_\infty}^2 A_D} = 4a(1-a)^2 \tag{2.2}$$

The maximum value of C_p that can be achieved at induction factor $a = \frac{1}{3}$ is 16/27. Which is called the Betz limit. This is due to the fact that if all energy is extracted from flow, the flow rate would become zero. Hence there will always be a portion of energy that cannot be extracted.

2.1.2. Actuator line method

The actuator line method, initially presented by Sørensen and Shen, 2002, is a comprehensive and time-dependent aerodynamic model used to examine the airflow around wind turbines. This method integrates a three-dimensional Navier-Stokes solver with an actuator line technique, where body forces are radially distributed along lines that represent the turbine blades. The basis of the model is the incompressible Navier-Stokes equations:

$$\frac{\partial V}{\partial t} + V \cdot \nabla V = \frac{1}{\rho}\nabla p + \nu \nabla^2 V + f, \;\; \nabla \cdot V = 0$$

where f denotes the body forces, which represent the loading on the rotating blades. The loading forces are calculated using the blade element theory. The following figure 2.2 illustrates a wind turbine blade divided into multiple actuator elements.

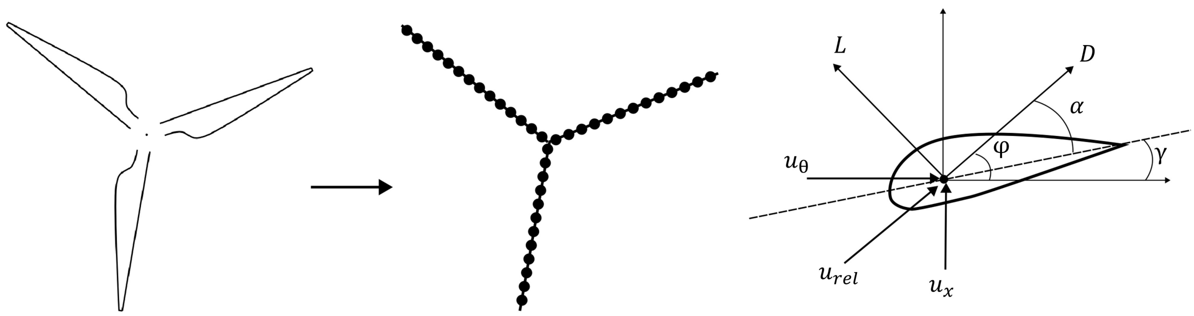


Figure 2.2: Wind turbine blades divided in multiple actuator elements with an airfoil crossection

Where each element has its own aerodynamic properties such as drag and lift coefficients. In ALM, all elements are represented at the 1/4 chord location. At each time step, the force per span length on each element is calculated from the following equations.

$$f_\theta = \frac{\rho c u_{\mathrm{rel}}^2}{2}\left(C_l \sin\varphi - C_d \cos\varphi\right), \quad f_x = \frac{\rho c u_{\mathrm{rel}}^2}{2}\left(C_l \cos\varphi + C_d \sin\varphi\right) \tag{2.3}$$

Where c is the local chord, φ is the angle between relative velocity u_{rel} and rotor plane. The negative value of these forces is added to the body forces, as the blade exerts the same force on the surrounding field in the opposite direction.

To prevent singular behaviour, it is necessary to evenly distribute the applied aerodynamic blade forces across multiple mesh points. In practice, these forces are distributed in a three-dimensional Gaussian manner along and away from the actuator lines. This distribution is achieved by convolving the computed local load, f, with a regularization kernel, η, also known as the "smearing" function as depicted below.

$$\eta_\epsilon(d) = \frac{1}{\epsilon^2 \pi^{3/2}} exp\left[-\left(\frac{d}{\epsilon}\right)^2\right] \tag{2.4}$$

Where the constant ϵ is force projection width. The bigger the constant, the bigger will be the force application volume. The optimal value of the ϵ plays a crucial role in achieving accurate and stable results in the flow field analysis.

Various studies have been conducted to determine this value. Troldborg, 2009 suggested setting ϵ to twice the CFD grid resolution ($\triangle$) as a good compromise between accuracy and avoiding oscillations.Jha et al., 2014 introduced a blended dissipative scheme with $\epsilon = 2(\triangle)$ to prevent oscillations and improve loading distributions. They also proposed a method to vary the ϵ along the blade span for better results around the tip region. Shives and Crawford, 2013 investigated the effect of the ϵ and CFD grid resolution for a fixed wing, recommending an ϵ around 1/4 of the chord to satisfy the vortex line approach. Martínez-Tossas et al., 2017 developed a theoretical model and concluded that for larger grid resolutions, the smoothing-kernel scale is determined by the mesh, while for finer discretizations, an ϵ of 1/5 of the chord is optimal, aligning with the guidelines from Shives and Crawford, 2013

TurbinesFOAM library developed by Bachant et al., 2018, is an implementation of the actuator line method in OpenFOAM. Apart from the above-discussed parameters, the library also uses various other models to most accurately mimic the flow around the turbine.

The library uses the dynamic stall model developed by Sheng et al., 2008 to simulate the stall condition in unsteady flow, which is originally derived from Leishman–Beddoes (LB) semi-empirical modelLeishman and Beddoes, 1989. An added mass model from Strickland et al., 1980 is utilized, to account for the extra inertial forces on fluid due to the moving actuator elements. Prandtl's lifting line theory is utilized for tip-end correction. Lastly, Flow curvature correction is required due to varying chordwise angle of attack, due to the circular shape of the crossflow turbine blade.

2.1.3. Wake and turbulence modelling

In wind turbine simulations, wakes are divided into near and far wakes. The near wake, extending from the turbine to approximately one or two rotor diameters downstream, exhibits distinct tip vortices generated by the turbine geometry. These vortices create sharp velocity gradients and peaks in turbulence intensity. In contrast, the far wake experiences reduced axial velocity and increased turbulence intensity, indirectly influenced by the turbine's actual rotor shape (Vermeer et al., 2003).

Turbulence in the far wake stems from atmospheric, mechanical, and wake sources, acting as an efficient mixer that aids in the recovery of velocity deficit and reduction in overall turbulence intensity. Over time, the velocity deficit becomes approximately axisymmetric, and self-similar in the far downstream region (Vermeer et al., 2003). Wake meandering, the large-scale movement of the entire wake may further decrease the velocity deficit but can increase fatigue and extreme loads on downstream turbines (G. C. Larsen et al., 2008,G. Larsen et al., 2007)

Various numerical models are used for wind turbine wake aerodynamics. Analytical methods assume the self-similar nature of the far wake. BEMT utilizes a global momentum balance and 2D blade elements. Vortex-lattice and particle methods describe inviscid flow with concentrated vorticity. Panel methods consider blade geometry and incorporate viscous effects, while the wake is modelled similarly to vortex-wake methods. However, these analytical models cannot predict the wake turbulence as accurately as numerically solving the Navier-stokes equation. All these wake models are discussed in Vermeer et al., 2003, Crespo et al., 1999, Snel, 1998. Detailed discussions are out of the scope of this thesis.

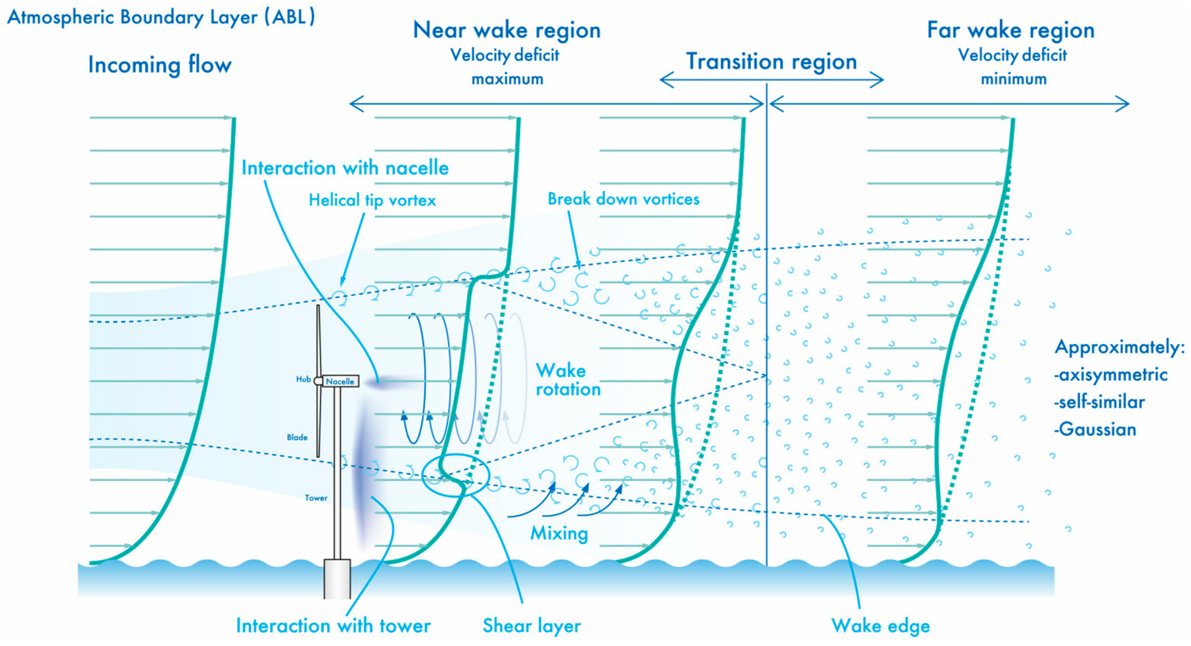


Figure 2.3: Different wake region and its characteristics Uchida, 2020

For the simulation of turbulence in CFD, two major methods are utilized. Reynolds's Averaged Navier Stokes method (RANS) and Large Eddy Simulation (LES).

RANS methods seek to provide a statistical representation of the fluid flow. The flow properties, such as velocity and pressure, are separated into an average component and a fluctuation component known as the Reynolds decomposition:

$$u(x,t) = \bar{u}(x) + u'(x,t) \tag{2.5}$$

Using this notation of flow properties, it is assumed that the average of the fluctuation term is zero(i.e. $u'(\bar{x},t) = 0$). This notation is then put into Navier Stokes (NS) equation and an average of the whole NS equation is taken. Upon simplification, the NS equation becomes:

$$\frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} = -\frac{1}{\rho}\nabla\bar{p} + \nu\nabla^2\bar{u} - \nabla \cdot (\overline{u'u'}) \tag{2.6}$$

Where the term $\overline{u'u'}$ is known as Reynolds stress tensor. Many models exist which aim to approximate this term and close the equation(Wilcox, 2006). But the two-equation models $k-\epsilon$ and $k-\omega$ are widely used. Where k is the turbulence kinetic energy, ϵ is the energy dissipation rate and ω is the specific energy dissipation rate.

In **LES**, the large eddies of the flow are directly computed, while the smaller eddies that are smaller than the grid size are modelled using a sub-grid scale model. To achieve this scale separation, the velocity field is spatially filtered, dividing it into a resolved (also known as large-scale, simulated, or filtered) velocity component and an unresolved (small-scale) component. This filtering operation is generally defined as a convolution integral.

Unlike RANS, which has a computational cost that is only slightly influenced by the Reynolds number (Re), the computational cost of LES increases approximately with Re squared Alam and Fitzpatrick, 2017. This means that LES becomes significantly more expensive, particularly near solid boundaries where boundary layers exist. This is because LES requires refinement in all three spatial directions, whereas RANS only requires refinement in the direction perpendicular to the wall.

To mitigate this computational burden, a hybrid approach called Detached Eddy Simulation (DES) can be employed. In this approach, RANS is used to accurately resolve the attached boundary layers, while LES is applied outside the wall region. By combining these two methods, the computational cost can be reduced while still capturing the important flow characteristics. This hybrid approach has been explored and documented in Spalart, 1999.

2.2. Structural dynamics of wind turbines

Different aeroelastic models are widely employed to accurately capture the complex behaviour of wind turbine components under various loading conditions. These models have been developed and utilized to simulate the structural response of wind turbines, considering factors such as material properties, geometric complexity, and dynamic interactions. The models offer diverse approaches for representing the structural behaviour of wind turbine blades, towers, and other components. In this context, this section provides an overview of the various models used for the structural simulation of wind turbines, highlighting their advantages, limitations, and applications.

The earliest work of dynamic wind turbine modelling can be traced back to Friedmann, 1976, proposing an equation of motion for a single blade, conceived as an elastic beam with a fixed root at the hub and a free tip. Utilizing Hamilton's principle and the Newtonian method, Hodges and Dowell, 1974 further developed these equations, allowing for moderate displacements in long, slender, straight, homogeneous, and isotropic beams. These equations incorporate various nonlinear structural and inertial terms, which significantly influence the aeroelastic stability and response of hingeless helicopter rotor blades.

Extending this initial work, Friedmann, 1976 formulated a new set of partial differential equations of motion for wind turbine blades rotating in a gravity field with variable rotor speed and pitch action. Hansen et al., 2006 implemented the principle of virtual work with modal shape function and nonlinear beam theory with FEM to devise the dynamic structural model of a wind turbine. However, the escalating flexibility and length of wind turbines challenge the capacity of classical beam theory to model the structural dynamic, necessitating a more nonlinear beam theory with fewer assumptions.

The selection of the beam theory (linear or nonlinear) and elements (shell or beam elements) is complemented by three frequently adopted discretization methods in relation to wind turbines, namely, modal reduction approach, multibody dynamics

(MBD), and Finite Elements Method (FEM).

In the **FEM approach**, the wind turbine system is divided into finite elements such as flexural beams, lumped masses, springs, and joints. This method's flexibility in considering various configurations, including geometric and material nonlinearity, results in a high degree of freedom but with correspondingly high computational demands and costs.

The **modal approach** superimposes the deflection of components like blades, towers, and support structures from a linear combination of physically realistic models, usually the lower eigenmodes such as 1st and 2nd flapwise and edgewise modes. While computationally efficient, this method suffers from several limitations, including a fixed number and type of degree of freedom, assumption of linearity, and inability to handle certain types of structures.

The **MBD** approach approximates the structure using a finite number of elements composed of rigid and flexible bodies linked by elastic joints. This discretized system is described by a finite number of ordinary differential equations, combining the advantages of the previous methods. It requires fewer equations of motion, considers nonlinearity, handles nonlinear kinematics more efficiently than FEM, and permits the modelling of mechanical systems with both large deflection and rotation.

Most of the latest structural models now used for aeroelastic simulations of wind turbines can be categorized into two categories: beam model and shell model.

The beam model represents the structural elements of a wind turbine, such as blades and tower, as one-dimensional beams. This simplified representation assumes that the cross-sections of the beam elements remain constant along their length. The beam model is computationally efficient, making it suitable for analyzing large-scale wind turbine structures. It allows for the examination of bending, torsion, and axial deformations, as well as the effects of loads and boundary conditions. However, it may not accurately capture the complex three-dimensional behaviour and local effects within the components.

On the other hand, the shell model represents wind turbine components as thin, two-dimensional surfaces, which better accounts for the geometrical complexities and localized phenomena. It considers the varying thickness and curvature of the components, enabling a more detailed analysis of their structural behaviour. The shell model is particularly useful for capturing the effects of local buckling, stress concentrations, and complex load distributions. However, the shell model typically requires a higher computational cost compared to the beam model due to its increased complexity.

In their paper, Kang et al., 2014 compared the geometrically exact shell and beam model for the accuracy of the dynamic behaviour of the rotating blade. Both models gave identical results for the natural frequencies of a straight blade with a slenderness ratio of less than 12.5. However, for a higher slenderness ratio and a high tip sweep angle greater than 30, there were significant differences in the results. For most modern wind turbines, the slenderness ratio is very high and the sweep angle is almost zero.

Kee and Shin, 2015 described a more accurate FEM model to simulate a rotating blade. Their model also includes Green-Lagrange non-linear stiffness term to include the non-linearity. The model includes shear deformation along the axial direction and thus is also accurate for very large deformation. Special importance is given to non-linearity as well. Reflecting on non-linearity, Ma et al., 2019 compared the result from their linear model with non-linear model. From figure 2.4, it can be said that in-plane deflections are less sensitive to non-linearity. While out-of-plane deflection has a lower value in the non-linear model as compared to the linear one.

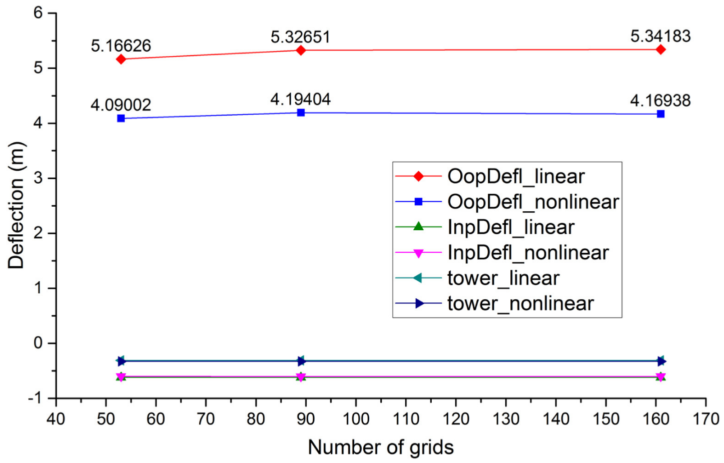


Figure 2.4: Value of tip out of plane, tip in-plane and tower deflection in model with different grid size

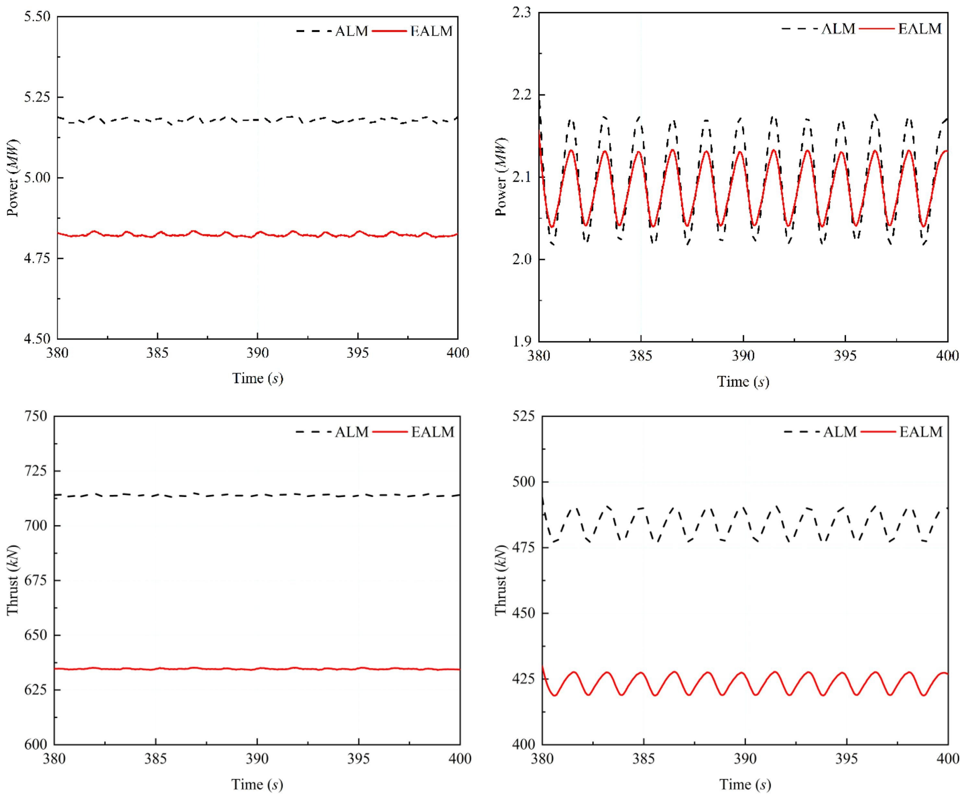
In the industry, ALM-based wind turbine and wind farm simulations are popular because of the low computation cost. FAST, and HAWC2 are widely used software for wind turbine and blade simulations. All of this software uses beam models for the structural dynamics module. BeamDyn (integrated in FAST) uses state-of-the-art Geometrically exact Beam Theory to simulate the blade response along with Wiener–Milenkovic rotation parameter to simulate a rotating beam (Wang et al., 2017). HAWC2 uses the multi-body approach to divide the structure into multiple small bodies. Connected with each other, these bodies efficiently capture large deformation despite having a linear model(Ahmed A., 2005).

2.3. Effect of structural elasticity on wind turbine simulation

The structural deformation can have an impact on different parameters such as rotor thrust, power, near and far wake wind velocity, turbulence etc. The paper by Muscari et al., 2023 used the coupling of SOWFA and OpenFAST to examine the importance of structural elasticity in the simulation. They used a linear structural solver- ElastoDyn to include elasticity in the SOWFA. Their research concluded that tip deflection of up to 4% of the blade span had a negligible effect on the wake performance. But a higher deflection of around 12% indeed showed a significant decrease in thrust, power and the wake wind velocity.

Similar results were found by Ahlström, 2006 as well. Ahlström, 2006 also mentioned that linear structural models used nowadays are accurate enough to simulate the turbines in normal conditions and for certification purposes. But for light and more flexible blades, non-linearity must be included to calculate large deflection in blades.

Research done by Zheng et al., 2023 showed that rotor deflection can cause a significant drop in turbine power and thrust. The authors compared two turbine configurations (One turbine in the wake of another) with the ALM and EALM (Elastic ALM) models side to side. The following figure shows the difference in torque and thrust at the hub due to the elasticity of the turbine.



(a)Upstream wind turbine

(b)Downstream wind turbine

Figure 2.5: Comparision of ALM and EALM turbines' thrust and power with one turbine standing in wake of another Zheng et al., 2023

A difference in velocity field was also apparent in both the rotor plane and the wake region. The overall increased velocity deficit in the wake region means that the wake region characteristics like wake expansion, and wake recovery were also affected by it.

Ma et al., 2019 investigated the aeroelastic coupled wake behaviour of the NREL 5MW turbine using the same EALM method. They also included the effect of nonlinearity in their paper. The authors concluded that an elastic turbine absorbs more energy from the air than a rigid one. The vorticity increases when going towards the far wake region. A major difference in velocity profile is seen at 5D-8D distance [Fig. 2.6], with the elastic turbine having more velocity deficit than the rigid one. This reduction in velocity can greatly affect the turbines in a wind farm configuration. The figure below [2.6] compares the velocity profiles for different kinds of non-linearity as given in Ma et al., 2019 .

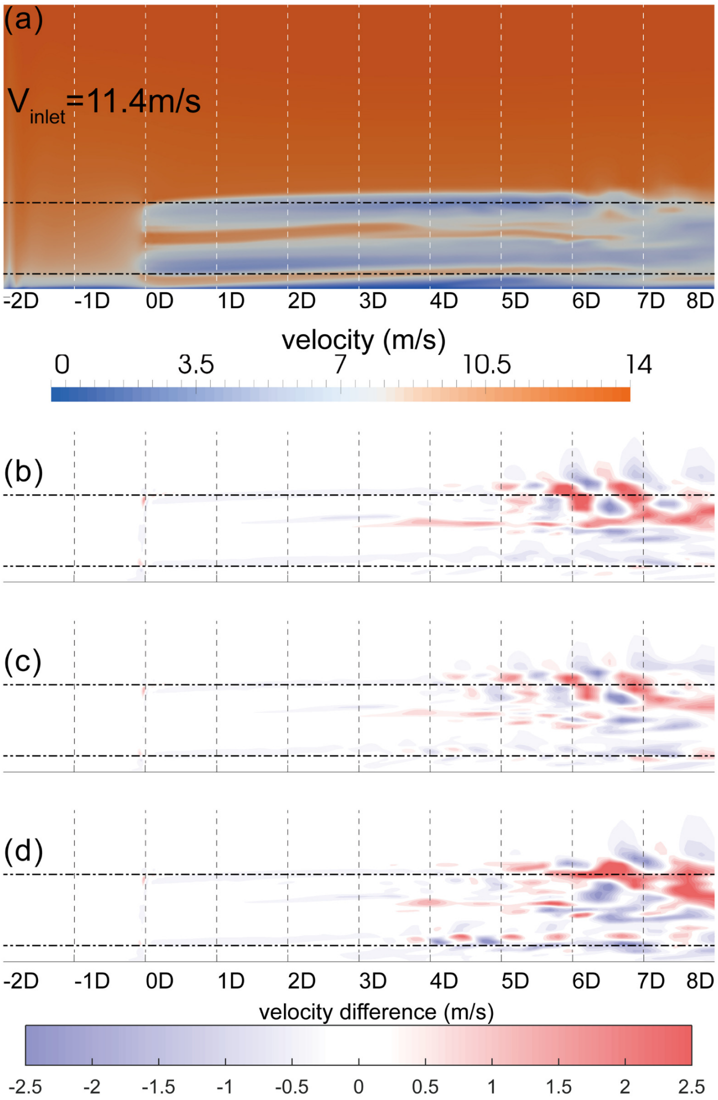


Figure 2.6: Differences in wake velocity profile for (b) spin (c) stress (d) non-linear when turbine compared with (a) rigid turbine Ma et al., 2019

While some research concluded an overall decrease in turbine performance, the investigation done by Sayed et al., 2019 showed a slight improvement of around 1% and 0.3% in rotor power and thrust respectively. Instead of using the ALM method, the authors investigated the case using high fidelity method of rotating mesh. The major reason for increased turbine performance was due to the increased axial forces on the blades which increased rotor torque due to deflection. The wake region was not included in this study.

Similarly, Liu et al., 2019 used blades resolved mesh and fully coupled CFD-MBD method to simulate NREL 5 MW wind turbine. Concluding that the flexible turbine outputs slightly more power than the rigid one. Negligible differences in turbine thrust were found as well. Thus blade resolved CFD - aeroelastic simulations tend to show almost no effect on turbine performance apart from obvious phase differences in loads on turbine blades.

3 Methodology

In this chapter, an overview of the building blocks of structural dynamics is presented. After which the actual FEM model to simulate the rotating wind turbine is gone through. This model is based on the blade FEM model as shown in Sajeer et al., 2020. Here, the same approach is used to derive the governing mass and stiffness matrices for a single element of the blade. The elemental matrices are then organized into global matrices to be finally used for the simulations.

3.1. The finite element method

3.1.1. Stiffness matrix

In the FEM, the stiffness matrix captures the stiffness properties of each element and their interconnections, enabling the calculation of displacements and internal forces under different loading conditions. A stiffness matrix can be independently formed for a beam and a truss, and later be assembled (Logan, 2000).

Axial deformation of 1D member
Consider a 1D spring element as shown below.

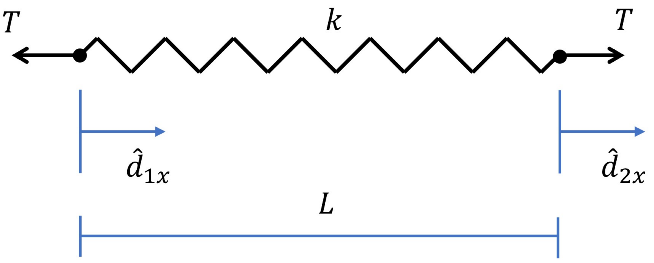


Figure 3.1: A 1-D axial spring element

The local x-axis is directed from node 1 to node 2. The spring is represented by labelling nodes at each end and by labelling the element number. The original distance between nodes before deformation is denoted by L. The material property (spring constant) of the element is k.

In advance, the mathematical function to represent the deformed shape of the spring element under loading must be chosen. Since the spring element resists axial loading only with the local degrees of freedom for the element being displacements $\hat{d}_{1x}$ and $\hat{d}_{2x}$ along the $\hat{x}$ direction, a displacement function $\hat{u}$is chosen to represent the axial displacement throughout the element. Here, a linear displacement variation along the $\hat{x}$ axis of the spring is assumed because a linear function with specified endpoints has a unique path (Logan, 2000).
Therefore,

$$\hat{u} = a_1 + a_2 x \tag{3.1}$$

In general, the total number of coefficients a is equal to the total number of degrees of freedom associated with the element. This relation can further be expressed in terms of a matrix.

$$\hat{u} = \begin{Bmatrix} 1 & x \end{Bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} \tag{3.2}$$

Now that the nodal displacements $\hat{d}_{1x}$ and $\hat{d}_{2x}$ are know. This gives,

$$\begin{aligned} \hat{u}(0) &= \hat{d}_{1x} = a_1 \\ \hat{u}(L) &= \hat{d}_{2x} = a_2 L + \hat{d}_{1x} \end{aligned} \tag{3.3}$$

Solving equation 3.3 for $a2$ and then substituting the value of a_2 in the equation 3.1 gives ,

$$\begin{aligned} a_2 &= \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L} \\ \hat{u} &= \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L} x + \hat{d}_{1x} \end{aligned} \tag{3.4}$$

And in matrix form, it can be written as,

$$\hat{u} = \begin{Bmatrix} N_1 & N_2 \end{Bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \tag{3.5}$$

Here

$$N_1 = 1 - \frac{\hat{x}}{L} \quad and \quad N_2 = \frac{\hat{x}}{L}$$

are shape functions. They represent the shape of the assumed displacement function across the domain of the element, specifically at the $\hat{x}$ coordinate, with the ith element degree of freedom having a value of one and all other degrees of freedom being zero.

Now, the tension in the spring can be written as

$$T = k(\hat{d}_{2x} - \hat{d}_{1x}) \tag{3.6}$$

The next step is to calculate the stiffness matrix for the spring element. Using the nodal force and equilibrium sign convention, the stiffness matrix can be obtained by the following equation.

$$\hat{f}_{1x} = -T; \;\; \hat{f}_{2x} = T$$

Hence

$$\begin{aligned} \hat{f}_{1x} &= -k(\hat{d}_{2x} - \hat{d}_{1x}) \\ \hat{f}_{2x} &= k(\hat{d}_{2x} - \hat{d}_{1x}) \end{aligned} \tag{3.7}$$

And in the matrix form, it can be described as,

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \tag{3.8}$$

Here,

$$\hat{k} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \tag{3.9}$$

is called the local stiffness matrix for the element.

Bending of a 2D beam

For the deformation of the beam, the minimum potential energy method can be used. After all, ultimately this method will be used to derive mass, stiffness and damping matrices for the wind turbine. According to Chandrupatla and Belegundu, 2011 the Lagrangian of a system can be given by;

$$L = K - \Pi \tag{3.10}$$

Where K is the kinetic energy and Π is the potential energy of the element. The Lagrangian in equation 3.10 will be discussed later. In the absence of external work, the potential energy, also called the strain energy, of a system can be written by the following equation (Leung and Fung, 1988)

$$\Pi = \int \int_v \int \frac{1}{2} \sigma_x \epsilon_x dV \tag{3.11}$$

Where σ_x and ϵ_x are the axial stress and strain at the given cross-section. The axial strain of the element due to traverse deflection can be given by,

$$\epsilon_x = -\hat{y} \frac{d^2 \hat{v}}{dx^2} = -\hat{y} \, [B] \, \{\hat{d}\} \tag{3.12}$$

Here, $\hat{y}$ is the distance in the traverse direction, and $\hat{v}$ is deflection in the same direction. $\hat{d}$ represents the four degrees of freedom (In matrix form, $\hat{d} = \left[\hat{d_{1y}}\ \hat{\phi_1}\ \hat{d_{2y}}\ \hat{\phi_2}\right]$). $[B]$ is the double derivative of shape function with respect to x, as shown in equation 3.5. But here, the shape function uses the Hermite shape function as shown in table 3.2 ($[B] = \frac{d^2[N_2(x)\ N_5(x)\ N_8(x)\ N_{11}(x)]}{dx^2}$).For bending, the stress-strain relationship can be given by

$$\sigma_x = [E]\,\epsilon_x \tag{3.13}$$

Where $[E]$ is the elasticity modulus. From equations equation 3.11 and equation 3.12, the following notation for the potential energy can be obtained in matrix form.

$$\Pi = \int_x \int_A \int \frac{1}{2}\{\sigma_x\}^T\{\epsilon_x\}dAdx \;=\; \int_0^L \frac{EI}{2}\{\hat{d}\}^T\,[B]^T\,[B]\,\{\hat{d}\}d\hat{x} \tag{3.14}$$

Where $I = \int\int_A y^2 dA$ is the area moment of inertia of the cross-section. And finally taking the derivative of the above equation with respect to $\{\hat{d}\}$ and equating it to zero gives the stiffness matrix of a 2D beam.

$$[K] = EI\int_0^L \{\hat{d}\}^T\,[B]^T\,[B]\,\{\hat{d}\}d\hat{x} \tag{3.15}$$

3.1.2. Mass matrix

In finite element analysis (FEA), the mass matrix is a fundamental component used to model the dynamic behaviour of structures. It represents the distribution of mass within the structure and is an essential part of the equations of motion in dynamic analysis.

The mass matrix is typically derived by integrating the mass density over the volume of each finite element. In FEA, the structure is discretized into a mesh of interconnected elements, and the mass matrix is assembled by summing the contributions of each element.

The mass matrix is often denoted as [M] and is a square matrix, with dimensions equal to the number of degrees of freedom in the system. Each element in the matrix represents the mass associated with a particular degree of freedom.

The lumped mass matrix, as the name suggests, approximates the mass distribution by lumping all the mass at the element nodes or at specific integration points. It assumes that the mass is concentrated at discrete locations within the element. The lumped mass matrix is simpler to compute and requires less memory compared to the consistent mass matrix. However, it does not capture the coupling effects between different degrees of freedom as accurately as the consistent mass matrix. In general form, the mass matrix for an element with two nodes and 6 DOF for each node can be given by,

$$[M] = diag\left[m_{x1}\ m_{y1}\ m_{z1}\ I_{x1}\ I_{y1}\ I_{z1}\ m_{x2}\ m_{y2}\ m_{z2}\ I_{x2}\ I_{y2}\ I_{z2}\right] \tag{3.16}$$

Where m and I are the concentrated mass and rotational inertia at the particular node and in particular directions.

The consistent mass matrix, also known as the consistent or fully populated mass matrix, is derived by considering the mass distribution within each element. It accounts for the coupling between different degrees of freedom and provides more accurate results for dynamic analysis. The assembly of the consistent mass matrix involves integrating the mass density over the element volume and considering the shape functions that describe the element's displacement behaviour. This matrix is typically larger and more computationally expensive to compute and store.

According to Bathe, 2006 and Chandrupatla and Belegundu, 2011, the kinetic energy of the element can be given by,

$$K_e = \frac{1}{2}\int_V \dot{u}^T \dot{u} \rho dV \tag{3.17}$$

where $\dot{u}$ is the velocity of the element. From equation 3.5, the velocity of the element , $\dot{u} = N\dot{d}_e$. Substituting that with equation 3.17 gives,

$$K_e = \frac{1}{2}\dot{d_e}^T \left[\int_V \left[N^T N \rho dV \right] \right] \dot{d_e} \tag{3.18}$$

Here, subscript e represents elemental matrices. Comparing it with a general kinetic energy equation ($\frac{1}{2}mv^2$), the elemental consistent mass matrix can be given by,

$$[M]_e = \int_V \rho N^T N dV \tag{3.19}$$

3.1.3. Damping matrix

The damping matrix introduces the necessary damping characteristics in the structural model to simulate the energy dissipation phenomena. Damping enables the simulation of the dynamic behaviour of the structure. For wind turbines, there can be multiple sources of damping such as aerodynamic damping, structural damping, hydrodynamic damping (for FOWTs), gyroscopic damping, damping by soil etc. Chen and Duffour, 2018 compared the effect of different damping sources on the turbine tower and concluded that for a wind turbine, aerodynamic damping has the greatest contribution. However, for this thesis, we will only focus on structural damping and gyroscopic damping.

