Validation and Application of a Fully Nonlinear Numerical Wave Tank

Thesis

N. Bruinsma

Author contact information
Niek Bruinsma
Phone: +31 6 54903169
Email: niek.bruinsma@gmail.com

Delft University of Technology
Validation and Application of a Fully Nonlinear Numerical Wave Tank

Thesis

by

N. Bruinsma

Master of Science

in Offshore and Dredging Engineering

at the Delft University of Technology.

Student: Niek Bruinsma
University supervisor: Prof. R. H. M. Huijsmans
Dr. ir. I. Akkerman
Dr. ir. M. Zijlema
Dr. ir. G. H. Keetels
Company supervisor: Dr. ir. B. T. Paulsen
The present master thesis is submitted as part of the requirement for obtaining a master of science degree in Offshore Engineering at the Technical University of Delft. The work has been conducted at Deltares, within the department of Harbour, Coastal and Offshore Engineering, under supervision of Dr. ir. Bo T. Paulsen. Supervision and guidance from the Technical University of Delft was provided by Prof. René H. M. Huijsmans, Dr. ir. Ido Akkerman, Dr. ir. Marcel Zijlema and Dr. ir. Geert H. Keetels. This thesis was performed as part of the 'TO2-Floating Wind' project, and the success of the presented work was stimulated by the collaboration of ECN, MARIN, NLR and Deltares.

I would like to express my gratitude towards Deltares for hosting me during my research, and MARIN for providing the experimental data essential for the success of the of this work. I am grateful for the guidance and supervision from my supervisors from the Technical University of Delft, but in particular for the unlimited support provided by Dr. ir. Bo T. Paulsen, who, to a great extent, is responsible for the success of this work. Finally, I would also like to express my gratitude towards Dr. ir. Niels G. Jacobsen for his support and stimulating discussions throughout the project.

N. Bruinsma
Delft, March 2016
The present thesis considers numerical computations of fully nonlinear fluid-structure interaction. The aim of the thesis is to establish a well validated fully nonlinear numerical wave tank for the simulations of complex fluid-structure interaction of moored floating offshore structures.

The numerical computations are carried out using the fully nonlinear numerical fluid-structure interaction solvers, interFoam and interDyM Foam, from the open-source CFD library OpenFOAM®. These solvers make use of a fully nonlinear Navier-Stokes/VOF solver for the computations of the two-phase flow field, where the interDyM Foam solver also utilises a 6-DOF motion solver to compute the motions of the floating structure. These solvers were extended with waves2Foam, a wave generation and absorption toolbox, developed by Jacobsen et al. (2012). The extended interFoam and interDyM Foam solvers, hereafter referred to as the waveFoam and interDyM Foam solver, were utilised for the computations of fluid-structure with fixed and moving structures respectively. Furthermore, an implementation of catenary mooring lines is provided by Niels Jacobsen (Researcher/advisor at Deltares), for the simulations of the mooring system of the floating structure. Finally, the fully nonlinear potential flow solver, OceanWave3D, was utilised in a fully nonlinear and fully parallelised domain decomposed solver, developed by Paulsen (2013) and Paulsen et al. (2014b), for the efficient computations of realistic sea states. Here, the outer wave field is described by the potential flow solver, whereas the inner wave field, in the vicinity of a given structure, is described by the interDyM Foam solver.

The waveFoam and waveDyM Foam solvers are carefully validated either in terms of convergence by grid refinement or by comparisons to experimental measurements. Special attention is paid to the waveDyM Foam solver with respect to the computation of the flow-induced motions of a moored floating wind turbine. The ability of the numerical model to accurately reproduce experiments, performed with the generic OC5 floating wind turbine model in the MARIN Concept Basin, is also investigated.

The Navier-Stokes/VOF based waveFoam and waveDyM Foam solvers were evaluated with respect to wave propagation and wave structure interaction in a two-dimensional set-up. A grid convergence study was performed on the propagation of a fully nonlinear stream function wave. The convergence rate of the two-phase numerical solution to the single-phase stream function solution was verified. Successful computation of wave loading on a partially submerged fixed horizontal cylinder was provided. The accuracy of the numerical model, with respect to fluid-structure interaction for free and forced motion of a structure, was verified with the generation of surface waves by the forced oscillation and the free heave decay of a horizontal cylinder.

These two-dimensional cases were also used to verify the efficiency of two meshing tools, provided by the OpenFOAM® toolbox. These meshing tools were shown to provide excellent results, and more importantly, provided a less complicated method for generating surface boundary meshes of complex three-dimensional structures.

The waveDyM Foam solver is used for the computation of free and moored decay test of
the three-dimensional floating wind turbine model. The accuracy of the numerical solution was verified against numerical computations from the work of Dunbar et al. (2015) and physical experiments performed at MARIN.

Finally a proof-of-concept case is performed, involving the modelling of a three-dimensional moored floating wind turbine subjected to irregular uni-directional waves. In these numerical computations the potential of the waveDyMFoam and the domain decomposition strategy are evaluated.
## Contents

1 Introduction .............................. 1
   1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
      1.1.1 Wind energy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
      1.1.2 Floating wind energy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
      1.1.3 Design tools . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
      1.1.4 International collaboration . . . . . . . . . . . . . . . . . . . . . . . . . . 3
      1.1.5 National collaboration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
      1.1.6 Hydrodynamical tools . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   1.2 Free surface waves . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
      1.2.1 Wave parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      1.2.2 Small linear waves . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      1.2.3 Steep nonlinear waves . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
      1.2.4 Wave loads on a surface piercing cylinders . . . . . . . . . . . . . . . . . . . 8
      1.2.5 Wave interaction with a floating body . . . . . . . . . . . . . . . . . . . . . 10
   1.3 Numerical simulation of wave-structure interaction . . . . . . . . . . . . . . . . . . . 11
      1.3.1 Flow theories . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
      1.3.2 Navier-Stokes-type models . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
      1.3.3 Navier-Stokes/VOF solvers . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
      1.3.4 Efficient domain decomposition . . . . . . . . . . . . . . . . . . . . . . . . . 13
   1.4 This Thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   1.5 Structure of this thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2 The numerical models ........................... 17
   2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
   2.2 InterFoam/waveFoam solver . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      2.2.1 Governing equations Navier-Stokes/VOF . . . . . . . . . . . . . . . . . . . . 18
      2.2.2 Boundary conditions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
      2.2.3 Relaxation zones . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
      2.2.4 Algorithm waveFoam . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
   2.3 InterDyMFoam/waveDyMFoam solver . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
      2.3.1 Motions of a body . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
      2.3.2 Under-relaxation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
      2.3.3 Dynamic mesh handling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
      2.3.4 Rotations and translations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
      2.3.5 Restraints for moving boundary surfaces . . . . . . . . . . . . . . . . . . . . 26
      2.3.6 Algorithm waveDyMFoam . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
   2.4 Fully nonlinear potential flow solver . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
      2.4.1 Governing equations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
      2.4.2 Algorithm waveDyMFoam / potential flow . . . . . . . . . . . . . . . . . . . 31
## 3 Wave propagation and fluid-structure interaction in 2D

3.1 Introduction .................................................................................................................. 33
3.2 Convergence study on the propagation of a fully nonlinear regular wave .................. 33
  3.2.1 Numerical set-up ....................................................................................................... 34
  3.2.2 Analysis convergence study ..................................................................................... 35
3.3 Wave loading on a partially submerged fixed horizontal cylinder .............................. 40
  3.3.1 Numerical set-up ....................................................................................................... 40
  3.3.2 Analysis wave loading ............................................................................................. 41
3.4 Surface waves generated by the forced oscillation of a horizontal cylinder ............... 47
  3.4.1 Numerical set-up ....................................................................................................... 47
  3.4.2 Analysis wave generation ......................................................................................... 48
3.5 Free heave decay of a horizontal cylinder .................................................................. 53
  3.5.1 Numerical set-up ....................................................................................................... 53
  3.5.2 Analysis motion decay ............................................................................................. 55
3.6 Summary ......................................................................................................................... 58

## 4 Meshing tools

4.1 Introduction ...................................................................................................................... 61
4.2 SnappyHexMesh ............................................................................................................. 62
  4.2.1 Wave loading on a partially submerged fixed horizontal cylinder ......................... 63
  4.2.2 Surface waves generated by the forced oscillation of a horizontal cylinder .......... 64
  4.2.3 Free heave decay of a horizontal cylinder ................................................................. 65
4.3 Multi-mesh grading ........................................................................................................ 66
  4.3.1 Wave loading on a horizontally fixed cylinder ......................................................... 67
  4.3.2 Surface waves generated by a forced heaving cylinder ........................................... 68
  4.3.3 Decay of heave motion of a free horizontal cylinder .............................................. 68
4.4 Summary ......................................................................................................................... 69

## 5 Fluid-structure interaction in 3D

5.1 Introduction ...................................................................................................................... 71
5.2 TO2 floating wind project ............................................................................................. 71
5.3 Physical experiments ...................................................................................................... 72
  5.3.1 Physical wave tank ..................................................................................................... 72
  5.3.2 OC5 floating offshore wind turbine model ............................................................. 73
  5.3.3 Experiments performed at MARIN .......................................................................... 73
5.4 Free heave, pitch and roll decay of a floating structure .............................................. 74
  5.4.1 Numerical set-up ....................................................................................................... 74
  5.4.2 Analysis of the free decay simulations .................................................................... 77
5.5 Motion decay of a floating structure with restraints .................................................... 81
  5.5.1 Numerical set-up ....................................................................................................... 81
  5.5.2 Analysis of the moored decay simulations ............................................................... 83
5.6 Summary ......................................................................................................................... 87

## 6 Proof-of-concept

6.1 Wave propagation using OceanWave3D ...................................................................... 89
  6.1.1 Numerical set-up ....................................................................................................... 90
  6.1.2 Analysis of wave propagation using OceanWave3D .............................................. 91
## Contents

6.2 Moored floating structure subjected to irregular waves .............................. 94  
   6.2.1 Numerical set-up .............................................................................. 94  
   6.2.2 Analysis moored floating structure subjected to irregular waves ...... 95  
6.3 Summary ......................................................................................... 99  

7 Conclusions and recommendations .......................................................... 103  
   7.1 Discussion ....................................................................................... 103  
      7.1.1 A converging solution ................................................................. 104  
      7.1.2 Propagation of a fully nonlinear wave ....................................... 104  
      7.1.3 Wave loads on a structure ......................................................... 105  
      7.1.4 Motions of a structure ............................................................... 105  
      7.1.5 Advanced meshing tools ............................................................ 106  
      7.1.6 Fully nonlinear numerical wave tank ......................................... 107  
   7.2 Conclusions ..................................................................................... 108  
   7.3 Further developments ....................................................................... 108  
   7.4 Future research ............................................................................... 109  

A WaveFoam code ..................................................................................... 111  
B WaveDyMFoam code ............................................................................. 115  
C SixDoFRigidBodyMotion code .................................................................. 119  
D Floating wind turbine model ................................................................... 127  

Bibliography .............................................................................................. 131
1

INTRODUCTION

1.1. BACKGROUND

Scientific understanding of global warming and the major environmental risk it poses is increasing and the notion that global warming is caused by increasing concentrations of greenhouse gases and other human activities has been widely recognized. Major industrialised nations are working together on possible societal responses, the most apparent being mitigation by emissions reduction. Part of this mitigation strategy is to change the world’s energy supply from being predominately dependent on fossil fuels to the use of low-carbon energy technology such as nuclear-, solar-, thermal-, hydro- and wind energy.

Recently the leaders of the European Union (EU) agreed on "The 2020 Climate and Energy Package", a set of binding legislation which aims to ensure the EU meets its ambitious climate and energy targets for 2020. These targets set three key objectives for 2020: a 20 % reduction in greenhouse gas emissions compared to the 1990 levels, 20 % energy consumption produced from renewable resources and a 20 % improvement in the EU’s energy efficiency (Moccia et al., 2014). It may be noted that the environmental risk is not the only reason for the EU’s increased interest in alternative energy sources. In recent years the relationship with Russia seems to deteriorate, resulting in an accumulated desire to be less dependent on Russian oil and gas. All this has stimulated investments in alternative energy sources.

1.1.1. WIND ENERGY

It is expected that wind energy will be a major contributor to the increase of renewable energy production as this technology is highly suitable for wide implementation throughout the EU. According to the European Wind Energy Association (EWEA) the total installed wind energy capacity in the EU amounted to 117.3 GW by the end of 2013. The majority of this capacity, 110.7 GW, was contributed by onshore wind farms and the remaining 6.6 GW came from offshore wind farms. That same year, 32 % of all the newly installed capacity was attributed to wind energy. This made wind power the fastest growing energy technology in terms of capacity growth. The wind energy capacity installed by the end of 2013 would, in a normal wind year produce 257 TWh of electricity, which is enough to cover 8 % of the electricity consumption of the EU, this is an increase of 1 % compared to 2012. EWEA predicts the combined wind energy production (offshore and onshore) to cover 14.9 % of the total EU power demand. This increased capacity will to some extent be achieved by replacing active
wind turbines by newer, larger and more efficient models, but for the greater part by the installation of new wind farms (Pineda et al., 2014).

Offshore wind energy installations are expected to produce approximately 2.9% of the total EU energy demand by 2020. This projected growth of offshore wind energy corresponds to the growth witnessed in the onshore wind sector at a similar time in that industry’s development. Onshore wind energy deployment picked up speed in the mid-1990s. With a 15 year difference, presently, the offshore wind seems to be following a similar growth path. The foreseen growth of the offshore wind sector will move it from being an emerging, immature technology to a key component of the EU’s energy mix (Arapogianni et al., 2011). As the technology develops, the offshore wind industry is moving into deeper waters, further from the coast with bigger farms. This reflects maritime spatial planning and wind farm developers desire to harness better energy resources further out at sea.

1.1.2. FLOATING WIND ENERGY

Although currently ongoing commercial scale offshore wind developments are based on concepts with fixed substructures, many countries have limited suitable sites in sufficiently shallow water to economically install wind turbines on monopile and jacket foundations, see Figure 1.1. A floating offshore wind turbine (FOWT) concept was introduced to harvest the vast energy resources found in the deeper waters. This concept has existed since the early 1970s, however, first full size test cases only initiated in 2008 (Arapogianni et al., 2013). The most apparent advantage of floating offshore wind over conventional offshore wind, is the possibility access the vast wind resources, located at deeper sites further out at see. Furthermore, since a FOWT will generally be located further from the coasts, the visual and noise impact to inhabited coastal areas will also be lower. On top of that, the relatively simple installation process is another major upside to this technology. The installation of the conventional offshore wind relies heavily on expensive special purpose ships for transport and on-site assembly and installation, while a FOWT can be assembled in port and towed into position using simple tugboats, whereupon the mooring system and electric grid can be connected.

Currently several FOWT concepts are actively undergoing full-scale prototype testing. The “Hywind”, developed by Statoil, was installed in the North Sea, just off the Norwegian coast and has been running successfully since 2009. The spar type structure with a 2.3 MW...
1.1. Background

A turbine was towed 10 km offshore, where it was installed in 220 m deep water. Statoil plans to build the first floating wind farm off the Scottish coast. The 30 MW pilot project, which should be commissioned by the end of 2017, will consist of five, 6 MW floating wind turbines operating in waters exceeding 100 m of water depth. The "WindFloat", developed by the joint-venture under the name of Windplus, was installed off the coast of North Portugal in 2011. This semi-submersible tri-floater has been successfully tested in water depths of approximately 45 m. Based on their sub-structure and mooring system, these, and other equivalent, FOWT concepts can be categorised into three main designs: the semi-submersible, the tension leg platform and the spar buoy. These designs, presented in Figure 1.1, were adapted from the offshore oil and gas industry.

1.1.3. Design tools

Offshore floating wind is seen as a promising solution to harvest the deep offshore wind resources. However booming, the offshore wind sector is constrained by high cost and relatively small economical margins compared to the offshore oil and gas industry. Reliable and purpose oriented design methods are essential to make this technology more cost efficient.

Some of the current engineering tools that are used to design offshore wind systems are based on design tools developed for land-based wind turbines. These design tools use codes that incorporate aerodynamic, control system, and structural-dynamic models. However, because offshore wind turbines are installed in marine environments, they require modelling tools that can also consider the hydrodynamic loads on a variety of substructure types as well as their mooring systems. To design these systems, engineers need engineering tools that can simulate incident waves, sea current, hydrodynamics, dynamics of the substructure and mooring system. The complex nature of these tools emphasises the need to verify and validate their accuracy.

1.1.4. International collaboration

The Offshore Code Comparison (OC3) project, part of the International Energy Agency (IEA) Wind Task 23, 2005-2009, was the first in a series of international collaboration projects that focused on the validation of offshore wind modelling tools (Jonkman and Musial, 2010). The two-phased Offshore Code Comparison Collaboration Continuation (OC4) project, IEA Wind Task 30, 2010-2013, was a continuation of the OC3 project (Jonkman et al., 2012). This IEA Wind Task 30 is currently extended with the Offshore Code Comparison Collaboration, Continued, with Correlation (OC5). These projects aim to verify the accuracy of complex engineering tools, utilised for the design of offshore wind systems, through code-to-code and code-to-data comparisons.

The Offshore Code Comparison Collaboration projects involve the modelling of fixed and floating wind systems. The first phase of the OC4 project involved the testing of three different generic floating wind turbine designs in a wave basin at Marine Research Institute Netherlands (MARIN) (Koo et al., 2014). To cover the main design concepts for the offshore floating wind industry these three design concepts included a spar, a tension leg platform and a semi-submersible, see Figure 1.1. The second phase of the OC4 project and the currently ongoing OC5 project concentrated on further development and validation of design codes using the semi-submersible FOWT.

The most recent semi-submersible FOWT design, the OC5, presented in Figure 1.2, features a 5MW horizontal-axis wind turbine with pitch controlled blades, developed by the Na-
1. Introduction

Figure 1.2: Schematic representation of the OC5 FOWT, a semi-submersible floating wind system consisting of a 5MW horizontal-axis wind turbine with pitch controlled blades mounted onto a semi-submersible try-floater, which is moored to the seabed using three catenary mooring lines.

The OC5 FOWT design was also utilised for a joint research project aimed at increasing the knowledge on floating offshore wind turbines. During 2015, four Dutch research institutes, ECN, MARIN, NLR and Deltares, worked together on this government subsidised project, referred to as the ‘TO2-Floating Wind’ project. A more detailed description of the project as well as the model can be found in Chapter 5.

The present research was conducted as part of the contribution from Deltares to this TO2-Floating Wind project. The main focus of Deltares was to extend the knowledge concerning numerical simulation of the motion of floating wind turbines and the calculation of loads in the mooring system. The project strived for code-to-data analysis, which was partly accommodated by physical model tests performed at MARIN.

1.1.6. Hydrodynamical Tools

At the moment no large body of knowledge exists of the behaviour of floating wind systems, hence there is still much work to be undertaken to improve their performance. Modelling such a system, especially in cases where the flow is more violent, can help to understand its behaviour in real seas.

Conventional hydrodynamical engineering tools, as used in the oil and gas industry, are often developed for designing floating structures which are installed in water depths of over 300 m. At these larger water depths linear wave theory may be sufficient to describe wave profiles and determine wave loading. However, the offshore wind industry initially intends
1.2. Free Surface Waves

Figure 1.3: Schematic representation of types of waves which can cause loading impacts on a floating wind turbine. (a) Linear waves, (b) Steep nonlinear waves and (c) highly nonlinear breaking waves.

To install floating structures at intermediate water depths, from 50 m to 200 m. Here, the waves can become strongly nonlinear in case of extreme weather conditions. Utilising linear or second-order wave theory is not sufficient to provide a realistic description of the wave profile and lack accuracy in terms of the determination of extreme wave loads.

Physical model testing is at present the only reliable method for investigating the strongly nonlinear hydrodynamical flow effects that occur during these extreme weather events. Constructing and testing a model is expensive, let alone conducting an intensive model campaign with multiple models subjected to a variety of wave conditions. It is often not possible, or highly expensive, to make adjustments to the model or the measurement equipment during these experiments and the number of sea states that can be tested is restricted by the time available in the test basin, thus of high influence to the total cost of the model campaign. Furthermore, only local measurements of the surface elevation, flow velocities and pressures can be acquired since they depend on the physically installed measurement equipment.

Computational Fluid Dynamics (CFD), which has been used as a successful design tool in many other areas of engineering, could be an alternative approach for modelling of a FOWT. It is a numerical method for the calculation of the nonlinear differential equations describing a fluid flow. One of the major benefits of numerical modelling over scale model experiments are the flexibility in the set-up of design cases, last minute design changes or changes to the environmental conditions. Also, since the modelling takes place in a numerical domain, it is possible to take measurements of the free surface, flow velocities and pressures at any given point in space and time. Furthermore, by using nonlinear equations for describing the air and water flow, CFD allows for the application of a much wider range of wave models, that can actually model higher order nonlinear effects. However, in order to use CFD as a reliable design tool for floating offshore structures, the accuracy of predictions, numerical convergence and computational efficiency need to be validated.

1.2. Free Surface Waves

To get a better understanding of the processes involved in fluid-structure interaction, this section will address different types of free surface waves, their measure of nonlinearity, the possible load effects they can have on a structure and the available wave theories that can be used to describe them. In the present work, waves are categorised as small linear waves, steep nonlinear waves and highly nonlinear breaking waves. A schematic representation of these three types of waves is presented in Figure 1.3.
1. INTRODUCTION

Figure 1.4: Schematic representation of a steep wave in a regular wave train (Fenton, 1990). Here, the wave length is denoted by $\lambda$ and the wave celerity, $c$, is defined as $c = \frac{\lambda}{T}$, where the wave period, $T$, is the time between consecutive crests.

1.2.1. WAVE PARAMETERS

A wave train is a series of waves, like the one presented in Figure 1.4, travelling in the same direction, spaced at regular intervals. There are three physical dimensions that uniquely define a wave train: the water depth $d$, the wave height $H$, which is delimited crest-to-trough, and the wavelength $\lambda$. There are many wave theories where the wave period, $T$, the time between two consecutive crests, replaces the wave length as the third parameter. The wavelength and wave period are affiliated through the wave velocity relative to the current (Fenton, 1990). This relative wave velocity is often referred to as the wave celerity and can be defined as $c = \frac{\lambda}{T}$.

Another important parameter of a wave is its steepness, which can be described by the ratio of the wave amplitude, $A$, and the wave length, $\lambda$. The wave steepness is a measure for the nonlinearity of the wave. A regular wave also has a limiting theoretical wave height, $H_{\text{max}}$, which is dependent on the water depth and the wave length. According to the theoretical work of Williams (1981) and Fenton (1990), this limiting wave height for regular waves can be defined by

$$
\frac{H_{\text{max}}}{d} = \frac{0.141063 \frac{A}{d} + 0.0095721 (\frac{A}{d})^2 + 0.0077829 (\frac{A}{d})^3}{1 + 0.0788340 \frac{A}{d} + 0.0317567 (\frac{A}{d})^2 + 0.0093407 (\frac{A}{d})^3}.
$$

Here, $H_{\text{max}}$ can be used as an alternative method for describing the measure of wave nonlinearity. The ratio of the wave height, $H$, and the limiting wave height, $H_{\text{max}}$, gives a more consistent measure for the nonlinearity across various water depths than the ratio of the wave amplitude and the wave length.

1.2.2. SMALL LINEAR WAVES

Small non steep regular waves, travelling in intermediate to deep water, where the free surface elevation is described by a simple sine function, are in general well defined by linear wave theory. The validity of various wave theories has been thoroughly investigated throughout the last decades, the result of which is presented in Figure 1.5. Here, the validity of a wave theory is presented as a function of the wave height and the water depth. It may be noted that these analytical wave theories are only valid up to the theoretical breaking limit, $H_b$. It can be observed that the wave height for which linear theory gives a valid description of the regular waves declines with the water depth. This can be explained by the fact that the profile of a wave changes as the wave moves towards shallower water. During this transition the wave
1.2. FREE SURFACE WAVES

Figure 1.5: Validity of various wave theories presented as a function of the wave height and the water depth, both of which were normalized by the gravitational acceleration multiplied by the wave period squared. $H_b$ is the theoretical breaking limit (Det Norske Veritas, 2014).

crest tends to get shorter while the trough gets flatter, i.e. the measure of nonlinearity of the wave increases. These changes to the wave profile are an effect of the enlarged influence of the seabed. This phenomenon is considered to take effect from the deep water limit, $kh \approx 3$, where $kh$ is the non-dimensional water depth, which is the water depth multiplied by the wave number, $k = \frac{2\pi}{\lambda}$.

Small non steep waves may be significant to the Fatigue Limit State (FLS) of an offshore structure due to their abundance in real sea states. However, their immediate impact load, when compared to extreme waves during storm events, is relatively insignificant to the Ultimate Limit State (ULS) of an offshore structure and therefore considered to be outside the scope of the present research.

1.2.3. STEEP NONLINEAR WAVES

Waves closer to the theoretical breaking limit, $H_b$, especially in shallow and intermediate water depths, are considered to have significant steepness. This implies that they are characterised by shorter crests and flatter troughs and do not resemble a single sine function, see Figure 1.4 for a schematic representation of such a wave profile. Therefore, linear wave theory can no longer be used to give an accurate description of the wave profiles of such waves. Instead, higher order wave theories, e.g. higher order Stokes, Cnoidal or the nonlinear stream function (Fenton, 1990), can be used to describe regular steep waves. The validity of various orders of the Stokes and the stream function theories are presented in Figure 1.5. It may be noted that these theories are restricted to the description of regular waves, or wave trains, and a flat sea bed and so do not represent realistic irregular sea states.

A limitation to the use of both the Stokes and Cnoidal theory is that they are specific to certain types of waves and thus not generally applicable. Stokes theory is known to lack in accuracy when describing shallow water waves, while the accuracy of Cnoidal theory is limited when it comes to the description of deep water waves. Both theories face difficulties describing extremely high waves. Various efforts were undertaken to develop methods which did not face these problems. Most of these wave theories focussed on the numerical determination of the Fourier coefficients, instead of using perturbation expansions as was done for the Stokes theory. This finally led to the fully nonlinear stream function method by Rienecker and
Fenton (1981), that could be utilised for the accurate description of steep nonlinear waves in both deep water and water of finite depth. This theory was later presented in a computer program by Fenton (1988).

Wave breaking, a process which causes large amounts of wave energy to be transformed into turbulent kinetic energy, can occur in the proximity of the theoretical breaking limit. Sarpkaya and Isaacson (1981) indicated that there are four possible types of wave breaking: spilling, collapsing, plunging and slurring. Such overturning waves are considered to be strongly nonlinear, where, air and water layers are mixing and multiple free surfaces start forming. It should be noted that no analytical solutions exist for these kind of waves.

1.2.4. WAVE LOADS ON A SURFACE PIERCING CYLINDERS

The phenomenon of wave loads on vertical surface piercing circular cylinders is a much researched topic since they appear as a fundamental part of many offshore structures. The inline forces on vertical surface piercing cylinders, subjected to incoming regular waves, can be classified as: drag, inertia and diffraction. The governing type is determined by the ratio between the wave length and the cylinder diameter, $\lambda/D$, and the ratio between the wave height and the cylinder diameter, $H/D$. Figure 1.6 can be used as a guideline for establishing the dominant force type in a specific design case. If a cylinder is relatively wide compared to the wavelength, $\lambda/D < 5$, wave diffraction is considered to be the dominant load factor, while the inertia forces are dominant for $\lambda/D > 5$ and drag forces become dominant for long waves where the cylinder is relatively slender compared to the height of the wave, or $H/D > 10$.

In context of the present research, the size of the main tubular members of the OC5 FOWT, $D = 12m$ combined with the relatively high, $H \approx 20m$, and long waves, $\lambda \approx 200m$, relevant for the ULS design of a floating offshore structure in intermediate water depth, give rise to moderate $\lambda/D$ and a large $H/D$. This would indicate that inertia forces play a dominant role in the investigation of these extreme wave load events. It may be noted that, not the direct impact of a large wave, but rather a secondary phenomenon, such as spray or wave run-up against a more fragile sub-component can be the cause of significant structural damage. Highly detailed numerical or physical models are needed to evaluate the potential dangers of these nonlinear fluid-structure interactions.

Analytical solutions to wave forcing by regular waves on vertical cylinders, that include second, and third-harmonic load components, have been derived by Faltinsen et al. (1995) and Malenica and Molin (1995). However, these solutions are only valid for regular non-steep waves and to date no higher order analytical solutions for wave forcing exist for real irregu-
lar sea states. At present the Morison equation (Morison et al., 1950) is widely applied for the calculation of hydrodynamic loads on offshore structures. This semi-empirical equation for determining the inline force on a body subjected to a flow consists of two parts; an inertia force and a drag force. State-of-the-art offshore wind turbine design codes, like the Det Norske Veritas (DNV), prescribe the Morison equation to calculate the inline wave forces on slender structures, such as tubular members of jackets and monopiles.

When it comes to calculating the loads on FOWT support structures, the Morison equation has a couple of shortcomings. Three of the main disadvantages were pointed out by Matha et al. (2011). First, to simplify the diffraction problem, the Morison equation assumes the wave potential to be constant across the body. This is in accordance with the long-wavelength approximation, so for bodies with a small diameter relative to the wave length. However, for many FOWT designs the body is relatively large and the wave disturbance may therefore not be neglected. Diffraction effects must be included if one is to correctly determine the local pressure forces and global wave loads. Secondly, the Morison equation assumes that viscous drag dominates the drag loading, and that wave radiation damping can therefore be ignored. This assumption is valid for slender structures and small motions, however, FOWT are in general of considerable volume and experience significant movement so that radiation forces should be taken into account. Lastly, the Morison equation does not take into account the added mass-induced coupling between the hydrodynamic force and the support structures accelerations. Apart from the limitations pointed out by Matha et al. (2011), the Morison equation is not able to describe strongly nonlinear free-surface effects encountered at fluid structure interaction in extreme sea states, e.g. large wave run-up, spray and wave breaking. Furthermore, the Morison equation fails to give detailed description of high-frequency loading, which is presumed to be critical for describing nonlinear phenomena such as slamming, springing and so called ‘ringing’.

