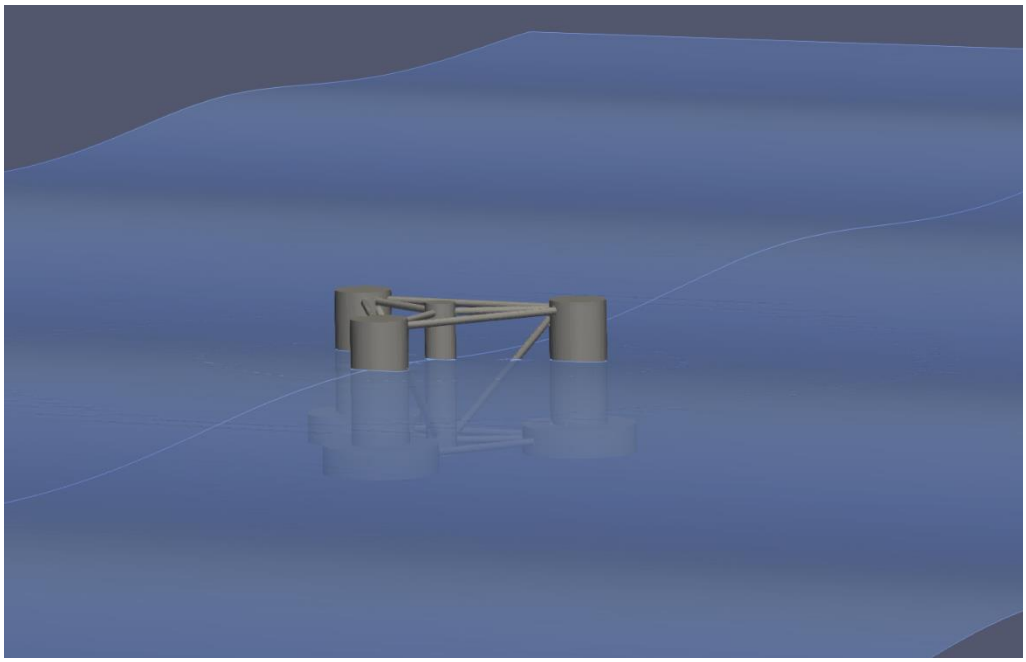
CHALMERS
UNIVERSITY OF TECHNOLOGY



Sensitivity analysis of a moored floating offshore wind turbine using a CFD based surrogate model

Master's thesis in Naval Architecture and Ocean Engineering

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2021

www.chalmers.se

MASTER'S THESIS IN NAVAL ARCHITECTURE AND OCEAN ENGINEERING

Sensitivity analysis of a moored floating offshore wind turbine using a CFD based surrogate model

Miguel Espinilla Soladana

Department of Mechanics and Maritime Sciences

Division of Marine Technology

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2021

Sensitivity analysis of a moored floating offshore wind turbine using a CFD based surrogate model

Miguel Espinilla Soladana

© Miguel Espinilla Soladana, 2021-06-04

Master's Thesis 2021:19

Department of Mechanics and Maritime Sciences

Division of Marine Technology

Chalmers University of Technology

SE-412 96 Göteborg

Sweden

Telephone: + 46 (0)31-772 1000

Cover:

Position of the floating platform for a 3D moored simulation at 19.33 s

Department of Mechanics and Maritime Sciences

Göteborg, Sweden 2021-06-04

Sensitivity analysis of a moored floating offshore wind turbine using a CFD based surrogate model

Master's thesis in Naval Architecture and Ocean Engineering

Miguel Espinilla Soladana

Department of Mechanics and Maritime Sciences
Division of Marine Technology

Chalmers University of Technology

Abstract

With increasing computational resources, the interest in high-fidelity simulations of wave-body interaction has increased. This is the case for the CFD modelling of Floating Offshore Wind Turbines (FOWT), as the industry moves towards commercial farms and the need for optimization increases. Previous computational models based on linear potential flow have shown a disagreement between the experimental and computational motions of the moored floaters subject to wave loading. The aim of this thesis is to help to clarify the influence of the mooring lines in this disagreement by performing a sensitivity analysis in the mooring line stiffness. A 1:50 scale model of a FOWT, defined by the OC5 project, is simulated using a coupled mooring analysis using CFD. The work uses OpenFOAM for the CFD part and Moody is used to model the mooring lines. The model is used to construct surrogate models for the floater motions and tensions in the mooring lines using Polynomial Chaos Expansion (PCE) in UQlab. The motions of the floater, the forces on the mooring lines fairleads and some flow characteristics are reported. It was found that for the wave case simulated the mooring stiffness had a negligible influence in the periodic motions of the floater, although it could affect significantly to the mean components of the mooring forces. The motions of the floater were found to be underpredicted with respect to the experimental data, but in agreement with simulations by other researchers.

Key words: FOWT, Offshore wind, CFD, OpenFOAM, Moody, UQ, gPC, mooring

Preface

In this thesis, a series of CFD simulations were made to build a surrogate model of a floating offshore wind turbine, which was then used to perform sensitivity analysis. The simulations were made between January and May of 2021. Chalmers Centre for Computational Science and Engineering (C3SE) provided access to its resources to perform the simulation in the Vera cluster. The project was carried out at the Division of Maritime Sciences of the Department of Mechanics and Maritime Sciences.

The thesis was done under the supervision of Claes Eskilsson, Senior Researcher at RISE, and Rickard Bensow, Full Professor at Chalmers University. This thesis would not have been possible without the help of both of them. To their unending kindness and understanding, as well as to the support of my family to provide this opportunity in these trying times, I am forever grateful.

Göteborg May 2021-06-04

Miguel Espinilla Soladana

Table of contents

Notations

Bold letters are used to denote vector fields.

Due to the variety of mathematical tools used on this project, some overlaps of notation occur with some variables. Those variables are either used in completely distinguishable contexts or marked with a different subscript.

Acronyms

CFD	Computational fluid dynamics
FOWT	Floating off-shore wind turbine
LCOE	Levelized cost of energy
O&M	Operations and maintenance
TLP	Tension Leg Platform
OC5	Offshore Code Comparison, Collaboration, Continued, with Correlation
RAO	Response Amplitude Operator
C3SE	Chalmers Centre for Computational Science and Engineering
RANS	Reynolds Averaged Navier Stokes
MARIN	Maritime Research Institute of the Netherlands
GUM	Guide to the expression of Uncertainty in Measurement
AIAA	American Institute of Aeronautics and Astronautics
ASME	American Society of Mechanical Engineers

UQ	Uncertainty Quantification
gPC	Generalized Polynomial Chaos
V&V	Verification and validation

Roman lowercase letters

b	Generic known vector
c	Wave velocity
d	Distance between floater and the far end of the simulation domain
e	Internal energy
f	Force, force per unit of volume, function
g	Gravity acceleration
h	Grid size
k	Turbulent kinetic energy, index for polynomial order
n	Number of grids
p	Pressure, convergence order, gPC polynomial order
q	Auxiliar variable for mooring equations
r	Position
t	Time
u	Fluid velocity
w	Grid weights
x	Generic unknown

Roman uppercase letters

A	Diagonal of M, section
C	Stiffness matrix of the floater
D	Damping matrix of the floater
E	Young modulus
F	Security factor
H	Auxiliar matrix that arises during SIMPLE loop
I	Turbulence intensity
L	Characteristic length

M	Matrix of terms arising from a discretization coefficient
N	Number of faces
Q	Generation per unit volume
S	Surface, error function to be minimized
T	Tension on the mooring line
U	Uncertainty
V	Volume
W	Mass matrix of the floater
Z	Random input for a surrogate model

Greek lowercase letters

α	Volume fraction, parameter for error power expansion
γ	Mass per unit length of the mooring line
ϵ	Axial strain of the mooring line, discretization error
ρ	Fluid density, probability function density
τ	Viscous stress tensor
μ	Dynamic viscosity
λ	Wavelength
σ	Security factor
ν	Kinematic viscosity
η	Kolmogorov length scale
ω	Turbulent specific dissipation rate

Greek uppercase letters

Δ	Threshold parameter for error fitting to a power expansion
Γ	Diffusion coefficient
Φ	Generic scalar or vector field, basis function

Figure index

Table index

1 Introduction

1.1 Aim

A CFD model of a moored floater has been developed in this work using the open-source software OpenFOAM v2012 [1]. The free surface flow and the floating platform motion have been simulated in a coupled way, using a 6 degrees of freedom model. A non-linear dynamic mooring model has been included using the software Moody [2]. The semi-submersible floating platform has been defined using the DeepCWind geometry, following the OC5 and subsequent OC6 project guidelines [3]. The cases have been run in the C3SE computer cluster provided by Chalmers University of Technology, Sweden.

The work intended to accomplish three goals:

1. To draw conclusions out of any computational model, the convergence of the results should be assessed. This constituted the first objective of this work. The solution verification study quantified the discretization error and the associated uncertainty of the results, allowing to select a mesh size adequate for our purposes.
2. In previous works, such as [4], an underprediction was identified between the motions of the floater reported in computational and in physical experiments with a test model. Section 1.2 expands upon this issue. The second objective of the project was to identify if this behaviour was still present when the mooring model became nonlinear.
3. Though dynamic mooring has been routinely used in potential flow models, its use in conjunction with higher fidelity CFD modelling is rarer. It increases the complexity of the model introducing dependencies on a higher number of input parameters. These parameters (mooring stiffness, drag and lift coefficients of the mooring lines, position of the mooring fairleads, etc) are often not available with the desired precision due to difficulties in the measurement process, or are subject to significant tolerances [5]. An evaluation of the relative importance of said parameters in the behaviour of the mooring model was deemed interesting both as a way of potentially explaining the discrepancies between physical and computational models and to gain insight into the design drivers of these structures. Therefore, the second goal was to perform a sensitivity analysis on these parameters. Due to time constraints, only one parameter (mooring stiffness) could be evaluated.

1.2 Project Background

The design tools used to simulate floating offshore wind turbines are constantly being improved. There is a great variety of methods that can be used to simulate the hydrodynamics of the floater, as well as the aerodynamics of the tower and blades. To compare between the options available, the International Energy Agency Technology Collaboration Program created the Offshore Code Comparison, Collaboration, Continued, with Correlation (OC5) project. Most of the methods were based in linear

potential flow for hydrodynamics, lumped mass for the motions of the tower and blade element theory for aerodynamics. It was concluded that such methods introduced simplifications to avoid paying too great of a computational price. In this way they lacked sufficient accuracy to model the floater motions in some design cases [4].

In contrast, full CFD RANS models take into account all nonlinearities in the equations, offering additional accuracy but increasing a great deal the computational effort. Previously, it would have been unthinkable to use such models as an engineering tool. However, the advances in computational capacity have made such simulations possible. CFD RANS codes require extensive verification and validation (V&V) to become a useful design tool. A non-exhaustive list include lift and drag on the wind turbine blades, deformation of and loads on the structure, motions of the floater, wave propagation, mooring line forces, sensitivity to different numerical schemes, turbulence models, and the coupling with different tools, such as mooring models, or structural deformation and rigid body motion solvers [6].

The Maritime Research Institute of the Netherlands (MARIN) tested a model FOWT at 1:50 scale in 2011 and 2013, and the results are still being used by many researchers for validation. In successive investigations it was concluded that one of the main causes of the discrepancies between the experimental results and the CFD ones could be the mooring model. To correct this, the use of a non-linear mooring model was suggested [4]. The program Moody, based in the discontinuous Galerkin formulation of the finite element method, have been applied successfully as a nonlinear mooring model for marine applications. The final objective tackled was to perform a sensitivity analysis using a generalized Polynomial Chaos (gPC) surrogate model in order to quantify the impact of deviations and tolerances from the nominal mooring stiffness in that model when applied to the FOWT problem.

1.3 Thesis structure

To achieve the goals of this project the work is structured as follows. It first starts introducing some generalities about the offshore wind field and describing the current efforts for developing reliable floating offshore wind platforms providing examples of some of the topologies that have been recently considered and its associated challenges. After that, the theoretical basis is laid out, explaining the models developed for fluid, motion and mooring simulations and the mathematical methods used to quantify uncertainty and perform sensitivity analysis. Following this, the methodology used to build the model, from the geometry modelling to the design considerations of the CFD simulation are described. The postprocessing steps are commented briefly. Lastly, the results are offered and reviewed. As an afterword, some possible future improvements are discussed.

2 The state of the wind industry

2.1 Introduction to wind energy and some figures

Wind energy turbines use the kinetic energy available by the movement of the wind to make a turbine spin. This rotational energy is transmitted through a shaft to an electrical generator or any other device that can make use of it. Wind turbines face different challenges depending on where they are placed and because of this they are classified in:

- Onshore wind, for turbines that are placed on land.

- Offshore wind, for turbines that are placed on a body of water, thus requiring special supports. If instead of using fixed foundations floating ones are to be used, the resulting turbine may be called a floating offshore wind turbine (FOWT).

Modern use of wind energy to produce electricity started shortly after the invention of the electric generator (1830). Electricity production started in the United Kingdom (1887) and the US (1888), but it was in Denmark where the first modern horizontal axis turbines were developed (1891). More than a century after that, the technology has developed in an exponential fashion. In the last 20 years, the worldwide installed capacity multiplied itself by 75 to reach 564 GW in 2018 [7]. A sustained reduction of costs due to technology improvements and economies of scale as well as the recent irruption of offshore wind ensures the continuity of the present relevance of wind energy.

In 2020 in Europe the total installed wind power reached 220 GW and the wind farms covered 16% of the demand of electricity. New generators accounting for 14.7 GW of power were installed. The COVID crisis reduced the amount of new power installed, but not by much (6% with respect of the previous year). Of the new installed generation, only 20% was planned to be produced by offshore installations. However, this proportion is greater than the current share for offshore of 11.4%. In the next 5 years, a slight increase in the market share of offshore is predicted by WindEurope [8], with 24% of the new installations being offshore. Offshore is gaining traction due to a greater resource availability and a cost reduction that is driving forward the whole wind industry [9]. This current trend is exemplified as seen in Figure 1.

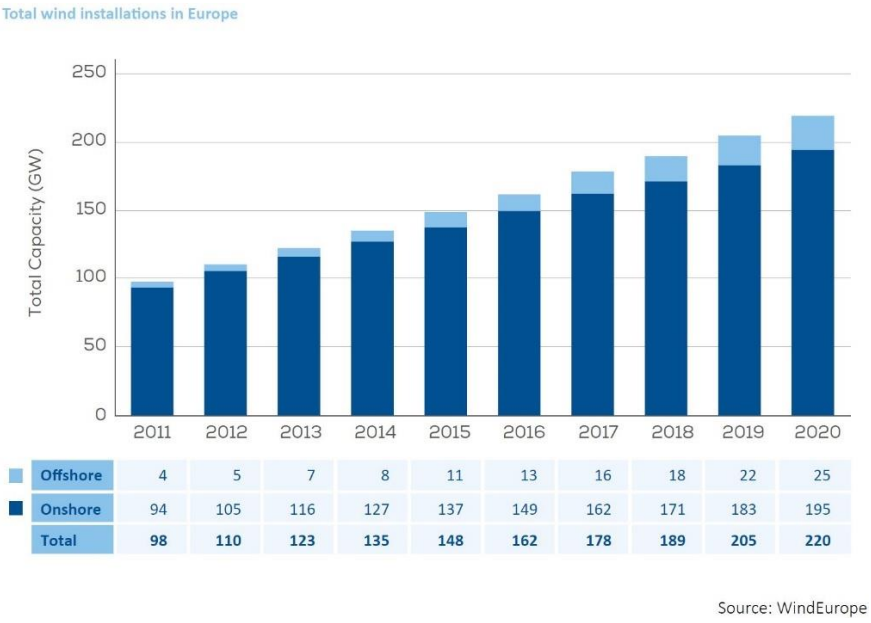


Figure 1: Increasing share of total installed power for offshore wind [8]

2.2 Rationale behind using wind turbines offshore

2.2.1 Causative factors

Offshore wind produced 3% of Europe's electricity demand in 2020. Onshore produced 13%. Wind turbines were initially deployed on land because lower costs could be achieved. Offshore incurs in higher cost as installation, operation and maintenance need to be done at the sea, where the labour cost is very expensive.

There are however several advantages of going offshore [10]:

- **Increased wind resource**: Figure 2 shows the mean power density (W/m^2) available at onshore and offshore locations in Sweden, Norway and Finland. Very few onshore locations can compare with the amount of power that is available at the sea. This is applicable to most land areas in the world. The wind speeds available near the surface are greatly affected by the surface roughness of the terrain. Water provides a comparatively smooth surface and the winds become stronger closer to the surface [11].

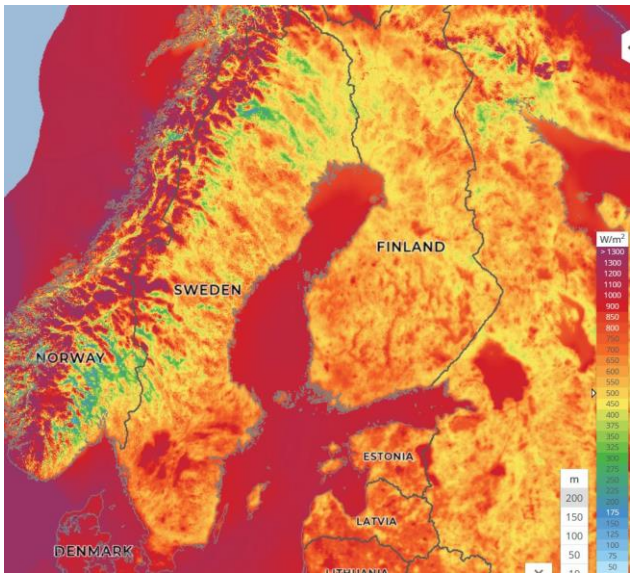


Figure 2: Mean Power Density layer of the Global Wind Atlas over the north of Europe [12]

- **More consistent wind speeds**: wind tends to blow more reliably at the design speeds. This translates into an increment of the capacity factor.

- **Wind production is closer to consumption centres**: Population centres tend to be in coastal areas in most countries with access to the sea. The ability to produce near those population centres means the energy does not need to be transported far away. This could alleviate the grid congestion, lower energy losses, and prevent potential grid overloads and blackouts.

- **Possibility of larger sized projects**: as the potential sites are not restricted to a single location but to a large sea area. This is beneficial because it enables economies of scale.

- **Makes available bigger turbines**: which in turn improve the total O&M cost by reducing the number of turbines needed to archive the same power. Figure 3 shows this. Turbine size is difficult to scale up for onshore. In addition to the structural problems that arise, the size of the blades (even more than 100m), which are normally produced in one piece, is getting so big that it is increasingly difficult to transport them [13]. The possibility of water transport eliminates that disadvantage as the turbine blades can be manufactured along the coast and transported by boat to the assembly site.

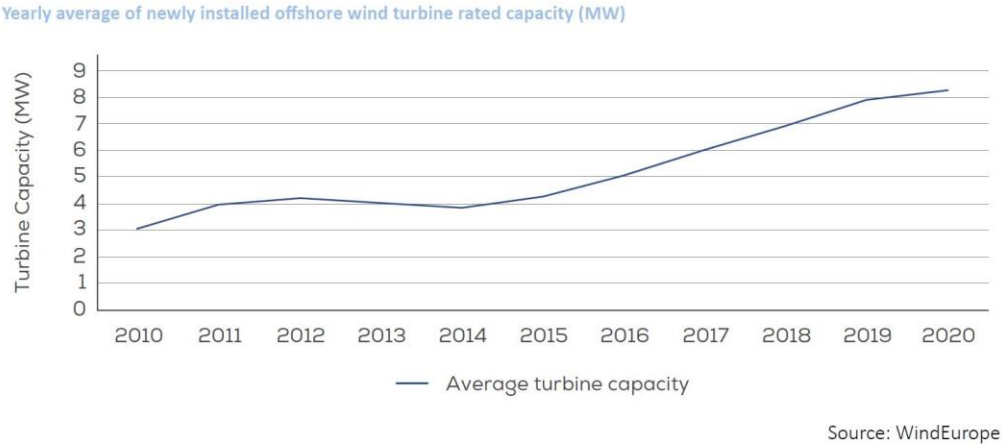


Figure 3: Yearly average of newly installed offshore wind rated capacity [8]

- **Relaxes some design constrains on the turbine**: turbines can rotate faster, have two blades or adopt a downwind configuration [11]. This is in part possible because, in contrast with the platforms used by the oil and gas sector, some safety standards can be relaxed as there is no permanent human presence nearby and operation and maintenance can be performed with calm seas. The noise intensity, a design factor with potentially harmful effects in health [14], can also be increased because of the same reason.