Structural damping

The structural damping matrix can be calculated using Rayleigh's method of proportional damping, where the damping is proportional to the mass and stiffness of the element.

$$[C] = \alpha[M] + \beta[K] \tag{3.20}$$

where α and β are mass and stiffness proportionality coefficients. The values of these coefficients are determined from experiments. The relationship between α, β and natural frequency can be given by Strømmen, 2013

$$\zeta_n = \left(\frac{\alpha}{\omega_n} + \beta\omega_n\right)/2 \tag{3.21}$$

where ζ_n is the damping ratio of nth mode, which can be determined from experiments. ω_n is the nth natural frequency. If two damping ratios for two natural frequencies are known, equation 3.21 can be used to determine the coefficients α and β. But as it is hard to get by these values, the structural damping ratio is assumed in literature most of the time. As mentioned in Strømmen, 2013 if the damping ratio of the first mode is known, subsequently, the coefficients α and β can be derived using the following equations.

$$\alpha = \omega_1\zeta_1 \; and \; \beta = \frac{\zeta_1}{\omega_1} \tag{3.22}$$

The structural damping ratio of a blade lies anywhere between 0.5% to 5%. G. Bir and Jonkman, 2007 used damping ratios of 0.5% and 2.5% for their analysis (for NREL 5MW). According to Jonkman et al., 2009, the structural damping ratio for the first mode of the blade is determined to be 0.477465%, while the damping ratio for the tower is 1%.

3.1.4. Transformation matrix

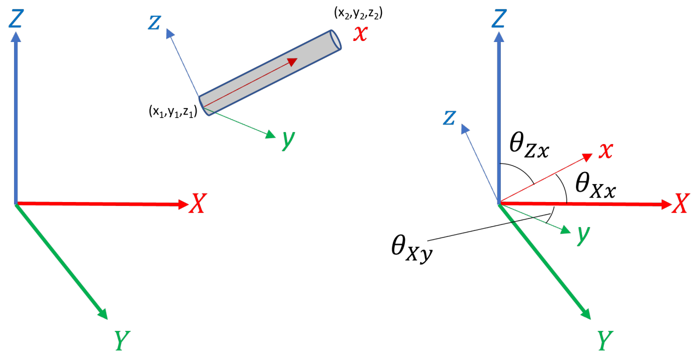


Figure 3.2: Representation of an element's local coordinate system in a global frame of reference

The transformation matrix converts the equations and variables from the local coordinate system to the global coordinate system, and vice versa. It enables the representation of the physical behaviour of elements in the global coordinate systems, simplifying the analysis and computations.

The transformation matrix to convert the global frame of reference to the local frame of reference can be given by,

$$T = \begin{bmatrix} cos\theta_{Xx} & cos\theta_{Yx} & cos\theta_{Zx} \\ cos\theta_{Xy} & cos\theta_{Yy} & cos\theta_{Zy} \\ cos\theta_{Xz} & cos\theta_{Yz} & cos\theta_{Zz} \end{bmatrix} \tag{3.23}$$

Where θ_{Ij} represents the angle between the axis I of the global frame and axis j of the local frame. The transformation matrix can be used to transform or offset structural properties of cross-section as well. An example to transform the local stiffness matrix to the global stiffness matrix is given below.

We know that

$$[F_l] = [K_l][d_l]\ ,\ [d_l] = T[d_g]\ and\ [F]_l = T[F_g] \Rightarrow T[F_g] = [K_l]T[d_g]$$
$$\Rightarrow [F_g] = T^{-1}[K_l]T[d_g] \tag{3.24}$$

Where $[F_l], [d_l], [K_l]$ are force, displacement, and stiffness matrix in the local frame of reference and $[F_g], [d_g]$ are force and displacement matrix in the global frame of reference. For an element with 2 nodes and 6 DOF for each node, the 12X12 transformation matrix can be given by,

$$T_{12X12} = diag\,[T\ T\ T\ T] \tag{3.25}$$

Transformation of cross-sectional properties

In this thesis, the forces are calculated on the blade axis which does not pass through the structural neutral axis of the cross-section. The offset of different sectional properties needs to be taken into account while calculating force on an axis other than the neutral axis as mentioned in *BeamDyn inputs from sectional beam properties*, 2019.

For offset in the angle of the neutral axis, the equation equation 3.23 can be used. To offset the distance between the axis, the following equation can be used.

$$T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

Where x,y and z are the offset distances. The composition of the above matrix will change based on the properties being calculated.

For the simulation of the NREL 5 MW turbine, the structural twist changes the orientation of the shear centre axis and tension/elasticity centre axis (Due to the linear model, it is assumed that both centres coincide).To compensate for the twist, the sectional stiffness properties need to be transformed. For COG, a translation is required because of COG offset, as mentioned in Jonkman et al., 2009. The following figure from G. S. Bir, 2007 illustrates the different locations of the shear centre, COG, elasticity centre and its orientation with the blade reference axis. However, the offset is only shown in an edgewise direction. The effect of flapwise offset of structural properties is negligible.

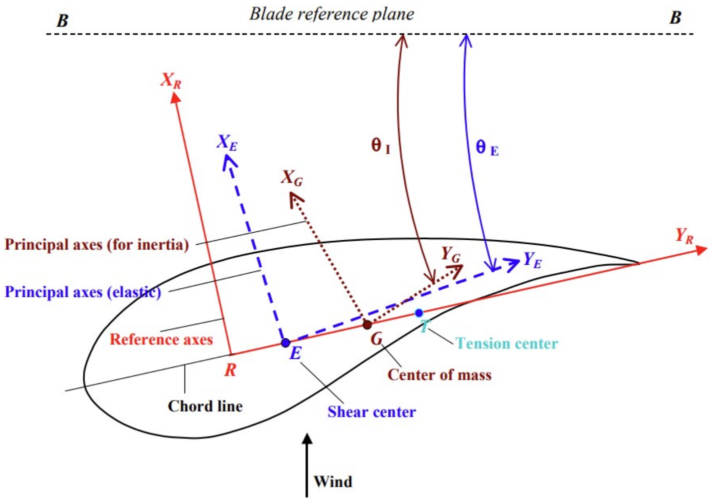


Figure 3.3: Orientation of sectional properties with respect to reference axis

3.1.5. Equation of motion

The Lagrange method offers a comprehensive approach for determining the equations of motion for any system, utilizing its kinetic and potential energy. Unlike Newton's method Morin, 2007, which involves constructing free-body diagrams for each body and requires knowledge of forces and their directions, the Lagrange method circumvents this need. This characteristic makes it especially well-suited for intricate systems with numerous degrees of freedom and for addressing constraints. For a given system, the Lagrangian L can be defined in relation to its kinetic energy (K) and potential energy (Π)

$$L = K - \Pi \tag{3.27}$$

The Lagrange-Euler equation of motion is,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial (L)}{\partial q_i} = 0 \tag{3.28}$$

3.1.6. Solving the dynamic equation

Once the mass, damping and stiffness matrices are defined, the equation of motion can be solved to get the dynamic response of the beam in the time domain. There are mainly two methods to calculate the dynamic response of a system as mentioned in Bathe, 2006. Direct integration over time and modal analysis.

The modal analysis uses eigenfrequencies and eigenvector/mode shapes to describe the dynamic response of the system by the law of superposition. Where the direct integration method manipulates matrices and uses direct matrix calculations to obtain the time series of a response. Multiple schemes are available under the direct integration method to solve the equation of motion such as the central difference method, Houbolt method, Wilson θ method etc. A detailed study of all methods including the stability analysis, which is out of the scope of this thesis can be found described in Bathe, 2006. For now, we will focus on one of the direct integration methods- the Newmark method.

The Newmark method

In the Newmark method, it is assumed that the acceleration of an element varies linearly with time ($i.e.{}^{d^2\hat{d}}/_{dt^2} = constant$). Thus, the following assumptions are made.

$$
\begin{aligned}
\dot{d}_{t+\Delta t} &= \dot{d}_t + \left[(1-\delta)\ddot{d}_t + \delta\ddot{d}_{t+\Delta t}\right]\Delta t \\
d_{t+\Delta t} &= d_t + \dot{d}_t\Delta t + \left[(\frac{1}{2}-\alpha)\ddot{d}_t + \alpha\ddot{d}_{t+\Delta t}\right]\Delta t^2
\end{aligned}
\tag{3.29}
$$

This method is unconditionally stable (i.e. stable even with bigger time steps, having less error and amplification) for $\delta \geq 0.5$ and $\alpha \geq 0.25(0.5+\delta)^2$. Hence this scheme is suitable for the purpose of this thesis. As we do not need to worry about convergence while developing a code. The following steps describe the whole method in order, to solve the equation of motion.

1. Calculate and assemble mass matrix M, damping matrix C and stiffness matrix K
2. Set initial condition, $d_0, \dot{d}_0, \ddot{d}_0$. The value of $\ddot{d}_0$ can be calculated from the below equation if not given.
$$\ddot{d}_0 = M^{-1}(F_0 - Kd_0)$$
3. Calculate effective stiffness matrix
$$\bar{K} = K + a_0M + a_1C$$
4. Calculate effective load at time $t + \Delta t$
$$\bar{F}_{t+\Delta t} = f_{t+\Delta t} + M(a_0d_t + a_2\dot{d}_t + a_3\ddot{d}_t) + C(a_1d_t + a_4\dot{d}_t + a_5\ddot{d}_t)$$
5. solve d at time $t + \Delta t$
$$\bar{K}d_{t+\Delta t} = \bar{F}_{t+\Delta t}$$
6. Calculate velocity and acceleration at time $t + \Delta t$
$$
\begin{aligned}
\ddot{d}_{t+\Delta t} &= a_0(d_{t+\Delta t} - d_t) - a_2\dot{d}_t - a_3\ddot{d}_t \\
\dot{d}_{t+\Delta t} &= \dot{d}_t + a_6\ddot{d}_t + a_7\ddot{d}_{t+\Delta t}
\end{aligned}
$$
7. Follow steps 4 to 6 for each time step.

The value of $a_0, a_1, ...a_8$ can be found in the following table.

Table 3.1: Integration constants for Newmarks method

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
$\frac{1}{\alpha\Delta t^2}$	$\frac{\delta}{\alpha\Delta t}$	$\frac{1}{\alpha\Delta t}$	$\frac{1}{2\alpha}-1$	$\frac{\Delta t}{2}(\frac{\delta}{\alpha}-2)$	$\frac{\delta}{\alpha}-1$	$\Delta t(1-\delta)$	$\delta\Delta t$

3.2. The spinning element method applied in wind turbine

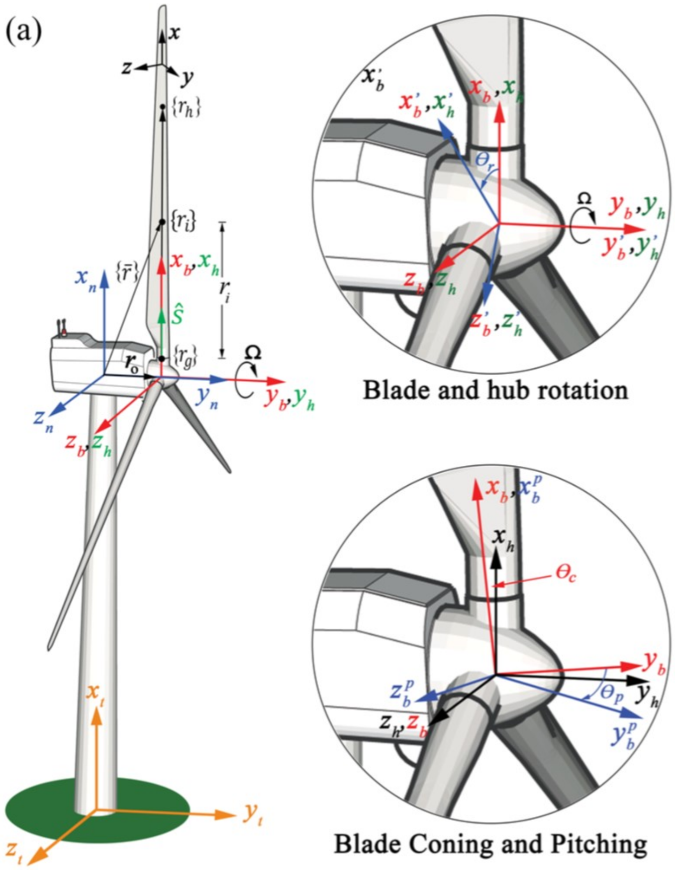


Figure 3.4: Schematic diagram of turbine co-ordinate system

Let us consider a point located on the blade as shown in 3.4 in the non-rotating reference frame $x_n y_n z_n$. Its position in the frame can be given by,

$$\{\bar{r}\} = \{r_o\} + \{r_i\} \tag{3.30}$$

$$\{r_i\} = \{r_g\} + r_i\hat{s} \tag{3.31}$$

$$\hat{s} = \frac{\{r_h\} - \{r_g\}}{|\{r_h\} - \{r_g\}|} \tag{3.32}$$

The blade's position can be expressed as a function of the position vector $\{r_i\}$ with respect to the moving co-ordinate system, the unit vector $\hat{s}$ along the blade pitch axis, and the distance r_o between the origins of the fixed and moving co-ordinates. In the case of a three-bladed turbine system where the origin of the fixed co-ordinate coincides with the origin of the rotating co-ordinate system (i.e., $\{r_o\}$=0), this expression can be used to determine the location of the blade.

$$\{\bar{r}\} = \{r_g\} + r_i\hat{s} \tag{3.33}$$

The inertial frame $x_n y_n z_n$, also known as the nacelle coordinate, provides a complete description of the rotating hub and blade system. When considering a rotating beam or blade, its torsional stiffness remains the same as that of a non-spinning beam. It is assumed that torsional deformation does not affect the longitudinal (u) and bending (v,w) deformations independently. Considering the blade as Euler-Bernoulli's beam, the two rotational deformations, other than the axial rotational deformation, are obtained from differential of the displacement in transverse direction i.e. $\theta_y = \partial_x w$ and $\theta_z = \partial_x v$. Thus the independent variables of deformation are $\mathbf{u}(x,t) = \{u, v, w, \theta_x\}$

However, the gyroscopic effect resulting from the blade's rotation around the y_n axis at a constant angular speed leads to the coupling of different rotational degrees of freedom perpendicular to the spinning axis.As mentioned in Leung and Fung, 1988, a matrix can be introduced so that these coupling between the perpendicular axis can be put into equations. The matrix also known as the spinning matrix can be described as Ω;

$$\Omega = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \tag{3.34}$$

From this spinning matrix, the velocity of any point at distance $\{r\}$ from the root can be described in the moving coordinates system $x_h y_h z_h$. The absolute velocity of the point will be,

$$\{V\} = \{\dot{r}\} + [\Omega]\{r\} \tag{3.35}$$

In the fixed frame $x_n y_n z_n$, the position of the deformed point can be given by,

$$\{r\} = \{r_i\} + \{\bar{d}\} = \{r_i\} + [R]^T\{d\} \tag{3.36}$$

Where, $\{d\}$ is deformation matrix in the local xyz co-ordinates. $\{\bar{d}\}$ is the deformation matrix in reference frame $x_n y_n z_n$. And $[R]$ is the transformation matrix between local xyz coordinates and reference coordinates $x_n y_n z_n$, also described in equation 3.23.Hence,

$$\{\dot{r}\} = [R]^T\{\dot{d}\} \quad (\{\dot{r}_i\} = 0) \tag{3.37}$$

From equation 3.35 and equation 3.36,

$$\{V\} = [R]^T\{\dot{d}\} + [\Omega]\left[\{r_i\} + [R]^T\{d\}\right] \tag{3.38}$$

3.2.1. Shape function

For the FEM model, the blade is divided into elements with two nodes, having 6 DOF for each node. Thus, a single element has 12 DOF freedom. As described in Eq. 3.5, the deformation can be described as a function of DOF with the help of the shape function,

$$\{d_e\} = \left[\phi\right]\{q_e\} \tag{3.39}$$

Where $\{q_e\}$ is a 12x1 matrix, representing the 12 DOF as shown in 3.5.

$$\{q_e\} = \{q_1\ q_2\ q_3\ .....\ q_{12}\}$$

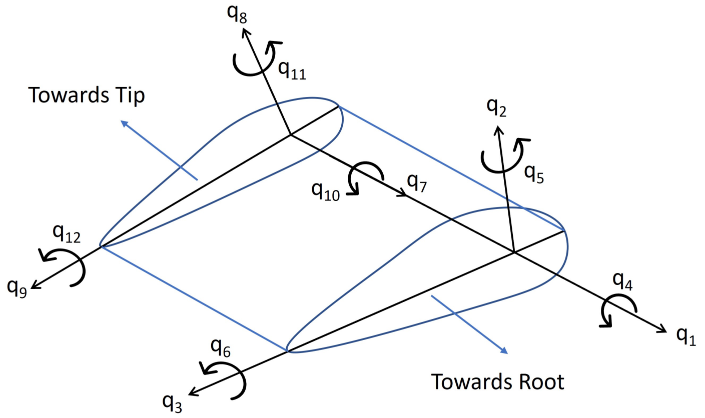


Figure 3.5: Degree of freedom of a blade element

The shape function can be described as,

$$\phi = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_7 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_2 & 0 & 0 & 0 & N_6 & 0 & N_8 & 0 & 0 & 0 & N_{12} \\ 0 & 0 & N_3 & 0 & N_5 & 0 & 0 & 0 & N_9 & 0 & N_{11} & 0 \\ 0 & 0 & 0 & N_4 & 0 & 0 & 0 & 0 & 0 & N_{10} & 0 & 0 \end{bmatrix} \tag{3.40}$$

In the above shape function, the first three rows represent the shape function for three directional degrees of freedom H such that deformation vector $\{d\} = \{u, v, w\} = [H]\{q_e\}$. The last row represents the shape function (θ_x) for torsional deformation such that $(\{d_{\theta_x}\} = [\theta_x]\{q_e\})$. Linear shape functions are utilized to model axial and torsional deformation in these formulations, whereas Hermite shape functions are employed to model bending deformation (Bathe, 2006). The table below represents the different shape functions as described in 3.40.

Table 3.2: Shape functions for blade element

Type	Node 1	Node 2
Axial displacement	$N_1(x) = 1 - (\frac{x}{l_e})$	$N_7(x) = (\frac{x}{l_e})$
Flapwise displacement	$N_2(x) = 1 - 3(\frac{x}{l_e})^2 + 2(\frac{x}{l_e})^3$	$N_8(x) = 3(\frac{x}{l_e})^2 - 2(\frac{x}{l_e})^3$
Flapwise rotation	$N_5(x) = x(1 - \frac{x}{l_e})^2$	$N_11(x) = x((\frac{x}{l_e})^2 - \frac{x}{l_e})$
Edgewise displacement	$\mathrm{N}_3(x) = N_2(x)$	$\mathrm{N}_9(x) = N_8(x)$
Edgewise rotation	$\mathrm{N}_6(x) = -N_5(x)$	$\mathrm{N}_{12}(x) = -N_{11}(x)$
Torsional deformation	$\mathrm{N}_4(x) = N_1(x)$	$\mathrm{N}_{10}(x) = N_7(x)$

3.2.2. Governing equation of motion

To arrive at the equation of motion, Lagrange's equation as mentioned in Bathe, 2006 and also Leung and Fung, 1988 can be used.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right)-\frac{\partial (L)}{\partial q_i} = f_i, \quad i = 1,2,3,....n \tag{3.41}$$

Here L is called $Lagrangian$; $L = K - \Pi$. The total energy can be found by summing up the energies of individual blade elements. The elemental kinetic can be given by,

$$K = \frac{1}{2}\int \rho A(x)\{v\}^T\{v\} + \frac{1}{2}\int \rho A(x){R_g}^2\{\dot{\theta}_x\}^T\{\dot{\theta}_x\} \tag{3.42}$$

Since the torsional deformation is decoupled from other deformation, the kinetic energy due to torsional deformation is evaluated separately.Now, using equation 3.38, 3.39 and 3.42

$$K = \frac{1}{2}\int_0^L \rho A(x)\left\{[R]^T\{\dot{u}\} + [\Omega]\left[\{r_i\} + [R]^T\{u\}\right]\right\}^T\left\{[R]^T\{\dot{u}\} + [\Omega]\left[\{r_i\} + [R]^T\{u\}\right]\right\}$$
$$= \frac{1}{2}\Big(\{\dot{q}_e\}^T H^T[R][R]^T H\{\dot{q}_e\} + \{\dot{q}_e\}^T H^T[R][\Omega]^T[\Omega][R]^T H\{\dot{q}_e\} + \{r_i\}^T[\Omega]^T[\Omega]\{r_i\}+$$
$$2\{\dot{q}_e\}^T H^T[R][\Omega]\{r_i\} + 2\{\dot{q}_e\}^T H^T[R][\Omega][R]^T H\{q_e\} + 2\{r_i\}^T[\Omega]^T[\Omega][R]^T H\{\dot{q}_e\}\Big)$$
$$+\frac{1}{2}\int_0^L \rho A{R_g}^2\{\dot{q}_e\}^T\{N_\theta\}^T\{N_\theta\}\{\dot{q}_e\}dx$$

As $[R]$ is an orthogonal matrix, $[R][R]^T = I$ (identity matrix); Thus simplifying gives,

$$K = \frac{1}{2}\{\dot{q}_e\}^T[M_e]\{\dot{q}_e\}+\frac{1}{2}\{q_e\}^T[K_{\Omega_e}]\{q_e\}+T_{ie}+\{\dot{q}_e\}^T\{f_e\}+\{\dot{q}_e\}^T[G_{\Omega_e}]\{q_e\}+\{F_{\Omega_e}\}^T\{q_e\} \tag{3.43}$$

Similarly, the potential energy of the blade element can be given by,

$$\Pi = \frac{1}{2}\int_0^L\Bigg[EA(x)\left(\frac{du}{dx}\right)^2 + EI_z(x)\left(\frac{d^2v}{dx^2}\right)^2 + EI_y(x)\left(\frac{d^2w}{dx^2}\right)^2 +$$
$$GJ(x)\left(\frac{d\theta}{dx}\right)^2 + Fc(x)\left(\frac{dv}{dx}\right)^2 + Fc(x)\left(\frac{dw}{dx}\right)^2 + Fg(x)\left(\frac{dv}{dx}\right)^2 + Fg(x)\left(\frac{dw}{dx}\right)^2\Bigg] \tag{3.44}$$

Here, $EA(x), EI_y(x), EI_z(x)$ and $GJ(x)$ are varying rigidity concerning x in the axial, flapwise, edgewise, and torsional direction respectively. $F_c(x)$ and $F_g(x)$ are centrifugal and gravitational force acting at the location x of the element. The equation of which can be found in Appendix A.

Now Using equation 3.44 and 3.39, the potential energy of the element can be written as,

$$\begin{split}\Pi = \frac{1}{2}\int\limits_0^L \Big[& EA\{q_e\}^T N_u'^T N_u'\{q_e\} + EI_x\{q_e\}^T N''_v{}^T N_v''\{q_e\} + EI_y\{q_e\}^T N''_w{}^T N''_w\{q_e\} + \\ & GJ\{q_e\}^T N_\theta'^T N_\theta'\{q_e\} + F_c(x)\{q_e\}^T N_v'^T N_v'\{q_e\} + F_c(x)\{q_e\}^T N_w'^T N_w'\{q_e\} + \\ & F_g(x)\{q_e\}^T N_v'^T N_v'\{q_e\} + F_g(x)\{q_e\}^T N_w'^T N_w'\{q_e\} \Big] dx \end{split} \tag{3.45}$$

Where, $N_u, N_v, N_w,\ and\ N_\theta$ are the first, second, third and fourth rows of the shape function 3.40. N' and N'' represent the derivative and double derivative of the shape function with respect to x respectively. After simplifying the above equation, we have

$$\Pi = \frac{1}{2} q_e\{[K_{Ee}] + [K_{Ce}] + \left[K_{ge}\right]\} q_e \tag{3.46}$$

Finally, putting equation 3.46 and 3.44 in the equation 3.42, and simplifying it gives us the equation of motion.

$$[M(t)]\{\ddot{q}\} + [C(t)]\{\dot{q}\} + [K(t)]\{q\} = \{f(t)\} \tag{3.47}$$

where $[M]$ and $[C]$ are the global mass and damping matrices, respectively. The damping matrix [C] is the sum of the structural damping matrix $[C_s]$ and gyroscopic damping matrix $[G_\Omega]$]. The stiffness matrix $[K(t)] = [K_E] + [K_c] + \left[K_g\right] - [K_\Omega]$. The definition of which is given in Appendix A.

3.2.3. Assemblage of matrices

While the elemental matrices derived in the previous sections are for blade elements, they can be adapted to obtain the corresponding tower matrices by assuming a stationary tower (i.e., zero rotational speed, $\Omega = 0$).However, the reference frame of the blade is a rotating frame (i.e. the matrices derived in the above chapter are with respect to the rotating blade coordinate system as mentioned in 3.6). While that of a tower needs to be a stationary one. To address this, a global stationary frame needs to be defined, which serves as a reference for both the tower and the blades. Since the rotor is rotating and changing its position at each time step of the simulation, all three blades need to be transformed to align with this global stationary frame.

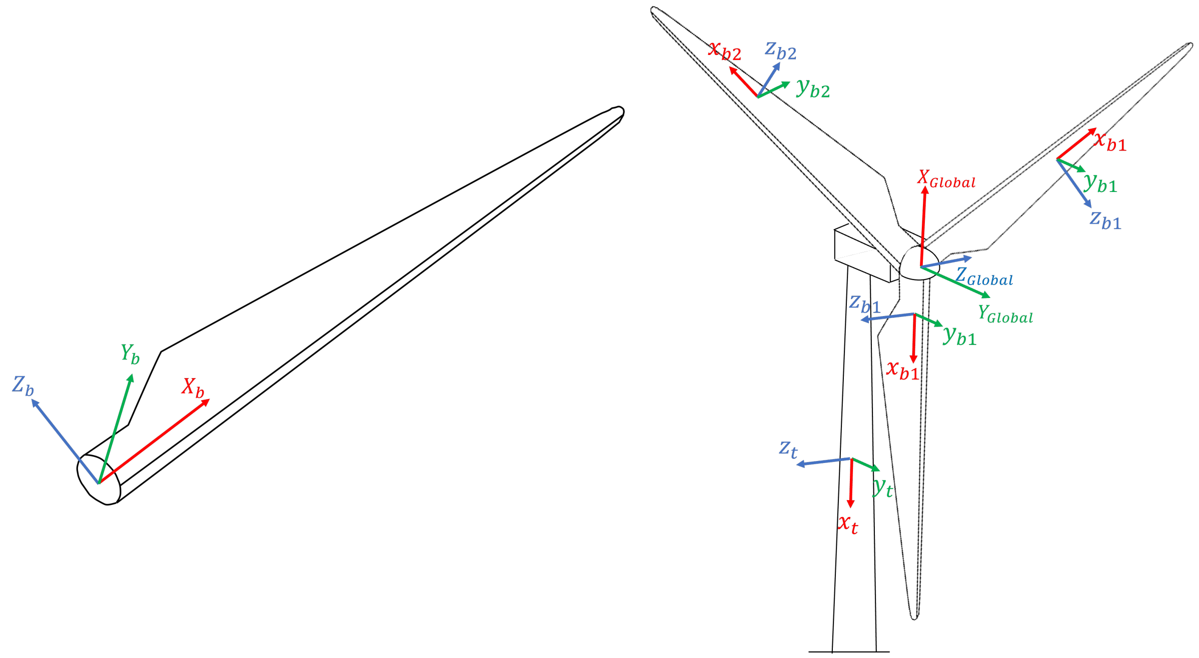


Figure 3.6: Representation of local blade frame and global frame in FEM

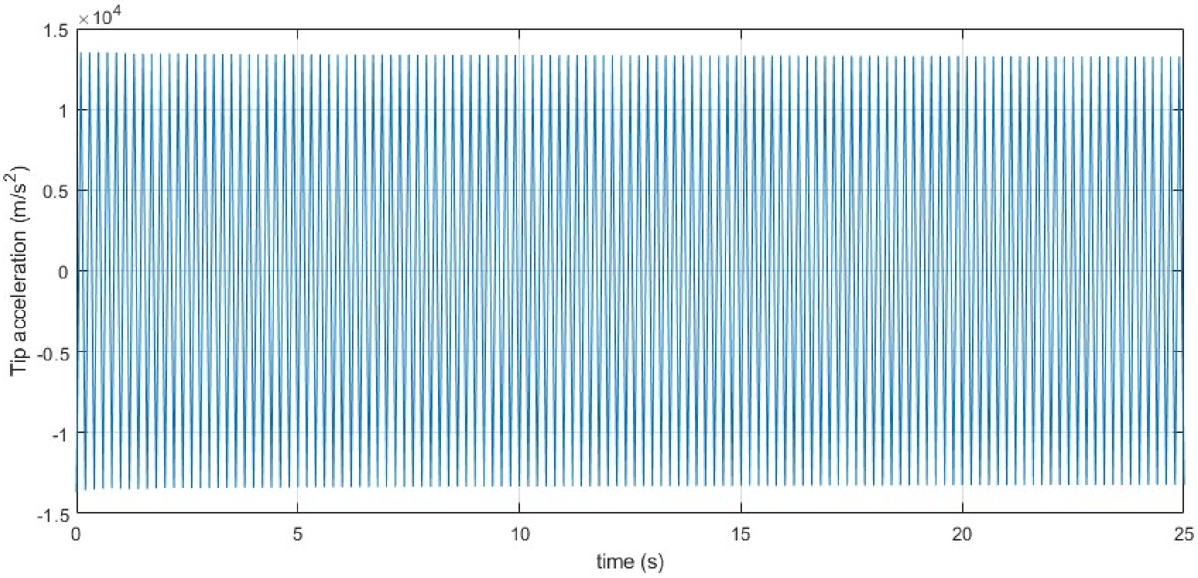
The transformation involves applying the coordinate transformation for each element, in the same manner, as described in 3.23 to convert the blade and tower element matrices from the rotating blade frame to the global stationary frame. At each time step the blade properties of, all three blades, need to be transformed by the angle which is equal to the rotor speed times the time-step. If the floating motion is included, those motions will also need to be taken into account separately. More about that is discussed later.

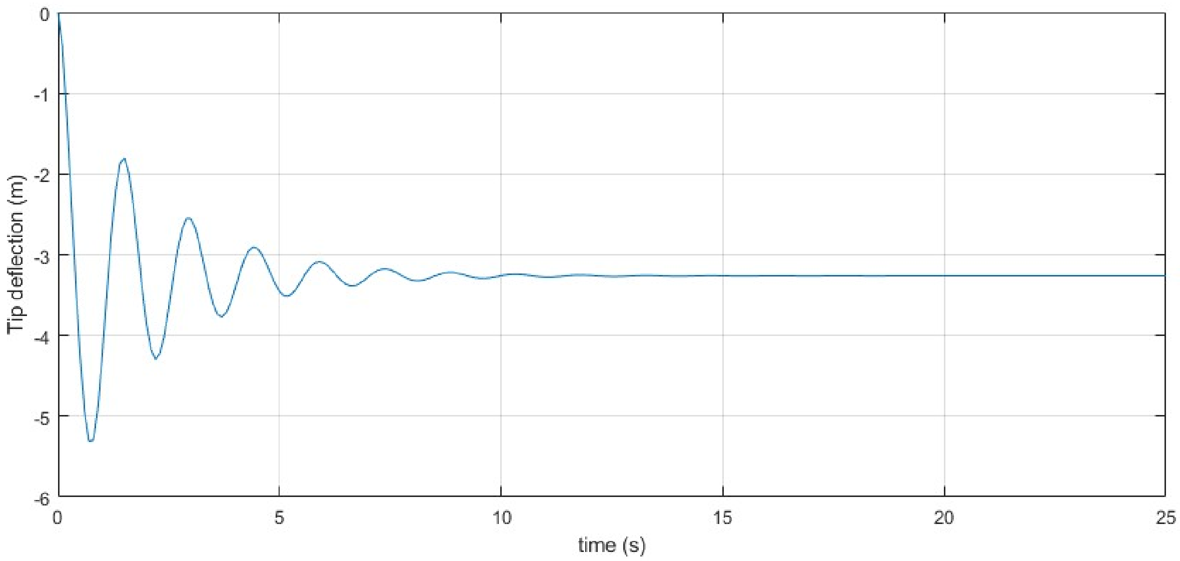
After the matrices are transformed into global coordinates, all three blades and the tower can be assembled to obtain the global matrices to be used in the equation 3.46. The composition of global matrices is given below.

$$\begin{bmatrix} Blade1.element1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Blade1.element2 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & Blade2.element1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & Blade3.element1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Tower.elementN \end{bmatrix}$$

3.2.4. Selecting the Newmark method's coefficients

As mentioned in 3.1.6, the Newmark method produces unconditionally stable solutions for $\delta \geq 0.5$ and $\alpha \geq 0.25(0.5+\delta)^2$. While this condition is only true for the displacement matrix, the velocity and acceleration matrix does not converge and show oscillatory behaviour for $\delta = 0.5$ and $\alpha = 0.25$. For the simulation in OpenFOAM, the convergence of velocity is as important as the displacement because the element's speed is also updated with the speed due to deflection. Hence, higher values for both coefficients are selected to further proceed with the simulation.





