

**A VLES and Two-Way Dynamic Fluid-Structure Interaction Study
of Wind Turbines**

Erkhan Sarsenov ID:201594086, B.Eng.

**Submitted in fulfillment of the requirements
for the degree of Master of Science
in Mechanical & Aerospace Engineering**



**NAZARBAYEV
UNIVERSITY**

**School of Engineering and Digital Sciences
Department of Mechanical & Aerospace Engineering
Nazarbayev University**

**53 Kabanbay Batyr Avenue,
Astana, Kazakhstan, 010000**

**Supervisor: Professor Yong Zhao
Co-supervisor: Professor Dongming Wei**

July 2023

DECLARATION

I hereby, declare that this manuscript, entitled “A VLES and Two-Way Dynamic Fluid-Structure Interaction Study of Wind Turbines”, is the result of my own work except for quotations and citations, which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.



Name: Erkhan Sarsenov

Date: 22.07.2023

Abstract

From year to year the energy demand is rising worldwide, which is leading to the development of renewable energy sources. One of them are the wind turbines that generate energy from the wind. However, in order to meet the energy demand and increase the efficiency, they are being made larger and larger, which is causing serious issues such as noise and vibration, as well as difficulties in studying their aerodynamics. Therefore, there is a need for Fluid-Structure Interaction (FSI) analysis of wind turbines, which can be useful in solving these problems and optimal design of the turbines. There are two methods of FSI analysis implemented in this work, high-fidelity and low-fidelity. The first method aimed to create a 2-way dynamic FSI of wind turbines with the application of the advanced VLES turbulence model. The FSI was tried to be performed by the coupling of the CFD and CSD solvers in preCICE software using the Arbitrary Eulerian Lagrangian (ALE) approach. The second method is done by coupling the Lifting Line Free Vortex Wake aerodynamic solver with the Chrono structural solver in QBlade software. The whole software setup was done, including the connection between solvers, dependencies as well as the test cases. However, the first method had some limitations, which were shown and explained. The findings and results are presented and discussed in the paper.

Table of Contents

Abstract.....	2
Table of Contents	3
List of Abbreviations & Symbols.....	4
List of Figures.....	5
Chapter 1 – Introduction.....	6
1.1. Background	6
1.2. Aims and objectives	7
1.3. Scope and Constraints	7
1.4. Novelty and Contributions	7
1.5. Problem Statement.....	8
1.6. Hypothesis.....	8
Chapter 2 – Literature Review	9
2.1. Overview: Fluid-Structure Interaction approaches	9
2.2. Turbulence Models	10
2.2.1. RANS Model.....	10
2.2.2. LES Model	11
2.2.3. VLES Model	12
Chapter 3 – Methodology.....	15
3.1. ALE method	15
3.2. PISO algorithm	16
3.3. Software overview	18
3.3.1. Coupling in preCICE.....	20
Chapter 4 – Results and Discussion.....	22
4.1. FSI Test Cases	22
4.1.1. Elastic 3D Tube	22
4.1.2. Perpendicular flap.....	24
4.2. NREL Phase VI wind turbine	25
4.2.1. Fluid model	25
4.2.2. Solid model	27
Chapter 5 – FSI Trials.....	29

5.1.	FSI run	29
5.1.1.	Initial trials	29
5.1.2.	Misalignment of the geometries	30
5.1.3.	Geometry with a spar	32
5.1.4.	Geometry with three spars	33
5.1.5.	Software Limitation	33
Chapter 6 – Low-Fidelity FSI		34
6.1.	Overview	34
6.2.	QBlade.....	36
6.2.1.	Aero-elastic coupling.....	36
6.2.2.	Lifting Line Free Vortex Wake Method	38
6.2.3.	The Chrono Library	39
6.3.	Results in QBlade	40
6.3.1.	NREL 5 MW wind turbine.....	40
6.3.2.	NREL Phase VI wind turbine	45
Chapter 7 – CFD Simulation		48
7.1.	Mesh	48
7.2.	Simulation results.....	49
Chapter 8 – Conclusions & Future works		51
References.....		52

List of Abbreviations & Symbols

<i>CFD</i>	Computational Fluid Dynamics
<i>FSI</i>	Fluid Structure Interaction
<i>CSD</i>	Computational Structural Dynamics
<i>VLES</i>	Very Large Eddy Simulation
<i>NREL</i>	National Renewable Energy Laboratory
<i>AHTM</i>	Arbitrary Hybrid Turbulence Modelling
<i>MDO</i>	Multidisciplinary Design Optimization
<i>ALE</i>	Arbitrary Lagrangian-Eulerian
<i>URANS</i>	Unsteady Reynolds-Averaged Navier-Stokes

List of Figures

Figure 2.1. Moving mesh algorithm flowchart.....	14
Figure 2.2. PISO algorithm flowchart.....	16
Figure 3.1. Concept of preCICE.....	18
Figure 3.2. Running solid solver (CalculiX).....	19
Figure 3.3. Running fluid solver (OpenFOAM).....	19
Figure 3.4. FSI analysis interface.....	20
Figure 4.1. Mesh for fluid solver.....	20
Figure 4.2. Mesh for solid solver.	21
Figure 4.3. Flow traveling through the tube	21
Figure 4.4. Deformation of the tube.....	21
Figure 4.5 Mesh for fluid domain.	22
Figure 4.6. Mesh for flap.....	22
Figure 4.7 Deformation of the geometry.	23
Figure 4.8. Deformation of the flap.	23
Figure 4.9. Blade dimensions.....	24
Figure 4.10. Blades.	24
Figure 4.11. Fluid domain.	24
Figure 4.12. Pressure on the blades.	25
Figure 4.13. Mesh for solid solver.	25
Figure 4.14. Fixed part.	26
Figure 4.15. Point of load.	26
Figure 4.16. Simulation in CalculiX.	27
Figure 5.1. Terminal output.....	28
Figure 5.2. Shell model mesh.	28
Figure 5.3. Floating point exception.	28
Figure 5.4. Comparison of the geometries.....	29
Figure 5.5. Meshes for fluid and structural solvers, respectively.....	29
Figure 5.6. Combined view of the meshes.....	30
Figure 5.7. Transferred mesh.....	30
Figure 5.8. Comparison of the geometries.....	31
Figure 5.9. Geometry with one spar.....	31
Figure 5.10. Blade with three spars.....	32
Figure 6.1. The flowchart of aero-elastic coupling.....	35
Figure 6.2. NREL 5 MW wind turbine.....	39
Figure 6.3. Internal structure of the blades.....	39
Figure 6.4. Structural loadings on the blades	40
Figure 6.5. Vorticity contours	41
Figure 6.6 Cp vs tsr graph.....	42
Figure 6.7. Twist angle vs R graph.....	43
Figure 6.8. NREL Phase VI wind turbine.....	43
Figure 6.9. Internal structure of the blades.....	44
Figure 6.10. Structural loadings on the blades	44
Figure 6.11. Vorticity contours	45
Figure 6.12 Cp vs tsr graph.....	45
Figure 6.13. Twist angle vs R graph.....	46
Figure 7.1. a) Mesh for fluid domain; b) Rotor rotation zone inside the domain.....	47
Figure 7.2. Mesh for blades.....	47
Figure 7.3. The change of pressure distribution	47
Figure 7.4. Vorticity contours around the blades	48
Figure 7.5. Pressure coefficients vs x/c	48

Chapter 1 – Introduction

1.1. Background

One of the alternative energy sources is wind that has a great promise in terms of cost-effectiveness and efficiency. With regard to its wide availability, high degree of development, and little effect on the environment, wind energy among other sustainable energy sources has significant potential [1]. Long, slender, composite structures that rotate in a wind turbines' plane to generate mechanical energy are called turbine blades. The air flowing across the foil portions creates a pressure difference between the two sides of the blades, which is what causes motion [2]. Even though the load of wind on the blades' surface generates rotational energy, there is a structural bending, which results in the phenomena where parts of the blade twist from their original position [2]. Bending or twisting of the blades results in direct changes to the field of fluid around the blade, which ultimately affects the profile of load. The result of this interaction, known as aeroelasticity, is a deviation from the original blade design. An airfoil section is utilized to enhance the lift force of a wind turbine blade for boosting the power output [3]. On the blade surface, unstable events like detachment, transition, and reattachment take place. It consequently experiences stall phenomena, where the force of lift and drag abruptly vary as the wind speed increases. The dominating aerodynamic properties of a blade surface pressure have a significant impact on the structural design process. The wind turbines would not be able to generate the amount of electricity they were expected to if these aspects were not taken into account when designing the blades. With rotor diameters rising from 10 meters to 160 meters during the last three decades, the wind turbines' average size has dramatically increased [4]. With the increasing size of the wind turbines, the issues of vibration and noise emissions rise as well [5]. However, the application of fluid-structure interaction (FSI) could possibly open the way for vibration and noise reduction as well as the further design optimization of the large wind turbines. Therefore, in order to optimize their design, it is crucial to model the intricate dynamics of turbulent and unsteady flow across the blades in rotation as well as the significant deformation in structure of blades [6].

1.2. Aims and objectives

- To develop a novel and more accurate methodology for analysis of wind turbines:
 - a) by implementing and using advanced Arbitrary Hybrid Turbulence Modelling (AHTM) approach called Very Large Eddy Simulation (VLES) in the DAfoam solver
 - b) by coupling the VLES solver with CalculiX structural solver through preCICE to study dynamic 2-way FSI, which is first in the field
- To Compare it to existing URANS FSI methods.
- To study vibration and noise with this novel approach.

1.3. Scope and Constraints

The scope of the study is the 3D large scale transient computational simulation for NREL Phase VI wind turbine.

The constraints, which can be faced during the work are:

- A limited number of studies, papers, and publications.
- A need for learning to work with a number of new software.
- Very high computational costs and turnaround time.

1.4. Novelty and Contributions

- The first to use advanced turbulence modeling such as VLES for 2-way dynamic FSI study.
- This approach can lead to accurate study of wind turbine vibration and noise and multidisciplinary design optimization (MDO).

- Implementation of preCICE to couple DAfoam and CalculiX with shell-element structural model which can be used for future transient Multidisciplinary Design Optimization (MDO) and for vibration and noise reduction (previously [7] used foam extend 4.0 which can only use solid elements and cannot be used for MDO).

1.5. Problem Statement

- Difficult to calculate the highly complex turbulent time dependent flow around wind turbines.
- Equally difficult to calculate fluid-structure interaction (FSI) between wind turbine blades and the surrounding turbulent air flow as blade deformation is also complex and large, and the blades are rotating and even yawing at the same time.
- Even more difficult to couple an advanced turbulence with FSI solvers.
- Thus, very few research works have been done and reported in the literature.

1.6. Hypothesis

- Use PreCICE to couple CalculiX and DAfoam by ALE coupling method.
- VLES turbulence model has been developed and shown to be versatile and accurate but less expensive than advanced models such as LES.

Chapter 2 – Literature Review

2.1. Overview: Fluid-Structure Interaction approaches

The simulation of 2-way FSI analysis of wind turbine was done by Zhangaskanov et al. [7]. The purpose of this project was to build and test the 2-way FSI analysis for NREL Phase VI wind turbine. In order to examine the impacts of FSI on the wind turbine, a two-way FSI model is created in OpenFOAM software. The coupling for FSI is done by combining the CFD and CSD, which provide highly accurate results. The authors stated that the major benefit of FSI modeling is the ability to simultaneously examine the impacts of aerodynamic loadings on structures, and deformations on the flow field. The work of Guma et al. [8] demonstrates the application of FSI analysis on the WINSSENT wind turbine. The Kratos structural solver and the FLOWer fluid solver were coupled to make the FSI, which allowed the turbine to be modelled using either shell or beam structural elements. The BEM model is sufficient for small wind turbines analysis in steady state. However, if the conditions flow and design of wind turbine become more complex, there is a need for FSI simulations, which can provide more accurate results. Kamalov et al. [9] using the OpenFOAM software, constructed an AHTM based on VLES for high fidelity wind turbine simulation and to compare with the conventional URANS model. The authors also used the wind turbine NREL Phase VI for this work. The results showed that the VLES outperforms the conventional URANS model in terms of accuracy, but still requires much more computational resources than URANS. In the study done by Grinderslev et al. [10], the fluid solver EllipSys3D and the structural solver HAWC2 were coupled to perform FSI modeling of a 2.3 MW NM80 wind turbine. For comparison, simulations based on the simpler BEM were also carried out, and measurements from field tests were utilized to validate the calculations. It was discovered that for intricate flows, BEM-based analyzes show roughly the same forces as FSI, but are not fully accurate. In contrast to FSI, for large and complex simulations, the application of BEM-based method is not adequate, because it can be deceptive. Lee et al. [3] have worked on Aerodynamics and Fluid-Structure interaction analysis of the wind turbine. The work was done in ANSYS software, where the CFD and BEM methods were used with FSI. The obtained results showed that the thrust force primarily affected the wind turbine, while rotational force did not have a significant impact on the features of deformation. According to authors, the created FSI test was effective in the estimation of the characteristics of aerodynamic force and the obtained outcomes are more

accurate. Bazilevs et al. [11] also stated that high accuracy is another benefit of FSI modeling. This method was used for the deformation analysis of wind turbines in this work. The authors state that proper wind turbine modeling can only be done by the FSI analysis. According to Wang et al. [12], by precisely predicting the deforming blade shape under various load circumstances, FSI modeling has the ability to enhance predictions of blade efficiency and noise emission. In order to produce aerodynamic stresses and the related structural responses, fluid-structure interaction modeling requires both structural and aerodynamic components. For the examination of the FSI behavior of blades, a range of model building techniques and coupling strategies are currently available. In this work of Santo et al. [13], FSI simulations are used to examine the influence of a wind gust on the blades of a wind turbine. The computational structural mechanics (CSM) model and the CFD model were used as the basis for the FSI model. The work was done using the ANSYS software. A spring-based smoothing technique was used to modify each blade component mesh, leading to the adoption of an ALE. The fluid mesh was distorted in accordance with the structural solver's instructions at the fluid-structure contact. In order to calculate the blade loads and damping coefficient for the 5MW wind turbine blade, discrete and control equations for the fluid domain and structural domain, respectively, were created in the work of Shi et al [14]. ANSYS and UG modelling software were used to finish the modeling of the blade object. Based on it, the wind tunnel test was used to verify the accuracy of the numerical computation of blade vibration characteristics under various wind and rotating speeds. Calculation results show that the first primary stresses at the blade surface along the span-wise direction are numerically consistent with the findings of the wind tunnel test, demonstrating the validity of the theory and numerical models.