2.2.2 Economical comparison

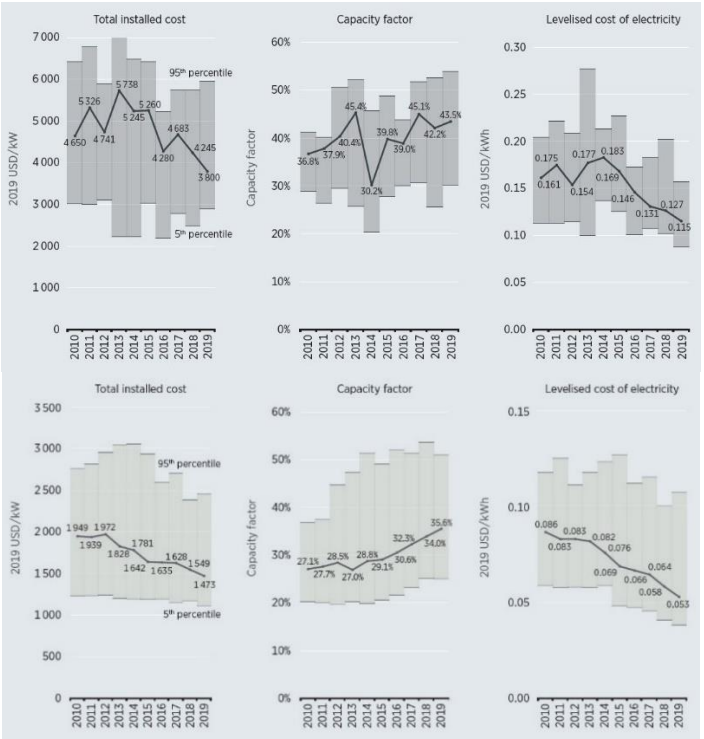


Figure 4: Total installed cost, capacity factor and LCOE for offshore (above) and onshore (below) [9]

Figure 4 offers a comparation between three important metrics to evaluate the cost performance of offshore and onshore energy. The levelized cost of energy (LCOE) is loosely defined as the total amount of money invested during the life cycle of a given project (including initial investment, installation, O&M, decommission, etc) divided by the energy produced during said product life. This metric provides a way of comparing power plants with vastly different cost structures. By inspection, one can see that the LCOE of offshore projects is more than double that of onshore ones.

However, the LCOE is not the only metric widely used to measure technology performance within the electrical market. The electrical grids sustain a very delicate balance of production and consumption which needs to be sustained day and night. The grid operators desire (and pay for) the predictability of the power sources. The capacity factor is defined for a given period as the amount of energy produced divided by the amount of energy that would have been ideally produced if the generator had been operating at rated power [15]. The high capacity factor of fossil fuels is one of the biggest problems that are to be solved to achieve a sustainable grid, and therefore the industry is making efforts for improving the capacity factor of the available technologies and finding new technologies with better capacity factors.

The existence of days without wind diminishes the capacity factor of wind energy. However, there is a great difference between offshore and onshore. For offshore energy the capacity factor is increased, due to more reliable and constant winds on the sea. For 2019, for offshore energy the capacity was of 43.5%, whereas the onshore one was of 35.6%. However, it could be even higher. The Hywind Scotland project [16], the world first floating wind farm, reports a capacity factor of 54%.

2.3 Foundation and mooring technologies

There are three main topologies that can be used for support structures of FOWT. They are spar, tensioned legs platforms (TLP) and semi-submergible concepts. The current projects deployed or in development use either one of these or a mix of several. This section intends to briefly describe these topologies and their associated engineering challenges. An illustration of these concepts can be found in Figure 5.

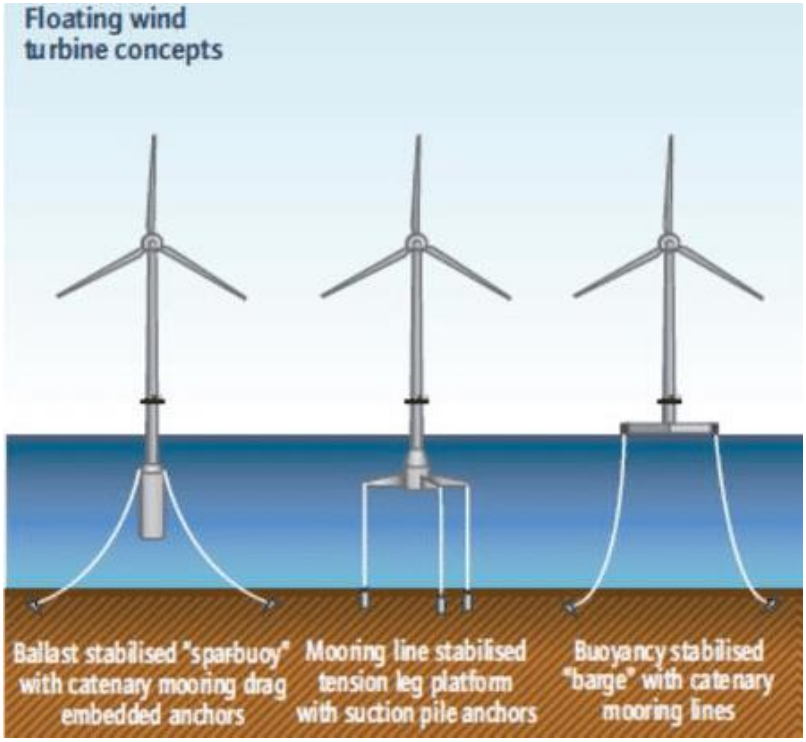


Figure 5: Floating foundation concepts for FOWT [17]

A spar foundation is stabilized by lowering the center of mass of the structure using ballast. It offers small waterplane areas, reducing the forces it experiences. It is efficient in harsh seas and deep waters, where the installation process is easier [18]. Some design challenges are to balance the size of the spar buoy to match the dynamic and static loading limits, to strengthen the turbine to withstand the heel movement produced during operation, and the assembly of the turbine on-site. This last step involves the complex process of raising the tower from the horizontal position in the sea [11]. The spar concept is the one that have been used most in already installed projects, such as the biggest current FOWT farm, Hywind Scotland.

The Tension Leg Platform (TLP) technology relies on the moorings to keep the buoyant structure fixed to the ground. The mooring lines are inextensible, which offers a great amount of stability, but may create problems in high or low tides, or facing extreme waves. A loss of the mooring lines may cause the full collapse of the structure, as TLP is not inherently stable unlike other concepts [11]. There are also installation challenges. The hulls can experience instability when submerged before the attachment of the mooring lines. To solve this, buoyancy aids can be added at a significant cost. New methods are being researched to solve this [18].

The semi-submergible type of foundation consist of a floating jacket which can be made from steel or concrete. It is moored to the floor to prevent excessive drifting. This kind of foundation is subject to higher motions and loads, as most of the structure is on the water surface where waves have a greater effect. Therefore the topology is designed to allow dampening of the structure motions. The heave plates that are installed below the floater in some topologies -including the OC5 floater- are a way of archieving that

damping. Although the fabrication process is complex and expensive, a key advantage comes from this process: the possibility to tow the assembly. For some designs, most of the structural components can be mounted in a dry dock in shallow waters, allowing for a simpler installation afterwards and reducing expensive labour at the sea. This also applies to repairs and O&M, which could be performed close to land.

As can be seen in Figure 6 spar designs are the most installed floating solutions, though semi-submergible foundations follow closely and both are still much less common than fixed foundations.

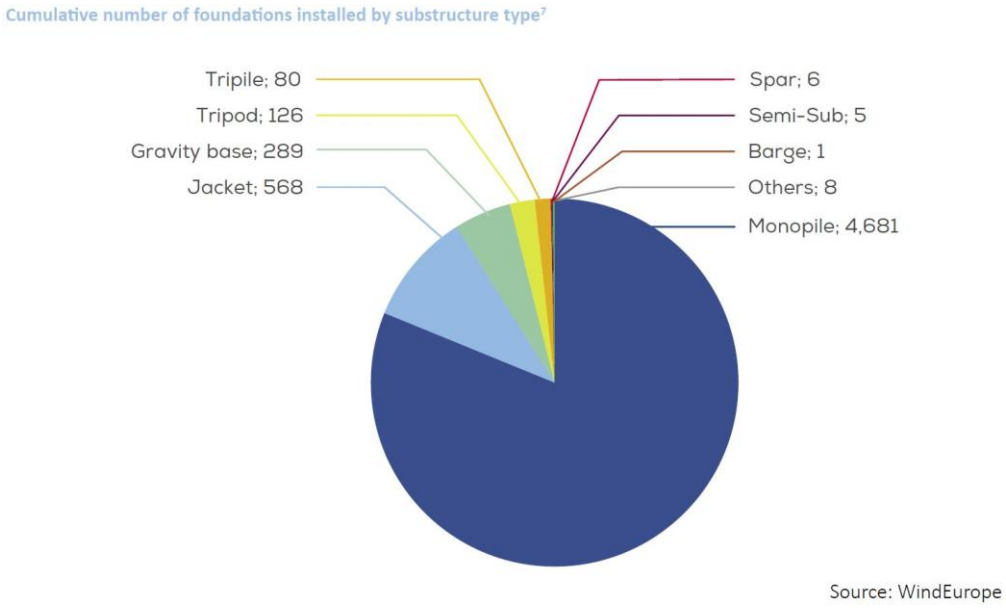


Figure 6: Number of foundations installed by type, including fixed foundations [8]

Mooring is a big part of an efficient design for a FOWT. Figure 7 provides a cost breakdown by components and processes. Although the cost of the mooring lines can account to only 5% of the total cost of the project, including the installation, it has a huge impact in the design of the foundations, which constitute up to 69% of the cost.

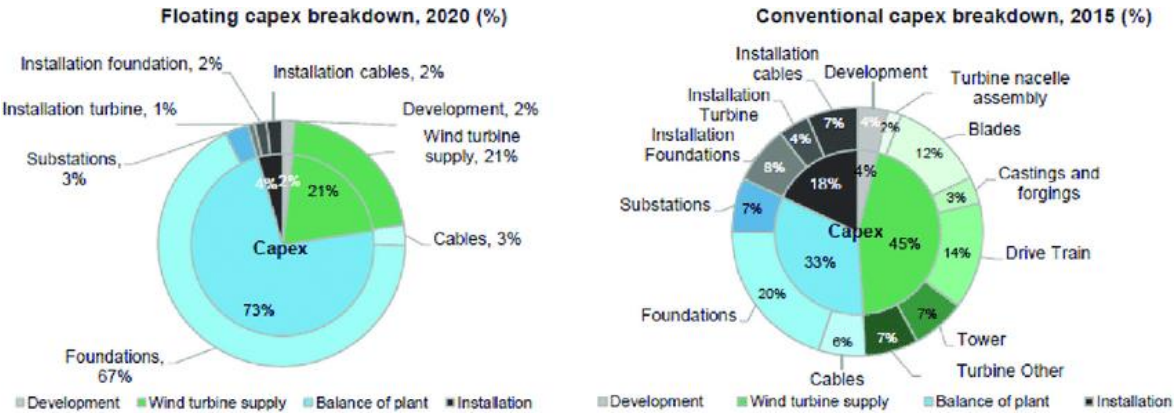


Figure 7: Comparison of costs between floating and fixed offshore wind [19]

Mooring lines affect to the dynamic response of the FOTW. This could be a cause of problems if the natural loading frequencies intersect the natural frequencies of the moored floater [20]. Previous research has identified discrepancies between the experimental movements of the FOWT and the computational experiments attributed to a simplified mooring system. Furthermore, failure rates of mooring lines are surprisingly high. While the industry targets useful lives of 10^4 to 10^5 years, the observed rate of failure of single lines was of only a few dozen years. 60% of the failures were attributed to a deficient design [5]. Further research needs to be done to achieve confidence in the design of mooring systems for FOWT.

The material used for the mooring lines has traditionally been steel wire. The chains of steel have the advantage of an excellent resistance to abrasion with the sea floor. Wire rope has also been used because of its shock absorption resistance. In offshore oil and gas platforms, some polymers have been used at very deep waters, where the weight of steel became too expensive [11]. Some research is being done to investigate the effect of polymers with a smaller Young modulus for shallow water applications, where steel could prove to be too stiff. Such a variety of materials means important fluctuations in the Young modulus for design purposes are to be expected. Even when the material of the mooring lines is known well, differences in manufacturing can contribute to a different stiffness values. In the model experiment of the OC5 floater [21], the uncertainty level of the mooring stiffness reaches about 10% of its nominal value. This same article claims that the mooring stiffness uncertainty is the single most influential parameter in the surge response of the floater. The mooring stiffness is therefore deemed a relevant parameter to perform a gPC sensitivity analysis with a CFD model.

3 Fluid dynamics equations and discretization

3.1 Conservation equations

Within the CFD field, there are many models that could potentially benefit from a certain discretization or solving technique. Many conservation equations from different fields describe the same phenomena differing only on the nomenclature. For example, the equations for the spread of heat in a solid and the change of concentration of a chemical species may be identical from a mathematical perspective. Therefore, it seems like a sensible idea to refer the mathematical derivations to a general equation and then substitute for particular variables later, instead of doing the work for each individual equation.

This general conservation equation represents the balance between four phenomena, and can be written as:

$$\frac{\partial}{\partial t}(\rho\Phi) + \nabla \cdot (\rho \boldsymbol{u}\Phi) = \nabla \cdot \left(\Gamma^{\Phi}\nabla\Phi\right) + Q^{\Phi} \tag{1}$$

Where:

1. ρ is density
2. Φ is the scalar or vector field that is conserved
3. $\boldsymbol{u}$ is velocity
4. Γ^{Φ} is a dissipation rate
5. Q^{Φ} is generation of destruction of Φ

The first term in the equation is the unsteady or transient term. It allows to describe the change of the solution with time. The second term is the convection term. It is associated with coherent, directional movement. The third term is called the diffusion term. It is associated with smoothing movement without a preferent direction. The fourth one is the source term. Is associated with creation or destruction (sources and sinks) of the quantity in the equation.

The integral form of this equation applied to a single cell, using the divergence theorem is:

$$\frac{\partial}{\partial \mathrm{t}}\int(\rho\Phi)\mathrm{dV} + \sum_{1}^{N}\int(\rho \boldsymbol{u}\Phi)dS = \sum_{1}^{N}\int\left(\Gamma^{\Phi}\nabla\Phi\right)dS + \int Q^{\Phi}\mathrm{dV} \tag{2}$$

Where N is the number of surfaces of that cell. This equation still does not include any assumption, so the solution field should be exact. To obtain an algebraic system of equations from this conservation equation, however, assumptions need to be made. For the FVM (Finite Volume Method) a key consideration is that the information stored in the cell centroids must be interpolated to the surfaces to be able to evaluate the integrals. The numerical evaluation of said integrals also create discretization error. Each term

should be discretized to reach a desired order of accuracy and some terms require special treatment to ensure numerical stability and physical results. For this problem the terms were discretized up to second order.

The Navier Stokes equations, as well as other equations used to model the fluid interface and for turbulent behaviour, can be described with this framework by choosing adequate Φ, Γ^{Φ} and Q^{Φ}.

3.2 The Navier Stokes equations

The Navier Stokes equations are the most important equations of fluid dynamics. They describe the pressure, velocity, and internal energy of a fluid as it develops trough a certain spatial domain during a certain time interval. They are partial differential equations, meaning the velocity, pressure, density, and energy rates of change depend on time and each spatial direction, adding mathematical complexity. The convective term in the momentum equation makes the equations nonlinear. The equations are written very generally and compactly as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{3}$$

$$\rho \frac{D\boldsymbol{v}}{Dt} = -\nabla \mathrm{p} + \nabla \cdot \bar{\bar{\tau}}' + \rho \, \bar{\boldsymbol{f}}_{\boldsymbol{m}} \tag{4}$$

$$\rho \frac{De}{Dt} = -\mathrm{p} \nabla \cdot \boldsymbol{u} + \bar{\bar{\tau}}' : \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{q} + Q_r + Q_c \tag{5}$$

Where:

1. u is the velocity vector field
2. p is the pressure scalar field
3. ρ is the density scalar field
4. e is the internal energy scalar field
5. $\bar{\bar{\tau}}'$ is the viscous stress tensor
6. $\bar{f_m}$ are the mass forces (such as gravity) per unit of volume
7. Q_r y Q_q are heat terms produced by radiation and chemical reaction respectively
8. The $\frac{D}{Dt}$ operator represents the material derivative:

$$\frac{D\Phi}{Dt} = \frac{d\Phi}{dt} + u \cdot \nabla\Phi$$

9. The ":" operator is the internal tensor product
10. The $\frac{D}{Dt}$ operator represents the material derivative:

$$\frac{D\Phi}{Dt} = \frac{d\Phi}{dt} + \boldsymbol{u} \cdot \nabla\Phi \tag{6}$$

11. The ":" operator is the internal tensor product

The first equation is the continuity equation. It is always needed, as it describes conservation of mass, which is always fulfilled. The second equation is the conservation of linear momentum. It is derived from Newton's Second Law of motion. Continuity and momentum conservation equations can describe incompressible fluid. The third equation is an energy equation. It can be formulated in terms of temperature and can be used to describe heat transfer problems. These equations are often completed with the equations of state of the fluid to describe compressible flow problems, giving a complete description of almost every fluid problem relevant for industry.

For a three-dimensional case, there are very few known analytical solutions to the Navier Stokes equations, and almost all of them are related to a very simple domain. For complex geometry or more involved fluid behaviour, the usage of numerical methods is always required.

For simulations involving water, such as the model of the FOWT, the fluid can be considered incompressible (ρ is constant) and developing the viscous stress tensor, substituting the mass forces by gravity, and including the turbulent quantities k and μ_t the equations become:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{7}$$

$$\frac{D\boldsymbol{u}}{Dt} = -\nabla\left(\mathrm{p} + \frac{2}{3}\rho\mathrm{k}\right) + \nabla \cdot [(\mu + \mu_t)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)] + \rho\boldsymbol{g} \tag{8}$$

Where the energy equation is dropped as we assume isothermal flow. The Finite Volume Method does not use the equations in this form, but rather requires an integral form. A problem that arises in the numerical treatment of these equations is the lack of the variable p in the first equation. To circumvent this, the equations are usually solved iteratively resulting in the SIMPLE (Semi-Implicit Method for Pressure Linked Equations), PISO (Pressure-Implicit with Splitting of Operators) or PIMPLE (Pressure-Implicit Method for Pressure Linked Equations) loops.

3.3 Turbulence modelling and boundary layers

Turbulence is a phenomenon that happens as the small instabilities in the flow get amplificated by the non-linear inertial terms. This causes chaotic, three-dimensional motion that amplifies mixing and diffusion. Turbulence is usually described as a sum of eddies of different scales. Energy is transferred between the biggest eddies to the smallest ones in a "cascade" process, until the eddies are so small that molecular viscosity dissipate their energy as heat. The scale of the smallest eddies that can exist is called the Kolmogorov scale:

$$\eta = \left(\frac{\nu^3}{\mathrm{g}}\right)^{\frac{1}{4}} \tag{9}$$

$$t_n = \left(\frac{\nu}{g}\right)^{\frac{1}{2}} \tag{10}$$

These scales are extremely small for most engineering applications. As the element size has to be smaller than the Kolmogorov scale to be able to resolve the eddies, the direct approach, called Direct Navier-Stokes (DNS) is barely used as it would result in extremely big meshes.

Most of the turbulence models used in industry work by averaging turbulent motion. Large Eddy Simulation (LES) only simulates the largest eddies, that carry more energy, while the smaller ones are approximated using a sub-grid model. This modelling strategy is quite accurate but proves to be too taxing for many applications.

The Reynolds Averaged Navier Stokes family of models are an alternative with a lower computational cost. These methods separate the variables into a mean value and a fluctuating component and average the equations in time. The resulting averaged equations are very similar to the original ones except for a tensor that adds six unknowns that shall be modelled, arising from the non-linear term.

Most models assume that the components on this tensor are a linear function of the velocity gradients (Boussinesq approximation). In that way, the problem for incompressible flows is often simplified to compute a turbulent viscosity, μ_t. This can be done using several methods. The two equation methods are popular in engineering modelling because they offer a trade-off between accuracy, computational cost, and stability. The $k-\epsilon$, and $k-\omega$ families of models are the most used. [22]

For our application, the model used was $k-\omega$ SST. It requires to solve a conservation equation for the turbulent specific dissipation rate, ω, and for the turbulent kinetic energy, k. It offered an advantage when modelling separated flow and adverse pressure

gradients in comparison with the baseline $k-\omega$ method. The equations for the $k-\omega$ SST model can be found in [23].

3.4 Free surface modelling

There are several approaches to model multiphase flows. For continuous-continuous phase interaction, and immiscible phases, the most used one is the Volume of Fluid method. It is based in considering each cell of the mesh as a homogeneous mixture of both fluids with a volume fraction α, ranging from 0 to 1. A value of 0 would mean that the cell only contains one of the fluids, and a value of one only the other one. The properties of the cell are linearly interpolated based in α [4]. This approach allows to solve the conservation equations only one time for both fluids. The drawback is that an additional conservation equation is added:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \boldsymbol{u}) = 0 \tag{11}$$

The discretization of the divergence term in this equation uses special schemes to maintain a sharp interface between the fluids. If a normal upwind scheme were to be used, there would be diffusion of α from the interface to the rest of the fluid, which would be unphysical. The Flux Corrected Transport (FCT) is a framework in which some of the most used schemes work. Often, high order schemes can create unwanted oscillations in the result. To avoid this, the FCT theory proposes to blend high and low order schemes, extending the stability of the method at high Courant numbers [24]. In OpenFOAM, such implementation is coded through the MULES solver. Common schemes used in this method are vanLeer, SuperBee, HRIC and CICSAM [25].