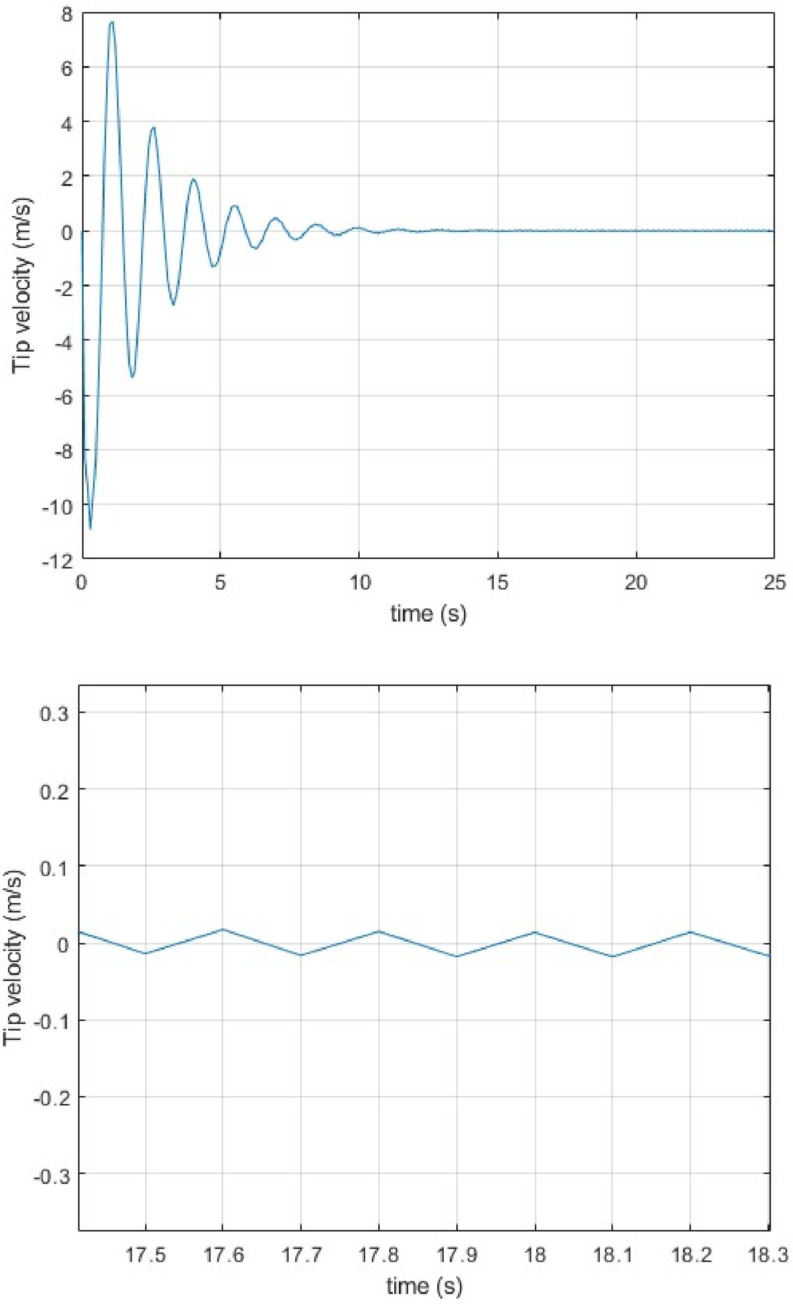
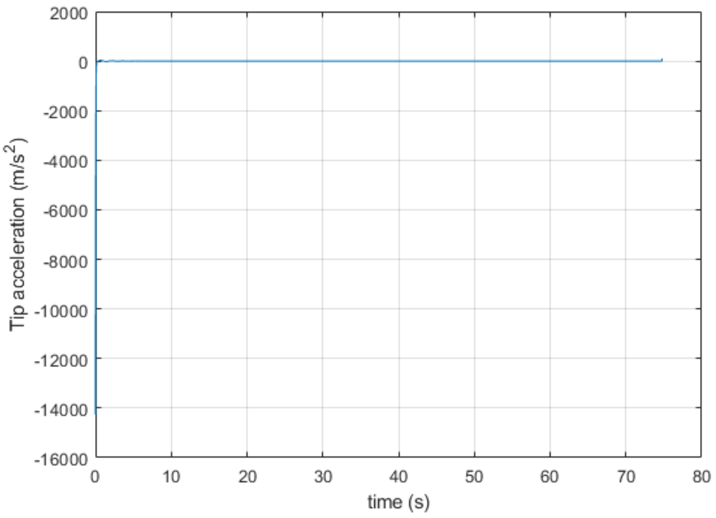
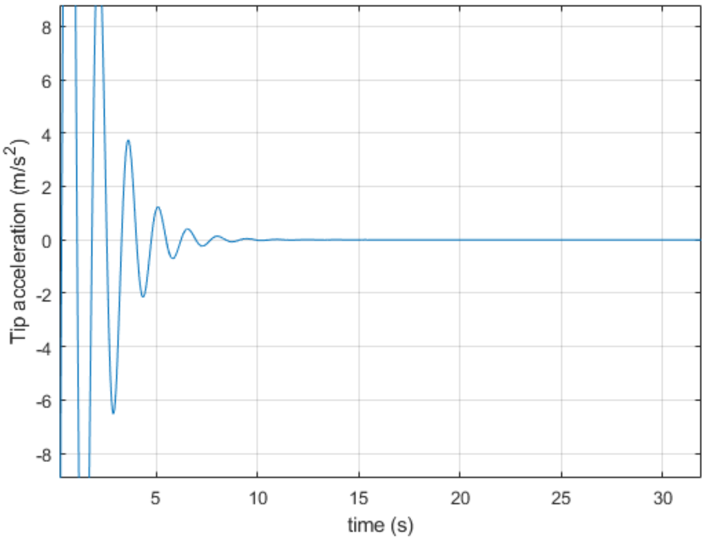
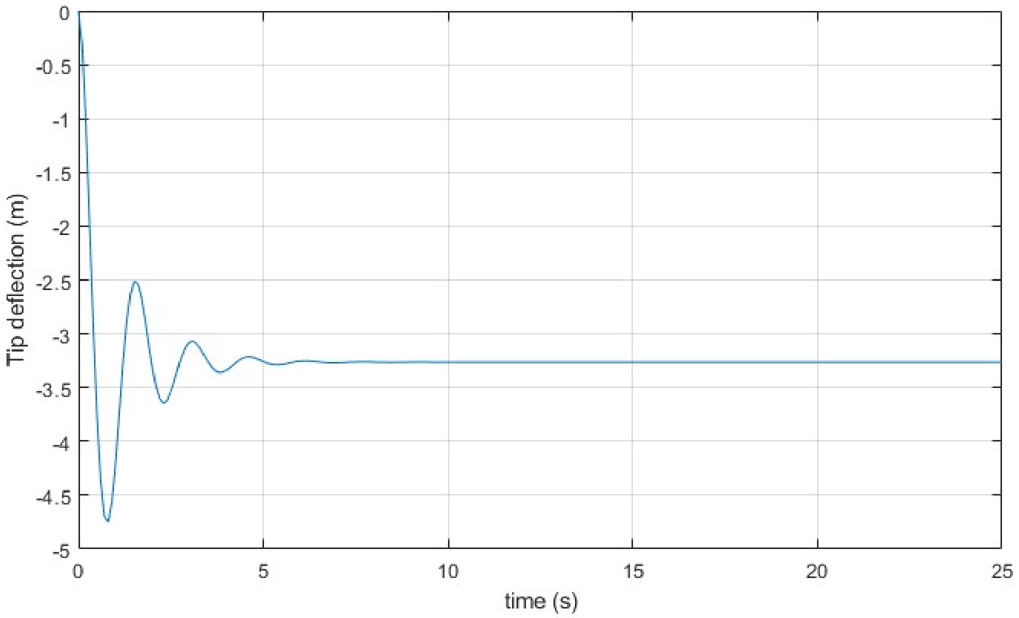


Figure 3.7: Time series of tip acceleration, deflection, velocity and zoomed in velocity when $\delta = 2\alpha = 0.5$

The figures 3.7 and 3.8 show the time series of tip deflection, tip velocity and acceleration of an NREL 5MW blade, subjected to a constant force of 3000 N in the flapwise direction. In the case of $\delta = 2\alpha = 0.5$, The acceleration and velocity of the tip do not converge even though the tip deflection is converging. But for $\delta = 1, \alpha = 0.6$ everything converges. Hence, the later values of Newmark coefficients are selected to proceed further.


Tip acceleration (m/s^2)
2000
0
-2000
-4000
-6000
-8000
-10000
-12000
-14000
-16000
0
10
20
30
40
50
60
70
80
time (s)


Tip acceleration (m/s^2)
8
6
4
2
0
-2
-4
-6
-8
5
10
15
20
25
30
time (s)


Tip deflection (m)
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
0
5
10
15
20
25
time (s)

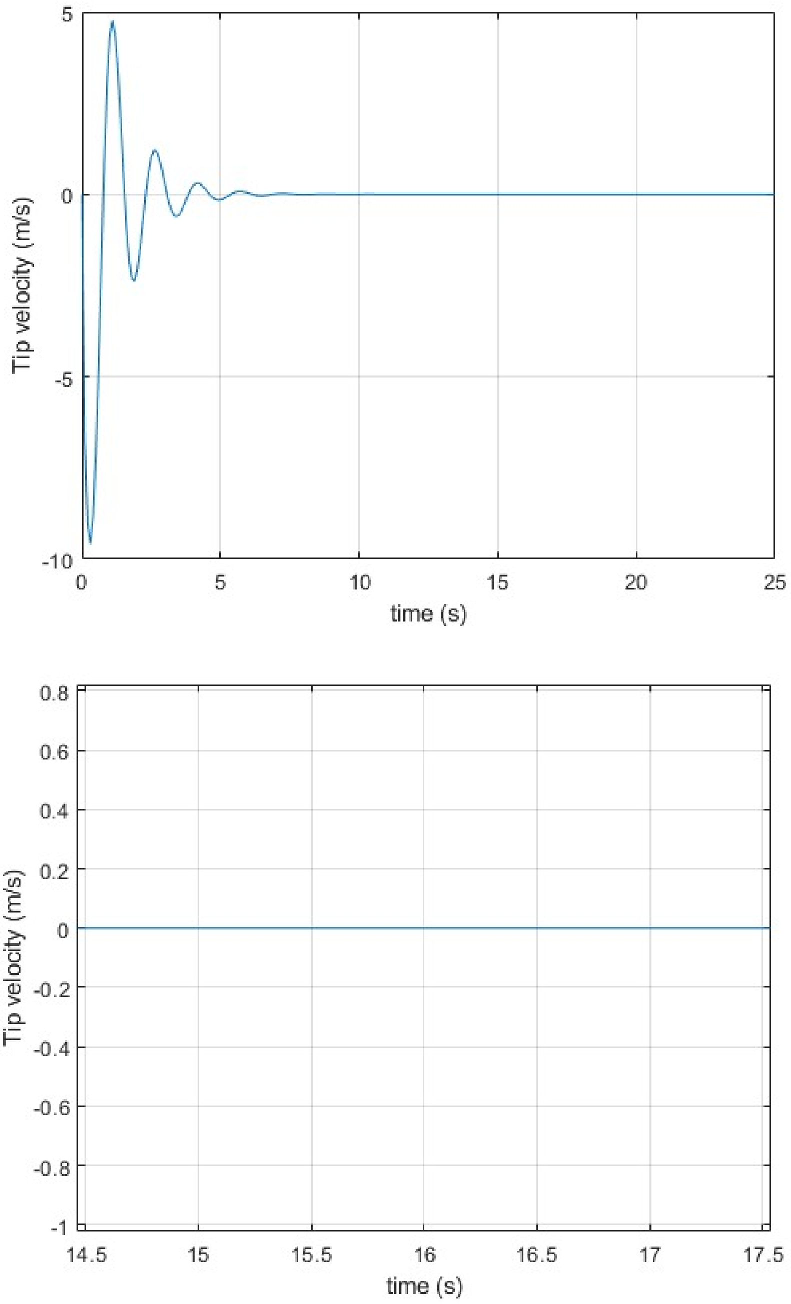


Figure 3.8: Time series of tip acceleration, zoomed in acceleration, deflection, velocity and zoomed in velocity when $\delta = 1, \alpha = 0.6$

3.3. Implementation in OpenFoam

This section elaborates on how the *TurbinesFoam* library, based on Bachant et al., 2018, is modified to include the structural deflection of the turbine model. The ultimate goal is to also update the *FloatingTurbinesFoam* library developed by Pericas, 2022. However, the implementation is done in such a way that the added deflections are valid for both steady and moving/floating turbines. At first, it is necessary to get familiar with TurbinesFoam and FloatingTurbinesFoam library (With the second being just an updated version of the first one, added with new functions and dictionaries). Indeed, the in-depth explanation of libraries are out of the scope of this thesis and requires a long time for new user to get familiarized. It is strongly recommended to go through the source code of libraries to understand the simulation flow. However, the following paragraphs are an attempt to explain the libraries on a higher level, without going too much into detail.

3.3.1. TurbinesFoam library

The TurbinesFoam library is based on multiple classes, coded in C++, intended to be used by OpenFOAM solvers. The classes of most importance are,

TurbineALSource: Writes, calculates and reads input and output to and from the user. The inputs can range from prescribed motion/rigid body motion criteria (rigid body motion refers to motion applied by the floater) to flow field, turbine performance etc. The class includes functions to calculate user-readable turbine performance such as C_t, C_q, turbine loads, blade loads etc.

AxialFLowturbineALSource: Includes function to create (/initialize) blades, nacelle, and tower in the field. Also includes functions to add an external force field created by the actuator lines to the added momentum term. Pericas, 2022 later updated this library to also accommodate turbine motions according to the given prescribed motion criteria or floater motion.

ActuatorLineSource: This class includes functions for individual actuator lines and can be used when there is no turbine motion/rotation. All different blades, tower, and nacelle with multiple actuator elements (AE) are treated as an object (object in the sense of object-oriented programming) of the actuatorLineSource class. Separate functions such as translating the whole line, finding force on the whole line etc. can be found here.

ActuatorLineElement: As an actuator line is made up of multiple actuator elements, functions to do operations on these elements are defined in the actuatorLineElement class. Functions such as but not limited to accessing different attributes of elements and changing those attributes

Apart from these classes, other classes such as addedmassmodel, dynamicStallmodel contain classes to calculate what the class name suggests! However, it is not required to modify it, as modifying the above-described classes will be enough to include deflection in the turbine.

3.3.2. Strategy for coupling aeroelasticity

The main strategy for including the deflection in existing code sounds straightforward: solve the flow field, calculate AE force, calculate deflections, apply deflection and then repeat. But including deflections along with moving turbine is more than that. Figure 3.9 shows the overall algorithm on how the element deflections are included.

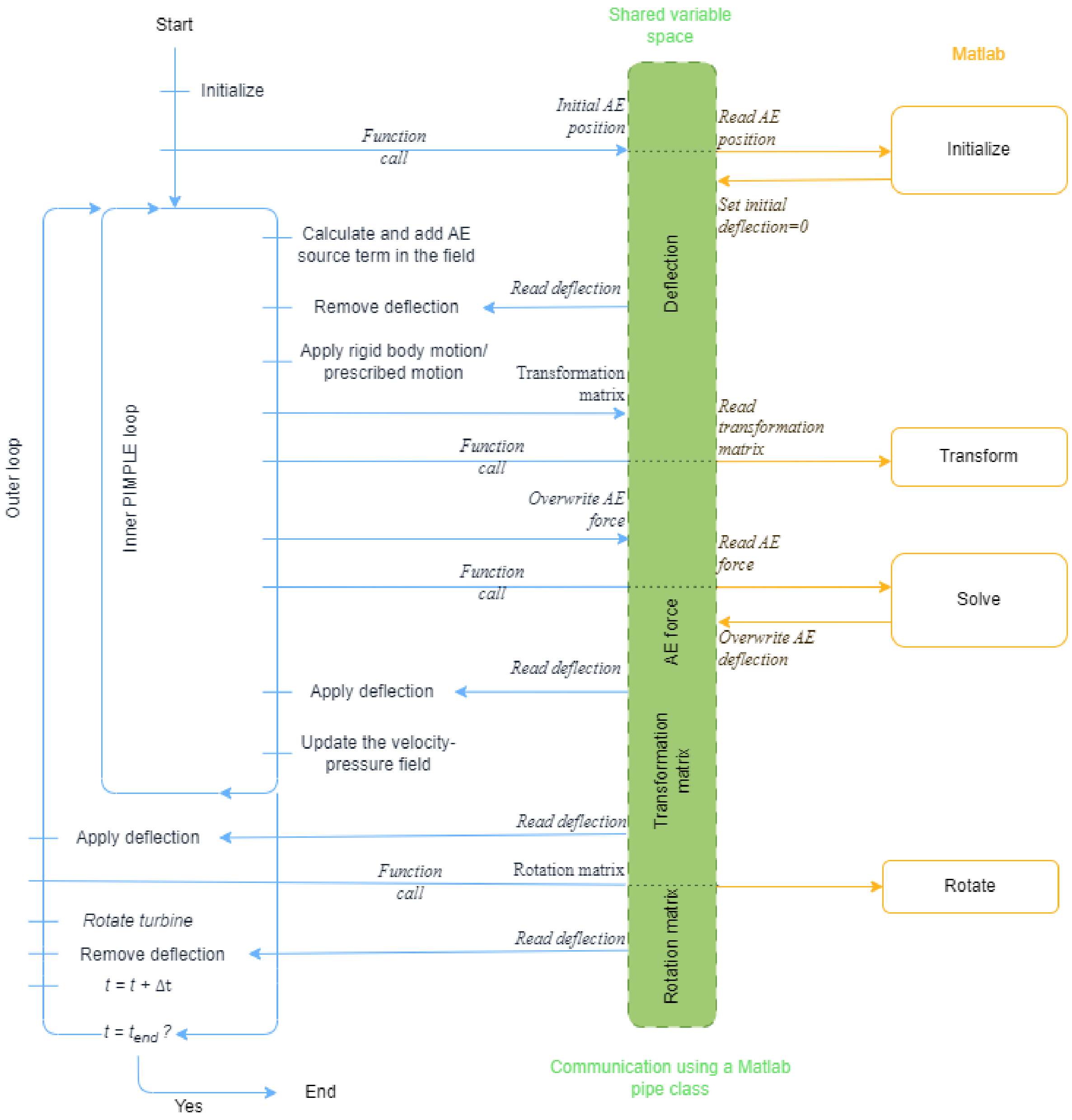


Figure 3.9: Implementation of the elasticity in OpenFOAM

The flow diagram only contains operations concerning the structural module. Other steps related to aerodynamic and hydrodynamic simulation are omitted and simplified, for the sake of simplicity. More details can be found in Pericas, 2022.

In the diagram, two loops are used one (the inner PIMPLE loop) for temporal convergence and another used for inter-temporal convergence. These two loops also work in favour of elasticity. As the turbine is subjected to rigid body motions, the location of the turbine element changes, the stiffness matrix changes, the aerodynamic forces change, and finally deflection changes; all within one time step. The initial inputs define the number of iterations of the inner loop.

Whenever the AE positions are updated, the deflections are needed to be removed.

Otherwise, the deflections will be carried forward in the applied motions/ rotation, which will result in the wrong location of the AE. We can subtract the deflection after the AE position is updated, but it will require sets of transformations for deflection as the deflections also have changed their frame of reference. It is also made sure that the velocity-pressure field is solved with the applied deflection.

3.3.3. Implementation in MATLAB

As MATLAB has many inbuilt functions and is easy to debug, it was selected to develop the structural module. The module itself is divided into 3 codes for easy handling and to enable particular function call from OpenFOAM whenever a separate operation is required to perform.

Initialize: This MATLAB function is called once whenever a simulation is carried out. It creates the blade and tower element matrices, described in section 3.2, in its own local frame of reference as mentioned in figure 3.6. It also finds the eigenfrequencies of the respective actuator line, required for the structural damping matrix. The sectional properties are interpolated from Jonkman et al., 2009 at span locations retrieved from OpenFOAM. For that initial positions of all the AEs are required. Constant gravitational force is calculated, to be applied at each element node in the solve function.

Transform/Rotate: These MATLAB functions take the rotation matrix as input and transform the elemental M,C,K matrices according to the applied prescribed motion or turbine rotor rotation, to get the matrices in the global frame of reference. After the elemental matrices are transformed, they are assembled to get the single global matrices as described in 3.2.3.

However apart from transforming the matrices; the deflection, velocity and acceleration vector (due to the deflection) from the previous iteration is also transformed to the current AE position. This step is well described in the figure 3.10 for deflection.

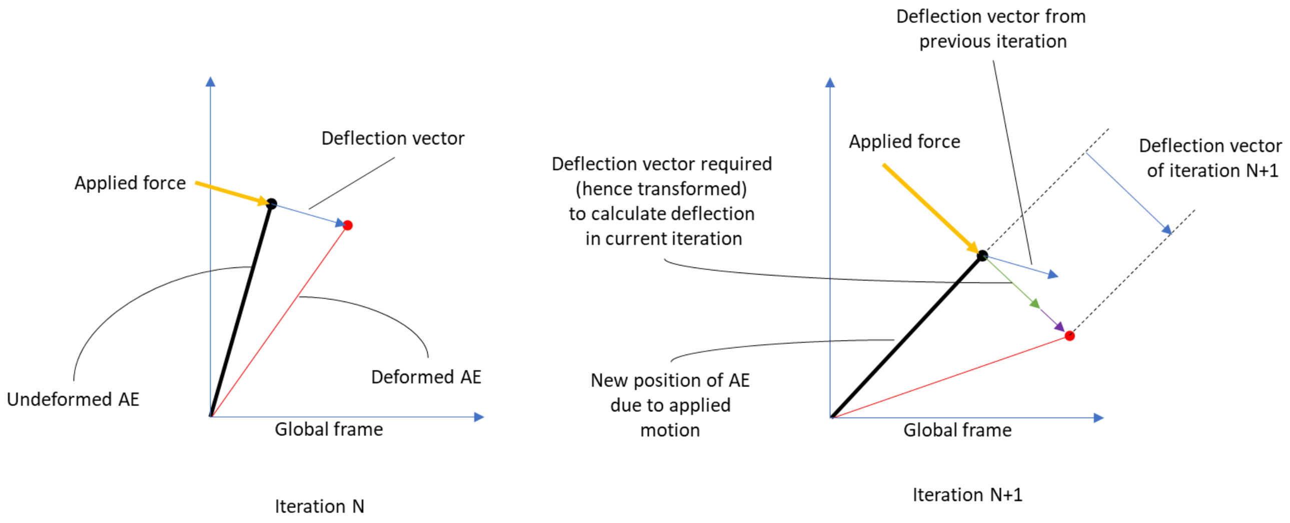

Deflection vector
Applied force
Undeformed AE
Deformed AE
Global frame
Iteration N
Deflection vector from previous iteration
Applied force
Deflection vector required (hence transformed) to calculate deflection in current iteration
Deflection vector of iteration N+1
New position of AE due to applied motion
Global frame
Iteration N+1

Figure 3.10: Transformation of deflection vector.

Solve: The *Solve* function reads the force value (along with gravitational force) and calculates the global deflection matrix. Newark's method is used as mentioned in section 3.1.6 to calculate the deflection. The transformed deflection, velocity, and acceleration vectors from the previous iteration are used to calculate deflection in the current iteration. The function then extracts deflection for all actuator lines from the obtained global deflection matrix, to be used by OpenFOAM for translating actuator lines (and their elements). Figure 3.11 explains the whole function in graphics.

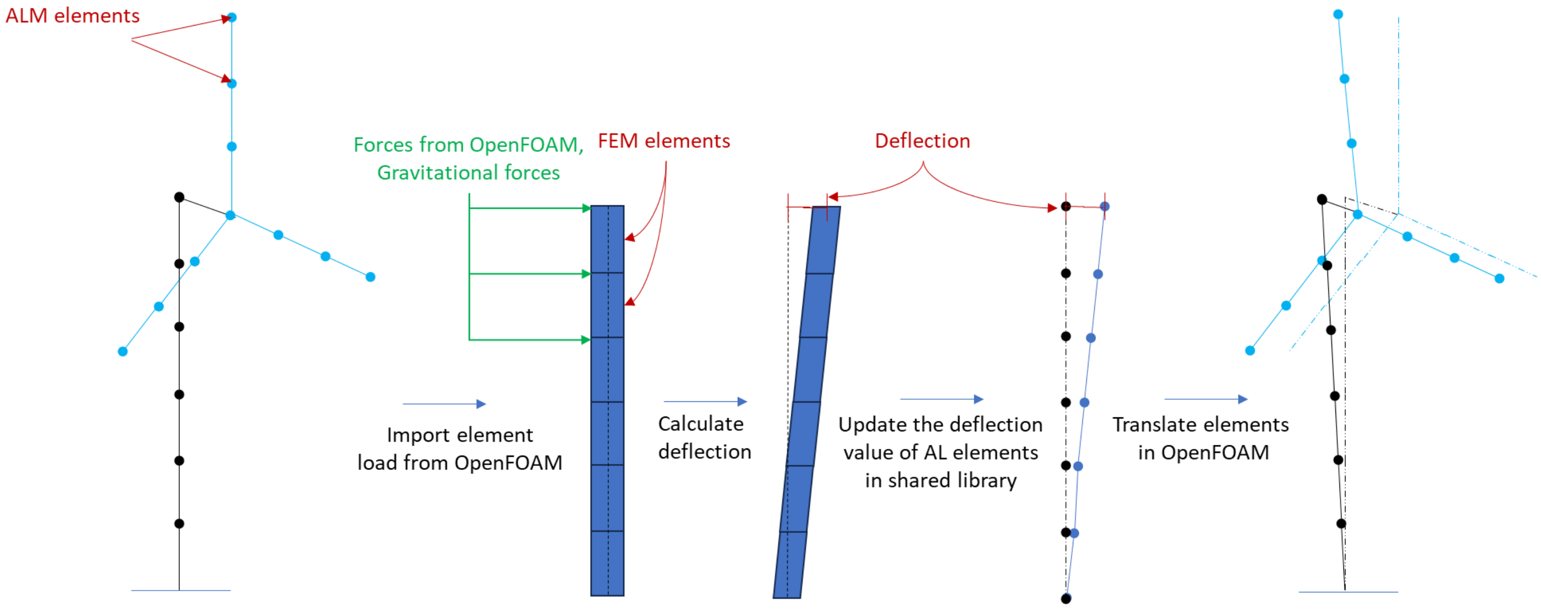


Figure 3.11: The solve function

3.4. Case setup in OpenFOAM

This section elaborates on different parameters used for the simulation in openFOAM such as boundary conditions, time discretization, wind shear etc.

Time discretization

The time is discretized based on the Courant-Friedrichs-Lewy (CFL) number. CFL defines how fast the particle propagates in the field in a time step.

$$CFL = \frac{\Delta t u}{\Delta x} \leq 1 \tag{3.48}$$

Where $u, \Delta t, \Delta x$ are particle velocity, time step, and local mesh size (assuming the mesh size is uniform in all directions).

The goal is to keep the particle propagation in the field to be less than that of the size of the local mesh. Thus, the CFL number must be kept less than 1 for the numerical stability of CFD simulation. But if an actuator element is introduced in the simulation, it is also required that the element does not travel more than the local grid size within a time step.

For wind turbine simulation, around the rated speed, the turbine tip moves much faster than the wind velocity. Thus the CFL condition for the turbine tip becomes the governing one. To select the maximum time step according to the turbine tip velocity, the following equation can be used. Pericas, 2022

$$\Delta t \leq \frac{\Delta x}{U_{\infty} TSR} \tag{3.49}$$

Where, Δx is the local grid size, $U_{\infty}TSR$ is turbine tip velocity. However, if deflections are included, the velocity due to deflection needs to be taken into account. The updated time discretization formula now is given by,

$$\Delta t \leq \frac{\Delta x}{U_{\infty} TSR + U_{deflection}} \tag{3.50}$$

Where $U_{deflection}$ is the velocity due to deflection. The value of deflection velocity will depend on applied forces on elements. So as a starting point, the time value obtained from equation 3.49 is multiplied by 0.75 (Thus CFL=0.75). At 12m/s hub wind speed and TSR of 7.5, the value of tip speed due to rotation is 90 m/s. The maximum deflection velocity of the turbine tip from simulations was found to be in the range of 15-20 m/s in the out-of-plane direction and 5-6 m/s in the in-plane direction. The reduced CFL number of 0.75 was enough to accommodate the increased element speed. But for higher wind speeds, the time step must be selected cautiously to accommodate even higher element deflection velocity.

Boundary condition and wind shear

The following table summarizes the boundary conditions for the variables.

Table 3.3: Boundary Conditions

Boundary	**U**	**P**	**k**	ω
Inlet	`logLawInlet`	`zeroGradient`	`fixedValue`	`fixedValue`
Outlet	`zeroGradient`	`fixedValue=0`	`fixedValue`	`fixedValue`
Walls	`slip`	`zeroGradient`	`fixedValue`	`fixedValue`

The velocity inlet and initial field velocity are modelled as logarithmic wind profile. The walls are modelled as an atmosphere except for the bottom wall, where velocity will always be zero. The used turbulence model is $k - \omega$. The fixed value of turbulent kinetic energy - k and the specific rate of dissipation of turbulent kinetic energy ω is given by

$$\begin{aligned} k &= \frac{3}{2} I |u_{ref}|^2, \\ \omega &= \frac{k^{0.5}}{{C_{\mu}}^{0.25} L} \end{aligned} \tag{3.51}$$

Where I is turbulence intensity, (taken to be 2% for all cases), u_{ref} is wind velocity at hub height,C_{μ} is constant with the value of 0.09, and L is reference length. (taken as the chord length).

The logarithmic wind profile is modelled as following equation

$$U(z) = U_h \frac{ln(z-d)/z_0}{ln(z_h-d)/z_0} \tag{3.52}$$

Where $U(z)$ is the velocity at height z,U_h is hub height velocity, z_0 is surface roughness (chosen to be 0.5), d is Zero-plane displacement below which wind speed is assumed to be zero (chosen to be 10 meters).

3.4.1. Mesh

The mesh in the computation region is divided into five sub-regions representing the actual turbine (Zone d), near wake (Zone b), surrounding the wake (Zone a), inlet region near the bottom wall (Zone c) and the remaining region (Zone 0). The cross sections are at the rotor centre

The figures 3.12,3.13.3.14 show the mesh of the case_1223 where 1223 represents the refinement level of zone-a,b,c and d respectively. Here the refinement level 2 means the cell size is 4 times smaller than the zero level mesh size, level 3 means the cell size is 8 times smaller than the zero level mesh size.

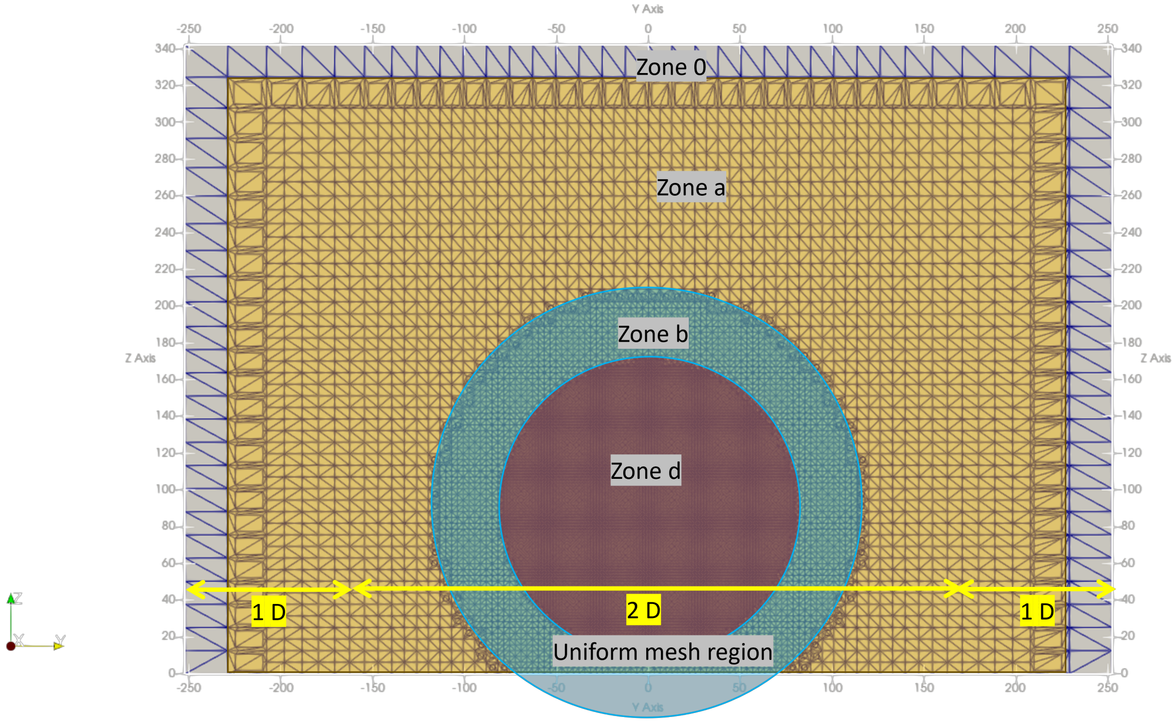


Figure 3.12: Mesh region: front view

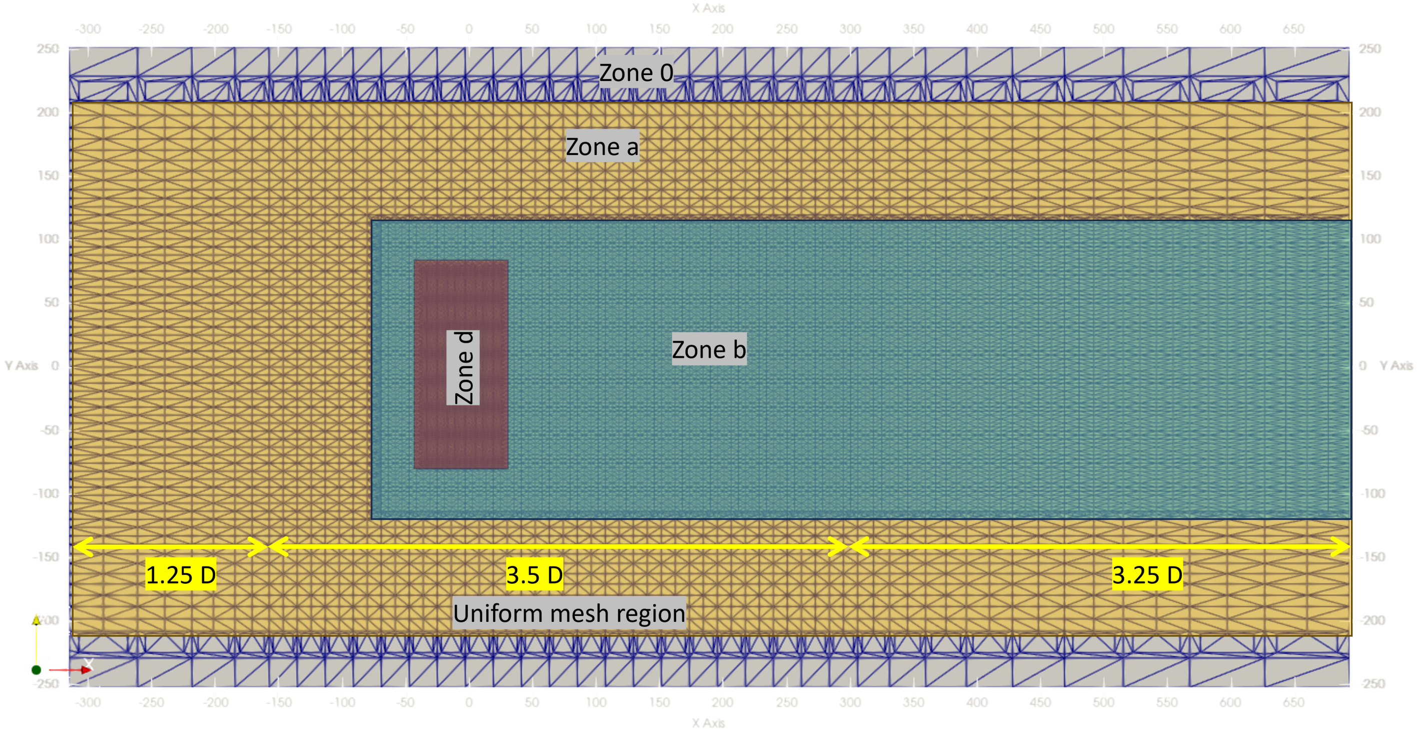


Figure 3.13: Mesh region: top view

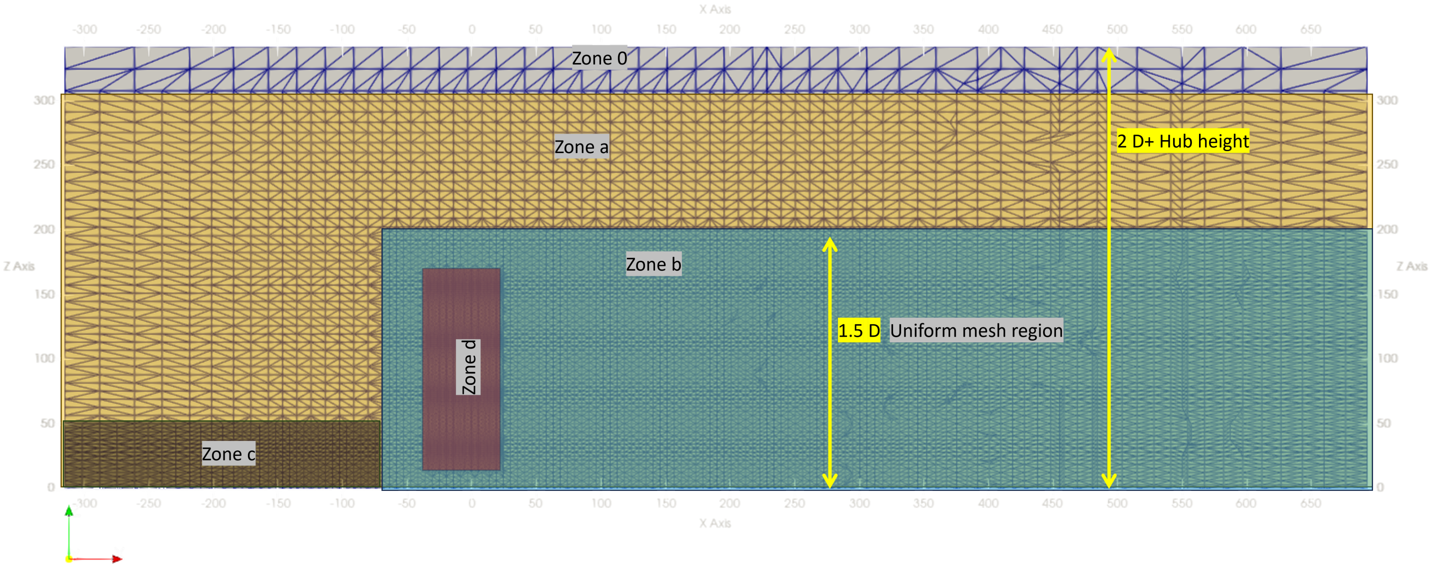


Figure 3.14: Mesh region: side view

The above figures show the computing region from 3 directions. The rotor centre is located at $x = -5m, y = 0m, z = 90m$. While selecting the mesh size, the guidelines from Pericas, 2022 and Malveiro, 2022 were followed. The 0-level mesh size (zone-0) in the Y and Z direction is taken as 12.8 meters ($\sim D/10$), applied to the uniform mesh region (the region in which the cell size does not change). The cell aspect ratio in the uniform cell region is kept $\sqrt{3}$. In the region before and after the uniform mesh region, the cell length increases in all directions by a preset overall expansion ratio. The local mesh size near the turbine rotor is kept between 0.8% to 2.4% of the rotor diameter. The mesh refinement level of 1,2,3 and 4 will give a mesh size of 5%,2.5%,1.25% and 0.75% of the rotor diameter respectively.