2.2. Turbulence Models

2.2.1. RANS Model

The Reynolds-averaged Navier-Stokes (RANS) equations are a set of partial differential equations used to describe the flow of a fluid, taking into account the turbulence effects. They are derived from the Navier-Stokes equations by applying a process called Reynolds averaging. The RANS equations are widely used in CFD to model turbulent flows in engineering applications. There are various RANS turbulence models, including models such as one-equation, and two-equation [15]. The two-equation turbulence models like k-omega, k-omegaSST, and k-epsilon are the most well-known ones that can typically generate precise predictions for a wide range of

applications in the industry under specific flow conditions. The turbulent viscosity needs to be calculated using two transport equations in the two-equation models. The $k - \epsilon$ is among the early turbulence models. It can be applied for the estimation of distant field turbulence. Apart from that, there is a $k - \omega$ model, which can be used to improve the accuracy of turbulence prediction close to the wall [15]. The term k used in this model stands for turbulence kinetic energy and, while ω represents the specific rate of turbulence dissipation, signifying the conversion of turbulence kinetic energy into internal thermal energy. At a specific range of speeds, the SST $k - \omega$ turbulence model can correctly predict the flow across the wind turbines [16]. All of these turbulence models eventually have drawbacks, but when it comes to simulating wind turbines, the SST $k - \omega$ model predicts the flow more accurately than other models [17]. For enhanced accuracy in resolving turbulence near walls and flow separation, the shear stress transport (SST) $k - \omega$ model serves as an improved version [15]. The SST $k - \omega$ model utilizes two equations. The first equation represents the turbulence specific dissipation rate:

$$\frac{D(\rho\omega)}{Dt} = \nabla \cdot (\rho D_\omega \nabla \omega) + \frac{\rho\gamma G}{v} - \frac{2}{3}\rho\gamma\omega(\nabla \cdot \mathbf{u}) - \rho\beta\omega^2 - \rho(F_1 - 1)CD_{k\omega} + S_\omega \quad (1)$$

The following is the equation for the kinetic energy of turbulence:

$$\frac{D(\rho k)}{Dt} = \nabla \cdot (\rho D_k \nabla k) + \rho G - \frac{2}{3}\rho k(\nabla \cdot \mathbf{u}) - \rho\beta^* \omega k + S_k \quad (2)$$

The turbulence viscosity is determined as follows:

$$\mu_t = \frac{\rho a_1 k}{MAX(a_1 \omega, \sqrt{2} S_t F_2)} \quad (3)$$

2.2.2. LES Model

The LES turbulence model stands for Large Eddy Simulation turbulence model. It is a CFD approach used to simulate turbulent flows. Unlike traditional RANS models, LES directly resolves large-scale turbulent structures while modeling the effects of small-scale turbulence. In turbulent flows, energy is transferred across a wide range of spatial scales [18]. The largest eddies contain most of the turbulent energy and are responsible for the dominant structures in the flow. The smaller eddies, which are more difficult to resolve due to their size, dissipate the turbulent energy into heat through viscous effects. In LES, the large-scale eddies are explicitly resolved by the numerical solver, while the small-scale eddies are modeled. This approach provides a balance

between accuracy and computational cost. The LES approach is particularly well-suited for flows with significant separation, complex geometries, and unsteady features. The LES equations are derived from the Navier-Stokes equations, and the filtering operation is used to separate the flow field into resolved large-scale components and subgrid-scale (SGS) components [19]. The resolved scales are directly computed, while the SGS scales are modeled using turbulence models. The most common LES models include the Smagorinsky model and its variants, dynamic models, and scale-adaptive models. The Smagorinsky model is one of the earliest and simplest LES models. It introduces a turbulent viscosity based on the local strain rate and the grid-filtered velocity gradients. The turbulent viscosity adds a dissipative effect that mimics the small-scale turbulent energy dissipation. LES simulations are computationally demanding because of the need to resolve large-scale structures explicitly. As computational resources have increased over the years, LES has become more accessible and widely used for a variety of engineering applications, such as aerodynamics, combustion, and environmental flows. Overall, LES offers a valuable tool for accurately predicting turbulent flows and capturing important unsteady phenomena that are critical for many real-world applications. However, its application requires careful consideration of the grid resolution, boundary conditions, and appropriate turbulence modeling to achieve accurate and reliable results.

2.2.3. VLES Model

The core concept of the control function F_r , which is derived from turbulence energy, is described by the equation below [20][21].

$$F_r = \frac{\int_{L_k}^{L_c} E(L) dL}{\int_{L_k}^{L_i} E(L) dL} \quad (4)$$

where, $L_k = \frac{\nu^{3/4}}{\varepsilon^{1/4}}$ stands for Kolmogorov length scale, $L_i = \frac{k^{3/2}}{\varepsilon}$ is the integral length scale and $L_c = C_{VLES}(\Delta_x \Delta_y \Delta_z)^{1/3}$ is a cut-off length scale. Additionally, the mesh dimensions are $\Delta_x \Delta_y \Delta_z$ in different directions and the ν is laminar kinematic viscosity. The resolution control function is defined by Equation (5) as the ratio of the total turbulent energy to the unresolved turbulent energy.

$$F_r = \min \left[1.0, \left(\frac{1.0 - e^{-\beta * L_c / L_k}}{1.0 - e^{-\beta * L_i / L_k}} \right)^2 \right] \quad (5)$$

VLES offers certain advantages over LES in specific scenarios, especially when computational resources are limited or when certain flow features are of primary interest [22][23]. Some of the advantages of VLES over LES include:

Computational Efficiency: VLES uses a grid resolution finer than LES but coarser than RANS. As a result, it requires fewer computational resources compared to LES. This makes VLES more feasible for simulations involving complex geometries or longer time scales where full LES would be excessively expensive.

Balance of Accuracy and Cost: VLES aims to capture both large and some small-scale turbulent structures directly, striking a balance between accuracy and computational cost. It may provide reasonably accurate results for many practical engineering applications without the computational burden of LES.

Moderate Modeling Complexity: The modeling of subgrid-scale (SGS) turbulence in VLES can be less complex than LES since it does not require modeling the smallest turbulent eddies. This can simplify the implementation and understanding of VLES turbulence models.

Transition Modeling: VLES can be useful for simulating flows with transitional turbulence. LES, while accurate, can sometimes struggle with transitional flows, especially if the resolution is not sufficient to capture the exact transition process. VLES can handle these situations better by partially resolving the transitional structures.

Time-Averaged Results: VLES can be suitable when one is interested primarily in time-averaged flow results. While LES provides detailed time-resolved information, it requires significant computational effort to obtain statistically converged results. VLES can offer reasonably accurate time-averaged results with less computational overhead.

Turbulent Mixing and Dispersion: VLES can effectively simulate turbulent mixing and dispersion phenomena in various engineering applications, such as pollutant dispersion in urban areas or combustion in industrial processes.

Larger Simulated Domains: Due to its lower computational cost, VLES allows for larger simulated domains and longer time scales compared to full LES. This can be beneficial for certain applications where capturing large-scale flow features is more critical than resolving all the small-scale turbulence. Despite these advantages, it's important to note that VLES is not a universally superior approach. LES remains the gold standard for accurately capturing all turbulent scales in the flow, especially for highly resolved simulations. The choice between VLES and LES depends on the specific requirements of the simulation, available computational resources, and the level of accuracy needed for the results.

Chapter 3 – Methodology

3.1. ALE method

Mesh deformation for FSI simulations should be determined after fluid and solid solutions have converged. One of the main methods used to couple fluid and solid solvers is the ALE method [7]. The ALE method combines the Lagrangian as well as Eulerian approaches, which are applied to moving and fixed meshes, respectively [24]. The momentum and continuity equations control incompressible transient flows in the framework of an ALE formulation. They are represented by:

$$\nabla \cdot (\mathbf{V} - \mathbf{V}_g) = 0 \quad (7)$$

$$\frac{\partial \mathbf{V}}{\partial t} + \nabla \cdot [(\mathbf{V} - \mathbf{V}_g)\mathbf{V}] = -\frac{1}{\rho}\nabla p + \nabla \cdot (v_{eff}\nabla\mathbf{V}) \quad (8)$$

The formulation of the governing equations with the $k - \omega$ SST model will be as follows:

$$\frac{\partial k}{\partial t} + \nabla \cdot [(\mathbf{V} - \mathbf{V}_g)k] - \nabla \cdot [(v - v_t\alpha_k)\nabla k] = \frac{1}{\rho}P_k - \beta^*\omega k \quad (9)$$

$$\frac{\partial \omega}{\partial t} + \nabla \cdot [(\mathbf{V} - \mathbf{V}_g)\omega] - \nabla \cdot [(v - v_t\alpha_\omega)\nabla\omega] = \frac{\rho C_1 P}{v_t} - C_2\omega^2 + \frac{2\alpha_\varepsilon(1-F_1)}{\omega}\nabla k \cdot \nabla\omega \quad (10)$$

The mesh deformation occurs after moving the wall boundary. The updated nodal coordinates and related nodal speed at each time-step are necessary for a new grid to be created in order to solve the governing Navier Stokes equations. Grid regeneration takes too long and is not cost-effective. Algebraic techniques can be used to update the distorted mesh's coordinates for minor displacement issues while maintaining the nodes' connectedness [25]. The figure below shows the moving mesh algorithm flowchart.

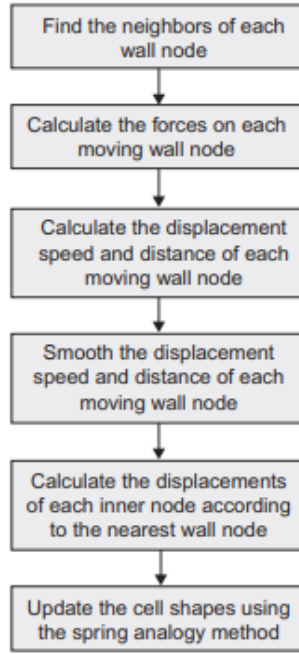


Figure 2.1. Moving mesh algorithms flowchart [20].

3.2. PISO algorithm

In 1986, Issa [26] proposed the PISO algorithm, which stands for Pressure-Implicit with Splitting of Operators. It does not require iterations, has big time steps, and requires less computational work. PISO uses predictor-corrector technique to meet mass conservation and consists of two corrector and one predictor steps. The $H_C[v']$ term in PISO algorithm is taken into account as a correction process part that includes two or more steps [24]. Recalculating the coefficients of the momentum equation leads to an explicit solution of the problem using the continuity satisfying pressure p^* and velocity v^{**} fields. The Rhie-Chow interpolation is applied to determine the mass flow rate field \dot{m}^{***} with the modified velocity field v^{***} . In a subsequent corrector stage, $H_C[v']$ is partially recovered, and the velocity correction is represented as:

$$\begin{aligned}
 v_c^{****} &= v_c^{***} + v''_c \\
 &= -H_c^{**}[v^{**}] - (D_c^v)^{**}(\nabla p^*)_c + v''_c \\
 &= -H_c^{**}[v^* + v'] - (D_c^v)^{**}(\nabla p^*)_c + v''_c \\
 &= -H_c^{**}[v^*] - H_c^{**}[v'] - (D_c^v)^{**}(\nabla p^*)_c + v''_c
 \end{aligned}$$

$$\begin{aligned}
&= \underbrace{-H_C^{**}[v^*] - (D_C^v)^{**}(\nabla p^*)_C}_{\approx v_C^{**}} - H_C^{**}[-D_C^v(\nabla p')_C] + v''_C \\
&\approx v_C^{**} + \underline{v''_C - H_C^{**}[-D_C^v(\nabla p')_C]} \tag{11}
\end{aligned}$$

The $H_C[v']$ obtained by the second corrector step is indicated by the underlined part. The second velocity correction satisfies:

$$v''_C = -H_C^{**}[v''] - (D_C^v)^{**}(\nabla p'')_C \tag{12}$$

When the Rhie-Chow interpolation is used with the Equation (11) between points F and C, we get the modified pressure correction field, represented as:

$$-\sum_{f \sim nb(C)} \rho_f \bar{D}_f \nabla p_f'' * S_f = -\sum_{f \sim nb(C)} \dot{m}_f + \sum_{f \sim nb(C)} \rho_f \bar{H}_f [v''] * S_f \tag{13}$$

One additional portion of $H_C[v']$ may be recovered each time this corrector step is performed, if desired. An illustration of the PISO algorithm is shown in Figure 2.2.

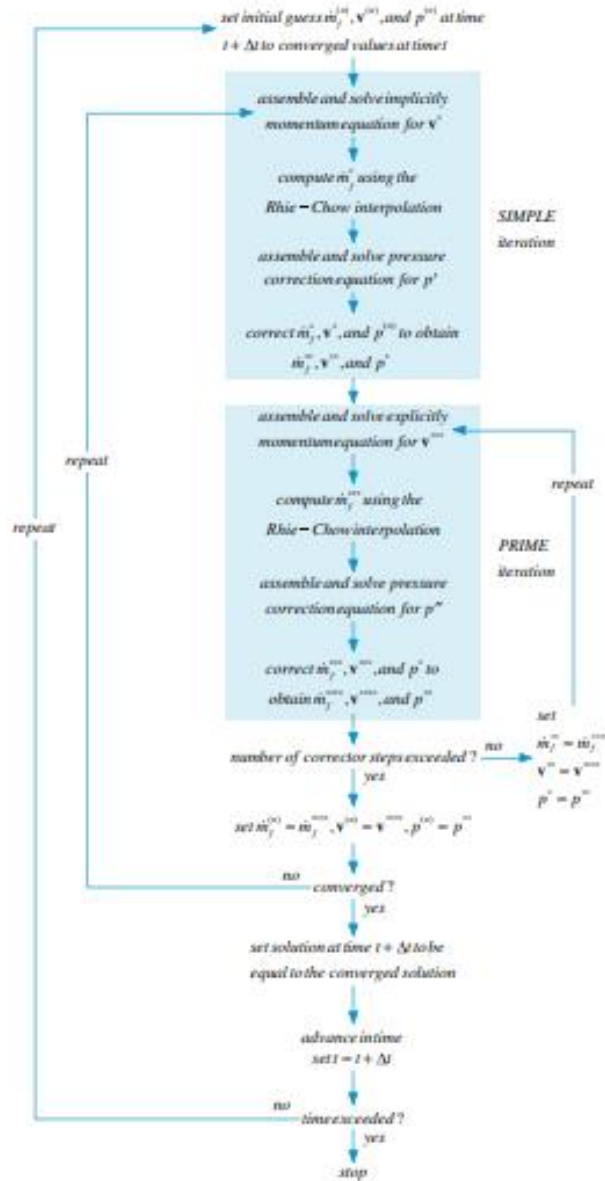


Figure 2.2 PISO algorithm flowchart [24].

3.3. Software overview

The computational simulation will be conducted on a model of National Renewable Energy Laboratory (NREL) Phase VI wind turbine. The work will be done by using software like OpenFOAM, DAFOAM (fluid solvers) and CalculiX (structural solver) coupled in the preCICE software for modeling FSI. OpenFOAM is an open-source and free CFD software designed largely in OpenCFD Ltd. In their investigations, researchers prefer open-source programs [27]. OpenFoam

is the most popular open-source application. It has a broad user base from both commercial and academic organizations in most fields of engineering and science [28]. OpenFOAM includes a wide range of capabilities for solving problems ranging from chemical processes, turbulence, and heat transport to acoustics, solid mechanics, and electromagnetics. It requires Linux (Ubuntu) Operating System but also can be used on Windows by creating a subsystem for Linux. The software is controlled using the command window with codes.

An object-oriented framework (DAFoam) is suggested to quickly construct the discrete adjoint technique for any steady-state OpenFOAM primal solver by adding or changing just a few hundred lines of source code, lowering the barrier for adjoint implementations [29]. DAFoam will be used for the implementation of VLES in it with the code modification and building dynamic link library.

PreCICE is an open-source software used for coupling used for multi-physics analysis, such as simulations of fluid-structure interaction [30]. Convenient techniques for data mapping, communication and transient equation coupling are provided by the software.

CalculiX is also an open-source finite-element analysis program. It has a pre-processor as well as the post-processor (CGX), also an explicit and implicit solver (CCX) [10]. It is used for building, calculating the Finite Element Models.

The coupling in a preCICE program is done by installing the special adapters for OpenFOAM and CalculiX. The developers of the preCICE design the adapters for fluid solvers for coupling them with the structural solvers for FSI. Furthermore, these adapters can be created by the users depending on their need. The fluid solvers include the Dirichlet boundary conditions, while the structural solvers include the Neumann boundary conditions [31]. The values of absolute displacement are read by fluid solvers and the forces are recorded. However, by structural solvers the forces are read, and the displacement is recorded.

Apart from these programs, there is a need for software like ParaView. The ParaView software will be used to visualize the obtained results from the simulations.

By applying the new entire study field methodology, it will be possible to fix the issues with the wind turbine analysis as it is currently done and gain a deeper comprehension of the real flow physics.

In order to be able to use the required software, there is a need for the Linux Operating System. The first step was to install it as a subsystem on Windows OS. It is done by using the WSL (Windows Subsystem for Linux), which permits the installation of Ubuntu 20.04.4 terminal environment. All of the operations are done by using this terminal. OpenFoam and CalculiX software were installed, and all of the dependencies were created. Following that, they were coupled in preCICE software. In order to check the functionality, the test cases with the perpendicular flap and elastic 3D tube were performed.

