A Comparative Computational Fluid Dynamics (CFD) Study of Wave-induced Motions and Loads on Floating Offshore Wind Turbine Platforms with Moonpools

Master thesis in collaboration with
the Department of Wind Energy of the
University of Rostock during the Erasmus+ mobility

Department of Industrial Engineering
Master degree course in Energy Engineering
Università degli Studi di Padova

Student
Marco Favatà
ID number
2054153

Research Supervisors
Prof.ssa **Anna Stoppato**
Prof.Dott.rer.nat.habil. **Uwe Ritschel**
Research Co-supervisor
M.Sc. **Qian Gao**

April 2024

Abstract

The objective of this master's thesis is to analyze the impact of the moonpool design on wave-induced motions and loads experienced by floating offshore wind turbine platforms through high-fidelity CFD simulations. This study aims to compare the behavior of three different 3D CAD platforms, at which, two of them present the moonpool. The verification and validation of the CFD model was based on the OC5 Semi-submersible platform. By examining different moonpool shapes, the research seeks to gain insights into their effects on the platform's dynamic response, including horizontal surge, vertical heave, rotational pitch, and wave-induced loads. Subsequently, the focus shift on the effect of the external force to the structure stability and how it respond to the change of the wave characteristics.

L'obiettivo di questa tesi è quello di analizzare l'impatto del design moonpool sui movimenti e sui carichi indotti dalle onde sperimentati dalle piattaforme galleggianti di turbine eoliche offshore attraverso simulazioni CFD ad alta fedeltà. Questo studio mira a confrontare il comportamento di tre diverse piattaforme CAD 3D, nel quale due di esse presentano il moonpool. La verifica e la validazione del modello CFD è basata sulla piattaforma semisommergibile OC5. Esaminando diverse forme di moonpool vogliamo ottenere informazioni l'effetto sulla risposta dinamica della piattaforma, tra cui l'impennata orizzontale, il sollevamento verticale, il beccheggio rotazionale e i carichi indotti dalle onde. Successivamente, l'attenzione si sposta sull'effetto della forza esterna sulla stabilità della struttura e su come questa risponde al cambiamento delle caratteristiche dell'onda.

Acknowledgments

Ein herzliches Dankeschön an die Universität Rostock für die Aufnahme, die Aufnahme und die Möglichkeit, die Räume des Fachbereichs zu nutzen, um die Abschlussarbeit so effizient wie möglich zu entwickeln. Besonderer Dank geht an Prof. Dr. rer. NAT. habil. Uwe Ritschel für den Vorschlag eines so faszinierenden Themas und M.Sc. Qian Gao dafür, dass er mich während der gesamten Entwicklung der Arbeit begleitet hat, auch in den kritischsten Phasen der Simulationsphasen des Clusters. Zu guter Letzt meinen Bürokollegen M.Sc. Robert Wustmann und Paul Neumann-Drews, die mir geholfen haben, durch ihre Zeugnisse mehr über Deutschland zu erfahren.

Ringrazio l'Università degli Studi di Padova e la professoressa Anna Stoppato per avermi seguito durante tutto il programma Erasmus e per avermi permesso di sviluppare lo sfidante argomento degli aerogeneratori offshore.

Un doveroso ringraziamento va soprattutto ai miei genitori, Lorella e Giovanni, che hanno lavorato sodo per sostenere economicamente tutto il mio periodo universitario ed a mia sorella Linda, che è stata una roccia di supporto sin dall'infanzia. Ringrazio Leonardo, per essere stato un amico leale con cui confrontarsi durante tutti questi anni.
Infine, a Federica che è sempre stata al mio fianco in questi anni, specialmente nei periodi più difficili. A lei, auguro di seguire i propri sogni con perseveranza ed umiltà, proprio come ha sempre fatto.

Contents

List of Figures

Chapter 1

Introduction

According to the World Energy Outlook 2023 WEO2023 document (see WEO2023), "The macro-economic mood is downbeat, with stubborn inflation, higher borrowing costs and elevated debt levels. Today, the global average surface temperature is already around 1.2 °C above pre-industrial levels, prompting heatwaves and other extreme weather events, and greenhouse gas emissions have not yet peaked. The energy sector is also the primary cause of the polluted air that more than 90% of the world's population is forced to breathe, linked to more than 6 million premature deaths a year". Many countries and an increasing number of businesses are committed to reaching net

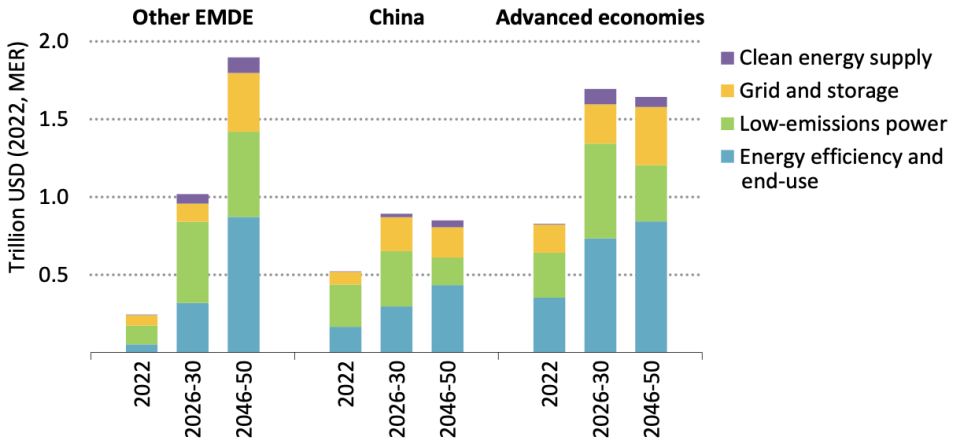


Figure 1.1: The bulk of increased investment in clean energy is needed in emerging economies other than China; it rises more than sevenfold in the second-half of the 2040s relative to 2022. From WEO 2023

zero emissions. As of September 2023, net zero emissions pledges cover more than 85% of global energy-related emissions and nearly 90% of global GDP. Ninety-three countries and the European Union have pledged to meet a net zero emissions target. Moreover, governments around the world, especially in advanced economies, have responded to the pandemic and the global energy crisis by putting forward new measures designed to promote the uptake of renewables, electric cars, heat

pumps, energy efficiency and other clean energy technologies. EV targets have driven a major transformation in the industrial strategies of car and truck manufacturers in recent years, together with fuel economy and CO2 emissions standards in the European Union and China, and more recently in the United States.

Solar PV and wind are set to dominate power capacity additions. Globally, they account for over 70% of total capacity additions between now and 2050 in all WEO scenarios. They do so because they are now the cheapest new sources of electricity in most markets, widely available and enjoy policy support in over 140 countries; Offshore installations account for just 15% of wind capacity additions today, but their share roughly doubles by 2030 across all scenarios. Offshore wind energy generation has experienced an important growth in the last five years, reaching currently an installed capacity of almost 8000 MW. The most important references in this type of technology are Denmark, Holland, Sweden, Germany and United Kingdom. In Spain, previsions for technology, have been set at around 750 MW of installed power during the period 2011–2020 [5]. The increases in spending required to meet climate goals appear challenging but within reach

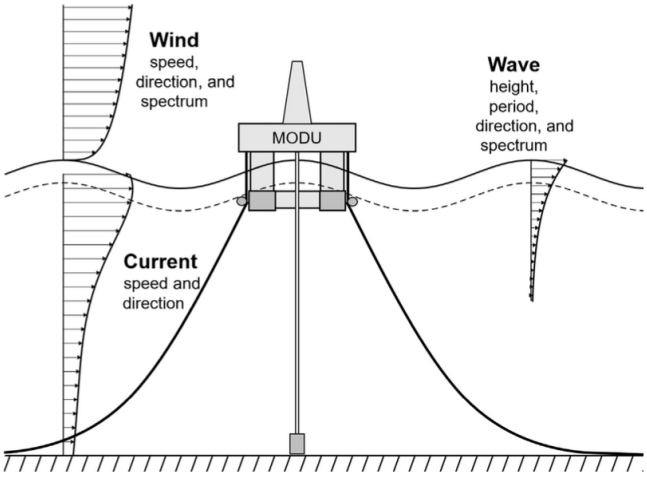


Figure 1.2: Loads acting on the structure, from [20]

for advanced economies and China. Finance is available for clean energy projects, and many of the main risks, including those related to permitting, now appear to be on the policy and regulatory side. The first turbines installed in marine environments were similar to the onshore ones. However, marine environment characteristics have promoted the development of offshore technologies with different features since ocean surface softness generates a low superficial roughness and, as a consequence, a low turbulence intensity. Wind velocity is higher at sea than on land at the same altitudes, so shorter towers can be installed and smaller fatigue damage will be caused [4]. Additionally, the higher offshore wind potential and the availability of larger areas have motivated the development of powerful turbines.

Which are the main logistical challenges in offshore wind supply chains? As the offshore wind farm supply chain lead firm, these firms maintain overall project management and financial man-

agement functions for the duration of the entire wind farm life-cycle which can be very challenging, especially for small-medium businesses. A wind farm life-cycle can generically be split into four key phases [27]:

1. Development & consent
2. Installation & commissioning
3. Operations & maintenance (O&M)
4. Decommissioning

In the development & consent phase, special geophysical, geotechnical, ornithological/mammal, and other survey vessels enable different surveys to be carried out as part of the site planning efforts.

The installation of offshore Balance of Plant components such as cables, offshore sub-stations, and foundations may happen with different supply chain constituencies acting as lead supply chain firm for different parts of the process such as the export cable, offshore sub-station, array cables, offshore accommodation solutions, wind turbine foundations, and finally wind turbine erection/ installation/commissioning. When unpredicted breakdowns to individual wind turbines occur, unscheduled maintenance is needed and this maintenance is more expensive and also more logistically challenging. This requires a different and very flexible logistical response where the break-down is first diagnosed and then repaired. An unexpected stoppage of the entire offshore wind farm due to e.g. a broken cable or a malfunctioning substation is the worst challenge of an offshore wind farm operator. Each producing country present different problems (related to the each weather conditions and financial structure of the country) and so different approaches to the investment's risk. In particular, for the floating off shore wind turbines (FOWT), a very clear picture of the complexity of the study and design of this type of technology is show in Figure 1.8.

Floating platforms for wind turbines are still at an early stage of development, and there are a wide range of platform designs; the choice depends mainly on the site of installation, structural equilibrium and, of course, economical evaluation regarding the capital investment. The desire to install wind turbines in more and more extreme conditions has led to an increase of the complexity of the technology in order to decrease the capital costs maintaining a constant power production. They can be classified based on the primary approach adopted to achieved the static stability in roll and pitch:

Spar-buoy. It consist of a large, vertical and cylindrical hull that provides stability to the structure by having the center of gravity positioned in a lower position respect to the center of buoyancy with the presence of a ballast, in the lower part of the hull cylinder, and the mooring chains. Compared to the other platforms, they can be more cost-effective for deep sea condition applications.

Tension Leg Platform (TLP). TLP's are made by a set of vertical tendons, attached to a semi-submersible structure, which provide stability to the structure from the counteraction of both wave and wind forces. They are suitable for a quite large range of water depths making them versatile for different locations, knowing that, compare to the other types of platform, generally the have a lower carbon footprint. The installation and connection of those tendons to the seabed is usually very complex (requiring specialized equipment, like vessels), because the mooring lines has

to ensure a sufficiently high tension but, at the same time, allowing flexibility. PelaStar, designed by Glosten company (see pelastar.com), is a very used design of this type of structure.

Barge-based system. It's a sea structure built on a relative simple design, making it more cost-effective compare to more complex platforms, with the ability of using ballast tanks to control the stability of the solid in different wave conditions. The design may face some challenges in extreme weather and/or very harsh sea conditions, but the absence of interfaces and the flat shape of the barge, reduces the overall draft.

Semi-submersible. This particular type of structure, is made by vertically and partially submerged columns anchored to the seabed with a catenary mooring system. The partially submerged composition helps to reduce the motions (surge, heave, pitch etc..) caused by the sea waves but the design can be more complex respect to the previous types, with a challenging installation. Usually, extra ballast (heavy plates) will be added to the cylinder base to compensate the turbine height and reducing wave excitation at low frequency.

A quantitative comparison between the first three types of platforms, that have been described

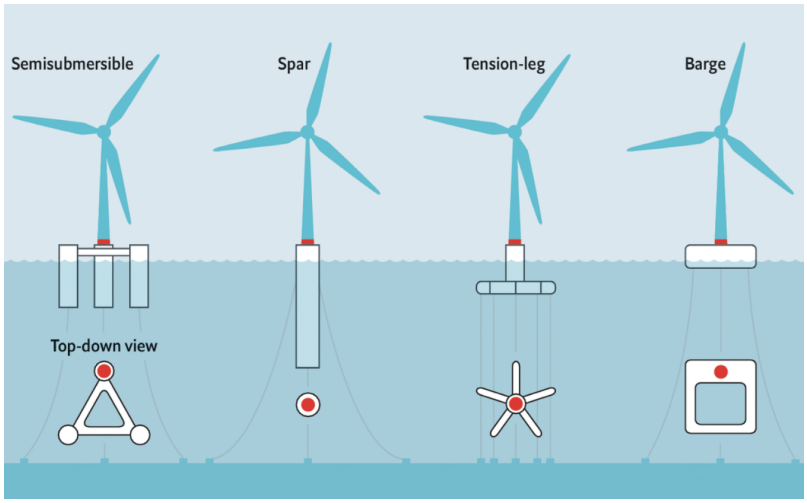


Figure 1.3: Classification of platforms for FOWT. Reproduced from [22]

previously, was made by J.Jonkman and D.Matha (2009, [16]) analyzed, using the aero-hydro-servo-elastic design code FAST with AeroDyn and HydroDyn. In this CFD simulations, the OC3-Hywind spar buoy, ITI Energy barge and MIT/REL TLP platforms with a NREL offshore 5-MW baseline wind turbine were analyzed, using the IEC 61400-3 offshore wind turbine standard as a guide.

The loads group bending moments in the wind turbine's component, from the DLCs extreme-event tables were compared based on the "Ratio of Sea to Land" (i.e. the ratio between the off shore wind turbine and the land based wind turbines): The Figure 1.4 shows two main things:

1. The barge-type platform, respect to the other ones, will increase the bending moment in the structural elements of the wind turbine.

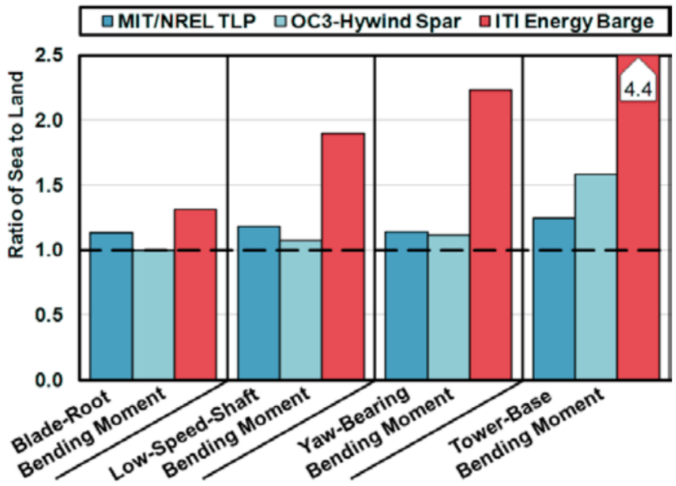


Figure 1.4: Loads comparison between FOWT. Reproduced from [16]

2. Sea based wind generators will be mainly subjected to more stress (related to the bending moments generation) compared to the onshore ones.

About the semi-submersible structure, considerations will be made it in the next lines.
Floating offshore wind turbines (FOWTs) have the potential to access wind resources deep water, which is so far prohibitive for conventional approaches. This, however, comes at a cost: the platform's extra degrees of freedom introduce complex aerodynamic and hydrodynamic behaviours, which need to be accurately modelled to reduce load uncertainties that ultimately prejudice the economic viability of FOWTs. Moored offshore structures in open water are exposed to time-varying environmental loads, such as wind, waves, and currents as we can see in 1.2. How the movement

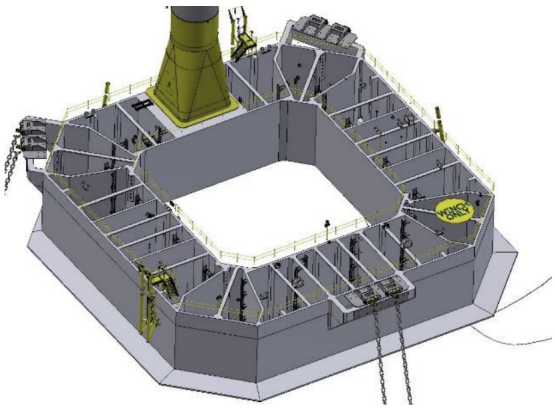


Figure 1.5: Barge type structure, from [3]

effects of platform on the FOWT are investigated?

Unlike its fixed counterpart, a floating wind turbine must be supported by a floating platform which, however, further complicates the design process. The upper turbine and the lower supporting platform are coupled/integrated in one way or another. It has be seen that from the perspective of a floating structure in reality, among all the six degrees of freedom of motion responses, surge, heave and pitch are usually present at the same time and, regarding the wind airfoil, various motion periods has to be tested together with aerodynamic thrust and torque of the blades (see Figure 1.6). According to the "Investigation of the effects of platform motion on the aerodynamics of a floating offshore wind turbine" [19], it was found that both thrust and torque are largely influenced by the prescribed platform motion, indicating that the motion response of the supporting platform for a floating wind turbine should be taken into account during the design procedure. Regarding the

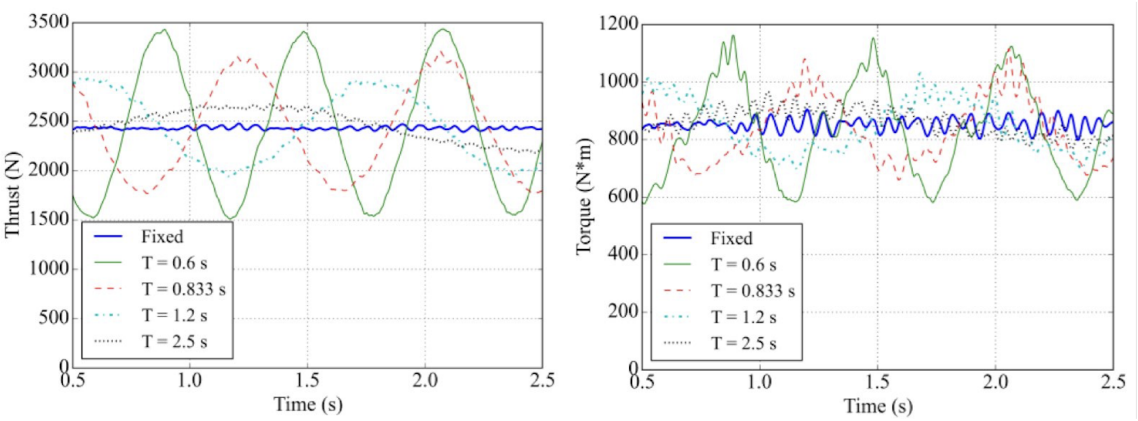


Figure 1.6: Thrust and torque of a NREL S809 airfoil FOWT [19]

heave motion, the article [36] from Krish Thiagarajan and Javier Moreno, by a repeatability experimental test using probes, calculate the phase between the wave and the heave plate signals using a frequency analysis (see Figure 1.7). The high fidelity computational fluid dynamic (CFD) models have been performed to predict the motions and loads of FOWTs. Numerous CFD calculations have been performed by a lot of researchers, both from the oil field (such as [20]) and from the wind turbine field, including decay motions simulations [15] and fully coupled aero-hydrodynamic analysis [11]. The type of modelling approaches used to study FOWTs can be categorised under coupled or uncoupled approaches. This emphasis arises from the highly coupled physical behaviour exhibited by the actual system. While in general, coupled approaches intrinsically include multi-physics modelling of the aero-servo-hydro-elastic system, the increased computational expense generally limits the adoption of advanced numerical tools.

Scaled model tests can provide a good understanding on the behavior of floating platforms at various stages of design and development. This reduces risks and helps to optimize the design of the prototype platform. Since the platform study and design is the most important part of the off shore wind power plant, industries are searching the best platforms both from the engineering, economic and environmental point of view by studying their behavior related to particular sea con-

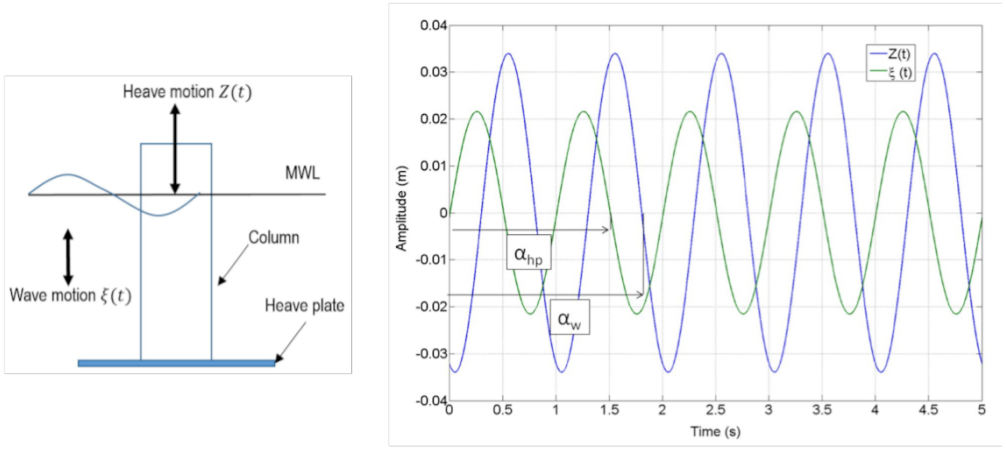


Figure 1.7: Wave Induced Effects on the Hydrodynamic Coefficients of an Oscillating Heave Plate in Offshore Wind Turbines [36].

ditions and testing new geometrical features [35] with the use of different structural materials [3].

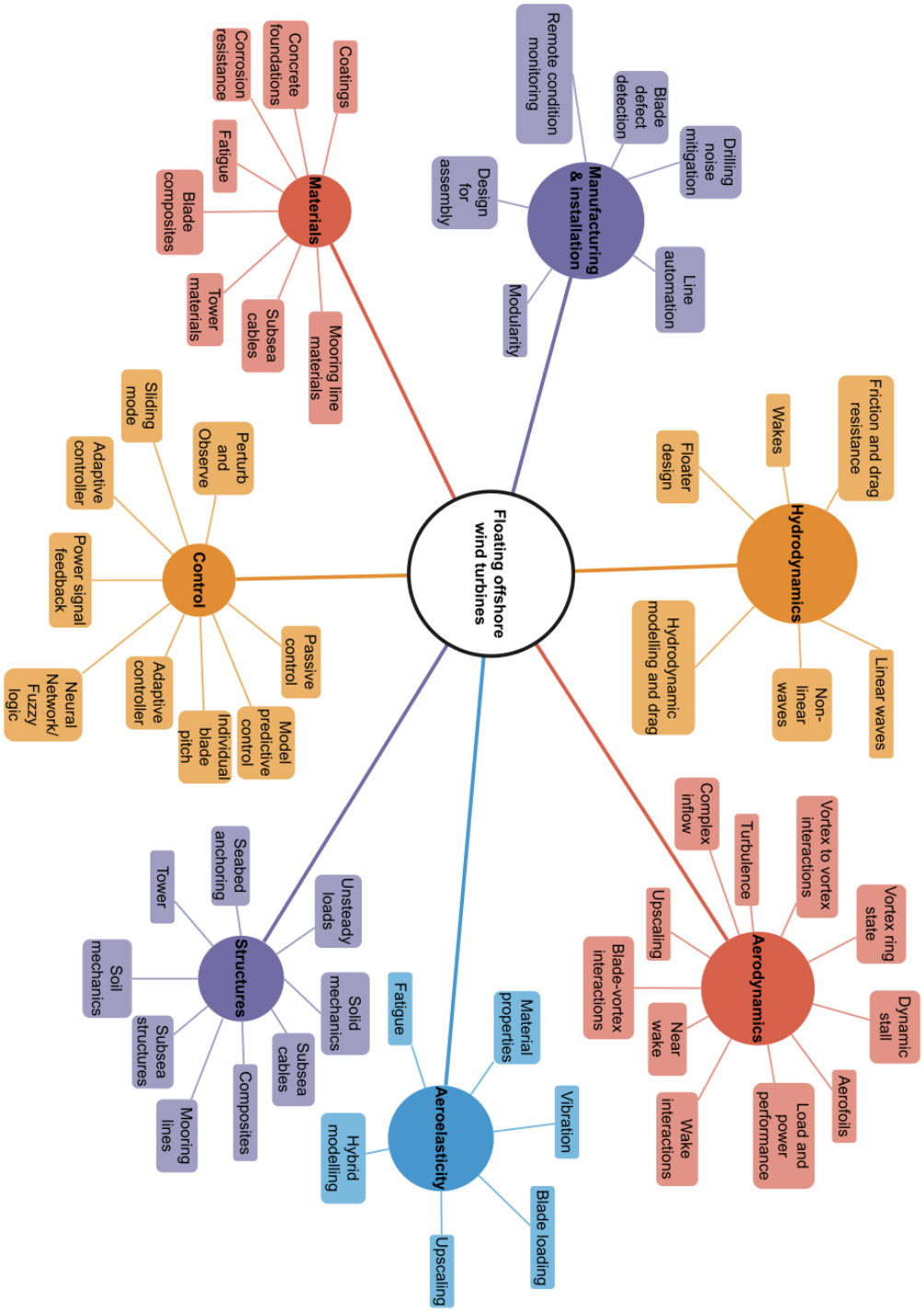


Figure 1.8: Research areas surrounding FOWT aerodynamics [23]

Chapter 2

Wave Theory

The aim of this chapter is to briefly summarize the book of [25] which explain in a very detailed way the theory of the water waves, defined as coastal hydrodynamics. After the description of the physical laws regarding the dynamic of the fluid, as we saw in the chapter (3), it's easily understandable how the Euler equation is mathematically modelled in order to approximate the waves in deep water conditions. The sea waves are defined as periodic undulations of the sea surface, the ultimate state of the wave growth depends on the wind conditions:

i. $\boldsymbol{F}$: fetch is the distance at which the wind blows

ii. $\boldsymbol{V}$: velocity of the wind

iii. $\boldsymbol{t}$: duration of the wind blow

so we understand that a correct wind intensity and direction frequency (represented by a wind rose) evaluation is necessary to correctly predict the ocean behavior. Ocean waves are random in nature and they can be studied by a superimposition of a discrete number of known sinusoidal waves each with different amplitudes and frequencies.