The developed elastic TurbinesFOAM library, MATLAB pipe class along with developed MATLAB structural dynamic modules, and used test cases can be found here.

4

Results

In this section, we will first go through the verification of the FEM model that was described in the previous section. After verification, we will investigate the effect of the aeroelasticity in wind turbine simulation in the OpenFOAM.

4.1. Verification of structural FEM model

To be sure that the model developed exhibits accurate behaviour, at first, we will compare the natural frequency of NREL 5MW blades and respective mode shapes with the BModes. BModes is a tool developed by NREL to calculate eigenfrequency and mode shapes of rotating as well as non-rotating beams. Bmodes can be used to do modal-frequency based analysis of beams. It can also be purposeful for verification of another developed FEM model.

The major reason for choosing Bmodes is the exclusion of non-linearity. As mentioned in G. S. Bir, 2007, BModes considers the beam as an Euler-Bernoulli beam and neglects any shear deformation in the axial direction. As the FEM model developed in this thesis also does not include non-linearity, it is best to compare it with a similar tool. After adding a non-linear source term, BeamDyn can be used to verify results. For now, having a non-linear model is out of the scope of this thesis and can be worked upon in future.

4.1.1. Natural frequency and mode shapes of blade

After obtaining the mass and stiffness matrix for blades as shown in section 3, natural frequency and mode shapes are obtained in Matlab. The Below table compares the eigenfrequency of a non-rotating (i.e $\Omega = 0$) blade with BeamDyn and Bmodes.

Table 4.1: Natural frequency of non rotating blade compared with BeamDyn and Bmodes

Modes	Current	BeamDyn	Bmodes
1st Flapwise	0.6931	0.6993 (0.88%)	0.6918 (0.19%)
1st Edgewise	1.1149	1.0793 (3.30%)	1.1101 (0.43 %)
2nd Flapwise	1.9943	1.9337 (3.14%)	1.9963 (0.1%)
2nd Edgewise	4.1356	N/A	4.095 (0.99%)

From the above table (4.1), it can be said that the natural frequencies do not deviate to a great extent from the tools which are used in industry for structural analysis of wind turbine blades.

Upon verifying the natural frequencies, we will verify the mode shapes of the model with that found in Bmodes. The following images compare the first four mode shapes of the non-rotating blade as well as with the blade rotating at 20 rpm. The graphs show normalized mode shapes (i.e. the max value in the mode shape is set to 1). From figure 4.1 and figure 4.2, it can be said that the mode shapes match with that of Bmodes.

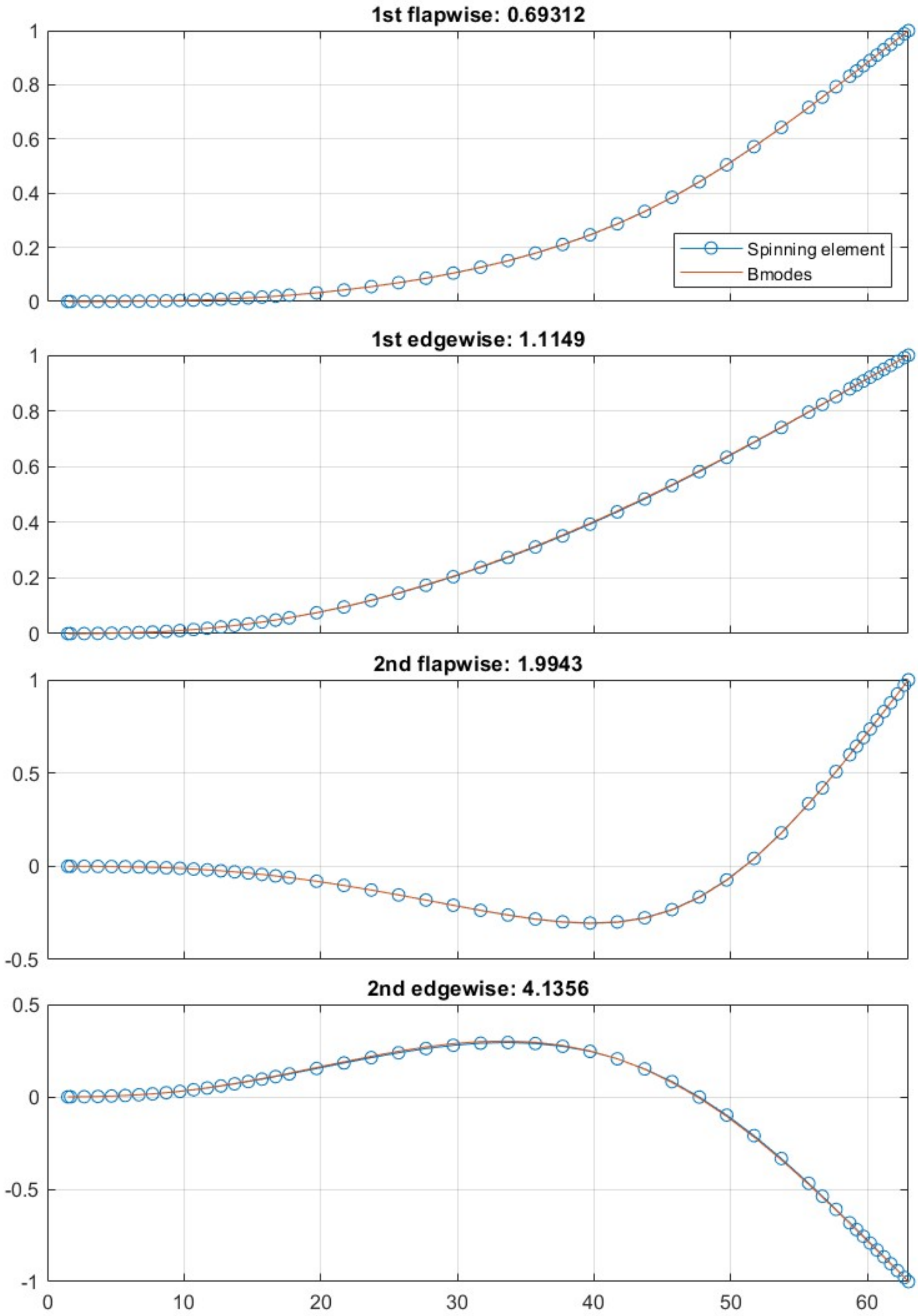


Figure 4.1: First four mode shapes compared with Bmodes for a non-rotating blade

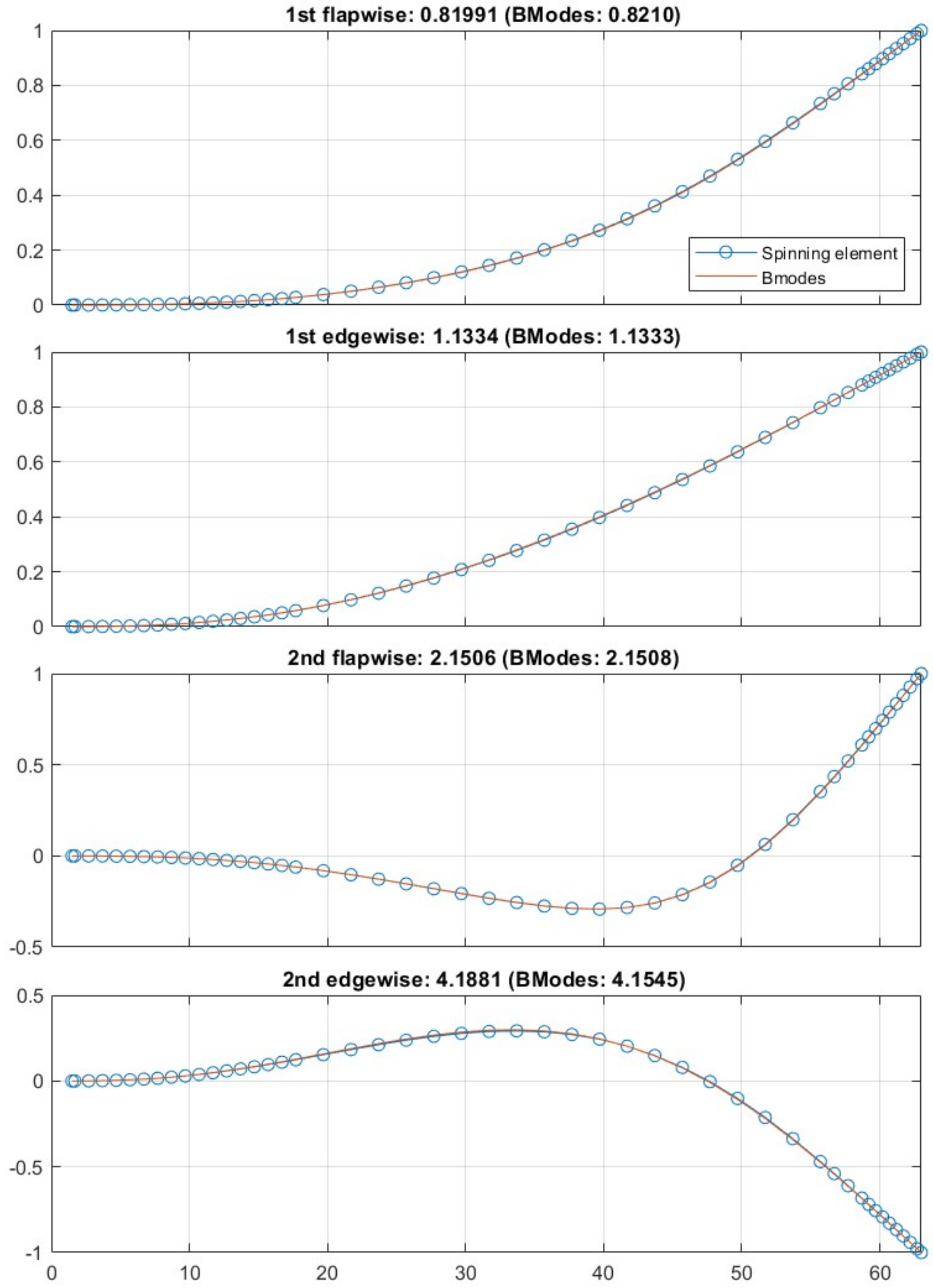


Figure 4.2: First four mode shapes compared with Bmodes for blade rotating at 20 RPM

As the turbine RPM increases, the blades become stiffer hence increasing the natural frequency. The natural frequencies of the blade at different rotor RPMs are given in figure 4.3 along with the frequency found by BModes. The model gives almost similar frequencies. As the RPM increases, the edgewise natural frequency can be seen deviating a little from BModes. This can be due to the different approaches used for mass and stiffness matrices.

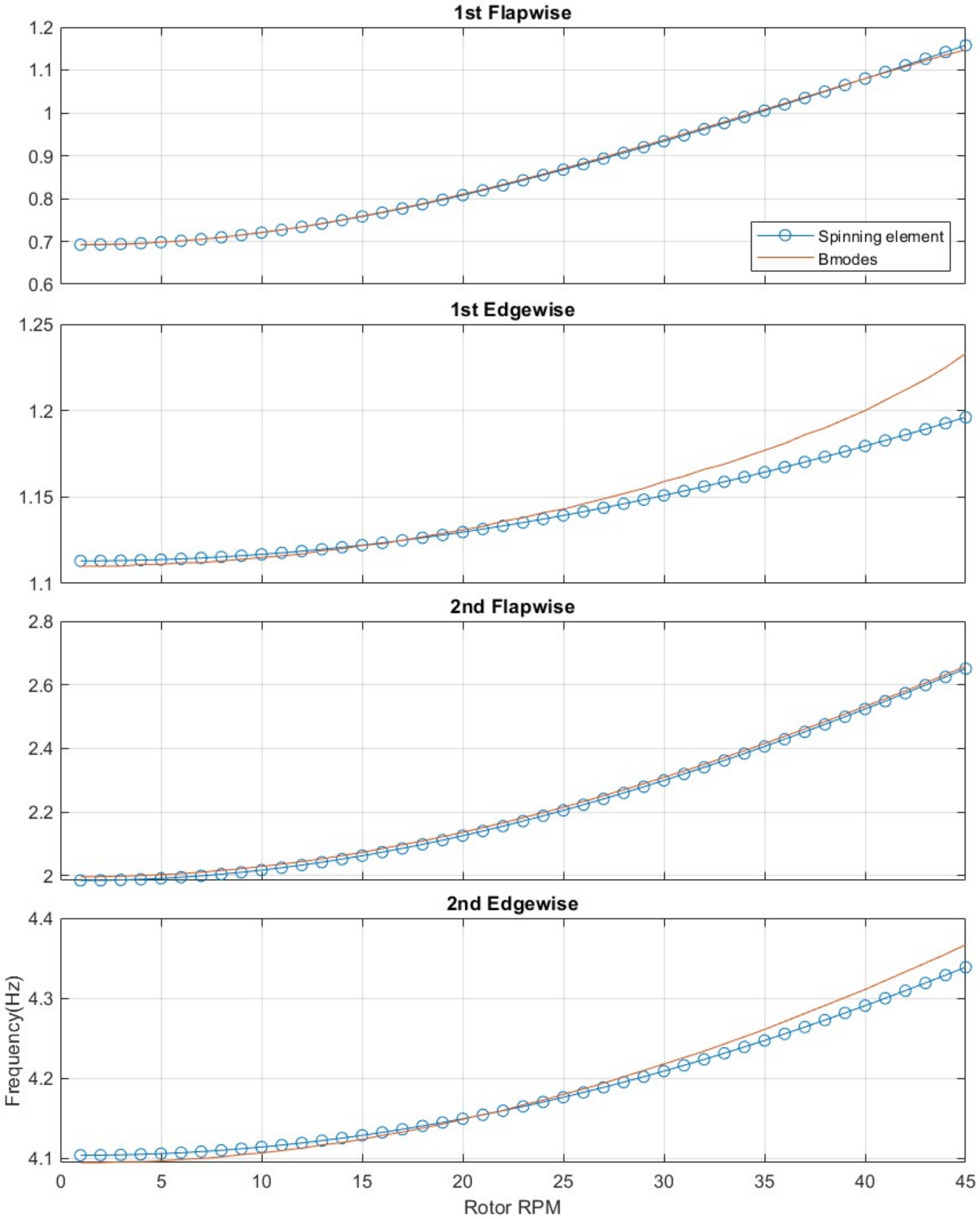


Figure 4.3: Variation of natural frequency with respect to rotor RPM

4.1.2. Dynamic response

Apart from comparing the mode shapes and eigenfrequencies of the blades, the dynamic deflection needs to be verified. The following figure shows the tip deflection of the blade under a distributed sinusoidal force at different frequencies. The blade is rotating at 20 RPM. We can see the peak amplitude at the natural frequency of the blade as described in figure 4.2. The damped case includes both gyroscopic and structural damping. While the other case does not contain any damping at all.

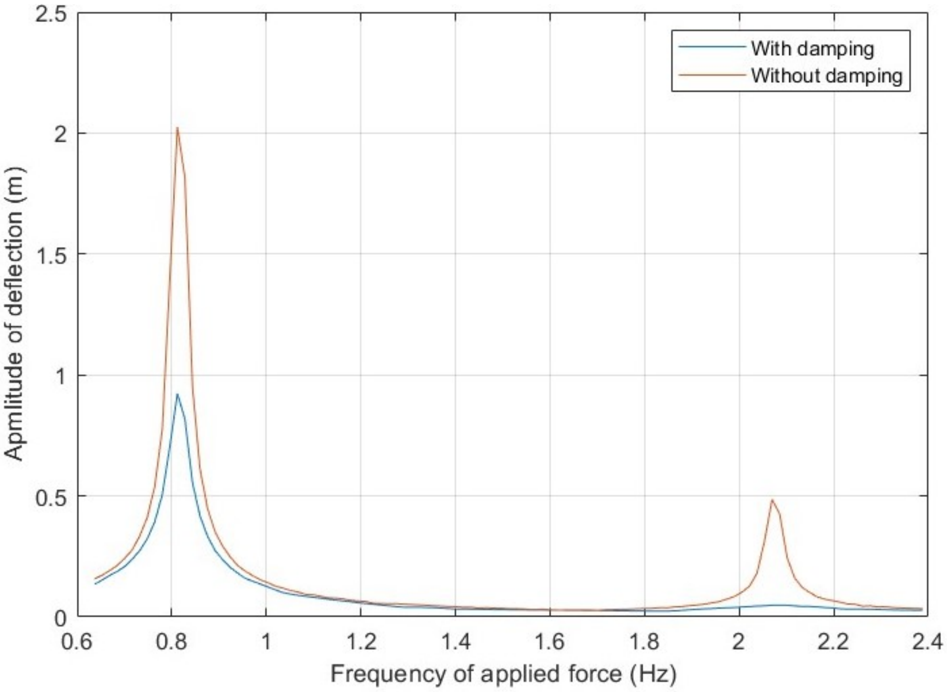


Figure 4.4: Flapwise variation of tip deflection with respect to dynamic forces applied at various frequencies

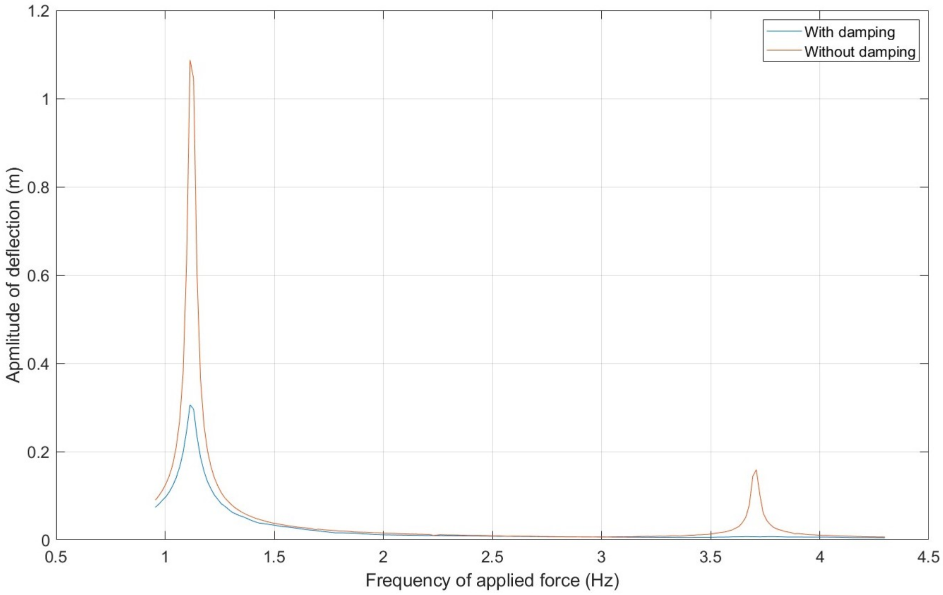


Figure 4.5: Edgewise variation of tip deflection with respect to dynamic forces applied at various frequencies

4.1.3. Verifying with OpenFAST

OpenFAST is an open-source wind turbine simulation tool designed for the analysis and design of wind turbines by NREL. It includes different modules such as AeroDyn, ElastoDyn, ServoDyn etc. Which can simulate different aspects of wind turbine simulation. For verification purposes, we will only focus on 3 modules- Elastodyne, InflowWind and AeroDyn; to verify a test case for the added deflections in the OpenFOAM. Elastodyn is used to compare the result as it utilizes a linear equation of motion for deflection. The comparison is not aimed to validate the results but to see if the developed structural model outputs comparable data to OpenFAST.

Test case description: For verification, the NREL 5 MW turbine (in steady condition) **with** the tower is considered in a uniform flow (no wind shear) field. The hub height is 90 meters from the ground. Other case parameters such as mesh, boundary conditions etc. are the same as mentioned in the section 3.4 and 4.2 unless mentioned specifically. Two test cases with two different uniform wind speeds are considered- 9 m/s and 12 m/s. The medium mesh size is used from 4.2

The following graphs compare the tip deflection for both the cases (9 and 12 m/s) in OpenFAST and OpenFOAM. The vertical dashed line shows that the blade-1 is in the vertical position with the tip facing upwards towards the sky.

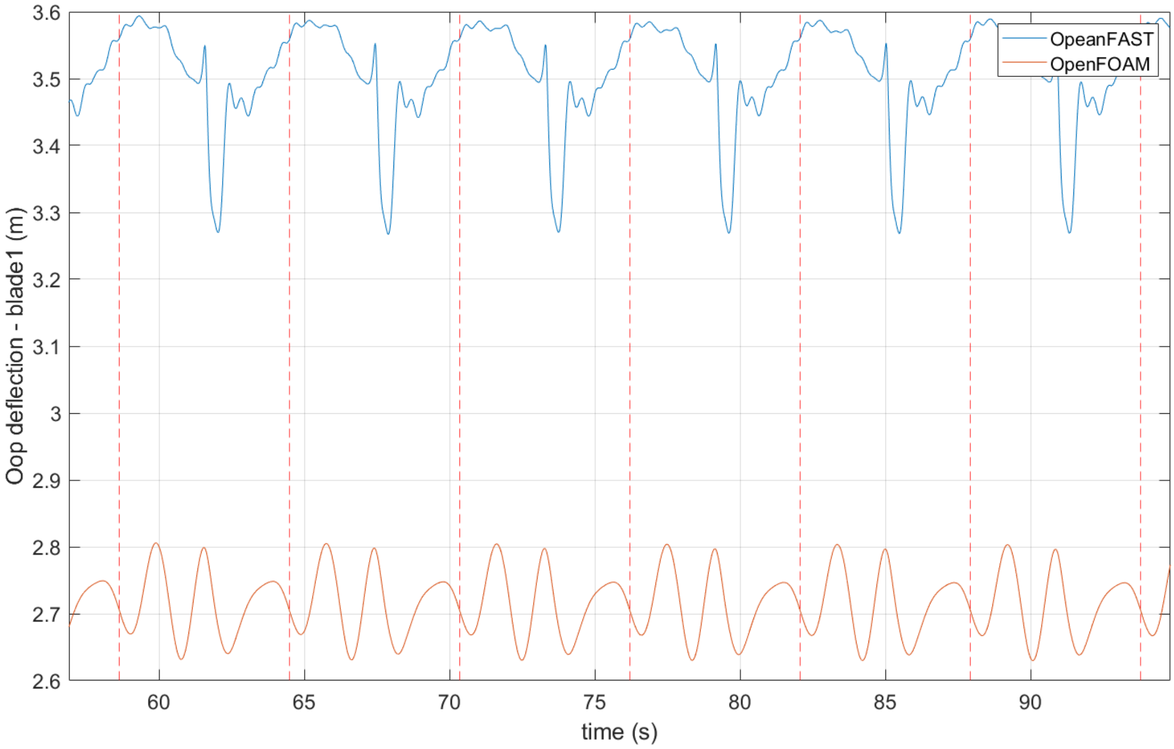


Figure 4.6: Time series of out of plane deflection- 9m/s

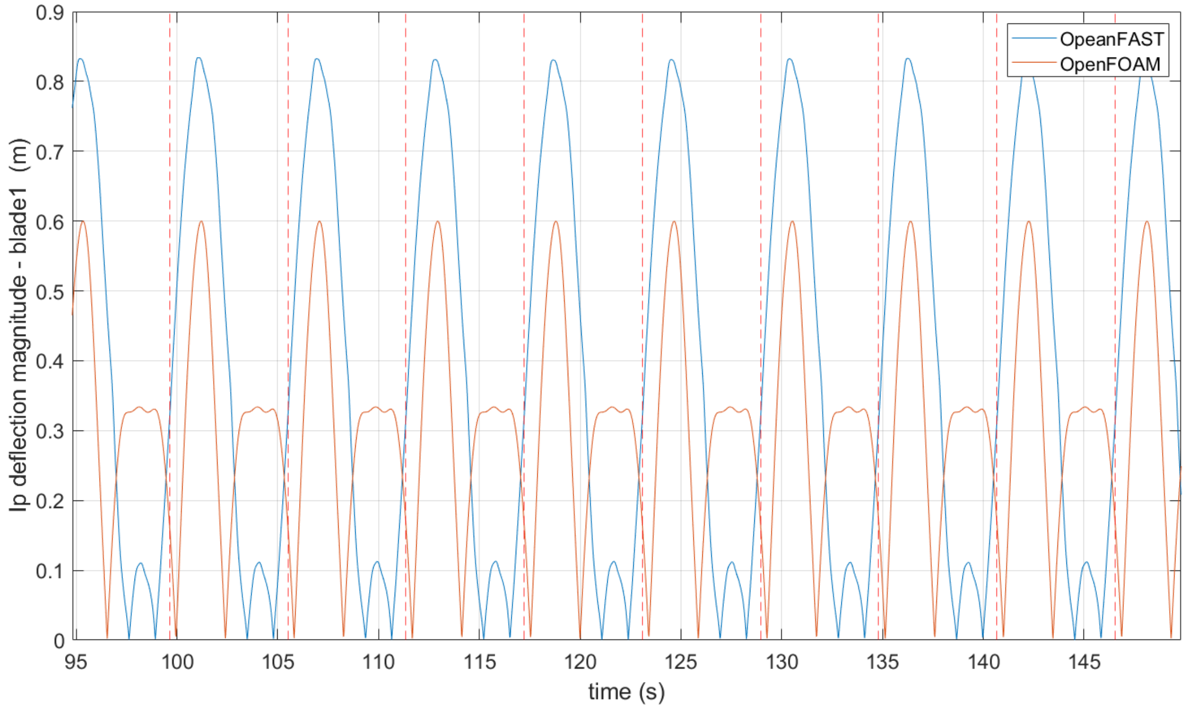


Figure 4.7: Time series of in plane deflection - 9m/s

There is a significant difference in the tip deflection value of OpenFAST and OpenFOAM. The reason behind it is the applied aerodynamic force on the blade. The following images compare the out of the plane aerodynamic force applied to the blade.

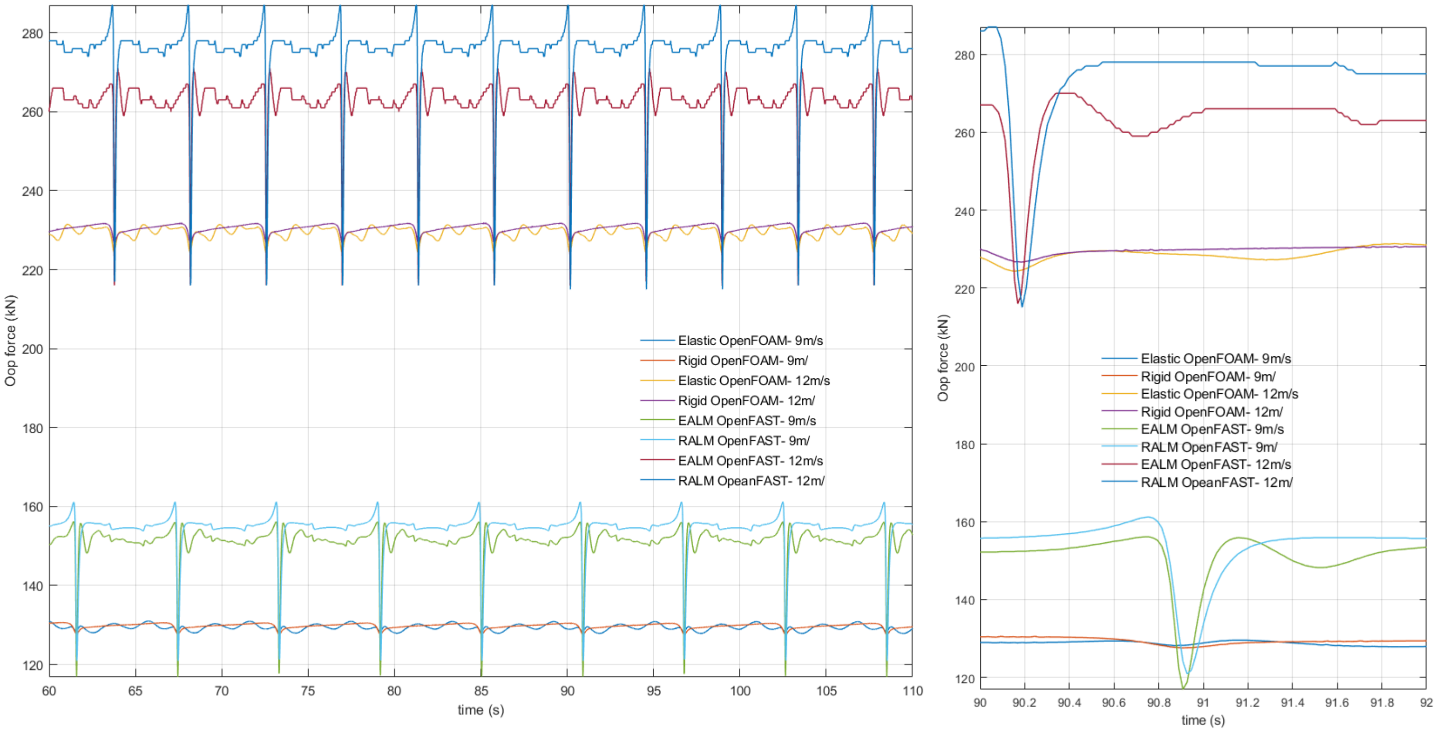


Figure 4.8: Time series of Out of plane force on blade -1 with zoomed-in values.

The out-of-plane force obtained in OpenFOAM is lesser than that found in OpenFAST. This behaviour can be explained by the inherent nature of both tools and both software's methodologies to calculate loads. TurbinesFoam is an ALM-based simulation tool, where the loads are resolved using CFD. OpenFAST is an analytical tool. Moreover, the input element distribution is also different in both tools. The drop in force value due to tower shadow is much higher in OpenFAST than in OpenFOAM. This phenomenon illustrates the inability of the ALM-based method to capture accurate velocity profile near the turbine surfaces.

However, to check the accuracy of the developed structural model, the forces obtained from OpenFAST are applied to the developed model. The total blade force output from OpenFAST is distributed evenly on the blades in the developed code. The out-of-plane deflection is given in figures 4.9 and 4.10.

