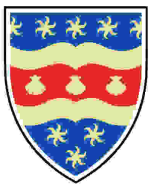
2019

NUMERICAL MODELLING OF RESPONSES OF OFFSHORE WAVE ENERGY CONVERTERS IN EXTREME WAVES

Musiedlak, Pierre-Henri



UNIVERSITY OF
PLYMOUTH

NUMERICAL MODELLING OF RESPONSES OF OFFSHORE WAVE ENERGY CONVERTERS IN EXTREME WAVES

by

MUSIEDLAK PIERRE-HENRI

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Engineering
In collaboration with DNV-GL

March 2019

Acknowledgements

Thanks to every person from the University of Plymouth CFD team, who have been answering kindly my questions. Thanks to Thomas Vyzikas, who has been a very good help with the starting of the PhD, providing simple documents and answers even for minor difficulties with OpenFOAM and CFD (and by mistake, introduce me to brewing and the lovely Kurdish restaurant). Edward Ransley, for his very big everyday contributions (and vegetables), such as papers correction and instantaneous OpenFOAM answers. Finally, Scott Brown, who was very helpful in coding, and included me in ISOPE papers even with my small contribution.

I would like as well to thank the rest of my supervision team. Thanks to Deborah Greaves for the time that she found for meeting within her busy planning. Martyn Hann, for his detailed answers about anything linked to experiments and data acquisition, and for giving me the opportunity to undertake some experiments myself at Plymouth University Ocean Basin. Benjamin Child, who supervised from the DNV-GL office in Bristol, was a very kind and efficient supervisor who took lots of his time to introduce me to several members of the DNV-GL team, facilitating software development and issues I have been facing. Gregorio Iglesias who has been always cheerful all my PhD long.

Of course, a big thanks to my family who even manage to bear some of my presentations, read some of my thesis, and help me to express difficult concepts with simple words and sentences: Mela la meilleure des soeurs, le Papa et la Maman. Merci Mamie, who watch over the Belledonnes for me, Papy who told me "Bonne route". The cousin.e.s team, Fanny, Alexandra, Anne-Sophie, Elisa, and Romain, for which I hope the best and thanks for their support at the yearly Chignin Christmas dinner (and random pop-up at their place), with the big family: Jeff, Cath, Maryse, Jean-Jacques.

Many thanks as well to my friends from Plymouth and France who help me make the most of my spare-time: Elodie, Clement, La Rue3, Feuyas, Ed, Kieran, Manu, Marina, Marta, François, Will, Clara, Ele, Olivier, Ross, the climbers, the B&R ...

Authors declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award. Work submitted for this research degree at Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment.

This work has been carried out by Pierre-Henri Musiedlak under the supervision of Prof. Deborah Greaves, Dr. Edward Ransley, Dr. Martyn Hann, Dr. Benjamin Child, and Prof. Gregorio Iglesias Rodriguez. This work was funded by the University of Plymouth and DNV-GL.

Word count for the main body of this thesis: **39211**

Signed: _____

Date: _____

Conference proceedings :

Musiedlak, P-H. , Greaves, D., Child, B., Iglesias, G., Hann, M., Ransley, E. (2017), 'Investigations of model validity for numerical survivability testing of WECs' *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*, Cork, Ireland.

Musiedlak, P-H. , Greaves, D., Child, B., Iglesias, G., Hann, M., Ransley, E. (2018), 'Starting a wave structure interaction CFD simulation from an advance time: hot-start', *Proceedings of the 6th European Conference on Computational Mechanics (ECCM 6) - 7th European Conference on Computational Fluid Dynamics (ECFD 7)*, 11-15 June 2018, Glasgow, UK

Brown, S. A., **Musiedlak, P.-H.**, Ransley, E. J., & Greaves, D. (2018), 'Numerical Simulation of Focused Wave Interactions With a Fixed FPSO Using OpenFOAM 4.1.' *International Society of Offshore and Polar Engineers*, 10-15 June, Sapporo,

Japan.

Ransley, E.J., Yan, S, Brown, S.A., Mai, T., Graham, D., Ma, Q., **Musiedlak, P-H**, Engsig-Karup, AP., Eskilsson, C., Li, Q., Wang, J., Xie, Z., Sriram, V., Stoesser, T., Zhuang, Y., Li, Q., Wan, D., Chen, G., Chen, H., Qian, L., Ma, Z., Mingham, C., Causon, D., Gatin, I., Jasak, H., Vukcevic, V., Downie, S., Higuera, P., Buldakov, E., Stagonas, D., Chen, Q., Zang, J., (2019), 'A blind comparative study of focused wave interactions with a fixed FPSO-like structure - CCP-WSI Blind Test Series 1' *nternational Journal of Offshore and Polar Engineering (IJOPE)*, ISSN 1053-5381

Posters presentations :

Musiedlak, P-H. (2016), 'Geometric parameters for a 2D-Numerical Wave Tank', Poster presentation at 3^{rd} PRIMaRE, 5-6 July 2016, Bath, UK; and at 10^{th} International Network for Offshore Renewable Energy, 12-18 June 2016: Nantes, France

Musiedlak, P-H. (2017), 'The validity of numerical models for testing the survivability of WECs', Poster presentation at 4^{th} PRIMaRE

Abstract

NUMERICAL MODELLING OF RESPONSES OF OFFSHORE WAVE ENERGY CONVERTERS IN EXTREME WAVES

Pierre-Henri Musiedlak

Wave Energy is a promising sector for sustainable low-carbon energy production. Despite the effort put in the last decade, concerning problems remained unsolved. Numerical modelling can play a key role in the assessment of extreme responses necessary to the understanding of Wave Energy Converter (WEC) capacity to survive extreme conditions. However, high-fidelity codes are too time-consuming for industries, while codes of lower fidelity are based on major physical assumptions inducing concerning uncertainties.

Therefore, the present study - sponsored jointly by DNV GL and the University of Plymouth - develops an hybrid model coupling the low-fidelity model WaveDyn, with a Computational Fluid Dynamic (CFD) Numerical Wave Tank validated against physical experiment of a single moored WEC under extreme events. A time-splitting technique allows to change between code as a function the confidence on the solution. The wave steepness is used to identify the limit in confidence for WaveDyn on survivability assessment. Additional developments allow to ensure a satisfactory start of the (CFD) simulation from an advance time where the wave-field is developed and the device is in motion. The coupled model overcomes identified inaccuracies in the WaveDyn code due to the inviscid assumption and the high computational cost of the OpenFOAM code.

Contents

List of Figures

List of Tables

Acronyms

$2D$ 2-dimensional

$3D$ 3-dimensions

BEM Boundary Element Method

CAD Computer Aided Design

CFD Computational Fluid Dynamics

CFL Courant-Friedrichs-Lewy

COP Conferences of Parties

CPWH cells per wave height

DoF Degree of Freedom

ELC External Load Controller

FFT Fast Fourier Transform

FSI Fluid Structure Interaction

FVM Finite Volume Method

GHG Greenhouse Gas

HPC High Performance Computing

IkA Instantaneous steepness

LES Large Eddy Simulation

MRE Marine Renewable Energy

MULES Multi-Dimensional Limiter for Explicit Solution

MWL Mean Water Line

NP Numerical Probe

NS Navier-Stokes

NWT Numerical Wave Tank

OWC Oscillating Water Column

OWSC Oscillating Wave Surge Converter

PIMPLE Pressure Implicit for Pressure Link Equations

PTO Power Take Off

PWT Physical Wave Tank

R&D Research and Development

RAM Random Access Memory

RANS Reynolds Averaged Navier-Stokes

RBM Rigid Body Motion

RMS Root Mean Square

SPH Smoothed Particle Hydrodynamics

VoF Volume of Fluid

WEC Wave Energy Converter

WG Wave Gauge

WSI Wave Structure Interaction

X-MED Extreme loading of marine energy devices due to waves, current, flotsam and mammal impacts

Nomenclature

α	Volume fraction field
α_{Ph}	Constant in Pierson-Moskowitz spectrum
χ_R	Scaled x-coordinate along the relaxation-zone
$\ddot{q}$	Joint acceleration $[m.s^{-2}]$
Δ	Laplacian operator
δ	Phase of linear wave
δ_f	Sampling frequency of a Fourier Transform
Δ_t	Time-step $[s]$
δ_{ij}	Kronecker delta ($\delta_{ij} = 1$ if $i = j$ else $\delta_{ij} = 0$)
$\dot{q}$	Joint velocity $[m.s^{-1}]$
η	Surface elevation $[m]$
Γ	Total forces acting on rigid-body $[N]$
λ	Wavelength $[m]$
λ_e	Linear interpolation factor from cell centres to cell face
λ_v	Second viscosity
μ	Dynamic viscosity
∇	Gradient operator
v	Fluid viscosity

ω Wave angular-frequency $[rad.s-1]$

$\overline{\phi}$ Time-averaged of the arbitrary field property ϕ

$\overrightarrow{\nabla}$ Divergence operator

$\overrightarrow{a}$ Acceleration vector of an arbitrary rigid-body $[m.s^{-2}]$

$\overrightarrow{F_p}$ Pressure force vector $[N]$

$\overrightarrow{F_v}$ Viscous force vector $[N]$

$\overrightarrow{F}$ Force vector $[N]$

$\overrightarrow{n}$ Normal vector

$\overrightarrow{u}$ Fluid velocity vector $[m.s^{-1}]$

$\overrightarrow{V}$ Velocity vector of an arbitrary rigid-body $[m.s^{-1}]$

ϕ Arbitrary field property

ϕ Velocity potential in potential flow

ϕ_E Arbitrary field property at cell centre E

ϕ_e Arbitrary field property on face e

ρ Fluid density $[kg.m^{-3}]$

σ Surface tension $[kg.s^{-2}]$

$\sigma(A)$ Standard deviation

τ Wave-period $[s]$

τ_{ij} Viscous stresses

A Wave-amplitude of largest wave (NewWave theory) $[m]$

a Amplitude of linear wave $[m]$

a_i Wave-amplitude of component i $[m]$

B Radiation damping $[Kg.s^{-1}]$

c Celerity of a linear wave $[m.s^{-1}]$

$C0$ Courant number

$E(A)$ Expected value of random variable

f Frequency $[Hz]$

f_p Peak frequency $[Hz]$

f_a Analytical solution of the imposed wave in the `waves2Foam` toolbox

f_e Excitation force in Cummins Equation $[N]$

f_e External force in Cummins Equation $[N]$

f_{hs} Hydrostatic force in Cummins Equation $[N]$

f_s Actual solution of the wave generated by the `waves2Foam` toolbox

g Acceleration due to gravity $[9.81\,m.s^{-2}]$

H Wave-height (crest to trough) $[m]$

$H_{n,s}$ n-year return siginificant wave-height $[m]$

Hs Siginificant wave-height $[m]$

IkA Instantaneous wave steepness defined as: $\frac{\partial \eta}{\partial t}$

k Wave-number $[m^{-1}]$

$k(t)$ Body impulse response function

kA Wave steepness

m Mass of an arbitrary rigid-body $[Kg]$

m_0 Zeroth moment $[m^2]$

M_a Added mass $[Kg]$

m_∞ added mass at infinite frequency $[Kg]$

N Number of components in an arbitrary sum

p Pressure $[Pa]$

p_rgh Non-hydrostatic pressure

p_0 Total pressure $[Pa]$

q Joint position $[m]$

R^2 Coefficient of determination defined as the square of the correlation

Re Reynolds number

$S(f)$ Power spectrum $[m^2.s]$

S_i Surface arbitrary face $[m^2]$

t_0 Focus location of a NewWave event in time $[s]$

T_p Peak wave period $[s]$

T_z Average wave period $[s]$

$T_{n,p}$ n-year return peak wave period $[s]$

u_i Component of the velocity vector $\overrightarrow{u}$ on axis i $[m.s^{-1}]$

x_0 Focus location of a NewWave event in space $[m]$

x_E Coordinates of cell-centre E in the mesh

x_e Coordinates of face e in the mesh

Chapter 1

Introduction

The effect of climate change is concerning and proven to be induced by human activities (IPCC 2014). The way energy is produced has a significant impact on Greenhouse Gas (GHG) emissions. The burning of fossil fuels directly generates them, whereas renewable energy sources, (those that are 'continuously replenished by nature' (Ellabban et al. 2014)), appear as more suitable for a low-carbon and sustainable development. The rise in renewable energy is noticeable, but the principal source remains fossil fuels, even if the European Renewable Energy Council (Zervos et al. 2010) estimates the potential of the renewable to be 3078 times the 2010 worldwide energy demand.

Marine Renewable Energy (MRE) is a source of significant power (potential of two times 2010 primary energy needs (Zervos et al. 2010)). MRE harnesses offshore or near-shore renewable energy sources, such as wave, tidal, current, temperature, salinity gradient, and offshore wind. Of course, MRE is more suitable for countries with a large coastline, and in Europe, the Atlantic arc from Scotland to Portugal gathers the biggest wave and tidal resource (SETIS 2013). Sheltered areas also have potential even if they have less energy. Survivability threatening events of less significance are likely to induce a reduction in device cost link to maintenance and robustness.

Wave energy is one source and sub-sector of MRE. Multiple of harvesting ideas and devices, called Wave Energy Converter (WEC), can be found because the systems interact with the wave train in different directions and ways. For example, an Oscillating Wave Surge Converter (OWSC), such as the WaveRoller device, harvests the wave perpendicularly using a flap that oscillates back and forth around the surge axis as

the wave passes; whereas, an attenuator like the Pelamis articulates along the wave direction. The third main category of device (Magagna and Uihlein 2015) is the point-absorber, which harvests the surface elevation difference of a wave by oscillating up and down like the buoy developed by Carnegie. Shoreline applications are also possible, for example with an Oscillating Water Column (OWC) system like the Pico that entrapped air inside a chamber to compress it due to the rise of surface elevation before powering an air turbine. The variety of devices and the extensive potential make the sector attractive, and it has gathered more investment than the tidal sector (Magagna and Uihlein 2015).

However, the WEC sector has suffered from lacks recently, with $700\,MW$ of the $1200\,MW$ projects announced capacity in Europe abandoned between 2009 and 2015 (Magagna and Uihlein 2015). Pelamis, the first offshore WEC to feed electricity to the grid in 2004, and the first project of WEC in an array (BBC 2005), went into administration in November 2014 (Magagna and Uihlein 2015). E-ON, the German electric supplier, one of the main supporters, had backed out from the Pelamis project the year before. Similarly, Aquamarine Power ceased to trade in 2015 (Magagna and Uihlein 2015) after ten years of development of their OWSC device. In 2014, the point-absorber of the company Carnegie did not survive the cyclone on the Reunion Island (Renewable Economy 2014).

The large number of failures in the WEC sector shows uncertainties that increase investor apprehension, hence slowing down the market growth. Key developments are required to break this cycle. Independently from the technology, the lifetime objective of a WEC device to capture energy divides into two required capacities dependent on the sea-state (Coe et al., 2018; Barstow et al., 2008), which both must be satisfied while ensuring the economic viability of the device. First, reliability defines that a device captures power in operational conditions through its design life-time. Second, survivability defines that a reliable device remains reliable after experiencing an event outside operational condition, i.e. extreme event. Therefore, the design obtained

assessing reliability, must be completed by one focusing on survivability.

Complex hydrodynamics phenomena are damaging the WEC in survivability conditions. Waves can break and generate violent impacts on devices. Such over-turning flows are highly turbulent and can entrapped air. Also, the size of the waves can exceed the maximum device motion and generate green water effects on the device deck. The consideration of the effect of these phenomena on survivability can be conducted using empirical models or by reproducing the physical process experimentally or numerically. Numerical models based on the weakly simplified Navier-Stokes equations, assumed of 'high-fidelity' (Figure 1.1), are necessary to model such hydrodynamics and the interaction with the device. Numerical models of lower fidelity are restricted by the assumptions made on the fluid properties (e.g. inviscid) or the equations solving procedure (e.g. linear potential flow).

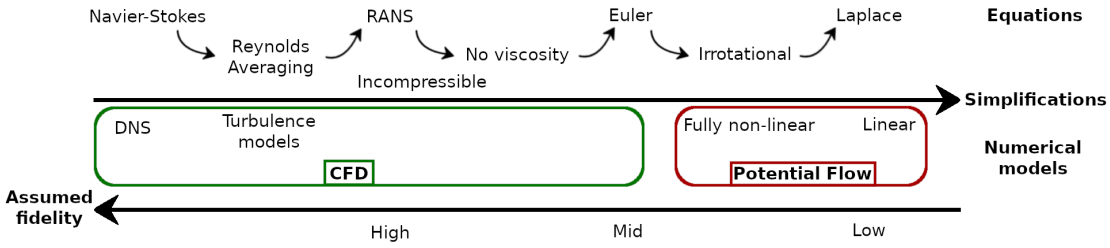


Figure 1.1: Numerical models assumed fidelity relative to the simplifications made on the Navier-Stokes Equations. Adapted from Guanche (2017) and Windt et al. (2018)

The use of models of low-fidelity is therefore debatable for survivability assessment of WEC. Henry et al. (2013) assessed the wave impact on an OWSC (a flap-type of WEC) using Computational Fluid Dynamics (CFD), which maximum load coincides with physical tank-testing experiments. CFD model is necessary to capture the disturbance of the water level near the flap, the consequences on the motion of the structure, and the resulting mechanical loads. Indeed, lower fidelity models cannot due to assumptions on fluid properties (e.g. inviscid). Therefore, it is often argued that modelling the extreme response of a WEC justifies the use of higher-fidelity models like CFD (Westphalen et al. (2009), Hu et al. (2016), among others) to include non-linear hydrodynamics.

For the engineering design, WEC developers benefit from Oil&Gas and shipping industry standards. Apart from the significant difference in investments, which justifies over-dimensioned of structure, another significant difference with the Oil&Gas traditional platforms is that WECs must not only be controlled to avoid damage but accentuated to generate power (Ransley 2015). As these standards assume small displacement as based on simplified physics, and in the light of the numerous failures among WEC devices, their applicability to WEC is concerning.

Indeed, Oil&Gas platforms and ships typical are smaller compared to WEC. Viscous effects therefore tend to become more important whereas diffraction and radiation effects tend to decrease. Hence, when radiation diffraction effects dominate on viscous one, Oil&Gas low-fidelity numerical models (see Figure 1.1) based on simplified physics remain reliable; whereas when viscous effects dominate high-fidelity numerical models like CFD are justified.

The assessment of WEC motion responses is crucial for survivability assessment. Assumptions made in lower fidelity models can induce under-estimation of motion responses, especially in surge and pitch (Heilskov et al. 2015). Inaccurate estimations of extreme response often lead to conservative design decisions that lower the commercial viability of a device (O'Neill et al. 2006). Therefore, numerical models of high fidelity are required to capture motions accurately and assess survivability precisely.

Despite the requirement of CFD for survivability assessment, the computational cost remains too expensive for the industry, especially as multiple irregular sea-states are recommended by the WEC standard (IEC TS 62600-2 (International Electrotechnical Commission 2016)). So, the use of CFD is restricted to single short extreme events (CCP-WSI Working Group 2016), which validity is unclear as unrealistic compared to an irregular sea-state. Therefore, low and mid-fidelity models remain widely used for their efficiency.

Uncertainties remain on which numerical model to use for a given application. The MERiFIC report classifies different numerical codes by the physical processes to model

(Vyzikas et al. 2014 - b). However, the absence of a parametric study function of a wave-parameter makes the choice of the selected model for survivability vague. Coe and Neary (2014) argues that in large waves up to the point of wave breaking, numerical models of lower fidelity (i.e. higher level of mathematical simplifications), are the most efficient methods as they give good agreement with experimental data for a fraction of the computational cost. Even though the level of accuracy is reasonably good in heave response, the accuracy of other Degree of Freedom (DoF)s is lower due to viscous effects (Heilskov et al. 2015), which requires CFD models. So, the present study conducts a parametric validity assessment of low and high fidelity numerical models below the breaking limit, which is beneficial to the overall MRE sector (Ransley et al. 2019).

Physical experiment of a WEC in survivability conditions assesses the validity of each numerical model. The first numerical model is the CFD code OpenFOAM, which solves the Navier-Stokes (NS) Equations for an incompressible two-phase flow. The second is the mid-fidelity linear time-domain model WaveDyn, which assumes an inviscid potential flow. The physical experiment done at the University of Plymouth Ocean Basin (COAST (2019) - later referred as such) was part of the X-MED project (Hann et al. 2015). This physical tank testing is the reference due to the known reliability of physical models heavily used though WEC design (CCP-WSI Working Group 2016). The motions responses of the X-MED buoy in all DoF assess the accuracy of each model to reproduce the experiment made of four extreme non-breaking events of increasing steepness. The parametric assessment uses the wave steepness (kA) to correlate validity with non-linearity, as the assumed use of CFD models is for highly non-linear flows, and the assumptions of mid-fidelity models is a linear flow. Besides, from this physical experiment, Hann et al. (2018) shows a significant effect of steepness on the amplitude of resulting motions. Therefore, the increase in steepness is expected to push the mid-fidelity model, WaveDyn, beyond its validity limit due to the assumptions on the fluid. On the other hand, the accuracy of the CFD model is expected to remain constant.

The assessment of each numerical model is expected to identify a parametric validity limit in the use of the mid-fidelity model, WaveDyn. So, WaveDyn use would be justified up until the limit, from which the CFD model is required to maintain the accuracy. Therefore, the present study develops a coupling between the two models based on the change from mid-fidelity to CFD at the limit; called time-coupling. The benefit of this coupling is the use of the most suitable model. Indeed, the low computational cost of WaveDyn model is beneficial up to the limit, hence, reducing the use of the computationally expensive CFD models. Therefore, this coupling model aims to make affordable for industries simulations to a level of accuracy of CFD, especially long irregular sea-state widely used in design procedures (e.g. CCP-WSI Working Group (2016), Coe et al. (2018), Rafiee et al. (2016)).

In summary, this research follows one principal aim, which is to

- Make computationally affordable simulations to the level of accuracy of CFD by developing and validating against physical experiments, a numerical tool modelling the motion responses of a WEC subjected to extreme events, by coupling the CFD package OpenFOAM with the potential flow time-domain model WaveDyn to benefit from the most suitable model.

This research requires the development of the CFD model as a Numerical Wave Tank (NWT) mirror of the physical one; the assessment of the validity of both models against physical experiment; and the coupling between the two models. This aim is decomposed into six objectives based upon the critical unknown of this research:

1. Examine the key components and parameters of a CFD based NWT for extreme Wave Structure Interaction (WSI)
2. Evaluate the ability of the NWT to reproduce accurately physical tank experiments of extreme event
3. Identify the range of use of the potential-flow time-domain WaveDyn as a function of wave steepness, and specify the reliability of each model

4. Investigate and implement the requirements for a CFD WSI simulation to start from an arbitrary time

5. Couple the potential-flow time-domain model, WaveDyn, to the CFD, OpenFOAM

6. Demonstrate the use of the coupled tool

After this introduction (Chapter 1), this research starts with a chapter of literature review and description of related background theory. Follows three chapters detailing the work performed in this study to achieve the development of the numerical coupling: first the NWT for the modelling of extreme WSI (Chapter 3); second, the assessment of models validity and the identification of WaveDyn validity limit (Chapter 4); and third, the coupling made possible from the two previous chapters (Chapter 5). Finally, a set of conclusions on the present work and potential future developments are presented in the final chapter, Chapter 6.

Chapter 2 starts by stating the mathematical model of fluid dynamics, the NS equation, in order to formulate the models used by the two numerical solver; the Reynolds Averaged Navier-Stokes (RANS) for OpenFOAM and potential flow for WaveDyn. After a brief presentation of wave-theory, survivability design of WEC is detailed to discuss on the use and applicability of numerical models. Potential flow solvers are then described to explain the solving procedure implemented in WaveDyn, followed by the one used by OpenFOAM for CFD applied to WSI. Coupling techniques between numerical models are then reviewed to present the solution chosen in this study. The experimental set-up of the single moored X-MED buoy under extreme events of increasing steepness is finally detailed, as it is used as the reference for the assessment of validity of the two numerical models - WaveDyn and OpenFOAM.

The third chapter relates the development of a CFD based NWT aiming to replicate extreme physical experiments. First, a 2-dimensional ($2D$)-NWT focusing on the wave-only accuracy is developed by comparing its solution with another numerical solution identified as greater accuracy. This investigation aims to develop a NWT adaptable

to other experiments made at the same facility. The NWT is then validated against, first, wave-only experiments, and then against the 6-DoF motion responses of the X-MED buoy under these four extreme events of increasing steepness. At the end of this chapter, a tool for extreme WSI is validated against experiments and can be used later for coupling.

Chapter 4 compares the motion responses of the two numerical models OpenFOAM and WaveDyn. The assessment is conducted against experiments to quantify the accuracy of each model for extreme WSI. From an identified inaccuracy, the validity of WaveDyn model is parametrically assessed as a function of the event steepness. Additional experiments data are used to confirm the assessment, and evaluate the limit of validity of WaveDyn where the OpenFOAM model is found to remain valid.

Therefore, Chapter 5 details the coupling strategy made possible by the two previous chapters: WaveDyn model is used up until its limit, where CFD is necessary and so, the NWT model starts. Starting the CFD model from an advance time using the rigid-body-motion state from WaveDyn requires developments that are detailed in this chapter. Then, the coupling strategy is applied to prove the concept of the coupled tool to a long irregular sea-state.

Conclusions made by this research with their limitations are discussed in Chapter 6 alongside potential for future work.

Chapter 2

Review of Literature and Background Theory

This chapter aims to give details on the background theory and literature review relevant to the numerical modelling of motion responses of Wave Energy Converter (WEC) in extreme conditions.

This chapter starts by drawing the mathematical model for Fluid Dynamics, the Navier-Stokes (NS) Equations. From it is formulated the mathematical model of each numerical solver. OpenFOAM establishes the Reynolds Averaged Navier-Stokes (RANS) Equations using the Reynolds averaging method for an isotropic, Newtonian, incompressible flow. WaveDyn considers an inviscid and irrotational, hence potential flow; from which wave-theory is presented.

Fluid dynamics mathematical models and wave-theory established, the chapter discusses the design practice for the survivability assessment of WEC. It starts with the evaluation of extreme events generating survivability loads. A discussion on the use of extreme events modelling techniques introduces the design-wave selected by the present study. Then, a review of numerical models for survivability assessment of WEC discusses the application of each numerical model (WaveDyn and OpenFOAM) as a function of physical assumptions.

The chapter presents the solving procedures of each numerical model, where the Computational Fluid Dynamics (CFD) solving procedure conducted for Wave Structure Interaction (WSI) in OpenFOAM receives specific attention. An overview of the coupling

techniques draws advantages and drawbacks to present the 'time-coupling' strategy developed by the present study. Finally, the experimental study used as the reference is detailed.

2.1 Mathematical Models for Fluid Dynamics

Fluid is a perfectly deformable matter that continuously deforms under shear stress, i.e. the ability to flow. Fluids can include liquids, gases and plasma, however, most commonly refers to a liquid. The difference between liquid and gas is that a liquid forms a free-surface when in a container, while gas takes the whole volume available.

Sir George Gabriel Stokes and Claude-Louis Navier established a mathematical model, the Navier-Stokes equations, that describes the motion of fluid by the variation of its properties, the pressure p, the velocity u, and the density ρ.

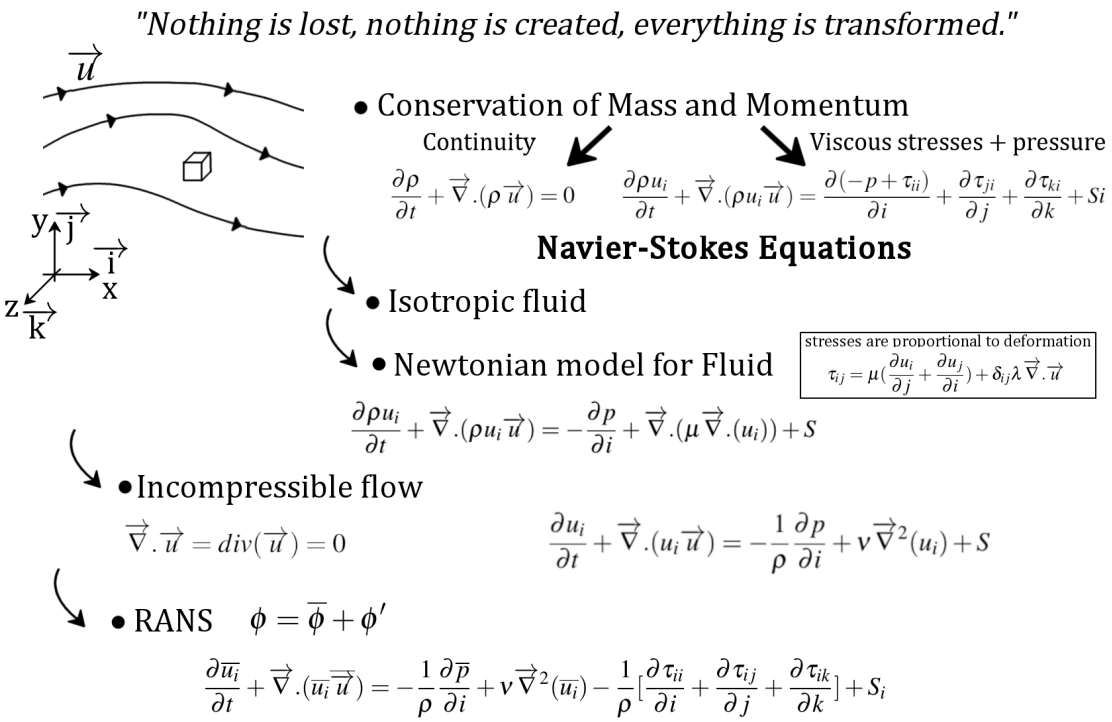


Figure 2.1: Illustration of the shaping of the NS equations applied to the WSI analyse in this study: An isotropic, Newtonian, incompressible, and Reynolds Averaged flow

Figure 2.1 illustrates the required steps in the establishment of the NS equations, and

its application to WSI for an isotropic, Newtonian, incompressible flow assuming an average variation of the flow properties. The following section refers to this diagram inspired by the work of Versteeg and Malalasekera (2007), which provided further details on the equations.

This study uses the over-right arrow notation for a vector. For example, the fluid velocity vector is $\overrightarrow{u}$, and the components of the vector the Cartesian frame ($\overrightarrow{i}$, $\overrightarrow{j}$, $\overrightarrow{k}$) are u_i, u_j and u_k. $\overrightarrow{\nabla}$ is the divergence operator, ∇ is the gradient, and Δ is the Laplacian.

2.1.1 Shaping of the Navier-Stokes Equation

The conservation of mass, the conservation of momentum (or Newton's Second Law of Motion), and the conservation of energy (or the First Law of Thermodynamics) are the three fundamental physics axioms describing the fluid dynamics of an isolated system. As the influence of thermal exchanges is negligible in the present application, the conservation of energy is not considered here.

As shown by the cube in left-hand side picture in Figure 2.1, the conversation principles apply specifically to a small volume of fluid within a flow field in a Cartesian frame ($\overrightarrow{i}$, $\overrightarrow{j}$, $\overrightarrow{k}$) for convenience.

Conservation of mass

The Conversation of Mass principle states that the rate of increase of mass is equal to the net rate of flow of mass. The continuity equation mathematically defines this for a compressible fluid as (Versteeg and Malalasekera 2007):

$$\frac{\partial \rho}{\partial t} + \overrightarrow{\nabla}.(\rho \overrightarrow{u}) = 0. \tag{2.1}$$

Conservation of momentum

The Conservation of Momentum links the sum of sources applying on the volume of fluid to the rate of change of the volume of fluids momentum. Forces applied to the volume surface are generated by pressure, p, and viscous stresses, τ. A source term

S gathers additional body forces applying to the whole fluid, such as gravity or Coriolis force. In the Cartesian frame for the projection on the i-axis, the Conservation of Momentum is:

$$\frac{\partial \rho u_i}{\partial t} + \overrightarrow{\nabla}.(\rho u_i \overrightarrow{u}) = \frac{\partial(-p+\tau_{ii})}{\partial i} + \frac{\partial \tau_{ji}}{\partial j} + \frac{\partial \tau_{ki}}{\partial k} + Si \tag{2.2}$$

where τ_{ij} is the stress on a face of normal $\overrightarrow{i}$ along the $\overrightarrow{j}$ direction. The set of Equations 2.1 and 2.2 are the NS Equations for an isothermal fluid.

2.1.2 Application to WSI

For common fluid dynamics problem in offshore applications, the NS equations model the flow assuming an isotropic, Newtonian, and incompressible fluid where instantaneous variations in the flow are averaged over a period of time (Reynolds Averaging) as the interest is the mean flow variation.

Isotropic Newtonian Fluid

The fluid being isotropic means that it deforms uniformly in all direction. The Newtonian model for fluid links the viscous stresses, τ_{ij}, to the deformation, $\frac{\partial u_i}{\partial j}$, using mechanical properties of the volume of fluid:

$$\tau_{ij} = \mu(\frac{\partial u_i}{\partial j} + \frac{\partial u_j}{\partial i}) + \delta_{ij}\lambda_v \overrightarrow{\nabla}.\overrightarrow{u}, \tag{2.3}$$

where, μ is the dynamic viscosity, λ_v the second viscosity, and δ_{ij} the Kronecker delta ($\delta_{ij} = 1$ if $i = j$ else $\delta_{ij} = 0$). Figure 2.1 illustrates this step by the two curved arrows.

Incompressible

The fourth step of Figure 2.1 assumes an incompressible flow, meaning that the fluid density is constant. The continuity equation 2.1 simplifies to :

$$\overrightarrow{\nabla}.\overrightarrow{u} = div(\overrightarrow{u}) = 0. \tag{2.4}$$

The Momentum Equations are re-written by introducing the fluid kinematic viscosity (defined as $\nu = \mu/\rho$) while using a Laplacian term coming from the Newtonian model of fluid (Equation 2.3):

$$\frac{\partial u_i}{\partial t} + \overrightarrow{\nabla}.(u_i \overrightarrow{u}) = -\frac{1}{\rho}\frac{\partial p}{\partial i} + \nu \overrightarrow{\nabla}^2(u_i) + S_i. \tag{2.5}$$

Due to the coupling between equations 2.4 and 2.5, no analytical solution exists for this system. Approximate solutions are obtained using time and space discretisation methods computed by CFD codes.

RANS

Due to non-linear instantaneous flows, fluctuations are averaged over a small period of time. Each flow property Φ decomposes into its average contribution, $\overline{\Phi}$, plus its fluctuating one, Φ' (Reynolds decomposition):

$$\Phi = \overline{\Phi} + \Phi'. \tag{2.6}$$

This method applied to the incompressible NS equations gives the final step in Figure 2.1, where the Continuity equation becomes:

$$\overrightarrow{\nabla}.(\overline{\overrightarrow{u}}) = 0, \tag{2.7}$$

while the Momentum equations become, for a given axis-i:

$$\frac{\partial \overline{u_i}}{\partial t} + \overrightarrow{\nabla}.(\overline{u_i}\overline{\overrightarrow{u}}) = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial i} + \nu \overrightarrow{\nabla}^2(\overline{u_i}) - \frac{1}{\rho}[\frac{\partial \tau_{ii}}{\partial i} + \frac{\partial \tau_{ij}}{\partial j} + \frac{\partial \tau_{ik}}{\partial k}] + S_i. \tag{2.8}$$

Compared to Equation 2.5, he extra terms that appear in the equation 2.8 are the Reynolds stresses coming from the Laplacian term. They are expressed as $\tau_{ij} = -\rho\overline{u_i' u_j'}$ (via the notation of Equation 2.3) and are six unknowns in the NS equations, in addition to the three velocity components and the pressure of the mean flow.

This set of equations, called the Reynolds Averaged Navier-Stokes (RANS), are used for the study of WSI in which the solving procedures are performed using CFD (later detailed in section 3.1 for OpenFOAM). Averaging the variation of the flow within a small period focuses the interest on the mean flow, rather than the fine definition of complex instantaneous local changes such as small turbulence eddies, that requires significantly small time and space discretisation for their modelling.

Potential Flow

By neglecting the viscosity, a laminar incompressible fluid becomes inviscid. It differs from a laminar assumption, as this one neglects the interaction between adjacent layers of fluid, but do consider viscosity.

Additionally, flow is considered as irrotational, meaning that its vorticity is zero (i.e. local spinning of a fluid element is zero). Mathematically, the curl of the flow is equal to zero at all point in the domain:

$$\overrightarrow{rot}(\overrightarrow{u}) = \overrightarrow{\nabla} \wedge \overrightarrow{u} = \overrightarrow{0}. \tag{2.9}$$

An inviscid and irrotational flow, i.e. potential flow, allows the existence of a potential ϕ defined as the gradient of velocity, which gives the Laplace Equation from the Continuity one:

$$\exists \phi \quad \text{satisfying} \quad \overrightarrow{\nabla}\phi = \overrightarrow{u}, \quad \Rightarrow \quad \overrightarrow{\nabla}^2\phi = 0. \tag{2.10}$$

This equation is the base for potential flow numerical models, like WaveDyn, that allows a less computationally expensive solving procedure than CFD, as the Laplace equation can be solved analytically. WaveDyn solving procedures is later detailed in section 4.1.

Equation of Motion

Neglecting hydroelastic deformation, modelling the motion of structures in hydrodynamics is commonly conducted using two approaches. The immersed boundary method

(Peskin 1972) considers the structure as an additional solid phase within the fluid. The fluid solid interface and the presence of the solid within the fluid is specifically treated by the numerical method forcing the solid properties at the interface. As a fluid phase with specific properties, the solid motion is solved via the momentum equation (Equation 2.2).

The second approach uses Newton second law of motion calculated from the pressure force plus the viscous shear acting on the solid. The first is calculated by integrating pressure over the solid surface. The second is the integration of shear forces due to viscosity and relative to the body-fluid motion. Newton second law of motion calculates the solid acceleration from the sum of forces. The solid is then displaced accordingly by integrating acceleration over the current time-step.

Due to the solving procedure, the immersed boundary condition is not affected by large displacement. Hence, it has the advantage of reducing the time-consuming mesh generation at each time-step induced by the solid motion, and the potential error linked to mesh quality (Windt et al. 2018). In the other hand, the immersed boundary method is often not implemented within the CFD code. Besides, the method can indicate concerning stability problems.

2.1.3 Wave Theory

Linear theory

Solving the Laplace equation 2.10 using boundary conditions on the surface and seabed (Figure 2.2), allows to established Stokes (Stokes 1847) or Airy (Airy 1845) wave-theory. It describes the evolution in space (x) and time (t) of the free-surface $\eta(x,t)$, represented in Figure 2.2 as:

$$\eta(x,t) = acos(kx - \omega t + \delta), \tag{2.11}$$

where, λ, k, ω, δ, a are respectively the wavelength, wave-number, angular-frequency, phase, and amplitude of the wave. This wave is either called regular, or linear, or Airy,

or Stokes 1^{st} order wave.

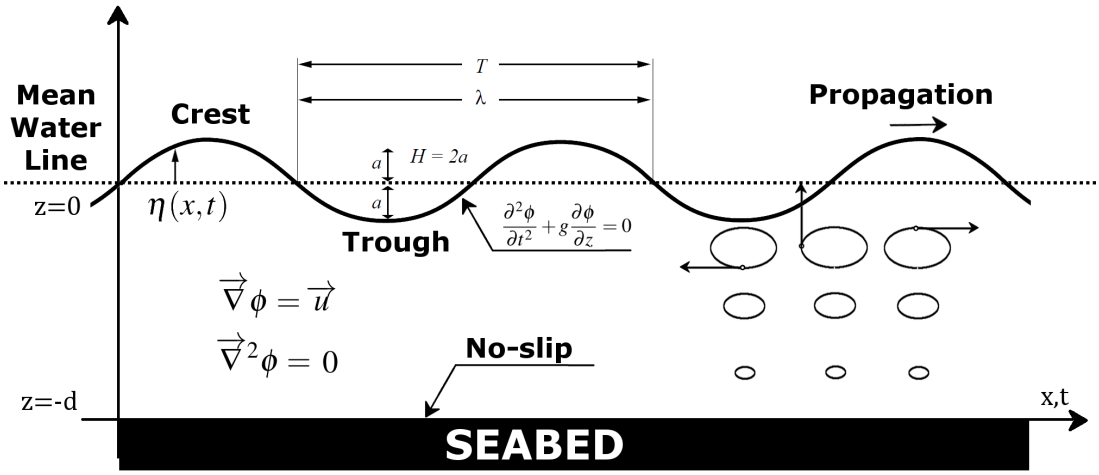


Figure 2.2: The surface Elevation η for an Airy wave, or linear wave or Stokes 1^{st} order wave - adapted from Le Boulluec et al. (2013) and the World Meteorological Organization, Bouws et al. (1998)

The wavelength λ is the distance between two successive crests, related to the wave-number k by $k = 2\pi/\lambda$. The wave-amplitude a is the maximum height displacement from the mean-water line. As a linear wave is symmetrical, the distance from crest to trough is the wave height H, equal to $H = 2a$. The period of the wave is τ, its frequency $f = 1/\tau$, so its angular-frequency $\omega = \dfrac{2\pi}{\tau} = 2\pi f$. The wave travels at speed $c = \lambda/\tau$. The wave propagation is described by the dispersion relation:

$$\omega^2 = gktanh(kd), \tag{2.12}$$

where g is the gravity, and d the depth.

Fully Non-Linear Potential Flow

Since the flow is irrotational (hence $\overrightarrow{\nabla}.(u_i\overrightarrow{u}) = \overrightarrow{u}.\overrightarrow{\nabla}.\overrightarrow{u} = \frac{1}{2}\overrightarrow{\nabla}\overrightarrow{u}^2$), inviscid ($\nu = 0$)), and therefore potential ($\overrightarrow{u} = \overrightarrow{\nabla\phi}$), the NS equation 2.5 becomes the Euler Equation (or unsteady Bernoulli):

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}.(\overrightarrow{\nabla \phi})^2 + g.z + \frac{p}{\rho} = f(t) \tag{2.13}$$

Where, $f(t)$ is a function dependent only of time specifically evaluated depending on the application.

The fully non-linear potential flow dynamic boundary condition on the free surface ($z = \eta$) is defined by specifying the pressure to the constant atmospheric pressure. At the free-surface the total time-derivative of the pressure is therefore zero:

$$\frac{Dp}{Dt} = 0 \Longleftrightarrow \frac{\partial p}{\partial t} + \overrightarrow{u}.\overrightarrow{\nabla p} = 0 \tag{2.14}$$

By merging the two preceding equations, the fully non-linear potential flow dynamic boundary condition on the free surface ($z = \eta$) is (Le Boulluec et al. 2013):

$$\frac{\partial^2 \phi}{\partial t^2} + g.\frac{\partial \phi}{\partial z} + \frac{\partial}{\partial t}(u^2) + \overrightarrow{u}\overrightarrow{grad}(\frac{u^2}{2}) = \frac{\partial f}{\partial t} \tag{2.15}$$

The linear boundary condition can be recovered by neglecting quadratic terms.

Wave Theories Applications

Specific environmental conditions use other wave-models, like the Cnoidal wave usually used for shallow water cases. Using the two quotients, $\frac{H}{g\tau^2}$ and $\frac{d}{g\tau^2}$, Le Méhauté (1976) defined the map of the validity of each wave-theory (Figure 2.3) commonly used in marine engineering.

Figure 2.3 shows the limits of each steady regular wave model. Linear theory is only valid to model intermediate to deep water waves of low height for a given period; identified by the yellow area on Figure. Cnoidal wave theory validity is indicated by the light-blue area. Higher orders of Stokes theory are separated by dashed blue lines.

Larger waves are modelled by wave theories of higher orders. In marine engineering design Stokes wave theories up to the fifth order are commonly used for extreme wave

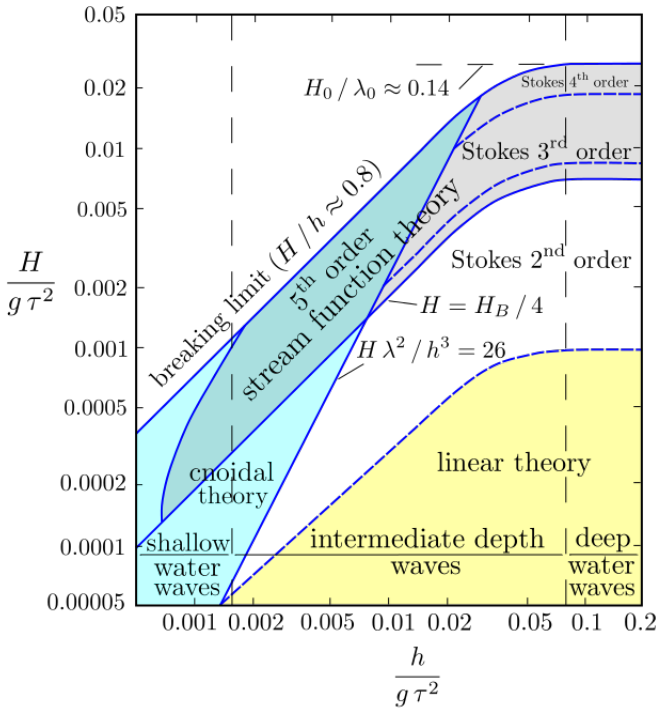


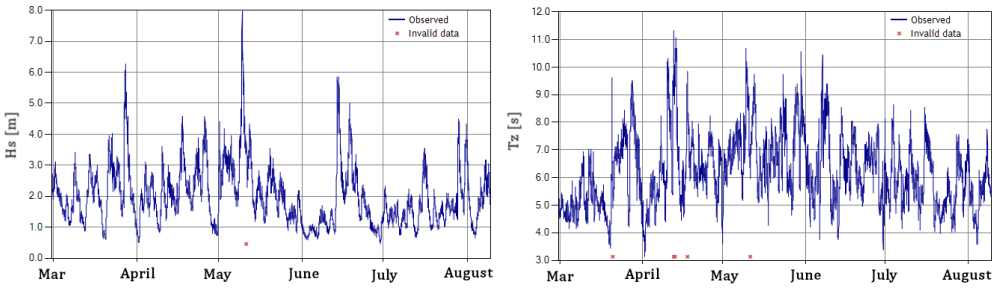
Figure 2.3: Validity of several theories for periodic water waves (Le Méhauté 1976)

loads assessment Tromans et al. (1991). Since, extreme waves are large and steep waves potentially breaking - hence near the breaking limit; more detailed on extreme waves is given in section 2.2 - linear theory seems unsuitable for the simulation of such waves.

Linear theory defines a symmetrical wave around the mean water line. Higher order theories (Stokes higher order in intermediate to deep water, and Cnoidal in shallow water) define waves with peaked crests above the water level and flat troughs below. The resulting shapes are more realistic when modelling a wave sensibly larger than the mean water level. Additionally, linear theory defines that the water particles move in closed orbits resulting in no net transport of fluid. Nonlinear theories predict a transport in the direction of the wave propagation. Laboratory measurements of linear wave show that particle orbits are not completely closed due to the mass transport phenomenon U.S. Army Corps of Engineers (2007). Hence, linear theory is not perfectly appropriate in modelling the wave motion of a single wave.

Description of a Sea-state

A given sea-state rarely corresponds to a unique wave, but a composition of multiples interacting waves. The World Meteorological Organization (Bouws et al. 1998) gives a detailed guide on the analysis of wave and sea-state using in-situ measured and statistical parameters available to characterise a sea-state.



(a) The significant wave height H_s *(b)* The average wave period T_z

Figure 2.4: Measures from March 2018 to August 2018 in Blackstones - obtained via WaveNet

Measurements tools, such as buoy data or satellite imaging, evaluates the wave parameters of a given sea-state. For example, the WaveNet project uses buoys along the UK coastline to monitor in real-time the significant wave height H_s (the average height of the $1/3$ highest waves (Bouws et al. 1998), Figure 2.4a), or the average zero-crossing wave period T_z (Figure 2.4b) for 30 minutes long time-series of the measured surface elevation η.

Figure 2.5 illustrates the different time scales used to describe sea-states, depending on the application. Such data are very useful for the prospective installation of an Marine Renewable Energy (MRE) device to calculate the power-generation or assess survivability loads. For example, considering a given site from the WaveNet project, the significant wave height larger than the yearly average (e.g. 2 times) can identify a storm, which surface elevation time-series could be then used to replicate survivability conditions.

By definition of the linear theory previously described, a sum of linear solutions is also

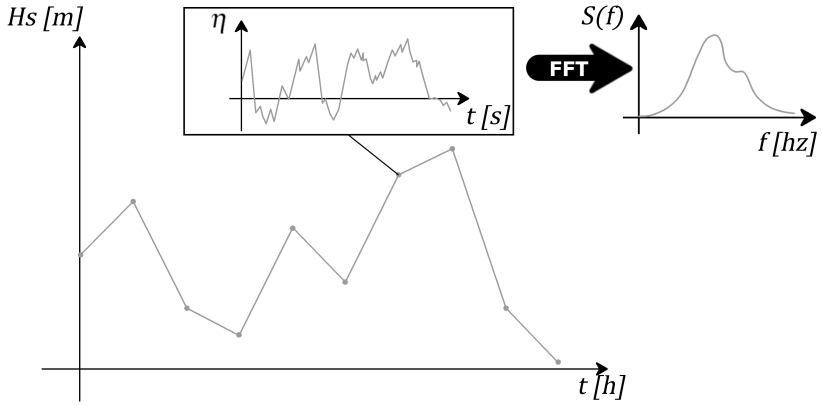


Figure 2.5: The different time scales for the description of a sea-state: long-term using the significant wave height H_s; short-term using the surface elevation η, from which the wave-spectrum of the short sea-state can be established

a solution. Therefore, a Fourier analysis (Fourier 1822) defines the surface elevation time-series of an irregular sea-state as a sum of N wave component:

$$\eta(x,t) = \sum_{n=0}^{N} a_n cos(k_n x - \omega_n t + \delta_n), t \tag{2.16}$$

where a_n, k_n and w_n are respectively the amplitude, the wave-number and the angular-frequency of the n^{th} linear wave component. A wave-energy spectrum represents the distribution of wave-components as a function of their contribution to the overall sea-state (right-hand side in Figure 2.5) using the density of power defined as:

$$S(f) = \frac{a_n^2}{2\delta_f}, \tag{2.17}$$

where δ_f is the sampling frequency used in the Fourier Transform.

If no on-site data are available, an idealised wave-spectrum can define a realistic sea-state from a spectral definition. The Pierson-Moskowitz spectrum is an example widely used across offshore industries because it defines the maximum wave height waves can reach due to the blowing of the wind. Its spectral density is defined as:

$$S(f) = \alpha_{Ph} g^2 (2\pi)^{-4} f^{-5} exp(-\frac{5}{4}(\frac{f_p}{f})^4), \tag{2.18}$$

where f_p is the peak frequency, and α_{Ph} is a constant (Pierson Jr and Moskowitz 1964).

The linearity of the sum in Equation 2.16 means that each wave component is independent of the other. However, an energy transfer exists between waves of different frequencies, called wave-wave interaction (Bouws et al. 1998). Techniques like Fourier transforms become no longer applicable (Dingemans 1997) since non-linear systems do not satisfy the superposition principle. This means that a wave-group described by the linear sum at a point in space and time will travel according to linear theory differently than its actual description, even for a unidirectional wave-group. The linear propagation of a wave-group is different from its actual propagation. Wang et al. (2017) compare the simulation of ocean waves in large spatial and temporal scale (128 peak wave-length and 1000 peak periods) using numerical models of different level of simplifications of physical equations. The linear model is found satisfactory (error inferior to 5%) compare to the fully non-linear model for extremely small steepness waves.

2.2 Survivability Assessment

From the NS equations ensue potential flow theory and the description of single wave and multiple sea-states. The description and identification of extreme events remain in the assessment of survivability.

Within the device economic viability, surviving extreme events is one of the two major requirements of an MRE device, along with extracting power in operational conditions (Coe et al. (2018), Barstow et al. (2008)). Survivability differs from reliability as, the first is the capacity to withstand all events through the design life-time (i.e. a reliable device before an event outside its normal use, is still reliable after); while the second is the insurance of power production in a normal use along the design life-time. The present study focuses on survivability since identified as the foremost concern in the following 5 years for WEC developers (CCP-WSI Working Group 2016). Also, this study chooses to focus on the estimation of extreme rather than fatigue loads, because the first threaten the integrity of the structure if applied once, whereas the second is of lower magnitude and requires multiple cycles to be a threat.

Survivability assessment starts from evaluating which loads are a threat for the WEC device and ends at modelling these loads. Figure 2.6 proposes a thinking principle formulated by Coe et al. (2018). Questions in italics highlight the steps of the thinking process. The overall idea is to reproduce such loads via the motion response under identified extreme events. So, from the survivability loads considered as damaging the integrity of the device, are questioned which sea-states could generate these loads: typically, a storm of significant importance. The storm is modelled as either a whole, or using a design wave aiming to represent the event of most significance for the survivability of the device. Finally, a numerical simulation or a physical experiment uses the model of the storm to assess the response motion of the structure, from which is deduced the extreme load.

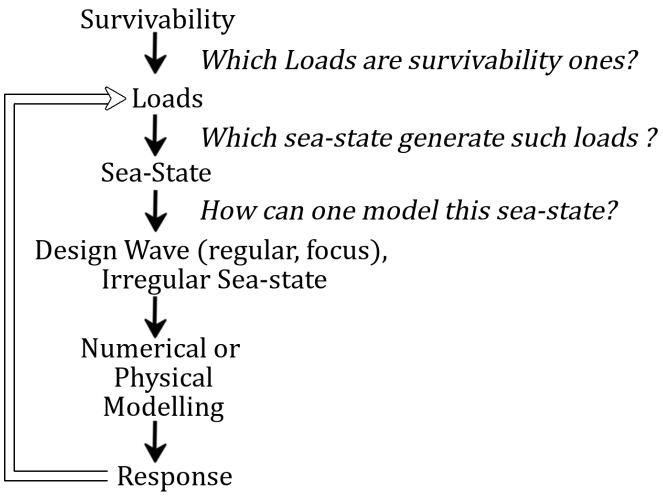


Figure 2.6: General survivability assessment thinking from extreme loads to response

2.2.1 Evaluation of Extremes Sea-States

Figure 2.7 illustrates the steps in the evaluation of the extreme sea-states used for survivability assessment. The objective is to characterise these extreme sea-states (i.e. storms) as a function of sea-state parameters, Hs and T_z (or Tp, the wave-period of the peak of the spectrum). Design practices select multiple sea-states to decrease the uncertainties of the evaluation.

The evaluation of extreme sea-states depends on the life-time and usage defined at

the early stages of the design considering the overall cost of the project (first step in Figure 2.7). The risk of failure developer considered as acceptable drives this first step. The risk is defined by the probability of occurrence for the response of interest (e.g. load, mooring tension, Power Take Off (PTO) force), to exceed a given limit. This can be expressed as a n-year return condition, meaning that response exceeding the limit will, on average, occur once every n years (second step in Figure 2.7). Standards and guidelines detail the procedure and recommend design conditions. The standard specific for WEC (IEC TS 62600-2 International Electrotechnical Commission (2016)) recommends a 50-year return value. Offshore oil and gas platform typically use a more restrictive 100-year period (NORSOK 2007), as a way of including a safety factor since failure can induce loss of life (Coe et al. 2018). A WEC life could be estimated from 20 to 30, which legitimates the use of 50-years return period (Coe and Neary 2014).

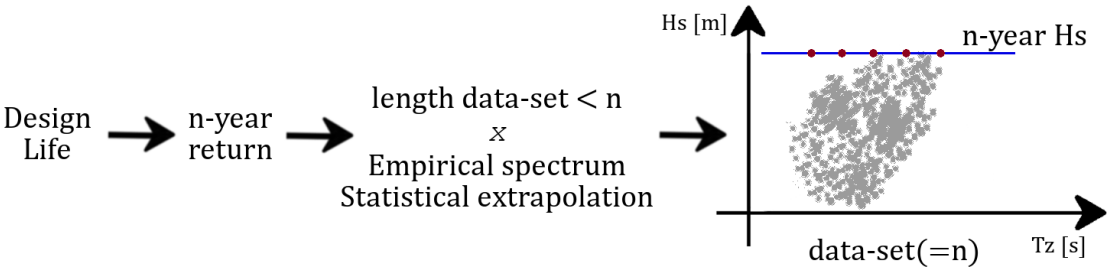


Figure 2.7: Evaluation of extremes sea-states from design life-time

Return periods require a long environmental recording to be used directly, which are unavailable (too long, challenging and expensive), and therefore extrapolation of the record is necessary (third step in Figure 2.7). Different statistical approaches (I-FORM, Nataf) are possible to obtain extreme sea-states from shorter recording. Also, it is common practice to use established record (i.e. empirical distribution such as Pierson-Moskowitz wave-energy spectrum, Equation 2.18) and adapt them to the deployment site of interest.

Design practices select several survivability conditions from the extrapolated data set of a n-year return period (fourth step in Figure 2.7). Each survivability condition characterises an extreme sea-state as a function of wave-parameters, Hs and T_z. The WEC

standard, IEC TS 62600-2, recommends selecting a range of wave period $T_{z,50}$ with the 50-year significant wave height $H_{s,50}$, red dot in Figure 2.7. More restrictive approaches can be used, and will select a larger range of events.

The present study uses the Pierson-Moskowitz wave-energy spectrum to model a 100-year return period storm from a hindcast ($Hs = 14.4m$, peak wave-period, $Tp = 14.1s$ Halcrow (2006)) at Wave Hub (2019) (Hann et al. 2015). A single extreme sea-state is used as the interest remains on modelling of motion responses from both numerical models, rather than the overall assessment of survivability for the X-MED buoy.

2.2.2 Definition of Extreme Event

Each identified extreme sea-state defines an extreme event (third question in Figure 2.6), either by reproducing one irregular sea-state or by defining a design-wave.

A $3h$ sea-state realizations (equivalent to 1000 waves (Hunt-Raby et al. 2011)) commonly defines one irregular sea-state. Due to its time requirement, this methodology is therefore only adapted to physical wave-tank experiment or fast (i.e. low/mid fidelity) numerical models; hence not CFD. The coupled model developed in the present study aims to make affordable these long simulations to a model to the fidelity of CFD by simulating the major part of simulation with the mid-fidelity model, hence limiting the use of CFD.

Design-waves describe the extreme sea-state as a short time-series making them affordable for high-fidelity modelling like CFD. The WEC developer Carnegie (2019) applied the irregular wave-design procedure using a scaled $3h$ sea-state at the COAST to assess the design wave-group (Rafiee et al. 2016). Once identified, a single design wave model was applied in a CFD numerical model to allow further investigations. This interesting combined approach highlights the use of the right design-wave depending on the model, physical or high-fidelity numerical.

A regular wave can be used as design-wave by defining its wave height proportional to the n-year significant wave height $H_{n,s}$. The NORSOK (2007) standard recommends a

1.9 proportionality assuming a Rayleigh distribution of the wave amplitudes in the extreme sea-state. However, the applicability of regular design-wave seems only adapted to specific environmental conditions of narrow banded wave-spectra (Coe et al. 2018). Besides, a regular wave does not represent a realistic sea-state (i.e. irregular), making the applicability of regular design-wave questionable.

Therefore, more elaborate definitions of design-wave have emerged to benefit from the short description of a single event with a specific surface elevation. Tromans et al. (1991) introduced the 'NewWave' wave model (later detailed in section 2.2.2) which proved its ability to model a more realistic description of the kinematics of the largest waves of a given sea-state. This model has been widely used for offshore and WEC applications (e.g. Walker et al. (2004), Bateman et al. (2012), Hann et al. (2015)). In the study previously mentioned (Rafiee et al. 2016), the most restrictive event for the PTO identified in several $3h$ irregular sea-states, is found with a shape in surface elevation similar to a NewWave event. These researches give confidence in the choice made by the present study to use NewWave events to assess survivability for the X-MED buoy.

However, the use of design-wave like NewWave is debated as no history of motion is incorporated in the resulting responses. Their applicability is especially questionable for structures capable of motions of large amplitude, as these can show that the extreme response does not necessarily occur due to the extreme design-wave (Taylor et al. (1997), Hann et al. (2015)). To tackle this problem, a NewWave event can be constrained into a regular or irregular sea-state to simulate the effect of load history and structural dynamics. Hann et al. (2018) shows that the modelling of extreme responses of the X-MED buoy using individual NewWave events is insufficient. However, this technique requires long simulations unaffordable in CFD, but where the coupled model aims to be used. The assessment of each two numerical models validity uses single NewWave design-wave besides these identified limits as long irregular sea-state are inapplicable for CFD but in light of an application using the coupled model to a long irregular sea-state.

NewWave model

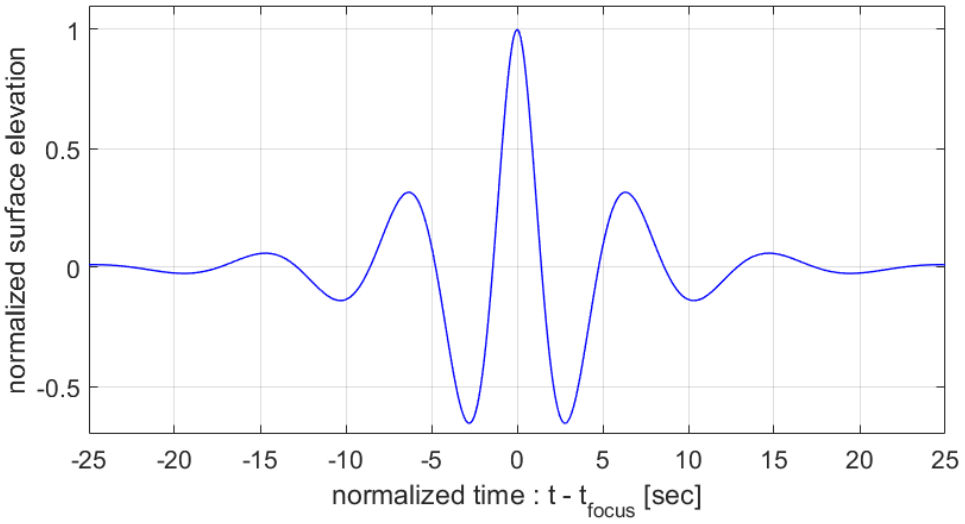


Figure 2.8: Normalized NewWave shape

NewWave theory, introduced by Tromans et al. (1991), describes the design-wave based on the wave-spectrum of the identified extreme sea-state representative for the assessment of survivability, as (at first order):

$$\eta = \eta^{(1)} = \sum_{i=1}^{N} a_i cos[k_i(x-x_0) - \omega_i(t-t_0) + \delta_i] \tag{2.19}$$

where, (same notation as Equation 2.18) x_0 and t_0 are the focus location in space and time, which is the specific point where all wave components focus to produce the average shape of the highest wave of the considered survivability sea-state. At focus, the free-surface is symmetric, as shown by Figure 2.8. The wave-spectrum of the considered sea-state defines the wave-amplitudes a_i as:

$$a_i = A * \frac{S(f_i)\delta f}{\sum_{p=1}^{N} S(f_p)\delta f} \tag{2.20}$$

where $S(f_i)$ is the spectral amplitude of the i-th wave components; δ_f is the frequency spacing used in the spectral description; and, A is the amplitude of the focused wave

defined as:

$$A = \sqrt{2m_0 ln(N)} \tag{2.21}$$

where m_0, the zeroth moment of the spectrum S is defined as (Bouws et al. 1998):

$$m_0 = (H_s/4)^2 \tag{2.22}$$

2.2.3 Numerical Models for Survivability Assessment

A physical or numerical model evaluates the motion responses resulting from the selected extreme event, either an irregular sea-state or a design wave; the last step in Figure 2.6. The modelling of the motion responses results in the assessment of the survivability load.

The use of a physical model is of debate compared to numerical modelling due to the expenses linked to laboratory testing (e.g. availability, renting, material, electrical power - Vyzikas et al. (2014 - b)). However, there is still a need of the two, because the use of a numerical model not previously validated against experimental (or on-site) data is highly questionable. Besides, physical modelling remains fundamental in the development and design optimisation of WEC (CCP-WSI Working Group 2016). Therefore, the following focuses uniquely on the use of numerical models for survivability assessment.

From Navier-Stokes Equations to Numerical Models

For each NS equation system presented in sections 2.1.1 and 2.1.2 there exist a numerical model capable of solving the system. As previously presented in Figure 1.1 in the Introduction, CFD gathers solvers capable of solving the NS equations with minors simplifications using temporal and spatial discretisation methods (later detailed in section 3.1) also called scheme. The spatial discretisation is either supported using a mesh made of multiple not-overlapping cells occupying the whole domain where the flow evolves (Eulerian method); or using a particles (called meshless, sometimes considered outside CFD) representing the flow itself (Lagrangian method). Outside CFD,

the NS equations are simplified to the Laplace Equation (Equation 2.10). Solvers are 'potential flow based' and do not require the spatial discretisation inherent to CFD.

The discretisation methods link mathematically a discrete field at a point in space (or time) by the contribution of its surroundings. The method discretises the NS equations or a simplified system (e.g. Euler, Laplace). Two different methods applied to a same equation system are expected to produce the same solution provided a sufficiently fine resolution. The difference is in the computational effort.

The Finite Difference Method (FDM) assumes a polynomial relation between neighbouring cells, such as a Taylor series or a polynomial. Development made by Yabe and Aoki (1991) to solved hyperbolic equations create the Constrained Interpolation Profile (CIP) method allowing third order accuracy in both space and time. Hu and Kashiwagi (2004) applied the CIP method to model violent free-surface flows. However conceptually simple and easily programmed, the FDM is restricted to simple geometries since their complexity increase significantly for moving boundaries or unstructured grid.

In the Finite Element Method (FEM) the mesh is made of elements and nodes, which respectively are the side of the cell and the angle. Elements contain physical properties, while nodes store the position in space and forces. FEM assumes that fields follow piecewise function (e.g. shape function) over an element, hence allowing interpolation over the entire domain. The main advantage of FEM compare to FDM is the capacity to cope with arbitrary geometries, unstructured grid and mesh refinements Ransley (2015). However, FEM is relatively unpopular in hydrodynamic problems (compare to its popularity in mechanical problems) due to the non-symmetry (i.e. non self-adjoint) of the equations leading to oscillatory solutions. The CV-FEM is another development based on the FEM. Elements are replaced by discrete Control Volume (CV). Piecewise functions are used to calculate the change of variable across the CV Westphalen et al. (2009). Westphalen et al. (2009) used it successfully to model extreme wave loading on a floating WEC by solving the RANS equations.

The numerical method commonly used in hydrodynamics is the Finite Volume Method

(FVM). It has the advantage of being by nature conservative due to the discretisation of the domain into CV requiring to integrate the NS equations over the CV surface and volume (later described in details in section 3.1.1). Therefore most CFD code, such as OpenFOAM - the one used in the present study-, used this method. However, the integration over the CV requires additional approximation to calculate fields at cell centre and faces, hence making FVM often more computationally demanding Ransley (2015). As other method, the FVM can be applied to discretise NS equations of different level of simplifications. The AMAZON code (Qian et al. 2006) solves the integral form of Euler equations for incompressible flow with variable density. The OpenFOAM code solves the RANS equations for an incompressible flow.

Meshless solvers (e.g. Smoothed Particle Hydrodynamics (SPH), Moving Particle Simulation) have specific schemes necessary to solve the NS equations expressed in Lagrangian frame, which also omit to deal with the complexity of the convective term. However, particles only interact with surrounding particles, which needs to be define by a kernel function. Recent developments show significant increase in efficiency and accuracy of these meshless solvers in Fluid Structure Interaction (FSI). Verbrugghe et al. (2018) use the DualSPHysics SPH code to model non-linear WSI around floating WEC. Additionally, single phase models can easily be implemented by defining uniquely water particles. The complexity of the free surface identification required in two phase mesh based CFD models is avoided. Highly distorted surface, such as splashing waves, are better reproduced (Henry et al. 2013). However, the computational effort is often even greater than mesh based CFD models.

Potential flow based solvers solve the Laplace Equation (Equation 2.10) for an irrotational flow using either a fully non-linear (such as for the NS equations), or using a second order; or using a linear approach. Fully non-linear potential flow solvers model the flow in a discretise domain, i.e. a mesh. Different numerical method are available for the discretisation. Yan (2006) (QALE-FEM) uses the FEM for the discretisation, while Engsig-Karup et al. (2016) use the spectral element method (Galerkin method).

Such solvers resemble NS based CFD ones in the setup but the flow properties differ (i.e. inviscid, irrotational) and a single-phase is considered.

Linear potential flow solvers calculate Response Amplitude Operator (RAO) from a spectral solution of Laplace Equation. The Boundary Element Method (BEM) (Delhommeau 1987) is probably the most common method and allows to discretise the structure wetted surface into panels allowing to deal with complex geometries. It is the method implemented into WAMIT or NEMOH potential flow codes. The numerical effort is insignificant relative to CFD code. Some numerical models, such as WaveDyn or WEC-SIM, use the input of these linear potential flow solver to compute hydrodynamics loads and structure response in the time-domain (later described in section 4.1).