3.5 Discretization process

The objective of a discretization process is to achieve a set of equations that can be solved with a computer. These equations, in general, are formulated in matrix form such as:

$$M\boldsymbol{x} = \boldsymbol{b} \tag{12}$$

To accomplish this, there are three steps that need to be followed:

- **Physical modelling**: The first decision that needs to be made is what is going to be included in the simulation. Specifically speaking, how are the boundaries of the domain going to look like and what are the physical phenomena that are going to be represented. In this simulation, for example, we are interested in the wave effect on the floater. If the domain boundaries were too close, the reflections would mix with the incoming waves and impede the retrieval of clean data. In this way there is a lower bound on the size of the domain. This kind of decisions are often highly uncertain, and should be made in accordance with previous research and industry experiences.

- **Domain discretization**: Once the geometry of the simulation is known, the domain should be discretized. This is done by dividing it into small partitions called cells, in a process called meshing. Figure 8 illustrates how a simple 2D mesh looks like. In general, a cell is composed of vertex, faces and a centroid. A cell can have any number of faces. There are two approaches to FVM meshes: cell-centred and node (or vertex) centred. The most used one, and the one that was used used, is the cell centred approach.

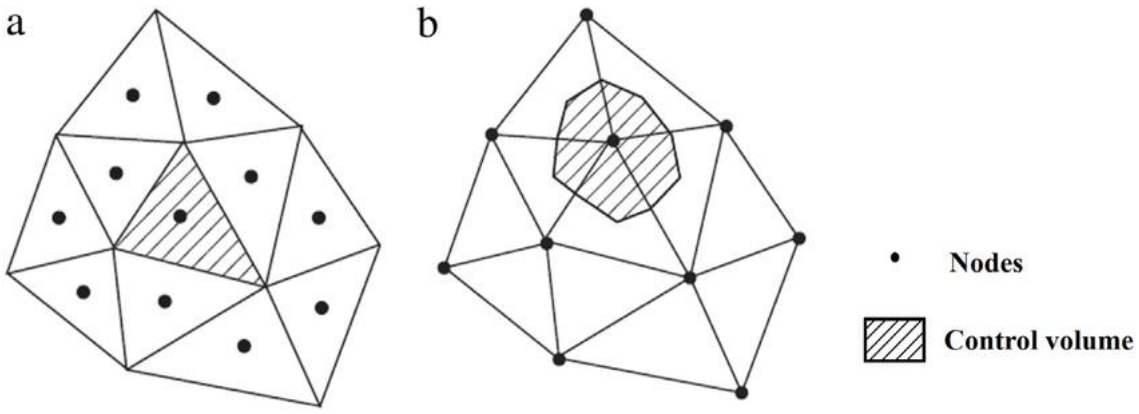


Figure 8: Cell centred mesh (a) and node centred mesh (b) approaches [27]

- **Equation discretization**: There are several methods to discretize the governing equations, which can be transformed to more convenient form. The Finite Difference Method substitutes the derivatives for an algebraic expression directly on the original equations. The Finite Element Method derives a weak or variational form of the equations. The Finite Volume Method works with the integral form of the equations, which then are applied (introducing approximations) to each single element. Most computer software for CFD analysis use the FVM method, as it inherently preserves fluxes between an element and its neighbours [22].

3.6 SIMPLE, PISO and PIMPLE loops

The incompressible Navier-Stokes equations are a challenge to solve numerically because they do not include an equation for pressure. The continuity equation can be seen more as a restriction for the momentum equation. To circumvent this problem, some manipulation is done to achieve an equation for pressure from the continuity equation, and then the equations are solved iteratively until a combination of pressure and velocity fields that satisfy the equations is found. A very simplified introduction to the process is exposed here. Starting from a semi-discretized version of the original equations [28]:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{13}$$

$$M\boldsymbol{u} = \nabla p \tag{14}$$

M is a matrix of coefficients known and constant at each time step that arise from the discretization of the momentum equation. u is the vector that contains each cell velocity. ∇p is the gradient of the pressure of each field. A H vector is defined also as:

$$H = A\boldsymbol{u} - M\boldsymbol{u} \tag{15}$$

Where A is the diagonal matrix of M, so that H is only a function of u. This decomposition is done because A is trivial to invert, which will help in later steps. With these variables, manipulating the momentum equation and substituting in the continuity equation yields an equation for pressure:

$$\nabla \cdot (A^{-1}\nabla p) = \nabla \cdot (A^{-1}H) \tag{16}$$

Once the pressure is known, the velocity field can be retrieved explicitly as:

$$\boldsymbol{u} = A^{-1}H - A^{-1}\nabla p \tag{17}$$

Though some methods solve the equation for pressure and velocity simultaneously, most solvers are iterative to reduce the computational expense. The SIMPLE loop iterates through these four equations until convergence is reached. First the momentum equation is implicitly solved for u. Then H is computed, and the pressure equation is solved. The velocity field is retrieved with the last equation. This is called an outer loop. Although the last one might not seem needed, it should be noticed that in this simplified explanation no distinction has been made between fluxes through the cell faces and

values at the cell centroids. A more rigorous mathematical description of the process can be found in [29]. As it solves the momentum equation at each time step, the SIMPLE algorithm is very stable.

The PISO loop starts as the SIMPLE by solving the momentum equation, computing H and ∇p, and solving for u. However, instead solving again the momentum equation, several iterations of the three last equations are made (inner or corrector loops). This reduces the computational load, but it is less stable. The PIMPLE loop, a combination of SIMPLE and PISO, uses a mix of inner and outer loops to improve stability while keeping computational speed. Figure 9 shows a flow diagram of the methods.

When simulating a transient problem, the transient term is dominant in the equations. This has the effect of making the simulation very stable. Hence, a fast method like the PISO or PIMPLE is often used. The SIMPLE loop excels at stationary cases where stability is critical, and often underrelaxation is needed to compute the loop in a stable way.

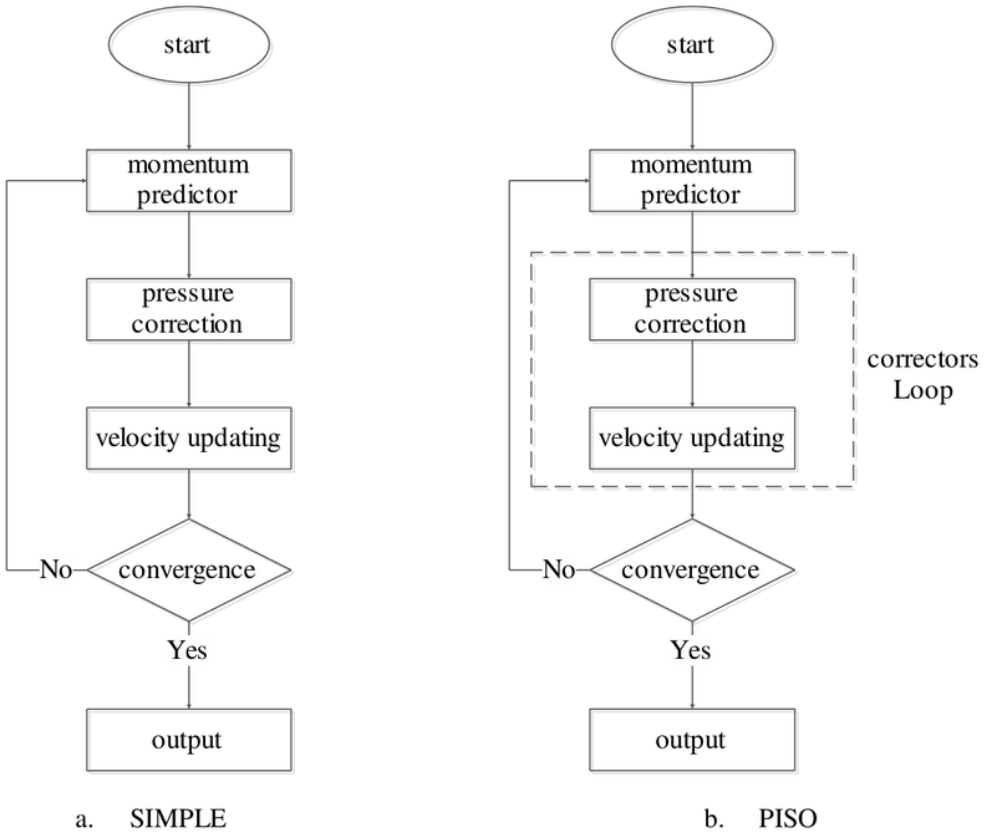


Figure 9: SIMPLE and PISO flow diagrams [30]

4 Rigid body motion and mooring equations

4.1 Equations for rigid body motion

The equations of motion are formulated as [6]:

$$W\ddot{\boldsymbol{r}} = \boldsymbol{f}_H - C\boldsymbol{r} - D\dot{\boldsymbol{r}} \tag{18}$$

W is the mass matrix, and C and D are the stiffness and the damping matrix respectively. For our problem this D matrix is zero, as the body is assumed to be completely rigid. The hydrodynamic forces $\boldsymbol{f}_H$ depend on the velocity of the floater $\dot{\boldsymbol{r}}$ because of the Navier Stokes equations, as:

$$F = \int\int_S (p\boldsymbol{n} + \boldsymbol{\tau})dS + \boldsymbol{F_M}$$

$$M = \int\int_S \left(\boldsymbol{r_{CS}} x (p\boldsymbol{n} + \tau)\right)dS + \boldsymbol{r_{CM}} x \boldsymbol{F_M}$$

Where:

1. p is the pressure
2. $\boldsymbol{n}$ is the normal vector
3. $\boldsymbol{F_M}$ is the mooring force
4. $\boldsymbol{r_{CS}}$ is the distance from the centre of mass to the element surface
5. $\boldsymbol{r_{CM}}$ is the distance from the centre of mass to the mooring attachment point

In this way the equations are non-linear, and the fluid and rigid body problem are coupled.

OpenFOAM can solve the rigid body motion equations through several methods: Crank-Nicholson, Newmark, symplectic integrator, or even custom made solvers. The weak stability of the coupling can be improved via acceleration relaxation (a direct multiplier on the acceleration) or acceleration damping (where acceleration is reduced more the higher its value is) [31]. The movement of the floater, which is a boundary condition of the model, implies that the adoption of a moving mesh formulation must be implemented.

4.2 Mooring equations

For the mooring, the differential equations relate the second time derivative of the position of the mooring $\boldsymbol{r}$, with the tension $\boldsymbol{T}$ and the external forces $\boldsymbol{f}$ [32]:

$$\frac{\partial^2 \boldsymbol{r}}{\partial t^2} = \frac{1}{\gamma_0}\frac{\partial \boldsymbol{T}}{\partial s} + \frac{\boldsymbol{f}}{\gamma_0} \tag{19}$$

$$\boldsymbol{T} = EA_0 \frac{\boldsymbol{q}}{1+\epsilon} \tag{20}$$

$$\boldsymbol{q} = \frac{\partial \boldsymbol{r}}{\partial s} \tag{21}$$

$$\epsilon = |\boldsymbol{q}| + 1 \tag{22}$$

Where ϵ is the axial strain, γ_0 is the mass per unit of length and $\boldsymbol{q}$ is an extra variable introduced to reduce the system to a first order system of equations. The external forces can be divided in 4 components [5]:

- Added mass
- Drag
- Buoyancy
- Contact force

The first two arise because of the relative velocities and accelerations between the fluid (assumed to be quiescent) and the mooring cable and are computed using the Morrison equations. The buoyancy force is derived straightforwardly from the Archimedes principle. The contact force is modelled as spring-damper system in the normal direction and a Coulomb friction model in the tangential direction.

The software used to solve the mooring model is Moody [2]. This software allowed to solve dynamics of cables, allowing the modelling of no bending stiffness chains, like the ones used for FOWT.. The solver implements a Discontinuous Galerkin method for spatial discretization. A few elements of high order are used to discretize the mooring line, taking advantage of the exponential convergence of high order methods for smooth solutions. Discretization in time is done via a third order Runge-Kutta scheme. An introduction to the mathematical formulation, as well as verification with several test cases relevant for offshore applications, can be found in [32].

The coupling algorithm specifies in OpenFOAM the positions of the fairleads, which are then passed as an input for Moody. In turn, Moody will return the mooring forces as an output for the rigid body solver. The time step of the fluid solver is in the range of $10^{-3}\ s$ to $10^{-4}\ s$, whereas Moody operates in the range of $10^{-5}\ s$. While the rigid body solver is still in the previous step, there is a need for interpolation of the variables that Moody need as an input. A lagged quadratic interpolation is used to solve this. Of practical interest is that this algorithm damps high frequency motions and therefore needs a low enough time step in the fluid and rigid body motion solver to properly converge

5 Uncertainty estimation method

For each experiment that is performed, be it with a physical system or a computer simulation, differences with respect to the real system conditions will arise. For a physical experiment, the precision and accuracy of the measuring instruments is limited, some variables could be uncontrolled, or some process could be not well understood or random in some way. These factors cause uncertainty in the measure, and the measure is not complete without the uncertainty estimation.

Providing an uncertainty estimation is useful for several tasks. It has importance in design, estimating how useful the data is to a certain design purpose. If the uncertainty is high, the data may only be relevant for qualitative information. Lower levels of uncertainty mean simulations can be used to study changes in the design once there is test data available, and a very low uncertainty is associated with a direct use of the magnitudes simulated for. Uncertainty estimation can also be a legal requirement to comply with a standard.

For physical experiments there are well established procedures to provide an estimation of the uncertainty of a measure. The Guide to the Expression of Uncertainty in Measurement (GUM), is maybe the most important of such procedures. However, no such thing exists for numerical simulations. Several procedures have been proposed by several authors and institutions [33] [34] [35], and some guidelines published by ASME and AIAA [36] [37], but no consensus have been reached on how to fully account for the uncertainty of the measure [38].

Within the CFD field, two complementary ways of addressing uncertainty can be found:

- **Verification** is the process used to determine whether the programming and computational implementation is correct or not. Code verification is performed by the programmers before releasing the CFD for public use. Solution verification is performed by the end users, and it aims to estimate the error/uncertainty for a computational solution even if the exact solution is unknown. The approach described in this section is concerned with the solution verification.

- **Validation**, on the other hand, assesses whether the simulation agrees with the performance of the real system. It addresses modelling errors. A validation assessment cannot be done for the CFD code itself, but rather to each specific kind of simulation performed with it.

It has been said that verification is about solving the equations right, whereas validation looks into solving the right equations. There are subtypes of numerical error generally associated with verification [6], although several classifications exist:

- **Computer round-off error**: The round-off error appears because of the storage of the numbers used in the simulation in a fixed-point way, with a fixed decimal length. This error is usually very small in comparison with other sources, as most computations are done using double precision arithmetic.

- **Iterative convergence error**: The Navier-Stokes equations are non-linear equations. To solve this kind of problems numerically an iterative procedure is usually required. Said procedure must have a stopping point, and the iterative error appears to account for the variation between the stopping point and the true solution. This error can be computed and made arbitrarily small. To make it negligible for the uncertainty estimation, it has been suggested to make it at least two or three orders of magnitude smaller than the discretization error [35].

- **Discretization error**: This is the error committed when transforming the continuous governing equations to an algebraic system that provides a discontinuous solution in the points defined by the mesh. This applies to the temporal discretization and to the spatial discretization. The sources of the discretization error are truncation error, dispersive error, and dissipative error. The parameter that controls this kind of error is the element size and the time step size. If the numerical scheme is consistent, when the element and time step size become infinitely small, the discretization error will tend to zero. This kind of error is dominant for practical CFD computations, and it is complex to quantify. Meshing is a complex problem. Its design shall consider, to cite some parameters, resolution, density, aspect ratio, stretching, orthogonality, grid singularities, and zonal boundary interfaces [38]. It has been said that meshing is as much of a science as of an art. Because of this, it becomes difficult to establish a concrete relationship between the quality of the mesh and the quality of the result prior to the simulation. Most verification techniques focus on this kind of error.

- **Modelling error** is associated with validation. It happens when modelling a poorly understood phenomena, when making the problem well posed requires making modifications to the domain or equations, or when making conscious assumptions for the sake of modelling ease or calculation speed. Even if the base equations solved for are well understood, usually there are still models based on empirical experience. For CFD codes these occur in turbulence modelling and when modelling some boundary conditions.

- In addition, one can talk about **unacknowledged error** (meaning there cannot be a systematic procedure to detect it) such as programming error or usage error.

For this simulation, we address verification by checking the discretization error. The round-off error is assumed to be negligible, and the iterative convergence error would be checked to make sure the residuals were at least two orders of magnitude below the discretization error [35].

The procedure used is the one described in Eça and Hoekstra [35]. In their paper they propose a way of estimating the discretization error for data in coarser grids, that may not present the mathematical characteristics to be estimated in the way previous works had focused on. To estimate the discretization error, several grids of increasing

resolution are needed. The error is approximated with a truncated power series expansion of the form:

$$\epsilon_\Phi = \Phi_i - \Phi_0 = \alpha h_i^p \tag{23}$$

where:

- Φ_i is for the mesh number i the variable we want to estimate the error for. This is known in advance.
- Φ_0 is our estimation for the true solution, unknown.
- α is a parameter we need to fit, unknown.
- h_i is the representative cell size for the i:th mesh, known in advance.
- p is a parameter we need to fit related to the order of convergence. It is unknown and would ideally be 2, as our OpenFOAM implementation offers second order precision.

The idea is to solve a system of equations for α, p and Φ_0 to get our ϵ_Φ estimate for each grid. As we have three variables to solve for, we would need three grids for that operation. However, it is advised in the paper to use at least four grids. The reason is that p is very sensible to small variations in the data (scatter) which often occur for engineering applications. Adding more grids adds to the reliability of the estimation and allows to check how good the fit is. As a counterpart, the more grids are used more expensive will be the simulations. If the fit is very bad (for example if the grids are too coarse), the uncertainty is higher, and therefore it will be computed using a greater security factor.

To solve the equation for more than three grids a least squares estimation is suggested:

$$S = \sqrt{\Sigma_{i=1}^{n_g} w_i \left(\Phi_i - \left(\Phi_0 + \alpha h_i^p\right)\right)^2} \tag{24}$$

Where α, p and Φ_0 are obtained by minimizing the function S, n_g is the number of grids used in the simulation and w_i are weights suitably selected to give more importance to the values of the finest grids.

Once the fit has been made, if p is too small (less than 0.5) or too big (greater than 2), the procedure is repeated using other estimators in order to avoid over or under conservative results, respectively. After obtaining the error values, the uncertainty for each grid $U_\Phi(\Phi_i)$ is estimated. This estimation will be different if the data fit has been bad, which is to say if the standard deviation of the fit σ is greater than a threshold value Δ_Φ. This value depends on the data scatter and number of grids:

$$\Delta_{\Phi} = \frac{\Phi_{i,max} - \Phi_{i,min}}{n_g} \tag{25}$$

Then if $\sigma < \Delta_{\Phi}$:

$$U_{\Phi}(\Phi_i) = F_s \epsilon_{\Phi}(\Phi_i) + \sigma + \left|\Phi_i - \Phi_{fit}\right| \tag{26}$$

with:

$$F_s = \begin{cases} 1.25 & 0.5 < p < 2.1 \\ 3 & otherwise \end{cases} \tag{27}$$

Otherwise. For $\sigma > \Delta_{\Phi}$:

$$U_{\Phi}(\Phi_i) = \frac{3\sigma}{\Delta_{\Phi}} \left(\epsilon_{\Phi}(\Phi_i) + \sigma + \left|\Phi_i - \Phi_{fit}\right|\right) \tag{28}$$

6 Sensitivity analysis and generalized polynomial chaos

The purpose of a sensitivity analysis to see how the output reacts for variation of the input. In more rigorous terms, the input is considered as a random variable with an associated probability distribution and our objective is to see what the probability distribution of the output would be. The obvious disadvantage is that our computational model does not allow to run for a random variable, only discrete values it could take. This forces us to choose a non-intrusive uncertainty quantification (UQ) method.

In many computational applications, there is often little reason to expect that the statistical distributions that will arise after running the model will have a canonical form. In addition, in most cases the relationship between the output and the input cannot be written in closed form. This rules out most analytical uncertainty propagation methods. For this cases, Monte Carlo (MC) simulations offer a way to perform uncertainty propagation numerically. The procedure is to take a random sample of inputs that follow the uncertainty distribution of that input variable and run each value through the model. The corresponding outputs approximate the real output distribution if the sample size is big enough. However, MC simulations converge slowly, and often require hundreds of thousands of iterations before reaching a stable distribution. This is impossible to do for a CFD simulation of this size.