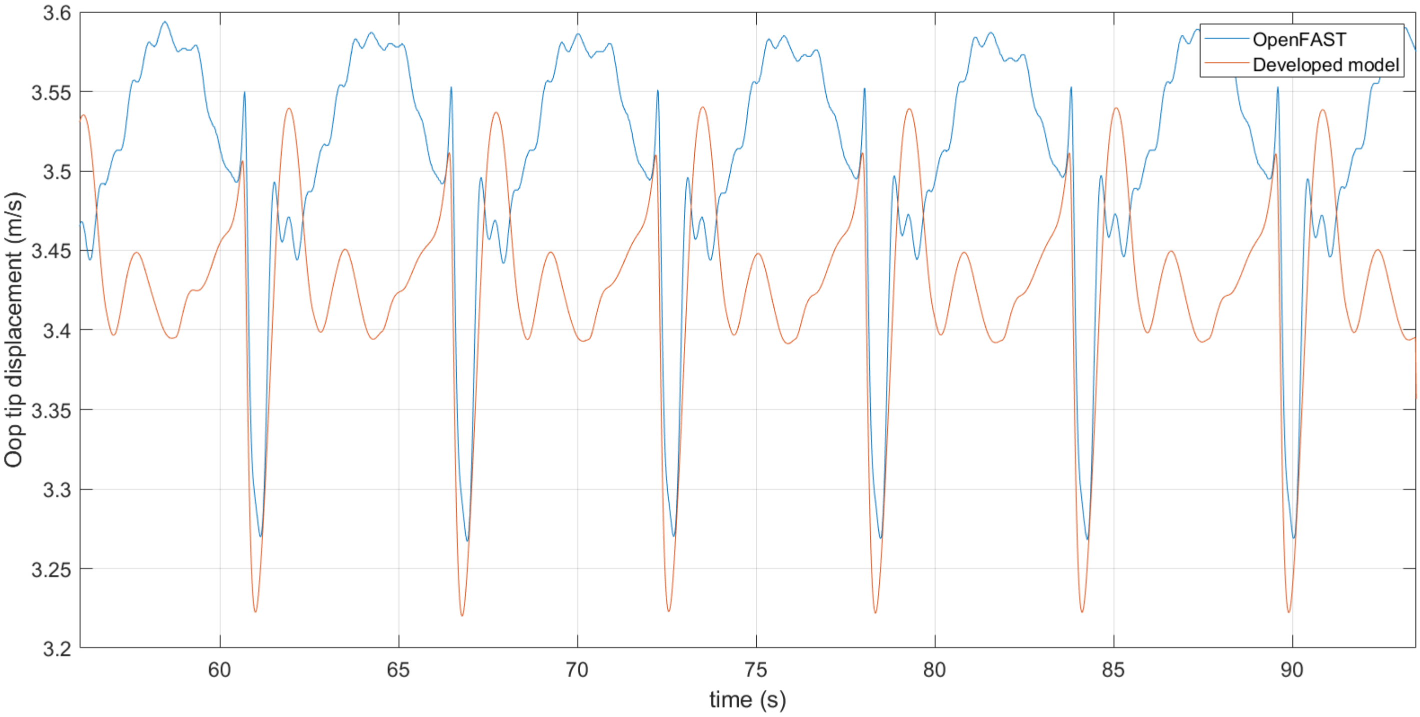


Figure 4.9: Comparison of OOP blade deflection in developed model and OpenFAST with same input loads at 9 m/s wind speed.

The deflection from the developed model is found to be comparable to the OpenFAST. There are some differences in deflection value when the blade is in the upright position. These differences are justifiable as the forces in developed models are evenly distributed spanwise. While in OpenFAST, it is not. Due to the absence of aerodynamic damping, the deflection values show more oscillations whenever there is a sudden change in input force. Otherwise obtained deflection values almost match that of OpenFAST. Thus, it is safe to say that the developed linear model predicts the comparable deflection when verifying against ElastoDyn.

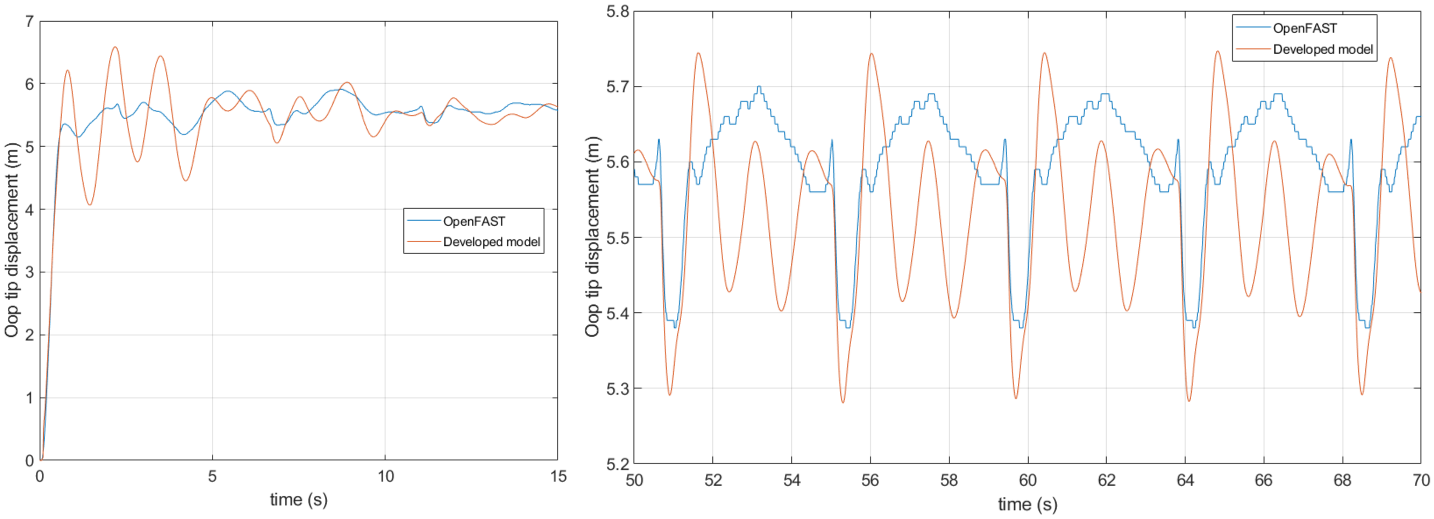


Figure 4.10: Comparision of OOP blade deflection in developed model and OpenFAST with same input loads at 12 m/s wind speed.

4.2. Mesh dependence analysis

Before proceeding with the actual elastic simulation, it is necessary to set up the correct mesh size to capture the accurate turbine loads and wake region. Studying the effect of elasticity on the wake region is essential for turbines in a wind farm configuration. This analysis can also serve as the grid dependence study for the turbine with prescribed motion.

Case setup: The computation region is set up in a rectangular box region. The simulated turbine (NREL 5MW) is moving with harmonic surge motion, with a frequency of 0.1 Hz and amplitude of 5 meters. Inlet velocity is 9 m/s with a logarithmic wind profile. The turbine TSR is of 7.5. The turbulence intensity is 0.02.

For the study, a total of three cases with different mesh configurations are considered. The coarse mesh (Case 1112), the medium mesh (Case 1223) and the fine mesh (Case 3334). Again, the computational region is divided into 5 regions as mentioned in 3.4. The mesh size in zero level mesh refinement level in the Y and Z directions is kept at 12.8 meters ($\sim 10\% D$). The coarse, medium and fine mesh has a total of around 0.27, 0.94 and 12.3 million cells respectively in the computational region. The table 4.2 summarizes the mesh size with respect to turbine diameter in the uniform mesh region.

Table 4.2: Grid sizes in uniform mesh region in Y and Z direction. Described as a percentage of turbine diameter

Case	Surrounding Turbine wake	Near bottom wall	Wake region	Turbine rotor
Coarse (case-1113)	5%	5%	5%	2.5%
Medium (case-1223)	5%	2.5%	2.5%	1.25%
Fine (case-3334)	1.25%	1.25%	1.25%	0.75%

Result comparison: The images 4.11, 4.12, 4.13 compare the wake velocity profiles (U_x, U_y, U_z) in the fully developed wake region (at t=250 s), at 1D (1 times the rotor diameter) to 5D location after the rotor. It can be seen that as the mesh gets finer, the velocity profile gets smoother and the resolution increases. However, not much of a difference is seen for the U_x velocity. **Relative** velocity differences in y and z are much more than that of x direction. This difference in U_y, U_z can further be visualized in the vorticity images as presented in figures 4.17- 4.19. The model uses the RANS method to calculate turbulence. The difference in U_y, U_z could further have been magnified if the higher model for turbulence was to be used such as LES.

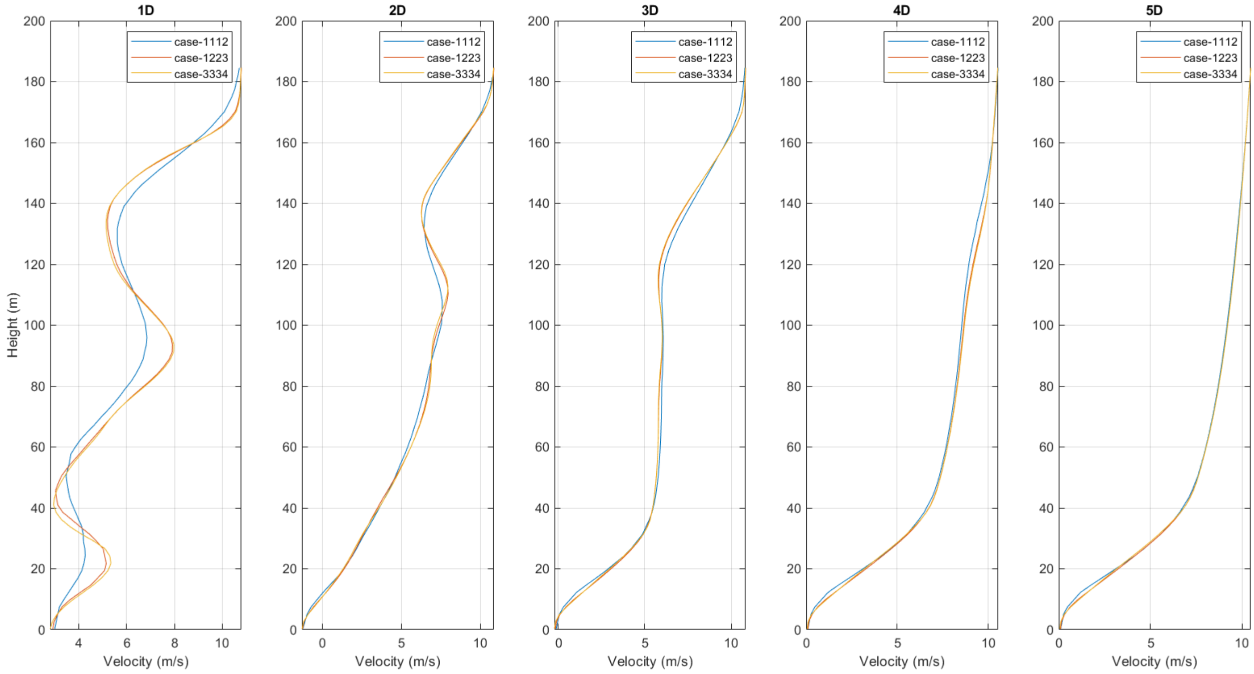


Figure 4.11: U_x for the test cases at different wake locations along a vertical line

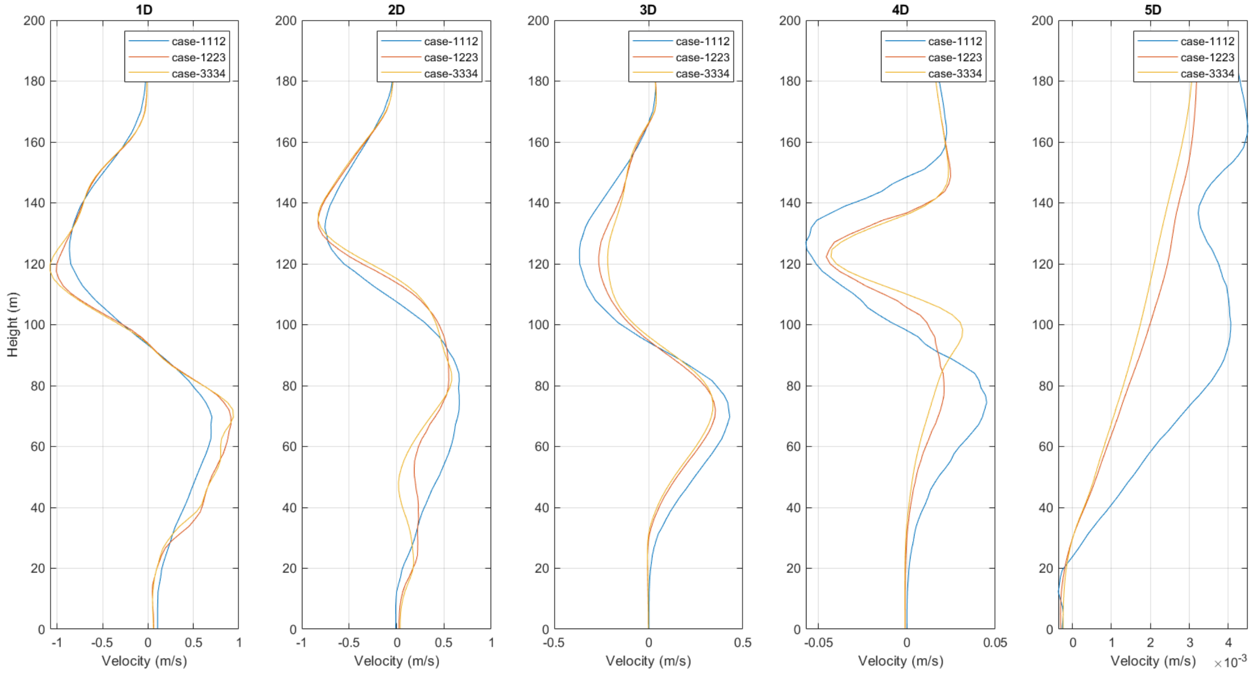


Figure 4.12: U_y for the test cases at different wake locations along a vertical line

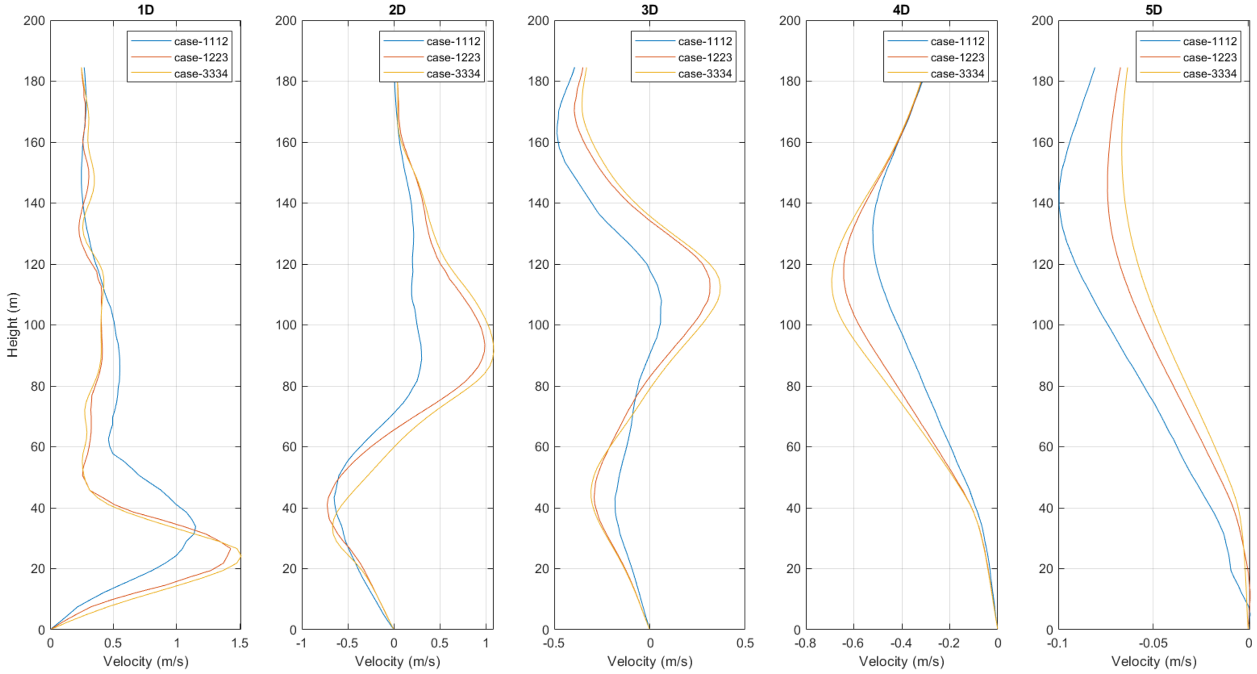


Figure 4.13: U_z for the test cases at different wake locations along a vertical line

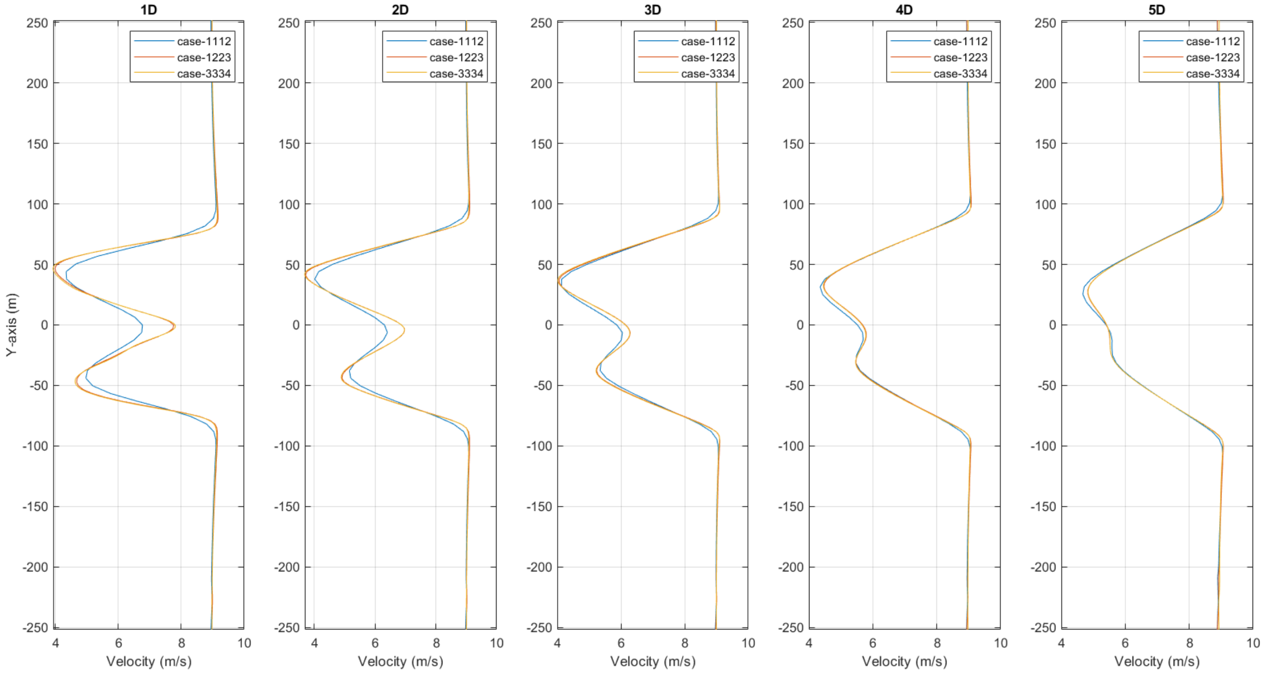


Figure 4.14: U_x for the test cases at different wake locations along a horizontal line

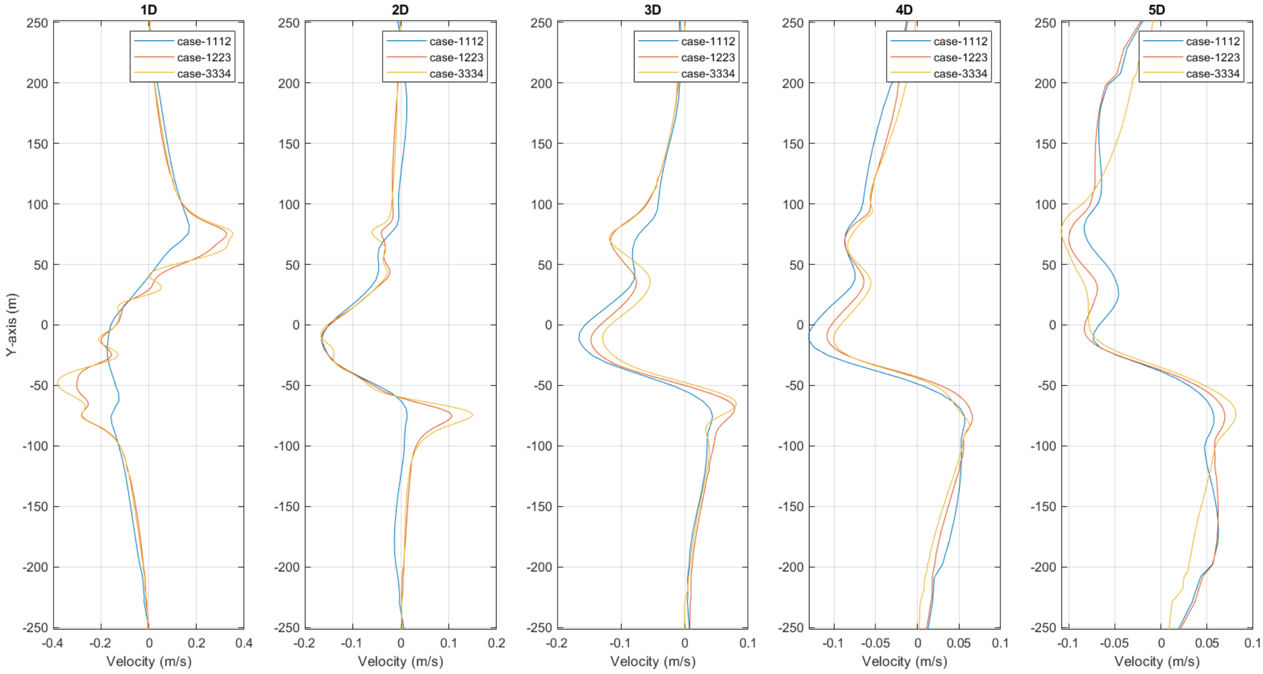


Figure 4.15: U_y for the test cases at different wake locations along a horizontal line

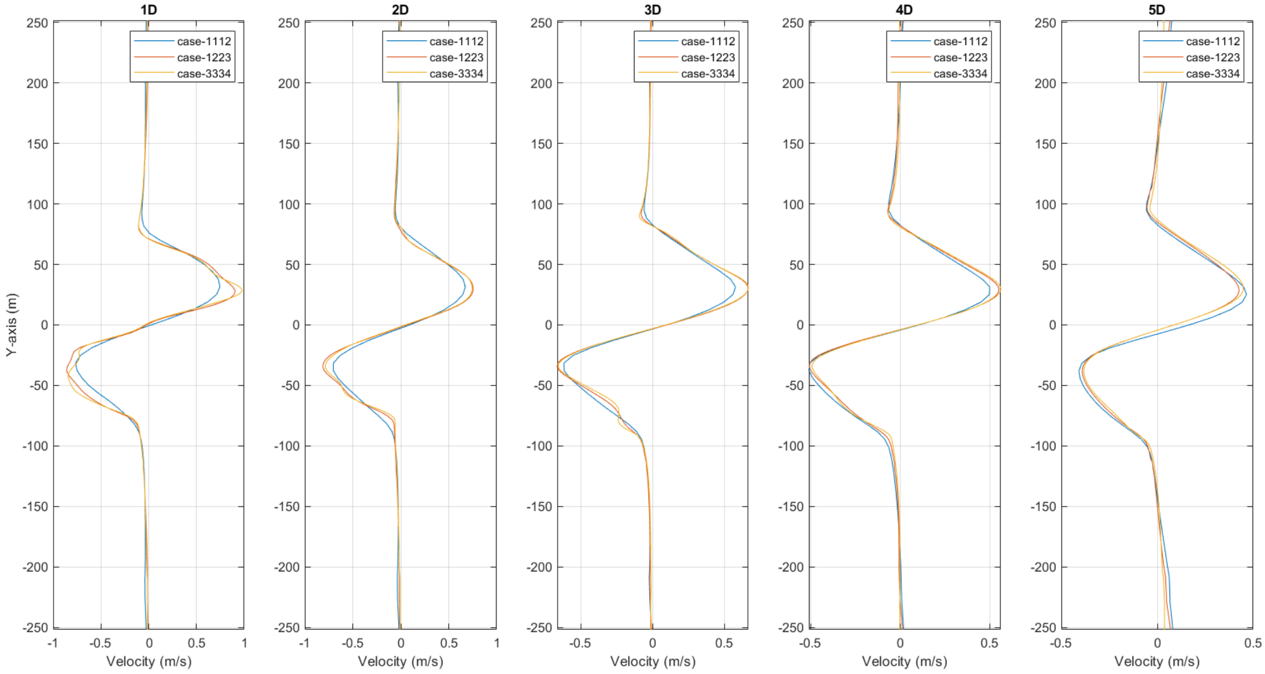


Figure 4.16: U_z for the test cases at different wake locations along a horizontal line

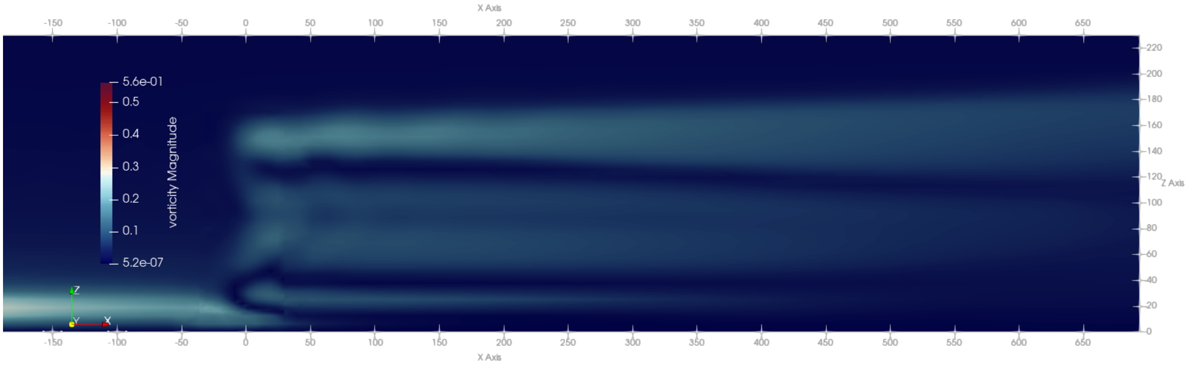


Figure 4.17: Vorticity for case-1112 at turbine mid-section plane as seen from side

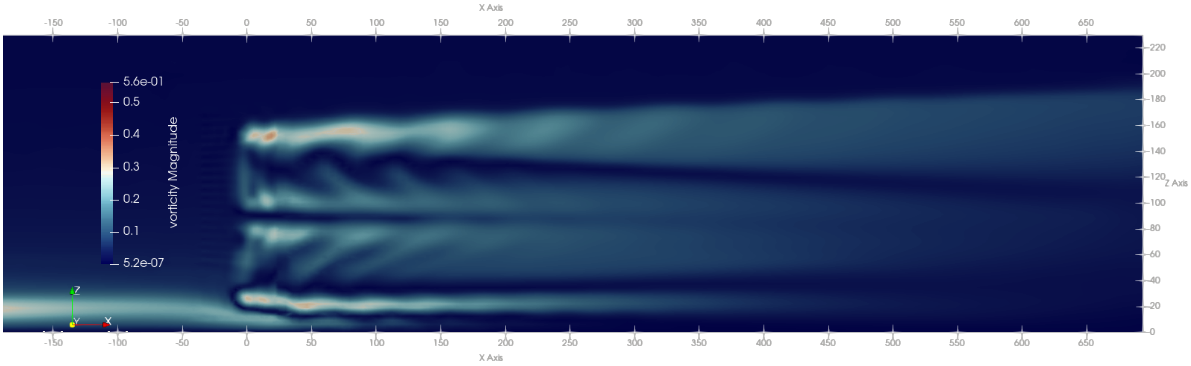


Figure 4.18: Vorticity for case-1223 at turbine mid-section plane as seen from side

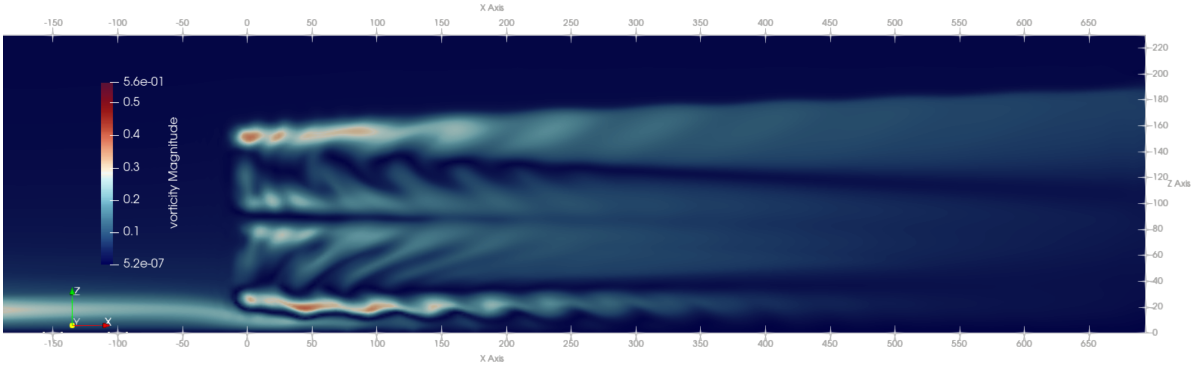


Figure 4.19: Vorticity for case-3334 at turbine mid-section plane as seen from side

Table 4.3: Wake velocity deficit at different wake location

Test cases	1D	2D	3D	4D	5D
case 1112	-3.60422	-3.62768	-3.47433	-3.29504	-3.08236
case 1223	-3.81809	-3.76049	-3.52471	-3.28115	-3.05438
case 3334	-3.81879	-3.78581	-3.55067	-3.29818	-3.07329

Apart from the wake characteristics, the following table 4.4 compares the turbine coefficients for different test cases. As the mesh size near the rotor decreases, the turbine performance decreases.

Table 4.4: Turbine performance for different test cases

Test cases	Power coefficient C_p	Thrust coefficient C_T	Torque coefficient C_q
case 1112	0.5491	0.7321	0.0732
case 1223	0.5162	0.7210	0.0688
case 3334	0.5145	0.7197	0.0686

Conclusion: The main purpose of the mesh dependence analysis was to observe, how much the turbine performance and wake velocities vary based on the mesh size. Case 1112 fails to capture accurate wake velocities. When comparing the other two cases, the finest mesh gives higher resolution in the near wake. But for the mid and far wake region, case 1223 and case 3334 give similar results. Turbine performance for both cases was found to be almost equal. While case 1223 took 18 hours with parallel computation (20 processes), case 3334 took almost 70 hours to run.

4.3. Effect of elasticity on wind turbine performance and wake region

This section compares the test cases for flexible turbines and their performance with the rigid turbine. For comparison purposes, the elastic or flexible turbine case is noted as EALM and the rigid turbine case is noted as RALM,

4.3.1. Effect on steady turbine

This section elaborates on the effect of elasticity on different aspects of the simulations of a steady non-moving wind turbine. For this, a total of 8 test cases were taken into consideration. Two hub wind speeds of 9 and 12 m/s, two types of mesh (fine and coarse), for each elastic and rigid simulation. The coarse mesh was included in the study to see if the developed structural model converges or diverges in the long run. Due to the unavailability of parallel processing, the fine mesh cases were only able to run for 150-190 seconds of simulation time, depending on turbine speed. The coarse mesh ran till the intended run time of 300 seconds, with converging results.

The mesh size is again the same as mentioned in section 4.2. The two mesh refinements 1223 (fine) and 1113 (coarse) are used (Here, case 1223 is referred to as fine mesh instead of medium, to emphasise more on the mesh size difference). The turbine tower is not included to run the simulation as long as possible. However, the logarithmic wind shear is used for inlet wind velocity.

Below rated speed - 9 m/s: T. Larsen et al., 2004 mentioned that the overall reduction of turbine rotor area due to deflection causes a reduction in turbine performance. However, for the current developed structural model, shear deformation in spanwise directions due to non-spanwise forces is neglected. The spanwise deflection is only governed by the spanwise forces. These deflections are very small to have a big impact on the rotor area. Here, the change in turbine performance is governed by the change in direction and location of applied ALM forces, rather than the changed rotor diameter. The following figures show the turbine power, torque and thrust coefficient over time.

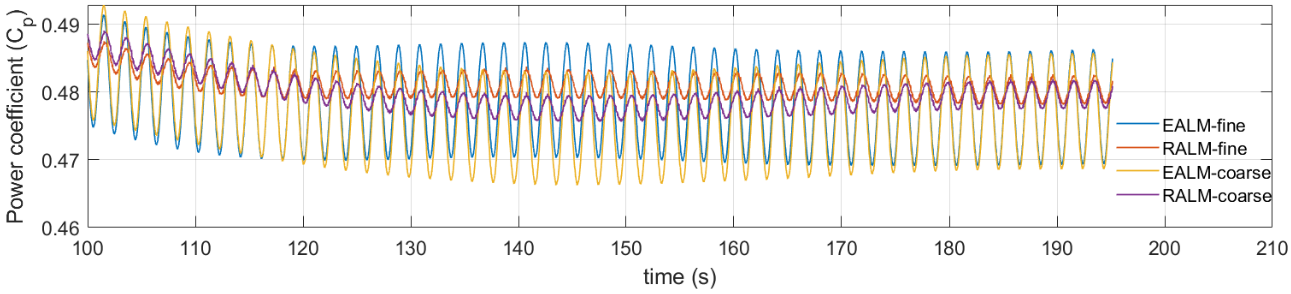


Figure 4.20: Power coefficient at 9m/s

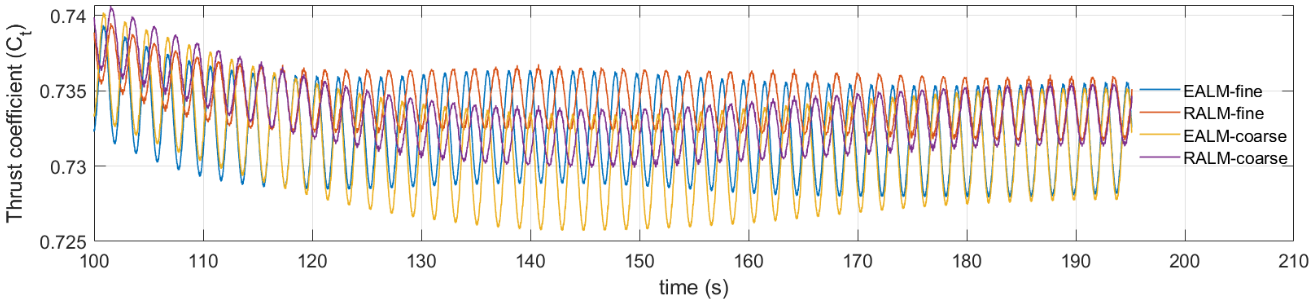


Figure 4.21: Thrust coefficient at 9m/s

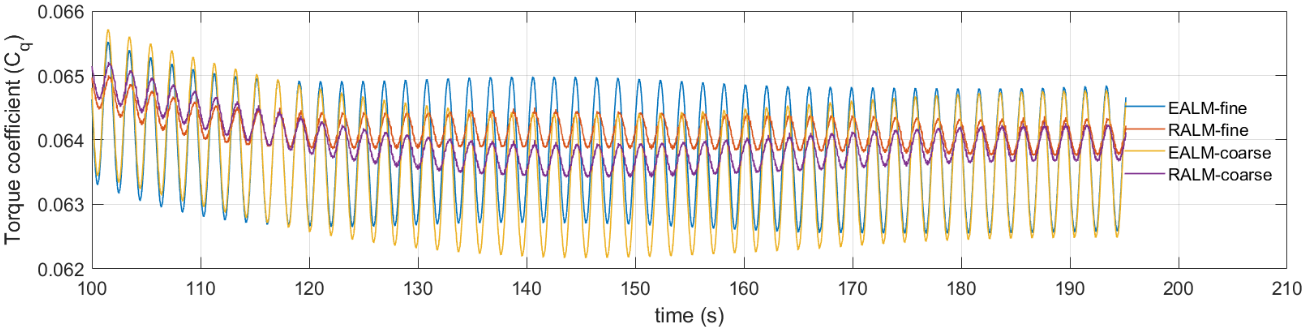


Figure 4.22: Torque coefficient at 9m/s

The elastic turbines have bigger amplitude for variations in power, thrust and torque when compared with rigid ones. These variations are not due to the reduced rotor area, but due to the dynamic response of blade deflections. As the blade velocity due to deflection increases, the relative wind speed increases or decreases depending on the deflection direction. This causes blade loads to oscillate depending on the dynamic deflection. This phenomenon can be seen in figure 4.8, where elastic turbine blades have oscillatory OOP forces and the rigid turbine blade has flat-lined loads.

