UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

MASTER FINAL THESIS

CFD Simulation of a Floating Wind Turbine in OpenFOAM:
an FSI approach based on the actuator line and relaxation zone methods.

Document:

Report

Author:

Pere Frontera Pericàs

Director:

Daniel Garcia-Almiñana

Degree:

Master's degree in Aerospace Engineering

Examination Session:

Autumn 2022

Abstract

Floating offshore wind turbines (FOWTs) have the potential to harness wind resources in deepwater, which is so far prohibitive for conventional approaches. This, however, comes at a cost: the platform's extra degrees of freedom (DoFs) introduce complex aerodynamic and hydrodynamic behaviours. Therefore, FOWTs must be accurately modeled to reduce load uncertainties that ultimately prejudice their economic viability.

This project implements a framework for the coupled, high-fidelity simulation of FOWTs in OpenFOAM. The tool is built upon two existing libraries: *turbinesFoam* [1] —for rotor modeling based on the actuator line method— and *waves2Foam* [2] —for wave-field generation and absorption based on the relaxation zone method. The multi-phase simulation uses the *interFoam* solver in combination with a morphing mesh technique and rigid-body model to represent the platform. The mooring restraints are computed with a quasi-steady, catenary model from *waves2Foam*. The *turbinesFoam* library, targeted at bottom-fixed turbines, is modified so that it can accommodate arbitrary motions along the rigid-body DoFs. The platform-turbine FSI coupling follows a serial sub-iterating strategy based on the PIMPLE scheme.

The simulation framework is built in a sequential style. First, the propagation of second-order waves in an empty tank is studied, followed by the decay oscillation of floating buoys from the experiments by Ito and Palm et al. [3, 4]. Then, the modified version of *turbinesFoam* is tested for the conditions from the OC6 Phase III campaign —a series of wind-tunnel tests carried out at *Politecnico di Milano* that analyzed the performance of a scaled 10-MW turbine under prescribed motions in pitch and surge [5]. Lastly, the coupled simulation of a 2-DoF (surge and pitch) semi-submersible FOWT under combined wind-wave conditions is achieved.

The presented framework proved capable of modeling the aerodynamic performance of turbines under prescribed motion and produced plausible results for a semi-submersible FOWT under combined wind and wave conditions. Once carefully validated, this tool will have the potential to serve as a reliable technique for the advanced modeling of FOWTs.

Resum

Les turbines eòliques flotants tenen el potencial d'aprofitar el recurs eòlic en aigües profundes, fins ara inaccessible mitjançant turbines convencionals. Però hi ha un preu a pagar: els graus de llibertat addicionals de la plataforma introdueixen comportaments aerodinàmics i hidrodinàmics complexos. Per això, els aerogeneradors flotants s'han de modelar acuradament per tal de minimitzar aquelles incerteses que puguin perjudicar la seva viabilitat econòmica.

En aquest projecte s'ha implementat, mitjançant OpenFOAM, un marc numèric per a la simulació d'alta fidelitat de turbines eòliques flotants. Aquesta eina es basa en dues llibreries preexistents: *turbinesFoam* [1] (per al modelatge de la turbina, basat en l'*Actuator Line Method*) i *waves2Foam* [2] (per a la generació i absorció d'onades, basat en el *Relaxation Zone Method*). La simulació multifàsica utilitza el solver *interFoam*, combinat amb una malla dinàmica i un model de sòlid rígid per a representar la plataforma. Les forces de les línies d'amarratge s'obtenen a través d'un model quasi estàtic inclòs dins *waves2Foam*. La llibreria *turbinesFoam*, originalment pensada per a aerogeneradors fixos, s'ha modificat per poder tenir en compte els moviments arbitraris de les turbines flotants. L'acoblament fluid-estructura entre la plataforma i la turbina segueix una estratègia de subiteracions, basada en l'algoritme PIMPLE.

Aquest marc numèric s'ha construit seqüencialment. Primer s'ha estudiat la propagació d'onades de segon ordre en un tanc bi-dimensional, seguit de l'oscil·lació amortida d'objectes flotants d'acord amb els experiments d'Ito et al. i Palm et al. [3, 4]. Després s'ha posat a prova la versió modificada de *turbinesFoam* amb les condicions de la campanya OC6: una sèrie de tests duits a terme al túnel de vent de la *Politecnico di Milano* on es va analitzar un aerogenerador a escala forçat a moure's en diferents graus de llibertat. Finalment, s'ha dut a terme la simulació completa d'una turbina eòlica flotant sotmesa a l'efecte combinat d'onades i vent.

El marc desenvolupat ha demostrat ser capaç de modelar l'aerodinàmica d'aerogeneradors amb moviments prescrits, de la mateixa manera que ha produït resultats plausibles per a una turbina eòlica del tipus semisubmergible sotmesa a l'efecte combinat d'onades i vent. Una vegada s'hagi validat de forma curosa, aquesta eina servirà per a modelar turbines eòliques flotants amb alta fidelitat.

Acknowledgements

This thesis sets the final milestone on a long journey. Getting here would not have been possible without the great deal of support and attendance I've received. First, I would like to express my gratitude towards my supervisor, Daniel Garcia-Almiñana, for his trust and guidance. Thank you to my professors, Axelle Viré and Delphine de Tavernier, for providing me with a renewed perspective with regard to floating offshore wind energy. To my colleagues Carlos and Likhitha, thank you for your valuable help and advice on the technical side of the thesis.

To all my friends, thank you for joining me on this adventure. To Xènia, for watching my back and standing me up every time I have fallen, for showing me the value of effort and perseverance, thank you. A very special shout-out to my family for their unconditional support and, more importantly, for putting up with me all these years. And finally, a deep acknowledgement to all the people who have contributed in whatever way to get me where I am now.

Pere Frontera Pericàs
Terrassa, December 2022

Preface

Climate change is non-negotiable. The *Intergovernmental Panel on Climate Change* (IPCC) puts it clearly: the aftermath of climate change is inevitable, but its magnitude will depend on the efforts made to reduce greenhouse gas (GHG) emissions over the next thirty years [6]. International will for collective action was palpable at the recent UN Climate Change Conference, the COP26, where 197 parties committed to keeping the global temperature increase below 1.5 ºC [7]. Given that almost three-quarters of the total GHG emissions in 2020 came from the energy sector [8], it should come as no surprise that to achieve the 1.5 ºC target, a goal of net-zero emissions by 2050 is needed [6].

Over the last decades, wind energy has proven to be a reliable and cheap source of renewable energy, with 770 GW of installed capacity in 2020 [9]. But even though 2020 saw the highest increase in wind energy installed capacity by more than 93 GW [9], this figure must be doubled by 2025 and tripled by 2030 to achieve the 1.5 ºC target according to the IEA's net-zero scenario [7]. But due to environmental, social, and regulatory restrictions, constructing large wind farms onshore is becoming increasingly difficult [10]. To exploit wind resource even further, one should also consider the technical potential of 4 TW of offshore wind energy, which is higher than the world's electricity demand [11].

Larger available areas with limited impact, ease of transportation, higher wind speeds (a 40–50% increase in capacity factors [10]) and lower turbulence levels are just some reasons why offshore wind energy is a key point in the 2050 agenda, with a global share of installed wind capacity of 6% in 2021 and a 21% estimation for 2025 [7]. Only in Europe, the aim is to increase offshore wind capacity from the current 12 GW to at least 30 GW by 2030 and 300 GW by 2050 [9]. But at the same time, the $\times 3$ increase (from 2014 to 2020) in offshore wind farms' mean distance to the shore [12, 13] points out the increasing trend to seek deeper waters. This strategy is twofold: on the one hand, most of the offshore wind resource is located in deep waters (80% in Europe and Japan and 60% in the USA [14]), and on the other hand, most countries do not have access to shallow waters where to deploy fixed offshore wind farms (in 2021, only five countries made up 99% of the total installed offshore potential in Europe [13]). This appetite for exploiting the wind resource in deep waters is higher in the USA, China, Japan, and several European countries, including the UK, Norway, France, Portugal, and Spain [14].

The main barrier here is that wind turbine cost increases linearly with shore distance and water depth, making conventional fixed foundations not economically viable in water depths greater than 50 m [12] thus raising the need for alternatives. In such a context, the idea of floating wind turbines arose. First conceptualized in 1972 by Professor E. Heronemus to harvest energy in deep waters [15], the first full-scale floating offshore wind turbine (FOWT) was installed in Norway in 2010 [12]. Since then, more than 30 different configurations have been conceptualized, 10 of which have already been deployed as part of the prototyping phase [12]. In 2021, Europe accounted for 83% of the global floating offshore wind capacity, equivalent to 62 MW, not even 1% of the offshore installed capacity [13]. Expectations for 2030 are to increase the floating capacity in Europe up to 7 GW [13], accounting for almost 25% of the total offshore capacity. But first, FOWTs need to overcome serious challenges and prove economically viable before they can aspire to a position in the market.

This is not the first time engineers must deal with floating structures, and much experience has been gained by the gas and oil sectors. However, knowledge transfer is not that easy: since oil platforms are manned systems, there is a need for high safety factors but not for optimization since these are usually *"one-off"* designs and cost is not the critical factor [12]. On the other hand, FOWTs are lightweight structures strongly influenced by wind and waves, where the resulting levelized cost of energy is the main driver for economic success. Unlike fixed structures, FOWTs are less constrained by water depth and seabed conditions, so they can benefit more from design standardization and the cost reduction it implies. These circumstances point out the need for FOWTs to have an optimized design, thus reducing structural mass and making these systems suitable for mass production [12]. This, added to the increased yield in deep waters, ease of installation, and reduced visual and environmental impact,

makes FOWT a promising newcomer in the wind energy sector.

Whether FOWTs will achieve economic feasibility is still a matter of debate [14]. Cost estimations for the current full-scale installations are difficult because conservative designs were adopted in these prototypes to reduce risk. Moreover, most of the floating wind concepts are turbine-agnostic, meaning that the floater can accommodate turbines from the fixed offshore sector. In the same vein, design tools for FOWT were initially constructed using the same building blocks as conventional offshore turbines in combination with extra modules to account for platform motions and hydrodynamic loads, which ultimately lead to high uncertainty in load prediction when applied in floating conditions. All of this results in a design that is far from optimized and thus not economically viable.

The design, analysis, and optimization of FOWTs require a multi-disciplinary approach because of all the different disciplines involved. It has been estimated that adopting an integrated design approach could lead to a potential cost reduction of 10% [12]. This figure could be further increased by optimizing the floater design, where most of the cost increase comes from when compared to fixed offshore [14]. But the dynamics of FOWTs are difficult to predict: the extra degrees of freedom of the floater, a flexible lightweight structure, the increased response to hydrodynamic loads, the more complex aerodynamic flowfields, and the action of the mooring lines all call for advanced numerical tools.

Great effort has been made by researchers from all over the world to verify and validate numerical tools specifically for FOWTs, most of which is encompassed within IEA Task 30 and the four OC campaigns. Since wind turbine design requires considering a large number of load cases [16], tools that require low computational power were prioritized in these campaigns. However, the increased aerodynamic and hydrodynamic complexity of FOWTs led to 20% load underprediction and thus an incorrect fatigue assessment [17]. The limited amount of experimental data calls for the need for high-fidelity simulation tools, mainly Computational Fluid Dynamics (CFD) which have the potential to shed light on the underlying physical mechanisms behind the behaviour of FOWTs. The OC6 phases Ib and III are already tackling the hydrodynamic and aerodynamic high-fidelity modeling of FOWTs, respectively [9].

The setup of CFD simulations is however relatively complex and requires large amounts of computational power, making it thus a non-feasible tool for the design stage. Most of this cost comes from the need to resolve the boundary layer around solid objects like turbine blades and floaters. The actuator line method (ALM) was introduced in 2002 [18] to help in this regard, at least on the aerodynamic side. The effect of turbine blades on the flow is modelled rather than simulated, removing the need for very fine grids and hence alleviating the computational cost of the simulation while keeping meaningful information about the wake and tip vortices.

With the aim of constructing a computationally affordable tool for the high-fidelity simulation of FOWTs, the present work will combine the ALM with the relaxation zone method (RZM) for wave generation and absorption. This simulation framework will be constructed in OpenFOAM for its open-source character and spread use throughout the CFD community. The work is based on two existing libraries: *turbinesFoam* by Bachant (ALM modeling of bottom-fixed turbines [1]) and *waves2Foam* by Jacobsen (RZM implementation in OpenFOAM [2]). The platform is modelled as a rigid-body coupled to the ALM turbine. Once carefully validated, this tool will have the potential to serve as a reliable technique for the advanced modeling of FOWTs.

Contents

List of Figures

List of Tables

Acronyms

ADM Actuator Disk Model
AE Actuator Element
ALE Arbitrary Lagrangian-Eulerian
ALM Actuator Line Model

BC Boundary Condition
BEM Boundary Element Method
BEMT Blade Element-Momentum Theory
BL Boundary Layer

CFD Computational Fluid Dynamics
CFL Courant-Friedrichs-Lewy
CNWT CFD-Based Numerical Wave Tank
CPWH Cells Per Wave Height

DoF Degree of Freedom

FOWT Floating Offshore Wind Turbine
FSI Fluid-Structure Interaction
FVM Finite Volume Method
FVM Free Vortex Method

HAWT Horizontal Axis Wind Turbine

IEA International Energy Agency

LES Large Eddy Simulation

MWL Mean Water Level

nRMSE Normalized Root Mean Square Error
NS Navier-Stokes
NWM Numerical Wave Maker
NWT Numerical Wave Tank

PWT Physical Wave Tank

RANS Reynolds-Averaged Navier-Stokes
RAO Response Amplitude Operator
RZM Relaxation Zone Method

TLP Tension Leg Platform
TSR Tip Speed Ratio
TWS Turbulent Wake State

VAWT Vertical Axis Wind Turbine
VOF Volume Of Fluid
VRS Vortex Ring State

List of Symbols

Mathematical notation

a	Scalar
$\mathbf{a}$	Vector
$\underline{\mathbf{A}}$	Tensor
$\underline{\mathbf{I}}$	Identity matrix
$\underline{\mathbf{R}}$	Rotation matrix
$\underline{\mathbf{Q}_{ab}}$	Transformation matrix ($a \rightarrow b$)
$\square^T$	Transpose
$\square^*$	Deviatoric component
$\overline{\square}$	Average
$\dot{\square}$	First temporal derivative
$\square_{(i)}$	Variable in the i frame

General variables

m	Mass
t	Time
ω	Frequency
R, D	Radius, diameter
S	Surface
V	Volume
$\hat{\mathbf{n}}$	Outward normal unit vector
$\mathbf{F}$	Force
$\mathbf{M}$	Moment
$\underline{\mathbf{J}}$	Inertia Tensor
$\mathbf{d} = \{x, y, z\}^T$	Surge, sway, heave
$\boldsymbol{\phi} = \{\phi, \theta, \psi\}^T$	Roll, pitch, yaw
$\mathbf{v} = \dot{\mathbf{d}}$	Rigid-body linear velocity
$\boldsymbol{\omega} = \dot{\boldsymbol{\phi}}$	Rigid-body angular velocity

Waves

η	Free-surface elevation
H	Wave height
h	Water depth
T	Wave period
k	Wave number
λ	Wavelength
ε	Wave steepness
g	Gravity acceleration

CFD

p	Pressure
ν	Kinematic viscosity
ρ	Density
α	Indicator field
$\mathbf{u}$	Velocity
$\boldsymbol{\omega}$	Vorticity
$\underline{\boldsymbol{\tau}}$	Stress tensor
$\mathbf{q}$	Source term
I	Turbulence intensity
k	Turbulent kinetic energy
ε	Turbulence dissipation rate
ω	Specific turbulence dissipation

Aerodynamics

$\boldsymbol{\Omega}$	Rotor angular velocity
$\mathbf{U}_\infty$	Freestream velocity
$\mathbf{W}$	Relative velocity
k	Reduced frequency
φ	Inflow angle
α	Angle of attack
a	Induction factor
c	Chord
N_B	Number of blades
c_l	Lift coefficient
c_d	Drag coefficient
C_T	Thrust coefficient
C_P	Power coefficient
η	Smearing function
ϵ	Projection width

Discretization

Δt	Timestep
Δx	Element size
τ	Cell aspect ratio
δ_{exp}	Expansion ratio

1

Introduction

On the road to net-zero emissions by 2050, many countries are demonstrating a fierce appetite for the technical potential of 4 TW of offshore wind energy, higher than the world's electricity demand [11]. However, many do not have access to this resource because it is found in deep waters and conventional turbines are no longer feasible for depths below 50 m [12].

Floating offshore wind turbines (FOWTs) have the potential to access wind resources deepwater, which is so far prohibitive for conventional approaches. This, however, comes at a cost: the platform's extra degrees of freedom introduce complex aerodynamic and hydrodynamic behaviours, which need to be accurately modelled to reduce load uncertainties that ultimately prejudice the economic viability of FOWTs. The strong coupling between the floater's motion and the aero/hydro flow-fields calls for advanced fluid-structure interaction (FSI) methods.

A trade-off between accuracy and cost ultimately drives the simulation approach. Blade-resolved computational fluid dynamics (CFD) represents the highest-fidelity tool to simulate FOWTs. However, fine grids near the blades and dynamic mesh strategies are required, heavily increasing computational costs. For this reason, blade-resolved CFD is reserved for particular load situations. On the other hand, actuator models can help in this regard by modelling the vorticity-driven flow around the blades rather than simulating it, alleviating the computational overhead.

In the present project, a simulation framework will be developed that can describe the coupled FOWT system (moorings + platform + turbine) with a high level of accuracy while maintaining computational costs reasonably low. For this purpose, the rotor's blades will be modelled using an actuator line model (ALM), while the wave field will be generated with the relaxation zone method. This framework will be constructed in OpenFOAM® given its open-source character and prevalent use in the CFD community.

1.1. Fundamentals of FOWTs

1.1.1. Extra degrees of freedom

In conventional, bottom-fixed wind turbines, the environmental loads (from turbulent wind and gusts to waves, currents and ice impact) are transferred through the foundations onto the soil. These turbines are said to be fixed because their rigid-body degrees of freedom (DOFs) are all constrained, although flexible deformations exist.

On the other hand, floating turbines cannot transmit the loads onto the soil with the same efficiency (they do so through the anchors of the mooring lines). Thus, they are susceptible to motion in all six rigid-body DoFs: three translations (surge, sway and heave) and three rotations (roll, pitch and yaw) as depicted in Figure 1.1. Depending on the type of FTW, they will be more sensitive to certain DoFs. A more detailed explanation of the different frames of reference used to describe the motion of FTWs is presented in Appendix 8.3.

These extra degrees of freedom come with an increase in modelling complexity. The underlying physics is now inherently unsteady, making aerodynamic (dynamic inflow, skewed wake or wake-blade interaction) and hydrodynamic (buoyancy and body-wave loads) estimations much harder. Moreover, the hydro- and aerodynamic domains become highly coupled: while the motion of the turbine is driven by the loads acting on it, these are closely related to the turbine position, orientation and velocity (see section 2.3 on fluid-structure interaction). The considered modelling approach now must take into account this two-sided coupling and solve for the six extra DoFs.

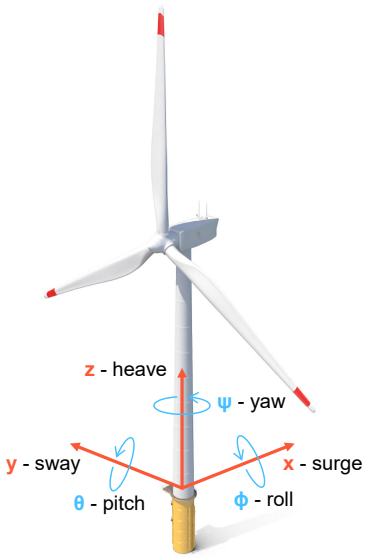


Figure 1.1: The six extra rigid body DoFs characteristic of FOWTs.

1.1.2. Classification

The desire to harvest wind resources in deeper waters has led to significant achievements in floating turbine technology. Although still far from the commercial phase, there is already a well-defined common ground between the different designs. Of all the drivers affecting the FOWT floater, there is a consensus to classify the designs based on the primary approach adopted to achieve static stability in pitch and roll. In other words, based on the source of the restoring moment counteracting the turbine thrust overturning moment [12, 19, 20, 10]:

Spar-buoy or ballast stabilised. Stability is gained by having the center of gravity lower than the buoyancy center, usually through a large cylinder with ballast in the lower part, and anchored to the seabed using catenary mooring lines. *Hywind*, *Sway* and *Advanced Spar* are examples of spar-buoy type FOWTs.

Tension leg platform (TLP) or mooring stabilised. Stability is provided by the tensioned mooring lines, vertically anchored to the seabed, while a shallow semi-submersible structure holds the turbine in place. *PelaStar*, *Blue H TLP* and *Eco TLP* are examples of TLP type FOWTs.

Barge or waterplane stabilised. Stability is gained by the increased waterplane-are and thus significant second moment of inertia. Barges are typically flat and without interspaces, thus having a minimal draft. Barges are seldom used because of the high exposure to wave loading.

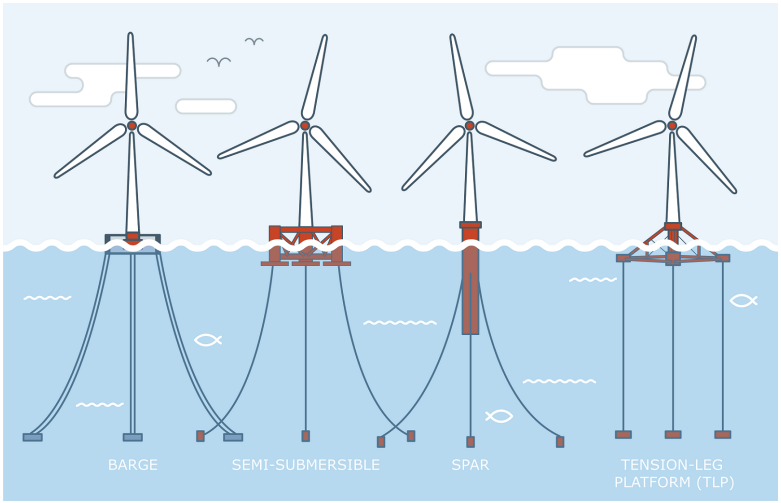


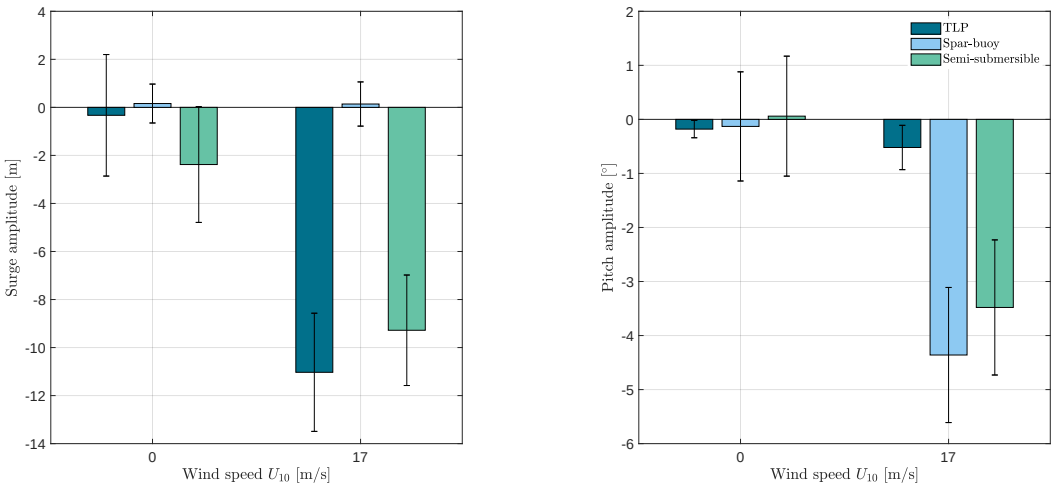
Figure 1.2: Main FOWT typologies. Reproduced from [21].

Semi-submersible. Hybrid between ballast and waterplane stabilised. A structure made of several cylinders floats semi-submerged on the ocean surface, anchored to the seabed by catenary mooring lines. The cylinders provide the extra buoyancy force, with usually one of them supporting the turbine.

In this case, extra ballast will be added to the other cylinders to compensate for the turbine weight and reduce inclination. Heave plates can be placed at the cylinders' base, increasing the added mass in heave and viscous damping, hence reducing wave excitation at low frequencies [10]. *WindFloat*, *WINFLO* and *VolturnUS* are examples of semi-submersible type FOWTs.

Figure 1.3 shows the results from the MARIN experimental campaign, where three different FOWT designs were tested in steady wind and irregular waves conditions (JONSWAP spectrum with significant wave height $H_s = 10.5$ m). Note that semi-submersible platforms are most sensitive to incident waves, as reflected by the higher pitch angle in the steady-wind case. Spars, however, are more susceptible to pitch and very rigid in surge. TLPs are the most rigid configuration in the rotational motion, although they experience surge drift.

Since up to a 16% reduction in the cost of FOWTs from prototype to commercial scale is expected to come from an optimised floater design, it is not surprising that more than 30 different floater concepts were under development in 2016 [14]. However, the optimum floater is just an idealisation: it will depend on the specific turbine, metocean conditions, water depth and seabed characteristics. Even though all of this was known, other factors such as control strategy, installation/maintenance costs, or footprint would come into play. A more pragmatic definition of optimum would be *"that which achieves the best functionality with the lowest cost"* [15], where functionality refers to the capability to keep the motion response within the acceptable envelope for the wind turbine [20].



(a) Surge response

(b) Pitch response

Figure 1.3: MARIN experimental campaign results for surge and pitch for three different floater configurations. The bars represent mean values while black lines the standard deviation of the measurements. Adapted from [22].

1.1.3. Mooring lines

In addition to buoyancy forces, mooring lines provide the required stiffness to maintain the floater position and orientation, thus acting as the main station-keeping system. The mooring lines are attached to the platform through the fairleads, which guide their orientation with respect to the platform. The vertical position of this fairlead will also modify the restoring moment from the mooring loads. Two types of mooring lines exist:

Catenary. Slack lines are freely hanging and anchored to the seabed at a significant distance from the platform. The lines can be partially resting on the seabed, in which case the anchors have to be designed for horizontal loads only, which reduces the dependence on soil properties. Catenary lines restrict the surge, sway and yaw motions [23] and govern the natural frequency of the former two [10].

Tensioned. Taut and lightweight lines anchored to the seabed in a vertical or slightly inclined orientation. This tension provides restoring forces in all six DoFs. On the other hand, the anchors need to support almost vertical loads, and thus soil characteristics become critical. Moreover, these lines can go slack and quickly taunt again due to waves, resulting in high shock loads [23].

1.2. Objective

High-fidelity simulation of FOWTs by means of the actuator line model is a promising approach capable of reducing the very high computational cost associated with CFD simulations. This fact has already been proven, and simulation tools with these characteristics can be found in literature [24, 25, 26]. They represent, however, a small portion compared with resolved CFD simulations. Moreover, an in-depth explanation of how to construct the model is often omitted, leaving the reader with sparse information in the case he or she wishes to replicate the model. They also miss specific information regarding the encountered difficulties, numerical cost, accuracy, or optimization alternatives, leaving no room for a *good-practices* manual.

To fill in this gap, the present project aims to develop a comprehensive description of how a CFD simulation framework for FOWTs based on the ALM can be implemented. To maximize the potential public that can benefit from such framework, it will be developed in the open-source tool OpenFOAM using the publicly available libraries *waves2Foam* and *turbinesFoam* which benefit from a well-established community that has validated and documented them. The research objective is posed as:

> **" To develop a comprehensive and accessible framework for the high-fidelity modelling of a full FOWT based on the actuator line and wave relaxation methods by performing and thoroughly documenting the CFD-FSI simulation of a FOWT in OpenFOAM using the *waves2foam* and *turbinesFoam* libraries "**

1.3. Scope

The aforementioned objective can be further split into five different sub-goals, which at the same time allow for the definition of the different work packages:

1. To understand the main challenges of FOWT modeling tools by:
   - 1.1 Performing a literature review on the underlying physics of FOWT: aerodynamics, hydrodynamics and moorings
   - 1.2 Examining the different fidelity tools used to simulate FOWTs, their main assumptions and limitations
   - 1.3 Getting acquainted with the current state of the art for FOWT research, including both experimental and numerical campaigns
2. To validate the implementation of the *waves2Foam* wave generation module in OpenFOAM by:
   - 2.1 Setting up a simulation with an empty numerical wave tank
   - 2.2 Comparing the numerical surface elevation with the algebraic solution
3. To validate the implementation of a dynamic mesh, mooring model and rigid-body coupling in OpenFOAM by:
   - 3.1 Setting up a free-decay simulation of a free and moored simplified platform (cylinder)
   - 3.2 Comparing the above results with data available from the literature
4. To adapt *turbinesFOAM* to the context of FOWTs:
   - 4.1 Modify *turbinesFOAM* to accommodate for arbitrary 6-DoF prescribed motions
   - 4.2 Test the modified library by considering the case from the OC6 Phase III campaign
5. Combine the different tools into a single framework for the coupled simulation of FOWTs:
   - 5.1 Couple *turbinesFOAM* with the rigid-body solver
   - 5.2 Test the final framework by simulating a semi-submersible FOWT under combined wind-wave conditions

1.4. Justification

1.4.1. The need for high fidelity

Floating offshore wind turbines have the potential to access a wind resource so far prohibitive for conventional approaches due to the limitations that deepwater implies. Whether they will ever achieve economic viability is still uncertain because of the simulation tools' limitation in predicting the operational loads. So far, prototypes have had to be designed with high safety factors to protect from these uncertainties, which resulted in a heavier and most costly structure, far from being an economically feasible design.

Within the IEA task 30 and other international projects, scientists are pushing hard to reveal the underlying physical mechanism of floating turbines to understand their behaviour better. Simulation tools with varying levels of fidelity were compared and validated against experimental data, showing that mid-fidelity tools often underpredict the response and thus, high-fidelity models —CFD— are required. The latter are unsuitable for everyday simulations since the computational cost is overly high. Instead, they are meant to shed more light on the underlying physics of floating turbines and for validation and correction of lower fidelity tools.

In the RANS domain, blade-resolved simulations are still a great challenge because of the required refinement around the blades and the need to incorporate a moving mesh around the rotor, resulting in long simulation times. A clear trade-off between accuracy and cost ultimately drives the simulation approach. Certain simplifications are possible to reduce computational costs and keep CFD simulations of floating turbines more manageable. For instance, the rotor can be modelled through actuator lines instead of resolving the detailed flow around the blades, leading to meaningful computational savings. Because most RANS-based solvers still require corrections to model near-wall phenomena (such as stall or boundary layer detachment), in wind turbine simulations, both resolved-RANS and ALM offer a similar level of fidelity.

1.4.2. The advantages of ALM

Because of their complex design and bluff shape, semi-submersible platforms cannot benefit from remarkable simplifications and thus must be modelled using resolved-CFD. On the other hand, turbine blades are streamlined bodies from which a long tradition of simplifying theories exists. The Free Vortex Method (FVM) has been shown to produce overall good results in floating conditions with much fewer computational resources than CFD, but they lack turbulence. Turbulence drives many phenomena occurring in FOWTs, such as near wake structures breakdown, wake instabilities, wake meandering and wake shear layer. For fatigue assessment, for instance, a turbulent flow description is crucial to model the loads' fluctuations correctly. In wind farm design, turbulent mixing in the wake determines the velocity deficit a neighbouring turbine will experience.

All in all, keeping the nonlinear advection term in the flow equations is necessary for many applications. Moreover, only partitioned coupling is possible between the FVM-rotor and the CFD-platform, requiring sub-iterations to stabilise the time-marching scheme. In any case, the low cost of FVM would be eclipsed by the high demands of the hydrodynamics CFD solver.

ALM fills the gap between blade-resolved CFD and vortex methods for turbine modelling. They can represent the main wake structures without the intense refinement needed near the blades for traditional CFD and without the use of moving mesh techniques. ALM can be coupled to the hydrodynamic simulation in a monolithic approach, leading to a more robust framework that has the potential to be extended to account for other elements, such as the tower and nacelle. The simulation of FOWTs with a resolved CFD for the platform and ALM for the rotor has already proved successful in [26, 25, 24]. There, the mooring dynamics and the platform motions are coupled into the CFD simulation in a partitioned manner. This same approach will be followed in the present project.

1.4.3. Towards an open-source FSI framework

High-fidelity ALM simulations of floating turbines are not new, although they represent only a tiny fraction of all high-fidelity studies. A comprehensive high-fidelity simulation framework based on this approach would allow scientific research on the dynamics of FOWTs with a relatively fast case setup and shorter simulation time. However, no such thing is available now, and one has to construct the simulation case

from scratch with few guidelines, which is remarkably cumbersome for research purposes. Moreover, only a reduced fraction of the available CFD software allows for rotor modelling through ALM.

To overcome this obstacle, the present project aims to develop a comprehensive and open-source framework for the high-fidelity simulation of FOWTs based on the actuator line (for the rotor) and relaxation zone (for wave generation and absorption) methods. Even though some work has been done in the development of FOWT simulation tools in OpenFOAM [25], to the author's knowledge the present work would represent the first attempt of such model combining *turbinesFoam* and *waves2Foam* for the simulation of FOWTs.

This framework should be able to delve into the underlying physics of FOWTs, where the coupling effects between the different elements results in complex behaviour that cannot be explained by modelling these elements in isolation. Because of the high-fidelity approach, this tool would have the potential to be used in environmental conditions such as severe sea states, turbulent or changing winds and large platform motions where traditional tools' assumptions are invalidated. Because of the lower computational overhead associated with ALM and its potential to describe the turbulent wake, this method is very well suited for simulating floating offshore wind farms under prescribed motions or wave conditions.

Thanks to the knowledge gained in these simulations, new corrections could be developed for these mid-fidelity tools. Moreover, even in milder conditions, the results for high-fidelity simulations stand out as an alternative to experimental data for validation and calibration of lower-fidelity tools. Test facilities for combined waves and wind are complex and expensive, with difficulties in accurately assessing the measurement uncertainties and the compromise of the aerodynamic or hydrodynamic similarities with the full-scale model.

1.5. Methodology

1.5.1. Set-up

For the present project, OpenFOAM is the selected software where this simulation framework is to be constructed due to its open-source character and prevalent use in the CFD community. A comprehensive description of OpenFOAM can be found in section 2.4 and Appendix 8.3. Different variants and versions of OpenFOAM exist, being version v2012 by *ESI-OpenCFD* the preferred choice throughout this report. This specific version has been chosen for its relatively recent release and compatibility with the different libraries:

— *waves2Foam*. By Jacobsen [27], allows for wave generation and absorption based on the relaxation zone technique (see section 4.1.3).

— *turbinesFoam*. By Bachant [1], models bottom-fixed turbines by means of the actuator line model (see section 6.1).

— *sixDoFRigidBodyMotion*. A rigid body motion is applied to a specified surface (e.g. a floater) to which different loads can be applied (see section 5.1).

How these different actors connect together is represented in figure 1.4.

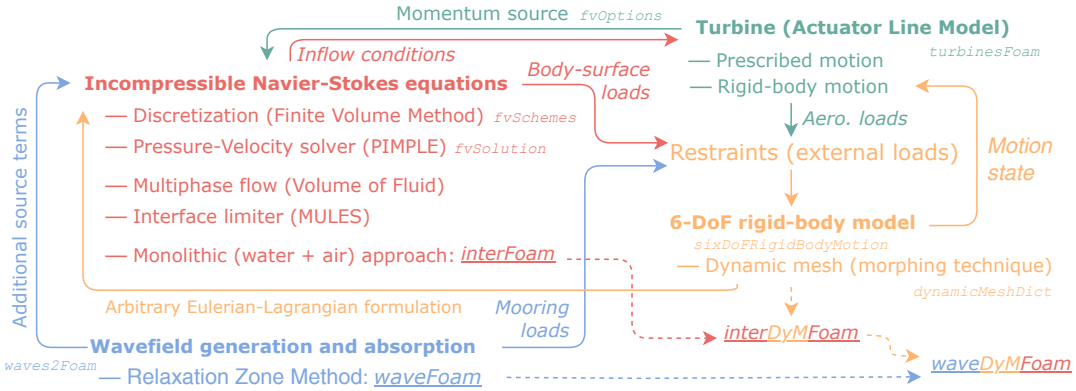


Figure 1.4: CFD simulation of a FOWT: FSI strategy in OpenFOAM v2012.

The mooring lines will be represented using a quasi-steady, catenary model included within *waves2Foam*. The main objective of the thesis is the coupling between the ALM and rigid-body libraries to fully describe the floating turbine behaviour. The turbine needs to follow the rigid body motion, while the latter should experience the aerodynamic load from the former. All in all, both *turbinesFoam* and *sixDoFRigidBodyMotion* will be extended so that they can be used in the context of FOWTs. To keep the source code transparent and available, the author's ambition is to eventually merge the resulting library for FOWTs with the original *turbinesFoam* by Bachant[1].

The simulation will be carried out following a monolithic approach, meaning that the same solver is used for the water (platform) and air (turbine) domains through a multi-phase modelling approach. Only the rigid body equations will be solved in a partitioned manner, that is, by a different solver. As the adage says, *"A picture is worth a thousand words"* and so the final framework is displayed in Figure 1.5 presenting the main actors involved in the simulation. As for the hardware side, most CFD simulations require computation powers unreachable by desktop computers. Instead, TU Delft's high-performance cluster (fpt-HPC12, mounted on *CentOS 7.9*) will be used, allowing for parallel jobs with up to 320 processors.

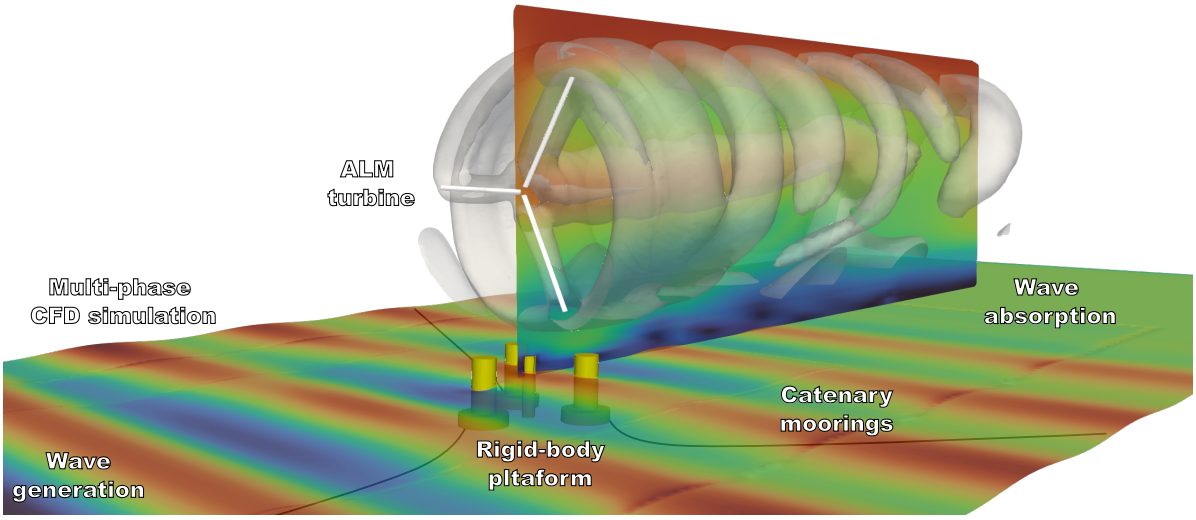


Figure 1.5: Final framework representation. The picture corresponds to the coupled FOWT simulation from Chapter 7.

Even in this very concrete scenario, many possibilities exist on how the simulation is performed: numerical domain, boundary conditions, CFD solver, moving mesh techniques, wave generation and absorption modules, free-surface treatment, turbulence modeling, force projection approach in ALM, mooring model, rigid body assumptions, coupling algorithm, etc. Those decisions are far from trivial and can ultimately result in various simulation fidelities and costs.

1.5.2. Verification philosophy

Any simulation tool needs to be verified —is the code doing what it is supposed to?— and validated —do the results match experimental observation?. Validation of both *turbinesFoam* and *waves2Foam* toolboxes has been carried out for a wide range of applications, but unfortunately not for FOWTs. An exhaustive validation or verification of the developed framework is a demanding endeavour which would likely deserve its own thesis and hence is left out of the present project.

The fact that no strict verification or validation procedure is adopted does not mean that the simulations are performed blindfolded, with no reference data to contrast with. All results will be compared with either analytical, numerical or experimental measurements. However, the broad scope of the project will not allow for grid/time-step independence or uncertainty analyses. All the different elements composing the simulation (grid, boundary conditions, numerical schemes and solvers, etc.) will be based on those from available literature on similar cases. The reader can expect coarser meshes and more

[1] To make the code transparent and rapidly available, both the modified libraries and simulation setups used throughout this research will also be released in the author's GitHub page.

diffusive methods that the ones required for high fidelity simulations, but which are more robust and computationally cheaper.

1.5.3. A sequential proceeding

OpenFOAM is a widely used tool for CFD in both industry and academia, but it is also famous for its considerably steep learning curve. No official manual is available that touches every specific detail and implementation, which can be frustrating for newcomers. Since this will be the author's first contact with this tool, straightly aiming for an FOWT simulation would likely result in a hopeless try. For this reason, the simulation framework will be constructed sequentially: first, an empty wave tank, followed by a free and moored floater, an ALM turbine, and finally, a full FOWT. Doing so means only one piece of OpenFOAM must be mastered at a time. Furthermore, the results can be more easily compared with available literature data.

Acknowledging that the reader might not be familiar with OpenFOAM, they will take advantage of this sequential order. Most of the available literature on ALM/FSI simulations of FOWTs omit the technical details and encountered difficulties, thus making it very hard to replicate the model. On the other hand, the present document will comprehensively cover most simulation aspects so that the steps can be quickly followed and replicated. This thesis should serve as a manual for anyone interested in the ALM/FSI simulation of FOWTs with OpenFOAM.

1.6. Report outline

Chapter 2 provides a comprehensive summary of the general concepts relevant to the thesis: from the fundamentals of hydrodynamics (stability of floating objects and wave propagation) to an overview on CFD (multi-phase flow and actuator models) and FSI (rigid body coupling and dynamic mesh), concluding with a short introduction to OpenFOAM (which is further extended in Appendix 8.3). Chapter 3 is presented as a literature review of the underlying physics of FOWTs, their main modelling approaches and current experimental campaigns. Chapter 4 focuses on the multi-phase simulation of the wave propagation phenomena in OpenFOAM, presenting the implementation of a 2D wave flume based on the relaxation zone method. Then, Chapter 5 extends this framework to floating rigid-bodies where dynamic mesh and FSI techniques are employed. Chapter 6 takes a breath from floating objects and focuses on the ALM, adapting the *turbinesFoam* library to prescribed motions and verifying the implementation with the results from the OC6 Phase III campaign. Finally, Chapter 7 puts everything together by coupling the ALM turbine and rigid-body floater. The resulting framework is tested through a FSI simulation of a semi-submersible FOWT. Chapter 8 outlines the key findings and provides with recommendations for future studies.

2

Theory

This chapter will provide the background theory and concepts on which the current thesis is built. It starts with a recap on hydrodynamics in section 2.1, which might turn useful for those coming from an aerodynamic background. This section will first cover the static equilibrium of floating bodies and then move to the ocean environment characterization where the wave propagation phenomena is reviewed. Sections 2.2 and 2.3 focus on the simulation side, explaining the basics of CFD (especially regarding multi-phase flows and actuator models) and FSI, respectively. Finally, section 2.4 presents the working principles behind OpenFOAM.

2.1. Fundamentals of hydrodynamics

The dynamics of floating platforms can be described through Newton's second law, where loads are computed from total pressure integration along the body's surface. This is, however, only possible with high-fidelity tools such as CFD. Even in undisturbed waters, analytically describing the motions of a floating body is a difficult task that marine engineers have faced with the help of linear hydrodynamic theory —see the *Marine Hydrodynamics* handbook by J. N. Newman [28] for a more in-depth explanation. This section, however, seeks for a simpler description of the underlying physics of floating bodies, first understanding its floating mechanism and then exploring the wave propagation phenomena and resultant loads.

2.1.1. Hydrostatics of FOWTs

Through Archimede's principle, one can compute the buoyancy force acting on a floating platform as the weight of the displaced body of fluid. Buoyancy forces can also be defined as the vertical component of the integrated hydrostatic pressure along the wet surface [29]. As pointed out above, describing the motion of an arbitrary floating body with 6 DoFs and large motions can become an overwhelming task. However, a simplified approach involving only one DoF can still provide meaningful information.

$$\mathrm{d}\mathbf{F_h} = -p_h\,\hat{\mathbf{n}}\,\mathrm{dS}, \quad \text{equilibrium: } \int_S \mathrm{d}\mathbf{F_h} = \sum \mathbf{F}_{\text{ext}}\,, \quad \int_S \mathbf{r}\,\mathrm{d}\mathbf{F_h} = \sum \mathbf{M}_{\text{ext}} \tag{2.1}$$

A common approach is to study the motion of the body close to static equilibrium, that is, when the hydrostatic (or buoyancy) force $\mathbf{F_h}$ and external (e.g. gravity, environmental or mooring) loads $\mathbf{F}_{\text{ext}}$ and moments $\mathbf{M}_{\text{ext}}$ are balanced. Note that surge, sway, or yaw motions (translations or rotations along the water surface) do not change the submerged volume and, thus, will not modify the equilibrium position. Similarly, heave motions do not introduce any moments about the center of gravity and hence will not perturb stable equilibrium. Only pitch and roll can change the stability condition by introducing a movement of the buoyancy and gravity centers.

The static initial-stability analysis presented here is based on previous works on floating structures [28, 29, 30], with the simplifying hypotheses that the body is always in equilibrium (constant wet volume and quasi-steady rotation with small angles) and the fluid is at rest. The goal is to get an analytic estimation

for the mean pitch angle θ in normal operation, even though the analysis can easily be extended to roll displacements ϕ.

Given the floating body in Figure 2.1, some geometrical definitions may be presented:

- An orthogonal axis system is defined, with X pointing in the downwind direction and Z upwards in the vertical direction. Then Y is defined such that the resulting base is semi-definite positive.
- The floater origin O_F is coincident with the center of flotation F, defined as the geometric centroid of the waterplane area.
- A bodyy-attached frame of reference, xyz, is defined with its origin at O. Both the inertial XYZ and body xyz frames coincide at zero pitch angle.
- The centers of gravity G and buoyancy B are defined as the point of action of the resulting gravity and buoyancy loads, respectively.
- The center of pressure of environmental loads C_e is the point of action of the horizontal component of the environmental forces F_e (mainly wind and currents) in an equilibrium state.
- The mooring lines acting point L is defined as the intersection of the line action for the horizontal component of mooring line force F_M with z.

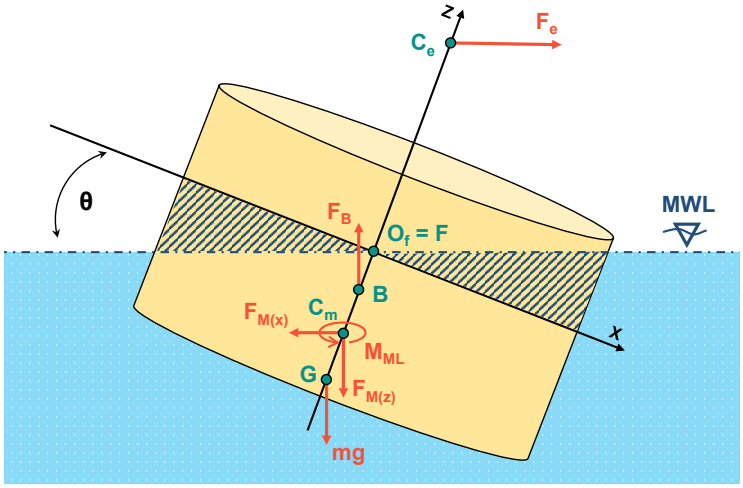


Figure 2.1: Simplified force diagram of a generic floating wind turbine.

The following analysis will assume that F_e is balanced by the mooring lines force in X, thus $F_{MX} = -F_e$. This pair of forces ultimately leads to the pitch inclining moment M_I with respect to O:

$$M_I = F_e \cdot (z_e - z_O) \cos\theta - F_{MX} \cdot (z_L - z_O) \cos\theta = F_e \cdot (z_e - z_L) \cos\theta \tag{2.2}$$

On the other hand, the pitching of the platform introduces a restoring moment M_R with three main contributions, which ultimately lead to the classification of FOWTs seen in section 1.1.2:

Waterplane area. Caused by the buoyancy loads of the additional submerged volume, which can be found by integrating the hydrostatic loads along this volume. The hydrostatic load resulting from an infinitesimal volume may be written as:

$$dF_h = \rho\, g\, dV = \rho\, g\, h\, dS = \rho\, g\, z\, dx\, dy = \rho\, g\, x\, \tan\theta\, dx\, dy \tag{2.3}$$

Where ρ is the water density and g the acceleration due to gravity. The waterplane area restoring moment can be found by integrating the above expression:

$$M_{WA} = \int_{S_0} x\, dF_h = \rho\, g \int_{S_0} x\, (x\, \tan\theta)\, dx\, dy = \rho\, g\, \tan\theta \int_{S_0} x^2\, dx\, dy = \rho\, g\, \tan\theta\, I_y \tag{2.4}$$

Where I_y is the second moment of area of the initial waterplane area S_0 (when $\theta = 0$). The above is true only if the cross section of the body is constant along z and for pitch angles small enough to ensure that the entire top surface of the body is above water level.

Moment arm effects. Caused by the relative movement of B and G, which introduce a lever arm into the weight $m\,g$ and buoyancy force $\mathrm{F_B}$. In FOWTs, $\mathrm{F_B}$ can be higher than the weight due to the downward mooring loads:

$$\sum \mathrm{F_Y} = 0 \;\rightarrow\; \mathrm{F_B} + \mathrm{F_{MZ}} - m\,g = 0\,,\;\; \mathrm{F_B} = m\,g - \mathrm{F_{MZ}} \tag{2.5}$$

Then the total moment arm effects $\mathrm{M_{MA}}$:

$$\begin{aligned} \mathrm{M_{MA}} &= \mathrm{F_B}\; z_B\; \sin\theta \;+\; \mathrm{F_{MZ}}\; z_L\; \sin\theta \;-\; m\,g\; z_G\; \sin\theta = (\mathrm{F_B}\; z_B \;+\; \mathrm{F_{MZ}}\; z_L \;-\; m\,g\; z_G)\; \sin\theta \\ \mathrm{M_{MA}} &= [m\,g\;\,(z_B - z_G) - \mathrm{F_{MZ}}\;\,(z_B - z_L)]\;\sin\theta \end{aligned} \tag{2.6}$$

Where the vertical mooring force $\mathrm{F_{MZ}}$ usually has negative sign. Note that the above moment is stabilising only if $\mathrm{M_{MA}} \geq 0$ or $z_G \leq z_B$ for freely floating bodies.

Mooring system. Mooring lines not only contribute to the inclining moment though $\mathrm{F_M}$, but some platforms such as TLPs can also provide stability via a restoring moment generated by a difference in the vertical components of the cables' tension. In a general case, this mooring lines moment $\mathrm{M_{ML}}$ depends on all six DOFs of the system, thus a 6×6 matrix $\mathbf{C_{ML}}$ is often used to represent the linear mooring stiffness, which accounts for coupling effects between DoFs. In this analysis, however, a linear pitch restoring moment is considered:

$$\mathrm{M_{ML}} = \mathrm{C_{ML\,(55)}} \cdot \theta \tag{2.7}$$

Finally, the equilibrium position of the entire system is found by:

$$\sum \mathrm{M} = 0 \rightarrow \mathrm{M_I} = \mathrm{M_R} \rightarrow \mathrm{M_I} = \mathrm{M_{WA}} + \mathrm{M_{MA}} + \mathrm{M_{ML}} \tag{2.8}$$

Leading to the following non-linear algebraic equation for θ:

$$\mathrm{F}_e \cdot (z_e - z_L)\,\cos\theta = \rho\,g\,I_x\,\tan\theta + [m\,g\;\,(z_B - z_G) - \mathrm{F_{MZ}}\;\,(z_B - z_L)]\,\sin\theta + \mathrm{C_{ML\,(55)}} \cdot \theta \tag{2.9}$$

Assuming small-angle approximation, the above expression can be linearised:

$$\theta = \frac{\mathrm{F}_e \cdot (z_E - z_L)}{\rho\,g\,I_x + m\,g\;\,(z_B - z_G) - \mathrm{F_{MZ}}\;\,(z_B - z_L) + \mathrm{C_{ML\,(55)}}} = \frac{\mathrm{M_I}}{\mathrm{C_{55}}} \;\rightarrow\; \mathrm{C_{55\,min}} \geq \frac{\mathrm{M_I}}{\theta_{\max}} \tag{2.10}$$

Where $\mathrm{C_{55}}$ is the linear pitch stiffness of the platform for small oscillations, which is close to $5 \cdot 10^8$ and $5 \cdot 10^9$ Nm rad^{-1} for a 5 MW turbine semi-submersible and spar platform, respectively [30].