Within the context of this research slamming is considered to be a hydrodynamic impact caused by a steep or braking wave. This strongly non-linear fluid-structure interaction can cause very large impact forces, which are highly dynamic and have a short duration. When it comes to FOWT structures, slamming loads can cause large local damage, but also accelerate the fatigue of the global structure due to the resulting dynamic response. The relative impact angle and the relative fluid-structure velocity are considered to be the main parameters determining slamming loads.

Springing is one of the more commonly observed non-linear phenomenon. It is a steady state resonance phenomenon caused by second-order wave effects, see Figure 1.7 for a representation of a springing response signal. Springing is observed in the vertical and bending modes of tension leg platforms and gravity based structures where it affects the fatigue level of the structure. Springing is therefore considered to be more significant to the FLS design then to the ULS design and will therefore not be incorporated in the present research.

Ringing, a potentially more dangerous nonlinear phenomenon, was first observed in some tension leg platforms and gravity based structures in the North Sea during the 1980’s. Since the discovery of the phenomenon great efforts have been made to identify the wave mechanisms that induce it in offshore structures. Ringing is a strong high-frequency transient response believed to be caused by very steep waves in extreme sea states. The structural response builds up over approximately one wave period and decays to steady state at a rate that depends on amount of damping in the system, see Figure 1.7 for a representation of a ringing response signal. Ringing of vertical cylindrical members of offshore structures was
once sought in third order theory, because early observations indicated third order wave frequencies to be corresponding to the natural frequency of the structure. A substantial amount of studies regarding the ringing phenomenon has been carried out since and it is now definitely known to be a strongly nonlinear phenomenon Rainey (2007). Higher harmonic nonlinear forcing as well as a secondary loading were identified by Grue and Huseby (2002), to be causing these ringing vibrations at a resonance frequency.

1.2.5. WAVE INTERACTION WITH A FLOATING BODY

Linear potential theory for waves acknowledges three contributions to the free surface elevation in the surroundings of a floating structure. First, the undisturbed incoming waves, which are present in spite of the presence of the structure. Secondly, the diffracted waves, generated due to the reflection of wave energy against the structure. Lastly, the radiated waves, resulting from the motion of the structure. For slender structures, waves pass relatively undisturbed, this is why the Morison equation assumes the wave potential to be constant across the body. However, significant disturbances to the wave kinematics, i.e. wave diffraction, are observed when the structure is relatively large compared to the wave length.

The DNV design code (Det Norske Veritas, 2014), states that wave diffraction becomes important for cylindrical shaped structures with a non-dimensional cylinder diameter $kD \gtrapprox 1.3$. Here, the non-dimensional cylinder diameter is the product of the wave number, $k = \frac{2\pi}{\lambda}$, and the cylinder diameter, $D$. This would imply that, if one considers the 12 m diameter cylindrical members and a wave climate where the highest waves have wave lengths in the order of 200 m, wave diffraction is not an important analysis criteria for a floating structure like the OC5 FOWT, since the non-dimensional cylinder diameter would stay well below the defined limit. It may however be the case that, if one looks at the structure as a whole instead of the individual tubular members, the presence of the relatively large structure can result in significant disturbances to the incoming waves. This can only be investigated if wave-structure interaction is taken into account.

Many design tools that are utilised to calculate hydrodynamic loads on offshore structures, were developed for stationary bodies. These models are often based on potential-flow theories which are only valid when the translational motion of the structure is small relative to the wavelength and the rotational motion is less than the wave steepness. Many floating configurations, however, can experience large translational displacements as a result of the low resistance to surge and sway in their catenary moored systems. This was denoted in Matha et al. (2011), where it was also concluded that these models, based on potential-flow theories, are no longer valid when it comes to the calculation of hydrodynamic loads on floating offshore wind turbines. Furthermore, the DNV design codes (Det Norske Veritas, 2014), state that the large translations of a FOWT, due to incoming waves, can produce significant
1.3. **Numerical Simulation of Wave-Structure Interaction**

To get an idea about the potential of numerical models with respect to engineering problems concerning free surface flow and fluid-structure interaction, this section will address some of the most important numerical methods currently at the disposal of hydrodynamic engineers.

### 1.3.1. Flow Theories

The numerical simulation of free surface flow problems is a challenging topic, let alone with the presence of a moving structure, however, it has been of considerable importance to the development of offshore structures. Several types of models are commonly seen in numerical simulation of the non-linear waves and wave-structure interaction. These models can be categorised as, either, Boussinesq, potential flow, or Navier-Stokes-type models, referring to the governing equations driving the model.

Boussinesq-type models have been widely used to solve free surface flows since the computational costs are reasonable and the accuracy of the approximation of water waves is valid for linear and weakly nonlinear waves, given that the wavelength is large compared to the water depth. A review of the developments of these types of models, as well as the description of a state-of-the-art model, is given by Madsen and Fuhrman (2010).

Potential flow theory has been very successful in modelling the wave loading on offshore structures and has become a standard tool in ocean engineering. The potential solver WAMIT is based on linear and second-order potential theory and is considered one of the best of its class, it has the capability of representing the geometry of the structure by a higher-order method (Lee, 1995). This is just one of the industries many state-of-the-art numerical models which are based on the potential flow theory, two other good examples are AQUAPLUS (Delhommeau, 1993) and AEGIR (Kring et al., 1999). These potential flow models in general well equipped to calculate first and second order wave loads and fluid-structure interaction at acceptable computational costs. They are very useful for demonstrating the initial feasibility of a floating structure. For instance by identifying the natural frequencies of the system, which, in order to minimise the dynamic response, need to be placed away from the wave spectrum.

However, for the investigation of the strongly nonlinear hydrodynamics phenomena involved with wave-structure interaction during extreme wave events, the potential flow theory is inadequate and fully non-linear numerical methods are required. The advancements in computational power in the last years have paved the way for fully nonlinear CFD mod-
els. These solve the Navier-Stokes equations in combination with the continuity equations to compute strongly nonlinear wave loads, fluid-structure and in some models it is even possible to simulate surface effect such as wave breaking.

1.3.2. **Navier-Stokes-type models**

The Navier-Stokes-type models can either be meshless or grid-based, each of these types has its own advantages and disadvantages. A major distinction between the two is the need for an additional equation to describe the free surface.

Smoothed-particle hydrodynamics (SPH) is a renowned fluid particle based meshless Navier-Stokes-type method that utilizes a Lagrangian specification of the flow field, i.e. the coordinates move with the fluid. The advantages of meshless methods is that they are generally better at dealing with large local distortions than grid-based methods. A thorough description of meshless methods, and in particular SPH, can be found in Omidvar (2010). Here, a SPH method was successfully implemented for the investigation of wave loading on a wave energy converter.

In a grid-based model the Navier-Stokes equations by themselves are unable to fully describe the interface of a two-phase flow. Therefore, an additional equation needs to be introduced. Methods for describing the free surface in multiphase flows can be categorized as Lagrangian-type surface methods and Eulerian-type volume methods, also referred to as surface tracking and surface capturing methods. A comprehensive analysis of these methods is given by Gopala and van Wachem (2008).

The surface tracking methods are based on a representation of the free surface by special marker points, this allows for explicit tracking of the interface. Interpolation is used to approximate the points between marker points. The advantage of this approach is that the position of the interface is known and that the interface remains sharp as it is advected across the domain. However, these models experience problems when simulating coalescence and breakup of the interface, so, for instance when a droplet of water is dropped into a bucket filled with water, or more apt, when wave spray is caused by a large wave colliding with an offshore structure. The most widely used surface tracking method is the level-set method developed by Osher and Sethian (1988). Here, the interface is defined as a zero level of a signed distance function from the interface. To distinguish between the fluids on either side of the interface a negative sign is attached to the distance function for one of the fluids. The level-set methods are conceptually simple, relatively easy to implement and yielding accurate results. However, the method is known to experience loss of mass when the interface is significantly deformed or when there is considerable vorticity.

In contrast to surface tracking methods, the surface capturing methods support an implicit tracking of the free surface of a two-phase flow. These approaches mark the fluid on either side of the interface, either by massless particles or by an indicator function. One of the drawbacks of these techniques is that the exact position of the interface is not known explicitly and special techniques are needed to reconstruct the interface. The most widely used surface capturing method is the Volume of Fluid (VoF) method from Hirt and Nichols (1981). Here, a scalar indicator function between zero and one, known as volume fraction is used to distinguish between two different fluids. The free surface is treated as the transitional layer between the two fluids. Due to the diffusive nature of this method, special interface compression techniques are developed to maintain the sharpness of the interface. A major advantage of this method is that, in contrary to the level-set method, it is mass conserving and a far
more robust approach to when it comes to the simulation of coalescence and breakup of the interface.

1.3.3. Navier-Stokes/VOF solvers

Navier-Stokes-type models coupled with a VOF method for capturing the free surface are a commonly seen combination in the field of fluid mechanics. They form one of the more robust methods for simulating the highly nonlinear phenomena caused by fluid-structure interaction during extreme wave events. A couple of state-of-the-art commercial software CFD packages are based on the combination of these two models. Note that these software packages are generally not single purpose, but instead designed for a broader spectra of fluid mechanics problems.

There are many researches in which commercial software was utilised for the analysis of floating offshore structures. An excellent example is the analysis of a point absorber wave energy conversion system, performed by Yu and Li (2013), using STAR-CCM+, a commercial CFD package, developed by CD-Adapco. Here, a dynamic analysis was performed on the predominant heave motion of the wave energy converter. ComFLOW is another commercial state-of-the-art, Navier-Stokes/VOF solver. An extensive review of this model was given by (Wellens, 2012), where an analysis is performed on an offshore semi-submersible subjected to waves. The unique feature of ComFLOW is that this method works with a single phase flow.

In addition to commercial software packages, some open-source models are available which can be of great scientific value if they have a progressive community. Open Field Operation And Manipulation (OpenFOAM®), is an excellent example of such an open source CFD package. The standard OpenFOAM® software package is distributed with a large number of solvers and utilities to cover a wide range of problems. Due to the open-source nature of OpenFOAM®, it is possible for users to write their own codes and solvers for their specific problems or to modify the existing solvers. Recently the waves2Foam toolbox was developed by Jacobsen et al. (2012), here, the standard two-phase incompressible Navier-Stokes/VOF solver was extended with a method to generate and absorb waves. A relaxation zone technique known as active sponge layer has been applied to the library as well as a large range of different wave theories. The toolbox is readily available as waveFoam, an extension of the OpenFOAM® basic interFoam solver. The capability to generate, propagate and absorb waves has been shown in Jacobsen et al. (2012). This toolbox was also used in an extensive analysis of wave forcing from steep and fully nonlinear waves on a bottom-mounted offshore wind turbines by Paulsen et al. (2014a). It should be noted that, although the accuracy with which the model can propagate waves, determine wave loading and simulate highly nonlinear wave-structure interaction was actively researched over the last decade, the wave-structure interaction with respect to floating structures has not been thoroughly examined.

1.3.4. Efficient domain decomposition

Although these Navier-Stokes/VOF models are proven to be very accurate in simulating waves and wave-structure interaction, the engineering problems which can be investigated are limited in space and time because in terms of demand of computational power, CFD is an expensive engineering tool. A method with which the computational time can be reduced, without losing the accuracy in the wave-structure interaction, is to apply a domain decomposition strategy. Two types of domain decomposition strategies exist: two-way coupling, where information propagates both from the outer to the inner domain and vice versa, or, one-way
one-way coupling, where information only propagates from the outer to the inner domain. The advantage of one-way coupling is its simplicity and the efficiency with which the coupling between fully nonlinear models can be established. Recently a fully validated one-way domain decomposed model was presented by Paulsen et al. (2014b). Here, the fully nonlinear three-dimensional potential flow model, OceanWave3D, from Engsig-Karup et al. (2009), was coupled with the fully nonlinear Navier-Stokes/VOF solver through the relaxation zones provided by waves2Foam. The potential flow model can be used to efficiently simulate realistic sea states while the computationally more expensive Navier-Stokes solvers simulates the highly nonlinear flow effects involved in the wave-structure interaction.

1.4. THIS THESIS

As follows from the discussion in Section 1.1, offshore floating wind can be a promising solution to harvest the wind resources further out at sea. However, in order make it possible to develop full scale FOWT farms in the near future, reliable and purpose oriented design methods are required to already reduce uncertainties and risks in the design phase. There are still many uncertainties with respect to the behaviour of FOWT and their mooring systems in extreme wave events. These types of structures will be deployed in intermediate water depths, where the waves during extreme events will be steeper than at deep water locations. The conventional Boussinesq- and Morison-type engineering tools are insufficient to accurately model the interaction of these extreme waves with a moored floating structure in order to investigate strongly nonlinear phenomena such as: ULS design loads from (near) breaking waves, local impact loads on sensitive structural members, peak forces in mooring and anchor systems, damping characteristics and diffraction including nonlinear free surface effects. Highly expensive physical model test campaigns are at present the only valid method to investigate these effects.

In order to reduce design uncertainties and to speed-up design iterations, an accurate and well validated numerical model is essential. For such a model to be considered a valuable supplement or alternative to physical model testing it shall as a minimum be capable of describing: strongly nonlinear breaking waves, local impact pressures and wave run-up, rigid body motion of a moored floating body and loads in the mooring system during wave impacts. Furthermore, in order to ensure a sufficiently high impact of such a model on the future development of floating offshore structures, the model should be: geometrical flexible in order to increase pertinency, fully parallelized for multi-core systems in order to enhance computational effectiveness, open-source in order to increase accessibility and last but not least it should be easy to couple with other flow-, soil-, and structural solvers in order to stimulate an integrated design method. The main objective and aim of this thesis is:

"To establish a well validated fully nonlinear numerical wave tank for the simulation of complex fluid-structure interaction of moored floating offshore structures."

There are a limited number of frameworks available to establish a numerical wave tank that meets the established requirements. The most suitable models, as discussed in Section 1.3, are generally based on the Navier-Stokes equation coupled with a geometrical flexible scheme for describing the free surface. One model, which fits the proposed requirements and stands out for this research is OpenFOAM®. This widely used open-source CFD library is capable of solving free surface Newtonian flows by using the Navier-Stokes equations coupled with a
1.5. Structure of this thesis

The structure of this document is as follows. A review of the utilized numerical models will be provided in Chapter 2. This chapter includes the description of the general algorithms, governing flow equations, boundary conditions, wave generation and absorption zones, dynamic mesh capabilities and mooring line implementations. Four different two-dimensional validation cases will be elaborated on in Chapter 3, this is followed by a short discussion, in Chapter 4, on the effects of the local mesh refinement and automated mesh generation tools, available in the standard distribution of OpenFOAM®. In Chapter 5 the model is used to simulate three-dimensional free and moored decay simulations of a FOWT, which are compared to physical experiments performed in the MARIN basin. The capabilities of the numerical model will be put to the test in a final proof-of-concept simulation, in Chapter 6, where a moored FOWT is subjected to irregular. This thesis concludes with a discussion of the presented work in Chapter 7.
2

THE NUMERICAL MODELS

2.1. INTRODUCTION

This section gives a description of the numerical models that were utilised in this research. To begin with, the fully nonlinear Navier-Stokes/VOF solver used for the accurate simulation of the complex free surface flows and fluid-structure interactions is presented in Section 2.2. Furthermore, the combination of this Navier-Stokes/VOF solver with a dynamic motion solver is treated in Section 2.3. This section also discusses the implementation of restraints to the floating structure in the form of catenary mooring lines and linear springs. On top of that, Section 2.4 gives a brief introduction to the potential flow solver which can be used for fast and accurate reproduction the wave fields. The numerical models applied in this research are open-source and thus in principle freely available as part of the OpenFOAM® and waves2Foam framework. It should be noted that the implementation of the waves2Foam toolbox in the coupled Navier-Stokes/VOF and dynamic motion solver is not readily available, however, the implementation is relatively trivial and documentation on this process is available. Furthermore, the implementation of the catenary mooring lines, as presented in Equation 2.3.5, is developed by Niels Jacobsen (Researcher/advisor at Deltares) and presently not publicly available.

2.2. INTERFOAM/WAVEFOAM SOLVER

This section elaborates on the waveFoam solver used to accurately describe the fluid-structure interaction for situations where no movement of the structure is required. The waveFoam solver, is based on the interFoam solver provided by the open-source CFD-toolbox OpenFOAM®, version 2.3.1., which was extended with the implementation of the wave generation and absorption toolbox, waves2Foam, developed by Jacobsen et al. (2012). The waveFoam solver utilises the two-phase incompressible Navier-Stokes equations in combination with a VOF-surface capturing scheme (Hirt and Nichols, 1981) to compute fluid-structure interaction. The focus of this section will be on the governing equations, the boundary conditions and the implementation of wave generation and absorption zones.
2.2.1. **Governing equations Navier-Stokes/VOF**

The governing equations, used in the Navier-Stokes/VOF solver waveFoam, for conservation of mass and momentum of an incompressible flow of air and water are given by

\[ \nabla \cdot \mathbf{u} = 0, \]  
(2.1)

\[ \frac{\partial \rho \mathbf{u}}{\partial t} + \rho \nabla (\nabla \cdot \mathbf{u}) \mathbf{u} = -\nabla p^* + (\mathbf{g} \cdot \mathbf{x}) \nabla \rho + \rho \nabla \cdot (\mu \nabla \mathbf{u}), \]
(2.2)

where \( \nabla = (\partial_x, \partial_y, \partial_z) \) is the three-dimensional gradient operator, \( \mathbf{u} = (u, v, w) \) is the velocity field in Cartesian coordinates, \( \mathbf{g} \) is the gravitational acceleration and \( p^* \) is pressure in excess of the hydrostatic pressure, which relates to the total pressure, \( p \), by

\[ p^* = p - \rho (\mathbf{g} \cdot \mathbf{x}). \]  
(2.3)

Furthermore, the local density, \( \rho \), and viscosity, \( \mu \), are given by the water volume fraction, \( \alpha \), consistent with

\[ \rho = \alpha \rho_{\text{water}} + \rho_{\text{air}} (1 - \alpha), \]  
(2.4)

\[ \mu = \alpha \mu_{\text{water}} + \mu_{\text{air}} (1 - \alpha), \]  
(2.5)

where \( \alpha \) is zero for air, one for water and a mixture of the two for all intermediate values.

OpenFOAM® uses a VOF method for tracking the air-water interface. After obtaining the velocity field by solving equations (2.1) and (2.2) for the two-phase flow of air and water, the VOF method (Hirt and Nichols, 1981) can be used to advance the \( \alpha \) field in time with the following scalar advection equation

\[ \frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha = 0. \]  
(2.6)

Using a standard finite-volume approximation for solving the hyperbolic advection equation (2.6) would lead to significant smearing of the interface. To reduce this, OpenCFD® extended the VOF method with an interface compression term. This is thoroughly discussed in Berberović et al. (2009). The interface compression term, which is the last term on the left-hand side of the following equation

\[ \frac{\partial \alpha}{\partial t} + \nabla \cdot \mathbf{u} \alpha + \nabla \cdot \mathbf{u}, \alpha (1 - \alpha) = 0, \]  
(2.7)

effectively reduces smearing of the air-water interface. The term is only active in the vicinity of the interface, \( 0 < \alpha < 1 \), where its strength is governed by the relative velocity, \( u_r \). It is stressed that even though \( u_r \) has the dimension of \( \text{m/s} \), it lacks any physical meaning. To ensure stability a multi-dimensional flux limited scheme (MULES) is used for solving the scalar advection equation (2.7).

In spite of the addition of the interface compression term to the volume of fluid method, the air-water interface can be smeared over several computational cells. Consequently these cells have a water volume fraction of \( 0 < \alpha < 1 \), which means that there is no clear location of the interface. If a free surface location is required, this can be defined by

\[ L_{\text{air}}^{\text{water}}(\alpha) = \{ \mathbf{x} | \alpha(\mathbf{x}) = 0.5 \}. \]  
(2.8)
2.2. INTERFOAM/WAVEFOAM SOLVER

Figure 2.1: Schematic representation of the computational domain in the Navier-Stokes/VOF solver, where I and II are relaxation zones implemented with the waves2Foam toolbox and all boundary surfaces were denominated with names used in the present research.

which defines the free surface by the iso-contour where $\alpha = 0.5$. Alternatively, the location of the free surface can be determined by making use of the wave gauge functionality in waves2Foam. One or more wave gauges can be placed at certain positions in the numerical domain, these determine the location of the free surface, $\eta$, relative to the still water level, according to

$$\eta = \int_{x_0}^{x_1} \alpha \, dz - h. \quad (2.9)$$

Here, $x_0$ and $x_1$ are the user defined start and end points of a vertical line over which the $\alpha$ field is integrated and $h$ is the water depth at the still water level condition. This approach yields a consistent value of the location of the free surface and is used throughout the present work for determination of the interface at specific location. The former method, using the iso-contours where $\alpha = 0.5$, is used purely for visualisation purposes.

2.2.2. BOUNDARY CONDITIONS

In order to solve the governing equations of the Navier-Stokes/VOF solver, boundary conditions need to be specified for all surfaces in the numerical domain. Before going into the allocation of particular boundary conditions, the general denomination of the boundary surfaces is given in a schematic representation of the numerical domain, see Figure 2.1. It should be noted that a structure will be present inside the numerical domain for a greater part of the numerical computations, the boundary surface of the structure is generally denominated as the floating object.

At the inlet and outlet boundary surfaces the velocities, $\mathbf{u}$, and the $\alpha$ field are given by a closed form analytic wave theory or by the potential flow solver (see Section 2.4). Assuming the wave impacts are inertia dominated, a slip condition is applied to the surfaces of, fixed or moving, structures (e.g. horizontal cylinders in Chapter 3 and 4, or the floating offshore wind turbine structure in Chapter 5 and 6. The viscous boundary layer at the seabed is also neglected, since typically the propagation of waves with a CFD-solver is limited to only a couple of wavelengths and in such simulations the influence of the seabed on the wave forcing is presumed small. The viscous boundary layer of side-walls of the three-dimensional numerical wave-tank, as presented in Chapter 5 and 6, are also subjected to a slip condition, while for the two-dimensional computations, as presented in Chapter 3 and 4, an empty condition was applied to these walls. It may be noted that, by imposing a slip condition, the viscous effects are artificially suppressed and although OpenFOAM® supports turbulence modelling,
it is deliberately excluded in the present work. At the atmosphere boundary layer of the numerical domain, an atmospheric boundary condition is applied for the velocity, \(u\), and the \(\alpha\) field. This means that air and water are allowed to leave the numerical domain, while only air is allowed to flow back in. Furthermore, the total pressure at the atmosphere boundary is equal to zero.

### 2.2.3. RELAXATION ZONES

In the wave generation toolbox, waves2Foam, developed by Jacobsen et al. (2012), relaxation zones were developed based on an extension of the relaxation technique by Mayer et al. (1998). These relaxation zones were implemented to prevent wave reflection from the outlet boundary and also to prevent internally reflected waves, e.g. by the presence of a structure, to interfere with the wave generation at the inlet boundary. An arbitrary number of relaxation zones can be defined, which can be characterised as either an inlet or outlet. In the present work all implemented relaxation zones were rectangular shaped, nevertheless circular ring shaped relaxation zones can also be implemented and with some minor changes to the code of waves2Foam it would also be possible to define other geometrical shapes. In the relaxation zones the velocity and \(\alpha\) field are updated every time step according to

\[
\psi = \chi \psi_{\text{target}} + (1 - \chi) \psi_{\text{computed}}, \quad \psi \in \{u_H, w, \alpha\},
\]

where the target solution, \(\psi_{\text{target}}\), can be specified by one of the wave theories available in waves2Foam or by the potential flow solver, presented in Section 2.4. The computed solution, \(\psi_{\text{computed}}\), on the other hand are the numerically computed velocity and \(\alpha\) values, obtained from the Navier-Stokes/VOF. The weight of the each of the two solutions is determined by

\[
\chi(\xi) = 1 - \frac{\exp(\beta \xi) - 1}{\exp(1) - 1},
\]

where \(\chi\) is the local coordinate in the relaxation zone, which, as Figure 2.2 indicates, is one at the outer end and zero on the inner end of the relaxation zone. The shape factor, \(\beta\), can be chosen arbitrarily, however in the present work the default value, \(\beta = 3.5\), was used.

### 2.2.4. ALGORITHM WAVEFOAM

The following algorithm gives a schematic overview of the process sequence of the waveFoam solver. The actual code is presented in Appendix A. It may be noted that the only difference between the interFoam solver and the waveFoam solver is the highlighted relaxation of the alpha and velocity field.
2.3. \textbf{INTERDyMFoam/waveDyMFoam solver}

This section shortly discusses features of the waveDyMFoam solver, which is utilised in the present work to describe the fluid-structure interaction for cases where motion of the structure is required. The waveDyMFoam solver, is based on the interDyMFoam solver provided by the open-source CFD-toolbox OpenFOAM®, version 2.3.1. This solver couples the Navier-Stokes/VOF solver with a dynamic motion solver. For the present research, the OpenFOAM® standard interDyMFoam solver was extended with the wave generation and absorption tool-box, waves2Foam, developed by Jacobsen et al. (2012), and the restraints for floating structures developed by Niels Jacobsen (Researcher/advisor at Deltares). This section will focus on the dynamic motion solver, the dynamic mesh handling and the implementation of the motion restraints.

2.3.1. \textbf{Motions of a Body}

In order to simulate the flow-dependent motion response of a floating structure, for instance in the heave decay simulation of the two-dimensional horizontal cylinder discussed in Section 3.5, the Navier-Stokes/VOF solver has to be coupled with a six degrees of freedom (6-DOF) motion solver.

The term six degrees of freedom refers to the freedom of motion of a body. If a body has six degrees of freedom, it is free to translate along and rotate around the x, y and z axes, see Figure 2.3. It is also possible to restrict the number of degrees of freedom to, for instance, just translation along the vertical direction as is done to single out heave displacement in the two-dimensional heave decay simulation discussed in Section 3.5.

The present only work considers rigid body motion, of which the accelerations, velocities and displacements can be calculated from the forces and moments on the body. The waveDyMFoam solver extracts the total force on and moment around the center of gravity (COG) by integrating all the separate forces and moments over the body according to
Figure 2.3: Axis describing six degrees of freedom, three translations along and three rotations around the x, y and z axes.

\[ F = \int_B F_{\text{flow}} + F_{\text{ext}}, \]  
\[ M = \int_B M_{\text{flow}} + M_{\text{ext}}. \]

Here, \( F_{\text{flow}} \) and \( M_{\text{flow}} \) are determined from integrating the pressures obtained by solving the Navier-Stokes equation (2.2) and the continuity equation (2.1). The \( F_{\text{ext}} \) and \( M_{\text{ext}} \) are considered to be external forces and moments, an example of an external force would be the gravitational force, which acts on the COG of the body. Other external forces and moments can be restraining forces due to the anchoring of a floating structure with the use of mooring lines, a discussion on the implementation of these follows in Subsection 2.3.5.

2.3.2. UNDER-RELAXATION

As introduced above, the hydrodynamics and the motions of the structure can be decoupled in the numerical simulation of floating structures. The concept of negative added mass in weakly coupled numerical methods can give rise to instabilities, this phenomenon is thoroughly discussed in the work from Seng (2012) and Dunbar et al. (2015). Here, strong coupling between the Navier-Stokes/VOF solver and the 6-DOF solver was established, which successfully improved the stability of the numerical model. It should be noted that their findings were based on an older version of OpenFOAM®.

In the newer version of OpenFOAM® that is used in the present research an under-relaxation method is available to counteract the stability issues. Under-relaxation methods make use of an under-relaxation factor, a comprehensive evaluation of the influence of this under-relaxation factor is presented in Ferziger and Peric (2012). The under-relaxation method is embedded in the 6-DOF motion solver, here, the acceleration of the COG of the structure, which is calculated from the forces, is ‘relaxed’ by an ‘accelerationRelaxation’ factor, \( f \), according to

\[ a_2^* = f a_2 + (1 - f) a_1, \quad (2.14) \]
\[ a_3^* = f a_3 + (1 - f) a_2^*, \quad (2.15) \]
\[ a_3^* = f a_3 + (1 - f)(f a_2 + (1 - f)a_1), \quad (2.16) \]
\[ a_3^* = f a_3 + f a_2 - f^2 a_2 + (1 - f)^2 a_1, \quad (2.17) \]
where, \( a^*_n \) is the under-relaxed acceleration of the COG of the structure at the instantaneous time step. This is defined by the acceleration calculated from the forces and moments multiplied by the ‘accelerationRelaxation’ factor plus the under-relaxed acceleration of the previous time step, \( a^*_{n-1} \), multiplied with one minus the ‘accelerationRelaxation’ factor. Note that the relaxation of \( a \), only starts from the second time step, this is defined in the full code of the 6-DOF rigid body motion solver and the embedding of the under-relaxation method which can be found in Appendix C. This implies that if, \( f = 0.0 \), the under-relaxation prevents the structure from moving at all, since \( a^*_2 = f a_2 + (1-f) a_1 = 0 \) because \( a_1 = 0 \). While the implementation of the under-relaxation method turns the acceleration into a total variation diminishing (TVD) property, it should be noted that by doing so it also introduces a diffusive term to the numerical model which can have a negative effect on the convergence rate of the solution. Throughout the present research an ‘accelerationRelaxation’ factor of 0.5 is used, this value was adopted from default settings in the ‘floatingBlock’ tutorial from OpenFOAM®.

### 2.3.3. Dynamic mesh handling

After the displacement of the body is determined, the mesh has to be adapted in order to comply with this motion. Two fundamentally different methods can be distinguished for handling moving boundary surfaces. One features a dynamic mesh that allows for topological changes, which means that the number of points, faces, and cells in the mesh are allowed to change in order to concur with the motion of the body. The other method, is dynamic mesh deformation, which, as the name suggests, allows the mesh to deform while the number of points, faces and cells remains constant. The latter of the two is applied in the waveDyM-Foam solver since the former involves data mapping between time-steps which can be associated with numerical errors.

Different mesh deformation algorithms exist that can be used to compute the motion of the internal mesh points based on the movement of the boundary surface. The work of Bos et al. (2013) gives a thorough discussion on the implementation of a state-of-the-art deformation algorithm, the radial basis function interpolation, in OpenFOAM®. The mesh deformation algorithm in the version 2.3.1. of OpenFOAM® is based on the mathematically related spherical linear interpolation (Slerp) function, as presented in Subsection 2.3.4. This technique is thoroughly discussed in the work of Maruyama et al. (2012).