3.3.1. Coupling in preCICE

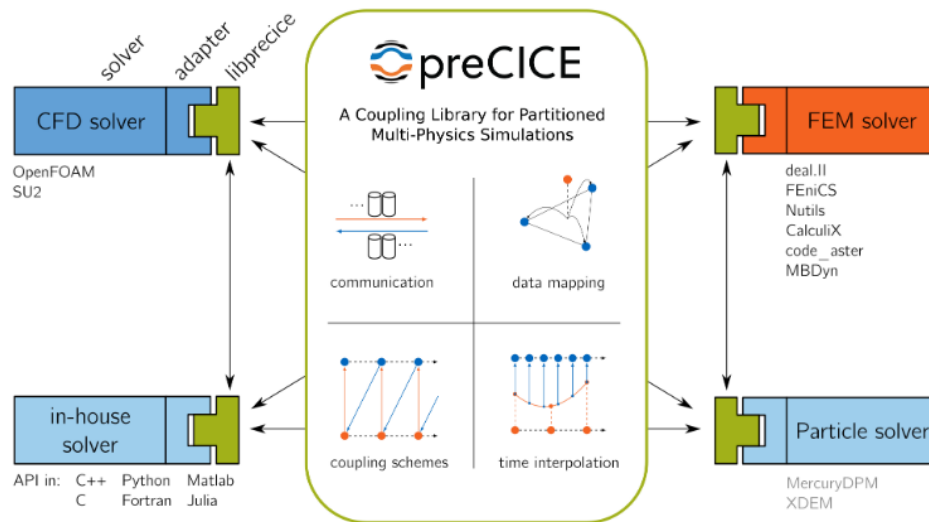


Figure 3.1. Concept of preCICE.[30]

The figure below represents the interface of the solid solver. As it can be seen that when it is runned, it waits for the fluid solver to start as well to perform the FSI. It also gives the message “Setting up primary communication to coupling partner/s”. The same thing can be seen on Figure 3.3, where the fluid solver is waiting for the solid solver to start. When the both solvers are runned, the FSI analysis starts, which is shown in Figure 3.4.

```
capstone@NLT058608NU: ~/preCICE/tutorials/perpendicular-flap/solid-calculix
3660
number of nonzero lower triangular matrix elements
63428

Starting FSI analysis via preCICE using the geometrically linear CalculiX solver...
Using up to 1 cpu(s) for the stress calculation.

Using up to 1 cpu(s) for the energy calculation.

Using up to 1 cpu(s) for the symmetric stiffness/mass contributions.

Factoring the system of equations using the symmetric spooles solver
Using up to 1 cpu(s) for spooles.

Using up to 1 cpu(s) for the stress calculation.

Using up to 1 cpu(s) for the energy calculation.

Setting up preCICE participant Solid, using config file: config.yml
---[preCICE] This is preCICE version 2.5.0
---[preCICE] Revision info: no-info [git failed to run]
---[preCICE] Build type: Release (without debug log)
---[preCICE] Configuring preCICE with configuration "../preCICE-config.xml"
---[preCICE] I am participant "Solid"
Using quasi 2D-3D coupling
Set ID Found
Read data 'force' found with ID # '3'.
Write data 'Displacement' found with ID # '2'.
---[preCICE] Setting up primary communication to coupling partner/s
-
```

Figure 3.2. Running solid solver (CalculiX).

```
capstone@NLT058608NU: ~/preCICE/tutorials/perpendicular-flap/fluid-OpenFOAM
Reading field p
Reading field U
Reading/calculating face flux field phi
Selecting incompressible transport model Newtonian
Selecting turbulence model type laminar
Selecting laminar stress model Stokes
No NRF models present
No fvModels present
No fvConstraints present
Constructing face velocity Uf
Courant Number mean: 0 max: 0
Starting time loop
---[preCICEAdapter] Loaded the OpenFOAM-preCICE adapter - v1.1.0.
---[preCICEAdapter] Reading preCICEDict...
---[preCICE] This is preCICE version 2.5.0
---[preCICE] Revision info: no-info [git failed to run]
---[preCICE] Build type: Release (without debug log)
---[preCICE] Configuring preCICE with configuration "../preCICE-config.xml"
---[preCICE] I am participant "Fluid"
---[preCICE] Setting up primary communication to coupling partner/s
```

Figure 3.3. Running fluid solver (OpenFOAM).

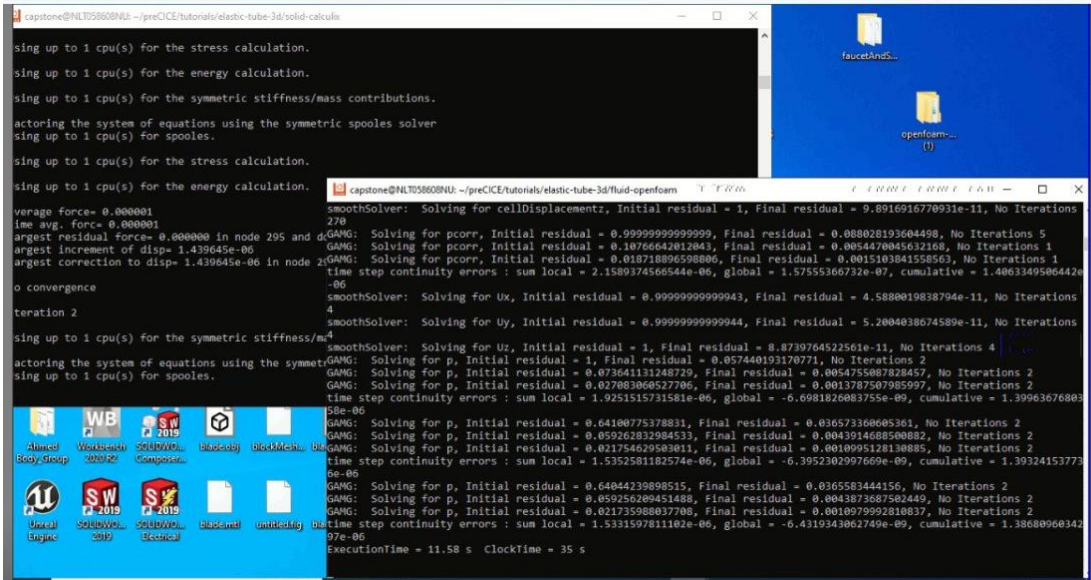


Figure 3.4. FSI analysis interface.

Chapter 4 – Results and Discussion

4.1. FSI Test Cases

4.1.1. Elastic 3D Tube

The FSI analysis requires two separate meshes for fluid and solid solvers. The meshes of the elastic 3D tube case are shown in Figures 4.1 and 4.2. A solid domain that surrounds a cylindrical fluid domain is used in the expanding tube test case. The fluid pressure causes the tube to expand, which subsequently contracts as the pressure is reduced.

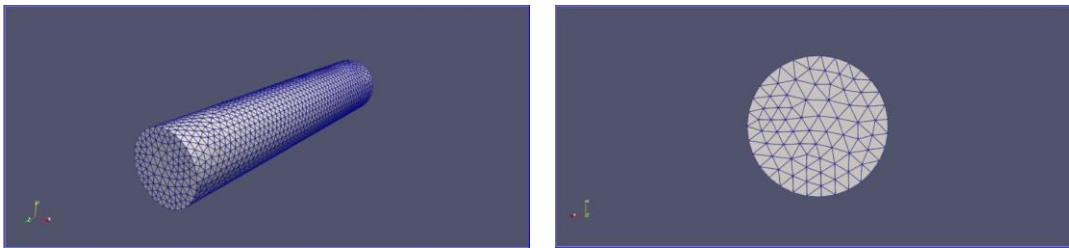


Figure 4.1. Mesh for fluid solver.

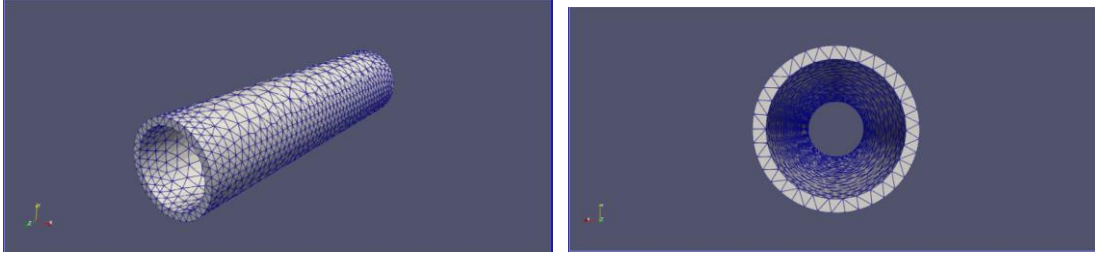


Figure 4.2. Mesh for solid solver.

The obtained results of this FSI case showed that the coupling of OpenFOAM and CalculiX was successful, and it is working properly. The figures below illustrate the obtained results. The tube deforms as the flow passes through it.

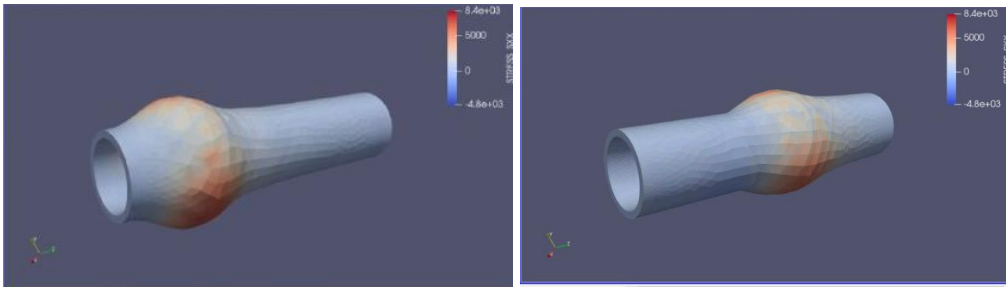


Figure 4.3. Flow traveling through the tube.

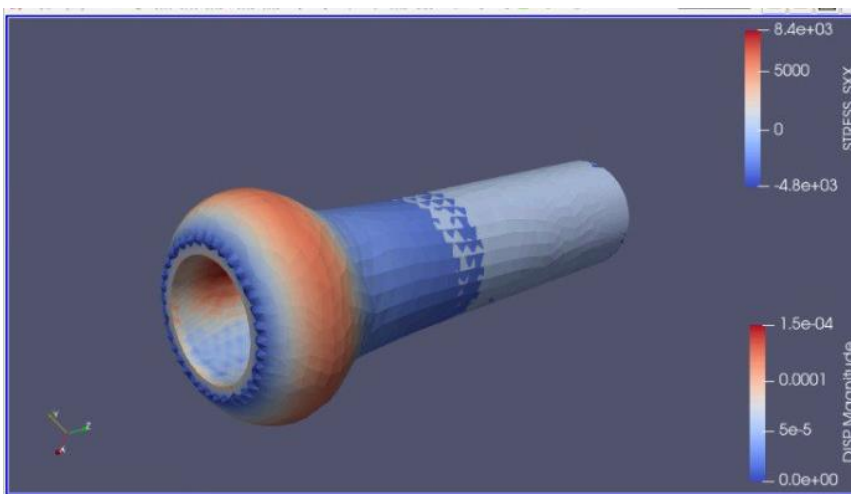


Figure 4.4. Deformation of the tube.

4.1.2. Perpendicular flap

This case includes a fluid flow through a channel in a 2D. There is a fixed flap on the floor of the channel. As fluid pressure increases on the flap's surface, the flap begins to oscillate. While the fluid participant reads displacement data, the solid participant reads forces at the interface. The same principle applies to the case of perpendicular flap. The fluid and solid meshes are shown in Figure 4.5 and 4.6.

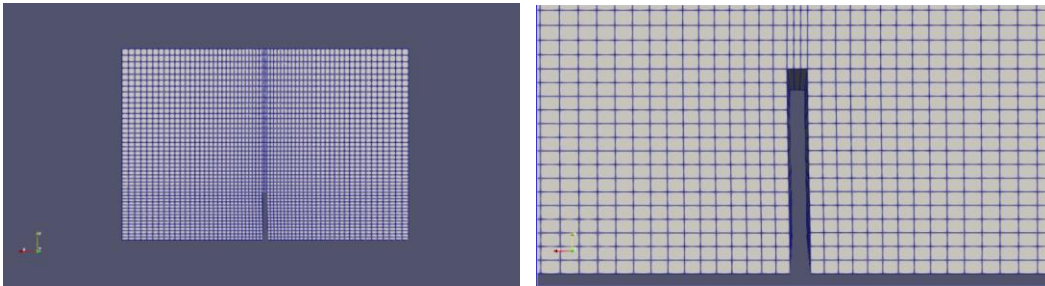


Figure 4.5. Mesh for fluid domain.

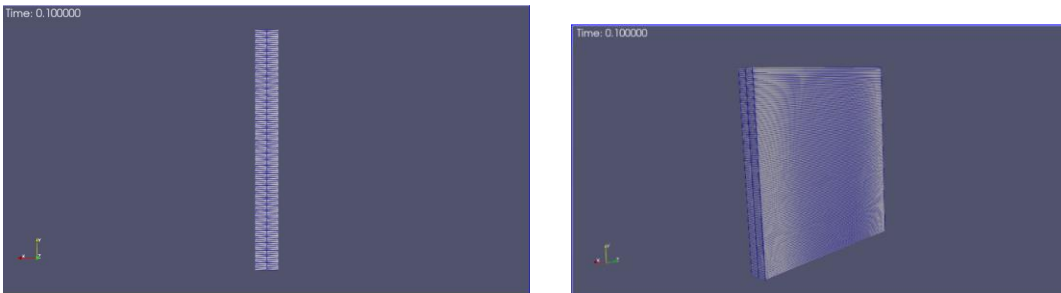


Figure 4.6. Mesh for flap.

The results of the case are represented in the figures below. As it can be seen, the fluid domain has undergone the displacement, except the area where the flap was placed. So, the flow hits the flap, then it deforms. This case also shows that the coupling in preCICE is operating correctly and can be applied to further works.

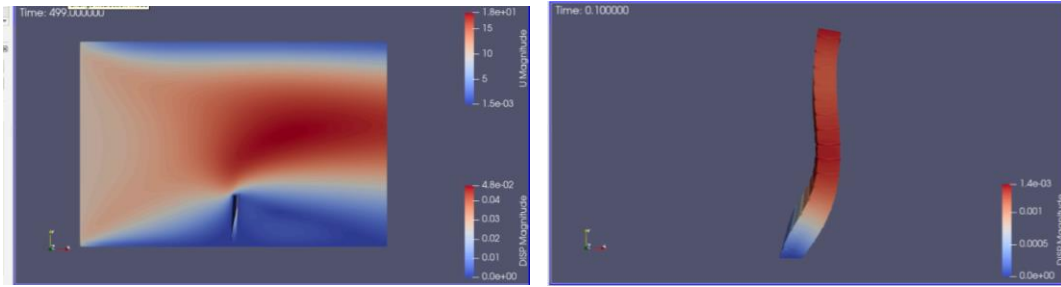


Figure 4.7. Deformation of the geometry.

The final deformation of the flap at 499 sec is shown in Figure 4.8.

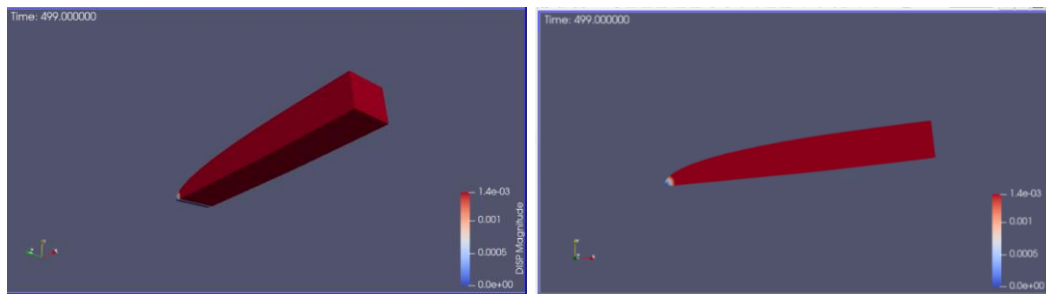


Figure 4.8. Deformation of the flap.

4.2 NREL Phase VI wind turbine

4.2.1. Fluid model

The goal was to run two solvers independently to see if they are operating properly and can be used conducting FSI. The geometry of the NREL Phase VI wind turbine was generated with regards to the dimensions from the work of Abate et al. [32]. The mesh was generated in OpenFOAM with the spherical fluid domain using the BlockMeshDict. Inside this fluid domain, there is a mesh of the blades. The fluid domain is used to study the fluid flow around the blades.