2.1.2. Ocean waves characterisation

The wave spectra

Even though many different types of waves coexist on the ocean surface (as depicted in Figure 2.2), wind-driven waves propagated by the effects of gravity are of utmost interest for offshore structures, with usual periods between 0.1 and 30s [20]. These waves can be generated by the local wind, thus leading to irregular and short crested shapes; or generated in a distant area with a more regular distribution and longer crests (swell). Wind-driven generation of waves is a dispersive and random phenomena [28], thus calling for a stochastic description of ocean waves.

The Airy or linear theory is preferred to describe irregular sea states (~ 3 h window) by the superposition of elementary monochromatic waves (wind-driven and swell) using the Fourier transform. Different energy spectra $S(\omega\,,\theta)$ exist to characterize sea states as a function of the significant wave height H_s (average of the highest one-third waves) and average spectrum frequency [28], being JONSWAP (*Joint North Sea Wave Project*) a popular option for almost fully developed seas. Given a wave spectrum $S(\omega)$ that specifies the wave energy distribution over different frequencies, its moments m_n can be used to obtain helpful information regarding the sea state:

$$m_n(\omega) = \int_0^\infty \omega^n\; S(\omega)\;\mathrm{d}\omega \;\rightarrow\; \mathrm{H}_s \sim 4\sqrt{m_0} \tag{2.11}$$

Time-domain modeling is still needed for large, steep, or non-linear waves, usually embedded within a stochastic wave distribution.

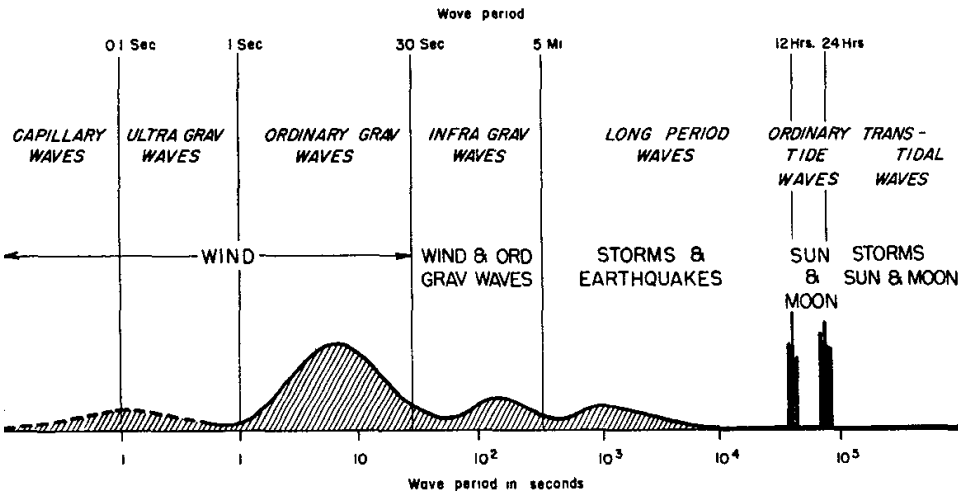


Figure 2.2: Wave energy content as a function of wave period. Reproduced from [31].

Unlike other waves, such as light and sound, water waves do not propagate (in a given medium) at a fixed speed, but each angular wavenumber has an associated frequency and phase speed. The equation describing the coupling between wavenumber and frequency is known as the dispersion relation and depends on the considered wave theory, as explored in the next section.

Wave theories

Generation and motion of gravity waves through the water surface is a rather complex process, which can be described through the Navier-Stokes equations in potential form. However, simplifying hypotheses are needed in order to reach manageable descriptions of waves. This gives rise to two main categories of wave theories:

Linear waves. Also known as Airy waves, they result from linearising the free surface kinematic and dynamic boundary conditions. In their complete form, these conditions state that the normal velocities of the fluid and free surface must equal at the boundary and that the fluid pressure should equal atmospheric. The linear theory ignores the high-order terms in these conditions and imposes them on the undisturbed plane of the free surface [28]. The simplest solution leads to plane progressive planes, with a surface elevation η of the form:

$$\eta_{(x,t)} = \mathrm{A}\cos\left(k\,x - \omega\,t + \epsilon\right) \tag{2.12}$$

Where A is the wave amplitude (half the wave height H), k the wavenumber, ω the frequency, and ϵ an arbitrary phase. Frequency and wavenumber are mutually dependent through the dispersion relation. Linear waves in deep water, where depth h is larger than $0.5\,\lambda$ ($\lambda = 2\,\pi/k$ is the wavelength), move through circular orbits whose radius attenuates exponentially with depth. Because of this circular motion, linear waves cannot capture wave drift.

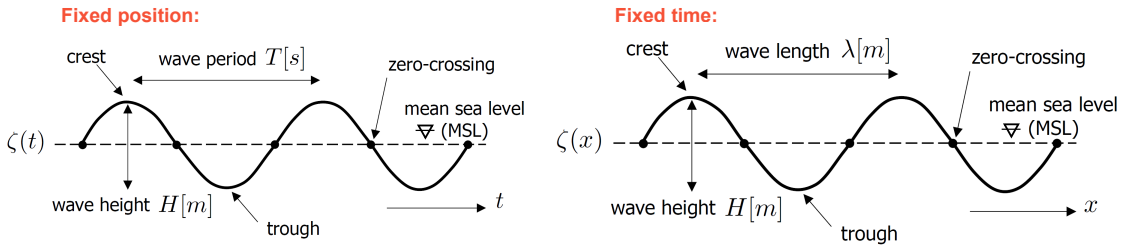


Figure 2.3: Main parameters characterizing the propagation of regular waves in space and time.

For shallow waters ($h < \lambda/20$), an impermeable bottom boundary condition must be imposed, leading

to elliptical trajectories that flatten with water depth. The dispersion relation for such waves reads as:

$$k \tanh(k\,h) = \omega^2/g\,, \qquad \text{deep water: } \lim_{h\to\infty} \tanh(k\,h) = 1 \;\rightarrow\; k = \omega^2/g \tag{2.13}$$

The greatest advantage of linear waves is that they allow for superposition, thus being capable of replicating arbitrarily irregular wave systems through Fourier series, which add ups a large number (theoretically infinite) of waves with different amplitude and frequencies. The relation between these amplitudes and frequencies is given by a spectral density function $S(\omega)$.

Non-linear waves. When wave amplitude or steepness increases, the assumptions of linear theory are no longer valid. Stokes tackled this issue by expanding the exact boundary conditions from the true free surface into an undisturbed plane by Taylor expansion. The order of these approximations can be arbitrarily high, although only up to fifth-order are used in practice. The free surface elevation in second-order Stokes waves, for instance, adopts the following shape:

$$\eta(x,t)^{\text{2nd}} = A\left\{\cos(kx-\omega t) + A\,k\,\frac{3-\sigma^2}{4\sigma^3}\cos(2\,[kx-\omega t])\right\} + \mathcal{O}\left(\varepsilon^3\right)\,, \text{ where } \sigma = \tanh(kh) \tag{2.14}$$

Neglecting higher-order terms, Stokes second-order waves can be re-written as an expansion of the first-order theory:

$$\eta(x,t)^{\text{2nd}} = \eta(x,t)^{\text{1st}} + A^2\,k\,\frac{3-\sigma^2}{4\sigma^3}\cos(2\,[kx-\omega t]) \tag{2.15}$$

The relative amplitude of second to first-order components is given by equation 2.16 and approaches zero for increasing values of relative water depth $h_r = h/\lambda$ and decreasing values of wave steepness $\varepsilon = H/\lambda$.

$$\frac{\hat{\eta}^{\text{2nd}}}{\hat{\eta}^{\text{1st}}} = \frac{A^2\,k\,\frac{3-\sigma^2}{4\sigma^3}}{A} = \pi\,\varepsilon\,\frac{3-\sigma^2}{4\sigma^3} \tag{2.16}$$

The dispersion relation is also affected by the non-linear terms, now weakly depending on the wave amplitude (i.e., larger waves travel faster than smaller ones). Stoke's waves are well suited for deep waters and short wavelengths.

Still, other non-linear solutions exist. Trochoidal waves exactly satisfy the free-surface boundary condition at the cost of violating the irrotational assumption of potential theory. Cnoidal waves are the solution of the Korteweg–De Vries equation, describing non-linear waves in shallow waters, which at the limiting case can model an infinitely-long solitary wave [32]. Finally, stream function theory defines a potential stream function whose coefficients are numerically determined to satisfy the kinematic and dynamic boundary conditions as closely as possible.

Note that above the wave-breaking limit, potential theory assumptions do not hold (viscosity, surface tension, and air mixing effects), and thus no satisfactory analytical solutions exist. A good starting value for the breaking height is $H_B = 0.8\,h$ in shallow waters or $H_B = 0.14\,\lambda$ for deep water. Some qualitative criteria on the validity of different wave theories for varying wave steepness and water depth are presented in Figure 2.4a.

It is important to note that, unlike Airy waves, non-linear waves are incompatible with the principle of superposition. The addition of two monochromatic waves of different periods and propagation directions gives rise to non-linear interacting effects, which manifest in the form of sum and difference frequency components. Due to the huge amount of wave-wave interactions, it is not feasible to model irregular sea states using non-linear waves.

Limitations of linear wave theory

As already presented in the previous section, the wave propagation phenomena is highly dependant on the water depth and wave height. Steep waves and shallow waters pose a great modelling challenge which linear theory misses. Because of the engineering interest in predicting wave loads, effort has been put into delimiting the range of applicability of the different wave theories. DNV, the world's largest classification society, provides an application limit for different wave theories depending on

wave steepness and water depth to wavelength ratio [33], as depicted in Figure 2.4a. Focusing on linear theory only, some authors have proposed simpler criteria for validity. Hedges [34] proposal is based on wave steepness $\varepsilon = H/\lambda < 0.04$ and Ursell number $U = H\,\lambda^2/h^3 < 40$. On the other hand, the CERC (*Coastal Engineering Research Center*) recommends restricting the use of linear theory to water depths above $h > gT^2/4\pi$ [35, 36].

For comparison purposes, these criteria can be transformed into the ε - h_r space, where $h_r = h/\lambda$ is the relative water depth. Rewriting the dispersion relation from equation 2.13 as a function of h_r gives:

$$\omega^2 = g\,k\,\tanh(k\,h) \;\rightarrow\; \left(\frac{2\pi}{T}\right)^2 = g\,\frac{2\pi}{\lambda}\,\tanh\left(\frac{2\pi}{\lambda}\,h\right) \;\rightarrow\; \lambda = \frac{T^2}{2\pi}\,g\,\tanh(2\pi\,h_r) \tag{2.17}$$

With the above transformation, the axes from DNV's criteria can be transformed:

$$\frac{H}{gT^2} = \frac{\varepsilon}{2\pi}\,\tanh(2\pi\,h_r)\,, \quad \frac{h}{gT^2} = \frac{h_r}{2\pi}\,\tanh(2\pi\,h_r) \tag{2.18}$$

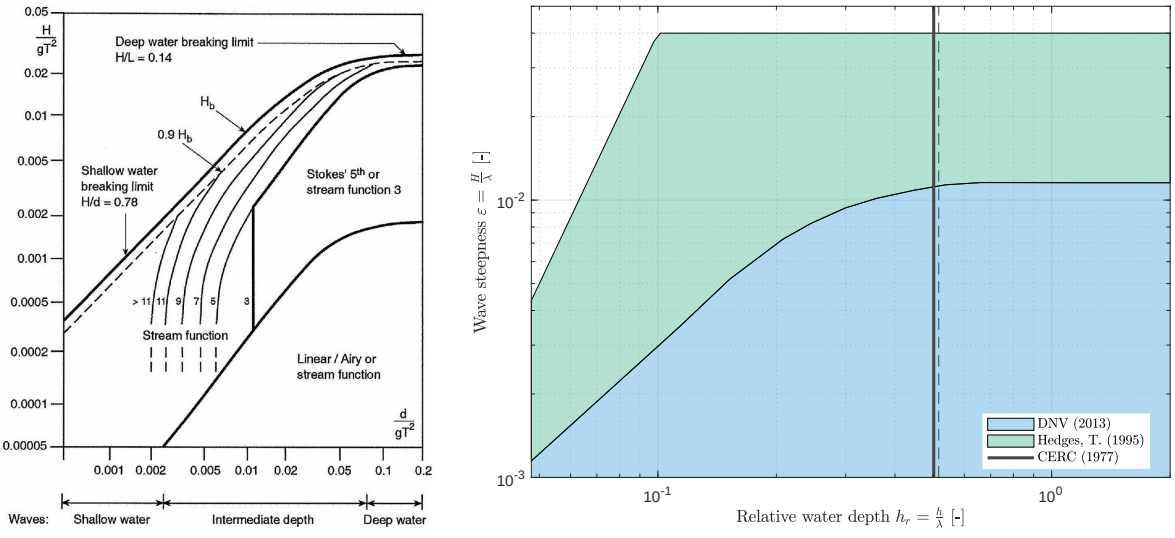
To map DNV's criteria into the ε - h_r space, the above system of non-linear equations is solved numerically. Similarly, equation 2.17 allows the CERC's criteria to be expressed as a non-linear inequation for h_r, which can also be solved numerically:

$$h > \frac{gT^2}{4\pi} \;\rightarrow\; h_r\,\tanh(2\,\pi\,h_r) - \frac{1}{2} > 0 \;\rightarrow\; h_r \gtrapprox 0.5018 \tag{2.19}$$

Finally, the Ursell number criteria from Hedges can also be transformed into the ε - h_r space:

$$U = H\,\lambda^2/h^3 = \frac{H}{h\,{h_r}^2} = \frac{\varepsilon}{{h_r}^3} < 40 \;\rightarrow\; h_r > \sqrt[3]{\frac{\varepsilon}{40}} \tag{2.20}$$

The linear theory delimiting regions for each of the presented criteria are shown in Figure 2.4b as a function of wave steepness and relative water depth. The criteria proposed by DNV is the most restrictive for the considered water depths.



(a) (b)

Figure 2.4: (a) DNV's criteria for the validity of different wave theories (reproduced from [33]) and (b) application range of linear wave theory using various criteria.

Wave-body loads
The interaction between waves and floating bodies is a reciprocal phenomenon in which the immersed body is affected by the fluid, and the latter is also altered by the presence of the body. If linear theory is applicable, then the resulting loads from this mutual interaction can be thought of as a combination of simpler load sources:

Excitation loads. Driven by the incident and scattered (diffraction) waves caused by the disturbance of the body. If the unsteady pressure field due to the incident waves is assumed to be unaffected by the presence of the body, then these loads are known as Froude-Krylov [28].

Radiation loads. Driven by the structure motions. These waves radiate outward from the body, thus extracting energy from it. Radiation loads are proportional to both the velocity and acceleration of the body, giving rise to damping and added mass effects which depend on the wave frequency.

Drag loads. Driven by viscosity and boundary layer detachment. They are usually confined within the boundary layer domain, but in certain regimes its detachment can ultimately lead to energy dissipation through vortex shedding.

Drag loads are highly non-linear in nature, just like other phenomena such as slamming, breaking waves, rogue waves and ringing, the latter described as a high-frequency vibration of the body affecting vertical cylinders in steep waves [38]. Other non-linear excitation loads arise when considering non-linear waves, such as difference-frequency and sum-frequency loads. The former are of great concern for FOWTs because of their high energy content at low frequency, potentially exciting the firsts structural modes.

An engineering formulation known as Morison's equation exists that approximates hydrodynamic loads as the addition of inertial, Froude–Krylov and drag forces acting parallel to the flow direction:

$$\mathbf{F} = \rho \mathrm{V} \dot{\mathbf{u}} + \rho\, c_a\, \mathrm{V}\, (\dot{\mathbf{u}} - \dot{\mathbf{v}}) + \frac{1}{2} \rho\, c_d\, \mathrm{S}\, (\mathbf{u} - \mathbf{v})\, \|\mathbf{u} - \mathbf{v}\| \qquad (2.21)$$

Where $\mathbf{u}$ and $\mathbf{v}$ are the wave and body velocities, V is the submerged volume, S is a reference area and c_d and c_a are the drag and added mass coefficients which depend on the specific body shape and flow regime. Because the equation ignores diffraction effects, it should be limited to slender bodies.

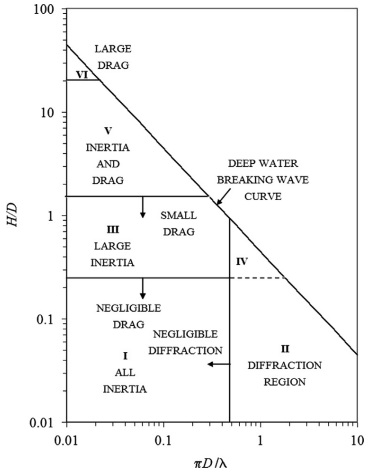


Figure 2.5: Main forcing regime for waves as a function of the characteristic length D, wavelength λ and wave height H [37].

2.2. Computational fluid dynamics

2.2.1. Modelling overview

Computational fluid dynamics (CFD) refers to a family of methods that numerically solve (i.e. approximate in a discrete domain) either the Euler or Navier-Stokes equations. The latter can be written in their incompressible form as:

$$\begin{gathered} \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\, \mathbf{u}^T) - \nu\, \nabla^2 \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{q} \\ \nabla \cdot \mathbf{u} = 0 \end{gathered} \qquad (2.22)$$

Where $\mathbf{q}$ is a general momentum source term. The most challenging term of the above equations is the non-linear convection of momentum, which ultimately drives turbulence. Different approaches exist to handle this term:

Direct Navier-Stokes (DNS). All length and time scales of turbulent flow are captured with no further assumptions. DNS is indispensable for fundamental research but its computational cost scales with the cube of the Reynolds number (Re), making it not suitable for engineering flow simulations.

Large Eddy Simulations (LES). The most energetic turbulent scales are resolved (inertial range of the energy cascade), but the small dissipative ones are modelled. LES simulations are independent of Re

except in boundary layers. LES can simulate highly complex flows, but in most cases the computation cost is inadmissible for everyday simulations.

Reynolds-Averaged Navier-Stokes (RANS). Only structures of the mean flow are resolved, while all turbulent fluctuations are modelled. This method can be extended to account for slow unsteady effects (URANS). Computational costs are lower but still more demanding than lower fidelity tools.

Hybrid methods exist that combine RANS and LES, such as zonal coupling or detached eddy simulations. Both RANS and LES are based on mathematical constructions that add more unknowns than equations. In RANS, these new terms are gathered in the Reynolds stress tensor. To solve the new equations, empirical approximations known as turbulence models are needed. These models will incorporate new equations, either algebraic or differential, that will allow solving the system. For external flows and in particular for wind turbine applications, the shear stress transport (SST) model is widely used. It is an eddy viscosity model that adds new differential equations for the transport of turbulent kinetic energy and specific dissipation rate. The SST model is a combination of the Wilcox $k-\omega$ model, well suited near the wall, and the $k-\varepsilon$ model, ideal for regions away from the wall.

The resulting differential equations can be discretised utilizing different methods, the most popular one being the finite volume method (FVM) for its relative ease of implementation and conservative nature. The FVM decomposes the domain into non-overlapping control volumes over which the conservation laws are integrated. Many discrete quadrature and interpolation approximations exist for this purpose, resulting in different truncation errors (numerical diffusion and dispersion) and order of convergence. Several time marching methods exist to integrate the resulting discrete system in time, usually divided into explicit and implicit. The former methods are less computationally intensive but require smaller time-steps to ensure stability. A helpful indicator is the Courant-Friedrichs-Lewy number (CFL), which defines how fast the solution physically propagates through the domain compared to the computational stencil:

$$\text{CFL} = \Delta t \, \frac{\|\mathbf{u}\|}{\Delta x} \leq 1 \tag{2.23}$$

Many different algorithms exist to solve the discrete system, being SIMPLE and PISO popular solvers based on pressure-velocity coupling for steady and transient simulations, respectively. There are many CFD softwares available, such as Ansys FLUENT/CFX, STAR-CCM, OpenFOAM or COMSOL. Specific solutions also exist for the pre and post-processing steps.

2.2.2. Multiphase flow

The CFD modelling for floating platforms follows the principles mentioned above. The total forces and moments acting on the floating body can be computed by integrating the pressure loads given by the CFD solver over the surface. However, adding a free surface between water and air requires the definition of a new formulation called volume of fluid (VOF).

The VOF method was first introduced by Hirt and Nichols in 1981 [39] to efficiently represent free surfaces in Eulerian simulations. An indicator scalar field $\alpha \in [0, 1]$ is defined for each cell representing the ratio of the volume occupied by a given phase, usually $\alpha = 0$ for gas and $\alpha = 1$ for liquid. The original formulation proposed a search algorithm to define the free surface, in which boundary cells contain a non-zero value of α and have at least one neighbouring cell with zero-valued α. Defining the free boundary as the cells with iso-contour $\alpha = 0.5$ is widely used for visualization purposes. Alternatives to the VOF approach exist, such as the line segments method, the marker particles method or the level set method [40].

Multiphase flows are inherently multi-scale in nature, meaning a cascade effect is acting at different scales, just like in turbulence. The free surface can generate large bubbles that progressively break down into smaller structures. The VOF method can be thought as an LES filter: bubbles larger than grid scales can be captured, but smaller structures are averaged. Multiphase flow poses great modelling and simulation challenges, such as surface tension effects, dynamic interfaces, small-scale structures interaction, turbulence modelling, phase mixing, etc., which fall beyond the scope of the present project. Overall, the VOF method incorporates a new transport equation to be coupled with Navier-Stokes [41]:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot [\mathbf{u}\,\alpha] + \nabla \cdot [\mathbf{u}_r (1-\alpha)\,\alpha] = 0\,, \quad \phi = \alpha\,\phi_l + (1-\alpha)\phi_g \tag{2.24}$$

$$\mathbf{u}_r = \min\left(c_\alpha \left\|\mathbf{u}\right\|, \max(\left\|\mathbf{u}\right\|)\right) \cdot \hat{\mathbf{n}} \tag{2.25}$$

Where the fluid variables ϕ (e.g. the velocity field $\mathbf{u}$, the density ρ or the viscosity μ) are computed as a weighted average based on the volume fraction of liquid and gas. Equation 2.24 models the advection of the indicator field and requires from a heuristic surface compression mechanism, $\nabla \cdot [\mathbf{u}_r(1-\alpha)\,\alpha]$, to avoid smearing of the free surface. Here, $\mathbf{u}_r$ is the vector of relative velocity between the two phases, normal to and pointing towards the free surface. It acts as a compression term that sharpens the free surface (avoiding smearing) and vanishes far away from it. The normal vector $\mathbf{n}$ of the free surface is found as the direction in which α changes more rapidly. The coefficient $c_\alpha \in (0, 2)$ is used to control the compression effects.

2.2.3. Actuator models

To drive down computational expenses while capturing the effects of nonlinear convection and turbulent transport, the Navier-Stokes equations can be coupled with actuator-type models that parametrize the turbine loading and hence do not resolve the fine details of boundary layers. Actuator models do not need intense refinement because the actual blades are not represented in the mesh, so moving grids are not needed either. The effect of the blades on the flowfield is modelled with tabulated airfoil data, just like BEMT and lifting line methods. Depending on the underlying assumptions, two main actuator-based models exist (Figure 2.6):

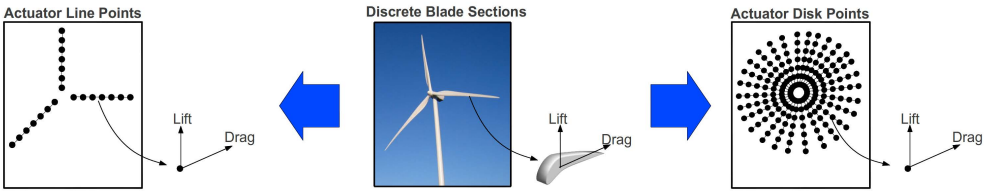


Figure 2.6: Schematic comparison of ALM and ADM discretisation, reproduced from [42].

Actuator line model

Originally developed by Sørensen and Shen [18] to validate the assumptions upon which lower-fidelity models are based, the actuator line model (ALM) has been popularised to fill the gap that exists between blade-resolved CFD and vortex methods, especially for modeling entire wind farms [1]. In ALM, turbine blades are treated as a combination of actuator elements (usually 25-60 [43, 44]) defined by their quarter-chord location. Since each blade is individually represented, the tip and root vortical structures can be captured. Each element introduces an additional forcing term in equation 2.22, which after being projected in the normal z and azimuthal θ directions reads as:

$$\mathrm{F}_\theta = \frac{\rho\, c\, \left\|\mathbf{W}\right\|^2}{2\, r\, d\theta\, dz}\left(c_l \sin\varphi - c_d \cos\varphi\right), \quad \mathrm{F}_z = \frac{\rho\, c\, \left\|\mathbf{W}\right\|^2}{2\, r\, d\theta\, dz}\left(c_l \sin\varphi + c_d \cos\varphi\right) \tag{2.26}$$

Where c is the local chord, $\mathbf{W}$ is the local velocity relative to the blade element, and φ denotes the inflow angle. The velocity W should be chosen such that the effects of the upwash and downwash created by the bound vortex are not seen [43], and thus a control point defined at the center of each line element is usually chosen. Velocity will then be sampled by linear interpolation of the surrounding volumes. More advanced methods [44] use an integral velocity sampling, where W at each element is obtained through the integration of the velocity field weighted by a force projecting function.

In ALM, time-step is more restricted by the tip speed than by the CFL condition, and it must be ensured that the blade's tip does not pass through more than a single cell each time-step [42]. To avoid singular behaviours and numerical instabilities, the forces must be spread over the body by convoluting the loads with a regularization kernel, also known as *"smearing function"* η. A typical choice is the spherical Gaussian function:

$$\eta = \frac{1}{\epsilon^3 \pi^{3/2}} \exp\left[-\left(\frac{\left\|\mathbf{d}\right\|}{\epsilon}\right)^2\right] \tag{2.27}$$

Where **d** is the vector pointing from the control point to the cell where the force is being projected and ϵ is a constant parameter specifying the smearing width. Different criteria exist for selecting ϵ [1]. Based on the grid size, the smearing width should be equal to twice the local cell length to guarantee stability. Based on the local airfoil chord, ϵ should be close to 25% of the chord. Another criterion by Jha et al. [43] states that the smearing width should be proportional to the sectional blade force rather than the blade platform and proposes a variable spreading factor based on an effective elliptic platform loading. Note that large values can increase the effective span of the blade, making it appear more efficient and incorrectly modeling the tip behaviour, while small values can ultimately lead to instabilities. Forsting et al. [45] used a vortex-core based correction for ϵ which mitigated the load overprediction at the tip and root.

The main weakness of ALM is its dependence on airfoil data. For this reason, corrections accounting for dynamic stall, added mass effects, and 3D effects are usually included in ALM. In general, ALM has shown good agreement with experimental results, but it is not able to accurately model the blade-wake interactions or the near wake structures, and in combination with RANS usually adds excessive dissipation into the wake, which adopts the shape of coarse vortex tubes [18]. Nonetheless, such methods allow for a reduction in computation cost of 2-4 orders of magnitude compared with blade-resolved CFD [46, 1]. Moreover, ALM can be extended to model the effects of the tower and nacelle onto the flowfield [47].

The ALM can be used in combination with various turbulence modeling approaches, e.g. RANS, URANS and LES. In the latter two cases, the modelled flow-field is inherently unsteady, and the resulting method is consequently named unsteady-ALM, UALM to shorten. Because the present report focuses on floating turbines with unsteady dynamics, both acronyms (ALM and UALM) will be used interchangeably.

Actuator disk model

The actuator disk model (ADM) is an extension of the BEMT model, where the hypothesis of annular independence along the streamtube is replaced by the non-linear convection of the Navier-Stokes equations [18]. The ADM changes the solid blades by a permeable surface representing its swept area, which can be a disk or a circular cone for blades with a conning angle [48]. The disk is then discretized into different elements, each of them exerting a load obtained through polar data and local airfoil characteristics. Since this will introduce a step pressure jump across the disk, a regularization kernel can be applied to distribute the loads onto the neighboring cells. Using a finite thickness to define an actuator volume is also a common approach [49]. More corrections must be applied in the ADM compared with the ALM since it does not inherently account for tip-root losses.

This method has proven to work well in axisymmetric flow conditions even for heavily loaded rotors and dynamic inflow [18]. But because the influence of the blade on the flowfield is taken as the integrated loads along the disk, there is no circulation distribution along individual blades and thus no asymmetric helical tip vortices. The wake is thus expected to be symmetrical, and contrary to ALM, ADM is not able to capture the asymmetric roll-up of Kelvin-Helmholtz-like instabilities in the wake's outer shear layer [42]. On the other hand, computational demands are lower than in ALM because the actuator disk always projects the loads onto the same neighboring elements and no search algorithm is needed. Furthermore, the time-step is no longer restricted by the tip speed. This method is thus recommended for cases where far wake effects are concerned but information regarding the near wake structures is not relevant.

2.3. Fluid-Structure Interaction

2.3.1. General concepts

Floating turbines are dominated by the aero- and hydrodynamic loads resulting from the wind-wave interactions with the rotor and platform. At the same time, however, these loads heavily depend on the position and motion of the platform as given by the elastic or rigid body equations. This is known as a fluid-structure interaction (FSI) problem: pressure loads cause the structure to move, leading to a change in the body-fluid interface that will, in turn, modify the flow-field and the resulting loads acting on the structure, giving rise to a closed-loop link between the solid and fluid domains. Two main strategies can be followed to tackle it:

Monolithic coupling. Fluid and structure equations are solved simultaneously. Boundary conditions between the two domains are automatically met, introducing no numerical error and thus higher stability and accuracy. However, the same solver needs to be used for both domains, which results in increased complexity and higher memory demand.

Partitioned coupling. Separate solvers are used for the fluid and solid domains, thus already existing solvers can be used. Since they are developed for a specific problem, these solvers are significantly simpler to maintain, validate, and optimise, making partitioned FSI the most popular approach. This, however, comes at a cost: since the boundary conditions between the domains are not automatically met, a partitioning error is introduced.

Partitioned coupling takes advantage of well-established solvers for individual domains, thus applying to a wider range of cases than monolithic coupling, where a different solver needs to be developed for each specific application. Partitioned schemes, however, require special coupling methods that limit the partitioning error. If these are not applied, the model is said to be *loosely coupled* and the error introduced by the discrete time integration may introduce instabilities and non-physical motions.

For FSI problems, this error is introduced as an artificial added-mass term proportional to the fluid-body density difference and the time-step size [50]. If the latter is too large, the simulation will diverge because the fluid-body coupling cannot be resolved in such a coarse time-step. Contrarily, for very small time- steps, the added-mass error will exponentially grow near the peaks of body acceleration, increasing numerical instability. Even though a compromise can be made and an *"optimized"* time-step could be selected, this is not known a priori, and even so, there is no guarantee that the time-advancing scheme will remain stable [51].

Loosely coupled methods solve the flow/structure equations once per time-step via either serial (Gauss-Seidel), parallel (Jacobi) or staggered integration schemes. Because the flow solvers usually require smaller time-steps, subcycling methods use different step sizes for the two solvers. If a stronger coupling is intended, then sub-iterations must be performed at each time-step to reduce the interface residual leading to *tightly coupled* models. If these iterations are performed until convergence to the monolithic solution is achieved, then the model is *fully coupled*. The most basic iteration methods are based on fixed-point modifications of the serial and parallel schemes, though under-relaxation is needed to increase the stability. Akiten's method uses an adaptive under-relaxation that is more stable than traditional Gauss-Seidel. To achieve faster convergence than with fixed-point methods, quasi-Newton methods can be implemented that solve the minimization problem of the interface residual.

In [50], N. Bruinsma et al. compared two different methodologies that tighten the FSI coupling in a floating body simulation. First, an under-relaxation method is applied to the acceleration of the mass center of the body, acting as a low-pass filter that removes high-frequency oscillations but introduces numerical diffusion that negatively affects the convergence rate. In this case, sub-iterations are performed only for the rigid body equations. Second, a predictor-corrector method is used that performs sub-iterations for both the fluid and rigid body equations, applying an under-relaxation factor to the pressure field. Both methods were found to significantly increase the numerical stability of the model, even though the predictor-corrector method was more effective at eliminating oscillations. A similar study is performed in [51] by J. Dunbar et al., where loosely and tightly coupled methods are compared, the latter making use of Akiten's dynamic under-relaxation method.

2.3.2. Rigid body dynamics

The extra degrees of freedom in FOWTs introduce a strong coupling between the external hydrodynamic and aerodynamic loads and the platform motion. In its simplest form, the relation between loads and motion is given by the rigid body equations. Given that all loads acting on the FOWT were known, then through the rigid body assumption, its motion could be described as a function of the mass center (G) displacements ($\mathbf{d}$) and rotations ($\boldsymbol{\phi}$) given by the Newton-Euler equations. The latter can be posed as a system of second-order differential equations:

$$\begin{aligned} m\,\ddot{\mathbf{d}} &= \sum \mathbf{F}_i = \mathbf{F}_{\text{hydro}} + \mathbf{F}_{\text{aero}} + \mathbf{F}_{\text{moor}} + \mathbf{F}_{\text{grav}} \\ \underline{\mathbf{J}}\,\ddot{\boldsymbol{\phi}} + \dot{\boldsymbol{\phi}} \times (\underline{\mathbf{J}} \cdot \dot{\boldsymbol{\phi}}) &= \sum \mathbf{r}_i \times \mathbf{F}_i + \sum \mathbf{M}_i + \mathbf{M}_{\text{gyro}} \end{aligned} \tag{2.28}$$

Where m is the total mass, $\underline{\mathbf{J}}$ is the time-invariant 3×3 inertia matrix with respect to G and $\mathbf{F}_i$ are the different load vectors containing the forces and moments acting in all six DoFs. Note that the Newton-Euler equation accounts for the non-linear gyroscopic coupling moment of the platform in the second left-hand term on the moment equation. However, the turbine effect is introduced as an external load since most rigid body simulations do not explicitly consider the turbine rotation motion:

$$\mathbf{M}_{\text{gyro}} = -\dot{\boldsymbol{\phi}} \times \underline{\mathbf{J}_R} \cdot \boldsymbol{\Omega}_0 \tag{2.29}$$

A more in-depth treatment of the gyroscopic load is presented in 8.3. Assuming that no relative angular acceleration acts on the turbine, the Coriolis force is the main responsible for gyroscopic effects [52]. In the above equation, $\boldsymbol{\Omega}_0$ is the constant angular speed of the rotor with respect to the platform and $\underline{\mathbf{J}_R}$ is the rotor inertia about its rotation axis. The cross product indicates that the direction in which the gyroscopic moment acts is perpendicular to both the floater and turbine rotation axes. Thus, for a FOWT undergoing pitch motion, the gyroscopic effect introduces a coupling moment in yaw.

In the most general form where $\mathbf{F}_i$ cannot be posed in terms of the state vector, the above equation must be solved through numerical integration. Nonetheless, from equation 2.29 it can be deduced that gyroscopic loads are proportional to $\dot{\boldsymbol{\phi}}$. Similarly, hydrodynamic loads from the water-body interaction (without the action of waves) can be posed as a function of the state vector by means of linear hydrodynamics and potential theory [28], which consider small-amplitude motions around equilibrium and neglect non-linear effects. As seen in section 2.1.1, mooring line forces can be assumed to be linear with the platform motions. All in all, these assumptions lead to the following system:

$$\left(\underline{\mathbf{M}} + \underline{\mathbf{A}}\right)\ddot{\mathbf{x}} + \left(\underline{\mathbf{B}}_{\text{hyd}} + \underline{\mathbf{G}}\right)\dot{\mathbf{x}} + \left(\underline{\mathbf{C}}_{\text{hyd}} + \underline{\mathbf{C}}_{\text{moor}}\right)\mathbf{x} = \mathbf{F} \tag{2.30}$$

Where $\mathbf{x} = [\mathbf{d}\ ,\ \boldsymbol{\phi}]^{\,T}$ is the state vector, $\underline{\mathbf{M}}$ is the mass-inertia matrix, $\underline{\mathbf{G}}$ the gyroscopic matrix, and $\underline{\mathbf{C}}_{\text{moor}}$ the mooring lines stiffness matrix. $\underline{\mathbf{A}}$, $\underline{\mathbf{B}}_{\text{hyd}}$ and $\underline{\mathbf{C}}_{\text{hyd}}$ are the hydrodynamic added mass, damping and stiffness matrices, respectively, with the first two being frequency-dependent. Typically, panel methods (also known as *Boundary Element Methods* or BEM) are the preferred numerical technique to obtain the latter matrices. The force vector $\mathbf{F}$ still needs to be computed for the wave, current, aerodynamic and gravity forces, for which an analytical description in terms of the state vector is not plausible. The benefit of equation 2.30 is that it can be transformed into the complex-frequency domain by assuming harmonic solutions [30]. One can then study the free (unforced) oscillations of the system by solving the eigenvalue problem associated with the homogeneous equation:

$$\left|\left(\underline{\mathbf{M}} + \underline{\bar{\mathbf{A}}}\right)^{-1} \cdot \left(\underline{\mathbf{C}}_{\text{hyd}} + \underline{\mathbf{C}}_{\text{moor}}\right) - \lambda\,\underline{\mathbf{I}}\right| = 0 \tag{2.31}$$

Where the eigenvalues correspond to the square of the system's natural frequencies. An early study on the natural frequencies is key to assess the platform response for a given wind and wave spectra. If coupling between different DOFs is neglected, a simple analytical form for the six undamped natural frequencies is found:

$$\omega^i = \sqrt{\frac{C^{ii}_{\text{hyd}} + C^{ii}_{\text{moor}}}{M^{ii} + A^{ii}_{(w_i)}}} \tag{2.32}$$

For hydrodynamic analyses, *Response Amplitude Operators* (RAOs) are often used to describe the system response due to wave excitation. For each DoF and wave frequency, the RAO is defined as the body motion caused by a wave of unit amplitude, assuming linear hydrodynamics and forces $\mathbf{F}_0$ proportional to wave amplitude. Because of these linear hypotheses, RAOs are superposable and thus popular in frequency-domain analyses, where they can be obtained through the following transfer function:

$$\text{RAO}(\omega) = \frac{|x_i|}{A} = \frac{\mathbf{F}_0}{\underline{\mathbf{C}} - (\underline{\mathbf{M}} + \underline{\mathbf{A}}(\omega))\omega^2 + i\underline{\mathbf{B}}(\omega)\omega} \tag{2.33}$$

RAOs are complex values, the imaginary part representing the response out of phase with the wave motion. But in most cases, forces cannot be linearized with wave amplitude since they are based on non-linear hydrodynamics and include aerodynamic, gravity, buoyancy and gyroscopic effects. In such cases, *effective* RAOs are defined as the *"difference in response amplitudes between non-linear time-domain simulations run with and without wave excitation"* [53]. A different effective RAO should be defined for every non-linear loadcase, e.g. each combination of wind-wave conditions.

2.3.3. Dynamic mesh methods

In blade-resolved CFD simulations, it is of special interest to capture the aerodynamic details near the walls such as boundary layer transition, separation, and stall. For this reason, the mesh is usually refined near the bodies, even with prism layers in the normal direction. In cases where the body position varies with time (i.e. rotating blades or moving floaters), some mechanism has to be used to account for the moving boundaries. Since regenerating the grid at each time-step would be time-consuming, other techniques that automatically take into account the body's motion are available:

Multiple reference frame. The computational domain is split into non-overlapping blocks, each one with an associated rotation axis and speed. New source terms are added to the modeling equations to account for the non-inertial loads (e.g., centripetal and Coriolis) with different methods to estimate fluxes at the boundaries. This method isn't truly a dynamic mesh technique, but rather aims to capture the steady solution of an unsteady problem.

Immersed boundary. In this method, fluid and bodies are represented with an Eulerian and Lagrangian formulation, respectively. The mesh is thus divided into fluid, solid, and boundary elements. The modeling equations must be modified in the vicinity of the boundary to incorporate boundary conditions. This method can account for large deformations and topology changes, but interpolation is usually non-trivial and boundary layers can be difficult to capture.

Morphing mesh. Only one mesh is created that adapts to the motion of the moving body. In grid connectivity schemes, nodes move depending on their neighbours (spring analogy or PDE based), whereas in point-by-point schemes nodes move independently (master-slave coupling or radial basis function interpolation). These methods are well suited when solids may undergo elastic deformations and can account for topology changes, but the introduced error increases as displacements enlarge.

Sliding mesh. The simulation is performed across disconnected but adjacent mesh domains moving relative to one another, with interface zones bounding each domain. The different zones slide (rotate or translate) relative to one another. Even though the domains are geometrically separated, an arbitrary mesh interface (AMI) ensures that field values remain continuous across the interface. For simple cases cylindrical domains are often used, while bodies rotating in various DOFs require spherical sub-domains.

Overset/Chimera grid. A static background mesh is coupled with an overlapping moving body-conformal mesh. A "hole" is cut in the static mesh and information is exchanged between different zones through interpolation in the overlapping regions. This method allows for different meshes to be used to represent distinct geometrical features. It does not account for solid shape deformation, but it can hold large displacements and rotations while keeping good boundary layer quality. Search algorithms must be employed to find which cells are overlapping, which can be computationally expensive.

Adaptive mesh refinement. The mesh is dynamically refined in certain regions of the domain. The amount of refinement is often based on scalar field or its gradient (e.g. $\nabla\alpha$ for free surface refinement). Because this method changes the number of grid cells during computation, the topology of the initial mesh is not preserved.

The use of dynamic meshes makes the simulation intrinsically unsteady, thus tightening the stability and accuracy requirements compared to static meshes. The mesh can experience instantaneous accelerations and consequently velocities that will be transferred onto the flow, hence robust numerical schemes and solvers should be preferred. If morphing meshes are used, the mesh quality must be monitored as the simulation progresses since there is no guarantee that the mesh can accommodate relatively large displacements or rotations. All dynamic mesh approaches are computationally expensive and intricate to set-up, so it is the user's responsibility to select the most appropriate method for

each specific case.

For the arbitrary motions characterising a FOWT, morphing and overset are the most suited techniques. Wind et al. investigated in [54, 55] the performance of both methodologies in OpenFOAM within the context of Wave Energy Converters (WEC), and compared the results with those from experimental campaigns. It was concluded that both methods yield similar results, but the associated costs of the overset mesh were significantly higher. All in all, they advocate for the use of morphing meshes for small motions that do not compromise mesh quality and hence numerical stability. Overset techniques remain the most versatile and robust tool for large motions along multiple DoFs [56].

2.4. OpenFOAM and the Finite Volume Method

OpenFOAM, acronym for *Open Source Field Operation and Manipulation*, is an open-source C++ library with extensive multi-physics simulation capabilities used to numerically solve partial differential equations (PDEs). It is widely used in computational fluid dynamics where it takes advantage of the finite volume method (FVM) to approximate the fluid equations. This section aims to present the basic FVM formulation and dissect the PIMPLE algorithm, which will be used throughout the present thesis to solve for the coupled multiphase flow and rigid body equations.

For a more general description of the OpenFOAM environment, the reader is referred to Appendix 8.3. Novice users are highly encouraged to also explore external sources. The official user tutorial and programmers guides provide a robust description of the case set-up, available functionalities and applications, discretization methods, programming environment, and more. Besides that, the manual by Gerhard Holzinger [57] contains a more extensive and detailed explanation of the most relevant topics whereas the book by Tobias Holzmann [58] focuses on the mathematics and numerics. The architects of OpenFOAM recently published an access-free book [59] focused on the main modelling techniques, numerical methods, and algorithms. The OpenFOAM tutorial collection is a great place to start, with a curated list of tutorials available for different levels of experience. Finally, the wiki and forum webpages are life-saving sources when dealing with everyday life simulations and troubleshooting.

2.4.1. Discretization of the governing equations

Even though many techniques exist to solve the Navier-Stokes equations, the FVM is widely used in commercial codes due to its conservative approach and robust implementation. This method splits the simulation domain into a union of control volumes (grid cells) representing a region of space where the physical laws are actually enforced. Inside each volume, the integral form of equation 2.22 obtained by applying Gauss' theorem is approximated:

$$\frac{\partial}{\partial t}\int_V \rho\,\mathbf{u}\,\mathrm{d}V + \int_S \rho\,\mathbf{u}\mathbf{u}\cdot\hat{\mathbf{n}}\,\mathrm{d}S = -\int_S p\,\mathrm{d}S + \int_S \underline{\boldsymbol{\tau}}\cdot\mathbf{n}\,\mathrm{d}S + \int_V \rho\,\mathbf{q}\,\mathrm{d}V \tag{2.34}$$

Where $\underline{\boldsymbol{\tau}}$ is the shear rate tensor. Fluxes on the cell surface S are commonly approximated from the cell average solution. Several quadrature methods exist to approximate the integral of these fluxes. On the other hand, interpolation methods are used to extract cell surface values from center values. Since the values at a cell face are computed using the center value of its neighbours, the governing equations of different cells are coupled together. Once discretised, these equations can be assembled into a matrix system. The FVM traditionally assembles individual matrices for each variable which are then solved in separate (segregated) steps, usually requiring iterative procedures. Appendix 8.3 presents the main algorithms that fulfil this purpose in OpenFOAM.

Still, for unsteady problems the time-derivative part of equation 2.34 must be modelled. Assuming a generic conservation law which represents the semi-discrete form of equation 2.34, where φ is the quantity of interest:

$$\frac{\mathrm{d}\,\varphi(t)}{\mathrm{d}t} = f(t,\varphi(t)) \;\rightarrow\; \varphi^{n+1} = \varphi^n + \int_{t_n}^{t_{n+1}} f(t,\varphi(t))\mathrm{d}t \tag{2.35}$$

Different time marching methods exist, both implicit and explicit, to approximate the integral term of the above equation.

2.4.2. The Arbitrary Lagrangian-Eulerian method

For moving meshes, a new formulation called Arbitrary Lagrangian-Eulerian (ALE) must be adopted to account for the additional convective terms due to mesh velocity. With ALE, the flow-field is described by the usual Eulerian approach, whereas the mesh motion is accounted for with a Lagrangian formulation. For the momentum equation, the term that is added on the left hand side of equation 2.34 reads as:

$$-\int_S \rho\, \mathbf{u}\, \mathbf{v}_m \cdot \hat{\mathbf{n}}\, \mathrm{d}S \tag{2.36}$$

Where $\mathbf{v}_m$ is the velocity of the mesh, which is generally unknown. For this reason, a geometric conservation law (GCL) for the mesh velocity must be formulated for moving meshes:

$$\frac{d}{dt}\int_{\mathrm{V}} \mathrm{dV} = \int_{\mathrm{S}} \mathbf{v}_m \cdot \hat{\mathbf{n}}\, \mathrm{dS} \tag{2.37}$$

The above equation simply states that the change in volume of a cell must equal the volume swept by the boundary. The presented GCL is said to be consistent: if satisfied, uniform flows are conserved on moving meshes. Evaluation of the right hand term will depend on the specific discretization scheme.

2.4.3. Understanding the PIMPLE algorithm

Even though many sequential (or segregated) solvers exist in OpenFOAM (see Appendix 8.3), the present thesis relies on PIMPLE: a merged PISO-SIMPLE algorithm suitable for FSI problems thanks to its iterative strategy. Because of the major importance PIMPLE has for the current work, its implementation in OpenFOAM is detailed here.

Pressure-velocity coupling

In sequential solvers, the FVM solves an individual matrix equation for each variable. For instance, equation 2.34 is decoupled into three linear matrix equations, one for each velocity component:

$$\underline{\mathbf{A}}\,\mathbf{u} = \mathbf{b}\,, \quad \mathbf{b} = \mathbf{H}(\mathbf{u}) - \nabla p \tag{2.38}$$

Where $\underline{\mathbf{A}}$ contains the linear terms whereas $\mathbf{H}(\mathbf{u})$ is a function of $\mathbf{u}$ and other sources. Non-linear, pressure and source terms are all contained within the source vector $\mathbf{b}$. Once the velocity field is known, pressure can be updated using the modified mass conservation law:

$$\nabla \cdot \mathbf{u} = 0 \;\rightarrow\; \nabla^2 p + \nabla \cdot [\nabla \cdot (\mathbf{uu})] = 0 \;\rightarrow\; \nabla \cdot \frac{1}{\underline{\mathbf{A}}} \nabla p = \nabla \cdot \left[\frac{\mathbf{H}(\mathbf{u})}{\underline{\mathbf{A}}}\right] \tag{2.39}$$

Because variables are solved once at a time, the overall system is solved by *"successive substitution of solved variables into the source vectors of subsequent equations"* [59]. This approach is widely known as predictor-corrector and consists of three basic steps:

1. **Momentum predictor**. Equation 2.38 is solved for $\mathbf{u}$, where the source vector $\mathbf{b}$ is explicitly computed from the current velocity and pressure values.
2. **Pressure equation**. With the new value of $\mathbf{u}$, pressure is solved from equation 2.39.
3. **Momentum corrector**. The velocity field is recomputed from equation 2.38 taking the updated values of velocity and pressure.

An intermediate step called *flux corrector* is often applied after the momentum corrector where $\mathbf{u}$ is interpolated to cell faces. If this sequence is applied once per time-step, then the resulting algorithm is known as SIMPLE (*Semi-Implicit Method for Pressure-Linked Equations*) and is suitable for steady-state solutions. For transient simulations, a number of PISO (*Pressure Implicit of Split Operations*) loops are performed between steps two and three, leaving the flux corrector step for the end. This method requires however sufficiently small Δt to ensure CFL~ 1. The PISO method may solve the pressure equation several times to correct for the non-orthogonality between cell faces.

The implementation
The PIMPLE algorithm results from the combination of the two presented methods. The PISO loop is maintained while a second PIMPLE loop encompassing all three steps is added. Because equations are solved using variables updated within the time step, temporal accuracy can be maintained for CFL>1. And more importantly, the PIMPLE loop allows for the FSI coupling between the turbine loads and rigid body motion. The number of PISO and orthogonal correction iterations is fixed while the PIMPLE loop can either be fixed or repeated until converge of the equation residuals.

This partitioning approach based on serial sub-iterations, as presented in Figure 2.7, will be used to strongly couple (i.e. with a fixed number of sub-iterations) the multiphase fluid and rigid body equations. Turbine ALM loads are obtained in the source term calculation step based on the current motion state of the rigid body, and applied to the rigid body as explicit restraints at the next PIMPLE iteration. External alternatives exist that expand the native capabilities of OpenFOAM in handling dynamic meshes and FSI problems, but its usage is kept out of the scope of the present thesis. One example is the foam-FSI library by David Blom [60], only available for the *foam-extend* version. It includes highly coupled algorithms like Aitken under-relaxation and IQN-ILS. Mesh motion solver is based on the efficient radial basis function (RBF) interpolation method.

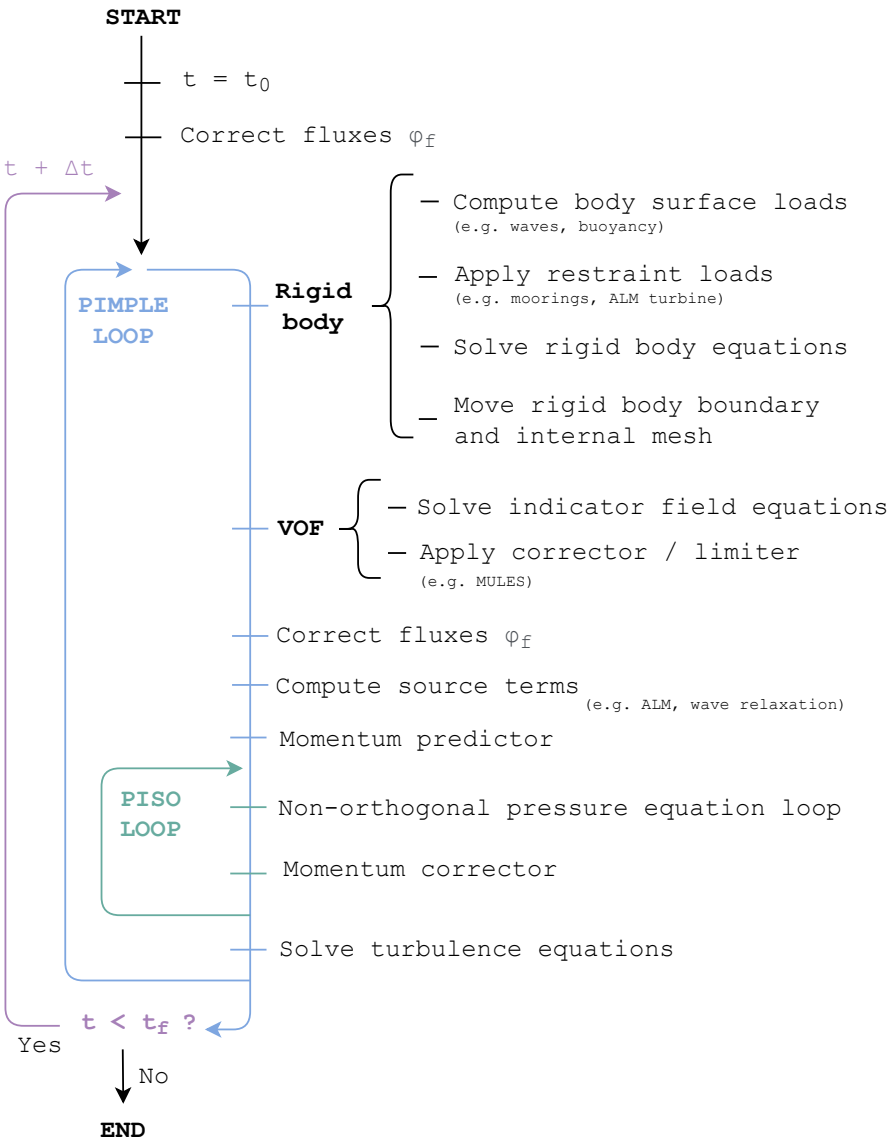


Figure 2.7: Diagram of the PIMPLE algorithm as used in the present work.