Selecting the correct model

The selection of a numerical model is the compromise between the desired level of physics (i.e. assumptions on NS equations) with the affordable computational cost. As shown in section 2.1.2, assumptions made on the fluid properties (i.e. incompressible, inviscid, irrotational) allows simplifying the NS equations. Each simplification induces a decrease in the accuracy (of the mathematical model) while allowing a decrease in computational cost of the solving method. Solving the RANS equations (2.7 and 2.8) requires the use of CFD (later described in section 3.1), but models an incompressible flow. Potential flow theory (Laplace Equation 2.1.2) can be solved in the frequency domain, which inputs can then be used in a time-domain description (later described in section 4.1) allowing fast computations but for an inviscid and irrotational flow. The use of low/mid fidelity model is driven by the efficiency of the simulation, while CFD focuses on accuracy.

No clear consensus appears on the appropriate numerical model for survivability assessment. In the review on modelling WEC in extreme sea-states, Coe and Neary (2014) justifies the use of mid-fidelity models up to the point of breaking wave, as results give an order of magnitude for a fraction of the computational cost. Mid-fidelity models reproduce well heave responses but show inaccuracies in surge an pitch (Heil-

skov et al. 2015), which allows these models to show an efficient estimation of mooring loads as mainly driven by heave. On the other hand, it is assumed that extreme event generates highly non-linear effects, generally meaning an increase in velocities, amplitudes, and forces (Penalba et al. 2017). To capture correctly these non-linear hydrodynamic effects, the modelling of extreme WSI commonly justifies the use of CFD (e.g. Westphalen et al. (2009), Hu et al. (2016), Henry et al. (2013), Ransley (2015)). Therefore, the consensus appears mainly related to the acceptable level accuracy of the solution and highlights the compromise between speed and accuracy as a function of the physics included.

The selection of the numerical model can be done as a function of the capacity of the numerical model the represent of physical phenomena. In regular and irregular waves, Chen et al. (2017) finds that the difference in responses between a mid-fidelity solver and a CFD increases with the wave amplitude for a point-absorber WEC, which is related to the incapacity of the mid-fidelity model to account for over-topping waves. For an Oscillating Wave Surge Converter (OWSC) WEC, Henry et al. (2013) found experimentally that the impacts of an extreme event results in a important distortion of the Mean Water Line (MWL) at the flap, which requires simulation of the flow to account for such a behaviour and modelling the impact of the wave (i.e. slamming event) itself. So, in the case of an extreme event including complex free-surface phenomenon, the use of higher-order models like CFD appears justified and mandatory.

To obtain the highest accuracy in motion response, the use of CFD is widely considered legitimate in the modelling of extreme WSI because lower fidelity numerical models are pushed beyond their limit due to physical assumptions. For a point-absorber WEC in a regular background where is embedded an extreme event, Sjökvist et al. (2017) validates against experiments the significant differences of responses found by two CFD models compare to an in-house developed linear model. Before the extreme event, the linear model is found to succeed in both heave and surge motion. However, at the event, the surge motion is not captured (opposite direction). Rafiee and Fiévez (2015)

compares to experiments in moderate wave conditions a mid-fidelity solver which drag coefficients were estimated from a previously run CFD simulation. The study identifies the mid-fidelity model as over-estimating motion responses (twice amplitude) for the point-absorber. CFD captures accurately surge and heave motion responses but shows more differences in pitch. Therefore, only the CFD model is used in the study on extreme events as the limit of the mid-fidelity model is attained. Therefore, the overall accuracy of mid-fidelity solvers in extreme events, even without complex free-surface flows (e.g. breaking waves, over-topping), remains low, which justifies the use of higher-order models.

However, the development of a CFD model is not trivial and the confidence in these high-fidelity models is often reassessed. CFD are used for the modelling of very challenging WSI, making the level of agreement subjective. The 'good agreement' for surge and heave responses for a point-absorber obtained by Sjökvist et al. (2017) for the two CFD models shows a decrease in accuracy starting at the main crest of the extreme event. Henry et al. (2013) found that the distortion of the free-surface, and hence the resulting motion response and slamming load, is better captured by the SPH (i.e mesh-less CFD) than the CFD. It shows uncertainties on the capacity of CFD models to accurately capture such complex WSI. Besides, due to its large computational requirement, CFD is restricted to short single design-wave, which shown uncertainties. Rafiee and Fiévez (2015) reproduces in CFD the design-wave identified by an experimental irregular sea-state using a NewWave. Resulting response motions in all Degree of Freedom (DoF)s (surge, heave, pitch) are under-estimated compare to the experiment, hence under-estimating the survivability assessment.

Additionally, the use of higher-order models like CFD suffers from excessive execution time (CCP-WSI Working Group 2016). The assessment of survivability via the use of multiple sea-states, as recommended by IEC TS 62600-2 (International Electrotechnical Commission 2016), makes the use of CFD restricted to short design-wave of questionable validity (CCP-WSI Working Group 2016). Also, the modelling of long

irregular sea-state is simply unrealistic in CFD, making the use of lower fidelity models mandatory despite the level of uncertainties.

Therefore, the coupling strategy develops by the present study aims to tackle these different problems. By assessing parametrically the limit of validity of WaveDyn (mid-fidelity) and OpenFOAM (CFD), the selection of the correct numerical model is expected not to be only driven by the affordable computational effort. Indeed, even if computational power would be available, the use of CFD below the mid-fidelity model's limit seems unnecessary as the lower model remains valid. Also, the parametric limit selects the 'most-appropriate' model for the assessment of extreme motion responses. Finally, the coupled model aims to make affordable long irregular sea-state simulations to the level of accuracy of CFD. The confidence in the use of CFD is assessed against the physical experiment of the X-MED buoy under four extreme design-wave (detailed in section 2.4).

2.3 Coupling in Wave-Structure Interaction

Numerical models across WEC design, both survivability and reliability, are significantly less sophisticated than required by developers (CCP-WSI Working Group 2016). Fully non-linear CFD models used in survivability assessment usually dissociate parts, like PTO or mooring lines, or make assumptions on that part for simplification. Coupling of numerical models (also called hybrid approach) refers to the use of two (or more) models simultaneously to benefit from each feature. It has been recently given much attention due to its potential of including complex parts description or developing more computationally efficient codes. Therefore, by coupling WaveDyn with OpenFOAM; OpenFOAM could benefit from a complex description of parts via WaveDyn; while, the coupled model would be computationally more efficient than CFD by maximizing the use of WaveDyn up until its validity limit. The present section gives an overview of other coupling strategies developed for WEC to discuss the pros and cons of each compare to the time-coupling strategy developed in the present study.

Coupling between models is qualified as weak compared to strong, and loose com-

pared to tight. When one model runs before the other in a unidirectional exchange of information, the coupling is weak (Verbrugghe et al. 2018). This strategy initialises one model by outputs from the other. When the exchange of information is two-ways, the coupling is strong. A strongly coupled model runs each models simultaneously. It benefits from each model feature during the whole simulation where models exchange information. Tight-coupling qualifies a strongly coupled model where the two models converge to the same time-step before advancing to next (Dunbar et al. 2015). In loose coupling, the time-step of a model is not imposed on the other. Dunbar et al. (2015) improve the stability of loose coupling between the Rigid Body Motion (RBM) and fluid (CFD) in the `interDyMFoam` OpenFOAM solver. Tight coupling is considered more robust than a loose coupling which, however, benefits from the independence between models. Tight coupling is not always achievable due to the difference between time step range, and because imposing the time-step might induce instabilities within one model. Indeed, for WSI modelling in CFD, the Courant number is required to define the time-step (discussed in section 3.1.2). Palm et al. (2016) interpolate the rigid-body position calculated using CFD before feeding it to the mooring model, hence benefiting from the flexibility of the loose coupling strategy.

The present study develops a loosely coupled model, which appears in-between weak and strong. The flexibility of the loose coupling assures that the Courant number governs the time-step in CFD. As Palm et al. (2016), WaveDyn requires interpolation between two following its time-step to exchange information with OpenFOAM. The coupled model uses WaveDyn only until the WSI requires the use of CFD. So, the coupled model is weak, as WaveDyn initialises OpenFOAM. However, when CFD is required, the two models communicate, and the coupling became strong. So the coupled model developed in the present study alternates between a weak and a strong-loose coupling.

Coupling strategies for the improvement of the WSI modelling of WEC can be separated into three categories. The toolbox technique (or function-splitting) aims to add an improved description of a part (e.g. mooring, PTO). The zonal approach (or space-

splitting) divides the space into domains each ruled by a different model. The time-coupling technique (or time-splitting), which is the use developed in this study, aims to swap between model for a specific period. The advantage of a strategy becomes, as a consequence, the drawback of the other.

2.3.1 Toolbox or Function-Splitting Strategy

The toolbox coupling strategy connects a model capable of a more precise description of a part of a WEC to a model modelling the WSI. The first model is therefore used as a tool for the second, the main. The more precise description of given parts aims to improve the overall modelling compare the main model alone.

A common example for WEC is the modelling of complex mooring like catenary lines via an additional solver coupled with a main responsible for the modelling of WSI. Palm et al. (2016) use the mooring software MooDy (in-house) to describe catenary lines, where the modelling of the RBM and flow are performed using CFD in OpenFOAM. A similar development is done by de Lataillade et al. (2017) between the CFD solver Proteus (2019) and the multi-body dynamics ProjectCHRONO (2019). These two research clearly improves the modelling of WEC, which could be applied to survivability assessment as required from developers (CCP-WSI Working Group 2016).

However, the improvement in accuracy is at the price of computational power due to the communication between codes. WaveDyn could describe in more details parts like the PTO system. The application of the present study remains the modelling of the X-MED motion responses under extreme events. As the buoy has no PTO or parts justifying additional toolbox for their description, this coupling strategy is excluded as not the priority. The implementation of more details description of PTO from WaveDyn into OpenFOAM is left of future work.

2.3.2 Zonal or Space-Splitting Strategy

The spatial coupling strategy separates the domain into sub-domains, each governed by one model. Typically the separation is a function of the complexity of the fluid dy-

namics. The area around a structure requires a high fidelity model to capture complex WSI (e.g. include viscous effects, slamming event, green water). On the other hand, the fluid dynamics of the far field away from the structure is of lower complexity and, especially, of minor interest. Therefore, the spatial coupling strategy is particularly suitable to CFD code with a lower fidelity model governing the far field. Coupling reduces the computational effort to make CFD based model affordable for industries (Kim et al. 2016), and capable to simulate more complex cases or larger domains.

Intensive development has been conducted in the past years to couple high-fidelity models to one of lower-fidelity in order to decrease the computational effort. Coupling was either weak or strong depending on the the degree of development of the coupling and on the interest of a two-ways exchange.

For weak coupling, the sub-domain ruled by the solver of lower fidelity inputs its solution (or part of it) into the sub-domain ruled by the higher-fidelity solver; while no information is exchanged from high to low fidelity model. Li et al. (2018), Hildebrandt et al. (2013), Lachaume et al. (2003) and Biausser et al. (2004) weakly coupled a two-phase NS-Volume of Fluid (VoF) model with a fully nonlinear potential theory (FNPT) based model solved using the FEM (QALE-FEM) and the BEM respectively. In these studies, the NS model uses the input from the FNPT based model. Hildebrandt et al. (2013) justifies the use of the weak coupling since no reflection is coming from the structure located in the NS sub-domain before the extreme event of interest reaches it. Hence, the 3-dimensions ($3D$) simulation made in the NS solver is limited to immediately before the impact and the impact itself. The 2-dimensional ($2D$) FNPT solver simulates the waves, which velocity and pressures at a specific location are used as boundary condition for the NS based model. In Janssen et al. (n.d.) the mesh based NS model is replaced by a particle-based meshless CFD model (Lattice Boltzmann), where the weakly coupled scheme is used as validation before a strongly coupled two-ways development.

Strongly coupled zonal models exchange information between models in a two-directional way. Strong coupling is often require for long simulation since sub-domains will affect

each other Sriram et al. (2014). A relaxation-zone (or sponge layer) or an interface provides the connection between sub-domains. Lachaume et al. (2003) develop further the weakly coupled into a strongly coupled model by creating a region where both FNPT and NS-VoF sub-domains overlap. Kim et al. (2010) uses the relaxation-zone technique between a BEM-FNPT sub-domain and a NS-VoF originally developed by YAN et al. (2003) for a similar two-ways coupling. In the relaxation-zone, the surface-elevation from the BEM-FNPT is forced to meet the one in the NS-VoF boundary. The two-ways coupling demonstrate bidirectional wave propagation. Sriram et al. (2014) strongly coupled a FEM-FNPT with a meshless NS based solver (Imrpoved Meshless Local Petrov Galerkin method with Rankine source, IMLPG_R). A moving overlapping zone connects the two sub-domains since fixed or moving interfaces are identified as only adapted for mesh based approach; and a fixed overlapping zone induces the addition or suppression of particles at every time steps (problem of conservation of mass). The moving overlapping zone maintains highly nonlinear physical phenomenon, such as over-turning or wave breaking, inside the meshless region.

In the WEC sector, the zonal approach was used by Verbrugghe et al. (2018) (among others) for the simulation of arrays. The SPH (mesh-less CFD) solver DualPhysics (Crespo et al. 2015) is only used on a cylindrical area surrounding of each device, while the fully non-linear potential flow model governs the rest of the domain, OceanWave3D (Engsig-Karup et al. 2009). So, DualPhysics assures the accurate description of the WSI, while OceanWave3D defines the wave-generation, absorption, and travel to a lower but chosen level of accuracy. An interface assures the transition between solvers domain (i.e. buffered layer), proven capable of transferring radiated waves between domain. The SPH domain being limited around devices, the computational saving is significant (from a quarter up to a half), which make arrays simulation affordable.

A drawback of the strongly coupled zonal strategy is that the time-consuming model (i.e. CFD) is used during the whole simulation independently from the complexity of the event, even in cases where such fidelity is not required. Therefore, there is also a

period (as there is an area) where WSI requires such a level of accuracy, and, outside this period lower fidelity models are accurate enough.

2.3.3 Time-Splitting Strategy

From this statement appears the third coupling strategy, the time-coupling; used in the present study. The principle is to reduce the usage of CFD (i.e. OpenFOAM) to the precise WSI instants identified as out of validity range for the mid-fidelity model (i.e. WaveDyn). Outside of these times, only the mid-fidelity model is used. At these times, the coupled model changes to a WSI modelled by OpenFOAM starting from the current RBM. Ideally, when CFD is no longer required, the mid-fidelity model regains ownership. This strategy benefits from an increase in the level of physics over a required period to increase the accuracy (compare to mid-fidelity only). The reduction of the use of CFD induces a reduction in computational power. However, the drawbacks are the identification of the range of use the mid-fidelity model, and the implementation of one model's solution into the other.

A major interest in this coupling is the potential for modelling long irregular sea-state at the level of accuracy of CFD. Indeed, the validity of WSI using short design-wave commonly model in CFD is argued due to the absence in motion history and loads induced by the use of these design waves (discussed in section 2.2.2). Therefore, the coupled model is expected to model the large motion response resulting from these events, in which CFD will benefit from the history of motion and load calculated by the mid-fidelity model. The development of the coupling between WaveDyn and OpenFOAM is detailed in Chapter 5, where the identification of the range of use of the mid-fidelity model WaveDyn is conducted in Chapter 4.

This coupling strategy is similar to the one developed by Wang et al. (2016). In the study, three wave models (QSBI, ESBI, ENLSE-5F) for large temporal and spatial scale simulations are used as a function of the instantaneous wave information. Ranked by order of validity assumed from underlying equations, which corresponds to their efficiency (computational effort), the coupled model will switch from one model to the

other. An estimation of the error of the modelled surface-elevation at each time-step triggers which model to use. The efficiency of the coupled model compared to the model of high fidelity depends on the wave spectrum used but ranges from half the computational effort to a tenth.

2.4 The Reference Model

This section is a presentation of the physical experiment (Hann et al. 2018) used as the reference in the assessment of both numerical models validity. The physical wave-tank is University of Plymouth Ocean Basin COAST, Figure 2.9. The X-MED buoy represents a simple point-absorber type WEC, Figure 2.10. Four extreme events of increasing steepness are used to assess the influence of steepness on the numerical modelling of motion responses.

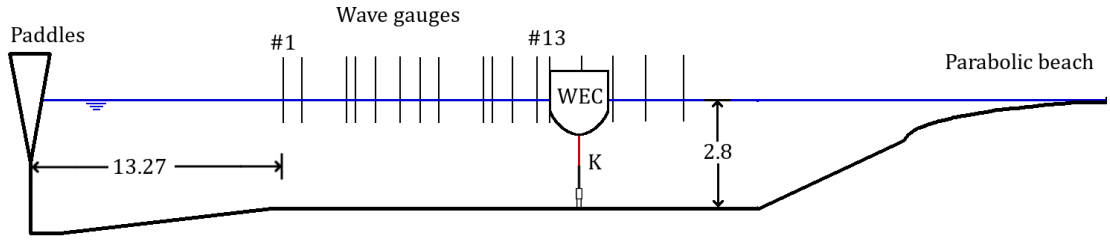


Figure 2.9: Physical Experiment set-up - based on Hann et al. (2015), Ransley (2015)

2.4.1 Physical Wave-Tank

The Physical Wave Tank (PWT) (Figure 2.9) is $35\,m$ long by $15.5\,m$ wide, the variable floor was fixed to $2.8\,m$. 24 individually controlled hinged flaps are used to generate the sea-state of interest, and the parabolic beach at the tank end minimises wave reflections.

Wave Gauge (WG)#	1	2	3	4	5	6	7	8
x in PWT [m]	13.27	14.32	14.77	15.23	15.7	16.11	16.54	16.82
WG#	9	10	11	12	**13**	14	15	16
x in PWT [m]	17.14	17.51	17.84	18.21	**18.54**	18.86	19.23	19.60

Table 2.1: Experiments Waves Gauges positions in the PWT - x is along the tank length and the wave direction - WG#13 is located at the front edge of the model, WG#14 almost at model centre

13 WGs were set-up along the tank length to measure the surface elevation, and their positions relative to the wave-paddles are reported in Table 2.1. The buoy front edge is located at WG#13, which was removed during the tests conducted with the model, along with WG#$14-15-16$.

2.4.2 Floating Structure

The structure model is a buoy which was designed as part of the EPSRC project on Extreme loading of marine energy devices due to waves, current, flotsam and mammal impacts (X-MED). The X-MED is a simple model (Figure 2.10) for the study of a generic point absorber type WEC under extreme events. A schematic representation of the X-MED buoy with its main characteristic is available in Figure 2.10, and table 2.2.

The geometry is a $0.5m$ diameter hemisphere with a $0.25m$ high cylinder of the same diameter on the top. A $2mm$ thick steel lid closes the whole, which weighs $43.2kg$. The buoy is moored by a single mooring line fixed to the sea-bed via a universal joint to ensure rotations in all DoF. The mooring line is a succession of a $1.38m$ rope (fixed to the bottom of the buoy), with a $0.9m$ spring fixed to the universal joint. The resonance frequencies of the model obtained using decay tests are available in Table 2.3.

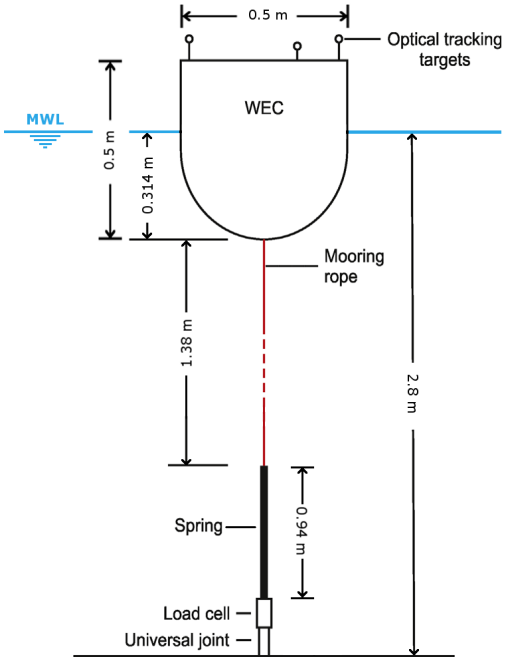


Figure 2.10: X-MED moored Hann et al. (2018)

Dry mass	$43.2\,kg$
Inertia	$Ixx = 1.61\,kg/m^2$ $Iyy = 1.61\,kg/m^2$ $Izz = 1.25\,kg/m^2$
Rope Spring	$35\,N/mm$
Linear Spring	$0.064\,N/mm$
Free spring length	$0.63\,m$
Mooring pre-load	$-20.5\,N$
Unmoored G	$-0.122\,m$
Moored G	$-0.133\,m$

Table 2.2: X-MED Model main characteristics

The mooring line is constantly under tension. The stiffness of the rope been over 500 times larger than the one of the spring, the elasticity of the rope is negligible. The centre of gravity of the moored buoy is $0.011\,m$ below the unmoored one. At rest, the force in the mooring line has $20.5\,N$. The pre-tension in the mooring line assures that the mooring line will stay in tension during the experiments.

Optical tracking targets are positioned on the top of the buoy and record its 6-DoF using Qualisys Oqus $300+$ high-speed cameras. The data are then translated to the centre of gravity, G. The maximal residual of the measure is $\pm 0.5\,mm$, and the sampling frequency is $128\,Hz$.

Decay test	No-mooring	With mooring line
Heave	0.926	0.917
Surge	not measured	0.07
Pitch	not measured	0.75

Table 2.3: Measured resonance frequencies in Hz from decay tests data

2.4.3 Extreme Events

Four focus events of increasing steepness model survivability events for the buoy. The extreme sea-state is a $3h$ data set from a 100-year storm hindcast at Wave Hub ($T_p = 14.1s$, $H_s = 14.4m$, Halcrow (2006)). Using a Pierson-Moskowitz wave-spectrum (Equation 2.18), the first event - least steep one named $ST1$ - is defined from NewWave theory (2.2.2). As the study is conducted at a $\frac{1}{50}$ scale, the theoretical crest amplitude of the NewWave event at focus is $0.268m$ (Equations 2.21 and 2.22). Theoretical wave steepness is defined as the product between the wave number k of the peak frequency, and the crest amplitude; $kA = 0.15$.

This wave-group is used to generate three steeper cases ($ST2$, $ST3$, and $ST4$) by multiplying the peak frequency by particular factors to obtain desire theoretical steepness values (Hann et al. 2015). The values used for the experiments are available in Table 2.4. They were obtained using the dispersion relation previously defined by Equation 2.12. It should be noted that these steeper wave-groups do not follow NewWave theory any longer, even if the overall shape of the surface elevation is similar. However, it does not remove their applicability as design-waves for survivability assessment.

This method is used to try to maintain the crest amplitude while increasing the steepness (Hann et al. 2015). A second method consists of increasing the crest amplitude while fixing the peak frequency. However, it induces a larger crest amplitude that generates a lager heave motion, so a larger mooring load, and hence reducing the assessment on steepness.

Due to the non-linearity of the wave-groups, from the least to the steepest case, the experimental focus location differs from the theoretical one. A trial and error procedure, described and used by Ning et al. (2009), is required so that the wave focuses at the front edge of the model (i.e. WG#13). The focus is considered achieved once the troughs, previous and next to the main crest, are symmetric. The trial and error process consists of adjusting the phase components of the wave-group until focus is achieved. The precision obtained by this method is $+/-10cm$.

Case	Peak frequency $[Hz]$	Theoretical kA	Measured kA	Measured A $[m]$
$ST1$	0.356	0.15	0.152	0.273
$ST2$	0.388	0.17	0.179	0.280
$ST3$	0.420	0.2	0.209	0.286
$ST4$	0.449	0.22	0.263	0.318

Table 2.4: Theoretical and measured characteristic for the four focus wave events

The measured characteristics are reported in Table 2.4, and the shape of the four wave-groups are plotted in Figure 2.11. The measured crest amplitude is the maximum of the surface elevation, which is then used to calculate the measured steepness considering the peak frequencies (measure from spectrum) as unchanged. Figure 2.11 shows that the amplitude increases with steepness. Hann et al. (2015) describe these as the effect of non-linear wave-wave interactions increasing with steepness as previously suggested by Ning et al. (2009). More informations on the accuracy of the experiments is available in Hann et al. (2015).

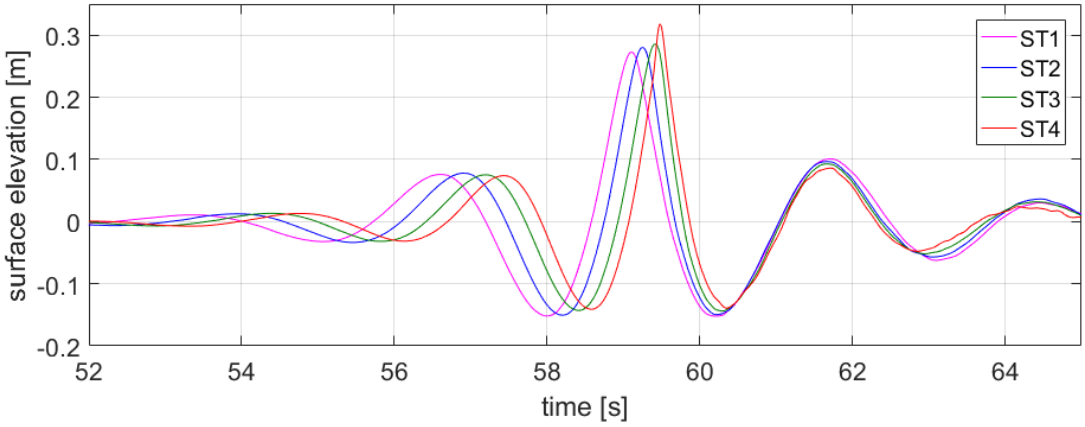


Figure 2.11: Surface elevations for the four extreme events at the focus location; i.e. the front edge of the model, WG#13

Chapter 3

A Numerical Wave Tank for Extreme Wave Structure Interaction

This Chapter presents the development of a Computational Fluid Dynamics (CFD)-Numerical Wave Tank (NWT) able to model the Wave Structure Interaction (WSI) of a Wave Energy Converter (WEC) device under extreme events to assess its survivability. The tool is validated against the physical extreme events experiments detailed in section 2.4, i.e. a single moored buoy under four extreme events of increasing steepness.

The NWT solves the Reynolds Averaged Navier-Stokes (RANS) equations for an isotropic, Newtonian, incompressible, two-phase flow using the Finite Volume Method (FVM) and the Volume of Fluid (VoF) method for the interface capturing scheme, as detailed in section 3.1.

After presenting the CFD solver is developed a 2-dimensional ($2D$)-NWT to assess only the modelling of the wave for the four extreme events. The number of cells per wave height (CPWH) and the wave absorption are identified as decisive (the first objective stated in Chapter 1) and are evaluated compare to a NWT of known greater accuracy. The least steep event is used since it is the least nonlinear case requiring a finer vertical resolution to minimize the numerical diffusion. In addition it is the case with least nonlinear physical phenomenon. Then, with no change in the NWT, the wave-only simulations of the four focus events are validated against the experiment.

Second, the NWT is expanded to 3-dimensions ($3D$) to include the rigid-body. The resonance frequencies of the X-MED buoy are validated against decay tests to assess

the Rigid Body Motion (RBM) and mesh-deformation.

Finally, the WSI is applied to $3D$ where is identified, and assessed, the potential requirement for a limiter of flow velocity to converge to $2D$ wave-only solutions. The width of the NWT is defined from a convergence over the 6-Degree of Freedom (DoF) responses of the buoy under the least steep event, and the NWT is then validated against experiments (second objective).

The methodology to develop NWT aims to apply to other Physical Wave Tank (PWT) experiments. The present NWT aims to be capable of reproducing other experiments done at the same PWT without changes (or minor ones), as it is developed using the least steep case but validated over the four events of increasing steepness. The resulting tool is used in future coupling specifically for the assessment of extreme WSI.

During the following, keywords referring to OpenFOAM C++ Source Code v4.1 (libraries, utilities, solvers...) and additional toolboxes are written using the Courier font: `relaxationZone`.

3.1 CFD Based Wave-Structure-Interaction using OpenFOAM

The second model used in the present study is the open-source CFD code OpenFOAM. The accessibility of the code for potential development offered by the open-source licence is a decisive advantage compared to commercial codes, such as ANSYS-FLUENT or STAR-CCM+. OpenFOAM is a set of C++ libraries (e.g. solvers, schemes, boundary conditions) accessible in the source codes OpenFOAM C++ Source Code v4.1 or OpenFOAM ESI. The code benefits from a community of users involved in the development of libraries or tutorials across a wide variety of applications. The present study benefits from the unlimited assess to High Performance Computing (HPC) of the University of Plymouth. In the following, names written with `typewriter` font refers to the corresponding OpenFOAM library.

OpenFOAM has been validated against experiment in the modelling of extreme WSI for point-absorber WEC. Rafiee et al. (2016) and Ransley (2015) (among others) replicate

the experiment of extreme focus events, showing good agreement of the event and the resulting motion responses (surge, heave and pitch). Ransley (2015) explains the accuracy of the motion due to the good replication of the event itself in the first place. The major difference appears after the main event (i.e. focus) and is considered as related to wave reflecting at the end of the domain. However, the error in the surface elevation at that time remains less significant than the error in the motion responses. So, either some improvements remain in the overall capture of motions response, or the replication of the event is not that accurate as the surface elevation does not uniquely define it. A more precise description of the event at the inlet might be necessary (e.g. use of second order NewWave). Rafiee et al. (2016) successfully replicates the experiment motion responses and loads resulting from the design-wave previously extracted from experimental irregular sea-state. Sjökvist et al. (2017) replicates motion responses of a moored floating point-absorber under a focus event embedded in a regular background. The overall agreement in both heave and surge remain good, but discrepancies appear at the main event inducing an under-estimation in surge motion with an over-estimation in heave. Similar to Ransley (2015), the good accuracy in replication of the surface elevation probably indicates that the error lies within the capture of the rigid-body-motion or in the description of the event. These different research, among others, confirm the validity of the CFD code OpenFOAM for extreme WSI, in which, however, potential improvement remain.

This section describes the required major steps made by the CFD code OpenFOAM in solving the RANS equations for an isotropic, Newtonian, incompressible two-phase flow using the FVM and the VoF surface capturing scheme. The overall CFD procedure is illustrated in Figure 3.1, and the specific FVM is detailed in Figure 3.2. Additional details on the equations and the solving procedure within OpenFOAM code can be found in Jasak (1996) and Rusche (2003). Anderson (1995) and Versteeg and Malalasekera (2007) provide more detail on the solving procedure in CFD independently from the code.

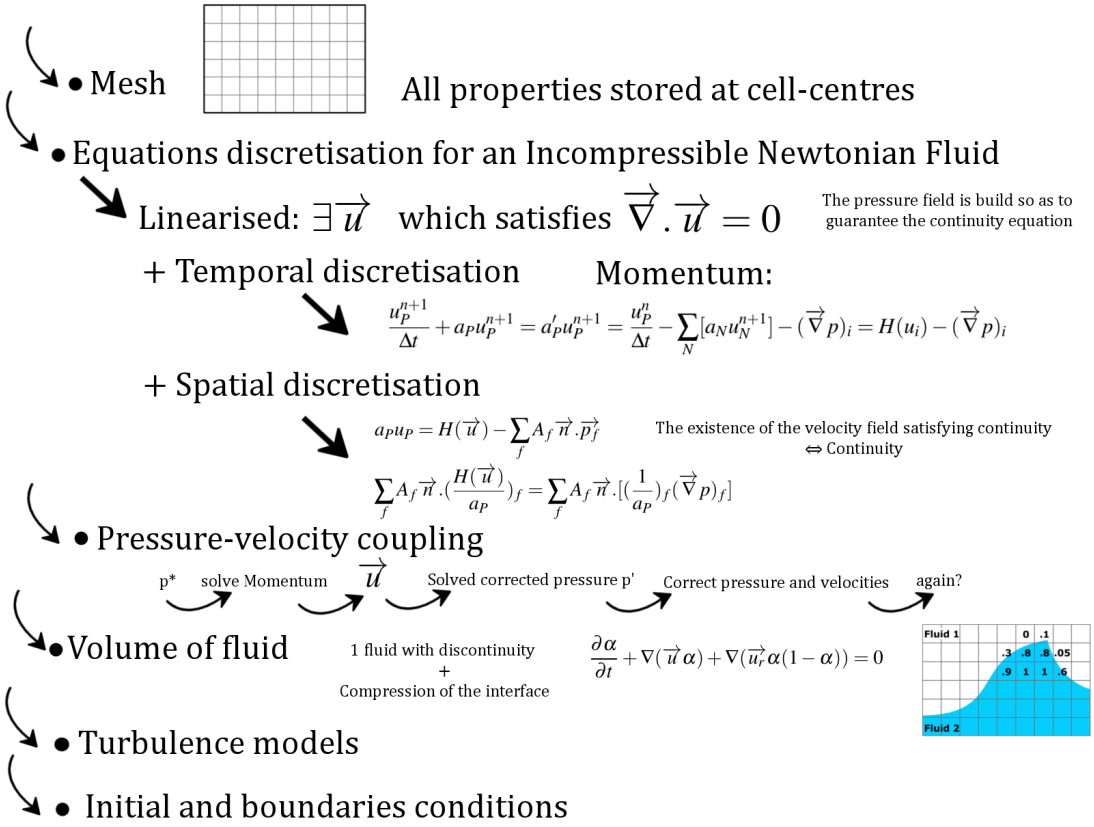


Figure 3.1: Steps required for the solving of the RANS Equations for an isotropic, Newtonian, incompressible two-phase flow based on the Finite Volume Method (FVM) and the Volume of Fluid (VoF) surface capturing scheme

3.1.1 Mesh and Finite Volume Method

CFD commonly refers to a mesh-based method, meaning that the domain where the fluid evolves is discretised into non-overlapping smaller volumes. Each volume is called a cell, and all the cells filling the domain constitutes the mesh. This first step of spatial discretisation is intrinsic to CFD and is the first step of the numerical method used in this study, the FVM. The FVM procedure implemented in OpenFOAM is reproduced in Figure 3.2, where the work of Jasak (1996) has been summarised, and adapted to this study notation.

OpenFOAM stores all the fields (velocity and pressure) at the cell centres. Using the staggered-grid method developed by Rhie and Chow (1983), it interpolates the velocity fields at the cell face. This interpolation step was proven required by Versteeg and

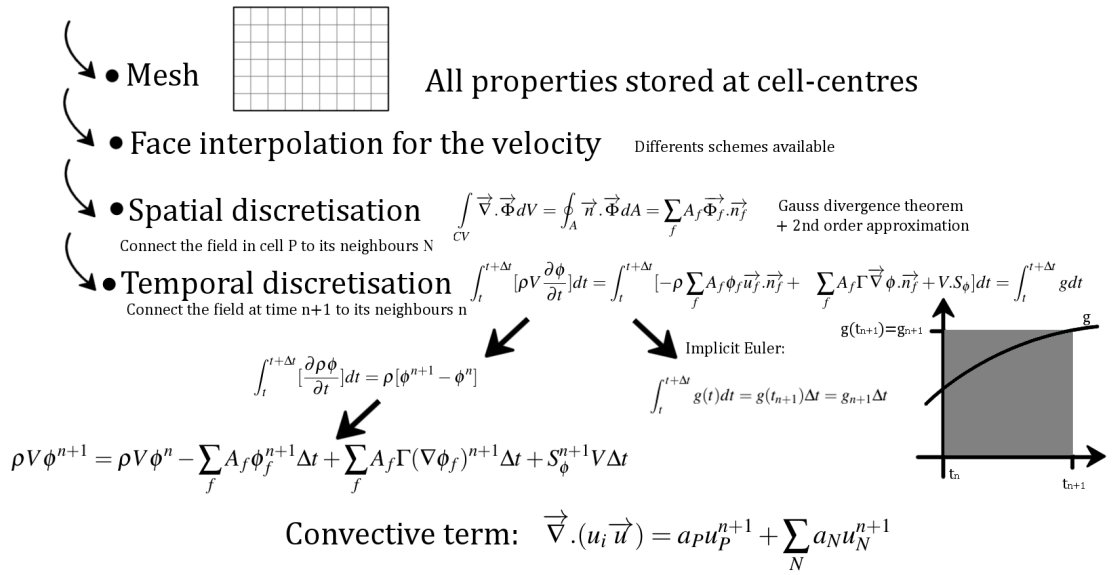


Figure 3.2: Illustration of the steps required for the Finite Volume Method (FVM) in OpenFOAM

Malalasekera (2007) to avoid highly non-uniform fields appearing from a uniform initial field due to the storage position of fields at cell centres.

Interpolation scheme

Interpolation of fields on cell faces (second step in Figure 3.2) uses different schemes defined depending on the neighbouring cells considered, the assumed evolution of the fields between cells, and grid. Figure 3.3 shows the considered cell P, its eastern neighbour E, in which both fields (Φ_P and Φ_E) and both position (x_P and x_E) are known. The face where the interpolation of fields Φ is required, is the east face, e at position x_e, hence field Φ_e.

The `interpolationSchemes` scheme used in this study is a linear interpolation and is specified in the `fvSchemes` dictionary (available in Appendix C.2). It uses the two closest cells centres from the considered face and assumes a linear variation between them. For face e, the cell centre P and E are used, with their respective grid coordinates, x_P and x_E:

$$\Phi_e = \lambda_e \Phi_E + (1-\lambda_e)\Phi_P \quad \text{where the linear interpolation factor} \quad \lambda_e = \frac{x_e - x_P}{x_E - x_P} \tag{3.1}$$

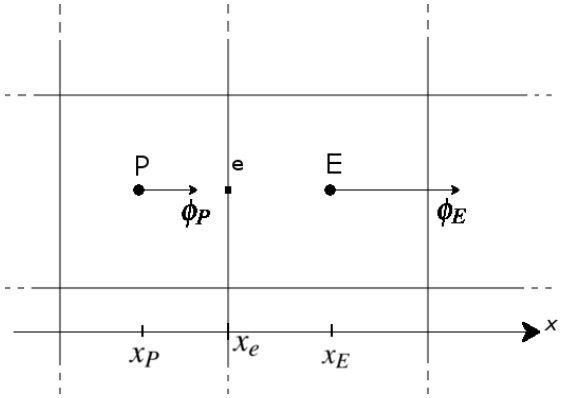


Figure 3.3: Interpolation of field Φ at cell face e (Φ_e) using its neighbours P and E

Other schemes of a different order (higher or lower) are available, and can be used to increase - respectively decrease - the accuracy for a higher - lower - computational cost or to increase the stability of a simulation. Higher orders schemes are defined using the further next neighbours (example: QUICK), i.e. the east cell of the east cell, to assume more complex variation. Or, simpler schemes assume a directional variation of the fields, like the Upwind Differencing Scheme, which estimates the field on the face e from the field at cell centre (P or E) depending on the direction of the flow.

Lower orders scheme were not found to be necessary for this study as stability was not an issue in the developed model (chapter 3). Otherwise, the Upwind scheme could have been used, because it is more stable as the fields are bounded, i.e. limited by boundary conditions values (Jasak 1996), which, however, would have potentially lead to a decreased of the accuracy of the solution.

On the other hand, the use of higher orders methods increases the computational cost for a supposed gain in accuracy. The order of magnitude of error obtained in this study did not legitimate the use of higher orders, because the major error in the representation of the event is considered due to the definition of the wave in the wave-generation model used (detailed further in section 3.1.5). Besides, the mesh used in this study will be refined in the area of higher complexity to justify the linear assumption between two neighbouring cells.

Spatial discretisation

The spatial discretisation is the third step in the FVM - Figure 3.2-, which requires to integrate the Navier-Stokes (NS) equations (Equations 2.4 and 2.5) over the volume of each cell. This integration is a specificity of the FVM over other discretisation technique (Finite Element or Finite Difference), which ensures that each volume and the overall domain satisfy the conservation of quantities (Versteeg and Malalasekera 2007).

Using Gauss divergence theorem, the integral over the volume of the divergence term of a given field Φ, is expressed as the sum of this field at each faces times the face normal vector, $A_f.\overrightarrow{n_f}$:

$$\int_{CV} \overrightarrow{\nabla}.\overrightarrow{\Phi}dV = \oint_A \overrightarrow{n}.\overrightarrow{\Phi}dA = \sum_f A_f \overrightarrow{\Phi_f}.\overrightarrow{n_f}. \tag{3.2}$$

Gauss divergence theorem is applied to the divergent terms in Equation 2.5, where the Laplacian term, $\overrightarrow{\nabla}^2(u_i) = \overrightarrow{\nabla}.\overrightarrow{u_i}$, induces the gradient of the velocity field on the cell face:

$$\int_{CV} \overrightarrow{\nabla}^2(u_i)dV = \sum_f A_f \overrightarrow{\overrightarrow{u_i}_f}.\overrightarrow{n_f}. \tag{3.3}$$

Interpolation schemes for the divergence and the Laplacian term are specified in the `fvSchemes` dictionary (Appendix C.2).

Temporal discretisation

In time-dependent problems, the transport equation needs to be integrated over a time interval; the time-step $\Delta t = t_{n+1} - t_n$ (step four in Figure 3.2). This final step connects the fields at the actual time t_n to the fields at time t_{n+1} using a temporal scheme. The difference with the interpolation scheme is that the time t_{n+1} is an unknown.

The temporal scheme used in this study is the implicit Euler, meaning that the integral over the time-step of a function $g(t)$ (representing the NS equations) is approximated by its value at t_{n+1} (i.e. $g(t_{n+1})$) multiply by the time-step.

From the spatial discretisation (Equation 3.2) and the implicit Euler, the NS equations for a given cell at time t_{n+1} are rewritten as a sum of the contributions of the field Φ at time t_n and t_{n+1}:

$$a_P \Phi_P^{n+1} + \sum_N a_N \Phi_N^{n+1} = R_P, \tag{3.4}$$

where a_P and a_N are specific coefficients depending on the discretisation schemes, that are detailed in Versteeg and Malalasekera (2007), and Anderson (1995).

3.1.2 Pressure-Velocity Coupling

The NS system (Equations 2.4 and 2.5) shows two issues. The first issue is the non-linearity in the convective term, $\overrightarrow{\nabla}.(u_i.\overrightarrow{u})$, which can contain quadratic terms. Second, is the fact that pressure and velocity cannot be considered separately, because the velocity is present in both momentum and continuity equations, and that no additional equation (like an equation of state) is available for the pressure. This is the pressure-velocity coupling, due to the incompressibility of the fluid, which states that density is constant and therefore not linked to the pressure.

To solve these issues, the common approach is to linearise the convective term by considering the existence of a velocity field, $\overrightarrow{u}$, which satisfies the Continuity equation (Equations 2.4), hence simplifying the Laplacian term from the momentum equations.

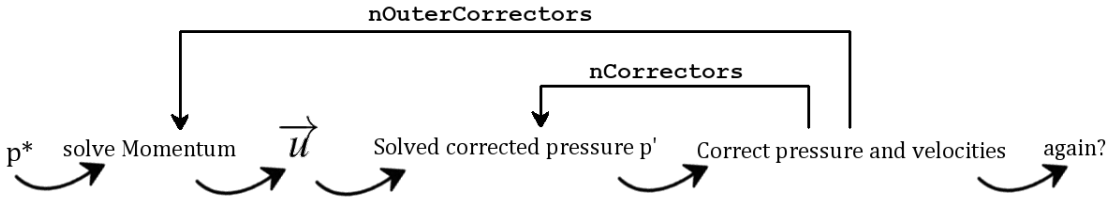


Figure 3.4: Pressure-Velocity Coupling PIMPLE algorithm as implemented in OpenFOAM - based on Versteeg and Malalasekera (2007) and OpenFOAM-WIKI (2019)

Then solving the system is commonly based on a pressure-velocity algorithm, which process follows the steps presented in Figure 3.4. It starts by a guess of the pres-

sure $p*$ (from previous time-step or initial state) allowing to calculate the convective term, then the velocity field, and then the corrected pressure (p') from this velocity field. The pressure and velocity are corrected, if required or specified multiple times (`nCorrectors` in Figure 3.4), from the difference between the original guess and the actual value. A second corrective loop (`nOuterCorrectors` in Figure 3.4) is included allowing the algorithm to restart from solving of the momentum using final values of pressure and velocity.

A second approach to the pressure-velocity coupling algorithm is the fraction step (or time-splitting) method developed by Chorin (1967). This method decouples the computations of the velocity and pressure field by splitting the momentum equation into different parts. This method is most commonly used with Finite Element Method (FEM) Wadhah et al. (2004). A comparative study between a pressure-velocity algorithm (SIMPLE) and the fraction step method on transient incompressible flows has been conducted by Hines (2008).

OpenFOAM uses the Pressure Implicit for Pressure Link Equations (PIMPLE) algorithm presented in Figure 3.4 or more details in Annexe A.1. It allows the user to specify the number of corrective iteration of the flow fields (`nCorrectors`); and a number of iteration for the overall loop (`nOuterCorectors`). The selection of these parameters depends on the application. The interest of PIMPLE is to use more than one overall corrective iteration to increase the time-steps and hence reduce the computational cost (OpenFOAM-WIKI 2019). However, in WSI, the Courant number commonly governs the time-step.

Courant-Friedrichs-Lewy Condition

The Courant-Friedrichs-Lewy (CFL), or Courant number, is the parameter, which ensures that the maximum fluid displacement between two time-step is smaller than the maximum displacement allowed by the mesh. It needs to be satisfied to assert numerical stability (Anderson 1995) and is defined for a given cell as:

$$Co = \frac{|u|\Delta_t}{\Delta_x} \leq 1 \tag{3.5}$$

where Co is the 'Courant number', $|u|$ is the norm of the fluid velocity, Δ_t is the time-step, and Δ_x is the distance between this cell centre and its neighbour in the direction of the velocity.

Therefore, the overall stability of the simulation is restrained by the maximum Courant number obtained across the entire mesh, hence for the highest velocity with the smallest cell-size. In unsteady cases, or if the velocity field is not predictable, it is common practice to adjust the time-step as a function of the Courant number so that the CFL condition is satisfied.

In WSI, Co can be limited to values lower than 1 to account for the surface elevation, which requires small time-steps to stabilize the simulation (The OpenFOAM Foundation 2019). Ransley et al. (2017) reproduces a NewWave extreme event in OpenFOAM using a Courant number restricted to 0.5 because it shows no difference in the surface elevation at the target location compare to a more restrictive value (0.25) for wave-only cases. This Courant-number value (0.5) is also widely used for WSI of WEC (Windt et al. 2018). Therefore, the present study uses a 0.5 Courant-number value because of the similarity of the application with Ransley et al. (2017) (i.e. solver, NewWave design-wave, floating object). Also, the PIMPLE algorithm is used with `nOuterCorectors`=1 and `nCorrectors`=2.

3.1.3 Free Surface

The modelling of the free surface requires specific attention because it is the interface between two fluids, air and water, and that it has been assumed the description of a unique fluid. A solution consists in modelling uniquely the water, and apply the preceding solving method. However, to capture the elevation of the free-surface (i.e. the wave), the domain needs to adapt accordingly. These moving mesh techniques are excluded because complex fluids flow, like breaking waves, are difficult to model

(Greaves 2004). Even if no breaking event is modelled in the present study, the CFD tool developed aims to be extended to more complex WSI involving breaking waves. So, the CFD domain requires the consideration of two fluids, air and water, where the free-surface lies in the interface. Therefore, an additional method is required for the implementation of the two fluids, and tracking of the free surface. The one chosen in this study is the VoF method as it is considered as simple but robust and reliable (Hirt and Nichols 1981). In order to avoids steep gradients due to hydrostatic effects, the pressure field, p, is commonly substituted by:

$$p_{rgh} = p - \rho_{water} g.x \tag{3.6}$$

where p is the static pressure, ρ is the density of the denser fluid, and x is the position vector.

Volume of Fluid

The Volume of Fluid (VoF) method considered a multiphase approach where the two fluids, air and water, are implicitly described. Considering both as immiscible, i.e. unable to mix, a single fluid with a step in density is stated. This is mathematically parametrized using a function over the physical properties of the two-phase flow defined, in the example of density, as:

$$\rho_{fluid} = \alpha\rho_{air} + (1-\alpha)\rho_{water} \qquad \text{where } \alpha \in [0;1] \tag{3.7}$$

where the multiphase fluid density ρ_{fluid} is linked to air and water densities, ρ_{air} and ρ_{water} respectively, using α, the volume fraction field. α is a non-dimensional advective field (meaning that it moves with the fluid in the manner of a substance) defined for each cell between 1, for water only cells, and 0, for air only cells. Strictly between 0 and 1 is, therefore, the interface, where the Equation 3.7 applies. Figure 3.5 represents schematically the VoF applied to a wave on a given 2-D grid.

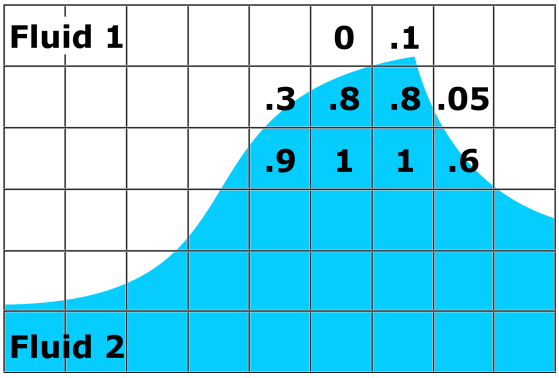


Figure 3.5: 2-D grid of an arbitrary flow using the VoF method - adapted from Rusche (2003)

Surface capturing

The VoF method reconstructs the free-surface using the transport equation applied to the phase field α:

$$\frac{\partial \alpha}{\partial t} + \nabla(\overrightarrow{u}\,\alpha) = 0. \tag{3.8}$$

The transport equation comes from the similarity between the momentum equations and expresses the motion of a field within a given flow. The Multi-Dimensional Limiter for Explicit Solution (MULES) method is implemented in OpenFOAM (developed by H. G. Weller) to assure boundedness and improve the capture of the interface (Jasak 1996) using an artificial compression term (function of the relative velocity between the two phases) added to the volume fraction. Rusche (2003) details the compression technique mathematically and its implementation.

3.1.4 Turbulence Modelling

The RANS equations averaged the flow fields to model the effect of turbulence while avoiding to capture all of the turbulent scales that requires significantly small space and time discretisation.

Depending on the ratio between inertial and viscous forces, the flow can become turbulent or remain laminar. The Reynolds number, introduced by G.Stokes and popularised by O.Reynolds, characterises the flow and identifies the laminar to turbulent transition

if above a threshold value. It is defined as:

$$Re = \frac{Inertia \quad Force}{Viscous \quad Force} = \frac{uL}{\nu}, \tag{3.9}$$

where ν the kinematic viscosity of the fluid. u and L are, respectively, the case-specific 'characteristic' velocity and length. At high Reynolds number, the inertia forces dominate and the flow is turbulent, while at low Reynolds number, the flow remains laminar as the viscous forces prevail (Versteeg and Malalasekera 2007).

The choice of a turbulence model is problematic. Multiple turbulence models are nowadays available and already implemented in OpenFOAM. Apart from the laminar model, which neglected turbulence, the most common one probably is the '$k-\varepsilon$', which characterised turbulence as an energy dissipation. But, due to the lack of this model in, for example, rotating flows (Versteeg and Malalasekera 2007), other models, such as the '$k-\omega$' and its variations, have emerged. This leads to a complex and wide range of different models, which are case specific, making unclear the selection of the right model.

In the Marine Renewable Energy (MRE) sector specifically, no clear consensus has been found on the required use of a given turbulent model compare to the other, and it seems that the choice of the appropriate model is left to the user responsibility. In the CFD review for WEC, Windt et al. (2018) indexes 18.3% for laminar, 49.6% for all $k-\varepsilon$ models, 27.8% for all $k-\omega$ ones, and 4.3% for Large Eddy Simulation (LES) ones. Even if the $k-\varepsilon$ models appear predominantly, the reasons for the choice of a model are often absented or silenced. So, the use of a turbulent model is case-specific.

It is considered that for the simulation of wave breaking (Brown (2017), Bredmose and Jacobsen (2010)), turbulence becomes important and therefore, turbulence models are required. This suggests their requirement as well for the modelling of extreme events, but lower to the breaking point their use might be not mandatory. In CFD simulation using a $k-\varepsilon$ turbulence model, Lin and Liu (1998) shows that turbulence appears in the

crest of the wave when this one is about to break in the surf zone. This conclusion obtained from a CFD model is said to agree quantitatively with experimental observations (Nadaoka et al. 1989). So, the use of a turbulence model is specifically mandatory when waves do break. Therefore, as none of the wave-groups were breaking in the physical experiment, no turbulence model is considered required for their numerical modelling.

In addition, estimating the maximum water velocity from linear theory (later detailed in section 3.6.1 Table 3.4), the Reynolds number ranges from $4.25x10^5$ for the least steep case, to $6.65x10^5$ for the steepest case. Comparing the present flow with the well-studied flow around a cylinder, the flow simulated in the present study is 'supercritical' (i.e. $3.5x10^5 < Re < 1.5x10^6$, Sumer et al. (2006)). The flow is expected to be turbulent in the wake (i.e. downstream the buoy) while the boundary layer will remain laminar. In this regime the drag coefficient is almost constant. The drag force is expected to be similar for all cases since the turbulent regime is the same. So, the same turbulent model can be used for the four cases. Since the Reynolds number is lower than the critical transition from laminar to turbulent, and since the turbulences are expected to be present only within the wake of the buoy, hence of minor interest, the turbulence model is chosen as laminar across the present study.

Furthermore, each turbulent model can give a different solution. In the event of a breaking wave in the surf zone (hence proven to require turbulence modelling), Brown et al. (2016) shown that five turbulence models give five different solutions with sometimes significant difference compare to experimental data on turbulence (i.e. turbulent intensity, mean vorticity, and eddy viscosity) from Ting and Kirby (1994). So, the choice of the model needs to be assessed in comparison to turbulent experimental data, which were not measured in the referent experimental set-up (section 2.4).

Therefore, the flow is considered laminar across the study. The author is aware of the potential, and supposed, lack in accuracy but would like to insist that the assessment of turbulences are not the point of interest in this study. Furthermore, if turbulences

became of concern, turbulent models can easily be added to the simulations, and the analysis should be conducted with experimental data on turbulence.

3.1.5 Boundary and Initial Conditions

The final requirement is the definition of the domain boundaries, and the initial state, from which the physical processes are evolving. Boundary conditions impose the behaviour of the fields across the length of study at domain's boundaries; while, initial conditions define the state of each field inside the domain at the starting time (i.e. $t = 0$) by satisfying the equations.

Physical boundaries (e.g. wall, atmosphere, inlet, structure) defining the domain are modelled by specifying a mathematical description for each field. The two mains types of boundary conditions are Dirichlet ones, which fix the field to a specific value; and Neumann ones, which define a gradient of the field normal to the boundary. For example, a wall is represented using a `noSlip` boundary condition specifying zero velocity, with a `zeroGradient` on the volume fraction, α, and the pressure gradient is forced to be equal to the velocity boundary condition using the `fixedFluxPressure` condition. Additional specific boundaries are implemented for wave-generation and absorption.

Wave-generation and absorption

Waves can be generated by reproducing the motion of wave-makers as in a PWT using moving boundaries to implement the movement of paddles. This dynamic boundary is implemented in the OpenFOAM toolbox `olaFlow` (Higuera (2019), detailed in Higuera et al. (2013)). This method benefits from the flow description resulting from the moving boundaries, hence implying a closer reproduction of the physics made in physical experiments as both are wedge/piston type wave-makers (Wei et al. 2016). It also induces common physical wave-makers problems, where reflected waves coming back at the wave-maker need to be accounted within the wave generation.

A mathematical description of the wave can be imposed to generate the wave. A static boundary defines the wave by its velocity field as a Dirichlet condition directly at

the boundary. This method is computationally efficient compared to the previous dynamic boundary as no mesh motion result from the wave-generation. (Higuera Caubilla et al. 2015) found an increase from 20% up to 40% when using a dynamic boundary compare to a static boundary. Mass or impulse source method, respectively adds a mass or source term to the continuity or momentum equation (Schmitt and Elsaesser 2015). These methods generate waves in multiple directions and are specified at a location different from the boundary condition. They require an absorption method at the boundary opposite to wave direction of interest. However, none of these techniques is yet implemented in OpenFOAM-4.1 and is therefore excluded.

A second mathematical solution is the relaxation method, illustrated in Figure 3.6, developed by Jacobsen et al. (2012) and implemented in the toolbox `waves2Foam`. A full description of the toolbox is available at Jacobsen (2017). The relaxation method defines an extended boundary condition (i.e. a section of the mesh), in which the analytical solution of the wave f_a is imposed gradually to the solution f_s across the extended boundary via the weighting function α_R by:

$$f = \alpha_R f_a + (1 - \alpha_R) f_s. \tag{3.10}$$

The solutions f_a, f_s and f define the velocities and surface elevation of the wave. α_R is defined by default as:

$$\alpha_R(\chi_R) = 1 - \frac{exp(\chi_R^{3.5}) - 1}{exp(1) - 1} \qquad \text{for} \chi_R \in [0, 1]. \tag{3.11}$$

where, χ_R is the scaled x-coordinate along the relaxation-zone. $\alpha_R(\chi_R)$ varies from 1 at the boundary start, to 0 at its end.

The analytic solution of the wave f_a follows potential theory. Irregular waves are generated as a superposition of regular ones based on the linear superposition defined by Equation 2.16. The given analytical solution is applied at wet (i.e. $\alpha = 1$) cells centres in the boundary. Velocity $\overrightarrow{u}$ is defined by the `waveVelocity` boundary condition.

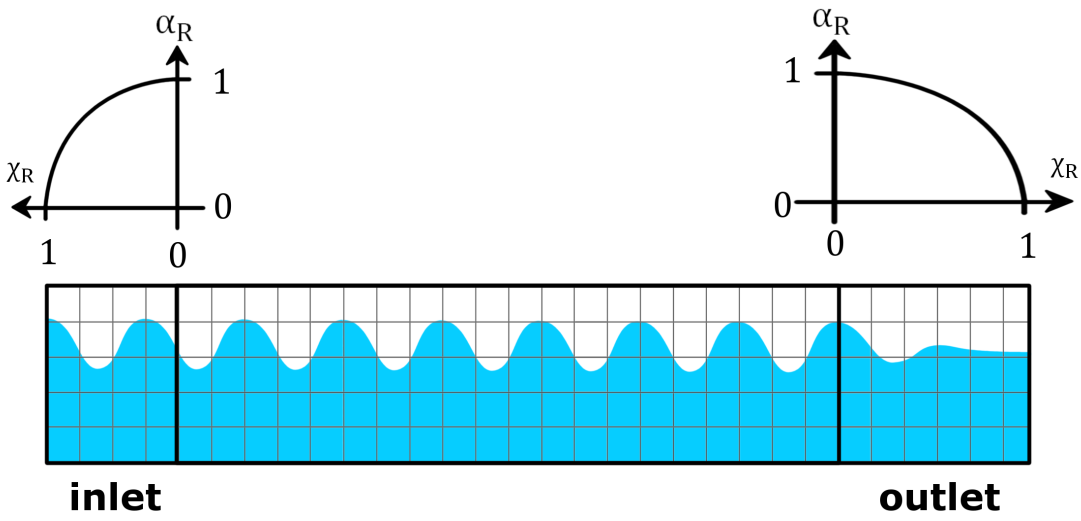


Figure 3.6: Sketch of the inlet and outlet relaxation zones defined using the weighting function, $\alpha_R(\chi_R)$ - adapted from Jacobsen et al. (2012)

For a regular wave, the velocity field is specified horizontally as (waves2foam Code Repository 2019):

$$u = a\omega\frac{cosh(k(z+d))}{sinh(kd)}cos(kx-\omega t+\delta), \tag{3.12}$$

And vertically as:

$$w = -a\omega\frac{sinh(k(z+d))}{sinh(kd)}sin(kx-\omega t+\delta), \tag{3.13}$$

Where, z is the depth of the given cell centre. The non-hydrostatic pressure `p_rgh` is also defined from potential theory at a given depth z as:

$$p_{rgh} = \rho g a\frac{cosh(k(z+d))}{cosh(kd)}cos(kx-\omega t+\delta). \tag{3.14}$$

Additionally, the normal gradient ($\overrightarrow{n}\nabla p_{rgh}$) remains driven by the velocity boundary condition using the `fixedFluxPressure` condition (specified as zero). Volume fraction fluid (α) is calculated as the ratio between the wet volume and the dry one from a cell. Wet volume is estimated from the volume below the surface-elevation assuming a liner interpolation between the points intersecting the cell faces. The corresponding

boundary condition is `waveAlpha`. Dry cells (i.e. $\alpha = 1$) are specified as a wall up to the surface-elevation.

The main advantage of the `waves2Foam` toolbox is that different waves theories are available to describe a wave at the boundary, and superposition of linear waves can be easily implemented. Therefore, the reproduction of the experiment can be done by imposing the linear superposition of a Wave Gauge (WG) at the boundary. This method has been widely validated for extreme waves (e.g. Vyzikas et al. (2014), Ransley (2015), Rafiee and Fiévez (2015)). The second advantage of the relaxation-zone technique is that it allows the absorption of waves using a still water solution. Therefore, reflected waves will not affect the imposed solution. Besides, a second relaxation-zone can be used for the absorption of the generated waves at the other side of the domain, Figure 3.6. The main drawback is that relaxation-zones require long zones to generate and absorb the wave, where the dynamic boundary method automatically adapt to account for reflected waves. Simonetti et al. (2015) found a similar level of error compared to experiments for the dynamic boundary (8%) and the relaxation method (9%), while the computational effort three times less for the relaxation method. Therefore, the `wave2Foam` toolbox is used for wave-generation and absorption.

3.1.6 Rigid-Body Motion

The Rigid Body Motion (RBM) solver solves the conservation of momentum (i.e. Newton 2^{nd} law of motion) applied to a moving structure:

$$\sum \overrightarrow{F} = m \times \overrightarrow{a} \tag{3.15}$$

where $\overrightarrow{F}$ gathers the forces acting of the structure (e.g. hydrodynamics, moorings, Power Take Off (PTO)), m is the rigid-body mass, and $\overrightarrow{a}$ is the rigid-body acceleration. Within the RBM solver is implemented the mesh motion solver that updates the mesh from the solved position of the structure (`dynamicMotionSolverFvMesh`).

The `sixDoFRigidBodyMotion` and the `rigidBodyDynamics` are two RBM solvers

implemented in OpenFOAM-4.1. `sixDoFRigidBodyMotion` calculates forces and moments applied on the rigid-body to then integrate the acceleration and angular acceleration in order to obtain the new position. Added in version 4.1, `rigidBodyDynamics` is a multi-body dynamic library based on (Featherstone 2014) (as WaveDyn; later detailed). It differs from the `sixDoFRigidBodyMotion`, as it is capable of dealing with the DoF of interconnected articulated bodies.

The `sixDoFRigidBodyMotion` specifies a zone in which the deformation of the mesh is restricted. This one can start away from the rigid-body surface. `rigidBodyDynamics` spreads the deformation across the entire mesh starting from the rigid-body surface. Both RBM diffuse the deformation according to a diffusion parameter, hence only with the zone, or in the whole mesh. So, `sixDoFRigidBodyMotion` concentrates the deformed mesh into the zone, hence decreasing the quality of the mesh only locally. Also, this can assure that the mesh near the model remains of high quality. However, motion responses are found to be dependent on the selected zone. On the other hand, `rigidBodyDynamics` is not dependent on the zone and the diffusion of a given deformation is necessary smoother in this RBM solver, as the zone can only be shorter than the mesh. Also, `rigidBodyDynamics` has been validated by Rafiee et al. (2016) in the modelling of extreme motion response of a point-absorber WEC compare to experiment. Therefore, `rigidBodyDynamics` has been selected for the present study.

The `rigidBodyDynamics` library decomposed a structure into multiple 'rigid bodies' linked together by 'joints'. The motion of a rigid body system can be written as a function of the joints position (q), velocity ($\dot{q}$), and acceleration ($\ddot{q}$), using a tensor (generalized 6 attributes vector or matrix) writing of Newton second law of motion (Kumar 2016):

$$I(q)\ddot{q} + C(q,\dot{q}) = \Gamma \tag{3.16}$$

where $I(q)$ is the inertia tensor, $C(q,\dot{q})$ is a force tensor gathering Coriolis, gravity and centrifugal forces, and Γ is the total force acting on the bodies. The accelerations of the joints, $\ddot{q}$, are functions of the forces, the joints actual velocities and position. Then, by integrating over the time-step, positions and velocities are updated, becoming the new input for the next time-step. Forces restraining the joints motion can be modelled using `restraints`, while DoF can be restricted using `constraints`.

3.1.7 Resume of Numerical Decisions

A Numerical Wave Tank (NWT) for extreme WSI is developed in Chapter 3 using the OpenFOAM CFD code, where the RANS Equations are solved for an isotropic, Newtonian, incompressible, two-phase flow, using the Finite Volume Method (FVM) with the Volume of Fluid (VoF) interface capturing scheme. The additional decisions are:

- Courant number $C0$ is limited to 0.5.
- Face interpolation uses a linear scheme.
- Gradient terms (∇) are solved using a Gaussian linear scheme.
- The convective term ($\nabla(\rho UU)$) is solved by a Van Leer Scheme considering the direction of the field. Other divergence terms ($\nabla.$) are solved using a Gaussian linear scheme.
- Laplacian terms (∇^2) are solved by a linear corrected scheme.
- Time discretisation solving uses an Euler scheme.
- PIMPLE algorithm is used with one outer and two inner corrective loop.
- The interface is captured by the VoF method improved by the MULES compression technique, where the divergence term uses a Gaussian linear scheme.
- The flow is considered laminar as below the wave breaking limit and since $Re < 10^6$.
- The wave-generation and absorption is performed by the `waves2foam` toolbox.
- The Rigid Body Motion (RBM) solver is the `rigidBodyDynamics`, with the corresponding mesh-motion solver, `dynamicMotionSolverFvMesh`.

Numerical schemes and these CFD decisions are reported in Appendix C.2.

3.2 Geometric Parameters for the 2D-NWT

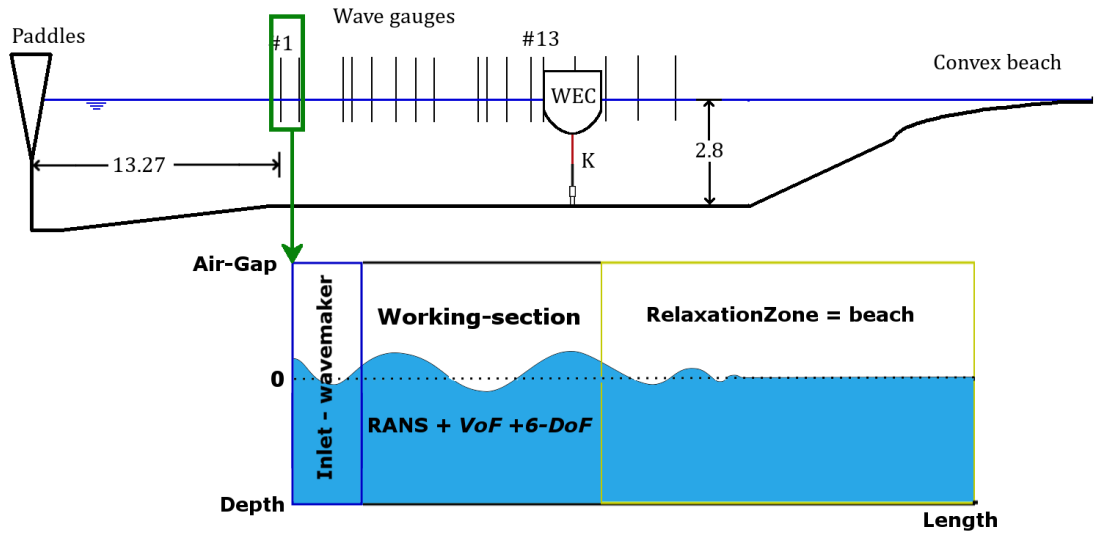


Figure 3.7: Schematic representation of a $2D$-NWT separated in three sub-domains: the inlet or wave-maker; the working-section where the fluid motion is solved using CFD; and the beach or relaxation-zone

As the wave propagation is considered unidirectional, the NWT is a two-dimensional slice along the length of its reference, the COAST PWT (section 2.4.1). The $2D$-NWT is schematically represented in Figure 3.7, where the wave-generation and absorption are conducted using the `waves2foam` library previously selected for different reasons presented in section 3.1.5 along with the method the library used to apply and absorb waves by imposing a solution via a weighting function through the boundary length.

It consists of three sub-domains by analogy with a PWT:

- The *inlet* acts as a wave-maker and generates the wave by imposing the linear superposition of the WG#1 via a weighting function through the inlet length (section 3.1.5).

- The *Working-section* is the zone of main interest, where the RANS equations are solved for an isotropic, Newtonian, incompressible two-phase flow using the FVM and the VoF interface capturing scheme (section 3.1).

- The *beach*, or relaxation-zone, is situated at the opposite of the inlet (sometimes called *outlet*) and aims to absorb the upcoming wave by imposing a still water solution via a weighting function through its length (section 3.1.5).

Figure 3.7 presents the different geometrical parameters of a NWT: the air-gap, the depth, the inlet length, working section length, and the beach length. The depth is chosen according to the reference PWT, hence $2.8m$. In a PWT, the area of interest and the first wave-gauge are some distance away from the paddles. The NWT is chosen to start at the first WG to save computational effort, as made possible by imposing a solution.