When a dynamic mesh problem is considered, the shape of the computational domain is varying in time, a schematic representation of this is presented in Figure 2.4. Here, the numerical domain \( \Omega \), alters its shape, during a time interval \( \Delta t \), to form a new configuration \( \Omega' \). In this transformation, a distinction can be made between the motion of the boundary points and the internal mesh points. The displacement of the boundary surface, \( B \), is determined by the forces and moments on that boundary, while the internal mesh points need to be moved as a reaction to the motion of the boundary points. Here, the mesh deformation algorithm has the responsibility to maintain the quality and validity of the mesh.

In order to enable the movement of the boundary surface, \( B \), and preserve the mesh quality, two different areas are defined around the moving boundary surface. The outer radius defines an area in which numerical cells are allowed to deform according to the described motion of the body. A smaller area is defined by the inner radius, in the area between the boundary surface and the inner radius, the computational cells translate and rotate according to the motion of the boundary surface without deforming, thus preserving the initial mesh orthog-
Figure 2.4: Mesh deformation in the computational domain, $\Omega$, during the motion of a body, $B$. Here, (a) is the initial configuration and (b) is the deformed configuration.

It may be noted that the internal point motion influences the solution only through discretisation errors (Ferziger and Peric, 2012).

### 2.3.4. Rotations and Translations

Older versions of OpenFOAM® used generic motion solvers based on solving an elliptic equation for the displacement. These motion solvers suffered from problems in terms of mesh quality, causing numerical instability. One of the most important entities for the mesh quality is the orthogonality of the grid lines, which is important for accurate flux calculations.

Version 2.3.1. of OpenFOAM® uses a specialised 6-DOF motion solver that handles the motion of the body and the mesh deformation by interpolating the translations and rotations using Slerp. These translations and rotations are expressed in quaternions en septernions, where a septerion consists of a vector and a quaternion. This subsection will give an insight in the use of Slerp and quaternions, more thorough studies on the subject can be found in the works by Shoemake (1985), Dam et al. (1998), Samareh (2002) and Maruyama et al. (2012).

#### Rotation using quaternions

Three different types of methods can be distinguished for the representations of rotation: Euler angels, rotation matrices and quaternions. The quaternion formulation is used in the waveDyMFoam solver, quaternions can be seen as a three-dimensional extension of complex numbers, also referred to as hyper complex numbers. They were first described by Hamilton (1844). A quaternion, $\mathbf{Q}$, can be represented by a vector in a four-dimensional hypersphere according to

$$\mathbf{Q}(w, v) = q_1 + i q_2 + j q_3 + k q_4, \quad w = q_1, \quad v = (q_2, q_3, q_4),$$

where $i, j$ and $k$ are imaginary units, $w$ is the real scalar and $v$ the imaginary vector, see Figure 2.5a for a schematic interpretation. The magnitude of the quaternion on the unit hypersphere is defined by

$$|\mathbf{Q}| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1,$$

and the multiplication of two quaternions is defined by

$$\mathbf{Q}_{12}(w_{12}, v_{12}) = [w_1 w_2 - v_1 \cdot v_2, w_1 v_2 + w_2 v_1 + v_1 \times v_2],$$

where it should be noted that the order of multiplication matters for the result, i.e.
A quaternion can also be represented as a rotation of a point by a certain angle, \( \theta \), about a unit vector, \( \hat{n} \), according to the following equation

\[
Q(w, v) = \left[ \cos \left( \frac{\theta}{2} \right), \hat{n} \sin \left( \frac{\theta}{2} \right) \right], \quad w = \cos \left( \frac{\theta}{2} \right), \quad v = \hat{n} \sin \left( \frac{\theta}{2} \right),
\]

where the inverse of the quaternion is defined by

\[
Q^{-1}(w, -v) = \left[ \cos \left( \frac{\theta}{2} \right), -\hat{n} \sin \left( \frac{\theta}{2} \right) \right].
\]

Using equations (2.21) and (2.22), a point \( p = (x, y, z) \), expressed as quaternion \( P(0, p) \) can be rotated over a quaternion \( Q \) according to

\[
P_{\text{rotated}} = QPQ^{-1} = P'Q^{-1},
\]

where it may be noted that the first operation in equation (2.23), \( QP = P' \), can be seen as the first half of the rotation, where one enters the fourth dimension, and the second operation, \( P'Q^{-1} \), the other half of the rotation, which cancels out the rotation in the fourth dimension, this is presented in a schematic manner in Figure 2.5(b).

**INTERPOLATION**

Slerp is a powerful interpolation method used for the interpolation of quaternions. They trace the minimum distance path between quaternions on the hypersphere. Here, a brief introduction to linear interpolation (Lerp) and spherical linear interpolation (Slerp) are given, where Lerp is used to interpolate the translation and Slerp to interpolate the rotation in body motion and mesh deformation.

**LINEAR INTERPOLATION (LERP)**

Lerp is the most widely applied technique for interpolation of values. The simplest example of Lerp, as presented in Figure 2.6, is interpolation using two weighted values, \( 1 - \alpha \) and \( \alpha \), according to the following formula

\[
Q_{12} = Q_1Q_2 \neq Q_2Q_1.
\]
2. The numerical models

Figure 2.6: Schematic interpretations of the different interpolation methods, where (a) is linear interpolation in a three-dimensional space and (b) is spherical linear interpolation in a four-dimensional space.

\[ t_{lerp}(t_1, t_2; \alpha) = (1 - \alpha)t_1 + \alpha t_2. \tag{2.24} \]

This type of interpolation is used for the interpolation of the translation vectors. Furthermore, the process of interpolation is commutative, which means it can be generalised for any number of interpolations according to

\[ t_{lerp}(t_1, t_2, \ldots, n; l_1, l_2, \ldots, l_n) = \frac{n}{\sum_{i=1}^{n} l_i} \sum_{i=1}^{n} \frac{l_i}{t_i}. \tag{2.25} \]

Spherical linear interpolation (Slerp)

Slerp is an interpolation method that divides the angle between two vectors by an arbitrary ratio. Slerp was introduced as a method for interpolation of quaternions by Shoemake (1985), it defines the minimum distance trace between two points on the hypersphere. Two quaternions can be interpolated with Slerp, see Figure 2.6, using two weighted values, \(1 - \alpha\) and \(\alpha\), according to the following formula

\[ Q_{slerp}(Q_1, Q_2; \alpha) = \frac{\sin[(1 - \alpha)\Theta]}{\sin \Theta} Q_1 + \frac{\sin[\alpha \Theta]}{\sin \Theta} Q_2, \tag{2.26} \]

where \(\Theta\) is defined as the angle between two quaternions, \(\theta = \cos(Q_1 \cdot Q_2)\). It may be noted that as \(\Theta \rightarrow 0\), this formula reduces to the formula for linear interpolation, equation (2.25).

2.3.5. Restraints for moving boundary surfaces

This subsection discusses the implementation of restraints for moving boundary surfaces provided by Niels Jacobsen (Researcher/advisor at Deltares). It should be noted that no official documentation of this implementation was available at present.

Two types of restraints have been implemented on version 2.3.1 of OpenFOAM®, a linear spring and a simple catenary type mooring line. It should be noted that this simple mooring line implementation does not take the interaction with the fluid into account, thus it is not influenced by waves or current. The restraints can be used to connect a floating body to a fixed anchor point, which, for the mooring line, is assumed to be located on a horizontal seabed. Another application, although not investigated in this paper, would be to interconnect two floating bodies, for instance to represent a towing line connection between a tug boat...
and a floating cargo. Each of these restraints gives rise to forces in the attachment points. This subsection discusses the derivation of these forces, as well as a brief description of the restraints.

**Linear spring**

The linear spring is implemented without any damping and the line is given a simple constant stiffness, $k$, and a rest length, $l_r$. The spring is defined such that the force in the spring is zero when the spring is at rest, i.e. has a length equal to $l_r$. The force exerted by the linear spring can be calculated according to

$$F_0 = -k \frac{r}{\|r\|_2} (\|r\|_2 - l_r) = -F_1,$$  \hspace{1cm} (2.27)

where $F_0$ and $F_1$ are the spring forces and $r = p_0 - p_1$ is the distance between the two connection points.

**Mooring line**

The mooring line implementation in the present work is used for anchoring a floating structure to the seabed, see Figure 2.7, where it is assumed that the seabed is horizontal. Three different mooring line states can be defined; simple state, resting state and hanging state, see Figure 2.8. These states and their corresponding forces in the attachment points of the mooring line are described in the following section.

**Simple state**

In the simple state, see Figure 2.8a, the mooring line is completely in suspension, i.e. the anchor point, $p_0$, is equal to the touchdown point, $p_t$. The implementation of the mooring line assumes that the line cannot break, while the distance between the connected objects can exceed the defined length of the line, $l_c$. This means that the state of the line can either be ordinary or length exceeded. The ordinary formulation follows the well known resistance force in a catenary line between two attachment points, where the distance between the attachment points is less than the length of the line, see e.g. Krenk (2001). The horizontal restraining force, $F_H$, is a function of the distance between the two attachment points following

$$F_H = \frac{s^2 - d^2}{2d} m_{sub} g,$$  \hspace{1cm} (2.28)

where $s$ is the instantaneous length of the mooring line, which in this case is equal to $l_c$, $d$ is the vertical distance between the attachment points and $m_{sub}$ is the submerged weight of the mooring line. The resultant force, $F_T$, can than be found according to
Figure 2.8: Schematic representation of the three different states of the catenary mooring line, where (a) is the simple state, (b) the resting state and (c) the hanging state. Furthermore, \( p_0 \) is the anchor point, \( p_1 \) is the attachment point to the floating object and \( p_t \) is the touchdown point of the line.

\[
F_T = F_H + m_{sub}g l 
\]  
(2.29)

where \( l \) is the vertical distance between the attachment point \( p_1 \) and the touchdown point, \( p_t \), which, in this case of the simple state, is equal to \( p_0 \).

When the distance between the two attachment points, \( ||r||_2 \), is larger than the 0.999\( l_c \), the line is treated as a linear spring. The stiffness of the line, \( k_c \), is based on the magnitude of the restraining force in the catenary line at the limit of its length and can be calculated according to

\[
k_c = \frac{F_{0.999} - F_{0.998}}{0.001l_c},
\]  
(2.30)

where \( F_{0.999} \) and \( F_{0.998} \) are the forces in the attachment point \( p_1 \), when the line has a length of respectively 0.999\( l_c \) and 0.998\( l_c \) respectively. The force exerted by the mooring line, when the length of 0.999\( l_c \) is exceeded, can be calculated by

\[
F_0 = -k\frac{r}{||r||_2}(||r||_2 - 0.999l_r) + F_{cr} = -F_1,
\]  
(2.31)

where \( F_{cr} \) is the tension in the mooring line at length 0.999\( l_c \). It may be noted that the mooring line is fully stretched and that the force \( F_{cr} \) is parallel with \( r \).

**RESTING STATE**

The resting state, see Figure 2.8b, is the mooring line state, where the floating structure is so close to the anchor point that a part of the catenary line will be resting on the sea bed. The remainder of the cable will behave as a catenary line. In order to evaluate the restoring force on the floating object in the attachment point, the touchdown point \( p_t \) is required, so that the instantaneous length, \( s \), of the suspended part of the mooring line can be determined. The touchdown point, \( p_t \), is evaluated by finding the resting length of the line that ensures that the slope of the line at the touchdown point is horizontal, i.e.

\[
\frac{\partial p_t}{\partial z} = 0.
\]  
(2.32)

Here, the touchdown point, \( p_t \), is defined as the resting length away from the anchor point. This requires iterations in the description of the mooring line. Once the resting length has been identified, the forces can be found in the attachment point following the catenary theory, see equations (2.28) and (2.29).
Hanging state

The hanging state is defined as the state where the floating object is so close to the anchor point that the piecewise linear configuration of the mooring line is shorter than the length of the mooring line, see Figure 2.7(c). It is defined that this state takes place when the angle, $\gamma$, between the line at the attachment point, $p_1$, and the horizontal exceeds $88^\circ$. Only a vertical force acts in the attachment point, $p_1$, in this state, which equals the weight of the suspended part of the line. It may be noted that both in the resting state and the hanging state, the force in the anchor point, $F_0$, vanishes.

2.3.6. Algorithm waveDyMFOAM

The following algorithm gives a schematic overview of the process sequence of the waveDyM-Foam solver. The actual code is presented in Appendix B. It may be noted that the waveDyM-Foam solver is basically the waveFoam solver with the addition of highlighted 6-DOF loop. The only difference with the interDyMFoam solver being the introduction of relaxation zones.

> Load case data
> Initialise waveDyMFoam solvers

while runTime
  > load run controls
  > set initial time step
  while PIMPLE loop
    while 6-DOF loop
      > extract forces and moments
      > move body / deform mesh
      > update flux field
    end 6-DOF loop
    if {first pimple loop} alpha
      > solve VOF equation
      > relaxation alpha & velocity field
      > update $\rho$ and $\mu$
      > update velocity field
    while pressure loop
      > update pressure field
    end pressure loop
  end PIMPLE loop
end runTime

2.4. Fully nonlinear potential flow solver

Recently the Navier-Stokes/VOF solver was coupled with the fully nonlinear potential flow solver (OceanWave3D), developed by Engsig-Karup et al. (2009), to form a one-way domain decomposed solver (Paulsen, 2013) (Paulsen et al., 2014b).

A schematic representation of the numerical domain of the domain decomposed solver is presented in Figure 2.9. In the outer domain, $\Omega$, the fully nonlinear potential flow problem is solved for the wave motion only. Additionally a smaller inner domain, $\Gamma$, is defined to
model wave-structure interaction and structure motions. Here, the Navier-Stokes/VOF solver is used to compute the velocity and \(\alpha\) field. The relaxation zones in the Navier-Stokes/VOF solver are used to extract information about the wave field from the outer domain, i.e. the velocity and surface elevation are interpolated from the outer domain, \(\Omega\), to the relaxation zones in the inner domain, \(\Gamma\). The Navier-Stokes/VOF solver is essentially using the fully nonlinear potential flow solution as target solution in the relaxation zones instead of one of the wave theories available in waves2Foam.

Since the potential flow solver is significantly faster and more accurate than the Navier-Stokes/VOF solver, due to stable higher order discretisation, it becomes possible to generate larger domains and longer time series. The coupling is set-up in a way that the potential flow solver can run a time series until a particular moment of interest, e.g. an extreme wave event, and only then start the computationally more expensive Navier-Stokes/VOF solver, after which there is runtime coupling between the two, see Subsection 2.4.2. This coupling supports parallel computations and utilises shared memory to improve performance.

2.4.1. Governing equations

Potential flow assumes the fluid to be inviscid and incompressible so that it can be described by the velocity potential, \(\phi\), and the position of the free surface. The physical fluid velocities can be expressed by \((u_H, v) = (\nabla_H \phi, \partial_z \phi)\), where \(u_H = (u, v)\) and \(\nabla_H = (\partial_x, \partial_y)\). The free surface, \(\eta\), relative to the still water level, can be evaluated according to the kinematic free surface condition

\[
\partial_t \eta = -\nabla_H \eta \cdot \nabla_H \phi + \tilde{w}(1 + \nabla_H \eta \cdot \nabla_H \eta),
\]

(2.33)

where \(\tilde{w} = \phi(x, \eta)\) and \(\tilde{\phi} = \phi(x, \eta)\) are the velocity and velocity potential at the free surface. By specifying the pressure at the free surface to be zero, the Bernoulli equation can be rewritten as the dynamic free surface condition,

\[
\partial_t \tilde{\phi} = -g \eta - \frac{1}{2} \left( \nabla_H \tilde{\phi} \cdot \nabla_H \tilde{\phi} - \tilde{w}^2(1 + \nabla_H \eta \cdot \nabla_H \eta) \right),
\]

(2.34)

The vertical velocity at the free surface, \(\tilde{w}\), is required to advance the free surface conditions in time. For this, the velocity potential in the fluid volume must be determined by solving the Laplace equation,

\[
\nabla_H^2 \phi + \partial_z \phi = 0, \quad -h \leq z < \eta,
\]

(2.35)

together with the kinematic sea bottom condition,

\[
\partial_z \phi + \nabla_H h \cdot \nabla_H \phi = 0, \quad z = -h,
\]

(2.36)

where \(h = h(x)\) is the water depth measured from the seabed to still water level. A detailed description of the fully nonlinear potential flow solver, one-way domain decomposition strategy and validation cases can be found in Paulsen (2013) and Paulsen et al. (2014b).
2.4. Fully nonlinear potential flow solver

2.4.2. Algorithm waveDyMFoam / potential flow

The following algorithm gives a schematic overview of the process sequence of the waveDyM-
Foam solver including a runtime coupling of the potential flow solver.

▷ Load case data
▷ Initialise waveDyMFoam solvers

while runtime
▷ load run controls
▷ set initial time step
▷ update wave field

while PIMPLE loop

  while 6-DOF loop
   ▷ extract forces and moments
   ▷ move body / deform mesh
   ▷ update flux field
  end 6-DOF loop

  if (first pimple loop) alpha
   ▷ solve VOF equation
   ▷ relaxation alpha & velocity field
   ▷ update \(\rho\) and \(\mu\)
   ▷ update velocity field
  end

  while pressure loop
   ▷ update pressure field
  end

end PIMPLE loop

end runtime
3.1. INTRODUCTION

This chapter explores the capabilities of the Navier-Stokes/VOF solver in a two-dimensional domain in terms of wave propagation with and without the presence of an object. For cases with a static mesh the waveFoam solver is used, whereas the waveDyMFoam solver is used for the cases where mesh motion is required. For this investigation, several cases of increasing complexity were modelled and analysed. A grid convergence study on the propagation of a fully nonlinear stream function wave (Fenton, 1988) is discussed in Section 3.2. Followed by a comparison between the experimental data from Dixon et al. (1979) and the numerical results for wave loading on a fixed, partially submerged horizontal cylinder in Section 3.3. In contrast to these first cases, the last cases investigate the capabilities of the Navier-Stokes/VOF solver when used in conjunction with a dynamic mesh. Section 3.4 reports on the generation of surface waves by the forced heave oscillating of a horizontal cylinder in a finite domain, where the result is compared with experimental data from Yu and Ursell (1961). The chapter concludes with the investigation of the heave decay of a free floating horizontal cylinder in Section 3.5. Here, the numerical result is compared to theoretical work by Maskell and Ursell (1970) and experimental work by Ito (1977).

3.2. CONVERGENCE STUDY ON THE PROPAGATION OF A FULLY NONLINEAR REGULAR WAVE

This section evaluates the ability of the Navier-Stokes/VOF solver to accurately describe the propagation of a fully nonlinear regular stream function wave in intermediate water depth. This is evaluated in a two-dimensional grid convergence study similar to that of earlier work presented by Paulsen (2013) and Paulsen et al. (2014a). The aim of such a grid convergence study is to evaluate whether the solution improves with the anticipated convergence rate when the spatial discretisation of the domain is changed. The predominant focus of this section is on the effect of the discretisation of the convective term in the momentum equation (2.2) by a first order Upwind scheme, also the investigation of a higher order scheme will be treated.
3. WAVE PROPAGATION AND FLUID-STRUCTURE INTERACTION IN 2D

3.2.1. NUMERICAL SET-UP

In the evaluation of the convergence rate of the Navier-Stokes/VOF solver, a continuously propagating fully nonlinear stream function wave is modelled in a two-dimensional domain. The properties of this nonlinear wave are specified in Table 3.1. The analytical wave profile was generated according to the stream function wave theory from Fenton (1988). It may be noted that this analytical solution only applies to a single phase flow of water, whereas the numerical solver gives the solution for a two-phase flow of water and air. The effect of the air phase is therefore part of the evaluation.

The convergence study was carried out in a rectangular domain where cyclic boundary conditions were applied at the inlet and outlet of the domain, as presented in Figure 3.1. By implementing these cyclic domain boundaries in the direction of propagation, a domain of only one wavelength could be used and the relaxations zones, used to generate and absorb waves, become unnecessary. This strategy, adopted from Paulsen et al. (2014a), was implemented to keep the computational time low and eliminate any influence the relaxation zones might have on the solution. The viscous boundary layer on the seabed was neglected and a slip condition was applied which is in accordance with the potential flow assumption of the stream function solution. An open inlet/outlet boundary condition is applied on the atmosphere boundary surface, satisfying that the total pressure is zero. The domain has a length, \( l \), which is equal to the wavelength, \( \lambda \). The height of the numerical domain was defined as the water depth, \( d \), at still water, plus a vertical dimension, \( a \), to accommodate for the free surface elevation and the air phase. Here, the length of \( a \) is equal to one and a half times the wave height, \( H \). The domain was discretised by \( n \times m \) cells with a unit aspect ratio, where the cell size is defined by the number of points per wavelength (p.p.w.) as,

![Figure 3.1: Schematic representation of the numerical domain used for the propagation of a fully non-linear regular stream function wave, where \( l = 200\text{m} \), \( d = 50\text{m} \) and \( a = 26\text{m} \).](image)

<table>
<thead>
<tr>
<th>Property</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water depth</td>
<td>( kh )</td>
<td>1.571</td>
</tr>
<tr>
<td>Wave height</td>
<td>( kH )</td>
<td>0.534</td>
</tr>
<tr>
<td>Wave period</td>
<td>( T / \sqrt{gk} )</td>
<td>20.37</td>
</tr>
<tr>
<td>Eulerian current velocity</td>
<td>( U_s / \sqrt{gk} )</td>
<td>0.00</td>
</tr>
<tr>
<td>Number of Fourier modes</td>
<td>( N )</td>
<td>32</td>
</tr>
<tr>
<td>Wave number</td>
<td>( k )</td>
<td>( 3.14 \cdot 10^{-2} \text{m}^{-1} )</td>
</tr>
</tbody>
</table>
3.2. **CONVERGENCE STUDY ON THE PROPAGATION OF A FULLY NONLINEAR REGULAR WAVE**

Figure 3.2: Visualisation of the numerical simulation of the propagation of a fully nonlinear stream function wave. Here, the iso-contour with $\alpha = 0.5$ is used to visualise the free surface at time interval (a) $t/T = 0.0$ and (b) $t/T = 15$. The spatial discretisation used for this simulation was 50 p.p.w.

$$\Delta = \frac{H}{p.p.w.} \quad (3.1)$$

The spatial discretisation of the domain is given in Table 3.2 by the number of p.p.w., the actual cell size in meters and the total number of computational cells in the domain.

The Courant-Friedrichs-Lewy (CFL) condition is utilised, for all the numerical computations performed throughout this research, to determine the time step. For a two-dimensional case, the CFL condition is described as

$$Co = \frac{|u_x|\Delta t}{\Delta x} + \frac{|u_y|\Delta t}{\Delta y} \leq Co_{max}, \quad (3.2)$$

where the dimensionless number, $Co$, is the Courant number, $u_{x,y}$ is the magnitude of the velocity, $\Delta t$ is the time step and $\Delta x$ and $\Delta y$ are the cell sizes in the x and y direction respectively. To ensure numerical stability the maximum Courant number for an explicit time solver should have a value between one and zero. This condition ensures that a particle cannot move further than the length of a computational cell. Unless otherwise specified a Courant number of $Co = 0.20$ is used throughout the two-dimensional simulations in Chapter 3 and 4.

A run-time analysis of the numerically computed free surface elevation was done with the use of 39 uniformly distributed wave gauges, i.e. a wave gauge was placed every 5 meters. Subsequently the phase error, $\delta \theta$, and the diffusive error, $\delta \eta$, are determined by comparing the numerically computed solution with the analytical stream function solution. The diffusive error, i.e. the change in wave height, was calculated using

$$\delta \eta = \sqrt{\langle (\eta'_x)^2 \rangle} \quad (3.3)$$

where the mean, $\langle \rangle$, of the difference between the numerically computed free surface elevation and the analytical stream function solution, $\eta'_x$, is calculated. Since the solution may also have a phase lag, the analytical stream function solution is shifted over a phase range, $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$, for each time step. The lowest diffusive error indicates the spatial position of the numerically computed wave and with it the phase error, $\delta \theta$, was established.

### 3.2.2. **ANALYSIS CONVERGENCE STUDY**

The numerical simulation of the propagation of the fully nonlinear stream function wave in a two-dimensional domain with cyclic boundaries is visualised in Figure 3.2. Here, the free surface is visualised using the iso-contour with $\alpha = 0.5$. It can be observed that the wave not
only becomes significantly smaller over time, expressed in this section as the diffusive error, but a phase shift is also introduced. The diffusive error is depicted as a function of time in Figure 3.3. Here, it can be observed that the initialisation of the wave field is the same for the simulations with a spatial resolution of 100, 150 and 200 p.p.w., while the diffusive error is slightly larger for the simulation where a spatial discretisation of 50 p.p.w. was applied. This may indicate that the 50 p.p.w. mesh is too coarse to fully grasp the detail of the initialised wave. In all four simulations it can be seen that the diffusive error significantly increases over the first half of the first wave period. When the $\alpha$ values are used to visualise the wave field it can be observed that the fluid interface, between the water and the air phase where $0 < \alpha < 1$, is sharply defined at time step zero. These visualisation also showed that, although the applied VOF method for tracking the free surface includes an interface compression term, equation (2.7), the interface is slightly smeared during the first half of the first wave period, as presented in Figure 3.4. This smearing effect is believed to be the main reason for the initial increase in the diffusive error during the first number of periods. After the initial increase of the diffusive error, over the wave first period, a more steady small increase is observed for the next fourteen wave periods. With increasing spatial resolution a reduction in the diffusive error can be seen and for the finest mesh resolution, with 200 p.p.w., the error remains in the order of $O(10^{-2})$ for the first 15 wave periods. It may be noted that the propagation of waves over 15 wavelengths with CFD software like OpenFOAM® is without a doubt beyond
The intended purpose of such software and that for most practical engineering purposes the propagation will be limited to a couple of wavelengths.

The logarithmic rates of convergence according to grid refinement are computed for each wave period and these are represented on the lower two axes of Figure 3.3. Here, the top axis depicts the convergence rate for the three coarsest meshes, 50, 100 and 150 p.p.w., whereas in the bottom axis the convergence rate for the three finest meshes, 100, 150 and 200 p.p.w. are shown. The initial logarithmic convergence rates are virtually zero because the wave field is initialised with almost the same precision in all four simulations. After the initialisation a first order convergence rate is expected, consistent with the first order upwind scheme used for the discretisation of the momentum equation. In both cases this is approximately true for the first couple of wave periods, while it is especially the rate of convergence for the three coarsest meshes, 50, 100 and 150 p.p.w., that starts to decline after the second wave period. This indicates that, over time, the increase of the diffusive error, for a simulation with a coarser resolution, becomes smaller relative to the increase of the diffusive error for a finer resolution. This could be explained by the fact that the numerical computation of a steeper higher wave is more diffusive than that of a flatter smaller wave, so, the more the wave is diffused, the less diffusive the numerical computation becomes.

The time step for each of the resolutions, presented in Figure 3.5, might give an extra indication why the rate of convergence for the coarser resolutions decline at a higher rate than that of the finer resolutions. The time step of the three finest resolutions, 100, 150, 200 p.p.w., become more or less steady after the initial estimated time step of $\delta t = 10^{-3}$ is corrected for. However, the time step of the coarsest resolution, 50 p.p.w., initially remains around the same value as the time step of the 100 p.p.w. simulation, after which it slowly starts to increase in an unsteady fashion. This is in contrast to what would be expected from the fixed Courant number of $Co = 0.2$ that governs the time step. This increased temporal resolution may have resulted in a lower level of diffusion in the numerical simulation using a resolution of 50 p.p.w., leading to the higher rate of decline in the rate of conversion observed in Figure 3.3.

Figure 3.6 presents the diffusive error and rates of convergence for wave propagation simulations performed with a second order Monotonic Upstream-Centered Scheme for Conservation Laws, otherwise referred to as a MUSCL scheme. Such higher order scheme are usually used to increase the performance of a numerical model, where the convergence rate of convergence is increased to reduce computational time. Ideally, discretisation of the convective term of the momentum equation with a second order MUSCL scheme should result in a second order convergence rate. However, the description of the free surface is still limited by the VOF scheme and the time integration is still carried out using a first order explicit Euler scheme, which will result in a less than second order convergence rate. From the results, presented in Figure 3.6, it can be observed that the numerical solution has improved in terms of the absolute diffusive error. However, the increase of the diffusive error over time is not
3. Wave propagation and fluid-structure interaction in 2D

Figure 3.5: Time step of the numerical computations, presented as a function, both of which have been normalized by the wave period.

Figure 3.6: Diffusive error, normalized by the wave height, as a function of time, normalized by the wave period. The convective term of the momentum equation is discretized with a second order MUSCL scheme. The lower two axes indicate the logarithmic rate of convergence, where (a) represents the convergence rate for 50, 100 and 150 p.p.w. and (b) represents the convergence rate for 100, 150 and 200 p.p.w.

steady, which results in an unsteady convergence rate. This is a problem as the quality of a numerical solution can not be checked by grid refinement studies. As a result the more stable first order Upwind scheme shall be used throughout this research.

The phase error is presented as a function of time in Figure 3.7 and Figure 3.8, where the computations were performed with Courant numbers of $Co = 0.2$ and $Co = 0.01$. From the results of the initial set-up, Figure 3.7, it can be seen that the phase error immediately starts to increase for the coarsest resolution of 50 p.p.w., while for the three finer resolutions it remains close to zero for the first three wave periods. While the phase error is positive in the case of the coarsest resolution, meaning that the wave propagates slower over time, the phase error becomes negative for the simulations with the finer resolutions, meaning that these waves propagate faster as time progresses. From the diffusive error it can be concluded that the amplitude of the waves declines over time in all four cases. If it is presumed that the wave
length remains constant, amplitude dispersion would dictate that the waves would propagate slower as the wave height decreases. As a reference a second set of computations were performed in which the Courant number was lowered to $Co = 0.01$, as presented in Figure 3.8. This change mainly affected the time step since all other variables used in the computation would remain approximately the same. The second set of computations, presented in Figure 3.8, shows a steady increase of the phase error for the coarsest resolution, while the phase error for the finer resolutions remains close to zero. This indicates that the time discretisation is governing the phase error.