3

Literature review

This chapter presents the results of a literature study devoted to floating offshore wind turbines. First, their aerodynamic behaviour is explored in section 3.1, followed by a revision of the main modeling techniques currently used for the simulation of FOWTs in section 3.2. The chapter concludes in section 3.3 with a review of the main projects and campaigns dedicated to the modeling of FOWTs.

3.1. Aerodynamics of FOWTs

The converged design of modern wind turbines seems to imply that there is a deep understanding of the underlying aerodynamic phenomena. But far from true, the aerodynamics of wind turbines is a very complex topic far from being closed. The adoption of simple tools to model rotors straightforwardly can make the engineer forget about the intricate flow phenomena driving the turbine. This complexity is further increased when taking into consideration the extra six degrees of freedom of FOWTs. A review of the aerodynamics of conventional HAWTs and the challenges introduced by the floating conditions is presented in this section.

3.1.1. General principles of HAWTs

Trying to explain the working principle of HAWTs by accounting for all the different aerodynamic phenomena occurring during their operation rapidly becomes overwhelming. A more adequate approach is based on simplifying assumptions, but nonetheless it serves to unfold the underlying nature of power extraction from wind. The latter, known as actuator model, serves as the backbone for a wide range of wind turbine aerodynamics modeling tools, see [10] and [61] for a more in-depth explanation. The grace of this model is that it assumes no specific device for the energy extraction process, and thus can be generalised for virtually any turbine design.

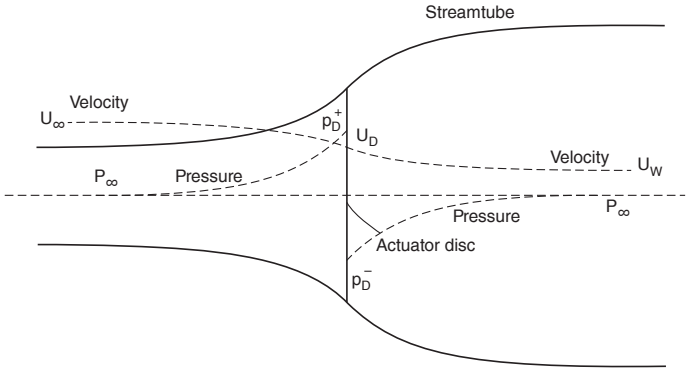


Figure 3.1: Energy extracting actuator disc and streamtube. Reproduced from [10].

Starting from the fact that the power from a wind turbine comes from the kinetic energy of the air that passes through it, this air will see a gradual velocity reduction compared to free-stream. The force originating such deceleration is a step pressure jump across the rotor plane. Assuming that the mass of air that passes through the rotor remains separated from the free-stream (no flow across the boundary) and neglecting compressibility and viscosity effects, then some bold general conclusions regarding the axial flowfield can be stated by applying basic conservation principles to the expanding streamtube:

$$U_D = (1-a)\ U_\infty \ ,\ U_W = (1-2\,a)\ U_\infty \tag{3.1}$$

Where U_∞, U_D and U_W are the freestream, disk, and far wake wind speeds, and a is referred to as the axial induction factor. The above expression shows that half of the axial speed loss takes place upstream of the rotor. This fact can be used to evaluate the load acting on the rotor and the power extracted from the wind, which are usually presented as non-dimensional coefficients:

$$C_T = \frac{\text{Thrust}}{\frac{1}{2}\rho\, {U_\infty}^2 S} = 4\,a\ (1-a) \quad , \quad C_P = \frac{\text{Power}}{\frac{1}{2}\rho\, {U_\infty}^3 S} = 4\,a\ (1-a)^2 \tag{3.2}$$

The maximum achievable value for C_P is $16/27 \sim 0.593$ for $a = 1/3$ and it is known as the Betz limit. This is a daring statement, since until now no specific design has been considered for the turbine. This limit comes from the compromise between kinetic energy extracted from the wind and massflow passing through the rotor. If we could extract all the energy from the wind, this would result in zero velocity and thus no net flow across the turbine.

By assuming an infinite number of blades, the wake can be described by a vortex cylinder model made by a succession of shed tip vortices resulting from a discontinuity in the radial circulation. Following Helmholt's second theorem, a vortex filament cannot end in a fluid, thus a root vortex must exist connected to the vortex cylinder by radial segments of bound circulation.

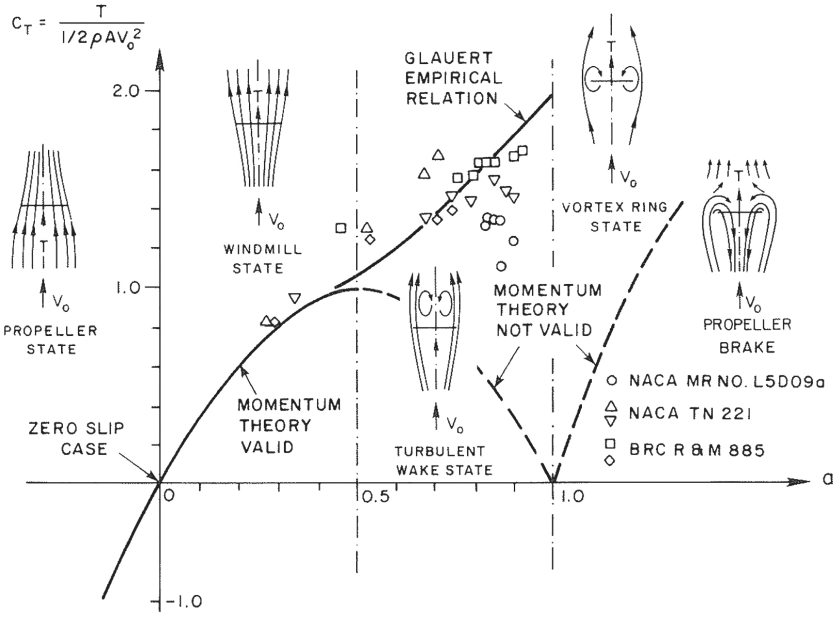


Figure 3.2: Thrust coefficient (C_T) as a function of average induction at the rotor plane (a). Reproduced from [62].

Variations of the above model exist to account for rotating wakes or yawed and tilted rotors, such as Glauert's momentum theory, which is however restricted to small angles and steady conditions. Both the effects of viscosity and unsteadiness can cause these methods to deviate further from nature, needing corrections (dynamic inflow and stall models). Momentum-based methods also struggle in heavily loaded rotors, where the induction factor is greater than $1/2$ (high tip-speed ratios). Even though empirical corrections exist (see Figure 3.2), one must be aware that wake stability is compromised by flow reversal and thus the assumptions made become invalid and higher fidelity modeling is needed. The wake characterisation in such conditions is further explored in the next section.

The presented method also fails to capture the effects of the wake on the wind turbine performance and other models need to be used to describe it. When regarding wakes, two distinct regions can be identified (fig. 3.3):

Near wake. It refers to the area just behind the rotor, approximately one rotor diameter downstream [63], and thus participates in the physical process of power extraction. The flowfield in this region is strongly influenced by the local properties of the turbine, including the tower-nacelle assembly, number of blades, and aerodynamic properties. The flow in this region is highly 3D, rotational, heterogeneous, and turbulent. Much of this complicated structure comes from the shed vorticity by the finite number of blades, which can extend up to 2-3 rotor diameters downstream, much more than the root vortex. Wind tunnel experiments [64] have shown that flow reversal can occur in $x/d < 0.4$ even in normal operating conditions. Due to this high level of complexity, high-fidelity tools are needed for a correct assessment of the interaction between the rotor and the near wake.

Far wake. Downwind of the near wake, turbulent convection and diffusion enhance the mixing of the wake with the undisturbed flowfield. The coherent vortex structures break down, and thus the local properties of the rotor do not influence this region. Predicting the far wake is most important for the interaction with other turbines in wind farms since the velocity deficit within the wake can lead to a decrease in the output power. The far wake can undergo slow, unsteady oscillations known as wake meandering, which effect needs to be taken into account for wind farm layout design. Even though high-fidelity tools are needed to capture the turbulent behaviour of the far wake, simpler models exist that predict the wake expansion and velocity deficit.

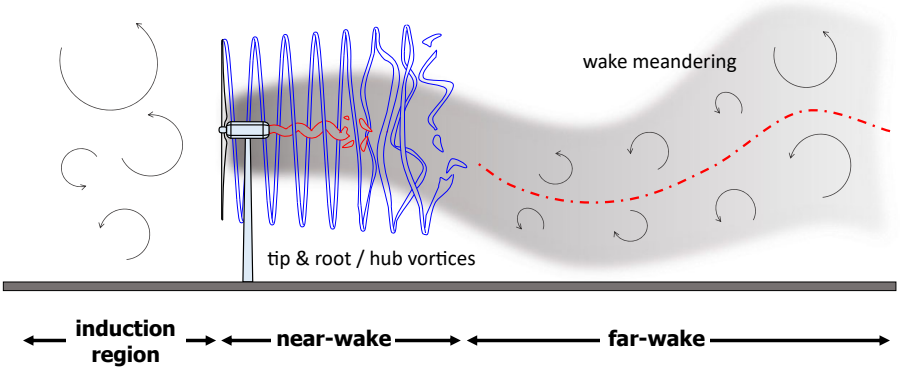


Figure 3.3: Representation of the features of the wake of a HAWT. Reproduced from [65].

3.1.2. The increased complexity of floating turbines

Flow states in floating conditions

The additional motions of a floating turbine introduce new complexities into the aerodynamic environment of the rotor [66], including significantly skewed flows in pitch and yaw, effective wind shear due to angular motions, increased unsteadiness because of the rapid changes in the effective wind component and rotor-wake interactions in intricate turbulent operating states. Even though the presented momentum theory has successfully been applied to turbines with small and steady yaw or pitch angles (tilted rotors), the aerodynamic environment of floating turbines breaks its simplifying assumptions. To characterize the degree of unsteadiness of an aerodynamic system, the reduced frequency from the non-dimensional Navier-Stokes equations is introduced:

$$k = \frac{\omega\, c}{2\ \|\mathbf{W}\|} \tag{3.3}$$

Where ω is the frequency, c the airfoil chord and W the incoming velocity. Per convention, quasi-steady flows are defined for $k \leq 0.05$ while unsteady flow occurs at $k > 0.05$. Spectral analysis of a floating turbine [66] revealed that surge and pitch modes introduce the highest aerodynamic unsteadiness, with barge floaters having the highest fraction of unsteady kinetic energy and TLPs the least. The most unsteady conditions were found in the inboard region of the blade, where the fraction of platform

induced velocity is highest. These effects are aggravated at below-rated conditions, where the tip speed ratio (TSR) is higher:

$$\text{TSR} = \frac{\|\boldsymbol{\Omega}\| \, R}{\|\mathbf{U}_\infty\|} \tag{3.4}$$

Where Ω is the rotor's rotating speed and R its radius. The relevance of surge and pitch motions comes from the fact that they directly affect the streamwise component seen by the rotor. In severe conditions, the velocity induced by these motions can be comparable to the freestream at high TSR, leading thus to scenarios of zero or negative flow across the rotor. This seemingly simple condition ultimately leads to the onset of highly complex flow conditions. These have been studied in detail in the helicopter field for axial flight, but investigation of this flow phenomenon in the context of FOWTs is relatively new. One of the first investigations was carried out by Sebastian [66], who pointed out that pitch and surge motions in combination with high TSR could result in the rotor cycling back and forth within these four states (Figure 3.4):

Windmill state. The turbine extracts energy from the wind, as described by momentum theory. Both the TSR and the reduced frequency stay relatively low.

Turbulent wake state. A turbulent region starts to develop in the wake due to both higher TSR and blade-wake interactions. This state is found to occur for $0.5 > a > 1$, and engineering corrections such as Glauert's exist to account for this effect.

Vortex ring state. For even higher TSR, a toroidal recirculation flow path normal to the rotor disc is developed [66]. The flow becomes unsteady and aperiodic, with successive forward and backward velocities across the rotor plane. Thrust and power become highly related to the energy dissipated in the toroidal vortex ring, leading to high fluctuations and increased blade fatigue.

Propeller state. At some value of TSR, the mean relative speed will become negative and the wake will reverse, leading to negative values of angle of attack and thrust. The turbine will start imparting energy into the flow, acting as a propeller.

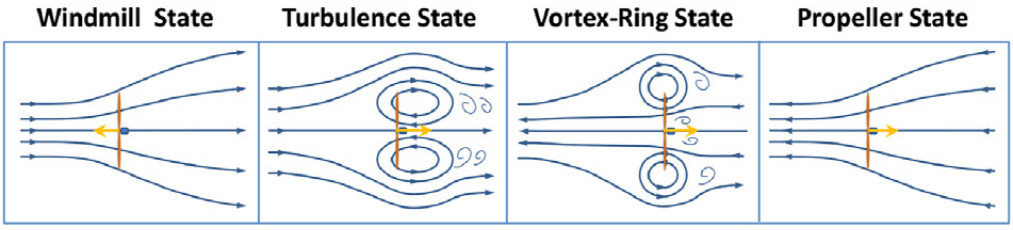


Figure 3.4: FOWT working states. Reproduced from [67].

These states have been identified for both surging and pitching motions, although unsteady and asymmetric wakes can be expected for all motions [68]. The transient vortex ring state (VRS) is of particular interest because of the lack of simple modeling tools capable of describing it. However, relatively simple indicators based on momentum theory have successfully been used to predict the occurrence of VRS [67] and validated with CFD simulations [69, 70]. An induction factor higher than one seems to be a good estimator to define the boundaries of occurrence of VRS, just like other criteria such as Piter's or Wolkovitch's. These studies found that VRS appeared even under normal sea and wind states, and was aggravated for higher TSR, motion amplitude and frequency.

Effect of single-DoF motions

The rotor dynamics in surge motion are characterised by large changes in relative wind speed and by the turbine shuttling in and out of its own wake, affecting both rotor performance and wake shape and stability. A very high-fidelity tool (improved detached eddy simulation) was used by Fang et al. to predict the unsteady aerodynamics of a turbine under prescribed surge motion [71], showing indicators of dynamic stall and separation occurring at the largest forward surge velocity.

Oscillation in power and thrust is expected as the turbines move up and downwind, as confirmed by wind tunnel experiments [72]. The highest blade-wake interaction was found at the maximum downwind position, caused by the accumulated tip vortices that could not be convected downstream because of

the low relative wind speed in VRS and propeller states. Complex phenomena occur in the near wake, such as vortex merging and breakdown, wake expansion and meandering [73] and oscillations of the vortex gaps (accordion shape).

Similar studies [74, 75] reinforce these conclusions, agreeing that blade-wake interactions are most likely to occur at below-rated conditions (high TSR) and that these interactions increase with the amplitude and frequency of the surge motion. In the end, the wake-blade interaction provides extra kinetic energy to the rotor resulting in a higher mean power coefficient, at a cost of larger oscillations (instantaneous power coefficients can exceed the Betz limit [76]) and fatigue.

Pitch motion is also responsible for highly unsteady flow conditions. As for surge, the back and forth motion introduces high platform induced speeds in the streamwise direction, resulting in the onset of TWS and VRS especially for high TSR [77, 78]. The cause is again the clustering of tip vortices behind the rotor when the turbine is pitching upward and the convection speed is low. A memory effect is produced in which the rotor performance is influenced by the wake lying within two rotor diameters downstream [66].

In normal sea-wind conditions, the turbine is not expected to shuttle in and out of the wake as strongly as in surge motion [73], nonetheless strong blade-wake interaction effects are expected. Again, a slight increase in mean power is observed under pitch motions. Very large oscillations in both power and thrust are found as the amplitude and frequency of the pitch motion increases [79, 80]. Wind tunnel experiments [81] revealed that pitch motions resulted in a skewed streamwise component and a larger vertical component of the wake. Pitching also introduces a platform pitch induced (PPI) wind shear, varying both with blade position and time. The PPI is proportional to the turbine height and dominant for high TSR, and it has been found to induce more severe and unsteady loads than typical wind shear [82].

The other DoFs are not as relevant as surge and pitch, but are still worth considering. Yaw, for instance, results in a smaller swept area by the rotor, hence a coupled yaw-pitch motion will reduce the amplitude of power and thrust oscillations compared to pure pitch [79]. However, the tower's interaction with the rotor and its wake is strongest in such yaw motions compared with other DoFs [83]. Regarding heave, wind tunnel experiments showed a wake pattern very similar to those of fixed turbines, with the wake slightly deflected up- and downwards when the turbine was moving in the opposite directions [84]. Very large heave amplitudes could lead to a change in the incoming flow velocity due to the atmospheric wind shear, whereas high heaving frequencies would result in induced velocities in the span-wise direction. Sway and roll motions lead to unstable wake structures but had reduced effect on power and thrust [72].

All in all, the extra degrees of freedom of FOWTs result in more unsteady, skewed, and complex flow states accompanied by blade-wake interactions, difficult to predict with conventional momentum methods. These effects are expected to dominate the aerodynamics of floating turbines as the amplitude and frequency of the oscillations increase, especially in below-rated conditions (high TSR). In such cases, high-fidelity tools are expected to be needed to correctly assess FOWT aerodynamics.

3.2. Simulation methods for FOWTs

Depending on the desired level of accuracy, different tools exist to predict the behaviour of floating wind turbines. In general, different equations drive the various individual components of FOWTs and thus a wide spectrum of simulation tools is available. This section will present a brief overview of the main modeling techniques along with their strengths and weaknesses when applied in floating conditions.

3.2.1. Turbine

Momentum-based methods

The blade-element/momentum theory (BEMT) is one of the most widespread engineering tools for rotorcraft and wind turbines in the low-mid fidelity range. It starts from the same assumptions as momentum theory (see section 3.1.1), but now it specifies that aerodynamic lift and drag are the driving forces causing a change in momentum of the air passing through the rotor. These aerodynamic forces acting on each blade element are obtained by means of 2D airfoil polars, given as a function of the local angle of attack and Reynolds number, which in turn are determined by the incident velocity at the

blade element.

Because of the many assumptions BEMT is built on, engineering corrections are often needed to extend the application range of the method. For instance, in heavily loaded rotors (high a) flow reversal and breakdown into turbulence start to occur (see turbulent wake state in section 3.1.2) and the modeling of the flowfield through the turbine requires corrections based on empirical analyses, being Glauert's a simple but still acceptable correction. Similar corrections need to be applied to account for skewed flows or yawed rotors, 3D effects, centrifugal pumping, tip and root losses due to vortex shedding (Prandtl's loss factor) or radial variations in the induced velocity [77].

BEMT methods can be extended to account for unsteady effects, which introduce a time lag between a change in inflow conditions and the resulting loads as vorticity is convected downstream. The two main approaches for this purpose are dynamic inflow models and acceleration potential methods or generalised dynamic wake (GDW). These methods are often coupled with dynamic stall models such as Leishman-Beddoes', which account for unsteady airfoil aerodynamics and the resulting hysteresis cycles in the 2D aerodynamic polars [10].

Because of the large amount of corrections upon which modern BEMT solvers are constructed, they must be carefully validated before being used in new conditions. For the floating turbine scenario, BEMT methods showed overall good agreement (below 10% discrepancy) with CFD simulations for small surge (1-4 m) and pitch (1-2º) amplitudes, but these differences grew with the TSR and motion amplitude/frequency with up to 24% discrepancy for the 4º pitch case [74, 77], which is, in fact, a conservative value of pitch. In general, the great deal of assumptions and corrections of BEMT make it unable to correctly model the complex flow states and blade-wake interactions expected in FOWTs caused by the motion in the extra DoFs [16]. What is more, BEMT can only provide quantitative data for the performance indicators of the turbine, mainly induction factors and loads. Information on the flow regime or wake structure can only be obtained in a qualitative manner [67].

Nonetheless, BEMT is still the preferred option among many wind turbine simulation tools, especially those coupling aerodynamics with hydrodynamics, structural mechanics and control systems such as FAST, HAWC2, SIMA, Bladed or SIMPACK [85], being the aerodynamic tool of choice for all the participants in the OC3-OC5 campaigns [17, 86]. This fact is due to the relative ease to prepare, run and couple BEMT tools, allowing for simulations of hundreds of different design loadcases needed to fulfil the standard requirements and certification guidelines [16]. For this reason, it is expected that BEMT-based methods will still dominate the realm of low-mid fidelity tools for FOWTs. Dynamic inflow models have already been developed that can predict the turbine performance in large surge motions when compared to vortex methods while maintaining computational efficiency [87], although they are still far from being integrated into wind turbine simulation programs. If we wish to extend these methods to a wider variety of conditions then the underlying aerodynamic phenomena of FOWTs must be further understood, either via experimental campaigns or high-fidelity tools so that new corrections for BEMT can be developed and properly validated.

Potential flow methods

From the governing equations for an inviscid, incompressible and irrotational flow it is found that the curl of the velocity field **u** is zero, and thus it should come from a potential function φ:

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} = 0 \;\rightarrow\; \mathbf{u} = \nabla\varphi \tag{3.5}$$

And the continuity equation becomes:

$$\nabla \cdot \mathbf{u} = 0 \;\rightarrow\; \nabla^2\varphi = 0 \tag{3.6}$$

Where the latter is known as the Laplace equation, whose linear nature allows for complex flow patterns to be constructed from elemental flow functions. Vortex or wake methods take advantage of this fact and represent the trailing and shed vorticity of the rotor with discrete elements (usually sources or vortex), which range from flat panels, curved filaments, particles, rings, etc. Because wind turbines generate bound circulation at the blades and release vorticity into the wake (Helmholtz's second theorem), vortex methods seem to be naturally suited for the simulation of wind turbine aerodynamics [88]. A grid-free or Lagrangian approach is the most common method to calculate the flowfield from the vorticity through the Biot-Savart law.

Different methods exist depending on how the lifting bodies are represented, such as lifting line, lifting surface, panel method or vortex lattice. The simplest representation is the lifting line method, where the blade is represented by a bound-vortex filament with trailing and shed vortices, which depend on the spatial and temporal variation of the bound circulation, respectively. The bound circulation along the span is computed through the Kutta–Joukowski theorem from aerodynamic loads, which in turn are obtained from 2D polars just as in BEMT.

Different methods exist based on whether the geometry of the wake is prescribed or the vorticity equation is used to compute the convection and stretching of the vortex structures (free-wake methods), which ultimately models the tip vortices roll-up. For the latter case, the induction between different vortexes must be computed, thus increasing the computation time up to two orders of magnitude compared with fixed wake methods [89]. Hybrid methods also exist that use lower fidelity tools to approximate the shape of the wake. Vortex and panel methods are included in several popular simulation software for wind turbines, such as openFAST (via the OLAF submodule), PUMA, AWSM, omniVor or FUNAERO to name a few.

Overall, free vortex methods (FVM) have proved to provide good results for FOWTs, being able to predict the aerodynamic performance of the rotor and characterise the wake. Because no specific assumption is made regarding the dynamics of the turbine, FVM can be easily coupled with the extra DoFs of FOWTs. This was first proven by Sebastian [66] who found that FVM is indeed able to predict TWS and VRS. The same conclusion was reached in [78]. The IEA Wind-15MW under pitch and surge was extensively analysed by Ramos et al. [90] using the FVM showing the great potential and versatility of vortex methods compared to BEMT. A similar conclusion is found by Xu et al. [91] and Farrugia et al. [92], where the method showed good agreement with experimental data from various test FOWTs. Similarly, FVM is validated by Fang et al. [71] against high-fidelity CFD simulations and actuator line models by Corniglion et al. [93] and Sarmast et al. [94]. Because in FVM the number of interactions between vortex elements increases as the wake is generated, vortex ring models have proven to reduce computational cost while maintaining accuracy [88].

Because vortex methods inherently model the rotor wake, they can be readily applied to a wider range of conditions than BEMT without the need for engineering corrections. Thanks to the Lagrangian approach upon which they are based, vortex methods do not suffer from numerical diffusion like other Euler approaches (e.g. CFD) and thus the wake geometry is longer conserved. This on the other hand presents a challenge when trying to incorporate viscosity models into vortex methods. Also, the quality of the simulation deteriorates as the vortex elements convect downstream, ultimately leading to instabilities. Vortex methods also struggle to represent cases where other elements have to be included (nacelle, tower, ground, etc.) or in cases where the flow is not driven by the shed vorticity but rather by viscosity and turbulence (recall the inviscid, incompressible and irrotational assumptions of potential flow). In general, vortex methods are an intermediate step between BEMT and CFD in terms of fidelity and computational cost for the simulation of FOWTs. They are much simpler to set-up, run and post-process than CFD, but still too expensive to be used for certification purposes.

Blade-resolved CFD

Blade-resolved CFD represents the highest-fidelity tool available to simulate floating wind turbines. No assumption is made a priori regarding the dynamics of the flowfield except the selected turbulence model, thus being immediately applicable to FOWTs with any additional geometric feature (e.g. tower or hub-nacelle assembly). A common approach to studying the aerodynamics of FOWTs through CFD is to prescribe the turbine's motions, corresponding to those an FOWT would experience due to environmental loads. Kim and Tram performed RANS SST simulations on pitching [77] and surging [77] turbines with prescribed harmonic motions. They found complex flow phenomena occurring in the near wake region that could not be captured by BEMT methods. Blade-resolved CFD was also compared against BEMT for pitching motions by Ortolani et al. [95], showing a 20% underprediction in the power coefficient for pitch amplitudes of 4º but overall good agreement when the turbine was fixed.

A similar setup was used by Lienard et al. [73] and Wu et al. [80] to simulate harmonic motions in surge, pitch and yaw, demonstrating that strong interactions (vortex pairing, wake meandering and instabilities) take place between the blades and the wake that cannot be modelled using FVM. CFD has also been used by Kyle et al. [69] to confirm the occurrence of VRS and TWS in FOWTs and

by Chen et al. [79] in coupled DOF motions. Fang et al. [71] carried out a detailed analysis of the surge motion is carried out using an improved delayed detached eddy simulation (IDDES), proving that rotor-wake interaction and stall phenomena are relevant in surge motions.

Blade-resolved CFD simulations do not only provide information regarding the flowfield characteristics but also from local phenomena such as boundary layer evolution or blade-wake interactions. This makes the outcome of well-designed CFD simulations almost equivalent to experimental campaigns, thus being an essential component of fundamental FOWTs research. They have the potential to shed more light on the underlying aerodynamic phenomena of floating turbines, and can also be used for tuning and validating lower-fidelity tools. But because the geometry of the blades needs to be represented and the boundary layer captured, very fine meshes are required near the wall, heavily increasing computational costs. For this reason, blade-resolved CFD is reserved for very specific load situations.

Actuator methods

Thanks to the reduced computational costs, actuator methods have been established as a popular CFD choice for floating wind devices. These models can easily be modified to account for the dynamic motion responses of the floating platform, resulting, for instance, in the unsteady actuator line model (UALM) [96]. For small surge and pitch amplitudes, UALM was compared with BEMT by Apsley et al. [97] showing that in such conditions, a quasi-steady model suffices to represent the main aerodynamic performance parameters. The slight differences between the two methods do not justify the higher computation cost of UALM in near-rated wind speeds and small motions. However, the results obtained with UALM proved useful for conceiving new corrections that can be applied to BEMT in floating conditions.

Li et al. validated in [98] an UALM against blade-resolved CFD for a fixed turbine and then used it to model an FOWT under pitch and surge motions, concluding that the pitch motion introduces complex wake flows that cannot be modelled with conventional BEMT. A comparison between a lifting line FVM and ALM is presented by Corniglion et al. [93], showing overall good agreement and a limited ability to predict aerodynamic loads in large surge movements with dynamic stall. It is still not clear where the use of UALM should be preferred over vortex methods since the former has shown to be at least two orders of magnitude more computationally demanding. The investigation by Spyropoulos et al. [46] points out that the main advantage of UALM over FVM is the enhanced capability to account for the wake turbulent convection and its interaction with surrounding bodies.

Johlas et al. [26] used an UALM-LES simulation to formulate FOWTs wake models that can be later used in lower fidelity tools. A similar study of the downstream wake characteristics of a FOWT under different metocean conditions is successfully presented in [99]. The reduced computation cost of RANS-UALM makes it a good candidate for full FOWT simulations, which are further reviewed in section 3.2.3. The actuator disk model (ADM), on the other hand, is usually reserved for far-wake computations, making it interesting for the simulation of floating wind farms. Xiao et al. [100] used LES in combination with ADM to successfully predict the interactions between different floating turbines in a farm layout, using considerably less computational resources than blade-resolved CFD simulations.

In his MSc thesis [101], Negroni validated an UALM in OpenFOAM against the UNAFLOW experimental campaign, which used a 1:75 scaled version of the DTU 10-MW turbine. The code, originally developed by Schito at POLIMI [102], was modified by Negroni to accommodate for prescribed motions in pitch and surge. He compared not only induction factors but also wake deficit profiles with both experimental data and lower fidelity tools. Compared to experimental results, the UALM underpredicted thrust and, to a lesser extent, torque. When it comes to the lower fidelity tools (BEMT and FVM), all produced similar values of mean thrust. Compared to the other two, UALM underpredicted thrust amplitude and overpredicted mean torque.

Sala focused her MSc thesis [103] on the investigation of FOWTs under prescribed surge motion by means of UALM. She modified the OpenFOAM library *turbinesFoam* so that a fluctuating, unsteady value of thrust coefficient could be imposed. Similarly, a prescribed surge motion could be assigned to the actuator lines. She compared the results with ADM and a lower fidelity dynamic inflow model [87]. She found the surge motion to cause oscillations in the blades' loading, which increased with surge frequency. The phase shift between thrust and platform motion was close to 90º. No evidence of a vortex ring state was found in any of her simulations.

3.2.2. Platform

Panel methods

Potential methods for hydrodynamic applications are based on the same inviscid, irrotational, and incompressible assumptions seen for aerodynamic applications in section 3.2.1. The velocity field can be expressed as the gradient of a potential scalar function, which in turn is driven by the Laplace equation. The added complexity in hydrodynamics comes from the need to satisfy extra boundary conditions [28]:

Impermeability condition. No flow can go through the surface of the floater, thus the normal component on the velocity field at the body surface must equal the floater's velocity. The same applies to the sea-bottom, assumed flat.

Far-field condition. The radiation potential must tend to zero far away from the body.

Free-surface kinematic condition. Fluid particles at the free surface remain at the free surface.

Free-surface dynamic condition. Pressure at the free surface must equal atmospheric.

The latter two conditions are non-linear in their most general form, thus complicating the analysis and invalidating the superposition principle. A linearization process is carried out through a perturbation expansion around the mean sea level and body equilibrium position. If only terms proportional to wave amplitude (or steepness) are retained, the first-order solution is obtained. This assumes linear Airy waves and enables the division of the potential function into radiation and diffraction, which can now be found independently. This boundary value problem is solved using Green's function for the radiation and diffraction potentials.

Typically, panel methods (also known as *Boundary Element Methods* or BEM) are the preferred numerical technique to solve this task. The geometry of the body is discretized into surface elements where a distribution of sources, vortices, or doublets with unknown intensities is placed [20]. This leads to an algebraic system of equations from which the velocity potential is obtained. Finally, forces acting on the body are computed from the integrated pressure along the surface, which in turn is given by Bernoulli's equation.

To capture the effect of drag forces, potential methods are usually coupled with Morison's equation (equation 2.21), a semi-empirical engineering model suited for slender bodies ($D < 0.05\lambda$) since wave diffraction is ignored. Morison's equation in its full version accounts for Froude-Krylov, inertia (added mass), and drag loads as covered in section 2.1.2. For its implementation in numerical solvers, the equation is usually presented for an infinitesimal strip ds of a cylindrical body [104]. The drag and inertia coefficients present in Morison's equations are chosen empirically based on the Reynolds and Keulegen-Carpenter (KC) numbers, even though a frequency-dependent drag coefficient might be preferred for a wider range of environmental conditions [20].

Potential flow theory and Morison's equations were the reigning tools in the OC campaigns due to their versatility and relatively low computation requirements, and they are also the main modeling tools of mid-fidelity FOWT codes. The OC4 [86] proved that RAOs are suitable for comparing the system response in various wave conditions between different codes. Codes based on Morison's equation provided similar results as potential solvers, thus showing that semi-submersible floaters may not need radiation/diffraction modeling. These codes were compared against experimental data in the OC5 [17], showing a general underprediction (20%) of the loads at the pitch and surge natural frequencies, with second-order potential methods being the closest to the experiment. Since these natural frequencies lie above the wave excitation range, they must be excited by a non-linear force.

A deeper study on this phenomenon was carried out during the OC6 [105], focusing on the low-frequency region. All participants correctly predicted motions and loads in the wave-excitation region but underestimated the response in the low-frequency region. It is suspected that certain assumptions about potential models might be violated with the smaller dimensions and larger motions that characterize FOWTs. Unsteady drag may not be quadratic in nature but be driven by a more complicated mechanism where viscous effects alter the wave radiation properties of the floating platform. To gain an understanding of the underlying hydrodynamic phenomena occurring in the low-frequency region, high-fidelity CFD simulations were carried out in subsequent phases of the OC6 [106].

Computational fluid dynamics
CFD simulations have been extensively used during the OC6 campaign for validation purposes of a simplified DeepCwind semi-submersible platform. In phase Ia, the feasibility of using high-fidelity CFD simulations to tune lower fidelity tools was investigated through free-decay motion tests [107]. Because of the interest in capturing the hydrodynamic damping coefficients, the mesh was refined near the body surface to accurately capture shear effects. A thorough uncertainty analysis revealed that turbulence models have a secondary effect compared to discretization uncertainty, with grid resolution being the dominant term. Several participants carried out the simulation (mainly in OpenFOAM and STAR-CCM+) showing meaningful estimates of the motion periods and damping ratios that were in agreement with experimental data.

Since the decay motions are driven by viscous damping rather than wave diffraction, the accuracy of potential methods is determined by the tuning of the linear and quadratic drag coefficients from CFD simulations. When properly tuned, these mid-fidelity methods can produce results consistent with the experiments, with computation times up to 5 orders of magnitude lower than CFD. Similar conclusions are stated by Rivera et al. [108], who simulated the same floater using both CFD and potential methods, showing that in severe waves, the potential method requires adjustments to the drag coefficients based on the CFD results. Wiley et al. [109] carried out CFD simulations of a semi-submersible platform under bi-chromatic forced oscillations over a range of KC numbers. Added mass, linear radiation damping, and quadratic drag coefficients were extracted from the results. The latter had the largest effect on improving potential methods' predictions in the low-frequency range. In this study, a $k-\omega$ SST turbulence model was used that treated the boundary layer as laminar and the wake as turbulent.

In phase Ib, special attention was paid to the load prediction in the low-frequency range [106, 110]. CFD simulations were performed with the fixed platform under the excitation of second-order bichromatic waves. While it was found that the sum-frequency loads had negligible effects on the floater motion, the difference-frequency component usually fell within the range of natural surge-pitch frequencies and thus required special attention. A good general agreement was found between CFD results and experimental data, even though the large uncertainties of CFD limit this confidence. Second-order potential methods were used for comparison, showing a severe underprediction of the difference-frequency surge force (by a factor of two) and pitch moment ($\sim 40\%$), likely caused by additional viscous excitation on the structure. During this campaign, it was found that numerically resolving the shear layer (non-slip condition) might not be critical for small waves. Since standard turbulence models were found to be overly dissipative for the bichromatic-wave problem, no turbulence model was used in the simulations.

Apart from the OC campaigns, CFD hydrodynamic modeling is a popular high-fidelity tool in cases where nonlinear or viscous loads are dominant. Bruinsma et al. [50] used a numerical CFD wave tank to simulate the flow-induced motions of a moored floating wind turbine, which was validated against experimental data. A study of the floater response under increasing wave steepness was performed by Zhou et al. [111], showing that non-linear loads increase with wave amplitude. Wave run-up was observed at large wave steepness, leading to enhanced peaks of the platform motion. Laminar flow was assumed in the simulation because no apparent disparity was found between turbulent and laminar models when the flow was dominated by waves rather than current. Discrepancies were found with potential methods (FAST), which increased with wave steepness. Burmester et al. [112] found that viscous effects and turbulence models did not play a major role in predicting the oscillation period, but they did affect the hydrodynamic damping of the platform. Simulating a scaled-down version of the platform was found to be crucial in order to successfully predict the experimental results.

3.2.3. Coupled simulations
Floating wind turbines are a specific FSI type of problem where not only two but four domains are coupled via structural, hydrodynamic, aerodynamic, and mooring equations. This number further increases in simulations considering the control system, electric components, etc. In certain cases, a monolithic approach can be implemented for the hydro-aerodynamics (e.g., CFD simulation with VOF) and for the mooring-structure (e.g., multi-body or FEM) thus the coupling is only needed between two models (fluid and structure). The latter is not a widespread approach in mid-fidelity tools, which re-use specific simulation tools and combine them to simulate floating turbines. Such tools often include BEMT and

vortex methods for the rotor, panel methods and Morison's equation for the hydrodynamics, and FEM or multi-body methods for the structure and moorings.

FAST by NREL, Bladed by DNV and HAWC2 by TUD are examples of mid-fidelity codes based on this heavily partitioned approach. They can even be coupled to external modules to model a wider range of conditions (e.g. WAMIT, SIMO or RIFLEX), making them a highly flexible option. Moreover, the relatively small computational cost of each model makes this approach suitable for the design stages, where hundreds of loadcases must be taken into account for certification purposes. But even though each of these models has been independently validated, that does not mean that they can be coupled to represent the dynamics of a FOWT without introducing significant errors, as shown during the OC campaigns [17, 105].

High-fidelity models are meant to shed more light on the coupling and interaction phenomena occurring between the different domains. These simulations, however, are very time-consuming due to the complex set-up and high computational costs, making them prohibitive for early design stages. On the other hand, not only can they be used to gain a deeper insight into the hydro-aero coupling mechanisms, but they also have the potential to be used to obtain reference data upon which lower-fidelity tools can be validated and calibrated. These high-fidelity simulations can be carried out using many different approaches and models, which will ultimately determine the level of accuracy and the computational requirements. Depending on the latter two, one should carefully choose which combination of methods is needed to obtain the desired simulation. An overview of high-fidelity coupled simulations of FOWTs is presented in Table 3.1.

3.3. Projects and campaigns

Because of the interdisciplinary dimensionality of floating turbines and the mutual interaction between the different components, fundamental research is intricate. Neither experimental campaigns nor numerical simulations are a piece of cake. This section presents the main organizations involved in the current research on floating wind energy.

3.3.1. IEA Tasks

The guiding organization in floating wind research during the last decades has been IEA Wind, which through its *Technology Collaboration Programme* (TPC) acts as an international vehicle for cooperative research and development projects called *Tasks*. These *Tasks* cover technology development, wind characterization, and social and environmental issues. For the purpose of the present report, *Task 30* is of special interest. Starting as the *Offshore Code Comparison Collaborative* (OC3) project in 2005 within *Task 23*, it extended up to the present time with OC4, OC5, and OC6 to address the need to verify and validate the load predictions of coupled modeling tools for offshore wind design [118]. These tasks have focused on engineering-level tools that consider simultaneous loads from wind and waves.

OC3: 2005-2009

The *Offshore Code Comparison Collaboration* (OC3) Phase IV [53] consisted of the simulation of the NREL 5MW turbine on a floating spar buoy, the OC3-Hywind, in deep water with various wind-sea conditions. Ten participants used different low- and mid-fidelity tools to model the FOWT, basically BEM or GDW for the turbine and potential solvers or Morison's equation for the floater. The structural dynamics of the blades and moorings, as well as the control system, were also taken into account. The results showed that natural frequencies were tightly related to the added mass terms in the hydrodynamic equations, while structural flexibility did not influence the overall motions. Gyroscopic coupling appeared to be the cause of yaw oscillations.

OC4: 2010-2013

The *Offshore Code Comparison Collaboration Continuation* (OC4) Phase II [86] involved the simulation of a semi-submersible design, the DeepCwind, chosen for its increased hydrodynamic complexity. The goal was to verify the modeling tools by comparing the results of simulated responses. Again, several wind-sea loadcases were defined and run by 23 participants from 13 countries with different modeling approaches, all within the mid-fidelity range.

Authors	*Case*	*Methodology*	*Remarks*
Tran, Kim [113]	OC4 DeepCwind + NREL 5-MW turbine under steady wind and regular waves	Monolithic hydro-aero CFD (VOF, $k-\omega$ model, overset mesh) + 6-DOF rigid body + Quasi-steady catenary mooring in STAR-CCM+	Validated against MARIN test data. Comparison with FAST revealed large differences in the dynamic responses
Leble, Barakos [114]	Semi-submersible + 10-MW turbine under steady wind and regular waves	Monolithic hydro-aero CFD (smoothed-particle hydrodynamics) + Rigid MB + Lumped-mass moorings in HMB3 with weak coupling	Full model was not validated, weak coupling showed good convergence
Calderer et al [115]	Simplified cylindrical platform under fully developed turbulent flow-field and JONSWAP waves	Monolithic hydro-aero: far field potential coupled with near field LES (rotor ADM) + Rigid body (pitch-heave only)	No moorings, verified against theoretical solutions
Ren et al [116]	TLP + NREL 5-MW turbine under maximum operation sea	Monolithic hydro-aero CFD (VOF + $k-\varepsilon$ model) + Rigid body (surge, sway and yaw) + Quasi-steady PEM moorings in FLUENT	Validation against experimental set-up
Liu et al [24]	Semi-submersible + 5-MW turbine under large 2nd order Stokes waves and steady wind	Monolithic hydro-aero CFD (THIC + immersed boundary method, rotor by UALM) + Rigid body + Linear mooring in OpenFOAM (RIAM-CMEN)	Validation against experimental data
Cheng et al [25]	OC4 DeepCWind + NREL 5-MW turbine under regular waves and steady wind	Monolithic hydro-aero CFD (VOF + SST, overset mesh, rotor by UALM) + Rigid body + Quasi-steady PEM moorings in OpenFOAM (naoe-FOAM-SJTU)	Validated and verified for wave/wind only cases
Cheng et al [96]	OC3-Hywind + NREL 5-MW turbine under 1st order Stokes waves and steady wind	Same as in [25]	In spars, motion responses are driven by aerodynamic loads rather by waves
Johlas et al [26]	OC3-U Maine spar + NREL 5-MW turbine under irregular JONSWAP waves and atmospheric boundary layer	Partitioned hydro-aero (HydroDyn + aero LES, rotor by UALM) + Elastic body + Dynamic mooring in SOWFA and openFAST	No validation, wake-effects limited by mesh resolution
Liu et al [117]	OC4 DeepCWind + NREL 5-MW turbine under regular waves and steady wind	Monolithic hydro-aero CFD (VOF + SST model, sliding mesh) + Rigid body + Linear mooring in OpenFOAM	Individual model validation and full-model comparison with FAST. Visible tower shadow effect

Table 3.1: Examples of high-fidelity coupled simulations of FOWTs.

OC5: 2014-2018

The *Offshore Code Comparison Collaboration Continuation with Correlation* (OC5) Phase II [17] aimed to solve the inherent weakness of OC3 and OC4 that simulation outputs could only be compared with each other (verification), not against experimental results (validation). For this purpose, the OC5 used measured data from a variety of experimental campaigns carried out with a 1/50-scale model of the DeepCwind at the MARIN wave basin. The system was tested under Froude-scaled wind and wave loads without any formal quality or uncertainty assessments, thus making the comparison with the engineering tools more of a qualitative evaluation.

Again, 21 participants simulated several wind-sea loadcases using various low- and mid-fidelity simulation tools. The participants were given information regarding static equilibrium, free-decay, wave-only and wind-only tests to calibrate their models. Results showed that all participants underpredicted the loads by an average error of up to 21%. The largest underprediction came from low-frequency responses below wave excitation frequency, and thus it was hypothesized that it was caused by a non-linear force.

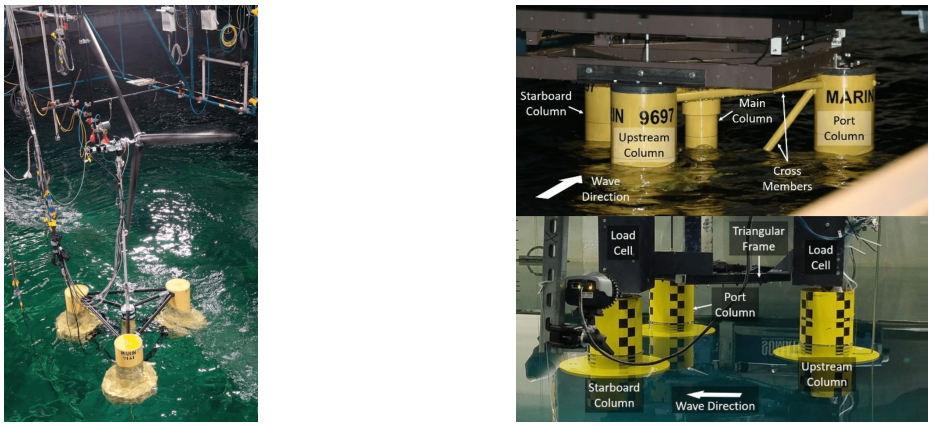


Figure 3.5: (a) OC5 phase II experimental setup at MARIN's offshore basin with both wind and wave condition [17] and (b) comparison between the experimental semi-submersible models for the OC6 phase Ia (up) and Ib (down) [106].

OC6: 2019-2022

The *Offshore Code Comparison Collaboration, Continued with Correlation and unCertainty* (OC6) used new wave-tank validation campaigns, where uncertainty was conscientiously assessed [105]. The floater geometry was simplified, and the turbine was changed to a rigid stout. Again, different participants run various loadcases with mid-fidelity tools, focusing this time on the low-frequency hydrodynamic loading. Again, the floating response at natural frequencies was underpredicted by all participants.

To examine the limitations of these models, higher fidelity tools were used in the OC6 phase Ib [106]. A new wave basin campaign at the University of Maine examined a cylinder in isolation, then a cylinder with heave plates, and finally a three-cylinder layout, all in a fixed configuration [119]. Neither the main column nor cross-members were included. To study the non-linear hydrodynamic force, a second-order bichromatic wave condition was imposed so that CFD simulations would only need to run for a few repeat periods to capture the non-linear difference-frequency loads. Three regular-wave cases were also performed. Eleven participants carried out CFD simulations that were compared to the experimental results. Validation was overall successful within the uncertainty ranges, especially in cases with lower wave heights because of the reduced viscous drag contribution, which is challenging to capture as it requires finer meshes. This proves that CFD tools are mostly consistent with experiments and can be used to tune mid-fidelity models, while providing good-quality data to better understand the underprediction of mid-fidelity tools. However, the high uncertainty of CFD simulations in certain loadcases limits confidence in the conclusion.

In the future, OC6 Phase III will focus on the validation of aerodynamic loading of FOWTs undergoing large motions, while Phase IV will validate methods combining potential flow and hydrodynamic load models for novel FOWT support structures.

3.3.2. INNWIND.EU

From 2012 to 2017, this EU integrated project focused on design methodologies and simulation tools for offshore wind turbines beyond 50m water depths, with the goal of a preliminary evaluation of a 20MW reference turbine. They used mid-fidelity simulation tools, but did not apply aero-hydro coupling. From the aerodynamic side, special attention was given to larger rotors with lower induction and integrated aero-structural simulations of two-bladed rotors. Various concepts for support structures were also developed. Two wave tank campaigns were carried out for a semi-submersible and TLP platform for a scaled model of the 10MW turbine.

3.3.3. LIFES50+

This European Horizon-2020 funded program was led by MARINTEK from 2015 to 2019 with the collaboration of 12 different partners. Its main goal was the development of floater designs for 10MW turbines that could qualify for a Technology Readiness Level of 5. Both industrialization and energy costs were considered in the early stages of the design. Experimental studies were carried out for two floater concepts, and then used to calibrate the mid-fidelity engineering tools. While some of the models produced results in line with the experiments in surge, none was able to predict the aerodynamic performance in pitch, which turned out to be more challenging to model.

3.3.4. UNAFLOW

The *UNsteady Aerodynamics for Floating Wind* (UNAFLOW) is an EU-IRPWIND experimental campaign focused on the unsteady aerodynamics of floating wind turbines [120], concentrating on the advanced aerodynamic modeling of Multi-Megawatt floating rotors. An extensive dataset was produced in a series of wind tunnel experiments with harmonic motions in surge direction, capturing both local airfoil and overall rotor performance variables. Then, three tools of different fidelity (BEM, lifting line, and CFD) were validated against this experimental data.

4

Wave tank implementation using the relaxation zone method

The goal of this chapter is to present a comprehensive method for the simulation of propagating waves in OpenFOAM by means of the relaxation zone method with `waves2Foam`. Its contents should serve as a guide for anyone interested in building a CFD-based numerical wave tank in OpenFOAM. Section 4.1 provides a survey of the available methodologies for numerical wave tanks, followed by a justification of the selected technique. Then, section 4.2 studies the wave propagation phenomena in OpenFOAM by constructing a numerical 2D wave flume and comparing the results with algebraic wave theory.

4.1. CFD-based numerical wave tanks

4.1.1. Motivation

For both fundamental research and engineering design, marine scientists rely on physical wave tanks (PWT) as the main source of experimental data. Those tanks can be thought of as pools with the capability to generate specific wave (and sometimes even wind) conditions by using wave makers and dampers. For the offshore engineering sector, the main objective is to study the response of fixed or floating bodies under loading conditions that resemble those of the open sea.

Note however that this experimental setup is just a recreation of the real marine environment, and hence the limitations of PWTs must be carefully examined before they are used to validate a model. Such limitations include wave reflection from walls and the difficult choice of a scaling parameter for a floating turbine, given that aerodynamic and hydrodynamic phenomena are driven by different non-dimensional numbers (Reynolds and Froude, respectively). Moreover, experimental campaigns in NWTs are restricted to very few facilities worldwide because of their complexity and cost.

The increase in available cheap computational power has raised the popularity of numerical wave tanks (NWTs), which nowadays stand as a meaningful tool in the field of marine engineering. These tanks do not suffer from the PWTs scaling dilemma and thus can be used both for direct comparison with full-scale devices and for validation purposes with PWTs campaigns. NWTs do not rely on the laws of nature to achieve this purpose, but rather on a given set of modeling equations. The degree of fidelity will be highly dependent on the chosen model. For instance, potential (BEM) methods are a popular choice due to their ease of implementation and low cost, allowing them to be used for long simulation timespans. But because they are built upon linear and potential flow assumptions, they miss important features of the flow that are of great importance to FOWTs [17, 105].

The highest fidelity method available is based on CFD —the CFD-based NWTs or CNWTs— which inherently account for all the non-linearities BEM methods miss. On the other hand, this approach requires a higher computational burden compared to potential methods, with the cost being highly dependent on the turbulence modeling approach. Among other requirements, CNWTs must be able to perform multiphase flow simulations, handle FSI problems, and implement numerical wavemakers

(NWM) for wave generation and absorption. All in all, CNWTs have already shown promising results for the simulation of FOWTs under various wave conditions [106, 108, 121]. Nonetheless, the engineer should keep in mind that CNWTs are subjected to the same limitations as any other CFD simulation.

4.1.2. Quality assessment

One of the most challenging aspects of CNWTs is their ability to reliably reproduce wave propagation according to the desired wave theory, given that the overall fidelity that can be achieved is directly related to the quality of the generated wave field. Apart from wave reflection, also present in PWTs, CNWTs suffer from artificial dissipation and dispersion errors, which ultimately distort the solution. Consequently, the quality of a CNWT must be properly assessed before it is used for validation purposes.

But because of the various existing NWM approaches, this quality check is difficult to generalize. In [122], Windt et al. performed an exhaustive comparison of the main NWM techniques with the goal of unifying a set of metrics that can be used to validate almost any CNWT. Not only did they take into account modeling inaccuracies such as spatial and temporal discretization or the sensitivity of solver settings, but also the NWM's ability to properly generate and absorb waves. In the end, they argue that no NWM technique is perfect, and the engineer should take a decision based not only on its accuracy but also its computational cost and availability.

4.1.3. The waves2Foam library

Developed in 2011 at the Technical University of Denmark by Jacobsen [27], *waves2Foam* is an open-source toolbox for the generation and absorption of free-surface waves based on the VOF method presented in section 2.2.2. A template on how to set-up a basic case with *waves2Foam* is presented in Appendix 8.3.

The relaxation zone method

The *waves2Foam* library is developed around the use of explicit relaxation zones (active sponge layers) for wave generation and damping. In those zones, a weighting is performed between the computed solution (i.e., **U** and α) and the indicator field with a target solution. Using an explicit approach:

$$\phi = (1 - \alpha_R)\,\phi_{\text{target}} + \alpha_R\,\phi_{\text{computed}} \tag{4.1}$$

Where $\alpha_R \in [0,1]$ is the relaxation weight and χ_R is a function of the local coordinate within the relaxation zone $\chi_R \in [0,1]$. This function can be defined in various ways (exponential, polynomial, etc.) and even corrected based on the local CFL number [2]. Commonly, two distinct relaxation zones are defined, as shown in Figure 4.1: one for the wave inlet, where the target field is given by a specific wave theory, and one for the wave outlet, which produces an undisturbed wave profile. The latter is used to remove spurious wave reflections that could interfere with the simulation.