One more phenomenon to be noticed is that the coefficient values for rigid and elastic turbines are not in the same phase, i.e. the peak occurs at different times. This is also the result of dynamic blade deflection. The table 4.6 compares the time-averaged coefficient values for all cases.

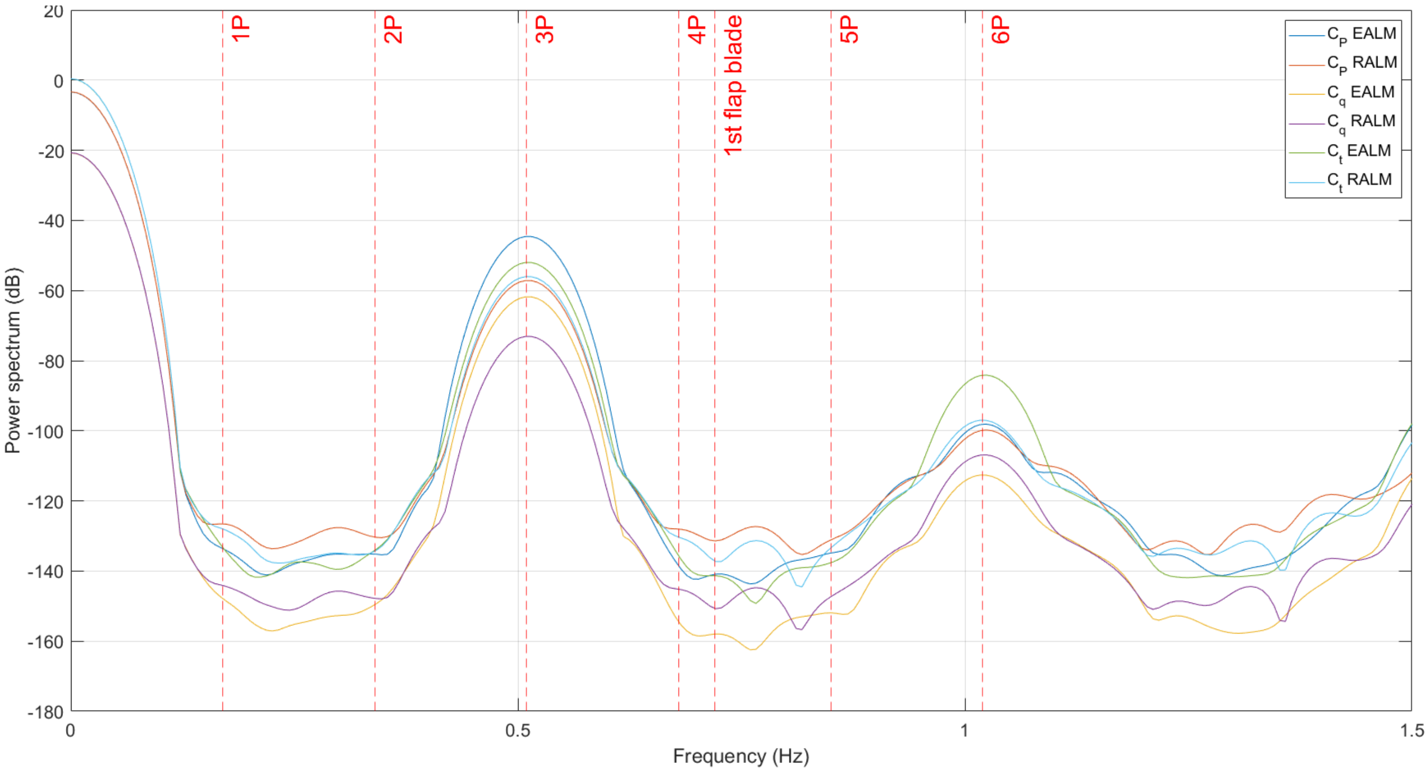


Figure 4.23: Turbine coefficients in frequency domain

The figure 4.23 shows the coefficients in the frequency domain. Because of shear, a peak is observed at 3P frequency. For elastic turbines, a small peak is present at the 1st blade flapwise frequency. Thus the blade's natural frequency affects the variation in turbine performance as well.

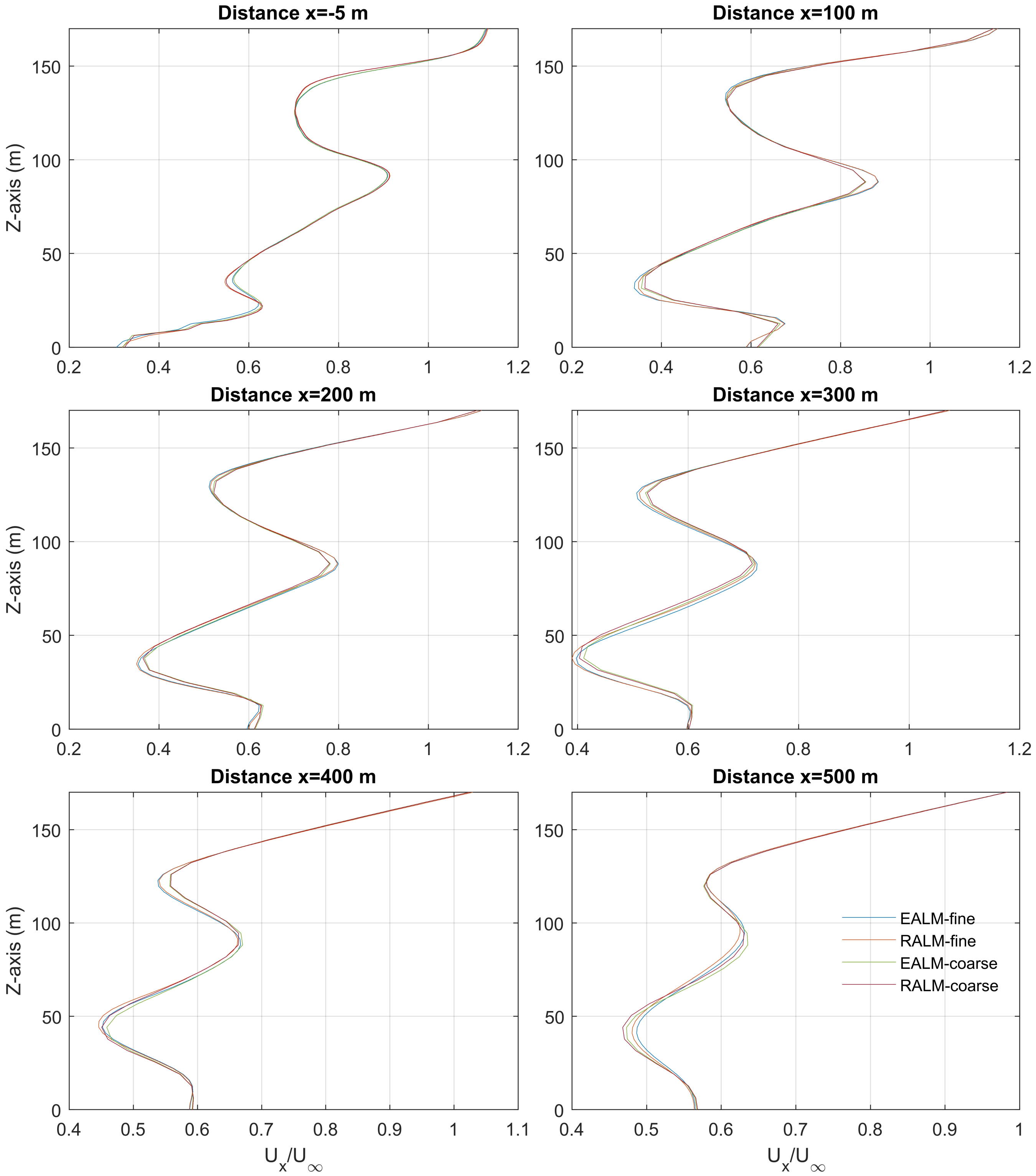


Figure 4.24: Vertical wind profile at different wake location, x=-5 being the rotor plane location

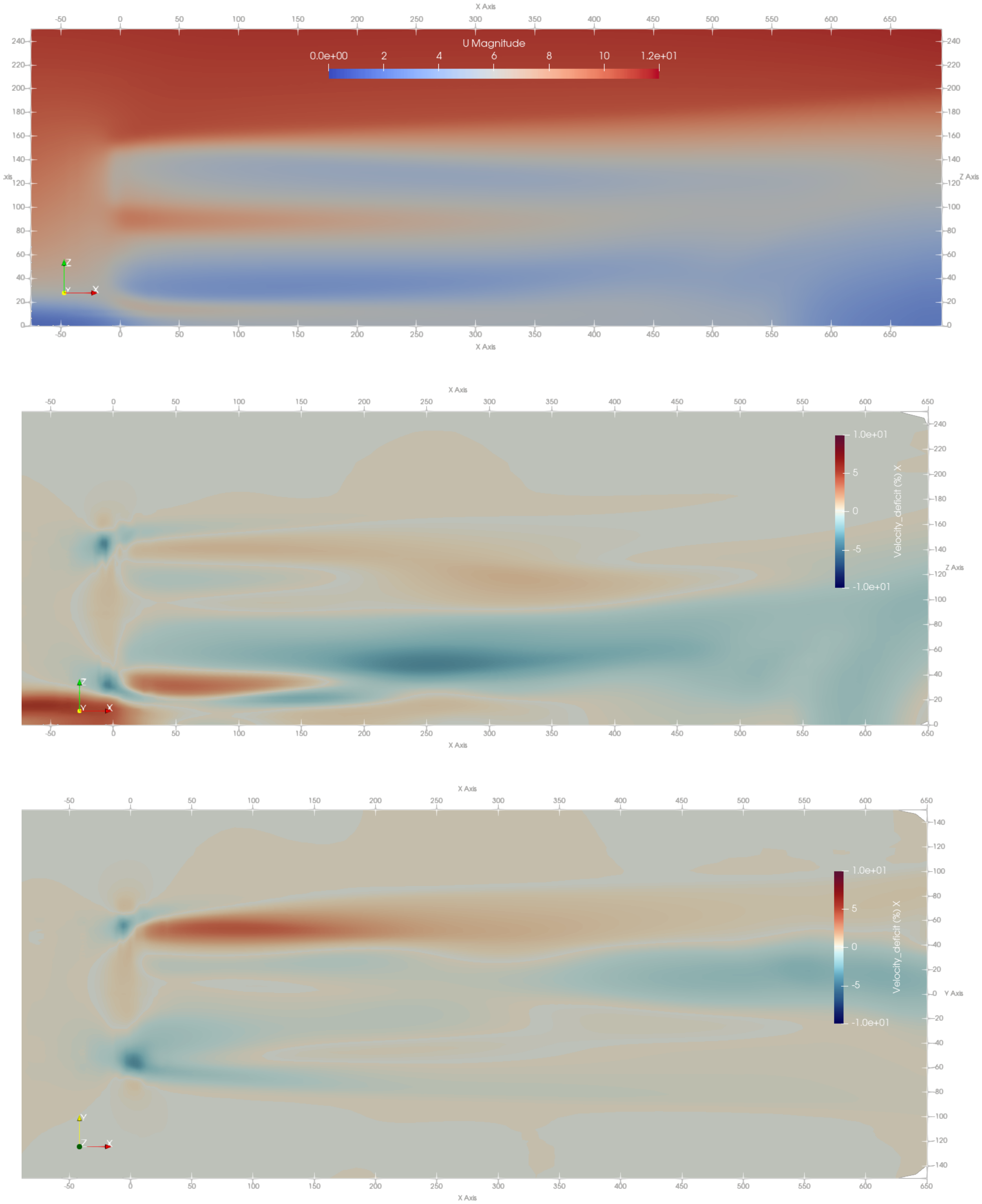


Figure 4.25: (a) Velocity profile of rigid turbine. Velocity difference of elastic turbine -(b) side view, (c) top view - for fine mesh

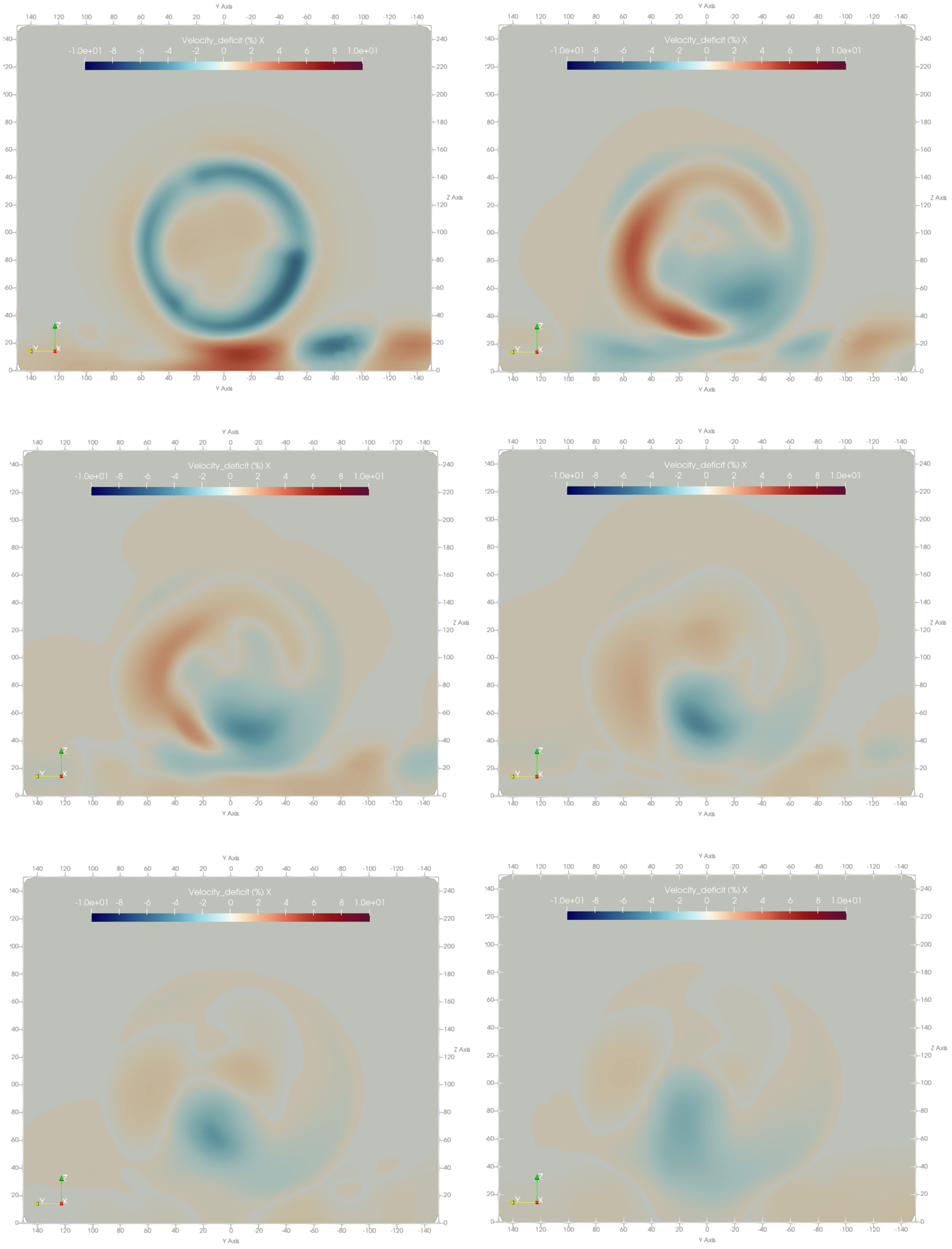


Figure 4.26: Velocity difference in planes parallel to rotor plane at different wake position (a) rotor plane (b) x=100m (c) x=200m (d) x=300m (e) x=400m (f) x=500m. From top to bottom and left to right, at time t=190 s, for fine mesh

Figure 4.24 shows the vertical wake profile at different wake locations The horizontal wake profile is given in appendix B. It can be seen that the EALM turbine wake has more velocity in some regions than the RALM turbine wake, and in some regions, it is reversed. To better visualize the effect of elasticity, the figure 4.25 shows the differences in velocity in the EALM turbine wake with respect to that of the RALM turbine. The difference is calculated using the formula:

$$Difference = \frac{(U_{RALM} - U_{EALM})100}{U_{RALM}} \tag{4.1}$$

where U_{RALM}, U_{EALM} are instantaneous wind velocity at the local point.

The velocity difference has both positive and negative values in the wake region. The difference also gets affected by the turbine rotation and shows helix-like rotation in the wake.

Above rated speed - 12 m/s: At higher wind speed, the deflection value increases hence its effects also get amplified. Figures 4.27, 4.28,4.29 illustrates variation in turbine performance at 12 m/s hub wind speed.

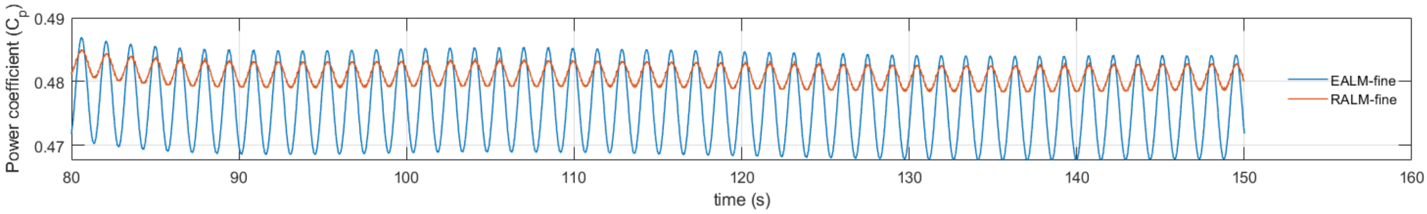


Figure 4.27: Power coefficient at 12m/s

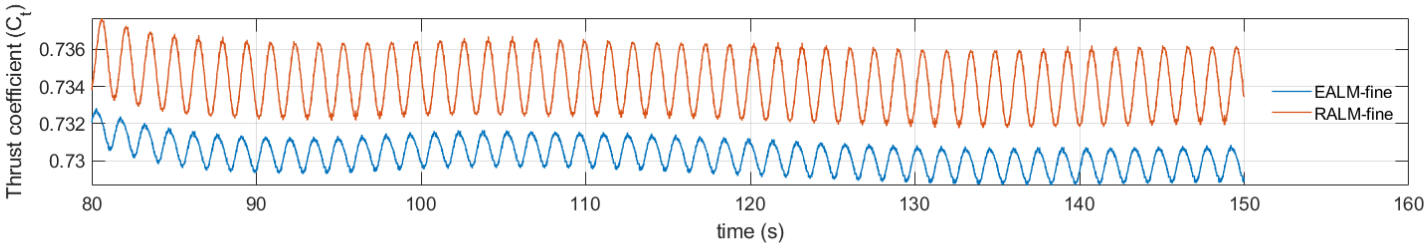


Figure 4.28: Thrust coefficient at 12m/s

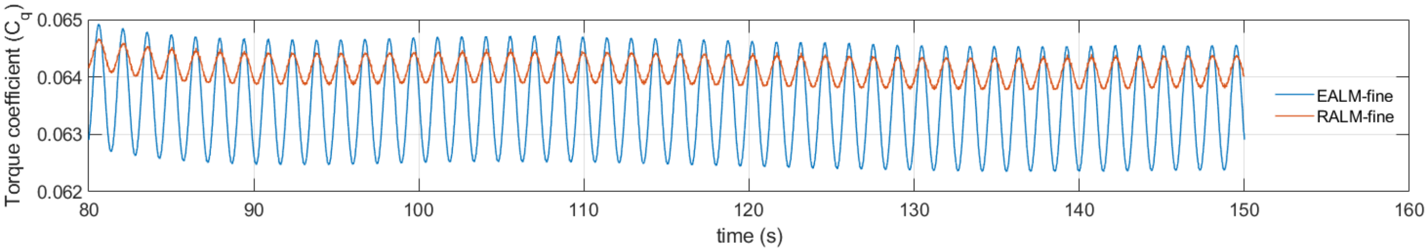


Figure 4.29: Torque coefficient at 12m/s

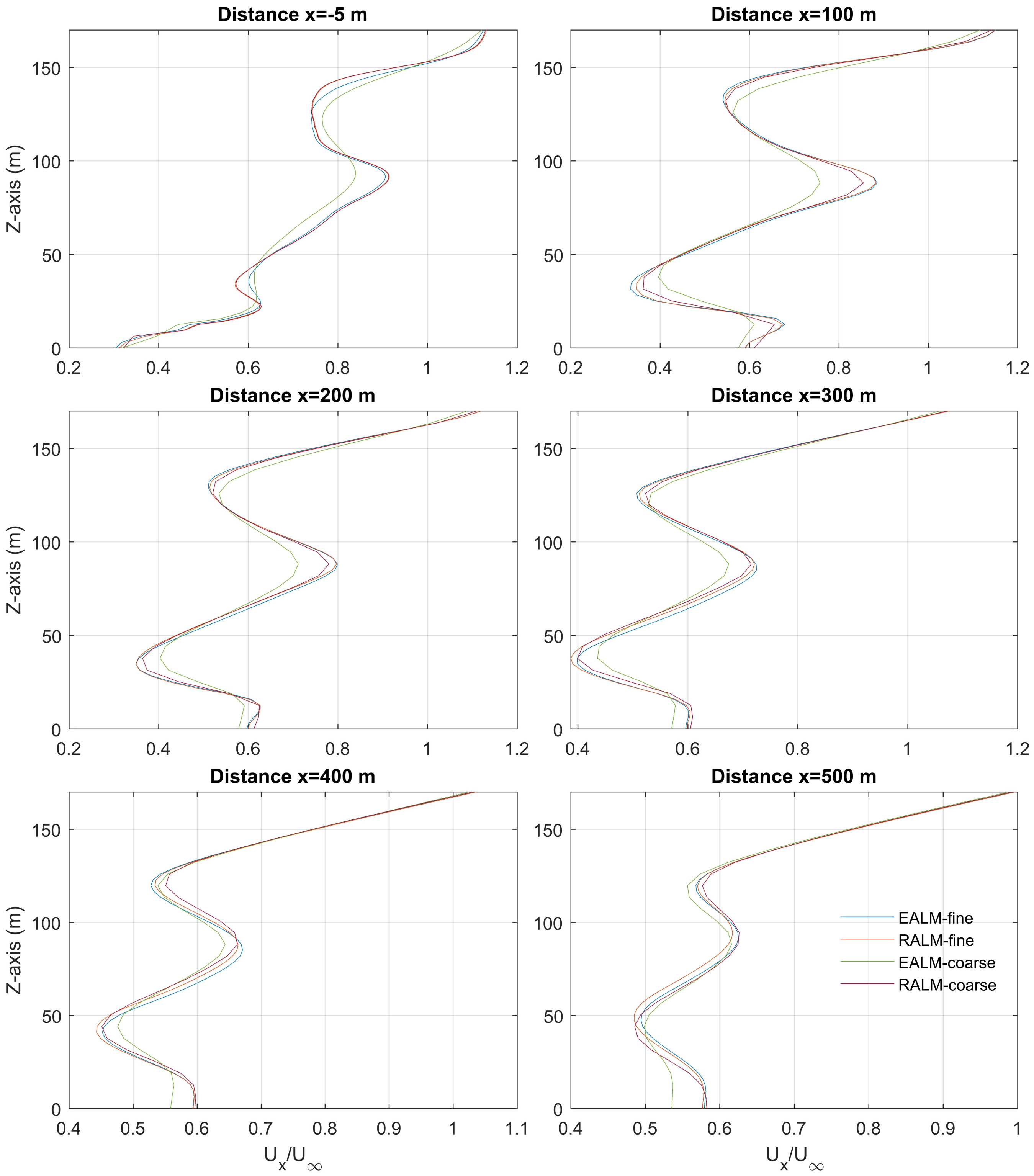


Figure 4.30: Vertical wind profile at different wake location

Table 4.5: Turbine performance summary for cases with fine mesh

Test cases	Power coefficient C_p	Thrust coefficient C_T	Torque coefficient C_q
EALM $9m/s$	0.4780	0.7320	0.0637
RALM $9m/s$	0.4807	0.7341	0.0641
EALM $12m/s$	0.4762	0.7302	0.0635
RALM $12m/s$	0.4808	0.7341	0.0641

Table 4.6 summarises the mean turbine performance for both inlet wind velocities.Not much of a difference can be seen in the overall turbine performance. A negligible drop in performance in the range of 1% is observed.

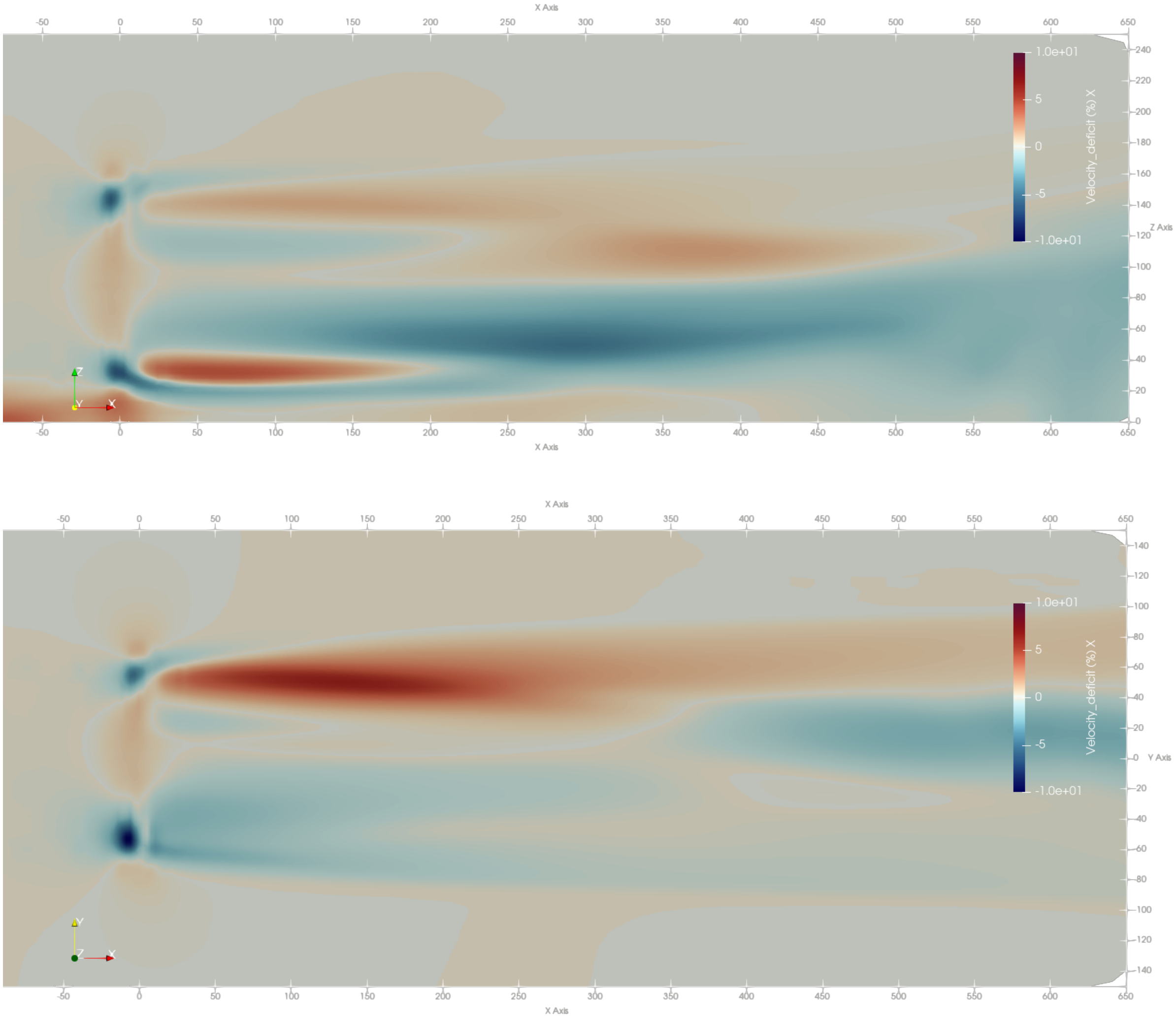


Figure 4.31: Velocity difference of elastic turbine -(a) side view, (b) top view, for fine mesh

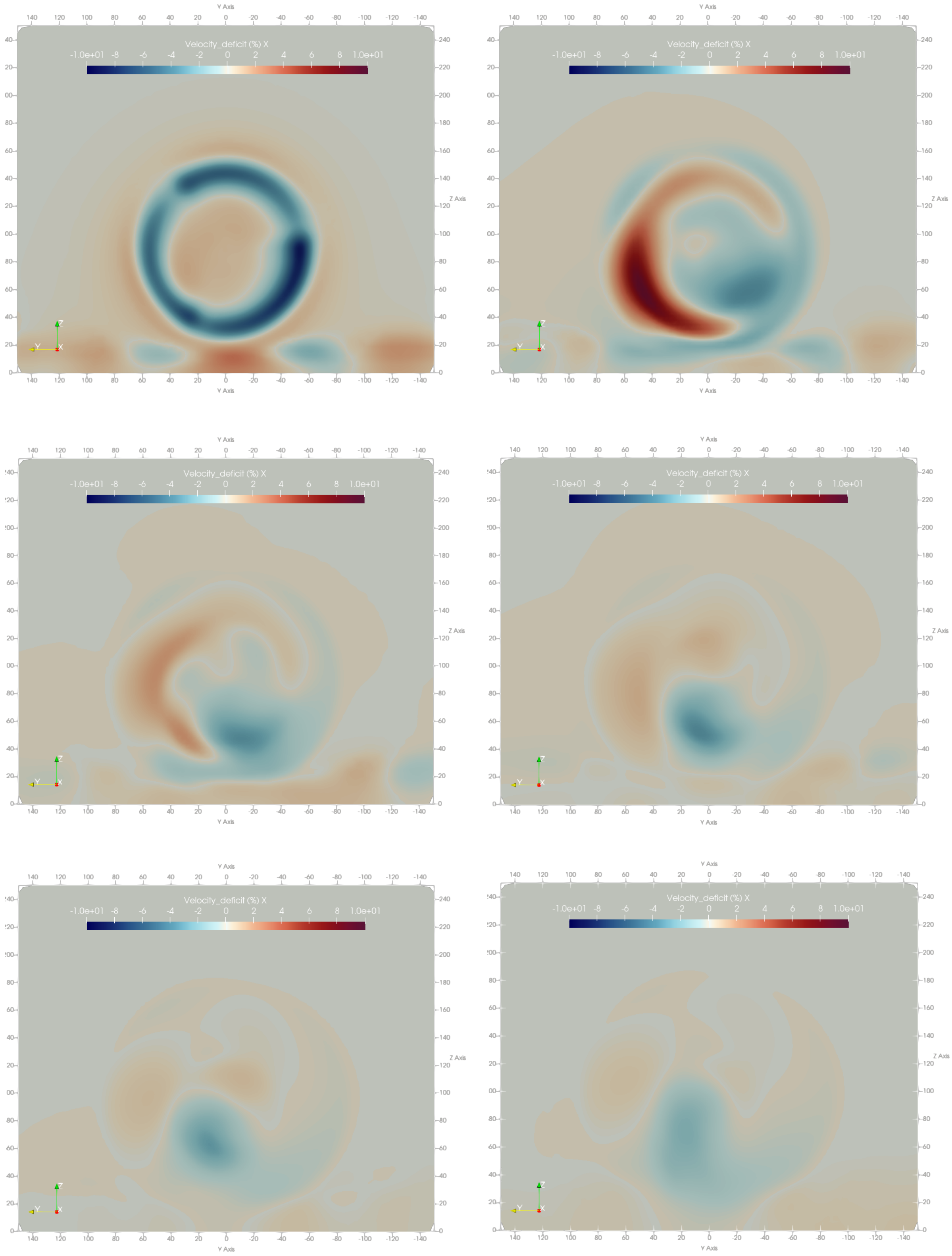


Figure 4.32: Velocity difference in planes parallel to rotor plane at different wake position (a) rotor plane (b) x=100m (c) x=200m (d) x=300m (e) x=400m (f) x=500m. From top to bottom and left to right, at time t=150 s, for fine mesh

However, apart from turbine performance, noticeable differences can be seen in the wake velocity profile. The figure 4.30 illustrates the wake velocity profile along the vertical direction in the mid-section of the wake. The difference gets broader. The figure B.11 shows again the velocity difference but now at 12 m/s hub wind speed. The percentage difference increases as well, as the wind speed and the deflection increase.

4.3.2. Effect on moving turbine

To see the effect of elasticity on a moving turbine, the turbine is rotated with a prescribed motion in surge and pitch direction. The M,C,K matrices are also rotated with the given rotation criteria to obtain the correct stiffness values.

case setup: The case is set up as same as the previous cases. But the turbine is moved with the pitching motion of amplitude one degree and surge motion of amplitude one meter simultaneously. Both with a frequency of 0.125 Hz. The point of rotation for pitching movement is the tower base. But the tower is not included in the simulation.

The fugure 4.33 shows the deflection values of the blades. Due to the prescribed motion, stiffness also changes depending on the rotor plane orientation.