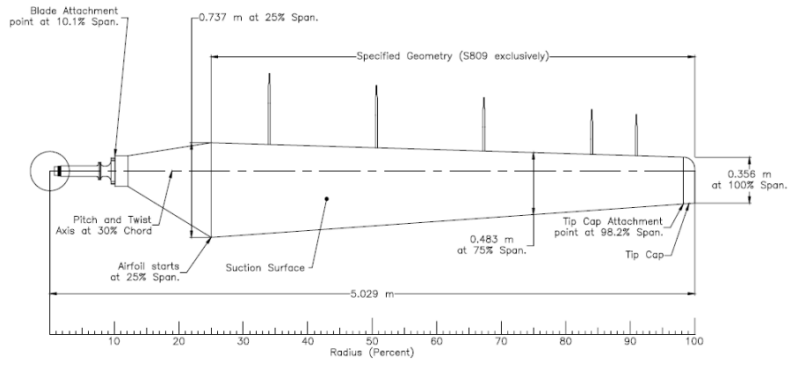


Figure 4.9. Blade dimensions [32].

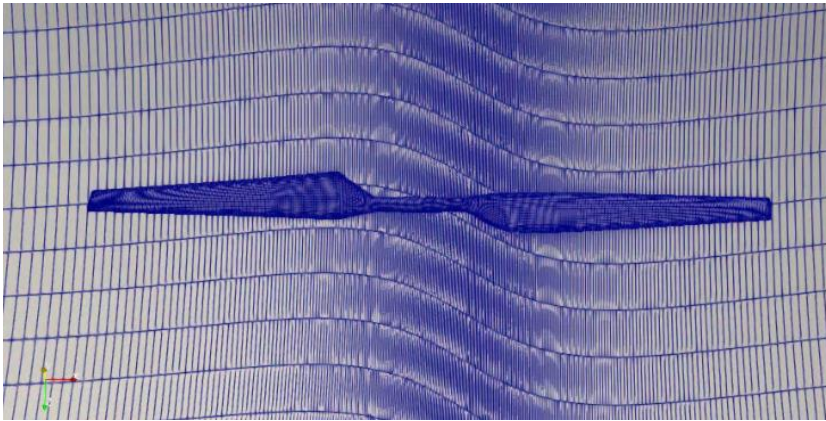


Figure 4.10. Blades.

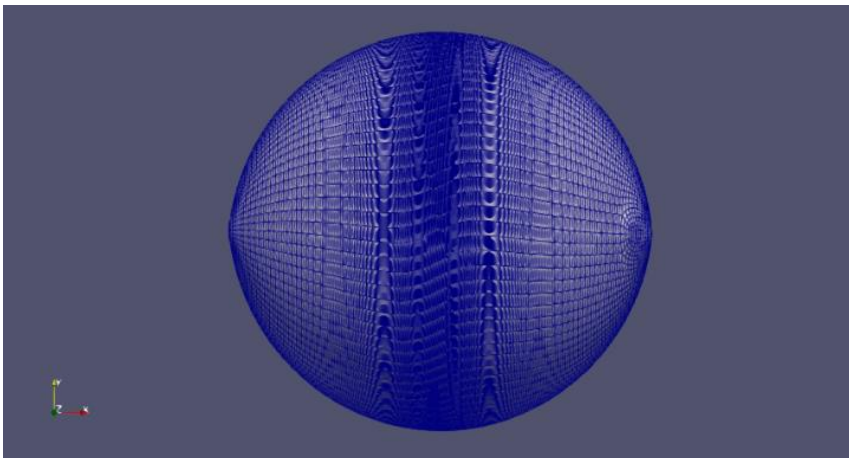


Figure 4.11. Fluid domain.

After running the OpenFoam solver, the following results were obtained. The wind speed was set to 7 m/s. The turbulence model k-w was applied. The blade is fixed in the middle to avoid the translation, and allow the rotation, for which the rotating frame approach was applied.

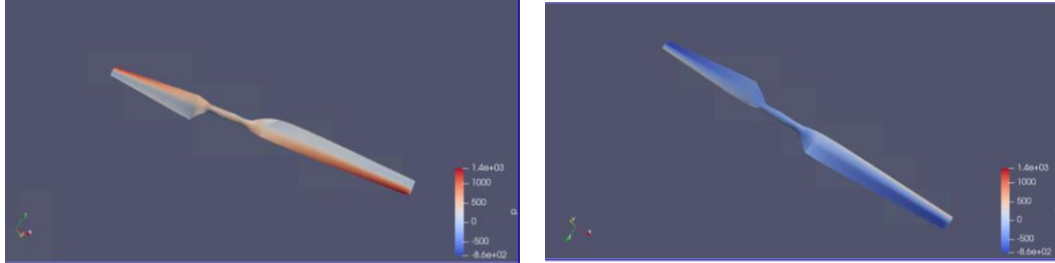


Figure 4.12. Pressure on the blades.

4.2.2. Solid model

For the generation of the mesh for solid solver it was necessary to use the ANSYS as well as Salome software. Firstly, the mesh for NREL Phase VI wind turbine was generated in Ansys software. The tetrahedrons meshing method is utilized. Then, the mesh was transferred to Salome software for definition of the necessary nodes and to transfer the mesh to CalculiX. The final mesh can be seen in Figure 4.13. Nevertheless, the mesh was changed several times because of the error occurring, when running the solver. The error description writes “there nonpositive Jacobian determinant in element. It was solved by using the tetrahedrons and making the mesh finer.

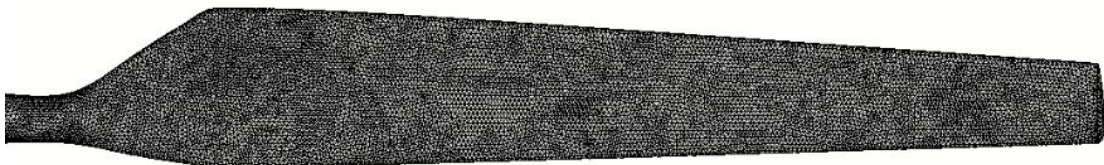


Figure 4.13. Mesh for solid solver.

The aim was to create a dynamic simulation of the rotating blades in a solid solver. By applying the force at the tip of the blade, it was tried to create the rotation and see the

displacements. One node was created in order to fix the blades from the translation in x, y, z axes as well as allow rotations.

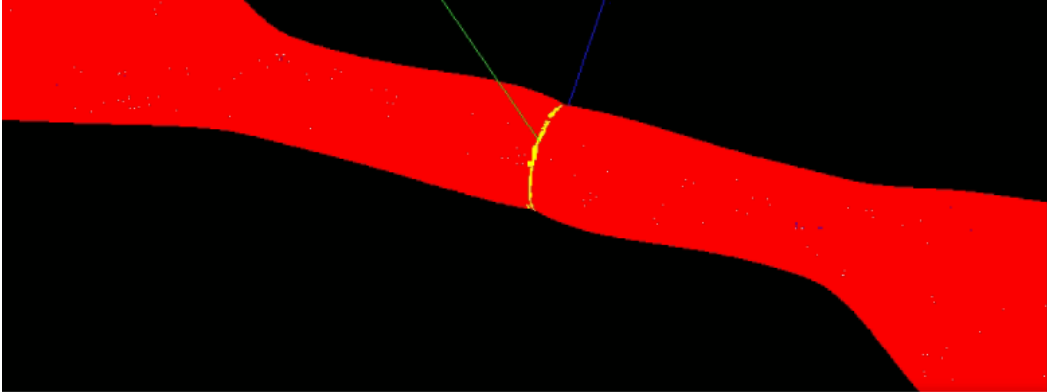


Figure 4.14. Fixed part.

Another node was created for the application of the force, which is shown in Figure 4.15. The force was applied in the direction of the Z axis with the magnitude of 1000 N.

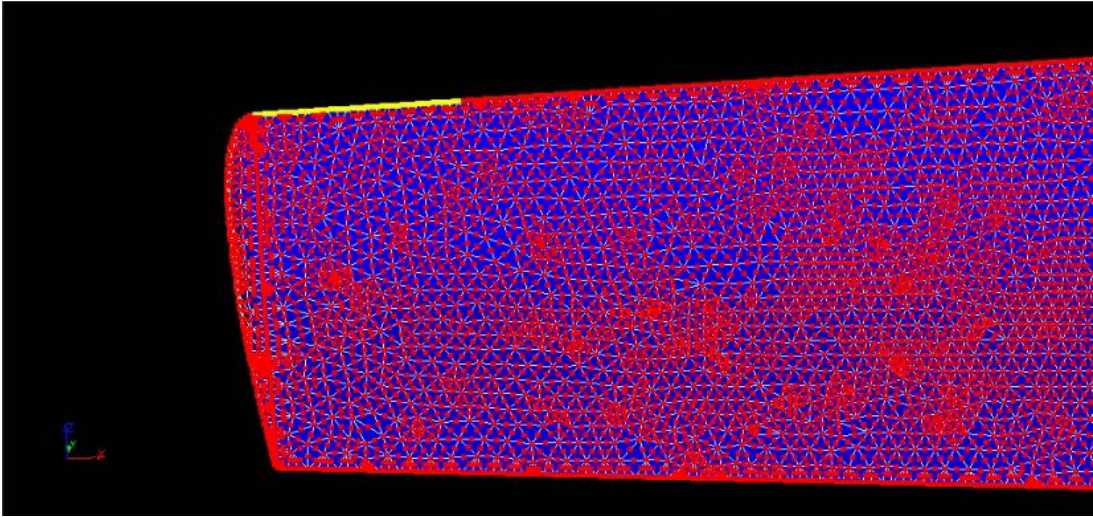


Figure 4.15. Point of load.

After setting up the case, the simulation in the CalculiX solver was started. The results of the simulation are represented in Figure 4.17. It can be seen that there is a rotational motion. Furthermore, the displacements are also shown.

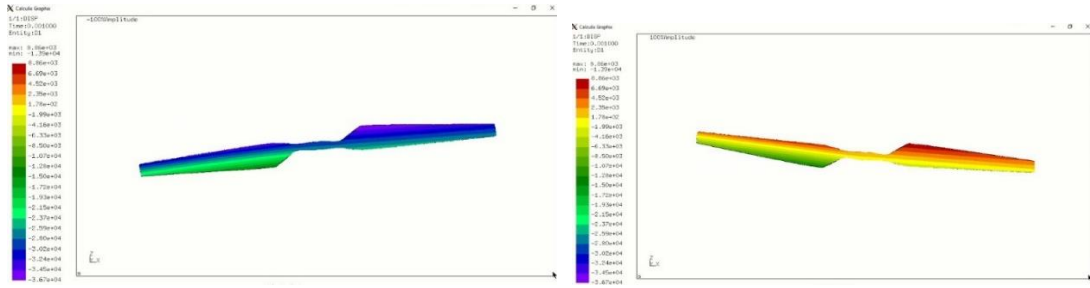


Figure 4.16. Simulation in CalculiX.

Chapter 5 – FSI Trials

5.1. FSI run

5.1.1. Initial trials

The first FSI run ended with an error. The terminal report showed `std::bad_alloc`, which appears when there is not enough memory to run the simulation. One of the main reasons was using the solid geometry blade. In order to solve this issue, it was decided to use the hollow blade.

```

Read data 'Force' found with ID # '3'.
Write data 'DisplacementDelta' found with ID # '2'.
--[precice] Setting up primary communication to coupling partner/s
--[precice] Primary ranks are connected
--[precice] Setting up preliminary secondary communication to coupling partner/s
--[precice] Receive global mesh Fluid-Mesh-Faces
--[precice] Prepare partition for mesh Solid-Mesh
--[precice] Gather mesh Solid-Mesh
--[precice] Send global mesh Solid-Mesh
--[precice] Setting up secondary communication to coupling partner/s
--[precice] Secondary ranks are connected
terminate called after throwing an instance of 'std::bad_alloc'
what(): std::bad_alloc
[NLT058608NU:14099] *** Process received signal ***
[NLT058608NU:14099] Signal: Aborted (6)
[NLT058608NU:14099] Signal code: (-6)

```

Figure 5.1. Terminal output.

The shell element blade was meshed using the Ansys software that is illustrated in Figure 5.2. Following that, the FSI was set up using this mesh. However, the FSI was stopped due to the floating-point exception error, which is shown in Figure 5.3.

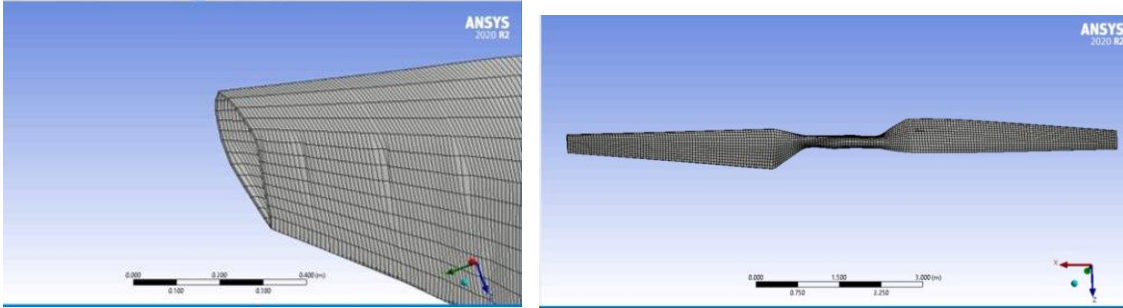


Figure 5.2. Shell model mesh.

```
#3 Foam::GaussSeidelSmoother::smooth(Foam::word const&, Foam::Field<double>&, Foam::lduMatrix const&, Foam::Field<double>
const&, Foam::UPtrList<Foam::lduInterfaceField const> const&, unsigned char, int) at ??:?
#4 Foam::GaussSeidelSmoother::smooth(Foam::Field<double>&, Foam::Field<double> const&, unsigned char, int) const at ??:?
#5 Foam::smoothSolver::solve(Foam::Field<double>&, Foam::Field<double> const&, unsigned char) const at ??:?
#6 Foam::fvMatrix<double>::solveSegregated(Foam::dictionary const&) at ??:?
#7 Foam::fvMatrix<double>::solve(Foam::dictionary const&) in "/opt/openfoam9/platforms/linux64GccDPInt32Opt/bin/pimple
#8 Foam::fvMatrix<double>::solve() in "/opt/openfoam9/platforms/linux64GccDPInt32Opt/bin/pimpleFoam"
#9 Foam::SolverPerformance<double> Foam::solve<double>(Foam::tmp<Foam::fvMatrix<double> > const&) at ??:?
#10 Foam::RASModels::SpalartAllmaras<Foam::IncompressibleMomentumTransportModel<Foam::kinematicTransportModel> >::corr
#11 ? in "/opt/openfoam9/platforms/linux64GccDPInt32Opt/bin/pimpleFoam"
#12 _libc_start_main in "/lib/x86_64-linux-gnu/libc.so.6"
#13 ? in "/opt/openfoam9/platforms/linux64GccDPInt32Opt/bin/pimpleFoam"
Floating point exception
capstone@MLT058608NU:~/preCICE/tutorials/test/Aero_shape_twist$ _
```

Figure 5.3. Floating point exception.

The structural solver transfers the data to the fluid solver, and it deforms the CFD mesh. When the deformation is too high, the OpenFoam exits with the floating-point exception error. Apart from that it also can happen because of the misalignment of the fluid and solid geometries.

5.1.2. Misalignment of the geometries

The geometries then were checked for misalignment in Paraview. It was found that the geometry in the structural solver was significantly greater than the one in the fluid solver. This issue happened due to transferring the mesh between the software, and the units did not match.