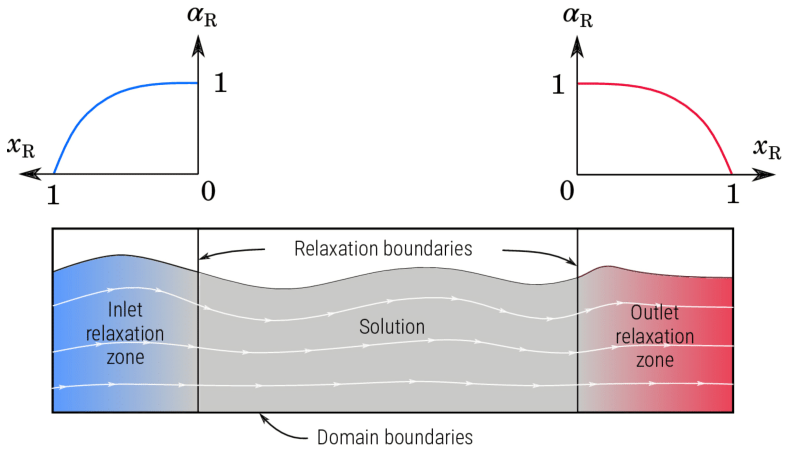


Figure 4.1: Sketch of the relaxation zones and the variation of α_R (χ_R). Reproduced from [123].

With *waves2Foam*, the user can choose between different shapes and weights for the relaxation zones. Rectangular shapes are useful for PWT experiment replication, whereas cylindrical zones are a great

option for open-sea simulations. The toolbox is compatible with a large range of algebraic wave theories, which are explained in detail in the manual [2]. The *waves2Foam* toolbox integrates the solver *waveFoam* to study the interaction and propagation of waves without permeable layers. This solver can be merged with the native *interDyMFoam* to account for moving meshes based on the VOF phase-fraction approach to capture the free-surface; the resulting coupled solver is known as *waveDyMFoam*. The *waves2Foam* library has been successfully validated for several cases, including wave loads in coastal bridges, breaking waves on a beach profile, and floating wave energy converters modeling [2].

The library also comes with utilities tailored for free-surface flows, such as surface elevation sampling. The user can define wave gauges that monitor the numerical surface elevation ζ relative to the initial mean sea level d at the specified positions. Surface elevation is then found through numerical integration of the water fraction α along the vertical line $z_0 - z_1$:

$$\zeta = \int_{z_0}^{z_1} \alpha \, dz \; - d \tag{4.2}$$

Mooring restraints

Mooring lines provide the required forces to restrain the system under the influence of environmental loads. For TLPs, they provide the principal means of stability. Its proper modeling is crucial to determining the restoring loads and assessing whether the lines will suffer from fatigue or yield. The *waves2Foam* library includes a quasi-steady mooring restraint model, meaning that static equilibrium is assumed at each time-step and dynamic effects such as inertia are ignored. The method is based on the closed algebraic solution of the lines suspended in a fluid by applying Hooke's law to a catenary shape [20].

Different formulations exist for the freely hanging portion of the line and the one resting on the seabed [124]. In combination, these analytic expressions can be used to obtain both the shape of the line and the restoring tension given by the anchor and fairlead positions. This method does provide individual mooring forces and accounts for buoyancy forces, seabed friction, and elastic stretching but is limited to lines with uniform properties and horizontal seabeds.

Justification

The choice of *waves2Foam* as the main NWM technique is based on the combination of accuracy, computational cost, and availability. In the work of Windt et al. [122], relaxation zone methods were found to provide robust performance in a wide range of sea states and great flexibility when it comes to accuracy control. As of 2022, the *waves2Foam* library is actively maintained and updated by *Deltares* which makes it a popular choice among the OpenFOAM community, leading to a large amount of available literature —from scientific articles to web posts— devoted to it.

On the counterpart however, the relaxation zone method in *waves2Foam* lacks the computational efficiency of its competitors (e.g. *olaFLOW* and *IHFoam*) due to the larger cell count that relaxation zones entail. To overcome this limitation, Jacobsen introduced the generating-absorbing boundary conditions (GABC) method to *waves2Foam* and validated it for heaving cylinders subjected to a variety of loading conditions [125]. Even though this method has the potential to reduce the computational overhead of FOWT simulations while limiting the effect of reflected waves, it will not be considered for the present project given its recent development and the lack of documentation.

Alternatively, there also exist wave generation techniques that are natively available in OpenFOAM and require no external libraries (e.g. *waves2Foam*, *olaFlow* or *IHFoam*). Recently, a new *fvModel* called `waveForcing` was included in the OpenFOAM-*dev* distribution that supported wave generation with great flexibility. It works by applying a forcing to the liquid phase-fraction field and to all components of the vector field, relaxing them toward those calculated from the current wave distribution. Waves do not need to be introduced at the inlet and can be generated in any direction relative to the mean flow. However, due to its very recent development and lack of compatibility with the selected OpenFOAM version (v2012), the `waveForcing` utility will not be considered for the current project.

4.2. Numerical wave flume in OpenFOAM

The high-fidelity simulation of FOWT semi-submersible platforms by means of CNWT has been successfully addressed in a large number of recent works [106, 108, 50, 105, 107, 121, 126]. The author is aware of this active field of research, and, trying not to re-invent the wheel, the present chapter does not focus on a thorough CNWT validation. Instead, this section presents a step-by-step implementation of a CNWT in OpenFOAM, to be seen as the first stage towards the full FOWT simulation. A 2D wave flume will be used as a case study to keep the set-up simple enough but representative of the wave propagation phenomena. The goal is to address the capability of the implemented CNWT in generating and absorbing water waves.

4.2.1. Case description

The objective of this first case is to get acquainted with multiphase flow modeling and wave generation by implementing a 2D wave flume across which monochromatic, second-order Stokes waves propagate. This step is fundamental as it will serve as the backbone for future CNWT simulations. Second-order wave theory is chosen to correct for the finite water depth and wave steepness effects discussed in section 2.1.2. Still, the values of water depth and wave steepness will be chosen such that the second- to first-order wave amplitude (given by equation 2.16) is kept relatively small, making the waves nearly linear. This should avoid non-linear interactions between harmonics, which could potentially arise when generating higher-order waves.

Still, the simulation of second-order waves poses an often underrated challenge. The engineer must be aware of the assumptions upon which this wave theory is built, which limit the range of values for wave steepness $\varepsilon = H/\lambda$ and relative water depth h/λ that can be accurately modelled. If not chosen properly, wave crests become higher and shorter, whereas troughs grow longer and flatter as a result of the higher-order terms [34]. Because wave simulations are very sensitive to spatial discretization in the free-surface normal direction, given as the number of cells per wave height (CPWH), very small steepness values are not feasible due to the small grid size required. Likewise, deep water simulations increase the overall cell count.

In a compromise between modeling accuracy and computational cost, a steepness value $\varepsilon = 0.5\%$ and a relative water depth $h/\lambda = 40\%$ are chosen for the present simulation, fulfilling all three criteria from section 2.1.2. In the end, the resulting second to first-order wave amplitude from equation 2.16 is kept at $\hat{\eta}^{2nd}/\hat{\eta}^{1st} \sim 0.83\,\%$. A time period $T = 8\,s$ representative for wind waves is chosen, resulting in a wavelength $\lambda \sim 98.62\,m$ according to the dispersion relation from equation 2.13.

Yet, the fact that second-order theory does not satisfy the complete non-linear boundary condition leads to the generation of higher harmonics, whose interaction causes energy transfer that can be identified as beat lengths [27]. This adds up to the errors arising from the discretization approach that can lead to artificial diffusion and dispersion phenomena. The different numerical parameters involved in the simulation, such as time-step, meshing approach, grid size, and relaxation zone layout (wave generation and absorption lengths, L_g and L_a) must be carefully examined before the simulation setup is used for scientific purposes. As this would be a time consuming task, reference values and advice from the literature will be used instead.

4.2.2. Literature review

As stated at the beginning of this chapter, aiming for validation or convergence analyses is beyond the scope of the present project. Instead, the work of other researchers in the field of CNWTs has been investigated to gain insight on what numerical parameters and strategies work best for wave propagation applications. Some of the reviewed publications aimed at high fidelity simulations and thus sought a numerical setup that would result in greater accuracy, while others prioritized a balance between accuracy and cost. A summary of the results is presented in Table 4.1, while the conclusions reached by the researchers are outlined next:

— In their exhaustive work of CNWT quality assessment [122], Windt et al. came to the conclusion that whether a certain discretization leads to monotonic convergence depends on the selected metric. For instance, the CPWH required based on the free surface elevation error was 10, but this value increased to 20 if the chosen metric was the velocity profile error. Different meshing

approaches were applied, with the conclusion that the best results were achieved using a uniform mesh. They noticed that this approach would be seldom used given its high computational demand and thus proposed a refinement zone at $\pm H$ from the free surface and a smooth grading away from it.

— Kim et al. performed in [127] an interesting study aiming for an optimal grid system and aspect ratio for a CNWT devoted to wave propagation. They achieve this value by discretizing the analytical wave dispersion relation (equation 2.13) in a cartesian grid, showing that an exact cell ratio of $\tau = \sqrt{3}$ vanished the second order error term. They also noted that non-smooth changes in grid size can cause undesired wave reflections.

— In a thorough investigation on the performance of the `interFoam` solver for wave propagation, Larsen et al. [128] carried out various simulations with different discretization approaches and solver settings in an attempt to unravel the underlying causes of unphysical behaviour in CNWTs, such as increased wave height and celerity. They believe that the overprediction in crest velocity arises from an imbalance in the momentum equation near the free surface, whereas the increased height might be caused by an insufficient amount of diffusion by the solver. All in all, they recommend aiming for a diffusive solver balance that limits these undesirable phenomena. When it comes to discretization, they followed Jacobsen's recommendation of using a unit cell ratio [96]. They surprisingly found that for long simulation timespans, only very small CFL numbers ~ 0.05 could push errors below 0.1% and concluded that most of the CNWT simulations do not achieve temporal convergence. Finally, they also studied the effect of the surface compression method and showed that, if tuned properly, both *MULES* and *isoAdvector* yield similar results at the same cost.

— In his thesis on semi-submersible FOWT platform simulations in a CNWT [121], Pinguet starts with a convergence analysis for a wave propagation case. In an attempt to reduce the computational overhead while maintaining reasonable accuracy, he defined a refinement zone at $\pm H$ from the free surface. For this purpose, he used the `snappyHexMesh` utility and specified a refinement level of 4, meaning that cells near the free surface are 16 times smaller than the base mesh. He then studied the effect of cell contraction in the relaxation zones, proving that it had almost no effect on the results but led to great computational savings. Thus, he decided to use a cell ratio $\tau = 4$ in these zones while keeping a uniform ratio $\tau = 1$ at the wave propagation zone.

Authors	***CPWH***	Δt	τ	L_g	L_a	***Solver notes***
Windt et al. [122]	10-20	$T/800$	1	λ	2λ	3 PIMPLE/MULES iterations, $c_\alpha = 1$
Windt et al. [129]	10	$0.001\,s$	1-4	-	-	-
Windt et al. [130]	10	$T/1400$	1	λ	3λ	-
Boucasse et al. [131]	4-20	$T/400$	2	λ	2λ	Second order time scheme
Larsen et al. [128]	12.5	CFL = 0.15	1	-	-	PCG solver is preferred
Pinguet [121]	10-20	$T/1750$	1-4	λ	2λ	Maximum pressure residual = 10^{-6}
Present work	10	$T/400$	$\sqrt{3}$-4	λ	2λ	Taken from *waveFlume* tutorial

Table 4.1: Representative numerical parameters used for wave propagation simulations in CNWTs.

4.2.3. Numerical set-up

From the lessons learned in the previous section, a setup that can produce accurate results with a low computational budget is sought. The baseline model is taken from the *waveFlume* tutorial available in the *waves2Foam* toolbox. Because turbulence modeling for water waves is a complex topic currently under research, and given that traditional turbulence models did not perform better than laminar flow [35, 129] and even caused excessive dissipation [110], the latter will be used hereinafter unless otherwise stated.

The domain is confined by six boundaries: inlet, outlet, front, back, bottom, and atmosphere. If not mentioned otherwise, the applied boundary conditions are those presented in Table 4.2. In its VOF formulation, OpenFOAM solves for the modified pressure p_{rgh} which corresponds to the difference between the static and hydrostatic ($\rho g h$) pressures, and thus BCs must be set accordingly. The wave

propagation zone has a length equal to 2λ, whereas relaxation zone lengths are set to the conservative values $L_g = \lambda$ and $L_a = 2\lambda$. The considered height for the air region above MWL is $2\,H$.

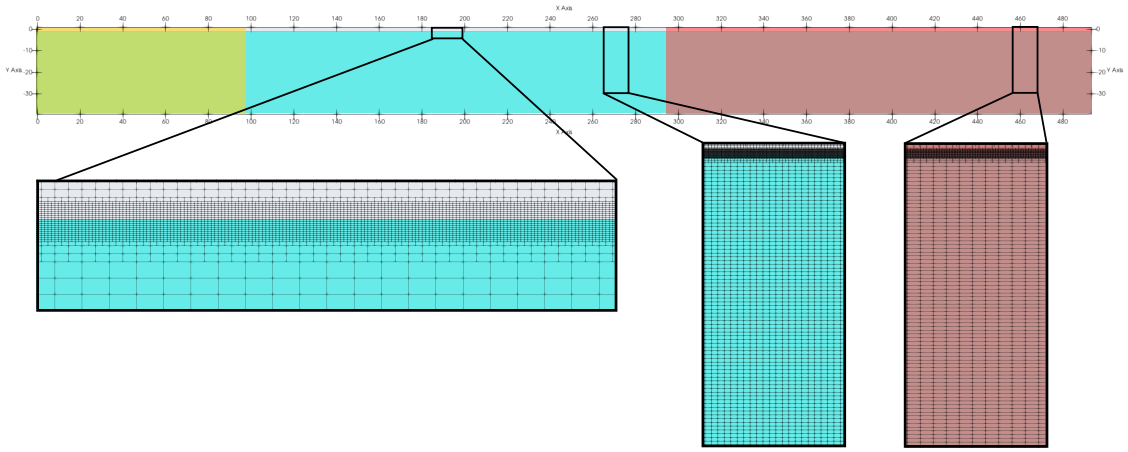


Figure 4.2: Representation of the mesh used for the 2D wave flume: the blue area represents water, whereas the grey area is air. Relaxation zones are shown in yellow (inlet) and red (outlet).

Using the `snappyHexMesh` utility and a refinement level equal to 3, a baseline mesh will be refined at $\pm H$ from MWL, aiming for a uniform grid in this zone with CPWH $= 10$ and cell aspect ratio $\tau = \sqrt{3}$. In the relaxation regions, the cell aspect ratio increases until reaching $\tau = 4$ at the inlet/outlet boundaries. The final mesh with the relaxation zones and mean water level is represented in Figure 4.2, where the change in cell aspect ratio towards the boundaries can be appreciated. Overall, the resulting grid consists of $\sim$ 0.15M cells.

The change in aspect ratio is given by a parameter called expansion ratio δ_{exp}, which is defined as the ratio of the final to the initial cell width along a specific expansion direction. For a given direction i, the relationship between the number of cells N_i, overall expansion ratio δ_{exp} and expansion edge length L_i is given by:

$$\beta_{\text{exp}} = \frac{L_i - d_0}{L_i - d_0\,\delta_{\text{exp}}}\,, \quad N_i = 1 + \frac{\log(\delta_{\text{exp}})}{\log(\beta_{\text{exp}})} \tag{4.3}$$

Where d_0 is the grid size at the start and β_{exp} is the cell-to-cell expansion ratio. This relationship assumes $\delta_{\text{exp}} \geq 1$ and is useful since OpenFOAM needs a value for N_i when defining a graded mesh.

	α	P_{rgh}	**U**
Inlet	zeroGradient	fixedFluxPressure	zeroGradient
Outlet	zeroGradient	fixedFluxPressure	fixedValue 0
Bottom	zeroGradient	fixedFluxPressure	fixedValue 0
Atmosphere	inletOutlet	totalPressure	pressureInletOutletVelocity
Front / Back	empty	empty	empty

Table 4.2: Boundary conditions for the volume fraction, modified pressure, and velocity.

The `interFoam` solver is used to model the multiphase flow, consisting of water and air. Surface tension effects are not considered. The MULES method is selected for interface compression. An adjustable time-step based on a maximum CFL of 0.5 is chosen, with the values taken from the reference *waveFlume* case and an initial time-step $\Delta t = T/400$. Solver options and discretization schemes are also taken from the reference case. Initial internal field values corresponding to a fully developed wave field (from Stokes' second-order theory) are set using the `setWaveField` utility. The case is decomposed using the *Scotch* method and run in parallel for a timespan of eight periods using 16 CPUs, with a total execution time of $\sim$40 minutes.

4.2.4. Results and discussion

Different quantities can be used for assessing the quality of the wave tank, namely surface elevation, wave height, and velocity profile. For sake of brevity, only the first will be addressed here. In the simulation, wave propagation is captured by means of 500 numerical gauges uniformly distributed along the wave propagation region, each producing a time series of the surface elevation $\eta\,(t)$.

$$x = c\,t + x_0 = \frac{\lambda}{T}\,t + x_0 \;\rightarrow\; \frac{x}{\lambda} = \frac{t}{T} + \frac{x_0}{\lambda} \tag{4.4}$$

Since the wave flume is 2D, waves propagate across one spatial dimension only. This makes the problem suitable for representation in a space-time plot, since free surface waves propagate in both domains simultaneously. This visualization allows for a preliminary inspection of the results at every position and instant. For instance, Figure 4.3 depicts the free surface elevation in a $x-t$ plot showing the expected wave pattern. Thanks to the propagating nature of waves, the so-called characteristic lines given by equation 4.4 are represented by stripped black lines and can be used to identify the wave fronts.

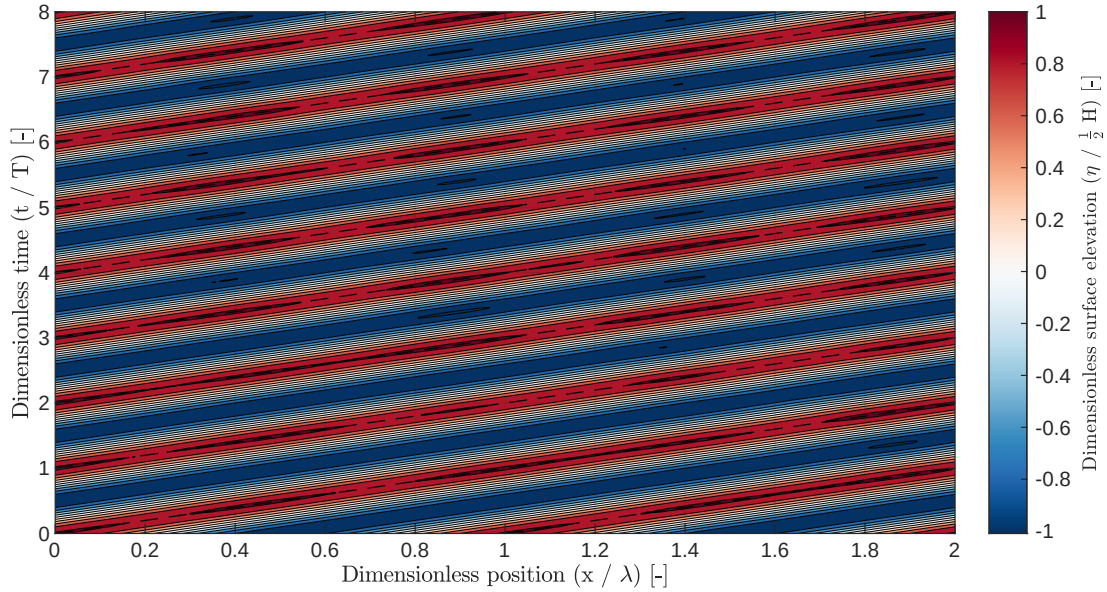


Figure 4.3: Position-Time representation of the normalized free surface elevation. Stripped black lines represent the characteristic lines from equation 4.4.

Yet, Figure 4.3 does not describe how close the numerical solution is to the theoretical second-order wave-field. To get a deeper insight on the results, the surface elevation relative error ϵ (defined in equation 4.5 as the difference between numerical and analytical solutions, normalized with wave height) is presented in Figure 4.4. This is a simple yet powerful metric, as its sign reveals whether the surface elevation is larger or smaller than the reference.

$$\epsilon = 100\,\frac{\eta(x,t)^{\text{num}} - \eta(x,t)^{\text{2nd}}}{H} \tag{4.5}$$

Notice from the colorbar range that the relative error is between $\pm 2.5\%$, which already gives a hint on the expected accuracy of the results. The error magnitude is highest during the first periods ($t/T < 2$), indicating an initial transient stage. Similarly, the error in this initial stage increases as waves propagate along x, probably due to numerical diffusion-dispersion. In this stage, the relative error seems to propagate along characteristics, meaning that waves are consistently propagated.

This behaviour changes after the second period, with the relative error now propagating along with the reflected wavefronts. This might indicate that the implemented relaxation zone is not damping all

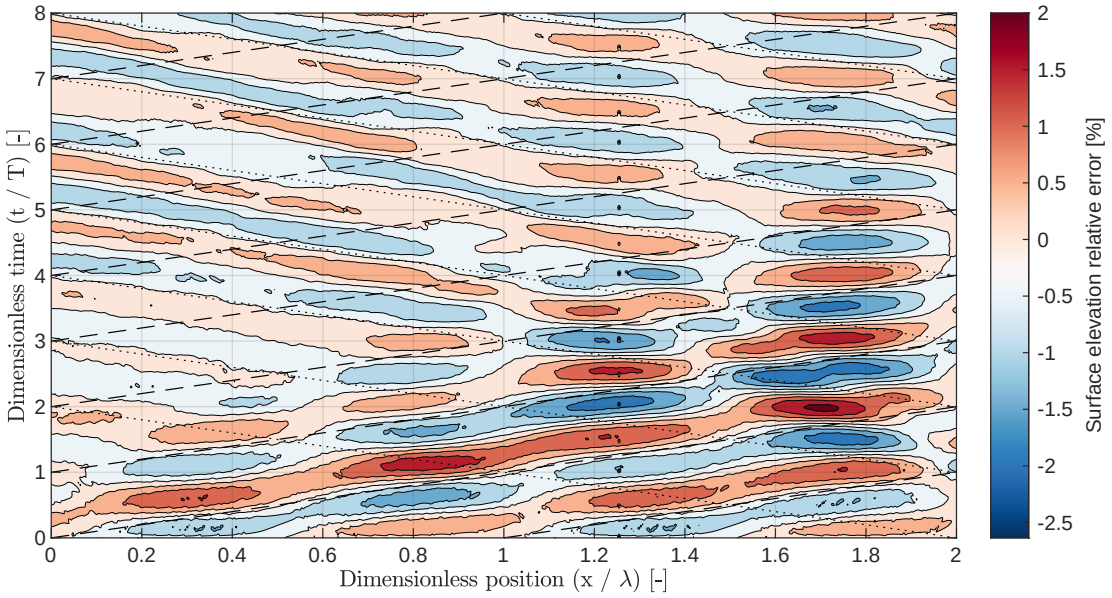


Figure 4.4: Position-Time representation of the surface elevation relative error. Stripped black lines represent the wave characteristics whereas dotted black lines depict the reflected wave fronts.

waves effectively. Hence, some are reflected and interact with the incident wavefield. Still, the error from these reflected waves is considerably smaller than the one from the initial transient phase.

During the transient phase, the lowest error is found at wave crests and troughs, and thus the error in wave height is expected to remain low. Contrarily, surface elevation is overpredicted at wave phase $\epsilon = 90º$ and underpredicted at $\epsilon = 270º$, just where the algebraic Stokes expression predicts zero elevation. This same behaviour is maintained when the error is driven by the reflected wavefield.

$$nRMSE = \frac{1}{H}\sqrt{\frac{\sum_{i=1}^{N_s}\left(\overline{\eta}_{(i)} - \eta_{\text{ref}\,(i)}\right)^2}{N_s}} \tag{4.6}$$

Waves propagating in a CNWT are exposed to numerical dispersion and diffusion errors. Following the recommendations by Windt et al. [122, 129, 130], surface elevation time series at each position x are used to compute the normalized root-mean-square error (*nRMSE*, equation 4.6). This metric will be used to assess the error as the wave propagates through the tank. The *nRMSE* is not computed for the whole timespan but rather for the 2^{nd} and 8^{th} periods, representative of the transient and converged stages, respectively.

The *nRMSE* for the two periods is depicted in Figure 4.5. The two seem correlated, although more simulations might be required to identify a meaningful pattern. In both cases, the error oscillatorily increases as the waves move forward in x, although this behaviour is more pronounced for the second period. This is due to the fact that, early in the simulation, wave reflections have not yet reached the tank zone close to the inlet.

The mean *nRMSE* in the 2^{nd} and 8^{nd} periods were $\sim 0.71\%$ and $\sim 0.43\%$ respectively, which is an acceptable level of accuracy for the implemented CNWT. As discussed earlier, the transient stage characterizing the 2^{nd} period results in higher errors since the interaction between forward and reflected waves has not yet converged. Following this trend, longer simulation spans could result in reduced error.

In the absence of a comprehensive grid refinement or uncertainty analysis and in light of the favorable presented results, the proposed set-up is considered suitable for cases involving the propagation of second-order waves. The main source of error has been associated with a higher-order perturbation.

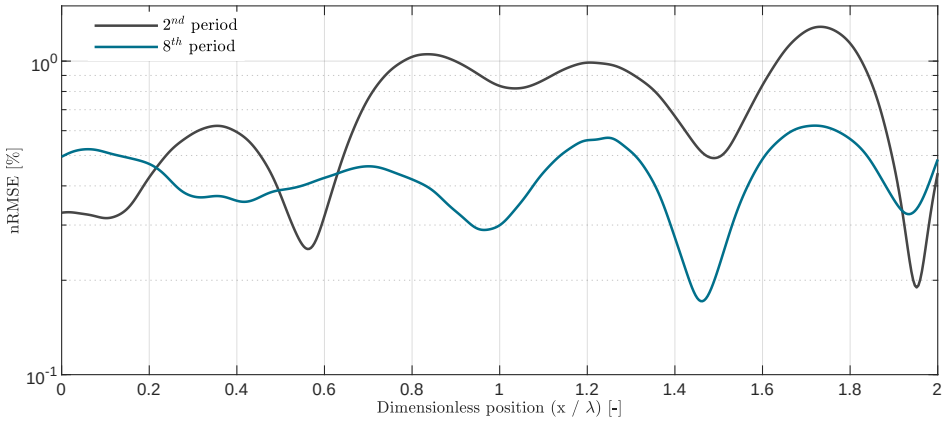


Figure 4.5: Free surface elevation nRMSE along x.

Whether this term stems from the tank discretization (diffusion and dispersion) or from the non-linear interaction with the reflected waves is still an unknown. The former hypothesis could be explored by monitoring the error during a mesh refinement study, whereas the latter could be checked by means of a standing wave simulation and reflection coefficient calculation as proposed by Windt et al. [122], although such analyses are beyond the scope of the present project.

5

Simulation of a floating rigid body

This chapter is meant as an extension of the wave propagation setup for floating rigid-bodies. For this purpose, section 5.1 summarizes how rigid-bodies can be modelled in OpenFOAM and how the mesh can be morphed accordingly. This knowledge is then applied in chapters 5.2 and 5.3 to the simulation of a 2D and 3D floating cylinder, respectively. In both cases, results are compared with both experimental and numerical data.

5.1. Modelling of moving bodies in OpenFOAM

5.1.1. Dynamic mesh

Technique selection

Dynamic meshing capabilities are a must in any modern CFD code. Not only for FSI-related problems but for a wider range of cases like time-varying domain geometries (e.g., a moving piston) or in situations where refinement is needed over a region that changes as the simulation progresses (e.g., along the liquid-gas interface of a wave tank). All the mesh motion methods described in section 2.3.3, including the latter-mentioned adaptive mesh refinement approach (AMR), are available in OpenFOAM.

The overset mesh technique was not included until recently, and specific solvers must be selected that are compatible with it (e.g. `overInterFoam` instead of `interFoam`). Even though it has the potential to be applied to arbitrarily complex motions, this technique should not be used as a silver bullet. Overset meshes can add numerical diffusion to the solution, not to mention the increased computational cost and the non-conservative nature of the interpolation step between grids. All in all, even though this method can account for large mesh motions without deteriorating mesh quality, it requires careful planning and expertise. Moreover, it does not offer increased accuracy compared to morphing meshes with limited motions (see the final note on section 2.3.3).

For these reasons, a dynamic mesh approach based on a morphing strategy will be used henceforth. Note that this choice will limit the range of applicability to small displacements so that mesh quality is not compromised. Recall that the goal of the present thesis is to construct the basis of a framework for the simulation of FOWT, so the capability to handle large motions is not needed at this initial stage.

Set-up

All the definitions regarding dynamic mesh capabilities in OpenFOAM are defined in the dictionary `dynamicMeshDict`, located inside the `constant` folder. A template for this file is presented in Appendix 8.3, corresponding to the case presented in section 5.2. The OpenFOAM implementation allows for seven different types of mesh motion, though for morphing meshes in a single domain, the `dynamicMotionSolverFvMesh` option is used. Many solvers exist to move the grid based on the desired motion, most of which solve a Laplacian equation based on boundary displacements.

For the current implementation, the `sixDoFRigidBodyMotion` solver based on a single six-DoF rigid-body motion is chosen. It applies a spherical linear interpolation (SLERP) of movement as a function of distance to the object surface. Should multiple rigid bodies be considered, then the `rigidBodyMotion`

solver must be selected. How the body motion is mapped onto the mesh can be selected via the diffusivity model, with `inverseDistance` being a common approach for rigid bodies (the farther away from the body, the less mesh morphing). This method considers that only the set of patches specified by the user are moving, whereas the rest is kept fixed. It is also possible to select the rate of change in diffusivity to linear, quadratic, or exponential. Via the `innerDistance` and `outerDistance` entries, the user can bound the mesh deformation limits within the fringe located between the two distances (normal to the body wall). The inner distance is typically in the order of the boundary layer thickness, whereas the outer distance is chosen based on the minimum distance to the closest domain boundary [4].

Any of the described methods for dynamic meshes requires new boundary conditions for the mesh motion. For rigid body motions, these are defined in *0/pointDisplacement*. All fixed patches should have a zero boundary field assigned except for the body patch, to which a `calculated` type will be assigned. The velocity boundary condition must be of type `movingWallVelocity` for the body patch. Regarding the pressure boundary condition at the body patch, `fixedFluxPressure` ensures that the pressure gradient is such that the flux on the boundary is that specified by the velocity boundary condition.

By default, the `waves2Foam` solver used for the generation of waves in Chapter 4 is not compatible with dynamic meshes. Even though it has not been validated for such cases, the integration of dynamic meshes into *waves2Foam* is possible as stated by its author in the manual, who recommends creating a separate solver (i.e. `waveDyMFoam`) to serve this purpose. A brief guide on how to set it up is presented in section 5.3 from the manual [2].

5.1.2. Six-DoF rigid body

Definition in OpenFOAM

A very broad and comprehensive description of the rigid-body implementation in OpenFOAM is given in the master's thesis by Limpens and Xu in the context of a generic floating object [132]. As seen in the previous section, the rigid-body motion is tightly related to the choice of a dynamic mesh solver. For that reason, its parameters are also defined inside the *dynamicMeshDict* dictionary; see Appendix 8.3 for a template. The user must specify six different types of information:

Body properties. These are the total `mass` and `momentOfInertia`. The latter is given in diagonal form, that is, as the body's inertia with respect to its principal axes.

Initial conditions. The motion state of the rigid body is defined at the initial time. Regarding translation motion, the user can specify the initial position, velocity, and acceleration through the keywords `centerOfMass`, `velocity` and `acceleration`. Angular variables such as initial orientation, angular speed, and angular acceleration are defined via the keywords `orientation`, `angularMomentum` and `torque`. Note that the latter two are in fact the product of the angular variables and the inertia tensor. Lastly, the user can also define the initial position of the rotation center, `centerOfRotation`, which by default is set equal to the center of mass.

Constraints. The user can restrict any of the six degrees of freedom by combining the following five types of constraints: `line`, `plane` and `point` for the linear motion, and `axis` and `orientation` for the angular motion.

Restraints. In this context, restraints refer to applied loads. The ones available by default depend on the body motion, such as linear and angular springs and dampers. In the *sixDoFRigidBodyMotion* library, no restraint exists that applies a constant force or torque to the rigid body. The *waves2Foam* library includes a custom restraint type for catenary mooring lines. Two new restraint types for constant and gyroscopic loads have been implemented; see Appendix 8.3.

Solver control. Three second-order solvers are available to integrate the rigid body equations in time, two of which are implicit (`Newmark` and `CrankNicolson`) and the other is explicit (`sympletic`). More information on the solver procedure is given in the next section.

Output control. The user can control whether the motion state of the rigid body will be written onto the solver output log or to an external file at each time-folder through the `report` and `reportToFile` keywords, respectively.

One very important note is that the `orientation` tensor corresponds to the transformation matrix $\underline{\mathbf{Q}_{pi}}$. The latter defines the body's principal axes of inertia rather than the more intuitive floater axes (see Appendix 8.3 for a detailed explanation). This is because the rigid-body solver stores angular momentum in the body-fixed principal frame, explaining why only the inertia tensor in its principal component form is needed. Therefore, the reference orientation of the body (where $\underline{\mathbf{Q}_{pi}} = \underline{\mathbf{I}}$) is such that the principal axes are aligned with the Cartesian. The user must be aware of this fact when imposing an initial orientation or when retrieving the Euler angles from the body orientation.

Take also into account that the user can specify the gravity vector with the `g` keyword which will act as a constant force. For incompressible flow simulations, the user can define a reference value for the fluid density to be used to compute the fluid loads onto the body via the `rhoName` entry.

Numerical method

The rigid-body library in OpenFOAM approximates the system of second-order differential equations presented in equation 2.28 by means of numerical integration. But first, external loads must be evaluated. Equations 5.1 and 5.2 show the force and torque expression in continuum media, where fluid (hydrodynamic and aerodynamic) loads are decomposed into viscous v and pressure p components. Loads caused by point forces $\mathbf{F}_i$ and moments $\mathbf{M}_i$ such as gravity or moorings are also included.

$$\mathbf{F}_{\text{body}} = \iint_S \left(p\,\hat{\mathbf{n}} + \underline{\boldsymbol{\tau}} \right) dS + \sum \mathbf{F}_i = \mathbf{F}_p + \mathbf{F}_v + \sum \mathbf{F}_i \tag{5.1}$$

$$\mathbf{M}_{\text{body}} = \iint_S \mathbf{r}_{PR} \times \left(p\,\hat{\mathbf{n}} + \underline{\boldsymbol{\tau}} \right) dS + \sum \mathbf{r}_{iR} \times \mathbf{F}_i + \sum \mathbf{M}_i = \mathbf{M}_p + \mathbf{M}_v + \sum \mathbf{r}_{iR} \times \mathbf{F}_i + \sum \mathbf{M}_i \tag{5.2}$$

Where $\underline{\boldsymbol{\tau}}$ is the viscous stress tensor, p is the fluid pressure, and $\hat{\mathbf{n}}$ is the body surface-normal vector. For incompressible flow, the stress tensor $\underline{\boldsymbol{\tau}}$ is computed as twice the symmetric part of the velocity gradient tensor $\nabla\mathbf{u}$ times the dynamic viscosity μ. The vector $\mathbf{r}_{PR}$ is the radii from the fluid load application point to the rigid body rotation center. Because OpenFOAM works with a discrete version of the problem, viscous and pressure loads are computed at each cell face f defining the fluid-body interface:

$$\mathbf{F}_p = \sum_f p_f\, S_f\, \hat{\mathbf{n}}_f\,, \quad \sum_f \mathbf{F}_v = S_f\, \hat{\mathbf{n}}_f \cdot \underline{\boldsymbol{\tau}}^* \tag{5.3}$$

By imputing the expressions for $\mathbf{F}_{\text{body}}$ and $\mathbf{M}_{\text{body}}$ into the Newton-Euler rigid body equations 2.28, a system of ODEs is obtained where the only remaining unknown is the motion state $\mathbf{x}$. Given an updated value of acceleration based on the loads acting on the body, a damping coefficient f_d can be defined to mitigate sudden jumps in acceleration that might transfer onto the fluid and destabilize the simulation. The user can specify this value using the `accelerationDamping` entry. To advance the system implicitly in time, OpenFOAM allows the user to select between two well-known numerical time-integrators:

Newmark-β. This method was first developed in 1959 for use in computational structural dynamics [133], but has expanded its use to a wide range of dynamic systems. The method uses the extended mean value theorem to compute first and second time derivatives:

$$\begin{aligned} \dot{\mathbf{x}}^{n+1} &= \dot{\mathbf{x}}^{n} + f_d\,\Delta t\,\left[(1-\gamma)\,\ddot{\mathbf{x}}_n + \gamma\,\ddot{\mathbf{x}}_{n+1}\right] \\ \mathbf{x}^{n+1} &= \mathbf{x}^{n} + \Delta t\,\dot{\mathbf{x}}^{n} + f_d\,\frac{\Delta t^2}{2}\,\left[(1-2\beta)\,\ddot{\mathbf{x}}_n + 2\,\beta\,\ddot{\mathbf{x}}_{n+1}\right] \end{aligned} \tag{5.4}$$

Where β and γ are numerical parameters. The parameter γ introduces damping to the numerical method, controlling the so-called artificial viscosity [134]. If $\gamma = 0.5$, no damping is added, whereas if $\gamma > 0.5$ the artificial viscosity added to the system is proportional to $\gamma - 0.5$. On the other hand, the parameter β defines how implicit or explicit the method is. An explicit central difference scheme is obtained with $\gamma = 0.5$ and $\beta = 0$, whereas a midpoint rule scheme (average constant acceleration) requires from $\gamma = 0.5$ and $\beta = 0.25$. The method is found to be unconditionally stable for $\beta \geq 0.5\,\gamma \geq 0.25$. The user can define the two numerical parameters through the `beta` and `gamma` keywords, which by default are set to those from the average constant acceleration scheme.

Crank-Nicolson. This method, originally developed as a numerical approach for heat conduction differential equations [135], is based on the trapezoidal rule and can be thought of as an implicit Runge-Kutta method.

$$\begin{aligned} \dot{\mathbf{x}}^{n+1} &= \dot{\mathbf{x}}^{n} + f_d\,\Delta t\,\left[(1-a_{oc})\,\ddot{\mathbf{x}}^{n} + a_{oc}\,\ddot{\mathbf{x}}^{n+1}\right] \\ \mathbf{x}^{n+1} &= \mathbf{x}^{n} + \Delta t\,\left[(1-a_{oc})\,\dot{\mathbf{x}}_{n} + v_{oc}\,\dot{\mathbf{x}}_{n+1}\right] \end{aligned} \tag{5.5}$$

Where a_{oc} and v_{oc} are the acceleration and velocity off-centering coefficients. By default, these are set to $\frac{1}{2}$ which makes it equivalent to a Newmark-β average constant acceleration scheme.

Though it is true that a third method called *sympletic* is available in OpenFOAM, its explicit nature makes it incompatible with multiple calls within a time-step, thus making it inadvisable for FSI problems, as explained in the section below.

Fluid-Structure Interaction

As introduced in section 2.3, rigid-body motions depend on the fluid loads, which in turn are a function of the body's position and velocity. With the proposed formulation, the fluid and body equations are said to be solved in a partitioned manner, meaning that the two systems cannot be solved simultaneously. How OpenFOAM deals with this problem must be well understood so that the user knows which are the inherent limitations of the method and which tools are available for reducing the partitioning error and avoiding non-physical behaviours and divergence.

Inside each PIMPLE outer-corrector loop, first the surface and restraint loads are computed, and then the rigid-body equations are solved (see Figure 2.7). The grid, fluxes, and boundary conditions are next updated according to the rigid-body motion and diffusivity scheme. If the rigid-body equations were solved only once per time-step, then the system would be loosely coupled since the loads applied to the rigid-body wouldn't be affected by the updated pressure and velocity fields. As described in section 2.3, both small and large time-steps could make the simulation diverge, thus calling in the need for sub-iterating methods.

In an attempt to reduce the partitioning error, multiple outer-corrector loops must be employed. With the `moveMeshOuterCorrectors` flag activated on the PIMPLE solver control sub-dictionary, the rigid-body equations will be solved once per outer-corrector iteration and the mesh will be updated accordingly, reducing the interface residual and tightening the coupling between solid and fluid domains. This iterative procedure should also help non-linear mooring models to converge. This procedure is summarized as a diagram in Figure 5.1, which for clarity omits most of the PIMPLE steps from Figure 2.7.

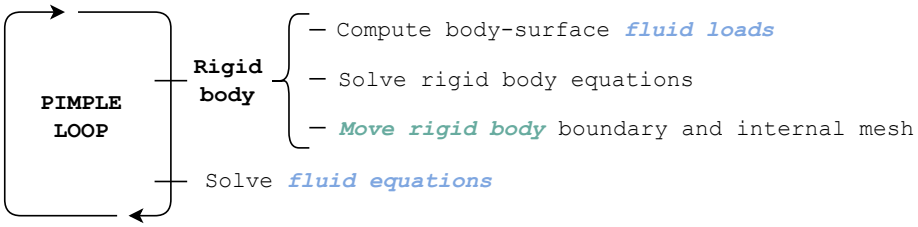


Figure 5.1: Diagram of the implemented FSI coupling between the rigid body and fluid solvers.

Unfortunately, in OpenFOAM, there is no way of achieving a fully coupled system: even though the PIMPLE algorithm can be run until the fluid residuals fall below the desired threshold, the same cannot be ensured for the body motion. Thus, a common approach is to fix the number of outer-correctors rather than making it depend on the flow-field residuals. By setting the `checkMeshCourantNo` flag to true, the user can monitor the mesh-based CFL number, which is a good stability indicator for the rigid body motion.

The presented strongly coupled approach is only available with the implicit solvers Newmark-β and Crank-Nicolson. Moreover, these iterative methods are compatible with the under-relaxation technique to further stabilize the simulation, which is controlled using the `accelerationRelaxation` keyword that controls the under-relaxation coefficient ϕ:

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}^{\text{old}} + \phi\,(\ddot{\mathbf{x}}^{\text{new}} - \ddot{\mathbf{x}}^{\text{old}}) \tag{5.6}$$

With $\phi < 1$, the stability of the numerical scheme will increase at the price of slower convergence rates. Advanced techniques such as Aitken under-relaxation are not natively available in OpenFOAM.

5.2. Decaying heave motion of a 2D cylinder

The objective of this first case is to get acquainted with rigid-body and dynamic mesh utilities from OpenFOAM by simulating the free decay oscillation of a cylinder in heave and comparing the findings with experimental measures and other numerical results.

5.2.1. Test description

The numerical implementations aim to replicate the experiment by Ito [3] in which a two-dimensional semi-immersed circular cylinder is released from an initially displaced position while the resulting transient oscillation is recorded. The cylinder, whose weight was adjusted to attain exactly a half-immersed condition, was attached to a rod so that only pure heaving motion would be obtained. The experiment was conducted at a towing tank, whose main characteristics are shown in Table 5.1.

h [m]	**L** [m]	**W** [m]	**R** [m]
1.2192	27.4320	2.5908	0.0762

Table 5.1: Towing tank properties from Ito's experimental campaign: depth **h**, length **L**, width **W** and cylinder radius **R**.

The goal of the original campaign was to compare the results with those of the 2D potential-flow theory (Maskell & Ursell), so a very long cylinder ($L/D = 17$) at a short distance from the lateral walls (1.27 cm) was used to satisfy the two-dimensional condition. To reduce the effects of viscosity and turbulence, a relatively small value is chosen as the cylinder's initial displacement ($z_0 = R/3$) from the MWL. This combination of a 2D domain (that keeps the numerical set-up easy to build and debug and the simulation time low) with a small displacement regime (the morphing mesh technique is best suited for modest motions) is what makes this test so attractive for validation purposes.

5.2.2. Numerical set-up

A CNWT wave tank is implemented in OpenFOAM to recreate Ito's setup. The current implementation follows the procedure from the wave propagation case, see section 4.2. Some aspects, however, must be adapted to meet the geometry and boundary conditions of the new case.

Grid geometry

Regarding the grid, it is important to produce a body-conforming mesh. Thanks to the simple circular cross section of the body, the `blockMesh` utility suffices for this purpose. The geometric set-up is based on the *cylinder* tutorial from OpenFOAM, although it must be adapted to the current case. A sketch on the `blockMesh` strategy and relaxation zones is presented in Figure 5.2.

From past experiences, it was seen that keeping the initial free surface aligned with the grid was crucial to avoid non-physical oscillations near the interface. Thus, properly accounting for the cylinder's initial displacement is imperative. To do so, the body's center is defined at the origin, while the MWL is set at $-z_0$. The mesh blocks surrounding the cylinder now vary in size and thus must contain different numbers of cells in order to produce a uniformly conforming mesh. The meshing procedure is parametrized as a function of the initial displacement and the number of cells along the cylinder's semi-perimeter and radial directions (n_{sp} and n_r, respectively). This allows various z_0 values and refinement levels to be tested without the need to edit the `blockMesh` file. Since the flow is treated with a laminar model, no mesh refinement in the near-wall region is required. The final mesh was constructed with $n_{sp} = 60$ and $n_r = 15$, leading to a total cell count of 8100 cells.

Boundary conditions

In the original experiment, no measures were taken to avoid wave reflections from the tank boundaries. This is because the total duration of the experiment ($\Delta T \sim 3\,s$) was shorter than the time it took for the waves generated by the oscillating motion to bounce from the walls and reach the cylinder again ($\Delta T \sim 6\,s$). From a numerical perspective, aiming for a full-length model will prohibitively raise the cell

count and computational overhead. For this reason, two wave relaxation (damping) zones are defined along the lateral boundaries of the CNWT. Since the wavelength of the radiated waves is not known a priori, the length of the relaxation zones is set to $L_a = 5\,R$. The wave propagation zone length is set equal to twice the water depth, $L_p = 2h$. All boundary conditions are taken from the baseline case in section 4.2.

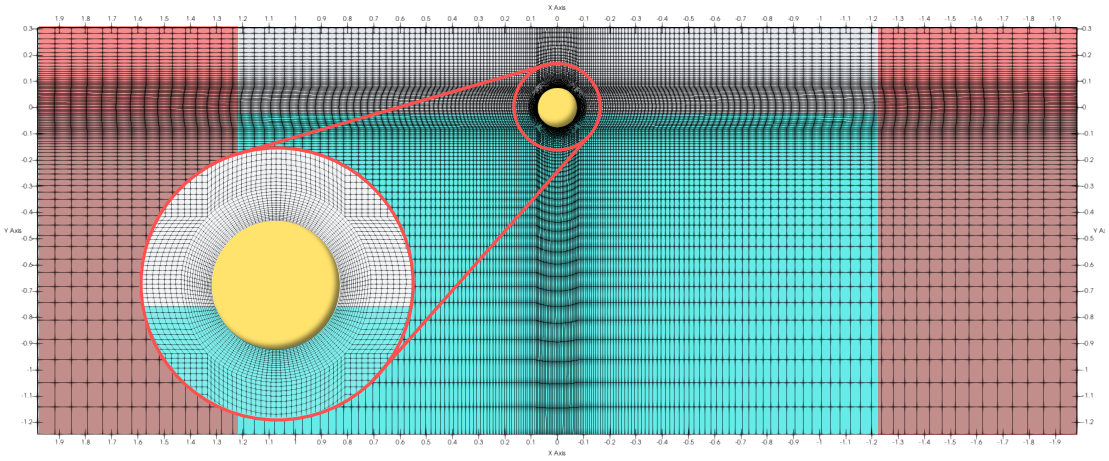


Figure 5.2: Representation of the mesh replicating Ito' set-up. Wave relaxation zones are shown in red.

Morphing mesh and rigid body

All the information related with the rigid-body simulation and mesh morphing procedure is defined in the `dynamicMeshDict` file. The mesh is allowed to move in the region between $2.5\,R$ and $6\,R$ from the cylinder's center, while the region's interior and exterior remain fixed. The morphing procedure is based on a diffusivity scheme whose effect is proportional to the square of the inverse distance from the cylinder. The rigid-body is restrained using *line* and *orientation* constraints so that only heave motion is allowed. The Newmark-β solver with an average constant acceleration scheme is used to advance the rigid body equations in time. No damping nor relaxation factors were used to stabilize the simulation.

Schemes and simulation control

The numerical schemes are taken from the linear wave propagation case in section 4.2. A GAMG solver is used for the dynamic mesh motion. Regarding the PIMPLE algorithm, five outer-corrector loops are used since this has been referred to as the minimum amount of iterations to achieve coupling convergence [136, 137]. Because the cells are now deformed by the cylinder motion, two non-orthogonal corrector steps are applied. The flags `moveMeshOuterCorrectors` and `correctPhi` are set to true. The initial time-step is set to $\Delta t = 1\,ms$, although an adjustable $\Delta\,t$ based on a maximum CFL number of 0.1 is used, following the recommendations from [128]. In practice, three different CFL numbers are computed based on the internal field velocity, interface velocity, and mesh velocity. Although the latter only has an informative purpose and does not actually affect the time-step, it is still a good stability indicator in the case of high-frequency body oscillations.

5.2.3. Results and discussion

During the simulation, the cylinder's vertical position at the last PIMPLE iteration of every time-step is stored. In order to compare the numerical results to Ito's, the latter is normalized by the initial displacement $z' = z/z_0$. Similarly, the time scale is non-dimensionalized as $t' = \sqrt{g/R}\,t$. Apart from the experimental campaign, the present results are also compared against those of Gatin et. al [138] , who also recreated Ito's experiment using an advanced FSI approach in OpenFOAM. The rigid-body equations were solved after each pressure correction step instead of only once per PIMPLE iteration, as done in the present work. They verified the model via uncertainty analysis based on mesh refinement, following the methodology proposed by Eça et al. [139]. Only the results from their finest grid ($\sim 1.1\,M$ cells) are used for comparison. The three decay oscillation time-series are presented in Figure 5.3.

Since a detailed uncertainty analysis is beyond the scope of the present project, a qualitative validation

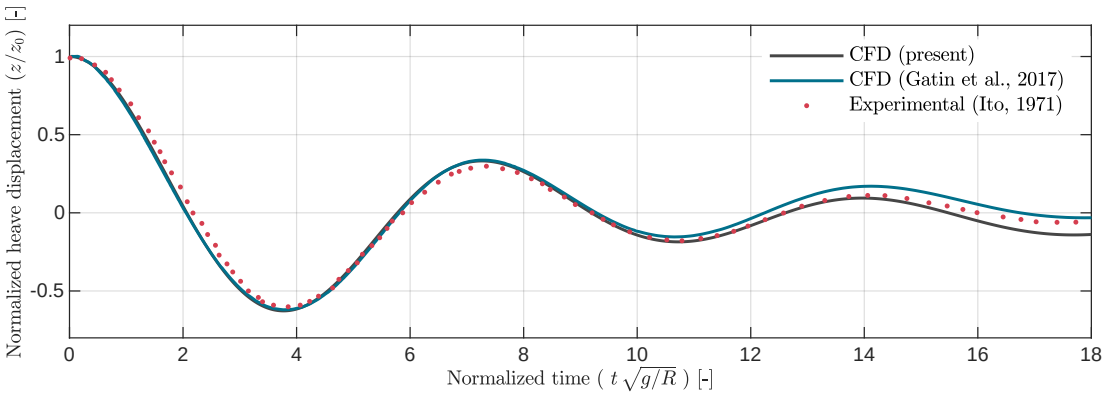


Figure 5.3: Comparison of the non-dimensionalized free decay oscillation amplitude with Ito's experiment and Gatin et al. numerical simulation.

m $[kg]$	**D** $[m]$	**h** $[m]$	$\mathbf{I_{xx}}$ $[kg \cdot m^2]$	$\mathbf{C_g}$ $[m]$	$\mathbf{d_f}$ $[m]$	$\mathbf{d_m}$ $[m]$	$\mathbf{l_m}$ $[m]$	$\boldsymbol{\rho_m}$ [kg / m]
35.85	0.515	0.401	0.90	0.0788	0.172	0.178	6.95	0.1447

Table 5.2: Buoy and mooring line properties: mass m, diameter D, height h, inertia moment I_{xx}, center of gravity C_g (distance from base), draft d (for the free and moored configurations), mooring line length l_m, and linear density ρ_m.

is preferred. Overall, there seems to be good agreement with the reference data. Both the present and Gatin's results slightly overpredict the response amplitude but coincide with each other until $t' \sim 9$. From there, the results from Gatin et al. increase this overprediction. Meanwhile, present results follow Ito's until $t' \sim 12$ after which they start underpredicting the response. Given that the cell count of the current simulation is ~ 140 times smaller than Gatin's, it can be argued that the current implementation achieves similar results with a substantial decrease in computational overhead. In the light of these favorable results, the presented set-up is considered suitable for the simulation of freely floating bodies under small displacements.