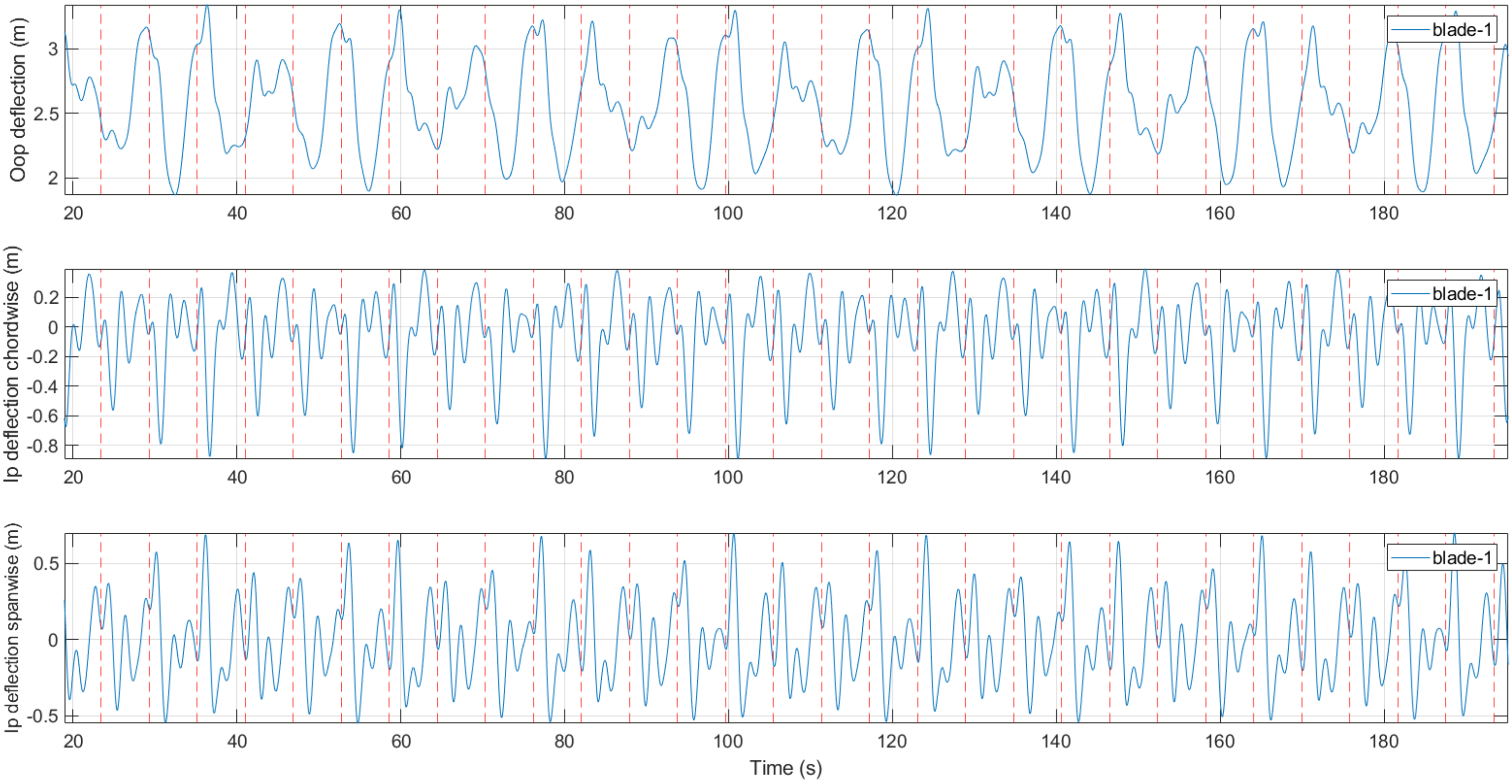


Figure 4.33: Blade tip deflection for moving turbine at 9 m/s hub speed for fine mesh

The turbine performance again showed a negligible difference when compared with the rigid turbine. But the overall turbine power increased when compared to the steady turbine in all cases.

Table 4.6: Time averaged turbine performance coefficients for cases with fine mesh

Test cases	Power coefficient C_p	Thrust coefficient C_T	Torque coefficient C_q
EALM $9m/s$	0.5040	0.7334	0.0672
RALM $9m/s$	0.5071	0.7355	0.0676
EALM $12m/s$	0.4969	0.7350	0.0663
RALM $12m/s$	0.5016	0.7390	0.0669

The remaining turbine performance and wake velocity plots are given in appendix B

5 Conclusion

This chapter summarizes the work carried out during this thesis work. After this, the main research questions are again looked upon and answered. Furthermore, the limitations and scope of future work are discussed to further improve upon the carried-out work.

5.1. Summary of the thesis

The major goal of this thesis is to explore the effect of elasticity in the different aspects of ALM-based wind turbine simulations. To study the effects, at first, a linear structural dynamic module was developed in MATLAB. This module is based on the spinning elements method, first discussed in Leung and Fung, 1988 and later applied for the wind turbine by Sajeer et al., 2020.

The developed module is first validated against the Bmodes for the natural frequency and mode shapes of the blades. After the validation, the TurbinesFOAM library was successfully modified to integrate the structural module by calling the MATLAB code while simulations were carried out. The link between MATLAB and TurbinesFOAM is a two-way link. Where the aerodynamic forces and deflection are exchanged.

After the successful integration of OpenFOAM with MATLAB, the added module in OpenFOAM is again compared with the results found in OpenFAST. The final conclusion about the comparison is discussed later in this chapter.

Later, the effect of elasticity on the simulation was explored. The major focus was on the turbine performance and the wake region. Negligible differences were found in terms of time-averaged turbine performance when comparing elastic turbines with rigid ones. However, conceivable differences were found in the wake region of the elastic turbine.

5.2. Research questions

The section answers the research questions which were presented in chapter 1. Here, all the sub-questions are discussed which autonomously answers the main research question.

- **How can structural dynamics of wind turbines be modelled using a simple yet accurate manner?**

  There are many structural models available in literature as described in 2. The models varied on different criteria i.e. element type, linearity, and beam type. All of the models have their own benefits. However, the major goal of this thesis has been to study the effect of elasticity in the CFD simulation. Developing a new structural model from scratch can prove redundant and time-consuming. Hence, the structural module was dependent on the already developed framework in the literature. Moreover, the model needed to be accurate and easy to implement in terms of coding.

  The selected model by Sajeer et al., 2020, derives the M,C,K matrices in the blade frame of reference. To further expand this model for the whole wind turbine, with 3 rotating blades and one stationary tower, global M,C,K matrices are assembled at each time step of the simulation. To extend this model to moving turbines, the global M,C,K matrices are rotated according to turbine motions.

  Finally, Newmark's method was used to solve the equation of motion to get the deflection values. However, the application of the model is only limited to small and linear deformation. The model will under-predict the structural deformation for very high wind speeds.

- **How to verify the developed structural model?**

  The verification process was carried out in a two-step process. At first, the developed MATLAB code for the blade's M,C,K matrices was used to determine the natural frequency and mode shapes of the blade. The frequencies and mode shapes were verified with Bmodes. The mode shapes of the model exactly matched that of BModes at different rotational speeds. While the natural frequency deviated by an acceptable margin with varying rotational speed.

  To further verify the global matrices consisting of three rotating blades and one stationary tower, ElastoDyn/OpanFAST was used. The deflection found in OpenFOAM was lesser than that found in Elastodyn due to the under-prediction of loads in OpenFOAM. However, applying the loads obtained from Elastodyn on the developed model gave comparable deflection values. Thus, the developed structural model was verified against the widely used tools in the field of wind turbine simulation.

- **How to strongly couple the ALM-based wind turbine simulation with the dynamic structural simulations?**

  The coupling between the structural module and the TurbinesFOAM was done within the already developed framework of the PIMPLE algorithm by Pericas, 2022. Three sub-iterations were carried out (and were also sufficient) within a time step to converge the forces applied by deflection velocity on the surrounding fluid.

  During each iteration, the aerodynamic forces from TurbinesFOAM are applied to the FEM element in MATLAB. And the deflections from MATLAB are applied on AEs of turbinesFOAM. These two-way operations are performed multiple times within a time step to achieve temporal convergence.

- **What is the effect of structural elasticity on the simulation?**

  The focal points of the effect of elasticity were the wake velocity profiles and blade loading. The elasticity caused the wake to divide into two different regions (positive and negative velocity difference) when compared to the rigid turbine. The percentage difference also increased with the inflow wind velocity. At 12 m/s for a steady wind turbine, the instantaneous mid-wake (4-5 D) wind velocity of the elastic turbine varied from -8% to 6% that of a rigid turbine. At higher wind speed and deflection values, the effect of elasticity on wake velocity cannot be ignored. It was difficult to quantify the change in the wake in the form of a conventional wake velocity deficit. The reason is the presence of both regions as mentioned above.

  However, the turbine performance was not affected by the added deflection severely. When comparing the power output, at 12 m/s, the elastic turbine showed a one percentage reduction on average when compared with the rigid turbine. Comparing that with openFAST, the percentage drop was around five per cent. The thrust and torque also showed similar trends. Nonetheless, with the negligible drop in performance, it will be worth exploring the effect of elasticity in multiple wind turbine configurations or in a wind farm.

  Commenting on the blade loads, the elastic turbine showed oscillatory blade forces as opposed to the almost constant force of the rigid turbine blade. Comparing the blade loads with the OpenFAST, ALM-based simulation gave a very different pattern of blade loading (figure 4.9).

5.3. Limitations and future scope

The current work in coupling aero-elasticity with ALM-based turbine simulations should be looked upon as the starting stone in the field. The carried out work enables future scope in the structural aspects of simulations such as blade stresses, fatigue and life-cycle. The following topics elaborate more on the limit of current work and the scope for future work.

- **Converting** the elasticity code from MATLAB to open foam's own C++ environment would be the next step for improvement. As current libraries require a MATLAB subscription, it defies the ultimate goal of this thesis to develop an open-source tool. However, this step will require in-depth knowledge of C++ and its utilities. The matrix operations and other mathematical operations such as integration, derivation, finding eigenfrequencies etc. can be performed very easily in MATLAB with the use of one command. This is not possible in C++; separate functions will need to be developed just for such operations. Switching from MATLAB to C++ will also enable parallel computation.

- As mentioned in the literature review, **Non-linearity** has a great impact on the accuracy of large deflection. Adding a source term for non-linearity in the derivation of the matrices of spinning elements will extend the use of this FEM model in the simulation of wind turbines, in high wind speed and turbulence.

- **Aerodynamic damping** is not modelled in the developed FEM model, due to its complex nature. Adding more damping in the model will reduce the initial oscillations with big amplitude, as observed in figure 4.10.

- **Control** is not included in the simulation as of now. The blade angle is taken to be zero in all cases. Adding varying blade angles with respect to incoming wind velocity will ultimately change the stiffness and thus the deflection values.

- **Verifying** the input model of NREL 5MW turbine is essential. The current input model overpredicts the torque and power values while underpredicting the thrust value when compared with OpenFAST. A validated input model will increase the accuracy of the simulation.

- Adding a **drive-train** elasticity instead of considering a rigid nacelle (Which is assumed in this model) will reduce the loads on the turbine tower and ultimately the deformation and stress of the tower.

References

Ahlström, A. (2006). Influence of wind turbine flexibility on loads and power production. *Wind Energy*, *9*(3), 237–249. https://doi.org/https://doi.org/10.1002/we.167

Ahmed A., S. (2005). *Dynamics of multibody systems.* (Vol. 3rd ed). Cambridge University Press. https://search-ebscohost-com.tudelft.idm.oclc.org/login.aspx?direct=true&db=nlebk&AN=138999&site=ehost-live&authtype=sso&custid=s1131660

Alam, J. M., & Fitzpatrick, L. P. J. (2017). Large eddy simulation of urban boundary layer flows using a canopy stress method.

Bachant, P., Goude, A., & Wosnik, M. (2018). Actuator line modeling of vertical-axis turbines.

Bathe, K. (2006). *Finite element procedures*. Prentice Hall. https://books.google.nl/books?id=rWvefGICfO8C

Beamdyn inputs from sectional beam properties. (2019). https://www.nrel.gov/wind/nwtc/assets/downloads/BeamDyn/BeamDyn_Manual.pdf

Bir, G., & Jonkman, J. (2007). Aeroelastic instabilities of large offshore and onshore wind turbines. *Journal of Physics: Conference Series*, *75*(1), 012069. https://doi.org/10.1088/1742-6596/75/1/012069

Bir, G. S. (2007). *User's guide to bmodes*. NREL. 1617, Cole Blvd, Golden, CO 80401. https://www.nrel.gov/wind/nwtc/assets/pdfs/bmodes.pdf

Burton, T., Jenkins, N., Bossanyi, E., Sharpe, D., & Graham, M. (2021). John Wiley Sons. https://app.knovel.com/hotlink/toc/id:kpWEHE0022/wind-energy-handbook/wind-energy-handbook

Chandrupatla, T., & Belegundu, A. (2011). *Introduction to finite elements in engineering*. Prentice Hall. https://books.google.nl/books?id=tOTAkQEACAAJ

Chen, C., & Duffour, P. (2018). Modelling damping sources in monopile-supported offshore wind turbines. *Wind Energy*, *21*(11), 1121–1140. https://doi.org/https://doi.org/10.1002/we.2218

Crespo, A., Hernández, J., & Frandsen, S. (1999). Survey of modelling methods for wind turbine wakes and wind farms. *Wind Energy*, *2*(1), 1–24. https://doi.org/https://doi.org/10.1002/(SICI)1099-1824(199901/03)2:1<1::AID-WE16>3.0.CO;2-7

Friedmann, P. P. (1976). Aeroelastic modeling of large wind turbines. *Journal of the American Helicopter Society*, *21*(4), 17–27. https://doi.org/doi:10.4050/JAHS.21.17

Griffith, D. T. (2009). Structural dynamics analysis and model validation of wind turbine structures. *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No*, 2408.

Hansen, M., Sørensen, J., Voutsinas, S., Sørensen, N., & Madsen, H. (2006). State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, *42*(4), 285–330. https://doi.org/https://doi.org/10.1016/j.paerosci.2006.10.002

Hodges, D. H., & Dowell, E. H. (1974). *Nonlinear equations of motion for the elastic bending and torsion of twisted nonuniform rotor blades* (tech. rep.).

IRENA. (2021). Renewable capacity statistics 2021.

Jha, P. K., Churchfield, M. J., Moriarty, P. J., & Schmitz, S. (2014). Guidelines for Volume Force Distributions Within Actuator Line Modeling of Wind Turbines on Large-Eddy Simulation-Type Grids [031003]. *Journal of Solar Energy Engineering*, *136*(3). https://doi.org/10.1115/1.4026252

Jonkman, J., Butterfield, S., Musial, W., & Scott, G. (2009). Definition of a 5-mw reference wind turbine for offshore system development. https://doi.org/10.2172/947422

Kang, H., Chang, C., Saberi, H., & Ormiston, R. A. (2014). Assessment of beam and shell elements for modeling rotorcraft blades. *Journal of Aircraft*, *51*(2), 520–531. https://doi.org/10.2514/1.C032217

Kee, Y.-J., & Shin, S.-J. (2015). Structural dynamic modeling for rotating blades using three dimensional finite elements. *Journal of Mechanical Science and Technology*, *29*(4), 1607–1618. https://doi.org/10.1007/s12206-015-0332-6

Larsen, G. C., Madsen, H. A., Thomsen, K., & Larsen, T. J. (2008). Wake meandering: A pragmatic approach. *Wind Energy*, *11*(4), 377–395. https://onlinelibrary.wiley.com/doi/abs/10.1002/we.267

Larsen, G., Madsen Aagaard, H., Bingöl, F., Mann, J., Ott, S., Sørensen, J., Okulov, V., Troldborg, N., Nielsen, N., Thomsen, K., Larsen, T., & Mikkelsen, R. (2007). *Dynamic wake meandering modeling*. Risø National Laboratory.

Larsen, T., Hansen, A., & Buhl, T. (2004). Aeroelastic effects of large blade deflections for wind turbines [Special topic conference: The science of making torque from wind ; Conference date: 19-04-2004 Through 21-04-2004]. *Proceedings*, 238–246.

Leishman, J. G., & Beddoes, T. S. (1989). A semiempirical model for dynamic stall. *Journal of the American Helicopter Society*. https://doi.org/doi:10.4050/JAHS.34.3.3

Leung, A., & Fung, T. (1988). Spinning finite elements. *Journal of Sound and Vibration*, *125*(3), 523–537. https://doi.org/https://doi.org/10.1016/0022-460X(88)90259-3

Liu, Y., Xiao, Q., Incecik, A., & Peyrard, C. (2019). Aeroelastic analysis of a floating offshore wind turbine in platform-induced surge motion using a fully coupled cfd-mbd method. *Wind Energy*, *22*(1), 1–20. https://doi.org/https://doi.org/10.1002/we.2265

Logan, D. L. (2000). *A first course in the finite element method using algor* (2nd). Brooks/Cole Publishing Co.

Ma, Z., Lei, L., & Zeng, P. (2020). The aero-elastic-wake coupling behavior for a two-wind-turbines case with power control process. *IOP Conference Series: Materials Science and Engineering*, *825*(1), 012003. https://doi.org/10.1088/1757-899X/825/1/012003

Ma, Z., Zeng, P., & Lei, L. (2019). Analysis of the coupled aeroelastic wake behavior of wind turbine. *Journal of Fluids and Structures*, *84*, 466–484. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2018.09.001

Malveiro, C. D. S. P. (2022). *Aerodynamics of a 15-MW and a 5-MW Reference Wind Turbine Using Varying Fidelity Tools* (Master's thesis). TU Delft. the Netherlands.

Martínez-Tossas, L. A., Churchfield, M. J., & Meneveau, C. (2017). Optimal smoothing length scale for actuator line models of wind turbine blades based on gaussian body force distribution. *Wind Energy*, *20*(6), 1083–1096. https://doi.org/https://doi.org/10.1002/we.2081

Morin, D. J. (2007). *Introduction to classical mechanics: With problems and solutions* (xvii, 719 pages). Cambridge University Press. https://tudelft.on.worldcat.org/oclc/191241203

Muscari, C., Giordani, R., & Schito, P. (2023). On wind turbine structural stiffness influence on wake flow. *Journal of Fluids and Structures*, *118*, 103862. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2023.103862

Naunidh, M. (2019). Reduced order wind turbine aeroelastic modelling for condition monitoring fault detection. https://doi.org/http://resolver.tudelft.nl/uuid:69a1c261-39ad-4b6b-b280-715f1fe94a1e

Pericas, P. F. (2022). *CFD Simulation of a Floating Wind Turbine with OpenFOAM* (Master's thesis). TU Delft,Universitat Politècnica de Catalunya. the Netherlands.

Sajeer, M. M., Mitra, A., & Chakraborty, A. (2020). Spinning finite element analysis of longitudinally stiffened horizontal axis wind turbine blade for fatigue life enhancement. *Mechanical Systems and Signal Processing*, *145*, 106924. https://doi.org/https://doi.org/10.1016/j.ymssp.2020.106924

Sayed, M., Lutz, T., Krämer, E., Shayegan, S., & Wüchner, R. (2019). Aeroelastic analysis of 10 mw wind turbine using cfd–csd explicit fsi-coupling approach. *Journal of Fluids and Structures*, *87*, 354–377. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2019.03.023

Sheng, W., Galbraith, R. A. M., & Coton, F. N. (2008). A Modified Dynamic Stall Model for Low Mach Numbers [031013]. *Journal of Solar Energy Engineering*, *130*(3). https://doi.org/10.1115/1.2931509

Shives, M., & Crawford, C. (2013). Mesh and load distribution requirements for actuator line cfd simulations. *Wind Energy*, *16*(8), 1183–1196. https://doi.org/https://doi.org/10.1002/we.1546

Snel, H. (1998). Review of the present status of rotor aerodynamics. *Wind Energy*, *1*(S1), 46–69. https://doi.org/https://doi.org/10.1002/(SICI)1099-1824(199804)1:1+<46::AID-WE3>3.0.CO;2-9

Sørensen, J. N., & Shen, W. Z. (2002). Numerical Modeling of Wind Turbine Wakes. *Journal of Fluids Engineering*, *124*(2), 393–399. https://doi.org/10.1115/1.1471361

Spalart, P. (1999). Strategies for turbulence modelling and simulations. In W. Rodi & D. Laurence (Eds.), *Engineering turbulence modelling and experiments 4* (pp. 3–17). Elsevier Science Ltd. https://doi.org/https://doi.org/10.1016/B978-008043328-8/50001-1

Strickland, J. H., Webster, B. T., & Nguyen, T. (1980). Vortex model of the darrieus turbine: An analytical and experimental study. https://www.osti.gov/biblio/5597154

Strømmen, E. (2013). *Structural dynamics*. Springer International Publishing. https://books.google.nl/books?id=Lcr-AAAAQBAJ

Troldborg, N. (2009). Actuator line modeling of wind turbine wakes.

Uchida, T. (2020). Effects of inflow shear on wake characteristics of wind-turbines over flat terrain. *Energies*, *13*(14). https://doi.org/10.3390/en13143745

Vermeer, L., Sørensen, J., & Crespo, A. (2003). Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, *39*(6), 467–510. https://doi.org/https://doi.org/10.1016/S0376-0421(03)00078-2

Wang, Q., Sprague, M. A., Jonkman, J., Johnson, N., & Jonkman, B. (2017). Beamdyn: A high-fidelity wind turbine blade solver in the fast modular framework. *Wind Energy*, *20*(8), 1439–1462. https://doi.org/https://doi.org/10.1002/we.2101

Wilcox, D. (2006). *Turbulence modeling for cfd (third edition) (hardcover)*.

Zheng, J., Wang, N., Wan, D., & Strijhak, S. (2023). Numerical investigations of coupled aeroelastic performance of wind turbines by elastic actuator line model. *Applied Energy*, *330*, 120361. https://doi.org/https://doi.org/10.1016/j.apenergy.2022.120361

.

A

Appendix A

In equation 3.47, the different elemental matrices, which are combined and then simplified from the respective kinetic and potential energy terms, are described as per the following equations.

$$\begin{aligned}
\text{Damping matrix } [C(t)] &= [C_s] + 2\,[G_\Omega] \\
\text{Where, Gyroscopic damping matrix } [G_{\Omega_e}] &= \int_0^L H^T\,[\bar{g}]\,H\,dx \\
\text{Mass matrix } [M_e] &= \int_0^L \phi^T\,[\bar{m}]\,\phi\,dx \\
\text{Stiffness matrix } [K(t)] &= [K_{E_e}] + [K_{C_e}] + \left[K_{g_e}\right] - [K_{\Omega_e}]
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\text{Where, Spinning stiffness matrix } [K_{\Omega_e}] &= \int_0^L H^T\,\left[\bar{k}_\Omega\right]H\,dx \\
\text{Element stiffness matrix } [K_{E_e}] &= \int_0^L \left[d\phi_K\right]^T\,\left[\bar{k}\right]\,\left[d\phi_K\right]\;dx \\
\text{Centrifugal stiffness matrix } [K_{C_e}] &= \int_0^L F_c(x)\left[d\phi\right]^T\left[d\phi\right]\;dx \\
\text{Gravitational stiffness matrix } \left[K_{g_e}\right] &= \int_0^L F_g(x)\left[d\phi\right]^T\left[d\phi\right]\;dx
\end{aligned} \tag{A.2}$$

Here,

$$\begin{aligned}
[\bar{m}] &= diag\left[\rho A(x)\ \rho A(x)\ \rho A(x)\ \rho I_p(x)\right] \\
\left[\bar{k_{\Omega}}\right] &= \rho A(x)\,[R]\,[\Omega]^T\,[\Omega]\,[R]^T \\
[\bar{g}] &= \rho A(x)\,[R]\,[\Omega]\,[R]^T \\
\left[d\phi_k\right] &= diag\left[\frac{d}{dx}\ \frac{d^2}{dx^2}\ \frac{d^2}{dx^2}\ \frac{d}{dx}\right]\left[\phi\right] \\
\left[d\phi\right] &= diag\left[0\ \frac{d}{dx}\ \frac{d}{dx}\ 0\right]\left[\phi\right] \\
\left[\bar{k}\right] &= diag\left[EA(x)\ EI_y(x)\ EI_z(x)\ GJ(x)\right]
\end{aligned} \tag{A.3}$$

For an element with two nodes,

$$\begin{aligned}
\text{Centrifugal force at location x } F_c &= (m_1 + (m_2 - m_1)\frac{x}{L})\Omega^2(L - x + r_i) + Frt \\
\text{Mass density } \rho A(x) &= (m_1 + (m_2 - m_1)\frac{x}{L})
\end{aligned} \tag{A.4}$$

In the above equations,r_i is the perpendicular distance of the element root from the axis of rotation. Frt is the centrifugal force acting on the tip of the element. m_1 and m_2 are the linearly varying mass densities at nodes 1 and 2, respectively. The other varying structural properties can also be obtained in the same manner. $EA(x), EI_y(x), EI_z(x), GJ(x)$, are spanwise varying axial, flapwise, edgewise and torsional stiffnesses. $I_p(x)$ is varying polar moment of inertia of blade cross-section in the same manner.

B

Appendix B

The presented figures are for cases with fine mesh unless mentioned.

B.1. Steady turbine

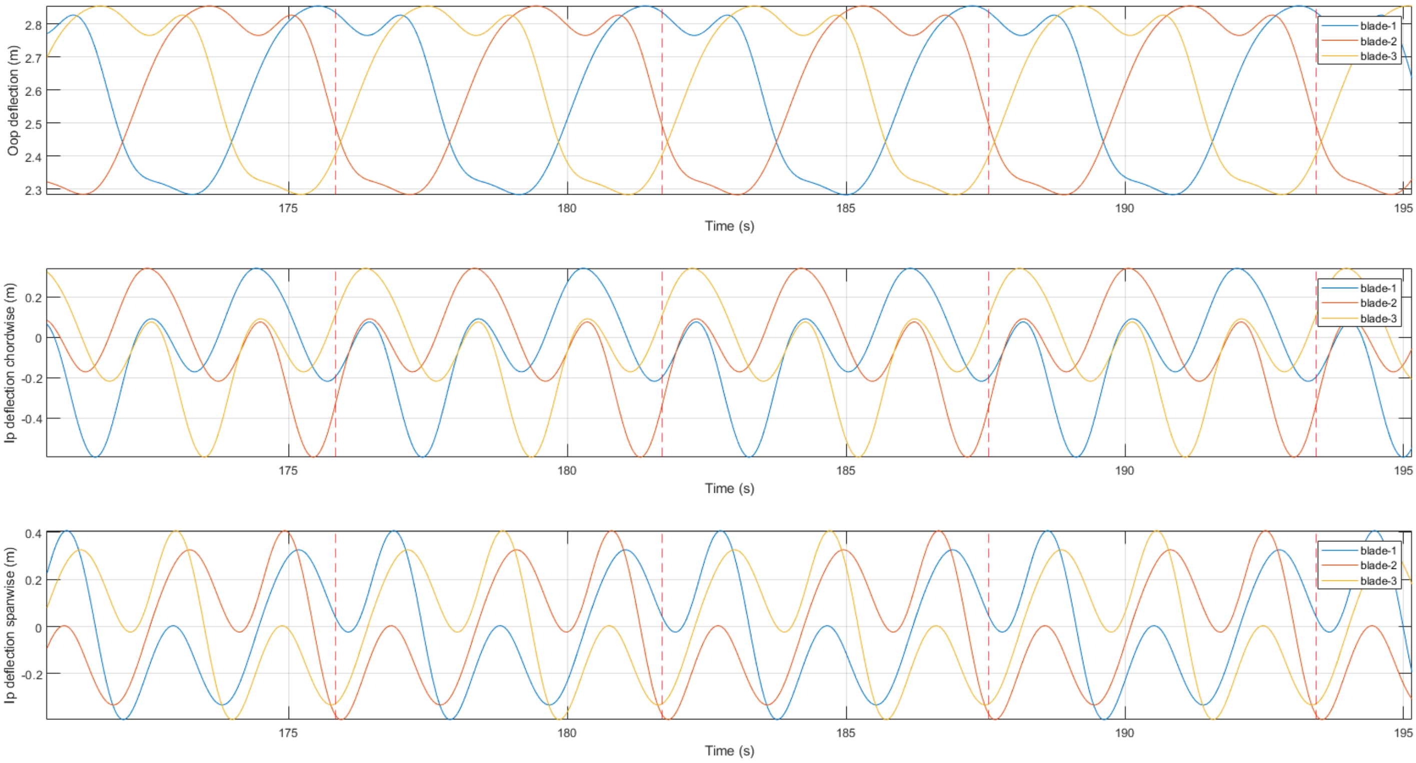


Figure B.1: Blade tip deflection for the steady turbine at 9 m/s hub speed

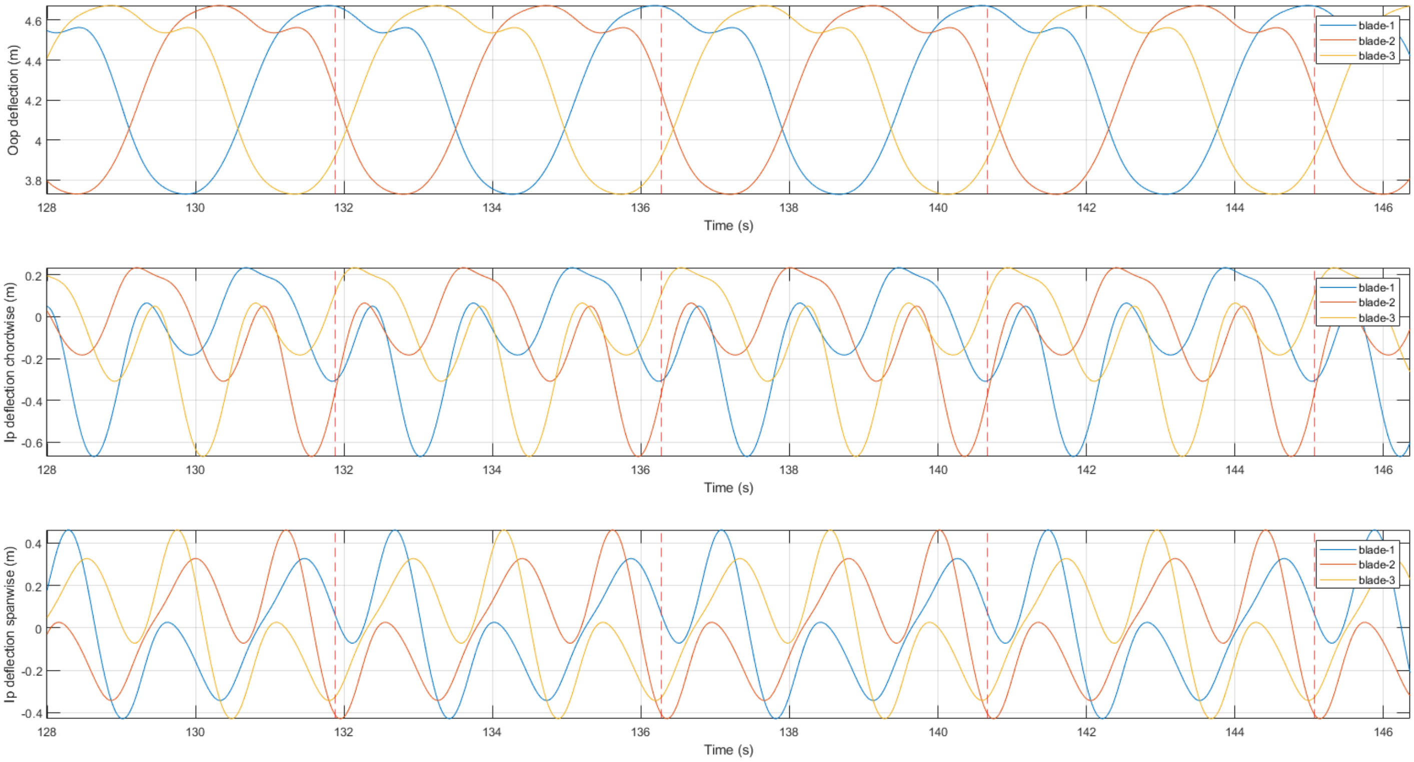


Figure B.2: Blade tip deflection for the steady turbine at 12 m/s hub speed

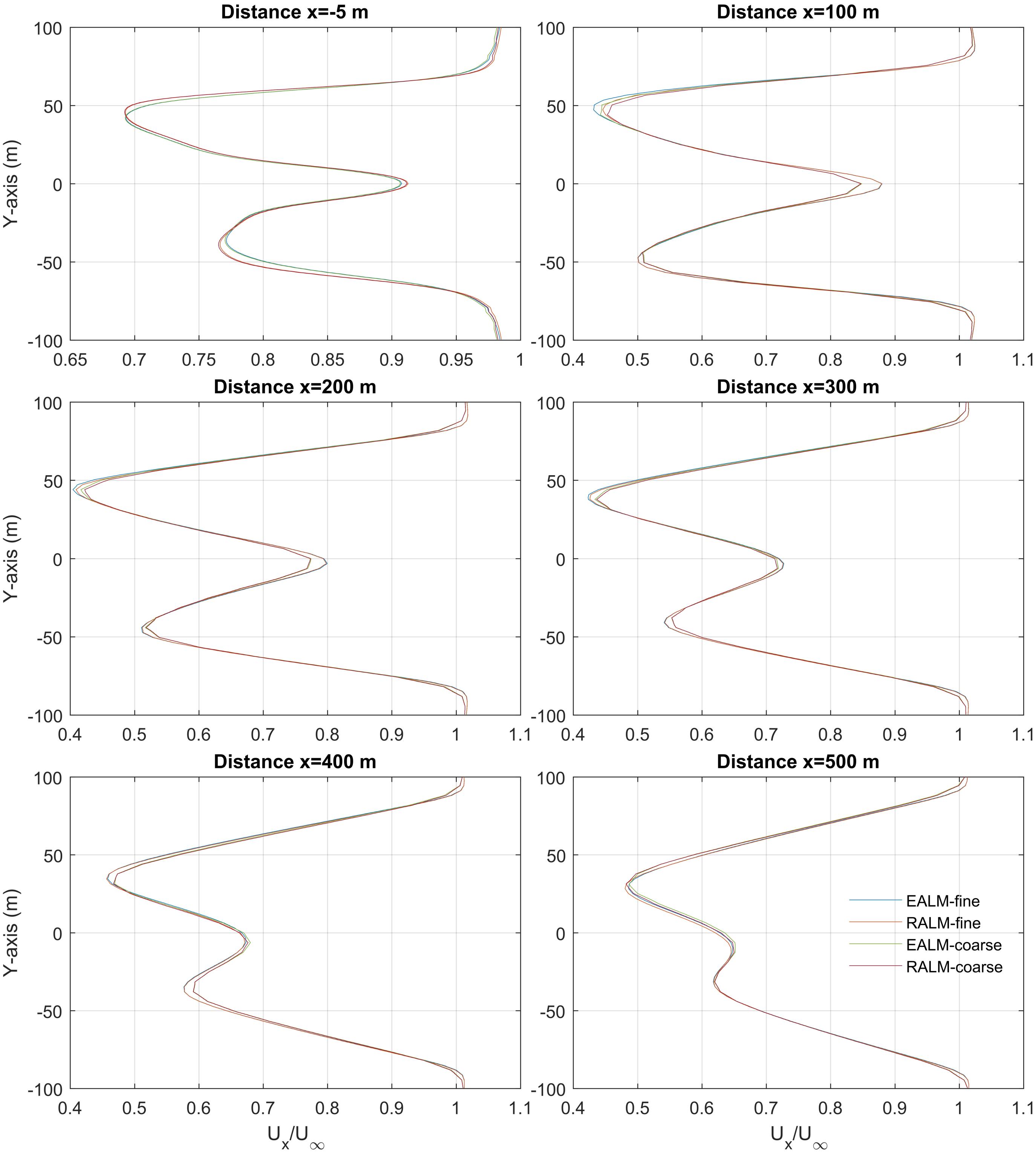


Figure B.3: Wind profile along a horizontal line passing the turbine height at different wake locations, for the steady turbine at 9 m/s hub speed