For general applications, WG data are not available. The surface-elevation at the model centre of gravity is defined from the extreme event used for the assessment. Therefore, the NWT aims to produce this exact solution at the model location without upstream (relative to wave propagation) WG. An example similar to this study would be to simulate the NewWave theoretical surface-elevation at the model centre of gravity. This is commonly done by imposing the wave spectrum at the inlet from a linear superposition of harmonics while setting the phase angles so that the wave-group is produced at the desired time and position (Jasak et al. (2015); Zhao et al. (2010)). This method is not used in the present study since the objective is to reproduce the experiment in a blind-test approach assuming that only the most upstream WG is available.

The maximum surge motion of the buoy under the four extreme events is approximately $0.6m$. In the light of future $3D-$NWT development, the working section should end $0.85m$ (maximum surge plus radius) away from the WG#13. The front edge of the buoy being aligned with WG#13, the buoy centre of gravity at rest is $5.58m$ away from WG#1, so the working section must be at least $6.43m$. The numerical model might over-estimate the surge response, which could bring the buoy inside the numerical beach, hence disturbing its motion as a still water solution is imposed on the flow in this section. To avoid this, i.e. make sure the buoy stays within the working-section through all events, a margin is considered and the working section ends $7m$ away from

WG#1.

No clear consensus across the NWT users is found for the air-gap. If chosen too small, the air-gap can influence the solution by compressing the air-phase when a wave travels, while the larger the air-phase the more computational effort is put specifically for this section of minor interest. Focus waves of similar maximum wave height ($0.2m$ and $0.25m$ respectively) realised at the COAST PWT are numerically reproduced by Vyzikas et al. (2013) and Ransley (2015) using a $0.5m$ air-gap in $2D$-NWT. However, Ransley (2015) increases the air-gap to $1.5m$ in $3D$ cases to account for the deformation of the mesh due to the motion of the structure, because this last is found to be influenced by the dimension of the deformable section of the mesh, and so the available mesh surrounding the structure. If the top of the structure is close to the top of the NWT due to a small air-gap, the deformation of the mesh due to vertical motions will be concentrated into the few cells above the structure. This will generate highly expanded or contracted cells, which decreases the quality of the mesh and influences the solution. The computational effort of wave-only cases in a $2D$-NWT remaining small, the present study fixes the air-gap to $1.2m$ in the light of future $3D$ simulations involving a moving structure, and to obtain an integer for the total NWT's height.

A given wave solution is imposed across the length of the inlet, meaning that the RANS equations do not govern the flow. The longer the inlet, the more the flow is governed by this wave solution and not by the RANS, hence imposing a theoretical propagation of the wave (e.g. linear). On the other hand, `waves2foam` library requires more than one cell (in the direction of the wave-propagation) to impose the wave (Jacobsen et al. 2012). Therefore, the inlet needs to be short enough not to impose a false theoretical propagation of the wave, while long enough to impose the wave itself and absorb waves reflected by the structure. The two previous studies agreed on the use of a $1m$ inlet, which is used in the present study.

The geometric parameter remaining to be defined is the length of the numerical beach (or relaxation-zone). Its assessment is specific as the influence on the solution and

computational effort are significant.

Relaxation-zone are selected as a function of the input wavelength. Jacobsen et al. (2012), whom develop the relaxation-zone technique in OpenFOAM, show for a single regular wave input, that the efficiency of the absorption is a function of the wavelength and relaxation length. Hence, the relaxation-zone length is selected from the input wavelength depending on the desired absorption. Ransley (2015) applies this method for focus NewWave wave-groups by selecting the component with the most energy (i.e. peak of the energy spectrum). The resulting absorption is estimated from Jacobsen et al. (2012) as less than 95%. However, this method applies a result obtained from single regular waves to a wave-group described by a linear superposition. It is therefore debatable that the wavelength of the peak frequency would be the one of largest influence on reflections. Besides, there exist non-linear wave-wave interaction between the components of the linear superposition. This interaction is not considered when selecting the relaxation-zone length from the peak frequency. Additionally, the present NWT aims to mirror its physical equivalent, hence achieve similar wave reflections. Yet the PWT reflections are believed to be a non-linear relationship between the wave frequency and amplitude (Ransley (2015) via an internal communication with Dr Keri Collins). Hence, selecting the relaxation-zone from a single wave component seems inadequate to mirror the PWT reflections. Therefore, the present study does not select the relaxation-zone length from a unique component of the input linear superposition. The relaxation-zone length is fixed across all events to obtain a unique NWT capable of reproducing multiple sea-state.

The CFL has been selected to 0.5 in section 3.1.2, so only the spatial discretisation remains, i.e. the resolution of the mesh. A convergence study is required to assess that the solution is not influenced by a refinement of the mesh (i.e. mesh independent (Versteeg and Malalasekera 2007)). The $2D$-NWT is assessed over the solution of the considered wave. So, the mesh convergence is conducted over the assessment of the minimal number of cells per wave height (CPWH) required to obtain a solution

independent from further refinement of the mesh. However, each refinement increases the computational effort.

No clear consensus on the number of CPWH is defined for WEC applications (Windt et al. 2018) due to the wide range of waves simulated. For example, Ransley (2015) showed that for regular waves of increasing steepness (kA) from 0.03 to 0.31, 10 CPWH is adequate to obtain less than 2% Root Mean Square (RMS) error in the free surface time-series. However, the extreme events of the present study are focus events, which might make this investigation inapplicable.

In the simulation of different NewWave events, Chen et al. (2016) found 56 CPWH as the optimal compromise between accuracy and efficiency, Vyzikas et al. (2013) found an optimum around 17 CPWH, and Ransley (2015) applies the previously investigated criterion and uses 10 CPWH. Across the three studies, a qualitative comparison of the surface elevation obtained generally shows a good agreement with other numerical model or experimental data, which highlights the necessity to assess the number CPWH specifically for the present study.

The present study chooses to use only square mesh cells (cube in $3D$) without mesh grading due to the influence of the aspect ratio on the solution. Ransley (2015) investigated the number of cells required to reach mesh independence using the run-up of a solitary wave in a NWT. The use of cells of aspect ratio above or below 1 (in the direction of the wave propagation or normal to the surface) were found to increases the number of cells required to reach mesh independence. These conclusions are considered as applicable for the simulation of extreme events in the present study, as the solitary wave used was the largest for the depth. It will be applied also in $3D$.

Square cells are defined by the ratio between vertical and horizontal resolution to 1:

$$\frac{\lambda . N_H}{N_\lambda . H} = 1 \tag{3.17}$$

Where, λ is the wave-length, H is the wave-height, and N_H and N_λ are number of cells

per wave-height and wave-length respectively. In the following, the vertical resolution is applied horizontally (i.e. $\frac{\lambda}{N_\lambda} = \frac{H}{N_H}$).

3.3 2D-NWT Methodology

The length of the numerical beach and the number of CPWH are identified as two decisive parameters that requires a specific assessment in the development of a $2D$-NWT able to reproduce extreme events realised at the COAST PWT.

The following section presents a methodology developed by the present study based on a numerical solution of known greater accuracy.

3.3.1 Surface-Elevation Measurements

The assessment is done on the surface elevation using numerical WG, called Numerical Probe (NP). Table 3.1 shows the equivalence of the thirteen WG positions between the PWT and the NWT, where the NP are specified in the NWT frame, i.e. zero being WG#1. The advantage of NP compared to WG is that NPs do not influence the flow, because a numerical scheme is used to track the interface. Additional NP are positioned in the NWT to assess precisely the surface elevation as a function of space, the wave-generation and absorption.

WG#	1	2	3	4	5	6	7	8
x in PWT [m]	13.27	14.32	14.77	15.23	15.7	16.11	16.54	16.82
x in NWT [m]	0	1.05	1.5	1.96	2.43	2.84	3.27	3.55
WG#	9	10	11	12	**13**	14	15	16
x in PWT [m]	17.14	17.51	17.84	18.21	**18.54**	18.86	19.23	19.60
x in NWT [m]	3.87	4.24	4.57	4.94	**5.27**	5.59	5.96	6.33

Table 3.1: Waves Gauges position in the PWT and NWT - x is the wave propagation axis - the four focus events were focused at the WG#13, which is aligned with the front edge of the X-MED model

3.3.2 Accuracy Criterion

Across the study, the comparison between two data sets is supported using the calculation of the Pearson 'correlation coefficient' (Galton 1886). It is a measure of the linear correlation between two variables A and B, noted $R(A,B)$ and is defined as:

$$Correlation(A,B) = R(A,B) = \frac{cov(A,B)}{\sigma(A) * \sigma(B)} \tag{3.18}$$

where σ is the standard deviation, and $cov(A,B)$ is the covariance. Using the statistical expected values $E[A]$, or $E[B]$, the correlation coefficient can be also expressed as:

$$Correlation(A,B) = \frac{E[AB] - E[A].E[B]}{\sqrt{E[A^2] - E[A]^2}.\sqrt{E[B^2] - E[B]^2}} \tag{3.19}$$

For strongly correlated variables, the correlation tends towards 1, while the correlation for uncorrelated variables decreases towards 0. In the study, the variable A is the reference data set, while B is the compared data set. For example, in Figure 3.9, the surface elevation of the full beach $60m$-NWT is the reference, A, while one of the other surface elevations measures is the comparison set, B.

By defining a threshold over the correlation coefficient, a certain level of accuracy of the compared data set can be considered as attained at this threshold. This is used to assess the convergence of a solution.

Unless stated otherwise, the analysis is conducted using $20s$ long time-series. Experiments data are therefore truncated. As OpenFOAM sampling differs from experiments, both time-series are interpolated assuming a spline variation at a fixed time vector of $0.01s$ time-step (i.e. $(0 : 0.01 : 20)$).

3.3.3 Assessment of Relaxation-Zone Length on Solution

The absorption of a wave or wave-group in the relaxation-zone is incomplete for any length. A better understanding of the absorption is required, where the objective is to

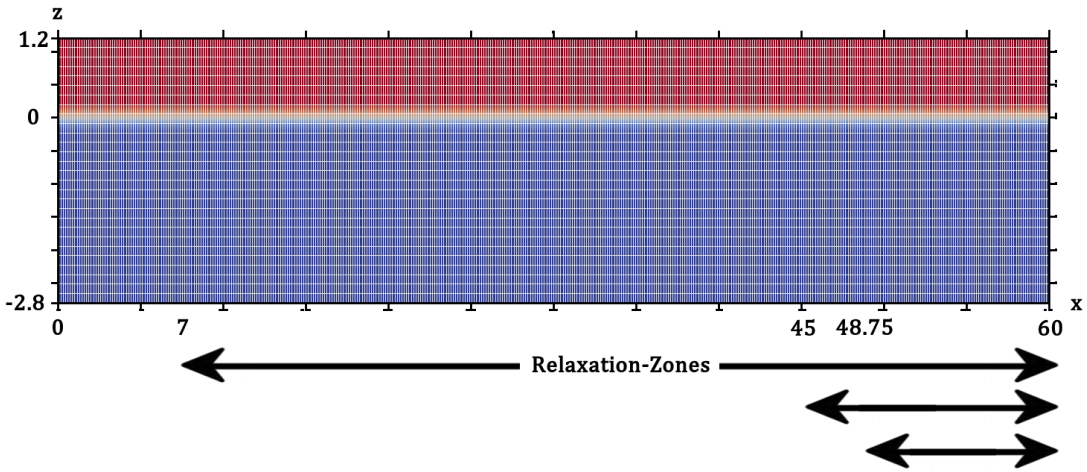


Figure 3.8: $60\,m$ $2D$-NWT at initial time - z is scale by 4

develop a NWT of known greater accuracy to use it as the reference. The objective is to get the best numerical solution for the given NWT, and not the best matching solution compared to theory or experimental data in order to avoid conclusions dependent to the experiment data-set or a given theory.To identify specifically the consequence of the wave absorption (or reflections) on the solution, the assessment needs to be conducted over the solution of a NWT using the same solving procedure but where the wave reflections are not affecting the solution.

Using a NWT 'sufficiently' long and building up from a still sea-state, the reflections of the wave-group of the NWT end are expected not to affect the solution for at least the time it requires for the wave to travel to the NWT end. The resulting solution before reflection will be of known greater accuracy and could be used as the reference. The principle behind this investigation joins the wave-absorption technique that consists of gradually increasing the NWT cells towards the propagation of the wave to model an infinitely long NWT (Windt et al. 2018).

Three $60\,m$ long $2D$-NWT of increasing relaxation-zones - $53\,m$, $7.5\,m$ and $15\,m$, Figure 3.8 - are used to highlight when a solution is affected by waves reflections. A Stokes 2^{nd} order wave ($H = 0.25\,m$, $T = 3.56\,s$, $\lambda = 15.29\,m$) is selected from Figure 2.3. The simulation starts from a still sea-state, and is $50\,s$ long, to show the development (i.e.

build-up) of the solution. The number of CPWH required for this wave was found to be 3 (result available on the poster presentation bounded - E.3.4), so the mesh is made of only square cells of $0.083\,m$ side length (i.e. $N_H = 3$ and $N_\lambda = 183$). The comparison is conducted between the surface elevations measured using NP at $x = 6.63\,m$ from the inlet.

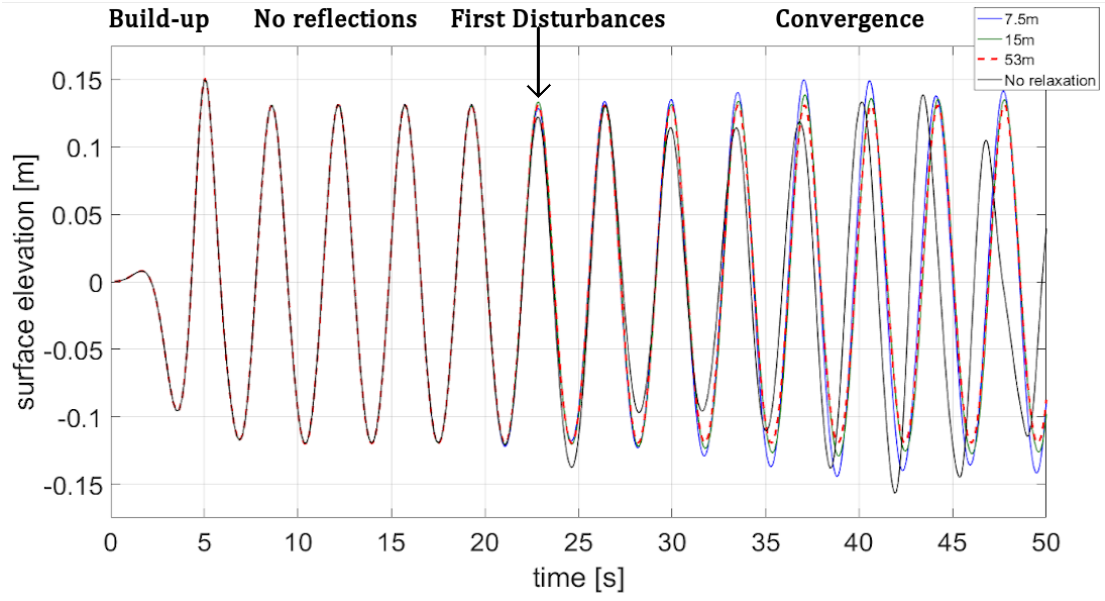


Figure 3.9: The surface elevation measured numerically at $x = 6.63\,m$ from the inlet for three $60\,m$ long NWT of different beach length. 4 sub-time-domains in the solution of the surface elevation are highlighted: Build-up, No-reflections, First disturbances, and Convergence.

Figure 3.9 shows the three surface elevations obtained from the three $60\,m$ long $2D$-NWTs of different relaxation-zones length and without (black line). It highlights that the time-series are divided into 4 sub-time-domains. From time $0\,s$ to $6\,s$, called 'Build-up' in Figure 3.9, the first generated at the inlet has travelled to the measurement position. The solution is building-up, towards the second sub-time-domain, named 'no-reflections' from time $10\,s$ to $22.5\,s$, where the three solutions are equal independently from the numerical beach. The first disturbances in the solutions appear at $22.5\,s$, indicated by the arrow in Figure 3.9. Finally, the disturbances converge to three different solutions, marked as the 'convergence' sub-time-domain.

If reflections were to start affecting the solution depending on the relaxation length,

each solution would diverge from the $53\,m$ long one at different times. For example, if reflections were generate at the entrance of the relaxation-zone. As all solutions diverge at $22.5\,s$, even when no relaxation-zone is used, the effect of reflection starts independently from the relaxation-zone length, and reflections happen at the end of the NWT.

The wave-number of the input wave is $k = 0.3954$, the wavelength $\lambda = 15.29\,m$. Therefore the phase celerity is $c = 4.29\,m/s$ and the group celerity $c_g = 3.19\,m/s$. If reflected waves were to have the same celerity as the wave-group, they would be at $x = 6.63\,m$ at $35\,s$ ($(2.NWT_{length} - x)/c = 35\,s$). When the wave do reflect at the NWT end in the case without relaxation-zone, the reflected waves travel back towards the inlet. The effect of these waves on the ones coming from the inlet is probably preceding them. This would explain why solution is affected by reflections at $22.5\,s$. The velocity of the effect of reflections is very fast since the wave-group reaches the NWT end at $t = 60/3.19 = 18.8\,s$, hence propagating at $u_r = 53/(22.5 - 18.8) = 14\,m/s$.

Additionally, by limiting the solution of a $60\,m$ $2D$-NWT to the first $20\,s$, the solution is proven to be unaffected by wave-reflections, hence proving the existence of a solution of known greater accuracy. Therefore, the $60\,m$ $2D$-NWT with a $53\,m$ long relaxation-zone will be used as the solution of greater accuracy in the assessment of the number of CPWH and the length of the relaxation zone.

3.3.4 Methodology: Numerical Beach and Convergence Study

Therefore, this study proposes a new methodology whereby the comparison is made with a numerical solution of known greater accuracy. Here, the solutions from the full beach $60\,m$-NWT are known not to be affected by wave-reflections before the first $20\,s$. Using this NWT as the reference, two parameters are still undefined in the development of a $2D$-NWT capable of reproducing extreme events.

First, the mesh resolution, i.e. the number of CPWH, needs to be established to assess the accuracy of the solution. The assessment is done using $20\,s$ long simulations of $60\,m$ long $2D$-NWTs with meshes of increasing resolution defined by the number of

CPWH. To develop a NWT able to reproduce extreme events, the least steep, $ST1$, is used. This event is selected since it is the least nonlinear, hence requiring the finer vertical resolution to minimise numerical diffusion. This wave-group is expected to be the least challenging to reproduce as the least steep event, i.e. the least non-linear. The measure of convergence is defined by the correlation between a solution and the assumed 'converge' solution evaluated qualitatively.

Using the mesh resolution, the second NWT parameter that needs to be defined is the length of the numerical beach because a $60m$ long $3D$-NWT will be inapplicable in terms of computational effort. The numerical beach efficiency is known to depend on the wavelength (Jacobsen et al. 2012), so a regular wave only cannot be used. The reference simulation is the least steep case $ST1$, done under the $60m$ $2D$-NWT with the adequate number of CPWH. It compares with NWT of increasing length, where the numerical beach starts from the end of the *working-section* previously defined at $x = 7m$, Figure 3.7. So, by increasing the numerical beach, the NWT increases.

The assessment of both parameters is done as a compromise between computational effort and accuracy. The comparison is conducted over the surface elevation measured at the focus location ($x = 5.58m$ away from the inlet WG#1 - table 3.1) for a $20s$ solution. The measure of accuracy is assessed by the correlation between the given solution and the reference, as defined in section 3.3.2. The selection of the correct parameter is made using a correlation criterion fixed to 0.9999 for the number of CPWH, and 0.999 for the relaxation-zone.

The computational effort is assessed by running all simulation on a single processor on the cluster (as not computationally costly to require parallel processing). Named Fotcluster2 (2019), it is a HPC facility made of 6 *Viglen HX425T*^{2}*i HPC 2U* Compute Nodes, which are equipped with *Dual Intel Xeon E5650* processors (6-cores, $2.66GHz$) and $12GB$ of memory per motherboard.

3.4 2D-NWT set-up

As a pre-processing step, the definition of the extreme event imposed at the inlet needs to be defined before the assessment of the number of CPWH and the length of the relaxation-zone.

3.4.1 Wave Input

To replicate the extreme event of the experiment, the surface elevation at WG#1 is imposed at the inlet. On the contrary to a regular or Stokes 2^{nd} order wave, there is no analytical definition of this surface elevation which can be imposed at the inlet. So, the approximation obtained by a Fast Fourier Transform (FFT) (detailed in section 2.1.3) of linear superposition of components is imposed at the inlet.

The accuracy of the approximation will affect the representation of the event, meaning that the number of components used for the superposition needs to be assessed. Vyzikas et al. (2013) reproduced a NewWave event similar to the least steep event used in the present study, by imposing at the inlet 243 wave components with frequencies evenly spaced between $7.8x10^{-3}$ and $1.89Hz$. The idea was to mirror the input of the actual wave-maker of the PWT. However, the inlet of the NWT is not limited by the wave paddles capacity and the range of frequencies is unlimited. Besides, in the development of NWT via a blind-test, Brown et al. (2018) found that the execution time increases exponentially with the number of components due to the increase in requirement of Random Access Memory (RAM). From a wave spectrum perspective, some components have more significance than others do. So, it seems possible to reduce the number of components by omitting the one of least amplitude, while maintaining the level of accuracy of the linear superposition.

Therefore, the present study develops a pre-processing tool which optimises the selection of the number of components based on an user-specified level of accuracy of correlation between the surface elevation described by the linear superposition, and WG#1. Figure 3.10 presents the algorithm made on a MATLAB script available in Ap-

pendix E.1.

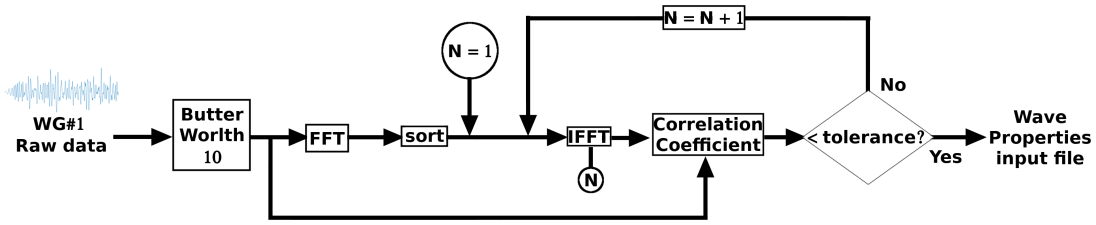


Figure 3.10: From WG measure to ready-to-use input in OpenFOAM

The algorithm starts by filtering the raw measurement obtained from WG#1 to remove the high frequencies. The filtered signal becomes the reference for the correlation. An FFT is applied to obtain all the components, which are then sorted by order of magnitude. Then, starting from the first component - i.e. the one of largest amplitude -, the linear superposition defines the surface elevation for the given number of components, which is then assessed against the filtered signal of WG#1 by calculating the correlation. If this one is lower than the tolerance, the next larger component not already included is added to the linear superposition. When the tolerance is reached, the number of components is selected, and the file used at the inlet (`waveProperties.input`) describing each component is generated.

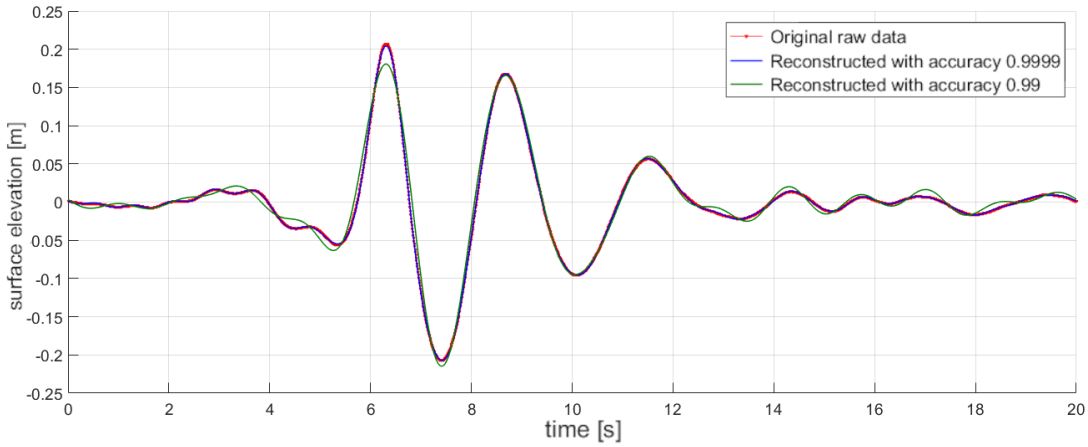


Figure 3.11: The original measure for the surface elevation measured by WG#1, compared to the reconstructed signal using a correlation threshold of 0.9999 or 0.99

The tolerance defines the accuracy of the linear superposition used at the inlet. Figure 3.11 shows how two linear superpositions obtained with a different level of tolerance, 0.9999 (blue line) and 0.99 (green line), compares with WG#1 (red dotted line). By lowering the tolerance, the reconstruction of the signal may become incomplete, and the higher level of tolerance, 0.9999, is deemed necessary and justified by the low number of component (28 - table 3.2).

The performance of the algorithm can reduce significantly the number of components compared to the method used by Vyzikas et al. (2013). Also, sorting the components by amplitude has been identified as beneficial. Using a correlation coefficient criterion of 0.9999, and without sorting of the components by amplitude, the linear superposition requires 76 components for the least steep event. No difference in the wave-only cases was identified compared to inlets using the sorting (result available in Appendix A.2.1). Therefore, this gives confidence that the present algorithm maintains the accuracy of the description of the inlet while improving the speed of the simulation.

Correlation Coefficient	ST1	ST2	ST3	ST4
0.9999	28	32	39	52

Table 3.2: Number of waves components obtained with a 0.9999 tolerance for the four extreme events, and use further

Table 3.2 shows the number of components required for each experimental case. As found by Brown et al. (2018), the number of components increases with steepness. The threshold over the correlation coefficient can be adapted depending on the level of accuracy of the final solution and the affordable computational effort. These decisions are available in Appendix C.8.

3.4.2 Number of Cells per wave height

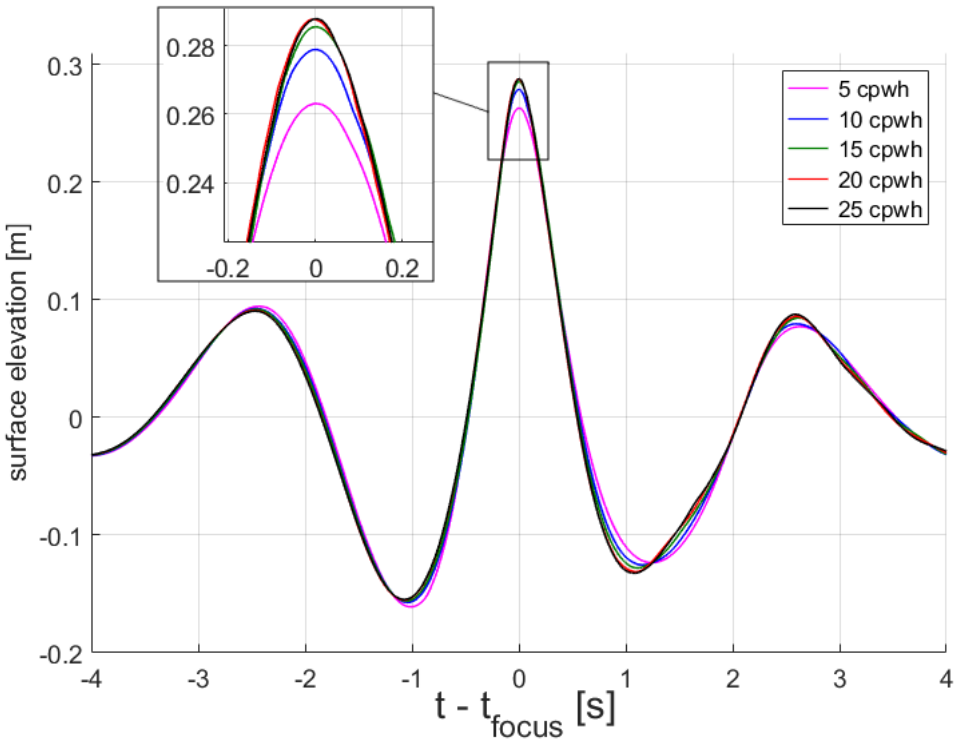


Figure 3.12: Surface elevation at focus location as a function of time for an increasing number of cells per wave height, from 5 to 25, where 25 is identified as the converged solution

From the conclusions on wave reflection disturbances, the convergence study is performed using the $60\,m$ NWT with a $53\,m$ long beach (Figure 3.9), with the comparison restricted between $10\,s$ and $20\,s$ where the surface elevation is undisturbed. The meshes are generated according to the number of CPWH for the least steep event, $ST1$.

The convergence of the simulated surface elevations can be appreciated qualitatively in Figure 3.12. The time has been normalized by the focus time, and the only $8\,s$ around the focus event are shown to highlight differences. Minor differences can be seen between 15 (green line), 20 (red line) and 25 CPWH (black line), meaning that the solution converges toward 25 CPWH. If a finer mesh resolution is used, the solution diverges at the second trough (Figure 3.13) due to the cell-size being too small ($\leq 0.015\,m$), which is considered to generate numerical instabilities. For different wave

regimes, Ransley (2015) also identified that solution were not simply converging as mesh gets finer, even when using very high number of CPWH (more than 100). The converge solution used as the reference is 25 CPWH, from which is assessed the difference in accuracy and computational effort to select the number of CPWH from a tolerance.

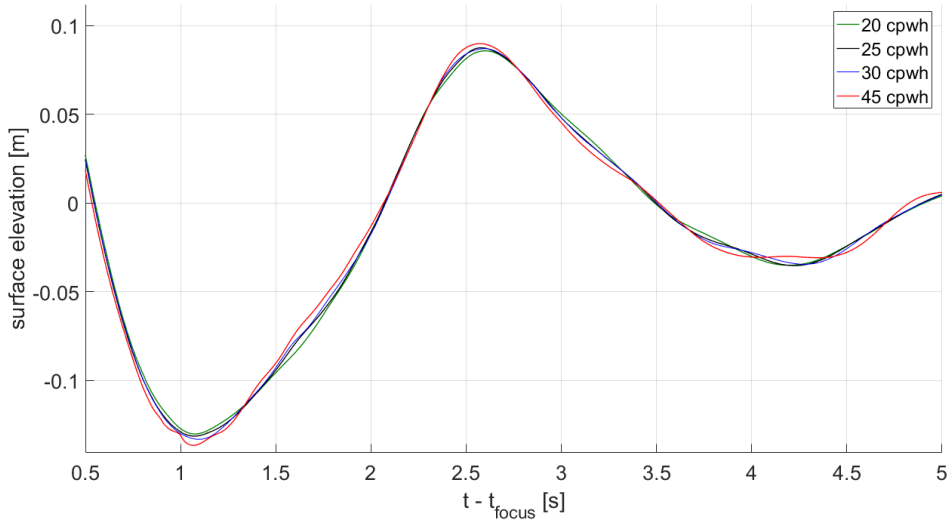


Figure 3.13: Numerical Instabilities for mesh size greater than 25 CPWH

Figure 3.14 left axis shows the correlation between each solution of an increasing number of CPWH with the 25 CPWH reference. Figure 3.14 right axis shows the computational effort.

The number of CPWH is selected as a compromise between accuracy and computational effort. The tolerance threshold of 0.9999 evaluates 20 CPWH, which allows saving more than twice the computational effort compare to 25 CPWH while obtaining the same accuracy. The minor differences using 15 CPWH could justify this mesh resolution, especially as the computational effort is half the one for 20 CPWH, and more than 4 compare to 25 CPWH. Considering the small amount of difference on the main peak, 2%, the use of 10 CPWH could be legitimate, but the error in the second trough will also have to be accounted for. Depending on the level of accuracy and the affordable computational effort, the CFD user can adapt the CPWH for its specific case.

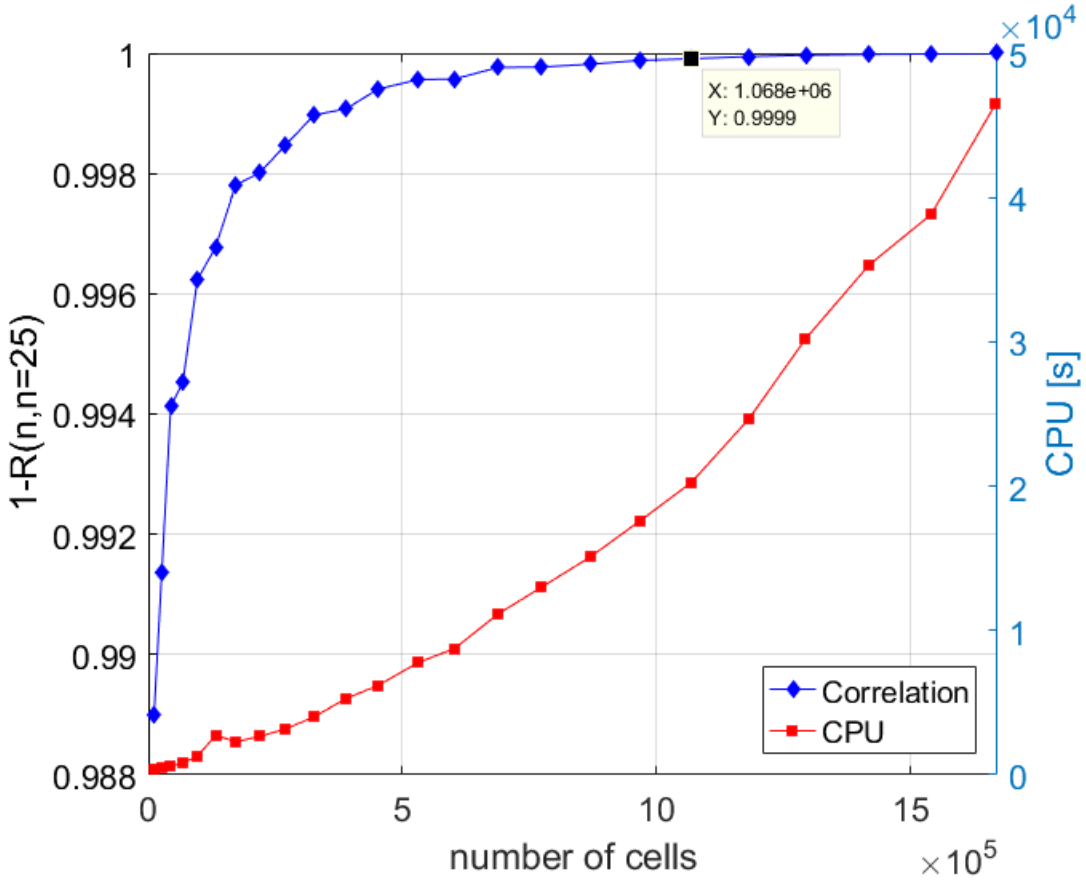


Figure 3.14: Assessment of the number of cells per wave height (CPWH) for the mesh independence considering accuracy and computational effort

As this convergence study is obtained using the least steep event, and in the light of applying the number of CPWH to events of higher complexity, the most restrictive tolerance (0.9999) is chosen, which corresponds to 20 CPWH.

Figure 3.15 presents the accuracy of the numerical scheme in the estimation of the wave-height, H. The error is calculated relative to the 25 CPWH solution. The error decreases as the mesh gets finer following a second order (quadratic, black lines on figure) rate of convergence. The convergence issue previously identified appears as well on Figure 3.15 for low cell size values ($< 10^{-2}$).

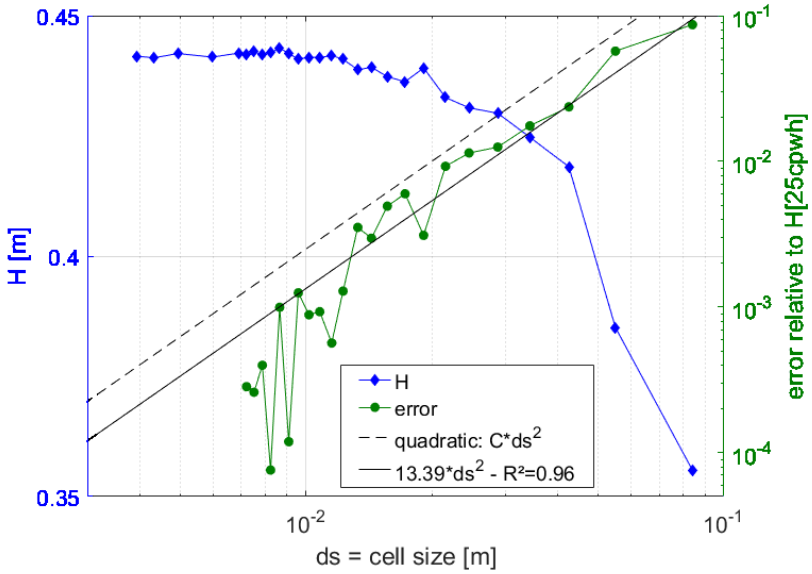


Figure 3.15: Accuracy and convergence rate of the numerical scheme based on the wave-height H

3.4.3 Numerical Beach Length

The selection of the appropriate beach length is done by comparing NWTs of increasing length. The beach starts at the end of the working section, previously defined at $x = 7\,m$, and runs until the end of the NWT. Simulations are $20\,s$ long to assess the effect of reflections on short NWT compared to the $60\,m$-NWT. The mesh is generated for each NWT according to the number of CPWH previously defined, 20.

The surface elevations of the NWT of increasing numerical beach length, and therefore total length, are plotted in Figure 3.16, where the legend is for NWT total length, and the time is normed by focus time.

Figure 3.16 shows that as the relaxation-zones increase, solutions work towards the $60\,m$ one (black line). The $10\,m$ ($3\,m$ relaxation (magenta line)) long NWT and $15\,m$ ($7\,m$ relaxation (blue line)) presents major differences with the reference, where the main crest is particularly under-estimated for the first, and both solutions are significantly affected by reflections after the main crest. $20\,m$ ($13\,m$ relaxation (green line)) long NWT and $25\,m$ ($18\,m$ relaxation (red line)) shows the improvement in the solution, where the

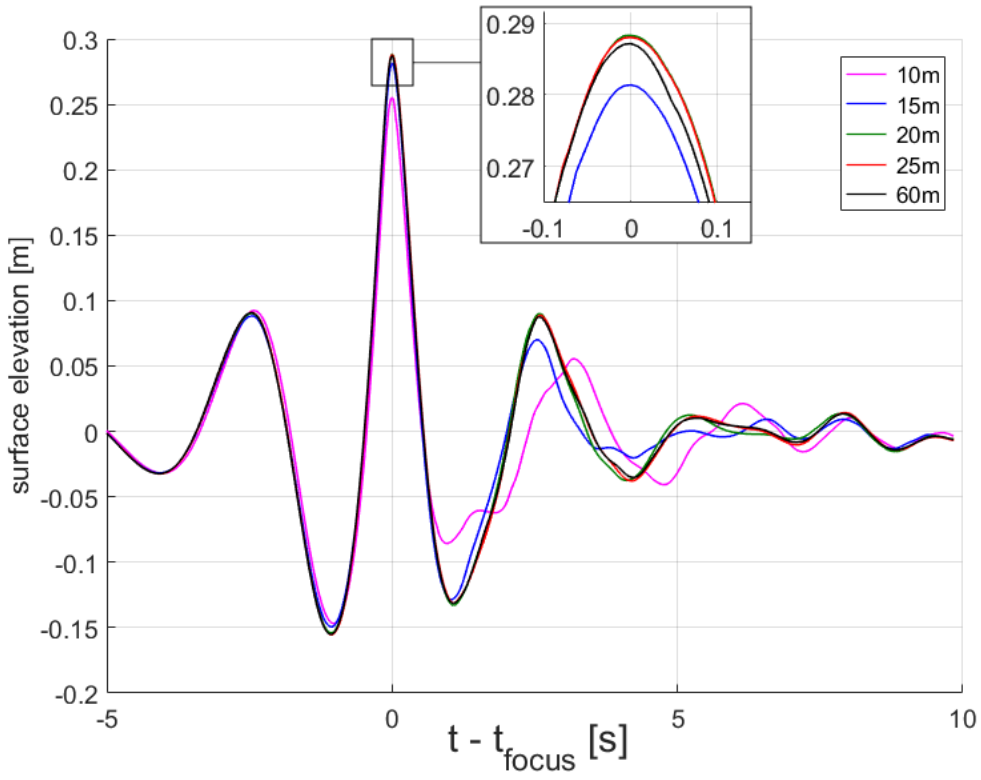


Figure 3.16: Surface elevation as a function of time for different NWT length

wave reflections are mostly not affecting the solution.

Figure 3.17 left axis shows the accuracy of each solution via the correlation with the reference as a function of the length of the NWT. Figure 3.17 right axis shows that the computational effort increases linearly as a function of the length of the NWT by slightly less than 10% per metre; nearly an additional hour per meter.

In Figure 3.17 left axis, the correlation is found to hit a plateau around $25\,m$ for a correlation above 0.9998, where a $45\,m$ NWT is required to attain 0.9999. It proves that only minor improvements to the solution can be obtained by increasing the numerical beach. So, the gain in accuracy to achieve the 0.9999 criterion on the correlation appears to require a significantly long NWT, which is unaffordable and unrealistic.

Therefore, the level of accuracy is reduced to 0.999, which is first considered attained by the $20\,m$ long NWT (0.9989). As shown by Figure 3.16, the difference between a $20\,m$ and a $25\,m$ NWT is negligible around the main peak, the previous and following trough;

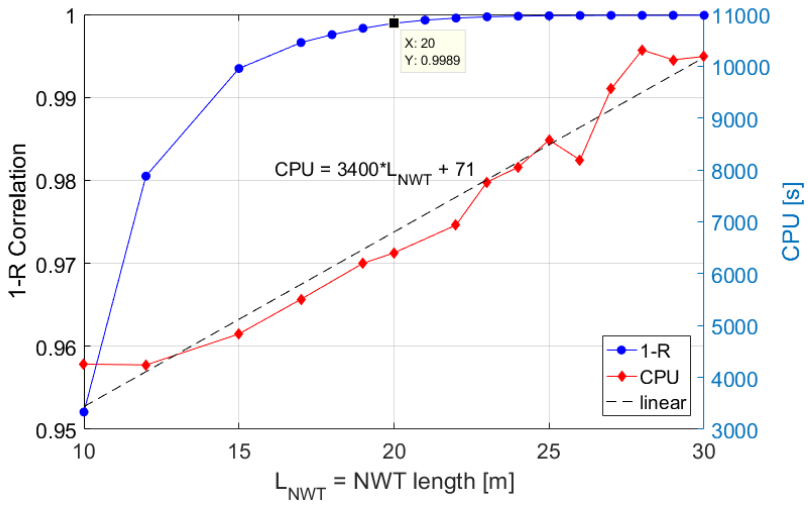


Figure 3.17: Assessment of the length of the relaxation-zone considering accuracy and computational effort

and only appears in after the event. So the gain in accuracy that might be obtained using a $21\,m$, or more, NWT are considered negligible, especially in light for the future $3D$ development where the concern will be over the main event. So, the $20\,m$ $2D$-NWT, presented in Figure 3.18, is fixed for the rest of the present study.

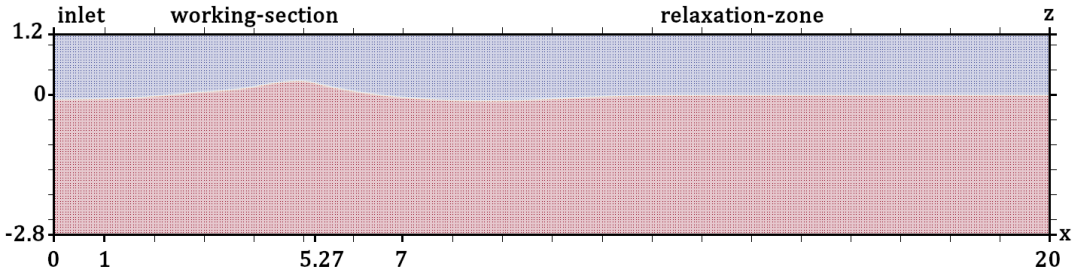


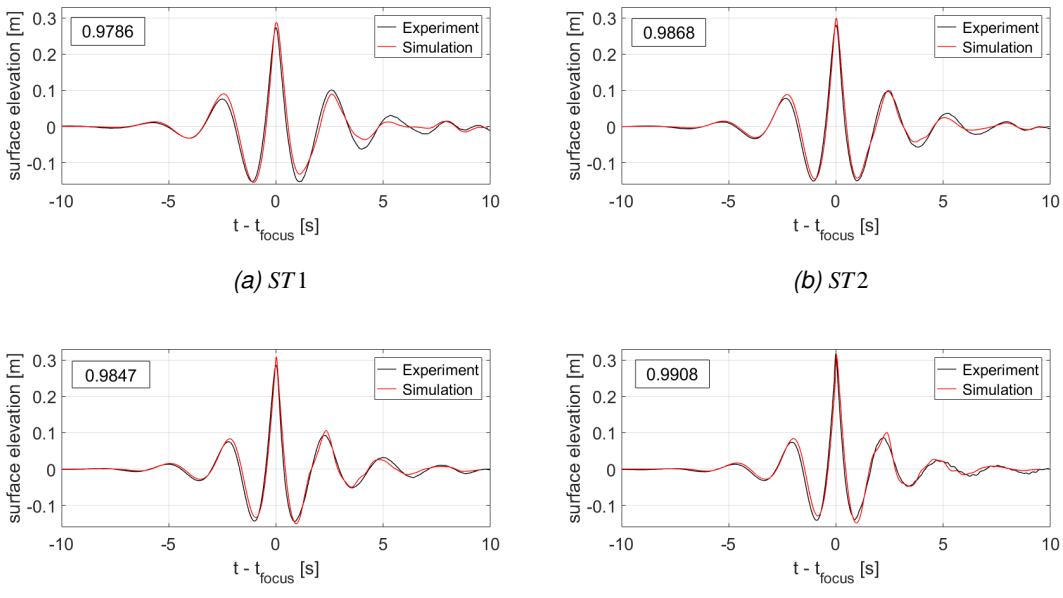
Figure 3.18: The $20\,m$ $2D$-NWT

So, the relaxation-zone is $13\,m$ long, which might appear significantly long. However, selecting the relaxation-zone as the largest wavelength among the linear components imposed at the inlet, evaluates a $105\,m$ relaxation-zone, while using the wavelength of the largest wave height, evaluates a $12\,m$ one.

3.5 Validation with Wave-Only Experiments

The $20m$ $2D$-NWT is validated against the experiments of the four extreme events detailed in 2.4.3, $ST1$, $ST2$, $ST3$ and $ST4$. The mesh is adapted for each simulation to respect 20 CPWH. At the inlet is imposed the linear superposition obtained for a 0.9999 tolerance over WG#1 of each experiment. The objective of this investigation is to assess the accuracy of the $2D$-NWT to replicate different extreme events, which will validate the method of development of the $2D$-NWT.

Figure 3.19 shows the surface elevation time-series at the focus location for the four extreme events obtained by the $20m$ $2D$-NWT (red line) compare to WG#13 (black line). From Table 3.3, the correlation between the experiments and the simulations are 0.9786 for $ST1$, 0.9868 for $ST2$, 0.9847 for $ST3$, and 0.9908 for $ST4$. So, the overall level of agreement is stable through the four cases and higher than 0.978, which validated the $20m$ $2D$-NWT over its capacity to replicate extreme events.



(a) $ST1$ *(b)* $ST2$

(c) $ST3$ *(d)* $ST4$

Figure 3.19: Comparison between experiments and simulation for the four events of increasing steepness at the focus location: WG#13 for experiment, and $x = 5.27m$ from inlet in the NWT

Event	ST1	ST2	ST3	ST4
Correlation	0.9786	0.9868	0.9847	0.9908

Table 3.3: Accuracy of the $20m$ long NWT for extreme events

However, and unexpectedly, the best correlation is found for the steepest case, $ST4$ Figure 3.19d, and the worse for the least steep case, $ST1$ Figure 3.19a. This is believed to be due to the different focusing location between the numerical model and the experiment. The focusing location is not affected by the number of component used at the inlet - i.e. not affected by the inlet pre-processing tool; section 3.4.1 - since no differences were found for the surface-elevation at model location when using the sorting of harmonics or not in the pre-processing tool (result available in Appendix A.2.1). The pre-processing tool proves that there is no loss of low frequency components when using it. It seems that, by chance, only $ST2$ event focuses at the right position. $ST1$ focuses further down the NWT, while $ST3$ and $ST4$ focus earlier. This is appreciated by the amplitude difference between the previous and second troughs, which, when zero, the event is focused; when positive, the event has already focused, and when negative, the event will focus further down the NWT. This offset in troughs amplitude is the largest for the $ST1$ case, meaning the focus location of the $ST1$ cases appears to be the furthest away from the experiment (Figure 3.19a), which explains the lowest correlation.

The lowest correlation of the $ST1$ event is considered not to be due to the length of the relaxation-zone, nor the number of CPWH. By comparing Figure 3.16 with Figure 3.19a, it appears that a lower length of the relaxation-zone, such as $15m$ (blue line) in Figure 3.16), would result in lower accuracy of the surface elevation due to the effect of wave reflections appearing around $4s$ after the focus event. The number of CPWH was selected compared to the converge solution, so no improvement is expected from an increase in CPWH.

The error in the focusing position is also considered not to be due to the length of

the relaxation-zone because this one is responsible for the absorption of the reflected waves, not for the propagation of the wave-group. Also, the effect of waves reflections on the $ST1$ case was proven in section 3.4.2 to appear after the second trough of the main-event for the selected length of the relaxation-zone. By assuming a similar result for the steeper events, the relaxation-zone is considered responsible for inaccuracies happening $4\,s$ after the event until the end, hence not the focusing ones.

Additionally, the number of CPWH is assumed not to significantly influence the focusing location, because the selected number of CPWH is compared to a converged solution. No major differences can be seen in Figure 3.12 between a low number of CPWH, such as 5 (magenta line), and the converged solution. So, even assuming that a higher non-linear event requires a higher number of CPWH, the focus location will remain mostly unchanged.

The difference in focusing location is considered due to the difference between the physical wave-maker and its numerical representation. Figure 3.19 shows the surface elevation at the focus location made in the experiments. The PWT uses a trial and error process over the phases components of the linear superposition used at the wave-maker to obtained the focus event at this specific location (detailed in section 2.4.3); whereas, the $2D$-NWT does not. Even though the NWT inlet uses the surface-elevation obtained from the experiment after the trial and error, this description remains a linear description of a nonlinear event, which therefore induces the same error in the focusing location as in the experiment. In other words, the physical model corrects the non-linear wave-wave interaction between the components (Ning et al. 2009), while the numerical does not.

Different solutions were developed for NWT to tackle this challenge already been identified numerically (e.g. Vyzikas et al. (2014), Ransley et al. (2017)). A first approach is to move the structure to the focus location found in the NWT. This approach is excluded because it means to change the numerical set-up to fit the experiment, which seems inappropriate.

A second approach follows the idea of the correction of the inlet made in the PWT (e.g. Chaplin et al. (1996), Rapp and Melville (1990)). Vyzikas et al. (2014) or Stagonas et al. (2018) use the methodology developed by Hann et al. (2014), to extract the linear part of a focus event from its four shifted (crest, trough, negative and positive) events. The correct set of amplitudes and phases are used to replicate the focus event at the focus location. Although this method allows a very good agreement of the focusing, some errors due to the over-estimation of the second order contributions disturb the surface elevation before the focusing. Besides, this methodology is not applicable for a non-focus event, or in a blind-test development of the NWT, in which the experiment surface elevation at the position of interest (i.e. the structure or focus position) is unknown. As the present methodology of development of a NWT for extreme events is aiming to apply to other extreme events done at the same PWT, the phases of the components will not be corrected.

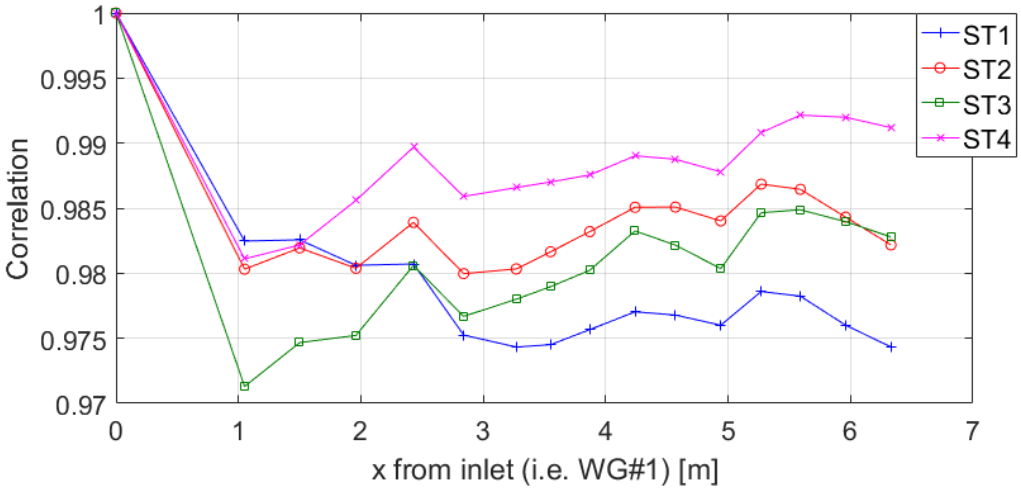


Figure 3.20: Accuracy of the solution at WG as the wave-group propagates

The error in the focusing position illustrates the issue inherent to the description of a non-linear event using a linear superposition. Figure 3.20 shows the accuracy of the solution along the NWT for the four events. Apart for the $ST1$ event, and at the inlet (i.e. $x = 0$), the accuracy increases as the wave-group travels to the end of the NWT. So, to replicate a non-linear event, the NWT needs space to build-up towards a fully

non-linear propagation of the wave. The distance between WG#1 and the position of interest needs to be large enough.

Additionally, Figure 3.20 proves that the accuracy of the present $20m$ $2D$-NWT is validated against experiments; in which it succeeded to model the extreme events within an accuracy above 0.97 independently from the non-linearity of the event, and along the length of the NWT. The decisions in the development of a $2D$-NWT are available in Appendix C.3.2.

3.6 From $2D$ to $3D$

The $20m$ $2D$-NWT is extended to $3D$, first by assuring the consistency of the flow description, then by inserting the X-MED model in the mesh while implementing the necessary changes to the solver to simulate the WSI of a moving structure.

3.6.1 Consistency of Wave-Only Events in a 3D-NWT

Mean-Water-Line Refinement

$3D$ simulations are significantly more computationally costly than $2D$ ones due to the increase in complexity of the flow and the increase in the number of cells. A refinement enveloping the free-surface is used, which maintains the advantages of square cells in the region while limiting the overall computational effort. Square cells are also maintained in the background mesh, i.e. the original mesh from which the refinement proceeds. The mesh refinement is performed by the `snappyHexMesh` utility, a mesh generator that allows refinement and chiselling of geometry from an existing background mesh.

The $2D$-NWT becomes $3D$ as the Mean Water Line (MWL) is refined considering the minima and maxima in amplitude attained by the surface elevation. NWT width is $0.1/,m$. The `noSlip` boundary condition remains applied on the side walls. The MWL refinement is required to be along the entire NWT length, as shorter refinements were found to converge slower while influencing the solutions. This connects to the reason to use square cells discussed in section 3.2.

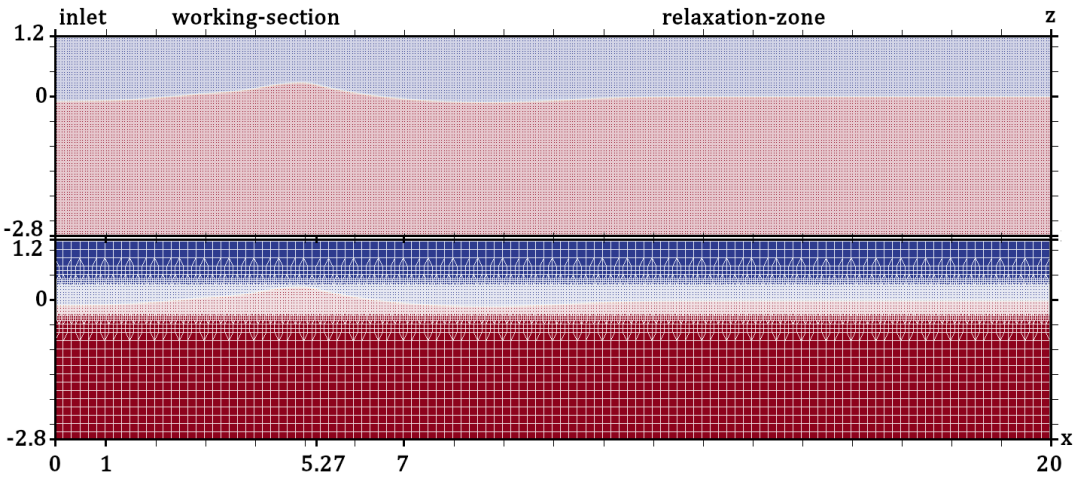


Figure 3.21: From $2D$ mesh to a $3D$ one, apply to the $ST1$ case

Outside of the refined area, the mesh is 3 levels coarser. This means that the cells are split in half three times between the background to the MWL using the octree refinement strategy (Greaves and Borthwick 1999). The mesh is adapted for each event, where the mesh-size of the refined region is set according to 20 CPWH (section 3.4.2). The level 3 is found to have a negligible influence on the solution of the four events when is used, at least, one cell more for both maxima and minima (result available in Appendix A.2.2).

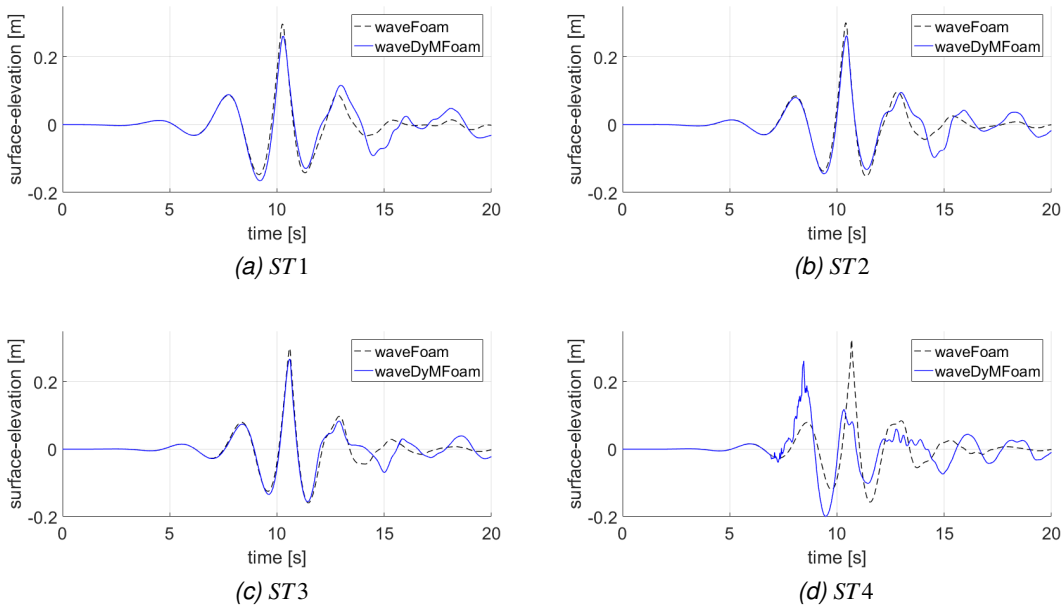
A fully squared mesh of a $1\,m$ wide NWT has $8X10^6$ cells, whereas using the refinement of the MWL for the $ST1$ event, the mesh has $1,3X10^5$ cells, hence a 98% decrease. Figure 3.21 shows the two meshes.

Level 4 of refinement is not used for the MWL because the background mesh becomes too coarse. As shown by Figure 3.21, a level 4 of refinement for the MWL will make the top part of the air-phase of only 1 cell.

Consistency of the Flow

The $3D$ solver later used to handle the motion of the X-MED buoy in the mesh, `waveDyMFoam` (detailed in section 3.1.6), is found to be inconsistent with the $2D$ solutions. Figure 3.22 shows the surface elevation obtained for the four focus events for a static $2D$

simulation solved by `waveFoam` (black dashed line), compared to a $3D$ one solved by `waveDyMFoam` (blue line) where the mesh remains undeformed. The inconsistency of the flow appears to be related to the non-linearity of the event.

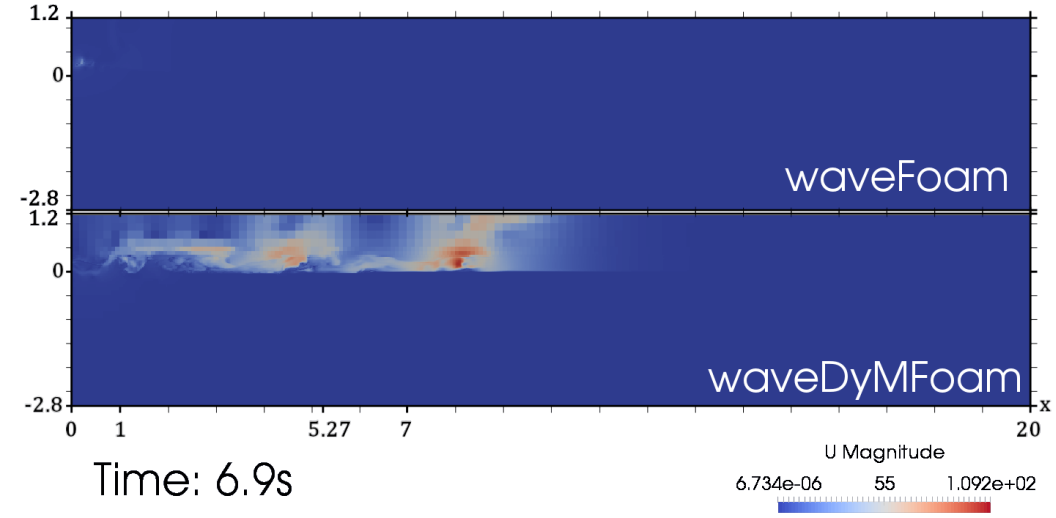


(a) $ST1$ *(b)* $ST2$ *(c)* $ST3$ *(d)* $ST4$

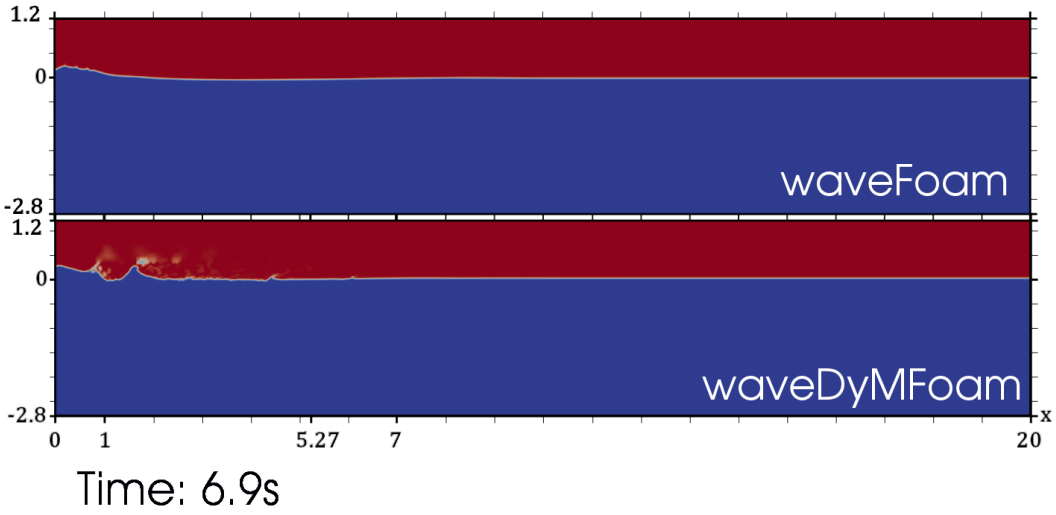
Figure 3.22: Inconsistency of the use of `waveDyMFoam` solver to replicate $2D$ wave-only cases obtained with `waveFoam`

This error is due to abnormal air flows velocities appearing in the inlet and propagating towards the outlet due to spurious air flows. Figure 3.23a shows these air flows for the steepest case $ST4$, where at the considered time-step the maximum flows speed is more than a $100 m/s$. It seems that these unrealistic values remain within the air-phase. However, the consequence of the surface elevation is significant (Figure 3.23b).

As the time-step of the simulation is ruled by the Courant number $C0$ (Equation 3.5 - section 3.1.2), the large velocities in the air flow bring the time-step down. Therefore, this limits drastically the progression of the simulation, slowing it down massively for necessarily wrong results; as it seems legitimate to assume that no such air flows are obtained in PWT laboratory.



(a) Velocity



(b) alpha.water

Figure 3.23: Unrealistic air-flows appearing in $3D$ dynamic simulation solved with `waveDyMFoam` for the $ST4$ wave-only event
views of the mesh obtained in post-processing using ParaView

These spurious air flows are believed to be intrinsic to the VoF method, as previously mentioned in section 3.1.3. Recent developments have been given much attention to tackle this challenge of the interface sharpening, such as the Ghost-Fluid method (IsoAdvector) (Pedersen et al. 2017) or the interface-compression technique (MULES) used in the present study. Changes of numerical schemes (in `fvSchemes`) might be

used to limit these numerical instabilities, however the air flows will remain as intrinsic to the VoF method. These different solutions are not investigated here and are left to future work.

In the description of the `waves2Foam` toolbox, Jacobsen et al. (2012) identifies the restrictive effect of air velocities on the Courant number in $2D$ cases performed by `waveFoam` solver, but not to such an extent with consequences on the accuracy of the solution. As the air phase is often considered irrelevant to many WSI, Liu and García (2008) found accurate to fix the air velocities to zero and each time-step in the study of scouring on the sea-bed over one wavelength. Jacobsen et al. (2012) insist that such a technique dissipates too widely the wave energy over multiple wave length (i.e. as the wave travels). The present study considers this approach because only the propagation of one event is modelled, restricting the simulation to a short one.

A similar method, developed by Brown (2017), is used. It sets to zero the flow velocity of a given cell, without consideration of the volume fraction, if this one is above a threshold value, $ULim$. A value too low will affect the water velocities, while a value too large will allow the generation of the abnormal air flows. This solution, however also non-physical, as violating the conservation of momentum, actually increases the accuracy of the simulation as it avoids the creation of these unrealistic air-flows, hence improving the consistency of the passage to $3D$.

An evaluation of the threshold value is performed not to influence the water phase of the flow. A minimum of the maximum velocity of the water can be estimated using linear theory, which defines the maximum velocity of a fluid particle as:

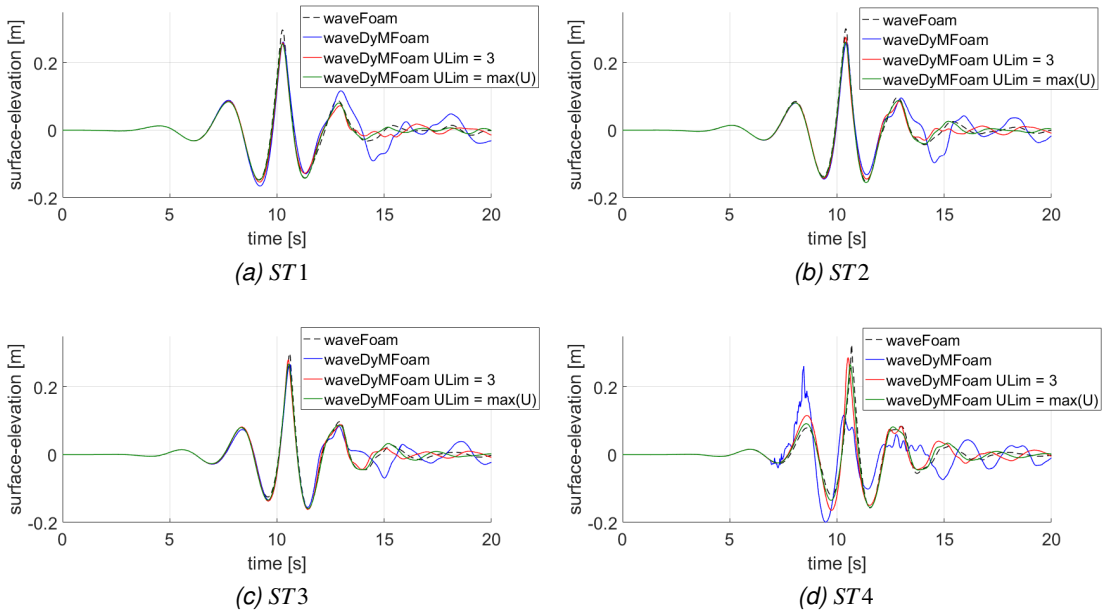
$$max(U) = \sqrt{g.k.tanh(k.d)} \tag{3.20}$$

where U is the velocity magnitude given by linear theory (other variable were defined in 2.1.3). Table 3.4 presents the maxima for each event.