The main parameters (see Figure 2.3) that define the sinusoidal wave are wave period (T_w), water depth (h) and wave height (H_w); other parameters like wave length (L_w or λ) and/or the wave amplitude (a) can be derived using formulas that depend on the water's condition.

We can classify the waves according to the *apparent shape* (progressive waves and standing waves, [40]), to the *origin* (capillary waves, gravity waves, long period waves etc..) and according to the *relative water depth* (small amplitude waves and finite amplitude waves); regarding the latter, the finite amplitude waves are subdivided into *intermediate depth waves* (or Stokes waves), at which $0.05 \leq \frac{h}{\lambda} \leq 0.5$, and *shallow water waves* if $\frac{h}{\lambda} > 0.5$. Small amplitude wave theory and finite amplitude wave theories can be used to introduce the velocity potential $\phi(x, z, t)$, assuming that:

- Fluid is ideal and the flow irrotational
- Surface tension is neglected with a constant pressure uniform at the free surface
- Wave height is small compared to its length

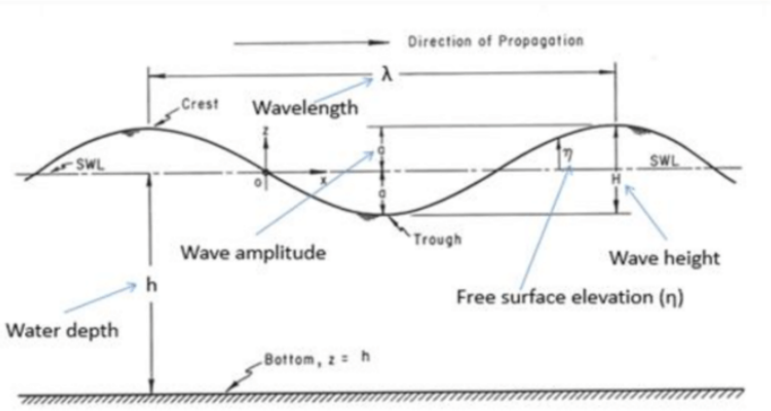


Figure 2.1: Parameters of a sinusoidal wave

From the Laplace equation $\nabla^2\phi = 0$ and the continuity equation (see 3.24 and the Bernoulli equation (which they are written as $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$ and $-\frac{\partial \phi}{\partial t} + \frac{1}{2}(u^2 + v^2 + w^2) + \frac{p}{\rho} + gz = 0$), using the **kinetic bottom boundary condition**[1] and the **dynamic free surface boundary condition**[2], we can derive:

$$\phi = \frac{ag}{\sigma} \cdot \cosh(k(h+z)) \cdot \frac{1}{\cosh(kd)} \cdot \cos(kx - \sigma t) \tag{2.1}$$

where $(kx - \sigma t)$ is a constant value, $k = \frac{2\pi}{\lambda}$ is the wave vector and the coefficient σ, usually defined as wave angular frequency, is computed using the **dispersion formula** from the linear theory:

$$\sigma^2 = gk \cdot \tanh(kh) \tag{2.2}$$

Other theories such as Hunt (1979, [14]), Fenten & Mckee (1990, [10]), Guo (2002, [13]) etc.. can be used too; from the velocity potential the wave elevation η in the x direction can be evaluated as:

$$\eta(x,t) = a\sin(kx - \sigma t) \tag{2.3}$$

This last equation and the linearized Bernoulli's equation are useful to understand the total pressure distribution, which is determine in this way:

$$p = \frac{\gamma H_w}{2} \cdot \frac{\cosh(k(h+z))}{\cosh(kh)} \cdot \sin(kx - \sigma t - \gamma z) \tag{2.4}$$

where γ is the specific weight of the fluid. This field distribution on x and z will be very useful to understand how the platform is subjected to the pressure forces below the mean sea level, which will follow an hyperbolic profile (as for the pressure). We can, also, use the equation 2.4 to evaluate the wave elevation. How can we calculate η for a group of waves? In the case of 2 waves with

[1] The Kinetic bottom boundary condition defines the vertical velocity component as null at the bottom of the sea. Typically this condition is specifically used to describe the behavior of the particles near the seabed.

[2] This other boundary condition, which instead takes into account the fluid flow behavior in the free water level area, is defined as $\eta = \frac{1}{g}\left[\frac{\partial \phi}{\partial t}\right]_{z=\eta}$.

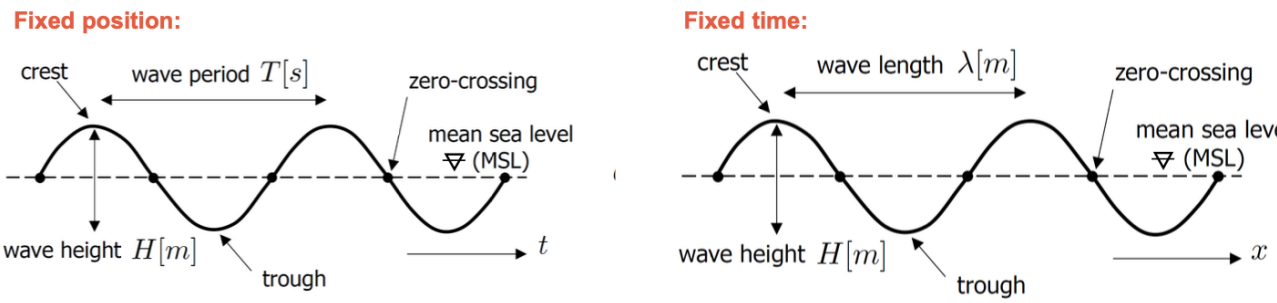


Figure 2.2: Zero-crossing points definition. Reproduced from [11]

$T_{w1} \neq T_{w2}$ and $H_{w1} \neq H_{w2}$:

$$\eta = \eta_1 + \eta_2 = a\sin(k_1 x - \sigma_1 t) + a\sin(k_2 x - \sigma_2 t) \tag{2.5}$$

the previous formula can be used also to define the position (on the x direction) of the zero-crossing points (or nodes) of the wave's group (i.e. points at which $\eta = 0$):

$$X_{\text{node}} = \frac{(2n+1)\pi}{k_1 - k_2} + \frac{(\sigma_1 - \sigma_2)t}{k_1 - k_2} \tag{2.6}$$

where $n = \frac{1}{2}\left(1 + \frac{2kh}{\sin(2kh)}\right)$ is the number of waves in a train. The distance between the nodes is computed as:

$$x = \left|\frac{\lambda_1 \cdot \lambda_2}{\lambda_1 - \lambda_2}\right|. \tag{2.7}$$

From 2.2, the celerity of the waves C can be computed based on the type of water condition:

$$C = \sqrt{\frac{g \cdot L_W}{2\pi} \cdot \tanh(kh)}$$

$$\text{where:} \quad \tanh(kh) \begin{cases} -1 & \text{for deep water} \\ \tanh\left(\frac{2\pi h}{L_W}\right) & \text{for intermediate water} \\ \frac{2\pi h}{L_W} & \text{for shallow water} \end{cases} \tag{2.8}$$

so, we understand that, based on the water's conditions, we can calculate in a very simple way the celerity. The last picture give also, not only the representation of the different water zone but, us the idea that the fluid's particle movement is different, and it's based on the different conditions of the sea; each particle will follow a circular path in the deep water zone and a more elliptical path as we get closer to the coast. These are called **progressive waves**, and, knowing the horizontal velocity u and the vertical velocity w of the particles, the displacement in x and z directions can

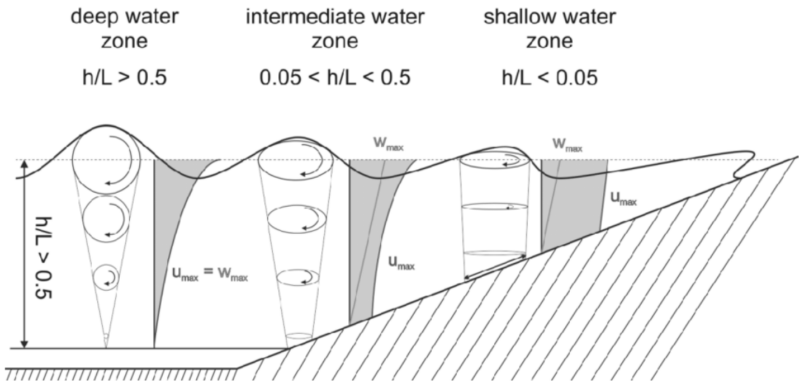


Figure 2.3: Water's condition. Reproduced from [17]

be computed as:

$$\delta_x = \int_t u\,dt = \frac{H}{2} \cdot \frac{\cosh(k(h+z))}{\sinh(kh)}$$
$$\delta_z = \int_t w\,dt = \frac{H}{2} \cdot \frac{\cosh(k(h+z))}{\sinh(kh)} \cdot \sin(kx - \sigma t) \tag{2.9}$$

from here, the local orbit equations are defined for both water's conditions:

$$\text{for shallow water} \rightarrow \left(\frac{\delta_x}{\frac{H}{2} \cdot \frac{1}{kh}}\right)^2 + \left(\frac{\delta_z}{\frac{H}{2} \cdot \frac{h+z}{h}}\right)^2 = 1 \tag{2.10}$$

$$\text{for deep water} \rightarrow \left(\frac{\delta_x}{\frac{H}{2} \cdot e^{kz}}\right)^2 + \left(\frac{\delta_z}{\frac{H}{2} \cdot e^{kz}}\right)^2 = 1. \tag{2.11}$$

2.1 *Comparison between wave theories*

As we said before, the main theories are the **small amplitude wave theory** (Airy, 1845 [6]) and the **Finite amplitude wave theories**, the latter ones can be classified in *Stokes wave*, *Solitary wave* and *Cnoidal wave* (see Figure 2.4). Airy's wave theory is based on the premises that motions are sufficiently small to allow the free surface boundary condition to be linearized (i.e. second and higher order terms of the wave amplitude can be neglected).

- **Stokes wave**: They referred to the wave's theory, developed by Stokes (1847, [28]) by an higher order of the sinusoidal and irrotational waves. For deep water condition, we can write:

$$\eta = \frac{H_0}{2}\cos\left(\frac{2\pi x}{L_0} - \frac{2\pi t}{T}\right) + \frac{\pi H_0^2}{4L_0}\cos\left(\frac{4\pi x}{L_0} - \frac{4\pi t}{T}\right) \tag{2.12}$$

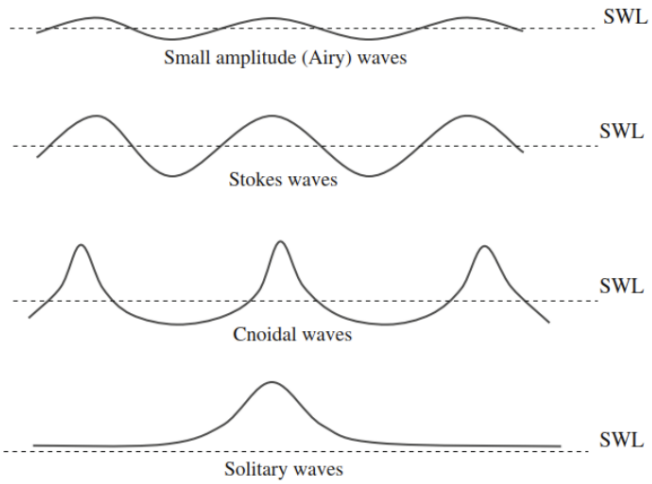


Figure 2.4: Wave theories representation

The 1st order Stokes wave theory is a linear theory that provides the solution of the linearized form of the Navier-Stokes equations, while higher orders take into account additional non-linear effects (such as wave-wave interactions, wave-wave modulation and higher harmonics) providing a more accurate representation of the real wave behavior and this was neglected in the previous case.

- **Solitary wave**: Very shallow water wave crests become peaked and trough flattened (the entire surface profile is placed above the SWL), so the wave will be non-periodic with a indefinite wave length; in deep water, we refereed them as tsunamis. Boussinesq (1872, [24]) derived, from the general equation of the steady flow, the characteristics of this type of wave in shallow water condition.

- **Cnoidal wave theory**: This is well described by Korteweg-de and Vries (1895), applicable for $\frac{1}{30} < \frac{h}{L_W} < \frac{1}{10}$. These types of waves exhibit amplitude modulation, meaning that the amplitude value varies periodically along the direction of propagation. While the stokes wave theory is suitable for small amplitudes, this theory provides a more accurate representation of larger amplitudes (which comprehend even the solitary waves).

Using the LeMehaute chart (1976, [18]), based on the wave's features, it'll be very easy to understand which theoretical case will better suit our situation.

Based on the characteristic of the wave used in this thesis, Stokes II model has been chosen.

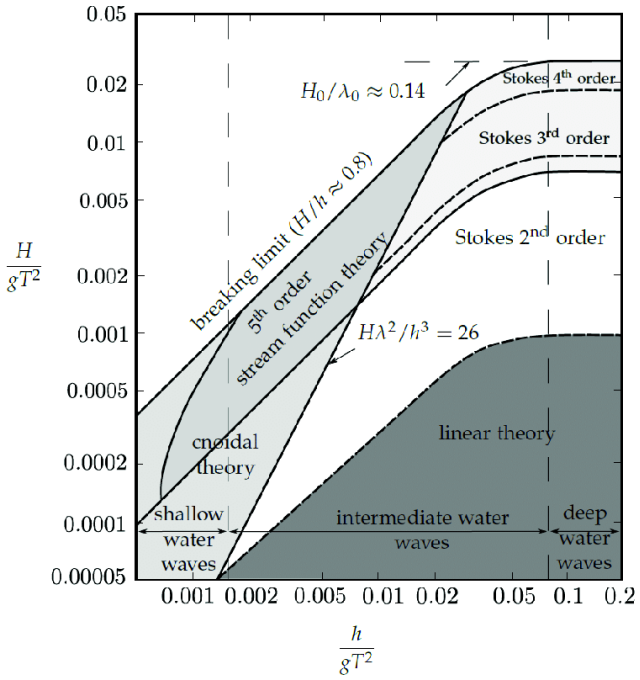


Figure 2.5: LeMehaute wave's theory chart. Reproduced from [18]

2.2 *Stokes II wave model*

Stokes II waves take into account higher-order corrections to the original Stokes wave theory, which improves its accuracy in describing real ocean waves, especially in cases where wave steepness is moderate to large. These corrections account for the effects of wave non linearity, which become significant as waves become steeper. In the following, we will be limited to the second-order approximation known as the second-order Stokes theory, so the full boundary value problem is formulated as follows [33]:

$$\phi_{xx}^{(2)} + \phi_{yy}^{(2)} + \phi_{zz}^{(2)} = 0 \quad \text{if} \quad 0 < z < h \tag{2.13}$$

$$\phi_t^{(2)} + \frac{1}{2}\left(\phi_x^{(1)}\right)^2 + \left(\phi_y^{(1)}\right)^2 + \left(\phi_z^{(1)}\right)^2 + \eta^{(2)} g + \eta^{(1)}\phi_{tz}^{(1)} = 0 \quad \text{a} \quad z = 0 \tag{2.14}$$

$$\eta_t^{(2)} + \eta_x^{(1)}\phi_x^{(1)} + \eta_y^{(1)}\phi_y^{(1)} - \phi_z^{(2)} - \eta^{(1)}\phi_{zz}^{(1)} = 0 \quad \text{a} \quad z = 0 \tag{2.15}$$

$$\phi_z^{(2)} = 0 \quad \text{a} \quad z = -h \tag{2.16}$$

Many research works, in the field of the study of the second-order Stokes theory, were developed. The nonlinear free surface boundary conditions are satisfied based on the perturbation method up to the second-order. The first and second- order velocity potential problems at each time step are solved through a finite element method (FEM) [38].

The "Numerical simulation of the second-order Stokes theory using finite difference method" from

MA Maâtoug, M Ayadi [21], numerically solve the second-order Stokes theory using the Crank–Nicholson finite difference method in order to simulate the potential flow and the surface elevation, then to deduct the pressure loads without taking into account the phenomena observed during the flow of the fluid (breaking, reflection, compressibility...). The lateral boundary conditions and the initial conditions are taken as follows:

$$\phi_x(0,y,z,t) = \phi_x(L_x,y,z,t) = f_1(y,z,t) \tag{2.17}$$

$$\phi_y(x,0,z,t) = \phi_x(x,L_y,z,t) = f_2(x,z,t) \tag{2.18}$$

$$\eta(0,y,t) = \eta(L_x,y,t) = f_3(y,t) \tag{2.19}$$

$$\eta(x,0,t) = \eta(x,L_y,t) = f_4(x,t) \tag{2.20}$$

$$\phi(x,y,z,0) = \phi_0(x,y,z) \tag{2.21}$$

$$\eta(x,y,0) = \eta_0(x,y) \tag{2.22}$$

To develop computational solutions of the problem, let the unknown functions be $\eta(x,y,t)$ and $\phi(x,y,z,t)$ and their values at the mesh points (x,y,t) and (x,y,z,t) are, respectively, $\eta(i\Delta x, j\Delta y, n\Delta t)$ and $\phi(i\Delta x, j\Delta y, m\Delta z, n\Delta t)$. From the Figure 2.6, the examination of the velocity potential behav-

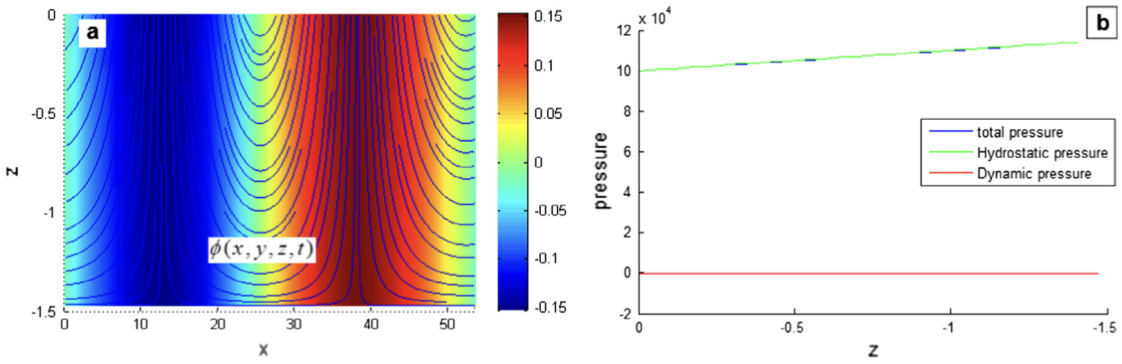


Figure 2.6: (a) Velocity potential and streamlines and (b) Hydrodynamic pressure, for $\frac{h}{gT_w^2} = 0.0015$ and $\frac{H_w}{gT_w^2} = 0.00002$. From [21]

ior, shows that the fluid flow is spread over the entire depth and it is obvious since we are already in shallow water where the particles are active throughout the depth.

For the fluid pressure, it can be observed that the depth insufficient allows the swell to exercise a total pressure equal to the hydrostatic pressure, due to the fact that the vertical accelerations of particles are very weak and because the flow is supposed to be irrotational which neglects the effects of turbulence. As the amplitude grows, the dynamic pressure contributions will increase. The contributions of this type of pressure are spread over the entire depth but they will not have the same effects throughout the depth. The effects of dynamic pressure are more important near the surface and will decrease approaching the bottom (Fig. 6.9b) because of the particles progressive movement at the surface.

The understanding of the pressure distribution was very useful to understand the force distribution on the platform surface.

2.3 *Standing waves*

Since the *background* domain of the simulation is a closed volume, when the wave (which is generated from the inlet surface) moves towards the platform and reaching the outlet surface of the domain, it will be reflected and moved against the next wave (which it has been generated 1.5 seconds after the first one) creating a **standing wave**. This will lead to an higher pressure distribution in the vertical direction with the generation of big forces against the sea structure. To reduce the reflection phenomena we adopted the `shallowWaterAbsorption` wave's model in the outlet surface which it'll act as a damping wave zone. Assuming a perfect reflection from the starting sinusoidal

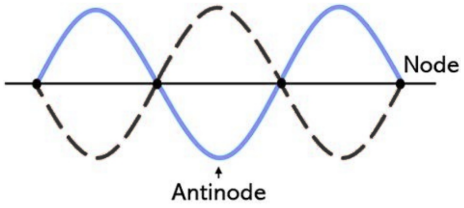


Figure 2.7: Reflection of a sinusoidal wave

wave, as we can see from Figure 2.7, using the Bernoulli equation, the pressure field distribution can be computed as:

$$\begin{aligned} \frac{p}{\rho} + \frac{1}{2}(u^2 + w^2) - \frac{\partial \phi}{\partial t} + gz &= \text{const}(t) \\ &\downarrow \\ p &= -\rho g z + \rho g k_p \eta \end{aligned} \tag{2.23}$$

knowing that k_p (defined as **response pressure factor**) is calculated in the same way as the progressive waves (or starting waves), since in the nodes $\eta = 0$ (the opposite respect to the antinodes); the total pressure will be equal to the hydrostatic one in those points. Integrating the equation 2.23 along the height (denominated as l of the sea structure and fixed to the sea bed and standing perpendicular to it) the force will be evaluated as:

$$F_S = \int_0^l p(z)\, dz = \rho \cdot \frac{gh^2}{2} + \rho g h \cdot \frac{\tanh(kh)}{kh} \cdot l^+ \tag{2.24}$$

where l^+ is the distance from the free water surface to the highest point of the solid and if the depth of the sea increase, F_S will increase too. Based on the velocity potential of both waves, we

can determine the particle's velocity in the x and z directions:

$$u_S = \frac{\pi H_w}{T_w} \cdot \frac{\cosh(k(h+z))}{\sinh(kh)} \cdot \sin(kx)\cos(\sigma t)$$
$$w_S = -\frac{\pi H_w}{T_w} \cdot \frac{\sinh(k(h+z))}{\sinh(kh)} \cdot \cos(kx)\cos(\sigma t). \tag{2.25}$$

Here, it's clearly understandable that both velocities present the same phase in the time trend perspective, but in order to achieve more information, we need to derive these two formulas to calculate the accelerations:

$$\dot{u}_S = a\sigma^2 \cdot \frac{\cosh(k(h+z))}{\sinh(kh)} \cdot \sin(kx)\cos(\sigma t)$$
$$\dot{w}_S = -a\sigma^2 \cdot \frac{\sinh(k(h+z))}{\sinh(kh)} \cdot \cos(kx)\cos(\sigma t) \tag{2.26}$$

Based on the 2nd Newton's law of dynamic, we understand that the force isn't uniform not only in the vertical direction (as we saw before from the pressure calculation) but it's also periodic in both time and direction; the maximum acceleration and so the maximum force in the x direction is present on the nodes, while the z component is maximized on the anti-node points. The SURF SIMILARITY APPROACH by Battjes, J. (1974, [1]), explain that the reflection phenomenon and so the force generating by it, depends also by the type of wall properties (slope, roughness and permeability) at which the wave impacts and from the wave's steepness; so, if the pitch angle of the platform and the wave steepness vary along the time of the simulation, F_S will vary too.

Other possible deformation phenomena are *wave refraction* and *wave diffraction*, the latter one represents the event at which the kinetic energy of the water is transferred laterally along a wave crest due to the presence of solid material in the sea.

In this case, the **circular diffraction** is present, as we can see from the small wave propagation (smaller kinetic energy compare to the incident ones) in the part of the sea immediately after the platform.

2.4 *Wave-body loads*

The interaction of a wave towards a solid structure can be describe by the forces acting on it; the dynamic interaction is the cause of the movements of the platform (surge, heave, pitch, roll etc..). The evaluation of the loads is necessary for the structure design, so in order to understand how our designed solid will respond to each load, we have to take into account:

- → The non-linear water particle displacement.
- → The variability of the wave profiles.

- $\rightarrow$ Turbulence phenomena.
- $\rightarrow$ Modification of the wave properties by the presence of the object (wave scattering) and the possibility of dynamic effects such as vortex shedding and structure resonance.

There are different types of forces that we can calculate from theoretical formulas and/or directly measured using pressure sensors in laboratory[3] and they can be classified in this way:

Drag force. This type of load, also known as **hydrodynamic resistance force**, is primarily related to viscous friction between the solid and the water layers. If this force is very high, it can lead to the variation of the thickness of the boundary layer with the generation of vortex by the dissipation of the kinetic energy. The magnitude of the drag force is related to several factors such as: object's shape, Reynolds number (related to fluid velocity and viscosity) and surface roughness.

Diffraction force. It is mainly related to the disturbance of the body generating scattered waves. The calculation of this type of load is defined by the Helmholtz equation:

$$\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2} + k^2 F(x, y) = 0 \tag{2.27}$$

where $F(x, y)$ is a complex number that contains both amplitude and phase of the load.