5.3. Decaying pitch motion of a 3D cylinder

With the lessons learned from the previous case, this section aims to further test the proposed methodology by investigating the free and moored decay motion in pitch of a cylinder in a 3D domain. Again, the outcome will be compared to both experimental measures and other numerical results. This section should serve as a close-up of more realistic FOWTs conditions compared to the previous case by introducing a 3D geometry, mooring restraints, and higher amplitude oscillations.

5.3.1. Test description

The numerical setup intends to replicate the experimental campaign by Paredes et al. [140] in which the decay motion of a cylindrical buoy is analyzed under different restraint conditions, including unrestricted motion and catenary moorings. This same case has been numerically replicated with CFD by Palm et al. [4] and Rivera et al. [108] whose results will be used to further validate the present implementation. Because no motion constraints are applied, coupling between different DoFs might show up (especially for heave and pitch, known as Mathieu instability). However, only pitch motion will be investigated in this section.

The experimental test was carried out in a wave tank that was 5 m wide with a water depth of 0.9 m. The properties of the used buoy and catenary moorings are presented in Table 5.2. These values already take into account the extra mass and inertia introduced by the mooring line attachments. Because the uncertainty associated with some of the presented values led to an amplified discrepancy with the experimental results when simulated in [4, 108], some properties of the buoy were changed in their CFD validation to better fit the experimental results (always within the experimental uncertainty range). Specifically, the center of gravity was shifted 3mm upward and the moment of inertia was increased to 0.95 kg m^2. These changes apply only to the free decay test.

5.3.2. Numerical set-up

The CNWT implementation follows the procedure from the heave decay case; see section 5.2. Only slight changes are needed to accommodate the new conditions.

Grid geometry

The grid definition for the present case is based on the work of Wang et al. [141] from the OC6 phase Ib campaign where multiple refinement zones were used to shrink the grid size near the floating body but coarsen it far away from it. The lessons learned in section 4.2 are also applied to the present case, aiming for a high level of refinement of the free surface and an increased cell aspect ratio towards the relaxation zones.

First, a background mesh is created by means of the `blockMesh` utility, already accounting for the aforementioned cell aspect ratio. By using the `snappyHexMesh` utility, various boxes with different refinement levels n_r are defined, as shown in Figure 5.4. The refinement level takes integer values, starting from zero for the background mesh. Each time n_r is increased, the grid length l is divided by a half (and the cell volume by eight):

$$l_{(n_r)} = \frac{l_0}{2^{n_r}} \tag{5.7}$$

A widely used metric to define mesh refinement at the boundary of the object of interest is the number of points per structure length. Given that the buoy is a cylinder, it is more appropriate to define the number of points per cylinder diameter (p.p.c.d.). Given that the background mesh size is set to $l_0 = 2/3\,R$, where R is the cylinder radius, and that the cylinder lies within a zone with $n_r = 3$:

$$\text{p.p.c.d.} = \frac{\text{grid points}}{\text{diameter}} = \frac{\text{diameter}}{\text{grid size}} = \frac{2\,R}{l_{(n_r)}} = \frac{R}{l_0} 2^{n_r+1} = \frac{3}{2} 2^4 = 24 \tag{5.8}$$

The cylinder geometry was defined using the `searchableCylinder` functionality rather than importing a geometry stl file. This allows the parametric definition of the buoy, which makes it easier to change its position and orientation. Because of the boundary layer developing at the walls of the buoy, a four-element prism layer mesh with a width of 2 cm was defined around the cylinder. Overall, the final mesh consisted of $\sim$ 1.25M elements contained within the box $x \in [-6, 6]$ m, $y \in [-2.5, 2.5]$m and $y \in [-0.9, 0.9]$m. Because of the relatively large cell count, the utility `renumberMesh` is executed, which reduces the bandwidth of the matrix system, increasing the efficiency of linear solvers.

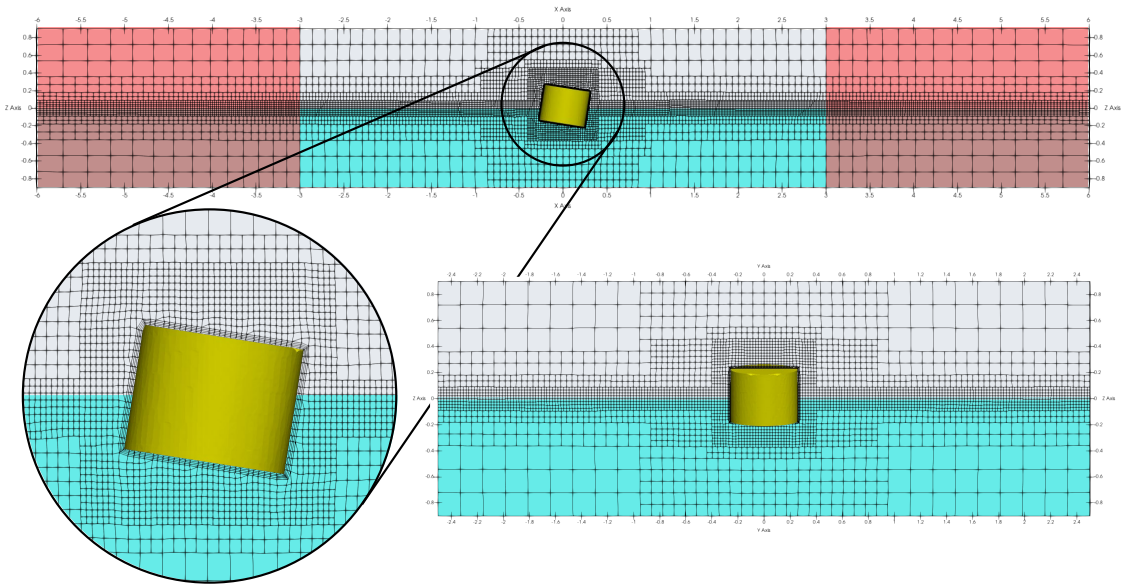


Figure 5.4: Representation of the mesh replicating Paredes' set-up. Wave relaxation zones are shown in red.

Boundary conditions
The numerical domain is taken from the work of both Palm et al. [4] and Rivera et al. [108] and consists of two relaxation zones that extend the same length as the simulation region, $L_a = L_p = 3$m. All boundary conditions are taken from the baseline case in section 5.2 except for the lateral walls, to which a slip condition is applied.

Morphing mesh and rigid body
The inner and outer distances that control the morphing mesh have been modified: the former is kept at 5 cm from the body, whereas the latter extends to 0.8m. The idea is to keep the boundary layer inside the inner distance and extend the outer one to the closest boundary face. Because no information was given regarding the polar moment of inertia I_z, it was computed assuming the cylinder had uniform density. No motion constraints were applied.

The quasi-steady catenary mooring restraint model from *waves2Foam* [2] was used for the moored case, which can account for three different states of the lines: simple, resting, and hanging. The three mooring lines were placed symmetrically, 120º apart. The attachments were located at the water line at a distance of 0.015m from the buoy. Because the attachments were defined with the buoy in its equilibrium position, the effect of initial inclination must be taken into account when specifying its position.

Schemes and simulation control
The numerical schemes are taken from the heave decay test, see section 5.2. When it comes to the solver for the water fraction (α), `nAlphaCorr` was set to 3, `nAlphaSubCycles` to 2, `cAlpha` to 0.5, and `nLimiterIter` to 5. The maximum CFL was increased to 0.5 (adaptive time-step), whereas the number of outer-corrector loops was reduced to 3 to save computation time. Aiming for a comparison with Rivera et al. [108], the total simulation time was set to three times the natural period of the buoy in pitch, $t_f = 3\,T_\theta = 3.51$s.

5.3.3. Results and discussion
Two different cases (free decay and moored) were simulated. The motion amplitude is normalized with the initial displacement, $\theta_0 = 8.898$º for the free decay and $\theta_0 = 11.353$º for the moored set-up. On the other hand, time is scaled with the natural period of the buoy in pitch $T_\theta = 1.17$s. In the experimental investigation, each test was repeated at least fifteen times to reduce uncertainty in the measurements. The oscillation period was then obtained by dividing the time elapsed between the first and last crest by the number of cycles encompassed in that timespan. Palm et al. [4] used a total timespan of seven periods, considerably longer than the three considered here and in [108].

The results for both the free decay and moored motions are presented in figure 5.5 and are compared with the aforementioned experimental and CFD data. The numerical oscillation period for the two cases is presented in Table 5.3 and compared with the reference data.

	Oscillation period [s]		Relative error [%]	
	Free	*Moored*	*Free*	*Moored*
Experimental [140]	1.170	1.163	-	-
Palm et al. [4]	1.163	1.136	-0.60	-2.32
Rivera et al. [108]	1.130	1.050	-3.41	-9.72
Present	1.157	1.110	-1.10	-4.55

Table 5.3: Pitch decay motion period: experimental and CFD results comparison for free and moored configurations.

The free-decay test results are in good agreement with the reference data within the uncertainty that can be expected from CFD simulations. All numerical simulations underpredict the pitch-natural period, with Palm's being the closest to experimental data. Given that the present simulation is based on the same methodology as Rivera's, it comes as no surprise that both yield similar results. Even though the current simulation might predict a natural period value closer to the experimental, Rivera's model better adjusts to the crests and troughs, which are slightly underpredicted in both the present and

Palm's results. The latter, even though it uses an 8M cell grid, heavily underpredicts the overall motion amplitude.

When it comes to the moored configuration, the discrepancy between CFD and experimental becomes evident. This might be caused by the limitations of the mooring models, and, to a lesser extent, by slight differences in the model properties. Mooring cables introduce loads not only in pitch but in all six DoFs, leading to highly coupled, more complex motion compared to the free case. Both the current and Rivera's results overpredict the motion amplitude, which is again underpredicted by Palm et al. Compared to the free decay test, the error in the natural period was increased in all three models, which again underpredicted its value.

Overall, the present results are still in good agreement with the reference data, especially in the free decay test. For the moored case, discrepancies with experimental data arise, although the current results are still consistent with the presented CFD data. Because further validation of the mooring model is outside the scope of the present project, the implemented set-up is considered suitable for the simulation of freely floating and moored 3D bodies in calm water. Adaptation to FOWTs will come in Chapter 7.

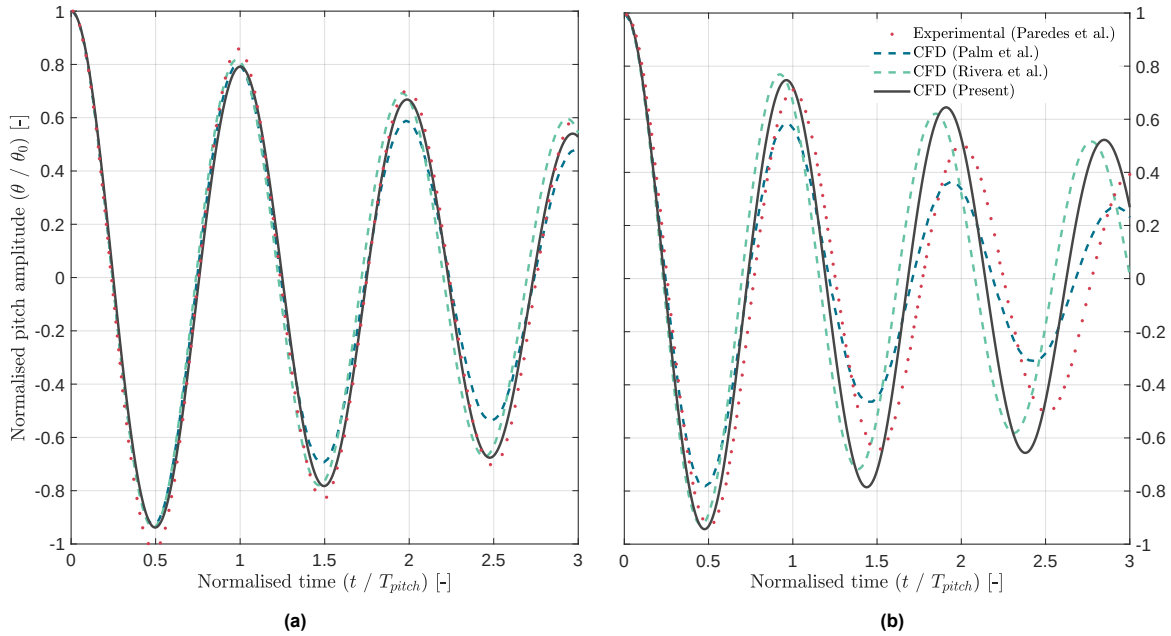


Figure 5.5: Results comparison of the cylindrical buoy decay motion in pitch with both experimental and CFD data for the free (a) and moored (b) configurations.

6

ALM simulation of a turbine with prescribed motion

In this chapter, the *turbinesFoam* library for the unsteady-ALM simulation of bottom-fixed turbines is modified so that it can handle any arbitrary prescribed motion along the six rigid-body DoFs. A general description of *turbinesFoam* and the proposed modifications are presented in sections 6.1 and 6.2. The modified library is then tested by replicating the experiment from the OC6 Phase III campaign, where a single-DoF prescribed motion is imposed on a scaled turbine. The experiment description, simulation set-up, and results discussion are reported in sections 6.3, 6.4 and 6.5, respectively.

6.1. The ALM in OpenFOAM

OpenFOAM allows the inclusion of momentum and energy source terms into the CFD simulations via the *fvOptions* dictionary. Various shapes and models are implemented, yet no native ALM is available in OpenFOAM. User-coded libraries are then needed, being *turbinesFoam* [142] one of the most popular options given its ease of use and open-source access. This library was developed (and is currently maintained) by Peter Bachant in the context of vertical-axis wind turbines [1], although it is also compatible with HAWTs. Given how simple yet powerful this library has proven to be, it has been chosen as the main ALM tool for the present project.

The author is aware of SOWFA (*Simulator fOr Wind Farm Applications*), an open-source OpenFOAM toolbox by NREL based on the ALM and coupled with FAST. Its use is advocated for wind turbine arrays in the atmospheric boundary layer. Because this project's goal is to adapt an ALM library so it can be used with FOWTs, it was thought that *turbinesFoam*'s architecture would be easier to understand and reverse-engineer. Because *turbinesFoam* works solely based on the *fvOptions* framework for source-term addition, it is compatible with any solver or turbulence model and hence suitable for multiphase simulations —which are the backbone of the present thesis. Moreover, it is more likely that the modifications proposed along this project will be accepted and included in a personal library (*turbinesFoam* by P. Bachant) rather than in an enterprise tool (SOWFA by NREL).

6.1.1. Introduction to *turbinesFoam*

The library by P. Bachant was initially developed for vertical-axis (or cross-flow) wind turbines and validated against experimental measurements and blade-resolved simulations [1]. Only qualitative near-wake flow features were recovered when used with a RANS model, while a more accurate representation of the shed vortical structures was captured by using a Smagorinsky LES model. Results were generally aligned with the experimental data, yet the computational burden was up to four orders of magnitude lower than that of blade-resolved RANS.

The ALM implementation follows the basic procedure described in section 2.2.3: loads are computed for each actuator element (AE) from the sectional aerodynamic polars (equation 2.26) and this force is then added to the Navier-Stokes equation as a body force or momentum source. To avoid instabilities

caused by step gradients, the force is projected away from the element position by means of a smearing or projection function. In *turbinesFoam*, the spherical Gaussian function given by equation 2.27) is used. Yet, the smearing or Gaussian width (a parameter controlling how much the force is spread) does not have a closed form. In *turbinesFoam*, three different criteria are used to determine it:

— **Load distribution**. The projection width for a lifting surface should be based on the load distribution, which is closely related to the local chord. Width values around $14 - 25\%$ of the chord length are recommended.

— **Drag-induced wake**. The projection width should fall within the order of the momentum thickness θ, which for a fluffy body is related to the drag coefficient by $\theta = \frac{1}{2} C_d \, c$, where c is the local chord.

— **Mesh size**. The projection width should be at least twice the local grid size.

All three are values computed for each AE at every time step, with the largest being chosen for the force projection step. The library allows the user to control each of the three criteria independently through the `chordFactor`, `dragFactor` and `meshFactor` coefficients that can be defined in the `GaussianCoeffs` sub-dictionary. Variable projection widths allow for more confined loads in finer meshes while protecting coarser grids from numerical instabilities.

By default, the inflow velocity at each AE is obtained at the quarter-chord position by means of OpenFOAM's class `interpolationCellPoint`. This helps keep the velocity smooth as the actuator lines rotate (and cross different cells), compared with using the cell values themselves. Yet, the user can make use of an integral velocity approach that uses a circle around the quarter-chord position as a sampling zone. The user can define the size of this circle via the `velocitySampleRadius` entry. The sampling radius is then obtained by multiplying the latter value by the projection width. The number of integration points can also be specified by the user with the `nVelocitySamples` tag.

The library also features a Leishman–Beddoes type dynamic stall model, flow curvature, added mass, and end-effects corrections (with several options available). The Glauert (or Prandtl) end-effect factor f is meant to reduce the loading near the blade ends. It is a function of the inflow angle φ and distance to the blade tip (or root) d. The end effect factor is given by equation 6.1, where N_B is the number of blades.

$$f_T(\varphi, r) = \frac{2}{\pi} \arccos \left[\exp \frac{\frac{N_B}{2}\left(1 - \frac{1}{d}\right)}{\sin(\varphi)} \right] \tag{6.1}$$

6.1.2. Library structure

The library follows a logical hierarchical structure where the turbine is defined as a collection of actuator line sources, each encompassing a set of actuator elements. The three main classes are:

— `actuatorLineElement`. Is the most fundamental class upon which the library is constructed. It samples the velocity at the quarter-chord and projects the resulting aerodynamic load. This class relies on the `profileData` class used for accessing, interpolating, and correcting the aerodynamic polars.

— `actuatorLineSource`. It generates the desired array of AEs that defines the actuator line. They can be used in isolation when no rotation motion is desired.

— `turbineALSource`. The turbine is defined as a set of actuator lines. The abstract class is used by the `axialFlowTurbineALSource` and `crossFlowTurbineSource` classes, each with their own definition of the rotation motion. This motion only applies to blades, while the tower, nacelle, and hub are defined as steady actuator lines. Overall turbine loads are computed within these classes.

6.1.3. Usage

Although no official guide is available, the library includes a set of basic tutorials to help the user set-up his or her first case. Even though it may seem tough at first, the user is encouraged to go through the highly commented, well-structured source code (at least the headers) to understand the meaning and purpose of the different variables. In most cases, all the information needed by *turbinesFoam* will be contained within these files:

— `fvOptions`. Main (mandatory) dictionary for the source term definition. The turbine type, geometry, and properties are declared here, as are the different correction methods to be applied.

— `topoSetDict`. The geometrical region enclosing the turbine (where source terms will be located) must be defined here as a *cellZoneSet*. Its name should correspond to the one used in `fvOptions`.

— `elementData`. Optional file where blade element data (chord, twist, etc.) distributions are defined. Since these properties are given at specific positions or stations, values in between are interpolated. Two consecutive stations define a span section containing an integer number of uniformly spaced AEs.

— *foilData*. Optional folder where the airfoil polars (aerodynamic coefficients as a function of the angle of attack for one or more values of Re) are stored.

6.2. Adapting *turbinesFoam* to floating motions

Even though *turbinesFoam* has proven adequate for conventional (fixed) VAWTs, at the moment of writing this report, it is not compatible with any type of turbine motion. The basic concept behind adapting the ALM to floating conditions is simple: update the actuator line position and relative velocity according to the platform and rotor motions. Yet simple does not always mean easy, and up to a thousand new lines were added to the original library to account for this floating motion. To avoid confusion with the original code, the new library will hereinafter be referred to as *modified turbinesFoam*.

Because *turbinesFoam* was verified using VAWTs under simple conditions, some minor bugs were reported and corrected[1], the most important being related to how aerodynamic twist is applied to the blades. Apart from that, a post-processing script was written to visualize the actuator lines, which turned out to be very handy for verification purposes when the floating motion was applied.

A technical description of the implementation is beyond the scope of this thesis, since the additions to the original library will eventually be made available through GitHub. The aim of this section is to give an overview of the approach followed and show the reader how to set-up a basic case involving a moving turbine.

6.2.1. Concept

As in many other engineering problems, it is wise to break down a complex problem into simpler components. The ALM uses a set of actuator lines to represent the FOWT, which are in turn composed of a set of individual actuator line elements. As seen in the previous section, each one corresponds to a different class and thus has its own methods and attributes. For that reason, the implementation is divided into these three building blocks.

Floating actuator elements

Actuator elements are the basic building blocks of the ALM, since their position and velocity ultimately determine the force field to be included as a source term in the Navier-Stokes equations. The most relevant change floating motions entail is the additional relative velocity due to the FOWT's extra degrees of freedom. This *floating* velocity term, $\mathbf{u}_f$, is computed at the position of each AE and included in the computation of the total relative velocity, $\mathbf{u}$:

$$\mathbf{u} = \mathbf{U} - \mathbf{u}_r - \mathbf{u}_f \tag{6.2}$$

Where $\mathbf{U}$ is the inflow velocity at the AE position and $\mathbf{u}_r$ is the element velocity due to the turbine rotation motion. Regarding the motion of the AE, different methods were created to translate the element and rotate it around the desired CoR. The rotation methods are slightly different depending on whether they are used for rotor or floater rotations: the former uses an axis-angle representation, whereas the latter receives a rotation matrix as input (see Appendix 8.3 for a description of these methods). Either way, AEs do not move in isolation but rather in groups called actuator lines (ALs), thus the floating motion routines are developed in the respective class.

[1] There appears to be a bug with OpenFOAM's `findCell` method. When the simulation is run in parallel, the library might return the error *"Inflow velocity point not found in mesh"*, even though the point is clearly inside. If this ever happens, try changing the number of processors or modifying the mesh.

Floating actuator lines

Because AEs are joined into actuator lines, it makes sense to associate a single motion to each AL, which will then apply to every AE contained within. The motion definition for a specific time instant t^{n+1} is given by four different parameters:

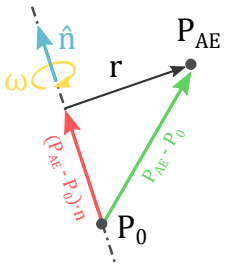
1. Displacement $\Delta\mathbf{d} = \mathbf{d}^{n+1} - \mathbf{d}^{n}$.
2. Rotation matrix of the floater body in the inertial frame $\underline{\mathbf{R}}_f^{n+1}$, which is equivalent to the transformation matrix from floater to inertial frames $\underline{\mathbf{Q}}_{fi}^{n+1}$.
3. Linear velocity $\mathbf{v}^{n+1}$.
4. Angular velocity $\boldsymbol{\omega}^{n+1}$.

At this point, the reader is encouraged to read section 8.3 for clarification on the geometric definitions. Velocities and displacements are needed in the inertial frame, even though a transformation matrix can be used to map between the floater and inertial frames. Given the first two parameters, the position of each AE contained inside the considered AL can be updated according to equation 3. Note that before applying the new rotation, the AE must first be returned to its original, un-rotated configuration. This means that the total rotation matrix in equation 3 must be computed as:

$$\underline{\mathbf{R}} = \underline{\mathbf{R}}_f^{n+1}\ \underline{\mathbf{R}}_f^{T\ n} \tag{6.3}$$

Finally, the AE motion velocity is computed from its linear and angular components:

$$\mathbf{u}_f\ = \mathbf{v}^{n+1} + \boldsymbol{\omega}^{n+1} \times \mathbf{r}^{n+1} \tag{6.4}$$



Where $\mathbf{r}^{n+1}$ is the radii vector between the AE position $\mathbf{p}_{AE}$ and the instantaneous rotation axis defined by the angular velocity. Given the unit vector $\hat{\mathbf{n}} = \boldsymbol{\omega}/\left\|\boldsymbol{\omega}\right\|$ and an arbitrary point contained along the instantaneous rotation axis $\mathbf{p}_0$, the rotation radius is computed as:

$$\mathbf{r}\ = \mathbf{p}_{AE} - \left\{\mathbf{p}_0 + \ \left[\left(\mathbf{p}_{AE} - \mathbf{p}_0\right) \cdot \hat{\mathbf{n}}\right] \hat{\mathbf{n}}\right\} \tag{6.5}$$

Where the superscript $n+1$ has been omitted for clarity. So far, the motion has been kept generic, meaning that the source of the aforementioned motion parameters has not been taken into account. However, it is not difficult to imagine how these could be computed for a prescribed motion. In the case of a pure harmonic motion:

$$\begin{aligned}
\Delta\,\mathbf{d} &= \mathbf{A}_\mathrm{d}\left[\sin\left(\mathbf{w}_\mathrm{d}\ \mathrm{t}^{n+1}\right) - \sin\left(\mathbf{w}_\mathrm{d}\ \mathrm{t}^{n}\right)\right] \\
\boldsymbol{\phi}^{n+1} &= \boldsymbol{\phi}_0 + \mathbf{A}_\phi\ \sin\left(\mathbf{w}_\phi\ \mathrm{t}^{n+1}\right) \\
\mathbf{v}^{n+1} &= \mathbf{w}_\mathrm{d}\,\mathbf{A}_\mathrm{d}\ \cos\left(\mathbf{w}_\mathrm{d}\ \mathrm{t}^{n+1}\right) \\
\boldsymbol{\omega}^{n+1} &= \mathbf{w}_\phi\,\mathbf{A}_\phi\ \cos\left(\mathbf{w}_\phi\ \mathrm{t}^{n+1}\right)
\end{aligned} \tag{6.6}$$

Where **A** and **w** are the amplitude and angular frequency of the harmonic motions and should be given as input by the user. The subscript d refers to linear motion, whereas ϕ does to angular. They are defined as vectors since they contain three different entries, one for each DoF. The Euler angles $\boldsymbol{\phi}^{n+1}$ are transformed into the rotation matrix $\underline{\mathbf{R}}_f^{n+1}$ by equation 7.

Floating wind turbines

Different ALs can be coupled together to describe a FOWT, each one representing a different component: not only the blades but also the tower, hub, and nacelle can be modeled through ALs [47]. Recall that only vertical axis FOWTs are considered in the present work. When the turbine moves with the floater, all ALs perform the same floating motion as defined in the above section. Blades, however, are a special case, since they also rotate with respect to the floater. This rotation results in the second velocity term of equation 6.2, which is already taken into account in the original library by means of the axis-angle representation, see Rodrigues' formula 9.

In addition, as the floater translates and rotates, so do the position and rotation axis of the rotor blades (i.e., the rotor frame). Hence, they both need to be updated according to the floater motion. As a final note, the implemented prescribed motion routine was verified against a MATLAB script (see Appendix 8.3), which was easier to control, adjust, and debug.

6.2.2. Usage

With the presented implementation, prescribed motions can already be applied to both individual actuator lines (`actuatorLineSource`) and axial flow turbines (`axialFlowTurbine`). Currently only the pure harmonic motion type presented in equation 6.6 is available. Despite its simplicity, it is still a powerful tool for validating the performance of the FOWT under various prescribed motions. The frequency and amplitude of the motion can be controlled in each DoF independently, thus allowing both single- and multiple-DoF motions. To activate the prescribed harmonic motion, the user must define a new sub-dictionary called `prescribedMotion` inside `fvOptions`:

```
axialFlowTurbineALSourceCoeffs
{
        // ...
        prescribedMotion
        {
                active                  on;
                rotationCenter          (0 1 -1); //Initial position of the CoR [m]
                initialRotation         (0 15 30); // (roll pitch yaw) [deg]
                motionType              harmonic1st;
                harmonic1stCoeffs
                {
                        translationAmplitude    (0.25 0.25 0.25); // (surge sway heave) [m]
                        translationFrequency    (2 2 2); // [Hz]
                        rotationAmplitude       (45 15 30); // (roll pitch yaw) [deg]
                        rotationFrequency       (2 1 0.5); // [Hz]
                }
        }
}
```

The initial orientation of the floater is given by a set of Tait–Bryan angles $\boldsymbol{\phi}_0$.

6.3. The OC6 Phase III campaign

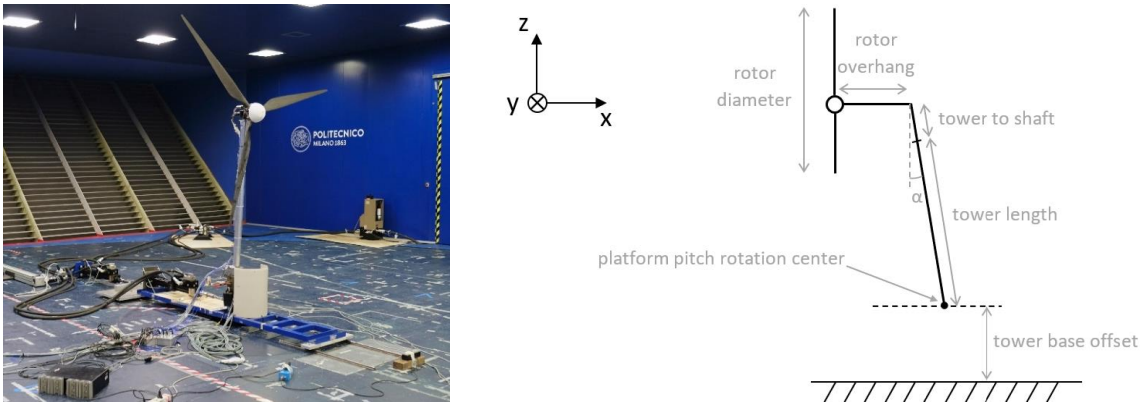
Contrarily to its predecessors, which focused on the non-linear hydrodynamics of the FOWT platform (see section 3.3), phase III of the OC6 campaign focused on the aerodynamic performance of a turbine under prescribed 1-DoF motion. The aim was to validate the aerodynamic loading on the turbine obtained with various fidelity tools by comparing them with the measurements from the *UNsteady Aerodynamics for FLOating Wind* (UNAFLOW) experimental campaign. The latter was performed at the *Boundary Layer Test Section* of the *Politecnico di Milano* wind tunnel, where the system was forced to oscillate in the surge and pitch directions using varying frequencies and amplitudes.

It was observed that the turbine motions introduced a variation in relative wind speed that impacted the rotor loads (thrust and torque), but still did not trigger an unsteady aerodynamic response. Overall, good agreement was found between the participants and the experimental results, with no significant differences between models of different fidelity.

6.3.1. Turbine definition

The turbine used in the experimental campaign was a 1:75 scaled version of the DTU 10-MW RWT [143]. The scaling aimed at preserving the thrust and power coefficients while reducing the physical dimensions by a factor of 75, the resulting physical properties being indicated in Figure 6.1b and Table 6.1. Because of the reduced size, the Reynolds number characterizing the flow field is also smaller (by a factor of 225), and special airfoils for such conditions (10% thick version of the SD7032) were used. Finally, the desired aerodynamic coefficients were achieved by modifying the blade pitch and chord distributions along the span.

Blade properties and aerodynamic coefficients were provided in 20 radial stations along the blade, as summarized in Table 1 from the OC6 Phase III paper [5]. The aerodynamic center is coincident with the blade pitch axis, and no aerodynamic moment coefficient is considered. The lift and drag coefficients



(a) Set-up for the UNAFLOW experiment 1

(b) Wind turbine geometry and coordinate system

Figure 6.1: The 1:75 scaled DTU 10-MW RWT in the Politecnico di Milano wind tunnel. Reproduced from [5].

were obtained (for 128 values of angle of attack) from two-dimensional sectional-model experiments conducted in the DTU Red wind tunnel for seven values of Reynolds number, ranging from 0.05 to 0.2 million. These polars were provided to the participants in a MATLAB file that already included the rotational augmentation corrections for 3D-delayed stall. Because the Reynolds number varies along the span, participants could choose between using interpolation routines or using a specific Reynolds value at each station as proposed by the campaign definition document.

Rotor diameter [m]	**Blade length** [m]	**Hub height (exp. 1)** [m]	**Hub height (exp. 2)** [m]
2.38132	1.10166	2.086	2.188

Table 6.1: Rotor properties from the UNAFLOW experimental campaign.

The blades were straight, without cone angle, and rigid. The tower, which had a diameter of 75 mm, was considered rigid and included a negative tilt angle of 5 degrees to counteract the wind turbine tilt, resulting in a rotor perpendicular to the ground. The scaled turbine also included a hub-nacelle assembly, which in general was not modeled by the participants.

6.3.2. Experimental setup

The experiment took place in the *Boundary Layer Test Section* of the *Politecnico di Milano* wind tunnel. The facility was 13.84 m wide × 3.84m high × 35m long. The wind turbulence intensity in the region covered by the rotor was close to 2% and the wind speed was constant over the rotor-swept area [5]. The air density was considered equal to 1.177 kg/m^3. A free-stream velocity of $U_\infty = 4.19\,m/s$ was chosen as representative for the DTU 10-MW turbine at model scale. This value is already corrected for wind tunnel blockage. The relatively small distance between the blade tip and the ceiling (compared to the floor) might affect the wake expansion in the vertical direction.

Two different experiments were carried out:

— **Experiment 1**. Hydraulic actuators were used in the first to induce the unsteady (surge only) conditions while the rotor was kept rotating at constant speed. Depicted in Figure 6.1(a).

— **Experiment 2**. A 6-DoF robot was used in the second to induce both surge and pitch motions. The pitch rotation center was defined 0.2666 m behind and 1.458 m below the rotor hub. Wake data was not recorded. Analytical post-processing corrections were needed because the controller induced rotor-speed oscillations.

The system geometry also changed between experiments, the main difference being the hub height (see Table 6.1). The experiments recovered mainly three types of data:

— **Hub loads**. Thrust force and torque moment though 6-DoF load cells.

— **Blade aerodynamics**. Radial distributions of normal and tangential forces, relative wind speed, angle of attack, and aerodynamic coefficients.

— **Wake characteristics**. Via PIV measurements and two sets of hot-wire probes at hub height, namely along-wind ($x = 5.48$m, $y \in [-1.6, 1.6]$m) and cross-wind ($x \in [2.18, 5.48]$m, $y = 0.9$m).

The thrust is defined in the local frame (load component perpendicular to the rotor plane), which will rotate during pitch motions. Load measurements were post-processed to remove all inertial components so that the loads being compared are purely aerodynamic. A 3 Hz low-pass filter was used in the post-processing of the results to remove 1P and 2P oscillations due to mass and aerodynamic imbalances, respectively. This filter also removed the tower shadow effect, and accordingly, participants did not include the tower influence in their models. No uncertainty information was made available for this campaign.

6.4. ALM simulation of the OC6 Phase III experiment

With the aim to test the prescribed motion capability implemented in section 6.2, the OC6 Phase III experimental campaign will be recreated through an ALM simulation in OpenFOAM.

6.4.1. Scope and limitations

The purpose of the numerical simulations that will follow is to serve as a proof of work for the modified *turbinesFoam*. As of now, only the positions and relative velocities of the AEs have been verified. Whether the resulting aerodynamic loads make physical sense is still unknown.

Even though results will be compared with experimental measurements, the reader should not consider this a validation task. Setting up a simulation that can actually match the experiment is an ambitious endeavor that deserves its own dedicated thesis. Here, no spatial nor temporal convergence studies will be performed. Instead, the various numerical parameters will be chosen based on the available literature.

Given that the scope of the present project covers not only the turbine but also its coupling with the platform, the available time to dedicate to this section is limited. For this reason, a relatively coarse mesh will be used along with a RANS turbulence model in an attempt to reduce the computational budget, even though it compromises accuracy. As will be seen when analyzing the results, this will considerably affect the wake, in which vorticity will not be successfully captured.

Yet, this should not conflict with the sought purpose: to evaluate the modified *turbinesFoam*'s capability of producing physically-consistent results when a prescribed motion is imposed, focusing on the differences between steady and floating conditions.

6.4.2. Investigated load-cases and metrics

Up to 92 different tests were carried out during this campaign. In the end, only seven were considered for validation purposes: one for steady (1.1) and three for surge (2.X) and pitch (3.X) conditions each. The amplitudes and frequencies of the prescribed motions were chosen such that the resulting reduced frequencies were preserved in representative FOWTs.

For the prescribed motion cases, a total of five periods will be simulated since the author heuristically found that the first two correspond to a transient start. The latter three will be averaged and used to obtain the desired metrics. Because of this, the case with the lowest frequency will not be considered since it would result in longer simulation times. For the steady case, 10 seconds of physical time will be simulated, and results will be averaged from the last two. All in all, a total of five cases will be recreated in OpenFOAM, as summarized in Table 6.2.

All cases were carried out with an inflow velocity $U_\infty = 4.19\, m/s$ and constant rotor speed $\Omega = 240$ rpm. Two different cases are considered in surge and pitch to analyze the effect of the oscillation frequency (and thus floater-induced velocity). Cases 2.5–3.5 and 2.7–3.7 have very similar prescribed velocities, so the effect of the motion type is better identified.

The different metrics that will be extracted for comparison with the experimental measurements vary between steady and unsteady cases. For the steady-rotor case, average thrust and torque will be the main metrics for hub loads. For the wake, velocity distributions at probe locations and the average wake deficit will be used. The averaged wake deficit within a two-dimensional domain (rotor region) is

defined as:

$$\Delta \mathrm{U}_{\mathrm{avg}} = \frac{\int_0^{2\pi} \int_0^R r \cdot \left(\mathrm{u}_{(\theta,r)} - \mathrm{U}_\infty\right) dr\, d\theta}{\pi R^2} \tag{6.7}$$

For wake visualization purposes, isosurfaces of the Q-criterion will be used. It is a scalar field obtained from the spatial gradient of the velocity components as shown in equation 6.8 (subscripts on the right-hand side represent differentiation). The Q-criterion can be thought of as the relative dominance of the rotational over stretching components, which is very helpful in representing the vortical structures of the flow.

$$Q = -\frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} = -\frac{1}{2}\left(u_x^2 + v_y^2 + w_z^2\right) - \left(u_y v_x + u_z w_x + v_z w_y\right) \tag{6.8}$$

For the unsteady cases, only hub-load (thrust and torque) metrics will be extracted. These include phase-averaged and peak-to-peak values. Also, the phase difference between the loads and rotor motion will be obtained by means of a sine cross-correlation analysis.

LC	Experiment	Platform motion			
		Direction	Frequency [Hz]	Amplitude [m] / [deg]	Speed [m/s]
1.1	1	None	-	-	-
2.5	1	Surge	1.0	0.035	0.22
2.7	1	Surge	2.0	0.008	0.10
3.5	2	Pitch	1.0	1.4	0.21
3.7	2	Pitch	2.0	0.3	0.09

Table 6.2: Selection of UNAFLOW load cases considered for their CFD numerical simulation.

6.4.3. Numerical set-up

The modified *turbinesFoam* library will be used within an OpenFOAM simulation to recreate the UNAFLOW experimental conditions.

Turbulence model

Contrarily to the previous hydrodynamic simulations, where the flow-field could be assumed laminar without major impact on the overall loads [35, 129, 110], turbulence has a strong effect on wind energy applications and must be properly modelled. Given the scope of the present project, URANS is the most adequate modeling approach given its reduced computational cost compared to LES and its solid capability to resolve the mean flow features, which are of prime interest in the sought application.

Because no physical walls are present in the ALM simulation, the $k-\varepsilon$ Reynolds-average stress (RAS) model will be used. It assumes equilibrium between turbulent production and dissipation and is thus tailored to the flow far away from boundary layers. With this model, two new differential equations must be solved: one for the turbulence kinetic energy k and a second for the turbulence dissipation rate ε. All model constants that need to be defined for the new pair of differential equations are set to their default value in the *RAS* field from the `turbulenceProperties` dictionary.

In OpenFOAM, we need to set the initial internal field conditions for both k and ε. The first one can be easily linked to the turbulence intensity:

$$I \equiv \frac{u'}{\|\mathbf{U}\|} = \frac{\sqrt{\frac{1}{3}\left(u'^2_x + u'^2_y + u'^2_z\right)}}{\|\mathbf{U}\|} = \frac{\sqrt{\frac{2}{3}k}}{\|\mathbf{U}\|} \rightarrow k = \frac{3}{2}\left(\|\mathbf{U}\|\, I\right)^2 \tag{6.9}$$

Where U is a reference velocity (e.g. freestream) and I is the turbulence intensity, which in the experimental campaign was registered to be $I = 2\%$. The turbulence dissipation rate is estimated by assuming isotropic turbulence:

$$\varepsilon = \frac{C_\mu^{3/4}\, k^{3/2}}{L} \tag{6.10}$$

Where $C_\mu \sim 0.09$ is a model constant and L is a reference length scale, chosen equal to the blade span.

Geometry and grid

In an attempt to simplify the problem and reduce computational costs, the boundary layer (BL) region near the section walls won't be simulated. Fine meshes and wall models are needed to capture the effect of BLs, requiring some expertise and experience with the various models. Moreover, the used $k-\varepsilon$ turbulence model is not suited for wall-bounded, anisotropic flows. Because the observed BL thickness of the experimental campaign was made available to all participants $\delta \sim 0.125$m, its width will be subtracted from all four walls and a velocity slip condition will be applied instead.

Following the procedure from Negroni [101], who also dedicated his MSc thesis to the ALM simulation of the UNAFLOW experiment, the domain length will be extended to a total of $16D$ with the turbine located $4D$ from the inlet. This reduces the rotor induction effect on the inlet and ensures a far-wake condition at the outlet. All in all, the numerical domain is 13.59 m wide × 3.59 m high × 47.6264 m long. For the first experiment (steady and surge conditions), the rotor hub is located at $(0, 0, 1.961)$m. For the second setup (pitch motion) the hub height is 2.063 m, while the rotation center is defined behind the turbine at $(0.2666, 0, 0.605)$m.

To reduce the element count, the cell aspect ratio increases toward the x and y boundaries, given that no BL refinement is needed. This especially applies to the outlet boundary, where numerical dissipation due to larger cells does not affect the upwind turbine. Still, a region with uniform square elements is maintained, which encompasses $x \in [-2D, 7D]$ and $y \in [-1.5D, 1.5D]$. This is done to ensure a good-quality mesh in the rotor and wake surroundings.

The base mesh is defined through the `blockMesh` utility and is made parametrically so that the user only needs to specify the desired cell width in the uniform region, Δx_0. From here, three zones with increasing refinement levels are defined by means of the `snappyHexMesh` utility:

— **Zone 1**. A refinement level of 1 is applied to a box within $x \in [-1.5D, 9D]$, including the 65% of the cells along the lateral direction and the 80% along the vertical. The aim of this first zone is to increase accuracy between the upwind and far-wake regions while leaving the boundaries unrefined.

— **Zone 2**. A refinement level of 2 is applied to a cylindrical region along x with a diameter of $1.8\,D$ that extends within $x \in [-0.5D, 2.5D]$. The goal is to better capture the structure of the near wake. The cylindrical shape should result in fewer cells compared to a rectangular box approach. The numerical gauges replicating hot-wire measurements are all located within this zone.

— **Zone 3**. Finally, a refinement level of 3 is applied to a short cylinder that encompasses the rotor. With a diameter of $1.25\,D$ extended along $x \in [-0.25D, 0.25D]$, the purpose of this region is the cell size reduction close to the rotor where ALM loads are to be calculated and applied.

For the cases with prescribed motion, the boundaries of zone 3 should be enlarged to account for the possible rotor motions. For surge, zone 3 was extended in both positive and negative x by the surge motion amplitude. In pitch, this is more intricate since the turbine moves in both x and z and is no longer perpendicular to x, so a cylindrical zone may no longer be the most suitable shape. Luckily, the amplitude of the pitch oscillations is sufficiently small for zone 3 to still contain the turbine trajectory. Figure 6.2 shows the rotor as it reaches its maximum and minimum pitch amplitudes ($\theta_{\text{max}} = 1.4^{\text{o}}$, x and z not to scale). The dotted red box encompasses the physical space where the turbine will move, which is safely contained within zone 3 (represented by the continuous red box).

At this point, the mesh will be completely defined given the cell width of the base mesh, Δx_0. The latter is related to the cell width at the rotor region (zone 3) by expression 5.7, $\Delta x_0 = 2^3 \Delta x_3$. Based on current literature on ALM-RANS simulations, Carlos Dos Santos reported in his MSc thesis [144] that the minimum grid length at the turbine should be within 0.8 and 2.4% of the rotor diameter. In an attempt to balance computational cost and accuracy, a turbine grid width $\Delta x_3 = D/120$ has been chosen, resulting in a base mesh cell size $\Delta x_0 = D/15$. Figure 6.3 shows various snapshots of the mesh, which for visualization purposes has been made coarser than the actual simulation grid ($\Delta x_0 = D/7.5$).

All in all, the cell count of the resulting mesh is around ~ 3.5M. Still, the used refinement will not be enough to capture the shed vorticity and wake, not even with a larger zone 3 that covers all the near wake region. As explained in section 6.4.1, the aim of the present analysis is not a strict validation but

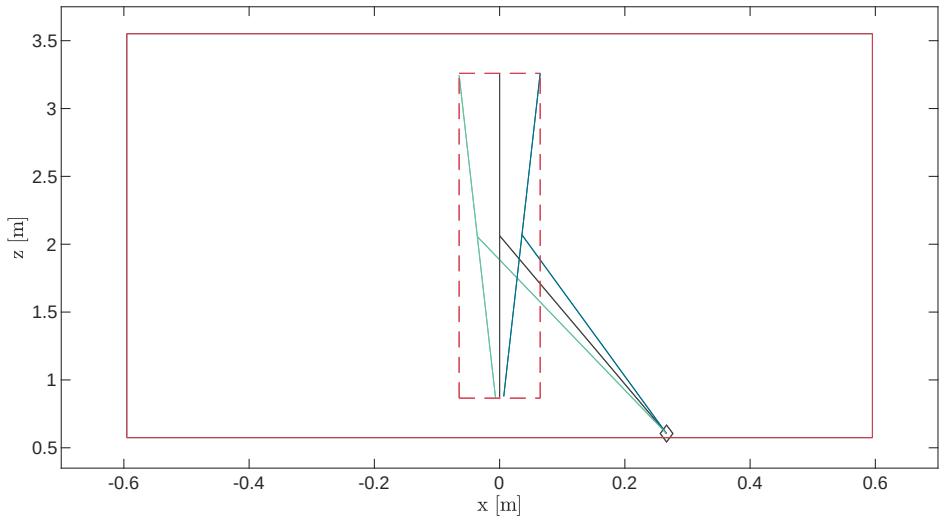


Figure 6.2: Scheme of the turbine geometry as it pitches, with $\theta_{\text{max}} = 1.4^{\text{o}}$ and motion phase $\varphi = 0$ (black), 90 (blue) and 270^{o} (green).

rather a proof of work for the implemented library. The proposed mesh has to be seen as a compromise between accuracy and computational cost.

ALM parameters

The *turbinesFoam* library allows the user to modify various numerical parameters. The most important is the rotor's discretization into elements. Given that the OC6 Phase III campaign divided the blade into 20 stations (i.e., 19 segments), two AEs will be defined between stations, resulting in a total of 38 AEs per blade, with a length $\Delta r \sim 0.029$ m. This means that there will be one AE for every ~ 1.46 grid elements, which falls within the recommended 1-2 range. The properties (chord, chord mount, twist, aerodynamic polars) at each element position are interpolated from the two closest blade stations. The aerodynamic polars are given with multiple Reynolds numbers, and the coefficients are obtained by interpolation with the element's local Re.

By default, the inflow velocity value for ALM load calculation is sampled from the closest cell to the AE, located at the quarter-chord position. This, however, results in a *noisy* inflow velocity, since there's a small jump every time the blade crosses one element. Instead, an integral sampling method will be used where the inflow velocity is obtained through the integration of the velocity field weighted by a projecting function of circular shape. The user can define the sampling radius and the number of samples. However, the library uses the smearing width ϵ to scale the chosen radius, making the integration region considerably smaller. In the end, a sampling radius $r_s = D/10$ is used, which after the ϵ-scaling results in a circular radius of $\sim D/100$. A total of 20 velocity samples will be taken within the specified domain.

Finally, end-effects are enabled that reduce aerodynamic loads towards the blade tip and root. Glauert's tip-root correction model (described in section 6.1) will be used for this purpose. To quantify the effect of such corrections, case L1.1 will be run twice: once with (L1.1a) and once without (L1.1b) tip/root corrections.

Time discretization

Regarding the time-step selection, a special treatment is needed involving the turbine rotation. Since blades are modeled through ALM rather than being physically simulated, their rotation velocity is not inherently accounted for in the Courant number. Because the highest CFL number is expected at the blade tip (given the high velocity and small grid size), we can make use of the TSR to define a new CFL number definition that includes the turbine rotation effect:

$$\text{CFL}_{\text{rot}} = \frac{\sqrt{{U_r}^2 + (U_\infty \cdot \text{TSR})^2}\,\Delta t}{\Delta x} \tag{6.11}$$

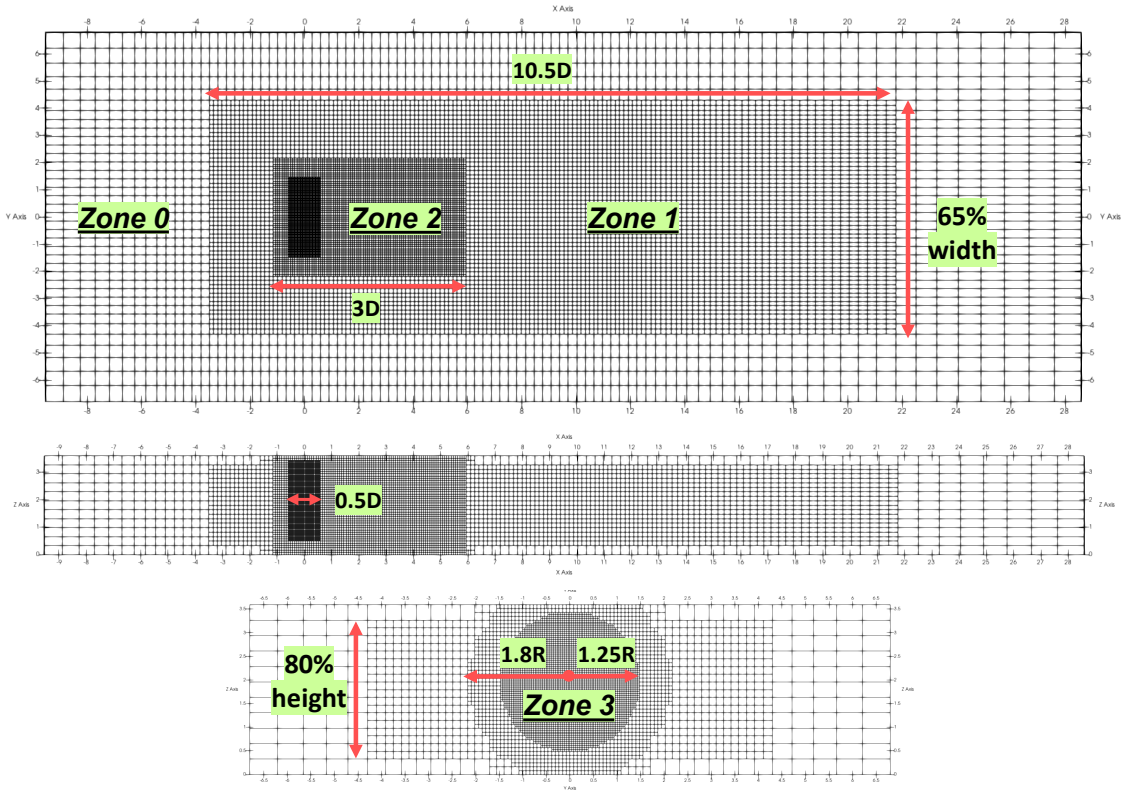


Figure 6.3: Mesh for the UNAFLOW experiment 1 (with $\Delta x_0 = D/7.5$), along with its main zones and dimensions.

From momentum theory (equation 3.1), the wind speed at the rotor plane U_r is related to free-stream through the induction factor. Assuming $a = 1/3$ yields:

$$U_\infty = \frac{1}{1-a} U_r = \frac{3}{2} U_r \;\rightarrow\; \text{CFL}_{\text{rot}} = \frac{U_r \, \Delta t}{\Delta x} \sqrt{1 + 9/4 \, \text{TSR}^2} \tag{6.12}$$

From the above equation, we can identify the first term in the right-hand side as the conventional definition of the CFL number that we use in OpenFOAM, leading to:

$$\text{CFL} = \frac{1}{\sqrt{1 + 9/4 \, \text{TSR}^2}} \, \text{CFL}_{\text{rot}} \tag{6.13}$$

For instance, a maximum $\text{CFL}_{\text{rot}} = 0.8$ and $\text{TSR} = 5$ would result in the highly restrictive maximum $\text{CFL} \sim 0.076$. Hence, we must expect small values of Δt when dealing with turbine simulations. This expression however does not include the effects of the platform motions, which will be neglected for the present analysis.