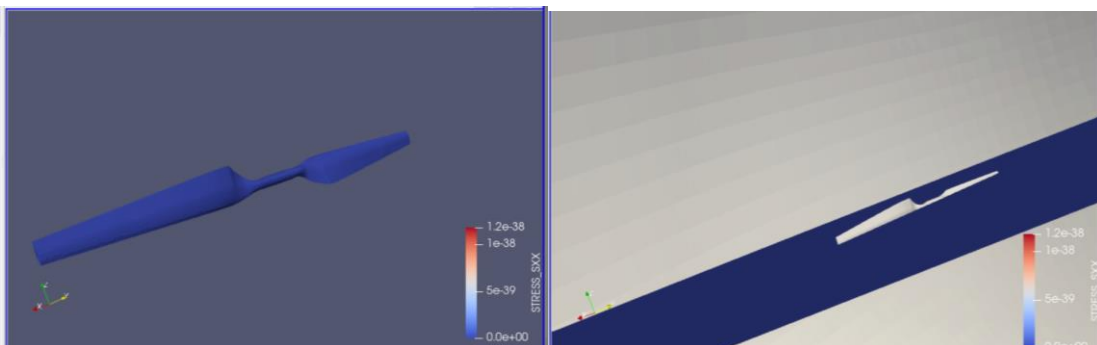


Figure 5.4. Comparison of the geometries.

After that a new structured mesh was created for the structural solver. It is represented on the right in the figure below.

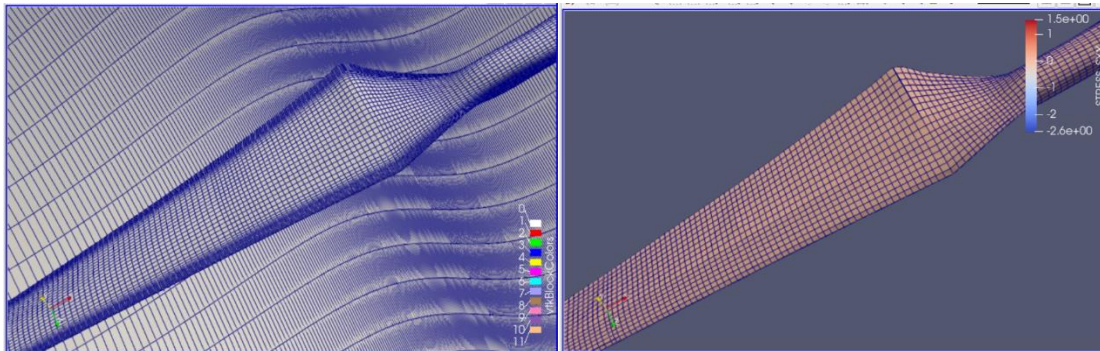


Figure 5.5. Meshes for fluid and structural solvers, respectively.

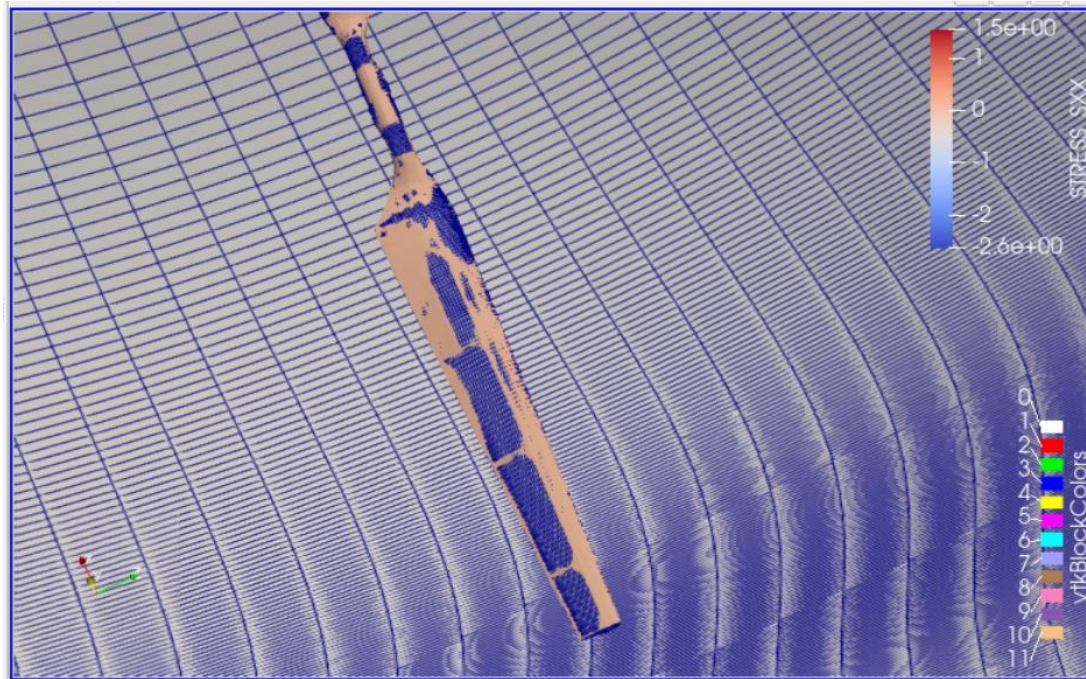


Figure 5.6. Combined view of the meshes.

Then, in order to make the meshes exactly the same and not have an issue with the misalignment, it was decided to transfer the OpenFoam mesh to Calculix mesh. It was accomplished by using the *FoamToFluent* utility in OpenFoam and by saving the Ansys Fluent mesh in UNV format for CalculiX.

Prepomax is a software tool that can be used in conjunction with the finite element analysis (FEA) software CalculiX. Prepomax is designed to simplify the process of generating finite element models for analysis in CalculiX. Using this program, the shell element model for CalculiX was created, and other input parameters were set up.

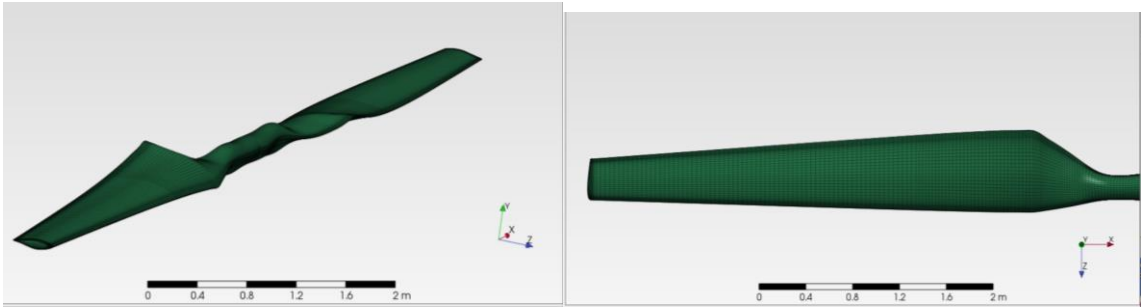


Figure 5.7. Transferred mesh.



Figure 5.8. Comparison of geometries.

Although the geometries did not mismatch and the meshes were the same, the floating-point exception error did not disappear. The next step was to decrease the deformations by using the blade with the spars inside.

5.1.3. Geometry with a spar

The first try was done by using the geometry that included one spar inside. The mesh is represented in the figure below. However, the implementation of this model also resulted in floating point exception error.

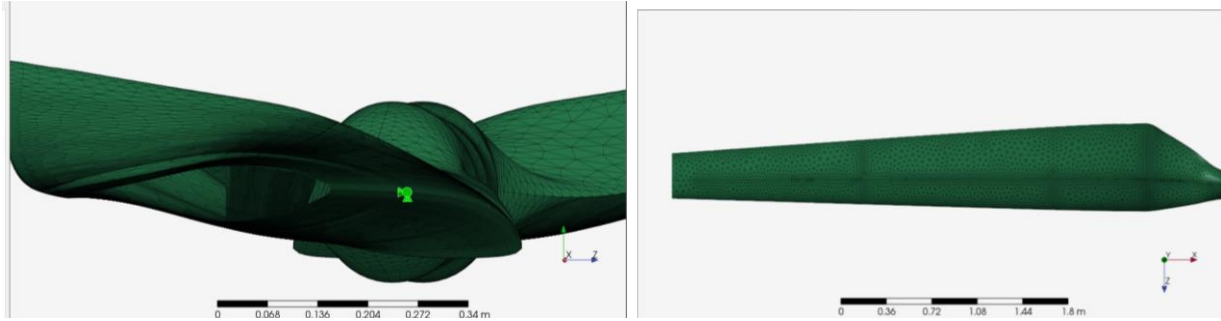


Figure 5.9. Geometry with one spar.

5.1.4. Geometry with three spars

The geometry with three spars inside was meshed and a case was set up. It is illustrated in Figure 5.11. Then, the one-way FSI run was conducted, Nevertheless, the two-way FSI was still showing the floating-point exception error.



Figure 5.10. Blade with three spars.

5.1.5. Software Limitation

After several trials and spending a lot of time, it was found that the combination of the preCICE software with OpenFOAM and CalculiX for two-way dynamic fluid-structure interaction (FSI) simulations involving rotational motion encountered challenges and limitations. Although Precice, OpenFOAM, and CalculiX are widely used and capable software tools individually, their integration for this specific scenario presented difficulties. Precice is a coupling library that facilitates the communication between different simulation codes, enabling FSI simulations. OpenFOAM is a popular open-source CFD software, while CalculiX is a finite element analysis

(FEA) solver for structural mechanics. Both OpenFOAM and CalculiX are versatile and widely used in their respective domains. CalculiX, as a structural analysis solver, focuses on simulating solid mechanics and finite element analysis. While it is a powerful tool for static structural simulations, it may have limitations when dealing with dynamic or rotational motion in the context of FSI simulations. Rotational motion introduces additional complexities such as the need to account for centrifugal forces, mesh deformation, and accurate coupling with the fluid domain. Therefore, it was necessary to switch to the other method. Then, it was decided to use the QBlade software to conduct low-fidelity FSI.

Chapter 6 – Low-Fidelity FSI

6.1. Overview

"Low fidelity FSI" typically refers to "low fidelity fluid-structure interaction." Fluid-structure interaction (FSI) is a computational method used to simulate the interaction between a fluid and a solid structure. It involves solving the equations that govern fluid flow and structural deformation simultaneously to predict the behavior of the coupled system. In FSI simulations, fidelity refers to the level of detail and accuracy of the modeling and analysis. Low fidelity FSI implies that the simulation is conducted with simplified or approximate models, which may sacrifice accuracy in exchange for computational efficiency or feasibility [33]. This approach is often used when the primary focus is on obtaining quick and approximate results or when high-fidelity simulations are not necessary. Low fidelity FSI may involve simplifications such as using reduced-order models, lumped-parameter models, or simplified fluid and structural models. These approximations can help reduce the computational complexity and simulation time, making it feasible to analyze larger systems or perform preliminary design assessments. However, it's important to note that low fidelity simulations may not capture all the intricate details and phenomena present in the real-world system and may provide less accurate results compared to high fidelity simulations.

A low-fidelity flow field typically involves simplified or reduced-order models that are computationally less demanding but may sacrifice accuracy and detail. The specific equations used in low-fidelity simulations can vary depending on the level of simplification. Euler's equations are a simplified form of the Navier-Stokes equations that neglect viscous forces [34].

They are commonly used in inviscid flow simulations, where the fluid viscosity is assumed to be negligible. The 3D form of Euler's equations for an inviscid flow are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \quad (14)$$

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{\nabla P}{\rho} \quad (15)$$

Where: ρ is the fluid density, t is time, \mathbf{V} is the velocity vector field, P is the pressure, ∇ is the del operator, and $\partial/\partial t$ is the partial derivative with respect to time. Euler's equations are computationally less expensive than the full Navier-Stokes equations since they neglect the complexities of viscosity and turbulence.

Potential flow theory assumes that the flow is irrotational and incompressible, which allows for further simplification of the governing equations [35]. Potential flow is suitable for cases with low-speed flows and where the effects of viscosity are negligible. The potential flow equation:

$$\nabla \times \mathbf{u} = 0 \quad (16)$$

where \mathbf{u} is the velocity vector

By the vector identity:

$$\nabla \times (\nabla \phi) = 0 \quad (17)$$

where ϕ is a scalar function, combining (1) and (2) leads to:

$$\mathbf{u} = \nabla \phi \quad (18)$$

where the velocity potential is referred to as ϕ .

Because the flow is incompressible, a divergence of the velocity provides mass conservation:

$$\nabla \cdot \mathbf{u} = 0 \quad (19)$$

equation (3) is incorporated into (4) to produce:

$$\nabla \cdot (\nabla \phi) = 0 \quad \text{Or} \quad \Delta \phi = 0 \quad (20)$$

Solving the potential flow equations can provide valuable insights into the overall flow behavior, but it neglects important aspects like boundary layers and flow separation that are crucial in high-fidelity simulations. Keep in mind that these low-fidelity flow field equations provide a basic representation of the fluid flow and may not capture all the complexities present in real-world scenarios. However, they serve as valuable approximations when computational resources or time constraints are limiting factors or for quick initial assessments.

6.2. QBlade

QBlade is an open-source wind turbine design and simulation software developed primarily for the aerodynamic analysis and design of wind turbines [30]. QBlade provides a range of tools and features to assist engineers and researchers in designing and optimizing wind turbine rotor blades. QBlade allows users to define the geometry of wind turbine blades, including airfoil shapes, chord lengths, twist angles, and tip shapes. It provides a graphical interface for easy blade design and modification. The software comes with an extensive database of airfoil shapes that users can choose from for their blade designs. It includes popular airfoil profiles used in wind turbine applications. QBlade utilizes CFD algorithms to simulate the flow of air over wind turbine blades. It calculates important aerodynamic parameters such as lift, drag, and moment coefficients, as well as power and thrust coefficients. The software can estimate the performance characteristics of wind turbines, including power output, thrust, and efficiency, based on the blade geometry and operating conditions. QBlade includes optimization algorithms that enable users to automatically optimize blade designs to achieve specific performance goals. It can adjust various design parameters, such as blade twist, taper, and airfoil selection, to maximize turbine performance. QBlade is a valuable tool for wind turbine designers and researchers, allowing them to explore different blade configurations and optimize wind turbine performance. It is widely used in the wind energy industry and academic institutions for both educational and research purposes.

6.2.1. Aero-elastic coupling

Fully aeroelastic simulations are possible in QBlade [36] owing to the link to the open-source multi-physics engine Project Chrono [37]. The Euler-Bernoulli beams used to simulate the

beam components of the wind turbine model are put up in a multi-body corotational framework. The structural framework assigns constant parameters along a single beam element.

The flowchart of aero-elastic coupling is shown in Figure 6.1. Time $t = t_0$ denotes the starting point of the simulation. Initially a search for a distribution of the convergent circulation for the blade's bound vorticity is conducted. Due to the unchanging wake shape during the evaluation, the costly process of the wake-induced velocities calculation for the panels of the rotor is only performed once at the start of the iteration. In this iteration, the unstable aerodynamics model along with different corrections are utilized in the computation of the lift and drag properties of the blade panels. The calculations are complete for the blade aerodynamics once a convergent solution is achieved for the bound circulation [36].

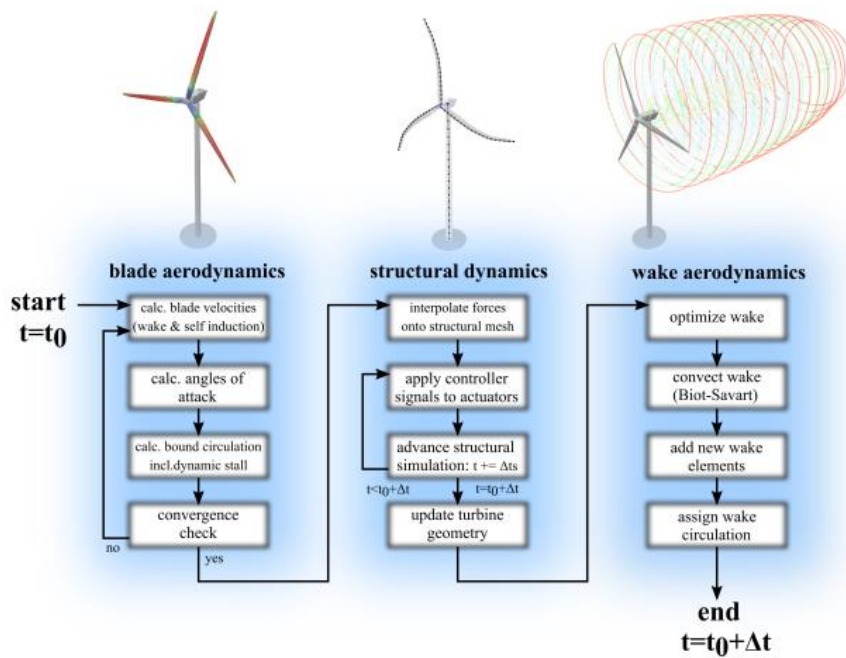


Figure 6.1. The flowchart of aero-elastic coupling [36]

The lift, drag, and moment parameters of the airfoil are used to interpolate the blade panel forces and moments onto the structural dynamics' discretization from the aerodynamic discretization. Additionally, the actuators receive the controller signals. The structural time step t_s has advanced the simulation at this point. The simulation of the structural dynamics is complete when the time reaches $t = t_0 + \Delta t$. The structural simulation is increased once more with a step size of Δt_s if $t < t_0 + \Delta t$, adding new controller signals, while the gradual rotor advancement

rotates the aerodynamic forces. This is done over and over again until the simulation time reaches $t = t_o + \Delta t$. The aerodynamic model then proceeds to move to its final position for this time step by interpolating back the locations of rigid bodies and beam components onto the aerodynamic mesh. The wake is updated in the final stage. By eliminating or combining wake elements, the optimization of the wake discretization is done [36].