Generalized Polynomial Chaos aims to accelerate MC simulations by only choosing a handful of points to recreate this output uncertainty distribution. In this method we create a surrogate model as a sum of polynomials that approximate the original model as a function of our random inputs. If there are few inputs, the model coefficients are adjusted using a few computational experiments. Once this surrogate model is created, it is straightforward to evaluate it for arbitrarily large sets of our input variable without having to run the original CFD model.

The mathematical foundations of the theory are intricate and are rooted in functional analysis. Here only an overview will be presented, following the introduction done in [5]. For clarity, the multivariate case is not discussed. A broader take with further references to the literature can be found in [39].

A model with random and deterministic variables can be described as a function $f(x, Z)$ where x refers to the deterministic and Z to the random variables. The idea of the polynomial chaos expansion is to decompose this function like:

$$\mathrm{f}(x, Z) \approx \sum_{k=0}^{p} \widehat{f_k}(x)\Phi_k(Z) \tag{29}$$

Where $\widehat{f_k}(x)$ are coefficients for the model and $\Phi_k(Z)$ polynomial basis functions whose form should be selected based on the probability distribution of the random inputs Z. The parameter p is the polynomial order of the simulation. To make a perfect fit, p should be infinite, but the coefficients decay quickly and the model can be trimmed to a finite number without much loss of precision.

The problem then becomes how to calculate $\widehat{f_k}(x)$. There are two main non-intrusive approaches to the computation of the model coefficient. They can be computed via a projection method or via a regression method. The projection method computes the coefficients via an inner product of $f(x,z)$ and $\Phi_k(z)$ with respect to the probability density function of the input variable $\rho(z)$:

$$\widehat{f_k}(x) = \frac{\int f(x,z)\Phi_k(z)\rho(z)dz}{\int \Phi_k^2(z)\rho(z)dz} \tag{30}$$

Where the integrals can be computed using a quadrature method, which also determines the input values z_i the CFD model will need to be evaluated for. However, for a high number of random input parameters (greater than 4-5), this approach requires a high number of model evaluations. This is known as the curse of dimensionality. In this case a regression method can be used.

In a regression method, the input values are first sampled randomly, and the outputs are computed. Then a regression fits the values of the model coefficients in a way that minimizes the error to those outputs. There are several regression techniques that can achieve this. UQlab [40], the program that was used to perform this calculation, includes an ordinary least-squares method as well as sparce regression methods. Even in this case, increasing further the number of random inputs leads quickly to need a large number of model evaluations, making the whole approach impractical for models with a large number of random inputs.

Once the coefficients are set, an arbitrarily large sample can be used as an input to the surrogate model following the uncertainty distribution of the input, like in a normal MC simulation. The uncertainty distribution at the output can be reconstructed from the resulting points.

The sensitivity analysis was done using the software UQlab. UQlab is a framework of tools to perform uncertainty quantification (UQ). It is based in Matlab and it can be used for free for academic purposes. Among its tools there is a suite for performing Polynomial Chaos Expansion (PCE). A detailed description, as well as manuals for the program, can be found in [40].

7 Methodology

7.1 Problem statement

To build the surrogate model and to verify the results two kind of simulations were performed:

- A series of seven 2D simulations with increasingly refined meshes ranging from 125.000 elements to 500.000 elements. These did not include the floater and were only intended to provide verification for the wave heights of the simulation.

- A series of seven 3D simulations with a mesh of 2.000.000 elements. These included the floater and a mooring simulation. They were required to build the surrogate model.

The simulations all included a water-air interface. The inlet generated the waves, which then propagated and eventually clashed with the walls of the floater. A $k-\omega$ SST turbulence model was included. A moving portion of the mesh took care of the coupling between the floater movement and the wave interaction. The domain and the coordinate system are outlined in Figure 10.

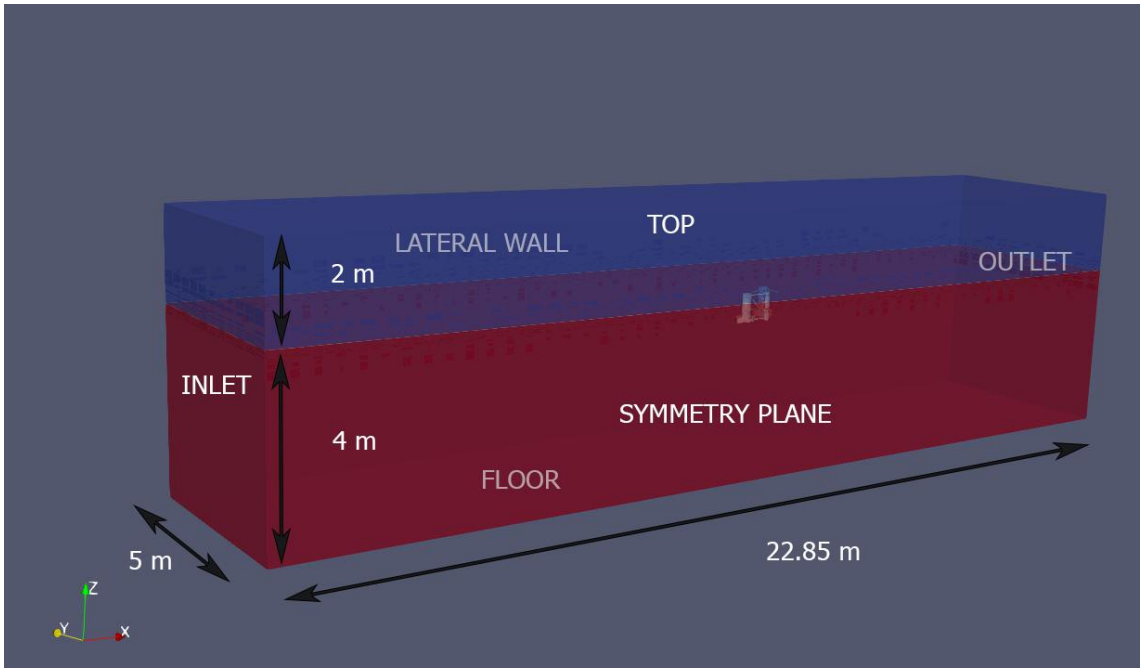


Figure 10: Domain of the simulation with measures and boundaries

The simulations were performed using a symmetry about the x-z plane. Essentially, only half of the floater and water domain were simulated. This introduced assumptions in the model, namely the force about the y axis, and the moments of the x and z axis were considered negligible, thus restricting the degrees of freedom of the model from six to three. However, it also reduced greatly the computational cost by reducing in half the amount of elements involved in each simulation. Without this simplification, this work could have hardly been done within the time and computational power constraints that existed.

In the next sections, some of the decisions that were made in the modelling and simulation process will be explained.

7.2 Physical properties

The physical properties were taken from [41] when available, and from online calculators when not available. Table 1 contains several physical properties from the floater and the environment relevant to the simulation. The fields that were not taken from the measurements in [41] are marked with an asterisk in Table 1.

Table 1: Properties of the fluid field and the floater

Property	Value
Density of the water	$1025\ kg \cdot m^{-3}$
Kinematic viscosity of water*	$1.81 \cdot 10^{-5}\ kg \cdot m^{-1} \cdot s^{-1}$
Density of the air*	$1.225\ kg \cdot m^{-3}$
Kinematic viscosity of the air*	$1.477 \cdot 10^{-5}\ m^2 \cdot s^{-1}$
Gravity*	$9.81\ \mathrm{m} \cdot s^{-1}$
Mass of the floater	$1.3958 \cdot 10^{7}\ kg$
Inertia about the x axis	$1.3947 \cdot 10^{10}\ kg \cdot m^2$
Inertia about the y axis	$1.5552 \cdot 10^{10}\ kg \cdot m^2$
Inertia about the z axis	$1.3692 \cdot 10^{10}\ kg \cdot m^2$
Z coordinate of centre of mass	$8.07\ m$ below sea level

The scaled model of the FOWT is depicted in Figure 11. The mass and inertia properties were referred in the original paper to the full floater. The way of downscaling them to the model properties was to perform Froude scaling. This approach does not predict viscous effects for the full floater, although some research is being done to circumvent this issue [6]. However, the use of this approach is general within the available papers. This dissimilarity should not affect our results, as the measurements were performed on the model scale, and the aim is to validate the computational model against these measurements.

Froude scaling is performed by multiplying the data by a scale factor "s" given by Table 2, where s is the geometric scale factor (in our case $\frac{1}{50}$). The scaled down data introduced in the CFD model is presented in Table 3.

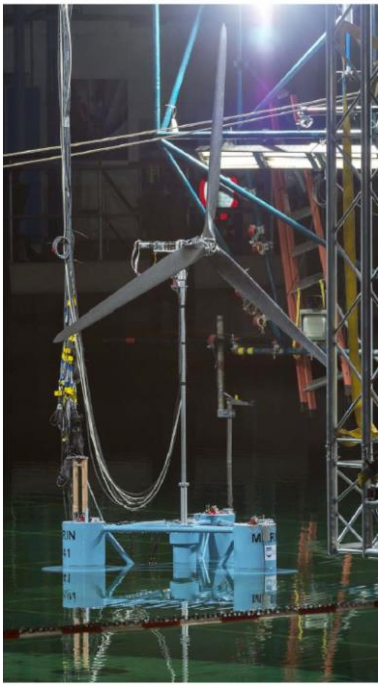


Figure 11: Experimental setup at MARIN [41]

Table 2: Froude scaling factors

Quantity	Scaling
Wave height and length	s
Wave period	$s^{0.5}$
Wave frequency	$s^{-0.5}$
Power density	$s^{2.5}$
Linear displacement	s
Angular displacement	1
Linear velocity	$s^{0.5}$
Angular velocity	$s^{-0.5}$
Linear acceleration	1
Angular acceleration	s^{-1}
Mass	s^{3}
Force	s^{3}
Torque	s^{4}
Power	$s^{3.5}$
Linear stiffness	s^{2}
Angular stiffness	s^{4}
Linear damping	$s^{2.5}$
Angular damping	$s^{4.5}$

Table 3: Scaled down mass properties of the floater

Mass of the floater	$111.664\ kg$
Inertia about the x axis	$22.3152\ kg \cdot m^2$
Inertia about the y axis	$24.8832\ kg \cdot m^2$
Inertia about the z axis	$21.9072\ kg \cdot m^2$
Z coordinate of centre of mass	$0.1614\ m$ below sea level

In most of the simulations performed, a symmetry condition was used and only half a floater was simulated. The mass properties were halved accordingly, and only the centre of mass location was unaffected.

For the mooring lines, the properties used were gathered in Table 4.

Table 4: Scaled down mass properties of the mooring lines

Mass per meter	$0.05024\ kg/m$
Stiffness (nominal)	5989.06 N
Stiffness standard deviation	$598{,}91\ N$ $(10\%\ of\ nominal)$
Diameter	0.0027733 m
Drag coefficient in normal direction	1.5
Drag coefficient in tangential direction	0.5
Added mass coefficient in normal direction	1.5

7.3 Geometry modelling

The geometry was modelled using the CAD software package SolidEdge [42]. SolidEdge is a CAD environment developed by Siemens. As many other CAD software packages, it allows to create and modify geometry in a more agile, robust and intuitive way than doing it directly in the OpenFOAM environment. SolidEdge was used under a student license, and it was chosen instead of other CAD software because of previous experience using it. The geometry definition was relatively straightforward, so this modelling could be replicated in any other CAD package if needed.

The geometry was replicated from Figure 12 with the dimensions scaled down by a factor of 50:

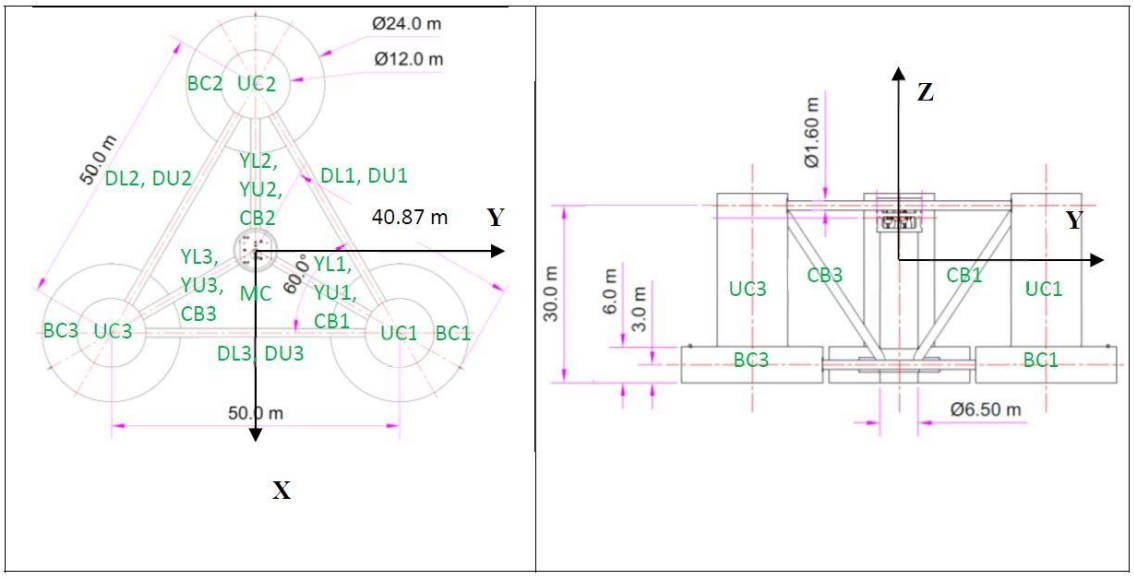


Figure 12: DeepCWind semisubmersible platform measures [41]

In the same way, the disposition of the mooring lines was gathered from Figure 13. The measures were also scaled down by a factor of 50:

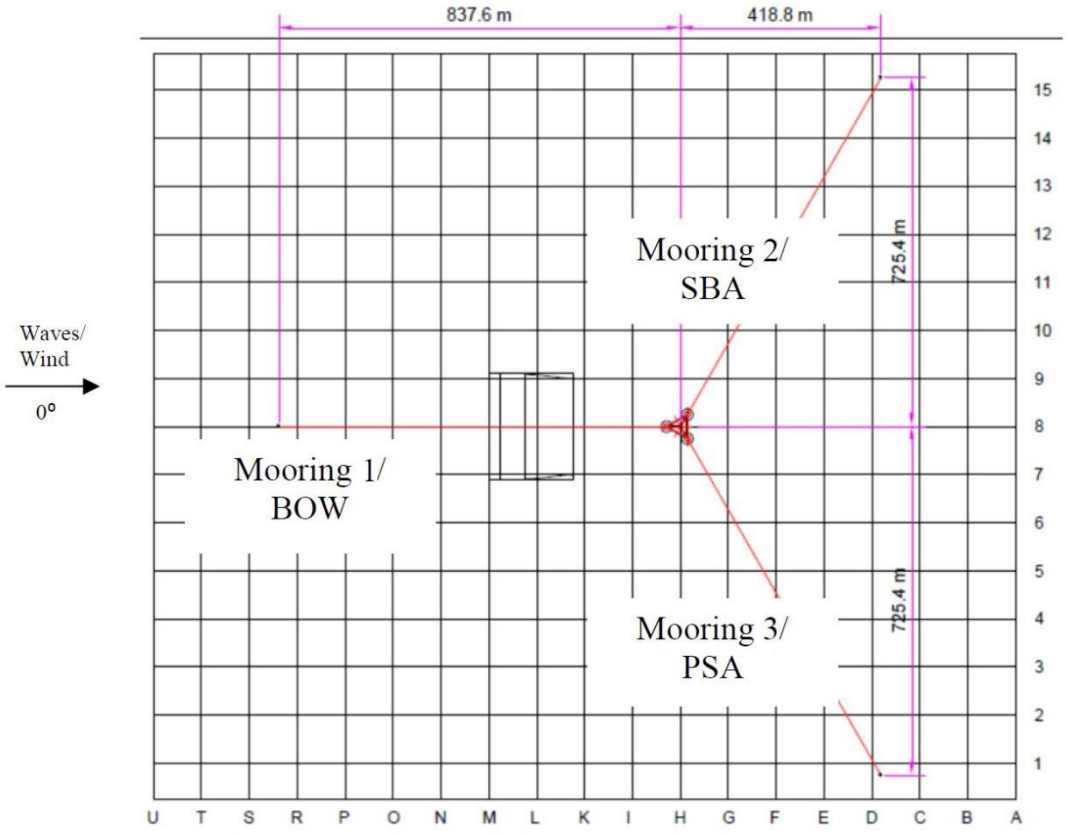


Figure 13: Mooring lines disposition [41]

The CAD model is presented in Figure 14. The geometry was exported as five different STL files. This division was done separating the floater in each of its towers, the central tower and the braces connecting them. This was useful to be able to mesh each part of the floater with a different element size, and to be able to postprocess the forces and moments over the members of the structure separately if needed. The quality of the STL file had to be improved from the standard settings offered by Solid Edge to be able to achieve a sufficient resolution to face the meshing process without issues.

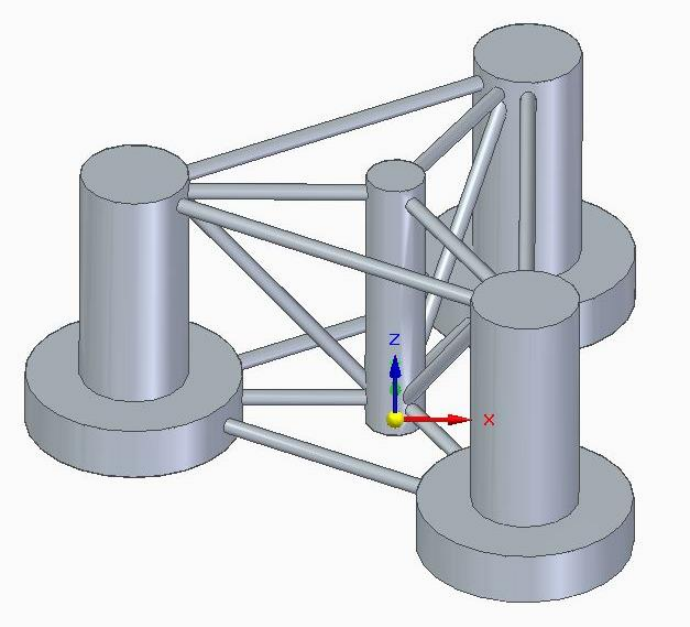


Figure 14: Final CAD FOTW model

Besides the floater, we had to define the box in which the fluid was enclosed. The depth of the water, 4 meters, was selected to replicate the setup from the experiments performed at MARIN, as described in [43]. From the free surface to the top of the box the distance was of 2 meters. The distance from the plane of symmetry to the edge of the simulation was of 5 meters. There were no STL files associated with the box. It was directly defined in the tool blockMesh during the meshing process.

It would have been prohibitively expensive to recreate the entire wave tank, so only a fraction of its length in the wave propagation direction (x direction) was modelled. This was not a trivial decision. On one hand, a short length would mean less elements to simulate and therefore a faster simulation. On the other hand, a shorter domain left less space for the waves to develop. The waves are generated using a simplified model equation, a mathematical abstraction of the wave. While they propagate, they adapt to their developed shape and start to experience the full-model effects not simulated in the boundary condition. Therefore, some space is needed between the wave generation boundary and the floater.

There is one more issue at hand that favoured a long domain: reflection. This phenomenon causes the waves to not stop at the outflow boundary that should absorb them, but rather to bounce back and travel towards the floater again. There is a difference in frequency between the model wave, which the wave absorption boundary condition would dissipate fully, and the real wave, which experiences variations in amplitude and numerical effects induced by the model. The consequence of this difference in frequency is that some components are not dissipated and instead travel back into the domain.

There are several solutions to avoid this in the literature [44] , one of the most common ones in wave absorption zones in which the wave predicted with the CFD model is blended with an analytical solution in a wide area instead of one point. However, that solution was not implemented.

For our simulation, reflection means that the simulation will only be a valid approximation of the intended sea state for a limited interval of time during the simulation. The wave is generated at the leftmost part of the domain and takes some time to arrive to the point where the FOWT model is located. This is considered the initial time stamp at which results ought to be collected. Eventually the wave front goes past the FOWT and arrives at the rightmost point of the free surface, where it is partly reflected and returns back to the floater. When the reflected wave interferes with the floater, the results are no longer recorded.

We introduced also a ramp-up time to allow the height of the waves to be developed gradually without inducing transient effects. A theoretical estimation of both initial time and end time can be performed using the linear wave theory approximation. A derivation can be found on Appendix A.
The range that we ended up selecting was slightly narrower to account for some transient effects. A discussion can be read in the results.

As a compromise between computational cost and accuracy and following the steps of [43], we ended up deciding the length in the x direction as 5 times the wavelength, with the floater situated in the middle.

7.4 Mesh creation

The mesh creation process comprised three processes, as three different OpenFOAM commands had to be run. The first process was to create a cuboid uniform mesh all over the domain box where the fluid goes in. This was done using the blockMesh tool. After that, the surfaceExtractFeatures tool was used to extract the surface features of the STL CAD geometry, so that said surface features could be interpreted by the final meshing tool. This tool is called snappyHexMesh. Meshing with snappyHexMesh supposed the most complex and time consuming step of all the set-up.