Keeping in mind that the intended purpose of the Navier-Stokes/VOF solver is to propagate waves over no more than a couple of wavelengths, a successful validation against the fully nonlinear stream function solution has shown that the two-phase Navier-Stokes/VOF solver is able to provide a respectable reproduction of the single-phase reference solution. The use of a first order Upwind scheme resulted in a diffusive error in the order of $O(10^{-2})$ for the finer spatial discretizations of 150 and 200 p.p.w. However, these discretizations do not ensure that the solution is grid independent, so grid studies are recommended for all following cases. Note that no consistent convergence could be obtained with the use of a second order spatial scheme, therefore, the first order Upwind scheme shall be used throughout the present research.
3. Wave Loading on a Partially Submerged Fixed Horizontal Cylinder

This section elaborates on the ability of the Navier-Stokes/VOF solver to accurately compute wave forces on a partially submerged fixed horizontal cylinder. The validation cases in this section were set-up in order to reproduce results from experiments performed by Dixon et al. (1979).

3.3.1. Numerical Set-up

The work from Dixon et al. (1979) presents a series of experiments in which vertical and horizontal forces on a fixed horizontal cylinder, induced by regular waves of varying amplitude, were determined for various levels of cylinder submergence. In their work, they discovered that the varying buoyancy force of a partially submerged cylinder can play an equally large role as the inertial force in the resultant vertical force. In some of their cases these two forces lead to entirely negative vertical forces that acted at twice the wave frequency. In the Morison's equations, this would result in a negative inertial coefficient, \( C_M \), to cope with this they introduced a varying volume and buoyancy term to the Morison's equation. A total of ten cases from Dixon et al. (1979) were simulated using the waveFoam solver, five with a constant relative axis depth and a varying relative wave amplitude and five with a constant relative wave amplitude and a varying relative axis depth. Here, the relative amplitude and axis depth, are the ratio of the wave amplitude, \( A \), and the axis depth, \( d' \), to the cylinder diameter, \( D \). In order to compare the obtained forces, the dimensionless Root Mean Square (RMS) vertical force was determined according to

\[
F_{\text{rms}} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} \left( \frac{F}{\rho g (\pi D^2 l/4)} \right)^2}
\]  

(3.4)

where \( F_{\text{rms}} \) is the RMS force over one wave period, calculated from the relative vertical force, which is the total vertical force component divided by the weight of the water displaced by a totally submerged cylinder. Both theoretically and experimentally obtained RMS vertical forces from Dixon et al. (1979) are presented in Table 3.3. A constant relative wavelength of 15.62 was applied for all cases. The relative wave length is defined as the wavelength, \( \lambda \), divided by the cylinder diameter, \( D \).

A schematic representation of the numerical domain is presented in Figure 3.9. The di-
mensions, $l = 4\lambda$, $d = 0.5m$, $a = 0.25m$ and $D = 0.1m$, were applied to comply with the experimental set-up from Dixon et al. (1979). The mesh, including the cylinder wall, was created with hexahedra by defining individual vertices, blocks and arcs, this can be seen from Figure 3.10, where the numerical simulation is visualised. A refinement area was defined around the cylinder surface to maintain the unit ratio cell dimensions as much as possible. This meant that the cells at the cylinder boundary surface were twice as small as the reference spatial discretisation. It may be noted that the more than ninety percent of the cells in the computational domain had the reference cell dimensions, which were based on the p.p.w. This same method was also applied to generate the computational domains for the simulations in Section 3.4 and 3.5.

### 3.3.2. Analysis wave loading

The following subsection will start with the evaluation of two different relaxation zone set-ups. This is followed by a grid refinement study performed to evaluate the sensitivity of the numerical solution to changes in the spatial discretisation. Finally the results from the ten simulations, with various relative amplitudes and axis depths, proposed in Table 3.3, will be analysed. The waves2Foam toolbox provides two methods for conducting the wave loading experiment. The wave field can either be gradually developed over a certain period of time using a step function, or it can be initialised at the start of the simulation. Both methods involve implementation of an inlet and an outlet relaxation zone. The inlet relaxation zone

<table>
<thead>
<tr>
<th>Relative amplitude</th>
<th>Relative axis depth</th>
<th>Theoretical $F_{rms}$</th>
<th>Experimental $F_{rms}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0</td>
<td>0.063</td>
<td>0.062</td>
</tr>
<tr>
<td>0.20</td>
<td>0</td>
<td>0.126</td>
<td>0.121</td>
</tr>
<tr>
<td>0.30</td>
<td>0</td>
<td>0.185</td>
<td>0.172</td>
</tr>
<tr>
<td>0.40</td>
<td>0</td>
<td>0.238</td>
<td>0.221</td>
</tr>
<tr>
<td>0.50</td>
<td>0</td>
<td>0.284</td>
<td>0.264</td>
</tr>
<tr>
<td>0.20</td>
<td>0.10</td>
<td>0.128</td>
<td>0.121</td>
</tr>
<tr>
<td>0.20</td>
<td>0</td>
<td>0.126</td>
<td>0.121</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.10</td>
<td>0.115</td>
<td>0.112</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.20</td>
<td>0.099</td>
<td>0.101</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.30</td>
<td>0.074</td>
<td>0.087</td>
</tr>
</tbody>
</table>
can be set up to reproduce various types of waves. The waves produced in the experiment were said to be linear, however, in general the waves generated by a piston type wave maker are not perfectly linear, therefore the stream function wave was used to reproduce the wave profiles. It should be noted that this could result in small differences between the numerically computed wave profiles and the ones generated in the physical wave tank. Please note that the analysis of physical experiments is not trivial and there are more uncertainties in terms of the reproduction of these experiments.

There are two ways the outlet relaxation zone can be set up: either with a zero current condition, or with a wave theory condition equal to the one applied at the inlet relaxation zone. In the former, the target solution is a flat water surface condition, resulting in gradual absorption of the wave. In the latter, the target solution is the undisturbed incoming wave, here, the wave disturbed by the presence of the cylinder, is relaxed towards the undisturbed wave condition. If the disturbance of the wave profile is small, the relaxation zone has a relatively small task since the computed and the target solutions are close to one another. Figure 3.11 presents the results of a comparison of the between both types of outlet conditions.

As mentioned before, the wave field can be initialised in two ways; either by using a ramp function to increase the wave amplitude over time, or initialising the full wave field at the start of the simulation. The former, in which the simulation starts with a flat free surface, is
3.3. **WAVE LOADING ON A PARTIALLY SUBMERGED FIXED HORIZONTAL CYLINDER**

Figure 3.12: Vertical forcing, normalized by the total buoyancy force of the cylinder, as a function of time normalized by the wave period. Comparison between theoretical and experimental data from Dixon et al. (1979) and numerical simulation. Grid sensitivity analysis where a resolution of 50, 100, 150 and 200 p.p.w. are applied. The relative wave amplitude and axis depth are 0.5 and 0.0.

preferred when the zero current condition is applied to the outlet relaxation zone, while it is more desirable to use the latter when the wave condition is applied to the outlet relaxation zone. To evaluate the effect of the two different relaxation zone set-ups a reference solution was created. The simulation for the reference solution was performed in a domain of twenty-two wavelengths long, twenty of which were located behind the cylinder so that the forcing on the cylinder would not be influenced by reflections from the outlet boundary. The relative axis depth of the cylinder was zero and a wave, with a relative amplitude of 0.5, was ramped up over one wave period, $T = 0.9962\, s$. This meant that the total wave height was equal to the cylinder diameter and that the cylinder would be half submerged in still water. The spatial discretisation applied in these simulations was 150 p.p.w. Figure 3.11 depicts the vertical forcing on the cylinder for a timespan of 15 wave periods. It is seen that the vertical forcing of the reference solution is fully developed after approximately five wave periods. From Figure 3.11a, it can be observed that the vertical forcing with the zero current relaxation, starts out consistently with the reference solution during the ramp up phase and halfway the fifth wave period the first minor differences start to appear in the form of an overestimation of the first and second peak. In the following wave periods these peak forces decline towards a slight underestimation, after which the solution becomes steady around the ninth wave period. With exception of these minor differences in the peak forces the force profiles generally show good correspondence. Although the differences with the reference solution are relatively small, the results presented in Figure 3.11b, match even better with the reference solution. When the wave is generated from the start of the computation, the vertical forcing is not merely in better correspondence with the reference solution, also, this method brings with it the advantage that it only needs about one wave period to reach the steady state condition. This not only saves a lot of computational time, but also requires less data to be stored. As a result of this the wave condition will be applied throughout this research where suitable.

To verify the grid dependency of the solution, a grid refinement study was performed with the spatial discretisations presented in Table 3.4. The results of this grid refinement study are presented in Figure 3.12, where the vertical forcing on the cylinder was compared with both the theoretical and experimental data from Dixon et al. (1979). It can be observed that, as the size of the computational cells decrease, the correspondence between the numerical
solution and the experimental force profile increase. The coarsest resolutions, with a spatial
discretisation of 50 p.p.w., was considered to low to produce the desired level of accuracy
in the description of the forces acting on the cylinder as the the overall forcing was severely
underestimated. The two finest resolutions, 150 and 200 p.p.w., preformed at a much more
acceptable level. As the difference between these two is hardly noticeable, a resolution of 150
p.p.w. was adopted for the remainder of the simulations in this section.

It can be observed that the numerically computed force profile has better correspondence
with the experimental than with the theoretical solution. By performing a Fast Fourier Trans-
formation on the forcing signals, the results of which are presented in Figure 3.13, the good
correspondence between the numerical and the experimental forcing was confirmed. The
mean of the forcing signal, which is presented on the zero frequency, has a good fit for all
data. The amplitudes of the first and second harmonics of the experimental forcing signal
are slightly overestimated by the theoretical values, while it can be observed that the numer-
ical forcing signal gives a better match. Furthermore, while the the higher frequency loading
amplitudes of the experimental forcing signal were relatively small compared to the primary
frequency, it can be seen that these were reproduced by the numerically computed forcing.
Note that the theoretical forcing hardly captures any of the higher order responses correctly.
It may be noted that an even better fit was established for smaller amplitude waves, as can be
seen in Figure 3.14 and 3.15.

The ten simulations can be split into two categories: In the first five simulations the rela-
tive axis depth, \( d' \), is equal to 0.0, such that the cylinder is half submerged, while the relative
amplitude of the incident waves is varied from 0.1 to 0.5, the second set of simulations has
a constant relative amplitude, \( A' \), of 0.2, while the relative axis depth is varying from +0.1 to

<table>
<thead>
<tr>
<th>Points per wavelength</th>
<th>Cell size, ( \Delta ) [m]</th>
<th>Computational cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.03124</td>
<td>6224</td>
</tr>
<tr>
<td>100</td>
<td>0.01562</td>
<td>18144</td>
</tr>
<tr>
<td>150</td>
<td>0.01041</td>
<td>40456</td>
</tr>
<tr>
<td>200</td>
<td>0.00781</td>
<td>71684</td>
</tr>
</tbody>
</table>
3.3. Wave loading on a partially submerged fixed horizontal cylinder

Figure 3.14: Vertical forcing, normalized by the total buoyancy force of the cylinder, as a function of time normalized by the wave period. Comparison between theoretical and experimental data from Dixon et al. (1979) and numerical simulation. The relative axis depth, $d'$, is kept constant at 0.0, while the relative amplitude, $A'$, is varying from 0.1 to 0.5.

Note that two of these ten cases, with $d' = 0.0$ and $A' = 0.2$, are the same so only nine simulations had to be performed.

The RMS force, $F_{rms}$, was computed for all the numerical simulations, the result of which is presented in Table 3.5. Here, a comparison was made with the experimental data from Dixon et al. (1979) by computing the absolute error expressed and express this as a percentage. It can be observed that the numerical solution generally underestimates the experimental values, whereas the theoretical solution gives a slight overestimation. Although the correlation is in general quite accurate, around 15 percent, the underestimation gets worse for the simulations with higher waves.

The RMS forces from by Dixon et al. (1979), Equation (3.4), can be interpreted as the average force over a single wave period, perhaps more interesting are the actual force profiles and peak loadings presented in Figure 3.14 and 3.15. Here, the vertical force is given as a function of time and the numerical results are compared with both the analytical and experimental data. The cases with a constant relative depth are depicted in Figure 3.14, where it is observed that the analytical solution has a good correspondence with the experimental data.

Table 3.5: Comparison of numerically computed RMS forces and experimental data from Dixon et al. (1979), whereby the deviation between the two is expressed in percentage.

<table>
<thead>
<tr>
<th>Relative amplitude</th>
<th>Relative axis depth</th>
<th>Theoretical $F_{rms}$</th>
<th>Experimental $F_{rms}$</th>
<th>Numerical $F_{rms}$</th>
<th>Error [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0</td>
<td>0.063</td>
<td>0.062</td>
<td>0.054</td>
<td>13</td>
</tr>
<tr>
<td>0.20</td>
<td>0</td>
<td>0.126</td>
<td>0.121</td>
<td>0.104</td>
<td>14</td>
</tr>
<tr>
<td>0.30</td>
<td>0</td>
<td>0.185</td>
<td>0.172</td>
<td>0.147</td>
<td>15</td>
</tr>
<tr>
<td>0.40</td>
<td>0</td>
<td>0.238</td>
<td>0.221</td>
<td>0.180</td>
<td>18</td>
</tr>
<tr>
<td>0.50</td>
<td>0</td>
<td>0.284</td>
<td>0.264</td>
<td>0.210</td>
<td>21</td>
</tr>
<tr>
<td>0.20</td>
<td>0.10</td>
<td>0.128</td>
<td>0.121</td>
<td>0.105</td>
<td>14</td>
</tr>
<tr>
<td>0.20</td>
<td>0</td>
<td>0.126</td>
<td>0.121</td>
<td>0.104</td>
<td>14</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.10</td>
<td>0.115</td>
<td>0.112</td>
<td>0.097</td>
<td>14</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.20</td>
<td>0.099</td>
<td>0.101</td>
<td>0.094</td>
<td>7</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.30</td>
<td>0.074</td>
<td>0.087</td>
<td>0.074</td>
<td>15</td>
</tr>
</tbody>
</table>
especially for the smaller amplitude waves. For larger amplitude waves, as higher harmonics start playing a more significant role, see Figure 3.13, the theoretical formulated force signal provides an underestimation of the first force peak, around $t/T = 0.1$, and an overestimating of the second force peaks, around $t/T = 0.4$. Better consensus with the experimental data is achieved by the numerically computed forcing. The force profiles are in general better aligned and especially the peak forces from the more extreme cases show more resemblance. For the cases with a constant wave amplitude and varying submergence level of the cylinder, presented in Figure 3.15, a similar result is obtained. Good correspondence is observed for both analytical and numerical results where the incoming wave has a relatively small amplitude. However, if the cylinder is submerged deeper and the waves are nearly over-topping the cylinder, an extra peak force becomes tangible, which is described better by the numerical than by the analytical solution.

These findings were also substantiated by the correspondence of the peak forces presented in Table 3.6. Here, the minimum and maximum of both the experimental and numerical normalized forces and the corresponding absolute errors between the two are portrayed. With the exception of a couple of outliers, the errors were found to be in the order of around 5 percent. Please note that there is something peculiar about the experimental forcing signal of the two cases which had the worst fit with the numerical simulations. The experimental forcing, of the cases with a relative amplitude equal to 0.2 and submergence levels of -0.2 and -0.3, seem to have moved in time. Although it should be mentioned that the depicted force profiles in this section were obtained by extracting data points from the figures from Dixon et al. (1979), which may have resulted into very small discrepancies, the shift of these experimental forcing profiles is not a result of this. In conclusion of the capabilities of the Navier-Stokes/VOF solver with respect to the accurate computation of wave forces on a partly submerged cylinder it is considered that the result obtained with the waveFoam solver is of acceptable quality.
3.4. SURFACE WAVES GENERATED BY THE FORCED OSCILLATION OF A HORIZONTAL CYLINDER

This section explores the ability of the Navier-Stokes/VOF solver to accurately simulate fluid-structure interaction, to be more specific, the wave generation by the forced heave oscillation of a two-dimensional horizontal cylinder. For these simulations, with a moving body, the waveDyMFoam solver was used. The validation cases in this section were set up in accordance with experiments performed by Yu and Ursell (1961).

3.4.1. NUMERICAL SET-UP

In their work, Yu and Ursell (1961), present a theoretical formulation for surface wave generation by the vertical oscillation of a horizontal cylinder. In this theoretical formulation the cylinder is initially half submerged, after which the cylinder is forced to oscillate vertically in a channel of infinite length. To substantiate this theory they performed experiments. Here, a cylinder was forced to oscillate midway a channel with a finite length of 30m. A selection of the results from their experiments are presented in Table 3.7. It may be noted that, although wave absorbers were used at both ends of the channel, the reflections in the experiment were reported to be in the order of 10 to 20 percent. Numerous experiments were performed, all with different combinations of heave amplitudes and periods, the results of which were compared using a relative amplitude

\[ R_A = \frac{\text{Wave amplitude at infinity}}{\text{Amplitude of motion of cylinder}} \] (3.5)

where, the 'wave amplitude at infinity' was evaluated as an average of measurements taken over a distance of 2 to 6 meter from the cylinder.

Figure 3.16 shows a schematic representation of the numerical domain, which dimensions are defined in correspondence with the experimental set-up from Yu and Ursell (1961). Relaxation zones with a length of one wavelength were implemented at both ends of the two-dimensional domain, these were given a zero current condition to absorb the waves generated by the heaving cylinder. The spatial discretisation was once again based on the p.p.w., where the wavelength was calculated from the wave period according to the deep-water limit according to

<table>
<thead>
<tr>
<th>Relative amplitude</th>
<th>Relative axis depth</th>
<th>Experimental $F_{\text{min}}$</th>
<th>Experimental $F_{\text{max}}$</th>
<th>Numerical $F_{\text{min}}$</th>
<th>Numerical $F_{\text{max}}$</th>
<th>Error min [%]</th>
<th>Error max [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0</td>
<td>-0.100</td>
<td>0.083</td>
<td>-0.093</td>
<td>0.077</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>0.20</td>
<td>0</td>
<td>-0.199</td>
<td>0.147</td>
<td>-0.187</td>
<td>0.136</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>0.30</td>
<td>0</td>
<td>-0.278</td>
<td>0.188</td>
<td>-0.287</td>
<td>0.177</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>0.40</td>
<td>0</td>
<td>-0.387</td>
<td>0.207</td>
<td>-0.386</td>
<td>0.209</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.50</td>
<td>0</td>
<td>-0.441</td>
<td>0.250</td>
<td>-0.447</td>
<td>0.244</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.20</td>
<td>0.10</td>
<td>-0.197</td>
<td>0.161</td>
<td>-0.171</td>
<td>0.158</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>0.20</td>
<td>0</td>
<td>-0.199</td>
<td>0.147</td>
<td>-0.187</td>
<td>0.136</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.10</td>
<td>-0.186</td>
<td>0.118</td>
<td>-0.196</td>
<td>0.105</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.20</td>
<td>-0.178</td>
<td>0.085</td>
<td>-0.207</td>
<td>0.091</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.30</td>
<td>-0.169</td>
<td>0.041</td>
<td>-0.189</td>
<td>0.025</td>
<td>12</td>
<td>38</td>
</tr>
</tbody>
</table>

Table 3.6: Comparison of the peaks of the relative force profiles of numerical solution and experimental data from Dixon et al. (1979), where the deviation between the two is expressed as a percentage.
A ramp function, with the duration of approximately one period, was implemented in the cosine function describing the oscillating motion of the cylinder to stimulate a smooth initialisation of the simulation. The dynamic mesh motion solver, waveDyMFoam, was used to allow the cylinder to move. Here, an area around the cylinder was defined in which numerical cells were allowed to deform according to the described motion of the cylinder. The deformation area in the dynamic mesh solver is defined as a radius around the object. A second radius defines a small area around the object in which the computational cells are not allowed to deform. It may be noted that the deformation area must be larger than the displacement of the cylinder to prevent failure of the numerical model, as this inevitably happens when the cylinder moves outside of this area. Numerical failure was experienced to occurs at an earlier stage due to the high level of cell deformation caused by the compression of the computational cells between the edge of the body surface and the border of the deformation area.

### 3.4.2. Analysis wave generation

This subsection will first explore the implementation and implications of the relaxation zones, provided by the waves2Foam toolbox, in combination with a moving mesh. Secondly, a grid refinement study will be discussed in which the sensitivity of the solution to changes in the spatial discretisation will be analysed. Finally, the simulations, with various heave amplitudes and periods, proposed in Table 3.7 are evaluated.

To verify that the implementation of the relaxation zones is not significantly influencing

Table 3.7: Relative amplitude for various periods and amplitudes of the heave motion (Yu and Ursell, 1961)

<table>
<thead>
<tr>
<th>Heave amplitude [m]</th>
<th>Heave period [s]</th>
<th>Experimental $R_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00615</td>
<td>0.92</td>
<td>0.451</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.84</td>
<td>0.514</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.76</td>
<td>0.543</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.69</td>
<td>0.610</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.60</td>
<td>0.695</td>
</tr>
<tr>
<td>0.00455</td>
<td>0.51</td>
<td>0.810</td>
</tr>
<tr>
<td>0.00465</td>
<td>0.46</td>
<td>0.852</td>
</tr>
</tbody>
</table>
3.4. SURFACE WAVES GENERATED BY THE FORCED OSCILLATION OF A HORIZONTAL CYLINDER

Figure 3.17: Visualisation of the numerical simulation of the wave generation by the forced heave oscillation of horizontal cylinder. Here, the iso-contour with $\alpha = 0.5$ is used to visualise the free surface. The heave amplitude and period were 0.00615 m and 0.92 s and the applied spatial discretisation was 100 p.p.w.

the solution, a reflection analysis was performed. In this analysis the cylinder was given a heave amplitude and period of 0.00615 m and 0.92 s. A visualisation of the numerical simulation is depicted in Figure 3.17, since the motions of the cylinder were small and the generated waves even smaller one can hardly observe them. The reference solution was developed by computing the simulation in an extremely long domain with twenty wavelengths to either sides of the cylinder, in order to prevent possible reflections contaminating the free surface elevation near the cylinder.

In Figure 3.18, the reference solution, is compared against a simulation performed in a 'short' domain, with a total length of eight wavelengths and the cylinder placed in the middle. Figure 3.18a represents the free surface elevation measured with 'wave gauge 1' located one wavelength from the cylinder. It can be observed that the amplitude of the wave develops over the first eight wave periods, after which the signal from the reference solution becomes constant. The solution obtained from the 'short' domain shows approximately the same result for the first eight periods, after which discrepancies start to appear. It could be, that although a linear wave maker is used, some spurious free waves are generated, similar to what happens with piston type wave makers in physical wave tanks, a phenomenon of which a comprehensive review is given in the work of Schäffer and Steenberg (2003). This should however mean, that these spurious waves would also occur in the reference signal, which does not seem to be the case. This leads one to believe that the generated waves were not fully absorbed by the relaxation zones and that the disturbances are caused by wave energy that is reflected back into the numerical domain. These reflected waves, propagating towards the cylinder, could be interfering with the outwards propagating waves, causing the disturbances. Furthermore, a slight change in period of the wave signal can be observed. It may well be that these both these changes are caused by the reflection of waves from the relaxation zones, as both changes seem to take effect from the seventh period, which is approximately the time it takes for the first wave to travel to relaxation zone and back to the wave gauge.

The same result can be observed from Figure 3.18b, where the surface elevation from 'wave gauge 2', located two wave lengths from the cylinder, is presented. Although logically, the signal starts to develop one wave period later, it can be seen that the peak in the secondary wave pattern, observed during the fourteenth wave period from the signal of 'wave gauge 1', is witnessed during the thirteenth period from the signal of 'wave gauge 2'. This would make sense if the changes in the signal were the effect of reflecting waves as it takes the wave one period less to travel from the cylinder to the relaxation zone and back to 'wave gauge 2'.

A reflection analysis, available as a post-processing tool in waves2Foam, was performed
on the three different signals from Figure 3.18 to identify if waves were indeed reflected by the relaxation zones. The result of this analysis is depicted in Figure 3.19, where the absolute amplitudes from the various harmonics are presented as a function of the frequency, normalized by the wave period. It can be observed that the zero frequency, or mean amplitude, of both the ‘normal’ and the ‘reflected’ signals all had approximately the same non-zero value. Furthermore, it can be observed that in contrary to the reference solution, the ‘reflected’ signals from the other two simulations contain waves of a certain amplitude. The difference in amplitude, between ‘wave gauge 1’ and ‘wave gauge 2’, may be the effect of numerical diffusion. This effect, although smaller and obviously in opposite direction, can also be observed in the ‘normal’ signal. The reflection analysis also revealed a small amplitude for the frequency where $f \cdot T = 7$. This could be the indication of the existence of the secondary wave pattern observed in the surface elevation signal presented in Figure 3.18. The analysis indicated that the amplitude of this response was approximately equal for both the ‘normal’ and the ‘reflected’ signals, which may indicated that a higher order wave frequency was travelling in both directions. Since no reflected waves were observed for the reference simulation, it was presumed that this was an effect caused by the application of a moving mesh in combination with the presence of relaxation zones. It may be noted that excellent functionality of the relaxation zones for static mesh cases was shown in numerous researches, e.g. Jacobsen et al. (2012), Paulsen et al. (2014a) and Paulsen et al. (2014b). As the main purpose of this validation case was to verify the accuracy with which the fluid-structure interaction is mod-
3.4. **Surface waves generated by the forced oscillation of a horizontal cylinder**

![Figure 3.19](image)

Figure 3.19: The absolute amplitudes from the reflection analysis, performed on the normalized surface elevation signals from Figure 3.18, as a function of the frequency, normalized by the wave period. Comparison between the, ’normal’ harmonics, travelling away and the, ’reflected’ harmonics, travelling towards the cylinder.

elled, no further time was spend on finding the exact cause of the observed disturbances, but instead, a method for excluding the disturbance from the measurements was devised.

The most simple solution to exclude the disturbances would be to run all computations in the extremely long domain in which the reference solution was created, however, to save computational time and storage space, a small study was performed to find a trade-off between domain length and the amount of disturbance. In Figure 3.20, the result from Figure 3.18a, is compared to a computation performed in a 'long' domain, which was twice as long as the 'short' domain. It can be observed from Figure 3.20b, that the solution was able to fully develop without the interference of reflected waves. Therefore, the 'long' domain set-up was applied for all following computations in this section.

The next step would be to evaluate the dependency of the solution to the spatial discretisation of the numerical domain. This was evaluated in a grid refinement study, where grid resolution of 50, 100, 150 and 200 p.p.w. were compared. The grid resolutions and the total number of computational cells in the numerical domain can be found in Table 3.8. Just as in the evaluation of the relaxation zones and domain length, a heave amplitude and period of 0.00615 m and 0.92 s were used. In Figure 3.21, the normalized free surface elevations were presented for duration of ten periods. Taking into account the ramp time of one period and the period which is necessary for the wave to reach the wave gauge, it takes the wave approximately eight periods to fully develop, as was also observed during the reflection analysis. Once more it can be observed that a resolution of 50 p.p.w. gives a poor result, not only a significant difference in the height of the progressing wave, but a small phase difference can be noticed comparing it to the solutions of the finer resolutions. The solution is seen to improve as the spatial resolution increases and the difference between the two finest resolutions, of 150 and 200 p.p.w., is so small that it cannot be observed from Figure 3.21. The error of the

<table>
<thead>
<tr>
<th>Points per wavelength</th>
<th>Cell size, $\Delta$ [m]</th>
<th>Computational cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.02</td>
<td>16600</td>
</tr>
<tr>
<td>100</td>
<td>0.01</td>
<td>65400</td>
</tr>
<tr>
<td>150</td>
<td>0.0066</td>
<td>149400</td>
</tr>
<tr>
<td>200</td>
<td>0.005</td>
<td>263600</td>
</tr>
</tbody>
</table>
three finer meshes were all in the order of $O = 10^1$, so for the sake of the computational time the 100 p.p.w. was considered adequate for the final computations.

A total of seven numerical simulations were performed where the cylinder was forced to oscillate with heave amplitudes and periods matching the experimental work by Yu and Ursell (1961). The results from these computations were presented in Table 3.9, in the form of relative amplitudes and a deviation error expressed as a percentage. The relative amplitudes were calculated from the surface elevation signal of a wave gauge located one wavelength from the cylinder. It may be noted that both the location of the wave gauge, the size of the domain and the size of the computational cells was different for each case as the wavelength for the individual cases varied. Furthermore, the time it took to fully develop the wave profile varied from case to case. This meant that, for each individual case, a period was selected from which the wave amplitude was extracted. In all cases a the numerical relative amplitude was a slight underestimation of the experimentally obtained value. The error between the two was in the order of ten percent. Considering a spatial discretisation of 100 p.p.w., this meant that, for the case with a heave amplitude and period of 0.00615m and 0.92, the error between the two numerically computed and experimentally obtained amplitudes was in the order of $O = 10^{-4}$, while the spatial discretisation was in the order of $O = 10^{-2}$. It was therefore concluded that the fluid-structure interaction concerning a dynamic mesh problem was successfully simulated by the collaboration of the waveDyMFoam solver.
3.5. FREE HEAVE DECAY OF A HORIZONTAL CYLINDER

This section elaborates on the ability of the Navier-Stokes/VOF solver to accurately simulate the decay of the heave motion of a free floating two-dimensional horizontal cylinder. Here, the flow-induced motions of the structure were modelled by a 6-DOF solver, while a dynamic mesh solver was used to deform the mesh to enable the motion of the boundary surface. The validation cases in this section are set up in accordance with theoretical work by Maskell and Ursell (1970) and experimental work by Ito (1977).

3.5.1. NUMERICAL SET-UP

In their work, Maskell and Ursell (1970), present a theoretical formulation for the decay of heave motion of a free heaving horizontal cylinder. This theory was later validated by experimental work from Ito (1977), where a horizontal cylinder was partially lifted out of the water surface and then dropped into still water to observe the transient motion of the cylinder. The experiments performed by Ito (1977) showed excellent correspondence with the theory from Maskell and Ursell (1970), indicating that this theory was an excellent estimation of the physics.