6.2.2. Lifting Line Free Vortex Wake Method

One of the methods implemented in QBlade for aerodynamic analysis is the Lifting Line Free Vortex Wake (LLFVW) method. The LLFVW method is a mathematical approach used to analyze and predict the aerodynamic performance of lifting surfaces, such as wings or rotor blades, in steady-state conditions [37]. It is particularly useful for analyzing the performance of finite wings with induced flow effects.

In QBlade, the LLFVW method is used to calculate the aerodynamic loads and performance of wind turbine blades [38]. It employs a lifting line model to approximate the blade as a series of lifting line elements along the span. These lifting line elements generate bound vortices that represent the circulation distribution on the blade. The LLFVW method in QBlade takes into account the influence of the bound vortices on the flow field and calculates the induced velocities caused by the vortices shed from each lifting line element. These induced velocities are combined with the freestream wind velocity to determine the effective velocity at each element. Based on the effective velocity and the local geometry of the blade section, QBlade calculates the lift and drag coefficients for each element. These coefficients are then used to determine the aerodynamic forces and moments acting on the blade, which are essential for assessing the turbine's performance and behavior. QBlade also incorporates other features and models to enhance the accuracy of the analysis, such as 3D correction factors, tip loss correction, and blade element momentum theory (BEM) integration [39]. These additional factors help to improve the prediction of performance, accounting for effects like tip losses and three-dimensional flow.

A vortex model of the flow field and inviscid potential flow concept serve as the foundation for the LLFVW technique [30]. The discretized components of the blade are shown as bound ring vortices. The vortex filament equation is used to determine the bound circulation distribution along the lifting line elements. This equation relates the circulation around the lifting line to the induced velocity. A vortex ring made up of four parallel vortex filaments represents each blade panel.

Based on the relative inflow velocity and the drag and lift coefficients obtained from tabulated data of airfoil, the bound vortex lines the circulation that makes up the lifting line is estimated. The Kutta-Joukowski theorem is used to calculate the sectional circulation calculation $\partial\Gamma$:

$$\partial F_L(\alpha) = \rho V_{rel} \times \partial\Gamma \quad (21)$$

The fluid density ρ affects the formulation for the sectional lift force, ∂F_L . A combination of the velocity of free stream V_∞ , the induced velocity V_{ind} , and the blade's motion V_{mot} yields the relative velocity V_{rel} . The Biot-Savart equation is used to calculate the induced velocity that accounts for the effect of all vortex components in the wake and on the blade:

$$V_{ind} = -\frac{1}{4\pi} \int \Gamma \frac{\vec{r} \times d\vec{l}}{r^3} \quad (22)$$

The circulation distribution across the blade is estimated at the beginning of every time interval. The blade element theorem and the Kutta-Joukowski theorem's anticipated forces are aligned by employing an iterative technique to do this. The effect of wake components on the blade is determined solely once throughout the iteration, while only the bound vorticity distribution is updated. The blade rotation progresses for a single time interval when convergence is reached. The local induced velocity as well as inflow are transferred along with all free wake vortex components. The Kutta condition is used to compute the circulation and allocate it to the vortex lines that are newly generated:

$$\Gamma_{trail} = \frac{\partial\Gamma_{bound}}{\partial x} \Delta x \quad (23)$$

$$\Gamma_{shed} = \frac{\partial\Gamma_{bound}}{\partial t} \Delta t \quad (24)$$

6.2.3. The Chrono Library

The multi-physics engine Project Chrono is the foundation for the structural model in QBlade, which uses its FEA module. The Chrono library provides a comprehensive set of tools and functionalities for modeling and simulating mechanical systems with a focus on multibody dynamics and structural analysis [40]. Within the Chrono library, the structural model represents the physical components and their interactions in a mechanical system. Chrono is part of the C++ Standard Library and provides a range of features for managing and manipulating time-related

data. It includes functionalities for measuring time intervals, working with clocks, calculating durations, and performing conversions between different time units. To solve the finite element problem, the EIGEN C++ template library's SparseLU solver is employed. This solver is utilized within the context of the Chrono module, which is compiled into a dynamic link library (DLL) sourced from the Project Chrono GIT repository. In order to incorporate the necessary functionalities, relevant header files from both Project Chrono and the EIGEN library are included in the source code. Additionally, the Chrono DLL is linked to QBlade's source code, allowing for the utilization of the Chrono module and its associated features within the QBlade simulation. This integration of the SparseLU solver, Chrono module, and QBlade's source code enables the solution of the finite element problem, utilizing the capabilities of the EIGEN library's SparseLU solver to efficiently compute the desired results [37]. This integration makes it possible to define the physical framework, to use finite elements, and to gain access to the solver, allowing QBlade to simulate structural dynamics in the time domain. The simulations were done for NREL 5MW and NREL Phase VI wind turbines in the Qblade software. The FSI in Qblade is done by a coupling of the LLTFVW+Chrono solvers.

6.3. Results in QBlade

6.3.1. NREL 5 MW wind turbine

The NREL 5 MW wind turbine refers to a specific type of wind turbine that was developed by the National Renewable Energy Laboratory (NREL), which is a research institution based in the United States. The 5 MW designation represents the turbine's power rating, indicating that it is designed to generate a maximum power output of 5 megawatts [41]. The NREL 5 MW wind turbine is a horizontal-axis wind turbine (HAWT), which means that its rotor rotates around a horizontal axis parallel to the ground. It features a three-blade design, where the blades capture the kinetic energy from the wind and convert it into mechanical energy. This mechanical energy is then further converted into electrical energy through a generator housed within the turbine. The NREL 5 MW wind turbine is a significant development in the field of wind energy, as it represents a high-capacity turbine with improved efficiency and reliability [41]. It has been extensively tested and used as a benchmark by researchers and industry professionals to assess and evaluate wind turbine technologies. The data and insights gained from studying the NREL 5 MW wind turbine have contributed to advancements in wind energy research, turbine design, and the overall

understanding of wind energy systems. The geometry of the created NREL 5MW wind turbine blade is shown in Figure 6.2.

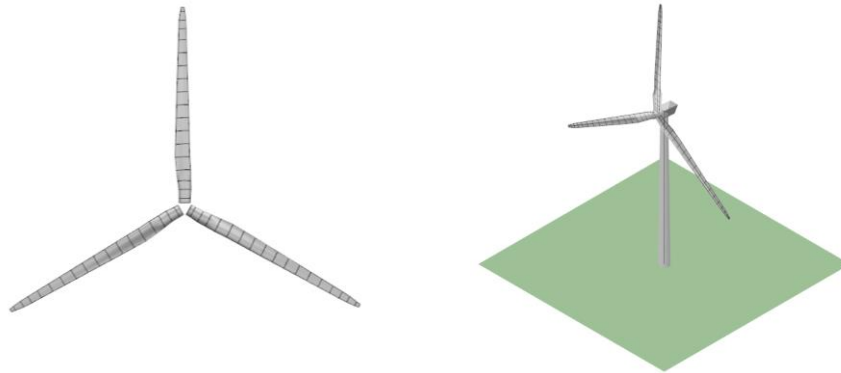


Figure 6.2. NREL 5MW wind turbine

The internal structure of the blades is illustrated in Figure 6.3. The blades are hollow with spar.

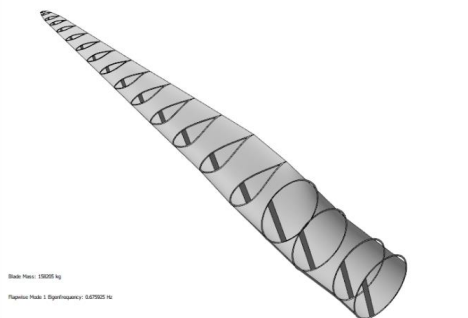


Figure 6.3. Internal Structure of the blades.

The simulation for NREL 5 MW wind turbine was conducted with 12.1 rpm, and a windspeed of 10 m/s. Figure below illustrates the structural loading of the blade, indicating that the regions of leading and trailing edges are under highest stress. The highest stress magnitude is 2.93 MPA. The simulation results show an edgewise tip displacement of - 0.963159 mm and a flapwise tip displacement of 24.8037 mm.

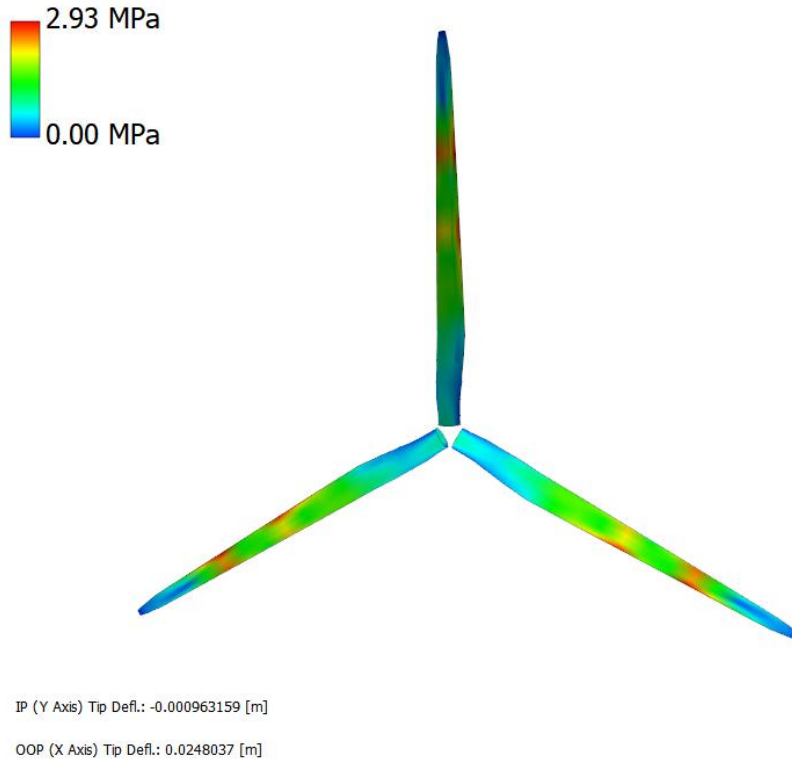


Figure 6.4. Structural loadings on the blades.

The pattern of vorticity throughout the flow field over a simulation is depicted by the vorticity contour. The regional rotational motion of the particles of the fluid in a flow is measured as vorticity. The vorticity contour generated by the QBlade modeling offers important insights concerning the flow dynamics near the blade [37]. It depicts the strength and shapes of the vortices that result from the blade's contact with the air around it. In general, a color map overlaid cross-sectional plane represents the vorticity contour. A variety of colours are used to indicate different vorticity magnitudes or intensities. The matching vorticity values for each colour are displayed in the colour chart. The resulting vortices can significantly affect the aerodynamic efficiency by increasing drag, altering the distribution of lift, or generating flow separation. In QBlade, the contours of vorticity can be used to evaluate the effectiveness of the design of the blade, spot areas with flow separation as well as high vorticity, and comprehend the flow phenomena which affect the performance in general. The contours of vorticity are shown in Figure 6.5.

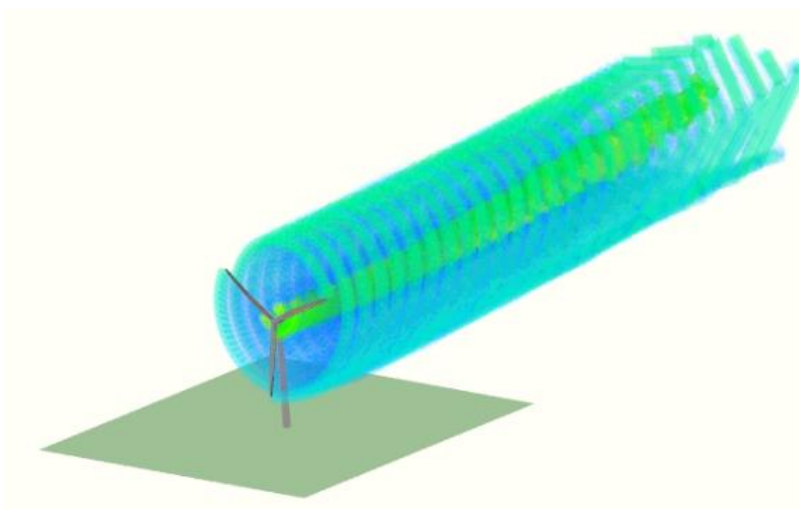


Figure 6.5. Vorticity contours.

The power coefficient and tip speed ratio are two important parameters used in wind turbine design and analysis. The power coefficient is a measure of how effectively a wind turbine converts the kinetic energy of the wind into electrical power. It is defined as the ratio of the actual power extracted by the wind turbine to the maximum possible power that can be extracted from the wind [37]. The power coefficient (C_p) is calculated using the following formula:

$$C_p = \frac{P}{\frac{1}{2}\rho AV^3} \quad (25)$$

Where: P is the power output of the wind turbine, ρ is the density of the air, A is the rotor swept area ($\pi * R^2$; R = rotor radius), V is the wind speed.

The maximum possible power that can be extracted from the wind is given by the Betz limit, which is approximately 59.3% (or 0.593) of the total kinetic energy in the wind. Therefore, the power coefficient (C_p) typically ranges from 0 to 0.593, with higher values indicating better conversion efficiency.

The tip speed ratio is a dimensionless parameter that compares the rotational speed of the wind turbine rotor to the speed of the wind at the rotor's tip. It is defined as the ratio of the tangential speed of the rotor blade tip to the wind speed [37]. The tip speed ratio (λ) is calculated using the following formula:

$$\lambda = \frac{R\omega_r}{V} \quad (26)$$

Where: ω is the rotational speed of the wind turbine rotor, R is the radius of the rotor, V is the wind speed.

The tip speed ratio affects the performance of the wind turbine. Different wind turbine designs have different optimal tip speed ratios for maximum power output. Generally, the tip speed ratio is kept within a range where the turbine operates efficiently without encountering excessive aerodynamic losses or structural limitations. It's important to note that both the power coefficient and tip speed ratio are interrelated and play a crucial role in determining the overall performance and efficiency of a wind turbine.

From the figure below, it is seen that the largest power coefficient was about 0.48, which was reached at tsr of 8. It started decreasing afterwards. In the work of Jonkman et al. [42], the results were close to these obtained results. The highest C_p was 0.482 at tsr of 7.55. In another work [43], the C_p of 0.49 was reached at a tsr of 7.55. The graph below represents the results from this work.

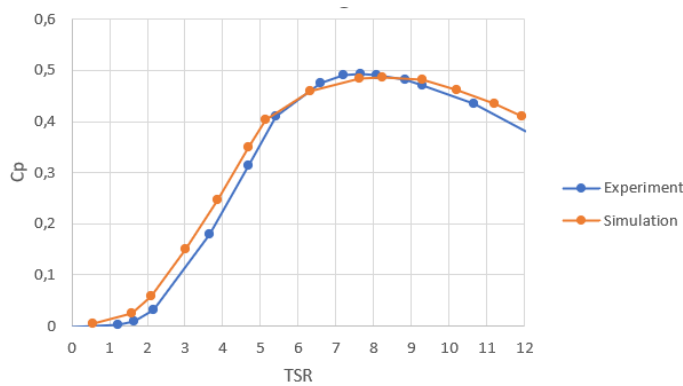


Figure 6.6. C_p vs tsr graph.