Another formulation based on a maximum Δt can be derived, which might turn out to be useful for solvers that do not adapt the time-step based on the maximum CFL number. The key idea is that the actuator lines should not cross more than one grid element per time-step, otherwise, the ALM force won't be evenly distributed along the rotor azimuth. Again, the most critical situation occurs at the blade tip, where the mesh size is Δx_3:

$$\Delta t \leq \frac{\Delta x_3}{U_\infty \, \text{TSR}} \tag{6.14}$$

Given that the grid size at the rotor is given as a fraction of its diameter, $\Delta x_3 = f_D \, D$:

$$\Delta t \leq \frac{f_D \, D}{U_\infty \, \text{TSR}} = \frac{2 \, f_D}{\Omega} \tag{6.15}$$

As expected, the higher the rotation speed, the lower the time-step must be. A finer mesh in the turbine zone will also result in more restrictive time-steps. For the present analysis, $f_D = 1/120$ and $\Omega = 240$ rpm, resulting in $\Delta t \leq 0.66$ ms. This is a fairly small value for RANS simulations and will considerably increase the computational requirements of the simulations.

Boundary conditions
The domain is confined by six boundaries: one inlet, one outlet, and four walls. All BCs are listed in Table 6.3. Because the BL has been removed from the domain, a slip condition will be imposed on all walls, and no wall functions will be used for k or ε. Removing the BL also reduced the cross-area of the domain, and thus the inlet velocity corresponds to the inflow speed already corrected for blockage, $U_\infty = 4.19\, m/s$. To account for any reverse-flow situation occurring at the outlet, an *inletOutlet* BC is imposed for the velocity and turbulent quantities. The values of k_0 and ε_0 at the inlet are the same as for the initial internal field, given by equations 6.9 and 6.10.

	U	**P**	**k**	**ε**
Inlet	fixedValue: U_∞	zeroGradient	uniformFixedValue: k_0	fixedValue: ε_0
Outlet	inletOutlet	fixedValue 0	inletOutlet	inletOutlet
Walls	slip	zeroGradient	zeroGradient	zeroGradient

Table 6.3: Boundary conditions for the velocity, pressure and turbulent quantities.

Schemes and solvers
The numerical schemes and solvers are taken from the *axialFlowTurbine* tutorial from *turbinesFoam*, with minor modifications. First-order schemes (`linear` and `upwind`) are used to ensure stability, even though they might increase numerical diffusion. Regarding the solvers, the main difference with the tutorial is the choice of a *PBiCGStab-DILU* solver for the velocity and turbulent quantities. The PIMPLE algorithm is used in PISO mode, with a total of three inner correctors and one non-orthogonal corrector step. Turbulent quantities are solved at every iteration.

All load cases were run in TU Delft's high-performance cluster using 48 AMD Opteron (2.4 GHz) processors. Using the $\sim$ 3.5M elements mesh, 38 AL elements per blade, and the constant time-step approach defined above, 10 seconds of physical time required $\sim$ 39 hours of simulation walltime.

6.5. Results and discussion

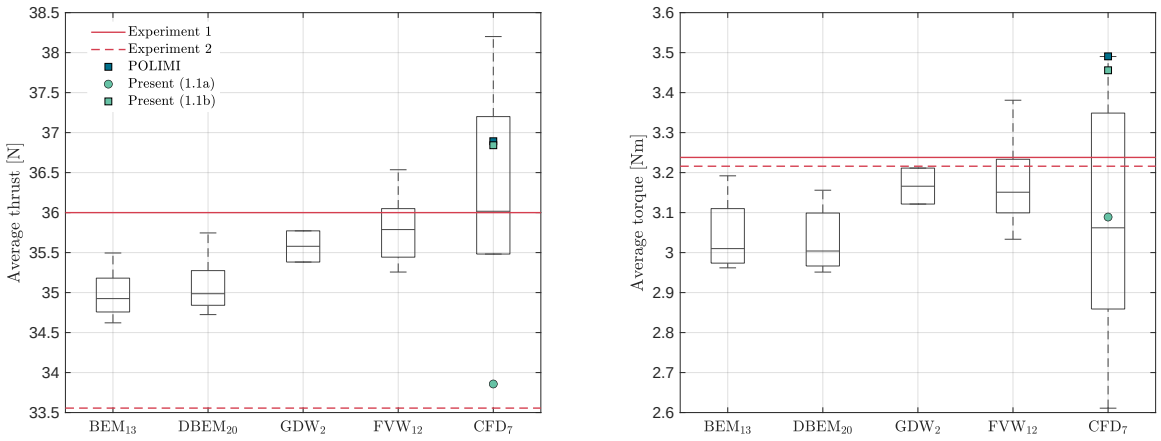
This section will present the main results from all the considered load cases and compare them with the experimental measurements. From all OC6 Phase III participants, results will also be compared with those from POLIMI, the only participant that used their in-house ALM within OpenFOAM as a modeling approach. For comparison purposes, the present results will be referred to as FALM (floating-ALM).

6.5.1. Steady turbine

The goal of this first case is to ensure that the aerodynamic model has been correctly implemented before it is used in unsteady conditions. The same case has been run twice, with (L1.1a) and without (L1.1b) Glauert's tip-root corrections. Hub-loads (thrust and torque) were averaged from the eight last blade rotations, from a total simulation time of 10s.

Average loads
Figure 6.4 presents a summary of the hub loads, along with experimental and numerical results based on the modeling approach (model fidelity increases to the right). The subindex next to each modeling technique indicates the number of available results from the participants. It can be observed that the experimental value differs from the first to the second experiment. This is thought to be caused by either the influence of the sensors' cable bundle or a small blade pitch angle offset [5].

Note how CFD results have the highest standard deviation of all considered techniques, with the outliers falling outside the experimental range. Yet, FALM results without tip-root corrections (1.1b) fall within the expected CFD range, very closely matching those from POLIMI for both thrust and torque. As expected, adding these corrections (1.1a) lowered the magnitude of hub loads. Thrust is no longer in



(a) Aerodynamic rotor thrust

(b) Aerodynamic rotor torque

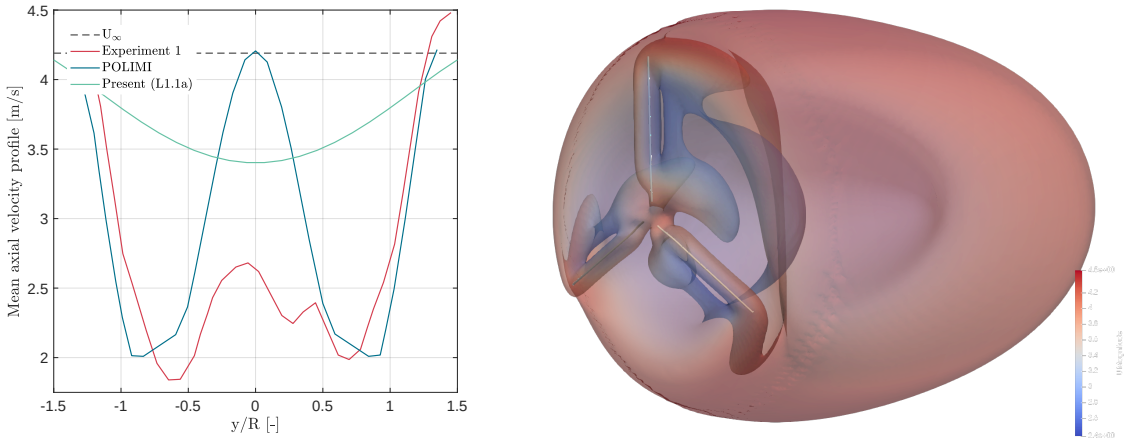
Figure 6.4: Summary of the OC6 Phase III hub loads for the steady-wind condition, compared with those from the present project. Black boxes represent numerical results from the participants based on the modelling approach.

the CFD range but now falls within the experimental range, although it is the lowest value across all techniques. When it comes to torque, the predicted value falls outside of the range of the wind tunnel observations but is now more aligned with the other techniques.

In general, the obtained hub loads fall within the range that is expected for the numerical models. Without uncertainty analysis, it is difficult to make more compelling statements regarding which model made the best prediction.

Wake

In the experiment, the wake profile was measured in the crosswind direction by a set of hot-wire probes. The measured axial component of the flow velocity is compared with the numerical prediction by POLIMI and the present project in Figure 6.5a at a distance $x = 5.48$ m downstream of the turbine. POLIMI predicts a higher wind speed at the hub location compared to the wind tunnel measurements because it does not include the hub nose blockage. The present FALM results, on the other hand, predict a much lower wake deficit with no spike at the hub location, even though they did not include the hub either.



(a) Comparison of wake profiles (U_x) at $x = 5.48$.

(b) Wake visualization ($Q = 0.05$, present project).

Figure 6.5: Wake results for the steady wind condition of the UNAFLOW experiment.

The velocity deficit profile mainly depends on the thrust coefficient along the blade span. The highest velocity deficit from POLIMI occurs near the blade tips, just where the maximum thrust coefficient is

	Experiment	POLIMI	Current
ΔU_{avg} [m/s]	-1.889	-1.827	-0.568

Table 6.4: Averaged wake deficit within the rotor area (from equation 6.7).

expected. This does not necessarily mean that the thrust distribution from FALM is wrong. By analyzing the wake profile much closer to the rotor, it turns out it also presents a peak at hub position.

The main cause of the differences between the observed and predicted wake profiles might be the inability to capture the wake structure, mainly tip-root vortices, as depicted in Figure 6.5b. This is believed to be caused by excessive diffusion in the near-wake region. Note from the Q-criterion isosurface how individual root and tip vortices are shed but intermediately merged into a single structure. The relatively high rotation speed of the turbine does not help either, since shed vortices are much closer together and more difficult to capture individually.

In order to improve the solution, three strategies were tried: a moderately finer mesh, a smaller force projection width, and high-order numerical schemes (QUICK) for the convective terms. A slight improvement was accomplished when combining them all, yet it was not enough to correctly predict the wake shape or the velocity deficit profile. It is thought that a denser mesh with a less diffusive turbulence model could improve the accuracy of the results.

The averaged wake deficits are presented in Table 6.4, confirming the gap between FALM and other results. With the presented evidence, it is concluded that the present set-up will not be capable of predicting wake shape or velocity deficit. This may not be a problem for the steady case since the focus is on hub loads, but it will definitely impact the unsteady cases where the turbine may interact with its own shed vorticity. However, a finer mesh or more advanced turbulence models are beyond the scope of the present analysis given their associated computational overhead.

A note on 2P oscillations

During the UNAFLOW experimental campaign, 2P blade oscillations were captured and associated with a rotor aerodynamic imbalance (e.g., blade pitch error or blades with different aerodynamic performance). However, the same phenomenon was observed in the FALM simulation, where the blades are an exact copy of each other, calling for a different hypothesis. This effect is also seen in the prescribed-motion cases later described.

The thrust loads of individual blades obtained for case L1.1a showed 2P oscillations, which became 1P when added up. The source of such oscillations is not an aerodynamic imbalance but has its origin in the inflow velocity sampled by the blade as it rotates. Figure 6.7a depicts the inflow velocity seen by the blade tip, and as expected, 2P oscillations are evident.

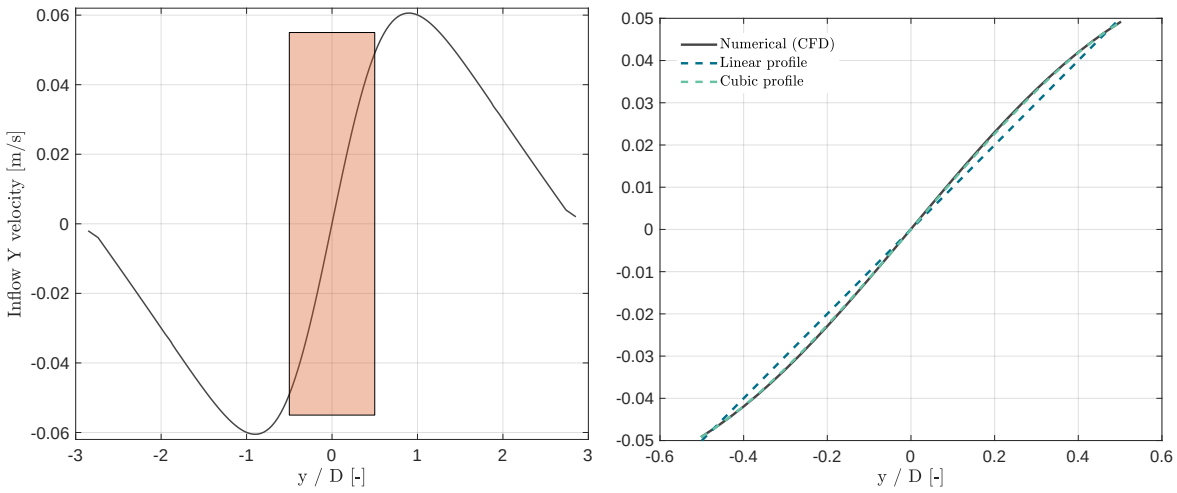
The proposed mechanism causing this phenomenon is related to the non-axial components U_y and U_z induced by stream-tube expansion as the flow goes through the turbine. Because of its divergent shape, non-axial velocity components increase as they approach the rotor plane (see Figure 3.1). At boundary walls, non-axial velocity components must become zero by definition.

This is confirmed by plotting the inflow y-component profile, $U_y\,(y)$, one turbine diameter upstream of the rotor plane. Note from Figure 6.6a how the non-axial inflow component is zero at hub position and increases as it approaches the stream-tube boundary (the shaded zone represents the rotor area), just to become zero again at the walls.

Now, let's build an analytical model to verify the proposed hypothesis. The magnitude of the velocity seen by an actuator element at a radial position r and azimuth angle φ, which includes both inflow and blade rotation components, is given by:

$$\begin{aligned} \|\mathbf{W}\|^2 &= {U_x}^2 + \left(U_y - \Omega\, r\, \cos\varphi\right)^2 + \left(U_y - \Omega\, r\, \sin\varphi\right)^2 \\ &= \left({U_x}^2 + \Omega^2\, r^2\right) + \left({U_y}^2 + {U_z}^2\right) + 2\,\Omega\, r\left(U_z\, \sin\varphi - U_y\, \cos\varphi\right) \end{aligned} \tag{6.16}$$

As shown in Figure 6.6b, the non-axial inflow velocity components can be modeled in the rotor area as



(a) CFD results (1.1a), shaded zone represents the rotor area.

(b) CFD and proposed analytical profiles along the rotor area.

Figure 6.6: Inflow velocity U_y distribution along y, at hub height and $x = -D$.

linear or cubic profiles. Following the former approach:

$$U_y = a_y\, y + b_y \;\;,\;\; U_z = a_z\, y + b_z \tag{6.17}$$

Which can be easily rewritten as a function of the azimuth angle, given that the rotation axis is defined along the negative x direction:

$$\begin{aligned} U_y &= a_y\ (r \sin\varphi + y_0) + b_y = a_y\, r \sin\varphi + c_y\ ,\;\; c_y = a_y\, y_0 + b_y \\ U_z &= a_z\ (r \cos\varphi + z_0) + b_z = a_z\, r \cos\varphi + c_z\ ,\;\; c_z = a_z\, z_0 + b_z \end{aligned} \tag{6.18}$$

To make the analysis simpler, consider $c_y = c_z = 0$, meaning that the turbine hub is located at the origin and the inflow profiles U_y and U_z are anti-symmetrical. Substitution into equation 6.16 yields:

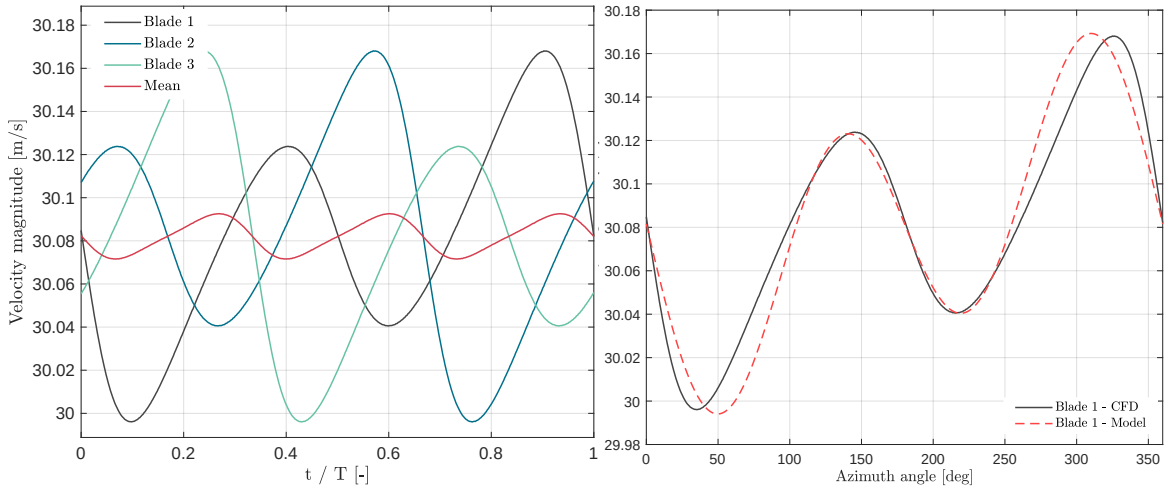
$$\|\mathbf{W}\|^2 = \left(\mathrm{U}_x{}^2 + \Omega^2\, r^2\right) + \frac{r^2}{2}\left[a_y{}^2 + a_z{}^2 + \left(a_z{}^2 - a_y{}^2\right)\cos\left(2\,\varphi\right)\right] + \Omega\, r^2\ \left(a_z - a_y\right)\sin\left(2\,\varphi\right) \tag{6.19}$$

Note that two 2P terms appear, proportional to $\left(a_z{}^2 - a_y{}^2\right)$ and $\left(a_z - a_y\right)$ respectively. The coefficient a represents the magnitude of the *"stream-tube"* effect, representing how much the flow deviates from the axial direction. In a symmetrical domain, where lateral and vertical walls are as far apart from the rotor, $a_y = a_z$ and all 2P effects would cancel out. The same applies to turbines that are not wall-bounded. An interesting property of equation 6.19 is that by adding up the forces from the three blades (with a phase shift of 0, 120 and 240°), the 2P terms cancel out no matter the values of a_y and a_z leaving only the 1P contributions.

In the experimental campaign, the turbine was closer to the vertical walls than to the lateral, hence $a_y \neq a_y$. Moreover, the distance to the ceiling was smaller than to the floor, thus $b_z \neq 0$. All in all, the expected tip velocity (obtained with the linear profile model) closely resembles the FALM results, as shown in Figure 6.7b.

To further verify the proposed 2P mechanism, a new simulation was carried out where the rotor was located at the center of a symmetrical domain, with its width equal to its height ($a_y - a_z = 0$). As predicted by the presented model, no 2P blade oscillations were found; see Figure 6.8a. Still, lower-amplitude 4P components appeared. These could be explained by adopting a cubic velocity distribution, like shown in Figure 6.6b:

$$U_y = a_y\, y + A_y\, y^3 + b_y \;\;,\;\; U_z = a_z\, z + A_z\, z^3 + b_z \tag{6.20}$$



(a) CFD results, all blades and average.

(b) CFD results vs analytical model with linear profiles.

Figure 6.7: Relative velocity magnitude at blade tip for L1.1a during the last rotation period, note the 2P oscillations.

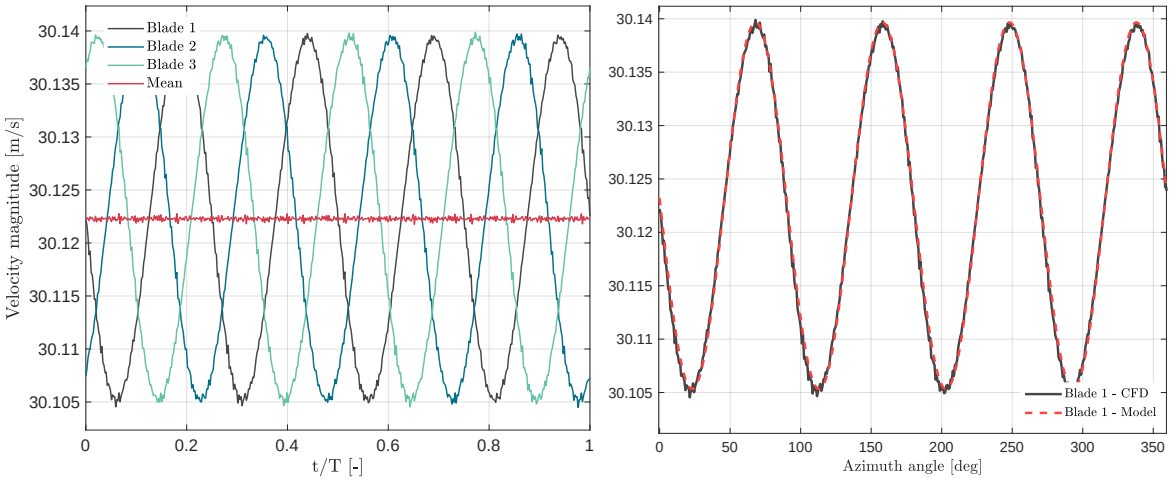
To keep the analysis simple, assume that the conditions are symmetrical in y and z, thus $a_y = a_z = a$, $A_y = A_z = A$ and $b_y = b_z = 0$. Also, $y_0 = z_0 = 0$. The velocities as a function of the azimuth angle are:

$$U_y = a\ (r \sin\varphi\ ) + A\ (r \sin\varphi\ )^3\ ,\ \ U_z = a\ (r \cos\varphi\ ) + A\ (r \cos\varphi\ )^3 \tag{6.21}$$

Given the linear behaviour of the above expression, we can study the effects of the linear and cubic components independently. Because we are interested in the cubic effects only, assume $a = 0$:

$$\|\mathbf{W}\|^2 = \left(\mathrm{U_x}^2 + \Omega^2 r^2\right) + \frac{A^2 r^6}{8}\ (3\cos(4\varphi) + 5) + \frac{\Omega A r^4}{4}\ \sin(4\varphi) \tag{6.22}$$

At this point, this analysis might have gotten too involved for the scope of the present thesis. Still, note how the 4P components have appeared even though we are assuming symmetrical conditions, meaning that they would be present even in unbounded flows. Moreover, the cubic profile analytical model suffices to predict such oscillations correctly, as shown in Figure 6.8b.



(a) CFD results, all blades and average.

(b) CFD results vs analytical model with cubic profiles.

Figure 6.8: Relative velocity magnitude at blade tip for case L1.1a within a symmetrical domain. Results for the last rotation period, note the 4P oscillations.

The lesson to take from here is that blade loads might oscillate slightly due to the *"stream-tube"* effect. Even though the blade loads might average out at the hub, their effect might be considerable for individual blades' aeroelasticity. Yet, this effect has only been observed in ALM numerical simulations, mainly

because the actuator line loads are very sensitive to the velocity at the quarter-chord. In CFD-resolved simulations or actual experiments, this effect might not be as relevant. The presented development is only meant as a justification attempt for anyone encountering these same 2P oscillations in their numerical model. Hopefully, they will find this information useful.

6.5.2. Prescribed motions

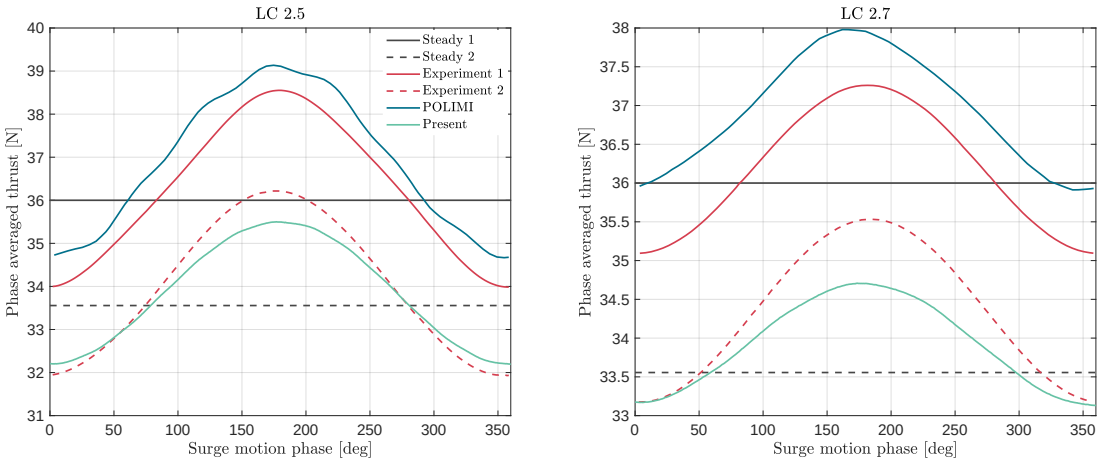
This sub-section will present and discuss the results from load cases 2.X and 3.X concerning prescribed motions in the surge and pitch directions. All simulations were performed with tip-root corrections enabled. The averaged results were obtained from the last three platform motion periods, while a minimum of five periods were simulated in total. Again, results will be compared to experimental measurements and POLIMI's numerical predictions.

Hub loads have been defined in the rotor local frame, which varies in time for the pitch load cases. In order to verify the implemented prescribed motions, the position and velocity of the actuator elements were first compared to the desired pitch and surge harmonic motions.

Phase-averaged thrust

The phase averaged thrust for both surge (Figure 6.9) and pitch (Figure 6.10) motions is clearly driven by the platform motion, and thus the results primarily exhibit a first-order sine component at platform frequency. This is generally true for all experimental results, while numerical predictions slightly deviate from a pure harmonic shape. In these figures, wind tunnel measurements are shown in red. Results for the steady wind tests are shown in black for comparison.

For the surge motions, the FALM results are slightly below the measurements from experiment 2, both in terms of mean and peak-to-peak values. When it comes to pitch, the situation reverses, and FALM overpredicts experiment 2. In all cases, the predicted average thrust is close to that of experiment 2 under steady conditions. On the other hand, POLIMI predicts the highest loads compared to both experiments and FALM. This was expected since, in the steady wind case, POLIMI's results were considerably greater than FALM's (with tip-root losses).



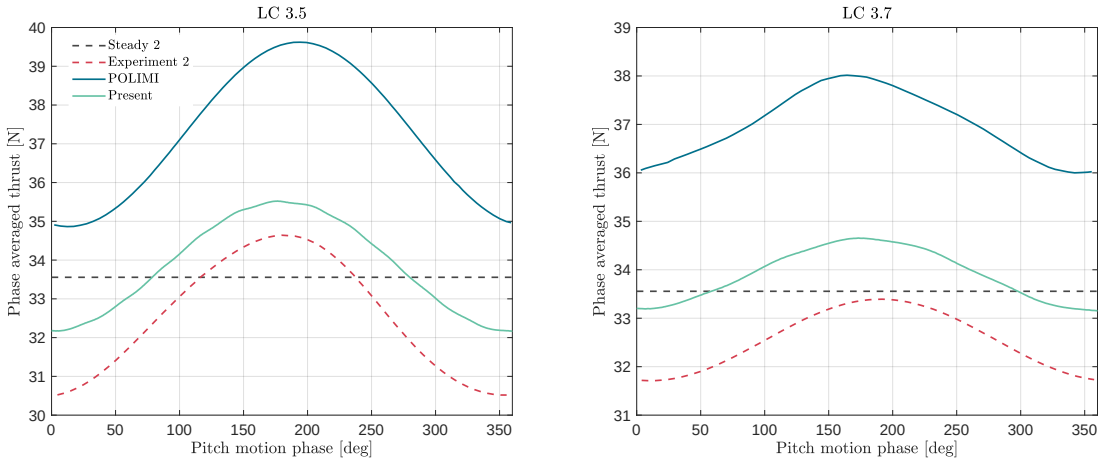
(a) Rotor thrust from LC 2.5

(b) Rotor thrust from LC 2.7

Figure 6.9: Comparison of the numerical and experimental phase-averaged thrust loads for prescribed surge motion.

Phase-averaged torque

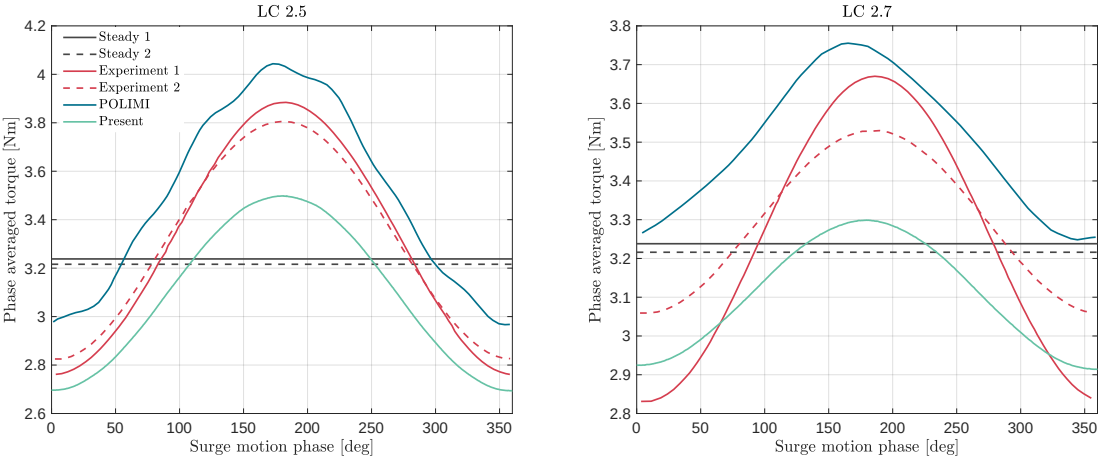
When it comes to torque, the same first-order sinusoidal behaviour is observed for both surge (Figure 6.11) and pitch (Figure 6.12) motions. The FALM results underpredict torque not only in surge but this time also in pitch. Yet, results are much closer to the experiment in pitch than in surge. Again, the present results fall well below POLIMI's predictions.



(a) Rotor thrust from LC 3.5

(b) Rotor thrust from LC 3.7

Figure 6.10: Comparison of the numerical and experimental phase-averaged thrust loads for prescribed pitch motion.



(a) Rotor torque from LC 2.5

(b) Rotor torque from LC 2.7

Figure 6.11: Comparison of the numerical and experimental phase-averaged torque loads for prescribed surge motion.

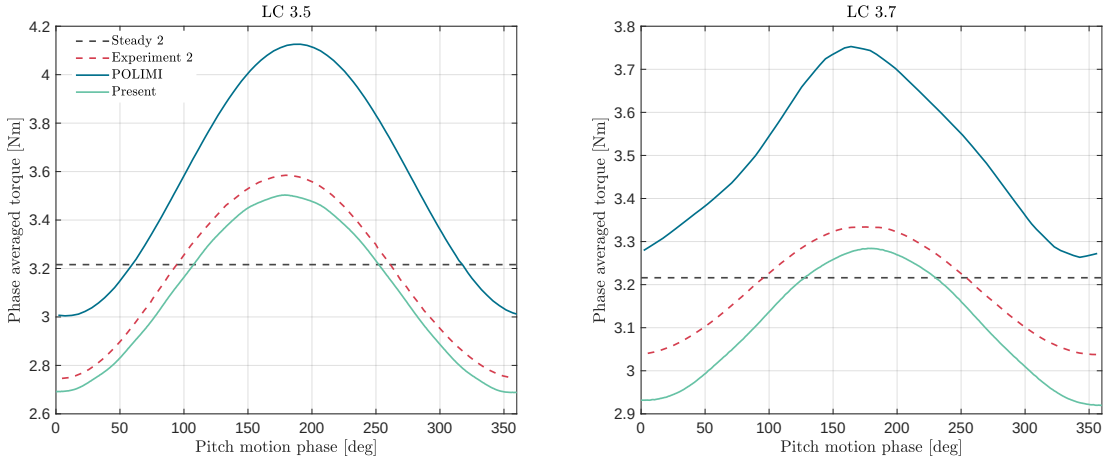
Peak to peak values

Figure 6.13 shows the normalized peak-to-peak values of thrust and torque. These plots compare experimental measurements with the median from all OC6 Phase III participants, POLIMI and FALM (present project). The loads have been normalized with the amplitude of the prescribed motion A. For pitch motions, this value is approximated as $A = l \cdot \sin(\theta_{\text{max}})$, where $l \sim 1.382$ m is the distance from the rotor hub to the tower base and θ_{max} is the angular amplitude of the pitch motion. Note how, once normalized, surge and pitch results return similar values, confirming that the rotor's apparent wind speed (which was maintained within cases 2.5-3.5 and 2.7-3.7) is the main driver of loads for the considered conditions.

The present FALM results display the lowest peak-to-peak values from all numerical models. Only for the pitch motions was the predicted peak-to-peak torque higher than the experimental value. In all cases, FALM showed greater peak-to-peak normalized amplitudes in pitch than in surge. The dispersion between results is much higher in the load cases with greater frequency (LC X.7).

In [5], a linear regression is constructed from the available data, which included an extra load case (X.1) at a lower frequency. This confirmed that peak-to-peak values were linearly proportional to changes in the rotor apparent wind and hence to frequency, with a zero peak-to-peak value in steady conditions. This outcome confirms that, for the studied cases, the aerodynamic behaviour of the rotor can be

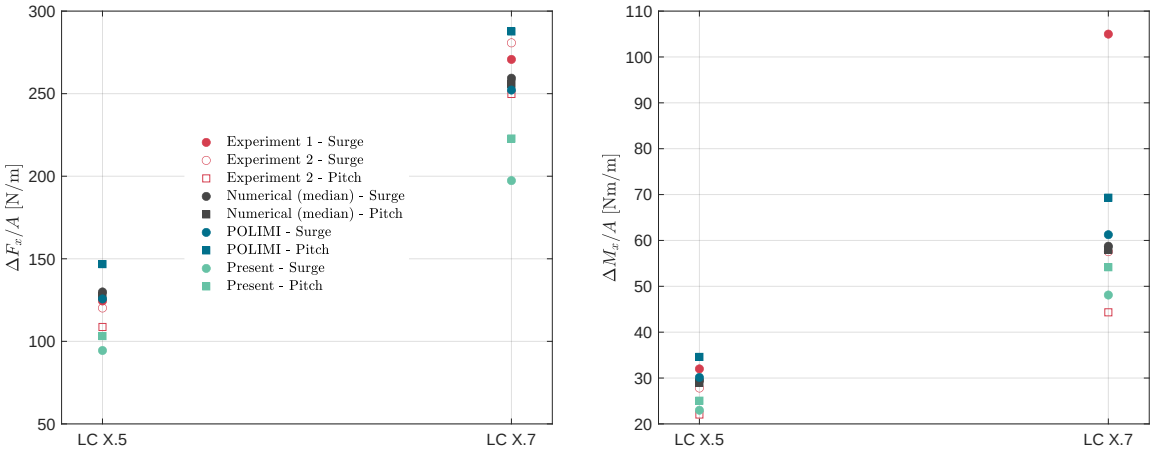
modeled as quasi-linear.



(a) Rotor torque from LC 3.5

(b) Rotor torque from LC 3.7

Figure 6.12: Comparison of the numerical and experimental phase-averaged torque loads for prescribed pitch motion.



(a) Aerodynamic rotor thrust

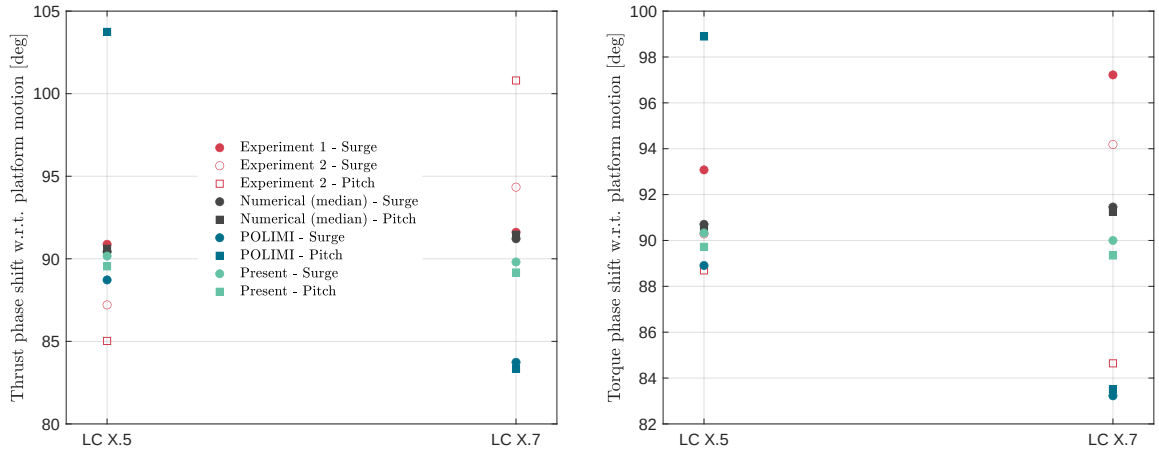
(b) Aerodynamic rotor torque

Figure 6.13: Peak-to-peak values of the phase-averaged hub loads, normalised with motion amplitude. Comparison of experimental, numerical and present results.

Phase shift

Next, the phase shift difference between the thrust and torque loads with respect to the platform motion is analyzed. Figure 6.14 compares experimental measurements with the median from all OC6 Phase III participants, POLIMI, and FALM (the present project). In all cases, results are close to 90º which is the expected value from quasi-steady theory. The latter predicts highest loads when the platform velocity is at its peak; hence, loads should be in phase with the platform's velocity (which, by definition, has a 90º shift w.r.t. the platform position). Again, results are more dispersed in load case X.7 (higher frequency).

Numerical models using unsteady airfoil aerodynamics predicted values slightly higher than 90º as a result of a small hysteresis in airfoil performance (rather than due to dynamic stall, which in the experiment was confined to the blade root) [5]. Regarding the FALM predictions, they show the closest phase shift to 90º from all results, suggesting a quasi-steady behaviour. Because the present approach could not capture the wake properly, it is hypothesized that the turbine loads were influenced by the relative wind speed alone rather than by unsteady turbine-wake interactions, explaining the quasi-steady behaviour.



(a) Aerodynamic rotor thrust

(b) Aerodynamic rotor torque

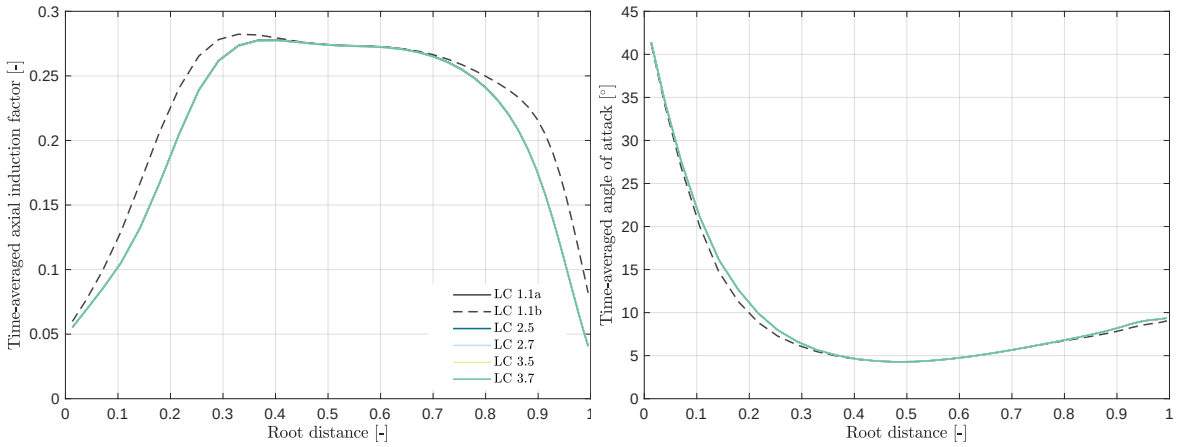
Figure 6.14: Phase shift between hub loads and platform motion.
Comparison of experimental, numerical, and present results.

Blade distributions

Even though the OC6 Phase III campaign did not measure any blade-distributed variable, it is still meaningful to analyze some results for the different motions considered. Since no participant submitted this information, the values presented here are taken from FALM only. Figure 6.15 depicts the radial distribution of the axial induction factor and angle of attack, whereas Figure 6.16 shows blade distributions of the aerodynamic coefficients. All values shown have been averaged over several motion periods.

Surprisingly, there is very little to no difference between the steady (1.1a) and unsteady load cases, no matter the frequency or type of motion. This means that the higher loads experienced by the turbine when moving upwind are balanced by the lower values when going downwind. Again, this is coherent with the quasi-steady aerodynamic theory, where loads are proportional to the relative wind speed.

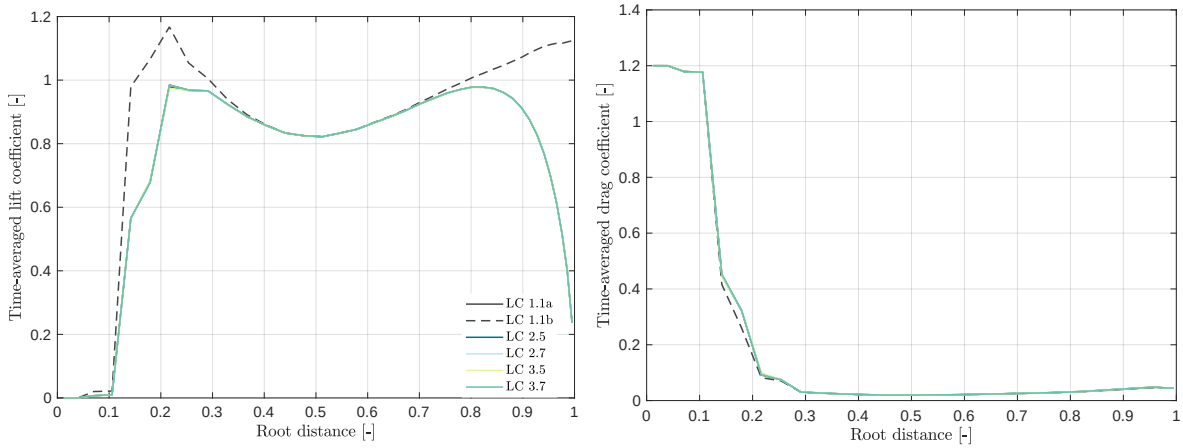
The second lesson to take from the figures is the impact tip-root corrections have on the results. They especially affect the wind speed seen by the rotor (see the axial induction factor) and the lift coefficient (to which the end-effect factor is directly applied). This has the overall effect of reducing the loading near the blade root and tip. The consequence is a reduced value for overall hub loads, as seen in Figure 6.4.



(a) Axial induction factor

(b) Angle of attack

Figure 6.15: Averaged a and α blade distributions for the different LC's from the present project.



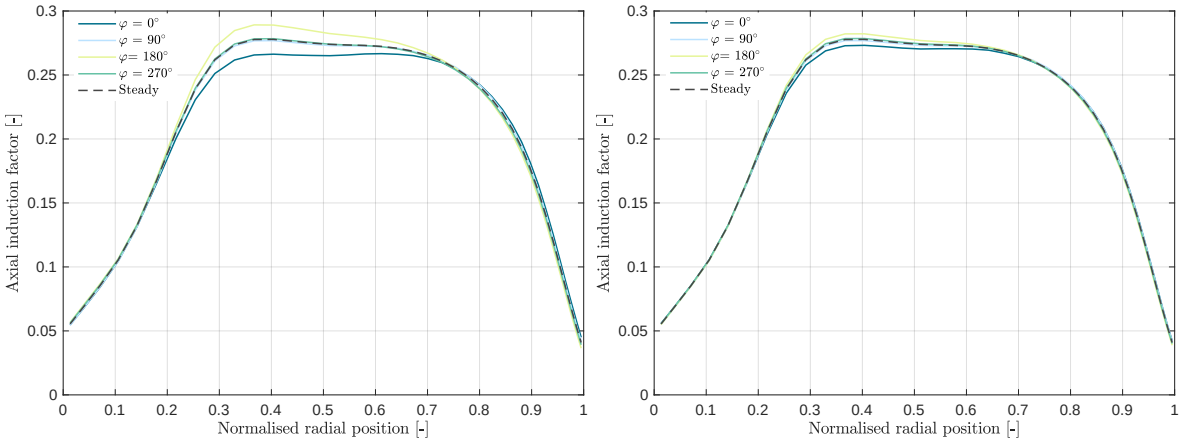
(a) Lift coefficient

(b) Drag coefficient

Figure 6.16: Averaged blade distributions of the aerodynamic coefficients for the different LC's from the present project.

Axial induction factor

Perhaps the most surprising conclusion from the previous section was the radial distribution of the axial induction factor a. As depicted in Figure 6.15a, the function $a(r)$ is almost identical for all load-cases when averaged in time for a whole motion period. This raises a new question: how does the prescribed motion affect the axial induction and consequently the rotor loads? In an attempt to answer it, the axial induction factors for the surge cases are shown in figure 6.17 for four time instants and compared to the steady case (L1.1a).



(a) LC 2.5

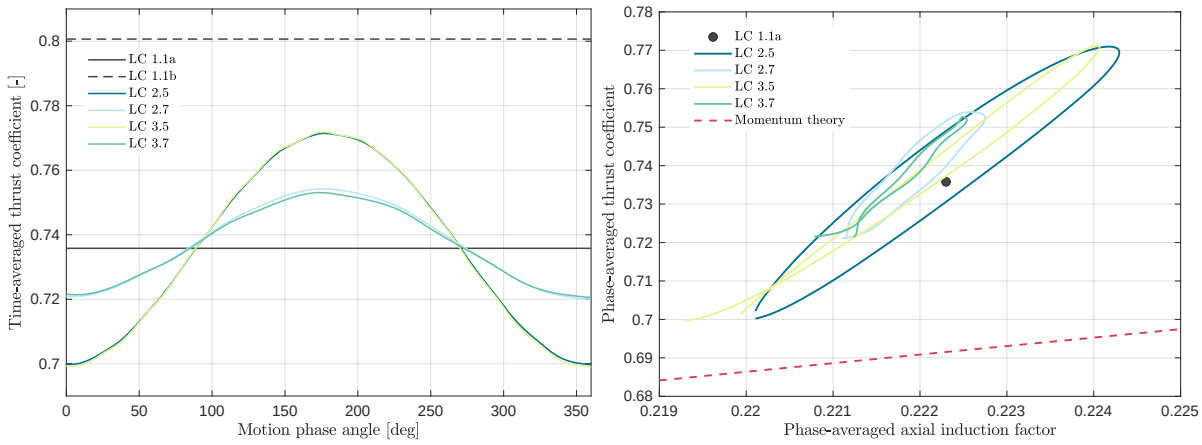
(b) LC 2.7

Figure 6.17: Radial distributions of the rotor axial induction factor at different time instants.

Now it becomes apparent that the induction changes as the turbine moves. As expected, a is higher when the rotor is moving upwind at maximum velocity ($\varphi = 180$º) and minimum in the opposite situation ($\varphi = 0$º). When the rotor achieves maximum displacement at zero prescribed velocity ($\varphi = 90$º and $\varphi = 270$º) the axial induction falls between the two former values. And going back to the original point, from the figures it is clear that the average induction over the whole period is the same as the one from steady conditions.

As expected, the peak-to-peak values of axial induction are higher in load-case 2.5, given that the platform velocity is more than twice that of load case 2.7. The effect of the prescribed motion velocity on loads is represented in Figure 6.18a, showing greater peak-to-peak values of thrust coefficient in the cases of higher speeds (2.5 and 3.5).

Finally, Figure 6.18b depicts the $C_T(a)$ diagram and compares it with the prediction from momentum



(a) Phase-averaged thrust coefficient

(b) C_T vs. a curves

Figure 6.18: Thrust coefficient (CT) as a function of average induction at the rotor plane (a).

theory (equation 3.2). Here, the axial induction factor is computed for the whole rotor at each time instant (weighted by swept area) and phase-averaged. Given the range of thrust coefficients and axial induction factors, the conditions considered here should fall within the validity range of the momentum theory (see Figure 3.2). For the steady case, the CFD result is slightly above (6.4%) the momentum theory prediction.

The amplitude of the orbits (hysteresis loops) described for each load case is coherent with the peak-to-peak thrust values from Figure 6.18a. Pitch motions show narrower trajectories than surge motions (less unsteady), but all are generally symmetric. The fact that the trajectories are not closed is due to the axial induction factor not being fully converged for the considered simulation timespan. Because the average a was found to be the same for all considered load-cases, as depicted in Figure 6.15a, it was expected that these curves would indeed coincide and resemble lines (quasi-steady) rather than orbits. A longer simulation period that eliminates all transient effects might reduce the gap between expected and obtained behaviour.

Recovering the phase-averaged axial induction factor turned out to be more challenging than the thrust coefficient. Since it was computed from the inflow velocity at each actuator element, both noise and 2P components (see section 6.5.1) were present and had to be filtered for a cleaner representation. Also, the inflow velocity component used for the pitch case was U_x rather than the component normal to the rotor. Given the small pitch angles, this should not cause a big difference.

6.5.3. Conclusions

In this chapter, the *turbinesFoam* library has been successfully modified so that any arbitrary rigid body motion can be imposed to a vertical axis turbine. To test the implementation, the experiment carried out during the OC6 Phase III campaign has been recreated for single DoF prescribed motions in surge and pitch. In general, hub loads were coherent with other participants. Because no uncertainty analysis was performed, it is difficult to assess whether the predictions match the experimental measurements.

Unfortunately, the simulation setup was not able to capture the rotor wake. The cause is still unknown, although the relatively high rotation speed might have something to do with it. In the next chapter, a simulation of a full-scale FOWT is performed with a much slower rotation speed. As shown in Figure 7.8, the wake shape and velocity deficit are properly captured. Because the numerical schemes and mesh of the two cases are very similar, it is likely that the underlying physics of the scaled-down turbine are the culprit of the failure in capturing the wake.

It was also possible to analyze local blade aerodynamics, surprisingly revealing that the phase-averaged aerodynamic coefficients and the induction factor did not change with the amplitude or frequency of the prescribed motion. This aligns with the observed quasi-steady behaviour of the model, predicting highest loads at the peaks of relative velocity. This aligns well with the findings by Sala [103], who also

used *turbinesFoam* to model FOWTs.

The hysteresis cycle in turbine loading was also analyzed, though longer simulation time spans are needed to produce more robust and converged phase-averages. After analyzing local blade results, 2P oscillations were encountered. The proposed probable explanation was the *"stream-tube effect"*, caused by the non-axial velocity components arising due to the fact that the turbine was not located at the center of the wind tunnel.

After all, the modified library has shown that it is capable of simulating vertical-axis turbines with prescribed motion. The actuator lines' positions and velocities are updated according to a generic six-DoF motion imposed by the user. Whether this will be enough to capture the aerodynamic phenomenology of FOWTs is still unknown and is left as an open question. Maybe more advanced techniques and corrections should be conceived before the ALM can be successfully applied to FOWTs.

7

Coupled FSI simulation of an FOWT

This chapter pulls together all the lessons learned so far and culminates with the development of a framework for the FSI simulation of FOWTs. First, section 7.1 presents the coupling methodology used to connect the ALM-rotor to the rigid-body platform. Then, section 7.2 puts this framework to the test by considering the 2-DoF motion (surge and pitch) of a moored semi-submersible FOWT (based on the *deepCWind* floater) under constant wind and regular waves. Finally, sections 7.4.1 and 7.4.2 serve as a preliminary verification step where the results from the coupled case are compared to those from platform-only and rotor-only simulations.

7.1. Coupling methodology

Being able to impose prescribed motions on an FOWT is just half the story: for most applications, a coupling mechanism between the floater's rigid-body model and the ALM-turbine is required. Note that this is a bi-directional coupling: the turbine follows the floater's rigid-body motion, which is in turn affected by the aerodynamic loads from the turbine. Because the solver in charge of computing the turbine loads is different from the one used to solve for rigid body motion, the FSI problem is said to be partitioned (see section 2.3). The solution to this issue has already been addressed in Chapter 5 where the fluid loads influence the body motion and vice versa, with sub-iterations being the chosen strategy to strongly couple the system.

Again, the technical details of the implementation will not be discussed here. Instead, the aim of this section is to examine how this bilateral interaction is handled and what the main differences are from the case of prescribed motions.

7.1.1. Concept

The key idea behind the rigid-body coupling is fairly simple: read the rigid body motion state and use it to move the FOWT accordingly, then compute the aerodynamic loads from the turbine and apply them to the rigid-body. For this to be successfully implemented, both the rigid body (*floatingSixDofRigidBodyMotion*) and floating ALM (modified *turbinesFoam*) libraries need to be adapted and *'connected'* accordingly. The writing-reading operation between different libraries is not straightforward —at least for OpenFOAM beginners— since the attributes are encapsulated within each class and cannot be easily accessed via an external library.

The adopted solution relies on OpenFOAM's object registry, which can be thought of as a large database that holds (i.e. *registers*) references to various objects that can be accessed during run-time. Many types of entities are compatible with it, with the `IOdictionary` being of special interest here since it can hold multiple attributes in it. Note that the purpose of this approach is to make attributes that would be restricted to a certain class globally available, thus pretending they are global variables. However, they won't be written into any external file. A template for the `IOdictionary` creation and access in OpenFOAM can be found in Appendix 8.3.