Impulse force. This load is about the wave breaking condition, condition at which the horizontal particle velocity at the crest exceed the celerity so the vertical particle's acceleration will be greater than the gravitational one with the increase of the wave's steepness higher than 120°. This aspect is fundamental, especially for the design of shallow water's harbors, and based on the type of breaker we'll have the formation of spilling, plunging and surging waves. Usually called as *resistant slamming force*, it become important when the **Froude number** $Fr = \frac{C}{\sqrt{gh}} > 0.6$

Radiation force. It's related to the motion of the body, where, the waves spread out from it in all directions, giving a rise to damping effects related to the wave period. They're related to the velocity and acceleration of the platform.

How can we quantify the total load at which the solid is subjected? The main ways are the *design wave approach* and the *irregular wave approach*, the latter one will consider a combination of waves, with different period and height, and each of them will generate a force. A more simply way is to use the first approach [43], by simplifying the wave as a sinusoidal one and assuming that all the forces classified before are acting parallel to the flow direction:

$$F_{\text{Morr}} = \rho V \dot{\mathbf{u}} + \rho c_a V(\dot{\mathbf{u}} - \dot{\mathbf{v}}) + \frac{1}{2}\rho c_d S(\mathbf{u} - \mathbf{v}) \left\|\mathbf{u} - \mathbf{v}\right\| \tag{2.28}$$

where V is the submerged volume, S is the reference area, c_d and c_a are the drag and the added mass coefficients (they depends on the flow regime and on the shape of the object), $\mathbf{u}$ and $\mathbf{v}$ are

[3]Basically the wave structures or wave test facilities are of two types: unidirectional and multi-directional. The first one is the most simple one since it can generate only mono directional regular or random waves (the latter one is used especially to test the wave-structure interaction in the worst sea possible conditions), while the second one is able to create 3D waves that can be generated to study the oblique waves interaction with mounted structure which cannot be rotated but it's very expensive.

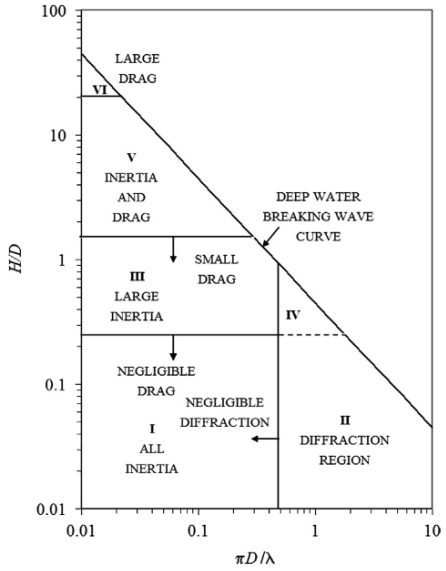


Figure 2.8: Force regime as a function of wave length λ
, characteristic body dimension D and wave height H from [2]

the wave and body velocities. From the linear theory, the diffraction force or Freude-Krylov force for a circular platform, can be wrote as:

$$F_{\text{Diff}} = -2\gamma r H_w \cdot \frac{\sinh(kh)}{k\cosh(kh)} \cdot \sin(kr)\sin(\sigma t) \tag{2.29}$$

where r is the radius of the platform. It's very important to state that, this equation, ignore the diffraction load, so it's used preferably for slender bodies. We understand that, knowing the wave properties and the body characteristics, we can identify the percentage of the drag component in the total force compare to the inertia component, as we can see in Figure 2.8.

The researchers Keulegan and Carpenter (1958) [30], from the study of a oscillating sinusoidal two dimensional flow having a horizontal velocity $u_{\max}\cos(\sigma t)$ applying on cylinders and transverse plates, they found that the c_d varies smoothly based on these two coefficients:

$$KC = \frac{(u_{\max} + U)\cdot T_w}{D} \quad \text{and} \quad R = \frac{(u_{\max} + U)\cdot T_w}{\nu} \tag{2.30}$$

at which, U is the current velocity, $u_{\max}$ is the maximum horizontal particle velocity and ν is the kinematic viscosity of the fluid. The sum of F_S, F_{Morr} and F_{Diff} will determine the total wave force F_e which will be used in the momentum equilibrium (see Chapter 3).

Chapter 3

Fundamental governing equations

This section will be only a brief introduction of the fundamental equations used in the dynamic study of offshore sea structure and ships (see the *Numerical Methods for Seakeeping Problems* handbook by Bettar Ould el Moctar , Thomas E. Schellin , Heinrich Söding [9]).

3.1 *Rigid body dynamics*

Offshore structures are elastic bodies, so their flexibility has a considerable influence on the stresses caused by wave loads. In order to compute these stresses, platform motions are superimposed from rigid body motions and elastic deformations. For a rigid body, we means that the distances between mass particles within the body don't change whatever is the force applied on it.

In order to describe the motion, we need to define two Cartesian right-hand coordinate systems: one inertial Cartesian coordinate system $\mathbf{O}(x, y, z)$ and a Cartesian system fixed to the body $\mathbf{o}(\underline{x}, \underline{y}, \underline{z})$ as we can see in Figure 3.1. The center of gravity of the body can be defined as $\mathbf{x}_G = (x_G, y_G, z_G)^T$. The origin of the body-fixed coordinate system is positioned at its center of mass while the position

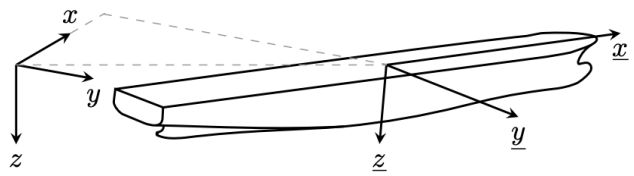


Figure 3.1: Coordinate systems for marine structures. Reproduced from [9]

of the global coordinate system is arbitrary, as for the orientation of the coordinate axes. The ship inclination about the longitudinal axis it's the roll angle ϕ and it's defined positive for a right-hand rotation along rotation of the body around $\underline{x}$ and, correspondingly, the pitch θ and yaw ψ angles describes the rotations along the transverse and vertical axes respectively.

The platform can be described both in translation and/or rotation motion:

Table 3.1: Description of the platform's motion. Reproduced from [9]

Name	Description	Velocity	Position, Euler angle
Surge	Translation along the x-axis	u	x
Sway	Translation along the y-axis	v	y
Heave	Translation along the z-axis	w	z
Roll	Rotation around the x-axis	p	ϕ
Pitch	Rotation around the y-axis	q	θ
Yaw	Rotation around the z-axis	r	ψ

In order represent mathematically the rotation of the object, we need to define the matrix $\mathbf{T}$, which is made by:

$$\mathbf{T}_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \tag{3.1}$$

$$\mathbf{T}_\theta = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \qquad \text{and} \qquad \mathbf{T}_\psi = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

so, we'll have:

$$\mathbf{T} = \mathbf{T}_\psi \mathbf{T}_\theta \mathbf{T}_\phi = \begin{pmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{pmatrix}. \tag{3.3}$$

This matrix is very useful because it permit us to calculate the position of any point respect to the center of mass $\mathbf{x_G}$ from this equation:

$$\mathbf{x} = \mathbf{x_G} + \mathbf{T}\mathbf{x}. \tag{3.4}$$

In the next chapters, we'll see the importance of this matrix to correctly set the constraints of the rigid body inside the program.

3.1.1 *Dynamic equilibrium*

Platform's translations are defined by the second law of Newton, and supposing that the motion of the body is defined based on the center of gravity:

$$m\ddot{\mathbf{x}}_G = \mathbf{F}, \tag{3.5}$$

where m is the platform mass, $\ddot{\mathbf{x}}_G$ as the linear acceleration of the center of gravity and $\mathbf{F}$ as the force acting on it. Rigid body rotations are governed by the equation

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) = \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \mathbf{M} \tag{3.6}$$

where, $\mathbf{M}$ is the external moment about G acting on the structure, $\mathbf{I}\omega$ is the angular momentum of the body and $\mathbf{I}$ is the inertial matrix referring to G. Both $\mathbf{M}$ and $\mathbf{I}$ are expressed in inertial coordinates. Of course, since the inertial matrix describe how the mass is distributed inside the body and how it responds to the forces, it's constant in ship-fixed coordinates if, all items inside the structure (m_i) belonging to it, do not move relatively to the body:

$$\underline{\mathbf{I}} = \begin{pmatrix} \underline{I}_{xx} & -\underline{I}_{xy} & -\underline{I}_{xz} \\ -\underline{I}_{yx} & \underline{I}_{yy} & -\underline{I}_{yz} \\ -\underline{I}_{zx} & -\underline{I}_{zy} & \underline{I}_{zz} \end{pmatrix} = \begin{pmatrix} \sum_i m_i(\underline{y}_i^2 + \underline{z}_i^2) & -\sum_i m_i \underline{x}_i \underline{y}_i & -\sum_i m_i \underline{x}_i \underline{z}_i \\ -\sum_i m_i \underline{y}_i \underline{x}_i & \sum_i m_i(\underline{x}_i^2 + \underline{z}_i^2) & -\sum_i m_i \underline{y}_i \underline{z}_i \\ -\sum_i m_i \underline{z}_i \underline{x}_i & -\sum_i m_i \underline{z}_i \underline{y}_i & \sum_i m_i(\underline{x}_i^2 + \underline{y}_i^2) \end{pmatrix}. \tag{3.7}$$

Since the mass moment of inertia, that we just described previously, depends on the body position, therefore the mass moment of inertia is conventionally computed in the body-fixed coordinate system $\mathbf{o}(\underline{x}, \underline{y}, \underline{z})$ and then transformed to the inertial coordinate system $\mathbf{O}(x, y, z)$ using the transformation matrix $\mathbf{T}$

$$\mathbf{I} = \mathbf{T}\underline{\mathbf{I}}\mathbf{T}^{\intercal}. \tag{3.8}$$

Now we can compute the body movements, which are defined by the linear and angular accelerations $\ddot{\mathbf{x}}_G$ and $\dot{\omega}$, from the forces and the momentum:

$$\begin{aligned} \ddot{\mathbf{x}}_G &= \frac{\mathbf{F}}{m} \\ \dot{\mathbf{\Omega}} &= \mathbf{I}^{-1}(\mathbf{M} - \mathbf{\Omega} \times \mathbf{I}\mathbf{\Omega}). \end{aligned} \tag{3.9}$$

From this formula $\dot{\boldsymbol{\alpha}} = \mathbf{R}^{-1}\boldsymbol{\omega}$, the velocity of the center of gravity ($\dot{\mathbf{x}}_G$) and the time derivative angular velocity or angular acceleration ($\dot{\boldsymbol{\alpha}}$) are numerically integrated to obtain $\mathbf{x}_G$ and $\boldsymbol{\alpha}$. For six-dimensional motion (three components each for the rotation and translation), is defined by a mass matrix $\mathbf{A}$ that depends on the actual immersion in the water and on the body orientation:

$$\begin{pmatrix} \mathbf{F} \\ \mathbf{M} \end{pmatrix} = -\mathbf{A} \begin{pmatrix} \dot{\mathbf{x}}_G \\ \dot{\boldsymbol{\omega}} \end{pmatrix} + \begin{pmatrix} \mathbf{F}_r \\ \mathbf{M}_r \end{pmatrix}, \tag{3.10}$$

where $\mathbf{M_r}$ and $\mathbf{F_r}$ are the rest moment and the rest force. Equation (3.10) can be solved by iteration under-relaxation (using the acceleration from the previous step) but it has to be state that, for light bodies in heavy fluids erratic accelerations and force histories, this can generate instability problems or wrong results; so, it'll need small under-relaxation factors in order to control the velocity of convergence of the loop. This may increase the number of iterations.

Each single rotation and translation movement, based on the variation of the position of the center of mass of the platform over the time, required a lot of time and computational power. So, as has been done in many research articles (which some of them are showed in the bibliography), it's usually need only the surge, heave and pitch values to analyze the cinematic behavior of the platform subjected to external forces and momentum.

Since this thesis is focused only on the platform and total FOWT structure, we didn't consider, in the definition of the rigid body dynamics, the gyroscopic load caused by the presence of the wind turbine. Here, we see how we can set the properties of the rigid body into **dynamicMeshDict** file (placed in the *constant* folder), using the RigidBodyMotion solver:

Listing 3.1: fvSolution.txt code part 1

```
dynamicFvMesh   dynamicOversetFvMesh;
motionSolverLibs (rigidBodyMeshMotion);
motionSolver    rigidBodyMotion;
report          on;

solver
{
    type CrankNicolson;
}

accelerationRelaxation 0.6;

bodies
{
    Offshore
    {
        type            rigidBody;
        parent          root;

        centreOfMass    (0 0 -0.1614);
        mass            115.16267;
        inertia         (44.6304 0 0 49.7664 0 43.8144);
        transform       (1 0 0 0 1 0 0 0 1) (0 0 -0.1614);

        joint
        {
            type    composite;
            joints
            (
                {
                    type Px;
                }
                {
                    type Pz;
                }
                {
                    type Ry;
                }
            );
        }

        patches         (Offshore);
        innerDistance   1000;
        outerDistance   1001;
    } }
```

The calculation of the **I** matrix for complex geometries (such as the "Shape" platform, see Chapter 5) is based on the *parallel axis theorem* since the geometrical center doesn't coincide with the CM and in the case of the cylindrical platform $I_{xx} = I_{yy}$, so:

$$I_{xx} = I_{xx_{\text{in}}} + md^2 \tag{3.11}$$

where d is the distance between the geometrical center of rotation and the center of mass.

3.1.2 *Hydrostatics of a semi-submersible body*

Every solid immersed in a fluid experiences an upthrust equal to the weight of the fluid displaced and in order to reach a dynamic equilibrium, the buoyancy force (which cause this upthrust) has to be equal to the gravity force acting on the center of gravity. A common approach is to study the motion of the body in the static equilibrium, which means that the gravitational and the vertical buoyancy forces are equal.

Using the Archimede's principle, we can compute the buoyancy force (described as a vertical component of the integrated hydro static along the wet surface) acting on a floating platform in a simplified way as:

$$\int_S \mathbf{dF}_h = \sum \mathbf{F}_{\text{ext}}, \quad \int_S \mathbf{r} \times \mathbf{dF}_h = \sum \mathbf{M}_{\text{ext}} \tag{3.12}$$

where:

$$d\mathbf{F}_h = -p_h \mathbf{n}\, dS \tag{3.13}$$

About the equation of equilibrium 3.12, $\mathbf{F}_{ext}$ and $\mathbf{M}_{ext}$ represent the force and the momentum generated by external loads (such as mooring and gravity acceleration) while $\mathbf{F}_h$ and p_h define the force and pressure of buoyancy. It's important to state that the surge, sway and yaw movements don't change the volume of body submerged in the water, vice versa only pitch and roll can change the stability condition of the platform; for this reason it'll be necessary to study only the pitch angle variation along the time. We adopt, in order to not complicate much the calculations of the understanding of the platform's behavior, an orthogonal axis system (at which X's axis orientation is the sea wave's motion direction and the Z orientation is upwards) where the axis Y is defined in a orthogonal direction respect to the plane XZ. A body-attached frame of reference xyz is defined with the origin 0. Both the inertial Cartesian system $\mathbf{O}(x, y, z)$ (in this case XYZ) and the yz body frames coincide at $\theta = 0$.

From the Figure 3.2, we state some geometrical definition in order, to the reader, to understand the next equations:

- O_F : Body origin
- F : Center of flotation
- G or x_G and B are respectively the center of gravity (point of action of the resulting gravity forces) and the center of buoyancy (point of action of the resulting buoyancy forces)

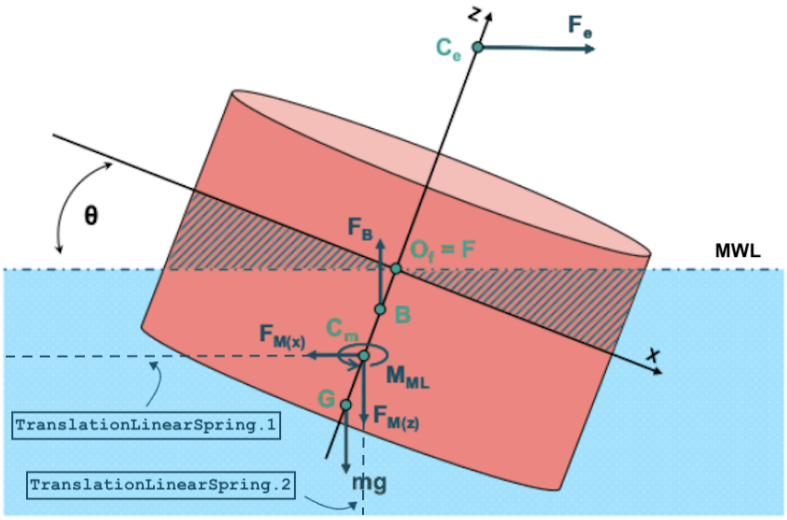


Figure 3.2: Simplified representation of a generic off shore structure

- L or C_m : Mooring point (which is defined in the metacentric height[1], but in our case, for simplicity, it's define in the center of mass G

While an accurate MoorDyN[2] model would be preferable, the coupling process requires study and test, which was not necessary for this thesis since it focus on comparing the effects of the moonpool design, using a spring instead of the mooring model, and by keeping the same spring force in all the moonpool simulations, does not introduce any additional impact on the moonpool structure.

We simply set in the **dynamicMeshDict** file the restraints of the body translation motion with two linear spring anchored to the external surface of the background mesh, in this way:

Listing 3.2: dynamicMeshDict.txt code part 2

```
TranslationLinearSpring.1
    {
        type                    linearSpring;
        refAttachmentPt         (0 0 -0.1614);
        anchor                  (0 0 -5.8);
        stiffness               7.76;    //units of N/m
        damping                 0;
```

[1]This value is defined as the difference between meta center and the gravity points of the body and it's a measurement of the initial stability of the ship; the larger is the metacentric height and the higher is the initial equilibrium of the element. It, also, influence the period of rolling, so it's necessary to choose a correct value of metacentric height in order to ensure comfortable travel condition for the passenger of a ship.

[2]MoorDyN is a simple, efficient and versatile mooring system dynamics model based on a lumped-mass discretization of a mooring line's dynamics, adds point-mass and rigid-body objects to enable simulation of a wide variety of mooring and cable arrangements. The code was initially developed in 2014 as a C++ library that could be easily coupled with other codes.

```
        restLength                          5.6386; //units of m
        body Offshore;
    }

TranslationLinearSpring.2
    {
        type                                linearSpring;
        refAttachmentPt                     (0 0 -0.1614);
        anchor                              (-5.6 0 -0.1614);
        stiffness                           33.84;  //units of N/m
        damping                             0;
        restLength                          5.6; //units of m
        body                                Offshore;
    }

AxialLinearSpring
     {
         type                               linearAxialAngularSpring;
         referenceOrientation               (1 0 0 0 1 0 0 0 1);
         axis                               (0 1 0);
         stiffness                          17.013;   //units of N/m
         damping                            0;
         body                               Offshore;
     }
```

For the calculation of the equilibrium we can define 4 main contributions specific for the FOWT (Floating Offshore Wind Turbines) platform:

i. **Water-plane area** → The hydro-static load, generated by the buoyancy force of the submerged volume, can be written as:

$$dF_h = \rho g dV = \rho g h dS = \rho g z dx dy = \rho g x \tan\theta dx dy \tag{3.14}$$

where g is the gravity acceleration and ρ the density of the fluid. Integrating the above expression, we can evaluate the water-plane area restoring moment:

$$M_{MA} = \int_{S_0} x dF_h = \rho g \int_{S_0} x(x\tan\theta) dA = \rho g \tan\theta \int_{S_0} x^2 dA = \rho g \tan\theta I_y \tag{3.15}$$

The above expression is correct only if the body's cross section is constant along the z axis and if θ is small enough to guarantee that the entire top area of the structure is above the mean water level (MWL). In this formula, I_y is the second moment of area of the water-plane, which basically it's the capacity of the solid's surface to resist against momentum that can rotate the platform around the y axis. It's defined based on the geometry of the surface S_0 if the surface is horizontal ($\theta = 0$).

ii. **Mooring component** → With the exception of TLPs (see Chapter 1), mooring lines are used to counteract the inclining momentum of the structure by the generation of a force F_M.

In the case of a system with six DOFs, the 6x6 matrix $\mathbf{C}_{ML}$ represent the linear mooring stiffness, taking into account the coupling effects between different DOFs; however, for this analyses, we consider only the linear pitch restoring moment (i.e. moment generated from the mooring tendons to reduce the platform's rotation around the y axis). So, we can evaluate:

$$M_{\mathbf{ML}} = \mathbf{C_{ML}} \cdot \boldsymbol{\theta} \tag{3.16}$$

iii. **Moment arm effects** → This concern the effects caused by the relative movement of the buoyancy point B and the gravity point G, which define a lever arm giving a rise to a momentum of rotation in counterclockwise direction. In the case of an offshore wind turbine's platforms, the buoyancy force $\mathbf{F_B}$ is even bigger than the weight due to the downward mooring loads. Based on $\mathbf{F_B}$, defined as:

$$dF_h = mg - F_{MZ} \tag{3.17}$$

and the total moment generated from the level arm is:

$$M_{MA} = F_z \cdot z_B \cdot \sin(\theta) + F_{MZ} \cdot z_L \cdot \sin(\theta) - m \cdot g \cdot z_G \cdot \sin(\theta) = (F_B \cdot z_B + F_{MZ} \cdot z_L - m \cdot g \cdot z_G) \cdot \sin(\theta). \tag{3.18}$$

From this formulation, we can easily notice that F_{MZ} (vertical mooring force) has a negative sign, it means that it counteracts the buoyancy force in order the stabilize the object. The moment M_{MA} stabilize the structure only if it's bigger or equal to zero, or if the center of gravity G is positioned lower respect to the center of buoyancy B ($z_G \leq z_B$).

iv. **Wave force** → The wave force is very important, because it's responsible the horizontal translation and the pitch inclination of the solid towards the X direction, so for this purpose, the mooring system will generate a force $F_{MA(x)}$ in an opposite direction to F_E. These will lead to a pitch inclining moment, with respect to the origin O, equal to M_I:

$$M_I = F_e \cdot (z_e - z_O) \cos\theta - F_M(X) \cdot (z_L - z_O) \cos\theta \tag{3.19}$$

It's fundamental to state that, in this thesis study, we don't consider the effect of the wind on the structure. Finally we can write the equation to find the equilibrium position of the entire system:

$$\sum M = 0 \rightarrow M_I = M_R \rightarrow M_I = M_{WA} + M_{MA} + M_{ML} \tag{3.20}$$

where M_R is the restoring moment. Assuming small-angle approximation on θ, a non-linear algebraic equation can be determine:

$$\theta = \frac{F_e \cdot (z_E - z_L)}{\rho g I_x + mg(z_B - z_G) - F_{MZ}(z_B - z_L) + C_{ML}} = \frac{M_I}{C_{ML}} \rightarrow C_{ML_{\min}} \geq \frac{M_I}{\theta}_{\max} \tag{3.21}$$

Based on the processed data of the "G2 with Δt_2" (see 4.1.2) of the oc5 testing cases, we found 3.3 and from it, we can compute the maximum value, defined as $\theta_{\mathrm{MAX}} = 1.065°$, so we'll have $C_{ML_{\min}} \geq \frac{M_I}{1.065}$ where M_I can be approximately assumed equal to M_y.

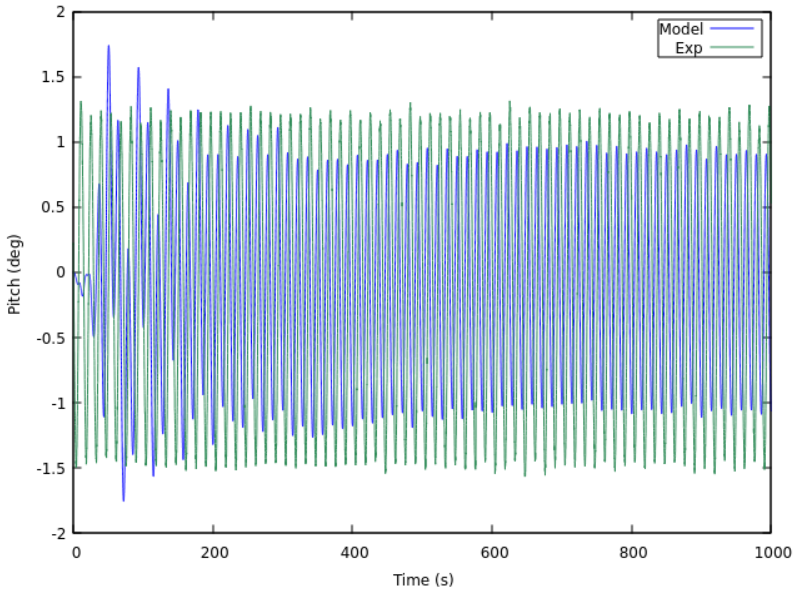


Figure 3.3: Pitch angle variation of the oc5 platform test

3.2 *Modelling of the fluid flow*

Liquid and gas are referred as fluids and fluids undergo continuous deformation under the action of shear forces. The main distinction between a liquid and a gas lies in their rate of change of density and the density of the latter changes more readily than the former one. Fluids are treated here as continua, so we'll neglect the motion of their molecules. Furthermore, we'll consider only Newtonian fluids, which are fluids that exhibit a linear relationship between the shear stress applied to them and the rate of shear deformation at which they undergo. This type of fluid can be described by four fields (which are physical quantities that varies on space): density (ρ), velocity ($\mathbf{u}$), pressure (p) and temperature (T). These properties will vary or not vary in the space and/or in time based on the type of fluid (ideal or real fluid) and the type of flow. We'll consider an unsteady, non-uniform and turbulent flow, so we'll have:

$$\left(\frac{\partial u}{\partial s}\right)_{t=\text{const}} \quad \text{and} \quad \frac{\partial u}{\partial t} \neq 0, \quad \frac{\partial p}{\partial t} \neq 0, \quad \frac{\partial \rho}{\partial t} \neq 0, \quad \frac{\partial T}{\partial t} \neq 0 \tag{3.22}$$

Based on this type of flow, from the 2nd Newton's law, we can derive the differential form of the Navier-Stokes equation which it's numerically solved by the CFD (Computational fluid dynamics) program, in our case Open Foam; it can be write as:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}^T) - \nu \nabla^2 u = -\frac{1}{\rho}\nabla p + q, \tag{3.23}$$

where q is a general momentum source term, ν kinematic viscosity and if the fluid is incompressible:

$$\nabla \cdot \mathbf{u} = 0. \tag{3.24}$$

The most difficult calculation of the CFD algorithm is related to the non-linear convection of momentum, which is related to the turbulence model applied to the CFD simulation; in the next pages, we'll introduce which are the approaches to solve the Navier-Stokes equation for turbulent flow.