In a very simplified way, snappyHexMesh creates a hexahedral mesh by iteratively splitting and moving the background mesh created by blockMesh. It works in three steps. First, it splits the mesh at the surfaces of the STL files and removes the cells in the interior of the files. Then it snaps the edges. That is to say that the mesh vertices are moved towards the STL surface to avoid rough edges. The last step of the process is to

add boundary layers to allow for turbulence models to accurately capture the flow characteristics near the wall. The process is illustrated in Figure 15.

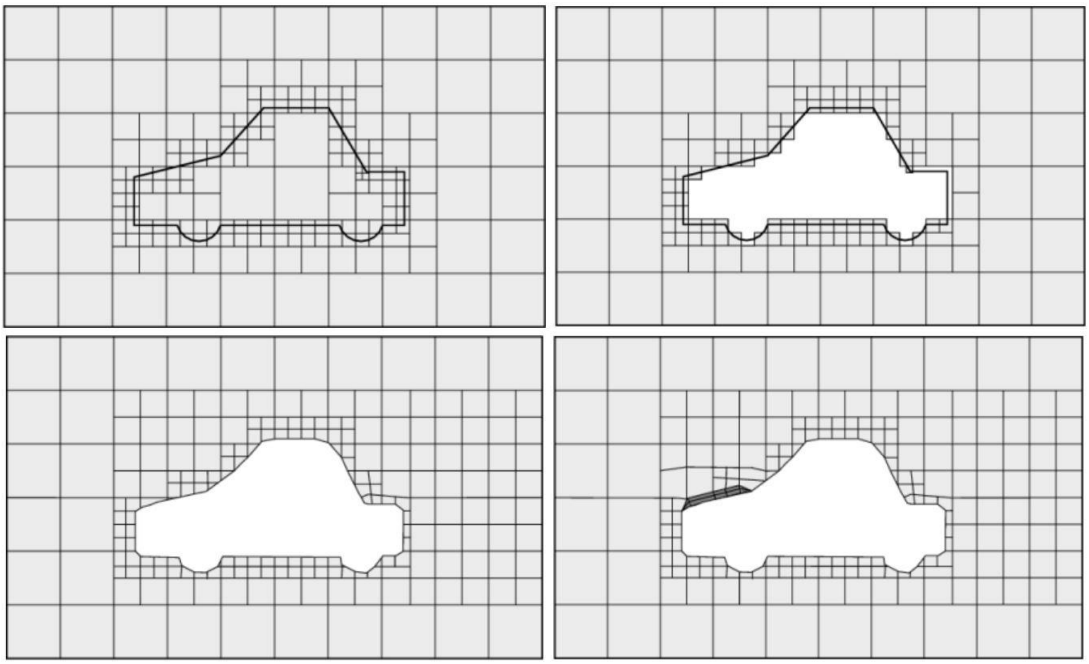


Figure 15: SnappyHexMesh meshing process [45]

Additionally, snappyHexMesh allows for mesh refinement in many ways. It allows to refine surface features, regions of the mesh defined by basic geometric shapes, or refine more those points that are closer to a surface. These refinement options should be matched with the regions we are interested in solving accurately. For our mesh, the regions of interest were:

- **The free surface**: Where the waves were constantly moving, and numerical dissipation associated with a coarse mesh would cause the wave to lose height.
- **The domain of fluid next to the floater**: Where all the interaction between the fluid and the structure would be happening.
- **The floater in itself**: Where a greater number of elements would be corresponded with a more faithful representation of the geometry, essential to avoid generating turbulence due to rough edges.

For the free surface, a box type refinement region was selected. For the domain of fluid next to the floater, a cylinder type refinement region was used. For the floater in itself, the surface features of the braces were refined sufficiently to avoid irregularities in the geometry. The goal was to make our mesh similar to the ones that could be found on the literature, such as the one in Figure 16.

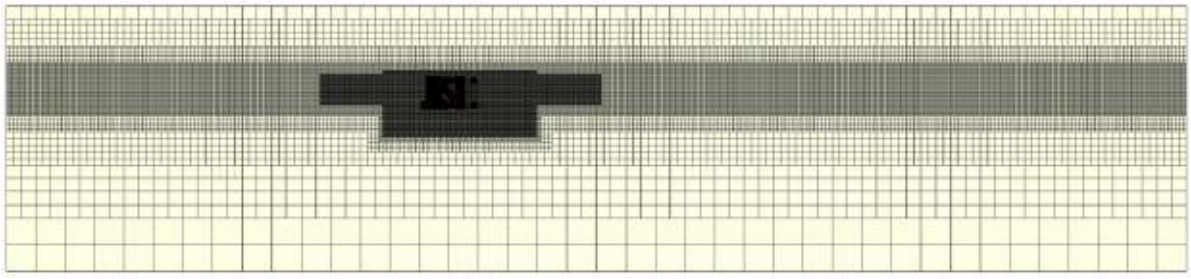


Figure 16: Domain mesh found in the literature [36]

One of the peculiarities of our work is that it includes a simplified 2D study to perform verification and validation. Therefore, both 2D and 3D meshes had to be created. Internally, OpenFOAM considers a 2D mesh a 3D mesh with only a cell width. Fortunately, a good portion of the code to create the 2D meshes could be reused to create the 3D ones, although some problems arose that made it impossible to make both type of meshes perfectly equivalent.

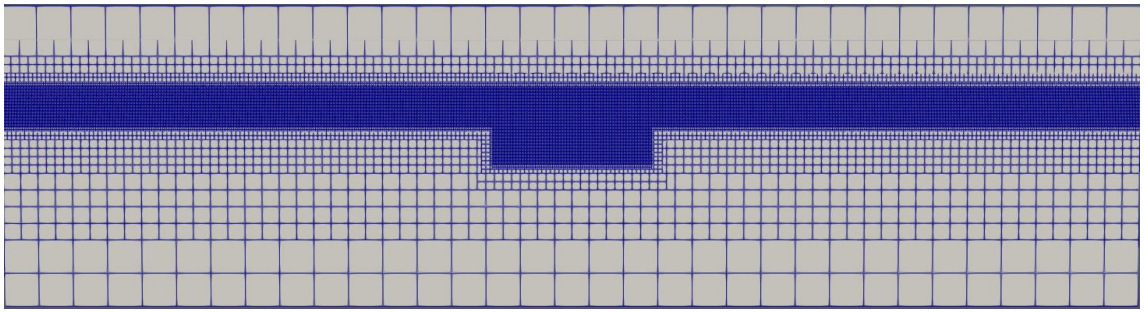


Figure 17: 2D mesh with 52.000 elements

The two-dimensional domains were meshed as seen in Figure 17. These 2D meshes did not include the floater, as the purpose of the 2D simulation was only to assess the quality of the wave generation. The 3D meshes however did include the floater. The 3D meshes were as can be seen in Figure 18 and Figure 19. The interested reader can find additional views of the mesh in Appendix D.

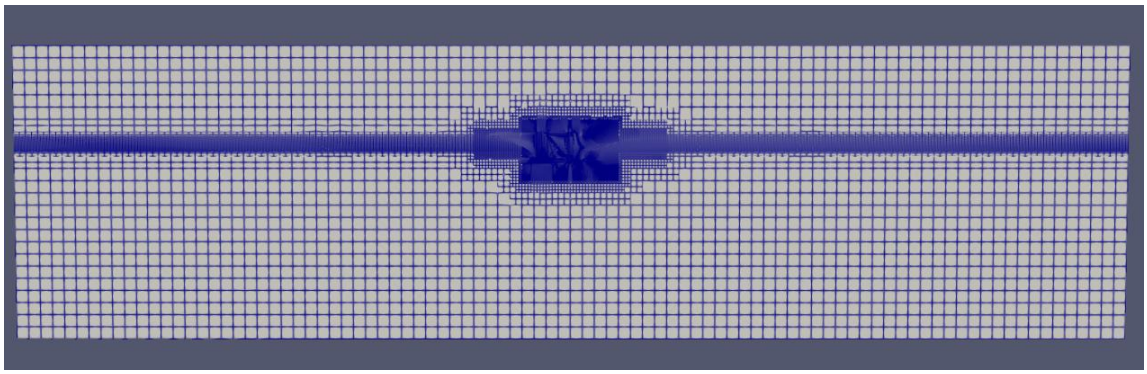


Figure 18: Side view of the simulation domain with a mesh of 2 million cells

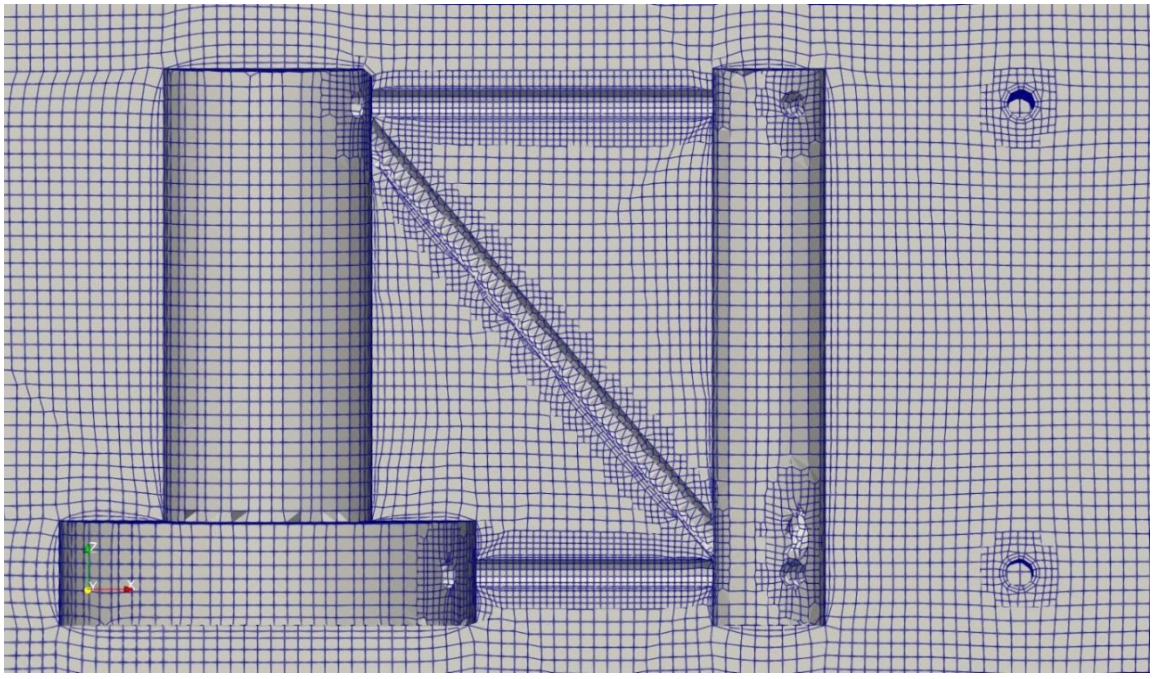


Figure 19: Side view detail of the FOWT with a mesh of 2 million cells

The meshes passed every test snappyHexMesh and checkMesh performs to ensure its quality. Despite this, they were not perfect. Upon closer inspection, one can see that some braces have not been meshed with the structured grid that appears in others. Irregular elements appear at some sharp corners of the floater, and the boundary layers are distorted in some places. The probable cause is that some of the settings of snappyHexMesh were relaxed to favour convergence in the meshing process. However, it was deemed that the meshes were good enough for our purposes. Although this imperfections on the mesh created additional turbulence, the case was mainly driven by inertia, so it was expected that the overall effect was small. The option of using more refined meshes if the uncertainty was too high also ensured the quality of the results.

Other relevant parameter to measure the quality of the mesh regarding turbulence modelling is the y+, which is linked to the size of the boundary layers. In our simulation, as the relative importance of turbulence viscosity was low, we did not undertake the computationally expensive endeavour of getting y+ below one. The

continuous wall function treatment that we used supported a wide range of values of y+, though values between 10-100 were targeted.

7.5 Case setup

Here, some justification of the election of said boundary condition and model settings is given. Although it describes the process alluding to some OpenFOAM tools, the modelling choices would be similar for other CFD solvers. An introduction to the OpenFOAM interface can be found in Appendix E.

OpenFOAM offers a variety of solvers for each possible application. For interphase modelling using the VoF method, the solver interFoam was used. An OpenFOAM case need both boundary conditions and initial values for the fields that are solved for. For interFoam and including dynamic mesh motion, the definition of the boundary conditions of seven fields was required. In addition, several files were used to define other properties of the model or controls of the simulation.

The interphase between the two fluids also had to be defined. This required interpolation of the values of the volume fraction α near the interphase. A precise interpolation method was used as it was deemed it could help improve the stability of the first time steps. SetAlphaFields was used to accomplish said precise interpolation.

For the velocity boundary conditions a no slip condition was chosen for the moving mesh of the floater. However, for the bottom boundary and the wall far from the floater, slip conditions were used. This discrepancy is present because as most of the movement of the wave happened near the free surface, investing computational resources in the full resolution of the bottom and side walls would be a waste of resources.

Turbulence was modelled with a k-ω SST model. This model required three input files: k (turbulent kinetic energy), ν_t (turbulent viscosity) and ω (turbulent specific dissipation rate). The correct initial values for k and ω are uncertain and very difficult to determine for our specific simulation. Ideally, they should be subject to a sensitivity analysis, but as the influence of the turbulent model in the results is small for this simulation, this approach was not implemented. Instead, a reasonable guess was derived with a generic equation offered by the manual. The calculation is available in Appendix C.

The combination of wall functions used allowed the solver to handle a wide range of y+. This was important as the periodicity of the wave movements could cause variations in the y+ that exceeded the recommended range for some formulations.

The waves were generated with a 5^{th} order Stokes model, and were absorbed using a shallow water wave absorption model. Unlike some of the works in the literature, and due to time and computational constraints, no wave absorption zones were used. The wave height and wave period were specified to parameters were selected to match the experimental and computational data that was published ($\mathrm{T} = 1.711$ s, $H = 0.143\ m$).

The wavelength λ was guessed initially using a deep wave approximation. Then this parameter was checked against the waves outputted by OpenFOAM.

To simulate the motion of the floater, a Newmark solver was used. The accelerations were relaxed with a factor of 0.8. As the case was symmetric, motion constraint were defined to keep the movements into a plane. Three degrees of freedom were restricted: roll, yaw, and sway.

7.6 Data operations

7.6.1 General postprocessing

At this step an important decision was how many time steps would the case run for before writing the resulting fields. This was important because of the storage space the files took up. For a mesh of 160 thousand elements 20MB were filled in the memory each time the fields were written, and this happened hundreds to thousands of times during a simulation. For bigger meshes this size increased too, and the sensitivity analysis required seven runs, resulting in a total computational cost on the order of hundreds of gigabytes. The compromise reached was to write the fields ten times per wave period, to allow for a fluid visualization, interpretation, and animation of the results. Forces, wave heights and y+ were less costly to write and therefore were written ten times more often.

The results retrieved from the model were:

- **Wave heights**: The height of the free surface was reported for 20 points equally spaced from inlet to outlet.
- **Interface iso-surfaces**: These iso-surfaces the appearance of the interface between the fluid and the solid and were used to animate this field over time.
- **Y+**: The maximum, average, and minimum y+ for each wall boundary were reported. The y+ field was also retrieved for visualization.
- **Forces and moments**: Were reported for each floater patch as a resultant and separated between viscous and pressure driven terms.
- **Solver information**: For each time step, the iterative error and the number of iterations needed to reach convergence were reported for velocity and pressure.

The results were postprocessed using Moody, Matlab and Bash scripts.

7.6.2 Surrogate model

The variable used for the sensitivity analysis was the mooring stiffness. In a previous experiment by [21] the importance of the mooring stiffness in the motions of the floater

was highlighted. However, the mooring system that was used for their experiment was very different from the original one that we were using and could not be used directly. The uncertainty they input to the mooring stiffness is of about 10% of the nominal value. As we did not have an uncertainty estimation for our mooring system, we used 10% of out nominal mooring stiffness value. Seven simulations were performed with different stiffness levels to create the surrogate model using gPC. The stiffnesses were decided as the points of a Gauss quadrature, as seen in

Table 5.

Table 5: Mooring stiffness for each simulation

Case	Mooring stiffness [kN]
1	3.7429
2	4.5715
3	5.2976
4	5.9890
5	6.6804
6	7.4065
7	8.2351

The sensitivity analysis evaluated the effect of stiffness in two important parameters in the floater response: the forces in the fairleads and the motions of the floater. In order to compare to the literature, the data was decomposed into mean and first order component of a Fourier series, and the uncertainty estimation was performed separately for each. The frequency spectrum is also kept as a result, as it contained useful information. A plot of the gPC model was also gathered.

A sample of a million values was used to perform the MC simulation in the polynomial surrogate model. The probability density function (PDF) was obtained. The uncertainty values and the 95% interval of confidence were also reported.

8 Results

8.1 2D simulations

8.1.1 Flow field

The 2D simulation ran for 40.577 s. Figure 20 shows the propagated waves at the end of the simulation. The focus for this part of the project was on evaluating the quality of the waves. Wave period and wave height were evaluated against the theoretical target. The average wave period was 1.64 s, which was 4.15% lower than the expected period of 1.771s. The wave heights were less accurate. With 0.1357 m at the floater, they were 5.07% lower on average with respect to the wave model used at the inlet. This is comparable with previous experiments that show an error of about 3.5% [6].

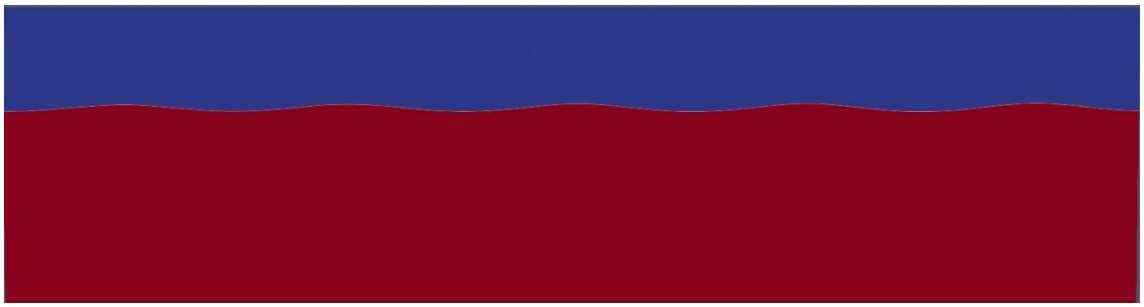


Figure 20: Water surface elevation at t=40.577s

A possible explanation for these differences is that they are produced due to diffusion, that is not predicted by the theoretical wave model. The case was run with a turbulence model, which could have contributed to the viscous dissipation. This was intended. The rationale of using said turbulence model was to make the results more extrapolatable to 3D.

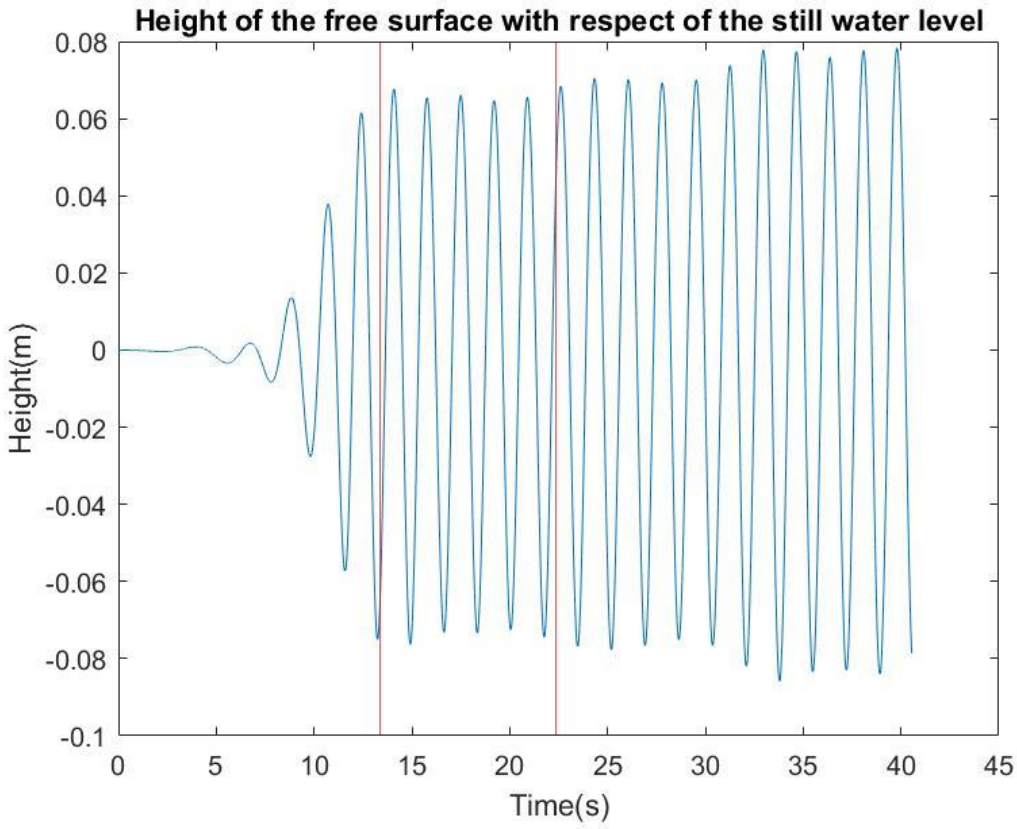


Figure 21: Free surface height at the floater wave gauge for a 500.000 elements 2D mesh

Figure 21 shows the wave height at the x disposition of the FOWT. From the time series, a clear distinction can be seen between the three regimes discussed in the Geometry section of the Methodology. First there is a ramping period, and the waves start to reach the floater location with increasing intensity. Then the wave height stays constant for some time, and after that the reflection begins to vary the water level inducing transient effects. The transitions should theoretically occur at 12.48 s and 29.45 s, respectively. However, even within these intervals, the wave height changes a little bit. Burmester et al. [6] cites two reasons for this behaviour:

- The inlet boundary condition for wave generation uses the Stokes 5th order theory, which only outputs a stable wave if the domain is fully developed.