The presented procedure is summarized as a diagram in Figure 7.1, which for clarity omits most of

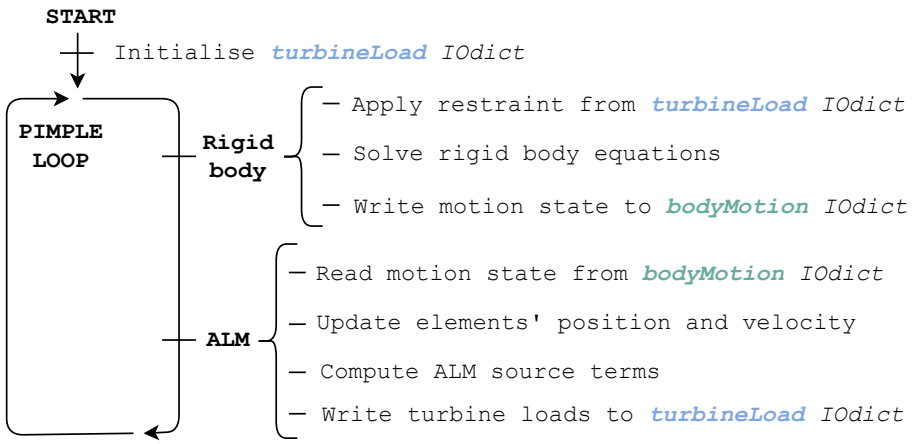


Figure 7.1: Diagram of the implemented FSI coupling between the rigid-body and ALM-turbine.

the PIMPLE steps from Figure 2.7. In the source term (ALM) calculation step, the turbine position and velocity are updated using the motion state from the rigid-body at the current iteration. The resulting loads will be applied to the rigid-body as external restraints at the next PIMPLE iteration.

Floater motions to the ALM turbine

The FOWT motion implementation described in section 6.2 was kept generic so that, given the four acknowledged motion parameters, the positions and velocities of the AEs could be updated. By creating a *bodyMotion* `IOdictionary` that contains the motion parameters of the floater rigid-body, the *turbinesFoam* library will be able to read them and update the FOWT state. The *bodyMotion* `IOdictionary` is constructed, initialized, and updated within the main source file of the *sixDofRigidBodyMotion* library and contains the following information for any given time instant:

- Position of the center of rotation of the floater ($\mathbf{O}_f$), given in the inertial frame.
- Orientation of the floater given as the transformation matrix from principal to inertial frames $\underline{\mathbf{Q}}_{pi}$.
- Linear velocity of the rigid body, given in the inertial frame.
- Angular velocity of the rigid body, given in the principal frame.

Because we are interested in the floater rather than the principal frame, the corresponding transformation matrix must be obtained:

$$\underline{\mathbf{R}}_f = \underline{\mathbf{Q}}_{fi} = \underline{\mathbf{Q}}_{fp}\,\underline{\mathbf{Q}}_{pi} \tag{7.1}$$

Where the transformation matrix from floater to principal frames $\underline{\mathbf{Q}}_{fp}$ is given by the user. For simple geometries, the floater and principal frames may coincide, and the latter would just be the identity matrix. Given these four variables, the motion routine implemented in section 6.2 can update the actuator lines' position and velocity according to the platform's rigid body-motion. Because the rigid-body equations are solved before the source-term addition step, the motion routine can be performed at the same PIMPLE iteration as the rigid-body solver step.

Turbine loads to rigid-body

Before the rigid-body library can read the aerodynamic loads from the ALM, they must first be computed. This includes not only blades, but also loads from the hub, nacelle and tower (if applicable). Within the original *turbinesFoam* library, the loads of each individual AE can be accessed, which comprise:

- Aerodynamic force, including both lift and drag effects.
- Total moment, including both the aerodynamic moment around the chord quarter position and the force moment around a reference point.

Because the floater is modelled as a rigid-body, it only requires the total force and torque as well as the application point of the former. To do so, the loads from each individual element are summed up,

including not only the blades but also the tower, nacelle, and hub (if present). The force moments are computed with respect to the origin of the rotor frame $\mathbf{O}_r$, although it can be chosen arbitrarily as long as it is kept consistent along all AEs. The resulting torque from the blades and hub along the rotor axis direction is not transmitted onto the floater, given that the rotor is free to rotate along this axis (generator effects are neglected). Since in incompressible simulations the loads are computed per unit density, they must be multiplied by a reference value of density given by the user (`rhoRef`) before they are transferred to the rigid-body solver.

Finally, these three variables (aerodynamic force, torque, and the application point of the former) are written in the *turbineLoads* `IOdictionary` which in turn is read by a new rigid-body restraint (`turbineAL`) that will apply the loads to the floater. This restraint is also in charge of creating and initializing (with zeroes) the *turbineLoads* `IOdictionary`. Because the source term addition step is performed after the rigid-body equations are solved, the ALM loads will be applied to the rigid body at the next PIMPLE iteration, as presented earlier in figure 7.1.

7.1.2. Usage

Coupling the ALM loads onto the rigid-body representing the floater requires modifications in both `dynamicMeshDict` and `fvOptions`. In the first file, the user must call the new restraint type `turbineAL`, in charge of creating the *turbineLoads* `IOdictionary` and applying the aerodynamic loads contained therein to the rigid-body:

```
        restraints
        {
                turbineLoad // Restraint name, must be unique
                {
                        sixDoFRigidBodyMotionRestraint turbineAL; // Restraint type
                }
        }
```

On the other side, a new sub-dictionary must be defined inside the `axialFlowTurbineALSourceCoeffs` dictionary of `fvOptions`:

```
        axialFlowTurbineALSourceCoeffs
        {
                // ...
                multiPhase          true; // Needed for multi-phase simulations
                phaseName           air; // Phase in which the turbine is located
                rhoRef              1.225;  // Reference density for load calculation
                rigidBodyMotion // New sub-dictionary for rigid body motion
                {
                  active                on;
                  coupleLoads           true; // Write aero loads to turbineLoads IOdict
                  isAligned             false; // Initial alignment is needed
                  principalOrientation  (1 0 0 0 1 0 0 0 1); // Transformation matrix Q_fp
                }
        }
```

At the time of writing this report, the rigid-body coupling is only available for axial flow turbines. The `coupleLoads` tag should be set to *false* when the ALM-turbine has to follow the floater's motion but the aerodynamic turbine loads don't have to be applied back to the rigid body (one-way coupling only). The `principalOrientation` entry is used to define the transformation matrix from floater to principal frames $\underline{\mathbf{Q}}_{fp}$. If not specified, the identity matrix is used instead.

It will usually happen that the initial configuration of the floater is already in a rotated orientation, meaning that the inertial and floater frames are not aligned. However, defining the geometry of the actuator lines easily becomes cumbersome if the user must take into account how the floater rigid body is oriented at the initial time. In cases where many simulations with different initial orientations are run, the user needs to modify the geometry accordingly.

To overcome this issue, an additional feature has been introduced that allows the user to define the turbine geometry only in the unrotated configuration. By setting the label `isAligned` to false, an initial alignment step will be performed that aligns the turbine with the floater frame. When set to true, on the

other hand, this rotation step is omitted since it is assumed that the FOWT geometry is already oriented with the floater rigid-body.

7.2. Coupled simulation of a semi-submersible FOWT

The objective of this section is to serve as a final proof of work for the developed framework. With this goal in mind, it will focus on the simulation of a moored semi-submersible platform coupled to an ALM-modeled rotor under steady wind shear and regular wave conditions.

Coupled turbine-floater simulations are scarce in the literature (see Table 3.1) and often relegated to low-mid fidelity range tools given their higher complexity and cost. The OC4 and OC5 (Phase II) campaigns are two examples of verification and validation approaches to the coupled simulation of FOWTs. However, they were tailored to statistical analysis with irregular wave spectra, which is out of the scope of CFD tools given the required simulation timespans. Moreover, these campaigns revealed that all participants underpredicted non-linear hydrodynamic loads, ultimately leading to the OC6 campaign. An accurate investigation of these loads arising in semi-submersible platforms has already been addressed during the OC6 campaign (Phase I and II) and by independent researchers (see the work of Rivera et al. [108] and Pinguet [121]).

Following the essence of the previous sections, the goal of the current is not a formal validation but rather a proof of work of the implemented coupling between floater and turbine. Thus, it is expected that the methods used will not yield results adequate for validation purposes. And given the high level of uncertainty among participants in OC4 Phase II, it might not be beneficial to aim for verification with other numerical tools. For these reasons, it has been decided that the case considered hereinafter will not be compared with external results. Yet, section 7.4.1 will compare the platform motions and mooring loads with those from a platform-only simulation with prescribed thrust force. Similarly, section 7.4.2 is devoted to the verification of the rotor loads, which are compared to those of a turbine with prescribed motion.

7.2.1. Case description

Because the results from the present case are not meant for comparison, there is complete freedom in choosing the turbine and environmental conditions. Still, only full-scale models will be considered. When working in scaled-down (laboratory) conditions, it is very difficult to achieve all geometric, aerodynamic, and hydrodynamic similarity (Froude number scaling often conflicts with Reynolds number scaling). Moreover, a scaled rotor must rotate much faster to achieve similar loading and TSR, increasing the temporal resolution needed for ALM simulations (as seen in equation 6.15).

Floating wind turbine model

Given their hydrodynamic complexity, semi-submersible floaters are the ones that can benefit the most from high-fidelity simulation tools. The relatively large motions of such platforms and the complicated design of the floater lead to a complex hydrodynamic problem, very sensitive to the modeling technique [86]. During the OC campaigns, it was demonstrated that the complex hydrodynamic modeling of semi-submersible platforms resulted in underprediction of both loads and motion responses, mainly due to the inability of mid-fidelity tools to capture non-linearities. This, added to the increased popularity of semi-submersibles [14] justifies why they could greatly benefit from CFD simulations.

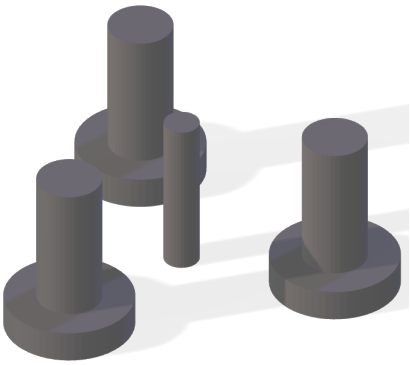


Figure 7.2: Simplified geometry of the *DeepCWind* semi-submersible floater used in the present case.

For this reason, the *OC4-DeepCWind* semi-submersible platform [145] is chosen for the sought-after simulation. This specific floater is selected given the availability of all its properties (geometry, mass, inertia, moorings, etc.) and its popularity among the research community. The platform is moored via three catenary lines 120º apart. In an attempt to reduce its associated hydrodynamic complexity and simplify the numerical grid, the inner braces have been removed from the model, as depicted

in Figure 7.2.

m [kg]	**CoM** $_{(x,y,z)}$ [m]	**I** $_{(xx,yy,zz)}$ [kg $\cdot$ m^2]
1.3473E+7	(0, 0, -13.46)	(6.827E+9, 6.827E+9, 1.226E+10)

Table 7.1: *OC4-DeepCWind* platform rigid-body properties: mass **m**, center of mass location **CoM** and moment of inertia along its principal axes **I** .

All the relevant information regarding the *OC4-DeepCWind* platform and mooring lines can be found in [145] and is summarized in tables 7.1 and 7.2. The mooring line properties are only given for the line going in the $-X$ direction. The attachment points of the other two can be easily calculated since they are 120º apart and the floater is centered at the origin. *Line length* refers to the total length of the catenary shape and is computed from the catenary equation by considering that the hanging part of the line measures 835.5m (see page 32 from [145]).

$\boldsymbol{\rho}$ [kg/m]	**L** [m]	**a** $_{(x,y,z)}$ [m]	**f** $_{(x,y,z)}$ [m]
108.63	865.5	(-837.6, 0, -200)	(-40.868, 0, -14)

Table 7.2: *OC4-DeepCWind* mooring line 1 properties: linear density $\boldsymbol{\rho}$, line length **L**, seabed anchor position **a** and fairlead position **f**.

When it comes to the turbine, the well-known 5-MW reference turbine from NREL [146] will be considered. Not only because the *OC4-DeepCWind* was tailored for this specific rotor, but also for its popular use and readily available properties (summarized in table 7.3). The rated tip speed is 80 m s^{-1}, thus the Mach number is $M \sim 0.23 < 0.3$ justifying the selection of an incompressible solver. Its shorter diameter in comparison with other offshore turbines will reduce the required size of the computational domain. Moreover, such a turbine has already been verified in steady-state conditions by Dos Santos [144] via an ALM simulation. For simplification purposes, the blades are considered rigid, with no pre-cone nor tilt. Neither the tower, hub, nor nacelle are included in the ALM model of the turbine.

D [m]	**d** [m]	**h** [m]	**m** [kg]	**CoM** $_{(x,y,z)}$ [m]
126	5	90	697460	(-0.2, 0, 64)

Table 7.3: NREL'S 5-MW reference turbine properties: diameter **D**, rotor overhang distance **d**, hub height **h**, mass m and overall center of mass location **CoM**.

Load case definition

When it comes to the loads acting on the turbine, a moderate yet realistic scenario has been considered, as summarized in Table 7.4. Since the expected dominant motions of the platform are surge and pitch (recall Figure 1.3), all other DoFs will be constrained (thus no gyroscopic coupling). This will considerably simplify the verification of aerodynamic loads from section 7.4.2.

It is very unlikely that representative sea states can be simulated with CFD when it comes to environmental conditions. Even though irregular sea states can be modeled with ocean wave spectra (e.g., JONSWAP), typical CFD simulation times are too short to extract meaningful statistical characteristics from the responses (a typical 3 hour time window is used for that purpose). Consequently, the waves generated at the inlet zone will follow Stokes' second order theory, which accounts for the second-order perturbations introduced by the finite water depth and wave steepness (see section 2.1.2 for a discussion on linear wave theory limitations).

The wave height has been set to 4m in an attempt to find a compromise between noticeable wave excitation and restricted floater motion, given that the morphing mesh technique is best suited for small displacements. The chosen wave period is 8s, close to the wind-wave peak in the ocean wave spectra (see Figure 2.2). Short wave periods have the advantage of reducing the total simulation time. The overall wavelength computed from the non-linear wave dispersion relation (equation 2.13) is $\lambda \sim 99.92$

$\mathbf{H_w}$ [m]	$\mathbf{T_w}$ [s]	$\mathbf{U(90)}$ [m/s]	**TSR** [-]	$\mathbf{z_0}$ [m]
4.0	8.0	9.0	7.5	0.3

Table 7.4: Proposed environmental conditions: wave height $\mathbf{H_w}$, wave period $\mathbf{T_w}$, wind speed at hub height $\mathbf{U(90)}$, rotor tip-speed ratio **TSR** and surface roughness $\mathbf{z_0}$.

m, while the considered water depth is $h_w = 150$ m. The resulting wave steepness is $\varepsilon = H/\lambda \sim 4\%$ and the relative water depth is $h_r = h_w/\lambda \sim 1.5$.

For the rotor side, it will rotate at a constant speed in below-rated conditions, corresponding to a tip-speed ratio TSR $= 7.5$. Recall from section 3.1 that the motions of FOWTs introduce aerodynamic unsteadiness and complex flow structures even under uniform free-stream conditions. No control strategy is followed. To resemble realistic atmospheric conditions, a wind shear vertical profile will be imposed. The latter is based on the well known logarithmic expression from equation 7.2. Even though a power-law profile should be preferred above blending height (~ 60 m), for simplification purposes only the logarithmic profile is considered. The parameter d is the *effective height if impenetrable roughness*, an offset that accounts for wind-slowing obstacles (in this case, waves), and is set to $d \sim 4\,H_w = 16$ m. The wind velocity at hub height is $U(z_{\text{ref}}) = 9$ m/s while the surface roughness has been set to $z_0 = 0.5$m, larger than it should be in open seas ($z_0 \sim 0.002 - 0.01$ m).

$$U(z) = U(z_{\text{ref}})\,\ln\left(\frac{z-d}{z_0}\right)/\ln\left(\frac{z_{\text{ref}}-d}{z_0}\right) \quad \text{if} \quad z > d + z_0 \tag{7.2}$$

The latter is done to reduce the wind speed gradient in the vertical direction: in a matter of centimeters, the velocity jumps from almost zero just above the waves to a value close to U_∞. The grid resolution is not fine enough to resolve the complex boundary layer generated in this region, resulting in spurious air velocities that arise from the momentum transfer from the water. Given the large difference in density between the two phases, even a slight imbalance in the momentum equation can introduce substantial air velocities. Accurate modeling of the water-air interface region under wave propagation conditions and wind shear profiles is out of the scope of this project, and thus a large value of z_0 has been chosen to minimize these effects.

7.2.2. Numerical set-up

The simulation setup is strongly influenced by the experience gained during the present research. Thus, a monolithic multi-phase simulation will be performed, where the rigid-body solver is combined with a morphing mesh technique. The ALM-rotor will be coupled to the rigid-body platform through the newly developed libraries, whereas quasi-steady catenary restraints from `waves2Foam` will be used to model the mooring lines.

Although it is true that the adaptive mesh refinement (AMR) technique would allow for cell refinement near the zones of interest (e.g., free surface or actuator lines) as the simulation progresses (i.e., refinement zones would follow wave propagation and turbine rotation), this method is not considered given the lack of experience with it. The same is true for more advanced methods such as the overset mesh technique.

Turbulence model

Because of the monolithic approach followed in this coupled simulation, the same turbulence model must be shared between the floater and turbine, even though they might benefit from different models (e.g., laminar transport for the hydrodynamics and $k-\varepsilon$ for the turbine). For this reason, the $k-\omega$ SST (shear stress transport) model will be used, which switches between $k-\varepsilon$ far away from the walls (free-stream conditions) and the classical Wilcox $k-\omega$ model in the near-wall region. The latter is a two-equation model that solves for the transport of turbulent kinetic energy k and specific turbulence dissipation rate $\omega = \frac{1}{C_D}\frac{\varepsilon}{k}$.

In OpenFOAM, the user must set the initial internal field conditions for both k and ω. The former is linked to the turbulence intensity by equation 6.9 whereas the latter is given as a function of k in equation 7.3.

There, $C_\mu \sim 0.09$ is a model constant, and L is a reference length scale chosen equal to the blade chord.

$$\omega = \frac{k^{1/2}}{C_\mu^{1/4} L} \tag{7.3}$$

Relaxation zones and wind profile

In all previous chapters, the wave generation zone length was set to $L_g = \lambda$ and the wave absorption zone to $L_a = 2\lambda$. However, the current simulation must account for the turbine, which requires a longer domain along the wind-wave propagation direction (especially downwind). For this reason, the relaxation zones have been extended to $L_g = 2D$ and $L_a = 4.5D$, leaving room for a wave propagation zone length $L_p = 2D$.

The wave field is initialized with Stokes second-order theory along the domain with the `setWaveField` utility from *waves2Foam*. This utility, however, initializes the air-phase velocity to zero. In order to ensure a smooth transition between the free surface and vertical wind shear profile, the relaxation zones are extended vertically up to $z = 4\,H_W = 16$m. The wind profile from equation 7.2 is applied to the internal field only above this height ($d = 4\,H_W$) by means of the `setExprFieldsDict` utility.

Geometry and grid

The mesh for the coupled case can be thought of as a merge of the grids from two previous cases: the floating buoy (section 5.3) and the OC6 Phase III turbine (section 6.4). Some changes have been made, however: the refinement zone around the platform is now a sphere instead of a box, and zone 3 (from the OC6 simulation) has been removed. A free-surface refinement is applied throughout the domain, with regions of smaller cell size near the wave propagation zone. The domain is enclosed by a box which is 1008m (8D) long, 504m wide (4D) and 492m high. Of the domain height, 342m correspond to the air phase. This value is chosen such that the rotor hub is located at the same distance from the ceiling (2D) as it is from the lateral walls.

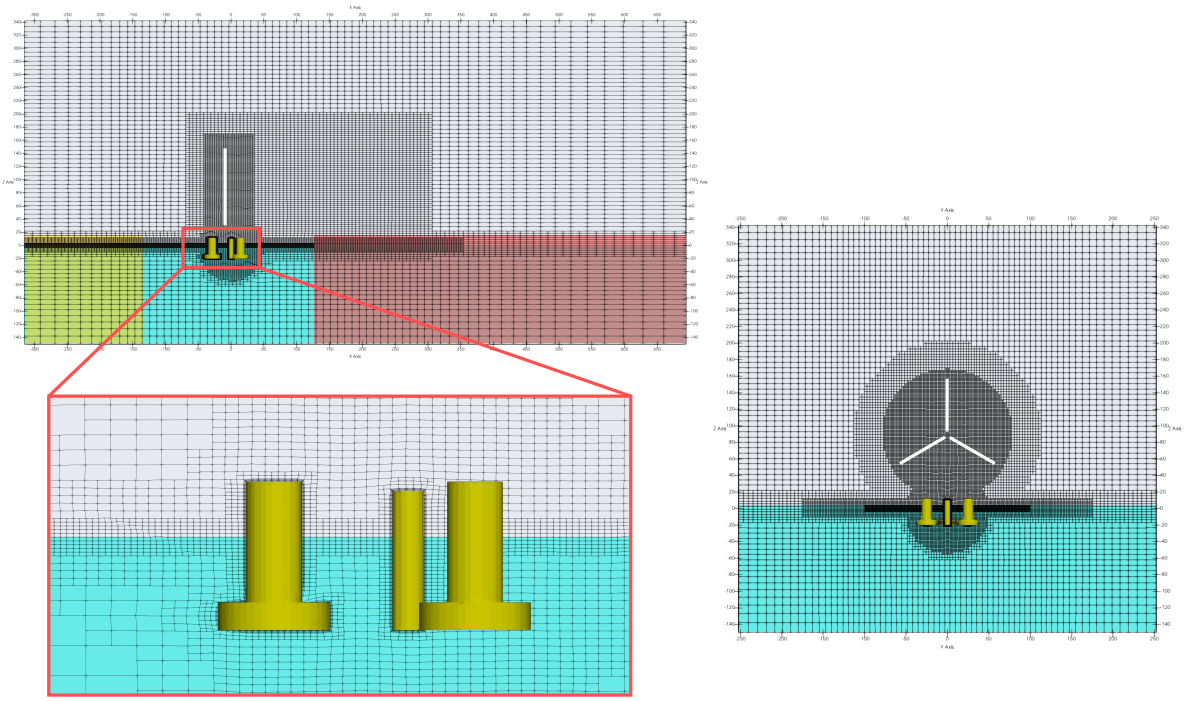


Figure 7.3: Mesh for the coupled simulation of a semi-submersible FOWT: front (right) and side (left) views. Relaxation zones are shown in yellow (inlet) and red (outlet).

A background mesh is created with *blockMesh* with increasing aspect ratio towards the boundaries to reduce the cell count. To push the number of cells even lower, the cells are not uniform but longer along

x. As discussed in section 4.2, an aspect ratio equal to $\sqrt{3}$ in the wave propagation direction improves the accuracy of the wave dispersion relation while reducing the total number of cells. The boundary layer along the floater surface (given as a *STL* file) is captured by means of a prism mesh created with *snappyHexMesh*. Its width is set to $\delta = 0.2$m and consists of 4 cells.

Refinement zones are created by means of the *snappyHexMesh* utility. As with all the other cases, the mesh is constructed parametrically based on the size of the unrefined grid, Δx_0. This value has been chosen to be equal to 6.4m as a compromise between accuracy and computation cost. The resulting grid is depicted in Figure 7.3, with a total of 5 cells per wave height along the wave propagation region (refinement level of 3) and a cell size at the rotor of $\Delta x_r = 1.6$ m (refinement level of 2).

This is a rather coarse mesh, especially for the free surface and boundary layer, and yet it resulted in a total of $\sim$ 2 M cells. Going any finer would be prohibitive given the long simulation timespan required (~ 300 s) and the multiple PIMPLE iterations needed to ensure a tight FSI coupling between the platform and ALM turbine. Because this simulation is not constructed for verification purposes, a coarse mesh should not affect the overall results.

Boundary conditions

The boundary conditions for the present case are very similar to those from the floating-body simulation in section 5.3. Because the domain is much higher now due to the presence of the turbine, the `inletOutlet` condition in the *top* patch has been replaced by a non-slip condition (now included inside the *walls* group). All BCs are presented in Table 7.5.

	α	P_{rgh}	U	k
Inlet	zeroGradient	fixedFluxPressure	logLawInlet	fixedValue k_0
Outlet	zeroGradient	fixedValue 0	zeroGradient	fixedValue k_0
Walls	zeroGradient	zeroGradient	slip	fixedValue k_0
Floater	zeroGradient	fixedFluxPressure	movingWallVelocity	kqRWallFunction

Table 7.5: Boundary conditions for the volume fraction, modified pressure, velocity and turbulent kinetic energy.

For the velocity inlet, a vertical profile based on equation 7.2 is applied using the *codeStream* utility. Regarding turbulence, only kinetic energy k is shown since the conditions are kept the same for all turbulent quantities (ω and ν_T): uniform value in all boundaries except along the floater surface where a wall function is used. Because there is no condition for the total pressure (only for modified pressure P_{rgh}), a *pvalue* of 0 is arbitrarily imposed on cell nº 1 within the PIMPLE dictionary in *fvSolution*. Recall that for simulations with a morphing mesh, an initial BC for the *pointDisplacement* field is needed.

Morphing mesh and rigid-body

As stated at the beginning of this section, a morphing mesh approach is used instead of the adaptive mesh refinement technique. Morphing meshes have been successfully implemented in Chapter 5 and can be easily coupled to the rigid-body solver. The morphing procedure is based on a diffusivity scheme, whose effect is proportional to the square of the inverse distance from the floater. The inner distance is set to 1.5 times the boundary layer thickness (~ 0.3m) whereas the outer distance is set to the wavelength of the incident wave field $\lambda \sim 100$m.

The total mass of the rigid body is computed as the sum of the platform and turbine masses from tables 7.1 and 7.3, leading to a total mass $m_T \sim 1.417 \cdot 10^6$ kg. The overall center of mass is located at $(-0.0098, 0, -9.6475)$ m. The rotational inertia of the turbine is neglected, and only the platform contribution from Table 7.1 is considered. Even though the gyroscopic moment from the rotor could be modelled using the restraint developed in section 8.3, it has been left out since the system is restricted to surge andpitch-only, leaving no room for gyroscopic coupling.

The aerodynamic loads from the ALM turbine are included through the `turbineAL` restraint from section 7.1.2. Three catenary restraints are defined according to the mooring line properties from Table 7.2. The seabed anchor positions along the surge (X) direction are moved by $m_0 = -23$m (chosen based on the platform-only simulations from section 7.4.1). This mooring offset in surge is meant to reduce the duration of the transient phase until the system reaches its equilibrium position. Furthermore, it

reduces the maximum platform displacements and hence helps maintain a good-quality mesh. It was also concluded that an initial pitch angle θ_0 did not improve the overall convergence time, although it did help reach pitch equilibrium sooner.

The rigid- body is restrained using `line` and `axis` constraints so that only surge and pitch motions are allowed. The Newmark-β solver with an average constant acceleration scheme is used to advance the rigid-body equations in time. No damping nor relaxation factors were used to stabilize the simulation.

ALM parameters

The blade property distributions (twist and chord) from NREL's 5-MW turbine are given at 17 elements along the span. Because *turbinesFoam* works with stations rather than elements for blade discretization purposes, linear interpolation and extrapolation routines were used to obtain the blade properties of the 18 stations bounding the 17 elements. A total of 2 AEs were defined between stations, leading to a total of 34 elements with an average length $\Delta r \sim 1.8$m. These are not evenly distributed along the span, but follow the element spacing from NREL [146]. Because the average cell width at the rotor region is $\Delta x \sim 1.6$m, there is approximately one AE per grid cell.

A total of nine different aerodynamic profiles were used along the span: cylinder shapes for the inner blade (no lift), variations of the Delft University (*DU*) airfoil series for the middle sections, and *NACA64* for the outward region. The aerodynamic coefficients are already corrected for rotational stall delay (3D correction). Because the polars were not given for a range of Re numbers, *turbinesFoam* by default applies a Re number-based correction. Glauert's end-effects are enabled which reduce aerodynamic loads towards the blade tip and root. The inflow velocity will be sampled only at the quarter-chord position to save computational resources, since the search algorithm (find closest cell to an AE) can turn expensive in large meshes.

The `cellSet` region defined in the *topoSetDict* should contain the AEs during the entire simulation, or the library would return an error and the simulation would stop. Because the turbine is moving with a non-prescribed motion, the platform-only simulation from section 7.4.1 is used to get an insight of the expected range of turbine motions. Larger zones are safer but increase the computational cost since the search algorithm has to check more cells. All in all, the `cellSet` is defined as a hub-centered cylinder along X, with a 25% larger radius than the rotor's and a length of $0.6\,D$.

To couple the platform motion to the turbine and the rotor loads onto the floater rigid-body, the *rigidBodyMotion* sub-dictionary from section 7.1.2 is defined with the `coupleLoads` flag set to *true*. The `isAligned` entry is set to *false* to ensure that the rotor will be oriented with the floater in case the initial pitch angle θ_0 is different from zero.

Time discretization

When it comes to the time-step selection, it has been decided to rely on a constant value of Δt rather than an adjustable value based on the maximum Courant number. During the simulation, some regions close to the free surface experience very high velocities due to a momentum imbalance between the air and water, amplified by the large density difference between phases. Luckily, these spurious currents arise in the wave damping zone, far away from the floater, and do not affect the turbine dynamics. Using an adjustable Δt approach would make the time-step prohibitively small as a consequence of the high CFL values (~ 0.5) in this region.

Contrarily to the previous simulations, now two different physical systems are in play: the turbine and the platform. The two have different requirements regarding time-discretization: while one is driven by the wave loads and body motion, the other is dependent on the turbine rotation period. From the literature review carried out in section 4.2, it was found that wave-propagation simulations used time-steps in the range of $T/400 \geq \Delta t \leq T/1750$, where T is the wave period. Conversely, the ALM turbine simulation from section 6.4 linked the minimum Δt to the time it takes an AE to cross a grid cell (see equation 6.15). From a combination of both criteria, the selected time-step is given by equation 7.4.

$$\Delta t = \min\left(\frac{T}{800}, \frac{2\,f_D}{\Omega}\right) \tag{7.4}$$

For the present case, the criteria based on wave period results in the smallest time-step, thus $\Delta t = T/800 = 0.01$s. From the platform-only and turbine-only simulations in sections 7.4.1 and 7.4.2, respectively, it is known that a relatively large transient phase can be expected. For this reason, the simulation will run for a total timespan of 40 wave periods, or 320 seconds.

Schemes and solvers

The numerical schemes are taken from the floater simulation in Chapter 5.3, with a `meshWave` method for wall distance computation in the turbulence model. In this case, the PIMPLE scheme couples not only the rigid-body and fluid equations but also the ALM rotor. To ensure rotor-platform coupling, at least two outer correctors must be employed. Because of the higher cost they require, the number has been reduced to `nOuterCorrectors` = 3. With these settings, the simulation was run on 120 processors (*Intel Xeon E5-2640v4*) and took ~60 hours to complete.

7.3. Results and discussion

7.3.1. Platform

Rigid body motions

The platform dynamics are described by the time evolution of its two DoFs: pitch and surge. The time histories of motion amplitudes and velocities are depicted in figures 7.4 and 7.5, respectively. By analyzing the pitch results, it is easy to identify a transient start followed by a steady state stage starting around $t \sim 20T$ where velocities are almost periodic. In this stage, there is an equilibrium between hydrostatic and turbine loads that results in a non-zero equilibrium pitch angle. This situation is justified by the almost zero mean velocity in pitch. Still, the platform oscillates back and forth at exactly the wave frequency due to the incoming waves.

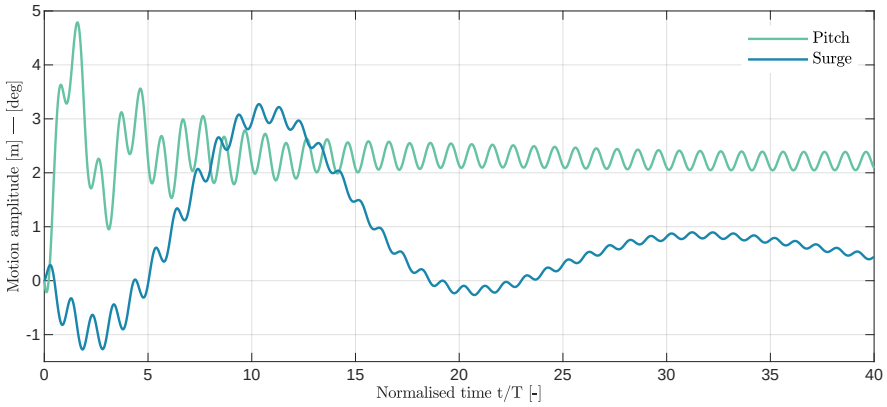


Figure 7.4: Time evolution of the floater position (surge and pitch).

In surge, the situation is different. Even though there is oscillation at wave frequency, the main contribution to surge has a much lower frequency. Contrarily to pitch, which is driven by environmental loads, the surge motion is forced by the mooring line loads. A clear correlation can be identified between the surge amplitude from Figure 7.4 and the horizontal mooring line loads from Figure 7.6a. Yet, it seems that this motion will dampen in time and reach an equilibrium position as surge velocity approaches a zero mean value.

	Surge		**Pitch**	
	Position [m]	Velocity [m/s]	Position [deg]	Velocity [deg/s]
Mean	0.6264	-0.0075	2.2175	2.89 E-5
Amplitude	0.0701	0.0425	0.1712	0.1008

Table 7.6: Mean and amplitude values of the platform surge and pitch motions between $34T \leq t \leq 40T$.

The main properties of the platform motion between $34T \leq t \leq 40T$ are summarised in Table 7.6.

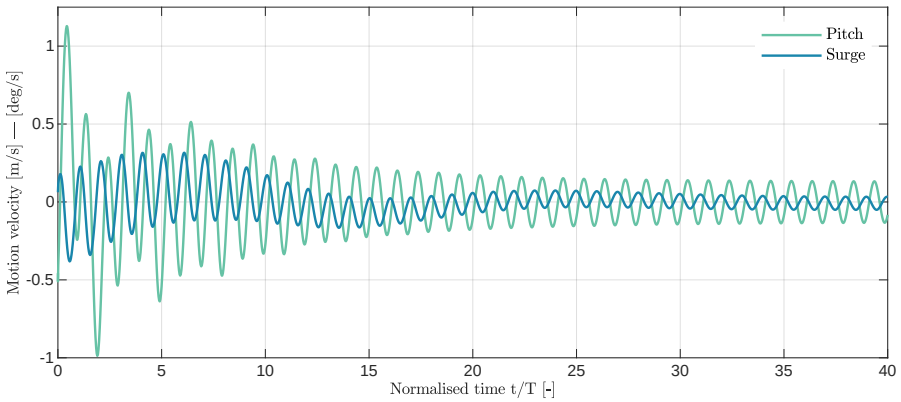
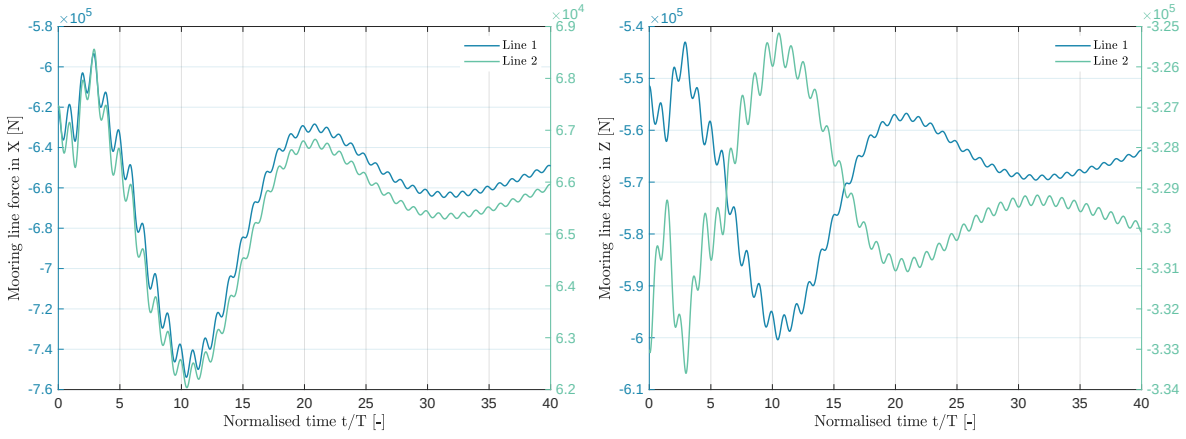


Figure 7.5: Time evolution of the floater velocity (in surge and pitch).

Although both surge and pitch oscillate at wave frequency, the two motions are not in phase with one another nor with the incoming waves. While the surge position shows a phase shift of 79.73° with the free surface elevation from equation 2.14, the pitch is delayed by 219.10°. With little error, both velocities can be described as pure harmonic waves. This fact will be used in section 7.4.2, where this same motion will be prescribed in a turbine-only simulation.

Mooring loads

One of the benefits of using a mooring model based on the catenary equations is that individual line loads are available. Figure 7.6 depicts the time evolution of such loads. Here, *Line 1* (L1) refers to the upstream line, while *Line 2* (L2) refers to any of the other two oblique lines (only one is shown since they experience the same loading). Although still appreciable as wiggles, wave loads (at wave frequency) are not the driving factor of mooring line loads. It is the lines themselves, with their elastic potential energy, that induce the lower-frequency conciliation. In the case set-up, a mooring offset of $m_0 = -23$m was applied, meaning that L1 was stretched while L2 was shortened. Because of their catenary shape, the lines do not exactly behave like linear springs: a stretched line will exert a greater load than another line that has been shortened by the same amount.



(a) X component **(b)** Z component

Figure 7.6: Time-evolution of the mooring lines forces.

With this in mind, it is not strange that the horizontal loads in surge from L1 are an order of magnitude higher than those from L2. Unsurprisingly, the total contribution from the mooring lines (including L3) reveals an overall negative force in surge, which is meant to counteract the turbine thrust and wave loads. On the other hand, vertical loads should balance the weight/buoyancy difference and provide pitch equilibrium, explaining the mirror-like shape. As with the floater's surge amplitude, mooring loads are expected to damp out as the FOWT reaches its equilibrium position.

7.3.2. Turbine

Aerodynamic loads

As discovered in section 6.5, the ALM allows for the recovery of many different variables concerning the turbine aerodynamic performance, such as blade distributions of loads, angle of attack, or inflow velocity. However, a detailed aerodynamic analysis of the coupled FOWT falls outside the scope of the present chapter. Hence only the most relevant metrics will be analyzed, namely the rotor's overall power and thrust (rotor-normal load) coefficients, whose time evolution is depicted in Figure 7.7.

Just like in the platform motion, a steady state is reached after $\sim 25\,T$ in which the aerodynamic coefficients become periodic, although not purely harmonic. Figure 7.7 also presents the rotor-perpendicular component of the hub velocity induced by the platform pitch and surge motions. It can be easily recognized that both C_P and C_T are in anti-phase ($\Phi = 180°$) with the hub velocity, which is a characteristic of quasi-steady behaviour. Since a negative minimum velocity implies that the rotor is moving against the wind, the relative velocity reaches its maximum at this point, as do the aerodynamic coefficients.

Because the hub motion and velocity oscillate at wave frequency, the same behaviour is expected for the aerodynamic coefficients. Still, their shape is far from harmonic due to the 3P oscillations that wind shear introduces. This effect will be further discussed in section 7.4.2.

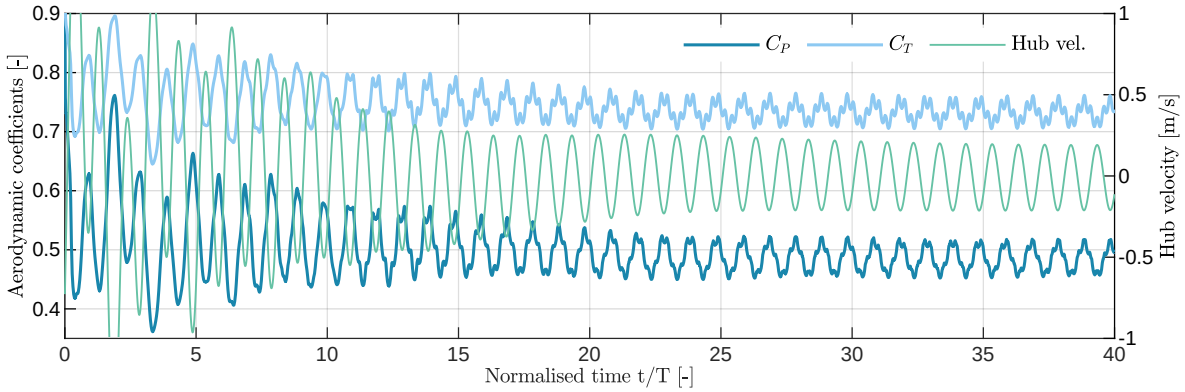


Figure 7.7: Time evolution of the turbine aerodynamic coefficients, compared with the rotor hub velocity.

Turbine wake

In FOWTs, the rotor is not static, and the wake will move following the rotor's trajectory. Because the turbine is no longer aligned with the incoming wind, the resulting wake is often skewed. The time history of the wake motion is of crucial importance in wind farm analyses since it can seriously affect the overall farm yield. In the present simulation, the used mesh is relatively coarse and the numerical schemes are on the diffusive end. Hence, such a detailed analysis loses its value.

Instead, Figure 7.8 presents the wake shape obtained at $t/T = 40$. In contrast with the results from section 6.4, now a clear wake shape is identified. Note that the wake is slightly skewed in the vertical direction, probably as a result of the rotor oscillations in pitch. Due to the coarse grid, the individual tip and root vortices rapidly merge together, leading to a cylindrical structure. From the work of Dos Santos [144], it is expected that individual shed vortices can be captured as the mesh is refined.

To better visualize wake development as it moves downstream, Figure 7.9 presents the wake velocity profile at three downwind positions. As expected, the wake core exhibits greater velocities close to the turbine but diffuses away from it. Note that none of the profiles is actually symmetrical along y, with a slightly higher velocity deficit towards the positive direction. This is an unexpected behaviour since the platform is fixed along y and all properties are symmetrical in this direction. Whether this lateral wake shift responds to the physics of the problem or is just an artifact of the chosen time instant is unknown. To discern between these two possibilities, time-averaged wake profiles should be analyzed instead.

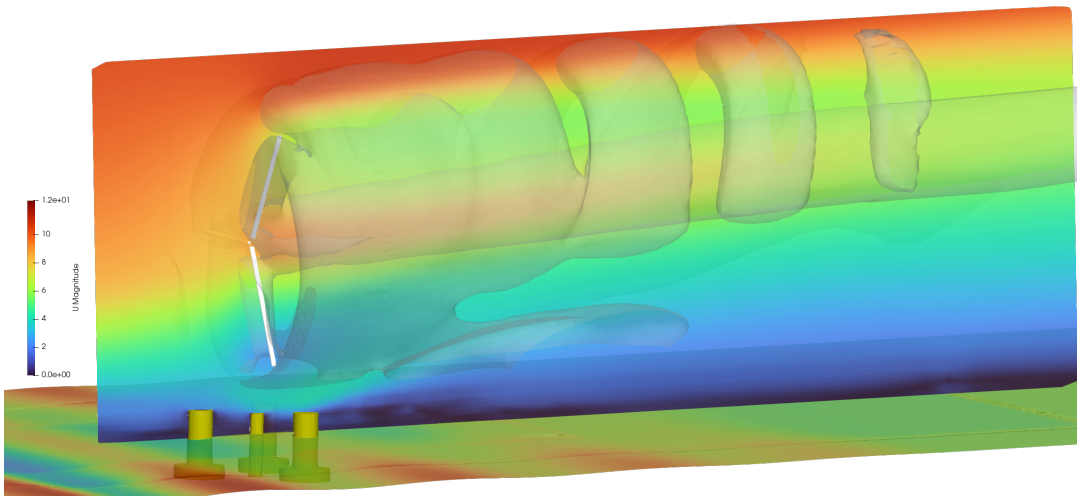


Figure 7.8: Snapshot of the FOWT at $t/T = 40$. The wake shape is obtained from the $Q = 0.001$ iso-surface.

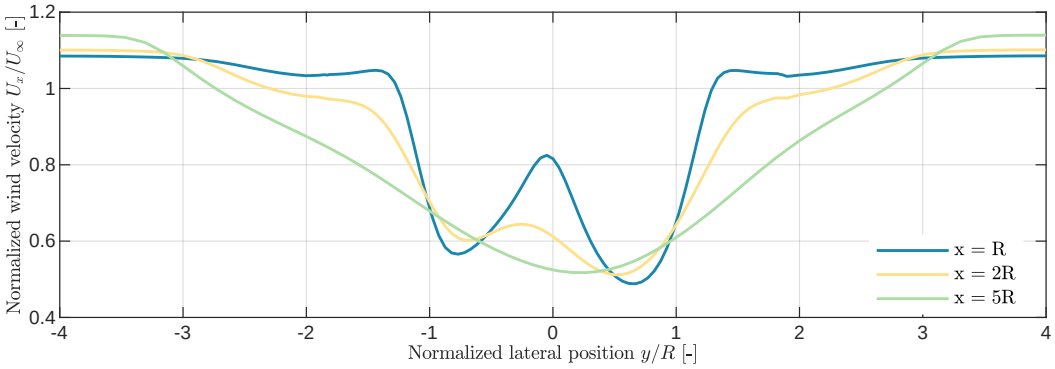


Figure 7.9: Wake velocity profile development as it progresses downwind. Results at hub height and $t/T = 40$.

7.3.3. Conclusions

In this section, an FSI simulation of a semi-submersible FOWT has been successfully performed using the numerical framework constructed throughout the thesis. The combination of the relaxation method for wave generation and absorption with the ALM for the aerodynamic modeling of the rotor has shown promising results. Even with a modest cell count ($\sim 2M$), the dynamics of the system could be well represented and captured. This method allows for the collection of interesting metrics regarding the floater's motion (amplitude and RAOs, velocity, nacelle acceleration), mooring lines (individual loads, stretching), aerodynamic performance (overall coefficients, blade local distributions, time-averaged properties), and the overall flow field (velocity contours, turbine wake).

One of the limitations of this study is the little focus dedicated to the mooring line model compared to its hydrodynamic and aerodynamic counterparts. Mooring loads have shown to drive the FOWT response in surge, at least for the analysed timespan. Recall how in section 5.3, after the simulation of a moored buoy, the mooring model was found to be the greatest source of discrepancies with the experimental results. Realistic FOWT simulations would need to consider the limitations of the mooring line model in more detail.

The present simulation was conceived as a proof of work to assess the capabilities of the developed framework. Even though it produced plausible results, there is no guarantee that it can accurately predict the coupled dynamics of FOWTs. For that to be proven, the results must be compared against different numerical and experimental data. In the next and final section, two new cases will be defined in an attempt to emulate a verification strategy.

7.4. Verification

Given the lack of both numerical results and experimental measurements of FOWTs under combined wind-waves loads, it is difficult to assess the quality of the results obtained in the previous section. In an attempt to replicate a verification campaign, two new simulations will be performed and the results compared with the previous one. First, a platform-only case where the ALM-rotor is substituted by a constant-thrust restraint. Then, a turbine-only case where the effects of platform motions are introduced by prescribing a turbine motion. The grid and numerical schemes have been kept as close as possible to the coupled FOWT set-up.

7.4.1. Platform-only simulation

In order to verify the hydrodynamic part of the coupled simulation, a new setup has been constructed without the effect of the turbine. The mesh consisted of ∼0.7M elements and the simulation took ∼13 hours to complete. Because there is no rotor nor wake, the domain is now shorter in all directions: 499.62m (5λ) long, 99.92 m (λ) wide, and 220m high (150m water, 70m air). The relaxation zone lengths have been shortened accordingly, with $L_g = \lambda$ and $L_a = 2\lambda$.

For this simulation to represent the same FOWT from the previous case, a constant value of thrust $T = 4.6 \cdot 10^5$N will be prescribed at hub height to resemble the effect of the turbine. This value corresponds to the mean rotor thrust at steady-state conditions from the coupled simulation. Ideally, the time-varying value of thrust should be imposed rather than its average, since these oscillations play a key role in the platform dynamics. However, this was ultimately discarded for the lack of time for the implementation of such restraint.

The final goal of this simulation is to assess how much the platform motion changes in comparison with the coupled simulation with an ALM-rotor. Contrarily to what was initially believed, the underlying physics of these two cases turned out to be surprisingly different as a result of applying a constant (instead of oscillating) thrust force. Nonetheless, the results shed light on the turbine effects on the platform dynamics and are hence considered relevant for discussion.

Initially, this setup was used to determine the optimal initial pitch angle θ_0 and mooring line offset m_0 that would result in the shortest transient phase for the coupled simulation. After some trials, it was decided to use θ_0 = 0º and $m_0 = -23$m. The mooring offset value is supposed to indicate the position at which mooring line forces are in equilibrium with the turbine thrust and wave loads. Apart from shortening the simulation timespan and thus alleviating the computational overhead, the mooring offset also reduces the displacement amplitude of the platform, which is appreciated when using a morphing mesh technique sensitive to large deformations.

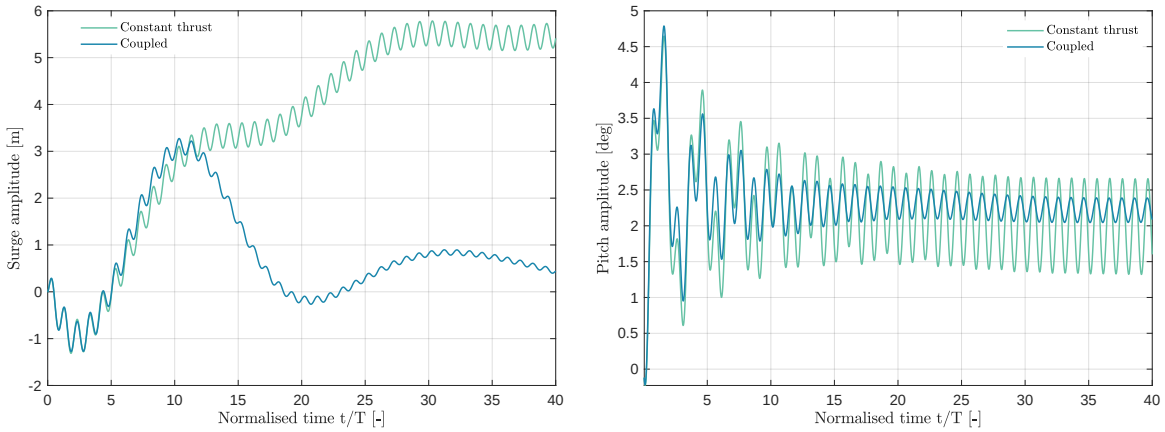
Platform motion

As with the previous case, platform dynamics are described by the time history of its position and velocity, depicted in figures 7.10 and 7.11 respectively. Plus, they are compared to the results obtained from the coupled-FOWT simulation. By inspecting the velocities, it is easy to identify a transient start followed by a steady-state periodic oscillation at wave frequency.

	Surge		**Pitch**	
	Position [m]	Velocity [m/s]	Position [deg]	Velocity [deg/s]
Mean	5.4340	7.93 E-4	1.9895	-3.70 E-4
Amplitude	0.2659	0.2101	0.6648	0.5239

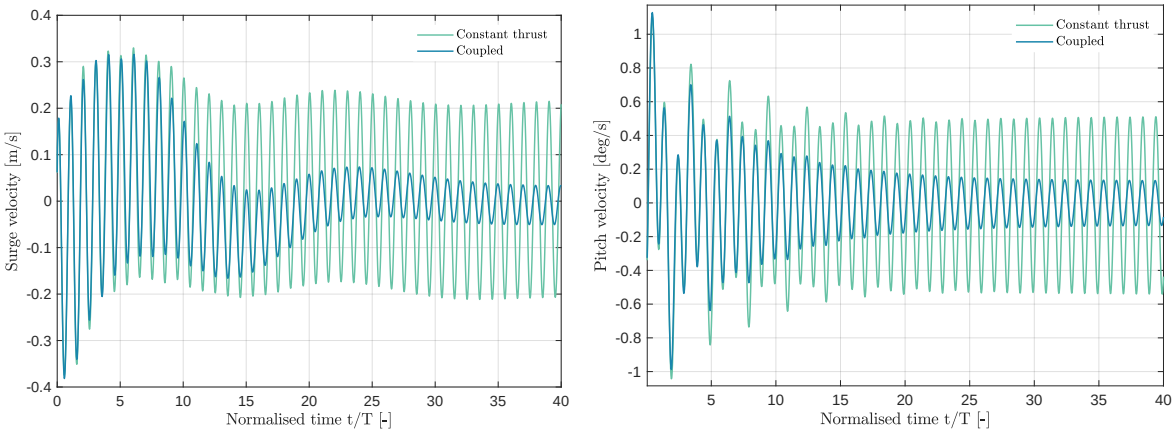
Table 7.7: Mean and amplitude values of the surge and pitch motions for the platform-only simulation between $34T \leq t \leq 40T$.

Although both simulations match nicely during the transient phase, the steady-state motion of the platform-only case presents much larger amplitudes (more than triple) than the coupled one. Yet, average values remain akin as indicated in Table 7.7. The highest disagreement between the two cases is found along surge, which achieves a higher equilibrium position ($\sim 5.4m$) as a result of the mean velocity in that direction being larger between $10T \leq t \leq 20T$. The amplitude of the wave-induced oscillations in surge is also larger than the one from the coupled FOWT.