Case	ST1	ST2	ST3	ST4
Linear theory maximum velocity $[m/s]$	0.85	0.96	1.10	1.33

Table 3.4: Estimation of the minima of $ULim$ using the maxima given by linear theory

The assessment of $ULim$ is performed using $3D$ wave-only simulations of increasing $ULim$ values starting from the minima given by linear theory and solved by `waveDyMFoam`. It aims to achieve consistency compare to the $2D$ wave-only solutions performed using `waveFoam` solver.



(a) $ST1$ *(b)* $ST2$ *(c)* $ST3$ *(d)* $ST4$

Figure 3.24: surface elevation obtained using the solver `waveFoam` compare to `waveDyMFoam` with and without $ULim$

Figure 3.24 compares the original $2D$ solutions (black dashed line), with the unlimited solution (blue line), a solution limited by the maximum given by linear theory (green line), and one limited by $ULim = 3m/s$ (red line). Figure 3.25 shows the accuracy of $3D$ solutions for different $ULim$ values compared to the $2D$ reference one. The computational effort of the least steep case is also added to the figure to illustrate its drastic increase, which, however remains the smallest of the four events.

Figures 3.24 and 3.25 show that $3D$ solutions converge toward $2D$ ones as $ULim$ value

decreases. Highest correlations are obtained for $ULim$ equal to the maximum defined by linear theory.

However, it seems inappropriate to use such values as the maximum velocity of the flow, especially because the events of interest are highly non-linear. Also, further applications to WSI are likely to generate high water velocities around the structure that needs to be captured. So, the following $3D$ study is conducted using $ULim = 3\,m/s$, chosen as a compromise, and considering a margin over the flow velocities.

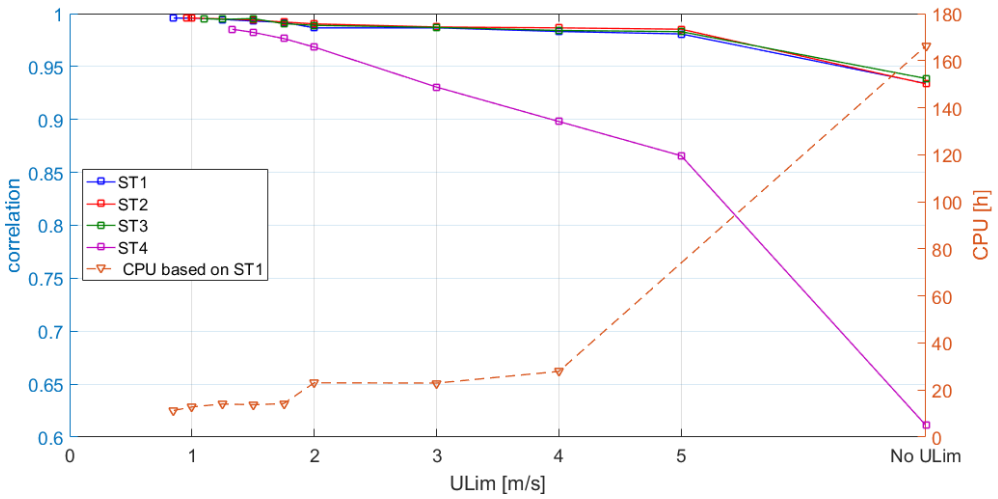


Figure 3.25: The assessment of $ULim$ using the correlation with the `waveFoam` simulation, with an additional plot of the computational effort of the $ST1$ case

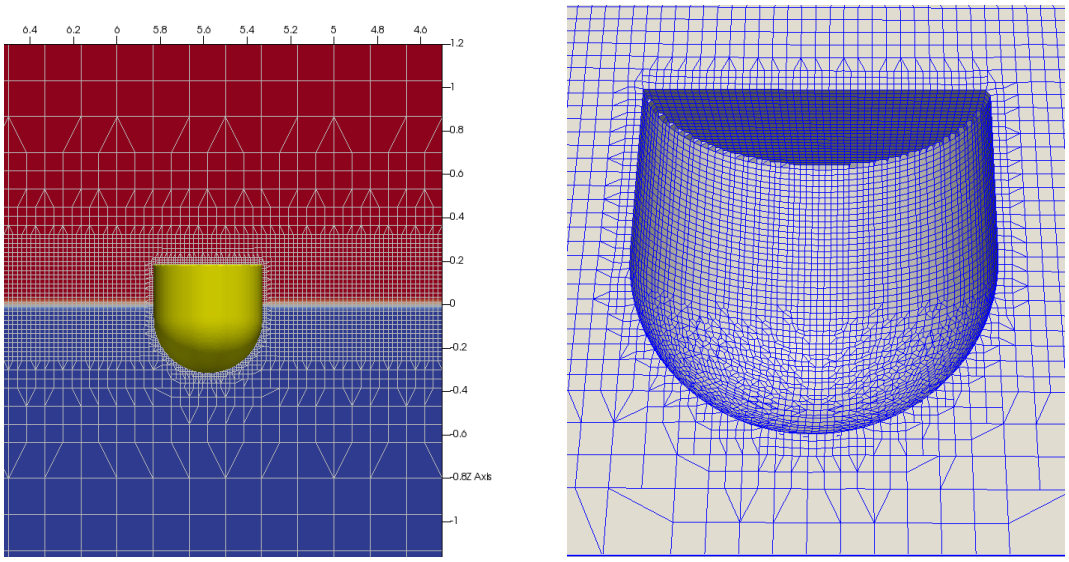
It should be noted that no differences, with or without $ULim$, were found in the time-series of the moored heave decay test (later detailed in section 3.7). This result is available in Annexe A.2.3 and is considered as the proof that the use of $ULim$ does not influence the motion responses of rigid bodies.

3.6.2 Rigid Body Motion

To account for the motion of a rigid body in WSI, three steps are required by the `rigidBodyDynamics` library selected in section 3.1.6. First, the contour (i.e. surface) of this body needs to be inserted into the mesh. Second, specific boundary conditions are specified on the surface, and the mesh is made deformable to account for the motion

(i.e. mesh-motion). Third, the motion of the body is calculated by the RBM solver (section 3.1.6) as a consequence of the hydrodynamic loads and potential others loads linked to the system (e.g. mooring).

Inserting the X-MED in the Mesh



(a) Buoy inserted in refined mesh

(b) Contour cut out of backgorund mesh

Figure 3.26: Mesh: refinements up to level 3 for the MWL, and 4 for the X-MED buoy

The outside surface of the X-MED buoy is generated by the Computer Aided Design (CAD) model SALOME (Ribes and Caremoli 2007) in a geometry file (.stl). `snappyHexMesh` chisels the surface from the background refined mesh to level 4, hence one level higher than the MWL section. The sharp top edge of the X-MED is purposely not accurately represented to accelerate the generation of the mesh while avoiding sharp edges that are likely to generate significant flow accelerations (Davidson et al. 2015). This approximation is believed to not influence the motion response as this edge remains in the air, the total loss of volume is negligible, and the rounded surface of the buoy is accurately represented. The mesh is shown in Figures 3.26a, and 3.26b.

To save computational effort, and because the events are unidirectional, a symmetry

condition is applied at the xOz plane, which cuts the domain and buoy in half along the $3D$-NWT length. The DoFs of the buoy are restricted to surge, heave and pitch.

Rigid Body Motion of the X-MED

The floating motion of the X-MED model is represented (Figure 3.27) by two prismatic joints (Px for surge, and Pz for heave) plus a revolute joint for pitch (Ry). The sum of those three joints is made via a `composite` joint, which is the 3-DoF symmetry equivalent to the 6-DoF `floating` joint.

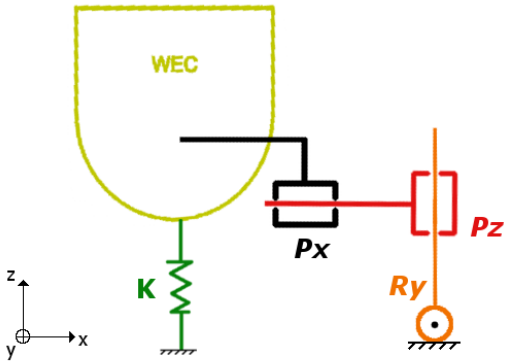


Figure 3.27: Rigid-body motion model of the X-MED buoy for the `rigidBodyDynamics` solver in a symmetric case - Px is a prismatic joint for surge, Pz for heave, and Ry is a revolute joint for pitch, K is a linear spring modelling the mooring line

If required, the mooring line of the X-MED model is modelled using a linear spring condition (`linearSpring` via the restrain `restraints`) on the distance between the bottom of the buoy and its anchor position on the sea-bed. This simplification allows not to model the line itself which will require a specific mesh and avoid the modelling of the complex flows happening around lines (e.g. drag on a riser line). It is justified as the mooring is constantly tensed through all events.

Only the spring of the actual mooring line is modelled because the Dyneema rope has a much bigger stiffness, hence a negligible extension. So, in a non-symmetric set-up, the linear spring of the model has a $66.3 N/m$ stiffness, and a $2.18 m$ rest length manually calculated to fit the law of Hooke on linear springs when sinking the X-MED

buoy from its unmoored to its moored position (Mooring pre-load is $-20.5N$ - Table 2.2 section 2.4.2). This values are confirmed by the initial times of simulations where the moored model is at rest.

Decisions for the rigid-body motion of the X-MED buoy are in Appendixes C.11 and C.13.

Boundary Conditions on the X-MED

To account the effects of the fluids on the structure, while considering its motion, the boundary conditions on the shell need to specified. As the structure is considered here as perfectly rigid, it acts as a wall. So, as for the NWT sides, `fixedFluxPressure` and `zeroGradient` conditions are respectively applied for the pressure and phase ratio fields.

The velocity condition requires to correct the flux due to the mesh motion while certifying that the total flux through the surface is still zero. This is done by the `movingWallVelocity` condition, which is an adaptation for moving meshes of the `fixedValue` condition used on walls (section 3.1.5).

Because the mesh is deformable, the nodes locations are changing for each time-step. The field `pointDisplacement` records the nodes locations in the global frame. All boundaries are specified with a zero `fixedValue` boundary condition for this field, as no displacement is forced into the simulation at the initial condition.

3.7 Methodology: Assessment of WSI for a Floating Rigid Body in a 3D-NWT

The flow has been proven consistent with $2D$ solutions, hence assuring the modelling of extreme events of increasing steepness. The RBM solver has been implemented in the flow solver to account for the floating motion of the X-MED buoy, and its consequence on the mesh. However, the width of the $3D$-NWT remains undefined, and the $3D$-NWT needs to be assessed over the modelling of the motion responses of the X-MED buoy under WSI, done in two steps.

First, the RBM solver is assessed over the reproduction of experimental decay tests to evaluate the capacity of the numerical model ($3D$-NWT) to reproduce Fluid Structure Interaction (FSI) while avoiding the complexity of wave. The assessment concerns the resonance frequencies of each DoFs, and the time-series (if available) to evaluate the decay of the motion.

Second, the least steep event is applied to the $3D$-NWT to assess the modelling of the motion responses of the X-MED buoy under an extreme event; a proof-of-concept. Finally, the $3D$-NWT can be validated over the four extreme events of increasing steepness.

3.7.1 Decay Tests

The specifications of the decay tests are reported in Table 3.5. Only the time-series of the moored and unmoored heave decay tests are available. The surge decay test is not conducted in the $3D$-NWT because the resonance period is too large ($14\,s$), which will make the simulation challenging as this slow motion is highly subjected to numerical errors. The X-MED buoy has no velocity at the initial time of all decay tests. Resonance frequencies are evaluated from the first oscillation of the response time-series.

Ransley (2015) reproduces the moored and unmoored heave decay experimental tests of the X-MED buoy using a square $3D$-NWT. Despite the use of a circular relaxation-zone on the sides, the width of the NWT was found of major influence on the heave responses.

	Heave		Pitch
	Moored	Unmoored	Moored
Resonance frequencies $[hz]$	0.917	0.926	0.75
Release offset from equilibrium	$0.204\,m$	$0.186\,m$	12.5°

Table 3.5: Specification of the moored and unmoored decay tests - pitch decay release position is chosen arbitrary as no time-series are available

Therefore, the moored heave decay tests are conducted in NWTs of increasing width

(e.g. $4m$ Figure 3.28) with no relaxation-zones on side walls to evaluate reflections due to radiated waves and assess the width of the NWT. A simulation of reference is performed using the NWT to the width of the COAST, $7.75m$. Solutions are expected to converge to this reference as the width increases. The moored heave decay test is chosen because the time-series is available and later WSI concern the moored X-MED.

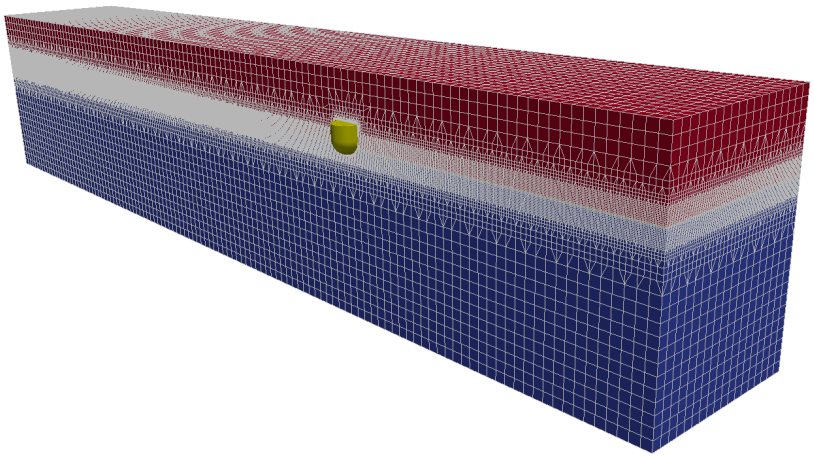


Figure 3.28: $20m$ long by $4m$ wide $3D$-NWT used for the heave decay test

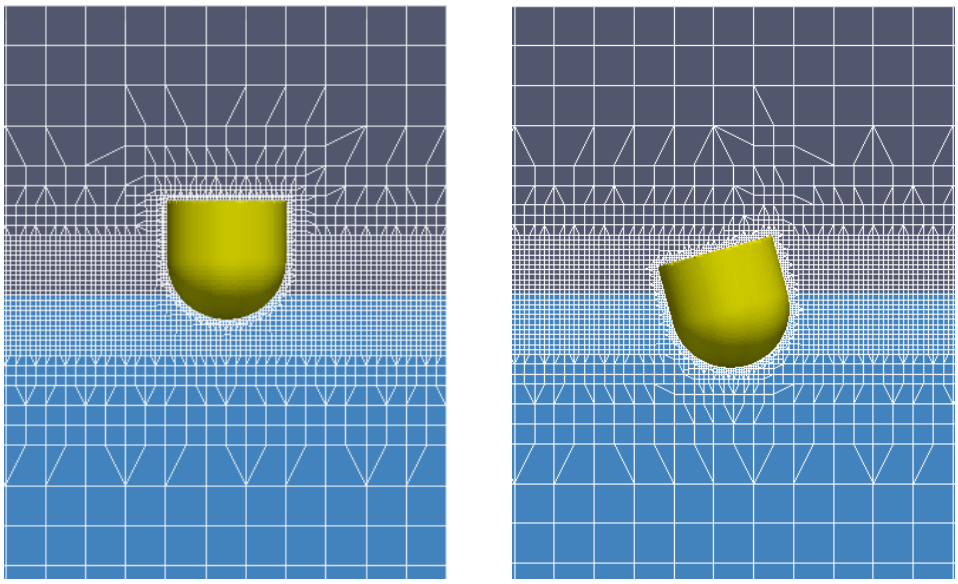
The unmoored heave decay is only performed using the width of the COAST to obtain the least disturbances from radiated waves. The moored pitch decay is done in a $4m$ wide NWT using 12.5° initial angle (arbitrarily selected due to the absence of time-series) as only the first oscillation is of interest to determine the resonance frequency.

Set-up	(x, y, z) position
Equilibrium Moored	$(5.58, 0.0, -0.133)$
Equilibrium Unmoored	$(5.58, 0.0, -0.122)$
Moored heave decay	$(5.58, 0.0, 0.071)$
Unmoored heave decay	$(5.58, 0.0, 0.064)$
Moored pitch decay	$(5.58, 0.0, -0.133)$

Table 3.6: Buoy centre of rotation (equivalent to centre of gravity), `centreOfRotation`, for the different sets-up

To consider each release positions - moored, unmoored, heave or pitch - a new ge-

ometry is designed in SALOME CAD model and inserted in a $3D$-NWT with the MWL refinement. Figures 3.29a and 3.29b show a section of the mesh generated for the moored heave and pitch decay tests. For each test, the initial position of the centre of gravity (equal for the X-MED to the centre of rotation - `centreOfRotation`) is changed in the `dynamicMeshDict` according to table 3.6. Decay tests are considering all 3-DoF of the buoy (i.e. not restricted to a single DoF).



(a) Moored heave decay *(b)* Moored pitch decay

Figure 3.29: Section of the meshes of the moored heave and pitch decay tests

All $3D$ simulations following were realised using 16 cores in parallel on the cluster of the University of Plymouth (detailed in section 3.3.4 Fotcluster2).

3.8 Validation of the 3D-NWT for Extreme WSI

The $3D$-NWT shown in Figure 3.28 is validated against experimental decay tests to validate the RBM solver and assess the width of the NWT, before being applied to extreme WSI.

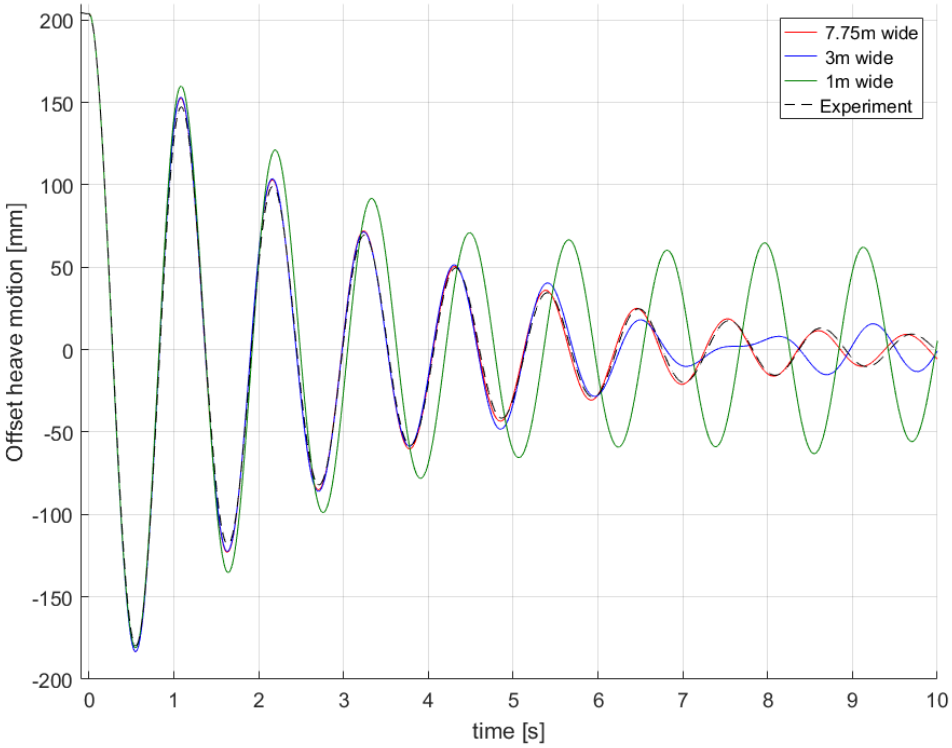


Figure 3.30: Heave responses time-series of the moored heave decay test

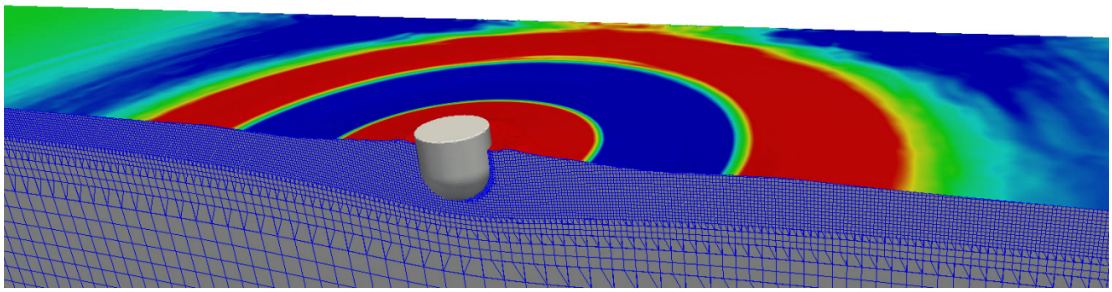
3.8.1 Heave Decay

Moored

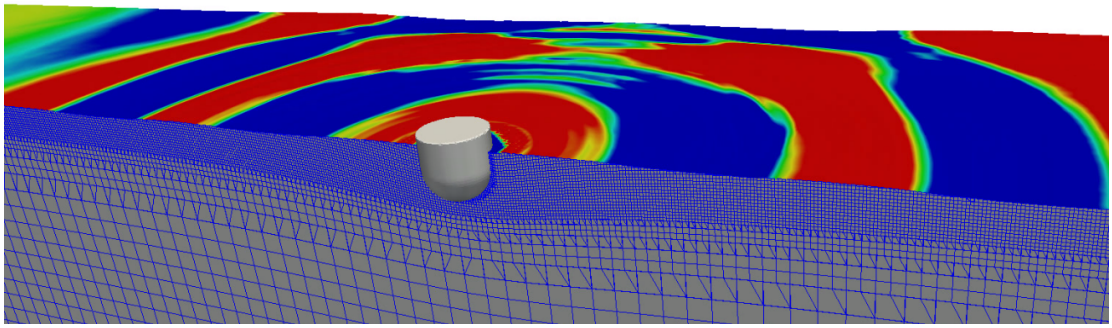
Figure 3.30 shows the offset from equilibrium of the heave responses time-series for the moored heave decay test. The experiment (black dashed line) compares with a $1\,m$ wide $3D$-NWT (green line), a $3\,m$ one (blue line), and the NWT to the width of the COAST (red line).

The $3D$-NWTs are in agreement with the experiment ($f = 0.917\,Hz$) for the evaluation of the moored heave resonance frequency, $f = 0.923\,Hz$, hence a $0.006\,Hz$ error. Indeed, the resonance frequency is independent of the width because the differences in the period of the first oscillation are negligible. A $0.001\,Hz$ difference is found between $1\,m$ and $7.75\,m$, and less for wider NWTs.

However, and as expected, the influence of the width on the decay is significant. The



(a) Heave decay test in the $3\,m$ wide NWT at $2.5\,s$



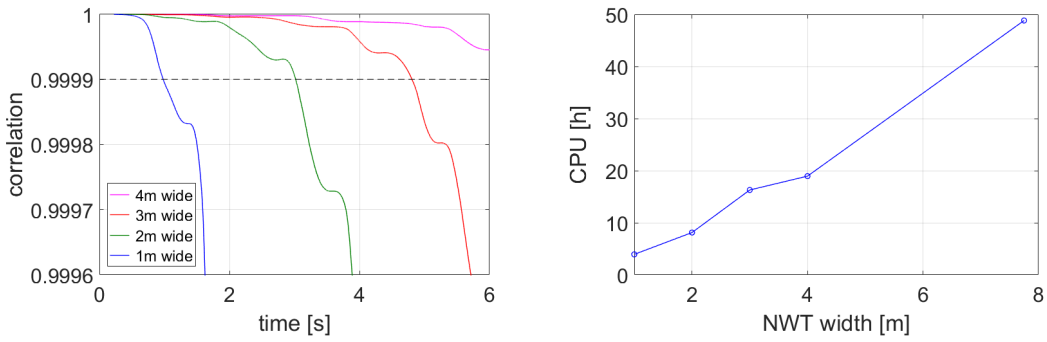
(b) Heave decay test in the $3\,m$ wide NWT at $4.8\,s$

Figure 3.31: The generation (a) and reflection (b) of radiated waves due to the heave motion of the buoy - the mesh was cut for $\alpha = 0.5$, the rainbow scale shows the surface elevation from troughs in blue to peaks in red

disturbance appears already at $1\,s$ for the $1\,m$ NWT, which probably influence very lightly the resonance frequency. The $3\,m$ wide NWT shows an important improvement where the decay diverges around $5\,s$ and is significantly disturb after. The best agreement is obtained for the NWT to the width of the PWT, which reproduces the decay almost perfectly.

The accuracy of the decay is due to the reflection of radiated waves generated by the drop on the side wall of the NWTs. Radiated waves are shown in Figure 3.31a for the $3\,m$ NWT, where the rainbows colours scale shows a difference in the surface elevation from the trough in blue, to the crest in red. Reflection can be seen at time-step $4.8\,s$ in Figure 3.31b. For the $1\,m$ NWT, the buoy probably resonates with the reflected radiated waves after $4\,s$. The effect is not so significant for the $3\,m$ wide NWT, and probably not present in the widest NWT because the waves have not reached the buoy yet.

So, the responses of shorter NWTs diverge from the one of the widest NWT as radiated



(a) Accuracy *(b)* Computational effort for first $6\,s$

Figure 3.32: Correlation between a $X\,m$ wide NWT and the $7.75\,m$ one as a function of time - 0.9999 tolerance is highlight as (black line) - computational effort

waves affect the decay. To illustrate this, the correlation between the solution of an $X\,m$ wide NWT compare to the $7.75\,m$ wide NWT is calculated for time-series of increasing length, from the initial time-step and up to $6\,s$. Figure 3.32a shows the divergence using the 0.9999 tolerance (black line), where the $1\,m$ NWT (blue line) drops out after $1\,s$, and $3\,m$ NWT (red line) before $5\,s$.

The increase in width, hence the increase in accuracy of the heave decay, is at the price of computational effort, as shown by Figure 3.32b. The number of cells increases by $1,2X10^6$ at each increase of a metre in width. The $1\,m$ wide NWT has $1,2X10^6$ cells, and the full width NWT has more than 9 million cells.

From the divergence and the computational effort, a first assessment of the width can probably be estimated. However, the heave decay test is very specific due to the generation of radiated waves, which are expected to be significantly larger than the ones potentially happening in future WSI. Therefore, the assessment of the width is postponed to the validation of extreme WSI later done with the least steep event.

Unmoored

The unmoored heave decay test is conducted using only the widest tank to avoid radiated waves reflections on side walls. Figure 3.33 shows the experimental time-series (black dashed line) compare with the numerical one (blue line).

The accuracy of both the assessment of the resonance frequency and the decay time-series is less accurate than the moored case. The resonance frequency is found lower as $f = 0.907 Hz$ for a $0.926 Hz$, hence a $0.02 Hz$ error. This error affects the decay time-series, which shows generally good agreement in amplitude, but where the difference increases with oscillations due to this offset in resonance frequencies.

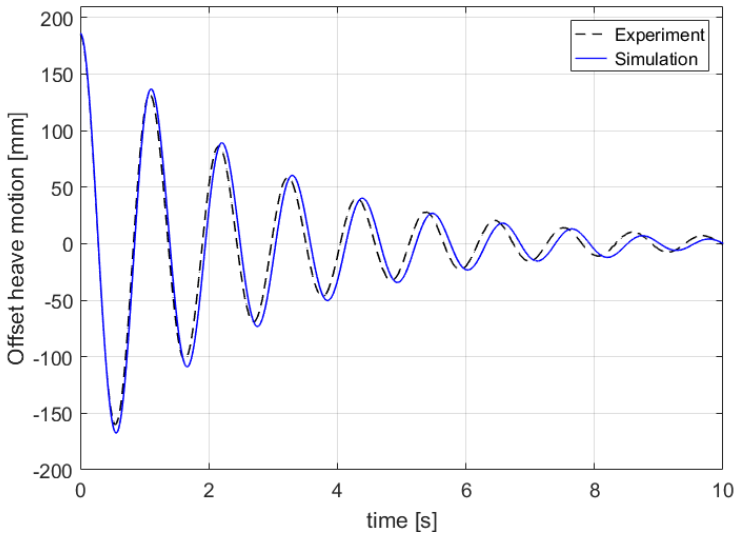


Figure 3.33: Heave responses time-series of the unmoored heave decay test

The difference in accuracy compared to the moored case is due to the presence of the mooring line, which mostly dictates the heave motion. However, due to the level of accuracy obtained in the moored heave decay, no further investigations are conducted in the present study to improve the quality of the unmoored case.

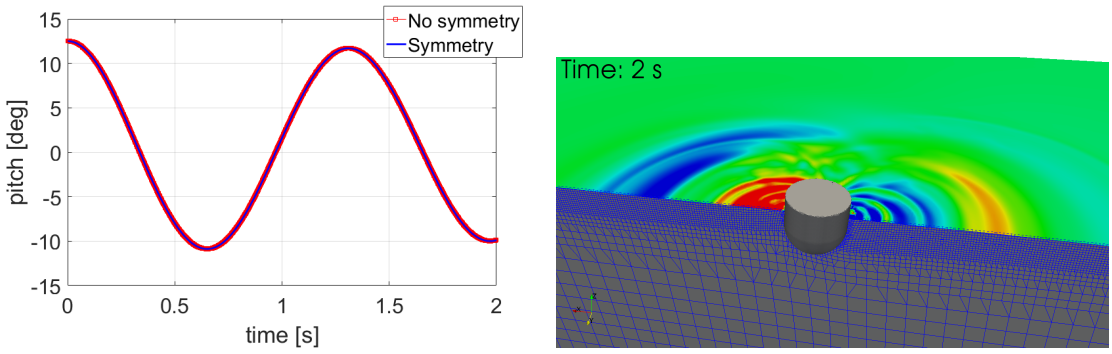
The level of accuracy obtained for both heave decay tests seems higher than the one obtained by Ransley (2015). This improvement is considered due to the use of a NWT of the size of the COAST PWT.

3.8.2 Pitch Decay

Due to the absence of the time-series, the pitch decay test is assessed over the response frequency only, and the initial pitch angle is arbitrary set to 12.5° in a $4 m$ wide NWT (Figure 3.34b). Figure 3.34a shows the pitch responses of a non-symmetrical

simulation (red line with squares) compare with a symmetrical set-up (blue line), which requires half, inertia, mass and mooring stiffness.

The pitch moored resonance frequency is $0.76Hz$ independently from the use or not of the symmetry condition, which is considered as a good agreement with the experimental measure, $f = 0.75Hz$ (Table 2.3).



(a) Pitch response

(b) $4m$ wide symmetrical NWT at $2s$

Figure 3.34: Pitch resulting motion and mesh for the pitch decay test in a $4m$ wide symmetrical and not NWT

3.8.3 Proof-of-Concept

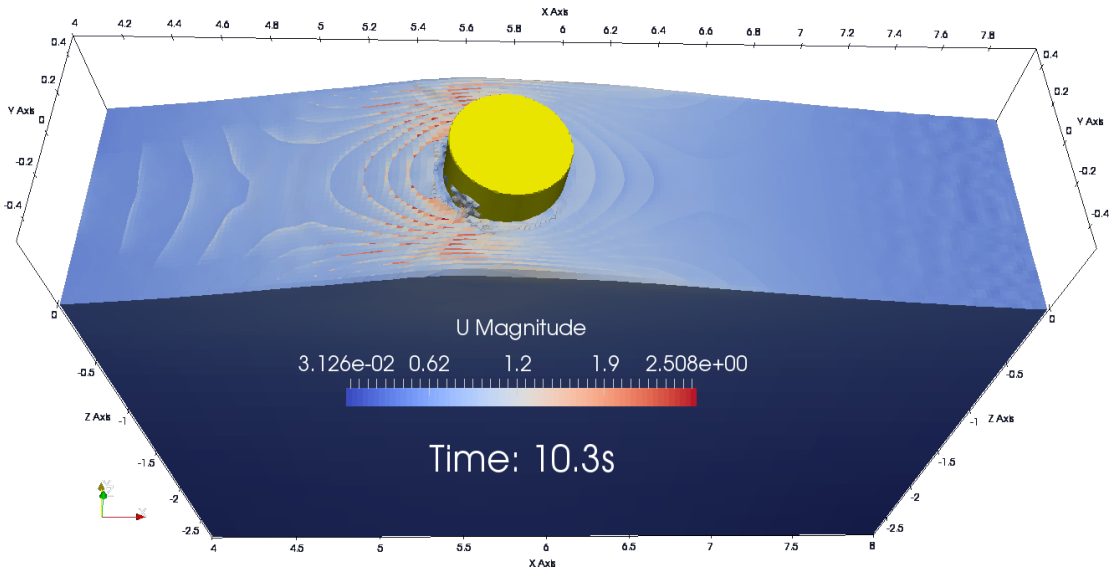


Figure 3.35: Apparition of high flow velocities near the X-MED buoy for a NWT twice wide as the radius of the buoy, considered as responsible for the instability of the simulation – view obtained in ParaView for a non-symmetric case via a cut of the domain at $\alpha = 0.5$, a second cut at $x = 4a,m$ and a third at $x = 8\,m$

The $3D$-NWT being validated against the decay tests, the validation is extended to extreme WSI. The least steep event $ST1$, obtained after the limitation of the flow for $3D$ cases in section 3.6.1, is used as believed to be the less challenging to simulate, because the non-linearity of an event is expected to generate instabilities in the motion responses. The assessment of the width of the NWT is also conducted as the heave decay test is found not relevant for the assessment. The width corresponds to the width of the NWT for a symmetrical case.

The only unstable simulation is found for a $0.5m$ wide NWT, hence twice the radius of the X-MED buoy. This proves that the $3D$-NWT can simulate extreme WSI using a NWT wider than three times the characteristic length of the structure. The instability of the $0.5m$ simulation is due to the apparition of high flow velocities near the X-MED buoy, as shown in Figure 3.35.

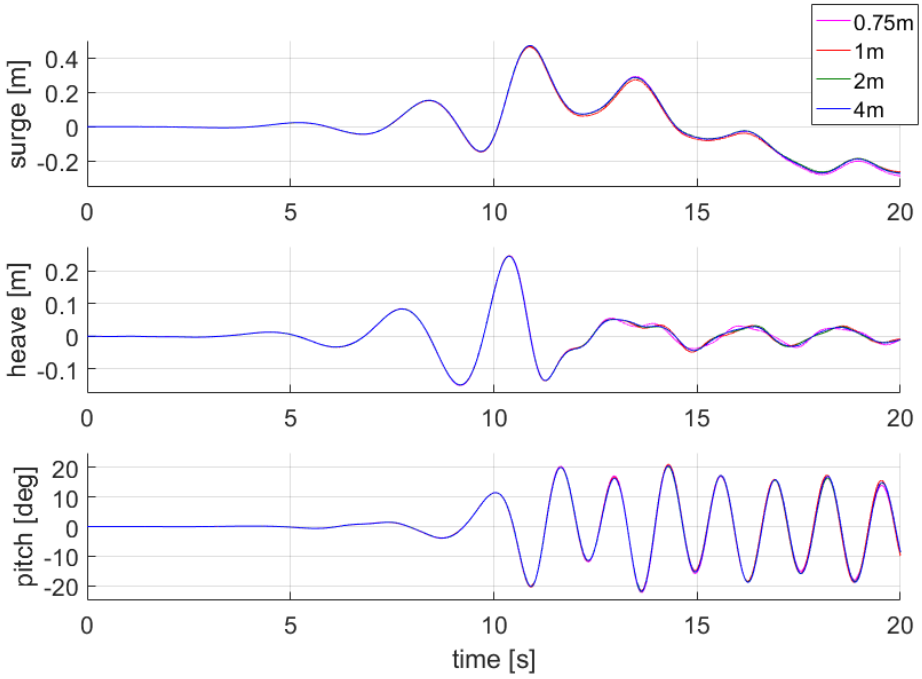


Figure 3.36: Comparison of the 3-DoF of the buoy for NWT of increasing width using the least steep event $ST1$

Figure 3.36 shows the motion responses of the buoy in the 3-DoF, surge, heave and pitch. Solutions of four NWTs of increasing width are presented - $0.75m$ (magenta line), $1,m$ (red line), $2m$ (green line), and $4m$ (blue line).

Figure 3.36 shows that motion responses have minor differences for NWTs of increasing width. As previously proven in the heave decay test (Figure 3.30), a wide NWT is necessary for cases involving large radiated waves. The extreme WSI of concern here is not generating large radiated waves because the overall motion of the buoy follows the wave (i.e. no drop). Also, the events are unidirectional, so the amplitude of the wave-groups are expected to predominate on potential radiated waves. Therefore, the width of the $3D$-NWT is assessed from the convergence of the motion responses.

Figure 3.37 quantifies the convergence as a function of the width for surge (blue line with squares), heave (red line with squares) and pitch response (green line with squares). The $4m$ wide NWT is used as the reference it has proven in the heave decay test to allow more than a $6s$ period before being affected by radiated waves (Figure 3.32a). The computational effort (magenta line with upside-down triangles) of each NWT are added to the assessment of the width to identify a compromise.

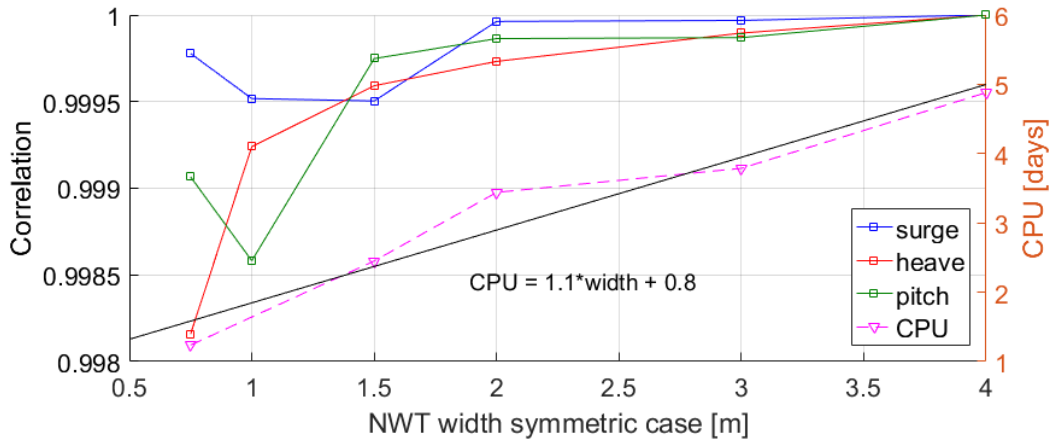


Figure 3.37: Assessment of the width from the convergence of the motion responses and the computational effort - $1m$ equivalent to 4 X-MED radius

All three DoF appears to converge towards the $4m$ wide NWT. The accuracy of the narrowest NWT, i.e. $0.75m$ equivalent to 3 X-MED radius, is already high in all DoF (0.998). So, the improvement in accuracy by increasing the width will be minimal.

A 0.9999 tolerance on the correlation identifies a $3m$ NWT, which requires a 50% increase (i.e. more than a day) in computational effort compared to a $1.5m$ wide NWT.

So, the $1.5\,m$ wide NWT is selected (Figure 3.38) because the marginal gains in accuracy do not justify a wider NWT.

3.8.4 Validation of the 3D-NWT for the WSI of Extreme Events

The $20 \times 1.5 \times 4\,m$ $3D$-NWT developed for extreme WSI is validated against the experiments for events of increasing steepness. The parameters defined by the present study are summarised in Table 3.7, and a representation of the NWT is shown in Figure 3.38.

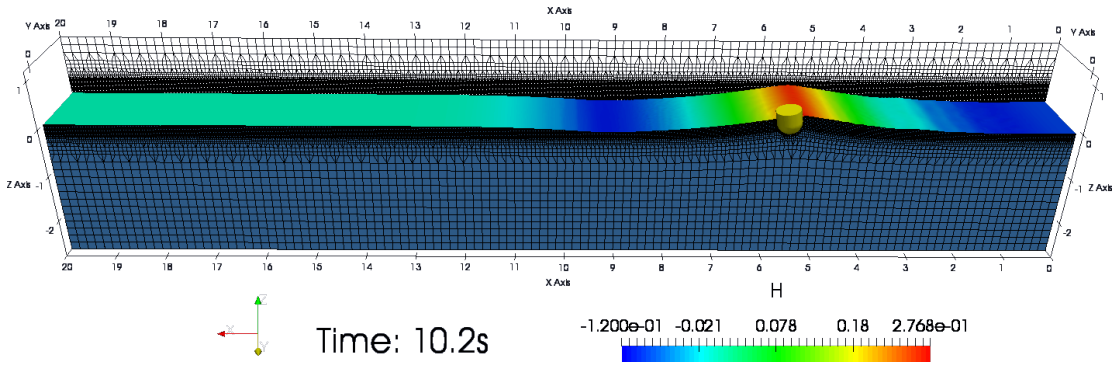
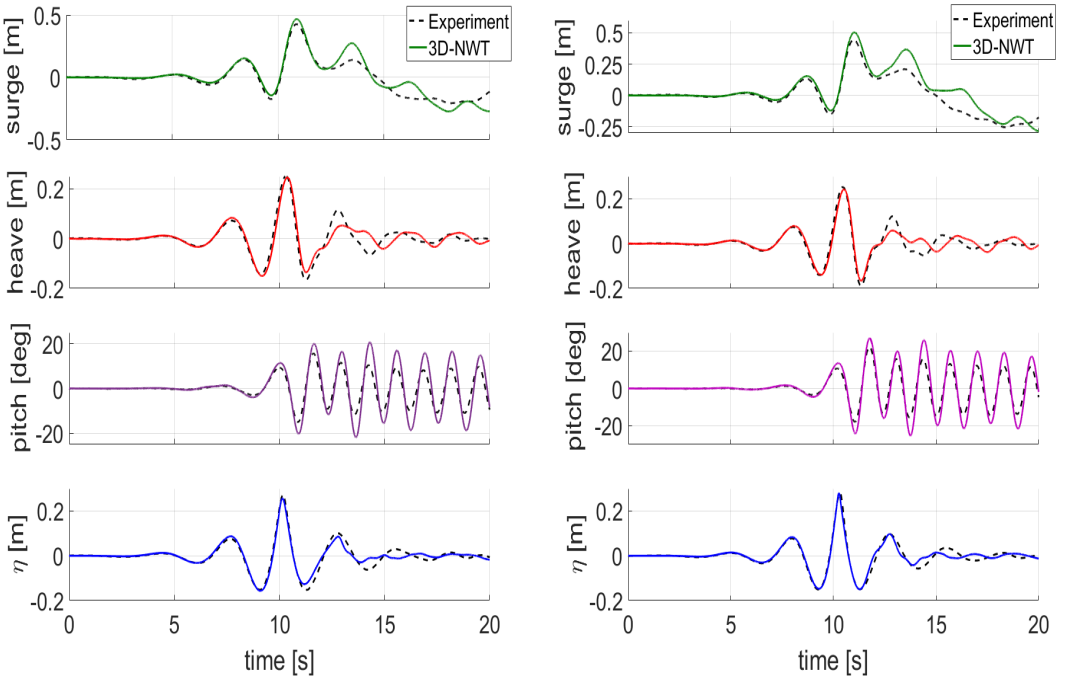


Figure 3.38: The $20 \times 1.5 \times 3\,m$ $3D$-NWT developed for extreme WSI, applied to the $ST1$ event - wave travels from right to left

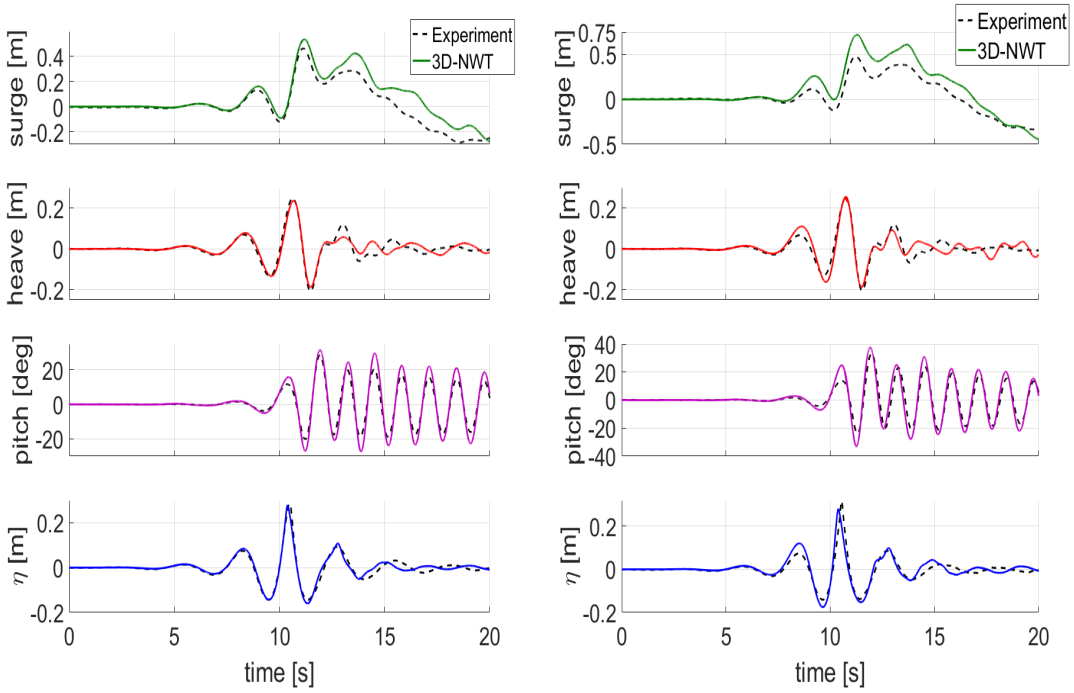
Dimensions	Inlet	Linear superposition	air-phase	CPWH
$20 \times 1.5 \times 3\,m$	$1\,m$	Correlation ≥ 0.9999	$1.2\,m$	20
Relaxation-Zone	MWL refinement	Surface refinement	$ULim$	CFL
$13\,m$	level 3	level 4	$3\,m/s$	$C0 \geq 0.5$

Table 3.7: Parameters defining the developed $3D$-NWT for extreme WSI

Figure 3.39 compares, for the four extreme events, the motion responses (surge (green line), heave (red line) and pitch (magenta line)) and surface elevation at the front face of the model (η (blue line)) simulated by the $3D$-NWT with the experiment (black dashed line).



(a) ST1 *(b)* ST2



(c) ST3 *(d)* ST4

Figure 3.39: Motion responses time-series of the X-MED buoy under the four extreme events of increasing steepness

Figure 3.39 shows that each DoF is captured independently from the non-linearity of the event. The accuracy of each motion responses is mostly constant through the events, where minor trends might be highlighted. Surge motion (green line) tends to be over-estimated as non-linearity increases. Heave responses (red line) are captured to a similar level of accuracy up to the second trough ($12\,s$). Pitch responses (magenta line) seem, however, to improve as the non-linearity increases even if the amplitude increases.

In the three less steep events, the overall agreement appears to decrease after the second trough ($12\,s$). The second crest of the heave response is not captured, while the surge motion over-estimates a bump existing in the experiments after the main peak. The pitch motion does not seem affected. However, the steepest event - $ST4$ Figure 3.39d -, deviates from the experiment earlier ($7\,s$) for both surge and heave responses. This is considered as a consequence of the over-estimation of the first peak previous to the main event, which is sole to the steepest event. Heave and surge motions seem to recover from this inaccuracy. The bump in surge response is captured but not-overestimated, while the peak following the main is captured for the heave motion. Therefore, the overall agreement in all three DoFs validates qualitatively the $3D$-NWT for extreme WSI independently from the non-linearity.

Figure 3.40 shows the experimental (dashed line) and numerical (solid line) motion response spectrum (Equation 2.17) for each event as a function of frequencies of the wave harmonics. The spectral analysis highlights difference as a function of harmonics.

Surge response low frequencies are over-estimated by the numerical model. This over-estimation increases with event steepness. Hence, the surge motion induces by harmonics of low frequencies (i.e. large period) is over-estimated by the numerical model. This could be due to the selection of inlet harmonics (section 3.4.1). However, low frequency components are the ones of largest amplitudes, therefore selected in priority by the inlet pre-processing tool.

In heave, the numerical model has a spectra made of two main frequencies (between

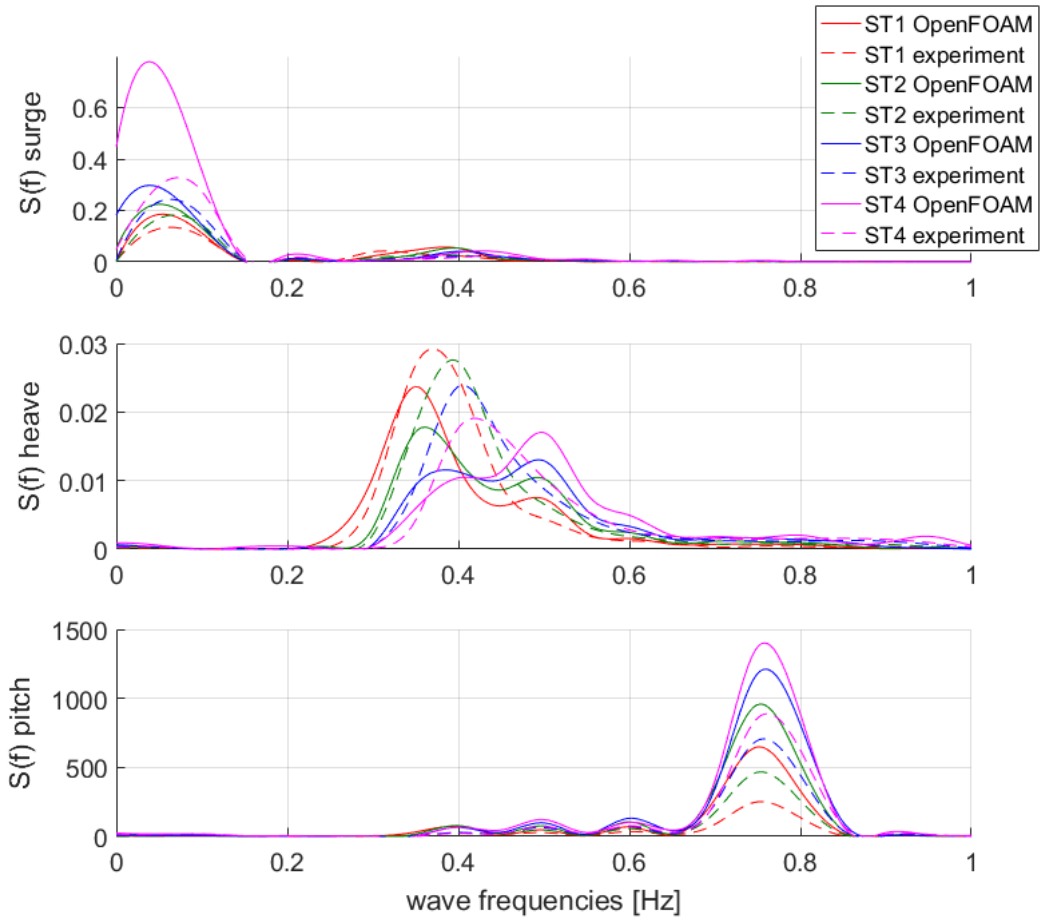


Figure 3.40: Motion spectra responses

$0.35Hz$ and $0.4Hz$, and one at $0.5Hz$), whereas the experimental spectrum have a single peak frequency (between $0.35Hz$ and $0.4Hz$). Numerical heave spectrum peak frequencies shift from the first main frequency to the second as steepness increases. Numerical spectrum deform towards the second higher frequency. In the other hand, experimental heave spectrum peak frequency increases with event steepness while spectrum widen towards higher frequencies.

Numerical and experimental pitch motion spectrum differ by the amplitude of the main peak frequency. Numerical model over-estimates the amplitude of pitch response, while accurately captures oscillation frequency.

Figure 3.41 shows a measure of the accuracy of each motion responses using its

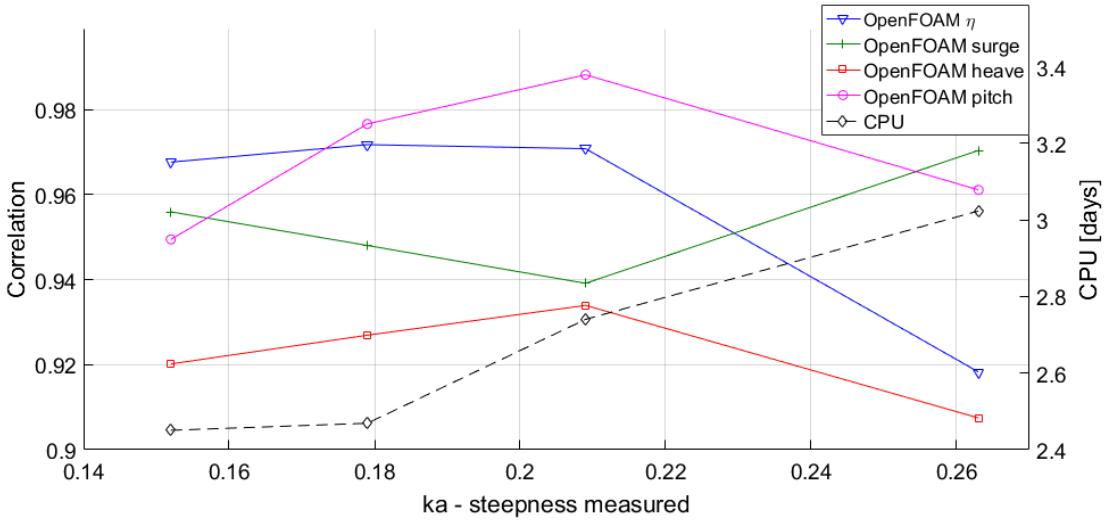


Figure 3.41: Assessment of the validity of the $3D$-NWT compare to experiment

correlation with the experiment as a function of the steepness (kA) of the given event. It aims to evaluate quantitatively the effect of non-linearity on the accuracy of the $3D$-NWT and assess its validity.

Figure 3.41 confirms that the accuracy of the $3D$-NWT is independent of the non-linearity of the event, where each DoF is captured to a similar level of accuracy and constant within events. Therefore, it can be stated that the $3D$-NWT is validated over the representation of extreme WSI within a 0.9 criterion on the convergence of the time-series of both motion responses and surface elevation.

The trends stated from the time-series are reassessed by the correlation, where the steepest event contradicts them. For the three least steep events, the accuracy of the surge motion (green line with x-signs) decreases, while both heave (red line with squares) and pitch (magenta line with circles) increases. However, the accuracy in pitch and heave appears to decrease just for the steepest event, while the surge one increases. This shows that the steepest event might need to be considered apart. This is considered due to the use of the correlation coefficient, which identifies similarity. It explains the high value obtained by the surge for the steepest case, even if the motion appears as over-estimated, and the lower value obtained by the pitch due to a time-shift appearing for the same event. Additionally, the surface elevation of the steepest event

is the most affected by the changes for the $3D$ solver, which is considered to induce a different behaviour in the surge response. Therefore, the independence of the accuracy of the $3D$-NWT from the non-linearity of the event might need to be attenuated, because when is omitted the steepest event ($ST4$), the trends in accuracy are confirmed:

- η is constant as kA increases,
- Surge decreases as kA increases,
- Heave increases as kA increases,
- Pitch decreases as kA increases.

The computational effort (black line with diamonds) is found, as expected, to increase with steepness since it induces an increase in fluid velocity and therefore a decrease of time step size. This increase is also probably due to the increase in the number of components needed for the linear superposition at the inlet (Table 3.2 section 3.4.1). The differentiation of the two is complex as the number of component induces the accuracy of the event. Reducing the number of components is likely to induce some simplifications on the flow, whereas increasing it might include components of high frequency inducing instabilities.

The development of the NWT was based on a blind test approach where the assessment of parameters was conducted solely using a numerical approach before the validation against experiments. Therefore, the present $3D$-NWT is deemed to be able to reproduce other experiments realised at COAST PWT with only minor changes. This statement will be proven in section 4.5.2.

3.9 Conclusion to Chapter 3

This Chapter presents the development of a Numerical Wave Tank (NWT) for extreme Wave Structure Interaction (WSI) based and validated against physical experiment of a moored floating buoy under four NewWave based focus events of increasing steepness.

First, the representation of the extreme events (i.e. wave-only cases) is assessed against a long NWT identified as greater accuracy since unaffected by wave-reflections. As a compromise between speed and accuracy, the number of cells per wave height (CPWH) required for an accurate representation of the least steep event is identified as 20, while the absorption-zone avoiding wave-reflections is found to be $13m$. The defined $20m$ NWT is validated against the four extreme events with a 0.978 minimum correlation over the surface elevation.

Second, the Rigid Body Motion (RBM) is implemented in the NWT to assess the WSI. For consistency from 2 to 3 dimensions simulations, a limitation restricting flow velocities is implemented as found necessary to avoid the development of abnormal air flows affecting the modelling of the events. Decay tests in each Degree of Freedom (DoF) of the buoy validate the RBM against resonance frequencies and decay time-series.

Finally, the NWT models the motion responses of the X-MED buoy under the four extreme events. It is validated against experiments within a 0.9 correlation over each DoF time-series. Overall agreement proves that the NWT accuracy in extreme WSI is mostly constant independently from event steepness (i.e. non-linearity); nevertheless, potential trends are suggested.

Chapter 4

Assessment of WaveDyn performance and limits on modelling extreme events

Wave Energy Converter (WEC) developers highlight the need for parametric validity of numerical models for survivability of WECs (CCP-WSI Working Group (2016), discussed in section 2.2.3). The aim is to assess as a function of wave parameters, rather than assume the validity based on the level of simplifications of the Navier-Stokes (NS) equations (section 2.1.1). This chapter aims to investigate parametrically the validity of two numerical models of sensibly different level of simplifications for the survivability assessment of WEC in extreme Wave Structure Interaction (WSI).

WaveDyn model is a mid-fidelity model based upon a linear time-domain description defined by the Cummins Equation of an inviscid, irrotational and frictionless, hence potential flow (sections 2.1.2 and 4.1.2). OpenFOAM model is a Computational Fluid Dynamics (CFD) package solving the Reynolds Averaged Navier-Stokes (RANS) equations (section 2.1.2) for incompressible two-phase flow, from which a Numerical Wave Tank (NWT) is developed in Chapter 3 and validated to model accurately extreme WSI.

Numerical model validity assessment is conducted against physical experiments (known for reliability Rafiee et al. (2016)) of the X-MED buoy's motion responses under the four extreme events of increasing steepness (ST cases - section 2.4). Validity is measured against event steepness as a non-linear wave-parameter, which is expected to identify limits of the mid-fidelity model WaveDyn induced by the simplifications of the NS.

This investigation is the second step in the future coupling work. The two numerical

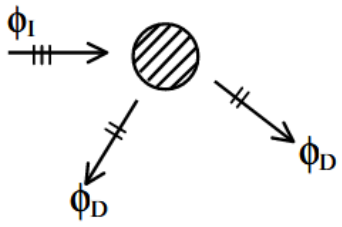
models will form one efficient tool that benefits from the computational efficiency of WaveDyn on any cases while being able to swap to OpenFOAM to include high fidelity WSI only at instant identified as out of WaveDyn's limits.

This Chapter first develops WaveDyn numerical model for the X-MED buoy. The comparison of the two numerical models starts against the experimental decay tests to validate each model Rigid Body Motion (RBM). Investigation extends to extreme WSI using the ST cases, where is assessed WaveDyn's validity for survivability. Finally, the deviation of WaveDyn motion responses from the experiment is used to parametrically evaluate WaveDyn limit as a function of the non-linearity of the event.

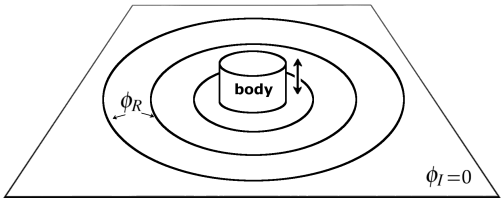
4.1 WaveDyn: A Time-Domain Potential Flow Based Solver

Potential flow models are created by solving in the frequency domain the Laplace Equation 2.10 for inviscid, irrotational flow. Hydrodynamic forces acting on a structure are computed assuming small displacement, where tools allow the description of complex geometries. These models (low/mid-fidelity) are widely used in marine engineering for the relative simplicity and fast computation.

4.1.1 Diffraction, Radiation, Excitation



(a) Diffraction - The static presence of the body affects the incident wave field



(b) Radiation - In the absence of an incident wave field, the motion of the body generates a wave field

Figure 4.1: Schematic representation of diffraction ϕ_R , and radiation ϕ_D effects of a structure in an incident wave field ϕ_I

The presence of a structure is separated between its static influence on the flow field, diffraction (Figure 4.1a); and its dynamic influence, radiation (Figure 4.1b). Assuming

small displacement, radiation and diffraction are decoupled. This description models the hydrodynamic force acting on the structure as the contribution of each field (incident, diffracted, radiated), plus the hydrostatic force (i.e. buoyancy). Further details on the theory can be found in Delhommeau (1987).

Radiated forces are modelled as a function of the structure velocity $\overrightarrow{V}$ and acceleration $\overrightarrow{a}$ via damping coefficient B, and 'added masses' M_a relating to the additional inertia the structure feels as it moves through the fluid in a specific Degree of Freedom (DoF):

$$\overrightarrow{F_{radiation}} = M_a \frac{\partial \overrightarrow{V}}{\partial t} + B\overrightarrow{V}. \tag{4.1}$$

The diffraction field is a consequence of the incident field. It is computed simultaneously as the excitation of the incident field assuming a linear superposition (Equation 2.16). The excitation force is the sum of the contribution of each wave component on the structure at a specific time. The response of the structure to a given linear wave defines the contribution. So, the excitation force is defined by the response function of the structure to a range of regular waves, i.e. amplitude of response as a function of wave frequency.

Radiation and excitation forces are dependent on the structure geometry, and the linear sum of the incident wave-field. Calculations of the coefficients - added mass, damping, and excitation responses - can be computed in the frequency domain by 'potential flow solver' (WAMIT, ANSYS-AQUA, or the open-source one NEMOH) based on the Boundary Element Method (BEM) (Delhommeau (1987), Lee and Newman (2013)). This method discretized the wetted surface of the structure at rest into panels. It allows the description of hydrodynamics of complex geometries as a function of wave frequency.

4.1.2 Time-Domain Solver, WaveDyn

The output from BEM solvers can be implemented into a time-domain formulation based on the conservation of momentum applied to the structure:

$$M\overrightarrow{d} = \sum \overrightarrow{F} = \overrightarrow{F_{excitation}} + \overrightarrow{F_{radiation}} + \overrightarrow{F_{hydrostatic}} \tag{4.2}$$

This formulation reduces the hydrodynamics assessment to a mechanical analysis using additional forces to account for the hydrodynamics effects acting on the body. A common formulation of the previous equation is the *Cummins* Equation (Cummins 1962):

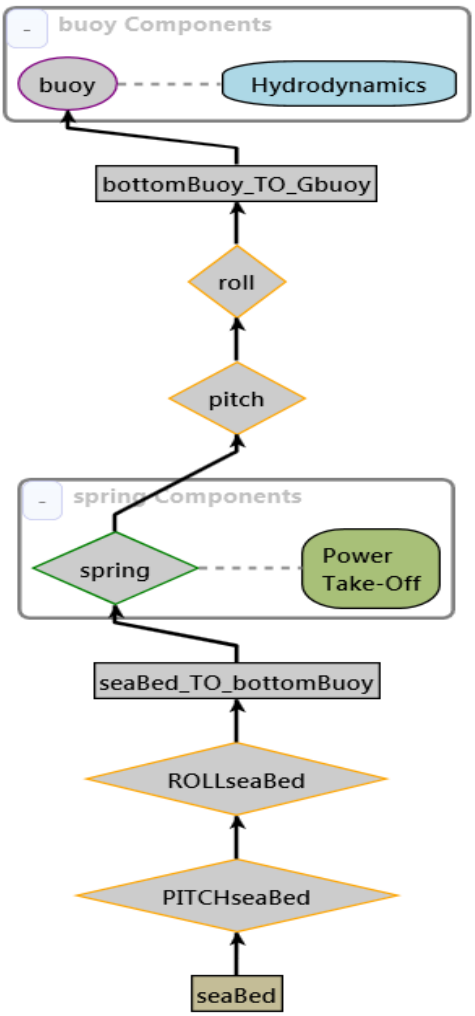
$$(m + m_r(\infty))\ddot{x}(t) + f_{hs}(t) + \int_{-\infty}^{t} k(t-\tau)\dot{x}(\tau)\partial\tau = f_e(t) + f_{ext}, \tag{4.3}$$

where x is the vector body displacement, $m_r(\infty)$ is the added mass at infinite frequency, $f_{hs}(t)$ is the hydrostatic force, $f_e(t)$ is the excitation force, and $f_{ext}(t)$ gathers additional forces that can be applied on the structure (mooring, Power Take Off (PTO), joints). $k(t)$ is the body impulse response function obtained by the BEM solver along with the added masses and excitation forces.

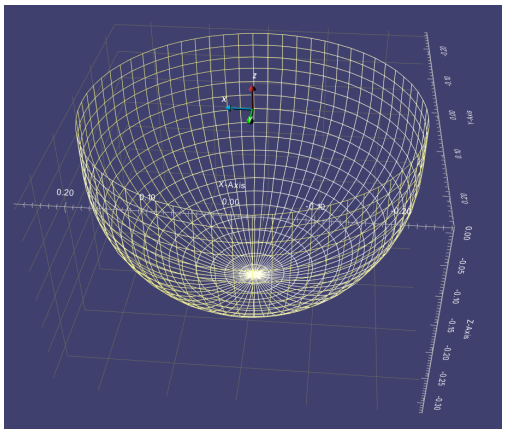
WaveDyn is a commercial code developed by the consultancy DNV-GL; the second body funding this research. WaveDyn solves the *Cummins* Equation using a multi-body description of interconnected components such as hinges or sliding joints to describe a WEC. This structural multi-body analysis is capable of modelling large scale kinematic (DNV-GL 2013). Numerical models based on the Cummins equation like WaveDyn have been used among WEC developers as they benefit from a low computational cost while allowing time-domain analysis where complex mechanics and non-linear hydrodynamics effects can be additionally implemented.

4.1.3 X-MED model

In WaveDyn, the X-MED model (section 2.4.2) is represented as a succession of hinges, mass-less rigid links, and slider, as shown in Figure 4.2a. The components are



(a) Mechanical description of X-MED model in WaveDyn



(b) Moored wetted surface made of 1214 panels - input for WAMIT

Figure 4.2: X-MED WaveDyn model

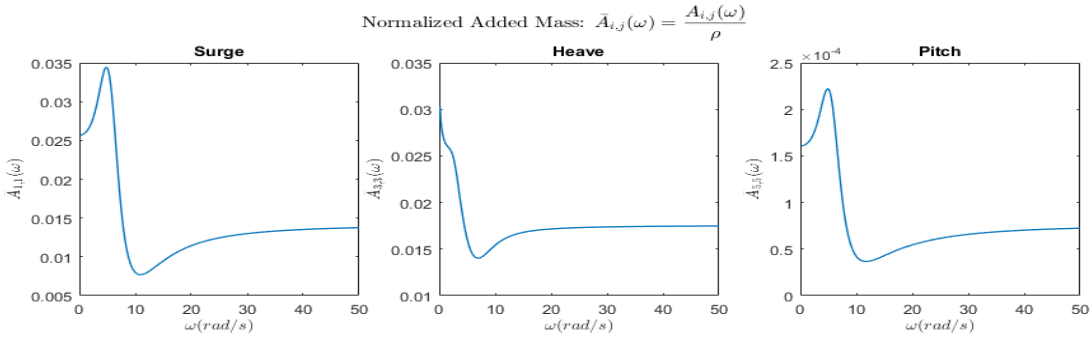
chosen to ensure the DoF of the X-MED buoy modelled as a rigid-body. Two hinges - name 'PITCHseaBed' and 'ROLLseaBed' - represents the roll and pitch of the experimental mooring line on the seabed. As for the OpenFOAM model, the mooring line is represented by a single linear spring using a slider (named 'spring') with a $66.3 N/m$ PTO acting as stiffness. A $2.18 m$ mass-less rigid link - name 'seaBed_TO_bottomBuoy' - connects the seabed to the spring. Two hinges - name 'pitch' and 'roll' - connect the spring to the bottom of the buoy - name 'bottomBuoy_To_Gbuoy' - while allowing the rotations. A $0.18 m$ mass-less rigid link connects the bottom of the buoy to its centre of

gravity. This model allows the buoy to move in 5-DoF, as the yaw motion is assumed as non-existent.