Figure 3.22 shows a schematic representation of the numerical domain, with the following dimensions; \( l = 6m \), \( d = 1.24m \), \( a = 0.2m \) and \( D = 0.1524m \), which were applied to closely

<table>
<thead>
<tr>
<th>Heave amplitude [m]</th>
<th>Heave period [s]</th>
<th>Experimental ( R_A )</th>
<th>Numerical ( R_A )</th>
<th>Error [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00615</td>
<td>0.92</td>
<td>0.451</td>
<td>0.406</td>
<td>10</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.84</td>
<td>0.514</td>
<td>0.464</td>
<td>10</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.76</td>
<td>0.543</td>
<td>0.535</td>
<td>1</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.69</td>
<td>0.610</td>
<td>0.598</td>
<td>2</td>
</tr>
<tr>
<td>0.00615</td>
<td>0.60</td>
<td>0.695</td>
<td>0.664</td>
<td>4</td>
</tr>
<tr>
<td>0.00455</td>
<td>0.51</td>
<td>0.810</td>
<td>0.757</td>
<td>7</td>
</tr>
<tr>
<td>0.00465</td>
<td>0.46</td>
<td>0.852</td>
<td>0.763</td>
<td>10</td>
</tr>
</tbody>
</table>
mimic with the experimental set-up from Ito (1977). The main difference being that, instead of physical wave absorbers, a relaxation zone with a length of one meter was implemented at both ends of the domain. The governing length scale for this experiment is the diameter of the cylinder, $D$, rather than the wavelength, $\lambda$. Therefore, the spatial discretisation was defined according to the number of points per cylinder diameter (p.p.c.d.) was adopted, see Table 3.10. In correspondence with the experiment, the initial displacement of the cylinder’s COG, measured from the still water level, was one sixth of the cylinder diameter, or $d' = 0.0254m$.

The waveDyM Foam solver used to describe the motions of the cylinder, based on the pressures on the boundary surface, is a combination of the Navier-Stokes/VOF and the 6-DOF motion solver, this process is described in Subsection 2.3.3. Furthermore, in order to allow displacement of the boundary surface of the cylinder, a dynamic mesh solver was used. Here, a fixed area around the structure was defined, in which the numerical cells are allowed to deform according to the motion of the cylinder. An inner area is also defined, where deformations of the mesh were not allowed, in order to ensure mesh orthogonality and numerical stability. An outer radius of $R_{outer} = d' \cdot 10 = 0.25m$ was defined in order to ensure that the cylinder could not move outside this boundary which would inevitably result in the termination of the numerical simulation.

The simulation of a floating structure, where the hydrodynamics and motions of the structure are decoupled, can introduce negative added mass which in turn may cause numerical instabilities. Dunbar et al. (2015) discussed this phenomenon on the basis of the two-dimensional heave decay simulation that is also being explored here. The results they obtain, presented in Figure 3.23, showed clear signs of instabilities in the version of OpenFOAM® they utilised for their research. They successfully improved the stability of their solver by the introduction of a strong coupling between the Navier-Stokes/VOF solver and the 6-DOF solver, a technique also discussed in the work of Seng (2012). The main difference,

<table>
<thead>
<tr>
<th>Points per cylinder diameter</th>
<th>Cell size, $\Delta$ [m]</th>
<th>Computational cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.015</td>
<td>55120</td>
</tr>
<tr>
<td>20</td>
<td>0.0072</td>
<td>215904</td>
</tr>
<tr>
<td>30</td>
<td>0.005</td>
<td>497316</td>
</tr>
<tr>
<td>40</td>
<td>0.0038</td>
<td>857020</td>
</tr>
</tbody>
</table>
3.5. Free heave decay of a horizontal cylinder

Figure 3.23: Solutions for the body displacement, velocity and force from the simulation of free heave decay of a two-dimensional circular cylinder with a time step size of \( \Delta t = 0.001 \text{s} \), using the standard OpenFOAM® interDyMFoam solver (Dunbar et al., 2015).

Figure 3.24: Visualisation of the numerical simulation of the free heave decay of a horizontal cylinder. Here, the iso-contour with \( \alpha = 0.5 \) is used to visualise the free surface. The applied spatial discretisation was 5 p.p.c.d.

besides the lack of relaxation zones, between the solver utilised in the work of Dunbar et al. (2015), and the waveDyMFoam solver used to simulate the motions of a floating structure in the present research, is that the waveDyMFoam solver is equipped with an under-relaxation method. This is a dexterity that is utilised to conserve numerical stability, the implementation is described briefly in Subsection 2.3.2. This section explores whether the under-relaxation method can provide numerical stability and decent results in terms of the accurate computations of the heave decay of a two-dimensional cylinder.

3.5.2. Analysis motion decay

This subsection will focus on the decay of heave motion simulated with the waveDyMFoam solver. First the displacement of the cylinder was evaluated and compared against experimental and theoretical data, after which the velocity and force profiles were evaluated in order to examine the stability of the numerical solution.

A visualisation of the numerical simulation is depicted in Figure 3.24. The vertical displacement of a the COG of cylinder, relative to the still water level, during the decay tests is presented in Figure 3.25. Here, the vertical displacement is normalized by the initial displacement, \( d' \), and presented as a function of time normalized by \( \sqrt{g/\alpha} \). The cylinder was dropped from an initial displacement of 0.0254 m, with zero velocity. The decay test was performed using four different grid sizes ranging from 10 to 40 p.p.c.d., see Table 3.10 for the applied discretisations an the total number of computational cells.

It was found that the numerical domain, with a length of 6m, corresponding to the experimental set-up described in Ito (1977), was sufficiently long enough to exclude reflections from disturbing the solution of the numerical computation. From Figure 3.25, it can be ob-
Figure 3.25: Vertical displacement of the center of mass, normalized by the initial displacement $d'$, as a function of time normalized by $\sqrt{g/r}$. Comparison of different grid resolutions to theoretical and experimental results. The spatial resolutions 10, 20, 30 and 40 p.p.c.d. were applied.

Served that, as the cylinder gains velocity on its first descend, there was hardly any difference between the solutions of different spatial discretizations, and that the correspondence with the theoretical solution was excellent. The minor differences between the numerically computed displacement and the theoretical and experimental solutions indicated that, in the numerical domain, the cylinder gained more velocity during its descent. This indicated that slightly less damping was present in the numerical domain.

The four numerically computed solutions, of different spatial resolution, were seen remain in good correspondence with the theoretical solution as the cylinder moves in the positive, upward, direction. Small discrepancies between numerical and theoretical results only started to appear around $t/\sqrt{g/r} = 7$, as the cylinder reached the maximum upward displacement, where a minor overestimation of the peak response can be observed. This could again be the effect of an underestimation of the damping in the numerical simulation.

Please note that a comparison is being made between a two-phase numerical and experimental solution and a single phase theoretical solution. For a larger part of the first heave period, no significant difference was observed between the different spatial discretisations, small discrepancies can however be observed around $t/\sqrt{g/r} = 7$. It may be seen as a token of grid convergence that, the higher the spatial resolution the better the correspondence with both the theoretical and the experimental value.

The first response peak, around $t/\sqrt{g/r} = 7$, was analysed in slightly more detail. Similar to Dunbar et al. (2015), the numerical and experimental signals were compared in terms of amplitude and phase error. The results are presented in Table 3.11, from which can be observed that the period of the numerically computed transient motion was in close proximity of the experimental value. The error in the amplitude however, was in the order 10 to 20 percent. This meant that, depending on the specific case, the amplitude error was in the

<table>
<thead>
<tr>
<th>Points per cylinder diameter</th>
<th>Error period [%]</th>
<th>Error amplitude [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.21</td>
<td>19.5</td>
</tr>
<tr>
<td>20</td>
<td>1.59</td>
<td>14.4</td>
</tr>
<tr>
<td>30</td>
<td>1.27</td>
<td>12.6</td>
</tr>
<tr>
<td>40</td>
<td>1.29</td>
<td>12.1</td>
</tr>
</tbody>
</table>
3.5. **FREE HEAVE DECAY OF A HORIZONTAL CYLINDER**

Figure 3.26: Vertical velocity (a), normalized by the maximum velocity $v_{\text{max}}$, vertical acceleration (b), normalized by the maximum acceleration, $a_{\text{max}}$, and vertical force (c), normalized by the buoyancy force of the completely submerged cylinder. All presented as a function of time normalized by $\sqrt{g/\alpha}$. The spatial resolutions 10, 20, 30 and 40 p.p.c.d. were applied.

order of one tenth to a half of the computational cell size. It shall be noted that the theoretical and experimental transient motions depicted in Figure 3.25, were obtained by extracting data points from the figures in Ito (1977). This may have resulted into small discrepancies in the displacement profiles and consequently in the calculated period and amplitude errors presented in Table 3.11.

From vertical displacements, presented in Figure 3.25, it seems that the numerical solver provides an accurate solution to the problem. To examine the reliability of the waveDyM-Foam in terms of stability, the velocity, acceleration and force profiles of the cylinder were evaluated, these are presented in Figure 3.26. The velocity of the cylinders COG, normalized by the maximum velocity is presented, in Figure 3.26a, as a function of time, normalized by the square root of the gravitational acceleration divided by the radians of the cylinder. No significant disturbances could be observed in the velocity profiles. Figure 3.26b, depicts the vertical acceleration of the cylinders COG and the vertical force on the cylinder, normalized by the buoyancy force of the totally submerged cylinder is presented Figure 3.26c, both of
these are also presented as a function of the normalized time. The vertical acceleration, velocity and displacement are all governed by the vertical forces on the cylinder, as is described in Section 2.3. This becomes especially clear when the vertical force in is compared to the vertical acceleration. The force spikes which can be observed in the solutions from all four numerical simulations are also observed in the vertical acceleration of the cylinders COG. The observed force spikes in the velocity profile are relatively small, and smaller even than the ones observed in the force profile. This may be an effect of the implemented under-relaxation method, which uses an ’accelerationRelaxation’ factor to relaxes the acceleration calculated from the forces on the cylinder, see Subsection 2.3.2 for a more elaborated description of this technique. It should be noted that the disturbances that can be observed from Figure 3.26, are by no means as exceedingly large as the ones presented by Dunbar et al. (2015), as presented in Figure 3.23. This could mean that the stability of OpenFOAM® was improved by the implementation of the under-relaxation method.

A small investigation was executed to evaluate the influence of the ’accelerationRelaxation’ factor on the solution of the flow-induced motions of a floating structure. For this study five heave decay simulations of the two-dimensional horizontal cylinder were performed, where the only variation between the cases was the ’accelerationRelaxation’ factor. A spatial discretisation of 20 p.p.c.d. was used since this was seen to produce descent results in terms of stability, as seen from the velocity, acceleration and force profiles in Figure 3.26. The simulations were performed with a ’accelerationRelaxation’ factor of 1.0, 0.7, 0.5, 0.3 and 0.0, the results of which are presented in Figure 3.27.

From Figure 3.27a, where the vertical displacement of the cylinders COG is presented, it can be seen that the fully relaxed simulations, i.e. the simulation for which the ’accelerationRelaxation’ factor equals 0.0, the cylinder does not move. The cause for this can be found in the definition of the under-relaxation method. The acceleration at the second time step, \(a_2\), is equal to \(f a_2 + (1 - f) a_1\), which is zero, because the initial acceleration, \(a_1\), is defined as zero. Furthermore, can be observed from the figure that, the higher the ’accelerationRelaxation’ factor, the better the correspondence between the numerical and experimental result. From the accelerations and force profiles, presented in Figure 3.27b and 3.27c, it may be seen that, the higher the ’accelerationRelaxation’ factor, the more severe the disturbances. The simulation performed without under-relaxation of the acceleration, so where the ’accelerationRelaxation’ factor equals 1.0, the acceleration and force profiles displayed heavy oscillations similar to those presented in the work of Dunbar et al. (2015). This shows that the inter-DyMFoam solver in its self shows signs of instability and that this stability may be improved by using an under-relaxation method. It should be noted that, as observed from the vertical displacement depicted in Figure 3.27, the implementation of under-relaxation seems to have a negative effect on the convergence of the solution. Taking into account both the reliability and the accuracy of the solution, it was decided that the default ’accelerationRelaxation’ factor of 0.5 will be used for all further simulations in the present research.

3.6. Summary

This chapter explored the capabilities of the Navier-Stokes/VOF solver in a two-dimensional domain. Wave propagation with and without the presence of an object was considered. Successful grid convergence on the propagation of a fully nonlinear stream function wave (Fenton, 1988) has shown that the two-phase Navier-Stokes/VOF solver was able to provide
a respectable reproduction of the single-phase reference solution of a fully nonlinear stream function wave Fenton (1988). First order convergence was observed in accordance with the first order Upwind and Euler schemes applied for the spatial and temporal discretisation. The use of spatial discretisations of both 150 and 200 p.p.w., lead to an acceptable diffusive error in the order of \( O(10^{-2}) \). No consistent convergence could be obtained with the use of a second order spatial scheme, this lead to the decision to apply first order schemes in all further numerical simulations in the present work.

Wave loading on a partially submerged fixed horizontal cylinder was evaluated using a theoretical and experimental reference solution from Dixon et al. (1979). Although the comparability of the RMS forcing values was frugal, with errors in the order of 10 to 20 percent, excellent results were obtained for the peak loading. Furthermore, it was shown that the numerical results were a better fit with the experimental reference solution than the theoretical solution proposed by Dixon et al. (1979).
In a numerical reproduction of the work by Yu and Ursell (1961), it was shown that the numerical model can generate free surface waves with the forced heave motion of cylinder. It may be noted that, the implementation of the forced oscillating cylinder lead to significant reflections from the domain boundaries, due to the limitations of the wave absorption zones. In spite of this, decent results were obtained for the relative wave amplitude, $R_A$, of the generated waves, showing that the fluid-structure interaction is simulated correctly by the numerical model.

Finally, the Navier-Stokes/VOF solver was used to simulate the heave decay of a free floating two-dimensional horizontal cylinder. The numerical simulation showed satisfactory similarity with the theoretical work by Maskell and Ursell (1970) and experimental work by Ito (1977). It should be noted that some irregularities, in the form of singular force spikes, were observed in the acceleration solution. These are a direct result of the spikes observed in the force signal, which are calculated from the pressure field. These minor signs of instability lead to the evaluation of the under-relaxation method implemented in waveDyMFoam. Here it was shown that an 'accelerationRelaxation' factor successfully improves the stability of the numerical simulation.
4

MESHING TOOLS

4.1. INTRODUCTION

In this chapter the capabilities of two meshing tools are evaluated that are provided with the standard distribution of OpenFOAM®. The validation cases discussed in Chapter 3, where fluid-structure interaction was considered, concerned simple two-dimensional structures. However, the structures encountered in engineering practise will, more often than not, have more complex three-dimensional shapes. It would be a tremendous amount of work to construct the computational mesh of such structures with the method used to generate the meshes in Chapter 3. Here, the meshes were constructed from hexahedra, by defining individual coordinates and connecting those with lines and arcs, see Figure 4.1a. To increase the applicability of the numerical model, simulations with more complex structures should also be achievable without too much effort.

As mentioned before, the discussed cases in Chapter 3 were two-dimensional while a lot of practical engineering problems, regarding free surface flows around objects, are three-dimensional. This change from two- to three-dimensional meshes, will significantly increase the amount of computational cells. Since CFD simulations are generally intensive in terms of computational effort, and the computational cost will only increase as the amount of computational cells rise, this chapter also explores a method for reducing the amount of computational cells by implementing efficient grading to the spatial discretisations.

Section 4.2 elaborates on the implications of snappyHexMesh, a versatile three-dimensional meshing tool available in OpenFOAM®. The three cases from Chapter 3, concerning fluid-structure interaction, were used for the evaluation of this meshing tool. The effectiveness of using multi-grading meshes will be evaluated in Section 4.3. This is a method that allows grading of the spatial discretisation, in the form of local mesh refinement, to reduce the amount of computational cells in the numerical domain. Since the multi-grading function was only recently developed and not available in version 2.3.1 of OpenFOAM®, version 2.4.0 was used to create the basic multi-graded block mesh, after which all subsequent processes and computations were performed in version 2.3.1 to uphold the comparability with the other computations.
This section examines the implementation and implications of the mesh generation utility, snappyHexMesh, supplied with OpenFOAM®. Three cases involving fluid-structure interaction, from Chapter 3, were simulated using computational domains generated with snappyHexMesh. The effectiveness of the mesh generation utility was evaluated.

SnappyHexMesh generates two- and three-dimensional meshes containing hexahedra, and split-hexahedra, from triangulated surface geometries in Stereolithography (STL) format. SnappyHexMesh is more of a mesh sculptor than a mesh generator, because it requires an already existing base mesh. The base mesh can, for instance, be an arbitrarily shaped numerical wave tank and the surface geometry, a horizontal cylinder. SnappyHexMesh approximately sculpts the cylinder in the base mesh, by iteratively refining the base mesh and morphing the resulting split-hex mesh to the surface geometry, as presented in Figure 4.1b. The specifications of the mesh refinement level are flexible and user-defined.

SnappyHexMesh is a more user-friendly method for constructing complex meshes compared to the ‘normal’ method, where the mesh is generated by defining individual vertices, blocks and arcs, as presented in Figure 4.1a. The solutions obtained with the ‘normal’ method, see Chapter 3, were used as the reference solution throughout this section.

It may be noted that, by applying a high refinement level to a mesh with the same reference mesh size, the computational cells near the cylinders surface become significantly smaller, as illustrated in Figure 4.1. The increase of the overall number of computational cells in the numerical domain was in the order of 10 percent, see Table 4.1. It may therefore

<table>
<thead>
<tr>
<th>Wave loading</th>
<th>Forced heave</th>
<th>Free heave</th>
</tr>
</thead>
<tbody>
<tr>
<td>cells [-]</td>
<td>cells [-]</td>
<td>cells [-]</td>
</tr>
<tr>
<td>time [-]</td>
<td>time [-]</td>
<td>time [-]</td>
</tr>
<tr>
<td>(a)Reference 4636</td>
<td>30</td>
<td>16600</td>
</tr>
<tr>
<td>Snappy 5396</td>
<td>83</td>
<td>17882</td>
</tr>
<tr>
<td>Increase[?]</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>(b)Reference 18144</td>
<td>74</td>
<td>65400</td>
</tr>
<tr>
<td>Snappy 19482</td>
<td>227</td>
<td>68198</td>
</tr>
<tr>
<td>Increase[?]</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>
4.2. **SNAPPYHEXMesh**

Figure 4.2: Vertical forcing, normalized by the buoyancy force of the totally submerged cylinder, as a function of time normalized by the wave period. Comparison of normal mesh generation and use of snappyHexMesh. The spatial discretisations of 50 and 100 p.p.w. were applied and the relative wave amplitude and axis depth are 0.5 and 0.0.

Figure 4.3: Time step, normalized by the wave period, as a function of time normalized by the wave period from the wave loading case. Comparison of normal mesh generation and use of snappyHexMesh.

be said that snappyHexMesh is not only a more user-friendly meshing tool, but also allows introduce local mesh refinement. In the following subsection the results summarised in Table 4.1 are discussed per case.

**4.2.1. Wave loading on a partially submerged fixed horizontal cylinder**

The effectiveness of the SnappyHexMesh utility was examined with the computation of wave loading on a horizontal cylinder, see Section 3.3 for a thorough description of the set-up of this case. The relative wave amplitude and axis depth for this particular case were 0.5 and 0.0. The simulation was performed with both a spatial discretisation of 50 and 100 p.p.w.

The theoretical, experimental, reference and ‘snappy’ solutions of the vertical forcing were presented in Figure 4.2. Here, the vertical forcing was normalized by the buoyancy force of the totally submerged cylinder and presented as a function of time, normalized by the wave period. It can be observed that, for both spatial discretisations, the similitude between the numerical and experimental solution was improved for the mesh generated by the snappyHexMesh utility. Especially the reproduction of the secondary loading peak was improved. It was believed that this result can mainly be credited to the smaller cells in the vicinity of the structure, increasing the accuracy with which the pressures were calculated and integrated.
over the cylinder surface.

From Table 4.1, it was found that the amount of computational cells, for the low and high resolution, had only increased by 16 and 7 percent, while the computational time had increased with 181 and 206 percent respectively. This difference was believed to be caused by the fact that the computational time is largely governed by the time step. In Figure 4.3, the time steps of all four computations are presented. The maximum Courant number, fixed at \( \text{Co} = 0.20 \), is controlled by the size of the computational cells, velocity and time step, see equation (3.2). Since the velocity of the flow is expected to remain approximately unchanged and the smallest computational cell was significantly smaller, for the mesh created with snappyHexMesh, this results in a smaller time step. It may be noted that, due to the smaller time steps, the total amount of time steps was increased by 130 and 190 percent for the low and high resolution. This seems to be closely related to the increase in computational time.

4.2.2. Surface waves generated by the forced oscillation of a horizontal cylinder

For the evaluation of the snappyHexMesh utility with respect to a moving mesh, the wave generation by a forced oscillating cylinder case was performed. An elaborate description of the numerical set-up of this case can be found in Section 3.4. A heave amplitude and period of 0.00615 m and 0.92 s were used. The simulation was preformed with both 50 and 100 p.p.w. spatial discretisation.

Figure 4.4 depicts the free surface elevation, normalized by the experimental wave amplitude at infinity, as a function of the time, normalized by the heave period. Here, it can be observed that, for the case with lower resolution of 50 p.p.w., the obtained amplitude was significantly enhanced by using snappyHexMesh, while the amount of cells in the numerical domain was only increased by 8 percent. This improvement was far less apparent for the case with higher resolution of 100 p.p.w. The lower improvement in the higher resolution case, may well be due to the fact the accuracy of the simulation of wave propagation, in contrast to wave loading, to a smaller degree governed by the smaller cell sizes in the direct vicinity of the cylinder surface. Instead it is more dependent on the spatial discretisation in the direction of wave propagation further away from the cylinder and this was more or less the same for both the simulations, as can be seen from Figure 4.1.

From Table 4.1, it can be seen that the amount of computational cells, for the low and high resolution, have only increased by 8 and 4 percent and in contrary to the high increase in computational time observed in the wave loading case, a minor decrease in computational time was observed. It is at present unclear as to how the decrease in computational time came about, there are many possible causes of which a relative change in the performance level of the numerical cluster is just one. As explained in the previous case, the Courant criteria would dictate a smaller time step with smaller computational cells which would result in an increased computational time. From the time steps, represented in Figure 4.5, it can be observed that there is a minor increase in time step, during the initialisation phase, where the motion of the cylinder is ramped up, after which the time steps converge towards an equilibrium. This more or less stable signal shows a periodic oscillation which may be caused by the velocity of the heaving cylinder. Note that there is hardly any difference in the mean of the time step signals between the snappyHexMesh and the reference solutions of equal spatial discretisation.
4.2. SNAPPYHEXMesh

Figure 4.4: Surface elevation, normalized by the experimental wave amplitude at infinity, as a function of time normalized by the wave period. Comparison of the reference solution and the solution using snappyHexMesh. The surface elevation signal was obtained from a wave gauge located at one wavelength from the cylinder. The spatial discretisations of 50 and 100 p.p.w. were applied and the heave amplitude and period of the cylinder were 0.00615 m and 0.92 s.

Figure 4.5: Time step, normalized by the wave period, as a function of time normalized by the wave period from the forced heaving cylinder case. Comparison of normal mesh generation and use of snappyHexMesh.

4.2.3. FREE HEAVE DECAY OF A HORIZONTAL CYLINDER

The numerical set-up for the heave decay of a free moving horizontal cylinder, was similar to the set-up described in Section 3.5. The difference being that the mesh was generated with snappyHexMesh, resulting in a higher resolution in the vicinity of the cylinder surface.

The simulation was preformed with a spatial discretisation of both 10 and 20 p.p.c.d. The results of the decaying heave motion, from the experimental data and numerical simulation with ‘normal’ and snappyHexMesh set-up, are presented in Figure 4.6. Here, the vertical displacement of the COG of the cylinder is normalized by the initial displacement with respect to the equilibrium. Although differences were small, it can be observed that, for both the low and the high resolution, the result improved by use of snappyHexMesh. This manifests itself in a smaller amplitude of the transient response of the cylinder, which is closer to the experimental result.

The amount of computational cells increased by approximately 3 percent for both the low and high resolution cases. However, the computational time of the lower resolution case was increased with 45 percent, while the increase for the higher resolution case was only 13 percent. When looked at the amount of computational time steps, an increase of approxi-
4. Meshing tools

Figure 4.6: Vertical displacement of the center of mass, normalized by the initial displacement $d'$, as a function of time normalized by $\sqrt{g/a}$. Comparison of the reference solution and the solution using snappyHexMesh. The spatial discretisations of 10 and 20 p.p.c.d. were applied.

...mately 70 and 30 percent was observed for the low and high resolution cases. This seems to be related to total computational time.

4.3. Multi-mesh grading

Using the standard single expansion ratio to describe mesh grading permits only “one-way” grading within a mesh block. To refine the mesh around the free surface, it is possible to describe the numerical domain by defining several blocks with one grading per block. However, this can become challenging as desired numerical domains become more complex. Furthermore, it has been experienced that using this technique to refine the mesh around the free surface poses problems, when snappyHexMesh is used to include a geometrical surface from a STL file. The snappyHexMesh utility defines an area around structure in which the mesh is refined, see Section 4.2, and if this area extends over several mesh blocks, failure of the meshing process was experienced.

OpenFOAM® version 2.4.0 and newer, include a multi-grading functionality that can subdivide a blockmesh in several sections in any given direction. Different grading levels can then be applied within each division, see Figure 4.7(b). Using this functionality the amount of computational cells can significantly be reduced, see Table 4.2, while maintaining an identical spatial resolution in the vicinity of the free surface. The following sections will examine the implementation of this functionality using the results obtained with snappyHexMesh on a uniform mesh, see Section 4.2, as a reference solution.

In the following evaluation, the multi-graded mesh was defined in a way that the resolution around the free surface, a horizontal strip slightly larger than the cylinder diameter, has

<table>
<thead>
<tr>
<th>Case</th>
<th>Wave loading cells [-]</th>
<th>Wave loading time [-]</th>
<th>Forced heave cells [-]</th>
<th>Forced heave time [-]</th>
<th>Free heave cells [-]</th>
<th>Free heave time [-]</th>
<th>Decrease[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>19482</td>
<td>227</td>
<td>68198</td>
<td>1329</td>
<td>223239</td>
<td>1970</td>
<td>34</td>
</tr>
<tr>
<td>Grading</td>
<td>12919</td>
<td>208</td>
<td>41893</td>
<td>1255</td>
<td>125077</td>
<td>1205</td>
<td>39</td>
</tr>
<tr>
<td>Decrease</td>
<td>8</td>
<td>39</td>
<td>6</td>
<td>44</td>
<td>39</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3. MULTI-MESH GRADING

Figure 4.7: Visualisation of the mesh around the cylinder boundary surface for the mesh generated for the wave generation case, where the reference spatial discretisation was 100 p.p.w. (a) the uniform reference mesh and (b) multi-gradient mesh, or ‘grading’.

Figure 4.8: Vertical forcing, normalized by the total buoyancy force of the cylinder, as a function of time normalized by the wave period. Comparison of uniform mesh and multi-graded mesh with a resolution of 100 p.p.w. around the free-surface. The relative wave amplitude and axis depth are 0.5 and 0.0.

approximately the same spatial discretisation as the uniform reference mesh. In the remainder of both the top and bottom part of the computational domain, a grading of 25 percent was applied, as presented in Figure 4.7b. It may be noted that small deviations can be present in the vicinity of the cylinder surface, since the refinement area used by the snappyHexMesh utility might contain various cell sizes in the base mesh and this could lead to discrepancies in the final mesh.

4.3.1. WAVE LOADING ON A HORIZONTALLY FIXED CYLINDER

The effectiveness of the multi-grading mesh was evaluated using the wave loading case discussed in the previous section. In this subsection, the multi-graded mesh will be compared to the reference solution with a resolution of 100 p.p.w.

From Table 4.2, it can be observed that, for the wave loading case, the overall amount of computational cells was reduced by approximately 34%, while only minor differences were experienced in the vertical force profile, see Figure 4.8. Minor discrepancies in the vertical forcing profile were observed around the secondary loading peak. This was most likely caused by the minor differences between the cell sizes in the vicinity of the cylinder, as this resolution was determined to be an important influence on the accuracy of the computation, as discussed in Section 3.3. In general the magnitude of the forcing was considered to
be consistent with the results obtained with the computationally more expensive reference solution. It may be noted that, although the amount of computational cells was decreased by 34 percent, the computational time only decreased by 8 percent. This is believed to be caused due to the fact that the computational time is, as least for a large part, governed by the Courant criteria, as discussed in the previous section.

4.3.2. Surface waves generated by a forced heaving cylinder

The multi-grading mesh was evaluated using the wave generation case previously discussed, where the resolution of the mesh around the free surface of the multi-graded mesh was approximately 100 p.p.w., which was consistent with the spatial discretisation of the uniform mesh of the reference solution.

By implementing the multi-graded mesh the total number of computational cells was reduced by approximately 39 percent, see Table 4.2. It is known that the propagation of waves, in a numerical domain, is highly dependent on the amount of cells in the direction of propagation. Since the grading was applied in the vertical direction of the domain, the amount of computational cells in the direction of propagation remains equal. From Figure 4.9 it can be observed that the implementation of the multi-graded mesh did not result in significant changes to the numerical solution. From Table 4.2, it is also seen that, without significant discrepancies to the solution, the computational time had decreased by approximately 8 percent.

4.3.3. Decay of heave motion of a free horizontal cylinder

A final evaluation of the multi-grading mesh was performed on the case concerning the heave decay of a freely oscillating horizontal cylinder. The results of the numerical simulation were presented in Figure 4.10. The resolution of the mesh around the free surface of the multi-graded mesh was approximately 20 p.p.c.d., which was consistent with the spatial discretisation of the uniform mesh of the reference solution.

From Table 4.2, it can be seen that the total number of computational cells was reduced by approximately 44 percent. The slightly larger decrease of the number of computational cells, can be attributed to the fact that the vertical direction of this case is relatively large compared to the previous cases, i.e. the efficiency of this method increases as deeper numerical wave
4.4. Summary

The results, presented in Figure 4.10, show that the vertical displacement of the COG of the cylinder was approximately identical for both the multi-grading mesh and the uniform reference mesh. It may be noted that the computational time was decreased by approximately 39 percent, which is in the same order of magnitude as the decrease of the number of computational cells.