The twist angle is found out to be higher closer to the center of the blades with almost 14 degrees, which is illustrated in Figure 6.7. The twist angle from the simulation is compared with the experimental twist angle from the work of Vesel [34]. From the Figure 6.7, it is shown that the results are almost identical, except for some small differences. In both cases, the twist angle decreases by radius.

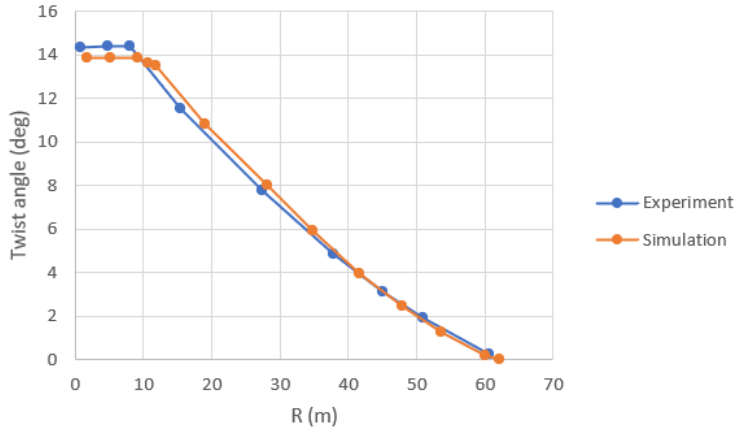


Figure 6.7. Twist angle vs R graph.

6.3.2. NREL Phase VI wind turbine

After conducting analysis for NREL 5MW wind turbine, another simulation was conducted for NREL Phase VI wind turbine. The NREL Phase VI wind turbine is well-recognized in the field of wind turbines. It is a horizontal axis wind turbine that has two blades. For this simulation windspeed was set to 10 m/s and the rpm of 72 was applied. The geometry of the NREL Phase VI wind turbine is shown in the following figure.

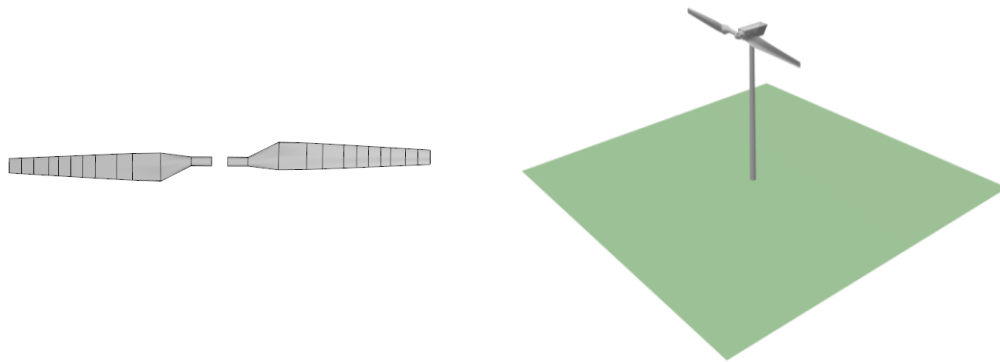


Figure 6.8. NREL Phase VI wind turbine

The blades of the wind turbine are also hollow and with spars inside. The internal structure of the blades is represented in Figure 6.9.

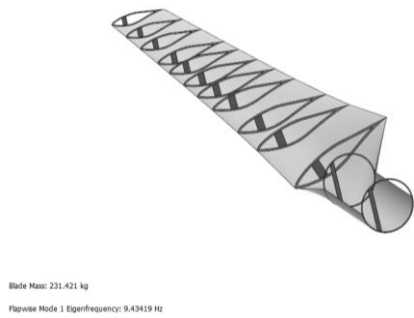


Figure 6.9. Internal structure of the blades.

The next figure represents the structural loading of the blades. It can be seen that the trailing edge and root regions experience the highest stress. The results show an edwise tip displacement of -0.620849 mm and a flapwise tip displacement of 1.61226 mm. The highest stress magnitude is 0.86 MPA.

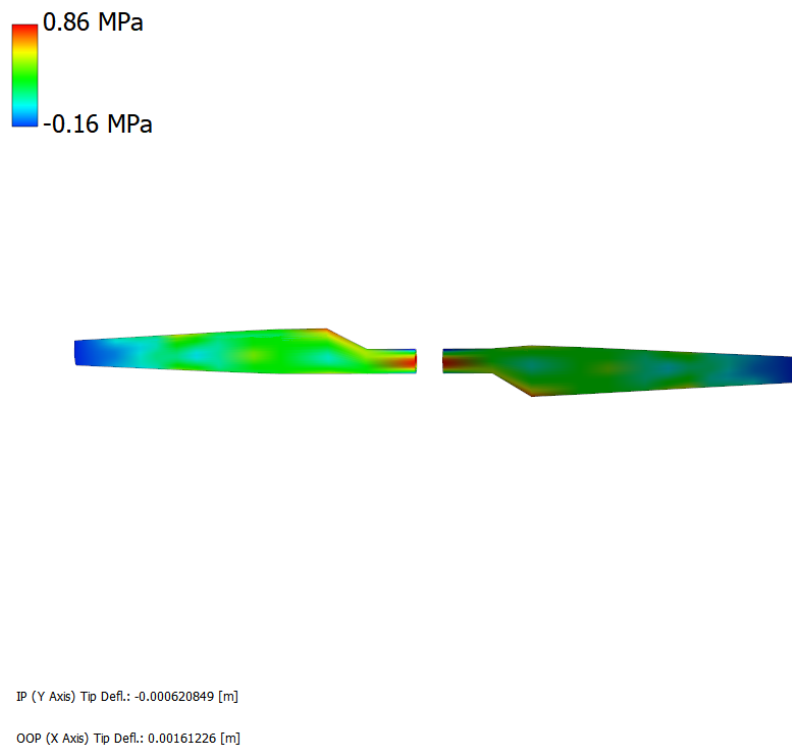


Figure 6.10. Structural loadings on the blade.

In Figure 6.11, the vorticity contours for NREL Phase VI wind turbine are illustrated. The vorticity contours are slightly less than the case of NREL 5 MW. The reason for that is simply the

number of blades of the wind turbines. The structure is turbulent at the start, but it becomes stable after some period of time.

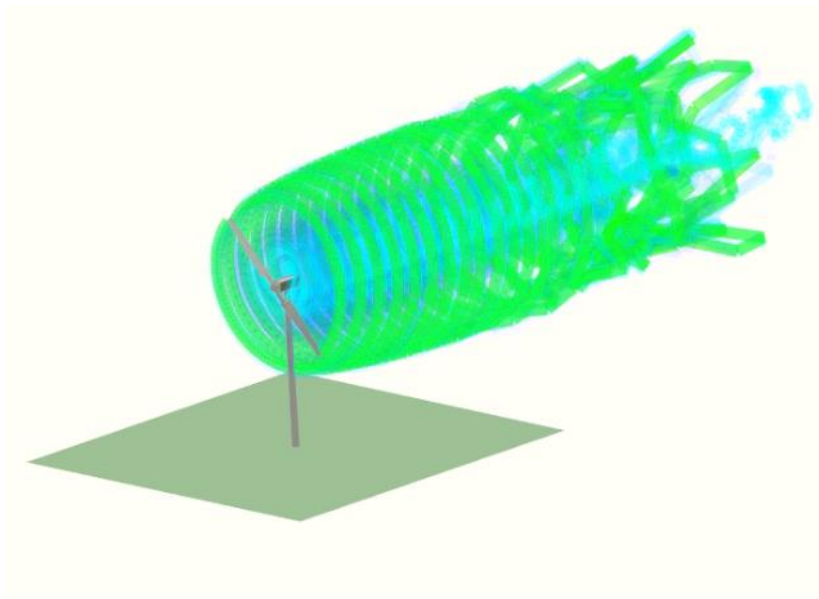


Figure 6.11. Wake structure.

The graphs for power coefficients from the simulation and experiment are illustrated in the next figure. The maximum C_p was achieved at tsr of approximately 6.5, hitting almost 0.41. In the experiment of Boudis et al. [45] the maximum C_p for this wind turbine was equal to 0.425 at a tsr of 7, which is illustrated in Figure 6.12. The obtained results and experimental results are very close to each other.

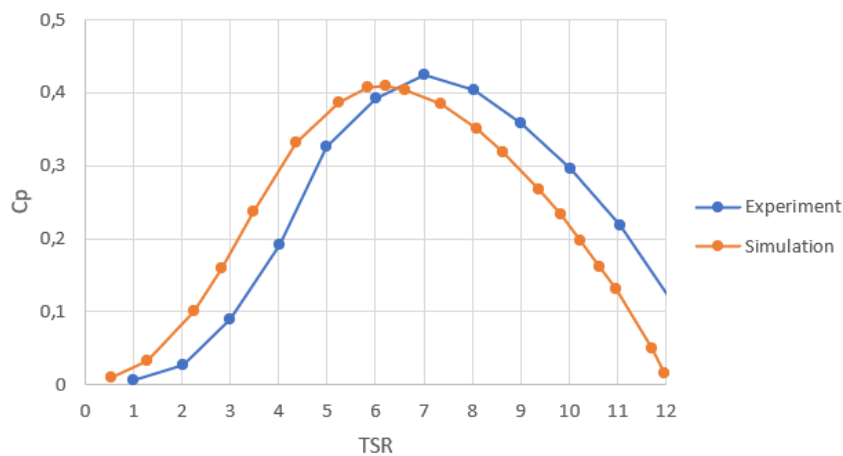


Figure 6.12. C_p vs tsr graph.

In Figure 6.13, the results for the twist angle from the simulation and experiment are plotted together for comparison. The trends of the results are similar, despite the parts having the highest twist angle. In the experiment of [46], the highest twist angle of 20 degrees was reached at $r/R=0.22$, while the simulation gave 19 degrees for angle of twist at $r/R=0.29$.

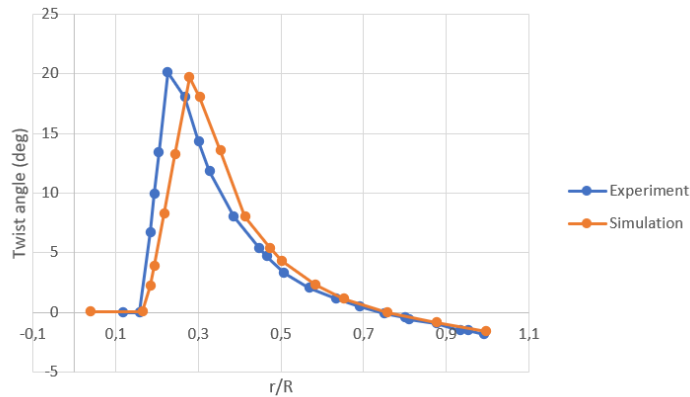


Figure 6.13. Twist angle vs r/R graph.

Chapter 7 – CFD Simulation

7.1. Mesh

For future works in this area a CFD simulation was set up and executed in OpenFoam on HPC. This model has a finer mesh, which contributes to get more accurate and correct results. The meshes of the fluid domain and propeller are shown in figures below. The windspeed of 7 m/s and 72 rpm was applied. The turbulence model is k-omegaSST URANS. This case further can be used for 1-way and 2-way FSI as well as the implementation of the VLES.

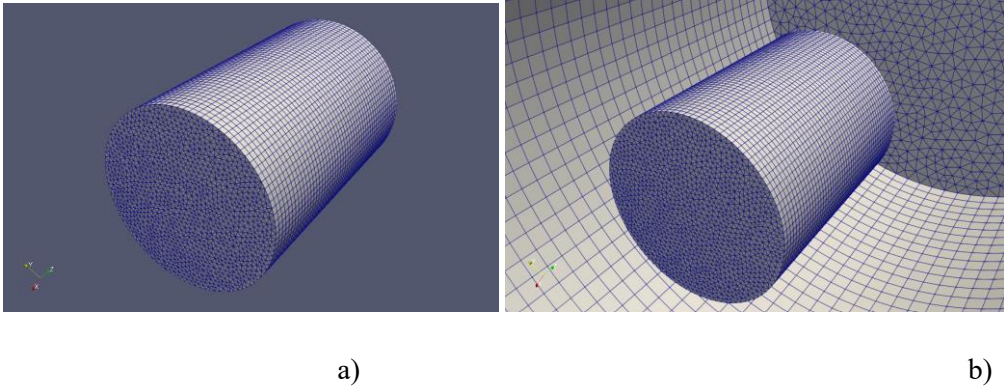


Figure 7.1. a) Mesh for fluid domain; b) Rotor rotation zone inside the domain

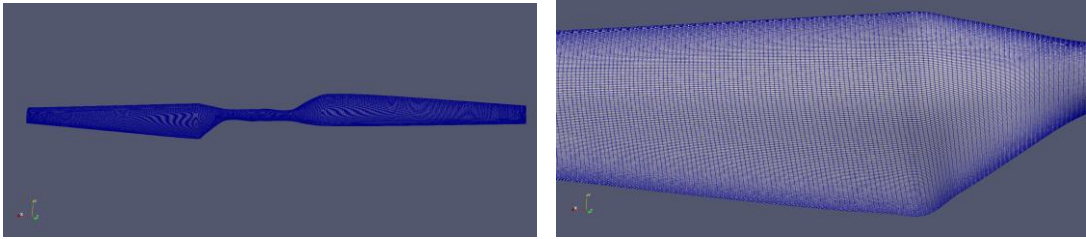


Figure 7.2. Mesh for blades.

7.2. Simulation results

The figure below shows the pressure on the blades. It can be seen that after rotation the regions of high-pressure change.

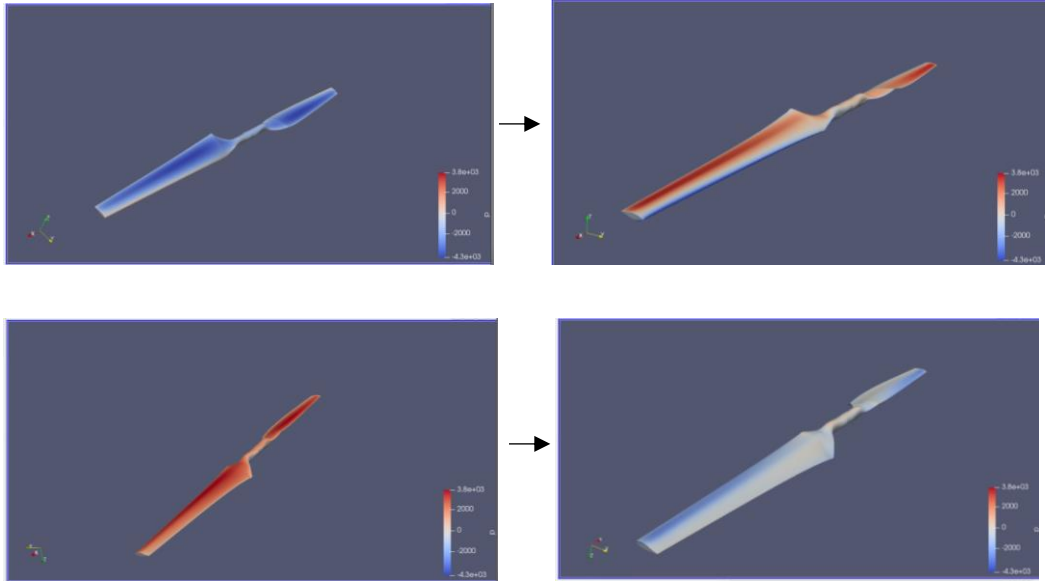


Figure 7.3. The change of pressure distribution on the blades.

Figure 7.4 represents the vorticity contours around the blades. The simulation can also be tried with higher windspeed values in order to analyze the turbulence eddies. This structure later can be compared with the results for the implementation of the VLES model in the future works. Then, use this model for FSI simulations.