3.3 *Turbulence modelling*

To understand what do we mean with eddies and how we can properly addressed them in CFD modelling, we can initially consider a simple 2D shear flow as represented in Figure 3.4. In this case,

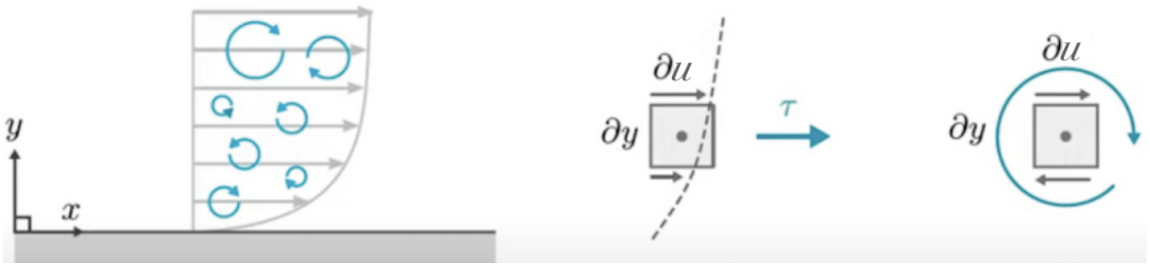


Figure 3.4: Simple 2D shear flow

due to the formation of turbulence mainly caused by the shear stress between the fluid particles and the wall, we have a turbulent fluctuation superimposed on top of the mean flow profile. If we look at a single fluid element (which is represented in the previous figure as a square block) it's sheared by the eddies (clockwise direction) of the flow and by the mean flow. This can be easily be seen in the mean flow block, which it presents an faster moving velocity at the top and a slower moving velocity at the bottom caused by the shear stress τ. The shear stress for the mean flow (generally, it's called viscous shear) τ can be determine as

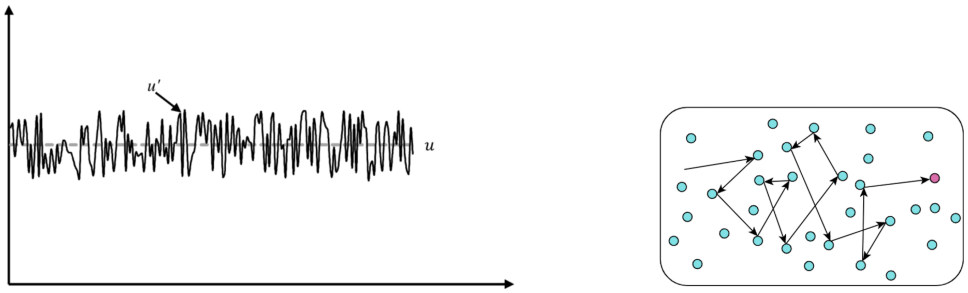
$$\tau = \mu \frac{\partial u}{\partial y} \tag{3.25}$$

where μ is the dynamic viscosity; the shear stress, generated from the eddies, is given by the Reynolds stress which is defined as:

$$\tau = -\rho\overline{(u'v')}. \tag{3.26}$$

If we look at the time trend of the x-axis velocity (u, Figure 3.5a) we understand that, compared to the linear velocity, the turbulent one is very unpredictable and not constant, so it'll be very difficult to compute it in the transport equation. There 3 possible approaches developed in the past to solve this issue:

Direct Navier-Stokes approach (DNS): this approach is the most complete and precise one because it captures all the scales of turbulence, from the smallest to the biggest vortex, but this required a very fine temporal and spatial discretization. We need a lot of computational power and time, that's why it's apply only for small scales project where we want to capture very precisely



(a) Horizontal velocity time trend

(b) Brownian motion

the vortex formation. Hence, it's used mainly in laboratory simulations and not for practical applications. In this thesis, we didn't use the eddy viscosity models.

Large Eddy Simulations (LES): it's used to solve only the big scale turbulence and the small dissipative ones can be solved statistically or modelled using a sub-grid scale model. The bid eddies (eddies that are bigger than the grid) are solved with a mesh, so the refinement of the mesh plays a decisive role if we want to adopt this approach. Due to the grid dependence, the requirement of high computational power and sensibility respect to the initial boundary conditions of this model make it not so used used for big marine CFD simulations.

Reynolds-Averaged Navier-Stokes (RANS): it's the most used approach for CFD simulations because, respect to the previous two, it doesn't required so much computational power. The eddy viscosity models, which define the RANS modelling concept, compute the equations of fluid based on the mean velocities $\bar{u}$, $\bar{v}$ and $\bar{w}$. In order to apply this concept, we assume that the turbulence fluctuations of the fluid follow the Brownian Motion theory[3] of gas particles (Figure 3.5b). In this way, we assume that the eddies fluctuation are random even; however, this main hypothesis is far from reality (especially for impinging jets, strong curvature and axial strain simulations) since the real individual fluctuations are correlated.

How can we actually used this Brownian Motion concept to determine the Reynolds stress? Assuming the fluid as the gas particles inside the stationary box, as we said before, but this time, we assume that the background fluid is moving through the box (not a stationary box anymore) following a shear profile (faster on the top region of the box and slower in the lower region). Thus, the particles which are randomly moving inside the box, will start to accelerate in the top part of the region colliding with the slower particles, so they'll transfer momentum down to the lower region particles (the velocity gradient will point downward). Essentially, we'll have a net transfer of momentum in the direction of the speed gradient (see Figure 3.6). Therefore:

$$\mu_t \frac{\partial \overline{u}}{\partial y} = -\rho \overline{u'v'} \tag{3.27}$$

[3]This type of fluctuation was named for the first time by Robert Brown in 1927 and it refers to the random movement displayed by small particles that are suspended in fluids. In this state, each particle changes its path when another particles collides with it, generating a transfer of momentum/energy between them. The smaller is the particle and the lower is the fluid viscosity, the higher will be the speed at which it moves inside the fluid causing an higher probability of collisions with the increase of the total entropy of the system.

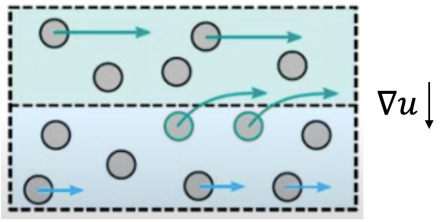


Figure 3.6: Acceleration of random fluid particles

where μ_t is the eddy viscosity, which it's a fictitious quantity that controls the rate of diffusion of turbulence (strength of the collisions); if it increases, it will increment the net momentum from the upper part particles to the lower ones. This assumption was firstly introduced in 1877 by Boussinesq. Extending this model to the 3D case:

$$2\mu_t \left(\frac{\partial \overline{u}}{\partial x} + \frac{\partial \overline{v}}{\partial y} + \frac{\partial \overline{w}}{\partial z} \right) = -\rho \left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2} \right) \tag{3.28}$$

For incompressible fluids, the calculation will be a little more complicated because it required the use of the Tensor notation (i,j) and since the demonstration of the shear stress equation isn't the objective of the thesis, we'll write directly:

$$-\rho \left(\overline{\mathbf{u}'\mathbf{u}'} \right) = \mu_t \left(\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T - \frac{1}{3} (\nabla \cdot \overline{\mathbf{u}}) \mathbf{I} \right) - \frac{2}{3} \rho k \mathbf{I} \tag{3.29}$$

at which, the turbulent kinetic energy is $k = \frac{1}{2} \left(\overline{u'u'} + \overline{v'v'} + \overline{w'w'} \right)$ and $\mathbf{I}$ is the identity matrix (it's called Kronecker delta δ_{ij} if we are using the Tensor notation).

k-ω SST model turbulent model

There are different formulations to calculate μ_t based on the model that we choose; the researchers in the past have developed 4 basic approaches; each of them can be more or less accurate based on the case of study; these are the **k-ϵ model**, **k-ω model**, **k-ω SST model** and the **Smagorinsky model**. There are no better turbulence models than others, but some of them are very different to each other and therefore it's necessary to choose the type that best suits our case study.

We have choosen the k-ω SST model, which it derives from the k-ω one, since the separation zones in the flow (an example of it, could be the separation of airflow passing through an aerodynamic airfoil) occurring around marine structures and, in the case of ocean waves, the free water surface is a critical factors; this CFD approach can provide a very detailed description of the eddies structure. Menter in 1994 found that, in both k-ϵ and k-ω models, the wall shear stressed were too high, so the flow wasn't separating correctly from the smooth surface of the wall, compare to the experimental measurements; So, the k-ω SST model attempts to handle this problem giving a better separation prediction. Without going into the mathematical demonstration of the results, which objective is far away from this thesis, we'll see how this model can accurately evaluate the

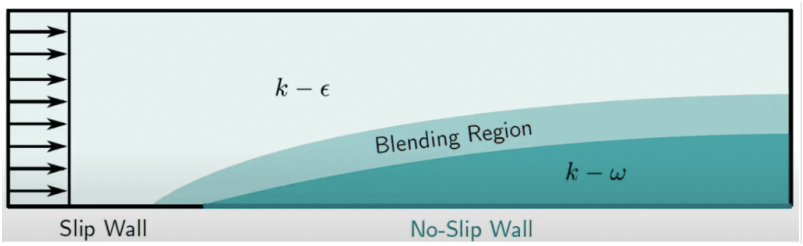


Figure 3.7: Models switch using the blending function. Reproduced from simscale.com

turbulence flow separation.

The k-ϵ model tends to be more accurate in the free-stream region area respect to the region of cells near the walls, vice versa, the k-ω is more precise in the region near the walls (as we can see from the Figure 3.7). A great idea was to introduce a trigonometric function called *blending function* F_2, it varies from 0 (k-ϵ model) to 1 (k-ω model) with the distance from the **closest wall** d, which it is computed as:

$$F_2 = \tanh(\arg_2) \quad \arg_2 = \max\left(\frac{2\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{\omega d^2}\right) \tag{3.30}$$

where β^* is a coefficient and ω is the turbulent frequency. Since the blending function is defined by an hyperbolic tangent, we'll have a smooth transition between the two basic turbulence models. This concept was initially applied to the k-ω BST model (using F_1 as blending function) but then

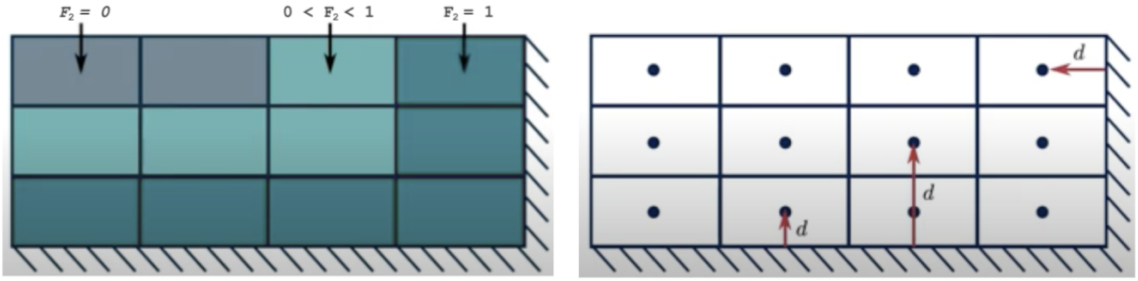


Figure 3.8: Blending function calculation. Reproduced from simscale.com

upgraded to the SST one. Now, we can finally calculate turbulent viscosity μ_t as:

$$\mu_t = \frac{\rho k}{\omega} \tag{3.31}$$

$$\mu_t = \frac{\alpha_1 \rho k}{\max(\alpha_1 \omega, SF_2)}. \tag{3.32}$$

The upper formula is related to the original k-ω while the down one represents an upgraded version of the k-ω SST model; in the equation 3.32, ρ is the fluid's density, S is the magnitude of the shear

strain and α_1 is a mathematical coefficient.

Now, knowing the main wave characteristics (H_w, T_w, d_w) we can calculate the not-scaled group celerity of the wave in deep water condition (in this case, it's equal to the inlet free stream velocity) as:

- *Standard wave*

$$U_s = C_{G.s} = \frac{1}{2}\sqrt{\frac{g \cdot L_{0.s}}{2\pi}} = 11.159\,\mathrm{m/s} \quad \text{where} \quad L_{0.s} = \frac{g \cdot T_{w.s}^2}{2\pi} = 319.14\,\mathrm{m}$$

- *Wave type 1*

$$U_1 = C_{G.1} = \frac{1}{2}\sqrt{\frac{g \cdot L_{0.1}}{2\pi}} = 9.418\,\mathrm{m/s} \quad \text{where} \quad L_{0.1} = \frac{g \cdot T_{w.1}^2}{2\pi} = 227.36\,\mathrm{m}$$

and supposing a turbulent intensity[4] T_u of 5% (it's the reference value for medium turbulence flows, see [37]), the scaled turbulent kinetic energy value for both waves is:

$$K_s = \left(\frac{3}{2}\right) \cdot (U_s \cdot T_u)^2 = 0.00936\,\mathrm{m^2/s^2}$$

$$K_1 = \left(\frac{3}{2}\right) \cdot (U_1 \cdot T_u)^2 = 0.00665\,\mathrm{m^2/s^2}$$

Now, considering the coefficient $C_\mu = 0.09$ and the turbulent length scale $I = 0.1 \cdot \phi_\mathrm{M}$ (where ϕ_M is the diameter of the moonpool) we can finally calculate:

$$\omega = (C_\mu)^{3/4} \cdot (k^{1/2}/l) \tag{3.33}$$

After setting the k and ω coefficients in the *0.orig* folder as initial conditions for the differential equations, we choose which turbulence model use in the **turbulenceProperties** file:

Listing 3.3: turbulenceProperties.txt code

```
simulationType  RAS;

RAS
{
    RASModel        kOmegaSST;

    turbulence      on;

    printCoeffs     on;
}
```

[4]This coefficient is the ratio of standard deviation of fluctuating flow velocity to the mean flow velocity.

3.3.1 *Volume of fluid method*

Since the study of the platform FOWT is a multi-phase flow case (i.e. analysis and simulation of flows involving multiple phases such as liquid and gas interacting within a system), the model has been modified by adding a free surface between water and air from the introduction of the Volume of Fluid (VOF) method.

This method, introduced by Steve Hirt and Barry Nichols in 1981, if applied in a CFD program, it divides the domain into a grid of cells and tracking the volume fraction of each phase (liquid and/or gas) using the transport equation within each cell. A scalar field α is defined, which, usually, is equal to 1 for liquid and 0 for gas. The free boundary is represented as the cells with iso-contour $\alpha = 0.5$ and it's very used in CFD analysis for an easy visualization of the water-air interface, as we can see here: About the previous figure, it's very important to highlight that the mesh has

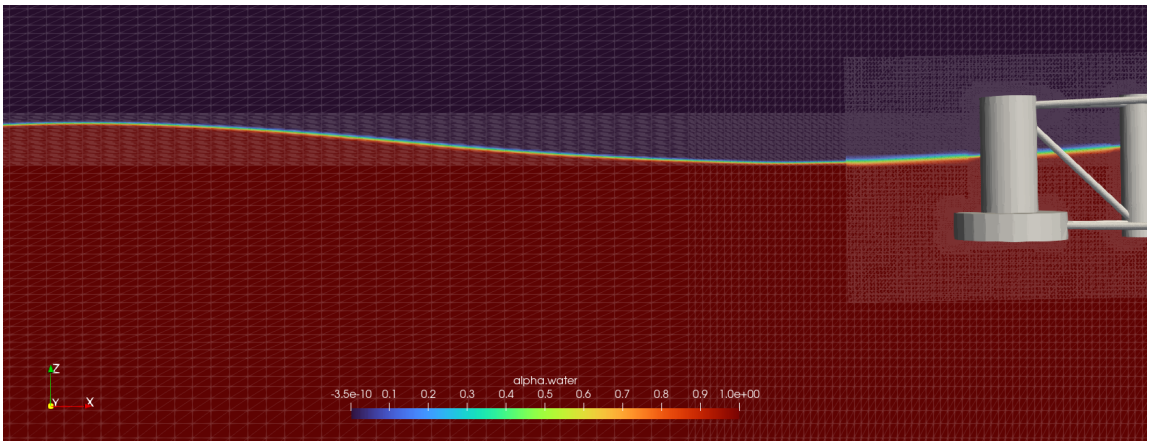


Figure 3.9: OC5 model ParaView, α is represented by alpha water

to be very fine in correspondence to the water-air interface, in order to achieve sufficiently precise results; regarding the wave, the number of cells both in the x and y directions is fundamental to the well computation, from the calculator, of the wave height (vertical direction) and lenght (horizontal direction) and so a better estimation of the results in the processing of the transport equation in each cell.

Overall, using this method, we need to add another transport equation that it'll be coupled with the Navier-Stokes equations:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{u}\alpha) + \nabla \cdot (\mathbf{u}_r(1-\alpha)\alpha) = 0 \tag{3.34}$$

where $\mathbf{u}_r$ is the relative velocity between the two phases normal to and pointing towards the free surface (related to the compression effects). If κ is a fluid property (like density ρ, dynamic viscosity μ, velocity field $\mathbf{u}$), which κ_l is for liquid and κ_g for gas, defined as the weighted averaged based on the volume fraction α:

$$\kappa = \alpha \cdot \kappa_l + (1-\alpha)\kappa_l. \tag{3.35}$$

There are other options, in alternative to the VOF, such as the *marker particles method*, the *level set method* and *level segments method*. This is set in the **transportProperties** file in OpenFoam and it contains:

Listing 3.4: transportProperties.txt code

```
phases (water air);

water
{
    transportModel   Newtonian;
    nu               1e-06;
    rho              998.2;
}

air
{
    transportModel   Newtonian;
    nu               1.48e-05;
    rho              1;
}

sigma                0.07;
```

where "nu" is the kinematic viscosity (in m^2/s) and "sigma" is the surface tension (in $\frac{N}{m}$) between water and air.

After the understanding of the mechanical fluid behavior and its model theory, we can finally set the **fvSchemes** to compute the transport equation in our CFD simulation:

Listing 3.5: fvSchemes.txt code

```
ddtSchemes
{
    default          CrankNicolson 0.5;
}

gradSchemes
{
    default           cellLimited leastSquares 1.0;
}
divSchemes
{
    div(rhoPhi,U)    Gauss linearUpwindV grad(U);
    div(U)           Gauss linear;
    div(phi,alpha)   Gauss vanLeer;
    div(phirb,alpha) Gauss linear;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
```

```
    div(phi,k)      Gauss linearUpwind default;
    div(phi,epsilon) Gauss linearUpwind default;
    div(phi,omega) Gauss linearUpwind default;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}
```

3.3.2 *Discretization of the governing equations*

In this section we'll see how a CFD program can mathematically compute the Navier-stokes equations thought the mesh. The most used approach for the commercial software is the Finite Volume Method (FVM), and it's an alternative of the well established FEM. The adopted FV method for structural dynamics was firstly introduced by Demirdžic´and Muzaferija in 1994 and then, extended in the OpenFoam framework.

The approach just listed, divides the simulated domain into a collection of control volumes (grid cells) that serve as representations of spatial regions where the fundamental physical principles are actively applied. So, inside each cell's volume, by applying the Gauss theorem, the integral form of the equation 3.23 is obtained:

$$\frac{\partial}{\partial t}\int_V \rho\mathbf{u}\,dV + \int_S \rho\mathbf{u}\mathbf{u}\cdot\hat{\mathbf{n}}\,dS = -\int_S p\,dS + \int_S \underline{\boldsymbol{\tau}}\cdot\mathbf{n}\,dS + \int_V \rho\mathbf{q}\,dV \tag{3.36}$$

where $\underline{\boldsymbol{\tau}}$ is the shear rate tensor and, the surfaces fluxes are approximated to the cell's average solution, which can be defined by the different quadrature methods. Interpolation methods are necessary to extract the value of the cell surface from its center, as we can see in the Figure 3.10.

The governing equations of different cells are coupled together because the values on a cell face are calculated using the central value of its neighbors and then assembled to a matrix system. Since, the case of study is a unsteady problem (i.e. the fluid properties are time-related quantities), the temporal term of the equation 3.36 must be modelled, assuming a generic fluid quantity with κ, as:

$$\kappa_{n+1} = \kappa_n + \int_{t_n}^{t_{n+1}} f(t,\kappa(t))\,dt \tag{3.37}$$

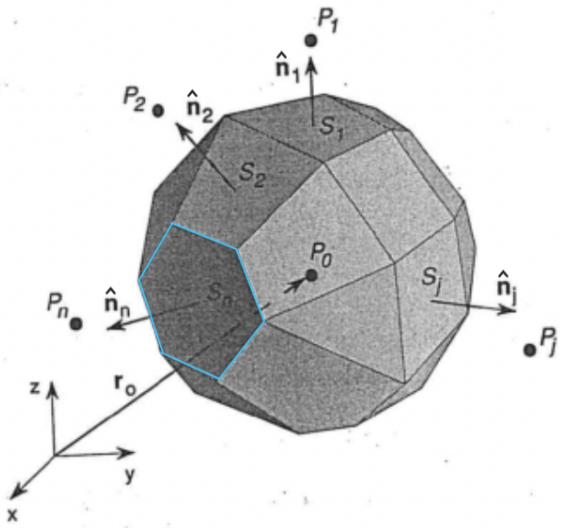


Figure 3.10: Cells domain representation

where $f(t, \kappa(t))$ is the time derivative of $\kappa(t)$.

In our model a moving overset mesh is present, so the Arbitrary Lagrangian-Eulerian method (ALE) must be adopted to account the additional convective terms related to the movement of the mesh domain. We can simply explain this method by defining two different domains, a physical domain that change in time (right one) and a time static domain (left one) which is easier to compute respect to the first one: so, the mapping function $\varsigma(t)$ (it's considered as a time function since

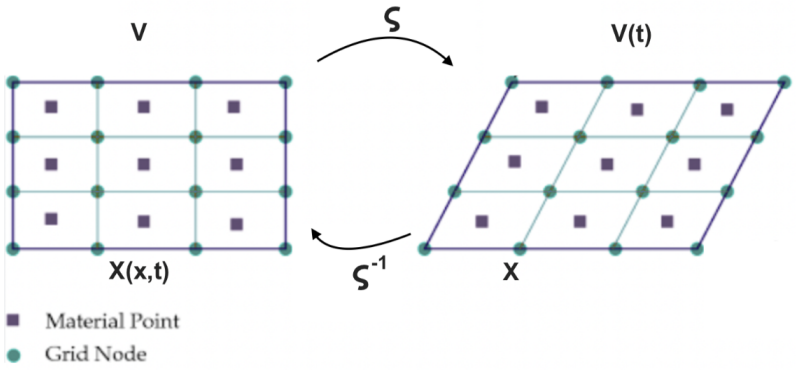


Figure 3.11: Volume demonstration of the ALE method

the V(t) depends on time) used to pass from the reference domain V to the physical domain V(t), is described using an Eulerian approach; we just need to apply the mapping function inside the transport equation. Without going into the mathematical demonstration (which include the use of the jacobian of the inverse matrix of the mapping function in the convection term of the transport

equation), we'll obtain:

$$-\int_S \rho u v_m \cdot \hat{\mathbf{n}}\, dS \tag{3.38}$$

this is the term that has to be added to the transport equation if we want to consider a transient simulation, and, because the mesh velocity v_m is unknown, by applying the geometric conservation law we obtain:

$$\frac{d}{dt}\int_V dV = \int_S \mathbf{v}_m \cdot \hat{\mathbf{n}}\, dS. \tag{3.39}$$

The latter equation define the variation of the mesh volume over time as the volume swept by the boundary and if it's satisfied, uniform flows uniformly flowing thought the cells is conserved.

3.3.3 *PIMPLE algorithm*

Even if the incompressible equations, momentum and mass transport equations are decoupled from the energy equation, so we don't need to find the pressure and the velocity fields or an equation of state (due to the constant density and temperature constraints required), since both of these two variables aren't present in the mass conservation equation.

If we want to solve the equation 3.36 (see 3.2) we'll need to suppose a guess value for the **p**. However, in this way, the **v** obtained from the Navier-Stokes equations cannot guarantee the continuity equation; this problem is called **pressure-velocity coupling**. To solve this issue, we need to developed an external equation that will permit us to calculate the pressure field, and inserting it in the equation 3.36, the resulting velocity will satisfy the mass conservation of the fluid.

Starting from the definition of the momentum equation in matrix form:

$$\mathbf{M}\mathbf{u} = \nabla p \tag{3.40}$$

where p is the kinematic pressure and **M** is a matrix of coefficients, coefficients that are derived by decomposing the differential terms of the momentum equation using the FVM; in this way we'll have n equations for each center of the cell. Now, if we separate the matrix of coefficient into a matrix made by the diagonal components **A** (which will be easy to invert) and another one made by the off diagonal components **H**, we obtain:

$$\mathbf{A}\mathbf{u} - \mathbf{H} = \nabla p, \tag{3.41}$$

in this way, with some mathematical simplifications, we can get the *Poisson equation of pressure*:

$$\nabla \cdot (A^{-1}\nabla p) = \nabla \cdot (A^{-1}\mathbf{H}). \tag{3.42}$$

→ **SIMPLE (Semi-Implicit-Method-Of-Pressure-Linked-Equations)**: This method is simply based on these 4 steps:

1. We define the initial value of ∇p to find the coefficient of **M** and, by solving the equation 3.40, the velocity field **u** is evaluated (usually defined as *Momentum predictor*).