- The ramp up period means that there is a difference between the height of the waves. The small waves at the beginning travel slower and there is interference with the higher, faster waves that come behind, resulting in a varying range of wave heights.

As a consequence of this, it was decided to include less wave periods and focus on the interval in which the wave height remains more or less constant (between bars in the figure).

8.1.2 Model verification

The verification study was performed to ascertain the number of elements needed to accurately resolve the waves. Six meshes were used, ranging from 125382 elements to 500.000 elements, as presented in Table 6. The verification was performed for the wave height, as it had been the most sensitive variable in previous experiments [6]. The wave heights were normalized using the theoretical wave height.

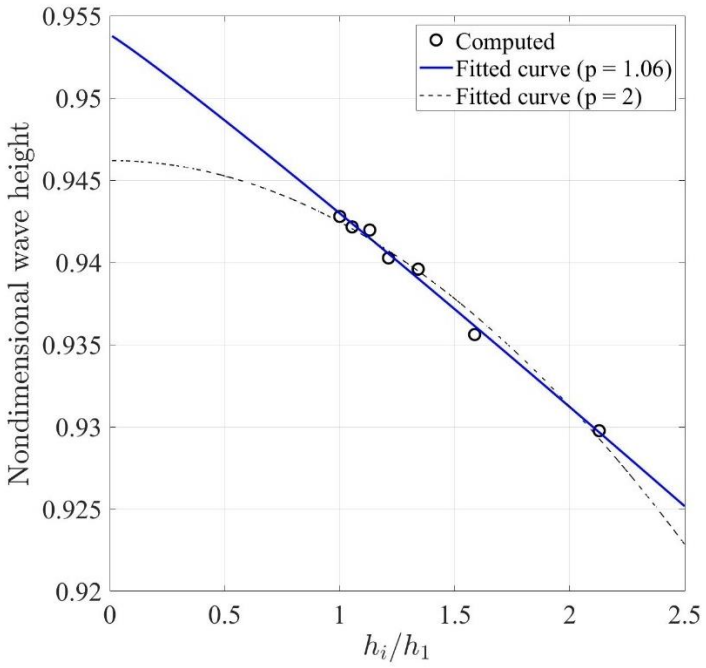


Figure 22: Computed fit (blue) vs theoretical fit (dotted line)

Figure 22 shows the fit of the meshes to the model. The obtained order of convergence was of 1.06. As most of the numerical schemes reached second order, 2 would be the theoretical convergence order. However, modelling assumptions (flux limiters, damping functions) and unstructured meses add scatter and make the convergence order always lower in engineering practice [35]. The convergence we got was deemed good, as it showed monotonic convergence.

Table 6: Uncertainties for each mesh

Mesh	Mesh size (elements)	Uncertainty (%)
Mesh 1	500668	1.493
Mesh 2	427877	1.577
Mesh 3	346574	1.695
Mesh 4	281133	1.824
Mesh 5	207560	2.026
Mesh 6	125382	2.420

The uncertainties associated can be seen in Table 6. They were in the range of 1-2% of the wave height. Other researchers showed uncertainties in the order of 0.3% to 9% for an equivalent experiment in a 3D setting [6]. For comparison, the height of the waves at the floater location exhibited a standard deviation of 1.46% within the selected interval. This shows that the uncertainty of the numerical method is comparable or even lower to the one due to the physics and the settings of the domain. Therefore, it was concluded that any mesh tested would be acceptable to reach the required uncertainty levels.

8.2 3D simulations

8.2.1 Flow field

Figure 23 show the velocity profile at the floater region. Reflected waves from the towers can be seen. The braces were the parts that caused greater velocities, and often water was ejected upwards in the smaller structures of the floater. As the mesh resolution did not allow these to be fully resolved, they appear smeared out.

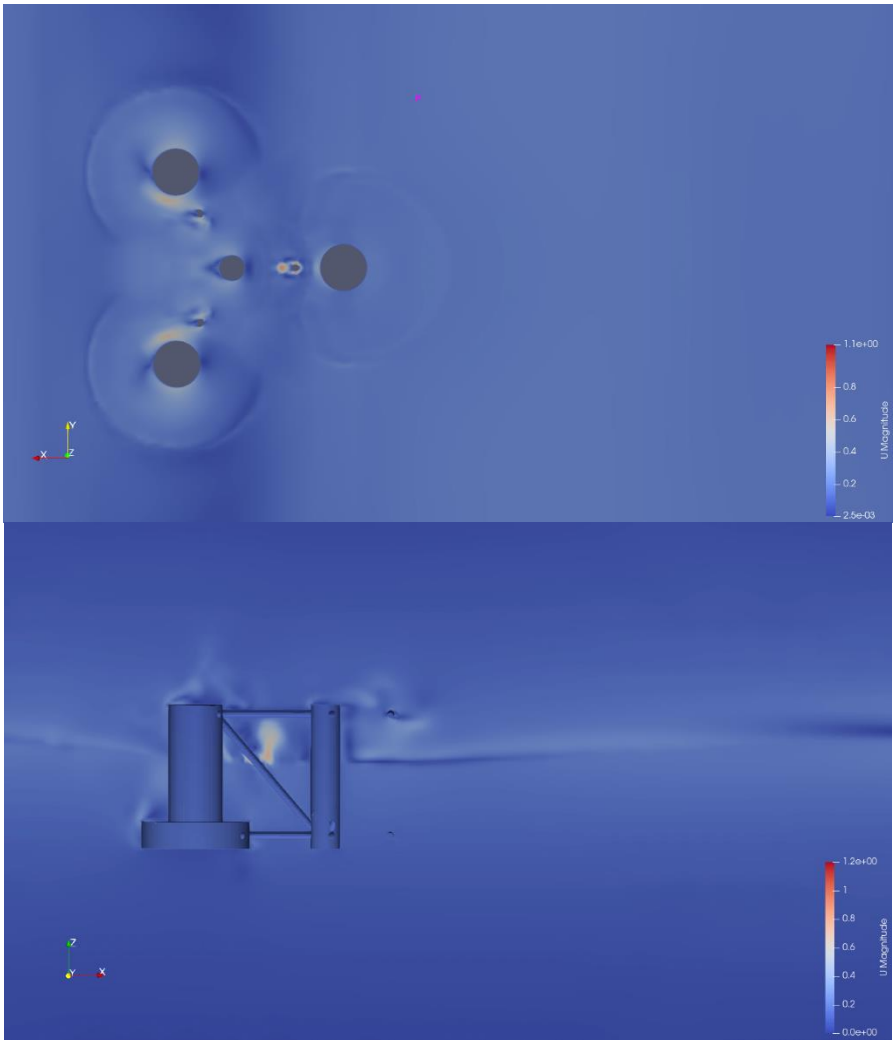


Figure 23: Flow velocity resultant in the floater region at t=19.33 s

Figure 24 shows the turbulent kinetic energy. Like velocity, it spreaded in a radial pattern from the towers and as a wake for the smaller braces.

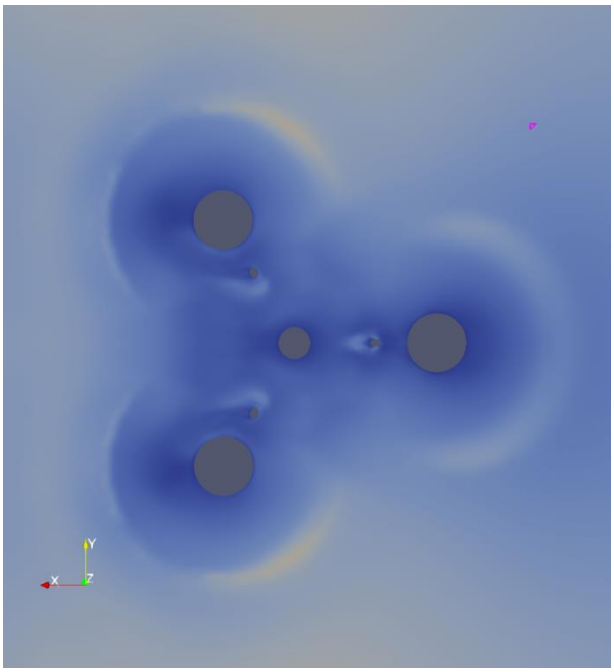


Figure 24: Turbulent kinetic energy at t=19.33 s

The turbulence model is checked against the y+ results. The mean y+ can be found in Figure 25. For our formulation, it was considered sufficient to have mean values on the order of $10 - 10^2$. This is plenty achieved. The results are higher at the towers than at the braces and in the central tower. This is expected, as the later were meshed up to a higher refinement to account for the smaller geometric features contained.

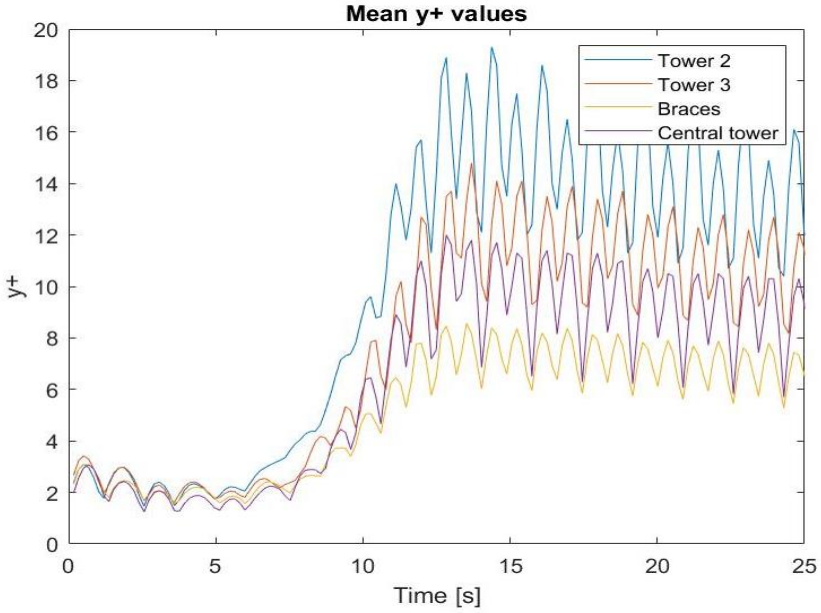


Figure 25: Mean y+ values for a 2e6 mesh

The residuals show the iterative error committed each time step the solver takes. For our approach to solution verification to be valid, the residuals should be two orders of magnitude below the discretization error. The discretization error is on the order of 10^{-2}, so the residuals should be below 10^{-4}. As seen in Figure 26, for the time interval of interest of 12 to 25 seconds, this happens. It is therefore considered the residuals are kept sufficiently low.

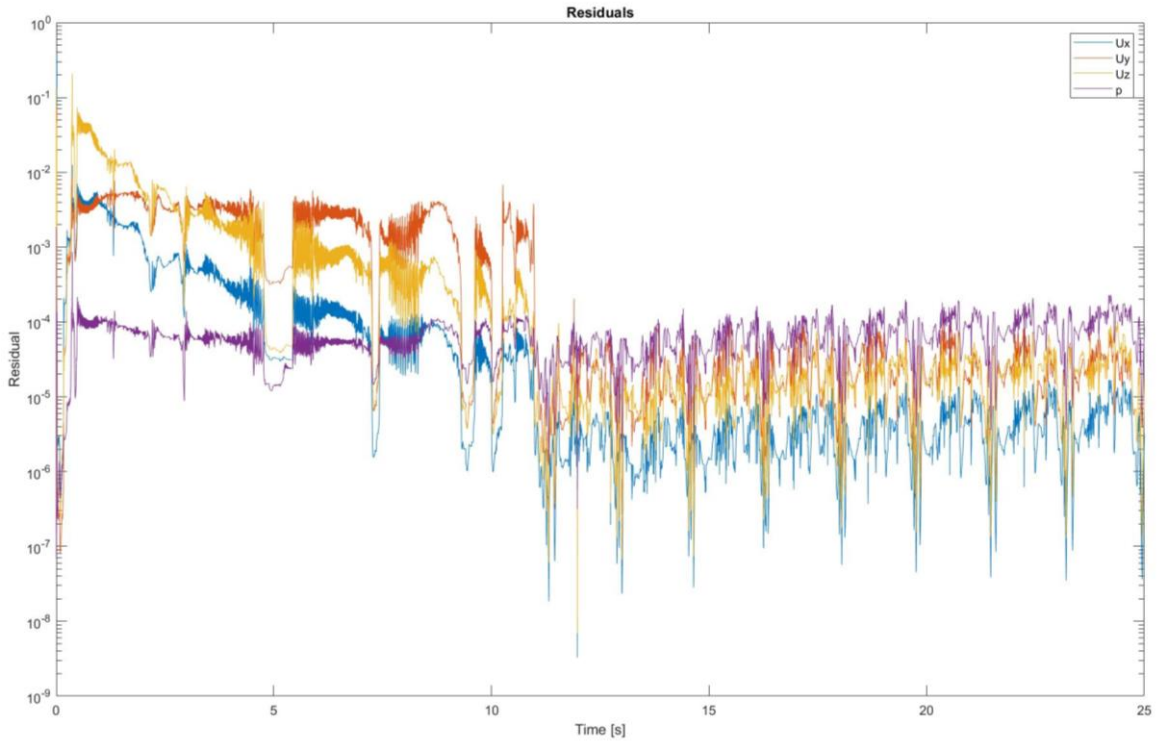


Figure 26: Residuals for a 25 s moored simulation in a 2e6 mesh

8.2.2 Verification

In the 2D simulation we concluded that the meshes resolved the waves with plenty of precision. As a target for the 3D meshes, the desired number of elements per meter on the free surface region was about 70 for an uncertainty level comparable with the one induced by transient effects. The meshes that were used ranged from 73.33 to 80, so it was concluded that the uncertainty level would be sufficiently low and the model would be verified.

As the meshes were small, it was deemed interesting to try to perform verification for the 3D cases. This could not be done in the end, as only three meshes could be included due to time constraints, and the convergence trend was too weak for the uncertainty estimation methods to give a sound result. One interesting observed effect was that the meshes appeared to be showing oscillatory convergence.

8.2.3 Sensitivity analysis

First, plots of the floater motions and the fairlead forces with respect to time are presented. At a first glance, the results of the motions of the floater show very little difference between the runs, while the mooring forces do present a visible difference. Figure 27 and Figure 28 illustrate this difference. It is therefore expected to obtain a higher uncertainty in the mooring forces than in the motions of the floater.

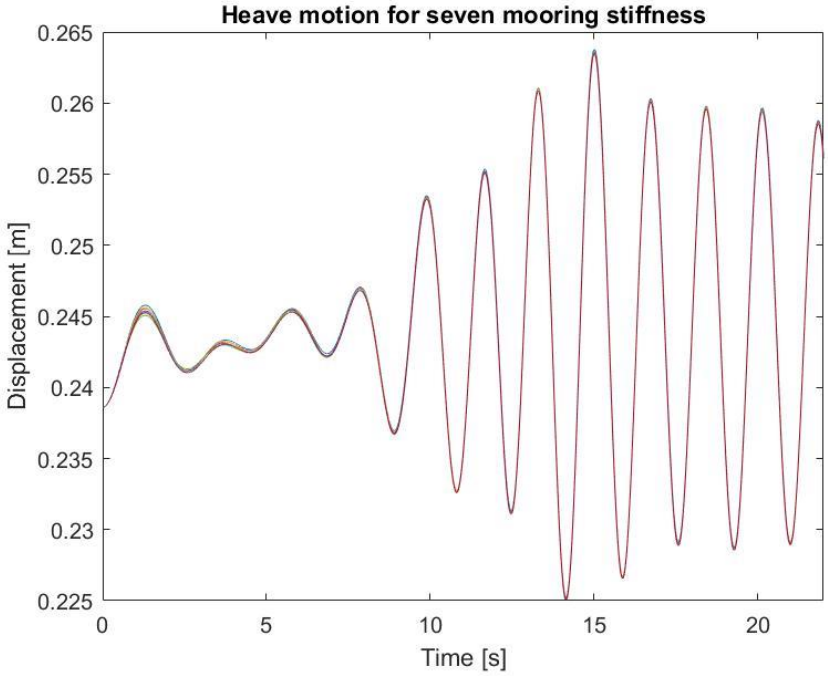


Figure 27: Heave motion for the seven runs

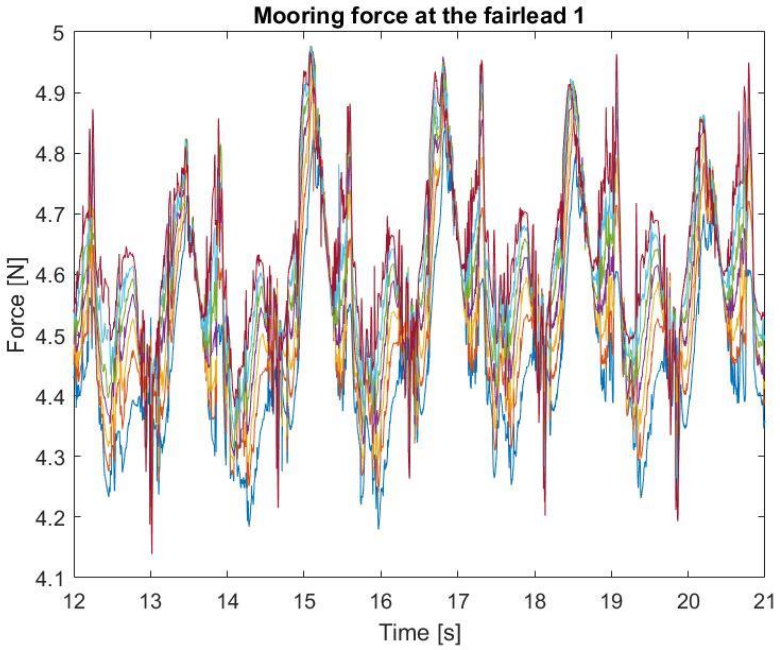


Figure 28: Forces in the fairlead of the fairlead most exposed to the waves

The results of the spectrum of frequencies, without the mean component, are presented next. Figure 29 shows the FFTs of the seven simulations performed. For the motions of the floater, the change in the spectrum with the mooring stiffness is inappreciable. The small differences also anticipate small influence of the mooring stiffness in the overall motion results. The frequency spectrum appears to be smooth, without other peaks at higher frequencies.

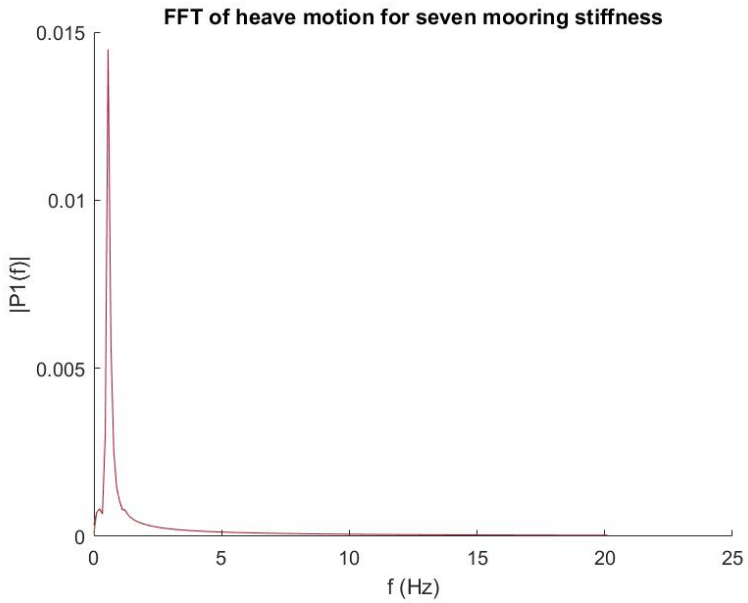


Figure 29: FFT of heave motion for the seven runs

The forces at the fairlead, however, show an interesting behaviour as the mooring changes. For the mooring line 1, directly in front of the waves, the first and second biggest peaks at Figure 30 and Figure 31 appear close to the excitation frequency (0.56 Hz) and three times this frequency (1.78 Hz) respectively. When the mooring is slack, the dominant frequency component is the wave excitation frequency one. However, as the mooring becomes stiffer, the higher frequency component becomes dominant. As a downside, in mooring lines 2 and 3 as the mooring becomes stiffer a greater force appears. The spectrum is quite rough, with peaks at several harmonics.