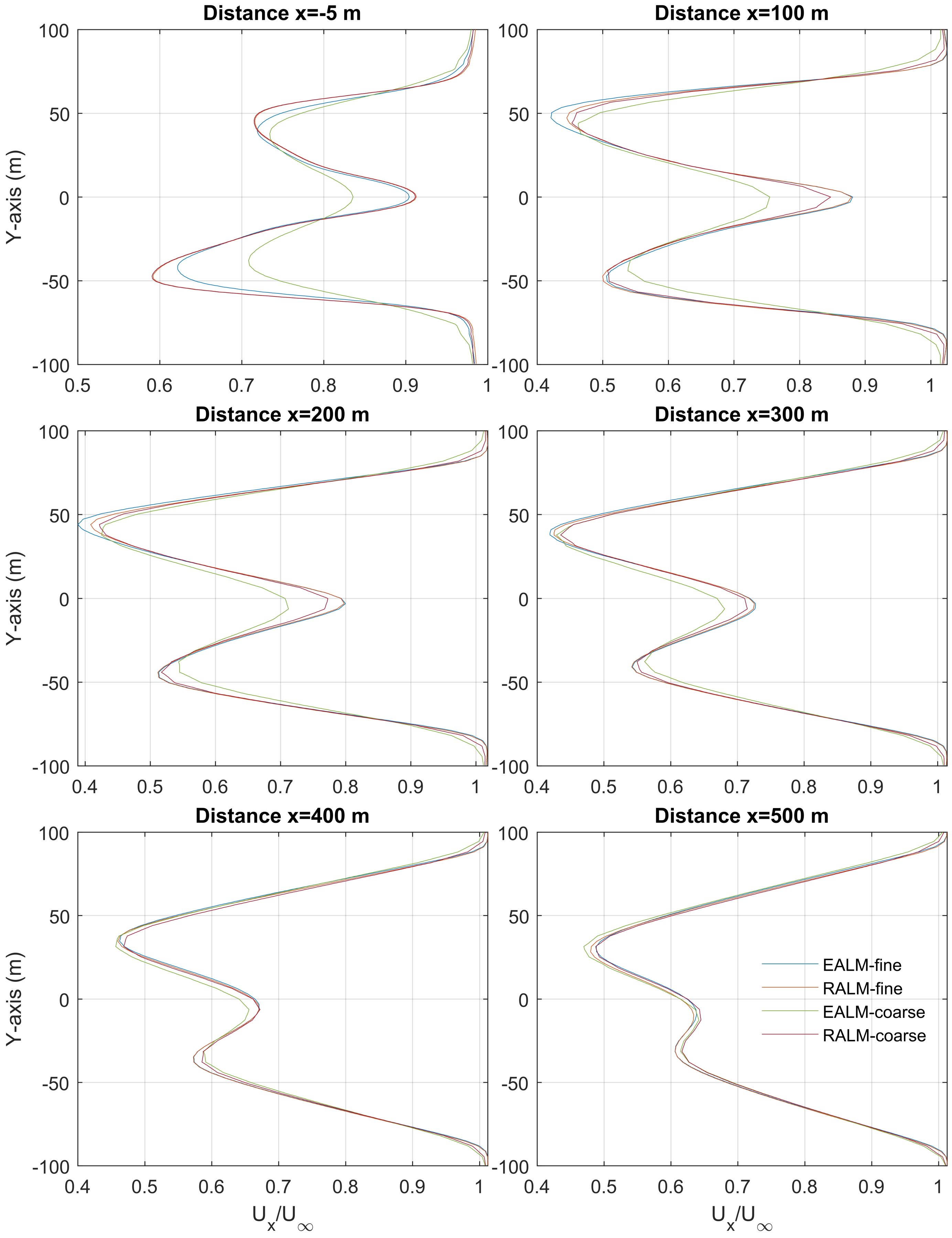


Figure B.4: Wind profile along a horizontal line passing the turbine height at different wake locations, for the steady turbine at 12 m/s hub speed

B.2. Moving turbine

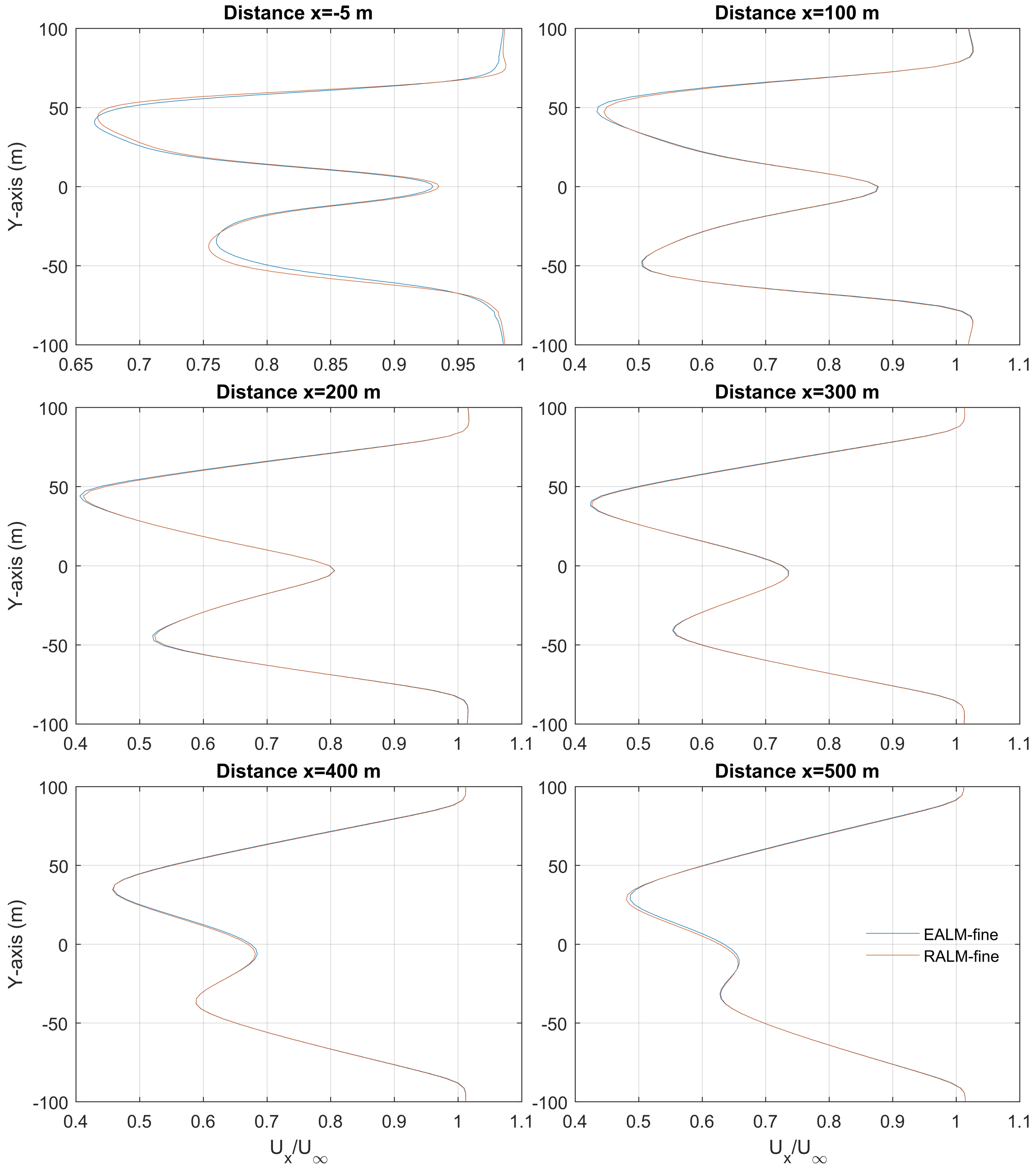


Figure B.5: Wind profile along a horizontal line passing the turbine height at different wake locations, for the steady turbine at 9 m/s hub speed

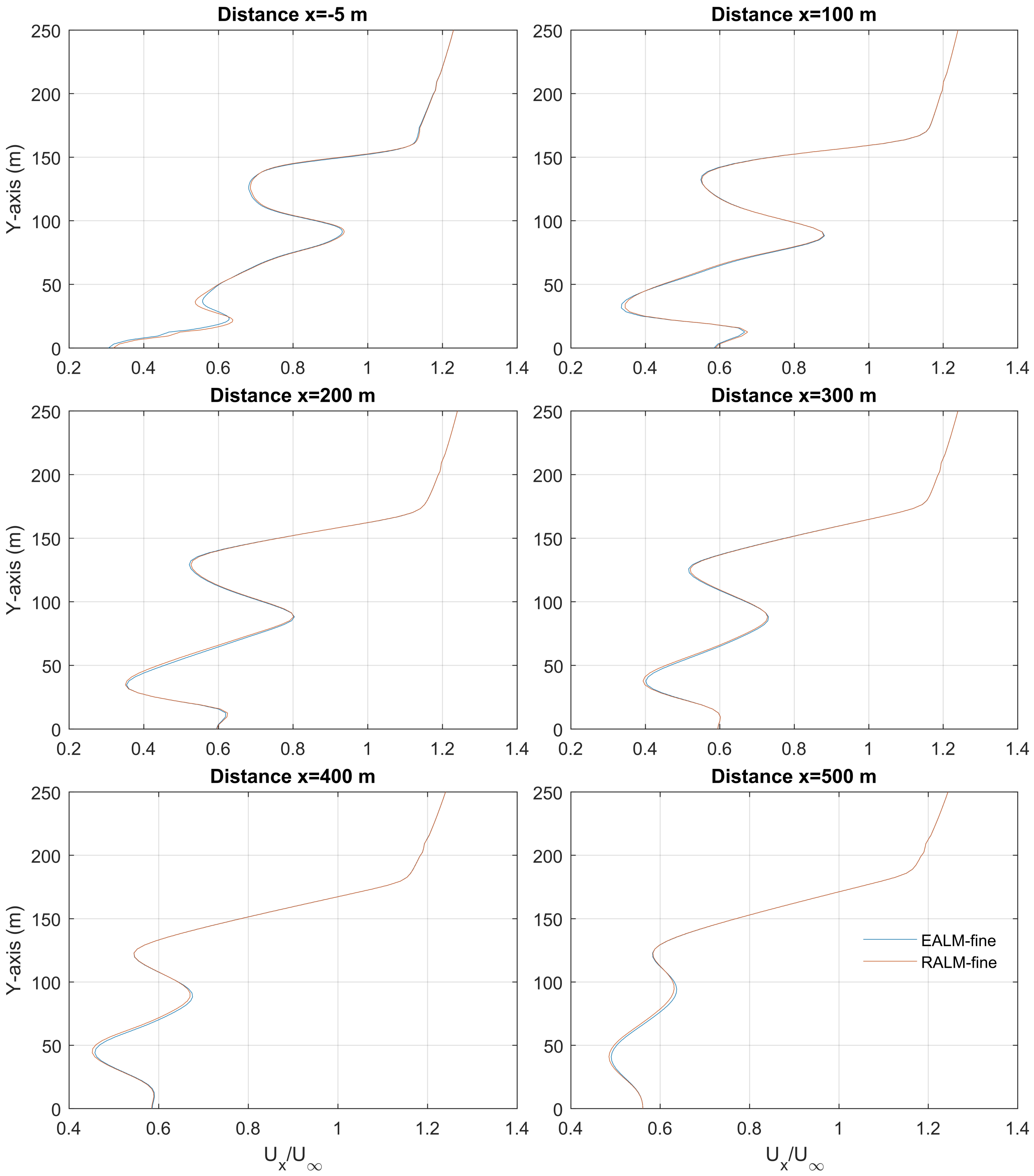


Figure B.6: Wind profile along a vertical line passing the turbine height at different wake locations, for the moving turbine at 9 m/s hub speed

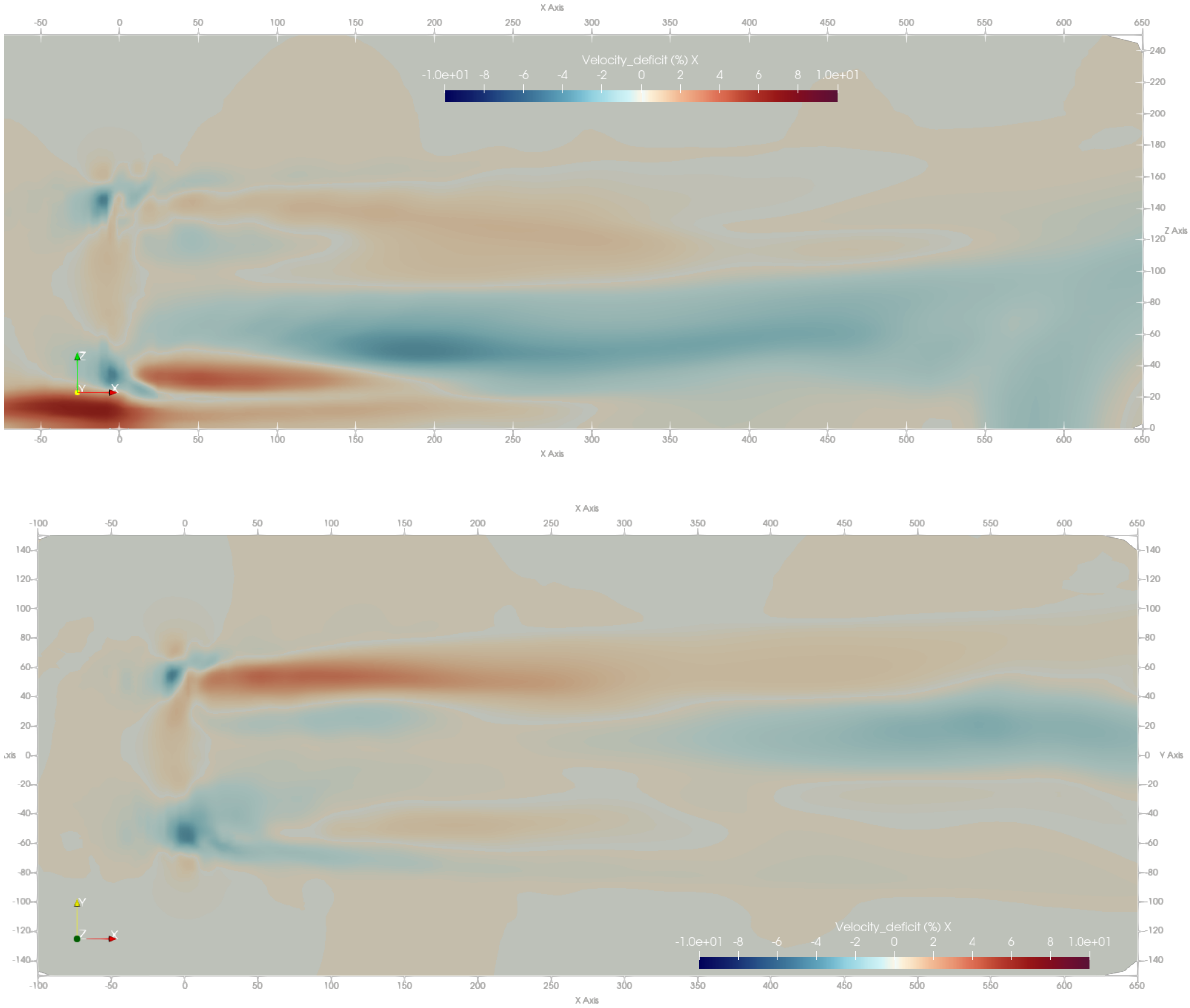


Figure B.7: Velocity difference of moving elastic turbine at 9 m/s hub speed-(a) side view, (b) top view

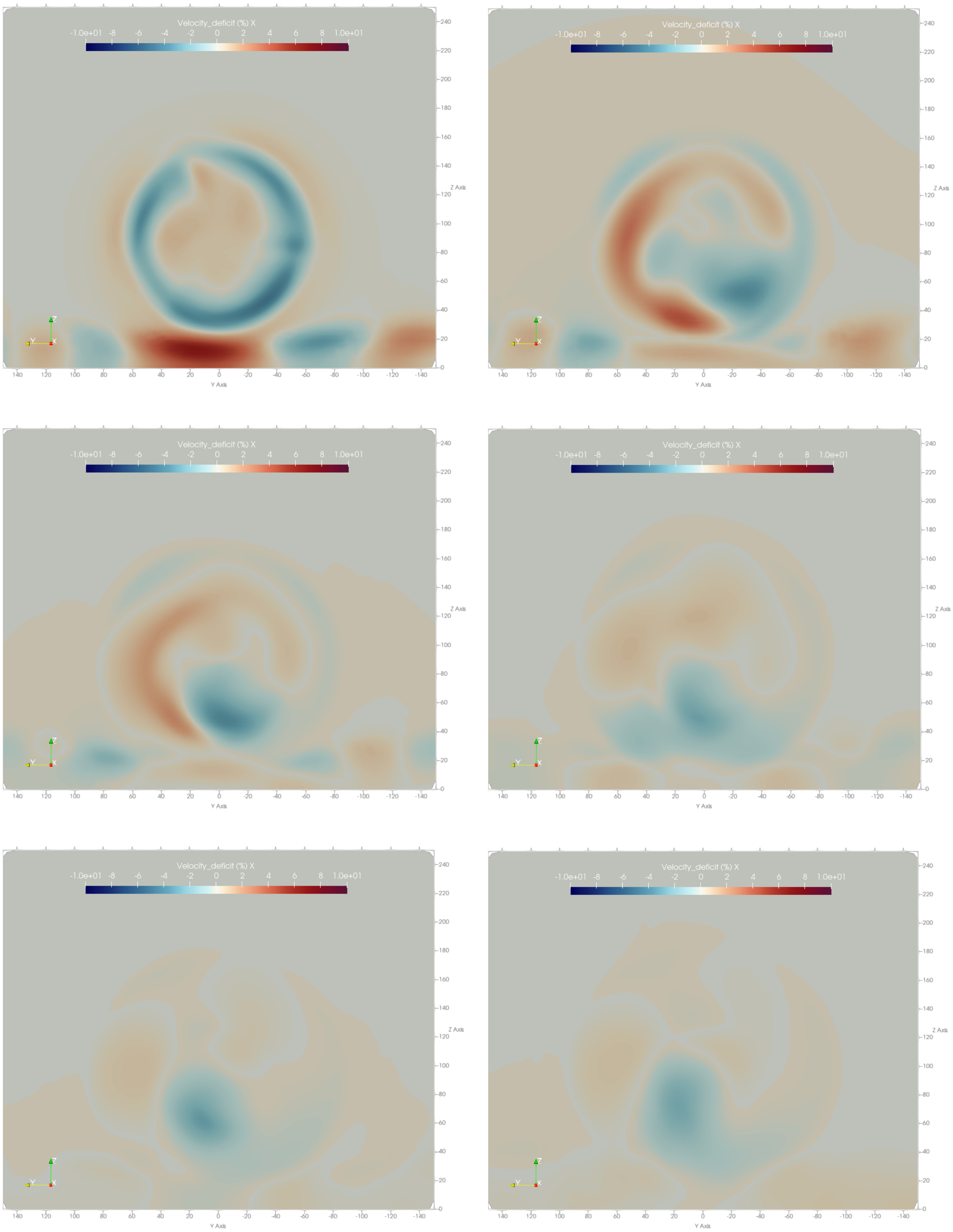


Figure B.8: Velocity difference in planes parallel to rotor plane at different wake position (a) rotor plane (b) x=100m (c) x=200m (d) x=300m (e) x=400m (f) x=500m. From top to bottom and left to right, at time t=150 s. For moving turbine at 9 m/s hub speed

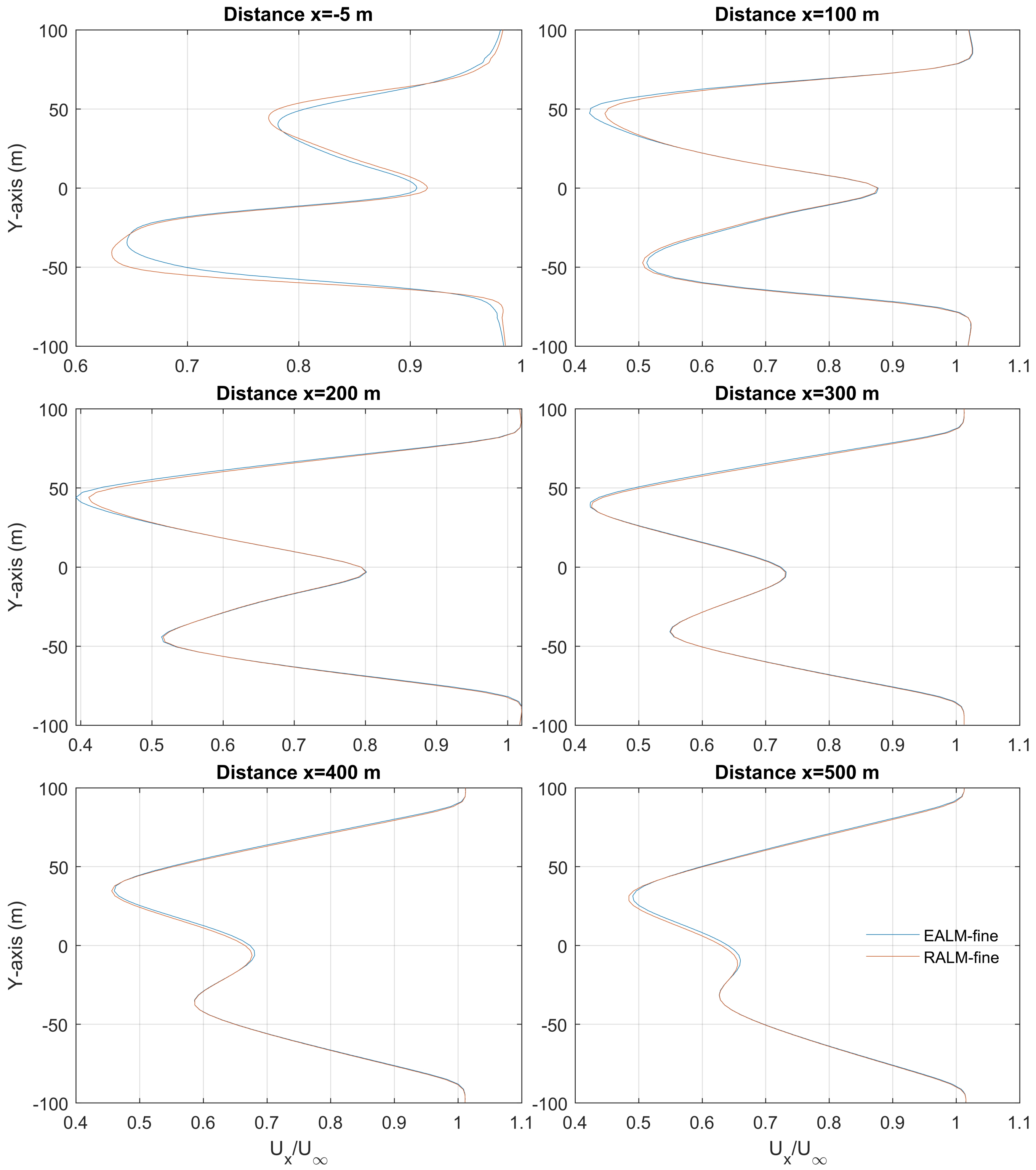


Figure B.9: Wind profile along a horizontal line passing the turbine height at different wake locations, for the steady turbine at 9 m/s hub speed

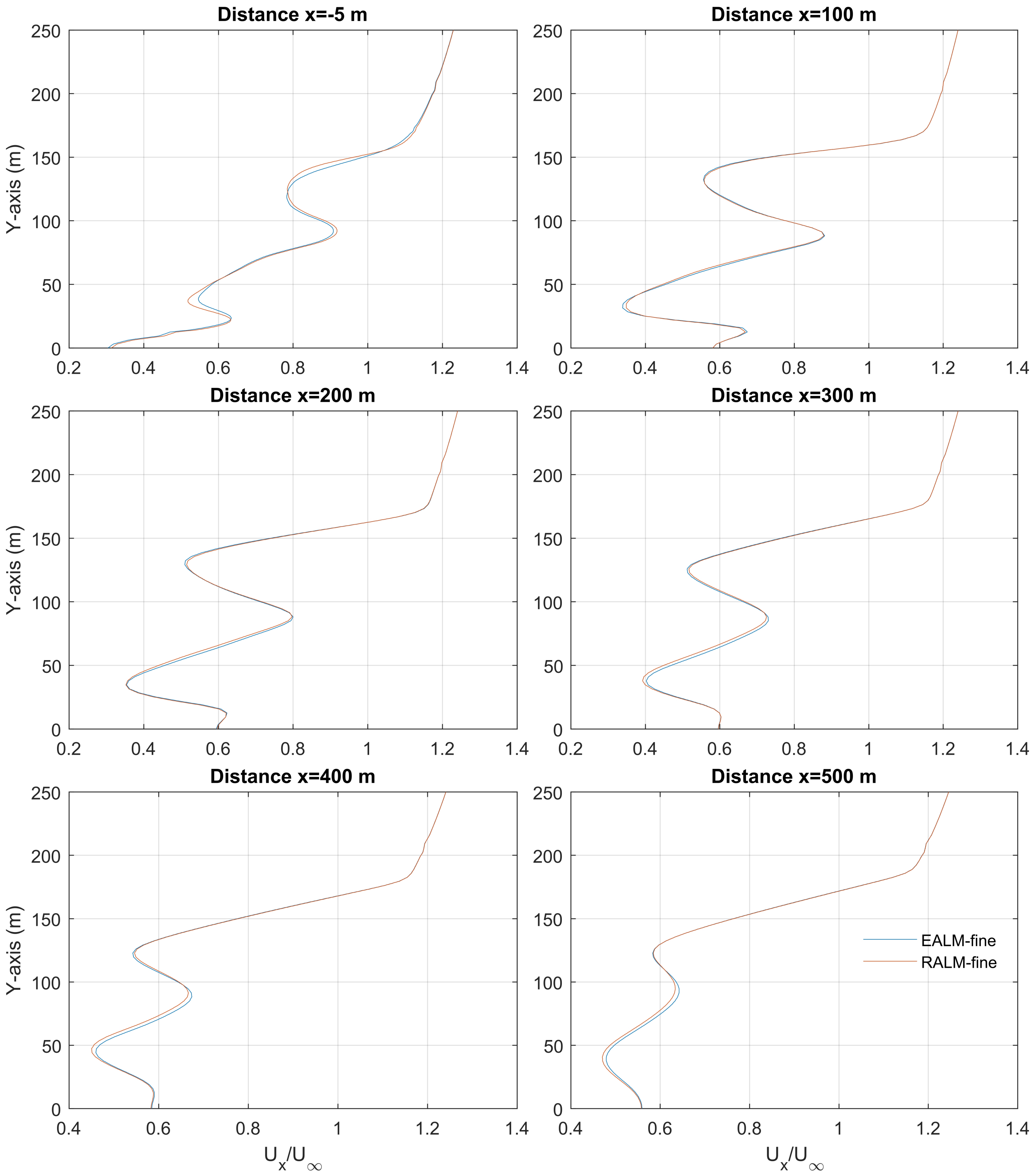


Figure B.10: Wind profile along a vertical line passing the turbine height at different wake locations, for the moving turbine at 9 m/s hub speed

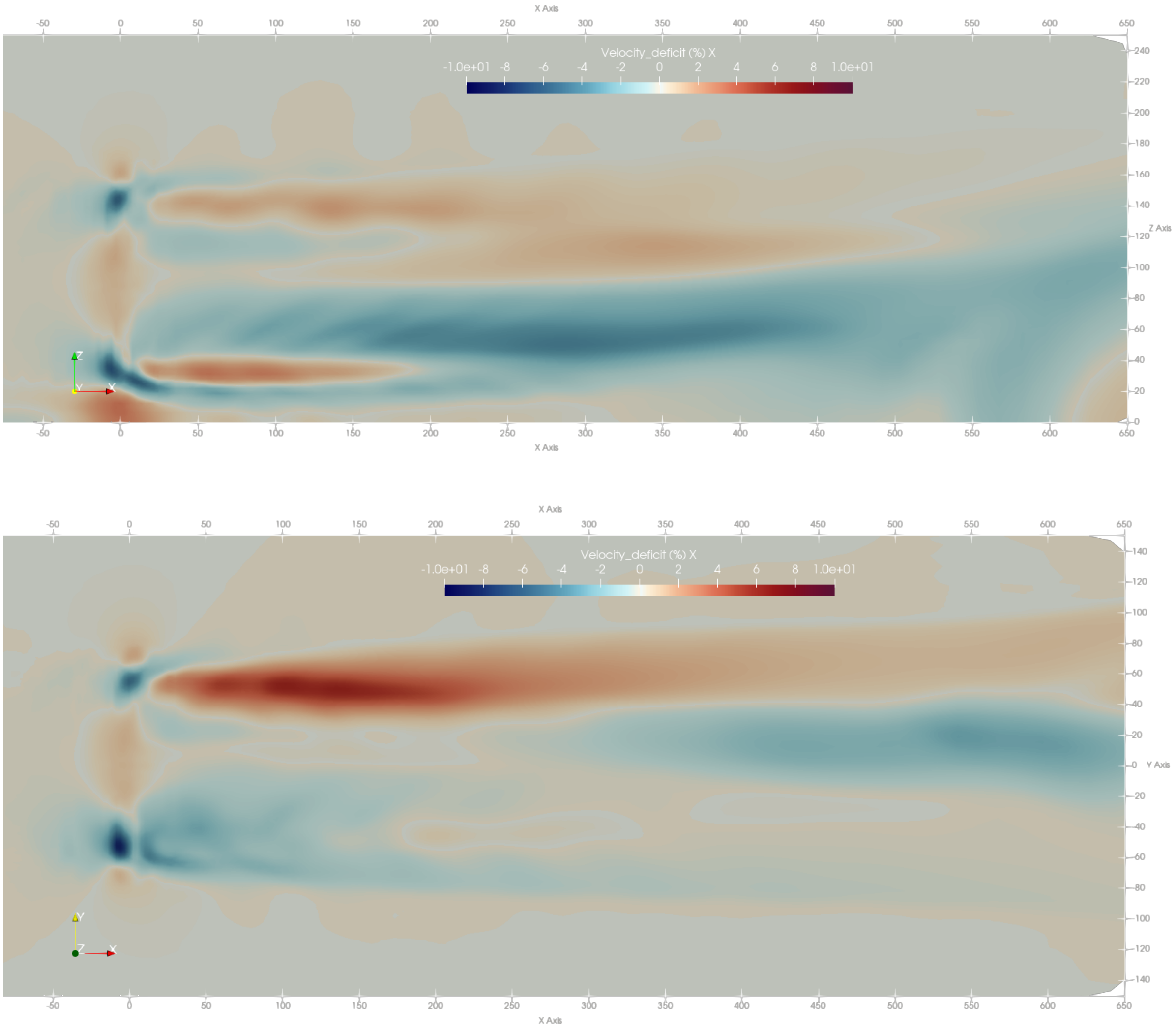


Figure B.11: Velocity difference of moving elastic turbine at 9 m/s hub speed-(a) side view, (b) top view

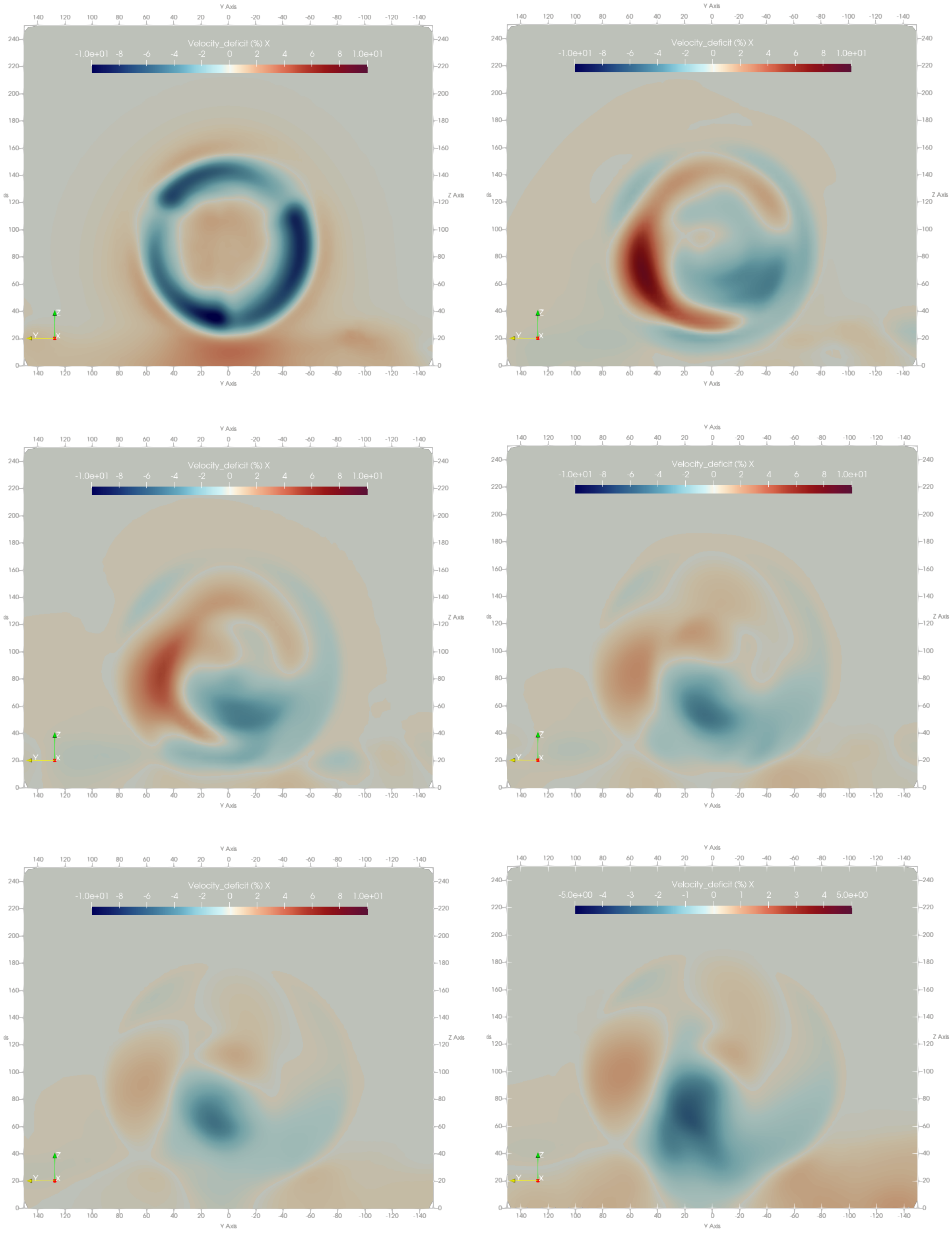


Figure B.12: Velocity difference in planes parallel to rotor plane at different wake position (a) rotor plane (b) x=100m (c) x=200m (d) x=300m (e) x=400m (f) x=500m. From top to bottom and left to right, at time t=150 s. For moving turbine at 12 m/s hub speed