2. Now, knowing $\mathbf{u}$, we can get the corrected $(\nabla p)_c$ from the Poisson equation 3.42.

3. Using the corrected pressure field to re-calculate the velocity field $\mathbf{u}_c$, it will satisfy the continuity equation, so:
$$\mathbf{u}_c = \mathbf{A}^{-1}\mathbf{H} - \mathbf{A}^{-1}(\nabla p)_c \tag{3.43}$$

4. Repeat the cycle if the velocity field doesn't satisfy the momentum equation, since the source term ($\mathbf{H}$) depends on the velocity field; this correction process is called *Outer corrector* or *Momentum corrector*.

The main downside of this approach is that it doesn't consider time derivative components, so it could be optimal only for steady-state problems so the pressure term is only approximated (which required under-relaxation factors to achieve stability); the latter issue can be solved using the SIMPLEC algorithm, but it requires an higher number of iterations for each step, but it ensure a smaller convergence rate.

→ **PISO (Pressure-Implicit-of-Split-Operations)**: This type of algorithm, introduced in 1986, instead, solves the momentum predictor (see equation 3.40) once and with the $(\nabla p)_c$ it'll compute the *inner loop*: Using this approach, usually, we don't need under-relaxation factors, based on the

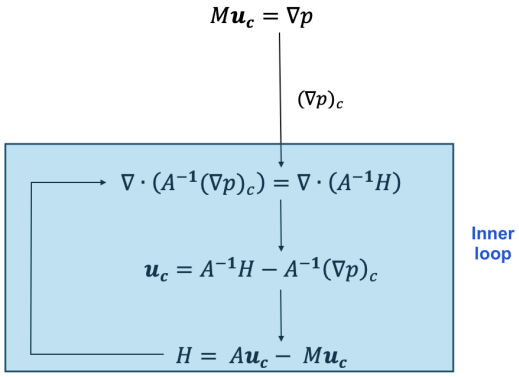


Figure 3.12: PISO algorithm scheme

fact that we use the PISO algorithm for very small time steps (assuming a very small CFL number); the SIMPLE algorithm requires more than 5000 iterations to earn the convergence for each time step, so, if we apply this method for a transient simulation (made by a thousands of time steps), this will required an enormous amount of computational time. That's why we prefer to use the PISO, since the inner loop is much shorter that the outer loop.

→ **PIMPLE (PISO-SIMPLE)**: It's the most used one and it can be applied for transient simulations with CFL≥1 (large time steps) at which, for each time step, the steady state solution is found with the SIMPLE loop. Before proceed to the next time step, as a normal SIMPLE algorithm may be do for transient problems, here the outer-correction loops are applied to ensure a convergence (if this loop isn't applied, the PIMPLE method will be exactly equal to the PISO approach). The PIMPLE algorithm suits well our case because it coupled the rigid body equations with the multi phase ones: all of this procedure is defined in OpenFoam in the **fvSolution** (file placed in the

system folder), in our case:

Listing 3.6: fvSolution.txt code

```
solvers
{
    "cellDisplacement.*"
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-08;
        relTol          0;
        maxIter         300;
    }

    "alpha.water.*"
    {
        nAlphaCorr      3;
        nAlphaSubCycles 1;
        cAlpha          1;
        icAlpha         0;

        MULESCorr       yes;
        nLimiterIter    15;
        alphaApplyPrevCorr  yes;

        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-9;
        relTol          0;
    }
    "pcorr.*"
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-8;
        relTol          0;
    }
   p_rgh
    {
        solver          PBiCGStab;
        preconditioner  DILU;
        tolerance       1e-8;
        relTol          0.01;
    }
      p_rghFinal
```

```
    {
        $p_rgh;
        relTol          0;
    }

    "(U|k|omega|epsilon).*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-08;
        relTol          0;
    }
}

PIMPLE
{
    momentumPredictor    no;
    nOuterCorrectors     50;
    nCorrectors          3;
    nNonOrthogonalCorrectors 3;

    ddtCorr                     yes;
    correctPhi                  no;

     moveMeshOuterCorrectors no;
    turbOnFinalIterOnly no;

    oversetAdjustPhi     no;

 residualControl
    {
          p_rgh
          {
           tolerance    1e-4;
           relTol       0;
          }

          U
          {
           tolerance    1e-4;
            relTol       0;
          }

        "(k|epsilon|omega)"
          {
           tolerance    1e-4;
```

```
            relTol          0;
        }
    }
}

relaxationFactors
{
    equations
    {
        ".*"  1;
    }
}
cache
{
    grad(U);
}
```

3.3.4 *Courant–Friedrichs–Lewy Condition(CFL)*

From the numerical point of view, in order to obtain convergence and stability by solving the PDEs within the domain, the CFL conditions has to be satisfied. How can we describe this condition? Initially supposing having a 1D cell where some flow pass thought it, we define the *Courant number* as the fraction of the cell that the flow moves, with a horizontal velocity $\mathbf{U}$, across in a time step Δt. Regarding the 3D case, the situation is more complicated because we'll need a method to specify the distance of the flow across the cell and the velocity incident to a face of the cell. The first one can defined as $\Delta x = \frac{V}{A} = \frac{\text{Cell Volume}}{\text{Total Surface Area}}$ and the second one can be seen as the normal component of the velocity to the external surface; according to OpenFOAM:

$$C_0 = \frac{1}{2} \cdot \Delta t \cdot \frac{\sum_f |u_f \cdot \hat{n}_f| \cdot A_f}{V} \tag{3.44}$$

where A_f is the face area, $\mathbf{u_f} \cdot \hat{\mathbf{n}}_\mathbf{f}$ is the normal component of the velocity to the surface of the block. Since most of the cells present different sizes (based on the fact that we increased the grid-spacing resolution in the areas near the water-air interface and close to the platform), each single block has a Courant number associated with it, which is bigger for smaller cells and higher flow field.

The CFD solver take into account only the maximum C_0 to measure the stability of the simulation and it has to be lower compared to a fixed threshold specifically related to the case study (in our simulation, we fixed $(C_0)_{\max}$ equal to 10); this can be seen as a first indicator to understand if the time step Δt was correctly selected.

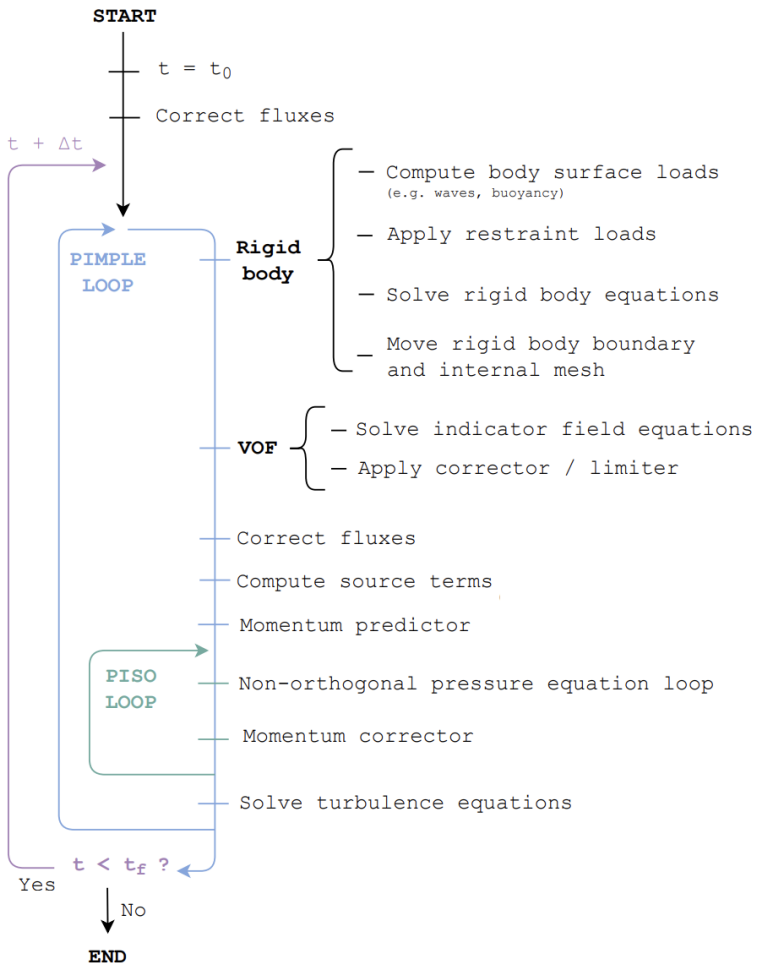


Figure 3.13: PIMPLE algorithm scheme (reproduced from [11])

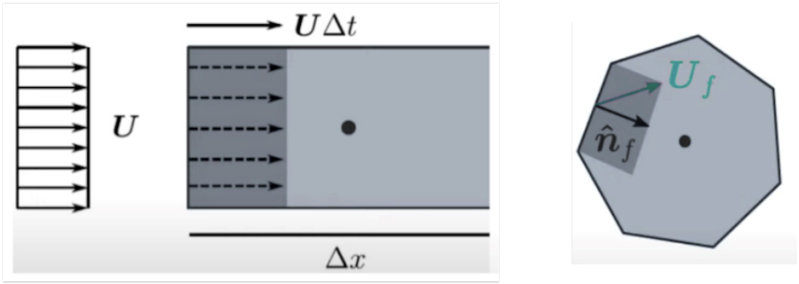


Figure 3.14: Cell fluidynamic representation. Reproduced from simscale.com

Chapter 4

Verification and validation of the model

Numerical grid and temporal verification, along with comparison with experimental data, are two crucial aspects within the field of the Computational Fluid Dynamics (CFD). These processes are vital to ensuring the reliability and accuracy of results obtained through numerical simulations. The grid, or mesh, forms the basis of CFD simulations. The quality of the grid significantly impacts the accuracy and stability of the model. Grid numerical verification involves analyzing its density, distribution, and quality. While, Temporal verification is fundamental in transient CFD simulations, where flow behavior varies over time. It's essential to select an appropriate time step for the simulation and ensure that the numerical solution remains stable and accurate throughout the time interval.

4.1 *Offshore IEA Project*

Since 2005, the International Energy Agency Wind Technology Collaboration Programme (IEA Wind), section of IEA, it started to act as an international vehicle of development and cooperation in research projects, called *Offshore Code Comparison Collaboration (OC)*, of sea and oceans wind turbines with the aim of "foster collaborative research and the exchange of best practises and data" (stated in the IEA Wind TCP annual report 2020, [26]). The first project was the OC3, started in 2005, which it was run under the Task 23 while its extensions OC4, OC5 and OC6 were carried out under the Task 30.

4.1.1 *OC5 DeepCwind model*

The Offshore Code Comparison Collaboration Continuation with Correlation or OC5, a 1:50 scale model of the DeepCwind semi-submersible FOWT was tested, for the first time in 2011 and then

re-tested in 2013 at the MARIN (acronym of Maritime Research Institute Netherlands) ocean basin and then used in 2015-2016 by IEA Wind in the Task 30. The NREL (National Renewable Energy Laboratory) led this project by conducting a series of calibration studies of the model with the objective of increasing the accuracy of the numerical model compare to the experimental data measured in the wave tank [29].

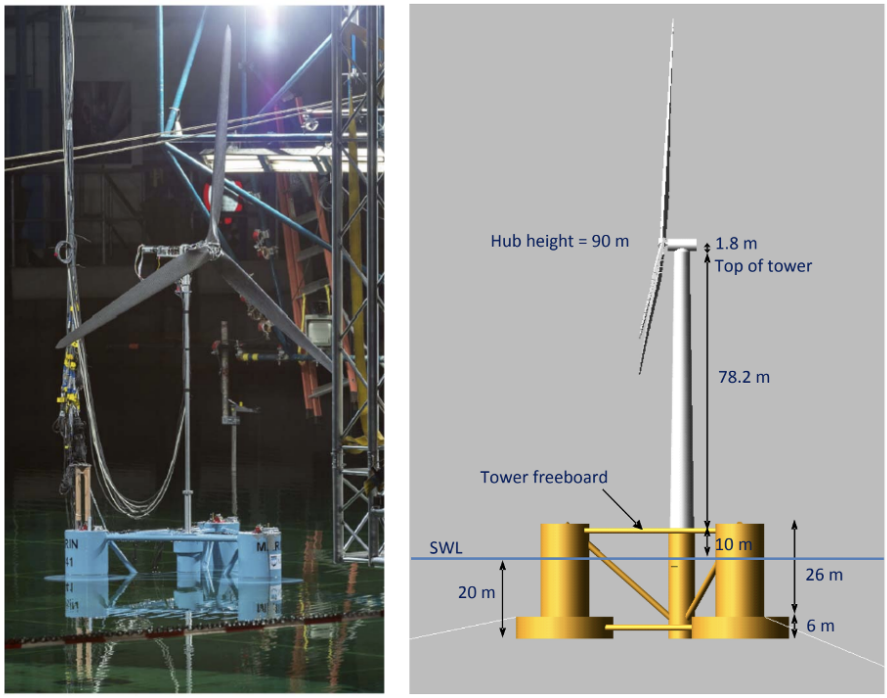


Figure 4.1: OC5 structure. Reproduced from [29]

The waves, instead, were generated by controlled flat-type wave makers (it can generate waves in all directions, but regarding this case, they preferred the waves straight to the turbine with a 180° angle). All measurements were sampled at the rate of 14.142 Hz (100 Hz model scale), so a time step ΔT_{exp} of 0.0707 seconds. It's important to notice that, the system was tested under the Froude-scale[1] for both wind and wave loads without any uncertainly assessments; in this way, a comparison with CFD tools will become a qualitative evaluation.

During the initial static simulations, NREL found that the system wasn't in equilibrium for the prescribed draft, so, for this purpose the University of Maine suggest that the biggest uncertainty is related to the total mass of the structure due to the the additional mass of cables and sensors of the OC5 DeepCwind experimental model and since, most of the mass in concentrated in the platform, they modified the mass of it even keeping the original center of mass. The selected material for

[1]In most of CFD models, the concept of scaling is applied to downscale a physical problem in a more appropriate size for the simulation on a scale length called Froude scale equal to the ratio between the fluid velocity and $\sqrt{gL}$

Table 4.1: Platform properties. Reproduced from [29]

Description	Value
Complete system mass	1.3958×10^7 kg
Draft	20 m
Displacement	1.3917×10^4 m^3
CM location below SWL	8.07 m
System roll inertia about CM	1.3947×10^{10} kg-m^2
System pitch inertia about CM	1.5552×10^{10} kg-m^2
System yaw inertia about CM	1.3692×10^{10} kg-m^2

the columns is Styrodur® 3035 CS with the use of steel weights inside the structure to obtain the defined CM position (reported in 5.1) and the tubes, that connect each platform's columns in order to consider the structure rigid, are made in aluminum. A summary of the geometry, with the representation of both plan and side view dimensions is well illustrated in Figure 4.2.

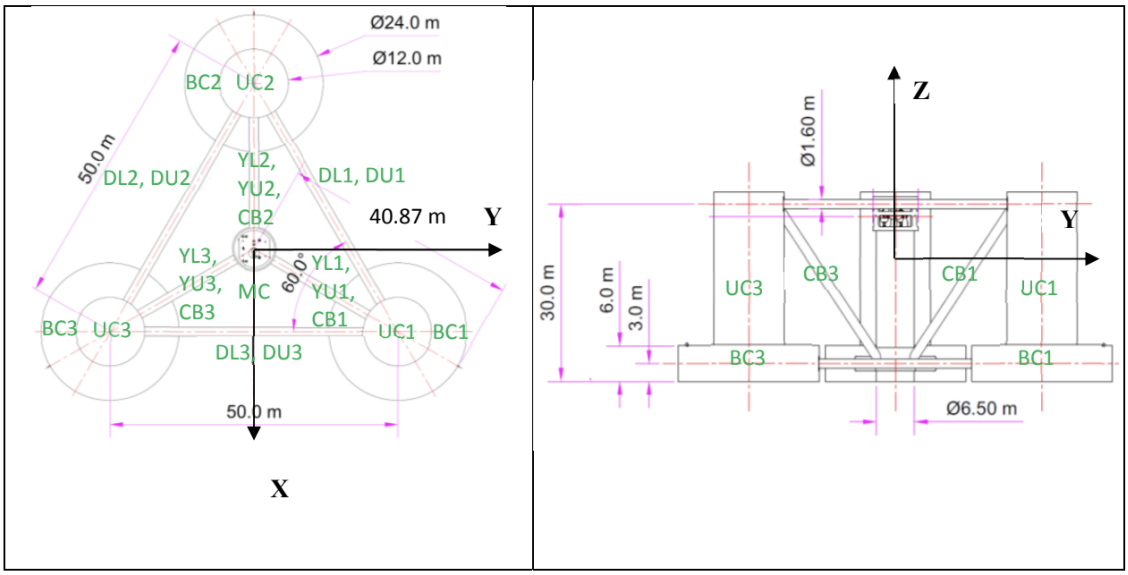


Figure 4.2: Plan (left) and side (right) view of DeepCwind Semisubmersible Platform. Reproduced from [29]

where gL is the characteristic length of the problem.

4.1.2 *Verification and validation process*

The numerical setup is the most important part in the entire simulation process since the accuracy of the results and the computational time are fundamentally based on the grid and time resolution. For this purpose we need to create a fined mesh only where we need an high resolution solution of the Navier-Stokes equations in order to guarantee enough computing time to obtain results that are accurate enough to the grid study process and to compare them with real experimental data (validation process).

The numerical computational domain, as shown in Figure 4.2, was defined as a rectangular box with a gradual variation of the refinement layers of the grid in the 3 directions towards the center of each box's surface; In the z-direction, a one more layer with a higher resolution, was set to achieve enough accuracy in the computation of the wave elevation. The *discretization error*, which is the

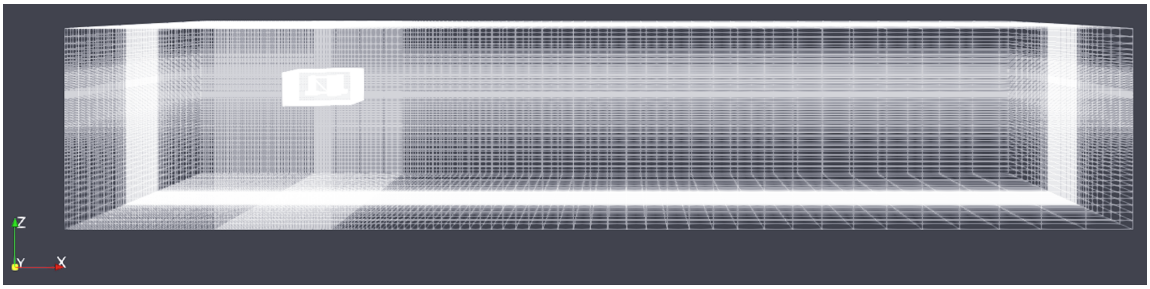


Figure 4.3: Plan (left) and side (right) view of DeepCwind Semisubmersible Platform.

consequence of the approximation of the finite-volume method (see 3.3.2) used to transform the PDEs into a system of algebraic equations (Eça and Hoekstra, 2014 [8]), is sensitive for both spatial and temporal discretization. Grid-space and time-step convergence were analyzed by a testing matrix of 5 possible configurations with a static refinement factor *r=1.25* for both grid and time-step solutions. The time resolutions selected were $\Delta t_1 = 0.005$, $\Delta t_2 = 0.0035$ and $\Delta t_3 = 0.0025$, while regarding the grid size, G_1 represent the coarse mesh, G_2 the medium one and G_3 the fine one. The starting point of the descretization error analysis starts with the definition of the convergence

Table 4.2: Spatial and temporal testing

Grid	N	Testing matrix		
G_1	2.78 M		Δt_2	
G_2	5.18 M	Δt_1	Δt_2	Δt_3
G_3	7.87 M		Δt_2	

ratio R, coefficient based on the solutions of the fine grid ϕ_3, medium one ϕ_2 and coarse one ϕ_1:

$$R = \frac{\phi_3 - \phi_2}{\phi_2 - \phi_1} \tag{4.1}$$

If $R \in (0, 1)$, a monotonic convergence is verified, so the Richardson extrapolation is applied to estimate the numerical error δ_{RE} and the observed order of accuracy p_{RE}; with the three solutions

only the leading term can be estimated, providing the following term:

$$p_{\mathrm{RE}} = \frac{\ln(1/R)}{\ln(r)} \tag{4.2}$$

$$\delta_{\mathrm{RE}} = \frac{\phi_3 - \phi_2}{1 - r^{p_{\mathrm{RE}}}} \tag{4.3}$$

The correction factor C_F, proposed by Stern et al. (2001) [32], quantify how far the uncertainties of solutions are respect to the asymptotic range ϕ_∞ of the grid size and the numerical error ϕ_D:

$$C_F = \frac{1 - r^{p_{\mathrm{RE}}}}{1 - r^{p_0}} \tag{4.4}$$

$$\delta_D = C_F \cdot \delta_{\mathrm{RE}} \tag{4.5}$$

where p_0 is related to the limiting order of accuracy as the grid-spacing approaches zero, in our case we adopt $p_0 = 2$ for a second-order accuracy method. From 4.5,

$$\phi_\infty = \phi_3 - \delta_D. \tag{4.6}$$

At the end, the uncertainty U_D, if the solution is closed to the asymptotic range (in this case $C_F \approx 1$), was estimated by (Wilson et al., 2004 [41]):

$$U_D = \begin{cases} [2.4(1 - C_F)^2 + 0.1]|\delta_{\mathrm{RE}}|, & \text{if } |1 - C_F| < 0.25 \\ |1 - C_F||\delta_{\mathrm{RE}}|, & \text{if } |1 - C_F| > 0.25 \end{cases} \tag{4.7}$$

otherwise, for C_F significantly less or greater than unity:

$$U_D = \begin{cases} [9.6(1 - C_F)^2 + 1.1]|\delta_{\mathrm{RE}}|, & \text{if } |1 - C_F| < 0.125 \\ [2|1 - C_F| + 1]|\delta_{\mathrm{RE}}|, & \text{if } |1 - C_F| > 0.125 \end{cases} \tag{4.8}$$

The estimation of the error, both for wave elevation based on the amplitude of the harmonic results, gave different results of the spatial study compare to the temporal one. Starting from the grid analysis: Here, the graphical representation of the grid study regarding the step: From the graphs, we

Table 4.3: Estimated results of the discretization error and uncertainty based on the **wave elevation** of the grid-size and time step. The the solution error in percentage of ϕ_∞ corresponds to ε_i.

Study	ϕ_1 [m]	ϕ_2 [m]	ϕ_3 [m]	R [-]	p_{RE} [-]	C_F [-]	ϕ_∞ [m]	U_D [%]	ε_1 [%]	ε_2 [%]	ε_3 [%]
Grid	4.4744	4.4979	4.5115	0.5638	2.568	1.375	4.5348	0.6375	1.33	0.8	0.52
Time	4.4449	4.4979	4.5325	0.6544	1.9	0.9387	5.0763	0.7	12.4	11.39	10.71

demonstrated what shown in the 4.3 with the ε_i coefficients, so, by increasing the number of cells the numerical solutions, the wave amplitude will converge to ϕ_∞ with an error of 0.6375%. Since the difference between ϕ_3 and ϕ_2 isn't so big, using G3 will not improve too much the accuracy of the results but it'll need a very high computational time. With $U_D = 0.7\%$, the Δt_2 was the solution chosen, even with the presence of some discrepancy (higher ε_i compare to the grid ones), for

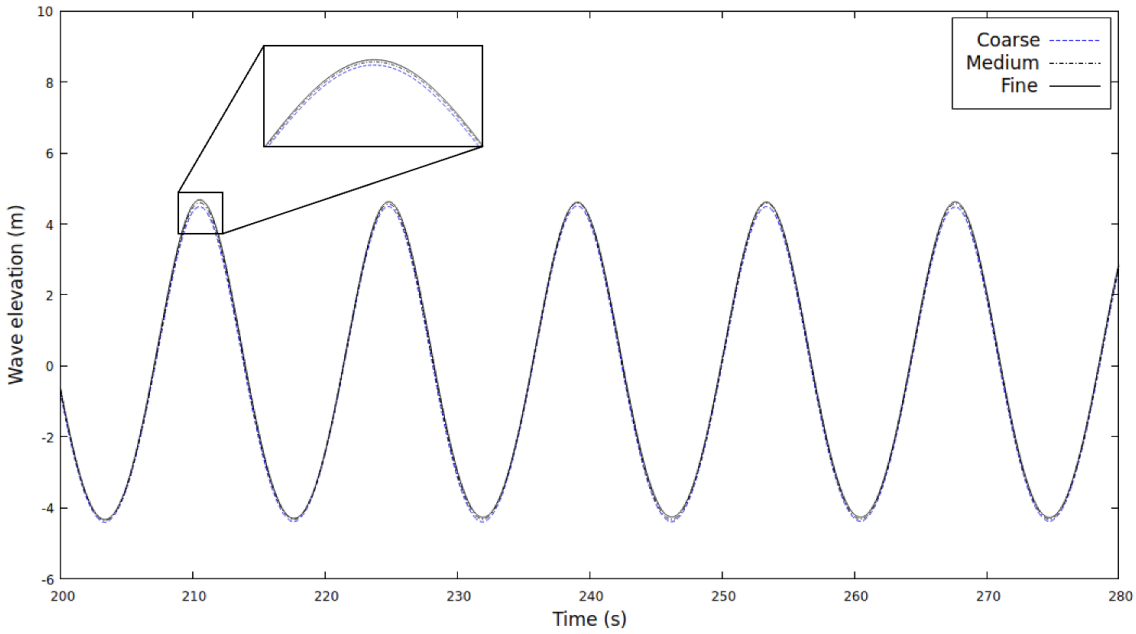


Figure 4.4: Time trend wave grid study chart

a benchmark value of $\phi_\infty = 5.07637\,\text{m}$. In both analyses, a quite big difference is present between G1 and G2, and regarding the spatial study, Δt_1 and Δt_2.