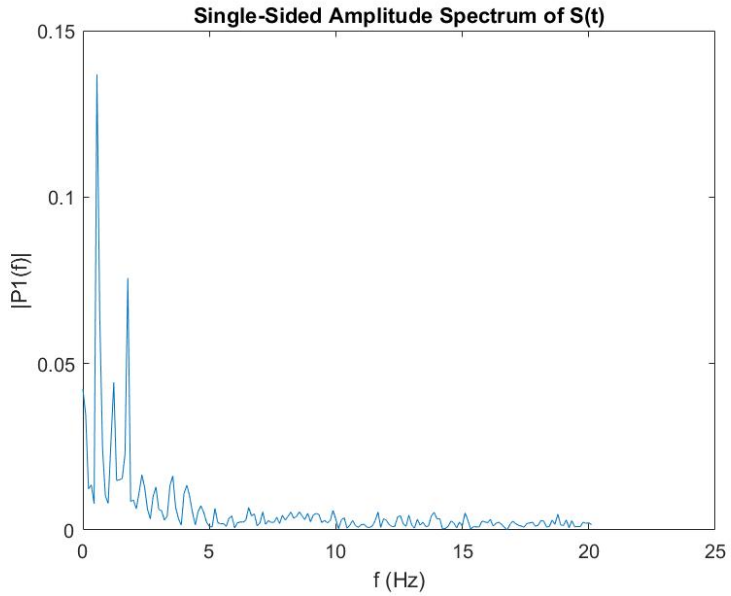


Figure 30: FFT of the mooring line 1 with a stiffness of 3.7429 kN

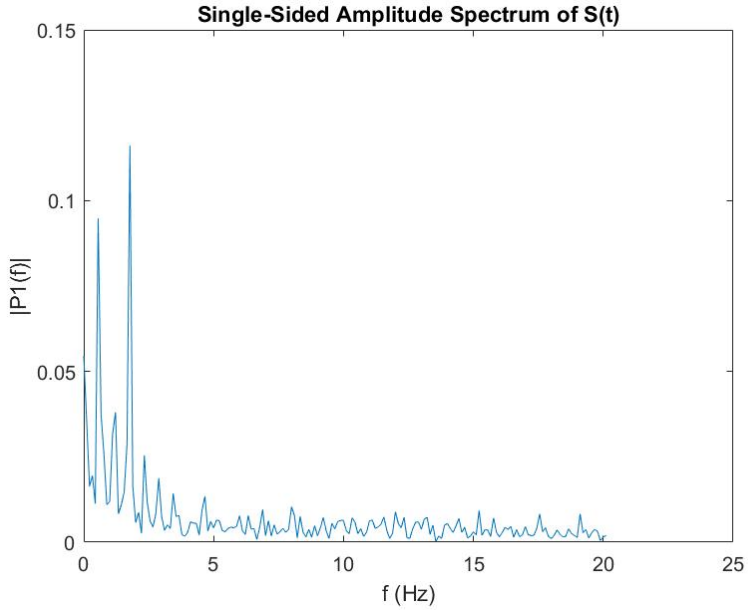


Figure 31: FFT of the mooring line 1 with a stiffness of 8.2351 kN

The surrogate models that arise from using gPC are plotted in Figure 32. The shape of the polynomial was distributed in a wide variety of arrangements, usually showing significant slopes at the extremes of the region of interest. The dots represent the points that were simulated using the CFD model. Here, only two plots are presented as an example.

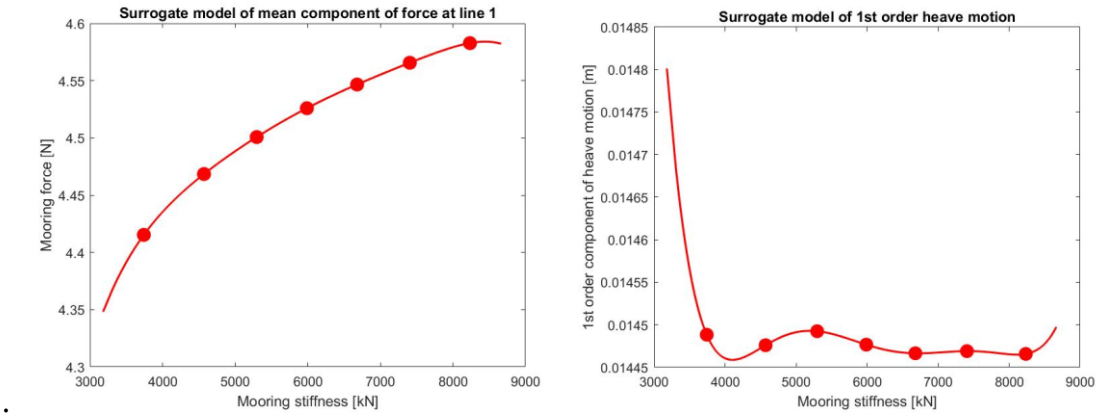


Figure 32: Example plots of surrogate models

The force component monotonically increase for the force surrogate models, while for the motions some of the plots show both increases and decreases on the motion amplitude when the stiffness increases.

From the surrogate models presented, a normal MC method can be used to estimate the probability density functions (PDF). These were estimated using a sample of a million random mooring stiffness values from the assumed distribution and classifying the resulting motions and forces in an histogram. Figure 33 and Figure 34 show the probability density functions (PDF) for the motion and forces. The probability density distributions were very smooth for most of the force results, while for the floater motions the PDF presented spikes and abrupt changes. The results only resemble a normal distribution for the forces. The motions are much less well behaved, inducing to believe other factors aside from the mooring forces, such as numerical artifacts, were disrupting the results.

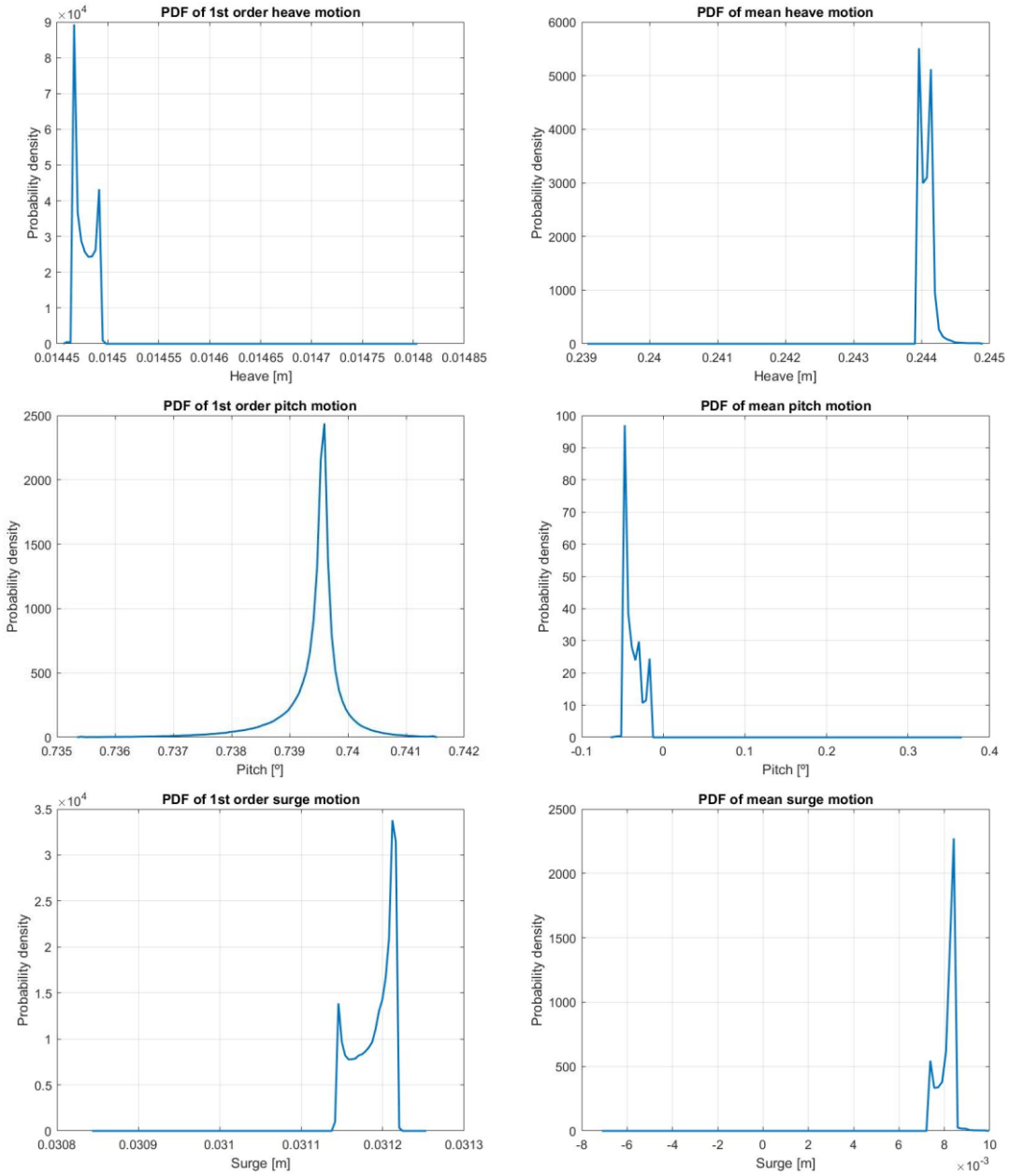


Figure 33: Probability densitity function for the FOWT motions

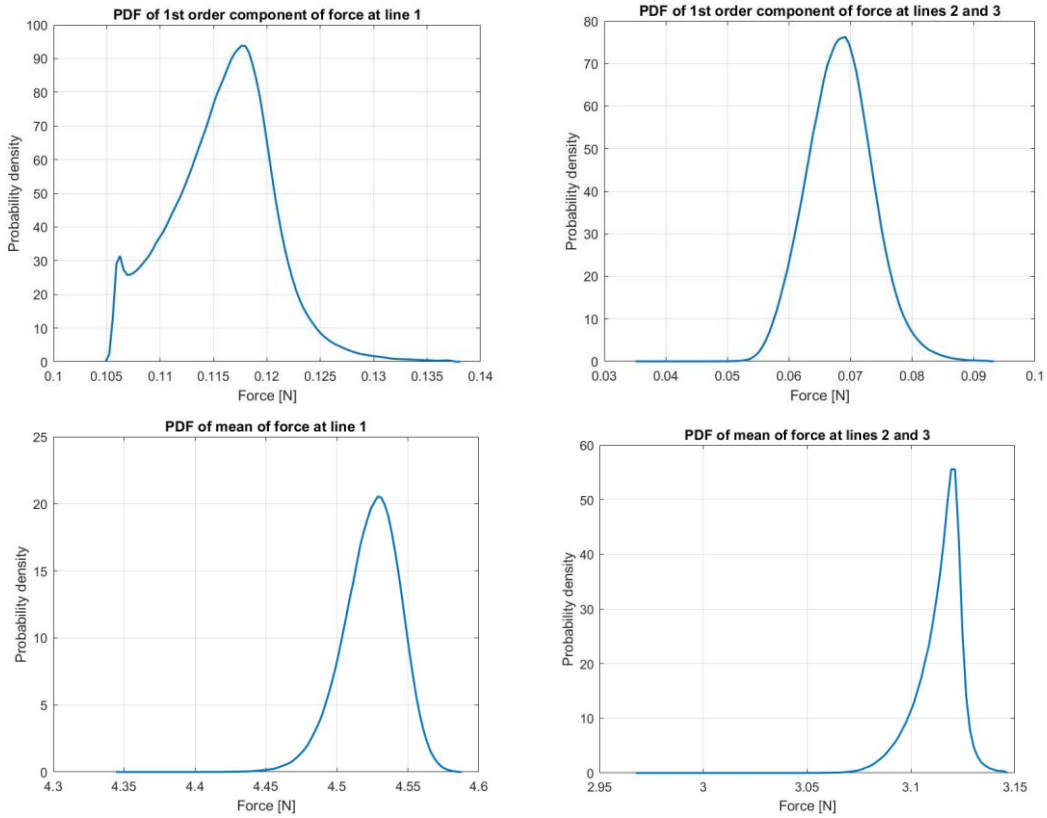


Figure 34: Probability density function of the mooring forces

Lastly, Table 7 and Table 8 show the results of the UQ analysis for the floater motions and forces. It is the the key result of this work, as the uncertainty derived from the mooring stiffness is a measure of the importance of said parameter in the behaviour of each force and motion. The results presented are the size of the confidence interval of the PDFs. This interval is defined as the one that contains 95% of the values of the MC simulation. In the last row, the value is normalized using the mean value to make the results easier to compare.

Table 7 shows the results for the motion components. The uncertainties for the first order components are under 0.2% for the three motions. The mean is much more sensible to variations, with pitch being the least well predicted with 41.5% uncertainty.

Table 7: Uncertainties of the floater motions

	Heave		**Pitch**		**Surge**	
	Mean	**1st order**	**Mean**	**1st order**	**Mean**	**1st order**
Value (mm)	24.4	1.44	-3.82	73.94	0.81	3.12
Uncertainty (mm)	0.016	0.0013	1.6	0.13	0.053	0.0035
Uncertainty (%)	0.065	0.090	41.50	0.17	6.52	0.11

For the mooring forces, the opposite happens. Table 8 shows the results for the mooring forces. The mean was not very sensible with uncertainties below 1%, while the first order components were less well represented, with uncertainties of up to 15%.

Table 8: Uncertainties for the mooring forces

	Mooring line 1		**Mooring lines 2 and 3**	
	Mean	**1st order**	**Mean**	**1st order**
Value (N)	4.52	0.12	3.11	0.07
Uncertainty (N)	0.0397	0.0093	0.0207	0.01049
Uncertainty (%)	0.87	7.75	0.67	14.98

For this wave height and period, the 1st order component motions of the floater are barely affected by the mooring stiffness. It suggests there is less likelihood of aligning the CFD simulations with the experiments by improving the mooring model when it comes to the RAOs. This result is in contrast with [21], which identifies the surge motion as a significantly sensible quantity to the mooring stiffness. More work is needed to identify the reason for this discrepancy.

The results also point at the alternating component of the mooring forces being significantly affected by the mooring stiffness, which is an interesting insight. More loading cases should be subjected to this analysis in order to identify patterns that could become useful for proper design of the mooring lines.

8.2.4 Motions of the floater compared to previous research

Figure 35 and Figure 36 show the agreement of the performed simulations with the experimental results at MARIN, as presented in [4]. The figures include the discrepancy as a percentage. The mean components are not at all well represented. This was however to be expected as the stationary position of the floater was very dependent on the experimental setup and cannot be compared directly.

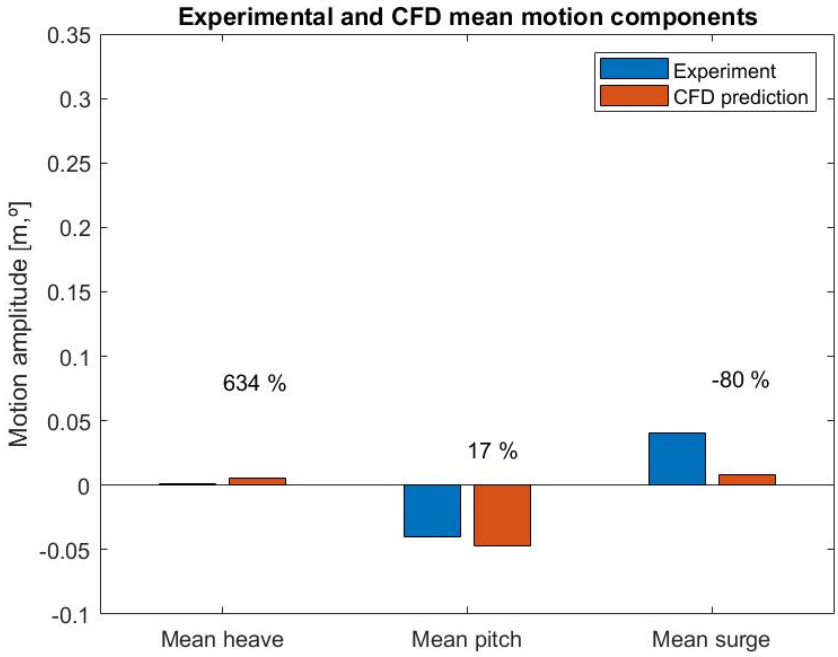


Figure 35: Comparison of mean motion results

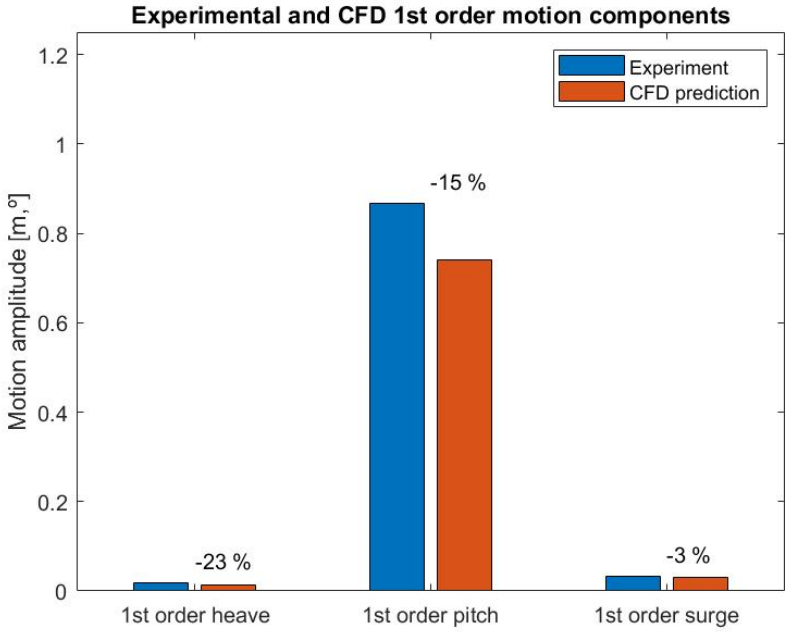


Figure 36: Comparison of 1st order component motion results

The first order component motion is more relevant. Our model is underpredicting heave by 23%, pitch by 15% and surge by 3%. This underprediction is also present in [4], which reports a difference of 27% for heave, 16% for pitch and 4.8% for surge. Though this underprediction is a common problem in most simulations performed in this problem, the similarity between our results and the results of other teams hint at the dynamic mooring model offering only a small improvement in motion prediction for this wave loading case.

Other possible causes for this underprediction may lay in the data gathered from the physical tests of the moored floater. Due to size constraints, the mooring lines are artificially corrected for axial stiffness with a spring, which may introduce differences in the mooring behaviour [41]. Additionally, no uncertainty estimation was provided for the first testing campaign, so there is no way to evaluate the effect of deviations of the measured properties or the RAOs.

9 Future work

Overall, the objectives of this thesis have been accomplished. We got results from the sensitivity analysis that added to the current research performed in the topic. Furthermore, we compared the model results to the results of other researchers with good agreement. However, several improvements could be made providing both time and computational resources were available for the project.

In order of perceived importance, they are:

- **Including more parameters in the gPC analysis**: Though this work focused on the mooring stiffness, other parameters were interesting. The drag and lift coefficients of the mooring lines and other such parameters could improve our understanding of the mooring model and thus provide assessment on where to focus the modelling efforts.

- **Verification of the 3D meshes**: The 2D verification performed was a compromise solution that was useful to avoid a great investment of resources. However, rigorous assessment of uncertainty for the 3D meshes would improve the credibility of the results.

- **Including more wave loading cases**: The simulations were performed only for one wave height and period. However, real sea states usually involve several. Including other wave cases would allow to compare the results and their behaviour as wave frequency changes.

- **Improvements of the meshes**: Some areas of the meshes had slight imperfections. A correction of those would improve the quality of the results. With enough computational power it could be interesting to resolve the floater walls to y+ values of less than one, and create smoother transitions between the free surface and floater area and the rest of the mesh.

10 References

[1] OpenFOAM, "OpenFOAM," OpenFOAM Foundation, 2021. [Online]. Available: https://www.openfoam.com/. [Accessed 29 5 2021].

[2] J. Palm and C. Eskilsson, "Moody: mooring dynamics. User manual," Göteborg, 2019.

[3] IEA Wind, "Annex 30 – Task Extension Proposal, 2019-2022. Offshore Code Comparison Collaboration, Continued, with Correlation, and unCertainty (OC6)," Golden, Colorado, 2018.

[4] Y. Wang, H.-C. Chen, G. Vaz and S. Burmester, "CFD Simulation of Semi-Submersible Floating Offshore Wind Turbine under Regular Waves," 2020.