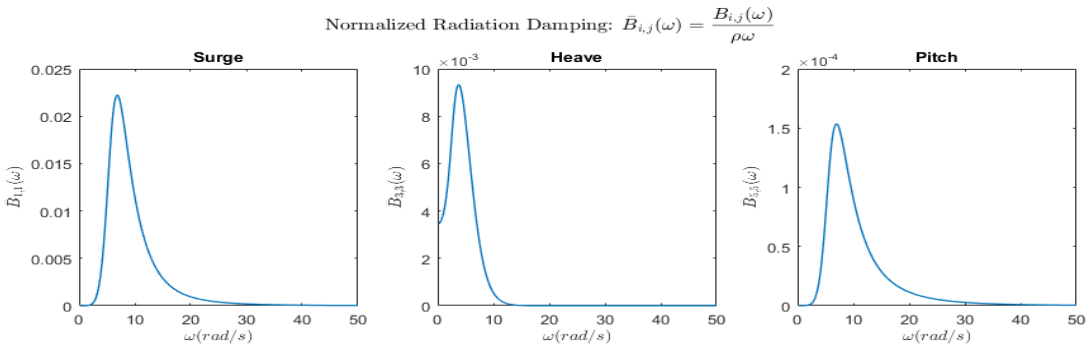
The mass and moments inertia of the buoy are the same as in the experiments (Table 2.2). Calculated at the equilibrium of the moored model using Archimedes' principle, the mean displaced mass, i.e. the hydrostatic force, is $45.291\,kg$. The centre of Buoyancy is specified $0.01014\,m$ above the centre of gravity. The extension of the spring at equilibrium is $0.306\,m$.

At the centre of gravity of the moored X-MED buoy are input the hydrodynamics coefficients calculated by the 'higher-order method' ($ILOWHI = 1$ - detailed in Lee and Newman) of the potential flow BEM solver WAMIT, using the moored wetted surface (Figure 4.2b) generated from MultiSurf v1.44 with 1214 panels of $0.025\,m$ panel size. Normalized radiation and added mass responses are shown in Figure 4.3. The centre of gravity of the moored buoy, situated at $2.667\,m$ above the seabed, is used as the origin for the hydrodynamic coordinate system. The hydrostatic stiffness is a diagonal matrix containing the heave stiffness, $1925.53\,N/m$, the pitch and surge, $30.0866\,N/m$.

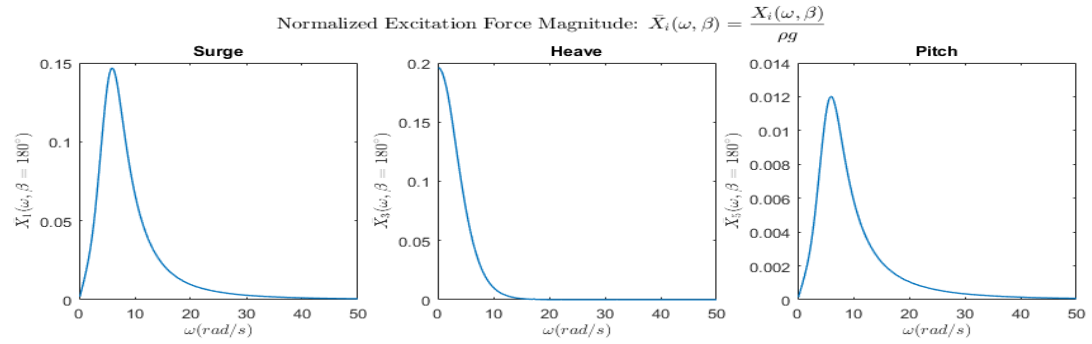
WaveDyn uses the linear superposition of the measurement made at the Wave Gauge (WG) the closest to the X-MED centre, WG#14 in Table 3.1; i.e. not the focus location. The number of components is determined by a level of accuracy compared to the experimental data defined by the correlation between the two. This is performed by the same procedure (script and tolerance 0.9999) than developed for the inlet of the NWT described in section 3.3.2.



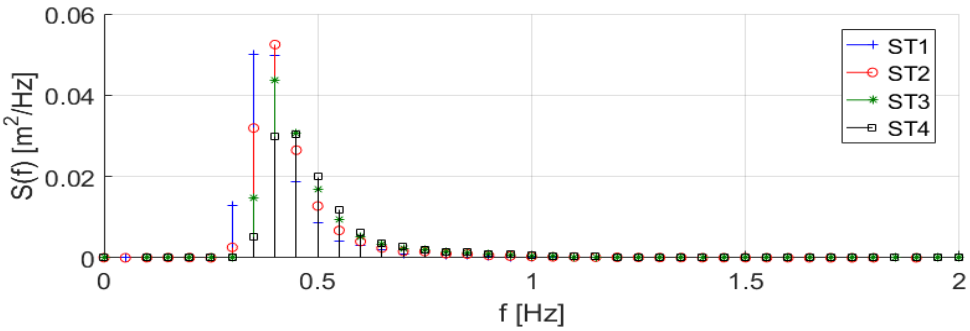
(a) Normalized Added Mass



(b) Normalized Radiation Damping



(c) Normalized Excitation Force



(d) Discrete wave spectrum used as input

Figure 4.3: WAMIT hydrodynamics used as input in WaveDyn

4.2 Validation of Rigid-Body Motion

The decay tests (moored heave, unmoored heave and moored pitch) are used to validate RBMs of each model against Fluid Structure Interaction (FSI) of a single DoF to avoid the complexity of the wave.

4.2.1 Method

This method is the same as previously used for the NWT development (section 3.7.1) in which the two models are compared against experiments. The assessment is conducted over the resonance frequencies of the considered DoF, and the decay (i.e. decreasing oscillations due to damping) time-series for the heave moored and unmoored cases. Resonance frequencies are defined as the inverse of the first oscillation period. Decrement of decays are calculated as $\frac{T}{\tau}$ from a fit of the damping oscillatory motion defined as:

$$d_{of} = Ae^{-\frac{t}{\tau}} \times cos(\frac{2\pi}{T}t + \theta), \tag{4.4}$$

where d_{of} means a given DoF, A, τ, T, θ are constants calculated via the fitting tool.

OpenFOAM model is the 3-dimensions ($3D$)-NWT to the width of the Physical Wave Tank (PWT), hence $7.75m$, to avoid the effects of radiated waves reflecting on side walls on time-series (developed and proven in section 3.8.1).

For the moored heave decay test, the initial spring displacement of the WaveDyn model previously described is set to $0.51m$. For the moored pitch decay test, the 'pitch' hinge (Figure 4.2a) is set to an initial angle of 12.5°. For the unmoored heave decay test, the spring stiffness is set to zero, and the mean displaced mass is adjust to the mass, $43.2kg$. The unmoored heave decay released position is $18mm$ lower than the moored one, so the initial spring displacement is set to $0.492m$.

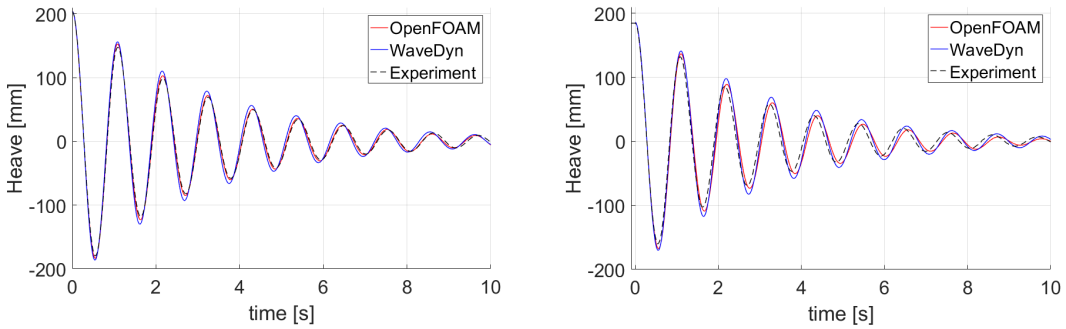
4.2.2 Results

Across the present chapter and unless specified differently, all Figures presenting motion responses show OpenFOAM as (red line), WaveDyn as (blue line), and experiment as (black dashed line). Table 4.1 reports the resonance frequencies and decrements.

Resonance frequencies $[Hz]$	Heave Moored	Heave Unmoored	Pitch Moored
Experiment	0.917	0.926	0.75
OpenFOAM	0.923	0.907	0.76
WaveDyn	0.926	0.901	0.75
Decrement T/τ	Heave Moored	Heave Unmoored	Pitch Moored
Experiment	0.359	0.384	–
OpenFOAM	0.357	0.390	–
WaveDyn	0.333	0.347	–

Table 4.1: Resonance frequencies and decrements obtained for decay tests

Heave Decay



(a) Moored *(b)* Unmoored

Figure 4.4: Heave responses time-series for moored and unmoored heave decay tests

WaveDyn is not performing as well as the OpenFOAM in the moored decay (Figure 4.4a), but the accuracy remains high. Resonance frequency ($0.917\,Hz$) is slightly more over-estimated by WaveDyn (0.926 compared to 0.923 for OpenFOAM). Decay time-

series is less accurate due to the shift in frequency, while the decrease in amplitude appears very similar as shown by decrement values: Experiment 0.359, OpenFOAM 0.357 and WaveDyn 0.333.

For the unmoored heave decay, Figure 4.4b, OpenFOAM model is also found to capture the motion slightly more accurately than WaveDyn model, but both to a lower accuracy than the moored decay. Both resonance frequencies ($0.907\,Hz$ for OpenFOAM, $0.909\,Hz$ for WaveDyn, and $0.926\,Hz$ experiment) are less well predicted than for the moored case. Both models over-estimate the amplitude of oscillations, WaveDyn being the less accurate. Decrement (0.384) is less over-estimated by OpenFOAM (0.390) than under-estimated by WaveDyn (0.347).

The slightly lower accuracy found for WaveDyn model can be due to the changes in the wetted surface. These are not accounted but can be significant in large heave decay, or for complex structures inducing large changes in volume (while decaying) or presenting discontinuities. The ratio between maximum offset and radius is high, 0.8, but the X-MED buoy does not present sever discontinuities. Lawson et al. (2014) found that differences between linear buoyancy (as used in WaveDyn) and the instantaneous wetted surface appears for an ellipsoid buoy of a ratio (half-height/offset) above 0.37. So, it is possible that the consideration of the wetted surface could improve decays.

Viscous effects are believed to play an important role as they accentuate the damping of the motion. This is indicated when comparing numerical results. OpenFOAM, which considers viscous damping, shows a larger energy loss identified by smaller amplitudes and larger periods than WaveDyn. Therefore, a viscous damping model could be implemented within WaveDyn to improve heave responses.

However, these inaccuracies are considered negligible because the overall agreement is good as proven by the resonance frequencies and decrements values (Table 4.1). Besides, the experiments are not perfect as decay tests are highly subjected to the experimental set-up. So, WaveDyn model is validated, such as the OpenFOAM one, against both moored and unmoored heave decay tests.

Pitch Decay

As no experimental time-series are available, Figure 4.5 only shows WaveDyn and OpenFOAM pitch motion responses. Table 4.1 only reports the resonance frequencies. WaveDyn accuracy is found higher than the OpenFOAM. The resonance frequency is found equal to experimental for WaveDyn, whereas $0.1\,Hz$ larger for OpenFOAM.

The amplitude of WaveDyn's pitch response is larger than OpenFOAM one, and the damping of less importance. The similitude with OpenFOAM constitutes the validation of the WaveDyn model in pitch because OpenFOAM has been proven accurate in the representation of pitch motion time-series for more complex cases.

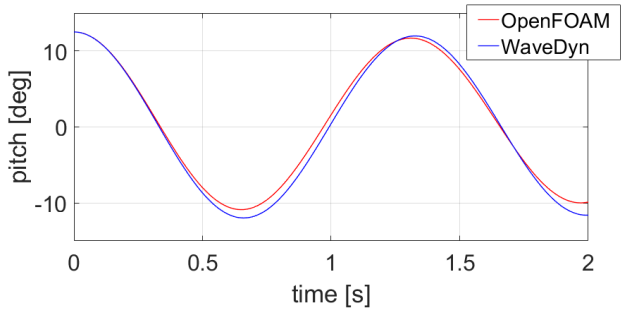


Figure 4.5: Pitch motion responses of the moored X-MED buoy

The computational effort for WaveDyn is incomparable to OpenFOAM. Each WaveDyn simulation runs under $10\,s$ on a normal desktop, whereas OpenFOAM $3D$-NWT simulations take multiples hours on a cluster. Decay tests are required to set-up the model correctly, and cannot be avoided to save computational effort. However, for simple FSI like decay tests, and probably also due to the simplicity of the structure model, mid-fidelity models like WaveDyn are significantly more efficient and accurate enough to obtain response frequencies.

The mesh used in the OpenFOAM model, however excessive, is required to capture the decay of DoF oscillations without the effects of radiated waves reflecting on side walls. This highlights a major difference between these two models: WaveDyn is not affected by wave reflections; apart from the one included in the wave-input by the measurement

of WG#14.

4.3 Validation to Extreme WSI

The two numerical models being validated for single DoF motion responses, the validation is extended to extreme WSI. The investigation assesses the modelling of 3-DoF large motion responses - surge, heave and pitch - faced by the X-MED buoy during the four extreme events of increasing steepness. This aims to highlight the validity limit of linear based models like WaveDyn for survivability application, and to quantify this limit as a function of the non-linear wave-parameter: kA, the event steepness.

The investigation focuses on the accuracy of the motion responses rather than the mooring loads because an accurate representation of the mooring load does not imply an accurate representation of the 6-DoF motion responses. However, the opposite is true. Furthermore, by restricting the analysis to the loads in the mooring line, the history of motion is omitted (i.e. cannot be tracked) despite known to influence the estimation of extreme loads for large dynamic structure (Hann et al. (2015), Taylor et al. (1997)). So, the investigation considers first the motion responses, then the loads to quantify the differences and identify the numerical model most suitable for motion responses or mooring loads assessment.

4.3.1 Method

No further changes in models are made to maintain a blind-test approach based on five parts:

- Experiment reference : X-MED buoy tested in the COAST PWT (section 2.4)
- 4 Extreme events: one event based upon NewWave theory from a 50 years storm, from which three events of increasing steepness are generated (section 2.4.3)
- OpenFOAM: $20 \times 1.5 \times 4\,m$ $3D$-NWT validated for extreme WSI (section 3.8.4)
- WaveDyn: potential flow based time-domain model

- Wave-input: linear superposition define by a 0.9999 tolerance compare to WG ($\#1$ for OpenFOAM, and $\#14$ for WaveDyn - section 3.3.2)

The definition of wave-input using the same tolerance ensures that no model benefit from a more precise definition of the wave-input. The number of components for each wave-input are presented in Table 4.2.

Correlation 0.9999	$ST1$	$ST2$	$ST3$	$ST4$
WG#1 OpenFOAM	28	32	39	52
WG#14 WaveDyn	34	38	45	80

Table 4.2: Number of components required for each event at a 0.9999 level of accuracy

The assessment of each DoF accuracy is conducted qualitatively via the comparison of motion responses time-series. The correlation between experiment and numerical motion responses are then used as a quantitative measure of the accuracy (section 3.3.2). Experiment and OpenFOAM time-series are interpolated at WaveDyn fixed time vector: $(0 : 0.01 : 20)$.

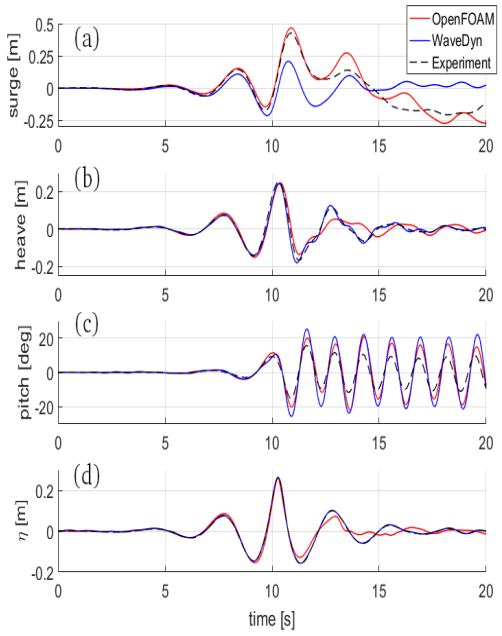
4.3.2 Results

3-DoF

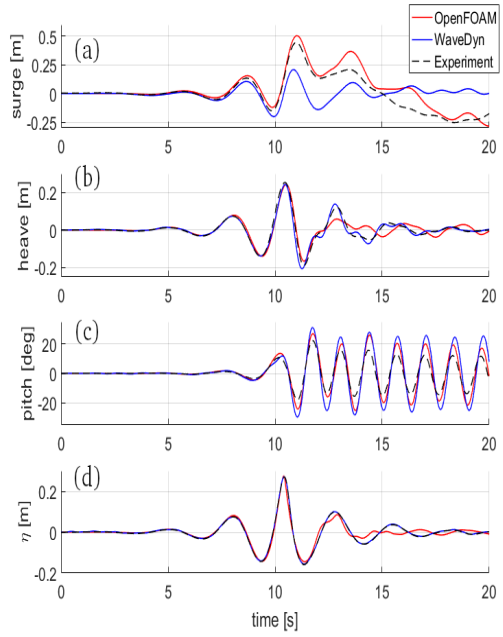
Each DoF motion responses of the X-MED buoy under the four events of increasing steepness are shown in Figure 4.6. Negative surge and pitch responses are towards the wave-maker or inlet, while positive ones are towards the tank end. The surface elevation at WG$\#14$ (i.e. closest to the model centre) is plotted on the lowest sub-plot. Figure 4.7 shows each DoF motion response spectrum for OpenFOAM (solid line), experiment (dashed line) and WaveDyn (dash-dotted line).

Qualitatively from each sub-figure (4.6a - 4.6d), WaveDyn captures heave and pitch motion responses, with significant accuracy in heave. The accuracy in pitch seems to decrease with steepness, while heave remains stable. However, the surge motion response is mostly not captured, as WaveDyn greatly under-estimates the main peak,

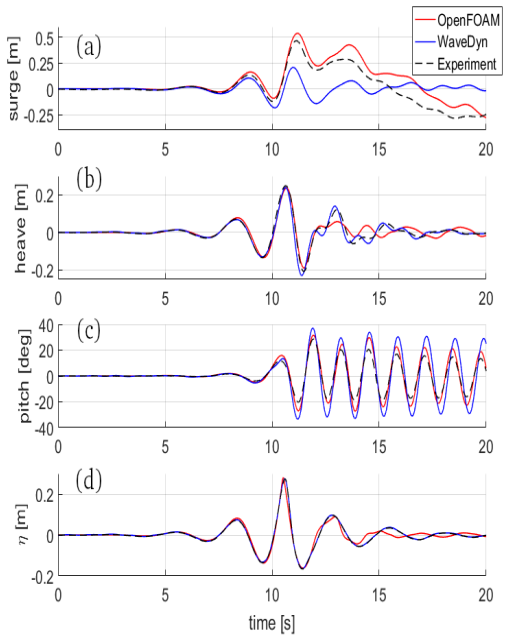
and does not reproduce the following backwards motion after $15\,s$.



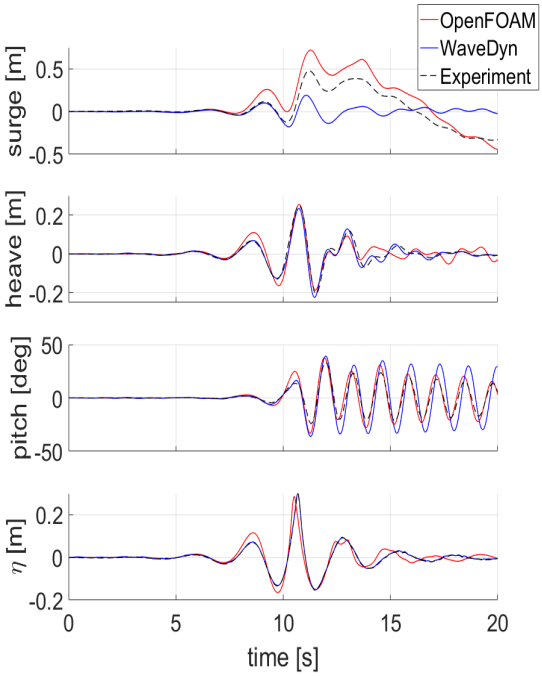
(a) ST1



(b) ST2



(c) ST3



(d) ST4

Figure 4.6: Motion responses time-series of the X-MED buoy under the four extreme events of increasing steepness - surface elevation η measured at WG#14

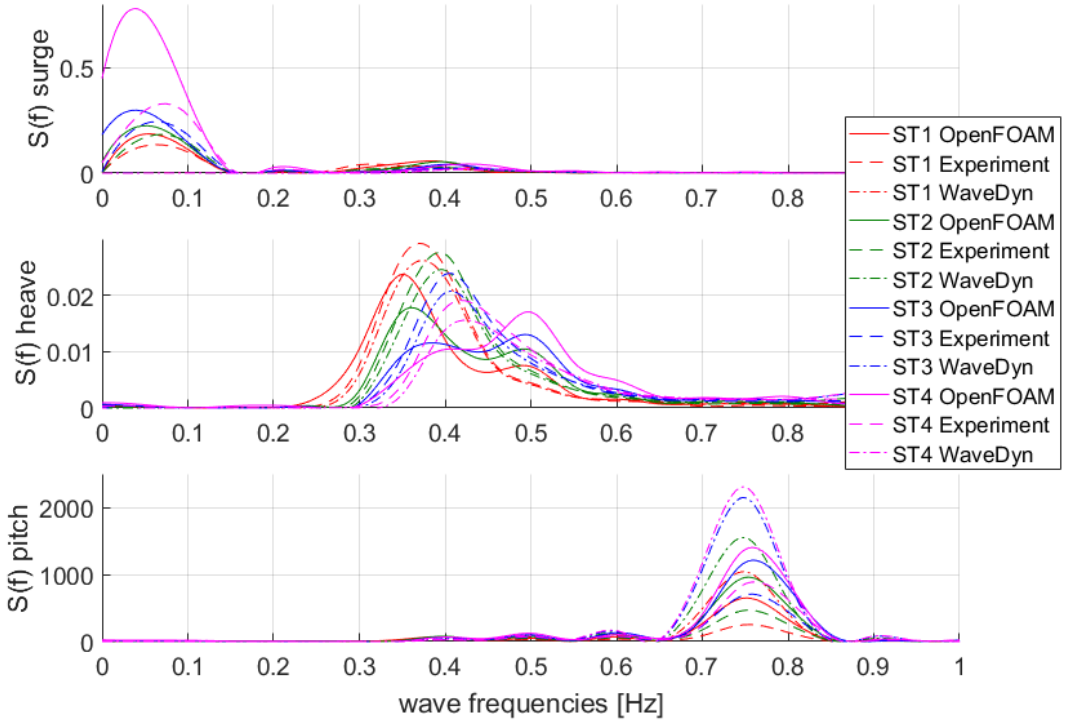


Figure 4.7: Motion spectra responses

Additional conclusions come from Figure 4.7. WaveDyn surge motion spectrum is almost inexistent on Figure 4.7, which indicates that low frequency motion (i.e. slow drift) is not captured. Heave and surge motion spectrum show great similarities with experimental ones, hence confirming the accuracy of the solutions for these DoF. Comparing motion peak frequencies, WaveDyn heave and pitch appear slightly shifted compare to experiment for all events.

As detailed in section 3.8.4, the OpenFOAM model captures at a similar level of accuracy each DoF motion response to an extent that can be considered as independent from non-linearity.

The best accuracy of a given DoF is found to be dependent on the numerical model. Heave is best replicated by WaveDyn, while pitch and surge by OpenFOAM. So, the OpenFOAM model is not found to succeed undoubtedly over WaveDyn in all DoF.

The quantitative assessment done via the correlation confirms these statements. Fig-

ure 4.8 shows the evolution of correlation as a function of the steepness of the event for the surge (green x-signs), heave (red squares), pitch responses (magenta circles), and the surface elevation (blue upside-down triangles). OpenFOAM is represented using coloured dashed lines, and WaveDyn coloured solid lines. WaveDyn surge motion (green line with x-signs) is set separately to highlight the difference in accuracy. All correlation values are available in Table D.2 in Appendix D.2.

The accuracy of the surface elevation (blue upside-down triangles) is higher for WaveDyn than OpenFOAM because of the difference in the description of the wave. WaveDyn imposes the WG#14 measurement with a 0.9999 accuracy, which assures a perfect description of the surface elevation as shown in Figures 4.6a - 4.6d. On the other hand, OpenFOAM imposes the linear superposition of the upstream WG at the NWT inlet. The wave-group then travels inside the NWT in a non-linear domain, in which the linear description of a non-linear wave-group necessarily induces inaccuracies.

The level of accuracy of the surface elevation induces the accuracy in heave since heave motion is generally a linear response to the incident wave. OpenFOAM accuracy in η shows a similitude with heave responses as steepness increases. WaveDyn description of events ensures an accurate modelling of heave found consistently more accurate than OpenFOAM. So, the WaveDyn model is found more suitable in the modelling of heave response across the range of steepness ($[0.152; 0.263]$) due to the use of the surface elevation at the model centre.

The accuracy of WaveDyn in heave appears to slightly decrease with the increase in event steepness while the accuracy of the event remains constant. So, the limit of WaveDyn in heave might require investigating over even steeper events. However, the accuracy of the event is expected to remain stable as long as a liner superposition can describe the shape of the surface elevation. Therefore, a highly distorted surface, such as a breaking wave, is probably required to reach WaveDyn's limit in the representation of the surface elevation, and hence in heave. This confirms the general idea that potential flow based models are successful until wave-breaking but restricts this

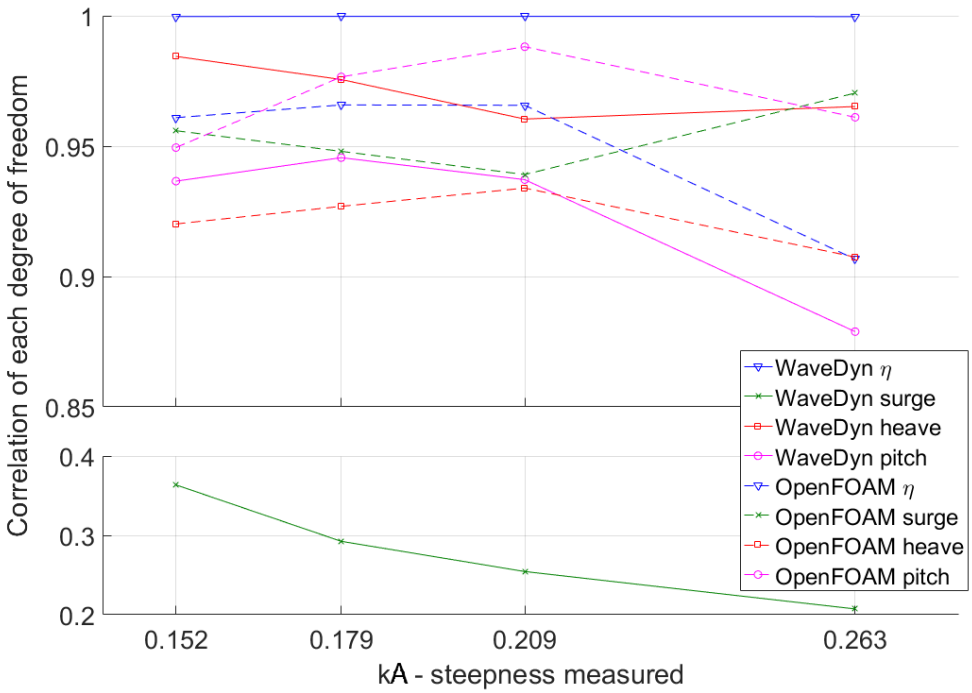
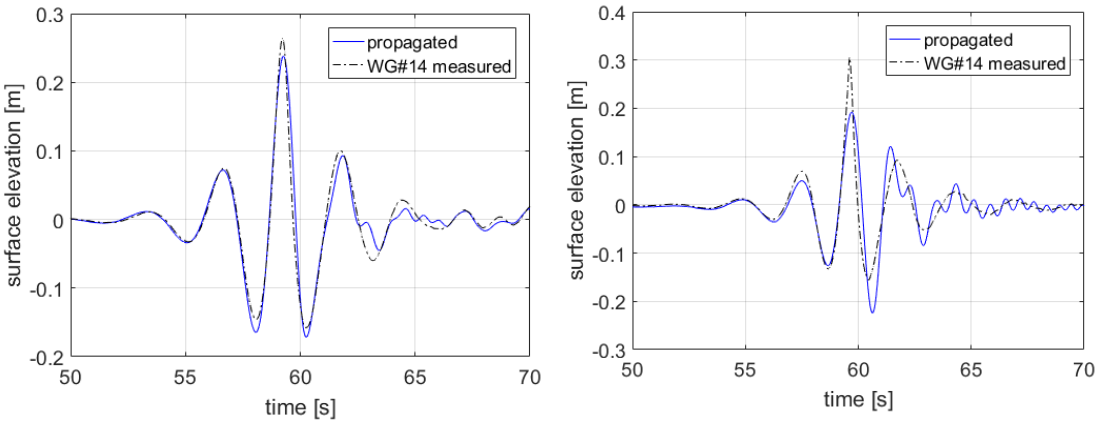


Figure 4.8: Correlation between each numerical model and the physical reference for each DoF motion response and the surface elevation at WG#14

statement to the assessment of heave motion responses.

However, WaveDyn accuracy in heave will decrease in the event of a test using only WG#1. Indeed, the propagation of non-linear wave-groups according to linear theory can induce significant error. Figure 4.9 shows the difference in the surface elevation for the least and steepest events obtained via a linear propagation. In this case, higher wave-propagation model can be used for WaveDyn. Or, a fairer comparison for both models might be to use OpenFOAM solution for the surface elevation as the wave-input for WaveDyn. This idea is not conducted any further by the present study, as significant inaccuracies are found for other DoF in WaveDyn.

WaveDyn captures pitch motion, but at a lower accuracy than OpenFOAM. The major differences are the over-estimation in amplitude and the shift in oscillations as steepness increases. This could be explained by the pitch decay (section 4.2.2), which already highlights that WaveDyn over-estimates pitch amplitudes and under-estimates

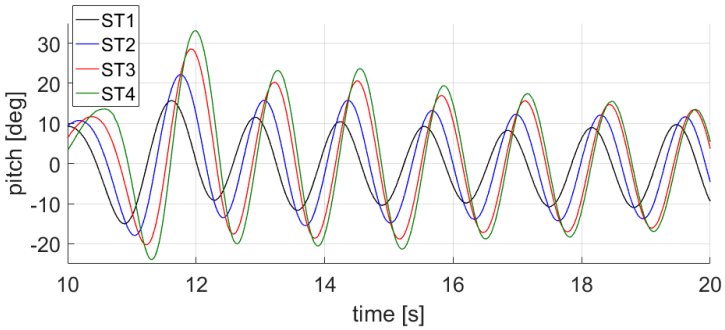


(a) ST1 *(b)* ST4

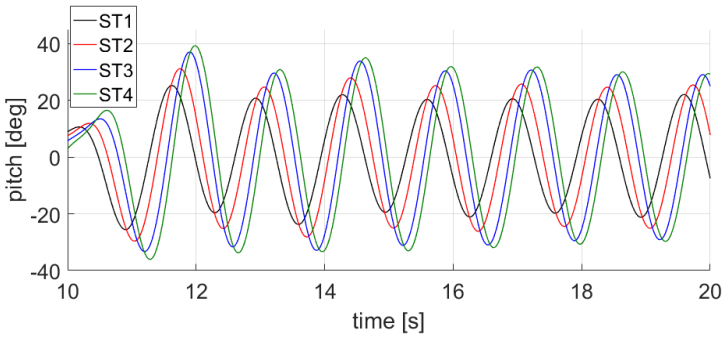
Figure 4.9: Difference between a linear-based propagation of the linear superposition imposed at the NWT's inlet and experiments

damping compare to OpenFOAM. Experimentally, the increase in steepness increases the amplitude of the first oscillations in pitch. For large amplitudes, the pitch is found to decay with more extent than for small ones (Figure 4.10a). WaveDyn does not capture this damping (Figure 4.10b), so its increase and therefore, WaveDyn accuracy decreases with the non-linearity of the event. So, WaveDyn's error in pitch is considered due to viscous effects as WaveDyn assumes no viscosity. The consistency in the pitch for OpenFOAM proves that the model captures the damping, while an overestimation in amplitude remains since OpenFOAM ignores turbulent viscous effects. WaveDyn accuracy remaining high, the limit of WaveDyn in pitch is not attained even for the steepest event.

Surge slow drift is not well capture by WaveDyn. In Figures 4.6a to 4.6d, surge motion appears as a consequence of the wave-group flowing past the X-MED buoy. For the main event, the wave carries the buoy along the direction of the wave-propagation. This slow drift motion appears as a consequence of lower frequency wave components as indicated by the main peak frequency in the motion spectrum on Figure 4.7. WaveDyn cannot capture this motion since no drift force is implemented, and since the change in position of the buoy is not considered within the input wave condition. Common formulation of drag force proportional to the square of velocity are expected to generate



(a) Experimental pitch responses



(b) WaveDyn pitch responses

Figure 4.10: Experimental damping on the pitch motion due to viscous effects not capture by WaveDyn

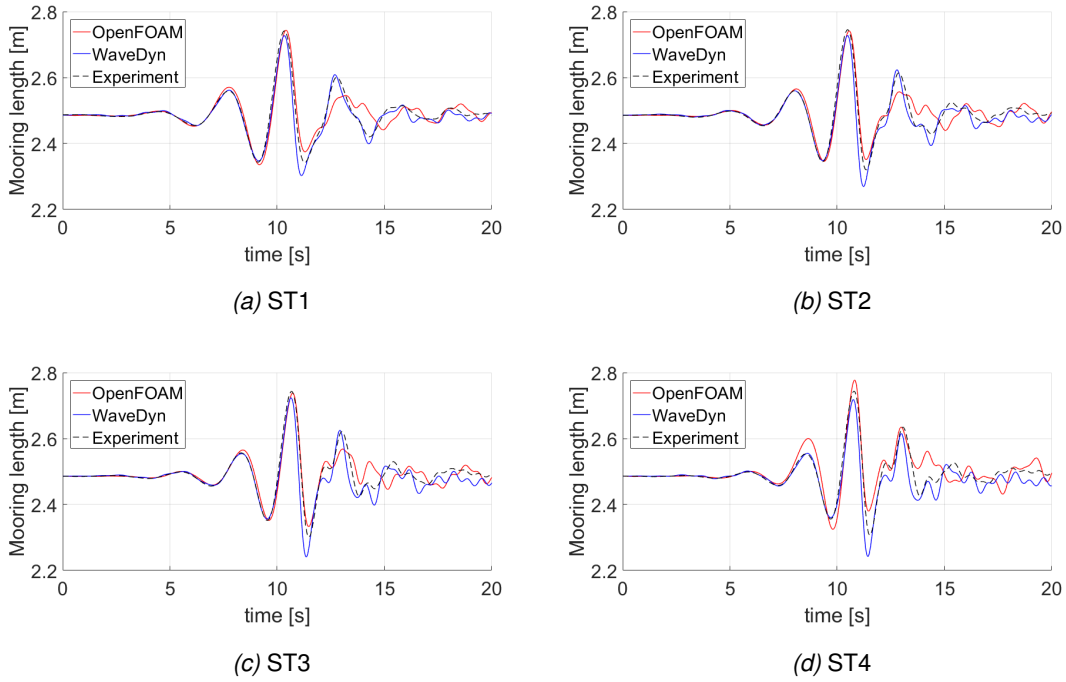
higher frequency response, and are therefore not used in the present study. Therefore, WaveDyn limit for the surge response is already attained at the least steep event, $ka = 0.152$.

In conclusion, for the modelling of extreme WSI, WaveDyn is found to capture independently from the non-linearity of the event, heave and pitch responses, while not surge due to dominating viscous effects. On the other hand, OpenFOAM is found to succeed independently from the non-linearity or the DoF. However, WaveDyn is considered more suitable than OpenFOAM for the estimation of the heave motion as long as the surface-elevation remains undistorted (below breaking point).

Mooring Load

Across all events the experimental mooring line is tense. Assuming Hooke's law for springs, displacement is proportional to force. Figures 4.11a to 4.11d shows the mooring length time-series calculated from the DoF motion responses. Table 4.3 presents

the accuracy of each model using the correlation of the time-series and the estimation of the maximal load.



(a) ST1 *(b)* ST2

(c) ST3 *(d)* ST4

Figure 4.11: Mooring length time-series (proportional to mooring load) of the X-MED buoy under the four extreme events of increasing steepness

Due to the depth $(2.8\,m)$, the mooring length is always multiple times larger than the amplitude of a given motion. WaveDyn accuracies in pitch or surge become of minor importance in the estimation of the mooring load time-series and maximum load. OpenFOAM is less precise in the accuracy of the mooring time-series due to the inaccuracies in heave, especially after the main crest (i.e. after $12\,s$). OpenFOAM remains more accurate in the estimation of the maximum load apart for the steepest case due to the over-estimation in surge induced by the lower representation of the event. WaveDyn estimates with very good accuracy the maximum load in the mooring line as the error in estimation are less than 1% independently from the steepness of the event. Therefore, the gain in accuracy obtained by using OpenFOAM is irrelevant, making WaveDyn more suitable in the estimation of the maximum load in the mooring line independently from the steepness.

Parameter	Model	ST1	ST2	ST3	ST4
Correlation	WaveDyn	0.9717	0.9575	0.9305	0.9205
	OpenFOAM	0.9218	0.9249	0.9276	0.8861
% Max	WaveDyn	-0.5	-0.6	-0.7	-0.9
	OpenFOAM	0.07	-0.1	-0.2	1.2

Table 4.3: Correlation of time-series and estimation of the maximal load in percent of the experiment load (negative means under-estimation, positive over-estimation)

However, this conclusion on the loads is expected to be specific to the use of a short event generating only one maximum load. The non-capture of the surge motion brings the buoy to a different position than the actual one. In the possibility of an event happening around $20\,s$, the inaccurate position of the X-MED buoy found by WaveDyn is likely to influence the resulting motion responses.

Therefore, in the survivability assessment based on an irregular sea-state (i.e. long simulation), surge motion is also expected not to be captured accurately by WaveDyn resulting in a different position of the buoy. Then, an unlikely event can be faced differently in WaveDyn, hence generating an under-estimation (or over-estimation) of the loads. Therefore, the correct estimation of each DoF motion responses remains of main interest, in which WaveDyn shows concerning inaccuracies, especially, in the capture of surge motion responses identified as threatening WaveDyn's validity for extreme WSI.

4.4 Assessment of WaveDyn Limits

This section aims to parametrically assess the limit of WaveDyn to model extreme WSI as a function of wave-parameters: the wave height H, and the non-linear parameter, the steepness, kA.

4.4.1 Measure of Error

The correlation coefficient evaluates the similitude between two time-series. To quantify the error relative to a simulation (i.e. the inaccuracy), the Root Mean Square (RMS) is used because it represents the cumulative error. To make the measure of error independent from the amplitude of motion, RMS is normed by the maximum measured experiment motion for the given DoF, $max(x_{exp})$. RMS is defined as:

$$RMS = \frac{1}{max(x_{exp})} \sqrt{\frac{1}{N} \sum_{n=1}^{N} |x_{exp}(n) - x_{WaveDyn}(n)|^2}, \tag{4.5}$$

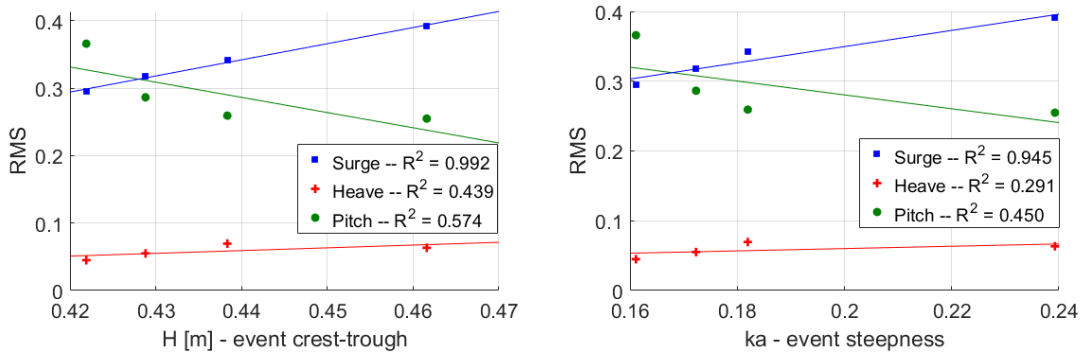
where, $x_{exp}(n)$ is the experimental motion response, $x_{WaveDyn}(n)$ is WaveDyn's solution, N is the number of point in the considered time-series ($[0 : 0.01 : 20]$ is 2001), and n is the motion response at time $n/100$ (n^{th} component of the motion vector). RMS can be calculated as a function of time by defining the length of the time-series from 0 to a specific time.

In the following investigation, the assessment of the correlation between two parameters is conducted using the coefficient of determination, R^2, defined as the square of the correlation coefficient previously introduced in section 3.3.2.

4.4.2 Parametric Analysis

The measure of error (i.e. RMS) for each DoF motion responses is calculated to relate the inaccuracy of WaveDyn per DoF to two event parameters: H, the maximum wave height, and kA, the event steepness. Figure 4.12a and 4.12b show respectively the evolution of error of surge (green squares), heave (red cross-signs), or pitch (blue dot), as a function of the events maximum wave height, H, or steepness kA. The R^2 between each DoF error and each event parameters - Table 4.4 - assesses a measure of correlation assuming a linear evolution (coloured lines).

According to coefficients of determination in Table 4.4, the error in heave and pitch correlates poorly with any of the two event parameters. However, the error in surge correlates with both. This suggests that the error in the surge response for a given



(a) Maximum wave height H *(b)* Event steepness kA

Figure 4.12: Parametric assessment of WaveDyn inaccuracy in each DoF

$R^2(RMS, --)$	Surge	Heave	Pitch
H	0.992	0.439	0.574
kA	0.945	0.291	0.450

Table 4.4: Coefficient of determination, R^2, between WaveDyn cumulative error for a given DoF and event parameters

event can be assessed as a function of these event parameters. Besides, surge motion is causing the inaccuracy of WaveDyn solution as the first DoF not to captured when the complexity of the event increases. Therefore, the parametric assessment of WaveDyn's limit in extreme WSI is performed as the assessment of error in surge response.

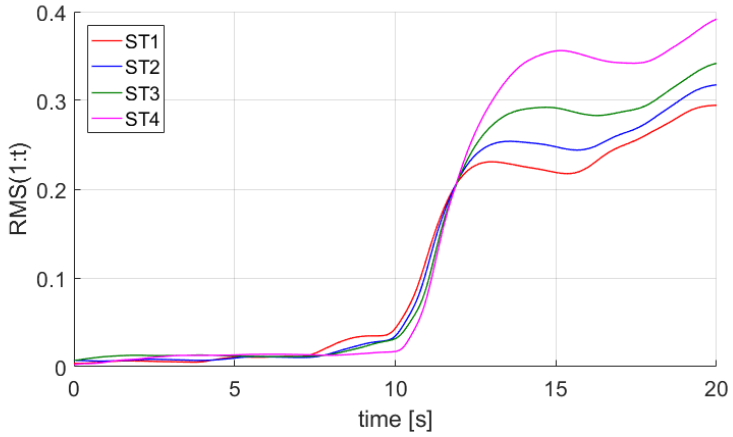


Figure 4.13: Cumulative error of the surge response as a function of time

Figure 4.13 shows the accumulation of error time-series for the four extreme events. Each case identifies a step highlighting a sudden increase of error, i.e. a point where WaveDyn's surge response suddenly deviates from the experiment. Figures (4.6a to 4.6d) of motion response time-series show approximately after the main crest that WaveDyn does not capture surge. Characterisation of these points, called 'deviation points', defines the limit in use of WaveDyn. As the error in the surge is found to be correlated to both event parameters (H and kA), deviations points are expected to be a function of these same parameters.

Deviation points identified the first time WaveDyn reaches its limit of validity in extreme WSI. The complexity at these times requires OpenFOAM. The coupled model developed between WaveDyn and OpenFOAM, will, therefore, use the parametric evaluation of the deviation points to swap from WaveDyn to OpenFOAM.

4.4.3 Methodology

Instantaneous steepness

To identify the time of deviation points, event parameters need to be changed to time-dependent parameters (i.e. instantaneous). H becomes the surface elevation η. The event steepness kA changes to the derivative of the surface elevation η relative to time, named Instantaneous steepness (IkA) and defined as:

$$ikA(t) = \frac{\partial \eta}{\partial t} \tag{4.6}$$

where η is the surface elevation at the buoy location input in WaveDyn model.

In potential flow theory, IkA is the vertical velocity (Equation 3.13 at $z = 0$) of the free-surface at a given location in space, which is proportional to the horizontal acceleration (time derivative of Equation 3.12 at $z = 0$):

$$w_{z=0} = -a\omega sin(kx - \omega t + \delta) = \frac{sinh(kd)}{\omega cosh(kd)} \frac{\partial u}{\partial t}_{z=0}. \tag{4.7}$$

Since WaveDyn model ignores viscous effects, the WEC motion is dominated by inertia forces (within the model). Hence, evaluating the vertical velocity - IkA - is equivalent to evaluate inertia forces acting on the WEC in the WaveDyn model.

Measure of deviation

The mathematical tool used to identify the lower angle of the step is the second derivative relative to the time of the cumulative error, $\frac{d^2RMS}{dt^2}$; the 'deviation'. As shown in Figure 4.14, the step in the RMS time-series generates a peak in the second derivative of amplitude relative to the severity of the angle that coincides with the angle time. Surface elevation η and IkA (absolute values) at this time identify the wave-parameters at deviation. The correlation (using R^2) between peaks of deviation and identified wave-parameters assesses WaveDyn's limit as a function of wave-parameters.

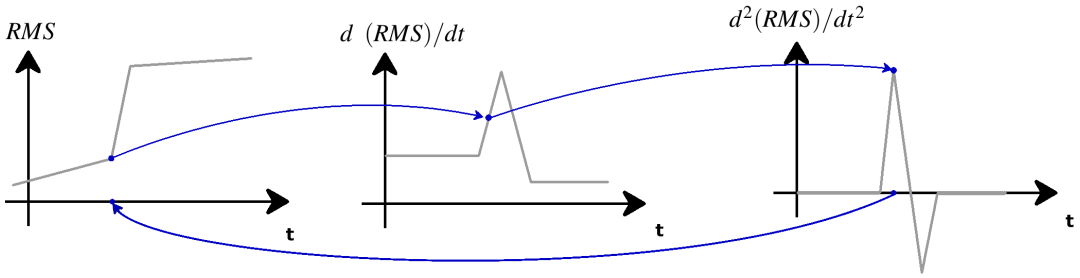


Figure 4.14: Evaluation of the lower angle in the step of the RMS time-series

To compare the four cases, the second derivative, $\frac{d^2RMS}{dt^2}$, uses the normed RMS as defined in Equation 4.5. This ensures that a larger peak in the second derivative is not due to an increase in the surge motion amplitude induced by the increase of event steepness, but rather to the increase of the wave-parameter itself.

4.4.4 Proof of methodology

Figure 4.15a shows the application of the methodology for the four events. On the left axis is the surface elevation η at WG#14 (light blue dotted line), IkA (magenta line), and experiment (black dashed line) and WaveDyn (blue line) surge responses. On the right axis is the RMS (red line) and the positive part of the measure of deviation (green line), as the only of interest. A low-pass butter-worth filter with a $4Hz$ cut-off frequency

is required to stabilise the measure of deviation and IkA.

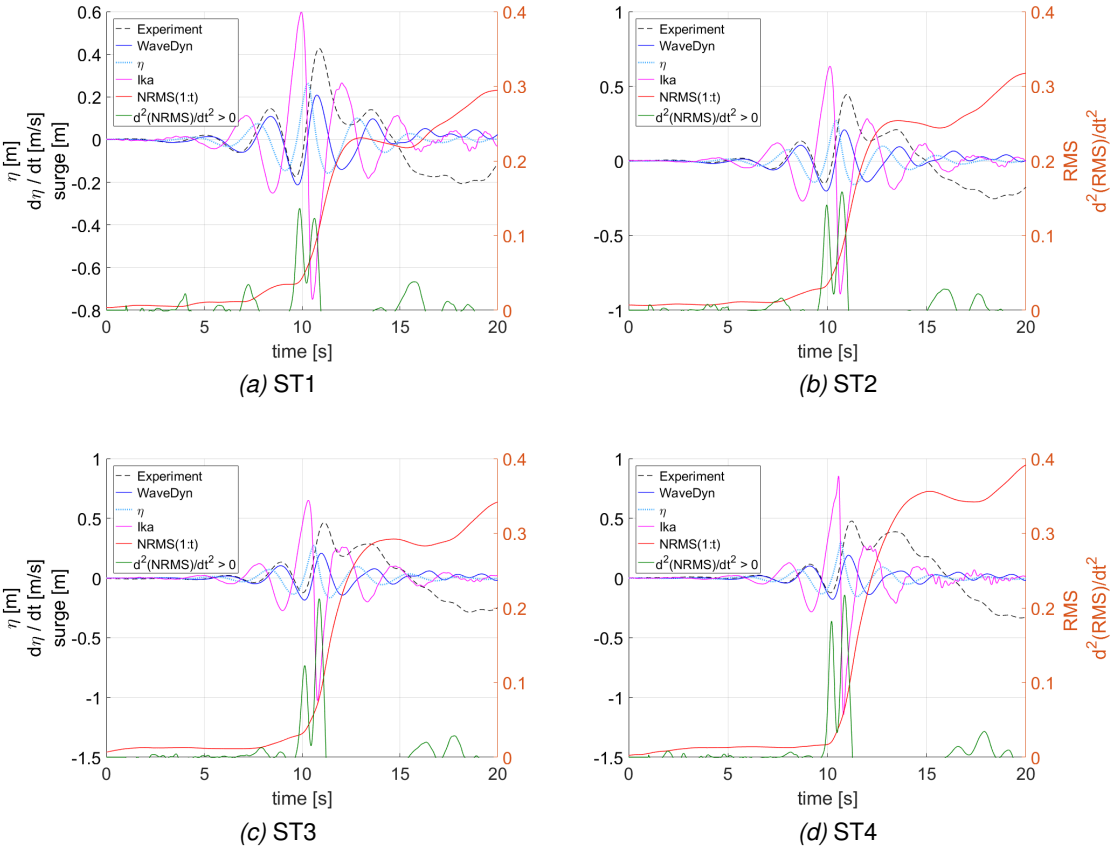


Figure 4.15: Identification of the first point of deviation of the WaveDyn surge solution by the second derivative of the cumulative error $\frac{d^2RMS}{dt^2}$

Figures 4.15 shows that the main angle in the cumulative error generates two peaks in the measure of deviation. This means that a first deviation occurs briefly followed by a second one. Smaller angles are also identified by the methodology and confirm that smaller angle in the cumulative error generates a smaller amplitude of peaks (e.g. Figure 4.15a around $7\,s$). This proves the use of the second derivative to identify and quantify the drastic changes in the cumulative error.

4.4.5 Results and Analysis

Deviation points

Deviation points are the largest peaks at which is identified surface elevation and IkA. Table 4.5 presents the coefficient of determination, R^2, between the peaks in deviation and each instantaneous parameter.

R^2	$\|\eta\|$ @max [m]	|IkA| @max
$max(\frac{d^2RMS}{dt^2})$	0.256	0.862

Table 4.5: Assessment of the relation between deviation points and instantaneous wave parameters

Maximum deviation does correlate more to IkA than the surface elevation according to R^2 values in Table 4.5, i.e. large deviation occurs for large IkA values rather than large surface elevation. Further analysis will only consider IkA.

As WaveDyn surge response matches experiment until deviation point. So, for lower value of IkA previous to deviation WaveDyn remains accurate. Therefore, a limit in IkA exists above which WaveDyn deviates. The IkA values identified at deviation points are increasing with events steepness (shown by Figure 4.16). So, the minimum of absolute IkA identified for case $ST1$, $0.58 m/s$, is - at least - the limit in IkA of WaveDyn.

Non-deviating points

Minor peaks in deviation are added to the analysis to consider points where the error increases while WaveDyn solution remains valid (i.e. non-deviating points). It aims to evaluate IkA as a function of deviation. The analysis is limited to the three largest peaks before the maximum (included) as smaller peaks will add points at the origin. This methodology is not applicable after the maximum because surge motion response is no longer captured.

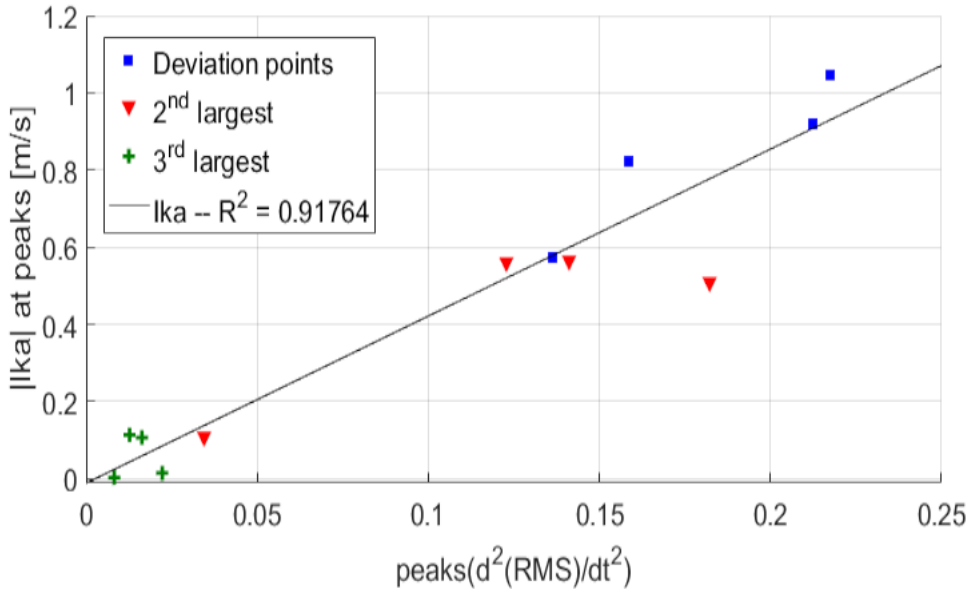


Figure 4.16: Evolution of IkA identified at deviation and non-deviating points
Origin $(0,0)$ is forced for the linear fit

Figure 4.16 shows that absolute IkA evolves linearly ($R^2 = 0.916$) with the measure of deviation (black line) considering deviation points (blue squares), 2^{nd} (red upside-down triangles) and 3^{rd} largest (green crosses). The linear assumption is forced at origin to consider that no deviation occurs for still sea-state. R^2 is higher than for deviation points only ($R^2 = 0.862$ Table 4.5) due to the presence of 5 points near the origin. If only the first two peaks were used, R^2 would remain similar as previously found (0.812).

However, the four events do not identify intermediate values in deviation ($[0.05; 0.1]$). The interval increases the value of R^2 as data points are gathered in two sets. It reduces the confidence in the linear evolution of IkA as a function of deviation.

4.5 Extended Study

The analysis is extended to a second set of extreme events based upon $ST1$ but with different shapes at buoy's position. It aims to identify WaveDyn's limit independently from the NewWave's shape. The assessment remains with OpenFOAM solutions using experimental data as the reference, to certify the constant accuracy of OpenFOAM and confirm the deviation of WaveDyn in surge responses.

4.5.1 Shifted Events

Experimental set-up

Experiments are also realised at the COAST PWT, but with different positions for the WGs (Annexe B.3.1). The focus location remains the front edge of the model, WG#8. WG#10, 11, and 12 are aligned to the centre of the X-MED buoy.

Twelve events are generated from the $ST1$ event by increasing the phase (δ_i in Equation 2.19) by $30°$ from $0°$ to $330°$. Changing the phase shifts the focus location in space and time. The shape of the surface elevation at a given position is therefore also changed. Figure 4.17 shows shifted events at the WG aligned with the front edge of the X-MED buoy: $ST1$, or $0°$ phase, is the 'crest focusing' event (blue line); $180°$ is the 'trough focusing' (green line); $90°$ is the 'up-crossing' (red line); and $270°$ is the 'down-crossing' (black line). All twelve events remain valid to NewWave theory.

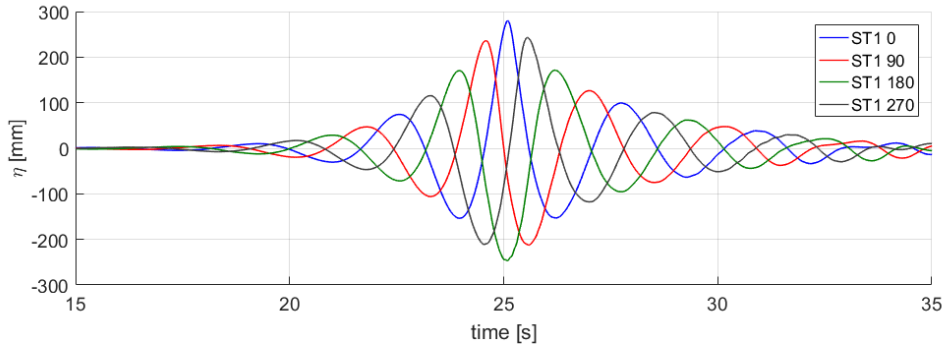


Figure 4.17: The surface elevation at the front of the X-MED buoy for the $ST1$ event with no phase shift, $90°$, $180°$, and $270°$ phase shift

Two minor differences exist between shifted and ST events. The distance between the X-MED centre and the WG aligned with it is slightly different. WG#10 in shifted events is aligned with the model centre, while WG#14 is $0.07m$ after the centre for ST events. Shifted events WG measurements are with the X-MED, while ST ones are wave-only cases. These differences are expected not to influence the analysis because the zero phase-shift event (green line) repeats $ST1$ (blue dashed line) accurately (Figure 4.18).

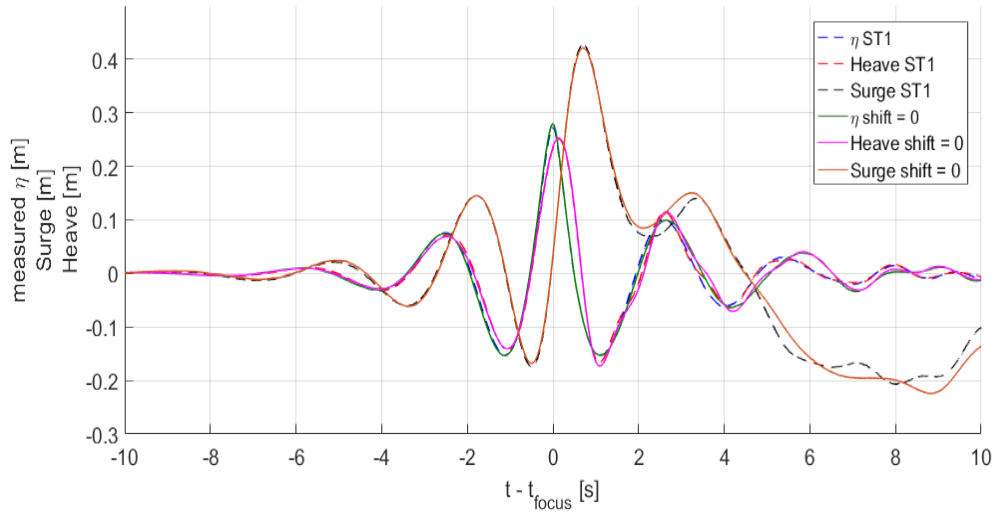


Figure 4.18: Repeatability of the $ST1$ event compared with the zero-phase one

The offset in time between motion responses and WG is unknown for the shifted events. Considering the repeatability of the zero phase-shift event with $ST1$, motion responses are manually shifted for the zero phase-shift to match with $ST1$. The offset in time vectors (surface elevation and heave response) is then applied to all shifted events.

Numerical models set-up

The NWT developed in Chapter 3 is deemed to remain valid to other experiment of extremes WSI made at the same PWT within minor changes. So, only the working-section (i.e. distance from the end of inlet to the beginning of relaxation-zone - Figure 3.7) of the NWT requires to increase by $1\,m$ to account the longer distance between the first WG and the X-MED buoy. Other NWT parameters (e.g. numerical beach, width, mesh resolution) are kept the same. This extended study is an opportunity to prove the adaptability of the NWT for other extreme WSI.

WaveDyn model is unchanged. WG#10 is used as wave-input. The differences between shifted and ST set-ups are considered negligible as part of the error in physical experiments.

4.5.2 Validation of Numerical Models

Figure 4.19 shows the accuracy of each DoF (surge (green x-signs), heave (red squares), pitch (magenta circles)) motion responses for OpenFOAM (coloured (dashed line)) and WaveDyn (coloured (line)) compare to physical experiment. It is similar to Figure 4.8 obtained for the four ST events and confirms previous conclusions on each model validity. Surge responses are available in Figure 4.20. All 3-DoF motion responses accuracy and time-series are available in Appendix D.2, Table D.2 and Figures D.1, D.2.

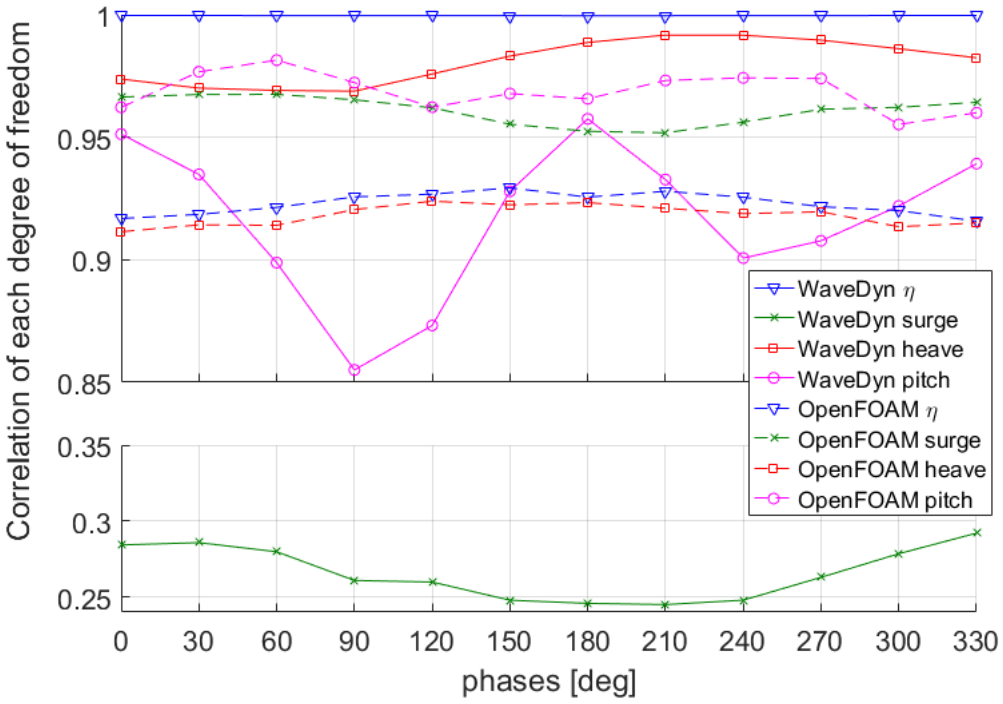


Figure 4.19: Correlation between each numerical model and the physical reference for each DoF motion response and surface elevation at WG#10

For each DoF and the surface elevation, OpenFOAM accuracy remains mostly constant independently from the event and above a 0.9 correlation. This validates the NWT for other extreme WSI done in the same PWT.

WaveDyn model obtains similar levels of accuracy than for ST events. Surface elevation and therefore heave responses remain at a higher accuracy than OpenFOAM. Pitch remains valid through all events, as it variates to a similar extent than for the four ST

events (between 0.85 and 0.95). Variation also suggests that accuracy in pitch is as more related to the shape of the event than the event non-linearity. For all events, the accuracy in surge (below 0.3) shows that the motion is not captured. Therefore, WaveDyn confirms to have attained its limit in extreme WSI due to the inaccuracy of surge. So, the methodology for the assessment of WaveDyn's limits developed in section 4.4.3 applies to shifted events.

4.5.3 Assessments of WaveDyn's Limits for Shifted Events

Identification of deviation

Figure 4.20 illustrates the identification of deviation points for each shifted (titles of each sub-figure). On the left axis is the surface elevation η (light blue dotted line), the IkA (magenta line), with experiment (black dashed line) and WaveDyn (blue line) surge responses. On the right axis is the RMS (red line) and the positive part of the measure of deviation (green line). Time-series are restricted to $5\,s$ to $15\,s$ for clarity, but, all WaveDyn solutions deviate from the experiment and do not recover as for ST cases. Figure 4.20 confirms the applicability of the method independently from the event.

The zero-phase event shows the group of two peaks in the measure of deviation around the focus time ($10\,s$), which were observed for the ST events. However, unlike the $ST1$ event, the first peak is smaller than the second is. This difference is due to differences in motion responses shown by Figure 4.18. Nevertheless, this difference does not remove the legitimacy of the methodology and its results.

Shifted events offer a wide range of shape of surface elevation. WaveDyn surge motion response is found to independently deviate from the experiment but at different rates. In Figure 4.20 all events present a step in the cumulative error with different severity of lower angle. All angles seem of lower severity than ST cases. Shifted events are therefore expected to identify deviation points of lower amplitude in deviation, which confirms the interest in their use.

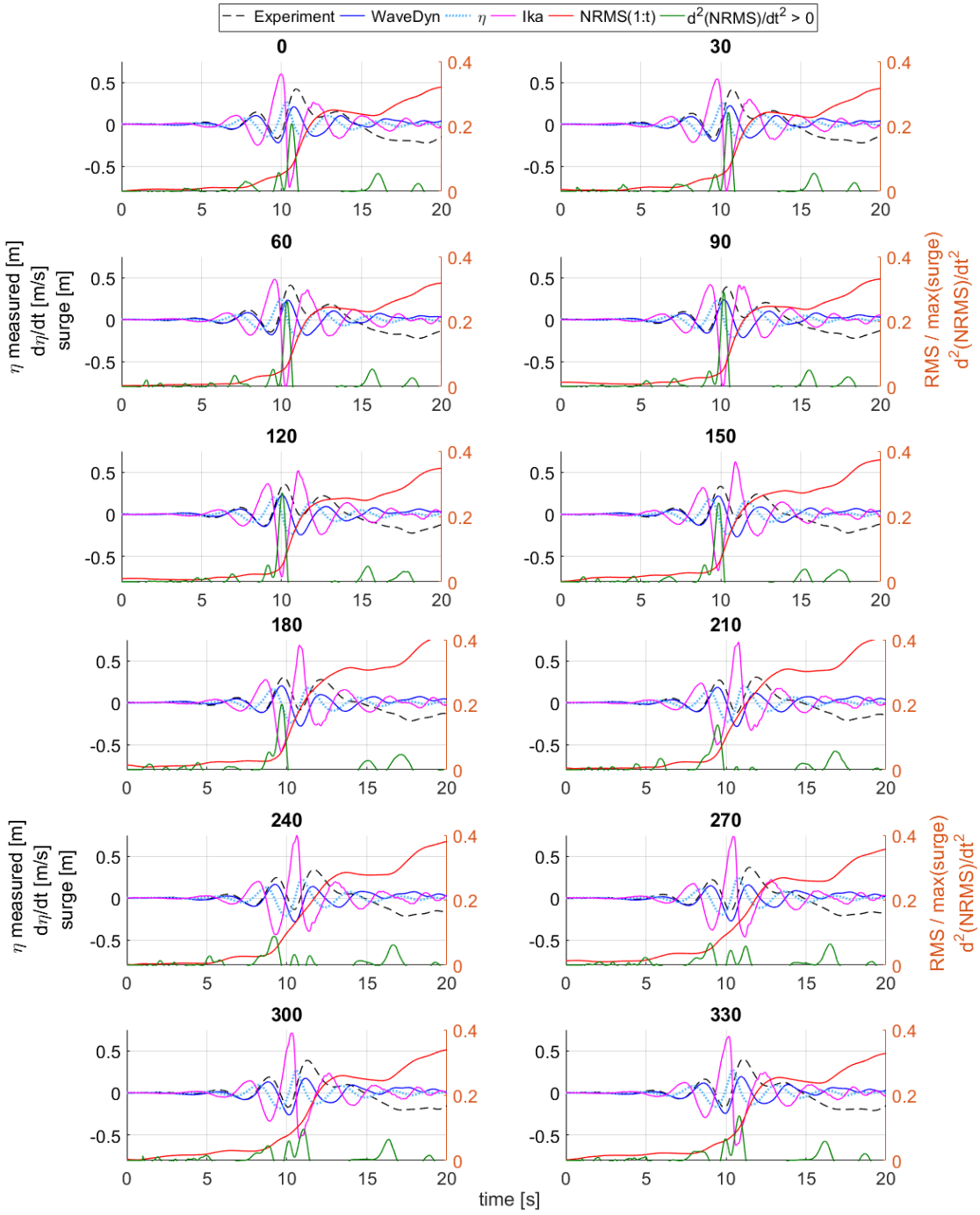


Figure 4.20: Identification of IkA as a function of the measure of deviation

A qualitative overview in Figure 4.20 identifies maximum peaks in deviation for absolute IkA above approximately $0.5\,m/s$. It confirms the first estimation of WaveDyn's limit in IkA made from the lowest IkA identified for ST cases in section 4.4.5, $0.58\,m/s$. Also, cases 150 and 180 shows that once deviation occurs, additional IkA values above the limit do not influence the solution. This highlights the existence of a limit in IkA above which WaveDyn is expected to deviate.