The discretization setup will be even more complete with the calculation of the errors related to the heave motion, pitch angle variation and surge[2], but it has been chosen not to delve further into it since it'll deviate the attention of the reader to the actual objective of the thesis; just to have an idea, we show the grid study chart of the heave:

4.1.3 *Comparison to experimental data*

To validate the adopted high-fidelity numerical model setup, the hydrodynamics results obtained from CFD solver are compared to experimental measurements (Robertson et al., 2016, [39]) but the aerodynamic influence of the tower turbine wasn't considered, since our case study is related only to the platform itself without considering the rest of the FOWT. The oc5 structure of the experimental paper was subjected to a regular Airy wave with $H_w = 9.41\,\text{m}$ and $T_w = 14.3\,\text{s}$. For a more clear comparison, the results analysis were reported to the full scale both in the charts and in the tables.

In the next figures, a comparison between the experimental and CFD model (G2 and Δt_2) results regarding the wave elevation, heave motion and pitch angle variation is shown: From the

[2]It's very important to note that for more precise results we need at least 2000 seconds of simulation data. Since, surge motion involves larger displacements and longer response times and natural period, we need enough simulation time to get stable and an accurate results.

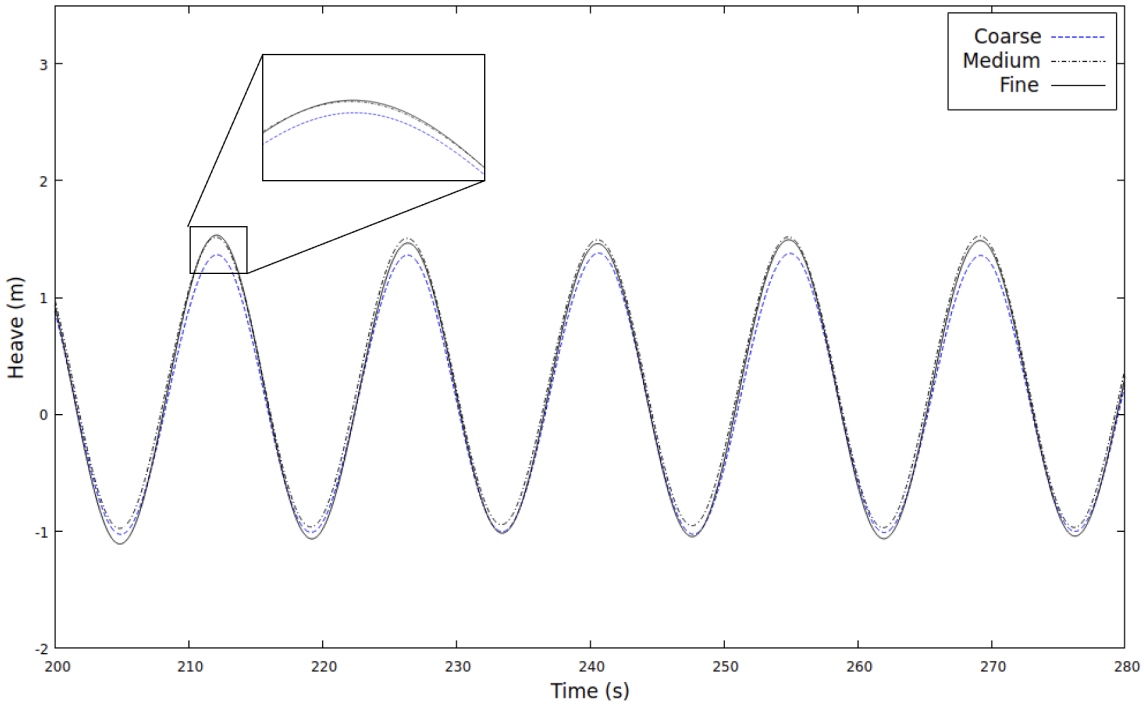


Figure 4.5: Time trend heave grid study chart

calculation of the amplitude of the previous curves, we can define the deviation of the model from the experiment: As we can see in 4.4, the deviation regarding the platform's motion (heave and

Table 4.4: Estimated deviations of the computed and measured results in terms of wave elevation η, heave motion z and pitch angle variation θ

Results	η [m]	z [m]	θ [deg]
CFD	4.4979	1.25435	1.06530
EXP	4.64	1.5776	1.44035
Deviation	3.06 %	19.4 %	26 %

pitch) is bigger respect to the wave elevation one; the main reasons behind these results are the non-perfect meshing on the off shore geometry (especially in the edge of the columns and in the tube external surfaces, see Figure A.3); mechanical inertia of the rigid body and steepness of the mooring systems generate other uncertainties. A more accurate results need a finer mesh of platform in the overset zone and the use of MoorDyN (coupled with OpenFOAM), see [12], to define the mooring-anchoring system in a more detailed way.

Observing the η values, both in 4.4 and 4.3, we can notice that, the more we increase the spatial and temporal resolution, the more the results tends to reach the experimental one. In the grid-space study, even if we use an infinite number of cells obtaining $\phi_\infty = 4.5348\ m$, we'll not

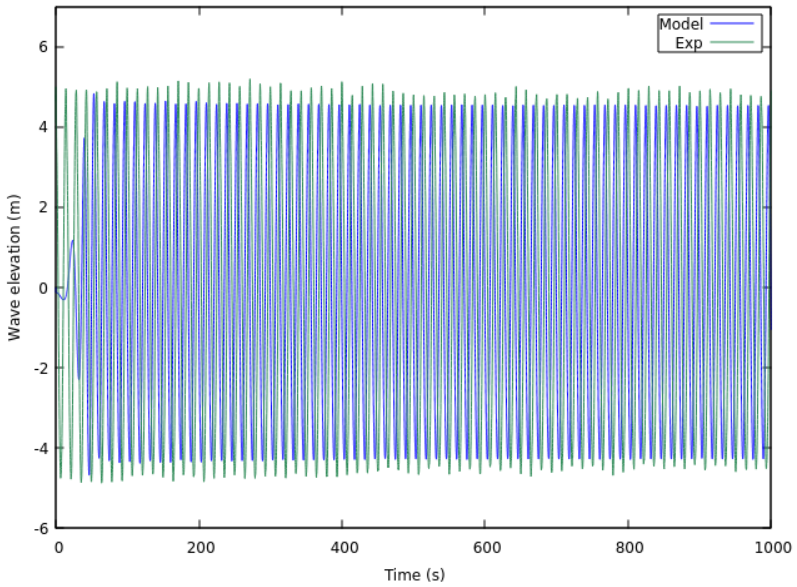


Figure 4.6: Time trend chart wave elevation

reach $\phi_{\text{exp}} = 4.64\ m$ basically due to the influence of experimental uncertainties (related to the conditions at which the buoy measurement was taken).

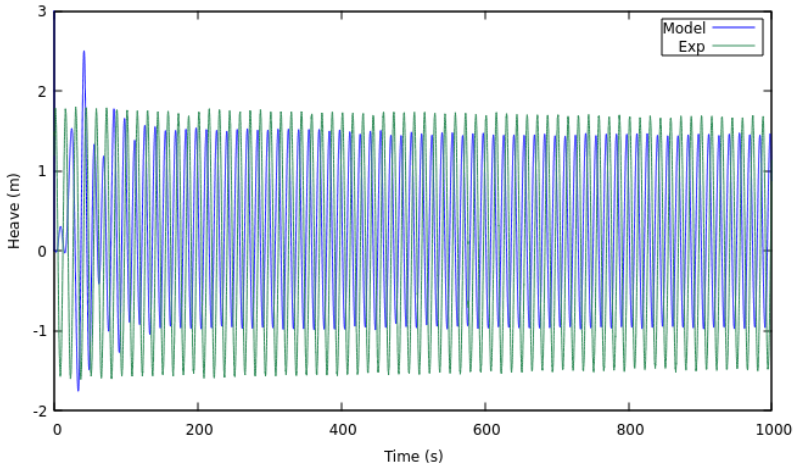


Figure 4.7: Time trend chart heave motion

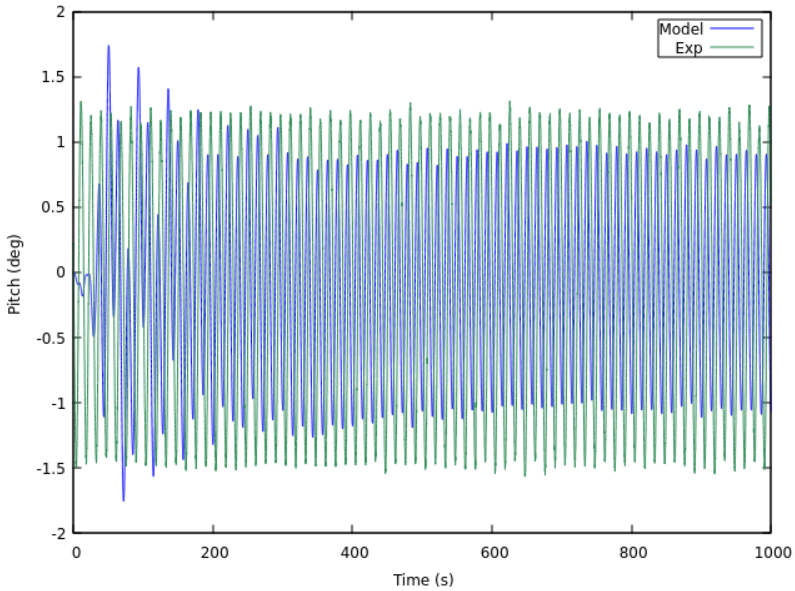


Figure 4.8: Time trend chart pitch angle variation

Chapter 5

The moonpool design concept

First of all, what is the meaning of the word "moonpool" and why such concept it's applied to FOWT?

A moonpool is a cavity located in the base of the hull of a ship, giving access to the water below, allowing researchers and technicians to lower instruments and submersible into the sea; also providing shelter and protection from ice in the case of high seas. The moonpool concept is especially applied in the offshore oil extraction field, allowing the access of the drilling rig to the water below without the need of another vessel. In the beginning of 2000, the concept of moonpool design was applied also to the FOWT platforms, various prototypes of the barge-type platform were developed with the aim to:

- **Safe maintenance operation**: An easy access to water is ensured without the use of external equipment, increasing the operational safety and reduce the maintenance costs.

- **Reduced exposure to weather conditions**: Reduction of the operators and equipment to severe weather conditions.

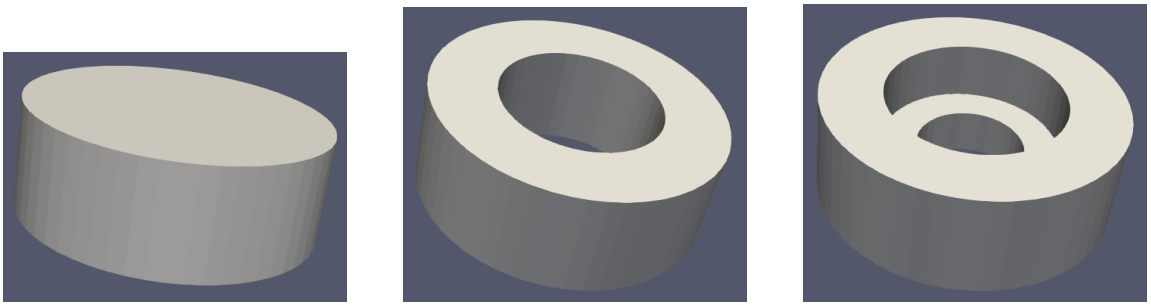
- **Risk mitigation**: This type of structure permit an efficient maintenance practices enable by the moonpool that helps the mitigation of risk of unexpected failures, increasing, in this way, the predictably of revenue streams.

Some studies, such as the "Effect of the Moonpool on the Total Resistance of a Drillship" [37], reveal that the moonpool cavity (placed at the base of the hull of a moving ship) will generated a piston-sloshing movement of the water inside the moonpool volume that, based on the size and shape of the moonpool, will generate a turbulence causing a reduction of the surge component of the ship's velocity. In light of this fact, in this chapters, we'll try to understand how the behavior, in deep-water sea conditions, of a simple geometry platform will change with the use of the moonpool design.

5.1 *Numerical analysis and comparison*

After the verification of the numerical model, using the OC5 platform's data with the spatial-temporal study and the comparison with the experimental data, we can finally test the behavior of the offshore structure of our specific case study.

The objective of this thesis is to analyze and comparing the physical performances of a simple cylindrical platform ($d_{\text{platform}} = 45\,\text{m}$ and $H_{\text{platform}} = 16.2\,\text{m}$) with 2 different types of moonpool shapes, to understand if the use of moonpool will bring advantages, not only from the economical point of view, but even for the dynamic point of view (see chapter 3). These are the 3 geometries: as we can see from Figure 5.1, both of these elements, present the same diameter and the same height, the simple moonpool structure (figure b) is made by a single through hole of $d_{\text{hole}} = 25\,\text{m}$ and the t-shape moonpool type of $d_{\text{top hole}} = 26\,\text{m}$, $h_{\text{top hole}} = 9\,\text{m}$ and $d_{\text{bottom hole}} = 8\,\text{m}$, $h_{\text{bottom hole}} = 7.2\,\text{m}$. These are the structural properties: The stiffness values of



(a) Normal **(b)** Moonpool **(c)** Shape

Figure 5.1: 3 different types of structure

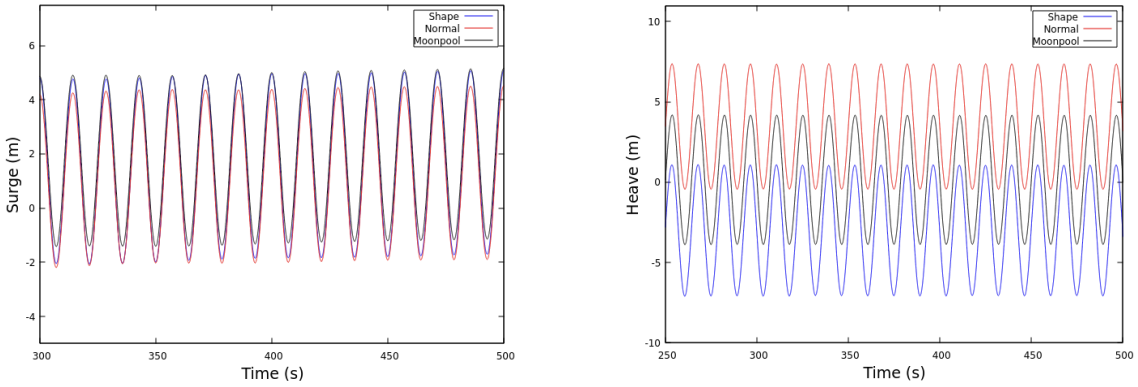
Table 5.1: Rigid body properties of the 3 platforms

Description	Value
Platform mass	1.169×10^7 kg
CM location below SWL	4 m
System roll inertia about CM	1.549×10^9 kg-m^2
System pitch inertia about CM	1.549×10^9 kg-m^2
System yaw inertia about CM	1.5281×10^9 kg-m^2
System roll inertia about CM	1.7549×10^9 kg-m^2
System pitch inertia about CM	1.7549×10^9 kg-m^2
System yaw inertia about CM	2.959×10^9 kg-m^2
System roll inertia about CM	2.2127×10^9 kg-m^2
System pitch inertia about CM	2.2127×10^9 kg-m^2
System yaw inertia about CM	3.8743×10^9 kg-m^2

`TranslationLinearSpring.1`, `TranslationLinearSpring.2` and `AxialLinearSpring` were computed from the values of the OC5 test case by applying a proportionality equation that takes into

account the difference of mass between the two platforms and the same concept was applied for the CM distance location below SWL.

The analysis is based on the study of the 3 main components of the motion trend (it's important to notice that some data aren't constant over the time, because only 500 seconds of simulation time were studied to avoid a lot of computational efforts from the cluster) supposing that, the same sea condition of the test platform $H_w = 9.41\,\text{m}$ and $T_w = 14.3\,\text{s}$:



(a) Surge movement chart

(b) Heave movement chart

Figure 5.2: Comparison between platforms

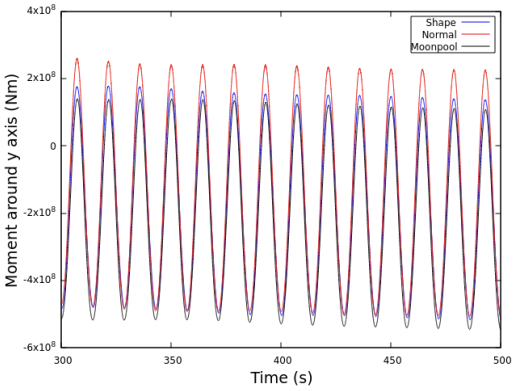


Figure 5.3: Moment around y axis

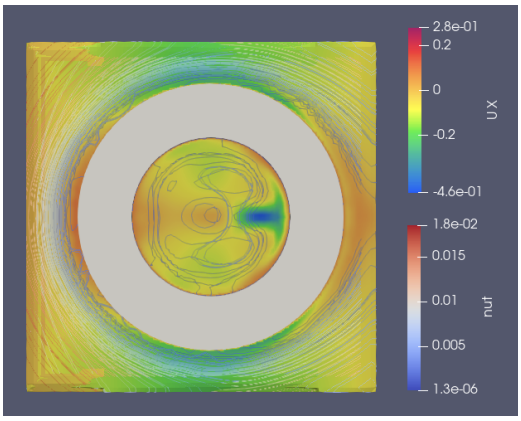
Quantitatively from the previous graphs, we can extract: from Figure 5.2 it discern the fact that, even with an increment of the amplitude of the surge and heave movements, compare to the normal geometry (element without the moonpool), the moonpool geometry reduce the moment around the y axis (consequently a smaller pitch angle) which is very effective for the overall FOWT equilibrium (see chapter 3, related to the wave force F_e). The placement of the "Shape" platform is well above the SWL compare the other ones, basically due to the non-constant, from the temporal

Table 5.2: Simulation results for moonpool analysis

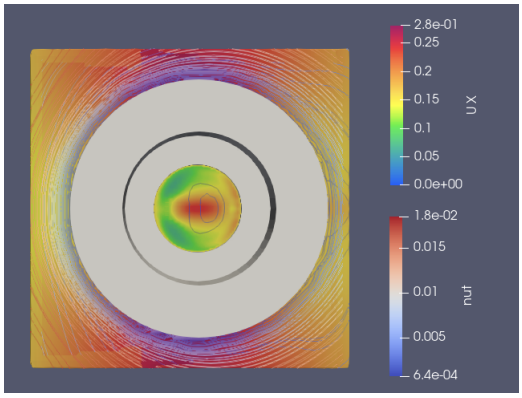
Heave (m)	Surge (m)	My $(N \times m)$
5.7456	4.7649	4.82×10^8
4.4247	4.4164	5.6×10^8
4.5975	4.5563	4.9×10^8

point of view, accumulation of water on the upper part of the hole that generates an impulsive gravitational downforce which "push" the platform down and so an increase of the heave is present (see 3.1.2).

The reduction of the momentum is based on the fact that, inside the moonpool area, eddies are generated creating a discontinuity on the shear layer (Sivabalan, Surendran [31]) as we can see from the next picture. The post-processing analysis, starting from the left to the right, allows the reader to



(a) Horizontal velocity field and turbulence generation above 10 m to the sea level, after 431 seconds of simulation time



(b) Horizontal velocity field and turbulence generation at sea level, after 431 seconds of simulation time

have a clear picture on how the turbulence is generated inside the moonpool (defined by the μ_t contour lines, it's called turbulence viscosity) that will cause a variation on the horizontal velocity field.

These pictures,also, permit us to visualize how the impact of the wave, towards the external surface of the platform, will decrease the velocity and increse the turbulence viscosity (even higher than the one inside the hole). The more complex is the shape of the moonpool and the more eddies are generated; this statement is verified from the fact that the reduction of M_y, compare to the simple cylindrical element, is higher in the t-shape respect to the simple moonpool and that's why we choose a turbulence model for the for this particular case-study (a classic laminar model wouldn't be able to describe accurately the torque change between the 3 platforms).

The shape moonpool is a recess type geometry [37], which is basically used to generate turbolence inside the structure; a variety of recess forms were studied in the past for ship solutions [34], since the moonpool shape is fundamental in that particular engineering field because it has direct

consequences on the reduction of ship surge speed and so, in the fuel consumption.

5.2 *External stabilizing force analysis*

About the 5.2b chart, it's highlighted the differences in buoyancy of the different structures: if the masses of the three structures are same, to keep equilibrium in water, the equilibrium positions in the vertical direction would vary for each structure. The Normal structure has the largest displacement per unit under water, resulting in the smallest equilibrium height (draft), and the Moonpool structure has the smallest displacement per unit, resulting in the largest equilibrium height (draft). The *risk of submergence* during rough sea conditions (extremely high waves) and *effects on the movements* like surge and heave, as we saw in Table 5.2, is responsible for a more difficult maintenance process.

Applying vertical forces to the structures is aimed at compensating the deficiency in buoyancy caused by structural changes, thereby keep consistent mass and consistent initial drafts for the three different structures is necessary. We added a **vertical force** F_+ (with an opposite sign compare to the gravitational one) applied on G point to the initial static body:

1) `case A`: Shape element with $F_+ = 250\,\text{N}$.
2) `case B`: Shape element without any external force added to it.
3) `case C`: Shape element with $F_+ = 595\,\text{N}$.

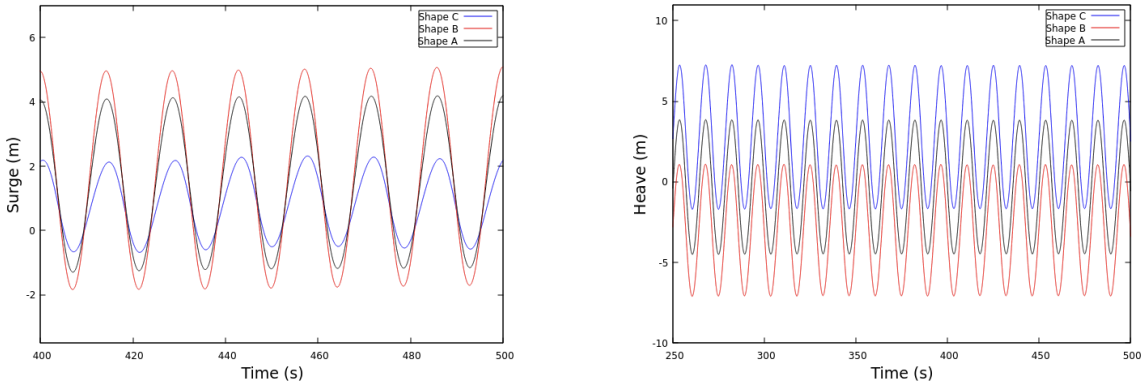
after 500 seconds of simulation, from the t-shape moonpool platform, we'll analyze how the surge (x), heave (y) and Moment (M_y) vary: A graphical point of view can be seen at Figure 5.5 and

Table 5.3: Simulation results for external force analysis

Type	Heave (m)	Surge (m)	My ($N \times m$)
A	4.6894	4.1696	4.51×10^8
B	5.7456	4.7649	4.82×10^8
C	5.0267	2.6645	3.01×10^8

Figure 5.6. A clearly reduction of the surge and heave movement and M_y momentum is denoted consequently to the application of the external positive force; with a closer look, we saw how the shapes of the surge and so the moment weren't anymore sinusoidal due to a generation of turbolence direclty in the upper part of the moonpool.

This idea of additional buoyancy system can be implemented from different solutions such as **Pneumatic and hydraulic systems** mounted beneath the structure; by controlling the fluid pressure they are able to create the exact upward force to generate the desired values of surge and heave amplitude. Another way can be using **Floats**, like the USFLOWT concept [7], based on inflatable devices filled with air or water.



(a) Surge movement chart

(b) Heave movement chart

Figure 5.5: Comparison between platforms

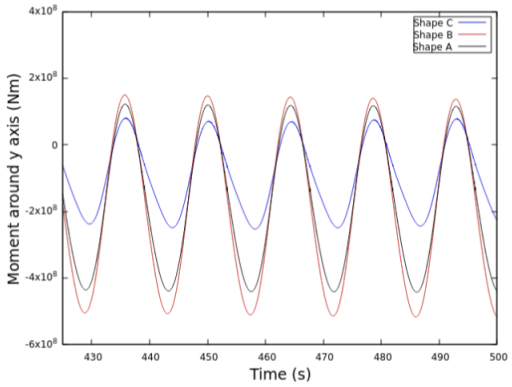
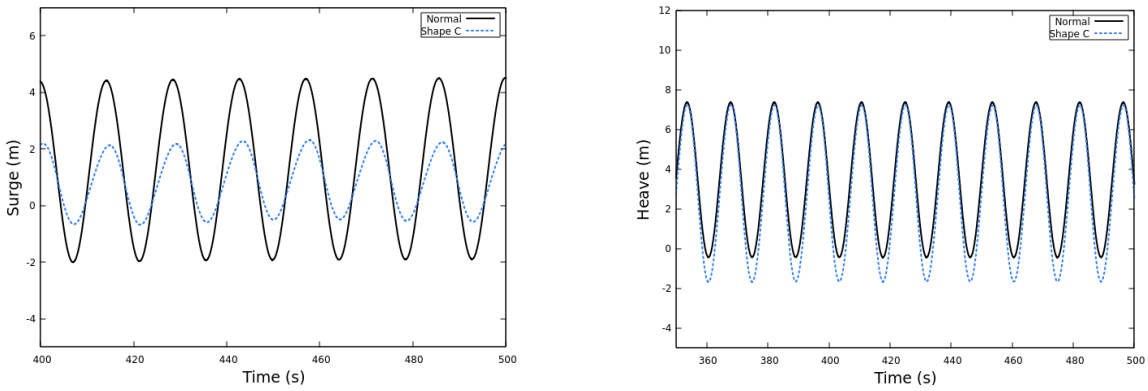


Figure 5.6: Moment around y axis

Understanding the previous concept, we can compare the improved moonpool platform with the basis cylindrical one (placed more or less at the same water level) obtaining Figure 5.7 and Figure 5.8.