(a) (b)

Figure 7.10: Time-evolution of the floater position: surge (a) and pitch (b).



(a) (b)

Figure 7.11: Time-evolution of the floater velocity: surge (a) and pitch (b).

All these differences between the two cases arise from the fact that the turbine-only simulation considered a constant thrust value. Remember that the aerodynamic loads from the coupled simulation were found to be in anti-phase with the platform velocity, meaning that the turbine will apply a higher thrust when the platform is moving upwind and vice versa. This oscillating behaviour ultimately results in the platform slowing down, since the turbine is always pushing back and forth in the opposite direction. This phenomenon is known as *aerodynamic damping* and will continue to diminish the motion amplitude until an equilibrium is reached with the wave loads.

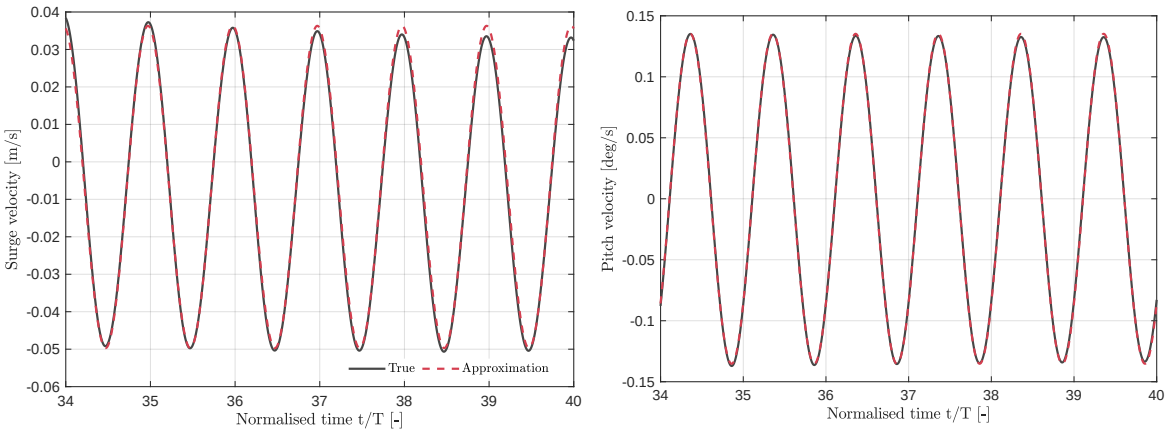
7.4.2. Turbine-only simulation

In an attempt to verify the aerodynamic model of the coupled simulation, a new single-phase set-up has been constructed without the effect of the platform or waves. Because there is no water phase, the domain has been shortened in the vertical direction, and the free-surface has been substituted by a slip-condition resembling the ground. The utilized mesh is an adaptation from the one used in Chapter 6, with cell refinement at the inlet and near the ground to better capture the imposed logarithmic wind profile. The mesh consisted of ∼0.6M elements and the simulation took ∼5 hours to complete.

To keep the aerodynamic loads similar to the ones experienced in the coupled FOWT simulation, a prescribed motion has been applied to the turbine that resembles the steady-state dynamics of the platform. For that purpose, an harmonic function at wave frequency has been fitted to the steady-state velocities from the coupled-FOWT case between $34T \leq t \leq 40T$. The imposed velocities are presented in equation 7.5 and compared to the originals in Figure 7.12. Positions can be easily obtained by

integrating expression 7.5.

$$\dot{x} = -0.0067 + 0.043 \sin\left(\frac{\pi}{4} t + 1.75\right), \quad \dot{\theta} = 0.1722 \cdot \frac{\pi}{4} \cos\left(\frac{\pi}{4} t + 4.03\right) \tag{7.5}$$



(a) Surge direction **(b)** Pitch direction

Figure 7.12: Comparison between prescribed and actual platform steady-state velocity.

Aerodynamic loads

The rotor power and thrust coefficients within the interval $34T \leq t \leq 40T$ are compared in figures 7.13 and 7.14, respectively. For reference, the results are also compared with those from a steady turbine. The mean and peak-to-peak values are summarised in Table 7.8. All cases produced very similar averages, with the prescribed motion case showing the largest, closely followed by the steady and coupled cases.

	C_P		C_T	
	Mean	P2P	Mean	P2P
Coupled	0.4847	0.0631	0.7319	0.0545
Prescribed	0.4927	0.0586	0.7423	0.0406
Steady	0.4921	0.0051	0.7416	0.0048

Table 7.8: Mean and peak-to-peak (P2P) values of power and thrust coefficient between $34T \leq t \leq 40T$.

The main difference appears in the peak-to-peak amplitude: while in the steady turbine the oscillations are 3P and induced exclusively by the wind shear, the prescribed and coupled simulations were also affected by the platform motions at wave frequency. These motions introduce variations in relative wind speed, which ultimately caused larger peak-to-peak amplitudes in the aerodynamic loads.

Still, a significant discrepancy is found between the coupled and prescribed motion cases. From Figure 7.14, it is clear that the wind shear 3P effect is much more intense in the coupled simulation than in the steady and prescribed motion ones. To explain this, the axial inflow speed at the blade tip is presented in Figure 7.15a. As expected, the wind shear intensity is higher in the coupled simulation, especially near the ground. This very low inflow speed is the cause behind the 3P drops in thrust coefficient in the coupled simulation.

To get a better insight on the root cause of this mismatch, the wind velocity profile at $x = -D$ and $t = 40T$ is represented in Figure 7.15b for both the coupled and prescribed-motion simulations along with the logarithmic profile imposed at the inlet boundary. Under $z/z_{\text{hub}} \sim 0.75$, the velocity from the coupled simulation falls below the other two profiles. The cause is thought to be the extra surface roughness introduced by the waves, which is absent in the prescribed motion case with slip ground. Eventually, the flow must speed up at higher elevations to fulfill the mass conservation principle. The

opposite is observed in the prescribed-motion case, where the velocity is slightly higher close to the ground and consequently lower at greater heights.

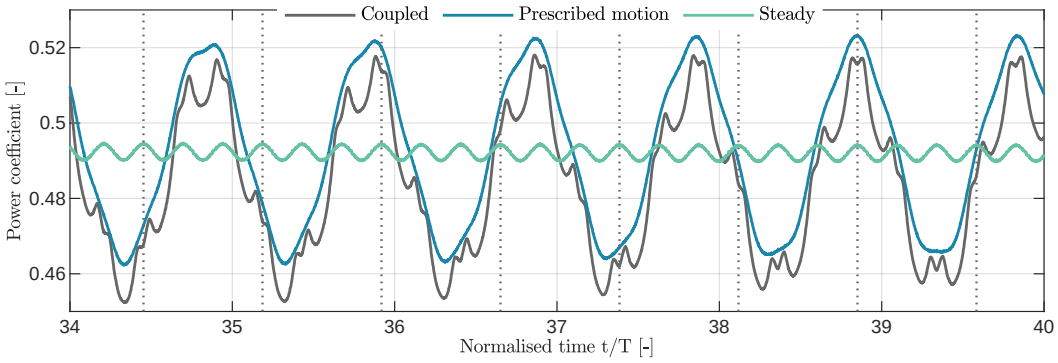


Figure 7.13: Steady-state power coefficient. Dotted lines represent the turbine rotation period.

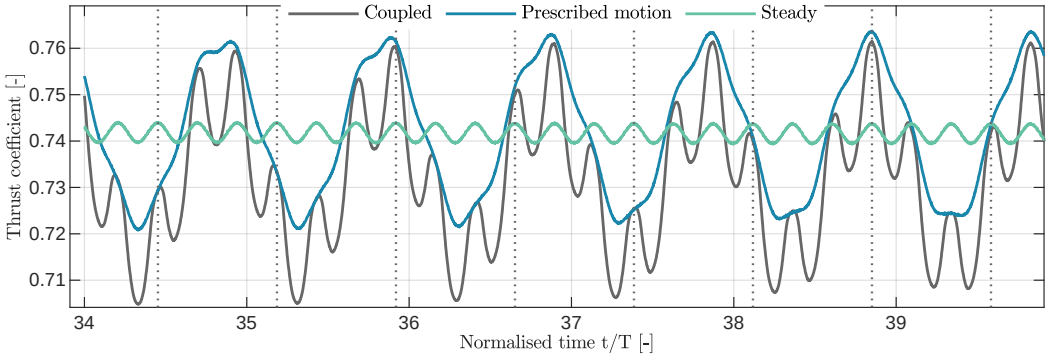
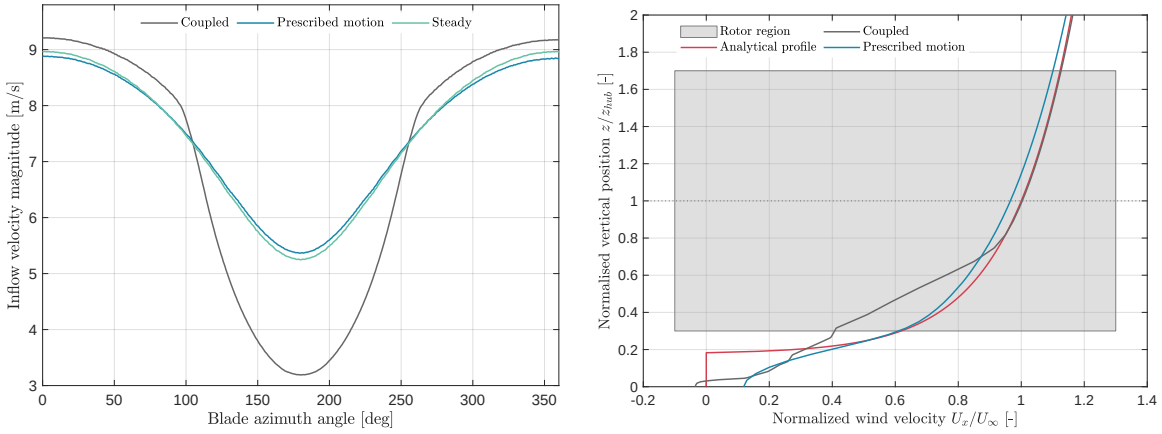


Figure 7.14: Steady-state thrust coefficient. Dotted lines represent the turbine rotation period.



(a) **(b)**

Figure 7.15: Axial inflow velocity at blade tip (a) and vertical wind profile at $x = -D$ (b).

7.4.3. Conclusions

Despite the success of the new platform-only and turbine-only simulations, the verification cannot be considered a complete win. Since a constant thrust was imposed in the platform-only simulation, the aerodynamic damping effect (which is induced by the oscillations in thrust) could not be captured. Hence, the platform motions from the platform-only case showed greater amplitudes and higher mooring loads, which ultimately led to a different surge equilibrium position. Nonetheless, the overall motion

dynamics did not differ much between the two cases. All in all, this simulation shed light on the turbine effect on the platform motions, revealing that rotor modeling is necessary in FOWT simulations: one cannot just impose the rated thrust and expect satisfactory results.

On the other hand, the turbine-only case was much like the coupled one. The main difference was introduced by the extra surface roughness induced by waves, which was absent in the turbine-only simulation, where the free-surface was replaced by a flat ground. This resulted in different vertical profiles, and hence the turbine-only case predicted lower amplitudes of the wind shear-induced 3P oscillations. Nonetheless, both average and peak-to-peak values of the thrust and power coefficients were almost coincident.

Even though more rigorous verification is still required with convergence and uncertainty analyses and comparison with other tools, the present section has laid the first stone in the path towards a robust and accurate model for the coupled FSI simulation of FOWTs in OpenFOAM.

8

Conclusions and further research

This chapter closes the thesis by presenting and discussing the main research results and conclusions. It starts with a short recap of the present project, followed by a summary of the key findings. The chapter finishes with a short piece of advice and recommendations for future research.

8.1. Thesis summary

In this thesis, a framework for the FSI simulation of FOWTs based on the relaxation zone method —for wave-field generation and absorption— and actuator line method —for rotor modeling— has been implemented in OpenFOAM. The work is constructed upon the *waves2Foam* library by Jacobsen [27] and *turbinesFoam* by Bachant [1]. The latter has been adapted to work with floating turbines and coupled to OpenFOAM's rigid body solver.

The thesis proceeded in a sequential style. First, the wave generation toolbox *waves2Foam* was used in a two-dimensional wave flume, which served as a first contact with CFD-based numerical wave tanks and proved suitable for the propagation of second-order waves. The lessons learned from this first case were then applied to the simulation of floating bodies. This second scenario required both the usage of a morphing mesh technique and a rigid body model, coupled to the *wavesDyMFoam* solver by means of a serial sub-iterating FSI strategy based on the *PIMPLE* algorithm. This setup was successfully applied to two different cases: the free decay of a 2D cylinder in heave and the moored decay of a 3D floater in pitch, the latter based on the quasi-steady mooring line model from *waves2Foam*. Results were compared to both experimental measurements and available numerical data.

Moving to the aerodynamic model, *turbinesFoam* was successfully modified to account for turbine motions. The user can now define a prescribed motion in any of the six rigid-body DoFs and impose it on the turbine. This motion will produce changes in orientation and relative velocities that result in loading conditions that differ from the steady rotor case. To test whether the aerodynamic behaviour of a moving turbine could be successfully captured by this modified library, the experiment carried out during the OC6 Phase III campaign was replicated. Single DoF motions in surge and pitch were imposed at different frequencies and amplitudes, with the overall results being in reasonable accordance with the experimental measurements and those of other participants.

Finally, all the pieces of the puzzle were put together in the coupled simulation of a full FOWT. Prior to that, the FSI coupling between the ALM rotor and rigid-body platform was implemented: while the turbine moves following the floater's motion, the latter is driven by the loads from the ALM rotor. This two-sided coupling was again built based on a serial sub-iterating FSI strategy via the *PIMPLE* algorithm. The resulting framework was put to the test in the simulation of a full-scale, 2-DoF (surge and pitch) semi-submersible FOWT under combined wave-wind conditions. Seemingly plausible results were obtained, which were then compared with those from non-coupled (platform-only and turbine-only) simulations.

All these steps have been presented in a detailed and comprehensive manner throughout the report

so that they can be replicated by any interested reader.

8.2. Final conclusions

A summary of conclusions is presented for the three main building blocks of this thesis:

Hydrodynamic model

- The relaxation zone method from *waves2Foam* has proven adequate for the generation and absorption of waves, although only second-order non-oblique waves have been tested.
- The combination of the morphing mesh technique with OpenFOAM's rigid body model has been considered adequate for the simulation of floating bodies with moderate motions after testing it in three different decay tests.
- A strategy based on serial sub-iterations between fluid and rigid-body solvers within the PIMPLE loop allows for a strong FSI coupling, although more optimized methodologies could be conceived.
- The mooring line model is thought to be the main source of mismatch with the experiment in the moored decay case. Switching from a quasi-steady catenary model to a dynamic one will likely increase the accuracy of the results.

Aerodynamic model

- Even though small bugs had to be fixed along the way, the ALM from *turbinesFoam* proved suitable for use with floating wind turbines. The library has been modified and now allows for the definition of any arbitrary prescribed motion along the six rigid-body degrees of freedom.
- The modified *turbinesFoam* is able to predict the aerodynamic performance of a turbine with prescribed motion, at least in the conditions of the OC6 Phase III experiment. However, the prediction is not necessarily better than that of lower-fidelity tools, hence questioning the use of high-fidelity tools in the considered load-cases.
- The ALM has been found susceptible to the small variations in inflow speed caused by the fact that the rotor was not placed in the center of the wind-tunnel section. This has been referred to as *"stream-tube effect"* and is responsible for 2P and 4P oscillations in blade loads. It is unclear whether the same effect would also be observed in an experimental setup or in blade-resolved CFD simulations.

Coupled model

- Once more, the serial sub-iterating strategy proved successful in strongly coupling the rigid-body floater and ALM rotor. Because these two elements are defined in different libraries, a communication framework was implemented in OpenFOAM based on the use of *IOdictionaries* and the PIMPLE algorithm.
- It is possible to simulate a full FOWT by integrating *waves2Foam* (for wave generation-absorption and a quasi-steady mooring model) with the modified version of *turbinesFoam* (for ALM modeling of moving turbines) and OpenFOAM's rigid body solver along with the morphing mesh technique.
- The resulting simulation framework has been tested for a semi-submersible, 2-DoF (surge and pitch) FOWT under combined wind-wave conditions. The results showed that even though the pitch response was driven by the turbine thrust, the surge displacements were largely affected by the mooring loads. The average aerodynamic coefficients were similar to those from a steady turbine, while significant wave-frequency oscillations were observed that can potentially induce higher blade fatigue.
- Although the computational cost of the coupled case is $\sim$4 times higher than for a platform-only simulation, it was observed that the floater-only setup fails in predicting the true dynamics of FOWTs as it lacks the aerodynamic damping induced by the moving rotor.
- Since these simulations are computationally intensive, they should be reserved for very specific cases where lower fidelity tools cannot accurately predict the FOWT's dynamics.

The author has tried his best to present the construction process of the current simulation framework in a sequential and comprehensive style so that it can be easily followed by the reader, no matter his or her initial experience with OpenFOAM. The extensive first chapters are meant as an introduction to the topic of floating turbines, their governing principles, and main modeling approaches. The remaining chapters tackle the high-fidelity simulation of FOWTs piece by piece, based on the relaxation zones and actuator line method. With the presented evidence, the main thesis' objective can be considered fulfilled:

> **" To develop a comprehensive and accessible framework for the high-fidelity modelling of a full FOWT based on the actuator line and wave relaxation methods by performing and thoroughly documenting the CFD-FSI simulation of a FOWT in OpenFOAM using the *waves2foam* and *turbinesFoam* libraries "**

8.3. Further research

In the end, the main thesis' objective was accomplished, and a framework for the CFD simulation of FOWTs was successfully implemented in OpenFOAM. Still, the scope of this project was limited, and different areas exist that could be expanded or improved. This section will explore such elements and present the potential directions that further research might take.

Verification and validation

Although the results obtained throughout the present research have been carefully compared with both numerical and experimental data, the procedure followed cannot be considered a strict verification. For the implemented simulation framework to be scientifically relevant, its accuracy needs to be proven through verification and validation.

- **Evaluate the uncertainty associated with the simulation** by performing time-step and mesh convergence analyses [139]. Try to identify which numerical schemes and solvers are best suited for the sought-after case. Also check for the influence of the different ALM parameters (e.g., number of actuator elements, amount of inflow sample points, projection width) and turbulence models on the results.
- **Verify the framework** by comparing the results with other numerical tools, preferably using a range of environmental conditions. Consider motions in all DoFs and account for the gyroscopic moment. It is possible that no results will be available for the considered load-case and FOWT configuration. In such a case, one should consider producing the results oneself (at least for the lower-fidelity tools, e.g. with *OpenFAST*).
- **Validate the framework** by replicating available experimental measurements and, if possible, with other numerical tools. At this point, it is important to understand the limitations of the model in use and take uncertainties into consideration.

Further improvements

Different choices had to be made when developing the simulation framework. Now we ask how different models and techniques would have improved the overall result.

- **Use more advanced FSI techniques** for the strong partitioned coupling between fluid and rigid-body equations. So far, only a basic coupling based on serial sub-iterations has been considered, given the straightforward implementation with the PIMPLE algorithm. More advanced solutions exist that can help reduce computational cost and minimize the interface residual, such as fixed or Aitken under-relaxation and IQN-ILS (quasi-Newton methods, better than fixed-point iterations [51]). Simpler techniques propose solving the rigid body equations and mesh motion less frequently than the fluid equations (i.e. sub-cycling [138]). Some of these methods, along with a more efficient morphing mesh technique based on radial basis functions (RBFs), are available through the *foam-FSI* library by David Blom [60].
- **Advocate for a higher-fidelity mooring line model**. Within this research, the mooring lines have been modeled based on the quasi-steady solution from the catenary equations. The mooring line solver is usually much cheaper than CFD, while it has a noticeable effect on the overall platform motions, as observed in sections 5.3 and 7.4. Consequently, the overall accuracy of the

framework could be largely improved by considering high-fidelity mooring models (e.g. lumped-mass, multi-body, FEM) while only minimally increasing the cost.

— **Extend the method to VAWTs**. The current implementation is only available for horizontal-axis wind turbines. Extension to VAWTs shouldn't pose a challenge: within *turbinesFOAM*, the modifications made in *axialFlowTurbineALSource* must be applied to *crossFlowTurbineALSource*.

— **Extend the method to multiple turbines**. For that to work, the dynamic mesh technique and rigid body solver must allow for multiple bodies. The OpenFOAM `rigidBodyDynamics` library extends the `sixDoFRigidBodyMotion` capabilities to various bodies.

— **Use an overset mesh for the floater** [121]. All the simulations performed in this thesis rely on the morphing mesh technique, a cheap yet robust method that can be easily linked to a rigid body. However, for large deformations, it can result in poor quality grids that ultimately stall the simulation. Overset meshes are more intricate and usually demand larger computational resources, but they can accommodate arbitrarily large motions.

— **Explore other wave generation techniques**. The relaxation zone technique has proven accurate and robust, but it requires higher cell counts due to the additional domain length. In coupled simulations, this might not be an issue since the turbine simulation already demands extending the domain in the axial direction. Nonetheless, it would still be interesting to try different wave generation techniques such as generating-absorbing boundary conditions (GABC, already available within *waves2Foam*) or other libraries like *olaFlow* or *IHFoam*.

— **Expand the framework**. For more realistic modeling, consider including flexible blades (a simple beam model would suffice) or control strategies for the rotor and platform.

Simulation of FOWTs

The present project focused on the development of a simulation framework that hasn't yet been put into practice.

— **Perform a coupled simulation of an FOWT**. Select an engineering-relevant FOWT configuration and use the ALM framework to characterize its response (e.g. under different sea states and TSR). Given how expensive these simulations are, only a handful of carefully selected load cases can be considered. If the framework has been extended to allow for multiple turbines, wind-farm simulations may be attempted.

— **Use the framework to improve lower-fidelity tools**. One of the objectives of the CFD simulation of FOWTs is to gain insight on the complex aero-hydrodynamic phenomena driving the turbine. Once the framework has been validated and the results can be trusted, its predictions can be used to develop engineering corrections that increase the accuracy of lower-fidelity methods or train surrogate models.

Framework optimisation

The main priority when implementing this framework was to make it workable, not efficient. Different approaches can be followed to diminish the associated computational cost.

— **Investigate the benefits of a partitioned simulation approach**. The framework presented in this thesis is based on a monolithic CFD approach (VOF method) meaning that the same solver is used for hydrodynamics and aerodynamics. This, however, implies that the same schemes, solvers, and turbulence models must be shared between the turbine and platform. Moreover, the longer domain required by the turbine in the axial direction also increases the free-surface area, which requires fine grids and thus increases the cell count. If different domains were used for the turbine and platform simulations, then their sizes, grids, and numerical methods could be set independently. This however comes at the cost of a partitioning error, which ultimately requires multiple iterations (and a higher cost) to achieve a coupled solution. OpenFOAM offers in-built strategies for multi-region simulations.

— **Use adaptive mesh refinement (AMR)**. Given the dynamic behaviour of FTWs, the near-rotor refinement zone needs to be wide enough to accommodate the turbine motions. An AMR technique based on the ALM force-field would refine the mesh only in the vicinity of the rotor, reducing the total cell count. The same could be applied to the free surface based on the gradient of the

indicator field. Still, it is unclear whether the reduction in the number of elements compensates for the increased cost of AMR.

— **Parallelize the ALM turbine motion procedure**. When *turbinesFoam* was modified to accommodate for arbitrary turbine motions, the code was implemented in a serial manner, meaning that the motion routines of all the actuator elements are executed by a single processor. However, CFD calculations commonly run on multi-core machines. Because the motion of one element is independent of the others', these motion routines could highly benefit from parallelization.

References

[1] Peter Bachant, Anders Goude, and Martin Wosnik. “Actuator line modeling of vertical-axis turbines”. In: *arXiv preprint arXiv:1605.01449* (2016).

[2] Niels G Jacobsen. “waves2foam manual”. In: *Deltares, The Netherlands* 570 (2017).

[3] Sōichi Itō. “Study of the transient heave oscillation of a floating cylinder.” PhD thesis. Massachusetts Institute of Technology, 1977.

[4] Johannes Palm, Claes Eskilsson, Guilherme Moura Paredes, and Lars Bergdahl. “Coupled mooring analysis for floating wave energy converters using CFD: Formulation and validation”. In: *International Journal of Marine Energy* 16 (2016), pp. 83–99.

[5] Roger Bergua et al. “OC6 project phase III: validation of the aerodynamic loading on a wind turbine rotor undergoing large motion caused by a floating support structure”. In: *Wind Energy Science Discussions* (2022), pp. 1–33.

[6] Ove Hoegh-Guldberg et al. “Impacts of 1.5 C global warming on natural and human systems”. In: *Global warming of 1.5 C. An IPCC Special Report* (2018).

[7] Lee Joyce and Zhao Feng. *GWEC Global Wind Report*. Global Wind Energy Council, Mar. 2021.

[8] Hannah Ritchie and Max Roser. *CO2 and Greenhouse Gas Emissions*. 2020. URL: https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions.

[9] Birte Holst, Hannele Holttinen, and Charlotte Hede. *IEA Wind TCP Annual Report 2020*. International Energy Agency, 2020.

[10] Burton Tony et al. “Wind Energy Handbook”. In: 3rd ed. John Wiley & Sons Ltd., 2021. ISBN: 9781119451099.

[11] Laure Cozzi et al. *Offshore Wind Outlook 2019: World Energy Outlook Special Report*. International Energy Agency, 2019.

[12] Chong Ng and Li Ran. *Offshore wind farms: Technologies, design and operation*. Woodhead Publishing, 2016.

[13] Lizet Ramírez, Daniel Fraile, and Guy Brindley. *Offshore Wind in Europe*. Wind Europe, Feb. 2021.

[14] James Rhodri and Marc Costa Ros. *Floating Offshore Wind: Market and Technology Review*. Carbon Trust on behalf of the Scottish Government, June 2015.

[15] Sandy Butterfield, Walt Musial, Jason Jonkman, and Paul Sclavounos. *Engineering challenges for floating offshore wind turbines*. Tech. rep. National Renewable Energy Lab. (NREL), Golden, CO (United States), 2007.

[16] Denis Matha, Markus Schlipf, Ricardo Pereira, and Jason Jonkman. “Challenges in simulation of aerodynamics, hydrodynamics, and mooring-line dynamics of floating offshore wind turbines”. In: *The Twenty-first International Offshore and Polar Engineering Conference*. OnePetro. 2011.

[17] Amy N Robertson et al. “OC5 project phase II: validation of global loads of the DeepCwind floating semisubmersible wind turbine”. In: *Energy Procedia* 137 (2017), pp. 38–57.

[18] Jens Norkær Sorensen and Wen Zhong Shen. “Numerical modeling of wind turbine wakes”. In: *J. Fluids Eng.* 124.2 (2002), pp. 393–399.

[19] J Jonkman and Denis Matha. *Quantitative comparison of the responses of three floating platforms*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2010.

[20] Cruz Joao and Atcheson Mairead. “Floating Offshore Wind Energy: The Next Generation of Wind Energy”. In: Springer, 2016.

[21] Erlend Gjelstad. *Oceans Unlocked - A Floating Wind Future*. June 2021. URL: https://www.cowi.com/insights/oceans-unlocked-a-floating-wind-future.

[22] Andrew J Goupee et al. “Experimental comparison of three floating wind turbine concepts”. In: *Journal of Offshore Mechanics and Arctic Engineering* 136.2 (2014).

[23] Michael Borg, Maurizio Collu, and Athanasios Kolios. “Offshore floating vertical axis wind turbines, dynamics modelling state of the art. Part II: Mooring line and structural dynamics”. In: *Renewable and Sustainable Energy Reviews* 39 (2014), pp. 1226–1234.

[24] Cheng Liu and Changhong Hu. “CFD simulation of a floating wind turbine platform in rough sea conditions”. In: *The Twenty-fourth International Ocean and Polar Engineering Conference*. OnePetro. 2014.
[25] Ping Cheng, Yang Huang, and Decheng Wan. “A numerical model for fully coupled aero-hydrodynamic analysis of floating offshore wind turbine”. In: *Ocean Engineering* 173 (2019), pp. 183–196.
[26] HM Johlas et al. “Large eddy simulations of offshore wind turbine wakes for two floating platform types”. In: *Journal of Physics: Conference Series*. Vol. 1452. 1. IOP Publishing. 2020, p. 012034.
[27] Niels G Jacobsen, David R Fuhrman, and Jørgen Fredsøe. “A wave generation toolbox for the open-source CFD library: OpenFoam®”. In: *International Journal for numerical methods in fluids* 70.9 (2012), pp. 1073–1088.
[28] John Nicholas Newman. *Marine hydrodynamics*. The MIT press, 2018.
[29] Lorenzo Cottura et al. “Dynamic modeling of an offshore floating wind turbine for application in the Mediterranean Sea”. In: *Energies* 14.1 (2021), p. 248.
[30] M Borg and M Collu. “A comparison between the dynamics of horizontal and vertical axis offshore floating wind turbines”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 373.2035 (2015), p. 20140076.
[31] Walter H Munk. *Origin and generation of waves*. Tech. rep. Scripps Institution of Oceanography La Jolla Calif, 1951.
[32] John D Fenton. “Nonlinear wave theories”. In: *Ocean Engineering Science* 9 (1990), pp. 1–18.
[33] Det Norske Veritas. “Design of Offshore Wind Turbine Structure”. In: *Offshore Standard DNV-OS-J101* (2013).
[34] TS Hedges and URSELL. “Regions of validity of analytical wave theories.” In: *Proceedings of the Institution of Civil Engineers-Water Maritime and Energy* 112.2 (1995), pp. 111–114.
[35] William Finnegan and Jamie Goggins. “Numerical simulation of linear water waves and wave–structure interaction”. In: *Ocean Engineering* 43 (2012), pp. 23–31.
[36] Coastal Engineering Research Center (US). *Shore protection manual*. Vol. 1. US Army Coastal Engineering Research Center, 1977.
[37] Michael Borg and Maurizio Collu. “Offshore floating vertical axis wind turbines, dynamics modelling state of the art. Part III: Hydrodynamics and coupled modelling approaches”. In: *Renewable and sustainable energy reviews* 46 (2015), pp. 296–310.
[38] RCT Rainey. “Weak or strong nonlinearity: the vital issue”. In: *Journal of Engineering Mathematics* 58.1 (2007), pp. 229–249.
[39] Cyril W Hirt and Billy D Nichols. “Volume of fluid (VOF) method for the dynamics of free boundaries”. In: *Journal of computational physics* 39.1 (1981), pp. 201–225.
[40] Pál Schmitt et al. “Beyond VoF: alternative OpenFOAM solvers for numerical wave tanks”. In: *Journal of ocean engineering and marine energy* 6.3 (2020), pp. 277–292.
[41] Edin Berberović et al. “Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution”. In: *Physical Review E* 79.3 (2009), p. 036306.
[42] Luis Martinez, Stefano Leonardi, Matthew Churchfield, and Patrick Moriarty. “A comparison of actuator disk and actuator line wind turbine models and best practices for their use”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2012, p. 900.
[43] Pankaj Jha, Matthew Churchfield, Patrick Moriarty, and Sven Schmitz. “Accuracy of state-of-the-art actuator-line modeling for wind turbine wakes”. In: *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2013, p. 608.
[44] Matthew J Churchfield et al. “An advanced actuator line method for wind energy applications and beyond”. In: *35th Wind Energy Symposium*. 2017, p. 1998.
[45] Alexander R Meyer Forsting, Georg R Pirrung, and Néstor Ramos-García. “The wake of an actuator line with a vortex-based tip/smearing correction in uniform and turbulent inflow”. In: *Journal of Physics: Conference Series*. Vol. 1256. 1. IOP Publishing. 2019, p. 012020.
[46] Nikos Spyropoulos, George Papadakis, John M Prospathopoulos, and Vasilis A Riziotis. “Investigating the Level of Fidelity of an Actuator Line Model in Predicting Loads and Deflections of Rotating Blades under Uniform Free-Stream Flow”. In: *Applied Sciences* 11.24 (2021), p. 12097.
[47] Matthew J Churchfield, Sang Lee, Sven Schmitz, and Zhixiang Wang. “Modeling wind turbine tower and nacelle effects within an actuator line model”. In: *33rd Wind Energy Symposium*. 2015, p. 0214.

[48] Idriss Ammara, Christophe Leclerc, and Christian Masson. “A viscous three-dimensional differential actuator-disk method for the aerodynamic analysis of wind farms”. In: *J. Sol. Energy Eng.* 124.4 (2002), pp. 345–356.

[49] H Daaou Nedjari, O Guerri, and M Saighi. “Full rotor modelling and generalized actuator disc for wind turbine wake investigation”. In: *Energy Reports* 6 (2020), pp. 232–255.

[50] N Bruinsma, BT Paulsen, and NG Jacobsen. “Validation and application of a fully nonlinear numerical wave tank for simulating floating offshore wind turbines”. In: *Ocean Engineering* 147 (2018), pp. 647–658.

[51] Alexander J Dunbar, Brent A Craven, and Eric G Paterson. “Development and validation of a tightly coupled CFD/6-DOF solver for simulating floating offshore wind turbine platforms”. In: *Ocean Engineering* 110 (2015), pp. 98–105.

[52] Jia-hao Chen, Ai-guo Pei, Peng Chen, and Zhi-qiang Hu. “Study on Gyroscopic Effect of Floating Offshore Wind Turbines”. In: *China Ocean Engineering* 35.2 (2021), pp. 201–214.

[53] Jason Jonkman and Walter Musial. *Offshore code comparison collaboration (OC3) for IEA Wind Task 23 offshore wind technology and deployment*. Tech. rep. National Renewable Energy Lab. (NREL), Golden, CO (United States), 2010.

[54] Christian Windt, Josh Davidson, Benazzou Akram, and John V Ringwood. “Performance assessment of the overset grid method for numerical wave tank experiments in the OpenFOAM environment”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 51319. American Society of Mechanical Engineers. 2018, V010T09A006.

[55] Christian Windt et al. “Evaluation of the overset grid method for control studies of wave energy converters in OpenFOAM numerical wave tanks”. In: *Journal of Ocean Engineering and Marine Energy* 6.1 (2020), pp. 55–70.

[56] Griet Decorte, Geert Lombaert, and Jaak Monbaliu. “A first assessment of the interdependency of mesh motion and free surface models in OpenFOAM regarding wave-structure interaction”. In: *MARINE VIII: proceedings of the VIII International Conference on Computational Methods in Marine Engineering*. CIMNE. 2019, pp. 795–806.

[57] Gerhard Holzinger. “OpenFOAM–a Little User-Manual”. In: *CD-Laboratory-Particulate Flow Modelling, Johannes Keplper University: Linz, Austria* (2015).

[58] Tobias Holzmann. “Mathematics, numerics, derivations and OpenFOAM®”. In: *Loeben, Germany: Holzmann CFD* (2016).

[59] C Greenshields and H Weller. “Notes on Computational Fluid Dynamics: General Principles”. In: *CFD Direct Ltd.: Reading, UK* (2022).

[60] David Blom. *FOAM-FSI*. https://github.com/davidsblom/FOAM-FSI. 2016.

[61] Emmanuel Branlard. *Wind turbine aerodynamics and vorticity-based methods: Fundamentals and recent applications*. Vol. 7. Springer, 2017.

[62] David M Eggleston and Forrest Stoddard. “Wind turbine engineering design”. In: (1987).

[63] LJ Vermeer, Jens Nørkær Sørensen, and Antonio Crespo. “Wind turbine wake aerodynamics”. In: *Progress in aerospace sciences* 39.6-7 (2003), pp. 467–510.

[64] Wei Zhang, Corey D Markfort, and Fernando Porté-Agel. “Near-wake flow structure downwind of a wind turbine in a turbulent boundary layer”. In: *Experiments in fluids* 52.5 (2012), pp. 1219–1235.

[65] Fernando Porté-Agel, Majid Bastankhah, and Sina Shamsoddin. “Wind-turbine and wind-farm flows: a review”. In: *Boundary-Layer Meteorology* 174.1 (2020), pp. 1–59.

[66] Thomas Sebastian. “The aerodynamics and near wake of an offshore floating horizontal axis wind turbine”. PhD thesis. University of Massachusetts Amherst, 2012.

[67] Jing Dong and Axelle Viré. “Comparative analysis of different criteria for the prediction of vortex ring state of floating offshore wind turbines”. In: *Renewable Energy* 163 (2021), pp. 882–909.

[68] Hakjin Lee and Duck-Joo Lee. “Effects of platform motions on aerodynamic performance and unsteady wake evolution of a floating offshore wind turbine”. In: *Renewable Energy* 143 (2019), pp. 9–23.

[69] Ryan Kyle, Yeaw Chu Lee, and Wolf-Gerrit Früh. “Propeller and vortex ring state for floating offshore wind turbines during surge”. In: *Renewable Energy* 155 (2020), pp. 645–657.

[70] Jing Dong, Axelle Viré, and Zhangrui Li. “Analysis the vortex ring state and propeller state of floating offshore wind turbines and verification of their prediction criteria by comparing with a CFD model”. In: *Renewable Energy* 184 (2022), pp. 15–25.

[71] Yuan Fang et al. “Effect of surge motion on rotor aerodynamics and wake characteristics of a floating horizontal-axis wind turbine”. In: *Energy* 218 (2021), p. 119519.
[72] Haoran Meng et al. “Experimental investigation on the power and thrust characteristics of a wind turbine model subjected to surge and sway motions”. In: *Renewable Energy* 181 (2022), pp. 1325–1337.
[73] Caroline Lienard, Ronan Boisard, and Camille Daudin. “Aerodynamic behavior of a floating offshore wind turbine”. In: *AIAA Journal* 58.9 (2020), pp. 3835–3847.
[74] Thanh Toan Tran and Dong-Hyun Kim. “A CFD study into the influence of unsteady aerodynamic interference on wind turbine surge motion”. In: *Renewable Energy* 90 (2016), pp. 204–228.
[75] Russell Farrugia, Tonio Sant, and Daniel Micallef. “A study on the aerodynamics of a floating wind turbine rotor”. In: *Renewable energy* 86 (2016), pp. 770–784.
[76] Binrong Wen et al. “On the power coefficient overshoot of an offshore floating wind turbine in surge oscillations”. In: *Wind Energy* 21.11 (2018), pp. 1076–1091.
[77] Thanh-Toan Tran and Dong-Hyun Kim. “The platform pitching motion of floating offshore wind turbine: A preliminary unsteady aerodynamic analysis”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 142 (2015), pp. 65–81.
[78] Minu Jeon, Seungmin Lee, and Soogab Lee. “Unsteady aerodynamics of offshore floating wind turbines in platform pitching motion using vortex lattice method”. In: *Renewable Energy* 65 (2014), pp. 207–212.
[79] Ziwen Chen, Xiaodong Wang, and Shun Kang. “Effect of the Coupled Pitch–Yaw Motion on the Unsteady Aerodynamic Performance and Structural Response of a Floating Offshore Wind Turbine”. In: *Processes* 9.2 (2021), p. 290.
[80] Jun Wu, Jin-hong Ding, Yan-ping He, and Yong-sheng Zhao. “Study on unsteady aerodynamic performance of floating offshore wind turbine by CFD method”. In: *The Twenty-fifth International Ocean and Polar Engineering Conference*. OnePetro. 2015.
[81] Morteza Khosravi. “An experimental study on the near wake characteristics of a wind turbine model subjected to surge, pitch, and heave motions”. PhD thesis. MS thesis (Iowa State University, 2015), 2015.
[82] Binrong Wen et al. “Wind shear effect induced by the platform pitch motion of a spar-type floating wind turbine”. In: *Renewable energy* 135 (2019), pp. 1186–1199.
[83] Danmei Hu, Liwei Deng, and Li Zeng. “Study on the Aerodynamic Performance of Floating Offshore Wind Turbine Considering the Tower Shadow Effect”. In: *Processes* 9.6 (2021), p. 1047.
[84] Stanislav Rockel et al. “Experimental study on influence of pitch motion on the wake of a floating wind turbine model”. In: *Energies* 7.4 (2014), pp. 1954–1985.
[85] Aldert Otter et al. “A review of modelling techniques for floating offshore wind turbines”. In: *Wind Energy* (2021).
[86] Amy Robertson et al. “Offshore code comparison collaboration continuation within IEA wind task 30: Phase II results regarding a floating semisubmersible wind system”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 45547. American Society of Mechanical Engineers. 2014, V09BT09A012.
[87] Carlos Ferreira, Wei Yu, Arianna Sala, and Axelle Viré. “Dynamic inflow model for a Floating Horizontal Axis Wind Turbine in surge motion”. In: *Wind Energy Science Discussions* (2021), pp. 1–22.
[88] Jing Dong et al. “A modified free wake vortex ring method for horizontal-axis wind turbines”. In: *Energies* 12.20 (2019), p. 3900.
[89] Jeanie Aird, Evan Gaertner, and Matthew Lackner. “Dynamic prescribed-wake vortex method for aerodynamic analysis of offshore floating wind turbines”. In: *Wind Engineering* 43.1 (2019), pp. 47–63.
[90] Néstor Ramos-García et al. “Investigation of the floating IEA Wind 15 MW RWT using vortex methods Part I: Flow regimes and wake recovery”. In: *Wind Energy* (2021).
[91] BF Xu, TG Wang, Y Yuan, and JF Cao. “Unsteady aerodynamic analysis for offshore floating wind turbines under different wind conditions”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 373.2035 (2015), p. 20140080.
[92] Russell Farrugia, Tonio Sant, and Daniel Micallef. “Investigating the aerodynamic performance of a model offshore floating wind turbine”. In: *Renewable Energy* 70 (2014), pp. 24–30.

[93] Rèmi Corniglion, Jeffrey Harris, Christophe Peyrard, and Matteo Capaldo. “Comparison of the free vortex wake and actuator line methods to study the loads of a wind turbine in imposed surge motion”. In: *Journal of Physics: Conference Series*. Vol. 1618. 5. IOP Publishing. 2020, p. 052045.

[94] Sasan Sarmast, Antonio Segalini, Robert F Mikkelsen, and Stefan Ivanell. “Comparison of the near-wake between actuator-line simulations and a simplified vortex model of a horizontal-axis wind turbine”. In: *Wind Energy* 19.3 (2016), pp. 471–481.

[95] Andrea Ortolani et al. “Computational Fluid Dynamics Analysis of Floating Offshore Wind Turbines in Severe Pitching Conditions”. In: *Journal of Engineering for Gas Turbines and Power* 142.12 (2020).

[96] Yang Huang, Ping Cheng, and Decheng Wan. “Numerical analysis of a floating offshore wind turbine by coupled aero-hydrodynamic simulation”. In: *Journal of Marine Science and Application* 18.1 (2019), pp. 82–92.

[97] DD Apsley and PK Stansby. “Unsteady thrust on an oscillating wind turbine: comparison of blade-element momentum theory with actuator-line CFD”. In: *Journal of Fluids and Structures* 98 (2020), p. 103141.

[98] Pengfei Li, Ping Cheng, DC Wan, and Qing Xiao. “Numerical simulations of wake flows of floating offshore wind turbines by unsteady actuator line model”. In: *Proceedings of the 9th international workshop on ship and marine hydrodynamics, Glasgow, UK*. 2015, pp. 26–28.

[99] HM Johlas et al. “Large eddy simulations of floating offshore wind turbine wakes with coupled platform motion”. In: *Journal of Physics: Conference Series*. Vol. 1256. 1. IOP Publishing. 2019, p. 012018.

[100] Shuolin Xiao and Di Yang. “Large-eddy simulation-based study of effect of swell-induced pitch motion on wake-flow statistics and power extraction of offshore wind turbines”. In: *Energies* 12.7 (2019), p. 1246.

[101] Filippo Negroni. “Pitch and surge motion effects on floating wind turbine unsteady aerodynamics”. In: (2021).

[102] Paolo Schito. “Large Eddy Simulation of wind turbines: interaction with turbulent flow”. In: (2012).

[103] Arianna Sala. “CFD study of induction and finite blade effect on a floating horizontal-axis wind turbine”. In: (2021).

[104] D Zhao et al. “Offshore wind turbine aerodynamics modelling and measurements”. In: *Wind Turbines Aerodyn Energy Harvest* (2019), pp. 373–400.

[105] Amu N Robertson et al. “OC6 Phase I: Investigating the underprediction of low-frequency hydrodynamic loads and responses of a floating wind turbine”. In: *Journal of Physics: Conference Series*. Vol. 1618. 3. IOP Publishing. 2020, p. 032033.

[106] Lu Wang et al. “OC6 Phase Ib: Validation of the CFD predictions of difference-frequency wave excitation on a FOWT semisubmersible”. In: *Ocean Engineering* 241 (2021), p. 110026.

[107] Lu Wang et al. “OC6 Phase Ia: CFD simulations of the free-decay motion of the DeepCwind semisubmersible”. In: *Energies* 15.1 (2022), p. 389.

[108] Irene Rivera-Arreba et al. “Modeling of a semisubmersible floating offshore wind platform in severe waves”. In: *Journal of offshore mechanics and Arctic engineering* 141.6 (2019).

[109] Peter Wiley. “Low-Frequency Hydrodynamic Modeling of a 12 MW Semi-Submersible Wind Turbine”. MA thesis. NTNU, 2021.

[110] Lu Wang et al. “Investigation of nonlinear difference-frequency wave excitation on a semisubmersible offshore-wind platform with bichromatic-wave CFD simulations”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 84768. American Society of Mechanical Engineers. 2021, V001T01A009.

[111] Yang Zhou et al. “Numerical modelling of dynamic responses of a floating offshore wind turbine subject to focused waves”. In: *Energies* 12.18 (2019), p. 3482.

[112] Simon Burmester et al. “High-fidelity modelling of floating offshore wind turbine platforms”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 84416. American Society of Mechanical Engineers. 2020, V009T09A068.

[113] Thanh Toan Tran and Dong-Hyun Kim. “A CFD study of coupled aerodynamic-hydrodynamic loads on a semisubmersible floating offshore wind turbine”. In: *Wind Energy* 21.1 (2018), pp. 70–85.

[114] V Leble and GN Barakos. “A coupled floating offshore wind turbine analysis with high-fidelity methods”. In: *Energy Procedia* 94 (2016), pp. 523–530.
[115] Antoni Calderer, Xin Guo, Lian Shen, and Fotis Sotiropoulos. “Coupled fluid-structure interaction simulation of floating offshore wind turbines and waves: a large eddy simulation approach”. In: *Journal of Physics: Conference Series*. Vol. 524. 1. IOP Publishing. 2014, p. 012091.
[116] Nianxin Ren, Yugang Li, and Jinping Ou. “Coupled wind-wave time domain analysis of floating offshore wind turbine based on Computational Fluid Dynamics method”. In: *Journal of Renewable and Sustainable Energy* 6.2 (2014), p. 023106.
[117] Yuanchuan Liu et al. “Establishing a fully coupled CFD analysis tool for floating offshore wind turbines”. In: *Renewable Energy* 112 (2017), pp. 280–301.
[118] IEA. *IEA WIND TCP TASK 30 REPORT*. 2020.
[119] Amy Robertson and Lu Wang. “OC6 Phase Ib: Floating Wind Component Experiment for Difference-Frequency Hydrodynamic Load Validation”. In: *Energies* 14.19 (2021), p. 6417.
[120] I Bayati et al. “UNAFLOW project: UNsteady aerodynamics of FLOating wind turbines”. In: *Journal of Physics: Conference Series*. Vol. 1037. 7. IOP Publishing. 2018, p. 072037.
[121] Romain Pinguet. “Hydrodynamics of semi-submersible floater for offshore wind turbines in highly nonlinear waves using Computational Fluid Dynamics (CFD), and validation of overset meshing technique in a numerical wave tank”. PhD thesis. Ecole Centrale Marseille, 2021.
[122] Christian Windt, Josh Davidson, Pál Schmitt, and John V Ringwood. “On the assessment of numerical wave makers in CFD simulations”. In: *Journal of Marine Science and Engineering* 7.2 (2019), p. 47.
[123] Josip Bašić. “Development of numerical model for green water loading by coupling the mesh based flow models with the meshless models”. PhD thesis. University of Zagreb. Faculty of Mechanical Engineering and Naval Architecture, 2019.
[124] Jason M Jonkman. “Dynamics of offshore floating wind turbines—model development and verification”. In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 12.5 (2009), pp. 459–492.
[125] Niels G Jacobsen. “GABC: Exploratory study of free and forced heave decay”. In: (2021).
[126] Nathan Tom et al. “Bichromatic wave selection for validation of the difference-frequency transfer function for the OC6 validation campaign”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 59353. American Society of Mechanical Engineers. 2019, V001T01A022.
[127] Jang Kim, Aldric Baquet, and Hyunchul Jang. “Wave propagation in CFD-based numerical wave tank”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 58769. American Society of Mechanical Engineers. 2019, V001T01A008.
[128] Bjarke Eltard Larsen, David R Fuhrman, and Johan Roenby. “Performance of interFoam on the simulation of progressive waves”. In: *Coastal Engineering Journal* 61.3 (2019), pp. 380–400.
[129] Christian Windt et al. “Validation of a CFD-based numerical wave tank model of the 1/20th scale wavestar wave energy converter”. In: *Fluids* 5.3 (2020), p. 112.
[130] Christian Windt et al. “Validation of a CFD-based numerical wave tank of the Wavestar WEC”. In: *International Conference on Renewable Energies Offshore*. CRC Press. 2018, p. 439.
[131] Benjamin Bouscasse et al. “Qualification Criteria and the Verification of Numerical Waves: Part 2: CFD-Based Numerical Wave Tank”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 85116. American Society of Mechanical Engineers. 2021, V001T01A007.
[132] Gauthier Limpens and Huai Yuan Xu. “Six degrees of freedom modelisation of an object in a fluid with OpenFOAM”. PhD thesis. Ecole Polytechnique de Louvain and Von Karman Institue for Fluid Mechanics, 2015.
[133] Nathan M Newmark. “A method of computation for structural dynamics”. In: *Journal of the engineering mechanics division* 85.3 (1959), pp. 67–94.
[134] Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.
[135] John Crank and Phyllis Nicolson. “A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type”. In: *Mathematical proceedings of the Cambridge philosophical society*. Vol. 43. 1. Cambridge University Press. 1947, pp. 50–67.

[136] Claus D Simonsen, Janne F Otzen, Soizic Joncquez, and Frederick Stern. “EFD and CFD for KCS heaving and pitching in regular head waves”. In: *Journal of Marine Science and Technology* 18.4 (2013), pp. 435–459.

[137] Vuko Vukčević and Hrvoje Jasak. “Seakeeping validation and verification using decomposition model based on embedded free surface method”. In: *Proceedings, Tokyo 2015 Workshop on CFD in Ship Hydrodynamics*. National Maritime Research Institute. 2015, pp. 437–442.

[138] Inno Gatin, Vuko Vukčević, Hrvoje Jasak, and Henrik Rusche. “Enhanced coupling of solid body motion and fluid flow in finite volume framework”. In: *Ocean engineering* 143 (2017), pp. 295–304.

[139] Luis Eça and Martin Hoekstra. “A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies”. In: *Journal of computational physics* 262 (2014), pp. 104–130.

[140] Guilherme Moura Paredes et al. “Experimental investigation of mooring configurations for wave energy converters”. In: *International Journal of Marine Energy* 15 (2016), pp. 56–67.

[141] Lu Wang, Amy Robertson, Jason Jonkman, and Yi-Hsiang Yu. “Uncertainty assessment of CFD investigation of the nonlinear difference-frequency wave loads on a semisubmersible FOWT platform”. In: *Sustainability* 13.1 (2021), p. 64.

[142] A. Goude P. Bachant and M. Wosnik. *Turbinesfoam v0.0.8, zenodo*. URL: https://doi.org/10.5281/zenodo.3542301.

[143] Christian Bak et al. “The DTU 10-MW reference wind turbine”. In: *Danish wind power research 2013*. 2013.

[144] Carlos Dos Santos Pereira Malveiro. “Aerodynamics of a 15-MW and a 5-MW Reference Wind Turbine Using Varying Fidelity Tools”. In: (2022).

[145] Amy Robertson et al. *Definition of the semisubmersible floating system for phase II of OC4*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2014.

[146] Jason Jonkman, Sandy Butterfield, Walter Musial, and George Scott. *Definition of a 5-MW reference wind turbine for offshore system development*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009.

[147] Suraj S Deshpande, Lakshman Anumolu, and Mario F Trujillo. “Evaluating the performance of the two-phase flow solver interFoam”. In: *Computational science & discovery* 5.1 (2012), p. 014016.

[148] Lionel Gamet et al. “Validation of volume-of-fluid OpenFOAM® isoAdvector solvers using single bubble benchmarks”. In: *Computers & Fluids* 213 (2020), p. 104722.

[149] Johan Roenby, Henrik Bredmose, and Hrvoje Jasak. “A computational method for sharp interface advection”. In: *Royal Society open science* 3.11 (2016), p. 160405.

[150] Pierre Blusseau and Minoo H Patel. “Gyroscopic effects on a large vertical axis wind turbine mounted on a floating structure”. In: *Renewable Energy* 46 (2012), pp. 31–42.