Additionally, for lower IkA values, WaveDyn accumulates error without deviating. Case 240 starts to deviate at $9.3\,s$ for an |IkA|$= 0.43\,m/s$. This confirms the interest of the investigation on non-deviating points conducted in section 4.4.5 to evaluate the evolution of the measure of deviation as a function of IkA.

However, the linear increase of IkA as a function of the measure of deviation found for ST events is debatable. Case 210 is the example opposite to previous conclusions. It shows a deviation point at $9.5\,s$ for an |IkA|$= -0.5\,m/s$ rather than for the maximum IkA. Besides, case 240 to 330 shows deviations of minor amplitude at deviation points compared to 0 to 150 cases for similar IkA values. Therefore, the characterisation of the evolution of IkA as a function of deviation is first conducted with shifted events only, before merging the two studies.

Evolution of IkA with deviation

The assessment uses the three largest peaks of deviation prior to the deviation itself. Figure 4.21 shows the evolution of IkA identified at deviation points (blue squares) assuming a linear evolution (blue line). Second (red x-signs) and third (red upside-down triangles) largest (i.e. non-deviating points) are also assumed to evolve linearly (red line). A linear (black dashed line) and a logarithmic (black line) fit are evaluated to all points and forced at the origin. The R^2 of each fit is calculated and added to the legend.

In overall, Figure 4.21 confirms that large absolute IkA values induce large deviations. The evolution of IkA as a function of deviation is found to be logarithmic ($R^2 = 0.766$) rather than linear ($R^2 = 0.597$). This suggests a difference in the evolution of deviation

for low IkA values compare to large ones.

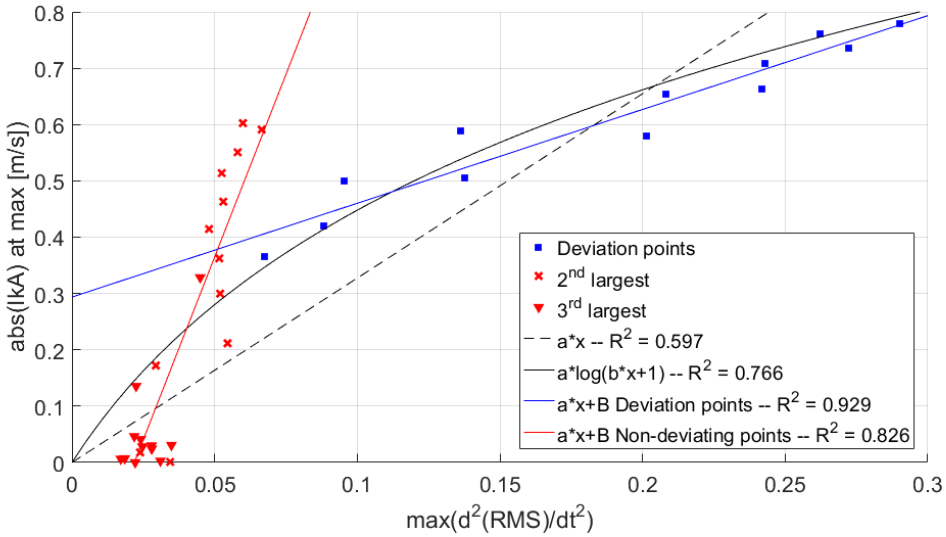


Figure 4.21: Evolution of IkA at deviation and non-deviating points
Origin is forced for linear and logarithmic fits

Identified IkA increases with deviation at a different rate between deviation and non-deviating points. As highlighted by respective linear fits ($R^2 = 0.929$ deviation points, $R^2 = 0.826$ non-deviating), the rate of increase is significantly higher for non-deviating points. This justifies the existence of a threshold in IkA. As long as the absolute IkA remains under the limit, an increase (in IkA) results in a low increase of deviation, meaning that the solution should remain valid. Above the given limit, deviation increases significantly for a small increase in IkA. Therefore, this limit is the validity limit of WaveDyn.

However, the definition of this limit remains unclear. Indeed, the second largest deviation peaks of many cases (0, 30, 60, 300 and 330) identify large IkA values above 0.5 for low deviation around 0.06, probably because the actual deviation occurs just after. It is similar to ST cases, where the step in error generates two peaks in deviation next to each other. Although for ST events the two peaks are of similar amplitude. In shifted events, the first peak remains small. This difference can be due to a slight offset in time for shifted events, that synchronises numerical solution with experiment, hence reduc-

ing deviation. As done for ST cases, the minimum IkA at deviations points - $0.36 m/s$, 270 case - can define the limit in IkA. However, this restrictive limit is overcome by the same cases (0, 30, 60, 300 and 330) with no deviation.

4.5.4 Definition of WaveDyn's Limit

Studies are merged for the final assessment of the limit of WaveDyn in IkA. Figure 4.22 adds to previous Figure 4.21 the ST events data points from Figure 4.16 as (green squares, green x-signs, green upside-down triangles). Logarithmic (black line) and linear (black dashed line) fits forced at origin are applied to the gathered data-set.

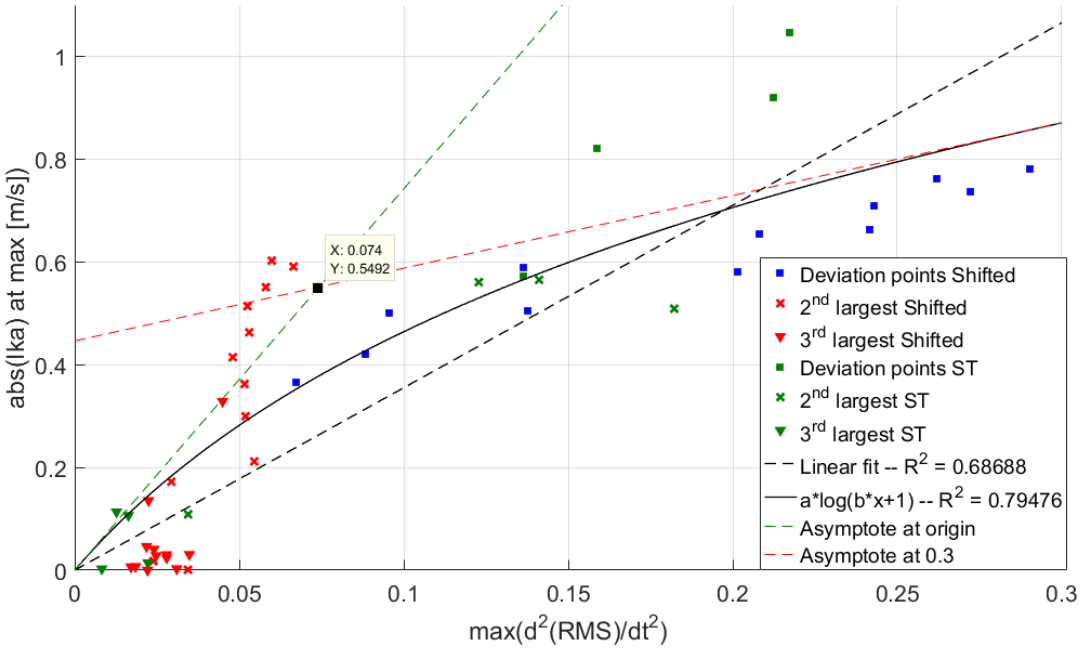


Figure 4.22: Evolution of IkA identified at deviation and non-deviating points for shifted and ST events

As expected, Figure 4.22 validates that WaveDyn deviates for large values of IkA. The logarithmic fit is found at slightly higher accuracy ($R^2 = 0.795$) than for the shifted events. This gives confidence in the use of both data-sets, and the extension of conclusions to future data-set. Although, this level of correlation remains not high. The inaccuracy is considered due to multiple deviation points of very low deviation amplitude (or even zero) identifying however small IkA. Also, deviations point for $ST 2-3-4$ cases identi-

fies IkA of very large amplitudes (three points of IkA$> 0.8\,m/s$) for relative low deviation. It is possible that the solution can be already considered as deviated because of the proximity of the previous deviation peak, or because the limit in IkA is already attained for the first peak. Despite reserve on the logarithmic evolution, the identification of the limit in IkA remains as two distinct patterns at low or high IkA values has been identified by the time-series.

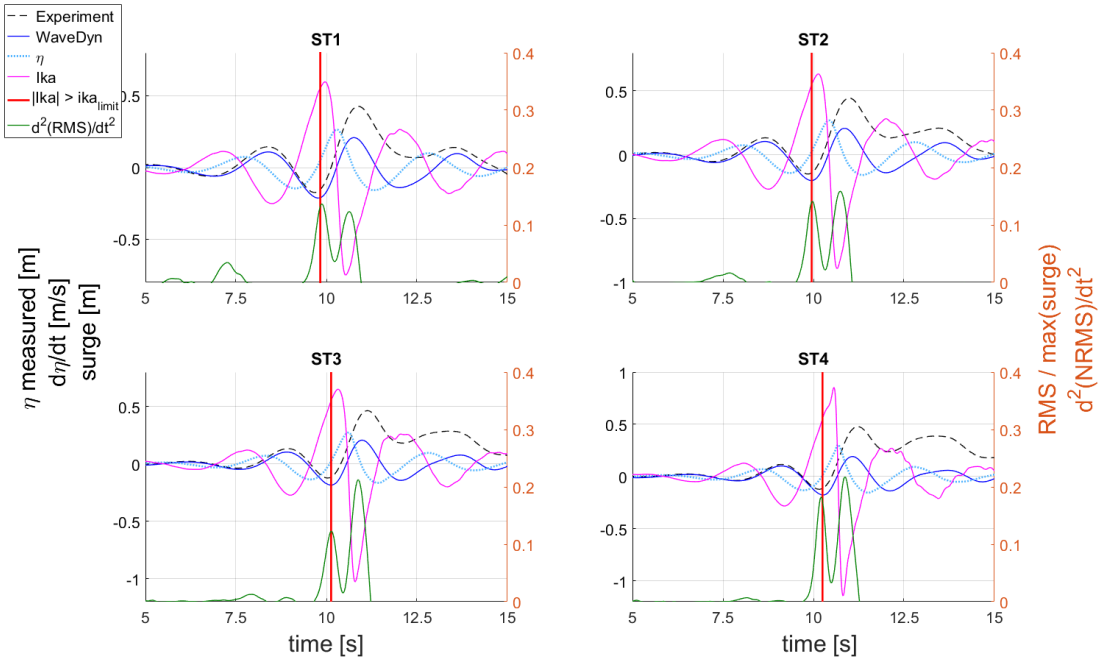
Asymptotes of the logarithmic fit in Figure 4.22 illustrate these two patterns. At origin (red dashed line) the asymptote outlines the low increase of deviation as long as IkA remain low. The second asymptote is defined at the end of the data-set end (deviation $= 0.3$) to illustrate the fast increase of deviation when IkA is too large. The intersection of the two asymptotes defines the moment in IkA from which the deviation changes from a slow increase to a sensibly larger one. Therefore, the limit of validity of WaveDyn is |IkA| $< 0.549\,m/s$.

Using the logarithmic since the best correlation, it is possible to define the limit in IkA differently. For example, a criterion over the deviation of 0.2 identifies a $0.7\,m/s$ limit, while a more restrictive 0.1 criterion gives a $0.46\,m/s$ limit. This technique of assessment is discarded as considered as arbitrary and because it does not illustrate the two patterns.

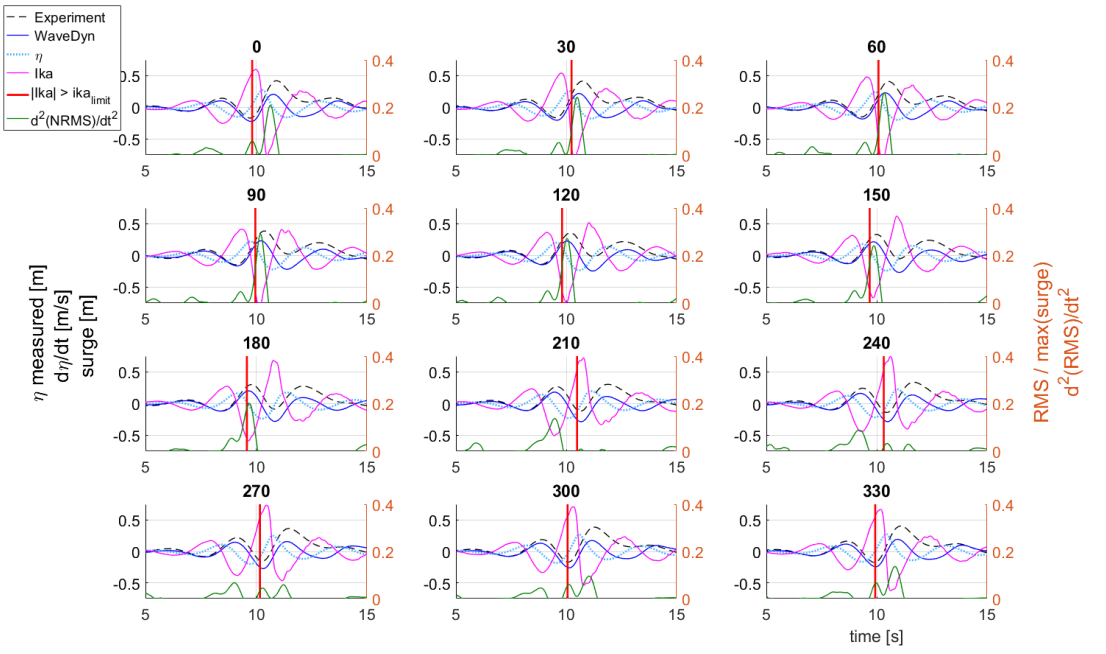
4.6 Application of Limit

In this section, the identified limit of WaveDyn in IkA is applied back to the two sets of events to assess its capacity to accurately identified deviation points. A long simulation of an extreme event embedded in an irregular sea-state is then used to prove the concept. It is in the meantime an assessment for the use of the limit as a trigger for the coupling between WaveDyn and OpenFOAM.

4.6.1 Focused and Shifted Events



(a) ST events



(b) Shifted events

Figure 4.23: Estimation of the deviation point using the limit in |IkA| $< 0.549 m/s$

In the following figures is identified as (red vertical line), the first point of absolute IkA above the limit - |IkA| $< 0.549 m/s$ - for ST (Figure 4.23a) and shifted cases (Figure 4.23b). The identified point is compared with the following peak in deviation: it is correct if the limit identifies the same deviation peak, restrictive for a previous one, and late for a past one.

The limit is found to be mostly accurate but restrictive and sometimes late. Deviation points for shifted events from 30 to 180 and the $ST1$ event are correctly identified. The second largest peak is identified for cases $ST2-3-4$, and shifted events 0, 300 and 330. However, for 210, 240 and 270 events, the limit is late and identifies points past deviation.

The error in the identification of the limit for these cases is due to deviation of minor amplitude generated from negative IkA minima which their opposite (i.e. $|ikA|$) would not have generated a deviation. For example, the minima in IkA for event 210, which does generate deviation, is of similar amplitude as the maxima of the 30 shift event, which does not generate deviation. It is also true between 240 event with 60; and 270 event with 90. It could be noted that these events are shifted by an additional π angle, hence wave-group profiles are opposite (i.e. $\eta[210] = -\eta[30]$). Therefore, this error shows a lack of accuracy in the identification of deviation, while suggesting that, for the same amplitude of IkA, deviation only occurs for negative IkA values. The limiter can probably be further specified by differentiating negative and positive IkA values.

4.6.2 Extreme Irregular Sea-State

The limit in IkA is now applied to an extreme irregular sea-state case not used for its assessment. Figure 4.24 compares surge motion responses for the experiment (black line) with WaveDyn (blue line). The red vertical line marks the first absolute value of IkA above the limit while the green line is the measure of deviation.

Figure 4.24 shows that the limit in IkA is late in the estimation of deviation when this one is defined by the maximum of the measure of deviation (occurring at $t = 20 s$). However, after the identified point, the amplitude of deviation becomes less significant, whereas

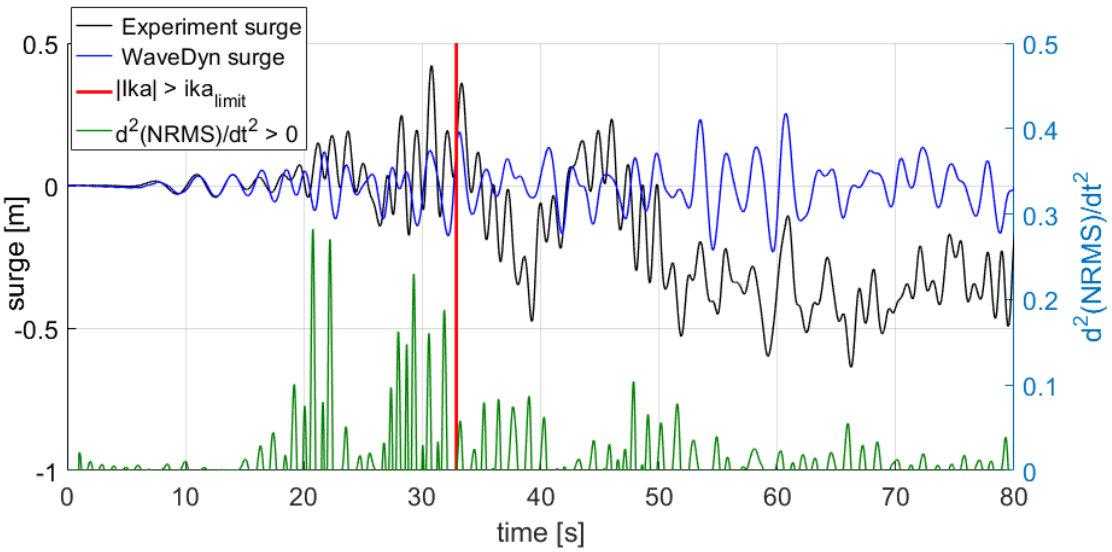


Figure 4.24: Estimation of the deviation point using the IkA limit applied to an irregular extreme sea-state

it remains large after the deviation maximum. The deviation precedes a large surge motion of low frequency, which is not captured by WaveDyn. This probably indicates that the current limit evaluates an upcoming slow drift in surge. This confirms the first assessment of WaveDyn model using the ST cases, where WaveDyn was found not to capture low frequency surge motion.

4.7 Conclusion to Chapter 4

Using the four events of increasing steepness, OpenFOAM Numerical Wave Tank (NWT) is found to remain valid independently from the non-linearity of the extreme events or the Degree of Freedom (DoF). The validity of WaveDyn for extreme events is questionable as motion responses are proven to be dependent on the DoF. For all events, heave and pitch motion responses are captured, while the surge is not. Therefore, the surge response is used to identify WaveDyn validity limit.

Event wave height and steepness correlate to the normed cumulative error in surge responses. However, surge motion responses are identified to deviate (i.e. no longer capture) from experiment due to a large derivative of the surface elevation relative to time, rather than a large surface elevation.

Additional experiments of extreme events of different shapes based on the least steep NewWave event confirm this while highlighting the existence of an upper limit in the derivative. The limit defines the change between WaveDyn being valid at low derivative value, to its validity decreasing suddenly above the limit. It is found as |Instantaneous steepness (IkA)| $< 0.549 m/s$.

Applied to all events, the limit demonstrates its use to identify times of deviation, while found to be restrictive with uncertainties for negative IkA values. For an irregular extreme sea-state, the limit is found late in the identification of the deviation point, but do identify the deviation induced by a slow surge drift motion of low frequency. The limit can, therefore, be used as the trigger time in the coupling between WaveDyn and OpenFOAM.

Chapter 5

Coupling of WaveDyn with OpenFOAM

In this chapter, the mid-fidelity numerical model WaveDyn is coupled with the Numerical Wave Tank (NWT) developed in OpenFOAM and previously detailed in Chapter 3. The coupling strategy consists of switching from WaveDyn to OpenFOAM at a time determined by WaveDyn validity threshold identified in Chapter 4. The coupled model benefits from the low computational cost of WaveDyn by reducing the usage of Computational Fluid Dynamics (CFD) to the identified Wave Structure Interaction (WSI) requiring the level of physics (i.e. assumptions on Navier-Stokes (NS) equations) of CFD (Reynolds Averaged Navier-Stokes (RANS)). Therefore, the coupled model aims to be able to simulate long time-series presently unaffordable in CFD, while improving the fidelity of the mid-fidelity model. This solution, named 'time-coupling', has been identified in section 2.3.3 as the most suitable coupling approach for the two numerical models compared to other coupling strategies.

This Chapter starts with a description of the time-coupling strategy by detailing the steps of the coupled simulation and the exchange of information between OpenFOAM and WaveDyn. The developments made to start a CFD simulation from non-still water to account for the build-up of the wave-field; and to specify the Rigid Body Motion (RBM) state (i.e. position, velocity and acceleration) different from at rest output from WaveDyn, are presented. The coupled model is assessed over the heave decay test before being applied to simulate extreme WSI using the $ST1$ case. Finally, the limit of fidelity of WaveDyn evaluated in Chapter 4 is applied within the coupled-model to identify the switch between WaveDyn and OpenFOAM for the irregular sea-state case used in section 4.6.2.

5.1 Time-Coupling

5.1.1 Strategy and Key Steps

Illustrated in Figure 5.1, the strategy consists of switching between models depending on the complexity of the WSI defined from the WaveDyn validity threshold (i.e. Instantaneous steepness (IkA) $|IkA| > 0.549 m/s$ (section 4.5.4). At the initial time, $t = 0$, WaveDyn models the WSI, while OpenFOAM is not used until :

1 An event outside of WaveDyn's validity range occurs. It is the hot-start time t_{hot}.

2 Solver changes to OpenFOAM, as an increase in the level of physics is required (i.e. potential flow to RANS). WaveDyn solver waits for OpenFOAM inputs (**4**).

3 Using a build-up period, t_{minus}, OpenFOAM sets-up the NWT wave-field (section 5.2.1) and the rigid-body motion-state (i.e. position, velocity and acceleration) at $t_{hot} - t_{minus}$ (sections 5.2.2 - 5.2.3).

4 OpenFOAM CFD simulation starts at $t_{hot} - t_{minus}$, while WaveDyn imposes OpenFOAM solution of loads and motion-state via the External Load Controller (ELC) (section 5.1.2): 2-ways solving (section 5.4).

5 The level of physics is no longer required and OpenFOAM stops, while WaveDyn continues without the ELC, i.e. WaveDyn-only.

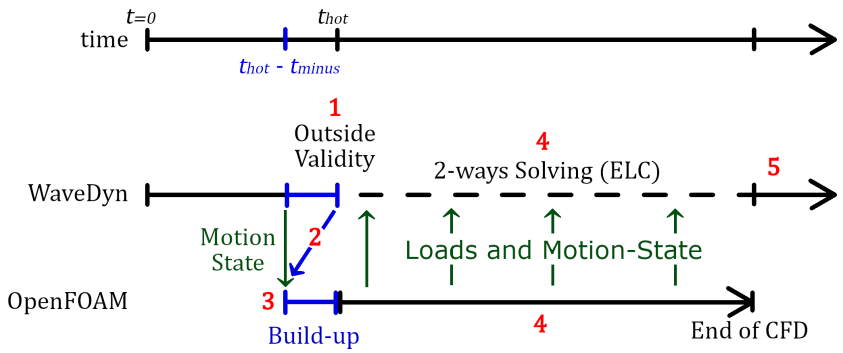


Figure 5.1: Time-coupling strategy as function of time
Numbers, such as **1**, corresponds to a step in a given coupled simulation

Step **5** is not investigated in the present study because in all events (ST and shifted) WaveDyn does not recover after deviating in surge (Figures 4.6 - 4.20). Therefore, the following investigation is restricted to simulations starting with WaveDyn, swapping to OpenFOAM at t_{hot} and stopping after the first use of OpenFOAM. Continuing a WaveDyn-only simulation when the use of CFD is no longer necessary is left for future work.

So, the proof of concept of the coupled model (section 5.4.3) consists in modelling one long irregular extreme sea-state WSI, where WaveDyn rules the simulation until the trigger is activated once from which OpenFOAM hot-starts and feeds solution back into WaveDyn until simulation end. This development is fitted to the objective of the present study since the computational cost of the coupled model will decrease compare to CFD only simulation, while maintaining the fidelity to the level of CFD.

The two solvers use different time-steps. Imposing a time-step to a solver induces instabilities, especially for OpenFOAM, where the Courant-Friedrichs-Lewy (CFL) condition (section 3.1.2) determines the time-step. During the 2-ways simulation, the OpenFOAM solution is imposed at each time-step of the WaveDyn solver using a linear interpolation between the two closest time-steps evaluates the input to the ELC.

5.1.2 External Load Controller

After the start of OpenFOAM at $t_{hot} - t_{minus}$ from WaveDyn solution, hence during the 2-ways simulation, OpenFOAM solution is imposed to WaveDyn as the only accurate solution.

The ELC applies an additional load and moment to WaveDyn model. So, to maintain the conservation of momentum between models - Newton second law of motion, $\sum \overrightarrow{F} = m.\overrightarrow{d}$ and $\sum \overrightarrow{M} = I.\frac{d\Omega}{dt}$ - during 2-ways solving, the balance of forces ($\overrightarrow{F}$) and moments ($\overrightarrow{M}$) between each model is assured by the ELC as:

$$\sum \overrightarrow{F_{OpenFOAM}} = \sum \overrightarrow{F_{WaveDyn}} + \overrightarrow{F_{ELC}}, \tag{5.1}$$

and,

$$\sum \overrightarrow{M_{OpenFOAM}} = \sum \overrightarrow{M_{WaveDyn}} + \overrightarrow{M_{ELC}}. \tag{5.2}$$

Therefore, the definition of the ELC force is defined from the difference between the two conservation of momentum. The rigid-body is subjected to three sources: gravity, hydrodynamics and ones inherent to the system (e.g. mooring, Power Take Off (PTO)). It is assumed that the forces inherent to the system are modelled the same way in both models, so their contribution can be simplified from the balance of forces and moments. In the application to the X-MED buoy, the mooring line is represented as a linear spring in both model and hence is simplified from equations. OpenFOAM models the hydrodynamics force as the sum of pressure, $\overrightarrow{F_p}$, and viscous, $\overrightarrow{F_v}$, forces. WaveDyn models it as the sum of radiation, $\overrightarrow{F_r}$, excitation, $\overrightarrow{F_e}$, and hydrostatic, $\overrightarrow{F_h}$ forces (Equation 4.2 in section 4.1.2). Therefore, the ELC force (moment is similar) is defined as

$$\overrightarrow{F_{ELC}} = \overrightarrow{F_p} + \overrightarrow{F_v} - \overrightarrow{F_h} - \overrightarrow{F_r} - \overrightarrow{F_e}. \tag{5.3}$$

OpenFOAM forces and moments are defined in the global coordinate system. Moments are transferred to the centre of gravity within the coupling script before being applied in WaveDyn.

Scripts developed for the coupling are available in Appendix E.3: the ELC, Appendix E.3.1; the *Allrun* responsible for the hot-start of NWT, Appendix E.3.2; the *python* interpolation WaveDyn's rigid-body motion state for the NWT hot-start, Appendix E.3.3; and the *forcesExtract* script, Appendix E.3.4, which generates the ELC input from OpenFOAM solution.

5.2 Starting a WSI CFD simulation from an Advance Time: Hot-Start

Starting the modelling of WSI in CFD from non-still water and a RBM state different from rest at an advance time is named 'hot-start'; steps 3 and 4 in Figure 5.1. Up until the threshold in validity (t_{hot}), WaveDyn model governs the simulation. OpenFOAM

starts from an anterior time (t_{minus}) to account for the building-up of the wave-field, from which is imposed WaveDyn RBM solution (i.e. position, velocity and acceleration).

To assess the capacity of OpenFOAM to hot-start while removing the influence of the coupling, the assessment is conducted only using OpenFOAM. The idea is to assess if OpenFOAM can hot-start from a previously non-hot-started OpenFOAM simulation. In light of the coupling with WaveDyn, accessible information is restricted to the RBM-state and the linear solution of the wave-field at advance-time. The objective is to remain consistent with the equivalent non-hot-started simulation, named a 'conventional' start such as proceeded in Chapter 3. The hot-started solution aims to maintain the validity of conventional ones while minimising the computational effort. Wave-field, rigid-body position and rigid-body motion (i.e. velocity and acceleration) are investigated separately because their requirements are independent of each other. Requirements are merged at last to assess the hot-start for WSI. Besides, investigations are uniquely based on CFD to avoid potential difficulties induced from the coupling. As the interest of the hot-start is to reduce the computational effort compare to CFD only simulation, the assessment of its reduction is added to the investigation.

5.2.1 Advance Conditions of Wave-field

Method

The correct wave-field at hot-start cannot be simply imposed across the NWT as it is unknown. The same linear superposition of waves components is defined at the inlet of the NWT. However, for hot-started simulation, the linear superposition is imposed along the NWT at advance-time, rather than starting from still water. So, at an advance time in an hot-started simulation, the surface elevation is imposed according to the linear propagation in space and time of the wave-group (Equation 2.16, section 2.1.3).

The wave-field imposed at hot-start does not considered radiation and diffraction of the rigid-body. This approximation induces uncertainties which should increase when the rigid-body motion becomes large. However, as the wave-fields of interest are extreme - i.e. of high energy - and unidirectional, diffracted and radiated waves are expected to

be of minor amplitudes relative to the input wave-field and will be merged within it as the wave-group propagates. Therefore, the present approximation is expected to have minor influence on the rigid-body motion response. The implementation of diffracted and radiated wave-field in addition to the main wave-field at hot-start is left for future work.

As the present study concerns extreme non-linear events, this solution is known to be inaccurate due to wave-wave interaction (as shown by Figures 4.9, and discussed in section 2.1.3). However, hot-start and conventional start both use the same linear superposition at the inlet. The difference sits in the building-up of the wave from a linear solution (hot-start) at an advance time, rather than from still water (conventional) at the initial time, $t = 0$. If the advance time is zero, the hot-start simulation imposes a linear superposition along the NWT. This imposed solution is still water as defined when the linear superposition is generated, hence it is the conventional solution. So, as the advance time decreases towards 0, the hot-start solutions converge to the conventional one.

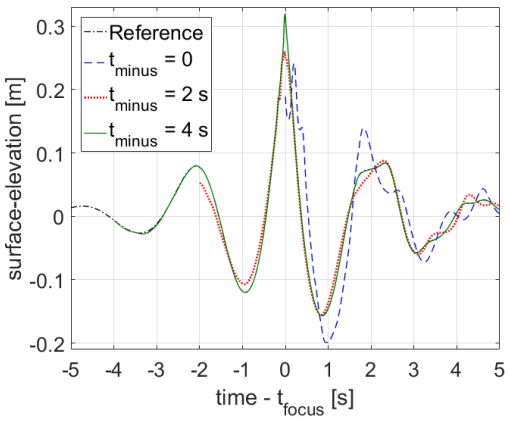
The present investigation is expected to work since the wave-field of interest are focusing waves which have linear profiles until near the focusing event where main non-linearities occur. Linear theory used at hot-start is therefore expected to define accurately these linear profiles until times near the focus. The applicability of the present method might be debatable when building-up an irregular wave-field. Such investigations are left for future work.

The assessment of the influence of the build-up period, t_{minus} (Figure 5.1), on the simulation of a focused wave is carried out. The investigation is conducted using 2-dimensional ($2D$) wave-only simulations of the four focus events of increasing steepness. The solutions obtained in section 3.5 are used as the reference. The hot-start time is set to be equal to the focus time, $t_{hot} = t_{focus}$. The build-up period, t_{minus}, is incrementally increased from $0s$ to hot-start, t_{hot}. Predictions of the hot-start model are compared to the result from the conventional start over surface elevation, velocity and

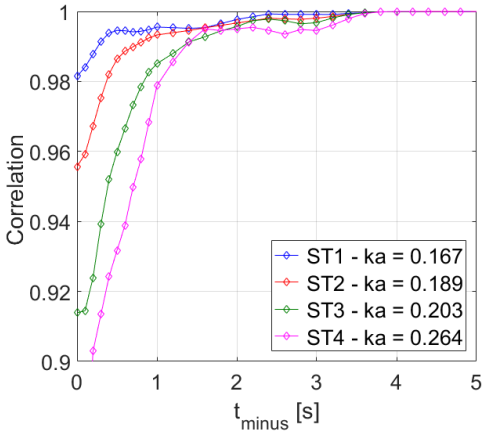
pressure profiles at focus location. The four events of increasing steepness are used to investigate the influence of non-linearities on the input wave field. Build-up periods are expected to increase with non-linearity because the linear solution increasingly deviates from the actual non-linear propagation with event steepness (Figures 4.9).

Results and analysis

Figure 5.2a shows the surface elevation time-series at the structure location for the steepest event $ST4$ for increasing build-up periods: no build-up (blue line), $2s$ (red line), and $4s$ (green line). The conventional set-up (i.e. starting from still water) is the reference (dash dotted line). Similar figures are obtained for least steep events, but $ST4$ being the most non-linear one generally shows the greatest differences between build-up periods. Figure 5.2b shows for all events ($ST1$ (blue line with dots), $ST2$ (red line with dots), $ST3$ (green line with dots), $ST4$ (magenta line with dots)) the correlation (section 3.3.2) between the surface elevation in the conventional set-up and that obtained for increasing build-up periods. Correlation is calculated after adjusting the length of the reference time-series to the length of the hot-start one.



(a) Surface elevation at the structure location (i.e. model centre) for the steepest event ($ST4$) comparing hot-started simulations of increasing build-up period to a conventional start



(b) Convergence of the correlation in the surface elevation as a function of the build-up period for events of increasing steepness

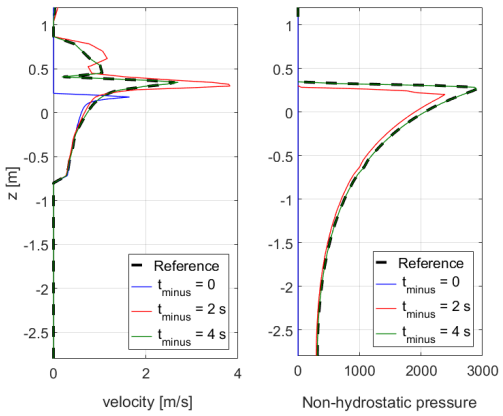
Figure 5.2: Assessment of build-up period t_{minus} required for hot-started wave-field to converge to a fully developed one, based on the surface elevation

Figures 5.2a and 5.2b show that the surface elevation converges to the solution of the conventional start as the build-up period increases. Correlation converges for any event steepness; and, when no build-up period is used, $t_{minus} = 0$, decreases as steepness increases. The steepest event, $ST4$, hot-started with no build-up period, (blue line) in Figure 5.2a, obtains a 0.865 correlation. The least steep event, $ST1$, is quite accurately represented by the linear solution and obtains a 0.979 correlation when no build-up period is used. This highlights that the less non-linear the event, the more accurate is the imposed linear solution at hot-start. Then, as t_{minus} increases (i.e. as $t_{hot} - t_{minus}$ decreases) the linear imposed solution is more representative for any given event because the inaccuracies due to the linear propagation decrease. Therefore, the accuracy of the solutions increases, and the difference between $ST1$ and $ST4$ decreases. For a long enough build-up period, $t_{minus} = 4\,s$, the error in the solution for all steepness approaches 0, meaning that the linear imposed solution is acceptable.

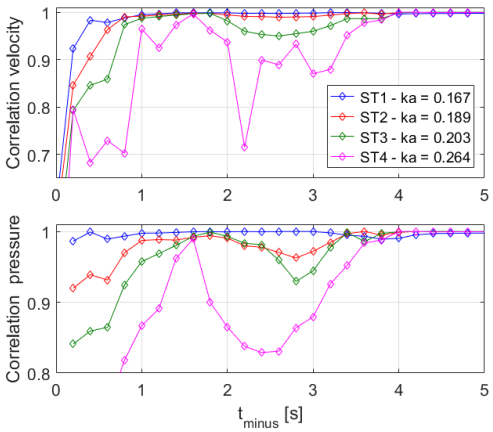
However, if a lower correlation level of the surface elevation is deemed sufficient (e.g. 0.99), the require t_{minus} is found to be a function of the non-linearity of the event. $ST1$ only requires a $0.5\,s$ build-up period, while $ST4$ a $1.6\,s$ one.

Figure 5.3 presents the influence of the build-up period, t_{minus}, on the simulated pressure and velocity profiles at the buoy equilibrium location at focus time. Profiles of the steepest event are shown in Figures 5.3a. Figures 5.3b shows the evolution of the correlation between hot-start and conventional start simulations for velocity and pressure profiles as a function of the build-up period, t_{minus}.

Figures 5.3a and 5.3b confirm that the hot-started solution converges to the conventional one as the build-up period increases. Convergence is reached at $t_{minus} = 4\,s$ for all 4 cases for both parameters, as hot-started profiles (green line) are matching the reference (black dashed line). Therefore, for the four NewWave based events used in the present study, hot-starting the simulation of wave-only cases $4\,s$ prior to the focus time is found to reproduce, at the rigid-body location, the wave-group modelled using a conventional CFD start.



(a) Velocity and pressure profiles at focus time, at the structure location (i.e. buoy centre), for the steepest case, comparing hot-started simulations of increasing build-up period, to a conventional start



(b) Convergence of the correlation of the velocity (top) and pressure (bottom) profiles as a function of the build-up period for events of increasing steepness

Figure 5.3: Assessment of build-up period t_{minus} required for hot-started wave-field to converge to a fully developed one, based on velocity and pressure profiles

However, the convergences are not as steady as for the surface elevation. Figures 5.3b show a first increase in correlation from $t_{minus} = 0$ to $t_{minus} = 1.6s$, followed directly by a decline. This decrease appears more significant for steeper events, and could already be noticed in the comparison of the surface elevation in Figure 5.2b with less significance. This error is due to the non-capture of the main crest of the event for a build-up period larger than $t_{minus} = 1.6s$, which therefore highly influences the description of the velocity and pressure profiles at the focus location. Indeed, for $t_{minus} > 1.6s$ the main crest is just imposed at $x = 0$, whereas for lower values, the linear solution inserts the main crest in the NWT. Therefore, for simulations hot-starting with $t_{minus} \leq 1.6s$, the solution builds-up from a linear solution including the main event. So, the hot-starting technique shows difficulties to build-up a NewWave based event when is imposed the main crest of the event at the inlet. By selecting a long enough period, the main crest is generated after a first crest. Therefore, to avoid these errors, the build-up period selected for further study is $t_{minus} = 4s$.

The applicability of this result to other events would require further investigation. In the

future application to a long irregular sea-state, the level of accuracy of the hot-started sea-state using a $4s$ build-up period is expected to be lower than for ST events. The accuracy obtained using a $4s$ build-up period is due to the representation of a short wave-group. However, for wave-groups of longer wave-period, the applicability is also questionable as the overall event length increases. As the parameter is applicable for the four events which peak frequency increases, it is expected to be applicable for NewWave events of larger peak frequency (i.e. lower period), hence $fp \geq 0.356Hz$. So, further investigations are required for the applicability of the parameter for events of longer period.

On the other hand, the build-up period cannot be extended indefinitely because it will remove the interest of the coupling. So, despite the expected inaccuracy, the $4s$ build-up period is later used in the application of the coupling for an irregular sea-state.

The assessment of the build-up period presents also an interest concerning the modelling of the wave-field from a CFD users perspective independently from the coupling. Indeed, it demonstrates that - for the four NewWave base events independently from the non-linearity - imposing the event linear solution $4s$ before its main event reproduces the given event as accurately (correlation above 0.9999) as a conventional start. The modelling of these events can be reduced from this time, and avoid modelling the first $6s$ of building-up of the wave field from still water. At $4s$ before the main event, the imposed linear solution is close to still water, where only the crest preceding the main is imposed linearly, Figure 5.4. So, it seems that the requirement to start modelling wave in CFD from still water is questionable and that wave-only cases can be restricted closely to the event.

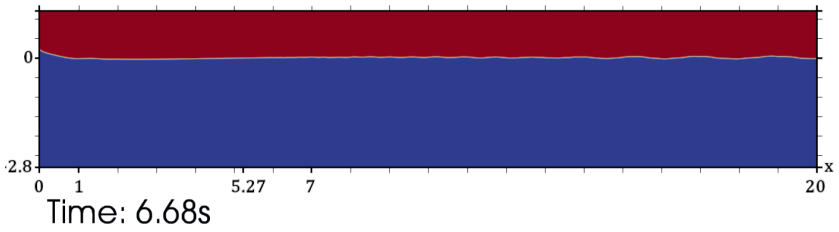


Figure 5.4: Linear solution imposed at advance time in the $2D$-NWT for the $ST4$ case

The applicability of this result is expected to be wider than the build-up period as it only means to start simulation at the first wave independently from the wave-group. The saving in computational effort compared to a conventional set-up goes from 25% for the least steep case to 12% for the steepest one. The hot-start removes the need to model the initial propagation of the wave-group onto still water, at which time the complexity of the flow remains low relative to the rest of the simulation. The level of complexity of the main crest increases with non-linearity, and so the rest of the simulation requires proportionally more computational effort than at the start.

Future developments on the build-up period of the wave-field will be focusing on a parametric assessment function of wave-parameters such as the local wave-steepness. The investigations would be conducted using irregular sea-states or NewWave events of smaller wave peak frequency.

5.2.2 Advance Position of the Body

Initial mesh deformation `deformDyMMesh` library

At hot-start, the position of the rigid-body is necessarily different from its equilibrium and needs to be specified accordingly into the mesh. A first method consists of generating the geometry at the advance position and inserting it into the mesh. This is the procedure used for the heave decay (section 3.8.1). It uses a Computer Aided Design (CAD) software, hence requiring a third coupling. This new mesh is undeformed with the geometry at the position different from equilibrium. The mesh quality is at its best and can only decreases as it deforms when the rigid-body moves. So, at equilibrium, the mesh is deformed and of lower quality.

To maintain the mesh quality at equilibrium while avoiding the use of CAD, a new OpenFOAM pre-processing library is developed. It uses uniquely the original undeformed mesh with the rigid-body at equilibrium and deforms the mesh according to the structure position at hot-start. The library, called `deformDyMMesh`, incrementally deforms

the original mesh by a $deformation$ of amplitude:

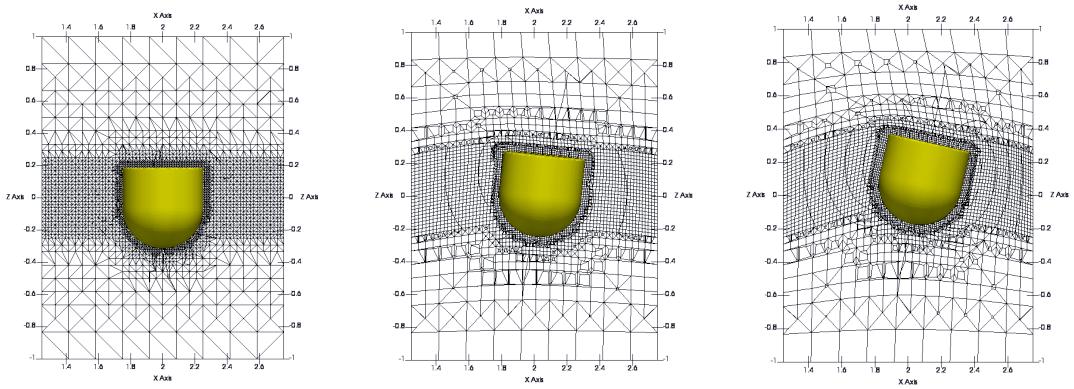
$$deformation = \texttt{deformPosition} \times \frac{i}{\texttt{deformIter}}. \tag{5.4}$$

where, `deformPosition` is the 6-Degree of Freedom (DoF) offset of the rigid-body from its equilibrium; i the iteration number out of the total number of iterations, `deformIter`.

`deformDyMMesh` is based on the `rigidBodyMotionSolver` solver detailed in section 3.1.6. `deformDyMMesh` uses as input an user-specified offset in displacement instead of one solved by the RBM solver. In the case of the solver, the deformation remains small as defined between two time-steps. The incrementation is necessary for `deformDyMMesh` to impose a deformation small enough to be implemented into the mesh. The number of iterations, `deformIter`, can be reduced to increase the efficiency of the mesh-deformation, up to the limit of a small deformation. The source code and a tutorial are available within the CCP-WSI Code Repository (2018) and are attached in Annexes E.2, E.2, E.2.

As an example, Figure 5.5 illustrates the capacity of the `deformDyMMesh` library applied to an offset of $0.06m$ in surge direction, $0.14m$ in heave, and 10° angle in pitch of the X-MED buoy. Figure 5.5a shows the original mesh with the structure at equilibrium used as input. Figure 5.5b shows an intermediate step in the mesh deformation, and Figure 5.5c shows the final deformed mesh.

When assigning the advance position of the rigid-body into the mesh at hot-start, `deformDyMMesh` is more efficient than the method using CAD, but take at least as long as the mesh generation (i.e. `snappyHexMesh`). The generation of the original mesh in Figure 5.5a takes $14s$ on serial mode on a virtual Ubuntu machine. Depending on the number of iterations specified, `deformDyMMesh` takes between $50s$ for the default 50 iterations and decreases to $10.2s$ for the minimum (15) iterations capable of maintaining a mesh of good quality.



(a) Original mesh *(b)* Half-deformed mesh *(c)* Mesh fully deformed

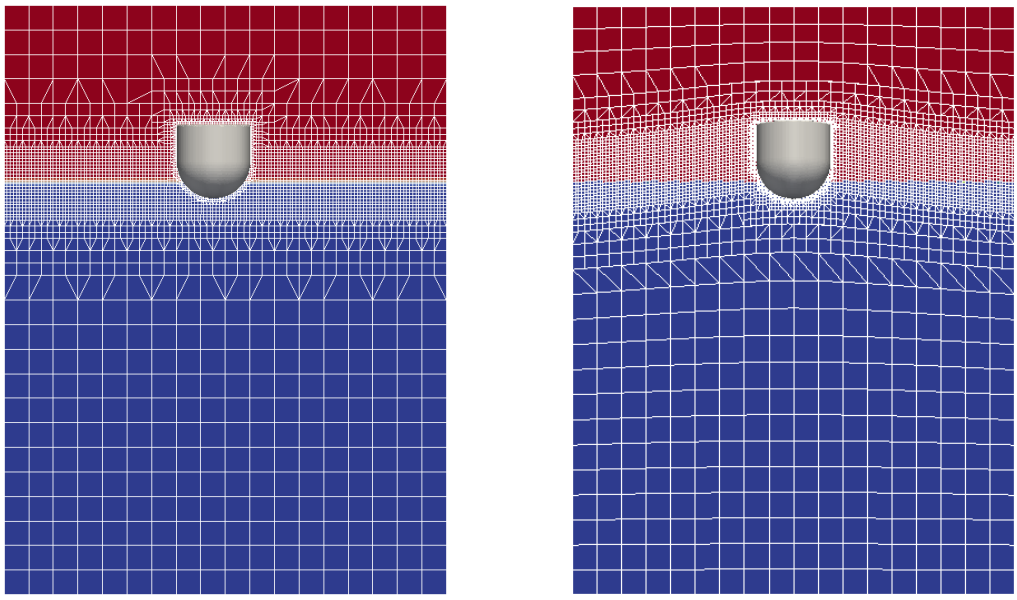
Figure 5.5: Steps of the $deformDyMMesh$ use

The heave decay test made in a $3m$ wide NWT described in section 3.8.1 is used to demonstrate the reliability of `deformDyMMesh` for Fluid Structure Interaction (FSI) applications. The comparison is made against the consistency of a case where the initial dropping position is specified by `deformDyMMesh` (i.e. `deformPosition`$= 0.204m$) and the conventional set-up. Both initial fields are set to still water.

Results and analysis

Figure 5.6 shows the differences in initial meshes between the conventional undeformed, and the deformed one obtained with `deformDyMMesh`. Figure 5.7a proves the reliability of `deformDyMMesh` in showing the heave motion time-series of the hot-started case (blue line) matching the conventional set-up (red line) and the experiment (black dashed line).

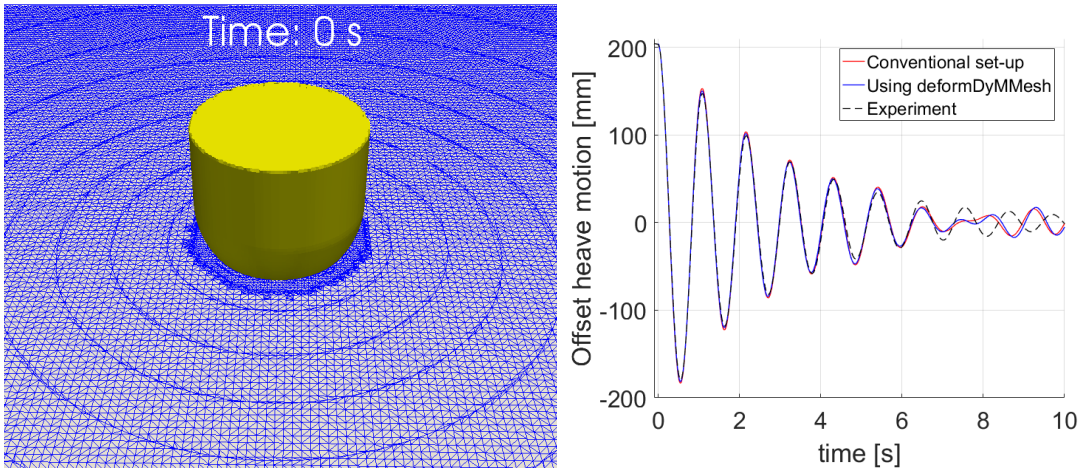
Differences between the two sets-up occurring after $6s$ are considered as negligible and specific to the heave decay test. Figure 5.7a shows that `deformDyMMesh` induces small circular waves when setting the fields (i.e. still water). These circular waves are due to the deformation that misaligns the mesh with the imposed Mean Water Line (MWL). The size of these waves is no more than a cell-size, hence $0.0208m$. In the application for WSI, the wave-field is expected to dominate over the misalignment induced by the mesh deformation.



(a) Conventional set-up

(b) `deformDyMMesh` set-up

Figure 5.6: Mesh differences at $t = 0$ between conventional undeformed, and hot-started deformed



(a) Initial set-up using `deformDyMMesh` - post-processing obtained via a cut at $\alpha.water = 0.5$

(b) Offset in heave motion of a conventional set-up simulation compared to one using `deformDyMMesh` library

Figure 5.7: Proof-of-use of `deformDyMMesh` for WSI using the heave decay test

5.2.3 Advance Motion State of the Structure

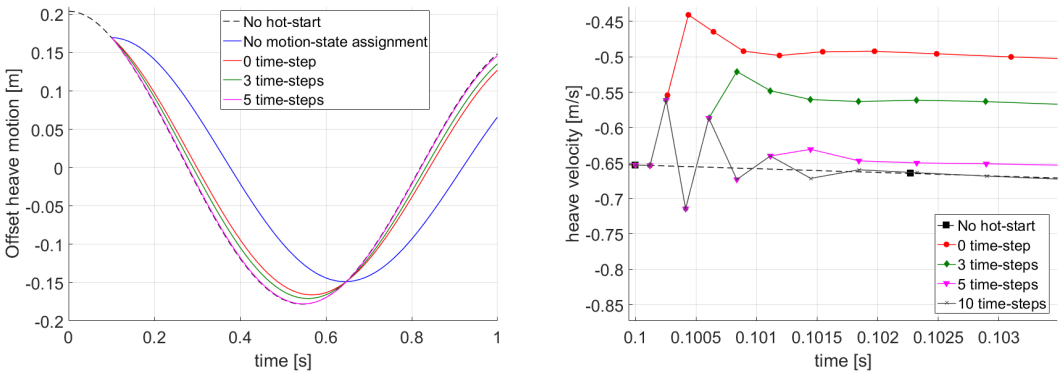
Method

In addition to the advanced position, the assignment of velocity and acceleration is required to consider the RBM dynamics at hot-start. Rigid-body velocity and acceleration can be specified at hot-start, but it is expected that the assignment requires a build-up period to converge to the correct motion. In the following, the hot-start time, t_{hot}, refers to the start of the CFD model, hence $t_{hot} - t_{minus}$ assuming a zero build-up period for the wave-field.

The input of the build-up period is investigated by assigning the RBM of the X-MED buoy for an increasing number of time-steps. The wave-field is set to still water. The heave decay test is hot-started at $t_{hot} = 0.1\,s$ using the RBM state from `deforMDyMMesh` solution, (red line) in Figure 5.7b. At the hot-started time, $t_{hot} = 0.1\,s$, the X-MED buoy velocity and acceleration are respectively $-0.65\,m/s$ and $-5.2\,m/s^2$. After t_{hot}, the velocities and acceleration from the conventional set-up solution are assigned to the buoy, with a linear interpolation between the closest time-steps of the conventional solution. When the build-up period ends, the buoy is 'released', so the motion is no longer assigned but calculated by the RBM solver. At the hot-started time, radiated waves remain small, so, the approximation of imposing still water for hot-started simulation is expected not to influence the overall motion. Investigation on the influence of build-up period on RBM assignment is restricted to the first second of the heave response, because the RBM of hot-started simulations is expected to have converged, or not, to the conventional solution.

Results and Analysis

Figure 5.8a shows the first second of the heave decay test for a conventional simulation (black dashed line), with the hot-started simulation with no RBM-state assignment (blue line), and with hot-started simulations of increasing build-up period (0 (red line), 3 (green line), 5 (magenta line) time-steps).



(a) Heave response *(b)* Imposed heave velocity

Figure 5.8: Convergence of hot-started simulation for an increasing number of time-steps

As expected, the assignment of velocity and acceleration is highly required. When set to zero, (blue line) in Figure 5.8a, the motion is equivalent to a heave decay test from a lower release height. Imposing the initial motion-state without build-up period (red line) in Figure 5.8a results in significant improvement in solution. However, it is insufficient, thus proving the requirement for a build-up period.

For the heave decay test, 5 time-steps for the RBM build-up period is found to be sufficient to impose the correct motion. Figure 5.8b highlights that imposing the motion at hot-start takes multiple time-steps to converge. Also, 5 time-steps (magenta line with upside-down triangles) is found to be slightly out. More time-steps - such as 10 ($-\times-$) - are required for a more precise convergence of the motion. However, heave response shows no improvement using more than 10 time-steps for the build-up period.

This investigation is restricted to a specific DoF, heave. The imposed velocity remained small while following a steady evolution (here increasing). In the example heave decay, a more challenging advance motion-start would be at maximum velocity (around $t = 0.25\,s$) or when the motion changes direction (crest or trough of heave response). The first is presented in Figure 5.9. The use of 5 time-steps gives the same solution as 25. Both solutions do differ slightly from the original. This is believed to be due to the radiated waves that are not correctly reproduced. This result suggests that increasing

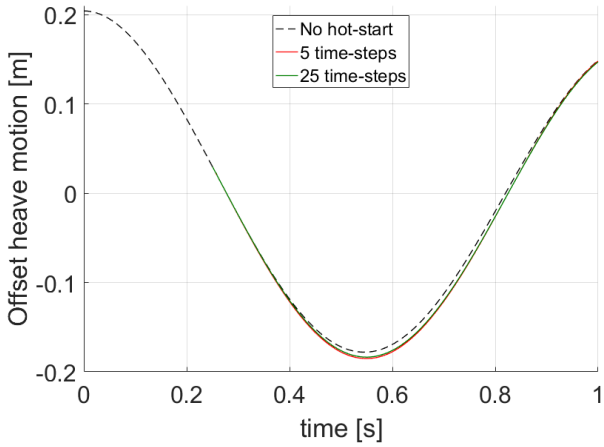


Figure 5.9: Heave response of hot-started simulation at $t = 0.25\,s$ for heave decay test

the build-up period beyond 5 time-step is not beneficial. So, 5 time-steps are used for the build-up of the RBM in the rest of the work. However, restrains remain on the general applicability of 5 time-steps for the RBM build-up period, and future development would assign the motion until the offset in motion between imposed and simulated has converged.

5.2.4 Starting a CFD-WSI from an Advance Time

The three previous investigations are applied as a proof of concept of an hot-started CFD WSI simulation. The $ST1$ event is hot-started at the focusing time, $t_{hot} = t_{focus}$, using a $4\,s$ build-up period (i.e. $t_{minus} = 4, s$). The least steep event can be used as the build-up period was found to modelled accurately any of the four events, so no difference would be expected using a steeper event. The RBM-state from motion responses (section 3.8.3) at hot-start time is imposed using `deformDyMMesh` for the position, and a 5 time-step build-up period for the RBM.

Figure 5.10 shows surge (a), heave (b), and pitch (c) responses for the conventional simulation (black dashed line), and the hot-started one (blue line). The agreement obtained for each DoF demonstrates the reliability of the hot-start technique and validates its parameters (i.e. $t_{minus} = 4\,s$, and 5 time-steps for the RBM) for focused events, with a correlation between the hot-start and the conventional solution above 0.997 for each

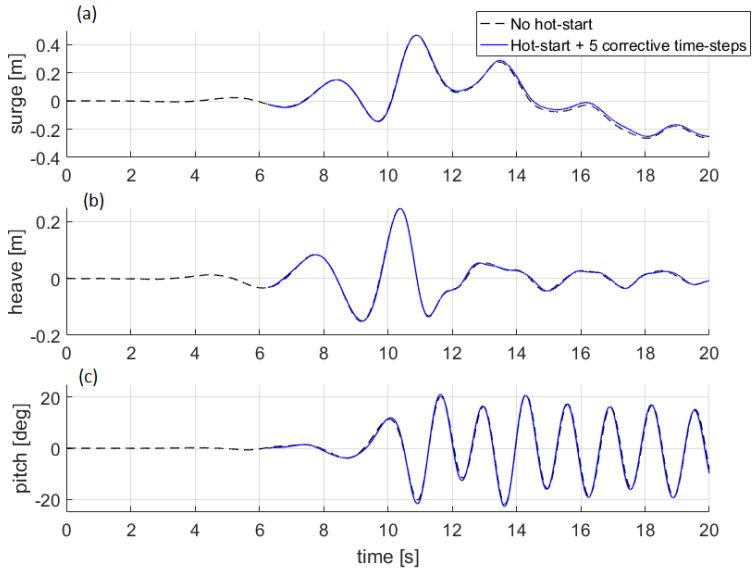


Figure 5.10: Time-series of DoF responses for the $ST1$ event hot-started at focus event and using $t_{minus} = 4\,s$ with 5 time-steps - each DoF correlation is above 0.997

DoF. No significance difference are found in the motion responses when a 10 time-step build-up for the RBM was tested.

The good agreement is probably due to the hot-start happening when rigid-body motion is very small. Diffracted and radiated waves are negligible at that time. Wave-group surface profile is mainly linear. Similar agreement is expected for the steeper cases for the same reasons. The application of the hot-start to other wave groups was discussed during the assessment of the wave-field build-up period in section 5.2.1. The hot-start approach and chosen build-up period are considered applicable to events of similar duration to $ST1$ (i.e. short wave-groups). However, the application to an irregular sea-state needs further investigation, with the definition of a build-up period function of the previous wave-train.

Further investigations where the rigid-body motion is large are necessary in order to assess the effect of neglecting radiation and diffraction. Since the non-linearity of $ST1$ is small, the wave-field build-up period can be reduced while maintaining the accuracy of the generated wave-field (Figures 5.2 and 5.3). At these times nearer to the focus,

the rigid-body motion is large and radiated/diffracted effects are of more importance. Hot-starting the simulation at that time might be used to assess the radiated/diffracted effects.

The application of the hot-start is now applied for the coupling between WaveDyn and OpenFOAM.

5.3 1-way coupling: OpenFOAM to WaveDyn

This section investigates the assignment of an OpenFOAM solution via the External Load Controller (ELC) to WaveDyn using Equation 5.3 (section 5.1.2). It is called '1-way coupling'. To avoid potential difficulties due to the hot-start, the OpenFOAM solution is imposed for the whole simulation (i.e. from $t = 0$ to the end). No trigger is developed in the ELC to swap between models. The conventional simulations from the NWT development in Chapter 3 are used. The ELC imports raw outputs from OpenFOAM simulations (loads, moments and 6-DoF motion responses). A linear interpolation is applied on outputs between the two OpenFOAM time-steps closest to WaveDyn's current time-step.

By imposing OpenFOAM loads, the 1-way coupled model is expected to replicate OpenFOAM motion responses. The moored heave decay simulation is first used to limit the assignment in the ELC to a single DoF. The least steep extreme event $ST1$ is then used to assess multiple DoFs.

5.3.1 Moored Heave Decay

The solution obtained by the $4\,m$ wide OpenFOAM NWT is imposed on the WaveDyn simulation via the ELC. The width of the NWT allows an investigation over the first $5\,s$ with no disturbances in the heave response due to radiated waves reflecting on side walls (Figure 3.32a). The additional load defined in the ELC is adapted from Equation 5.3 (F_h is detailed) and defined only on the z-axis as:

$$F_{ELC} = F_p + F_v - F_r - F_e + Kz.(z - z_0) - \rho.V_{moored}.g + \Delta(momentum) \tag{5.5}$$

where, Kz is the heave hydrostatic stiffness, z is OpenFOAM heave solution, z_0 the position of the centre of mass at equilibrium, V_{moored} is the immersed volume at equilibrium when moored $(45.291\,kg)$, and $\Delta(momentum)$ is the difference between the two models conservation of momentum ensuring consistency.

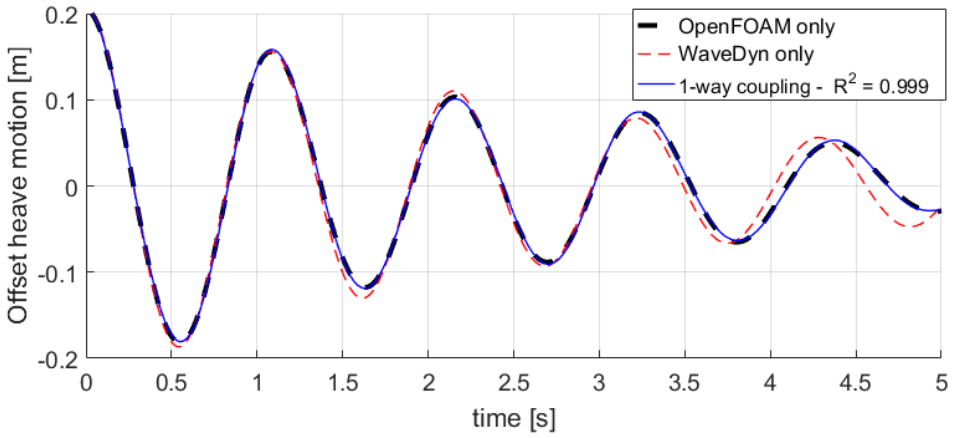


Figure 5.11: 1-way coupling applied to a single DoF motion using the moored heave decay test

Figure 5.11 shows that the 1-way coupling (blue line) changes the WaveDyn only solution (red line) to OpenFOAM one (black dashed line), hence proving the use of the ELC for single DoF.

5.3.2 WSI: ST1 case

The $1.5\,m$ wide OpenFOAM NWT solution, Figure 3.39a section 3.8.4, is imposed via the ELC on the WaveDyn model. Only the buoy three DoF - heave, surge and pitch - are considered. Surge is assigned using an expression similar to Equation 5.5. To assign the pitch, the moment around y-axis is defined as the sum of the moments resulting from each force:

$$M_{ELC} = M_p + M_v + Ky.(\theta - \theta_0) + M_s - M_r - M_e, \tag{5.6}$$

where, Ky is the pitch hydrostatic stiffness, θ and θ_0 are the pitch response and initial pitch angle in OpenFOAM, and M_s the moment due to the spring forces. Moments are transferred to the centre of gravity within the ELC.

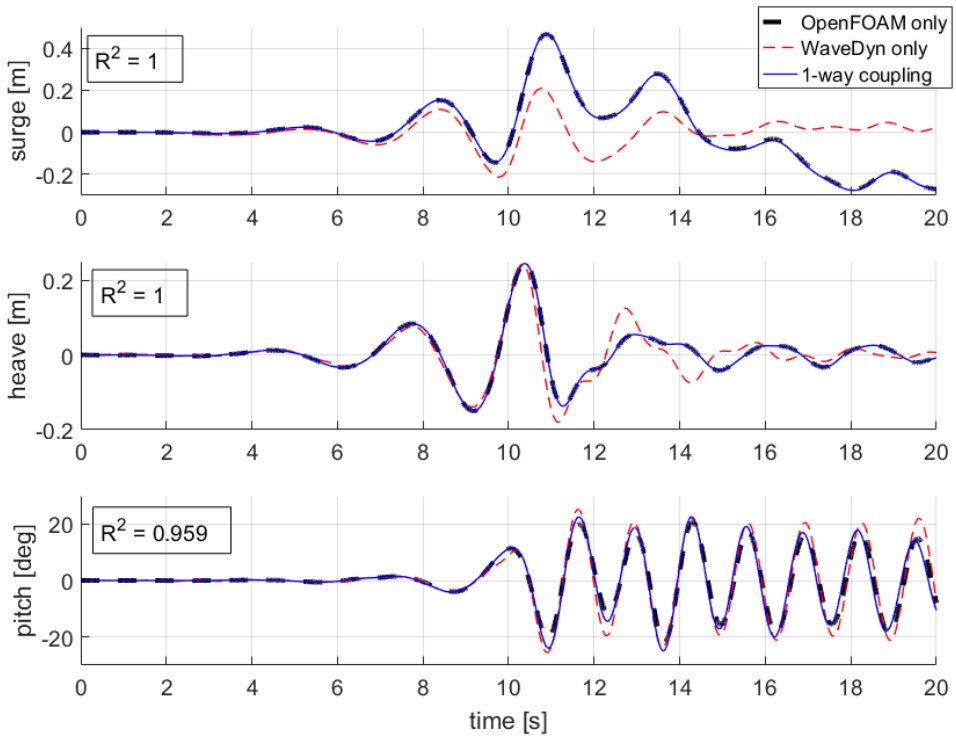


Figure 5.12: 1-way coupling applied to modelling WSI using the $ST1$ case

Figure 5.12 shows that the 1-way coupling (blue line) solution compares well with the OpenFOAM one (black dashed line), and differs from the WaveDyn only solution (red dashed line). All DoF are reproduced with a high level of accuracy, with a correlation above 0.999 for surge and heave. The minor differences in the pitch motion ($R^2 = 0.959$) are considered due to differences between models in the calculations of moments. The agreement obtained across DoFs proves the correct setting-up of the 1-way coupling.

5.4 2-ways coupling

WaveDyn is fully coupled with OpenFOAM by implementing the trigger hot-start (section 4.5.4) that changes from WaveDyn to OpenFOAM. The coupled model, therefore, follows WaveDyn only solution from initial time, $t = 0$, up to the hot-start, t_{hot} (steps **1** and **2** in Figure 5.1). Depending if the wave-field is initialised, the build-up period of the wave-field, t_{minus}, is set to 0 (no wave-field) or $4\,s$ (for WSI). At $t_{hot} - t_{minus}$ (step **3**), the OpenFOAM $1.5\,m$ wide NWT is hot-started with the WaveDyn RBM solution using `de-`

`formDyMMesh`, the 5 time-steps build-up period for the assignment of the motion, and if required the $4\,s$ build-up period for the wave-field. From $t_{hot} - t_{minus}$ (step **4**), WaveDyn is driven by OpenFOAM solution via the ELC; it is the 2-ways coupling.

The coupled model is first assessed using the moored heave decay run by the coupled model with a hot-start specified at $t_{hot} = 0.1\,s$ and $t_{minus} = 0$. A second assessment is made, for the coupled model to simulate the $ST1$ event with a hot-start at the focusing time, $t_{hot} = t_{focus}$, and the required build-up period to reproduce accurately the event, $t_{minus} = 4\,s$.

Finally, to present a proof-of-concept of the coupled model in application to extreme WSI, the hot-start is defined by the limit of WaveDyn model in IkA as defined in section 4.5.4; and, the coupled model is applied to the long irregular sea-state (used in section 4.6.2). As previously mentioned, the coupled simulation ends after the first use of CFD as the development for continuing a WaveDyn-only simulation is left for future work.

5.4.1 Moored Heave Decay

The hot-start time, $t_{hot} = 0.1\,s$, is chosen because the radiated wave-field is negligible at this point limiting the error in initializing the NWT as still water at hot-start. Besides, WaveDyn and OpenFOAM motion response are identical at this time (Figure 4.4a section 4.2.2). Therefore, the fully coupled solution should converge to OpenFOAM-only one.

Figure 5.13 illustrates the change from the WaveDyn only simulation (red dashed line), to the fully coupled-model (blue line), which matches the OpenFOAM only solution (black dashed line). It demonstrates the implementation of the trigger in the ELC that, once activated, sets-up the NWT and initialises the simulation at hot-start. Besides, the $0.1\,s$ of WaveDyn simulation saves 16% of the computational effort of the OpenFOAM-only simulation.