[5] G. M. Paredes, Claes Eskilsson and A. P. Engsig-Karup, "Uncertainty Quantification in Mooring Cable Dynamics Using Polynomial Chaos Expansions," *Journal of Marine Science and Engineering,* vol. 8, no. 62, 2020.

[6] S. Burmester, G. Vaz and O. e. Moctar, "Towards credible CFD simulations for floating offshore wind turbines," *Ocean Engineering,* vol. 209, 2020.

[7] IRENA, "Wind Energy," [Online]. Available: https://www.irena.org/wind. [Accessed 2021].

[8] WindEurope, "Wind energy in Europe: 2020 Statistics and the outlook for 2021-2025," 2021.

[9] IRENA, "Renewable power generation costs in 2019," 2020.

[10] NREL, *Webinar: Overview of Floating Offshore Wind,* 2020.

[11] J. Cruz and M. Atcheson, Floating Offshore Wind Energy, Springer, 2016.

[12] DTU Wind Energy, ESMAP, "Global Wind Atlas," 2020. [Online]. Available: https://globalwindatlas.info/about/introduction. [Accessed 2021].

[13] R. Mckenna, P. Leye and W. Fichtner, "Key challenges and prospects for large wind turbines," *Renewable and Sustainable Energy Reviews,* vol. 53, pp. 1212-1221, 2016.

[14] Swedish Environmental Protection Agency, "Noise annoyance from wind farms: a review," 2003.

[15] NMPPenergy, "Energy Education 101: Understanding Energy Capacity and Capacity Factor," [Online]. Available: https://www.nmppenergy.org/feature/capacity_factor. [Accessed 2020].

[16] Equinor, “What we do. Floating wind,” [Online]. Available: https://www.equinor.com/en/what-we-do/floating-wind.html. [Accessed 2021].

[17] E. Konstantinidis and P. Botsaris, “Wind turbines: current status, obstacles, trends and technologies,” IOP Conference Series: Materials Science and Engineering, 2016.

[18] Z. Jiang, “Installation of wind turbines: a technical review,” *Renewable and Sustainable Energy Reviews,* no. 139, 2021.

[19] K. Michael, A. Frank, D. Frank and J. Grossmann, “Floating offshore wind - Economic and ecological challenges of a TLP solution,” *Renewable Energy,* vol. 126, 2018.

[20] W. West, “The Influence of Synthetic Mooring Line Stiffness Model Type on Global Floating Offshore Wind Turbine Performance,” *Journal of Physics,* 2020.

[21] A. Robertson, E. E. Bachynski, S. Gueydon, F. Wendt and P. Schünemann, “Total experimental uncertainty in hydrodynamic testing of a semisubmersible wind turbine, considering numerical propagation of systematic uncertainty,” *Ocean Engineering,* vol. 195, 2020.

[22] F. Moukalled, L. Mangani and M. Darwish, The Finite Volume Method in Computational Fluid Dynamics, Springer, 2016.

[23] OpenFOAM Foundation, “OpenFOAM v2012 UserGuide. k-omega Shear Stress Transport (SST),” 2017. [Online]. Available: https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence-ras-k-omega-sst.html. [Accessed 5 2021].

[24] R. Bardia, “ Understanding MULES,” 2013. [Online]. Available: https://sites.google.com/site/trujillolabgroup/understanding-mules. [Accessed 5 2021].

[25] M. Ge and R. Bensow, “Comparison of Free Surface Capturing Approaches in OpenFOAM for Ship Resistance Prediction,” Gothenburg.

[26] NASA, “ FAST User Guide,” 1999. [Online]. Available: https://www.nas.nasa.gov/Software/FAST/RND-93-010.walatka-clucas/htmldocs/chp_16.surferu.html. [Accessed 5 2021].

[27] Q. Li, W. Chen, S. Liu and H. Fan, “Topology optimization design of cast parts based on virtual temperature method,” *Computer-Aided Design,* vol. 94, 2017.

[28] A. Wimshurst, “[CFD] The PISO Algorithm,” 2019.

[29] H. Jasak, “Error Analysis and Estimation for the Finite Volume Method with Application to Fluid Flow,” 1996.

[30] M. Muhammad, “Wind Prediction Modelling and Validation using Coherent Doppler LIDAR Data,” 2016.

[31] OpenFOAM Wiki, “Parameter Definitions - dynamicMotionSolverFvMesh,” 16 11 2016. [Online]. Available: https://openfoamwiki.net/index.php/Parameter_Definitions_-_dynamicMotionSolverFvMesh#Solver_Type. [Accessed 05 2021].

[32] J. Palm, C. Eskilsson, G. M. Paredes and L. Bergdahl, “Coupled mooring analysis for floating wave energy converters,” *International Journal of Marine Energy,* no. 16, pp. 83-99, 2016.

[33] T. Xing, “A general framework for verification and validation of large eddy simulations,” *Journal of Hydrodynamics,* vol. 27, no. 2, pp. 163-175, 2015.

[34] T. Xing and F. Stern, “Factors of Safety for Richardson Extrapolation,” *Journal of Fluids Engineering,* vol. 132, no. 6, 2010.

[35] L. Eça and M. Hoekstra, “A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies,” *Journal of Computational Physics,* vol. 262, no. ISSN 0021-9991, pp. 104-130, 2014.

[36] ASME, “Standard for Verification and Validation in Computational Fluid Dynamics,” 2009.

[37] AIAA, Guide: Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (AIAA G-077-1998(2002)), American Institute of Aeronautics and Astronautics, Inc., 1998.

[38] NASA, “NPARC Alliance CFD Verification and Validation,” February 2021. [Online]. Available: https://www.grc.nasa.gov/www/wind/valid/tutorial/bibliog.html#AIAA98. [Accessed March 2021].

[39] S. Marelli, N. Luthen and B. Sudret, “UQlab User Manual: Polynomial Chaos Expansions,” Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Zurich, 2021.

[40] UQlab, “UQlab: The framework for Uncertainty Quantification,” ETH Zurich, 2021. [Online]. Available: https://www.uqlab.com/. [Accessed 5 2021].

[41] A. Robertson, J. Jonkman, F. Wendt, A. Goupee and H. Dagher, “Definition of the OC5 DeepCwind Semisubmersible Floating,” NREL and University of Maine.

[42] SIEMENS, “SolidEdge,” [Online]. Available: https://solidedge.siemens.com/. [Accessed 4 2021].

[43] Y. Wang, H.-C. Chen, G. Vaz and S. Mewes, “Verification study of CFD simulation of semi-submersible floating offshore wind turbine under regular

waves," in *IOWTC2020-3558*, Boston, 2020.

[44] M. Israeli and S. A. Orszag, "Approximation of radiation boundary conditions," *Journal of Computational Physics*, vol. 41, no. 1, pp. 115-135, 1981.

[45] C. Greenshields, "OpenFOAM v6 User Guide: 5.4 Mesh generation with snappyHexMesh," CFD Direct, 2018. [Online]. Available: https://cfd.direct/openfoam/user-guide/v6-snappyhexmesh/. [Accessed 2021].

[46] A. McMaster, "BEng (Hons) Mechanical Engineering with Renewable Energy Final Year Project," [Online]. Available: https://adammcmaster.wordpress.com/wave-tank-testing/froude-scaling-laws/. [Accessed 2021].

Appendix A: Time until reflection calculation

The wave that we are simulating can be approximated as a deep wave. The expression for the wave phase speed is:

$$c_p = \frac{1}{2}\sqrt{\frac{g\lambda}{2\pi}} \tag{31}$$

where λ is the wavelength. For this problem:

$$\lambda = 4.56\, m \tag{32}$$

$$g = 9.81\, \frac{m}{s^2} \tag{33}$$

The wave speed turns out to be:

$$c_p = 1.345\, \frac{m}{s} \tag{34}$$

And knowing the distance between the FOWT and both inlet and outlet:

$$d_{i/0} = 11.41\, m \tag{35}$$

And the ramp up time:

$$t_{ramp} = 4s \tag{36}$$

The initial and final times are:

$$t_{ini} = t_{ramp} + \frac{d_{i/o}}{c_p} = 12.48\, s \tag{37}$$

$$t_{end} = t_{ramp} + 3\frac{d_{i/o}}{c_p} = 29.45\ s \tag{38}$$

Appendix B: Boundary and initial conditions

In this appendix, the boundary and initial conditions for each field are presented:

Boundary condition	Top	Bottom	Inlet	Outlet	Left wall	Symmetry	Floater
A	Zero Gradient	Zero Gradient	Wave Alpha	Wave Alpha	Zero Gradient	Symmetry Plane	Zero Gradient
u	Pressure inlet-outlet velocity	Slip	Wave velocity	Wave velocity	Slip	Symmetry Plane	Moving wall velocity
p_{rgh}	Total pressure	Fixed flux pressure	Fixed flux pressure	Fixed flux pressure	Fixed flux pressure	Symmetry Plane	Fixed flux pressure
k	Inlet-outlet	Zero Gradient	Fixed value	Inlet-outlet	Zero Gradient	Symmetry Plane	k Low Re Wall Function
ω	Inlet-outlet	Zero Gradient	Inlet-outlet	Inlet-outlet	Zero Gradient	Symmetry Plane	ω Wall Function
ν_t	Calculated	Zero Gradient	Calculated	Calculated	Zero Gradient	Symmetry Plane	ν_t Blended Wall Function
Point displacement	Fixed value	Fixed value	Fixed value	Fixed value	Fixed value	Symmetry Plane	Calculated

	α	U	p_{rgh}	k	ω	ν_t	Point displacement
Initial condition	Uniform 0	Uniform (0 0 0)	Uniform 0	Uniform $1.5e-6$	Uniform 4.35e-3	Uniform 0	Uniform 0

Appendix C: Turbulence initialization

The initial values were computed as the OpenFOAM reference recommended if there was no other data, using the formulas:

$$k = \frac{3}{2}\left(I\left|u_{ref}\right|\right)^2 \tag{39}$$

$$\omega = \frac{k^{0.5}}{0.548L} \tag{40}$$

I is the turbulent intensity, which was chosen to be of 1%. The reference velocity was chosen as the mean velocity over the area of the floater, of about 0.1 m/s. The reference length was chosen as the diameter of one of the towers, 0.48 m. The final values used as initial condition were:

$$k = 1.5 \cdot 10^{-6}\frac{J}{kg} \tag{41}$$

$$\omega = 4.65 \cdot 10^{-3}\ \mathrm{s}^{-1} \tag{42}$$

Appendix D: Additional mesh views

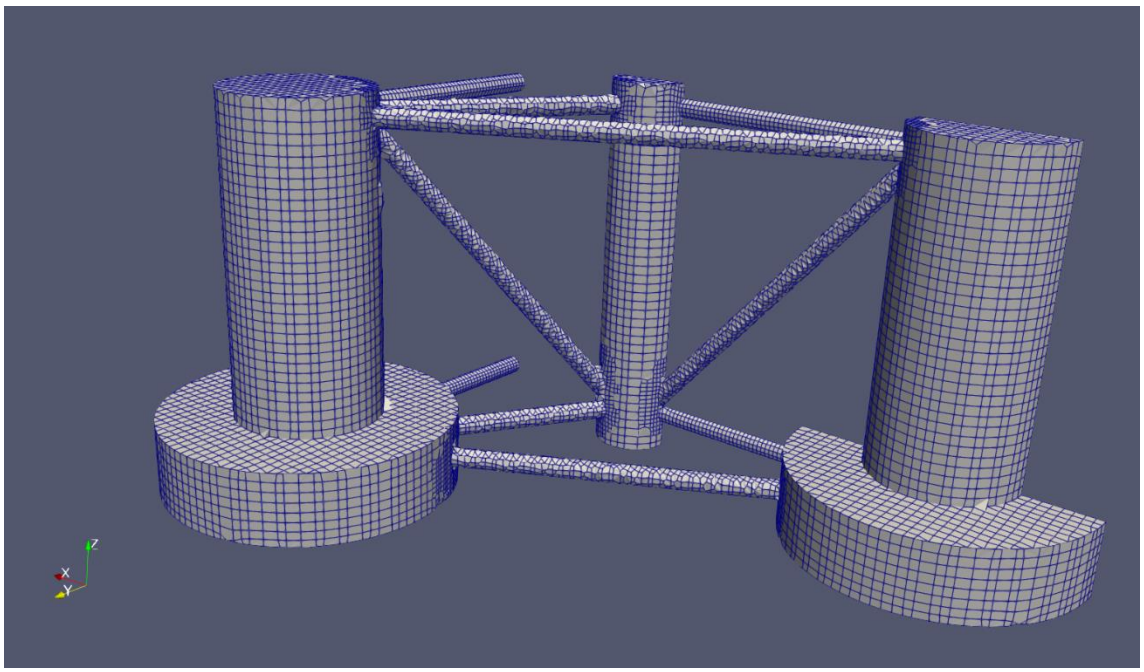


Figure 37: Mesh detail of the FOWT surface

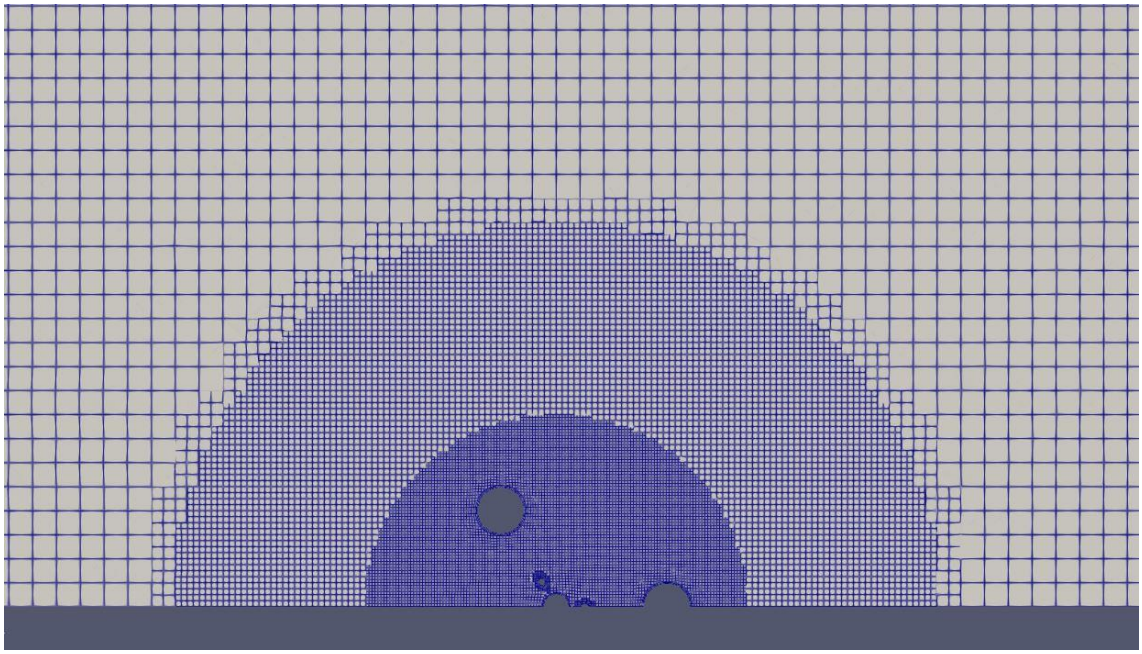


Figure 38: Top view of the FOWT and its surroundings

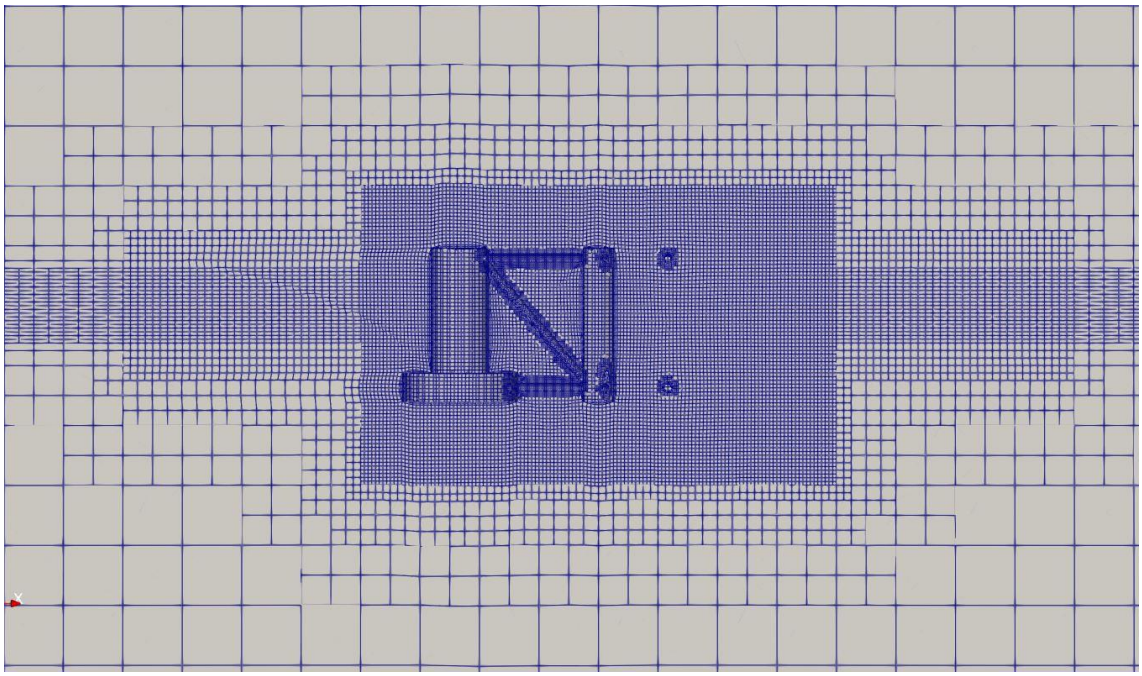


Figure 39: Side view of the FOWT and its surroundings

Appendix E: Introduction to the OpenFOAM interface

OpenFOAM is an open-source suite intended for Field Operation And Manipulation (hence FOAM). It is primarily focused in CFD simulations, but it also offers tools to perform thermal, mechanical, and electromagnetic field computations. It is written in C++ and based in the Finite Volume Method. OpenFOAM is a powerful tool, and it is gaining traction within the research community and some companies due to its open-source nature and its capability to be adapted for tailor-made applications and external tools. For this project, it was chosen as it allowed to use the Moody mooring tool without a need for further code modification.

The OpenFOAM environment is not immediately intuitive for the new user. A brief discussion of its structure and peculiarities is beneficial to understand better the process followed to set up the cases.

An OpenFOAM case is a collection of folders that contain files with instructions on how to simulate it. The case is simulated by executing certain commands in a Linux console when browsing the directory that contains the case instruction, or via a script. The case directory takes the form showed in Figure 40:

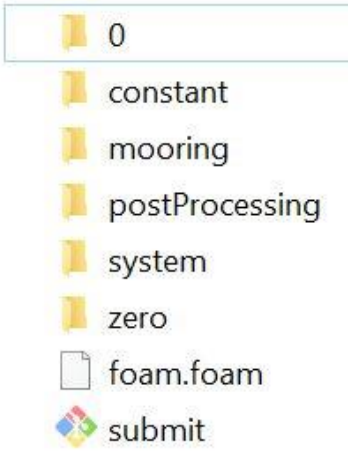


Figure 40: General file structure of and OpenFOAM case

General view of an OpenFOAM case

Where:

- The **0** folder includes the initial fields and boundary conditions. Successive numerical folders include the output fields from the model, the number being the time stamp at which they are written out.
- The **constant** folder include the mesh and the rigid body motion solver settings, as well as the properties that do not vary over time (such as density, viscosity, etc.).
- The **mooring** folder is generated because of the coupling with the external tool Moody. It includes the mooring settings and results.

- The **postProcessing** folder includes the results of the simulation that are not fields, such as forces, moments, residuals, y+, wave heights, etc.
- The **system** folder includes control for the meshing process, the simulation itself and the process of decomposition and recompositing for parallel computing.
- The **zero** folder is just a backup of 0 that does not get changed during the simulation, to reset the simulation more easily.
- The **foam.foam** file allows to open the simulation with the visualization program Paraview.
- The **submit** script is a program that allows to run the case in the cluster

In each of these folders, several text files exist. These control each individual parameter of the model. Some relevant parameters for this simulation are discussed within the Metodology section of this thesis.

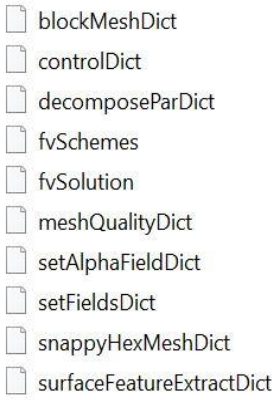


Figure 41: System folder

Each of this text files can be opened with a text editor and modified:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

field alpha.water;
type plane;
origin (0 0 0.4);
//direction (0 0 1);
normal (0 0 -1);

// ************************************************************************* //
```

Figure 42: Code of an OpenFOAM case file