4.4. Summary

In this chapter, the capabilities of two meshing tools, provided with the standard distribution of OpenFOAM®, were investigated. The validation cases discussed in Chapter 3, where fluid-structure interaction was considered, were re-simulated using meshes created with these tools. It was shown that snappyHexMesh, a versatile three-dimensional meshing tool, could be used to create cylindrical boundary surfaces in a standard block mesh and that this technique could also be applied on a multi-graded block mesh, generated with a multi-grading tool available in the slightly newer, OpenFOAM® version 2.4.0. and higher. The numerical simulations showed no unexpected instabilities as a result of the use of either of these methods. However, an increase computational time was observed for the use of snappyHexMesh. This was attributed to the smaller computational cells in the vicinity of the cylinder, when compared to the reference mesh, which had a large influence on the time step due to the Courant criteria applied in this research. Furthermore, it was shown that the amount of computational cells could be significantly reduced by the use of the multi-grading tool, resulting in smaller computational times without a significant loss of accuracy.

Figure 4.10: Vertical displacement of the center of mass, normalized by the initial displacement $d'$, as a function of time normalized by $\sqrt{g/a}$. Comparison of uniform mesh and multi-graded mesh with a resolution of 20 p.p.c.d. around the free-surface.
5

**FLUID-STRUCTURE INTERACTION IN 3D**

5.1. **INTRODUCTION**

This chapter explores the capabilities of the Navier-Stokes/VOF solver in a three-dimensional set-up with respect to fluid-structure interaction. The aim of this chapter is validate the numerical model by comparison to laboratory measurements performed at MARIN.

This chapter is structured as follows; first a short introduction is given on the ‘TO2 Floating Wind’ project and the physical model campaign. This is followed by a comparison of measured and computed decay tests. First the results from a free heave decay test are compared to results from Dunbar et al. (2015), secondly, the results from a free pitch and roll decay simulation are compared to experimental data and finally the results from moored heave, pitch and roll decay tests are evaluated with respect to the results from physical model tests.

5.2. **TO2 FLOATING WIND PROJECT**

The 'Toegepaste Onderzoek Organisaties' or TO2-institutes are a group of Dutch research institutes and universities which are stimulated by the government. In 2015 several TO2-projects, focussed around various societal challenges, were carried out by these institutes. The present research is part of a TO2-project which is related to ‘safe, clean and efficient energy’. The participating companies in this so called ‘TO2 Floating Wind’ project are the Dutch research institutes; ECN, MARIN, NLR and Deltares. The project has two main task areas, which are a) motion and control of the floating wind turbine and b) helicopter movements in offshore wind farms. Within task a) ECN, MARIN and Deltares focussed on extending their knowledge and integral approach of simulation and testing of floating wind turbines. Numerical simulation were performed by ECN, to improve the control strategy for the controllable pitch blades of the floating wind turbine. MARIN refurbished their old high speed basin, equipping it with a new wave maker and a state of the art wind field generator to perform experiments with the floating wind turbine model, see Section 5.3. The main focus of Deltares was to extend the knowledge of numerical simulation of fluid structure interaction with a floating object. Since the maintenance of offshore wind farms, be it floating or fixed, will partly be done by helicopters, the knowledge on helicopter movements in and around the wakes of wind turbines is essential. NLR and ECN joint forces to extend the knowledge in this field.
5.3. PHYSICAL EXPERIMENTS

As part of the TO2 project, physical experiments with a floating offshore wind turbine were performed in one of the wave tanks at MARIN. Unless otherwise stated all dimensions and properties will be presented in model scale, for which a 1:50 Froude scaling was applied. This section will discuss the floating wind turbine model and the experimental set-up.

5.3.1. PHYSICAL WAVE TANK

MARIN had transformed their old high speed basin into the new concept basin, this transformation included the installation of a new wave-maker and the development of a state-of-the-art wind field generator. This wind field generator is neither depicted nor discussed in the present research, as no numerical simulations will be preformed with a generated airflow.

An illustration of the 200 m long wave tank is depicted in Figure 5.1. The width of the wave tank is 4 m as is also the water depth. The model was positioned approximately in the center of the wave tank. The original design of the OC5 mooring system had a total of three mooring lines, one connected to each of the heave plates of the sub-structure of the floating wind turbine, and spread out evenly over the seabed. Due to the limitations of the wave tank, MARIN had to design and develop a different mooring system. They opted for a front and back mooring line and two horizontal linear springs, this would give the system restoring properties close to those of the original three line system. In order to get the balance right, an extension arm was constructed. All attachment points, dimensions and properties of the mooring system can be found in Appendix D. It may be noted that the front mooring line is slightly longer than the back mooring line, which had the original design length. This extension was implemented, because an earlier model campaign had shown that the original line length was considered to be to short for extreme conditions loading conditions, which could lead to complete suspension of the mooring line.
5.3. PHYSICAL EXPERIMENTS

Figure 5.2: Schematic representation of the sub-structure of the floating wind turbine model. (a) left side view, (b) back view. Description of the various parts: A = transition piece, B = center column, C = offset column and D = heave plate. Description of locations: I = COG, II = front mooring, III = back mooring, IV = left spring and V = right spring.

5.3.2. OC5 FLOATING OFFSHORE WIND TURBINE MODEL

The design of the sub-structure of the OC5 FOWT, hereafter referred to as the floater, was inspired on the semi-submersibles found in the offshore oil and gas industry. This generic model was created by a worldwide collaboration of research institutes to stimulate the increase in knowledge about FOWT. A schematic representation of the floater is depicted in Figure 5.2. More detailed description of the floating wind system, including dimensions and other physical properties like the moment of inertia, draft and position of the COG can be found in Appendix D. Note that these are the values for the experimental model and that this includes the floater, tower and nacelle.

The sub-structure of this particular model was a try-floater, meaning it has three main vertical columns. Heave plates were attached to the bottom of these, in order to increase the stability of the floater. The three main columns are connected to a, more slender, center column via support braces. The turbine tower, with at its top the nacelle and blades, is connected to this center column. As mentioned in the previous subsection, an extension arm was constructed to allow the back mooring line to be behind the aft of the floater.

5.3.3. EXPERIMENTS PERFORMED AT MARIN

Several tests were performed during the model campaign at MARIN. These tests included: wave calibration tests, decay tests with and without the mooring system and some wave loading cases with and without wind generation. The cases with an induced wind field will be not discussed within the scope of this report.

Four different wave cases, presented in Table 5.1, were used for wave calibration tests. These can be divided into two regular and two irregular cases. For the two regular waves the wave height and period are given, and for the irregular, JONSWAP, wave spectra the significant wave height and peak wave period, as well as a peak enhancement factor, \( \gamma \), are provided. These properties were based on the wave climate on the East Coast of the United States. The wavelength presented here was determined using the deep water limit, \( \lambda = \frac{g}{2\pi} T^2 \), note that the MARIN basin had a depth of 4.0 m, which comes down to 200 meters in real scale.

Decay tests were performed in still water to gain insight in the physical properties of the floating wind turbine. Pitch and roll decay test were performed without the presence of any mooring system. Heave, pitch, roll and yaw decay test were performed with the floating wind turbine with the mooring system installed. Please note that the mooring system was altered as discussed in Subsection 5.3.1. The decay tests will be discussed and analysed in more detail in Section 5.4 and 5.5.

The four wave loading cases, as presented in Table 5.1, were part of the model test campaign performed with the moored floating wind turbine. Regretfully the data analysis, of
5. FREE HEAVE, PITCH AND ROLL DECAY OF A FLOATING STRUCTURE

This section explores the effectiveness of the waveDyMFoam solver to accurately describe the heave, pitch and roll motion of the floating offshore wind turbine. In these computations consider the free floating structure, so, no mooring system is taken into account. First the numerical set-up is described after which the experimental and numerical results are reviewed.

5.4.1. NUMERICAL SET-UP

The numerical domain is presented in a schematic manner in Figure 5.3. It was set-up to closely resemble the experimental set-up. First a cubically shaped domain was formed, here, the length, $l$, and width were equal to 4.0 m while the height was 4.624 m, of which 4.0 m was the water depth, $d$, and the rest was to leave enough space for the motions of the floater and the presence of the air phase, $a$. Relaxation zones of two-thirds of a meter were implemented at the inlet and outlet boundary. Note that for these decay simulations, both of these relaxation zones were used to absorb waves generated by the floater. As the physical experiments these cases, was not performed in time to be included in the present research. To evaluate the performance of the waveDyM Foam solver, it was therefore opted to perform a proof-of-concept for the irregular wave loading of a moored floating wind turbine, as a code-to-experiment validation was not possible, this will be discussed in Chapter 6. Please note that, previous to this proof-of-concept, a thorough evaluation of the free and moored decay tests is performed, the results of which are described in the present chapter.

Table 5.1: The wave heights and periods, in model scale, for the wave calibration and wave loading tests. The peak enhancement factor, $\gamma$, of the JONSWAP spectrum, was based on the wave climate of the East Coast of the United States. The wave lengths were calculated using the deep water limit, $\lambda = \frac{g}{2}\pi T^2$. Please note that all data is presented in 1:50 Froude scaling.

<table>
<thead>
<tr>
<th>Description</th>
<th>Type</th>
<th>Wave height [m]</th>
<th>Period [s]</th>
<th>Wave length [m]</th>
<th>$\gamma$ [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular 7</td>
<td>regular</td>
<td>0.142</td>
<td>1.7112</td>
<td>4.572</td>
<td>-</td>
</tr>
<tr>
<td>Regular 10</td>
<td>regular</td>
<td>0.21</td>
<td>2.0023</td>
<td>6.385</td>
<td>-</td>
</tr>
<tr>
<td>Operation</td>
<td>JONSWAP</td>
<td>0.142</td>
<td>1.7112</td>
<td>4.572</td>
<td>2.20</td>
</tr>
<tr>
<td>Design</td>
<td>JONSWAP</td>
<td>0.21</td>
<td>2.0023</td>
<td>6.385</td>
<td>3.00</td>
</tr>
</tbody>
</table>
took place in a 4.0 m wide wave tank with concrete walls, both the lateral boundaries of the numerical domain were set up as fully reflecting walls.

The domain was then graded in the vertical direction using the multi-grading technique, as discussed in Section 4.3. The multi-gradient mesh was defined in such a way that a horizontal strip around the free surface, over the full length and width of the domain, had a spatial discretisation of 5 p.p.c.d. Here, the cylinder diameter, $D = 0.24 \, m$, of the offset columns was used as a reference. A visualisation of the computational mesh is presented in Figure 5.4 and a detailed representation of the computational mesh of the floater surface boundary is provided in Appendix D.

The floater is the only part of the floating wind turbine system that was modelled for the numerical simulations. Note that the physical properties, such as the draft, mass and moments of inertia of the structure are for the complete floating wind turbine, so, including the tower and nacelle. These values, presented in Table 5.2, were determined by MARIN for their physical model. The floating wind turbine, of which only the floater was modelled, was defined as a ridged body.

The design drawings of the OC5 model were utilised to generate an STL file of the floater, after which snappyHexMesh could be utilised to sculpt the surface boundary face in the base.

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Unit</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft (mooring)</td>
<td>$T_{\text{moored}}$</td>
<td>[m]</td>
<td>0.40</td>
</tr>
<tr>
<td>Draft (no mooring)</td>
<td>$T_{\text{free}}$</td>
<td>[m]</td>
<td>0.3898</td>
</tr>
<tr>
<td>Mass</td>
<td>$M$</td>
<td>[kg]</td>
<td>111.664</td>
</tr>
<tr>
<td>Roll inertia</td>
<td>$I_{xx}$</td>
<td>[kg/m$^2$]</td>
<td>49.768</td>
</tr>
<tr>
<td>Pitch inertia</td>
<td>$I_{yy}$</td>
<td>[kg/m$^2$]</td>
<td>47.556</td>
</tr>
<tr>
<td>Yaw inertia</td>
<td>$I_{zz}$</td>
<td>[kg/m$^2$]</td>
<td>43.814</td>
</tr>
</tbody>
</table>
Figure 5.5: Visualisation of the numerical simulation of the free pitch decay test of a floating wind turbine. Here, the iso-contours with $\alpha \geq 0.5$ is used to visualise the water. The position of the floater is visualised at time interval (a) $t/T = 0.0$, (b) $t/T = 0.5$, (c) $t/T = 1.0$ and (d) $t/T = 1.5$.

mesh, as discussed Section 4.2. It should be noted that, even with snappyHexMesh, the generation of a reliable surface boundary mesh for such a complicated three-dimensional structure is not trivial. For this a 4-3-2 refinement was applied, this meant that the cells furthest from the boundary surface would have half the size of the reference mesh, of 5 p.p.c.d., while the cells close to the surface boundary would be approximately one twenty-fourth of the size of the reference mesh. This level of mesh refinement was necessary to make sure that the mesh did not contain any errors. This resulted in a total number of computational cells in the domain of almost 1.5 million, as presented in Table 5.3, while the base mesh only had 268128 computational cells.

For all the heave, pitch and roll simulations the initial position of the floater had to be modelled in the base mesh. This meant that, as OpenFOAM® lacks the ability to rotate or translate an STL file, the STL file itself had to be customised. This just goes to show that the set-up of these cases is difficult to say the least, and small errors are easily made. The same can be said for the set-up and analysis of physical model tests, which makes the comparison of these two far from trivial.

The free pitch decay test is visualised in Figure 5.5, to give an idea of how such a numerical simulation looks. The iso-contours with $\alpha \geq 0.5$ is used to visualise the water. The position of the floater is captured at its maximum inclination angle the first 1.5 period of pitch motion. Small disturbances can be observed around the surface piercing cylinders, it is questionable

Table 5.3: Spatial discretisation and total number of computational cells of the numerical domain used for both the free and restrained decay cases. Here the spatial discretisation is defined for the cells around the free surface in the base mesh.

<table>
<thead>
<tr>
<th>Points per cylinder diameter</th>
<th>Cell size, $\Delta$ [m]</th>
<th>Computational cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.048</td>
<td>1487575</td>
</tr>
</tbody>
</table>
whether these are a result the motions of the floater, small errors in the numerical model or a result of the way the free surface is visualised.

### 5.4.2. Analysis of the Free Decay Simulations

It is likely that there is a difference between the physical and numerical model, be it by a fault in the numerical calculation of the buoyancy force or by discrepancies between the properties described in Table 5.2 and the actual test model. This may, among others, cause the draft of the numerical model to be different from the one presented for the model tests. These discrepancies may also have an effect on the heave, pitch and roll behaviour of the floater. Before evaluating the decay tests, the draft of the numerical model was established.

A numerical simulation was performed where the floater was initialised in the proposed equilibrium position of the physical model, with a draft of 0.3898 m. The draft of the floater is represented in Figure 5.6. The results of this simulation were quantified in Table 5.4, from which it can be observed that the there was in fact a difference between the draft of the physical and numerical model. The draft of the numerical model was found to be 3.64 percent larger, which amounts to approximately 0.7 m in real scale. This newly determined draft was applied as the standard for the free decay tests discussed in this section. All other physical properties, as presented in Table 5.2, remained unchanged. Note that, to some degree, the change in draft will have an effect on the dynamic response of the floating wind turbine. Furthermore, it should be noted that, for the simulation for the draft evaluation, no restrictions were defined for the floater. The pitch, roll, and yaw angles experienced during this draft examination were in the order of $O(10^{-2})$ degrees, which give an indication that the numerically computed floater was balanced correctly.

Now that the correct numerical draft is known, the validation cases could be simulated and evaluated, starting with the free heave decay of the floating structure. As no model experiments were performed for this particular case, a code-to-code comparison was performed with the numerical solutions provided by Dunbar et al. (2015) as a reference. In their research they provided a ’tightly coupled’ version of the OpenFOAM® interDyM Foam solver and com-

<table>
<thead>
<tr>
<th>Description</th>
<th>Draft physical [m]</th>
<th>Draft numerical [m]</th>
<th>Increase [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without mooring</td>
<td>0.3898</td>
<td>0.404</td>
<td>3.64</td>
</tr>
</tbody>
</table>

Figure 5.6: Draft of the floating wind turbine, i.e. position of the keel with respect to the still water level. Initial position of the floater was bases on the draft of the experimental model.
pared it to FAST, a NREL engineering tool. Among others, they performed two heave decay tests with the OC5 semi-submersible floating wind turbine. It should be noted, that although no intricate mooring system differences had to be overcome, the comparison of these results is far from a validation of any sort. It should be regarded more, as an initial step in the evaluation of the capabilities of the waveDyMFoam solver. Furthermore, it is a check whether the under-relaxation technique applied in this research, is as capable of providing a stable numerical result, as the ‘tightly coupled’ model presented by Dunbar et al. (2015).

The results of the first of these two heave decay tests is presented in Figure 5.7. A comparison between heave displacement computed by the three different numerical models is presented in Figure 5.7a, where the displacement is normalized by the initial displacement. For this first heave decay test this was equal to 0.02 m, or 1 m in real scale, while the second heave decay test, presented in Figure 5.8, is performed with an initial displacement six times as large. From the results of the first decay test it can be observed that the results of the three different numerical models are in close proximity of each other in terms of the computed heave period. This is quantified by the evaluation of the first response amplitude, as presented in Table 5.5. It can also be seen that damping of the heave motion, computed by the waveDyMFoam solver, is much higher than that in the simulations performed by Dunbar et al. (2015). This becomes even more clear from the displacement of the, more extreme, second heave test, presented in Figure 5.8a. Although the period is still in good correspondence with the one computed with the ‘tightly coupled’ interDyMFoam solver, the error in the amplitude, computed after one heave period, is almost forty percent. Aside from the differences
5.4. **Free Heave, Pitch and Roll Decay of a Floating Structure**

Figure 5.8: Heave displacement of the COG (a), normalized by the initial displacement of 0.12 m. The acceleration (b), normalized by the maximum acceleration. Both presented as a function of time, normalized by the obtained heave period. Comparison of the result obtained with the waveDyMFoam solver and results from Dunbar et al. (2015), both from FAST and their ‘tightly coupled’ interDyMFoam solver.

present in the amount of damping of the system, which may have numerous causes, it can be observed that no significant disturbances were observed in the accelerations of the floater, as presented in Figure 5.7b and 5.8b. This indicates that, although there were differences between the numerical solutions, the waveDyMFoam solver was able to produce acceptable results in terms of numerical stability by using the under-relaxation method.

Since the capability of the waveDyMFoam solver was proven to provide adequate solutions to the flow problems concerning the heave decay of a complex three-dimensional structure, it could now be used in the comparison against experimental results from the decay tests performed at MARIN.

The experimental and numerical pitch angles of the floating wind turbine are presented in Figure 5.9a. Here, the pitch angle, normalized by the initial inclination angle of 3.13 degrees, is presented as a function of time, normalized by the experimentally determined pitch period. It should be noted that, in order to isolate the pitch motion, the rotations in roll and yaw direction were restricted. The model was however allowed to translate in all directions.

<table>
<thead>
<tr>
<th>Description</th>
<th>Initial displacement</th>
<th>Error period [%]</th>
<th>Error amplitude [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heave</td>
<td>0.02 m</td>
<td>0.7</td>
<td>17.6</td>
</tr>
<tr>
<td>Heave</td>
<td>0.12 m</td>
<td>1.5</td>
<td>38.1</td>
</tr>
</tbody>
</table>

Table 5.5: Comparison between the numerical solutions obtained with the waveDyMFoam solver and the ‘tightly coupled’ interDyMFoam solver from Dunbar et al. (2015). Here the absolute period and amplitude error were determined after one wave period.
tions making it a 4-DOF system. This technique was also applied to all further pitch and roll cases in the present research. In general a good correspondence was found between the experimental and numerical result, as can be seen from the amplitude and period comparison presented in Table 5.6. It is however worrisome that the numerically computed response amplitude at $t/T = 0.5$ was slightly higher than the initial inclination angle. This increase indicates that the numerical model of the floating wind turbine is either out of balance, the illegitimate assumption that the axis of rotation could be fixed, a small error was made in the set-up of the simulation or that there is some underlying numerical problem. Despite the small disturbance, the numerically computed result fits quite well with the experimentally obtained solution. Since the time available for this thesis is limited, it was decided not to take this small disorder into further consideration. From Figure 5.9, where angular accelerations of the floating wind turbine are displayed, it can be seen that the numerical solution remains relatively stable up to around $t/T = 2$, where the solution starts displaying oscillations, as was observed in the work of Dunbar et al. (2015). This indication of instability was also observed, in a similar magnitude and at a similar time interval, in the roll accelerations.

Table 5.6: Comparison between the numerical solutions and experimental data from MARIN in the decay tests without mooring. Here the absolute period and amplitude error were determined after one wave period.

<table>
<thead>
<tr>
<th>Description</th>
<th>Initial displacement</th>
<th>Error period [%]</th>
<th>Error amplitude [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch</td>
<td>3.13 deg</td>
<td>1.42</td>
<td>5.58</td>
</tr>
<tr>
<td>Roll</td>
<td>3.16 deg</td>
<td>0.49</td>
<td>2.14</td>
</tr>
</tbody>
</table>
Figure 5.10: Roll angle (a), normalized by the initial angle of inclination of 3.13 degrees. The angular acceleration (b), normalized by the maximum angular acceleration. Both presented as a function of time, normalized by the experimentally obtained roll period. Comparison of the result from the free decay test performed at MARIN and numerical solution obtained with the waveDyMFoam solver.

presented in Figure 5.10b. This remarkable similarity could have many reasons, and therefore needs more elaborated investigation, such an analysis does however not fit within the scope of this thesis. In general it was shown that the waveDyMFoam solver was able to provide accurate solutions to the flow problems concerned with the free decay tests of a complex floating structure, where it should be noted that the execution of a seemingly simple physical decay test is far from trivial. Also, indications of instability in the waveDyMFoam solver were observed for the cases that concerned angular displacement.

5.5. **MOTION DECAY OF A FLOATING STRUCTURE WITH RESTRAINTS**

This section examines the waveDyMFoam solver with respect to the accurate computation of the heave, pitch and roll motion of the moored floating offshore wind turbine. First the numerical set-up is described, with special attention to the mooring system, after which the experimental and numerical results are evaluated.

5.5.1. **NUMERICAL SET-UP**

The numerical set-up, used for the moored decay cases, is illustrated in Figure 5.11. The dimensions and spatial discretisation of the numerical domain, $\Gamma$, were adopted from the free decay set-up, described in Section 5.4. The main difference being the presence of the mooring system. As illustrated in the figure, the dimensions of the mooring system do not limit the size of the numerical domain. This is because the restraints are not a part of the
5. FLUID-STRUCTURE INTERACTION IN 3D

Figure 5.11: Schematic representation of the numerical set-up used in the moored heave, pitch and roll decay cases. The dimensions of the domain, \( \Gamma \), are the same as in Figure 5.11.

Figure 5.12: Visualisation of the numerical simulation of the moored heave decay test of a floating wind turbine. Here, the iso-contours with \( \alpha \geq 0.5 \) is used to visualise the water. The position of the floater is visualised at time interval (a) \( t/T = 0.0 \), (b) \( t/T = 0.5 \), (c) \( t/T = 1.0 \) and (d) \( t/T = 1.5 \).

mesh, but simulated as a force in the attachment point on the floater. It should be noted that mooring line implementation, provided by Niels Jacobsen (Researcher/advisor at Deltares), provides a post-processing tool for the visualisation of the catenary mooring lines. This can be observed from the visualisations of the moored heave decay test presented in Figure 5.12.

The mooring system consists of four restraints: two catenary mooring lines and two horizontal linear springs. The main properties of both the mooring lines and the linear springs are presented in Table 5.7. The attachment point and anchor coordinates can be found in Appendix D. Here, the attachment points are denoted for the equilibrium position of the floater. Please note that these had to be altered according to initial conditions of each individual decay case.

Most of the properties presented in Table 5.7 were the directly copied from the design drawings of the mooring system for the physical experiment. However, no measurement data was available of the tension in the horizontal springs. Therefore, an estimated pre-tension of 0.256N, provided by MARIN, was adopted. Knowing the pre-tension, the length of the springs at rest could be calculated, according to the formulas for linear springs described in Subsection 2.3.5. Possible discrepancies between the estimated pre-tension and the actual
pre-tension in the physical model tests could lead to irregularities in the results. Please note that this is just one of many causes for the conceivable variations between the experimental and numerical results.

5.5.2. Analysis of the Moored Decay Simulations

In the analysis of the free decay cases, discussed in Section 5.4, it was already observed that there were small differences between the numerical and physical properties of the floater. The addition of a mooring system, especially with the uncertainties in the pre-tension of the horizontal springs, made it more likely that the floater would have a different equilibrium position in the numerical computations. Therefore, again a numerical simulation was performed where the floater was initialised in the equilibrium position of the physical model, with a draft of 0.40 m. The vertical displacement of the floater is presented as a function of time in Figure 5.13, where the vertical displacement is represented as the draft of the floater. A couple of observations can be made from the result of the draft evaluation. First, the most obvious observation, that there was in fact a difference between the draft of the physical and numerical model. An increase in draft of 2.75 percent, as represented in Table 5.4, was observed, which amounts to approximately 0.011 and 0.55 meter in model and real scale respectively. The increased draft of the numerical model would be applied as the standard for the

Table 5.7: Properties of the restraining system of the physical floating wind turbine model, as provided by MARIN. $W_{sub}$ is the submerged weight of the mooring line and $k_{spring}$ is the spring stiffness.

<table>
<thead>
<tr>
<th>Description</th>
<th>Length [m]</th>
<th>$M_{sub} / k_{spring}$ [N/m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front mooring</td>
<td>18.38</td>
<td>0.04348</td>
</tr>
<tr>
<td>Back mooring</td>
<td>16.71</td>
<td>0.04348</td>
</tr>
<tr>
<td>Left spring</td>
<td>1.2114</td>
<td>1.51886</td>
</tr>
<tr>
<td>Right spring</td>
<td>1.2114</td>
<td>1.51886</td>
</tr>
</tbody>
</table>

Table 5.8: Comparison of the drafts of the physical and the numerical models, where the increase is presented as a percentage.

<table>
<thead>
<tr>
<th>Description</th>
<th>Draft physical [m]</th>
<th>Draft numerical [m]</th>
<th>Increase [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without mooring</td>
<td>0.3898</td>
<td>0.404</td>
<td>3.64</td>
</tr>
<tr>
<td>With mooring</td>
<td>0.40</td>
<td>0.411</td>
<td>2.75</td>
</tr>
</tbody>
</table>
Figure 5.14: Heave displacement of the COG (a), normalized by the initial displacement of 0.0272 m. The acceleration (b), normalized by the maximum acceleration. Both presented as a function of time, normalized by the experimentally obtained heave period. Comparison of the results of the moored decay test performed at MARIN and the numerical solution obtained with the waveDyMFoam solver.

following computations, while all other physical properties, presented in Table 5.2, remained unchanged. Please note that neither the position of the floater nor the attachment point and anchor coordinates were altered, instead, the water level was increased by 0.011 meter. Secondly, the displacements of the free and moored floating wind turbines, displaced in Figure 5.13, show that the implementation of the mooring system lowers the natural frequency of the heave response of the system. It should also be noted that, despite of the implementation of the restraints, with all the uncertainties that come with it, the observed pitch, roll, and yaw angles experienced, were in the order of $O(10^{-2})$ degrees. These relatively small rotations give the indication that the moored floating wind turbine was balanced correctly.

With the correct draft of the numerically simulated model established, the validation against the physical model tests, performed at MARIN, could commence. First a heave decay test was performed, where the initial displacement of the floater was a modest 0.027 m. The displacement and accelerations of the COG of the floater are presented in Figure 5.14. Here, the heave displacement is normalised by the initial displacement of the floater, while the acceleration is normalised by the maximum acceleration. Both were presented as a function of time, normalized by the experimental heave period of 17.5 seconds. It can be observed that, for the first two heave periods, the numerical solution for the heave displacement provides a good fit with experimental results. The error in period and amplitude is determined after one heave period, this is presented in Table 5.9. Although the first correspondence of the first heave period is good, it is observed from the following two periods that the damping of the numerical system is higher than the damping in the physical experiment. In terms of stabil-
5.5. Motion decay of a floating structure with restraints

Figure 5.15: Pitch angle (a), normalized by the initial angle of inclination of 3.34 degrees. The angular acceleration (b), normalized by the maximum angular acceleration. Both presented as a function of time, normalized by the experimentally obtained pitch period. Comparison of the result from the moored decay test performed at MARIN and numerical solution obtained with the waveDyMFoam solver.

ity, the accelerations, presented in Figure 5.14, only show some minor spikes during the first half period of the simulation.

The numerically and experimentally obtained displacements of the moored pitch decay test are presented in Figure 5.15a, where the pitch angle, normalized by the initial pitch displacement of 3.34 degrees, is presented as a function of time, normalized by the experimental pitch period of 33.1 seconds. To isolate the pitch rotation, the numerical model was restricted to the pitch rotation, while it was free to translate in the x, y and z direction. As seen from Figure 5.15 and the calculated period and amplitude error, presented in Table 5.9, the correspondence between the numerical and experimental result was less good than previous obtained results. It may also be noted that, in contrast to the significantly larger damping observed for the heave decay test, the damping observed in the numerically computed pitch motion is much more equivalent to the experimental result. From Figure 5.15b, where the pitch acceleration is presented, it can be seen that the numerical model, minor spikes put aside, performed well for over two pitch periods. At around $t/T \approx 2.3$ the first significant in-

<table>
<thead>
<tr>
<th>Description</th>
<th>Initial displacement</th>
<th>Error period [%]</th>
<th>Error amplitude [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heave</td>
<td>0.0272 m</td>
<td>1.39</td>
<td>3.04</td>
</tr>
<tr>
<td>Pitch</td>
<td>3.34 deg</td>
<td>1.01</td>
<td>26.48</td>
</tr>
<tr>
<td>Roll</td>
<td>3.54 deg</td>
<td>3.06</td>
<td>2.76</td>
</tr>
</tbody>
</table>
indications of instability can be observed in the form of violent oscillations in the acceleration profile. Despite these irregularities, the waveDyM Foam solver managed to produce a decent result in terms of the pitch decay of a moored floating wind turbine.