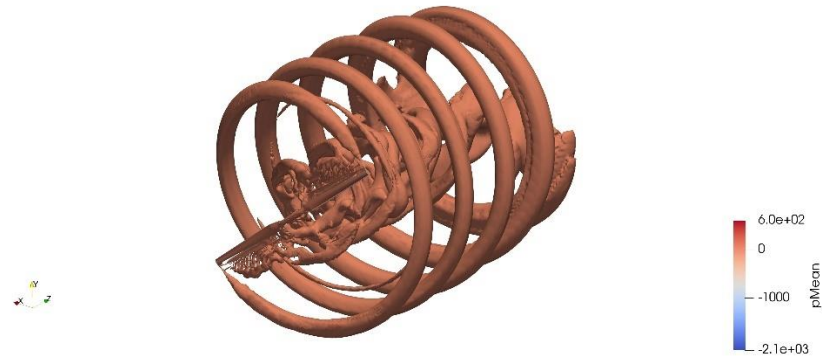


Figure 7.4. Vorticity contours around the blades.

The figure below illustrates the pressure coefficients obtained from the CFD simulation of the URANS model, which are then contrasted with experimental data at various spans from the work of Hand et al. [47]. It can be seen that the pressure coefficients from the simulation and the ones from the experiment are in good agreement. However, there is a noticeable difference in pressure for a leading edge.

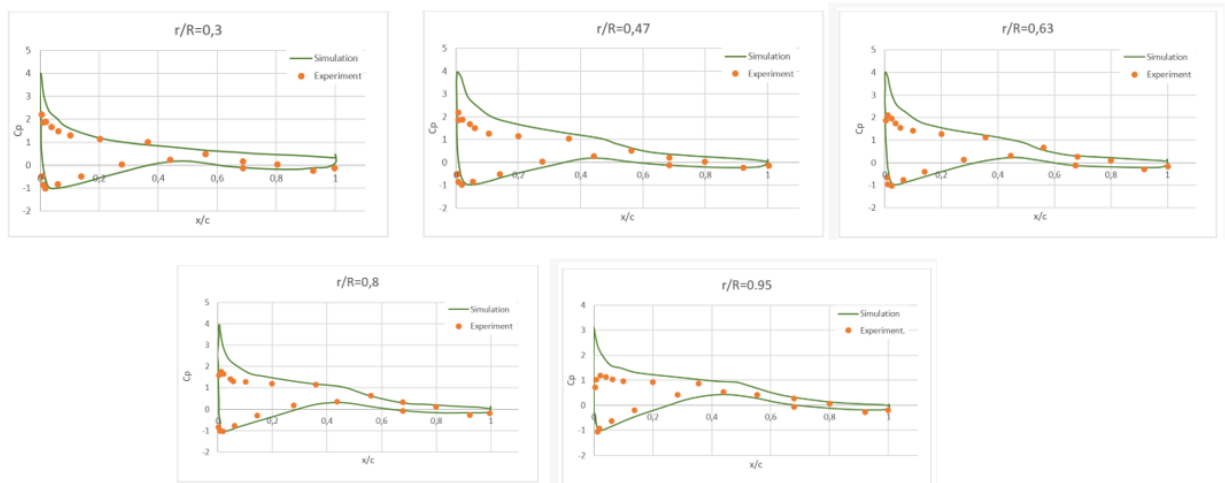


Figure 7.5: Pressure coefficients vs x/c

Chapter 8 – Conclusions & Future works

Initially, the goal of this work was to create a two-way dynamic FSI by coupling the DAfoam and CalculiX solvers in preCICE software. The coupling was being made between OpenFoam and CalculiX as an initial case. Apart from that, the FSI test cases such as perpendicular flap and elastic 3D tube were done. These works helped to obtain a better understanding of the software and FSI analysis. Nevertheless, this methodology had its limitations and was not suitable for two-way dynamic FSI analysis. Mainly, the issue was coming from the CalculiX solver. Despite that, this method for FSI analysis is possible for 2-way dynamic FSI simulations without rotational bodies. On the other hand, the second method showed better performance. QBlade software was used for carrying out a low-fidelity FSI by coupling the LLFVW aerodynamic solver with the Chrono structural solver. With this software setup the simulations were conducted for NREL 5 MW and NREL Phase VI wind turbines. The obtained results were examined and showed good agreement with the experimental results. The QBlade simulation findings provide a comprehensive knowledge of the flow phenomena present in the blade, such as flow separation, vorticity formation, and wake structure. Furthermore, a thorough study of the simulation results is possible because of QBlade's visualization features including vorticity contours and other post-processing tools. This makes it easier to examine the aerodynamic behavior in detail and to spot areas with significant vorticity, flow separation, or other flow characteristics that might affect the wind turbine's functionality and effectiveness. Moreover, a new domain was used for CFD simulation with URANS model. The obtained results were compared with the experimental results. The results are almost identical to the results from the literature. This model can be used for future works, such as implementation of the VLES as well as 1-way and 2-way FSI simulation for more precise and accurate results.

References

- [1] Ren, Z., Verma, A. S., Li, Y., Teuwen, J. J., & Jiang, Z. (2021). Offshore wind turbine operations and maintenance: A state-of-the-art review. *Renewable and Sustainable Energy Reviews, 144*, 110886.
- [2] Lee, Y. J., Jhan, Y. T., & Chung, C. H. (2012). Fluid–structure interaction of FRP wind turbine blades under aerodynamic effect. *Composites Part B: Engineering, 43*(5), 2180-2191.
- [3] Lee, K., Huque, Z., Kommalapati, R., & Han, S. E. (2017). Fluid-structure interaction analysis of NREL phase VI wind turbine: Aerodynamic force evaluation and structural analysis using FSI analysis. *Renewable Energy, 113*, 512-531.
- [4] Hewitt, S., Revell, A., & Margetts, L. (2016). A Novel Black-Box Massively Parallel Partitioned Approach to Fluid-Structure Interaction Problems. [2] Xu, Z., Wei, J., Zhang, S., Liu, Z., Chen, X., Yan, Q., & Guo, J. (2021). A state-of-the-art review of the vibration and noise of wind turbine drivetrains. *Sustainable Energy Technologies and Assessments, 48*, 101629.
- [5] Xu, Z., Wei, J., Zhang, S., Liu, Z., Chen, X., Yan, Q., & Guo, J. (2021). A state-of-the-art review of the vibration and noise of wind turbine drivetrains. *Sustainable Energy Technologies and Assessments, 48*, 101629.
- [6] Kim, D. H., Lim, O., Choi, E. H., & Noh, Y. (2017). Optimization of 5-MW wind turbine blade using fluid structure interaction analysis. *Journal of Mechanical Science and Technology, 31*(2), 725-732.
- [7] Zhangaskanov, D., Batay, S., Kamalov, B., Zhao, Y., Su, X., & Ng, E. Y. K. (2022). High-Fidelity 2-Way FSI Simulation of a Wind Turbine Using Fully Structured Multiblock Meshes in OpenFoam for Accurate Aero-Elastic Analysis. *Fluids, 7*(5), 169.
- [8] Guma, G., Bucher, P., Letzgus, P., Lutz, T., & Wüchner, R. (2022). High-fidelity aeroelastic analyses of wind turbines in complex terrain: FSI and aerodynamic modelling. *Wind Energy Science Discussions, 1*-25.
- [9] Kamalov, B., Batay, S., Zhangaskhanov, D., Zhao, Y., Ng, E. Y. K. (2022). Arbitrary Hybrid Turbulence Modeling (AHTM) Approach for High Fidelity NREL Phase VI Wind Turbine CFD Simulation, *Fluids 2022, 7*(7), 236; <https://doi.org/10.3390/fluids7070236>
- [10] Grinderslev, C., González Horcas, S., & Sørensen, N. N. (2021). Fluid structure interaction simulation of a wind turbine rotor in complex flows, validated through field experiments. *Wind Energy, 24*(12), 1426-1442.
- [11] Bazilevs, Y., Takizawa, K., Tezduyar, T. E., Hsu, M. C., Kostov, N., & McIntyre, S. (2014). Aerodynamic and FSI analysis of wind turbines with the ALE-VMS and ST-VMS methods. *Archives of Computational Methods in Engineering, 21*(4), 359-398.
- [12] Wang, L., Quant, R., & Kolios, A. (2016). Fluid structure interaction modelling of horizontal-axis wind turbine blades based on CFD and FEA. *Journal of Wind Engineering and Industrial Aerodynamics, 158*, 11-25.
- [13] Santo, G., Peeters, M., Van Paeppegem, W., & Degroote, J. (2020). Fluid–structure interaction simulations of a wind gust impacting on the blades of a large horizontal axis wind turbine. *Energies, 13*(3), 509.
- [14] Shi, F., Wang, Z., Zhang, J., Gong, Z., & Guo, L. (2019). Influences of wind and rotating speed on the fluid-structure interaction vibration for the offshore wind turbine blade. *Journal of Vibroengineering, 21*(2), 483-497.

- [15] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics*, vol. 113. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-16874-6.
- [16] Jasni, N. A. H., & Lajis, M. A. (2012). A comprehensive study on surface roughness in machining of AISI D2 hardened steel. *Advanced Materials Research*, 576, 60–63. <https://doi.org/10.4028/www.scientific.net/AMR.576.60>
- [17] Tachos, N. S., Filios, A. E., & Margaritis, D. P. (2010). *A comparative numerical study of four turbulence models for the prediction of horizontal axis wind turbine flow. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 224(9), 1973–1979. doi:10.1243/09544062jmes1901
- [18] Nichols, R. (2003). Applications of RANS/LES turbulence models. In *41st Aerospace Sciences Meeting and Exhibit* (p. 83).
- [19] Taghinia, J. H., Rahman, M. M., & Siikonen, T. (2018). A sub-grid scale model with natural near-wall damping. *Aerospace Science and Technology*, 74, 1-16.
- [20] Maulenkul, S., Yerzhanov, K., Kabidollayev, A., Kamalov, B., Batay, S., Zhao, Y., & Wei, D. (2021). An Arbitrary Hybrid Turbulence Modeling Approach for Efficient and Accurate Automotive Aerodynamic Analysis and Design Optimization. *Fluids*, 6(11), 407.
- [21] Stephens, D., Sideroff, C., & Jemcov, A. (2016). A two equation VLES turbulence model with near-wall delayed behaviour. In *7th Asia-Pacific International Symposium on Aerospace Technology. DOI (Vol. 10)*.
- [22] Tiwari, P., Xia, Z., & Han, X. (2020). Comparison of VLES and LES turbulence modeling for swirling turbulent flow. *Journal of Applied Fluid Mechanics*, 13(4), 1107-1116.
- [23] Liu, D., Tang, W., Wang, J., Xue, H., & Wang, K. (2016). Comparison of laminar model, RANS, LES and VLES for simulation of liquid sloshing. *Applied Ocean Research*, 59, 638-649.
- [24] Mangani, L., Buchmayr, M., Darwish, M., & Moukalled, F. (2017). A fully coupled OpenFOAM® solver for transient incompressible turbulent flows in ALE formulation. *Numerical Heat Transfer, Part B: Fundamentals*, 71(4), 313-326.
- [24] Zhao, Y., & Su, X. (2019). Arbitrary Lagrangian–Eulerian (ALE) Method and Fluid-Structure Interaction. *Computational Fluid-Structure Interaction*, 127–144. doi:10.1016/b978-0-12-814770-2.00009-x
- [25] Issa, R. (1982). Solution of the implicit discretized fluid flow equations by operator splitting. *Mechanical Engineering Rep*, in.
- [26] OpenFOAM. (2022). Retrieved 29 April 2022, from <https://www.openfoam.com/>
- [27] Lorentzon, J. (2009). Fluid-Structure Interaction (FSI) case study of a cantilever using OpenFOAM and DEAL. II with application to VIV.
- [28] Dafoam.github.io. 2022. *DAFoam: Discrete Adjoint with OpenFOAM for High-fidelity Multidisciplinary Design Optimization | DAFoam*. [online] Available at: <<https://dafoam.github.io/>> [Accessed 30 August 2022].
- [29] preCICE - The Coupling Library | preCICE - The Coupling Library. (2022). Retrieved 1 September 2022, from <https://precice.org/>
- [30] Chourdakis, G., Davis, K., Rodenberg, B., Schulte, M., Simonis, F., Uekermann, B., ... & Koseomur, O. Z. (2021). preCICE v2: A sustainable and user-friendly coupling library. *arXiv preprint arXiv:2109.14470*.

- [32] Abate, G., Mavris, D. N., & Sankar, L. N. (2019). Performance effects of leading edge tubercles on the NREL Phase VI wind turbine blade. *Journal of Energy Resources Technology*, 141(5).
- [33] Sodja, J., De Breuker, R., Nozak, D., Drazumeric, R., & Marzocca, P. (2018). Assessment of low-fidelity fluid–structure interaction model for flexible propeller blades. *Aerospace Science and Technology*, 78, 71-88.
- [34] Fernandez-Escudero, C., Gagnon, M., Laurendeau, E., Prothin, S., Michon, G., & Ross, A. (2019). Comparison of low, medium and high fidelity numerical methods for unsteady aerodynamics and nonlinear aeroelasticity. *Journal of Fluids and Structures*, 91, 102744.
- [35] Nagawkar, J., & Leifsson, L. (2022). Multifidelity aerodynamic flow field prediction using random forest-based machine learning. *Aerospace Science and Technology*, 123, 107449.
- [36] *QBlade - Next Generation Wind Turbine Simulation (2023) QBlade.org*. Available at: <https://qblade.org/>
- [37] Marten, D. (2020). QBlade: a modern tool for the aeroelastic simulation of wind turbines.
- [38] Hashem, I., Ke, W., & Zhu, B. (2021). On the Wake Characteristics of the NREL Phase VI Wind Turbine under Turbulent Inflow Conditions. In *14 th European Conference on Turbomachinery Fluid dynamics & Thermodynamics*. EUROPEAN TURBOMACHINERY SOCIETY.
- [39] Perez-Becker, S., Papi, F., Saverin, J., Marten, D., Bianchini, A., & Paschereit, C. O. (2020). Is the Blade Element Momentum theory overestimating wind turbine loads?—An aeroelastic comparison between OpenFAST's AeroDyn and QBlade's Lifting-Line Free Vortex Wake method. *Wind Energy Science*, 5(2), 721-743.
- [40] Tasora, A., Serban, R., Mazhar, H., Pazouki, A., Melanz, D., Fleischmann, J., ... & Negrut, D. (2016). Chrono: An open source multi-physics dynamics engine. In *High Performance Computing in Science and Engineering: Second International Conference, HPCSE 2015, Soláň, Czech Republic, May 25-28, 2015, Revised Selected Papers 2* (pp. 19-49). Springer International Publishing.
- [41] Marten, D., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2016). Implementation, optimization, and validation of a nonlinear lifting line-free vortex wake module within the wind turbine simulation code qblade. *Journal of Engineering for Gas Turbines and Power*, 138(7), 072601.
- [42] Jonkman, J., Butterfield, S., Musial, W., & Scott, G. (2009). *Definition of a 5-MW reference wind turbine for offshore system development* (No. NREL/TP-500-38060). National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [43] Marten, D., Paschereit, C. O., Huang, X., Meinke, M., Schroeder, W., Mueller, J., & Oberleithner, K. (2020). Predicting wind turbine wake breakdown using a free vortex wake code. *AIAA Journal*, 58(11), 4672-4685.
- [44] Vesel Jr, R. W. (2012). *Aero-structural optimization of a 5 MW wind turbine rotor* (Doctoral dissertation, The Ohio State University).
- [45] Boudis, A., Hamane, D., Guerri, O., & Bayeul-Lainé, A. C. (2023). Airfoil Shape Optimization of a Horizontal Axis Wind Turbine Blade using a Discrete Adjoint Solver. *Journal of Applied Fluid Mechanics*, 16(4), 724-738.
- [46] Mo, J. O., & Lee, Y. H. (2012). CFD Investigation on the aerodynamic characteristics of a small-sized wind turbine of NREL PHASE VI operating with a stall-regulated method. *Journal of mechanical science and technology*, 26, 81-92.

- [47] Hand, M. M., Simms, D. A., Fingersh, L. J., Jager, D. W., Cotrell, J. R., Schreck, S., & Larwood, S. M. (2001). *Unsteady aerodynamics experiment phase VI: wind tunnel test configurations and available data campaigns* (No. NREL/TP-500-29955). National Renewable Energy Lab., Golden, CO.(US).