Without the need to compute the amplitudes of the previous chart, we can easily understand the advantages in terms of surge and pitch movements, except for the heave because, even if they're placed more or less at the same water level, the amplitude of the platform without moonpool is smaller. The wave frequency of these curves is the same between "Normal" and "Shape C" since the wave is the same; in the next subsection, a variation in the curves period (or frequency) is present, due to the change of wave characteristics.



(a) Surge movement chart

(b) Heave movement chart

Figure 5.7: Comparison between platforms

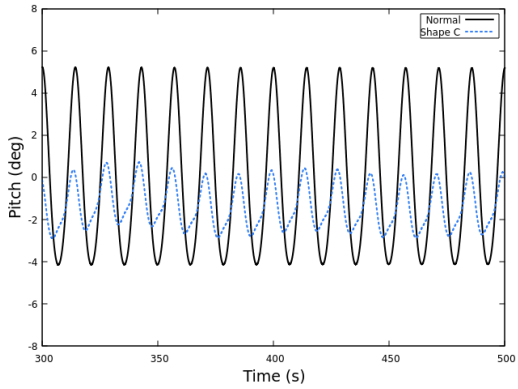


Figure 5.8: Pitch data comparison

5.3 *Variation of the wave characteristics*

Until now, we based the platform's study using the same wave (here defined as "Standard"), but in reality, based on the weather and so on the wind generation phenomena, the wave characteristics will change everytime. By changing the parametric background mesh and keeping the same moon-pool platforms, we check the element basic movements (surge, heave and pitch) of the 2 different case:

1^{st} *case*: "Shape" structure without any external force.
2^{st} *case*: Same platform with the presence of an external force $F_+ = 595\,\mathrm{N}$.

The two waves tested were:

- **Standard wave**: Regular airy, $H_w = 9.41\,\mathrm{m}$ and $T_w = 14.3\,\mathrm{s}$
- **Wave 1**: Regular airy, $H_w = 7.37\,\mathrm{m}$ and $T_w = 12.07\,\mathrm{s}$

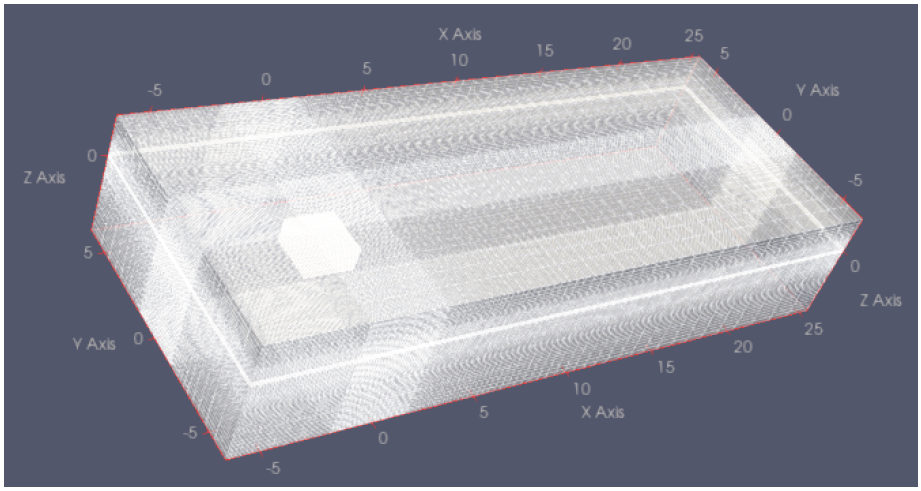


Figure 5.9: Mesh's scaled view of the standard wave case

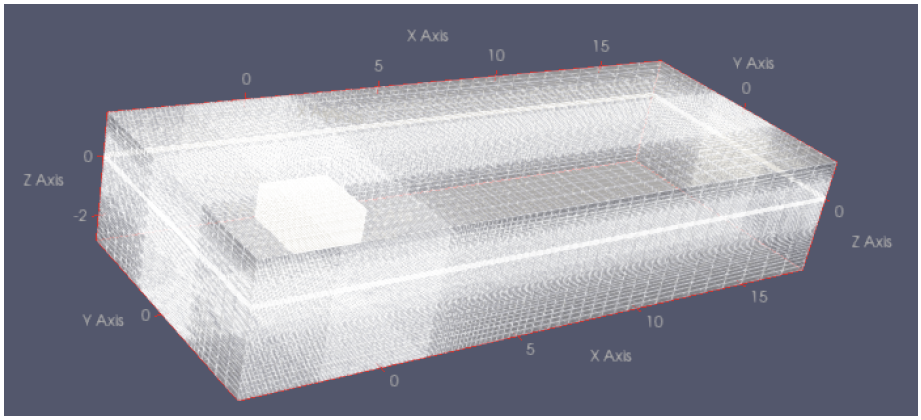
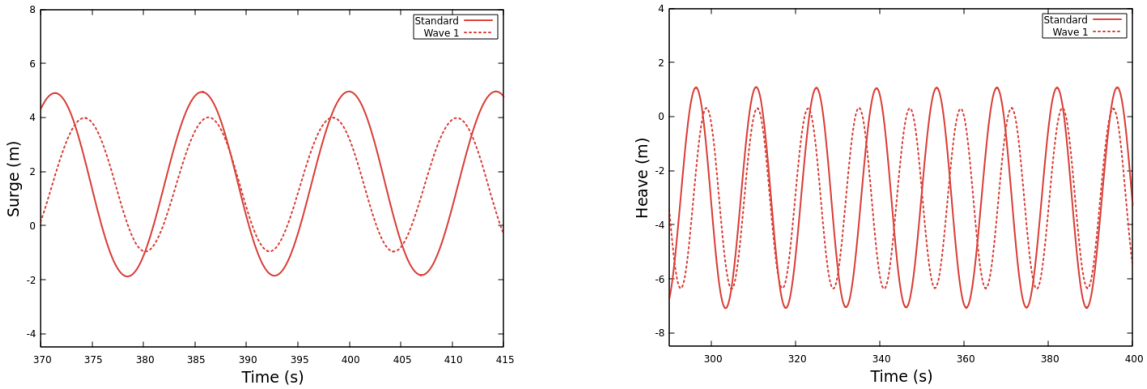


Figure 5.10: Mesh's scaled view of the wave 1 case

In the background mesh, all the distances of the mesh vertices respect to the center of the axis, where the platform is located, will change both in x,y and z directions (see 5.10). The reduction of the size of the background mesh, while the overset remains the same, will reduce the computational time.

What do we expect from the theory illustrated in the past chapters? Without even computing the results, we already expected from the formulas 2.9, a reduction of the displacements both in x-direction and z-direction of the water's particles and considering the platform as one of them, we expected a decrease on both heave and surge. Another way to understand the platform's behavior is to indirectly looking at the forces acting on it (see 2.4).

Regarding the pitch variation, it's not so straightforward since, as we saw in the equation 3.21 in the 3.1.2 section, the pitch depends by multiple factors such as dimensions and shape of the moonpool which cause the eddies generation, the positioning of the center of mass related to the weight distribution, the mooring system constraints etc...



(a) Surge movement chart **(b)** Heave movement chart

Figure 5.11: Comparison between platforms

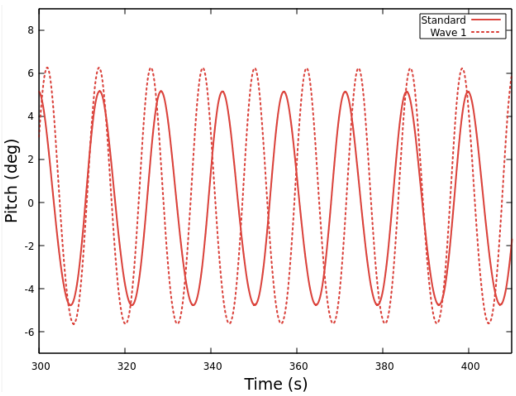
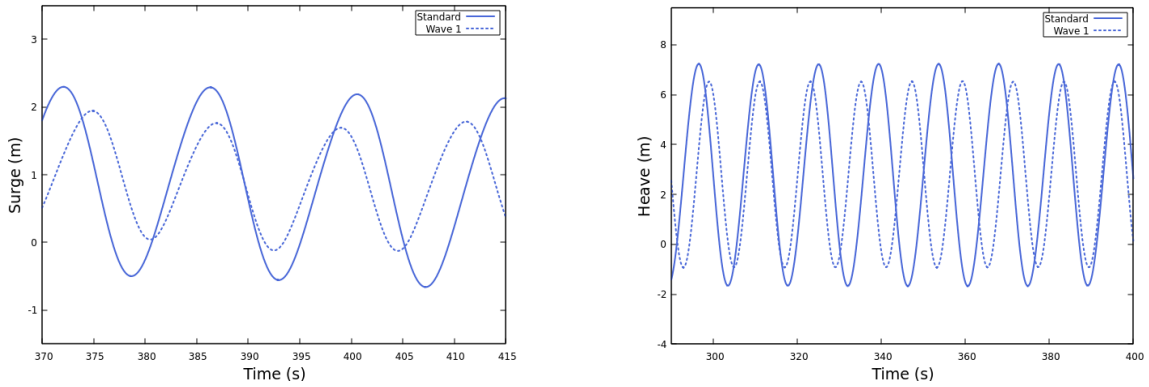


Figure 5.12: Pitch data comparison

From 5.14, we see how the moonpool platform enhance the rotation around its y-axis if the sea characteristics change, even if the heave and surge movements were decreased.

If an external vertical force is applied to the initial static body, what will happen? Will the situation change? Or will it remain the same?

As we already know from 5.2, the amplitudes are lower compare to the case without external force, but an interest thing has been noticed concerning the fact that, by changing from the standard wave to wave 1, the pitch rotation has been increased much more compare to the 1^{st} one:



(a) Surge movement chart

(b) Heave movement chart

Figure 5.13: Comparison between platforms

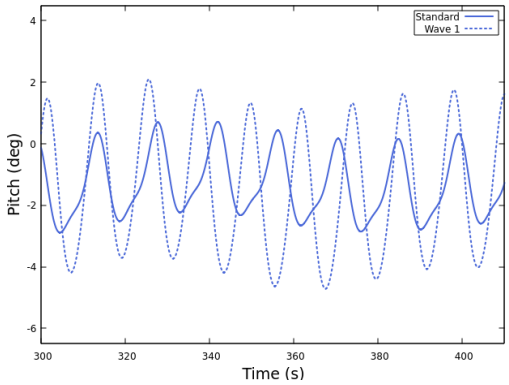


Figure 5.14: Pitch data comparison

The previous analytical tables allowed us understand the effect of the external vertical force to the moonpool platform regarding the structure's motion is better, in surge and heave terms but worse related to the pitch results, compare to a normal moonpool platform, when H_w and T_w decrease.

A very important element which generates both surge and angular momentum on a offshore object is the force and it's basically related to the magnitude of the fluid velocity (which is strictly correlated to the wave celerity)(see 2.4); to see how it's distributed on the external surface of the platform, we need to study the post-processing, as we can see in the picture 5.15. From the post processing, we have a visual representation on the platform movement of the "wave 1" case (see 5.16).

In the figure 5.16, it's clear how the overset mesh is following the platform's movement, as it's attached to it; so accurate results can be obtained, otherwise no simulations will guarantee sufficiently accuracy regarding the study of the performance data used to make a comparison between

Motion type	Standard	Wave 1	Difference
Surge $[m]$	4.7649	4.0778	-14.4 %
Heave $[m]$	5.7456	5.4174	-5.7 %
Pitch [deg]	6.3582	7.2611	+12.43 %

Table 5.4: 1^{st} case table

Motion type	Standard	Wave 1	Difference
Surge $[m]$	2.6645	2.3970	-10.04 %
Heave $[m]$	5.0267	4.2327	-15.79 %
Pitch [deg]	3.7607	5.9542	+36.83 %

Table 5.5: 2^{st} case table

different situations.

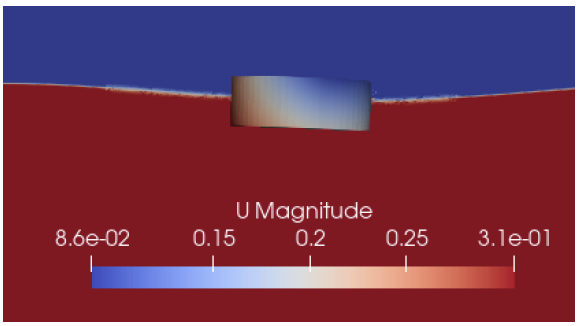


Figure 5.15: Velocity distribution on the external surface

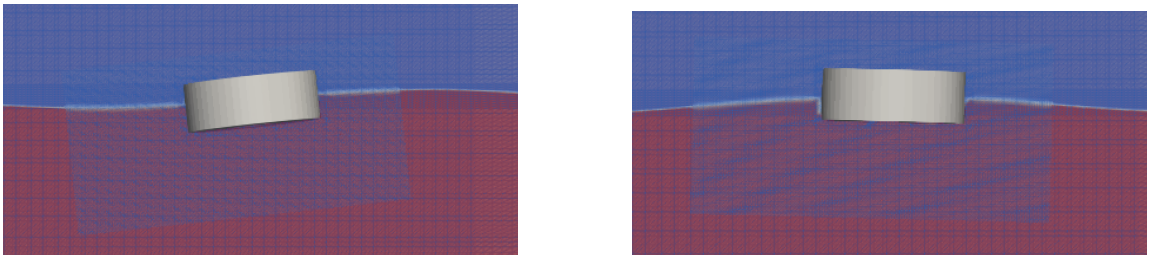


Figure 5.16: Wave 1 case, image capture at 431 and 434 seconds of simulation time

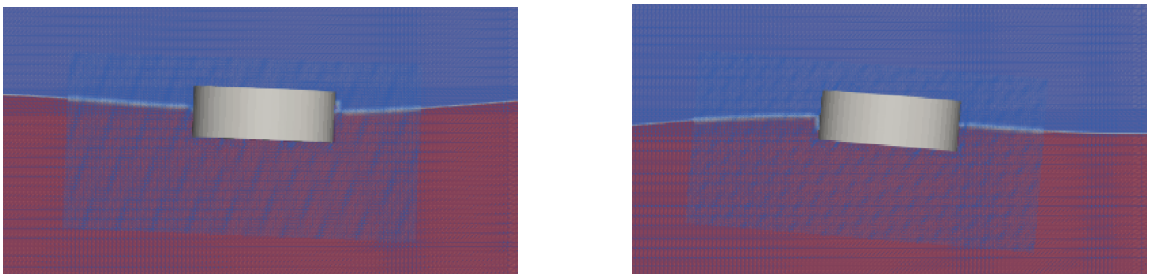


Figure 5.17: Wave 1 case, image capture at 438 and 441 seconds of simulation time

Chapter 6

Conclusions and considerations

The main objective of this thesis is the study of the effect of the moonpool design applied to a FOWT platform and, based on the analysis of the main movements and torque, to understand if this design is effective from the mechanical point of view.

Firstly, to achieve this goal, a precise multi-phase laminar CFD model was prepared using OpenFoam; starting from the CAD generation of the OC5 structure, grid and time step analyses has been made in order to test the case together with the comparison of the results to the experimental performance data; to figured out, if the model had sufficient accuracy. We figured out that the grid was sufficiently fine enough to define the wave conditions without the need of a big computational period and, we kept the same anchoring system keeping the same spring force.

After the integration of the k-ω SST turbulence model, with the aim was to comprehend what does happened if we simply apply two different moonpool design to the same cylindrical platform directly without adding an external force to it. We learned from both vertical and horizontal movements, that the platform behavior is worse than the normal one but, the more complex is the moonpool shape, the less torque is generated, and so the overall FOWT stability is improved due to the generation of eddies. Negative effects, mainly related to different water drafts of the moonpool platforms compare to the simple cylindrical one aren't negligible, so an additional external force was necessary to create a more realistic comparison between the platforms and since the performance of the "Shape" type (related to the moment around the y axis) were better compare to the simple moonpool one, it was the choosen one.

The external force impact on the performances, as we saw in the Table 5.3, increase even more the platform's stability, avoiding the draft's problems for a better stability.

In reality, the water conditions won't remain the same everytime so, with the wave simulations, we wanted to understand how the selected platform will behave in a different sea condition and what is the influence of the external force in this particular analysis. These simulations were computed also to verify if the wave's theory was successfully considered in the OpenFoam software and how the theory predicts the behavior of the offshore structure. The reduction of the surge and heave was expected, as we saw in 2, but an increase of pitch was unforeseen, at least from the theory,

probably because it's related to the mass distribution (i.e. position of the center of mass), inertia effects etc...
If an external force was applied, the pitch angle increment increases even more, so we understand how the effect of the external force has to be considered in order to make objective comparison.

In the beginning of the chapters 1 and 5, considerations from the economical and operational point of view has been made, so it was already very clear why the moonpool design (already applied in the barge type platform) is used a lot, especially in deep sea conditions; however from the physical point of view, the understanding on the behavior isn't so straightforward.

This thesis had the aim, starting from the tested CFD model, to study how the platform behave under different conditions, by highlighting the main factor, which was the increasing of the stability of the structure with the consequence of loosing heave and surge.
At the end we can say, both from mechanical, economical and maintenance points of view, that the moonpool design, for a simple cylindrical platform, will increase the performance of the FOWT.

6.1 *Further research*

How can we improve this project from the numerical point of view? Which other characteristics of the moonpool design can be studied in the future? How a more realistic comparison can be made based on the used moonpool design? And many other questions needs to be explained in order to bring this thesis's idea to a broader overview.

Improvements are always possible but are fundamentally related on the cluster's power and computational time, knowledge of OpenFoam, mesh software and availability of wave test facilities. An improvement category list can be made:

- **Numerical analysis**: The use of a finer mesh both in the background and overset zones, coupling of the MoorDyN code with OpenFOAM, multi-phase CFD turbulence model of a FOWT system (wind turbines positioned on the semi-submersible platform) using turbinesFoam with AMI and the definition of the wind component (velocity and direction of propagation).
- **Experimental analysis**: A precise scale model can be realised, after the CFD study, and tested in a wave laboratory in order to verify the results comparing them with the experimental data.
- **Geometrical analysis**: By varying geometrical aspects, such us the variation of the moonpool diameter, the cross-section of the moonpool (for example, the use of a square hole), variation of the height of the moonpool (dead-end hole).
- **Dynamic analysis**: introducing ballast masses to the platform, in this way, a variation of the position of the center of Mass is obtained, or by varying the inertia and/or the mass and/or change stiffness of the anchoring system and other constraints.

Bibliography

[1] JA Battjes. Surf-zone dynamics. Annual Review of Fluid Mechanics, 20(1):257–291, 1988.

[2] Subrata Kumar Chakrabarti. Hydrodynamics of offshore structures. WIT press, 1987.

[3] Thomas Choisnet, Bruno Geschier, and Geoffroy Vetrano. Initial comparison of concrete and steel hulls in the case of ideol's square ring floating substructure. In Proceedings of the 15th World Wind Energy Conference and Exhibition, Tokyo, Japan, volume 31, 2016.

[4] Antonio Colmenar-Santos, David Borge-Diez, Enrique Rosales-Asensio, and África López-Rey. Recent developments of offshore marine power in spain. RESEARCH ADVANCES IN MARINE RESOURCES, page 43, 2017.

[5] Antonio Colmenar-Santos, Javier Perera-Perez, David Borge-Diez, and Carlos dePalacio Rodríguez. Offshore wind energy: A review of the current status, challenges and future development in spain. Renewable and Sustainable Energy Reviews, 64:1–18, 2016.

[6] Alex DD Craik. The origins of water wave theory. Annu. Rev. Fluid Mech., 36:1–28, 2004.

[7] Rick Damiani and Max Franchi. An innovative second-order design method for the structural optimization of the spiderfloat offshore wind platform. Ocean Engineering, 228:108792, 2021.

[8] Luis Eça and Martin Hoekstra. A procedure for the estimation of the numerical uncertainty of cfd calculations based on grid refinement studies. Journal of computational physics, 262:104–130, 2014.

[9] Bettar Ould el Moctar, Thomas E Schellin, and Heinrich Söding. Numerical methods for seakeeping problems. Springer, 2021.

[10] John D Fenton and WD McKee. On calculating the lengths of water waves. Coastal Engineering, 14(6):499–513, 1990.

[11] Pere Frontera Pericàs. Cfd simulation of a floating wind turbine in openfoam: an fsi approach based on the actuator line and relaxation zone methods. Master's thesis, Universitat Politècnica de Catalunya, 2022.

[12] Qian Gao, Changqing Jiang, Youjun Yang, and Uwe Ritschel. Comparative study of numerical methods for predicting wave-induced motions and loads on a semisubmersible. Journal of Marine Science and Application, 22(3):499–512, 2023.

[13] Junke Guo. Simple and explicit solution of wave dispersion equation. Coastal Engineering, 45(2):71–74, 2002.

[14] John N Hunt. Direct solution of wave dispersion equation. Journal of the Waterway, Port, Coastal and Ocean Division, 105(4):457–459, 1979.

[15] Changqing Jiang. Mathematical modelling of wave-induced motions and loads on moored offshore structures. PhD thesis, Dissertation, Duisburg, Essen, Universität Duisburg-Essen, 2021, 2021.

[16] Jason M Jonkman and Denis Matha. Dynamics of offshore floating wind turbines—analysis of three concepts. Wind Energy, 14(4):557–569, 2011.

[17] Nils Karow, Leonie Kandler, Martin Brede, and Sven Grundmann. Turbulent transport of discharged ground water in oceanic bottom boundary layers in a water channel experiment. Ocean Science Discussions, 2020:1–20, 2020.

[18] Bernard Le Méhauté and Bernard Le Méhauté. An introduction to water waves. An introduction to hydrodynamics and water waves, pages 197–211, 1976.

[19] Yuanchuan Liu, Qing Xiao, Atilla Incecik, and De-cheng Wan. Investigation of the effects of platform motion on the aerodynamics of a floating offshore wind turbine. Journal of Hydrodynamics, 28(1):95–101, 2016.

[20] Kai-Tung Ma, Yong Luo, Chi-Tat Thomas Kwan, and Yongyan Wu. Mooring system engineering for offshore structures. Gulf Professional Publishing, 2019.

[21] MA Maâtoug and M Ayadi. Numerical simulation of the second-order stokes theory using finite difference method. Alexandria Engineering Journal, 55(3):3005–3013, 2016.

[22] Xuan Mei and Min Xiong. Effects of second-order hydrodynamics on the dynamic responses and fatigue damage of a 15 mw floating offshore wind turbine. Journal of Marine Science and Engineering, 9(11):1232, 2021.

[23] Daniel Micallef and Abdolrahim Rezaeiha. Floating offshore wind turbine aerodynamics: Trends and future challenges. Renewable and Sustainable Energy Reviews, 152:111696, 2021.

[24] John W Miles. Solitary waves. Annual review of fluid mechanics, 12(1):11–43, 1980.

[25] John Nicholas Newman. Marine hydrodynamics. The MIT press, 2018.

[26] Miriam Noonan, Tyler Stehly, David Fernando Mora Alvarez, Lena Kitzing, Gavin Smart, Volker Berkhout, and Yuka Kikuch. Iea wind tcp task 26: Offshore wind energy international comparative analysis. 2018.

[27] Thomas Poulsen and Rasmus Lema. Is the supply chain ready for the green transformation? the case of offshore wind logistics. Renewable and sustainable energy reviews, 73:758–771, 2017.

[28] M Rahman. Fundamentals concerning stokes waves. WIT Transactions on Engineering Sciences, 9, 1970.

[29] Amy N Robertson, Fabian Wendt, Jason M Jonkman, Wojciech Popko, Habib Dagher, Sebastien Gueydon, Jacob Qvist, Felipe Vittori, José Azcona, Emre Uzunoglu, et al. Oc5 project phase ii: validation of global loads of the deepcwind floating semisubmersible wind turbine. Energy Procedia, 137:38–57, 2017.

[30] Turgut Sarpkaya. Force on a circular cylinder in viscous oscillatory flow at low keulegan—carpenter numbers. Journal of Fluid Mechanics, 165:61–71, 1986.

[31] P Sivabalan and S Surendran. Cfd simulation of the moonpool on the total resistance of a drillship at low forward speed. Procedia engineering, 194:31–37, 2017.

[32] Fred Stern, Robert V Wilson, Hugh W Coleman, and Eric G Paterson. Comprehensive approach to verification and validation of cfd simulations—part 1: methodology and procedures. J. Fluids Eng., 123(4):793–802, 2001.

[33] James Johnston Stoker. Water waves: The mathematical theory with applications, volume 36. John Wiley & Sons, 1992.

[34] Jia-Yu Sun, Shi-Li Sun, Shu-Zheng Sun, and Hui-Long Ren. The impact of piston and sloshing motions on added resistance from moonpool configurations. Ocean Engineering, 267:113179, 2023.

[35] Lei Tan, Tomoki Ikoma, Yasuhiro Aida, and Koichi Masuda. Mean wave drift forces on a barge-type floating wind turbine platform with moonpools. 2021.

[36] Krish Thiagarajan and Javier Moreno. Wave induced effects on the hydrodynamic coefficients of an oscillating heave plate in offshore wind turbines. Journal of Marine Science and Engineering, 8(8):622, 2020.

[37] John Törnblom and Erik Hammargren. Effect of the moonpool on the total resistance of a drillship. 2012.