Lastly, the roll decay test was performed, the numerical and experimental result of which are presented in Figure 5.16. Here, a similar phenomenon to all previous pitch and roll simulations can be observed in the acceleration profile presented in Figure 5.16b. It seems as if the under-relaxation method is capable of maintaining stability during a certain part of the computation after which small signs of instability start appearing. No evident reason was found for the instigation of the oscillations observed in the acceleration profile, other than that they are a result of irregularities in the forces acting on the body, as is discussed in Section 3.5. The roll angle, normalized with the initial inclination angle, was presented in Figure 5.16a, as a function of time, normalized by the experimentally determined roll period of 34.2 seconds. Both from the figure and from the analysis of the first response amplitude, presented in Table 5.9, it was observed that an excellent correspondence with the experimental result was achieved. However, again, a small increase in the inclination angle was observed at \( t/T = 0.5 \), the cause of which could be the restriction of the pitch and yaw rotations, to name one. This phenomenon still needs further investigation. Note that the experimental roll angle shows an irregularity at negative roll angles, so when the floater is leaning to port side. Although this abnormality does not seem to have a significant influence on the general progression of the roll signal, and could just be a measurement error, it shows how non-trivial the evaluation discussed in this section are. It may therefore be noted that the interpretation of these results

Figure 5.16: Roll angle (a), normalized by the initial angle of inclination of 3.54 degrees. The angular acceleration (b), normalized by the maximum angular acceleration. Both presented as a function of time, normalized by the experimentally obtained roll period. Comparison of the result from the moored decay test performed at MARIN and numerical solution obtained with the waveDyM Foam solver.
5.6. **Summary**

This chapter explored the capabilities of the Navier-Stokes/VOF solver in a three-dimensional set-up with respect to fluid-structure interaction. The aim of this chapter was to validate the numerical model by comparison to laboratory measurements performed at MARIN. A short introduction was given on the ‘TO2 Floating Wind’ project and the physical model campaign, after which the results from the free and moored decay tests with the OC5 floating wind turbine were discussed.

It was shown that snappyHexMesh could be used to generate a sound mesh for the complex three-dimensional sub-structure of the OC5 model. This model was used in the simulation of two free decay tests of various initial displacement. Here, a comparison was made between the solution obtained with the waveDyM Foam solver and the results presented by Dunbar et al. (2015). It was shown that the under-relaxation method, implemented in the waveDyM Foam solver, was equally capable, as the ‘tightly coupled’ interDyM foam solver from Dunbar et al. (2015), of providing a numerical solution to the intricate flow problems involved in the decay test. The damping of the heave motion, observed in the waveDyM Foam solution, was somewhat higher, which became even more apparent in the second decay test, where the initial displacement of the cylinder was higher.

Numerical simulations of the free pitch and roll decay compared against measurements from MARIN. Here, excellent results were obtained in terms of the amplitude and period of the response of the floater. The under-relaxation method was shown to provide a stable solution for the larger part of the computation, however, at some point during the computation, oscillations were observed in the angular acceleration signal. These irregularities showed similarities with the stability issues discussed in Section 3.5.

The numerical and experiment set-up of the moored decay test was discussed. A first indication of the functionality of the mooring line implementation was provided. Considering the uncertainties involved in the comparison of the numerical and experiments results of the heave, pitch and roll decay tests, an acceptable result was obtained. The under-relaxation method was shown to be capable of providing a solution to flow problems involved in these decay tests, indications of numerical instability were observed in the form of oscillations in the acceleration profile.
This chapter explores the capabilities of the Navier-Stokes/VOF solver in a three-dimensional set-up with respect to wave structure interaction. The aim of this chapter is to provide a proof-of-concept for this type of three-dimensional computations. Please note that it is not within the scope of this thesis to investigate fatigue loading, impacts on secondary structural elements or strongly nonlinear phenomenon such as ringing, let alone come up with an integrated design method.

This chapter is structured as follows; first, the capabilities of the potential flow solver, OceanWave3D, will be discussed, where special attention will be paid to the reproduction of an irregular wave field and the domain decomposition strategy, and secondly a proof-of-concept case is of irregular wave loading on a moored floating wind turbine is presented.

6.1. Wave propagation using OceanWave3D

Through various validation cases it has been shown that the Navier-Stokes/VOF model can be used for accurate simulation of waves and wave-structure interaction. The simulations however, are expensive, both in terms of storage capacity and computation power. In Section 4.3, a technique was evaluated which reduced the computational time by reducing the number of computational cells in the CFD domain. This OpenFOAM® functionality has shown to reduce the amount of computational time while preserving accuracy of the solution and was also used successfully in the three-dimensional decay simulations, as discussed in Chapter 5.

Another method for reducing the computational time is to apply a domain decomposition strategy. Here, the numerical domain is split into an inner domain where the Navier-Stokes/VOF equations are solved, for the accurate simulation of the wave-structure interaction, and an outer domain where the potential flow equations are used for the time efficient simulation of realistic sea states. Recently Paulsen et al. (2014b) presented the coupling of the waveFoam solver with OceanWave3D, a fully nonlinear three-dimensional potential flow model developed by Engsig-Karup et al. (2009). The implementation of the OceanWave3D is discussed in Subsection 2.4.2.

This section examines the effectiveness of OceanWave3D and the coupling with OpenFOAM® in a simulation where the irregular wave field from the MARIN wave calibration tests is reproduced. The characteristics of the JONSWAP spectrum are presented in Table 5.1.
Figure 6.1: Schematic representation of the domain of the potential flow solver, $\Omega$, and the waveDyMFoam solver domain, $\Gamma$. WG2 was a wave gauge positioned at location of the center column of the floater and WG1 was located 5.36 m in front of WG2. Note that $I + \frac{1}{2} \lambda = II + \frac{1}{2} \lambda = 5.36$ m.

6.1.1. Numerical Set-up

The fully nonlinear potential flow solver, OceanWave3D, can use both a wave maker signal from a piston type wave maker, or a free surface elevation signal from a wave gauge to recreate a certain time series of propagating waves. It may be noted that the solver can produce both uni- and multi-directional wave fields, as was presented in Paulsen et al. (2014b). Since no multi-directional wave propagation was considered and the floating structure did not have to be modelled, the present evaluation was performed in two-dimensional numerical domains.

As a result of policy regarding the newly installed wave maker in the MARIN concept basin, unfortunately, no wave maker signals were available. The signals from two wave gauges could be acquired, which presented the entire time series of the wave calibration tests. As mentioned above, no structure was present during these tests, which meant that one of the wave gauges, from here on referred to as wave gauge 2, was located at the position of the centroid of the center column of the floater. The other wave gauge, from here on referred to as wave gauge 1, was located 5.36 m in front of wave gauge 2. Since no wave maker signals were available, the potential flow solver tot reproduce the free surface elevations from a wave gauge signal. As the waves did not have to be generated at the wave maker, and propagated approximately 200 m further, to the location of the floater, it may be said that the potential flow solver was not utilised to its full potential in terms of efficient and accurate wave propagation.

OceanWave3D generates a complete wave spectrum from the free surface elevation signal from wave gauge 1, which is then used to reproduce the free surface elevation at the location of the floater, where the free surface elevation is then evaluated by a numerical wave gauge. Please note that neither in the physical experiments nor the numerical simulations contained a floating structure as the wave calibration tests were considered.

The potential flow solver determines a cut-off frequency based on the Nyquist frequency, so the wave spectrum is truncated for the higher frequencies. It should be noted that this method does not take into account any form of wave reflection that might have been present in the physical wave tank.

The numerical domain of the coupled OceanWave3D and waveDyMFoam solvers are depicted in Figure 6.1, here the smaller CFD domain, $\Gamma$, is located inside the larger potential flow domain, $\Omega$. The coordinates of wave gauge 2 form the center of both the numerical domains. Two wave generation and absorption zone with a length of two wavelengths were utilised in the potential flow domain. Wave gauge 1 was located at the end of the wave generation zone. It may be noted that the coupling relaxation zones in the CFD domain were relatively short, at only a quarter of a wavelength, which was an effect of the short distance
6.1. Wave propagation using OceanWave3D

Figure 6.2: Surface elevation, normalized by the significant wave height, as a function of time, normalized by the peak wave period. (a) experimental wave signals from wave gauge 1 and wave gauge 2, (b) comparison of the experimental and numerical results for the propagation of a irregular wave measured at wave gauge 2.

between the two wave gauges. The spatial discretisation of both of the numerical domains is presented in Table 6.1. Note that the resolution of the domain of the potential flow solver is significantly coarser. This is because it uses a higher order Finite Difference method for solving the Laplace equation (2.35), and a fourth order Runge-Kutta-scheme for integrating the dynamic free surface conditions, equation (2.34), which makes it far more accurate than the lower order Navier-Stokes/VOF solver.

6.1.2. Analysis of wave propagation using OceanWave3D

The free surface elevation obtained from both the physical wave calibration experiment and the OceanWave3D computation are presented in Figure 6.2. Here, the surface elevation, normalized by the significant wave height, is presented as a function of time, normalized by the peak wave period. Figure 6.2a presents a time series of around 500 waves from the physical wave calibration tests performed at MARIN. These were based on a JONSWAP spectrum with

Table 6.1: Spatial discretisation and total number of computational cells of the two-dimensional numerical domain used for the wave propagation cases using the potential flow solver coupled with the waveDyMFoam solver. Here the spatial discretisation of the waveDyMFoam solver is defined for the cells around the free surface in the base mesh.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Points per wave-length</th>
<th>Cell size, $\Delta$ [m]</th>
<th>Computational cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>OceanWave3D</td>
<td>10</td>
<td>0.64</td>
<td>540</td>
</tr>
<tr>
<td>WaveDyMFoam</td>
<td>100</td>
<td>0.064</td>
<td>7800</td>
</tr>
</tbody>
</table>
a peak enhancement value, $\gamma$, equal to 3.00 and a significant wave height and period are 0.21 m and 2.0023 s. A comparison can be made between the experimental wave signal from wave gauge 2 and the numerically computed free surface presented in ??.

On the presented time scale it is difficult to observe that in general a good correspondence between the numerical and experimental signals was established. Note that the OceanWave3D simulation starts out with a flat water condition, a ramp function was utilised to initiate the wave propagation, since the potential flow solver did not have the ability to initialise a wave field directly.

The crest elevation distribution of both the experimental and numerically evaluated, the result of which is presented in Figure 6.3. Here, the exceedance probability was presented as a function of the crest elevation, normalized by the significant wave height. The figure depicts the experimental and numerical result as well as the second order Forristall distribution (Forristall, 1978), which was generated from the characteristics of the JONSWAP spectrum. It can be observed that for approximately 90 percent of the waves, i.e. the waves up to a crest elevation of $2 \cdot \eta_{max}/H_s \approx 1.2$, all three exceedance probabilities are in good correspondence. Especially for the top one percent of the waves it can be seen that OceanWave3D generally underestimates the crest height measured during the experiment. It may be noted that the wave heights in the top one percent are significantly higher than what the Forristall distribution suggests, this may however be a result of the relatively low number of waves that was evaluated here.

Since Figure 6.3 only presents general comparison of the wave spectra, a straight signal-to-signal evaluation is presented in Figure 6.4. In this analysis a comparison is made, each time step, between the experimental free surface elevation and the numerical counterpart. In case of perfect positive correlation this approach would depict a straight line of dots, where $\eta_{OceanWave3D} = \eta_{measurement}$. As expected from Figure 6.3, the majority of the measure point, depicted in Figure 6.4, are within the area for which $1 \leq 2 \cdot \eta/H_s \geq 1$ is true. Please note that the darker color, as seen from the colorbar in the figure, indicates more measurements with an approximately equal value. The correlation between the two signals was found to be acceptable, as majority of the values is centred around the line for which $\eta_{OceanWave3D} = \eta_{measurement}$. Please note that this analysis does not correct for any differences in phase, which results in irregularities, like the circular patterns observed. This is deemed to have a negative influence on the correlation.

From the free surface elevation signal, presented in Figure 6.2b, a substantially high wave
is seen to pass the position of the structure around $t/T_p = 430$. The computational effort would be enormous if the entire time series had to be computed in the waveDyMFoam solver. Therefore, a coupling between OceanWave3D and waveDyMFoam was established. Here the time efficient potential flow solver was used to simulate the entire time series, while the, computationally more expensive, CFD computations are only active during short period around the interesting wave event. There is a one-way coupling of the flow characteristics from the OceanWave3D solver to the relaxation zones in the CFD domain, more detailed explanation of the coupling is presented in Section 2.4. The free surface elevation of wave gauge 2, from experiment, OceanWave3D and waveDyMFoam, are presented in Figure 6.5. Please note that no floating structure was present and that the simulations were performed in two-dimensional numerical domains. It can be observed that some of the higher frequency phenomena are not fully captured by the numerical models. This may partly be caused by the way OceanWave3D interprets the wave signal data, it generates a uni-directional wave signal form the experimental wave signal measured at wave gauge 1, as mentioned before, it does not take into account any waves travelling in the opposite direction. Furthermore it can be seen that the waveDyMFoam solution is slightly underestimating the wave peaks. Despite the small discrepancies an acceptable correspondence was established between the free surface
6.2. MOORED FLOATING STRUCTURE SUBJECTED TO IRREGULAR WAVES

In this section presents an exploratory evaluation of the capabilities of the waveDyMFoam solver to model the wave loading of the OC5 moored floating wind turbine. The domain decomposed solver, consisting of the potential flow solver, OceanWave3D, and the waveDyMFoam solver, is used to subject the moored floating wind turbine to uni-directional irregular waves. The aim of this section is to provide a proof-of-concept for this type of three-dimensional computations.

6.2.1. NUMERICAL SET-UP

The numerical set-up of the domain decomposed solver, used in the present section, is similar to the one described in Subsection 6.1.2. The main difference is that the inner domain, $\Gamma$, essentially the set-up of the inner domain, $\Gamma$, is equal to the set-up used for the moored decay tests, as presented in Section 5.5. The sole difference is the length of the domain and relaxation zones, these are adopted from the two-dimensional set-up described in Subsection 6.1.2. The inner numerical domain, $\Gamma$, is visualised in Figure 6.6, and the spatial discretisation and the total number of computational cells is presented in Table 6.2. Note that, with respect to the two-dimensional computations from Subsection 6.1.2, no changes were applied to the outer domain, $\Omega$. Since no multi-directional wave propagation was required, there was also no need to use a three-dimensional mesh.

The coupling between OceanWave3D and the Navier-Stokes/VOF solver was fully functional if a static mesh was considered, as shown in Subsection 6.1.2. However, when mesh motion was introduced, there was a problem with the initialisation of the two-phase flow field. A pragmatic solution was found to solve this problem: first, the initial time step, $t/T_p = 425.5$, of the waveDyMFoam was created with a zero-current flat water condition, secondly,

Table 6.2: Spatial discretisation and total number of computational cells of the two- and three-dimensional numerical domain of the potential flow solver and the waveDyMFoam solver used for the simulation of a moored floating wind turbine subjected to irregular waves. Here the spatial discretisation of the waveDyMFoam solver is defined for the cells around the free surface in the base mesh.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Points per wave-length</th>
<th>Cell size, $\Delta$ [m]</th>
<th>Computational cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>OceanWave3D</td>
<td>10</td>
<td>0.64</td>
<td>540</td>
</tr>
<tr>
<td>WaveDyMFoam</td>
<td>100</td>
<td>0.064</td>
<td>1600547</td>
</tr>
</tbody>
</table>
the OceanWave3D solver was used to compute the flow field until the initial time step of the waveDyMFoam solver and finally, utilising the end condition of the potential flow solver and the initial still water condition of the waveDyMFoam solver, the coupled simulation could be initiated from $t/T_p = 425.5$. Note that, in order for this to work, a ramp function was to be applied in the waveDyMFoam relaxation zones. This is required, so that the target solution, which is driven by the potential flow solver, would start from a still water condition.

6.2.2. Analysis Moored Floating Structure Subjected to Irregular Waves

This subsection describes the findings of the simulation of the irregular wave loading of a moored floating wind turbine in a fully nonlinear wave tank. This proof-of-concept study was performed to gain insight in the possibilities of the waveDyMFoam solver. The irregular wave loading conditions were supplied by a fully nonlinear potential flow model, OceanWave3D. The following is discussed: free surface elevation, motions of the structure and the forces in the mooring lines.

The free surface elevation from the experimental measurements, the OceanWave3D simulations and the waveDyMFoam solution are presented in Figure 6.7. Due to the presence of the moving floating structure, the interpretation of the free surface, with wave gauge 2, at the location of the center column would not work. Therefore wave gauge 2 was shifted 1 meter to port side, in the waveDyMFoam domain, so it was located in between the floating structure and the wall and maintained the same x coordinate. It should be noted that, distance from the floater to the wave gauge was half a meter, and the free surface elevation signal might be influenced by wave diffraction and radiation.

Due to the implementation of the ramp function, initially, there is little correspondence between free surface elevation in the waveDyMFoam. It can be seen that the second wave peak, around $t/T_p = 428$, is overestimated. This is probably a result of the initialisation from still water, as this was not observed in the result of the two-dimensional coupling presented in Figure 6.5. Furthermore, an underestimation of the two wave peaks, around $t/T_p = 429.8$ and $t/T_p = 430.8$, are underestimated, even more so than in the two-dimensional simulation. Then, around $t/T_p = 431$, the surface elevation signal from the waveDyMFoam solver stopped. A numerical error, or floating point error to be more precise, had caused the simulation to stop at this point. Although the waveDyMFoam solver was shown to be susceptible
Figure 6.8: Visualisation of the motions of the moored floating wind turbine subjected to irregular waves. Here, the iso-contours with $\alpha \geq 0.5$ is used to visualise the water. The position of the floater is visualised at time interval (a) $t/T = 428.5$, (b) $t/T = 429$, (c) $t/T = 429.5$, (d) $t/T = 430$, (e) $t/T = 430.5$ and (f) $t/T = 431$.

to numerical instabilities, as presented in Section 3.5 and Chapter 5, this does not directly imply that the computational error was caused by this instability.

Visualisations of various time steps of the simulations are presented in Figure 6.8. Here, the background mesh, front and back mooring lines, the floater and the free surface are depicted. The flow-induced motions of the floating structure, as well as the resulting stretching of the front mooring line can be observed from these illustrations. Furthermore, small disturbances to the free surface are observed around the surface piercing cylinders. Since all these effects are of a slightly different spatial scale, they will each be addressed separately.

The most predominant motions of the floating structure subjected to irregular uni-directional head waves are heave and pitch, since the floating system is symmetrical perpendicular to the wave direction. Therefore, the heave displacement and pitch angle were presented in Figure 6.9a and 6.9c. It was shown in Chapter 5, that signs of instability of the numerical simulation could be observed in the acceleration profiles, therefore, the acceleration in the heave direction as well as the angular acceleration were also presented, see Figure 6.9b and 6.9d. Both the heave displacement and pitch angle show a smooth profile, however, from the acceleration profiles it becomes clear that there are signs of numerical instability. These are observed in the form of oscillations in the acceleration profile, these oscillations are more intense near the peak oscillations. Furthermore, significant irregularities can be seen in both the acceleration profiles at the exact same time intervals, just a little after $t/T_p = 430$ and again around $t/T_p = 431$. The numerical solver recovers from the first irregularity, while the second one causes numerical failure. This gives an indication that, while the Navier-
Figure 6.9: Heave displacement of the floater (a), normalized by the maximum displacement. The acceleration (b), normalized by the maximum acceleration. The pitch angle (c), normalized by the maximum pitch angle. The angular acceleration (d), normalized by the maximum angular acceleration. All of these are presented as a function of time, normalized by the peak wave period.
Figure 6.10: Visualisation of the catenary mooring lines of the moored floating wind turbine subjected to irregular waves. Here, the iso-contours with $\alpha \geq 0.5$ is used to visualise the water. The position of the floater is visualised at time interval (a) $t/T = 429.5$, (b) $t/T = 430$, (c) $t/T = 430.5$ and (d) $t/T = 431$.

Stokes/VOF solver coupled with the 6-DOF motion solver was shown to provide excellent solutions of intricate fluid-structure interaction problems in Chapter 3, 4 and 5, the numerical solution is susceptible to numerical errors and the stability issue of the model is not fully resolved by the implementation of the under-relaxation method.

Figure 6.10 shows some time frames of the numerical simulation, in which the entire length of the mooring lines is visualised. It can be seen from the figure that the front mooring line is fully suspended in Figure 6.10b and 6.10d. Both of these time frames visualise the numerical simulation just after a large wave collided with the floating structure. From Figure 6.11, where the horizontal and vertical mooring forces are presented, it can be seen that these incidents correspond with large forces in the attachment points of the mooring lines. It may also be noted that these peak mooring forces coincide with the acceleration peaks observed in Figure 6.9. Please note that it is not argued here, that latter is a result of the former. What is shown here, is that the mooring line implementation can be utilised to investigate the mooring system. The accuracy with which this is done remains to be established in thorough code-to-experiment validation. Also, to further improve the pertinence of the implementation, the functionalities should be expanded. For example, the fully suspended mooring line would induce a vertical force in the anchor point, however, the current mooring line implementation does not provide the calculated anchor loads.

Finally a detailed visualisation of the fluid-structure interaction, during the wave loading of the floating wind turbine, is presented in Figure 6.12 and 6.13. Here, the free surface is visu-
Figure 6.11: The horizontal (a), and vertical (b), forces in the attachment points of both the front and the back catenary mooring line. The forces are normalized by the initial mooring force and presented as a function of time, normalized by the peak wave period.

alised using the iso-contours where $\alpha \geq 0.5$. In these figures highly detailed nonlinear fluid-structure interactions can be observed, such as wave run-up and wave over-topping of the structure. Although the accuracy of these computations remains to be validated against experimental work, they provide a first proof-of-concept that the capabilities of the waveDyMFoam solver could provide a contribution to the analysis of complex fluid-structure interactions.

6.3. **Summary**

This chapter explored the capabilities of the Navier-Stokes/VOF solver in a three-dimensional set-up with respect to wave structure interaction. The aim of this chapter was to provide a proof-of-concept for this type of three-dimensional computations. An evaluations of the capabilities of the potential flow solver, OceanWave3D, was provide. Here it was shown that a decoupled domain strategy could be used to simulate irregular waves in the waveDyMFoam solver. This coupling was found to provide a acceptable correspondence with experimental data. Some limitations were indicated with the coupling when a floating structure was modelled in the waveDyMFoam solver. An adequate solution was provided for the one-way decomposed model, which allowed for the simulation of a moored floating wind turbine subjected to irregular uni-directional waves. The analysis of this simulation showed that: a) the numerical solution was susceptible to numerical instabilities in the form of oscillations in the acceleration profile, b) the mooring line implementation provided functional restraints for the floating structure, and c) the waveDyMFoam solver is capable of providing detailed de-
Figure 6.12: Visualisation of the motions of the moored floating wind turbine subjected to irregular waves. Here, the iso-contours with $\alpha \geq 0.5$ is used to visualise the water. The position of the floater is visualised at time interval (a) $t/T = 430.5$, (b) $t/T = 430.6$ and (c) $t/T = 430.7$.

Details of the complex fluid-structure interactions involved in the wave loading of a floating structure. Please note that, although a proof-of-concept was provided, the merit of the numerical model remains to be evaluated with further validation.
Figure 6.13: Visualisation of the motions of the moored floating wind turbine subjected to irregular waves. Here, the iso-contours with $\alpha \geq 0.5$ is used to visualise the water. The position of the floater is visualised at time interval (a) $t/T = 430.8$, (b) $t/T = 430.9$ and (c) $t/T = 431.0$. 
CONCLUSIONS AND RECOMMENDATIONS

In this thesis, state-of-the-art numerical models were used to investigate the propagation of free surface waves and their interaction with fixed and moving structures in two- and three-dimensional set-ups. For the investigation, two fully nonlinear numerical models were applied. The fully nonlinear Navier-Stokes/VOF solver, provided as part of the open-source CFD-toolbox OpenFOAM®, version 2.3.1., which was extended with the implementation of the wave generation and absorption toolbox, waves2Foam, developed by Jacobsen et al. (2012). This solver, referred to as waveFoam, was used for validation cases with a static mesh. For cases where structure motion was demanded, this solver was extended with a 6-DOF motion solver, to form the waveDyMFoam solver. In an attempt to increase the spatial and temporal reach of the model a domain decomposition strategy was applied, where the fully nonlinear Navier-Stokes/VOF solver was coupled with a fully nonlinear potential flow solver, OceanWave3D, developed by Engsig-Karup et al. (2009). Please note that the fully parallel one-way domain coupling was established and fully validated by (Paulsen, 2013) and (Paulsen et al., 2014b). The numerical models applied in this research are open-source and in principle freely available as part of the waves2Foam framework. It should be noted that, the mooring line implementation, provided by Niels Jacobsen (Deltares), has not been published at present nor is the implementation publicly available.

The structure of this final chapter is as follows: first, the presented results will be discussed, secondly, the main conclusions are summarised, thirdly, a few further developments of the numerical models are suggested and finally recommendations are made for possible future research.

7.1. DISCUSSION

The results from a series of validation cases were presented throughout Chapter 3, 4 and 5. Furthermore, a proof-of-concept simulation was provided in Chapter 6. The first chapters concerned an extensive exploration of the capabilities of the Navier-Stokes/VOF solver with respect to wave propagation and wave structure interaction in a two-dimensional set-up. Chapter 4 presented the evaluation of two meshing tools provided by the OpenFOAM® toolbox, which were then utilised to accommodate the investigation of fluid-structure interaction in a series of decay tests with a three-dimensional floating wind turbine model. This research was concluded with a proof-of-concept case, discussed in Chapter 6, involving the modelling of a three-dimensional moored floating wind turbine subjected to irregular uni-
directional waves. All of these numerical computations were performed in an attempt to realise the main research objective:

"To establish a well validated fully nonlinear numerical wave tank for the simulation of complex fluid-structure interaction of moored floating offshore structures."

This section provides a discussion of the results presented in Chapter 3, 4, 5 and 6, in order to establish the success and limitations of the presented work related to the fulfilment of this main objective.

7.1.1. A CONVERGING SOLUTION

If this Navier-Stokes-VOF solver is to become a supplement or even a substitute to physical model testing, one has to be able to count on the solution. Therefore, the solution of the model should improve as the spatial or temporal resolution increases. A grid convergence study was successfully performed on the propagation of a fully nonlinear stream function wave, as described in Section 3.2. The convergence rate was verified for both a finer and a coarser set of spatial discretisations. An approximately first order convergence rate, corresponding with the first order Upwind and Euler schemes used for the spatial and temporal discretisation, was observed for the first couple of wave periods. However, the convergence rate was shown to decrease over time and was lower for the coarser set of spatial discretisations. In order to increase the computational efficiency an attempt was made to use a second order spatial scheme, while the results from these simulations showed an improved solution at some time intervals, no reliable convergence rate could be established. As the present research would benefit more from a stable result, the choice was made to use the first order schemes for all the numerical simulations. The various grid sensitivity analysis performed on all individual validation cases each showed a reliable result, justifying the choice. It should be noted that increasing computational efficiency is an important issue for these computationally expensive models and that it would be beneficial to perform more detailed studies on higher order discretisation schemes.

7.1.2. PROPAGATION OF A FULLY NONLINEAR WAVE

To be able to investigate the impact of nonlinear waves on floating structures one has to provide an accurate description of propagating nonlinear waves in the first place. In Section 3.2, numerical computations with the two-phase Navier-Stokes/VOF solver were used to simulate the propagation of the single-phase fully nonlinear stream function wave (Fenton, 1988). Here, the diffusive nature of these types of models, based on the VOF method, became clear. Although an excellent resolution was obtained for the initialisation of the flow field, the diffusivity was shown to increase rapidly over the first wave period, during which the air-water interface was smeared. The description of the free surface by means of a VOF method therefore remains an area which is in need of improvement.

In spite of the smearing of the interface during this first wave period, the description of the fully nonlinear stream function wave was reasonably good in terms of diffusivity, considering it was a numerical two-phase reproduction of a single-phase flow. An acceptable error in the order of $O(10^{-2})$ was observed for spatial discretisations of 150 and 200 points per wavelength. The observations made regarding the phase error of the propagating wave was reason for concern. The time step, determined by the maximum Courant number, seemed to be governing the phase error of the wave. A high Courant number, resulting in a larger time
7.1. **Discussion**

Step, lead to a negative phase lag, indicating the unexpected phenomenon that the wave was moving faster as it got smaller. This is a reason for concern and it would be wise to perform a more elaborated sensitivity analysis on this subject matter. Considering the time attributed to this thesis, the remainder of the research questions to be answered, and primarily due to the fact that an acceptable result was obtained for the intended propagation length of one to two wavelengths, a more elaborated sensitivity analysis was considered unjustified within the scope of this thesis.

7.1.3. **Wave loads on a structure**

The computation of wave loads on a structure was validated on the basis of theoretical and experimental work by Dixon et al. (1979). The numerical computations lead to accurate results in terms of peak loading. The Root Mean Square vertical force provided less promising results, the importance of this force remains questionable at this point, because the numerically computed loading cycle has a considerably better correspondence with the experimental than the theoretical values. It should be noted that, as a simplification, the viscous boundary layers on all solid walls were neglected. Despite this simplification the computed wave loads on the fixed structure were considered to be highly accurate. However, with respect of the main objective of the thesis, it is clear that thorough validation of wave loading on floating structures is still required. Regretfully, this did not fit within the scope of this research in terms of time and resources.

The heave decay of a horizontal cylinder was simulated in Section 3.5. The presented absolute forcing on the cylinder surface showed irregularities, while stable solutions were provided for the motion and velocity of the cylinder. The observed force spikes may have been a result of the interpolation of the pressures, which makes it an interpretation problem, or the force spikes could be a result of errors in the pressure field, making it a numerical problem. Either way, the disturbances were nowhere near the extreme velocity and force oscillations observed in Dunbar et al. (2015), where the instability of the standard OpenFOAM® interDyMFoam solver was shown. It should be noted that this research probably used a different version of OpenFOAM® and did not implement the waves2Foam toolbox. Since no instability problems were observed for the various cases, it was concluded that no further attention would be paid to the force spikes during this thesis.