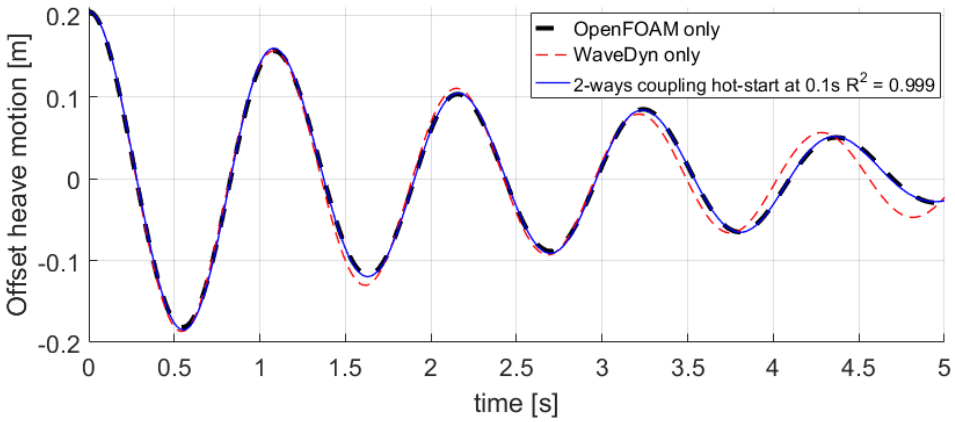


Figure 5.13: Heave decay test of the coupled model using a trigger at hot-start time $t_{hot} = 0.1\,s$ ($t_{minus} = 0$)

5.4.2 WSI: ST1 case

The coupled model is tested against the $ST1$ case hot-starting at focusing time, $t_{hot} = t_{focus}$, with a build-up period, $t_{minus} = 4\,s$. OpenFOAM hot-starts with the RBM-state evaluated by the WaveDyn-only solver at $t_{hot} - t_{minus}$. The fidelity of the OpenFOAM solution being of higher fidelity than WaveDyn, the coupled model imposes OpenFOAM solution from that time (i.e. $t_{hot} - t_{minus}$) into WaveDyn.

Further investigation is required by the present development. The steeper case is one of them. However since no difference in the wave-group (conclusions from the $4\,s$ wave-field build-up period in section 5.2.1) was found for the steepest event, and since the rigid-body motion remains small at $t_{hot} - t_{minus}$ which assures minor radiation/diffraction effects, conclusions from the least steep group might be applicable for the steepest.

In Figure 5.14, the coupled-model (blue line) differs from the WaveDyn only model (red dashed line) after $t_{hot} - t_{minus}$, from where it follows the OpenFOAM only solution (black dashed line). As evaluated for the 1-way coupling, section 5.3.2, a small error in pitch exists ($R^2 = 0.96$), while surge and heave are captured to a higher accuracy (respectively $R^2 = 0.989$ and $R^2 = 0.998$). Therefore, the implementation of the hot-start for the 2-ways coupling is validated.

Besides, the 2-ways coupling model is validated against experimental data (green

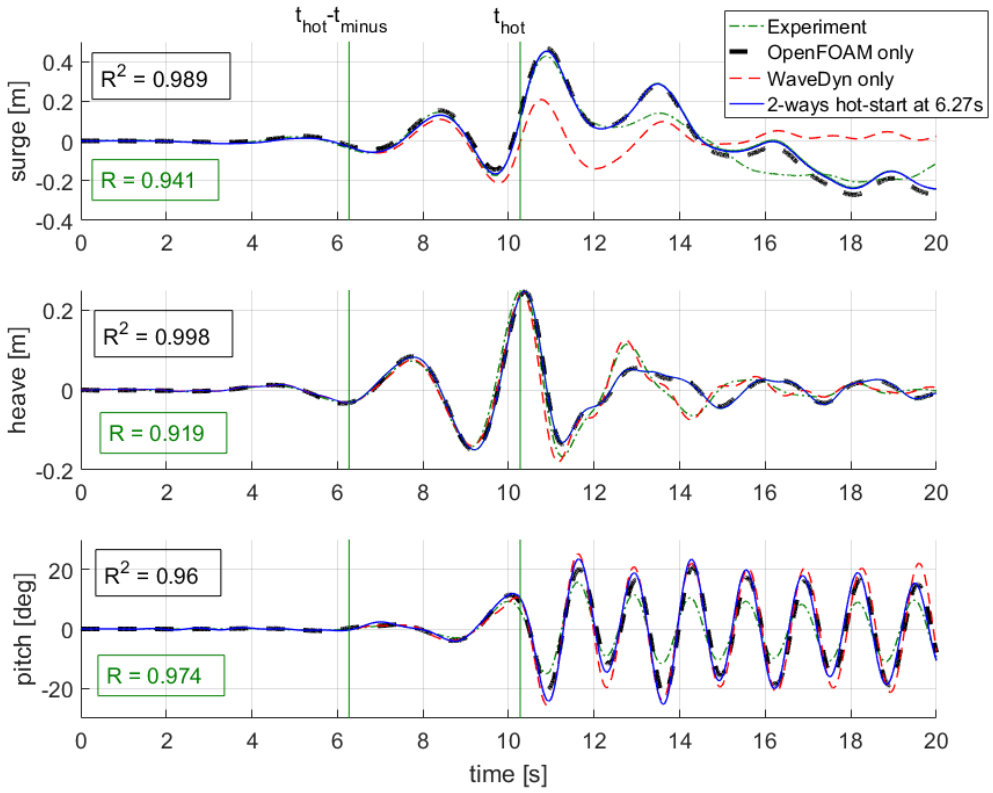


Figure 5.14: WSI applied to $ST1$ case comparison with the hot-start at $10.27\,s$ with $t_{minus} = 4\,s$)

dots). The accuracy of the coupled model in each DoF, measured by the correlation (i.e. R) between time-series, is similar to the CFD only (values available in Appendices D.2): surge is 0.941 against 0.956; heave is 0.919 against 0.92; and pitch is 0.974 against 0.95. Therefore, the coupled model is validated against experiment for the $ST1$ event, where it demonstrates the modelling of extreme WSI to a similar level of accuracy than CFD using the X-MED buoy. This result is considered applicable to the three steeper events, $ST2$, $ST3$ and $ST4$, and to events of similar duration as previously discussed in section 5.2.1.

The benefit in using the coupled model results in a 26% reduction in computational effort. Therefore, the coupled-model has proven its capacity to model extreme WSI at the level of accuracy of CFD for a lower computational effort, hence demonstrating the concept.

It should be noted that the WaveDyn validity limit identified by IkA threshold is at the trough before the main crest (Figure 4.23a, section 4.6.1). If the hot-start had been specified at that time, the coupled model would have swapped to OpenFOAM at an earlier time than done in this case. So, a similar replication of the OpenFOAM solution and experiment would be expected as it is equivalent to a hot-start at the main crest with a larger build-up period, t_{minus}, hence assuring that the wave-field would be correctly modelled. Therefore, Figure 5.14 demonstrates the full use of the coupled model to model extreme WSI in which the trigger is defined from the WaveDyn validity threshold identified in section 4.5.4.

5.4.3 Proof-of-concept: Extreme Irregular Sea-State

The WaveDyn limit in IkA identified in section 4.5.4 is implemented as a trigger in the ELC. The 2-way coupling model is assessed against the modelling of WSI of the X-MED buoy under the long irregular sea-state previously used in section 4.6.2. WaveDyn validity limit is identified for the long irregular sea-state at $t_{hot} = 32.86\,s$ (section 4.6.2), hence $t_{hot} - t_{minus} = 28.86\,s$. The stop in the use of CFD so that WaveDyn continues without the ELC (step 5 in 5.1) is not investigated in the present study. So, the simulation stops at $40\,s$ for a $120\,s$ irregular sea-state, after the first use of the OpenFOAM-CFD model.

Figure 5.15 compares the motion of the X-MED buoy under an extreme irregular sea-state between the WaveDyn-only (red dashed line) model, the coupled model (blue line), and the experiment (black dashed line). The time where WaveDyn validity threshold is attained, t_{hot}, and the time where CFD is used, $t_{hot} - t_{minus}$, are highlighted as (green vertical line).

The coupled model performs a simulation not affordable in CFD only, and where the mid-fidelity model WaveDyn becomes outside its validity. Figure 5.15 shows that the coupled model benefits from the OpenFOAM solution for the capture of the surge motion. From $t = 35\,s$ to $t = 40\,s$, the displacement in the negative x-direction is captured by the coupled model, whereas not by WaveDyn only. The couple model captures the low

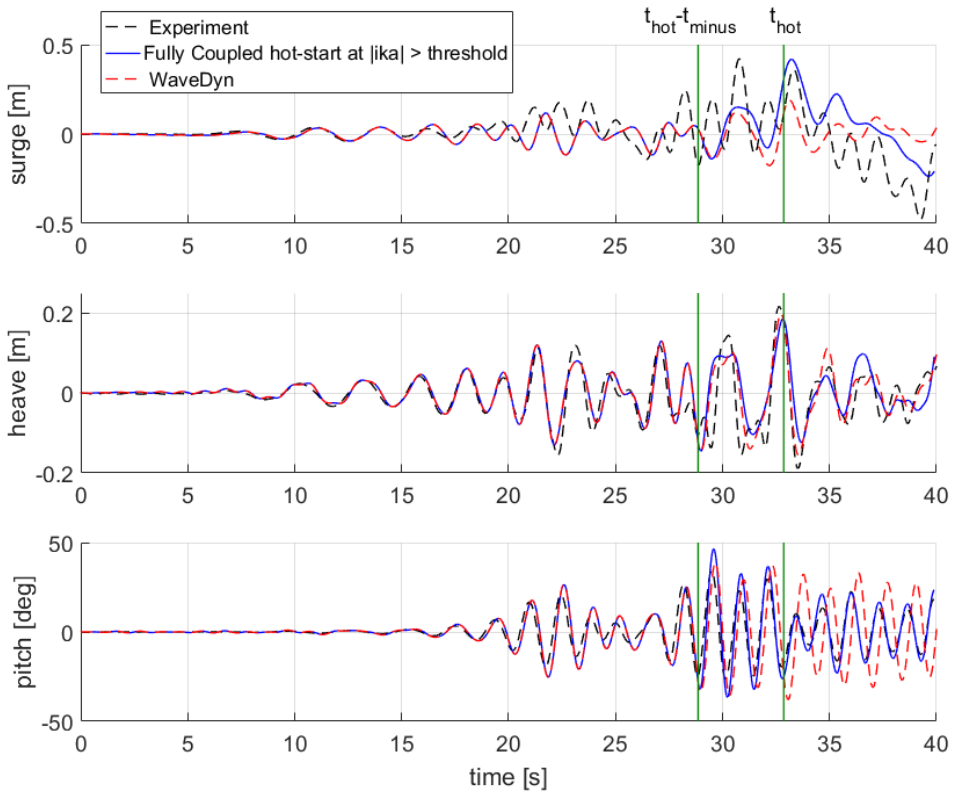


Figure 5.15: Proof-of-concept of the use of the WaveDyn-OpenFOAM time-coupled model for the assessment of the 6-DoF motion responses of the X-MED buoy in an irregular sea-state

frequency surge motion actually identified by WaveDyn fidelity criteria (as discussed at the end of Chapter 4). Therefore, the present demonstrates the use of WaveDyn fidelity criteria to identify low frequency surge response not captured by WaveDyn, and implement it in the coupled model.

Therefore, it appears that, despite the unknown on the use of the $4s$ build-up period t_{minus} for the wave-field, the overall accuracy of the solution increases with the use of the coupled model. This demonstrates the interest in the coupling between WaveDyn and OpenFOAM to model extreme irregular WSI.

The evolution of the accuracy in the capture of given DoF by the coupled model is expected from the previous investigation and discussion of the accuracy of WaveDyn and OpenFOAM taken separately (section 4.3.2). The decrease in accuracy in heave

is due to the use in WaveDyn of the surface elevation at the model location, which does not require the propagation of the wave-group compared than in OpenFOAM. The increase in accuracy in surge and pitch relates to the increase in the level of physics (i.e. viscosity) included by the coupled model via the use of CFD. However, the increase in accuracy pitch is significant and unexpected compared to the previous investigation. As pitch appears to deviates at hot-start, it suggests that a second trigger might be identified from the deviation in pitch response using the methodology developed by the present study for surge responses.

However, the coupled model does not successfully capture all the recorded motion response of the buoy. The surge motion found by OpenFOAM does not capture the peak just after the hot-start, or the oscillations from $t = 35\,s$ to $t = 40\,s$. The first error is most certainly related to the wrong motion-state at $t = 28.86\,s$ due to the error in the WaveDyn only solver, but the second one might be an inaccuracy in the capture of the CFD model.

The next step in the development of the coupling would be the investigation over the stop of CFD so that WaveDyn continues until a possible next activation of the trigger (step 5 in Figure 5.1). As for the build-up period of the wave-field, it is expected that a minimum period of use of CFD would be required before 'releasing' WaveDyn model. During the 2-ways coupling, OpenFOAM corrects the acceleration of the rigid-body via Newton's second law of motion. After the release, WaveDyn-only needs to maintain the capture of the surge motion. A solution would be to investigate the release of WaveDyn using the least steep event by releasing WaveDyn at increasing times after the hot-start. From this investigation a require period could be defined and then tested for the steeper cases and the present irregular sea-state. However, the applicability of this period might be suggestive to the resulting large surge motions WaveDyn does not capture. A second possibility would compare motion responses between a WaveDyn-only simulation and the coupled model. When the difference tends towards zero, the model could swap back to WaveDyn-only. However, this strategy is expected to also

swap at times when the two solutions cross rather than converge. A third possible approach would consist in comparing WaveDyn-only sum of forces with OpenFOAM ones. When the difference tends towards zero, WaveDyn and OpenFOAM will calculate the same RBM, and CFD would be no longer required.

5.5 Conclusion to Chapter 5

In this chapter, the time-domain potential flow based model WaveDyn is coupled with a Computational Fluid Dynamics (CFD) based Numerical Wave Tank (NWT) validated for extreme Wave Structure Interaction (WSI) developed in OpenFOAM in Chapter 3: At times where the wave-field is identified as outside WaveDyn's validity from Chapter 4, OpenFOAM hot-starts (i.e. starts from that time) from WaveDyn's solution in order to increase the level of physics and fidelity.

Starting the wave-field from an advance time in the NWT requires the linear imposed description to build-up to the equivalent wave-field starting from still water at the initial time, $t = 0$. A $4s$ build-up period relative to the focusing time is found necessary to replicate accurately any of the four NewWave based design-waves, and is considered acceptable for other design-waves of shorter wave-period, hence $Tp \leq 2.8\,s$.

To assign the Rigid Body Motion (RBM)-state (i.e. position, velocity, acceleration) in the NWT, a new library - specifically developed - uses the offset from equilibrium to deform the mesh which rigid-body is inserted at equilibrium. The library avoids the generation of a new geometry file, maintains mesh quality at equilibrium, and has been demonstrated to be consistent with initially undeformed mesh for a heave decay test. Simulations of large initial displacement are suspected to benefit from the library because the mesh quality increases as the rigid-body returns to equilibrium. The RBM-state is found to require a build-up period of 5 time-steps, where the RBM-state is assigned, to converge to a given motion.

The coupling imposes OpenFOAM solution of the conservation of momentum via an additional load applied to the rigid-body in WaveDyn model. Using a heave decay

test, and the $ST1$ case, the 1-way coupling first demonstrates that the heave motion response is accurately reproduced by imposing OpenFOAM solution via the additional load into WaveDyn. In second is demonstrated the 2-ways coupling that is governed by WaveDyn solution until a trigger time. From that time, or minus the $4s$ build-up period in cases involving a wave-field, WaveDyn RBM-state is imposed to OpenFOAM via the 5 time-steps build-up period and the mesh-deformation library. Using a manual trigger at $0.1s$, the coupled model replicates a heave decay test to the level of accuracy of CFD, while saving 16% of the computational effort. The coupled model reproduces to the level of CFD, and validate to experiment, the motion responses of the X-MED buoy under an extreme event ($ST1$) using a trigger time specified at the focus time. This trigger time being more restrictive than a trigger selected by WaveDyn validity threshold, it proves the use of the coupled model for extreme WSI for a lower computational effort, 36%, than CFD, while maintaining the accuracy.

The coupled model concept is demonstrated against a long extreme irregular sea-state where the use of OpenFOAM starts at WaveDyn's validity limit defined from the wave-field. The gain in accuracy in multiple Degree of Freedom (DoF)s obtained by the coupled model demonstrates the applicability of the coupling to simulations not affordable in CFD only, and out of linear potential flow based solver validity.

Chapter 6

Conclusions

Marine Renewable Energy (MRE) energy devices can play a key role in the development of an electric supply of low-carbon emission in the context of the fight against global warming. However, the numerous failures among MRE projects, especially among Wave Energy Converter (WEC), has proven the requirement in the assessment of survivability for validated numerical models of high fidelity within a computational effort affordable for the industry.

The originality of the present study is the development of a coupled model. This study demonstrates the validity of a new numerical model coupling WaveDyn (mid-fidelity model) to OpenFOAM (high-fidelity Computational Fluid Dynamics (CFD) model) in the modelling of extreme Wave Structure Interaction (WSI) for a decrease in computational effort compare to CFD, while improving the mid-fidelity solution to the level of CFD. This overall achievement corresponds to the objectives detailed at the end of Chapter 1 and was made possible by several original contributions:

- The development and validation against experiment of a CFD based Numerical Wave Tank (NWT) modelling the extreme WSI of a single moored buoy representative to point-absorber type WEC (X-MED buoy) using four NewWave based extreme events of increasing steepness; a NWT which is also validated against twelve NewWave shifted events.
- The assessment of WaveDyn and OpenFOAM fidelity to model extreme WSI using the motion responses of the X-MED buoy under the four focus events of increasing steepness, plus the twelve shifted events.

- The evaluation of the WaveDyn validity threshold as a function of a time-dependent non-linear wave-parameter, the Instantaneous steepness (IkA), that identifies times where WaveDyn surge motion response solution deviates.
- The assessment of requirements in the start of a CFD simulation for WSI modelling via the definition of a build-up period for the wave-field, one for the Rigid Body Motion (RBM)-state, and the development of a new library for the assignment of rigid-body position.
- The implementation of the time-coupling strategy and its validation for the modelling of extreme WSI at a lower computational effort than CFD in addition to an increase in accuracy compared to potential flow time-domain solver.

This chapter summaries the work per chapter in order to highlight most important original contributions. A full summary of a given chapter is available at its end. In a final section is suggested possible directions for future work.

6.1 Numerical Wave Tank for Extreme Wave Structure Interaction

In Chapter 3 is developed a NWT replicating the experiments performed in the Physical Wave Tank (PWT) of the University of Plymouth of the extreme WSI between the X-MED buoy and four NewWave focus events of increasing steepness (Hann et al. 2018).

This NWT has been developed using a blind-test approach, where only the Wave Gauge (WG) the most upstream (i.e. the first hit by the wave-group) is available. Other experimental measure, such as other WG and the rigid-body motion response, are only used for the comparison. Therefore, only a numerical assessment is used in order to select the NWT specifications. The interest of this blind-test development is to provide a NWT capable of representing other extreme events performed in the same PWT.

In NWT, wave-generation and absorption remain intrinsic problems. The present study identifies a NWT of greater accuracy unaffected by wave reflections since its length is longer than the time required for the wave-group to propagate and affect the solution. This long NWT tackles the wave absorption problem. Wave-generation of focus

event requires to impose the linear superposition of harmonics of the first WG. A pre-processing tool is developed to limit the number of harmonics necessary at the inlet - since harmonics have been identified to increase exponentially the memory requirement (Brown et al. 2018) -; while not affecting the solution.

In 3-dimensions $(3D)$, abnormally large air flows velocities induced by the solver are affecting significantly the solutions. Using an approach adapted from by Liu and García (2008), the flow is limited to maintain consistency with 2-dimensional $(2D)$ solutions of the surface elevation. For the four focus events, convergence between $2D$ and $3D$ solutions is obtained when limiting the overall flow to the maxima of water particles velocities evaluated from linear theory. However, a $3 m/s$ limit is selected to account for fast flows that might happen near rigid-body. In addition to an increase in the accuracy of the surface elevation, the reduction of computational effort is significant as flow velocity drives the length of the time-step via the CFD condition.

The $20 \times 1.5 \times 4 m$ NWT is validated against the experiments with correlations above 0.9 for any Degree of Freedom (DoF) in any of the four focus events. The level of agreement demonstrates the use of the NWT to model extreme WSI of the X-MED buoy under NewWave based events independently from their non-linearity, and up to just below breaking limit ($kA = 0.263$). As expected, in Chapter 4, the NWT is proven reliable (correlation above 0.9) in the modelling of extreme WSI of the X-MED buoy for short-crest events of different shapes using the twelve shifted events of the NewWave least steep event. Therefore, the NWT is considered applicable to a wider range of extreme sea-states.

6.2 Assessment of WaveDyn performance and limits on modelling extreme events

Following the blind test development for the CFD base NWT, a linear potential flow time-domain model is developed based on the code WaveDyn. In order to later coupled both code, i.e. OpenFOAM and WaveDyn, this chapter assesses the performance of WaveDyn relative to experiment and OpenFOAM. The evaluation of each model validity

is made function of the event steepness since a parametric range of use of numerical models was identified as beneficial for the WEC sector (Ransley et al. 2019). The event steepness is chosen for its relation to non-linearity.

WaveDyn demonstrates the known capacity of linear potential flow model to capture accurately heave motion since this one is a linear response to the surface-elevation below the breaking limit. However, large surge motions of low frequency involving a large drift are not captured. This error is concerning in extreme response modelling since WECs typically show large motion response with influence on time-history.

This error is used to identify WaveDyn validity limit from which the coupling would start. The identification of the points where WaveDyn deviates from experimental solution is conducted by introducing a time-dependent wave-parameter, IkA. The definition of this criteria is one of the main original contributions of the present study. IkA is the vertical velocity of the free-surface, which is proportional to the horizontal acceleration in potential flow theory. Since WaveDyn model neglects viscous effects, the motion responses obtained by WaveDyn are dominated by inertia forces, hence related to the IkA. Using the four focus events plus twelve phase-shifted version of the least steep, a maximum of the absolute value of IkA is defined. Above this threshold, WaveDyn is expected to deviate. This criteria allows to track down WaveDyn fidelity with time.

Applied to the different events plus an extreme irregular sea-state, the criteria is found to identify points preceding large surge motion of low frequency.

6.3 Coupling of WaveDyn with OpenFOAM

In Chapter 5 is developed the coupling of the CFD-NWT validated in Chapter 3 with WaveDyn model. The time-coupling strategy restricts the use of CFD to the modelling of WSI requiring this level of physics, which times are identified from WaveDyn limit in validity evaluated in Chapter 4. The coupling models extreme WSI at a reduce computational effort compare to CFD while increasing the fidelity of WaveDyn to the level of CFD. This tool aims to make affordable for the industry the simulation of long extreme

irregular sea-state recommended for survivability assessment of WEC (International Electrotechnical Commission 2016).

Several developments were made to allow the CFD simulation to start from an advance time, where both rigid-body and wave-field are non-zero:

- The identification of a build-up period for the linear imposed solution across the NWT to converge to a conventional CFD start;
- An initial mesh deformation library that deforms the mesh with rigid-body at equilibrium from the position offset in all DoF;
- and, the identification of a second build-up period for the assignment of a RBM-state using a linear interpolation between solution and assignment.

The build-up period for the wave-field is considered applicable for other design-waves of equal or shorter wave-period, $Tp \leq 2.8s$. Reserves are taken on the extension to irregular sea-states, where further investigations are required. The mesh-deformation pre-processing tool avoids the use of Computer Aided Design (CAD) to generate a new geometry file, maintains the mesh quality at rigid-body equilibrium, and is consistent in the heave motion response of a decay test with initially undeformed mesh. CFD simulations of large initial displacement (e.g. splashing plate on still water) constitutes potential external use for the library because the mesh quality increases as rigid-body returns to equilibrium.

These different developments, and the ones done in previous chapter, allow to couple WaveDyn with OpenFOAM. The coupled model demonstrates its used for the least steep event by obtaining the level of accuracy of CFD in all DoF while reducing the computational cost by 26%.

Finally, a proof of the coupling concept is demonstrated against a long irregular sea-state, where the use of CFD only is unaffordable and WaveDyn becomes outside its validity. The trigger is activated from the IkA limit. The coupled model captures surge

and pitch responses that are not captured by WaveDyn only, hence increasing the accuracy of the coupled solution and demonstrating the use of the coupled tool for long irregular sea-states. Heave motion remains captured to a higher level of accuracy by WaveDyn only as this one benefit from the measure of the surface elevation at model location.

6.4 Future Work

Potential improvement exists in the optimisation of the NWT. Using lower criterion the accuracy of the surface elevation or the relaxation-zone, the computational effort of $3D$ WSI simulations could be decreased with minor effect on the accuracy of motion responses. This investigation could provide a parametric assessment of NWT optimisation as a function of computational effort and accuracy.

Another domain for potential investigations concerns the simulation of multi-bodies in the CFD model, where the RBM solver - `rigidBodyDynamics` - is expected to demonstrate its interest compare to other RBM solvers. The first interest would be the implementation of additional parts of the WEC like the Power Take Off (PTO) as required by developers (CCP-WSI Working Group 2016). This investigation will require the validation against different model than the X-MED buoy. The experiment made by Rafiee et al. (2016) on the CETO WEC in the University of Plymouth physical wave tank could be used to investigate the effect of including more complex description of the PTO in the modelling of extreme events. A second interest in the development of multi-bodies in CFD concerns the assessment of WEC in an array, where experimental data made at the same physical wave-tank are available for validation. Such simulation being significantly computationally expensive, the use of the coupling strategy can play a key role in the reduction of this expense.

It would interesting to extend the assessment of models validity. CFD is considered capable of simulation the breaking of wave, but the simulation of such events is more challenging and might require higher fidelity model like Smoothed Particle Hydrodynamics (SPH). WaveDyn validity limit could be extended to extreme events of lower

intensity using the same NewWave event but for lower steepness.

As reserves were taken on the applicability of the coupled model for irregular sea-states due to the build-up period of the wave-field, a potential investigation would be to assess the applicability of the $4s$ build-up period, or its re-evaluation, for arbitrary hot-starting times in multiples irregular sea-states. Besides, an investigation on the build-up period for design-waves of longer wave-periods is suspected to show the limit of applicability for the $4s$ build-up period. These investigations would provide material to widen the range of application of the coupled model, in which a parametric assessment function of the event period could be considered.

DNV-GL is planning on using the coupling between OpenFOAM and WaveDyn as an optional complementary toolbox included within WaveDyn software. Further investigations would be relevant to ensure the validity of the coupling strategy. The coupling will need to be validated to other point-absorber WECs, where WaveDyn validity threshold is the most likely to be applicable. A non-dimensional investigation would be interesting using simple different geometries for the floater to investigate the effects of viscosity on validity range. It would also be interesting to extend the investment of numerical models validity to other types of WEC, such as an Oscillating Wave Surge Converter (OWSC). It is expected that the validity threshold would not remain applicable as the motion responses are significantly different between WEC types. However, further investigations could lead to a more generic limit of potential-flow time-domain models. Besides, DNV-GL also wants to investigate the applicability of the coupling strategy to 'Tidal-Bladed', the tidal optimization tool developed by DNV-GL equivalent to WaveDyn for tidal WEC.

Appendix A

Additional Material

A.1 PIMPLE Algorithm

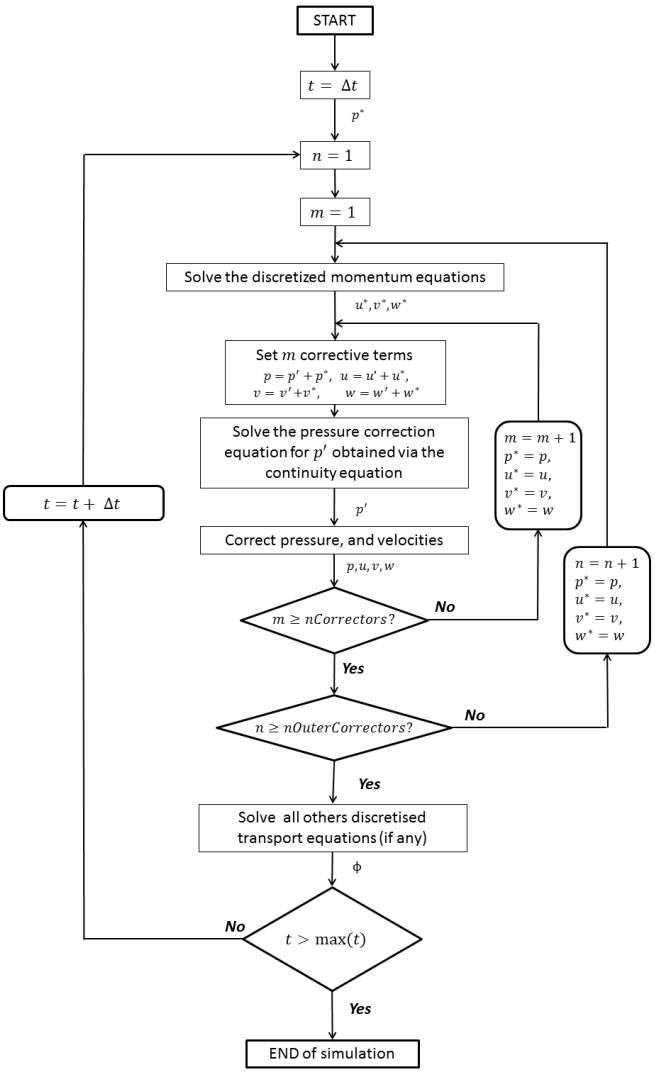


Figure A.1: PIMPLE algorithm schematic representation

A.2 Additional proofs

A.2.1 Consistency of the surface-elevation from pre-processing tool selecting the harmonics

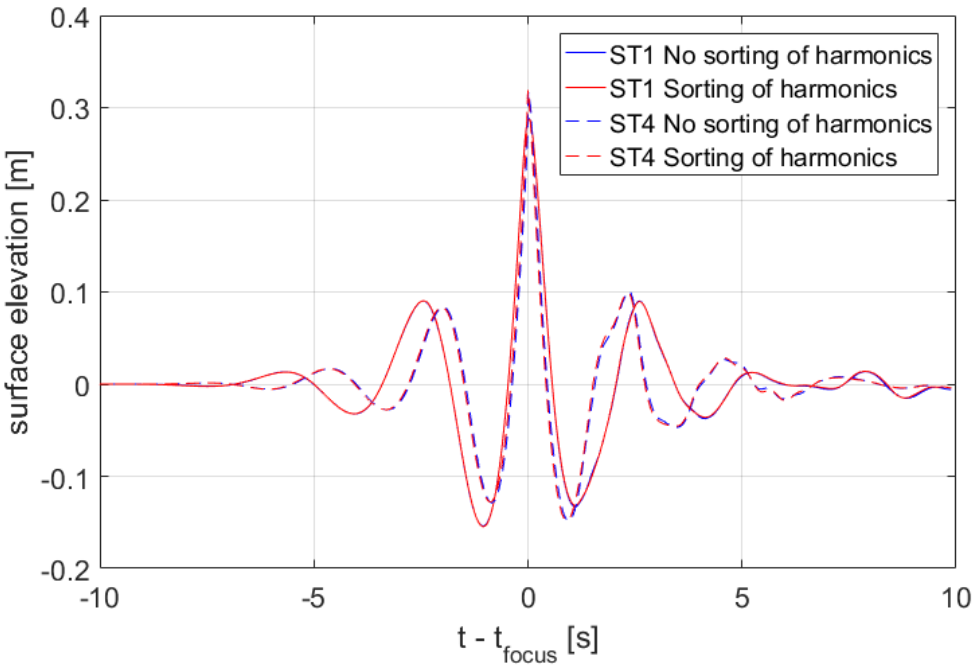


Figure A.2: Surface elevation obtained at focus location for least and steepest events using the sorting of harmonics at inlet or not

A.2.2 Consistency of surface-elevation when using a refinement around the Mean Water Line (MWL) to level 3

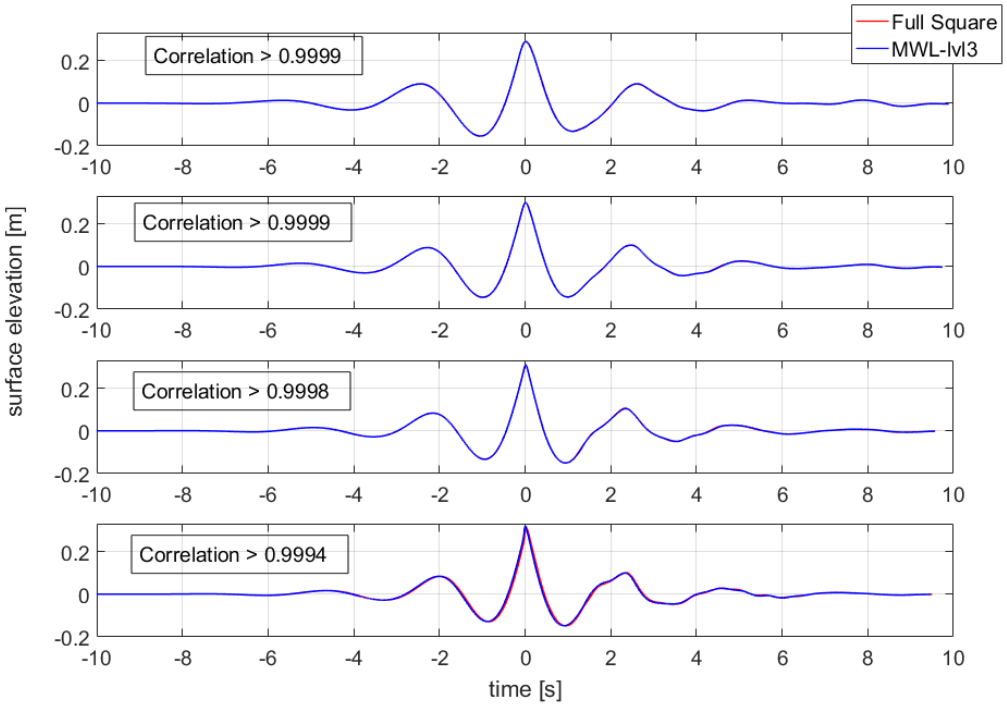


Figure A.3: Comparison of the surface elevation at focus position for the four events, between a full square mesh and one using a refinement around the MWL up to level 3

A.2.3 Consistency of heave decay when using ULim

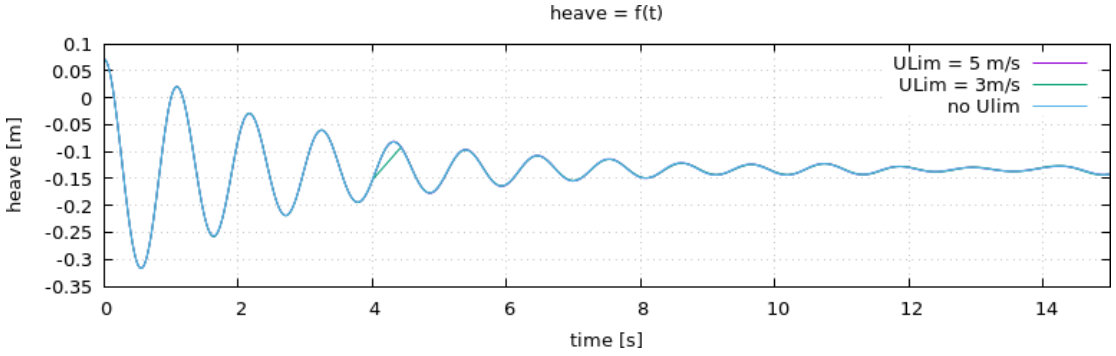


Figure A.4: Heave decay test comparing simulating using different values of $ULim$ with one simulation without $ULim$

Appendix B

Physical Experiments

B.1 X-MED buoy

B.1.1 Mechanical specifications

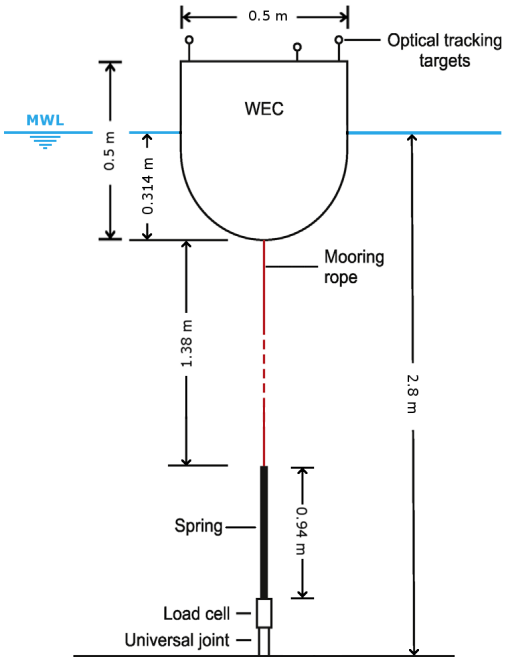


Figure B.1: Extreme loading of marine energy devices due to waves, current, flotsam and mammal impacts (X-MED) moored

Dry mass	$43.2 kg$
Inertia	$Ixx = 1.61 kg/m^2$ $Iyy = 1.61 kg/m^2$ $Izz = 1.25 kg/m^2$
Rope Spring	$35 N/mm$
Linear Spring	$0.064 N/mm$
Free spring length	$0.63 m$
Mooring pre-load	$-20.5 N$
Unmoored G	$-0.122 m$
Moored G	$-0.133 m$

Table B.1: X-MED Model main characteristics

Decay test	No-mooring	With mooring line
Heave	0.926	0.917
Surge	not measured	0.07
Pitch	not measured	0.75

Table B.2: Measured resonance frequencies in Hz from decay tests data

B.1.2 Resonance frequencies

B.2 ST-1/2/3/4

B.2.1 Wave-Gauges

WG#	1	2	3	4	5	6	7	8
Position [m]	13.27	14.32	14.77	15.23	15.7	16.11	16.54	16.82
WG#	9	10	11	12	**13**	14	15	16
Position [m]	17.14	17.51	17.84	18.21	**18.54**	18.86	19.23	19.60

Table B.3: Waves Gauges positions in the PWT - distance from wave-paddles - WG#13 is the focus location which is aligned with the X-MED buoy front edge - WG#14 is approximately aligned with the X-MED buoy centre

B.2.2 Steepness

Case	Peak frequency $[Hz]$	Theoretical kA	Measured kA	Measured A $[m]$
$ST1$	0.356	0.15	0.152	0.273
$ST2$	0.388	0.17	0.179	0.280
$ST3$	0.420	0.2	0.209	0.286
$ST4$	0.449	0.22	0.263	0.318

Table B.4: Theoretical and measured characteristic for the four focus wave events

B.3 Shifted

B.3.1 Wave-Gauges

WG#	1	2	3	4	5	6
Position [m]	11.206	12.929	13.328	13.641	15.928	16.53
Sides WG#	7	**8**	9	10	11	12
Position [m]	16.81	**17.8**	17.8	18.05	18.05	corrupt

Table B.5: Waves Gauges positions in the PWT - Distance from wave-paddles - WG#8 is the focus location which is aligned with the X-MED buoy front edge; WG#10 is aligned with the X-MED buoy centre

Appendix C

Numerical Wave Tank and CFD decisions

C.1 Physical properties

	Physical symbol	OpenFOAM name	Value	Units
Phase fraction Water	$\alpha.water$	`alpha.water`	1	–
Water viscosity	ν	`nu`	10^{-6}	$m^2.s{-}1$
Water density	ρ	`rho`	1000	$kg.m^{-3}$
Phase fraction air	$\alpha.air$	`alpha.air`	0	–
Air viscosity	ν	`nu`	$1.48x10^{-5}$	$m^2.s{-}1$
Air density	ρ	`rho`	1	$kg.m^{-3}$
Surface tension	σ	`sigma`	0	$kg.s^{-2}$
Gravity	g	9.81	`g`	$m.s^{-2}$

Table C.1: Physical properties

C.2 Solvers decisions

Use	Schemes	Type
Face interpolation	`interpolationSchemes default`	linear
Gradient terms ∇	`gradSchemes default`	Gauss linear
Divergence terms $\nabla.$	`div(rhoPhi,U)`	Gauss Van Leer V
Divergence terms $\nabla.$	`div(phi,alpha)`	Gauss linear
Divergence terms $\nabla.$	`div(phirb,alpha)`	Gauss linear
Divergence terms $\nabla.$	`div(((rho*nuEff)*dev2(T(grad(U)))),`	Gauss linear
Laplacian terms ∇^2	`laplacianSchemes default`	Gauss linear Corrected
Component of gradient normal to cell face	`snGradSchemes default`	corrected
Temporal discretisation	`Euler default`	Euler
Inner corrector	`nCorrectors`	2
Outer corrector	`nOuterCorrectors`	1

Table C.2: Numerical schemes (`fvSchemes`)

Type	OpenFOAM names	Value
Inner loops	`nCorrectors`	2
Outer loop	`nOuterCorrectors`	1
Non-Orthogonal loop	`nNonOrthogonalCorrectors`	0
Correction of α	`nAlphaCorr`	1
Inner correction of α	`nAlphaSubCycles`	3
Correction of α	`cAlpha`	1
Correction of ϕ	`correctPhi`	*yes*
Momentum predictor	`momentumPredictor`	*no*
MULES iteration limiter	`nLimiterIter`	3

Table C.3: PIMPLE algorithm

Solver	Properties	Tolerance	relative tolerance
`smoothSolver`	alpha	10^{-8}	0
	U	10^{-6}	0
GAMG	`cellDisplacement`	10^{-5}	0
	`p_rgh`	10^{-8}	0
PCG	`pcorr.*`	10^{-5}	0
	`p_rghFinal`	10^{-8}	0

Table C.4: Solvers tolerances

C.3 2D-NWT

C.3.1 Set-up

	Inlet	Walls (left, right, seabed, outlet)	Atmosphere
Volume fraction - α	`waveAlpha`	`zeroGradient`	`inletOutlet`
Velocity - `U`	`waveVelocity`	`NoSlip`	`pressureInlet OutletVelocity`
Non-hydrostatic pressure - p_rgh	`fixedFluxPressure`	`fixedFluxPressure`	`totalPressure`

Table C.5: Boundary conditions for the NWT and rigid-body

C.3.2 Decisions

Parameter length	Depth	Air-gap	Total height	Inlet	Working-section	Beach
Value in $[m]$	2.8	1.2	4	1	6	13

Table C.6: Geometrical parameters for the $2D$-NWT

Parameter	Cell-size	CPWH	MWL refinement	Courant-Number
Value	$0.02m$	20	level 3	< 0.5

Table C.7: Mesh and simulations decisions for the $2D$-NWT

Correlation Coefficient	ST1	ST2	ST3	ST4
0.9999	28	32	39	52

Table C.8: Number of waves components defined by a 0.9999 correlation between WG#1

C.4 3D-NWT

C.4.1 X-MED

Volume fraction - α	Velocity - `U`
`zeroGradient`	`movingWallVelocity`
Non-hydrostatic pressure - p_rgh	Mesh-deformation `pointDisplacement`
`fixedFluxPressure`	`fixedvalue`$= 0$

Table C.9: Boundary conditions on the rigid-body

Property	OpenFOAM name	Value	Units
Mass	`mass`	21.6	kg
Centre of Mass	`centreOfMass`	$(0\ 0\ 0)$	m
Inertia Matrix	`inertia`	$(0.805\ 0\ 0\ 0.805\ 0\ 0.625)$	$kg.m^{-2}$
Orientation Matrix	–	$(1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$	*radians*

Table C.10: Buoy input characteristics as specified in the `dynamicMeshDict` in the symmetrical CFD model. The non-symmetrical model doubles the mass and inertia.

Case	Joint type	Role
No Symmetry	`floating`	6-DoF
Symmetry	`composite`	Composition of multiples:
	`Px`	displacement along x-axis
	`Pz`	displacement along z-axis
	`Ry`	rotation around y-axis

Table C.11: Joint type for symmetrical and non-symmetrical case

Set-up	(x, y, z) position
Equilibrium Moored	$(5.58, 0.0, -0.133)$
Equilibrium Unmoored	$(5.58, 0.0, -0.122)$
Moored heave decay	$(5.58, 0.0, 0.071)$
Unmoored heave decay	$(5.58, 0.0, 0.064)$
Moored pitch decay	$(5.58, 0.0, -0.133)$

Table C.12: Buoy centre of rotation (equivalent to centre of gravity), `centerOfRotation`, for the different sets-up

Property	OpenFOAM name	Value	Units
Anchor	`anchor`	$(5.58, 0.0, -2.8)$	m
Stiffness	`stiffness`	66.3	$N.m^{-1}$
Symmetry stiffness	`stiffness`	33.15	$N.m^{-1}$
Rest length	`restLength`	2.18	m
Attachment point	`refAttachmentPt`	$(0, 0, -0.181)$	m
Pitch attachment point	`refAttachmentPt`	$(0.05, 0, -0.175)$	m

Table C.13: Mooring line characteristics in the symmetrical CFD model, called `restraints`, for different set-up. The attachment point is an offset from centre of rotation.

C.4.2 Decisions

ULim	Rigid-body surface refinement	NWT width
$< 3\,m.s^{-1}$	level 4	$1.5\,m$

Table C.14: Mesh and simulations decisions for the $3D$-NWT

Appendix D

WaveDyn performance and limits on modelling extreme events

D.1 Decay test resonance frequencies

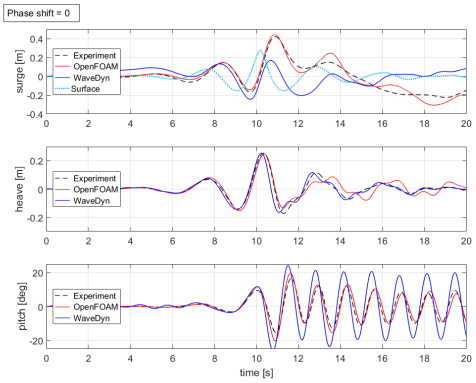
Model	Heave Moored f [Hz]	Heave Unmoored f [Hz]	Pitch Moored [Hz]
Experience	0.917	0.926	0.75
OpenFOAM	0.923	0.907	0.76
WaveDyn	0.926	0.909	0.75

Table D.1: Resonance frequencies

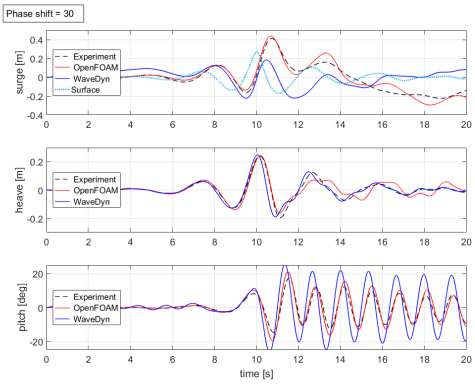
D.2 Extreme WSI

	WaveDyn				OpenFOAM			
	ST1	ST2	ST3	ST4	ST1	ST2	ST3	ST4
η	0.9997	0.9998	0.9998	0.9997	0.9609	0.9658	0.9657	0.9067
Surge	0.3640	0.2923	0.2534	0.2057	0.9561	0.9481	0.9391	0.9703
Heave	0.9844	0.9756	0.9604	0.9652	0.9202	0.9269	0.9339	0.9075
Pitch	0.8982	0.9108	0.9015	0.8278	0.9498	0.9766	0.9881	0.9611

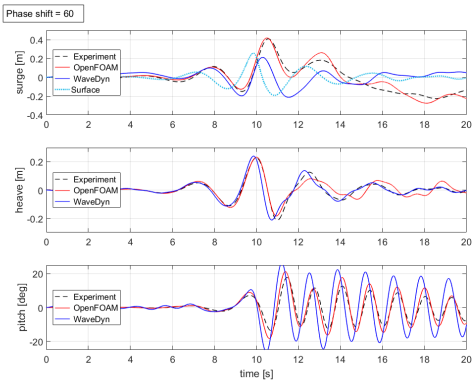
Table D.2: Correlation between each numerical model and the physical reference for each DoF motion response and the surface elevation at WG#14



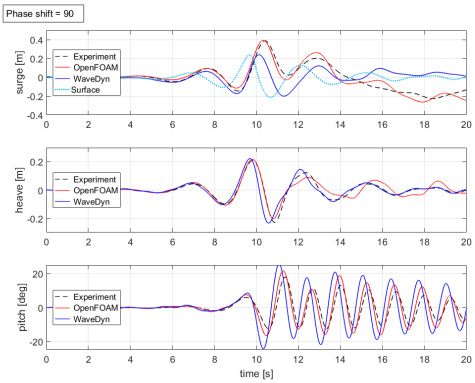
(a) Shift = 0



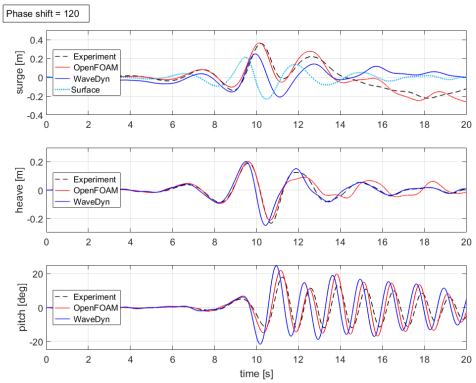
(b) Shift = 30



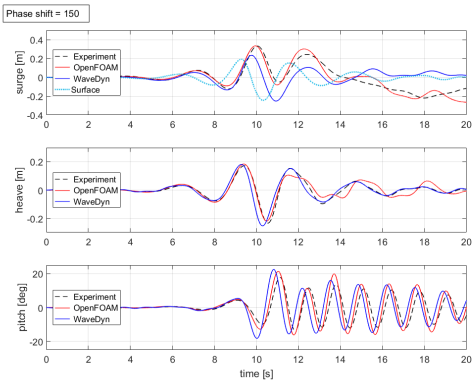
(c) Shift = 60



(d) Shift = 90

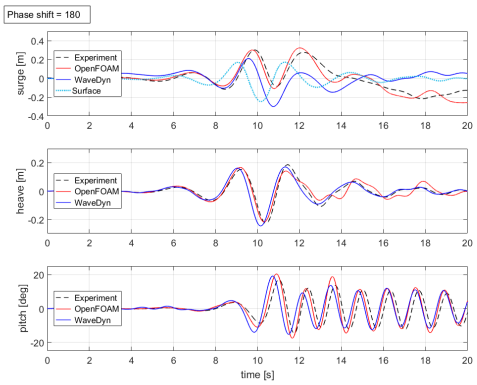


(e) Shift = 120

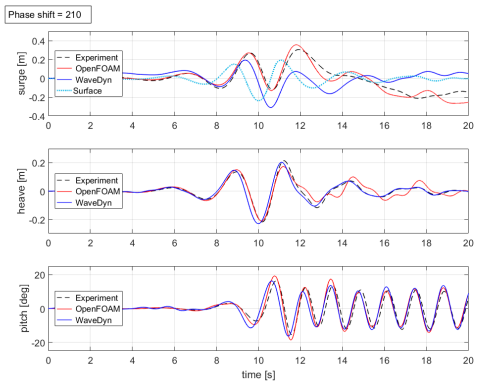


(f) Shift = 150

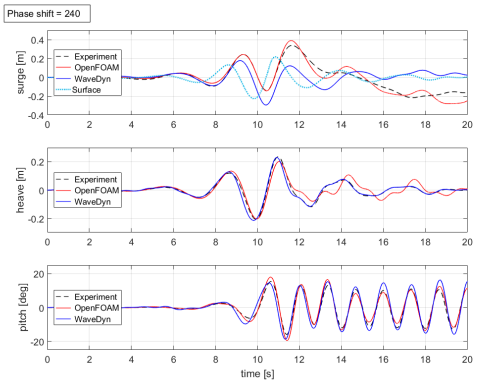
Figure D.1: Motion responses time-series of the X-MED buoy under shifted events (0 - 150) - η is the measurement from WG#10 (i.e. aligned with model centre)



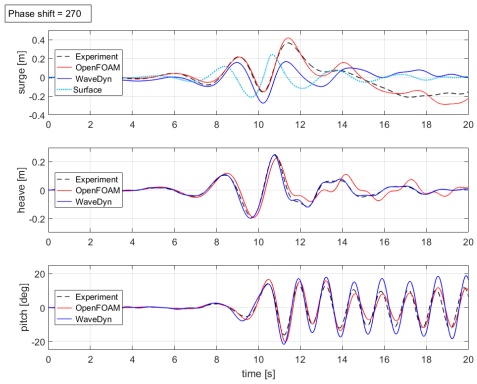
(a) Shift = 180



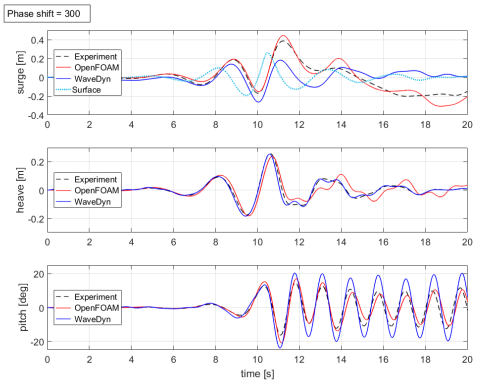
(b) Shift = 210



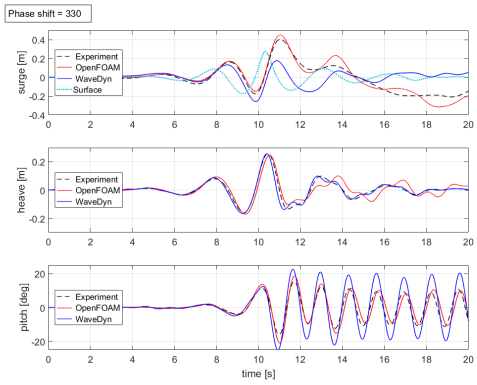
(c) Shift = 240



(d) Shift = 270



(e) Shift = 300



(f) Shift = 330

Figure D.2: Motion responses time-series of the X-MED buoy under shifted events (180 - 330) - η is the measurement from WG#10 (i.e. aligned with model centre)

Appendix E

Developments

E.1 Selection of the number of components for the linear superposition used at wave-input

```
function [numComponents,calculatedSurface, ampl,f, phi] =
    number_of_components_2(timeSeries, precision, fs )
%% Find the number of components necessary to represent the measure within a
    precision
% timeSeries : is a two column matrix [time, surface_elevation];
% fs is the sampling frequency used - usually for lab data 128hz

%% precision is a scalar < 1
if precision >= 1
    error('precision should be <1, usual value: 0.9999')
end

%% Make a spectrum
[~, ampl,f, phi] = FFTcoarse_phm(timeSeries, fs, 0, 1000) ;
% plot(f,ampl)

%% Remove frequencies out of range
fmin = 0 ; fmax = 1000;
for kk = 1:length(f)
    if f(kk)<fmin || f(kk)>fmax
        f = [f(1:kk-1);f(kk+1:end)] ;
        ampl = [ampl(1:kk-1);ampl(kk+1:end)];
        phi = [phi(1:kk-1);phi(kk+1:end)];
    end
end

%% sort amplitude by descending value and return their index
[ sorted_amp , index ] = sort(ampl, 'descend') ;

%% LOOP
```

```
numComponents = 0 ;
correlation_coeff = 0 ;
time_local = [0:1:length(timeSeries(:,2))-1]/fs;
while correlation_coeff < precision

        numComponents = numComponents + 1 ;
    %% Take the 1st X main components of the spectrum and make a new_signal = sum( X mains components)
    newSignal = zeros( length(timeSeries(:,2)) , 1); % Predefines the size of the array
    for kk = 1 : numComponents                                  % Sums the calculated wave components and stores them as a time series
        newSignal = newSignal + sorted_amp(kk)*cos(2*pi*f(index(kk))*time_local' + phi(index(kk)));
    end

    %% IS corr_coeff(new_signal , timeSeries) < precision ?
    hh = corrcoef(newSignal,timeSeries(:,2));
    correlation_coeff = hh(1,2);

end

ampl = sorted_amp(1:numComponents) ;
f_saved = zeros(numComponents,1) ; phi_saved = zeros(numComponents,1) ;
for kk = 1 : numComponents
    f_saved(kk) = f(index(kk)) ;
    phi_saved(kk) = phi(index(kk));
end
f = f_saved ; phi = phi_saved ;
calculatedSurface = newSignal ;
['The number of components needed within an ' num2str(precision) ' accuracy is ' num2str(numComponents) ]
end
```

E.2 Initial mesh deformation: `deformDyMMesh` library

```
/*---------------------------------------------------------------------------
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2016 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

```

```
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
            License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  Application
25      deformDyMMesh
26
27  Description
28      Solver for 2 incompressible, isothermal immiscible fluids using a VOF
29      (volume of fluid) phase-fraction based interface capturing approach,
30      with optional mesh motion and mesh topology changes including adaptive
31      re-meshing.
32
33  \*---------------------------------------------------------------------------
34
35  #include "fvCFD.H"
36  #include "dynamicFvMesh.H"
37
38
39  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
40
41  int main(int argc, char *argv[])
42  {
43      #include "addOverwriteOption.H"
44
45      #include "setRootCase.H"
46      #include "createTime.H"
47
48      const bool overwrite = args.optionFound("overwrite");
49
50      IOdictionary dict(
51      IOobject(
52          "dynamicMeshDict",
53                      runTime.constant(),
54                      runTime,
55                      IOobject::MUST_READ,
56          IOobject::AUTO_WRITE,
```

```
57          true
58                  )
59      );
60
61      word solver(dict.lookupOrDefault<word>("solver","rigidBodyMeshDeform"));
62      // Info << "solver is : " << solver << endl;
63      solver=solver+"Coeffs";
64
65      int deformIter(dict.subDict(solver).lookupOrDefault("deformIter",50));
66
67      dict.subDict(solver).add("initializeDeform","true",true);
68      dict.Foam::regIOobject::write(); // needed to re-write down the
            dynamicMeshDict
69
70      //bool initializeDeform (dict.subDict(solver).lookup("initializeDeform"));
71
72      Info << "Mesh deformation will be done in " << deformIter << " steps" << nl <<
            nl
73      << "If checkMesh fail after the deformation, please increase the number of
            iterations" << nl
74      << "To do so add: " << nl
75      << "   deformIter N; "<< nl
76      << "Where N is the number of iterations - 50 per default" << nl
77      << "to the dynamicMeshDict inside rigiBodyMotionSolverCoeffs." << nl << endl;
78
79      #include "createDynamicFvMesh.H"
80      Info << "Mesh is now a deformable one. Deformation can start" << nl << endl;
81
82      runTime.setDeltaT(1,true); // set time-step to 1
83
84      for (label i=1; i<=deformIter; i++)
85      {
86      Info << "DeformDyMMesh iteration = " << i << nl << endl ;
87
88          mesh.update();
89
90        mesh.moving(true);
91
92      if (!overwrite)
93      {
94          runTime++;
95          runTime.write();
96      }
97       }
98
99
100      if (overwrite)
```

```
101     {
102     runTime+=(0);
103     //runTime++;
104     runTime.write();
105
106     //  deleate deltaT  in  0/uniform/time
107     IOdictionary  timeDict0(
108         IOobject(
109             "0/uniform/time",
110                 runTime,
111                 IOobject::MUST_READ,
112             IOobject::AUTO_WRITE,
113             true
114              )
115         );
116     timeDict0.remove("deltaT");
117     timeDict0.Foam::regIOobject::write();
118      }
119
120      dict.subDict(solver).remove("initializeDeform");
121      dict.Foam::regIOobject::write();  //  needed to re-write down the
           dynamicMeshDict
122
123      Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
124          << "   ClockTime = " << runTime.elapsedClockTime() << " s" << nl << endl;
125          Info<< "End\n" << endl;
126
127      return  0;
128  }
129
130
131  //
       ************************************************************************
       //
```

```
1   /*---------------------------------------------------------------------------
2     =========                 |
3     \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
4      \\    /   O peration     |
5       \\  /    A nd           | Copyright (C) 2016 OpenFOAM Foundation
6        \\/     M anipulation  |
7   -------------------------------------------------------------------------------
8   License
9       This  file  is  part  of  OpenFOAM.
10      OpenFOAM is free software: you can redistribute it  and/or modify  it
11      under the terms of  the  GNU General Public License as published by
```

```
12      the Free Software Foundation, either version 3 of the License, or
13      (at your option) any later version.
14      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
15      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
16      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public
            License
17      for more details.
18      You should have received a copy of the GNU General Public License
19      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
20  \*---------------------------------------------------------------------------
21
22  #include "rigidBodyMeshDeform.H"
23  #include "addToRunTimeSelectionTable.H"
24  #include "polyMesh.H"
25  #include "pointPatchDist.H"
26  #include "pointConstraints.H"
27  #include "uniformDimensionedFields.H"
28  #include "forces.H"
29  #include "mathematicalConstants.H"
30
31  // * * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //
32
33  namespace Foam
34  {
35      defineTypeNameAndDebug(rigidBodyMeshDeform, 0);
36
37      addToRunTimeSelectionTable
38      (
39          motionSolver,
40          rigidBodyMeshDeform,
41          dictionary
42      );
43  }
44
45
46  // * * * * * * * * * * * * * * * * * Constructors * * * * * * * * * * * * * * * //
47
48  Foam::rigidBodyMeshDeform::bodyMesh::bodyMesh
49  (
50      const polyMesh& mesh,
51      const word& name,
52      const label bodyID,
53      const dictionary& dict
54  )
55  :
56      name_(name),
57      bodyID_(bodyID),
```

```
58      patches_(wordReList(dict.lookup("patches"))),
59      patchSet_(mesh.boundaryMesh().patchSet(patches_))
60  {Info  << "  In  rigidBodyMeshDeform.C
        Foam::rigidBodyMeshDeform::bodyMesh::bodyMesh " << nl << endl ;}
61
62
63  Foam::rigidBodyMeshDeform::rigidBodyMeshDeform
64  (
65      const polyMesh& mesh,
66      const IOdictionary& dict
67  )
68  :
69      motionSolver(mesh, dict, typeName),
70      model_
71      (
72          coeffDict() ,
73          IOobject
74          (
75              "rigidBodyMotionState",
76              mesh.time().timeName(),
77              "uniform",
78              mesh
79          ).headerOk()
80        ? IOdictionary
81          (
82              IOobject
83              (
84                  "rigidBodyMotionState",
85                  mesh.time().timeName(),
86                  "uniform",
87                  mesh,
88                  IOobject::READ_IF_PRESENT,
89                  IOobject::NO_WRITE,
90                  false
91              )
92          )
93        : coeffDict()
94      ),
95      test_(coeffDict().lookupOrDefault<Switch>("test", false)),
96      rhoInf_(1.0),
97      rhoName_(coeffDict().lookupOrDefault<word>("rho", "rho")),
98      curTimeIndex_(-1),
99      meshSolverPtr_
100     (
101         motionSolver::New
102         (
103             mesh,
```

```
104                 IOdictionary
105                 (
106                     IOobject
107                     (
108                         "rigidBodyMotionSolver:meshSolver",
109                         mesh.time().constant(),
110                         mesh
111                     ),
112                     coeffDict().subDict("meshSolver")
113                 )
114             )
115         ),
116         meshSolver_(refCast<displacementMotionSolver>(meshSolverPtr_())),
117         initializeDeform(coeffDict().lookupOrDefault<Switch>("initializeDeform",false)),
118         deformIter_(coeffDict().lookupOrDefault<int>("deformIter",50)),
119         iterationNumber(0),
120         finalDeformPosition_(coeffDict().lookupOrDefault<scalarField>("deformPosition",
                scalarField(6,Zero) ))
121     {
122     Info << " In rigidBodyMeshDeform.C
            Foam::rigidBodyMeshDeform::rigidBodyMeshDeform " << nl << endl;
123         if (rhoName_ == "rhoInf")
124         {
125             rhoInf_ = readScalar(coeffDict().lookup("rhoInf"));
126         }
127
128         const dictionary& bodiesDict = coeffDict().subDict("bodies");
129
130         forAllConstIter(IDLList<entry>, bodiesDict, iter)
131         {
132             const dictionary& bodyDict = iter().dict();
133
134             if (bodyDict.found("patches"))
135             {
136                 const label bodyID = model_.bodyID(iter().keyword());
137
138                 if (bodyID == -1)
139                 {
140                     FatalErrorInFunction
141                         << "Body " << iter().keyword()
142                         << " has been merged with another body"
143                             " and cannot be assigned a set of patches"
144                         << exit(FatalError);
145                 }
146
147                 bodyMeshes_.append
148                 (
```

```
149                     new bodyMesh
150                     (
151                         mesh,
152                         iter().keyword(),
153                         bodyID,
154                         bodyDict
155                     )
156                 );
157             }
158         }
159     }
160
161
162     // * * * * * * * * * * * * * * * * * Destructor * * * * * * * * * * * * * * * * * //
163
164     Foam::rigidBodyMeshDeform::~rigidBodyMeshDeform()
165     {}
166
167
168     // * * * * * * * * * * * * * * * * * Member Functions * * * * * * * * * * * * * * * * * //
169
170     Foam::tmp<Foam::pointField>
171     Foam::rigidBodyMeshDeform::curPoints() const
172     {
173     Info<< "Start of rigidBodyMeshDeform::curPoints()" << nl << endl;
174         return meshSolverPtr_->curPoints();
175     }
176
177
178     void Foam::rigidBodyMeshDeform::solve()
179     {
180
181         // Info<< "Start of rigidBodyMeshDeform::solve()" << nl << endl;
182         const Time& t = mesh().time();
183
184     // Info << "Time =" << t.value() << endl;
185
186     // Info << " initializeDeform in rbmd = " << initializeDeform <<endl;
187
188     // Info   << "coeffDict().found(deformPosition)" <<
            coeffDict().found("deformPosition") << endl;
189
190         if (mesh().nPoints() != meshSolver_.points0().size())
191         {
192             FatalErrorInFunction
193                 << "The number of points in the mesh seems to have changed." << endl
194                 << "In constant/polyMesh there are " << meshSolver_.points0().size()
```

```
195             << " points; in the current mesh there are " << mesh().nPoints()
196             << " points." << exit(FatalError);
197     }
198
199     // Store the motion state at the beginning of the time-step
200     if (curTimeIndex_ != this->db().time().timeIndex())
201     {
202         model_.newTime();
203         curTimeIndex_ = this->db().time().timeIndex();
204     }
205
206     if (db().foundObject<uniformDimensionedVectorField>("g"))
207     {
208         model_.g() =
209             db().lookupObject<uniformDimensionedVectorField>("g").value();
210     }
211
212
213     if (test_)
214     {
215         label nIter(readLabel(coeffDict().lookup("nIter")));
216     Info<< "if test_" << nl << endl;
217
218         for (label i=0; i<nIter; i++)
219         {
220     Info<< "for test " << nl << endl;
221             model_.solve
222             (
223                 t.deltaTValue(),
224                 scalarField(model_.nDoF(), Zero),
225                 Field<spatialVector>(model_.nBodies(), Zero)
226             );
227         }
228     }
229     else if (coeffDict().found("deformPosition") && initializeDeform )
230     {
231         iterationNumber++;
232
233     // Info << "Deformation iteration steps = " << iterationNumber << endl;
234
235         scalarField presentDeformPosition = finalDeformPosition_ *
                iterationNumber/deformIter_;
236
237     // Update joint position to next small deformation
238     model_.state().q() = presentDeformPosition; //t.value()/deformIter;
239
240     // - Store the motion state at the beginning of the time-step
```

```
241     model_.newTime(); //  needed otherwize Newmark.C solve() reset to 0
242
243     model_.solve
244     (
245         t.deltaTValue(),
246         scalarField(model_.nDoF(), Zero),
247         Field<spatialVector>(model_.nBodies(), Zero)
248     );
249     // Info  << "END of deformOnly" << nl << endl ;
250      }
251      else
252      {
253     Info  << "else  forces  ???" << nl  << endl ;
254           Field<spatialVector> fx(model_.nBodies(), Zero);
255
256           forAll (bodyMeshes_, bi)
257           {
258               const label  bodyID = bodyMeshes_[bi].bodyID_;
259
260               dictionary  forcesDict;
261               forcesDict.add("type",  functionObjects::forces::typeName);
262               forcesDict.add("patches", bodyMeshes_[bi].patches_);
263               forcesDict.add("rhoInf",  rhoInf_);
264               forcesDict.add("rho",  rhoName_);
265               forcesDict.add("CofR", vector::zero);
266
267               functionObjects::forces  f("forces",  db(),  forcesDict);
268               f.calcForcesMoment();
269
270               fx[bodyID] = spatialVector(f.momentEff(), f.forceEff());
271           }
272
273           model_.solve
274           (
275               t.deltaTValue(),
276               scalarField(model_.nDoF(), Zero),
277               fx
278           );
279      }
280
281      if  (Pstream::master() && model_.report())
282      {
283     //  This  is  the  line  that  prints  the buoy status  if  report  is  on
284           forAll (bodyMeshes_, bi)
285           {
286               model_.status(bodyMeshes_[bi].bodyID_);
287           }
```

```
288
289         }
290
291
292
293         // Update the displacements - VERY IMPORTANT STEP
294         forAll (bodyMeshes_, bi)
295         {
296     Info<< "START in rigidBodyMeshDeform::solve() forAll(bodyMeshes_, bi) " << nl <<
        endl;
297             forAllConstIter (labelHashSet, bodyMeshes_[bi].patchSet_, iter)
298             {
299                 label patchi = iter.key();
300                 pointField patchPoints0
301                 (
302                     meshSolver_.pointDisplacement().boundaryField()[patchi]
303                    .patchInternalField(meshSolver_.points0())
304                 );
305                 meshSolver_.pointDisplacement().boundaryFieldRef()[patchi] ==
306                 (
307                     model_.transformPoints
308                     (
309                         bodyMeshes_[bi].bodyID_,
310                         patchPoints0
311                     ) - patchPoints0
312                 )();
313         }
314         }
315     Info<< "In rigidBodyMeshDeform::solve(); just before meshSolverPtr_->solve()" <<
        nl << endl;
316          meshSolverPtr_->solve();
317     Info<< "End of rigidBodyMeshDeform::solve()" << nl << endl;
318     }
319
320
321     bool Foam::rigidBodyMeshDeform::writeObject
322     (
323         IOstream::streamFormat fmt,
324         IOstream::versionNumber ver,
325         IOstream::compressionType cmp
326     ) const
327     {
328         IOdictionary dict
329         (
330             IOobject
331             (
332                 "rigidBodyMotionState",
```

```
333                 mesh().time().timeName(),
334                 "uniform",
335                 mesh(),
336                 IOobject::NO_READ,
337                 IOobject::NO_WRITE,
338                 false
339             )
340         );
341 
342         model_.state().write(dict);
343         return dict.regIOobject::write();
344 }
345 
346 
347 bool Foam::rigidBodyMeshDeform::read()
348 {
349         if (motionSolver::read())
350         {
351             model_.read(coeffDict());
352 
353             return true;
354         }
355         else
356         {
357             return false;
358         }
359 }
360 
361 // not used by deformDyMFoam
362 void Foam::rigidBodyMeshDeform::movePoints(const pointField& points)
363 {
364 Info<< "Start of rigidBodyMeshDeform::movePoints" << nl << endl;
365     meshSolverPtr_->movePoints(points);
366 Info<< "Start of rigidBodyMeshDeform::movePoints" << nl << endl;
367 }
368 
369 // not used by deformDyMFoam
370 void Foam::rigidBodyMeshDeform::updateMesh(const mapPolyMesh& mpm)
371 {
372 Info<< "Start of rigidBodyMeshDeform::updateMesh" << nl << endl;
373     meshSolverPtr_->updateMesh(mpm);
374 Info<< "Start of rigidBodyMeshDeform::updateMesh" << nl << endl;
375 }
376 
377 
378 //
    ************************************************************************
```

```
    //
_____________________________________________________________________
_____________________________________________________________________
/*---------------------------------------------------------------------
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2016 OpenFOAM Foundation
     \\/     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/>.
Class
    Foam::rigidBodyMeshDeform
Description
    Rigid-body mesh motion solver for fvMesh.
    Applies septernion interpolation of movement as function of distance to the
    object surface.
SourceFiles
    rigidBodyMeshDeform.C
\*---------------------------------------------------------------------

#ifndef rigidBodyMeshDeform_H
#define rigidBodyMeshDeform_H

#include "displacementMotionSolver.H"
#include "rigidBodyMotion.H"

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

namespace Foam
{

/*---------------------------------------------------------------------
        Class rigidBodyMeshDeform Declaration
\*---------------------------------------------------------------------
```

```
44
45 class rigidBodyMeshDeform
46 :
47     public motionSolver
48 {
49     // - Class containing the patches and point motion weighting for each body
50     class bodyMesh
51     {
52         // - Name of the body
53         const word name_;
54
55         // - ID of the body in the RBD::rigidBodyMotion
56         const label bodyID_;
57
58         // - List of mesh patches associated with this body
59         const wordReList patches_;
60
61         // - Patches to integrate forces
62         const labelHashSet patchSet_;
63
64
65     public:
66
67         friend class rigidBodyMeshDeform;
68
69         bodyMesh
70         (
71             const polyMesh& mesh,
72             const word& name,
73             const label bodyID,
74             const dictionary& dict
75         );
76     };
77
78
79     // Private data
80
81         // - Rigid-body model
82         RBD::rigidBodyMotion model_;
83
84         // - List of the bodyMeshes containing the patches and point motion
85         //   weighting for each body
86         PtrList<bodyMesh> bodyMeshes_;
87
88         // - Switch for test-mode in which only the
89         //   gravitational body-force is applied
90         Switch test_;
```

```
91
92          // - Reference density required by the forces object for
93          //   incompressible calculations, required if rho == rhoInf
94          scalar rhoInf_;
95
96          // - Name of density field, optional unless used for an
97          //   incompressible simulation, when this needs to be specified
98          //   as rhoInf
99          word rhoName_;
100
101         // - Current time index (used for updating)
102         label curTimeIndex_;
103
104         autoPtr<motionSolver> meshSolverPtr_;
105
106         displacementMotionSolver& meshSolver_;
107
108     // boolean
109     Switch   initializeDeform ;       //bool initializeDeform   ;
110
111     // deformIter
112     int deformIter_ ;
113
114     int iterationNumber;
115
116     // deformPosition
117     scalarField finalDeformPosition_;
118
119
120      // Private Member Functions
121
122         // - Disallow default bitwise copy construct
123         rigidBodyMeshDeform
124         (
125             const rigidBodyMeshDeform&
126         );
127
128         // - Disallow default bitwise assignment
129         void operator=(const rigidBodyMeshDeform&);
130
131
132 public:
133
134     // - Runtime type information
135     TypeName("rigidBodyMeshDeform");
136
137
```

```
138         //  Constructors
139
140             // - Construct from polyMesh and IOdictionary
141             rigidBodyMeshDeform
142             (
143                 const polyMesh&,
144                 const IOdictionary& dict
145             );
146
147
148         // - Destructor
149         ~rigidBodyMeshDeform();
150
151
152         //  Member Functions
153
154             // - Return point location  obtained from the current  motion  field
155             virtual  tmp<pointField> curPoints() const;
156
157             // - Solve for  motion
158             virtual  void  solve();
159
160             // - Write state  using given format,  version  and compression
161             virtual  bool writeObject
162             (
163                 IOstream::streamFormat fmt,
164                 IOstream::versionNumber ver,
165                 IOstream::compressionType cmp
166             )  const;
167
168             // - Read dynamicMeshDict dictionary
169             virtual  bool read();
170
171             // - Update local data for  geometry changes
172             virtual  void  movePoints(const pointField&);
173
174             // -  Update local data  for  topology changes
175             virtual  void  updateMesh(const mapPolyMesh&);
176 };
177
178
179 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
180
181 }  //  End namespace Foam
182
183 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
184
```

```
185 #endif
186
187 //
    ********************************************************************************
    //
```