[38] CZ Wang and GX Wu. Time domain analysis of second-order wave diffraction by an array of vertical cylinders. Journal of Fluids and Structures, 23(4):605–631, 2007.

[39] Yu Wang, Hamn-Ching Chen, Guilherme Vaz, and Simon Burmester. Cfd simulation of semi-submersible floating offshore wind turbine under pitch decay motion. In International Conference on Offshore Mechanics and Arctic Engineering, volume 59353, page V001T01A002. American Society of Mechanical Engineers, 2019.

[40] Sophie L Ward, Peter E Robins, Matt J Lewis, Gregorio Iglesias, M Reza Hashemi, and Simon P Neill. Tidal stream resource characterisation in progressive versus standing wave systems. Applied energy, 220:274–285, 2018.

[41] Robert Wilson, Jun Shao, and Fred Stern. Discussion: Criticisms of the "correction factor" verification method. J. Fluids Eng., 126(4):704–706, 2004.

[42] Christian Windt, Josh Davidson, Pál Schmitt, and John V Ringwood. On the assessment of numerical wave makers in cfd simulations. Journal of Marine Science and Engineering, 7(2):47, 2019.

[43] Xiaoxiao Zan and Zhenhua Lin. On the applicability of morison equation to force estimation induced by internal solitary wave on circular cylinder. Ocean Engineering, 198:106966, 2020.

Appendix A

OpenFoam environment for overInterDyMFoam

In this section, a synthetic summary about the structure of OpenFoam is present, with the aim to understand which are the fundamental elements of the software and how they work together in order to solve the fundamental governing equations and produce the postProcessing results, results that help us to analyze the behavior of the semi-submersible platform anchored to the bottom of the sea.

A.1 *Structure*

The solver ***overInterDyMFoam*** is applied for two incompressible (assuming air as a incompressible fluid), isotherms fluids, using the VOF phase-fraction interface capturing with optional mesh motion and mesh topology changes, including adaptive re-meshing [see overInterDyMFoam].

From Figure A.7, at the end of the Appendix's chapter,a complete view of each folder and file for each case study is displayed (from the OC5 modelling to the different moonpool cases),which it will permit us to acknowledge how the software organize the files to guarantee a precise cooperation between them; it's important to underline that, in this thesis, the C++ environment behind the whole composition is not shown in order to avoid a too detailed explanation on the software itself, which may divert from the true objective of the paper; if the reader is particularly interested on the scripts behind the different codes, he can refer to this guide [see OpenFoam.guide]. The CASE folder structure, defined by the user, is divided into 2 main parts:

- **background**: This folder is related to the background environment (sea and atmosphere) that reproduce the sea waves and it contains the overset block. Inside this folder, important features of the simulation are defined, such us the rigid body properties of the platform, wave characteristics and the boundary conditions.
- **overset**: This part, instead, is a moving mesh block coordinately with the platform, in order

to ensure a precise calculation of the Navier-stokes equation near the sea structure.

Both of these two elements contains the *constant* and *system* directories, and now we'll realize why they are so important. The *constant* directory hold the properties (like the `polyMesh`, and in the case of the background folder, the flow's and the platform characteristics) that will remain the same during all the simulation time. The *system* defines the way at which the program solve the differential equations. The output data of the process will be automatically updated, based on the specific time interval, in the `postProcessing` directory. Of course each file follow a syntax rule that has been initially defined in the `dictionaries` files.

A.2 *Mesh*

We refer with the word "mesh" as the grid or structure of data that define the geometrical environment of the CFD simulation where the differential equations are solved. The mesh plays an essential part on the discretization of the domain, so the way at which we build it, determine the precision at which the equations will be solved by the solver. The developer of OpenFoam have designed the in-built utilities that permit to design the mesh of different shapes of cells like hexahedrons, wedges, and prisms, without the use of external software; since it hasn't a visual straightforward interface it's quite complex to design one, especially for beginners.

However, the design of the mesh has to follow particular rules, these are well explained here [see blockMesh.guide] (like the right-end rule for the normal surface's orientation) as we can see in A.1, otherwise errors may occurs. The two main mesh utilities used in the case study are:

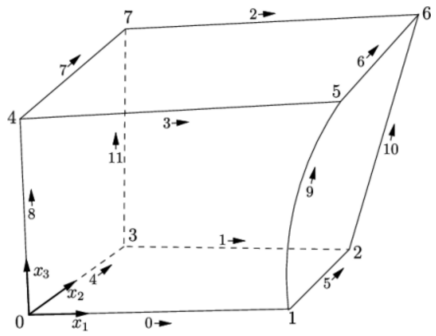


Figure A.1: Block's orientation (Reproduced from the OpenFoam guide)

blockMeshDict $\longrightarrow$ This file, placed in both background and overset folders, define the hexahedral blocks domains with straight and curve edges and the number of cells in each direction with their respective cell expansion ratio. Nevertheless, it let the user to denominate each domain's surface, recognised as `patches` (where the boundary and initial conditions of the fluid mechanics equations will be applied) by the constant folder in the program. Since, in our case, we require two meshes working together, we need to merge the surfaces and so, the patches, of both blocks

(background and overset) into a domain that simulates the computations as a single interconnected system; this process can be done by creating an utility called `mergeMeshDict` and by directly writing in the terminal the command "`mergeMeshes . ../overset -overwrite`" (see Figure A.2).

A very interesting way to design a mesh that presents dimensions that are based on external characteristics that may change, can be the **parametric mesh**, where basically the dimensions of the domain and the so the hexahedral cells depends on the variable that we set at the beginning of the code (i.e. the wave lenght for horizontal dimension and wave height for vertical dimensions).

snappyHexMeshDict $\longrightarrow$ The domain is partitioned into hexahedra and split-hexahedra built on surface geometries in Stereolithography (STL) format, which before has to be extracted in the `extendedFeatureEdgeMesh`, using .emesh format, by `surfaceFeatureExtractDict`. This is achieved through an iterative process that involves refining an initial mesh (typically generated using blockMesh) and subsequently adapting it to the STL surface (in the Figure A.7, it's defined as geometry.stl).

The latter surface is generated using a 3D CAD program called FreeCAD[1] and import it in the Open Foam's folders `triSurface` (using the MeshLab software, where a de-codification of the geometry file from binary to STL is made) that, in our case, it's placed in the overset folder due to the fact that the platform mesh (that we can see from the Figure A.3) is inserted inside the overset's blockMesh.

In the Figure A.3, we can even notice that the mesh is refined precisely to the borders of the structure because they're the most weak parts of the domain, which means that it's very probable to have errors in these parts of the mesh if the refinement isn't enough accurate. All the specific characteristics of the .stl cells are stated in the `snappyHexMeshDict` file, which contains:

- `castellatedMeshControls`: Feature used to control and manage the mesh generation process.
- `refinementSurfaces` and `refinementRegions`: They are key components defined as the surfaces and volumes where we want to apply a specific refinement of the cells grid.
- `snapControls` and `addLayersControls`: The first one control the snapping process (it's the automatic adjustment of the mesh vertices to conform the surface of the STL geometry), while the second one, set the boundary layers added to the boundary surfaces regarding the near-wall flow phenomena.
- `meshQualityControls`: It ensure both geometrical and numerical characteristics (like orthogonality, cell quality, skewness etc..) of the generated mesh to achieve accurate results.

By typing "`checkMesh`" into the terminal, a mesh validity control based on the patch topology is made, in order to look at the main statistics of the grid just created (type and number of cells, surfaces) that represents the quality of the mesh; remembering that the higher is the refinement (i.e. number of cells) the more precise will be the calculations performed but the higher will be the

[1] FreeCAD is an open source 3D parametric modeling application, not designed for one particular type of work but rather intended for a wide range of uses. Being an open source and multi platform, it benefits from the efforts and contributions of enthusiasts and programmers from all over the world.

computational power needed by the cluster, so a more computational time is needed. If very precise results are required, the cell's mesh distribution refinement has to be set in order to have a finer mesh only where we want to have more accurate results, in our case in the overset zone and in the wave region to a good prediction of the wave elevation, so progressively bigger blocks far from them.

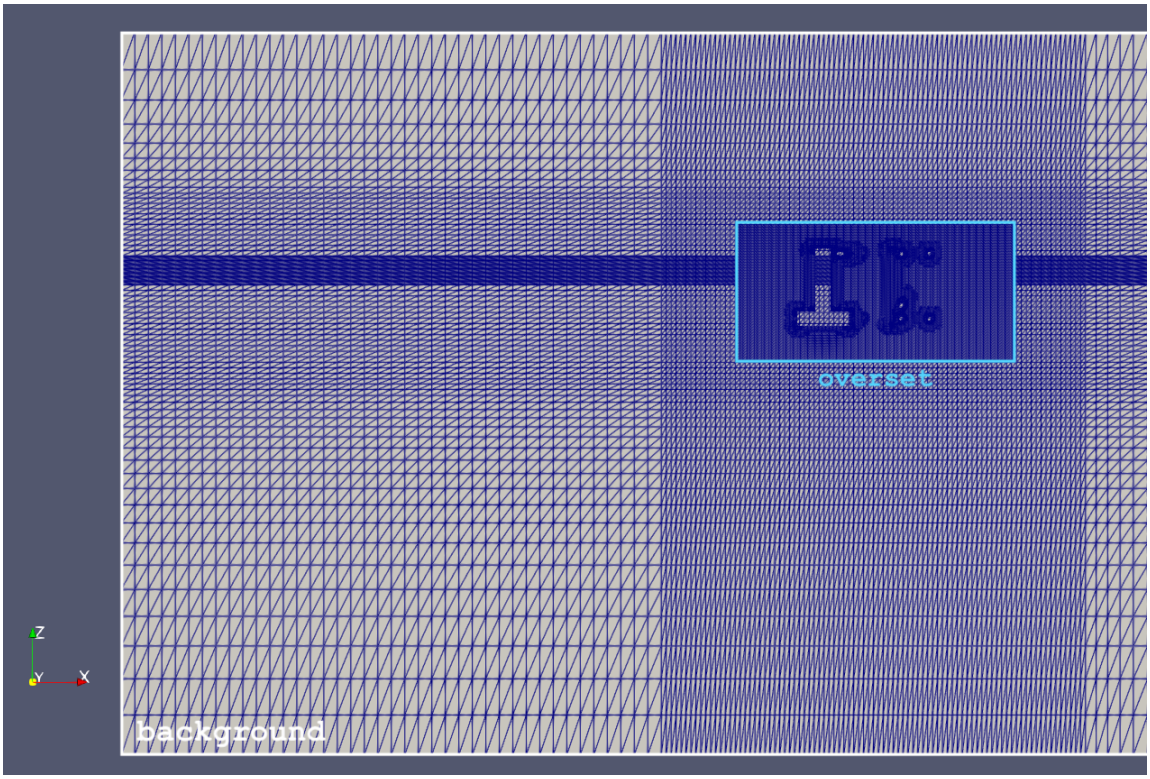


Figure A.2: Slice of the merge domain of the OC5 test model

A.3 *Wave generation*

OpenFOAM provides various options for simulating waves, including linear wave theory, Stokes waves, and higher-orders wave models. In the **waveProperties** file, placed inside the *constant* folder, the characteristic of the waves such us:

- `waveModel`
- `waveHeight`, `wavePeriod` and `waveAngle`
- `rampTime`[2]

[2]It's the time at which, after the start of the simulation, the program will generate the wave from the inlet surface.

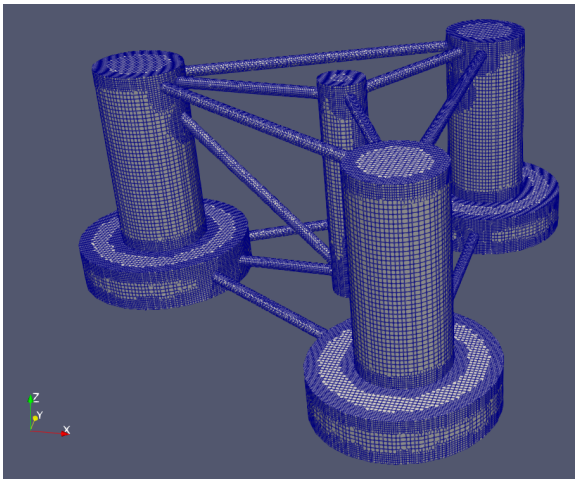


Figure A.3: OC5 platform mesh

they are defined in the inlet patch, while, in the outlet one, an absorption model is selected to avoid boundary reflections, as we mimicking a infinite background domain (just like a sea) with a reduced domain size (see Figure A.4). The software present different types of waves that can be generated,

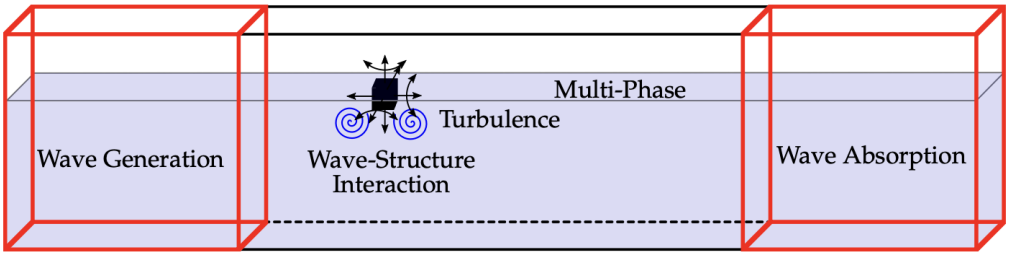


Figure A.4: Generic CFD-based numerical wave tank schematic [42]

such as Cnoidal, Stokes I, II and V, based on the sea conditions (see Figure 2.5).

A.4 *Numerical discretization and solution algorithm control schemes*

The numerical schemes used to discretize and solve the CFD problem are defined in the **fvSchemes** file; OpenFoam gives the user the freedom to choose different solvers to compute the differential equations by mapping them into a discrete domain (see 3.3.2), achieving systems of linear algebraic equations. Here, in this table, we'll see how each component of the differential equation can be

discretized:

Table A.1: Description of different elements in the fvSchemes file

Numerical scheme	Description
`interpolationSchemes`	Point-to-point interpolations of values
`snGradSchemes`	Gradient component normal to the cell face
`gradSchemes`	Gradient (∇)
`divSchemes`	Divergence ($\nabla\cdot$)
`laplacianSchemes`	Laplacian (∇^2)
`timeSchemes`	Time derivatives of the first and second order ($\frac{\partial}{\partial t}$, $\frac{\partial^2}{\partial t^2}$)

Each of the numerical schemes can be discretize in different ways based on the application, for example, the `interpolationSchemes` can be represented with different method,s such as `default linear`, `default upwind phi` (where `phi` is the velocity flux ϕ) and `default limitedLinear phi 1.0`. The equations solvers algorithm and the tolerances of the residuals are controlled by the **fvSolution** file, placed with the **fvSchemes** in the *system* directory. A first way to solve the system's equation is to use an iterative solver and setting the residuals for each unknown variable, so the solver algorithm (in our case the PIMPLE algorithm, see 3.3.3) will iterate until the tolerance criterion is fulfilled; another way to proceed is, instead, by the definition of the maximum number of iterations desired. Before running the algorithm, Open Foam impose the user to set the preconditioners in order to avoid too many iterations.

Since the understanding of this chapter required some advanced numerical mathematical notions (especially regarding the **fvSolution** part) and this will go well beyond the thesis objective, for a more detailed view, we suggest the reader to read the Chapter 6 of the OpenFoam guide [see Chapter6.guide].

A.5 *Initial field values*

To set or initialize the field values within the mesh domain, before the beginning of the computations, the user needs to run the **setFieldsDict** utility, utility that, in this specific study case, will divide the background mesh into two types of fluids (based to the scalar parameter α) with different physical properties such us density and viscosity. Here:

Listing A.1: setFieldsDict.txt code

```
defaultFieldValues
(
    volScalarFieldValue alpha.water 0
    volScalarFieldValue zoneID 123
);

regions
```

```
(
    boxToCell
    {
        box ( -1000 -1000 -1000 ) ( 1000 1000 0 );
        fieldValues
        (
            volScalarFieldValue alpha.water 1
        );
    }

    boxToFace
    {
        box ( -1000 -1000 -1000 ) ( 1000 1000 0 );
        fieldValues
        (
            volScalarFieldValue alpha.water 1
        );
    }

    cellToCell
    {
        set c0;

        fieldValues
        (
            volScalarFieldValue zoneID 0
        );
    }

    cellToCell
    {
        set c1;

        fieldValues
        (
            volScalarFieldValue zoneID 1
        );
    }
);
```

However, to apply the "`setFields`" command in the terminal, it's necessary to identify and generate cells sets (like `c0` and `c1`) based on topological criteria. Each of these cells sets will contains a number of cells that will share common properties and/or boundary conditions. To make this topological selection happened we need to define the **topoSetDict** folder (placed with the setFields one in the *system* directory):

Listing A.2: topoSetDict.txt code

```
actions
(
    {
        name    c0;
        type    cellSet;
        action  new;
        source  regionToCell;
        insidePoints ((10 1 1));
    }

    {
        name    c1;
        type    cellSet;
        action  new;
        source  cellToCell;
        set     c0;
    }

    {
        name    c1;
        type    cellSet;
        action  invert;
    }
);
```

A.5.1 *Boundary conditions*

To solve the partial differential equations, using the FV method, it's necessary to set the boundary conditions (BCs) appropriate for the case study and it's crucial to couple each patch surface, wall or element with its corresponding boundary condition. The software presents more than 70 BCs, which are divided in 3 categories that are *basic*, *constraints* (related to geometrical restrictions) and *derived* (for specific conditions such as inlet, outlet and wall patches) present in the `0.orig` folder and then assigned to each patch in the `0` directory after the generation of the mesh (see Figure A.5).

The more complex the case is, the more specific boundary conditions are needed, for example, the turbulent cases required the setting of the k, omega, nut BCs for each patch, they will be added to the laminar ones. In the case of a symmetrical block, the symmetry patch for the side surfaces is applied in order that, all the property applied to that surface, will be automatically copied to the other one.

Listing A.3: part of blockMeshDict.txt code

```
boundary
(
    oversetPatch
    {
        type overset;
```

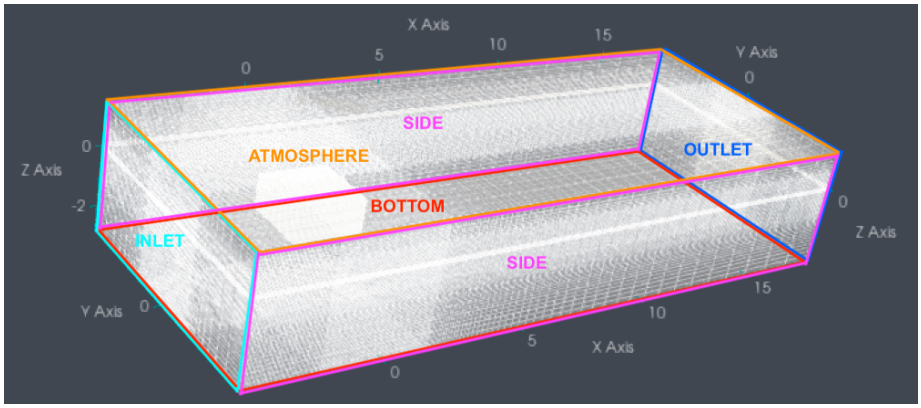


Figure A.5: Study case boundary background surfaces

```
    faces ();
}

sides
{
    type symmetry;
    faces
    (
        (1 5 4 0)
        (3 7 6 2)
    );
}

bottom
{
    type wall;
    faces
    (
        (0 3 2 1)
    );
}

atmosphere
{
    type patch;
    faces
    (
        (4 5 6 7)
    );
}
```

```
    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }

     outlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
);
```

A.6 *Case flow chart*

In this section, there will be presented the case flow chart that, after the brief introduction of the software's functionalities in the previous pages, will permit the reader to understand what are the temporal steps that have been followed in order to simulate the project case with OpenFoam and studied the post processing results (charts of the main variables using *gnuplot* and space-time distribution of the resulting variables from ParaView). The scheme in the Figure A.6 will give a clear view on how to operate with the used CFD computational tool: and to conclude, the CFD software's directories structure is presented in the Figure A.7.

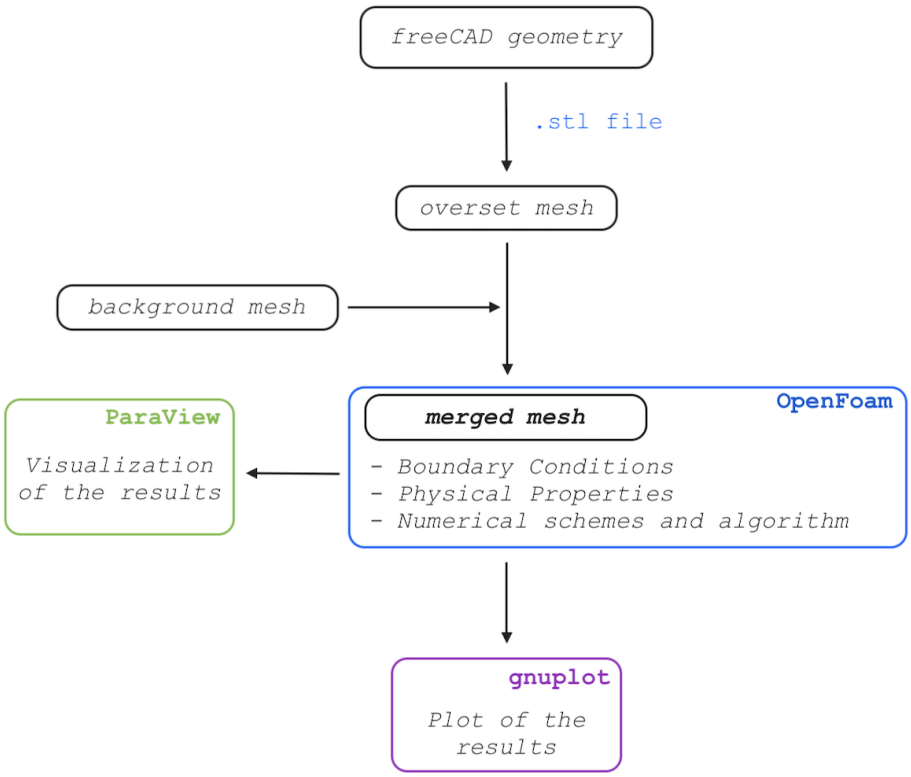


Figure A.6: OpenFoam case flow chart

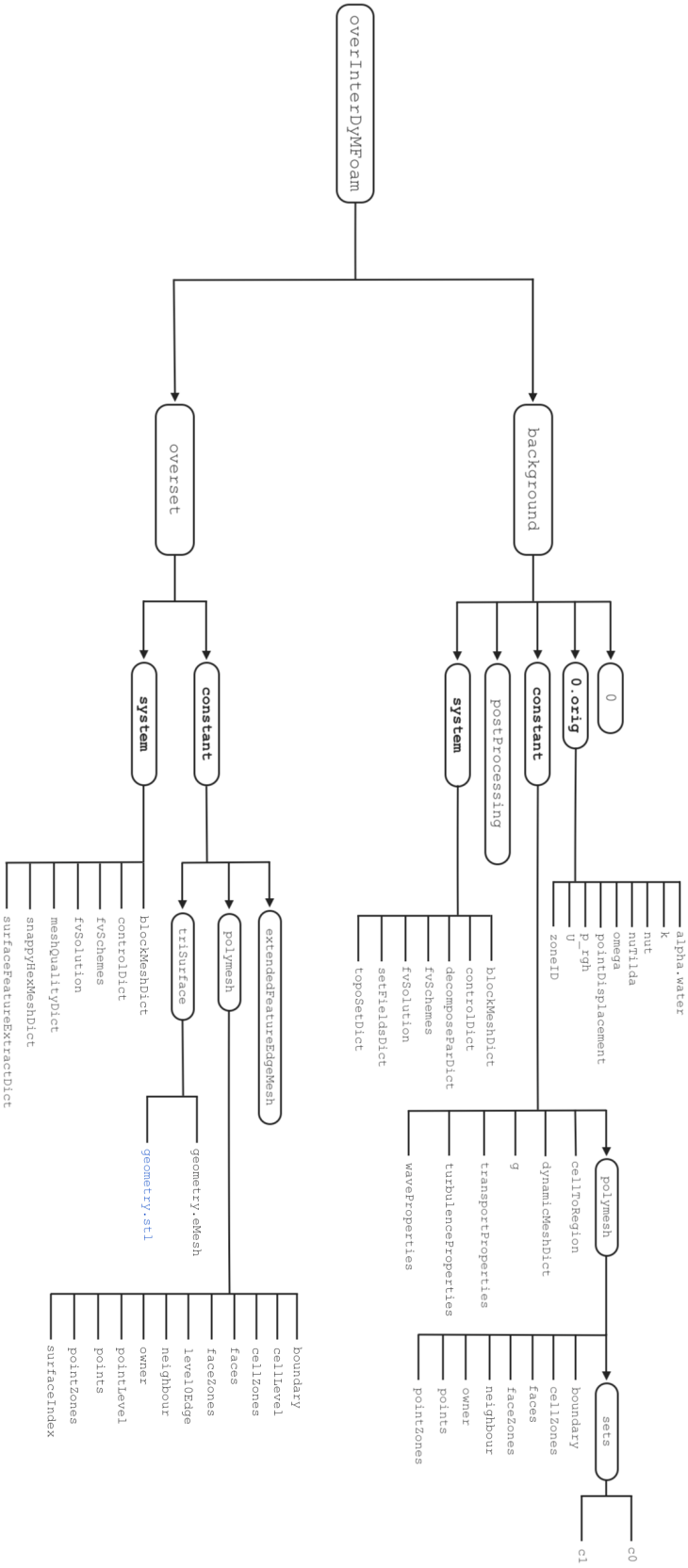


Figure A.7: Case directories structure