7.1.4. **Motions of a structure**

For the simulation of the motions of a floating structure, the model relies on the 6-DOF motion solver and dynamic mesh handling of the standard OpenFOAM® interDyMFoam solver. This model was extended with the waves2Foam toolbox in order to implement the generation and absorption of waves. In Section 3.4, this model was utilised to describe the generation of waves by the forced heave oscillation of a horizontal cylinder. The results of these two-dimensional numerical simulations proved to be very sensitive to changes in the domain set-up. Although eventually an acceptable correspondence with the experimental result from Yu and Ursell (1961) was achieved, initially, reflections from both of the outer boundaries, in the order of $O(10^1)$, were disturbing the flow pattern. This meant that the relaxation zones, which were shown to be perfectly capable of handling wave generation and absorption by Jacobsen et al. (2012), were not working properly for this particular dynamic mesh set-up. This phenomenon was not observed to effect the results of rest of the dynamic mesh cases involving free moving bodies. No further time was spend on finding the cause of this problem, as it was
solely established for this forced motion case and the main goal, to prove capability of the model to accurately model the fluid structure interaction, was accomplished.

The simulation of the free decay of a horizontal cylinder was presented in Section 3.5. Here, acceptable correspondence with the theoretical work by Maskell and Ursell (1970) and experimental work by Ito (1977) was achieved. It should be noted that some irregularities, in the form of singular force spikes, were observed in the acceleration solution. These are a direct result of the spikes observed in the force signal, which are calculated from the pressure field. These minor signs of instability lead to the evaluation of the under-relaxation method implemented in waveDyMFoam. Here it was shown that an ‘accelerationRelaxation’ factor, which was provided to relax from the forcing determined accelerations, successfully improved the stability of the numerical simulation.

After the successful completion of these two-dimensional simulations, the waveDyMFoam solver was used for the decay tests of a three-dimensional floating wind turbine support structure. The free and restrained decay cases were presented in Section 5.4 and 5.5. Especially the free decay tests provided good similarity with the measurements from the physical model tests in terms of amplitude and period. Similar accuracy was obtained compared to the results of the two-dimensional free decay test with the horizontal cylinder.

The presented results of numerical computations of the restrained decay tests showed minor differences with the measurements from the physical experiments. The results of the heave decay showed that the numerical model had a little more damping than the physical model. This could be due to the differences between the numerical representation of the mooring system and the physical one, due to the estimation that was made for the pretension in the linear springs, the horizontal restraints of the system, or because the numerical model actually computed more damping in the heave direction. Since the actual pretension in the horizontal springs is unknown and no free heave decay tests were done with the model it is difficult to conclude on this matter.

A comparison against numerical computations of two free heave decay tests, presented by Dunbar et al. (2015), did show that damping in the waveDyMFoam solver was significantly higher than in their ‘tightly coupled’ interDyMFoam solver. This may indicate that the under-relaxation method, applied in the present research, introduces extra damping, although it should be pointed out that there are many uncertainties in this code-to-code comparison.

The differences observed between the numerical and physical results for the restrained pitch and roll simulations give an indication that there could be something wrong with the definition of the numerical mooring system. As mentioned, there were some uncertainties regarding the set-up of the physical mooring system, this makes it difficult to distinguish whether the errors are caused by the uncertainties in the set-up of the moored floating wind turbine or that the numerical computation of the governing physics is slightly inaccurate. Furthermore, it should be noted that indications of numerical instability were observed in the form of oscillations in the acceleration profile. These indications were supported by the observations made in the proof-of-concept simulation.

7.1.5. Advanced meshing tools
In Chapter 4, the capabilities of two advanced meshing tools, provided with the standard distribution of OpenFOAM®, were investigated. It was report that the versatile three-dimensional meshing tool, snappyHexMesh, was an effective tool for sculpting a structure from a standard block mesh. The method was used to recreate the two-dimensional numerical sim-
ulations from Chapter 3 and proved capable of reproducing an improved level of accuracy in the solution. The way the mesh generation was set-up lead to smaller computational cells in the vicinity of the structure. This made for an unfair comparison, because on the one hand the accuracy of the solution increased, while on the other hand the computational time increased. More importantly however, the simulations did not show numerical instabilities due to the implementation of a snappyHexMesh domain and it would therefore be a proper starting point for the meshing of more complex three-dimensional structures.

In order to lower the computational effort of the model a multi-grading tool, from a newer version of OpenFOAM®, was utilised to generate mesh with a high cell concentration around the free surface. Provided the cells in the vicinity of the free surface in the graded meshes had the same aspect ratio as the uniform reference meshes, a very high similarity was achieved in the accuracy of the result. The result in terms of time saving did regretfully not prove to be as consistent. Although a decrease was observed in all cases, just in one of the cases, this was proportional to the amount of decrease in computational cells.

Both methods were used successfully in Chapter 5, where they were used for the generation of the numerical reproduction of the physical model experiments of the floating wind turbine. SnappyHexMesh provided an excellent representation of the substructure of the floating wind turbine with small computational cells in the vicinity of the structure, while the overall amount of computational cells was reduced by concentrating the mesh around the free surface.

7.1.6. FULLY NONLINEAR NUMERICAL WAVE TANK

Chapter 6 explored the capabilities of the Navier-Stokes/VOF solver in a three-dimensional set-up with respect to wave structure interaction. Here, a proof-of-concept simulation was evaluated on the irregular wave loading of a moored floating wind turbine. A decoupled domain strategy was shown to be capable of providing irregular wave boundary conditions to the Navier-Stokes/VOF solver. The potential flow solver, OceanWave3D, was seen to provide efficient and accurate solutions for the propagation of irregular uni-directional waves. It should be noted that there were some limitations with the coupling of OceanWave3D and the waveDyMFoam solver when a moving floating structure has to be modelled, however, an acceptable solution was provided.

The evaluation of the irregular wave loading of the moored floating wind turbine confirmed that, although the under-relaxation method implemented in the waveDyMFoam solver was shown to provide adequate solutions to intricate flow problems, the numerical solution was susceptible to numerical instabilities in the form of oscillations in the acceleration profile. It should be noted that in the extreme case these irregularities may lead to break-down of the numerical computation. In order to use this numerical wave tank as a supplement or even substitute to physical wave tanks, improvement of the stability of this numerical model is therefore still a top priority. The mooring line implementation was proven to be functional, although the capabilities of the implementation could be expanded on some aspects and a full validation of the system is still to be provided. The proof-of-concept simulation also provided detailed descriptions of the complex fluid-structure interactions, however, full validation of nonlinear phenomenon such as local impact loads and wave run-up should be performed to confirm the accuracy of the model.
7.2. **Conclusions**

In regard to the main objective and the research questions, proposed in Section 1.4, it can be concluded that:

- A thoroughly validated fully nonlinear solver was presented for the accurate computation of wave-structure interaction.
- The implementation of a multi-gradient grid can improve computational efficiency without significant loss of numerical accuracy.
- The waveFoam solver is capable of providing an accurate description of the wave loading on a fixed structure.
- The waveDyMFOam solver can be utilised for the accurate computation of free decay tests of both two- and three-dimensional floating structures.
- The waveDyMFOam solver is of providing acceptable solutions for the moored decay tests of a three-dimensional floating structure.
- A functioning restraint system was implemented, which can be used to simulate the mooring system of a floating structure.
- The domain decomposition strategy can be used to efficiently simulate realistic sea states in the computational domain of the waveDyMFOam solver.
- The waveDyMFOam solver can be used for the simulation of a moored floating structure in realistic sea states.
- The numerical stability provided by the under-relaxation method has its limitations and further improvement of the stability of the waveDyMFOam solver is required.

7.3. **Further developments**

As part of this thesis, a fully nonlinear Navier-Stokes/VOF type numerical model was validated and verified with respect to wave propagation and wave-structure interaction. Even though the model has been shown capable of computing highly detailed fluid-structure interactions, including dynamic motion response of a ridged structure, further development is still necessary. Possible extensions could include:

- **Improved stability waveDyMFOam solver**: Numerous simulations were performed in this research that showed the capabilities of the waveDyMFOam solver, some of the simulations were seen to be susceptible to numerical instability. The under-relaxation method which was used here to increase stability was observed to have limitations, as fluid-structure interaction became more complex and flows more violent. Efforts should be made in further developments of the numerical models discussed in the present thesis to increase the numerical stability. Dunbar et al. (2015) provided a different approach to increase the numerical stability of a similar numerical model, a thorough code-to-code verification might provide new insights in the matter.
- **Improved performance Navier-Stokes/VOF solver**: A first order convergence rate of the Navier-Stokes/VOF solver, consistent with the first order spatial and temporal discretisation, was provided in this research. During a brief investigation of the effects of using a second order spatial scheme no stable solution was found. As the applied CFD model supports various spatial discretisation methods, a thorough validation could be beneficial to the computational performance.
• **Improved turbulence modelling:** In all the simulations performed with the Navier-Stokes/VOF solver, the viscous boundaries were neglected by applying a slip conditions to solid wall boundaries. As the in most cases the computations were in good correspondence with their experimental counterparts, it was concluded that this approach was an effective method to simplify the flow problem. A logical next step is to include turbulence models to improve the wall shear description.

• **Improved mooring line description:** The mooring line implementation, provided by Niels Jacobsen (Deltares), was based on a simple catenary type mooring line, which was not influenced by waves or current. Firstly, note that this model is still to be made publicly available. Secondly, this incomplete model, could result in underestimation of the mooring forces. An extensive validation research, as suggested in the next section, for future work, can help identify the need for an improved mooring line model.

• **Coupling with other models:** The present research focussed on applying the CFD model for the investigation of wave loading on floating structures. When it comes to moored floating structures, the evaluation of the anchor strengths is an important topic. For such an analysis it would be interesting to provide real time information of the loads in the anchor points to a geotechnical model such as PLAXIS. As the Navier-Stokes/VOF solver is already equipped to provide these loads, only the coupling between these models has to be established. One can think of numerous other fields of application, with respect to offshore wind and more generally hydrodynamical engineering, that could benefit from highly detailed CFD computations. An example would be the installation of a gravity-based foundation of an offshore wind turbine on the seabed, again, the coupling with a detailed geotechnical model could provide detailed information about soil pressures and deformations through the simulation of the touchdown moment in the installation process.

### 7.4 Future research

The presented validation of the Navier-Stokes/VOF solver, with respect to wave propagation, is considered extensive. However, the focus of the present work was on the validation of a fully nonlinear numerical wave tank and still many research topics have been left untouched. The two numerical models have, even without the suggested improvements, the potential to significantly improve the computations of fluid-structure interaction in numerous applications. For future work the following research topics could be of interest:

• **Wave loads on a moored floating structure:** Regretfully, the proof-of-concept computation, of the loading on a moored floating wind turbine by irregular waves, could not be validated with experiment measurements, due limitation of time and availability of data. This being said, the presented model can be used for the numerous subjects that need substantial validation:
  • Motions of the floating structure as a result of wave loading.
  • Impact loads on the floating structure from of irregular phase-focussed waves.
  • Loads in the mooring system as a result of wave impacts.
  • Wave loads on secondary structures from irregular waves:

• **Multi-directional phase-focused waves:** Multi-directional phase-focused waves show the potential to introduce larger peak forces than unidirectional waves. Since the work
of (Paulsen, 2013) and (Paulsen et al., 2014b) has demonstrated the excellent capabilities of the potential flow solver to generate these types of free surface waves, it would be a logical next step to investigate the effects of such waves on floating structures. Please note that many of the research topics related to wave loads on moored floating structures, as mentioned in the previous bullet point, are interesting fields of study when it comes to multi-directional phase-focused waves.

- **Related offshore research topics:** It can be argued that wave loading on floating wind turbines is just one of the many interesting research topics regarding the search for a cleaner offshore energy source. At the moment, many attempts are made with respect to the advancements of these technologies and one could imagine using the, in this thesis discussed, Navier-Stokes/VOF solver for the investigation of extreme wave loads on wave energy converters, detailed modelling of tidal energy conversion systems subjected to waves, detailed towing operations and installation of gravity-based foundations for offshore wind turbines to name a few.
#include "fvCFD.H"

/* **************************************************************************
   OpenFOAM: The Open Source CFD Toolbox
   Copyright (C) 2011-2013 OpenFOAM Foundation
   
   This file is part of OpenFOAM.
   
   OpenFOAM is free software: you can redistribute it and/or modify it under
   the terms of the GNU General Public License as published by the Free
   Software Foundation, either version 3 of the License, or (at your option)
   any later version.
   
   OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
   ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
   FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
   for more details.
   
   You should have received a copy of the GNU General Public License along
   with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

   Application
   interFoam

   Description
   Solver for 2 incompressible, isothermal immiscible fluids using a VOF
   (volume of fluid) phase-fraction based interface capturing approach.

   The momentum and other fluid properties are of the "mixture" and a single
   momentum equation is solved.

   Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.

   For a two-fluid approach see twoPhaseEulerFoam.
*/

#include "fvCFD.H"
```c++
#include "CMULES.H"
#include "subCycle.H"
#include "interfaceProperties.H"
#include "incompressibleTwoPhaseMixture.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"
#include "fixedFluxPressureFvPatchScalarField.H"
#include "relaxationZone.H"
#include "externalWaveForcing.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main( int argc, char * argv[]) {
    
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    pimpleControl pimple(mesh);
    
    #include "initContinuityErrs.H"
    #include "readGravitationalAcceleration.H"
    #include "readWaveProperties.H"
    #include "createExternalWaveForcing.H"
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "createPrghCorrTypes.H"
    #include "correctPhi.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    Info<< "\nStarting time loop\n" << endl;
    while ( runTime.run() )
    {
        #include "readTimeControls.H"
        #include "CourantNo.H"
        #include "alphaCourantNo.H"
        #include "setDeltaT.H"

        runTime++;

        Info<< "Time = " << runTime.timeName() << nl << endl;

        externalWave->step();

        // --- Pressure-velocity PIMPLE corrector loop
        while ( pimple.loop() )
        {
            #include "alphaControls.H"

            if ( pimple.firstIter() || alphaOuterCorrectors )
            {
```
twoPhaseProperties.correct();

#include "alphaEqnSubCycle.H"
relaxing.correct();
interface.correct();
}

#include "UEqn.H"

// --- Pressure corrector loop
while (pimple.correct())
{
  #include "pEqn.H"

  if (pimple.turbCorr())
  {
    turbulence->correct();
  }
}

runTime.write();

Info << "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
<< " ClockTime = " << runTime.elapsedClockTime() << " s"
<< nl << endl;

// Close down the external wave forcing in a nice manner
externalWave->close();

Info << "End\n" << endl;

return 0;

}
/*--------------------------------*-\*-
 \ \ / Field \ OpenFOAM: The Open Source CFD Toolbox
 \ \ / O peration \ Copyright (C) 2011-2014 OpenFOAM Foundation
 \ \ / A nd \ License
 \ \|/ M anipulation
--------------------------------------------*/

License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Application
interDyM Foam

Description
Solver for 2 incompressible, isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based interface capturing approach, with optional mesh motion and mesh topology changes including adaptive re-meshing.

\*---------------------------------------------*/

#include "fvCFD.H"
#include "dynamicFvMesh.H"
#include "CMULES.H"
#include "subCycle.H"
#include "immiscibleIncompressibleTwoPhaseMixture.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"
#include "fixedFluxPressureFvPatchScalarField.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[]) {
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createDynamicFvMesh.H"
    #include "initContinuityErrs.H"

    pimpleControl pimple(mesh);
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "createPrghCorrTypes.H"

    volScalarField rAU
    {
        IOobject
        {
            "rAU",
            runTime.timeName(),
            mesh,
            IOobject::READ_IF_PRESENT,
            IOobject::AUTO_WRITE
        },
        mesh,
        dimensionedScalar("rAUf", dimTime/rho.dimensions(), 1.0)
    };

    #include "correctPhi.H"
    #include "createUf.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
    Info<< "
Starting time loop
" << endl;

    while (runTime.run()) {
        #include "readControls.H"
        #include "alphaCourantNo.H"
        #include "CourantNo.H"

        #include "setDeltaT.H"

        runTime++;;

        Info<< "Time = " << runTime.timeName() << nl << endl;

        // --- Pressure-velocity PIMPLE corrector loop
        while (pimple.loop()) {
            if (pimple.firstIter() || moveMeshOuterCorrectors)
scalar timeBeforeMeshUpdate = runTime.elapsedCpuTime();

mesh.update();

if (mesh.changing())
{
    Info << "Execution time for mesh.update() = " << runTime.elapsedCpuTime() - timeBeforeMeshUpdate << " s" << endl;

    gh = g & mesh.C();
    ghf = g & mesh.Cf();
}

if (mesh.changing() && correctPhi)
{
    // Calculate absolute flux from the mapped surface velocity
    phi = mesh.Sf() & Uf;

    #include "correctPhi.H"

    // Make the flux relative to the mesh motion
    fvc::makeRelative(phi, U);

    mixture.correct();
}

if (mesh.changing() && checkMeshCourantNo)
{
    #include "meshCourantNo.H"
}

#include "alphaControls.H"
#include "alphaEqnSubCycle.H"

mixture.correct();

#include "UEqn.H"

// ---- Pressure corrector loop
while (pimple.correct())
{
    #include "pEqn.H"
}

if (pimple.turbCorr())
{
    turbulence->correct();
}

runTime.write();

Info << "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
<< " ClockTime = " << runTime.elapsedClockTime() << " s"
<< nl << endl;

Info << "End\n" << endl;

return 0;

// ..............................................................
//
SIXDOFRIGIDBODYMOTION CODE

// * * * * * * * * * * * * * P r i v a t e M e m b e r F u n c t i o n s  * * * * * * * * * * * * * //

void Foam::sixDoFRigidBodyMotion::applyRestraints()
{
    if (restraints_.empty())
    {
        return;
    }
    if (Pstream::master())
    {
        forAll(restraints_, rI)
        {
            if (report_)
            {
                Info<< "Restraint " << restraints_[rI].name() << ": ";
            }
        }
    }
}
// Restraint position
point rP = vector::zero;

// Restraint force
vector rF = vector::zero;

// Restraint moment
vector rM = vector::zero;

// Accumulate the restraints
restraints_[rI].restrain(*this, rP, rF, rM);

// Update the acceleration
a() += rF/mass_;

// Moments are returned in global axes, transforming to
// body local to add to torque.
tau() += Q().T() & (rM + ((rP - centreOfRotation()) ^ rF));

// * * * * * * * * * * * * * * * * Constructors * * * * * * * * * * * * * * //

Foam::sixDoFRigidBodyMotion::sixDoFRigidBodyMotion()
:
  motionState_(),
  motionState0_(),
  restraints_()
  constraints_(),
  tConstraints_(tensor::I),
  rConstraints_(tensor::I),
  initialCentreOfMass_(vector::zero),
  initialCentreOfRotation_(vector::zero),
  initQ_(1),
  mass_(VSMALL),
  momentOfInertia_(diagTensor::one-VSMALL),
  aRelax_(1.0),
  aDamp_(1.0),
  report_(false)
{}

Foam::sixDoFRigidBodyMotion::sixDoFRigidBodyMotion
( const dictionary& dict,
  const dictionary& stateDict )
:
  motionState_(stateDict),
  motionState0_(),
  restraints_()
  constraints_(),
  tConstraints_(tensor::I),
  ...
rConstraints_(tensor::I),
initialCentreOfMass_
{
    dict.lookupOrDefault
    {"initialCentreOfMass",
     vector(dict.lookup("centreOfMass"))
    },
initialCentreOfRotation_(initialCentreOfMass_),
initialQ_
{
    dict.lookupOrDefault
    {"initialOrientation",
     dict.lookupOrDefault("orientation", tensor::I)
    },
}
mass_(readScalar(dict.lookup("mass"))),
momentOfInertia_(dict.lookup("momentOfInertia")),
aRelax_(dict.lookupOrDefault<scalar>("accelerationRelaxation", 1.0)),
aDamp_(dict.lookupOrDefault<scalar>("accelerationDamping", 1.0)),
report_(dict.lookupOrDefault<Switch>("report", false))
{
    addRestraints(dict);
    // Set constraints and initial centre of rotation
    // if different to the centre of mass
    addConstraints(dict);
    // If the centres of mass and rotation are different ... vector R(initialCentreOfMass_ - initialCentreOfRotation_);
    if (magSqr(R) > VSMALL)
    {
        // ... correct the moment of inertia tensor using parallel axes theorem
        momentOfInertia_ += mass_*diag(1*magSqr(R) - sqr(R));
        // ... and if the centre of rotation is not specified for motion state
        // update it
        if (!stateDict.found("centreOfRotation"))
        {
            motionState_.centreOfRotation() = initialCentreOfRotation_;
        }
    }
    // Save the old-time motion state
    motionState0_ = motionState_;
}

Foam::sixDoFRigidBodyMotion::sixDoFRigidBodyMotion
(const sixDoFRigidBodyMotion& sDoFRBM)
:
    motionState_(sDoFRBM.motionState_),
    motionState0_(sDoFRBM.motionState0_),
    }
```cpp
exDoFRigidBodyMotion::addRestraints
{
    const dictionary& dict
    {
        if (dict.found("restraints"))
        {
            const dictionary& restraintDict = dict.subDict("restraints");
            label i = 0;
            restraints_.setSize(restraintDict.size());
            forAllConstIter(IDLList<entry>, restraintDict, iter)
            {
                if (iter().isDict())
                {
                    restraints_.set
                    {
                        i++,
                        exDoFRigidBodyMotionRestraint::New
                        (iter().keyword(),
                        iter().dict());
                    }
                }
                restraints_.setSize(i);
            }
        }
    }
}

exDoFRigidBodyMotion::addConstraints
(
    const dictionary& dict
    {
        if (dict.found("constraints"))
        {
        }
    }
```
const dictionary& constraintDict = dict.subDict("constraints");

label i = 0;

constraints_.setSize(constraintDict.size());

pointConstraint pct;
pointConstraint pcr;

forAllConstIter(IDLList<entry>, constraintDict, iter)
{
    if (iter().isDict())
    {
        constraints_.set
        (i,
         sixDoFRigidBodyMotionConstraint::New
         (iter().keyword(),
          iter().dict(),
          *this)
         );

        constraints_[i].setCentreOfRotation(initialCentreOfRotation_);
        constraints_[i].constrainTranslation(pct);
        constraints_[i].constrainRotation(pcr);

        i++;
    }
}

constraints_.setSize(i);

tConstraints_ = pct.constraintTransformation();
rConstraints_ = pcr.constraintTransformation();

Info << "Translational constraint tensor " << tConstraints_ << nl
    << "Rotational constraint tensor " << rConstraints_ << endl;
}

void Foam::sixDoFRigidBodyMotion::updatePosition
(
    scalar deltaT,
    scalar deltaT0
)
{
    // First leapfrog velocity adjust and motion part, required before
    // force calculation
    if (Pstream::master())
    {
        v() = tConstraints_ & (v0() + aDamp_*0.5*deltaT0*a());
        pi() = rConstraints_ & (pi0() + aDamp_*0.5*deltaT0*tau());
    }
// Leapfrog move part
centreOfRotation() = centreOfRotation0() + deltaT*v();

// Leapfrog orientation adjustment
Tuple2<tensor, vector> Qpi = rotate(Q0(), pi(), deltaT);
Q() = Qpi.first();
pi() = rConstraints_ & Qpi.second();
}
Pstream::scatter(motionState_);

void Foam::sixDoFRigidBodyMotion::updateAcceleration
(const vector& fGlobal,
const vector& tauGlobal,
scalar deltaT)
{
static bool first = false;

// Second leapfrog velocity adjust part, required after motion and
// acceleration calculation
if (Pstream::master())
{
// Save the previous iteration accelerations for relaxation
vector aPrevIter = a();
vector tauPrevIter = tau();

// Calculate new accelerations
a() = fGlobal/mass_;
tau() = (Q().T() & tauGlobal);
applyRestraints();

// Relax accelerations on all but first iteration
if (!first)
{
a() = aRelax_*a() + (1 - aRelax_)*aPrevIter;
tau() = aRelax_*tau() + (1 - aRelax_)*tauPrevIter;
}
first = false;

// Correct velocities
v() += tConstraints_ & aDamp_*0.5*deltaT*a();
pi() += rConstraints_ & aDamp_*0.5*deltaT*tau();

if (report_)
{
status();
}
Pstream::scatter(motionState_);
void Foam::sixDoFRigidBodyMotion::status() const
{
    Info << "6-DoF rigid body motion" << nl
    << " Centre of rotation: " << centreOfRotation() << nl
    << " Centre of mass: " << centreOfMass() << nl
    << " Orientation: " << orientation() << nl
    << " Linear velocity: " << v() << nl
    << " Angular velocity: " << omega() << endl;
}

Foam::tmp<Foam::pointField> Foam::sixDoFRigidBodyMotion::transform
(const pointField& initialPoints) const
{
    return 
    (centreOfRotation()
     + (Q() & initialQ_.T() & (initialPoints - initialCentreOfRotation_))
    );
}

Foam::tmp<Foam::pointField> Foam::sixDoFRigidBodyMotion::transform
(const pointField& initialPoints,
 const scalarField& scale) const
{
    // Calculate the transformation septernion from the initial state
    septernion s
    (centreOfRotation() - initialCentreOfRotation(),
     quaternion(Q() & initialQ_.T())
    );

tmp<pointField> tpoints(new pointField(initialPoints));
pointField& points = tpoints;

forAll(points, pointi)
{
    // Move non-stationary points
    if (scale[pointi] > SMALL)
    {
        // Use solid-body motion where scale = 1
        if (scale[pointi] > 1 - SMALL)
        {
            points[pointi] = transform(initialPoints[pointi]);
        }
        // Slerp septernion interpolation
        else
        {
            septernion ss(slerp(septernion::I, s, scale[pointi]));
        
    } // else, ignore pointi
} // forAll
points[pointi] =
  initialCentreOfRotation() 
+ ss.transform
  (initialPoints[pointi] - initialCentreOfRotation());

return tpoints;

// ************************************************************************** //
Figure D.1: Two-dimensional representation of the floating wind turbine model.
Figure D.2: Two-dimensional representation of the sub-structure of the floating wind turbine model, where the Froude scale was 1:50. (a) left side view, (b) back view. The dimensions are: 1 = 0.06m, 2 = 0.52m, 3 = 0.817m, 4 = 0.577m, 5 = 0.50m, 6 = 0.817m, 7 = 0.48m, 8 = 0.12m, 9 = 1.48m and 10 = 1.00m.

Table D.1: Coordinates of COG and restraining system of the floating wind turbine model depicted in Figure D.1 and Figure D.2, where the Froude scale was 1:50.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Description</th>
<th>Attachment [m]</th>
<th>Anchor [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>COG</td>
<td>(0.0, 0.0, 0.1186)</td>
<td>-</td>
</tr>
<tr>
<td>II</td>
<td>front mooring</td>
<td>(0.818, 0.0, 0.0)</td>
<td>(18.434, 0.0, 0.0)</td>
</tr>
<tr>
<td>III</td>
<td>back mooring</td>
<td>(-0.818, 0.0, 0.0)</td>
<td>(-16.762, 0.0, 0.0)</td>
</tr>
<tr>
<td>IV</td>
<td>left spring</td>
<td>(-0.408, -0.620, 0.48)</td>
<td>(-0.408, -2.0, 0.48)</td>
</tr>
<tr>
<td>V</td>
<td>right spring</td>
<td>(-0.408, 0.620, 0.48)</td>
<td>(-0.408, 2.0, 0.48)</td>
</tr>
</tbody>
</table>
**Table D.2:** Dimensions of the sub-structure of the floating wind turbine model depicted in Figure D.2, where the Froude scale was 1:50.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Description</th>
<th>Number of pieces</th>
<th>Diameter [m]</th>
<th>length [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>transition</td>
<td>1</td>
<td>0.14</td>
<td>0.12</td>
</tr>
<tr>
<td>B</td>
<td>center column</td>
<td>1</td>
<td>0.13</td>
<td>0.52</td>
</tr>
<tr>
<td>C</td>
<td>offset columns</td>
<td>3</td>
<td>0.24</td>
<td>0.52</td>
</tr>
<tr>
<td>D</td>
<td>heave plates</td>
<td>3</td>
<td>0.48</td>
<td>0.12</td>
</tr>
<tr>
<td>E</td>
<td>braces</td>
<td>9</td>
<td>0.032</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table D.3:** Properties of the restraining system of the floating wind turbine model depicted in Figure D.1 and Figure D.2. $W_{sub}$ is the submerged weight of the mooring line and $k_{spring}$ is the spring stiffness, where the Froude scale was 1:50.

<table>
<thead>
<tr>
<th>Description</th>
<th>Length (at rest) [m]</th>
<th>$M_{sub} / k_{spring}$ [N/m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front mooring</td>
<td>18.38</td>
<td>0.04348</td>
</tr>
<tr>
<td>Back mooring</td>
<td>16.71</td>
<td>0.04348</td>
</tr>
<tr>
<td>Left spring</td>
<td>1.2114</td>
<td>1.51886</td>
</tr>
<tr>
<td>Right spring</td>
<td>1.2114</td>
<td>1.51886</td>
</tr>
</tbody>
</table>

**Table D.4:** Properties of the floating wind turbine model depicted in Figure D.1 and Figure D.2. These are properties including the floater, turbine and tower.

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Unit</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft (mooring)</td>
<td>$T$</td>
<td>[m]</td>
<td>0.40</td>
</tr>
<tr>
<td>Draft (no mooring)</td>
<td>$T$</td>
<td>[m]</td>
<td>0.3898</td>
</tr>
<tr>
<td>Mass</td>
<td>$M$</td>
<td>[kg]</td>
<td>111.664</td>
</tr>
<tr>
<td>Roll inertia</td>
<td>$I_{xx}$</td>
<td>[kg/m$^2$]</td>
<td>49.768</td>
</tr>
<tr>
<td>Pitch inertia</td>
<td>$I_{yy}$</td>
<td>[kg/m$^2$]</td>
<td>47.556</td>
</tr>
<tr>
<td>Yaw inertia</td>
<td>$I_{zz}$</td>
<td>[kg/m$^2$]</td>
<td>43.814</td>
</tr>
</tbody>
</table>
Figure D.3: Detailed two-dimensional representation of the computational mesh of the sub-structure of the floating wind turbine model.