E.3 Coupling

E.3.1 External Load Controller

```
1  /*********************************************************************************
2  *
3  * External Load Controller Developped for Hot-Start with OpenFOAM-CFD solver
4  *  author: Pierre-Henri MUSIEDLAK - PhD at Univeristy of Plymouth
5  *  supervisor: Benjamin Child : benjamin.child@dnvgl.com
6  *********************************************************************************/
7
8
9  #include "ExternalLoadsDll.h" /**< this provides a definition of IExternalLoads. */
10 #include "functions.h"
11 #include <iostream>
12 #include <fstream>
13 using namespace std;
14 #include <array>
15
16 vector<double> time_simu;
17 vector<double> time_simu_complete;
18 vector<GHExternalLoads::DOF3> dof_simu;
19 vector<GHExternalLoads::DOF3> rot_simu;
20 vector<GHExternalLoads::DOF3> dof_simu_complete;
21 vector<GHExternalLoads::DOF3> rot_simu_complete;
22 vector<GHExternalLoads::DOF3> speed_simu;
23 vector<GHExternalLoads::DOF3> acc_simu;
24
25 // Buoy characteristics
26 double hydro_stiffness[6][6] = { { 0 , 0, 0,        0,        0,        0 } ,
27 { 0 , 0, 0,        0,        0,        0 } ,
28 { 0 , 0, 1925.53, 0,        0,        0 } ,
29 { 0 , 0, 0,        30.0866, 0,        0 } ,
30 { 0 , 0, 0,        0,        30.0865, 0 },
31 { 0 , 0, 0,        0,        0,        0 } };
32
33 GHExternalLoads::DOF3 position_rest = { 0, 0, 2.667 }; // [m][m][m]
34 GHExternalLoads::DOF3 rotation_rest = { 0, 0, 0 };    // [rad][rad][rad]
35 double mean_displaced_mass = 45.2913; //[kg]
36 double inertia[3][3] = { { 1.61, 0, 0 },{ 0, 1.61, 0 },{ 0,0,1.25 } }; // kgm^2
37 double mass = 43.2; //[kg]
```

```
38
39 double added_mass[6][6] = { { 14.1435, 0, 0, 0, 1.02951, 0 },
40                           { 0, 14.1436, 0, −1.02952, 0, 0 },
41                           { 0, 0, 17.5333, 0, 0, 0 },
42                           { 0, −1.02952, 0, 0.0758922, 0, 0 },
43                           { 1.02951, 0, 0, 0, 0.0758921, 0 },
44                           { 0, 0, 0, 0, 0, 2.06116E−13 } };
45
46 // Environmental properties
47 double rho = 1000; // water density [kg/m^3]
48 double gravity = 9.81; // gravity [m/s^2]
49 double pi = 3.14159265;
50 vector<double> eta; // surface elevation [m]
51 vector<double> time_eta;
52 vector<double> ika; // instantaneous steepness (i.e. derivative of eta) [m/s]
53 double ika_threshold = 0.54; // Max ika WaveDyn is capable of
54
55 // COUPLING
56 double t_trigger = 40; [s] // a manual trigger to stop WaveDyn solving and
       swap to OpenFOAM
57 double t_minus = 4; [s] // the backward time the simus needs to go for the hot−start
58
59  // for the sake of coupling : input previously independantly ran WaveDyn
       simulation
60 int read_input_wdy = 0;
61
62 // six dof coupling at Swapping time : t_hot − t_minus
63 GHExternalLoads::DOF3 position_at_swap;
64 GHExternalLoads::DOF3 rotation_at_swap;
65
66 // OpenFOAM inputs
67 int read_opf_input = 0;
68 GHExternalLoads::DOF3 position_rest_opf = { 5.58 , 0 , −0.133 };
69 GHExternalLoads::DOF3 rotation_rest_opf = { 0 , 0 , 0 };
70 // Pressure load and moment
71 vector<double> time_input = { 0 };
72 vector<double> px = { 0 };
73 vector<double> py = { 0 };
74 vector<double> pz = { 0 };
75 vector<double> p_m_x = { 0 };
76 vector<double> p_m_y = { 0 };
77 vector<double> p_m_z = { 0 };
78 // Viscous load and moment
79 vector<double> vx = { 0 };
80 vector<double> vy = { 0 };
81 vector<double> vz = { 0 };
82 vector<double> v_m_x = { 0 };
```

```
83  vector<double> v_m_y = { 0 };
84  vector<double> v_m_z = { 0 };
85  //  6-DoF
86  vector<double> dofx = { 0  };
87  vector<double> dofy = { 0  };
88  vector<double> dofz = { 0  };
89  vector<double> dofroll  = {  0  };
90  vector<double> dofpitch = {  0  };
91  vector<double> dofyaw = { 0 };
92  vector<double> d_roll_dt = {  0  };
93  vector<double> d_pitch_dt = { 0  };
94  vector<double> d_yaw_dt = { 0 };
95  //  Mooring Load
96  vector<GHExternalLoads::DOF3> mooring_load = { { 0, 0 ,0 } };
97  vector<GHExternalLoads::DOF3> mooring_moment = { { 0, 0 ,0 } };
98  vector<double> mooring_load_x = { 0 };
99  vector<double> mooring_load_y = { 0 };
100 vector<double> mooring_load_z = { 0 };
101
102 //vector<double> fake_moment = { 0 };
103 vector<GHExternalLoads::DOF3> sum_of_forces_opf;
104 vector<GHExternalLoads::DOF3> speed_opf;
105 vector<GHExternalLoads::DOF3> acc_opf;
106
107 //  Exciation  load
108 vector<double> time_input_wdy = { 0 };
109 vector<double> excitation_x = {  0  };
110 vector<double> excitation_y = {  0  };
111 vector<double> excitation_z= {  0  };
112 vector<double>  excitation_roll  = {  0  };
113 vector<double> excitation_pitch  = {  0  };
114 vector<double> excitation_yaw = {  0  };
115
116 //  Radiation load
117 vector<double> radiation_x = {  0  };
118 vector<double> radiation_y = {  0  };
119 vector<double> radiation_z = {  0  };
120 vector<double>  radiation_roll  = {  0  };
121 vector<double> radiation_pitch = {  0  };
122 vector<double> radiation_yaw = {  0  };
123
124 //  6dof
125 vector<double> dof_wdy_x = { 0 };
126 vector<double> dof_wdy_y = { 0 };
127 vector<double> dof_wdy_z = { 0 };
128 vector<double> dof_wdy_roll= { 0 };
129 vector<double> dof_wdy_pitch = { 0 };
```

```
130 vector<double> dof_wdy_yaw = { 0 };
131
132 // speeds
133 vector<double> speed_wdy_x= { 0 };
134 vector<double> speed_wdy_y = { 0 };
135 vector<double> speed_wdy_z = { 0 };
136 vector<double> speed_wdy_roll = { 0 };
137 vector<double> speed_wdy_pitch = { 0 };
138 vector<double> speed_wdy_yaw = { 0 };
139
140 // Added Mass load
141 vector<double> ma_wdy_x = { 0 };
142 vector<double> ma_wdy_y = { 0 };
143 vector<double> ma_wdy_z = { 0 };
144 vector<double> ma_wdy_roll = { 0 };
145 vector<double> ma_wdy_pitch = { 0 };
146 vector<double> ma_wdy_yaw = { 0 };
147
148 /// <summary>Initialise is called once at the start of the simulation. This is
149 /// intended as a point where the user can set up logging variables and initialise
150 /// their own code. Loads should not be applied during this call .</summary>
151
152 extern "C" void __declspec(dllexport) __cdecl
        Initialise (GHExternalLoads::IExternalLoads* ext_loads)
153 {
154     ext_loads->SetLoggingLevel(GH_DISCON_LOG_INFO); /**< This sets the
            threshhold of the logging so that Notes are displayed. */
155
156     wstring message = L"Using input from ";
157     message += ext_loads->GetInputFilepath();
158     ext_loads->ReportInfoMessage(std::string(message.begin(),
            message.end()).c_str());
159     /**< This writes a Note to the console and the $ME file. */
160
161     /// Setting up logging:
162     ext_loads->AddLogValue("Time", "T");
163     /**< This adds a logging variable 'Time', whose units are T (for Time). */
164     ext_loads->AddLogValue("X position", "L");
165     /**< This adds a logging variable 'X position', whose units are L (for Length).
            */
166     ext_loads->AddLogValue("Y position", "L");
167     ext_loads->AddLogValue("Z position", "L");
168
169     ext_loads->AddLogValue("X force", "N");
170     ext_loads->AddLogValue("Y force", "N");
171     ext_loads->AddLogValue("Z force", "N");
172
```

```
173     ext_loads->AddLogValue("X Moment", "Nm");
174     ext_loads->AddLogValue("Y Moment", "Nm");
175     ext_loads->AddLogValue("Z Moment", "Nm");
176
177     ext_loads->ReportInfoMessage("Initialising external loads dll ... ");
178
179     double time_step = ext_loads->GetSimulationCurrentTime();
180     GHExternalLoads::DOF3 position = ext_loads->GetMultibodyNodePosition("C");
181     GHExternalLoads::DOF3 rotation =
            ext_loads->GetMultibodyNodeOrientation("C");
182
183     time_simu.push_back(time_step);
184     dof_simu.push_back(position);
185     rot_simu.push_back(rotation);
186
187     time_simu_complete.push_back(time_step);
188     dof_simu_complete.push_back(position);
189     rot_simu_complete.push_back(rotation);
190
191     // clean OpenFOAM output file
192     ofstream
            all6dof_input_for_opf("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\all6dof_constrainedNWE52_hotstart");
193     all6dof_input_for_opf << time_step << "\t" << -position.x - position_rest.x <<
            "\t" <<
194         -position.y - position_rest.y << "\t" <<
195         position.z - position_rest.z << "\t" <<
196         -rotation.x << "\t" << -rotation.y << "\t" << -rotation.z << endl;
197     all6dof_input_for_opf.close();
198
199     // surface-eleavtion
200     double a;
201     int kk = 0;
202     ifstream dotseafile("C:\\Users\\pmusiedlak\\OneDrive - University of
            Plymouth\\waveDyn\\AsSimilarToOpenFOAMasPossible\\XMED_6Dof_ConstrainedNWE52\\XMED_6D
203     while (dotseafile >> a)
204     {
205         time_eta.push_back(kk*0.01);
206         kk = kk + 1;
207         eta.push_back(a);
208     }
209     // ika
210     int ind = eta.size();
211         for (int kk = 1; kk < ind; kk++)
212     {
213         ika.push_back((eta[kk] - eta[kk - 1]) / (time_eta[kk] - time_eta[kk - 1]));
214     }
215
```

```
216 }
217
218 /// <summary>ApplyExternalLoads is called every time that loads are applied to
219 /// the multibody model - which includes the integrator's trial steps, which are
220 /// then ignored.</summary>
221
222 extern "C" void __declspec(dllexport) __cdecl
        ApplyExternalLoads(GHExternalLoads::IExternalLoads* ext_loads)
223 {
224     time_simu.push_back(ext_loads->GetSimulationCurrentTime());
225
226     GHExternalLoads::DOF3 position = ext_loads->GetMultibodyNodePosition("C");
227     GHExternalLoads::DOF3 rotation =
            ext_loads->GetMultibodyNodeOrientation("C");
228     /**< This interrogates the multibody model for the position of node 'C'. An
            error will be raised if there is no node of this name. */
229
230     dof_simu.push_back(position);
231     rot_simu.push_back(rotation);
232
233     speed_simu.push_back(ext_loads->GetMultibodyNodeVelocity("C"));
234     // Acceleration:
235     GHExternalLoads::DOF3 acc;
236     int ind = speed_simu.size();
237     acc.x = (speed_simu[ind].x - speed_simu[ind - 1].x) / (time_simu[ind] -
            time_simu[ind - 1]);
238     acc.y = (speed_simu[ind].y - speed_simu[ind - 1].y) / (time_simu[ind] -
            time_simu[ind - 1]);
239     acc.z = (speed_simu[ind].z - speed_simu[ind - 1].z) / (time_simu[ind] -
            time_simu[ind - 1]);
240     acc_simu.push_back(acc);
241
242     /// Setting the log values on each call of ApplyExternalLoads:
243     ext_loads->GetLogValueByName("Time")->SetLoggingValue(ext_loads->GetSimulationCurren
244     ext_loads->GetLogValueByName("X position")->SetLoggingValue(position.x);
245     ext_loads->GetLogValueByName("Y position")->SetLoggingValue(position.y);
246     ext_loads->GetLogValueByName("Z position")->SetLoggingValue(position.z);
247
248         // IkA @ time-step
249     int pp = 0;
250     while (time_eta[pp] < time_simu.back())
251     {
252         pp++;
253     }
254
255     if (read_opf_input == 0 && ( abs(ika[pp]) >= ika_threshold || time_simu.back()
            >= t_trigger ))
```

```
256     {
257         t_trigger = time_simu_complete.back();
258         cout << "Trigger got activated - we now run OpenFOAM starting at t_trigger
                - t_minus" << endl;
259         int jj = 0;
260         while (time_simu_complete[jj] < t_trigger - t_minus)
261         {
262             jj++;
263         }
264         cout << "jj = " << jj << endl;
265
266         ofstream
                hot_start_input_for_opf("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\6_DoF_constrainedNWE52_hotsta
267
268         // Print WaveDyn output (rigid-body motion state) for OpenFOAM hot-start
269         hot_start_input_for_opf << "Time :\t" << time_simu_complete[jj] << "s" <<
                endl;
270         hot_start_input_for_opf << "Offsets :\t" << -dof_simu_complete[jj].x -
                position_rest.x << "\t" <<
271             -dof_simu_complete[jj].y - position_rest.y << "\t" <<
272             dof_simu_complete[jj].z - position_rest.z << "\t" <<
273             -rot_simu_complete[jj].x - rotation_rest.x << "\t" <<
274             -rot_simu_complete[jj].y - rotation_rest.y << "\t" <<
275             -rot_simu_complete[jj].z - rotation_rest.z << "\t" << endl;
276         hot_start_input_for_opf.close();
277
278         position_at_swap = { -dof_simu_complete[jj].x - position_rest.x ,
                -dof_simu_complete[jj].y - position_rest.y , dof_simu_complete[jj].z -
                position_rest.z };
279         rotation_at_swap = { -rot_simu_complete[jj].x - rotation_rest.x ,
                -rot_simu_complete[jj].y - rotation_rest.y , -rot_simu_complete[jj].z -
                rotation_rest.z };
280
281         ofstream
                all6dof_input_for_opf("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\all6dof_constrainedNWE52_hotstart"
                std::ofstream::app);
282         all6dof_input_for_opf << time_simu.back() << "\t" << -position.x -
                position_rest.x << "\t" <<
283             -position.y - position_rest.y << "\t" <<
284             position.z - position_rest.z << "\t" <<
285             -rotation.x << "\t" << -rotation.y << "\t" << -rotation.z << endl;
286         all6dof_input_for_opf.close();
287
288         cout << "Waiting loop while OpenFOAM is running ; i.e. checking the file for
                updates" << endl;
289         ifstream
                is_openfoam_finished("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\pressure_loads_constrainedNWE52_
```

```
290
291         bool is_good = is_openfoam_finished.good();
292         while (!is_good)
293         {
294             ifstream
                    is_openfoam_finished("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\pressure_loads_constrai
295
296             is_good = is_openfoam_finished.good();
297             _sleep(10000);
298         }
299         cout << "OpenFOAM simu is now finished - restart the WaveDyn one with
                input from OpenFOAM" << endl;
300         cout << "time_simu.back() = " << time_simu.back() << "\t t_tigger = " <<
                t_trigger << "\t read_opf_input = " << read_opf_input << endl;
301         double t;
302         double a, b, c, d, e, f;
303
304         ifstream
                fin ("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\pressure_loads_constrainedNWE52_hotstart.
305
306         cout << "import pressure loads " << endl;
307         while (fin >> t >> a >> b >> c >> d >> e >> f)
308         {
309             time_input.push_back(t);
310             px.push_back(a);
311             py.push_back(b);
312             pz.push_back(c);
313             p_m_x.push_back(d);
314             p_m_y.push_back(e);
315             p_m_z.push_back(f);
316         }
317
318         a = 0; b = 0; c = 0; d = 0; e = 0; f = 0;
319         ifstream
                fin_v("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\viscous_loads_constrainedNWE52_hotstar
320
321         cout << "import viscous loads " << endl;
322         while (fin_v >> t >> a >> b >> c >> d >> e >> f)
323         {
324             vx.push_back(a);
325             vy.push_back(b);
326             vz.push_back(c);
327             v_m_x.push_back(d);
328             v_m_y.push_back(e);
329             v_m_z.push_back(f);
330         }
331
```

```
332     a = 0; b = 0; c = 0; d = 0; e = 0; f = 0;
333     ifstream
            fin_m("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\t_vs_springLoad_constrainedNWE52_hotstart.txt");
334
335     cout << "import Mooring loads " << endl;
336     while (fin_m >> t >> a >> b >> c >> d >> e >> f)
337     {
338         mooring_load.push_back({ a, b, c });
339         mooring_moment.push_back({ d, e, f });
340     }
341
342     cout << "Don't import the mooring moment - as calculated later " << endl;
343     cout << "calculate the sum of forces in OpenFOAM" << endl;
344     int ind = time_input.size();
345     double x, y, z;
346     for (int kk = 0; kk <= ind; kk++)
347     {
348         x = px[kk] + vx[kk] + mooring_load[kk].x;
349         y = py[kk] + vy[kk] + mooring_load[kk].y;
350         z = pz[kk] + vz[kk] + mooring_load[kk].z - gravity*mass;
351         sum_of_forces_opf.push_back({ x,y,z });
352     }
353
354     a = 0; b = 0; c = 0; d = 0; e = 0; f = 0;
355     // read the 6DoF file
356     ifstream
            fin_dof("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\6Dof_opf_constrainedNWE52_hotstart.txt");
357
358     cout << "import 6DoF " << endl;
359     while (fin_dof >> t >> a >> b >> c >> d >> e >> f)
360     {
361         dofx.push_back(a);
362         dofy.push_back(b);
363         dofz.push_back(c);
364         d_roll_dt.push_back(d);
365         d_pitch_dt.push_back(e);
366         d_yaw_dt.push_back(f);
367     }
368
369     cout << "Integrate roll, pitch, yaw OpenFOAM inputs as they are angular
            velocities." << endl;
370     ind = d_roll_dt.size();
371     x = 0; y = 0; z = 0;
372     GHExternalLoads::DOF3 int_a_b = { 0,0,0 };
373     GHExternalLoads::DOF3 cum_int = { 0,0,0 };
374     for (int kk = 1; kk <= ind; kk++)
375     {
```

```
376         x = (dofx[kk] - dofx[kk - 1]) / (time_input[kk] - time_input[kk - 1]);
377         y = (dofy[kk] - dofy[kk - 1]) / (time_input[kk] - time_input[kk - 1]);
378         z = (dofz[kk] - dofz[kk - 1]) / (time_input[kk] - time_input[kk - 1]);
379         speed_opf.push_back({ x,y,z });
380
381         if (kk >= 2)
382         {
383             x = (speed_opf[kk].x - speed_opf[kk - 1].x) / (time_input[kk] -
                    time_input[kk - 1]);
384             y = (speed_opf[kk].y - speed_opf[kk - 1].y) / (time_input[kk] -
                    time_input[kk - 1]);
385             z = (speed_opf[kk].z - speed_opf[kk - 1].z) / (time_input[kk] -
                    time_input[kk - 1]);
386             acc_opf.push_back({ x,y,z });
387         }
388
389         int_a_b.x = 0.5*(time_input[kk] - time_input[kk - 1])*(d_roll_dt[kk - 1] +
                d_roll_dt[kk]);
390         cum_int.x = cum_int.x + int_a_b.x;
391         dofroll.push_back(cum_int.x);
392
393         int_a_b.y = 0.5*(time_input[kk] - time_input[kk - 1])*(d_pitch_dt[kk - 1]
                + d_pitch_dt[kk]);
394         cum_int.y = cum_int.y + int_a_b.y;
395         dofpitch.push_back(cum_int.y);
396
397         int_a_b.z = 0.5*(time_input[kk] - time_input[kk - 1])*(d_yaw_dt[kk - 1] +
                d_yaw_dt[kk]);
398         cum_int.z = cum_int.z + int_a_b.z;
399         dofyaw.push_back(cum_int.z);
400     }
401     cout << "dofpitch[1] = " << dofpitch[1] << endl;
402     cout << "Calculate the Moments at the buoy centre of Gravity : M[G] = M[O] +
            GO^Force" << endl;
403     ind = p_m_x.size();
404     for (int kk = 1; kk <= ind; kk++)
405     {
406         GHExternalLoads::DOF3 vector_f = { px[kk] , py[kk], pz[kk] };
407         GHExternalLoads::DOF3 go = { -dofx[kk] , -dofy[kk], -dofz[kk] };
408         GHExternalLoads::DOF3 c;
409         c = cross_product(go, vector_f);
410         p_m_x[kk] = p_m_x[kk] + c.x;
411         p_m_y[kk] = p_m_y[kk] + c.y;
412         p_m_z[kk] = p_m_z[kk] + c.z;
413
414         vector_f = { vx[kk] , vy[kk], vz[kk] };
415         c = cross_product(go, vector_f);
```

```
416             v_m_x[kk] = v_m_x[kk] + c.x;
417             v_m_y[kk] = v_m_y[kk] + c.y;
418             v_m_z[kk] = v_m_z[kk] + c.z;
419
420
421             GHExternalLoads::DOF3 vector_anchor_G = { 5.58 - dofx[kk] , 0 -
                    dofy[kk] , -2.8 - dofz[kk] };
422             mooring_moment[kk] = cross_product(vector_anchor_G, mooring_load[kk]);
423         }
424
425         read_opf_input = read_opf_input + 1;
426         cout << "Finish importing all OpenFOAM outpout, simu can now restart" <<
                endl;
427     }
428
429     if (read_opf_input == 0 && read_input_wdy == 0)
430     {
431         // input previously independantly ran WaveDyn simulation
432         cout << "Input previously independantly ran WaveDyn simulation" << endl;
433
434         // Input directly from already run WaveDyn case
435         double a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r;
436         ifstream fin_wdy_alreadyruncaseLoads("C:\\Users\\pmusiedlak\\OneDrive -
                University of
                Plymouth\\waveDyn\\AsSimilarToOpenFOAMasPossible\\XMED_6Dof_ConstrainedNWE52\\XMED_
437         while (fin_wdy_alreadyruncaseLoads >> a >> b >> c >> d >> e >> f >> g >> h
                >> i >> j >> k >> l >> m >> n >> o >> p >> q >> r )
438         {
439             time_input_wdy.push_back( (time_input_wdy.size())*0.01);
440             excitation_x.push_back(a);
441             excitation_y.push_back(b);
442             excitation_z.push_back(c);
443             excitation_roll.push_back(d);
444             excitation_pitch.push_back(e);
445             excitation_yaw.push_back(f);
446
447             radiation_x.push_back(g);
448             radiation_y.push_back(h);
449             radiation_z.push_back(i);
450             radiation_roll.push_back(j);
451             radiation_pitch.push_back(k);
452             radiation_yaw.push_back(l);
453         }
454
455         ifstream fin_wdy_alreadyruncaseAcc("C:\\Users\\pmusiedlak\\OneDrive -
                University of
                Plymouth\\waveDyn\\AsSimilarToOpenFOAMasPossible\\XMED_6Dof_ConstrainedNWE52\\XMED_
```

```
456         while (fin_wdy_alreadyruncaseAcc >> a >> b >> c >> d >> e >> f >> g >> h
                >> i >> j >> k >> l)
457         {
458             dof_wdy_x.push_back(a);
459             dof_wdy_x.push_back(b);
460             dof_wdy_x.push_back(c);
461 
462             speed_wdy_x.push_back(d);
463             speed_wdy_y.push_back(e);
464             speed_wdy_z.push_back(f);
465 
466             dof_wdy_roll.push_back(g);
467             dof_wdy_pitch.push_back(h);
468             dof_wdy_yaw.push_back(i);
469 
470             speed_wdy_roll.push_back(j);
471             speed_wdy_pitch.push_back(k);
472             speed_wdy_yaw.push_back(l);
473         }
474 
475         // calculate acc
476         for (int kk = 1; kk <= time_input_wdy.size(); kk++)
477         {
478             ma_wdy_x.push_back(-(speed_wdy_x[kk] - speed_wdy_x[kk - 1]) /
                    (time_input_wdy[kk] - time_input_wdy[kk - 1]));
479             ma_wdy_y.push_back(-(speed_wdy_y[kk] - speed_wdy_y[kk - 1]) /
                    (time_input_wdy[kk] - time_input_wdy[kk - 1]));
480             ma_wdy_z.push_back((speed_wdy_z[kk] - speed_wdy_z[kk - 1]) /
                    (time_input_wdy[kk] - time_input_wdy[kk - 1]));
481 
482             ma_wdy_roll.push_back(-(speed_wdy_roll[kk] - speed_wdy_roll[kk - 1]) /
                    (time_input_wdy[kk] - time_input_wdy[kk - 1]));
483             ma_wdy_pitch.push_back(-(speed_wdy_pitch[kk] - speed_wdy_pitch[kk -
                    1]) / (time_input_wdy[kk] - time_input_wdy[kk - 1]));
484             ma_wdy_yaw.push_back(-(speed_wdy_yaw[kk] - speed_wdy_yaw[kk -
                    1]) / (time_input_wdy[kk] - time_input_wdy[kk - 1]));
485         }
486 
487         cout << "after the import of the WaveDyn files " << endl;
488         read_input_wdy = read_input_wdy + 1;
489     }
490 
491     // Find closest time-step
492     int ii = 0;
493     if (time_simu.back() >= t_trigger)
494     {
495         while (time_input[ ii ] < time_simu.back())
```

```
496         {
497             ii  =  ii  + 1;
498         }
499     }
500     else
501     {
502         while (time_input_wdy[ii] < time_simu.back())
503         {
504             ii  =  ii  + 1;
505         }
506     }
507 
508     GHExternalLoads::DOF3 p_at_timestep;
509     p_at_timestep.x = px[ ii ];
510     p_at_timestep.y = py[ ii ];
511     p_at_timestep.z = pz[ ii ];
512 
513     GHExternalLoads::DOF3 p_m_at_timestep;
514     p_m_at_timestep.x = p_m_x[ii];
515     p_m_at_timestep.y = p_m_y[ii];
516     p_m_at_timestep.z = p_m_z[ii];
517 
518     GHExternalLoads::DOF3 v_at_timestep;
519     v_at_timestep.x = vx[ ii ];
520     v_at_timestep.y = vy[ ii ];
521     v_at_timestep.z = vz[ ii ];
522 
523     GHExternalLoads::DOF3 v_m_at_timestep;
524     v_m_at_timestep.x = v_m_x[ii];
525     v_m_at_timestep.y = v_m_y[ii];
526     v_m_at_timestep.z = v_m_z[ii];
527 
528     GHExternalLoads::DOF3 dof_at_timestep;
529     dof_at_timestep.x = dofx[ ii ];
530     dof_at_timestep.y = dofy[ ii ];
531     dof_at_timestep.z = dofz[ ii ];
532 
533     GHExternalLoads::DOF3 rot_at_timestep;
534     rot_at_timestep.x =  dofroll [ ii ];
535     rot_at_timestep.y = dofpitch[ ii ];
536     rot_at_timestep.z = dofyaw[ii ];
537 
538     GHExternalLoads::DOF3 mooring_load_at_timestep;
539     mooring_load_at_timestep = mooring_load[ii];
540 
541     GHExternalLoads::DOF3 mooring_moment_at_timestep;
542     mooring_moment_at_timestep = mooring_moment[ii];
```

```
543
544     // Linear interpolation of OpenFOAM inputs
545     if ( ii > 0 && time_simu.back() >= t_trigger)
546     {
547         // pressure
548         double ratio_x = (px[ ii ] - px[ ii - 1]) / (time_input[ ii ] - time_input[ ii -
                1]);
549         double ratio_y = (py[ ii ] - py[ ii - 1]) / (time_input[ ii ] - time_input[ ii -
                1]);
550         double ratio_z = (pz[ ii ] - pz[ ii - 1]) / (time_input[ ii ] - time_input[ ii -
                1]);
551         p_at_timestep.x = ratio_x*(time_simu.back() - time_input[ii - 1]) + px[ ii -
                1];
552         p_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii - 1]) + py[ ii -
                1];
553         p_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 1]) + pz[ ii -
                1];
554
555         ratio_x = (p_m_x[ii] - p_m_x[ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
556         ratio_y = (p_m_y[ii] - p_m_y[ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
557         ratio_z = (p_m_z[ii] - p_m_z[ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
558         p_m_at_timestep.x = ratio_x*(time_simu.back() - time_input[ii - 1]) +
                p_m_x[ii - 1];
559         p_m_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii - 1]) +
                p_m_y[ii - 1];
560         p_m_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 1]) +
                p_m_z[ii - 1];
561
562         // viscosity
563         ratio_x = (vx[ ii ] - vx[ ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
564         ratio_y = (vy[ ii ] - vy[ ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
565         ratio_z = (vz[ ii ] - vz[ ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
566         v_at_timestep.x = ratio_x*(time_simu.back() - time_input[ii - 1]) + vx[ ii -
                1];
567         v_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii - 1]) + vy[ ii -
                1];
568         v_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 1]) + vz[ ii -
                1];
569
570         ratio_x = (v_m_x[ii] - v_m_x[ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
571         ratio_y = (v_m_y[ii] - v_m_y[ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
572         ratio_z = (v_m_z[ii] - v_m_z[ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
573         v_m_at_timestep.x = ratio_x*(time_simu.back() - time_input[ii - 1]) +
                v_m_x[ii - 1];
574         v_m_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii - 1]) +
                v_m_y[ii - 1];
```

```
575         v_m_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 1]) +
                v_m_z[ii - 1];
576
577         // 6Dof
578         ratio_x = (dofx[ ii ] - dofx[ ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
579         ratio_y = (dofy[ ii ] - dofy[ ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
580         ratio_z = (dofz[ ii ] - dofz[ ii - 1]) / (time_input[ ii ] - time_input[ ii - 1]);
581         dof_at_timestep.x = ratio_x*(time_simu.back() - time_input[ii - 1]) + dofx[ ii
                - 1];
582         dof_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii - 1]) + dofy[ ii
                - 1];
583         dof_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 1]) + dofz[ ii
                - 1];
584
585         // rot i.e. roll pitch yaw
586         ratio_x = ( dofroll [ ii ] - dofroll [ ii - 1]) / (time_input[ ii ] - time_input[ ii
                - 1]);
587         ratio_y = (dofpitch [ ii ] - dofpitch[ ii - 1]) / (time_input[ ii ] - time_input[ ii
                - 1]);
588         ratio_z = (dofyaw[ii ] - dofyaw[ii - 1]) / (time_input[ ii ] - time_input[ ii -
                1]);
589         rot_at_timestep.x = ratio_x*(time_simu.back() - time_input[ii - 1]) +
                dofroll [ ii - 1];
590         rot_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii - 1]) +
                dofpitch [ ii - 1];
591         rot_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 1]) +
                dofyaw[ii - 1];
592
593         // Mooring Load
594         ratio_z = (mooring_load[ii-1].z - mooring_load[ii - 2].z) / (time_input[ ii -1]
                - time_input[ ii - 2]);
595         mooring_load_at_timestep.z = ratio_z*(time_simu.back() - time_input[ii - 2])
                + mooring_load[ii - 2].z;
596
597         // Mooring Moment
598         ratio_y = (mooring_moment[ii].y - mooring_moment[ii - 1].y) / (time_input[ii]
                - time_input[ ii - 1]);
599         mooring_moment_at_timestep.y = ratio_y*(time_simu.back() - time_input[ii -
                1]) + mooring_moment[ii - 1].y;
600
601     }
602
603     GHExternalLoads::DOF3 excitation_at_timestep;
604     excitation_at_timestep.x = excitation_x[ ii ];
605     excitation_at_timestep.y = excitation_y[ ii ];
606     excitation_at_timestep.z = excitation_z[ ii ];
607
```

```
608     GHExternalLoads::DOF3 excitation_rot_at_timestep;
609     excitation_rot_at_timestep.x = excitation_roll [ ii ];
610     excitation_rot_at_timestep.y = excitation_pitch [ ii ];
611     excitation_rot_at_timestep.z = excitation_yaw[ ii ];
612
613     GHExternalLoads::DOF3 radiation_at_timestep;
614     radiation_at_timestep.x = radiation_x[ ii ];
615     radiation_at_timestep.y = radiation_y[ ii ];
616     radiation_at_timestep.z = radiation_z[ ii ];
617
618     GHExternalLoads::DOF3 radiation_rot_at_timestep;
619     radiation_rot_at_timestep.x = radiation_roll [ ii ];
620     radiation_rot_at_timestep.y = radiation_pitch [ ii ];
621     radiation_rot_at_timestep.z = radiation_yaw[ ii ];
622
623     GHExternalLoads::DOF3 ma_at_timestep;
624     ma_at_timestep.x = ma_wdy_x[ii];
625     ma_at_timestep.y = ma_wdy_y[ii];
626     ma_at_timestep.z = ma_wdy_z[ii];
627
628     GHExternalLoads::DOF3 ma_rot_at_timestep;
629     ma_rot_at_timestep.x = ma_wdy_roll[ii];
630     ma_rot_at_timestep.y = ma_wdy_pitch[ii];
631     ma_rot_at_timestep.z = ma_wdy_yaw[ii];
632
633
634     if ( ii > 0 && time_simu.back() < t_trigger)
635     {
636         // excitation
637         double ratio_x = (excitation_x[ ii ] - excitation_x[ ii - 1]) /
                (time_input_wdy[ii] - time_input_wdy[ii - 1]);
638         double ratio_y = (excitation_y[ ii ] - excitation_y[ ii - 1]) /
                (time_input_wdy[ii] - time_input_wdy[ii - 1]);
639         double ratio_z = (excitation_z[ ii ] - excitation_z[ ii - 1]) /
                (time_input_wdy[ii] - time_input_wdy[ii - 1]);
640         excitation_at_timestep.x = ratio_x*(time_simu.back() - time_input_wdy[ii -
                1]) + excitation_x[ ii - 1];
641         excitation_at_timestep.y = ratio_y*(time_simu.back() - time_input_wdy[ii -
                1]) + excitation_y[ ii - 1];
642         excitation_at_timestep.z = ratio_z*(time_simu.back() - time_input_wdy[ii -
                1]) + excitation_z[ ii - 1];
643
644         ratio_x = ( excitation_roll [ ii ] - excitation_roll [ ii - 1]) /
                (time_input_wdy[ii] - time_input_wdy[ii - 1]);
645         ratio_y = ( excitation_pitch [ ii ] - excitation_pitch [ ii - 1]) /
                (time_input_wdy[ii] - time_input_wdy[ii - 1]);
```

```
646         ratio_z = (excitation_yaw[ ii ] - excitation_yaw[ ii - 1]) / (time_input_wdy[ii]
                - time_input_wdy[ii - 1]);
647         excitation_rot_at_timestep.x = ratio_x*(time_simu.back() - time_input_wdy[ii
                - 1]) + excitation_roll [ ii - 1];
648         excitation_rot_at_timestep.y = ratio_y*(time_simu.back() - time_input_wdy[ii
                - 1]) + excitation_pitch [ ii - 1];
649         excitation_rot_at_timestep.z = ratio_z*(time_simu.back() - time_input_wdy[ii
                - 1]) + excitation_yaw[ ii - 1];
650
651         // radiation
652         ratio_x = (radiation_x[ ii ] - radiation_x[ ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
653         ratio_y = (radiation_y[ ii ] - radiation_y[ ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
654         ratio_z = (radiation_z[ ii ] - radiation_z[ ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
655         radiation_at_timestep.x = ratio_x*(time_simu.back() - time_input_wdy[ii - 1])
                + radiation_x[ ii - 1];
656         radiation_at_timestep.y = ratio_y*(time_simu.back() - time_input_wdy[ii - 1])
                + radiation_y[ ii - 1];
657         radiation_at_timestep.z = ratio_z*(time_simu.back() - time_input_wdy[ii - 1])
                + radiation_z[ ii - 1];
658
659         ratio_x = ( radiation_roll [ ii ] - radiation_roll [ ii - 1]) / (time_input_wdy[ii]
                - time_input_wdy[ii - 1]);
660         ratio_y = ( radiation_pitch [ ii ] - radiation_pitch [ ii - 1]) /
                (time_input_wdy[ii] - time_input_wdy[ii - 1]);
661         ratio_z = (radiation_yaw[ ii ] - radiation_yaw[ ii - 1]) / (time_input_wdy[ii]
                - time_input_wdy[ii - 1]);
662         radiation_rot_at_timestep.x = ratio_x*(time_simu.back() - time_input_wdy[ii
                - 1]) + radiation_roll [ ii - 1];
663         radiation_rot_at_timestep.y = ratio_y*(time_simu.back() - time_input_wdy[ii
                - 1]) + radiation_pitch [ ii - 1];
664         radiation_rot_at_timestep.z = ratio_z*(time_simu.back() - time_input_wdy[ii
                - 1]) + radiation_yaw[ ii - 1];
665
666         // acceleration
667         ratio_x = (ma_wdy_x[ii] - ma_wdy_x[ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
668         ratio_y = (ma_wdy_y[ii] - ma_wdy_y[ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
669         ratio_z = (ma_wdy_z[ii] - ma_wdy_z[ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
670         ma_at_timestep.x = ratio_x*(time_simu.back() - time_input_wdy[ii - 1]) +
                ma_wdy_x[ii - 1];
671         ma_at_timestep.y = ratio_y*(time_simu.back() - time_input_wdy[ii - 1]) +
                ma_wdy_y[ii - 1];
```

```
672         ma_at_timestep.z = ratio_z*(time_simu.back() - time_input_wdy[ii - 1]) +
                ma_wdy_z[ii - 1];
673
674         // acceleration Rot
675         ratio_x = (ma_wdy_roll[ii] - ma_wdy_roll[ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
676         ratio_y = (ma_wdy_pitch[ii] - ma_wdy_pitch[ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
677         ratio_z = (ma_wdy_yaw[ii] - ma_wdy_yaw[ii - 1]) / (time_input_wdy[ii] -
                time_input_wdy[ii - 1]);
678         ma_rot_at_timestep.x = ratio_x*(time_simu.back() - time_input_wdy[ii - 1]) +
                ma_wdy_roll[ii - 1];
679         ma_rot_at_timestep.y = ratio_y*(time_simu.back() - time_input_wdy[ii - 1]) +
                ma_wdy_pitch[ii - 1];
680         ma_rot_at_timestep.z = ratio_z*(time_simu.back() - time_input_wdy[ii - 1]) +
                ma_wdy_yaw[ii - 1];
681     }
682
683     GHExternalLoads::DOF3 force;
684     GHExternalLoads::DOF3 moment;
685
686     if (time_simu.back() >= t_trigger)
687     {
688         // Definition of ELC loads
689         force.x = -(p_at_timestep.x + v_at_timestep.x) ;
690         force.y = 0;
691         force.z = (p_at_timestep.z + v_at_timestep.z) + hydro_stiffness [2][2] *
                (dof_at_timestep.z - (-0.133)) - mean_displaced_mass*gravity ;
692
693         // Conservation of Momentum
694         double sum_of_forces_wdy = hydro_stiffness[2][2] * (position.z - 2.667) -
                mean_displaced_mass*gravity + 66.3*(position.z - 2.667 + 0.306) -
                force.z + mass*gravity;
695         double error_sumofforces = sum_of_forces_opf[ii].z + sum_of_forces_wdy;
696         force.z = force.z - error_sumofforces;
697
698
699         moment.x = 0;
700         moment.y = - hydro_stiffness[3][3] * (rotation_at_swap.y) -
                hydro_stiffness [3][3] * (rot_at_timestep.y - dofpitch[1]) -
                (p_m_at_timestep.y + v_m_at_timestep.y -
                mooring_moment_at_timestep.y);
701         moment.z = 0;
702     }
703     else
704     {
705         //cout << "Applying WaveDyn previously run dat-set" << endl;
```

```
706         force.x = added_mass[0][0] * ma_at_timestep.x + added_mass[0][4] *
                ma_rot_at_timestep.y + excitation_at_timestep.x - radiation_at_timestep.x;
707         force.y = 0;
708         force.z = -added_mass[2][2] * ma_at_timestep.z + excitation_at_timestep.z
                - radiation_at_timestep.z;
709
710         moment.x = 0;
711         moment.y = added_mass[4][4] * ma_rot_at_timestep.y + added_mass[4][0] *
                ma_at_timestep.x + excitation_rot_at_timestep.y -
                radiation_rot_at_timestep.y;
712         moment.z = 0;
713     }
714
715     ext_loads->GetLogValueByName("X force")->SetLoggingValue(force.x);
716     ext_loads->GetLogValueByName("Y force")->SetLoggingValue(force.y);
717     ext_loads->GetLogValueByName("Z force")->SetLoggingValue(force.z);
718
719     ext_loads->GetLogValueByName("X Moment")->SetLoggingValue(moment.x);
720     ext_loads->GetLogValueByName("Y Moment")->SetLoggingValue(moment.y);
721     ext_loads->GetLogValueByName("Z Moment")->SetLoggingValue(moment.z);
722
723     ext_loads->ApplyMultibodyNodeForce("C", force, "buoy");
724     /**< Applying a force to node 'C' [just to the side nearest component 'Flap']. */
725     ext_loads->ApplyMultibodyNodeTorque("C", moment);
726     /**< Applying a torque to node 'C' [next to the proximal connection]. */
727 }
728
729 /// <summary>StepComplete is called once the integrator has found a consistent
730 /// state for the model. This can be used to interrogate the model, but should
731 /// not be used for logging or applying loads</summary>
732
733 extern "C" void __declspec(dllexport) __cdecl
        StepComplete(GHExternalLoads::IExternalLoads* ext_loads)
734 {
735     /**< The multibody model can be interrogated. */
736     time_simu_complete.push_back(ext_loads->GetSimulationCurrentTime());
737     GHExternalLoads::DOF3 position = ext_loads->GetMultibodyNodePosition("C");
738     GHExternalLoads::DOF3 rotation =
            ext_loads->GetMultibodyNodeOrientation("C");
739
740     dof_simu_complete.push_back(position);
741     rot_simu_complete.push_back(rotation);
742
743     ofstream
            all6dof_input_for_opf("C:\\Users\\pmusiedlak\\Dropbox\\PhD\\all6dof_heavedecay_0",
            std::ofstream::app);
744
```

```
745     all6dof_input_for_opf << time_simu_complete.back() << "\t" << -position.x -
            position_rest.x << "\t" <<
746         -position.y - position_rest.y << "\t" <<
747         position.z - position_rest.z << "\t" <<
748         -rotation.x << "\t" << -rotation.y << "\t" << -rotation.z << endl;
749     all6dof_input_for_opf.close();
750 }
```

E.3.2 Coupling Allrun script

```
1  #!/bin/bash
2
3  bash Cleanall
4  cp -r 0.org 0
5  . $WM_PROJECT_DIR/bin/tools/RunFunctions
6
7  runApplication blockMesh
8  runApplication surfaceFeatureExtract
9  runApplication snappyHexMesh -overwrite
10
11 #paraFoam
12 while [ ! -f /home/pilou16/Dropbox/PhD/6_DoF ]
13 do
14     echo 'I am spleeping'
15     sleep 2
16 done
17
18 # need to comment the restraint in dynamicMeshDict
19 cp -r constant/dynamicMeshDict constant/dynamicMeshDict_restraint
20 sed -i '/restraints/,+12 d' constant/dynamicMeshDict # delete line where
       something is and the following 13
21 sed -i 's/rigidBodyMotionSolver/rigidBodyMeshDeform/g'
       constant/dynamicMeshDict # change something by something else after skipping
       number of lines
22
23
24 # the 6-Dof
25 dof="$(grep "Offsets:" /home/pilou16/Dropbox/PhD/6_DoF | cut -d ":" -f 2 | cut -d
       " " -f 2)"
26 dof="deformPosition 6 ("$dof");"
27 sed -i "/report/a $dof" constant/dynamicMeshDict
28 sed -i "/report/a deformIter 20" constant/dynamicMeshDict
29
30 foamDictionary system/controlDict -entry writeControl -set 'timeStep'
31 foamDictionary system/controlDict -entry writeInterval -set '1'
32 deformDyMMesh -overwrite &> log.deformDyMMesh
```

```
33
34 checkMesh -latestTime
35
36 # in order to startAt t_{hot}-t_{minus} (already considerate in the C++ script)
37 startAt="$(grep "Time:" /home/pilou16/Dropbox/PhD/6_DoF | cut -d ":" -f 2 | cut
       -d "s" -f 1 | sed -e 's/^\s *//') "
38 echo $startAt
39 mv 0 "$startAt"
40
41 setWaveParameters > log.setWaveParameters
42 setWaveField > log.setWaveField_afterDeform
43 #paraFoam
44
45 # Before running the case, forcing q,qDot,qDdot for X iterations is necessary:
46 foamDictionary system/controlDict -entry writeControl -set 'timeStep'
47 foamDictionary system/controlDict -entry writeInterval -set '1'
48 foamDictionary system/controlDict -entry stopAt -set 'nextWrite'
49 foamDictionary system/controlDict -entry endTime -set '20'
50 cp -r constant/dynamicMeshDict_restraint constant/dynamicMeshDict
51
52 # need to specify at least the qDot and qDdot in
       startAt/uniform/rigidBodyModelState
53 # i.e. forcingIterations = 0 , however recommanded is 5
54 declare -i forcingIterations
55  forcingIterations =5
56 for ((i =0 ; i <=$forcingIterations ; i++));
57 do
58     echo '/n i= ' $i
59     # last time-step
60     timeStep="$(ls -1 | sort --numeric | tail -1)"
61     echo $timeStep
62     # change q,qDot,qDdot in LastTime-step/uniform/rigidBodyMotionState
63     python script.py /home/pilou16/Dropbox/PhD/all6dof.txt $timeStep
64      if [ $i -gt 0 ];
65     then
66         echo 'waveDyMFoam - forcing Iteration = ' $i
67             waveDyMFoam &>> log.waveDyMFoam
68      fi
69 done
70
71 checkMesh -latestTime
72 #paraFoam
73
74 # To run the case:
75 foamDictionary system/controlDict -entry writeControl -set 'adjustableRunTime'
76 foamDictionary system/controlDict -entry writeInterval -set '0.1'
77 foamDictionary system/controlDict -entry stopAt -set 'endTime'
```

```
foamDictionary system/controlDict -entry endTime -set '20'

waveDyMFoam &>> log.waveDyMFoam

forcesExtract

extractData
cp -r results/t_vs_6DoF /home/piloust/Dropbox/PhD/dof_opf.txt

paraFoam
```

E.3.3 Python script

```
#!/usr/bin/python

# Python 3.6

# This script use the WaveDyn time-series as input
# find the closest time-values from the last time-step and interpolate
# author: Pierre-Henri MUSIEDLAK - PhD at Univeristy of Plymouth

import sys

print 'Number of arguments:', len(sys.argv), 'arguments.'
aa = sys.argv
print aa[1]
print aa[2]

from numpy import *
import numpy as np

# Last time-step
time = float(aa[2])
dof = np.loadtxt(aa[1], delimiter="\t")
dd = nonzero(dof[:,0]>time)
ind = min(dd[0])

interpolation = zeros((6,1))
q = zeros((6,1))
qDot = zeros((6,1))
qDdot = zeros((6,1))
for ii in range(6):
    print ii
    ratio = (dof[ind, ii+1]-dof[ind-1,ii+1])/(dof[ind,0]-dof[ind-1,0])
    q[ii] = ratio*(time-dof[ind-1,0]) + dof[ind-1,ii+1]
    qDot[ii] = ratio
```

```
34      ratio_2 = (dof[ind-1,ii+1]-dof[ind-2,ii+1])/(dof[ind-1,0]-dof[ind-2,0])
35      qDdot[ii] = (ratio - ratio_2)/(dof[ind,0]-dof[ind-1,0])
36
37  # Overwrite time-step/uniform/rigidBodyMotionState
38  filename = aa[2] + '/uniform/rigidBodyMotionState'
39   print  filename
40  rgbm = open(filename,'w')
41
42  rgbm.write('/*--------------------------------*- C++
        -*----------------------------------*\ \n')
43  rgbm.write('| =========                 |
                                                | \n')
44  rgbm.write('| \\\\      /  F ield         | OpenFOAM: The Open Source CFD
        Toolbox      | \n')
45  rgbm.write('|  \\\\    /   O peration     | Version:  4.1
                                        | \n')
46  rgbm.write('|   \\\\  /    A nd          | Web:      www.OpenFOAM.org
                          | \n')
47  rgbm.write('|    \\\\/     M anipulation  |
                                                | \n')
48  rgbm.write('\*---------------------------------------------------------------------------
        \n')
49  rgbm.write('FoamFile\n')
50  rgbm.write('{\n')
51  rgbm.write('    version      2.0;\n')
52  rgbm.write('    format       ascii ;\n')
53  rgbm.write('    class        dictionary ;\n')
54  rgbm.write('    location     0/uniform;\n')
55  rgbm.write('    object       rigidBodyMotionState;\n')
56  rgbm.write('}\n')
57  rgbm.write('// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        * * * //  \n')
58  rgbm.write('\n\n')
59
60  rgbm.write('q       6 (' + str(float(q[0])) + ' ' + str(float(q[1])) + ' ' +
        str(float(q[2])) + ' ' + str(float(q[3])) + ' ' + str(float(q[4])) + ' ' +
        str(float(q[5])) + ');\n\n' )
61  rgbm.write('qDot      6 (' + str(float(qDot[0])) + ' ' + str(float(qDot[1])) + ' '
        + str(float(qDot[2])) + ' ' + str(float(qDot[3])) + ' ' + str(float(qDot[4])) +
        ' ' + str(float(qDot[5])) + ' );\n\n' )
62  rgbm.write('qDdot     6 (' + str(float(qDdot[0])) + ' ' + str(float(qDdot[1])) + '
        ' + str(float(qDdot[2])) + ' ' + str(float(qDdot[3])) + ' ' +
        str(float(qDdot[4])) + ' ' + str(float(qDdot[5])) + ' );\n\n' )
63
64  rgbm.write('//
        ************************************************************************
        // \n')
```

```
65
66 rgbm.close()
```

E.3.4 forcesExtract script

```
1  #!/bin/sh
2
3  # remove the 3 headlines:
4   tail  -n +4 "postProcessing/forces/0/forces.dat" > out
5
6  sed -i 's/(// g' out
7  sed -i 's/) // g' out
8
9  awk '{ print  $1,$2,$3,$4,$11,$12,$13}' out >
       "/home/pilou16/Dropbox/PhD/pressure_st1_0.txt"
10 awk '{ print  $1,$5,$6,$7,$14,$15,$16}' out >
       "/home/pilou16/Dropbox/PhD/viscous_st1_0.txt"
11
12 gnuplot forcesPlot
```

List of references

Airy, G. B. (1845), 'Tides and waves', *In Encyclopaedia metropolitana* **5**, 241–396.

Anderson, J. J. D. (1995), *Computational Fluid Dynamics: The basics with applications*, McGraw-Hill International Editions.

Barstow, S., Mørk, G., Mollison, D. and Cruz, J. (2008), The wave energy resource, *in* 'Ocean wave energy: Current Status and Future Prespectives', Springer, Berlin, Heidelberg, pp. 93–132.

Bateman, W. J. D., Katsardi, V. and Swan, C. (2012), 'Extreme ocean waves. Part I. The practical application of fully nonlinear wave modelling', *Applied Ocean Research* **34**, 209 – 224.

BBC (2005). http://news.bbc.co.uk/1/hi/scotland/4563077.stm, Online accessed Mar-2019.

Biausser, B., Fraunié, P., Grilli, S. and Marcer, R. (2004), 'Numerical Analysis of the Internal Kinematics and Dynamics of Three-dimensional Breaking Waves on Slopes', *International Journal of Offshore and Polar Engineering* **14**, 247–256.

Bouws, E., Draper, L., Shearman, E., Laing, A., Feit, D., Mass, W., Eide, L., Francis, P., Carter, D. and Battjes, J. (1998), 'Guide to Wave analysis and forecasting', *World Meteorological Organization* (702).

Bredmose, H. and Jacobsen, N. G. (2010), Breaking wave impacts on offshore wind turbine foundations: focused wave groups and CFD, *in* '*Proceedings of the 29th International Conference on Ocean, Offshore and Arctic Engineering*', American Society of Mechanical Engineers, pp. 397–404.

Brown, S. A. (2017), Numerical modelling of turbulence and sediment concentrations under breaking waves using OpenFOAM®, PhD thesis, University of Plymouth.

Brown, S. A., Musiedlak, P.-H., Ransley, E. J. and Greaves, D. (2018), Numerical simulation of focused wave interactions with a fixed FPSO using OpenFOAM 4.1, *in* '*Proceedings of the 28th International Ocean and Polar Engineering Conference (ISOPE)*'. Sapporo, Japan, 10-15 June.

Brown, S., Greaves, D., Magar, V. and Conley, D. (2016), 'Evaluation of turbulence closure models under spilling and plunging breakers in the surf zone', *Coastal Engineering* **114**, 177–193.

Carnegie (2019). https://www.carnegiece.com/wave/, Online accessed Mar-2019.

CCP-WSI Code Repository (2018). A Collaborative Computational Project in Wave Structure Interaction.

CCP-WSI Working Group (2016), Wave Structure Interaction Computation and Experiment Roadmap - Part 1: A Report on the 1st CCP-WSI Focus Group Workshop, Technical report. Produced by the University of Plymouth.

Chaplin, J. R. et al. (1996), 'On frequency-focusing unidirectional waves', *International Journal of Offshore and Polar Engineering* **6**(02).

Chen, W., Dolguntseva, I., Savin, A., Zhang, Y., Li, W., Svensson, O. and Leijon, M. (2017), 'Numerical modelling of a point-absorbing wave energy converter in irregular and extreme waves', *Applied Ocean Research* **63**, 90–105.

Chen, Y., Wu, Y., Bahuguni, A., Gullman-Strand, J., Zhang, Y., Stewart, G., Lv, X. et al. (2016), Advances in Computational Hydrodynamics Applied to Wave-in-Deck, *in* '*Proceedings of the Offshore Technology Conference (OTC)*'. Asia, Kuala Lumpur, Malaysia 22-25 March.

Chorin, A. J. (1967), 'The numerical solution of the navier-stokes equations for an incompressible fluid', *Bulletin of the American Mathematical Society* **73**(6), 928–931.

COAST (2019), 'Coastal, Ocean and Sediment Transport'. Online accessed Mar-2019.

Coe, R. G. and Neary, V. S. (2014), Review of methods for modeling wave energy converter survival in extreme sea states, *in* '*Proceedings of the 2nd Marine Energy Technology Symposium (METS)*'. Seattle, WA, April 2014.

Coe, R., Yu, Y.-H. and Van Rij, J. (2018), 'A survey of wec reliability, survival and design practices', *Energies* **11**(1), 4.

Crespo, A. J., Domínguez, J. M., Rogers, B. D., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A. and García-Feal, O. (2015), 'DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH)', *Computer Physics Communications* **187**, 204–216.

Cummins, W. E. (1962), The impulse response function and ship motions, Technical report, David Taylor Model Basin Washington DC. (No. DTMB-1661).

Davidson, J., Cathelain, M., Guillemet, L., Le Huec, T. and Ringwood, J. (2015), Implementation of an openfoam numerical wave tank for wave energy experiments, *in* '*Proceedings of the 11th European wave and tidal energy conference (EWTEC)*'. Nantes, France, 6-11th September.

de Lataillade, T., Dimakopoulos, A., Kees, C., Johanning, L., Ingram, D. and Tezdogan, T. (2017), CFD Modelling coupled with Floating Structures and Mooring Dynamics for Offshore Renewable Energy Devices using the Proteus Simulation Toolkit, *in* '*Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*'. Cork, Ireland, 27th August - 1st September, 2017.

Delhommeau, G. (1987), Les problèmes de diffraction-radiation et de résistance de vagues: étude théorique et résolution numérique par la méthode des singularités, PhD thesis, Ecole Nationale Supérieure de Mécanique de Nantes, France.

Dingemans, M. W. (1997), *Water Wave Propagation Over Uneven Bottoms*, World Scientific Publishing Company.

DNV-GL (2013), 'WaveDyn Theory Manual'. Available within WaveDyn installation.

Dunbar, A. J., Craven, B. A. and Paterson, E. G. (2015), 'Development and validation of a tightly coupled cfd/6-dof solver for simulating floating offshore wind turbine platforms', *Ocean Engineering* **110**, 98–105.

Ellabban, O., Abu-Rub, H. and Blaabjerg, F. (2014), 'Renewable energy resources: Current status, future prospects and their enabling technology', *Renewable and Sustainable Energy Reviews* **39**, 748–764.

Engsig-Karup, A., Eskilsson, C. and Bigoni, D. (2016), 'A Stabilised Nodal Spectral Element Method for Fully Nonlinear Water Waves', *Journal of Computational Physics* **318**.

Engsig-Karup, A. P., Bingham, H. B. and Lindberg, O. (2009), 'An efficient flexible-order model for 3D nonlinear water waves', *Journal of computational physics* **228**(6), 2100–2118.

Featherstone, R. (2014), *Rigid body dynamics algorithms*, Springer, Boston, MA, US.

Fotcluster2 (2019). Plymouth Univeristy's HPC, Online accessed Mar-2019.

Fourier, J. J. (1822), 'Théorie analytique de la chaleur, par m. fourier', **82**. Chez Firmin Didot, père et fils.

Galton, F. (1886), 'Regression towards mediocrity in hereditary stature', *The Journal of the Anthropological Institute of Great Britain and Ireland* **15**, 246–263.

Greaves, D. (2004), 'A quadtree adaptive method for simulating fluid flows with moving interfaces', *Journal of Computational Physics* **194**(1), 35–56.

Greaves, D. M. and Borthwick, A. (1999), 'Hierarchical tree-based finite element mesh generation', *International Journal for Numerical Methods in Engineering* **45**(4), 447–471.

Guanche, R. (2017), On the importance of calibration and validation procedures: hybrid modeling, *in* '*Proceedings of the BCAM Workshop Hydrodynamics of Wave Energy Converter*'.

Halcrow (2006), 'South West of England Regional Development Agency: Wave Hub development and design phase coastal processes study report', p. 19.

Hann, M., Greaves, D. and Raby, A. (2015), 'Snatch loading of a single taut moored floating wave energy converter due to focussed wave groups', *Ocean Engineering* **96**, 258–271.

Hann, M., Greaves, D. and Raby, A. C. (2014), A new set of focused wave linear combinations to extract non-linear wave harmonics, *in* '*Proceedings of the 29th International Workshop for Water Waves and Floating Bodies*', pp. 61–64.

Hann, M., Greaves, D., Raby, A. and Howey, B. (2018), 'Use of constrained focused waves to measure extreme loading of a taut moored floating wave energy converter', *Ocean Engineering* **148**, 33–42.

Heilskov, N. F., Petersen, O. and Fuchs, J. (2015), Structural design of wave energy converters - State-of-the-Art and implementation of design tools for floating wave energy converters, Technical report. Danish Hydraulic Institute.

Henry, A., Schmitt, P., Whittaker, T., Rafiee, A., Dias, F. et al. (2013), The characteristics of wave impacts on an oscillating wave surge converter, *in* '*Proceedings of the 23rd International Offshore and Polar Engineering Conference (ISOPE)*', International Society of Offshore and Polar Engineers. Anchorage, Alaska,30 June-5 July.

Higuera Caubilla, P. et al. (2015), Aplicación de la dinámica de fluidos computacional a la acción del oleaje sobre estructuras. Application of computational fluid dynamics to wave action on structures, PhD thesis, Universidad de Cantabria.

Higuera, P. (2019), 'phicau/olaFlow: CFD for waves (Version v1.1)'. Online accessed Mar-2019.

Higuera, P., Lara, J. L. and Losada, I. J. (2013), 'Realistic wave generation and active wave absorption for Navier–Stokes models: Application to OpenFOAM®', *Coastal Engineering* **71**, 102–118.

Hildebrandt, A., Sriram, V. and Schlurmann, T. (2013), Simulation of focusing waves and local line forces due to wave impacts on a tripod structure, International Society of Offshore and Polar Engineers, pp. 575–581. Anchorage, Alaska, 30 June-5 July.

Hines, J. (2008), 'A comparative study of the simple and fractional step time integration methods for transient incompressible flows'.

Hirt, C. W. and Nichols, B. D. (1981), 'Volume of fluid (VOF) method for the dynamics of free boundaries', *Journal of computational physics* **39**(1), 201–225.

Hu, C. and Kashiwagi, M. (2004), 'A cip-based method for numerical simulations of violent free-surface flows', *Journal of Marine Science and Technology* **9**(4), 143–157.

Hu, Z. Z., Greaves, D. and Raby, A. (2016), 'Numerical wave tank study of extreme waves and wave-structure interaction using OpenFoam®', *Ocean Engineering* **126**, 329–342.

Hunt-Raby, A. C., Borthwick, A. G., Stansby, P. K. and Taylor, P. H. (2011), 'Experimental measurement of focused wave group and solitary wave overtopping', *Journal of Hydraulic Research* **49**(4), 450–464.

International Electrotechnical Commission (2016), 'Marine Energy—Wave, Tidal and Other Water Current Converters—Part 2: Design Requirements for Marine Energy Systems', *IEC TS 62600-2:2016* . IEC: Geneva, Switzerland.

IPCC (2014), Climate Change 2014 : Synthesis Report. Contribution of Working

Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC), Technical report. IPCC, Geneva, Switzerland.

Jacobsen, N. (2017), waves2Foam Manual, Technical report.

Jacobsen, N. G., Fuhrman, D. R. and Fredsøe, J. (2012), 'A wave generation toolbox for the open-source CFD library: OpenFoam®', *International Journal for numerical methods in fluids* **70**(9), 1073–1088.

Janssen, C. F., Grilli, S. T. and Krafczyk, M. (n.d.), Modeling of Wave Breaking and Wave-Structure Interactions by Coupling of Fully Nonlinear Potential Flow and Lattice-Boltzmann Models, International Society of Offshore and Polar Engineers. Beiing, China, June 20-25.

Jasak, H. (1996), Error analysis and estimation for the finite volume method with applications to fluid flows, PhD thesis, Imperial College London (University of London).

Jasak, H., Vukčević, V. and Gatin, I. (2015), *Numerical Simulation of Wave Loading on Static Offshore Structures*, Springer International Publishing, Cham, pp. 95–105.

Kim, J. W., Jang, H., Baquet, A., O'Sullivan, J., Lee, S., Kim, B. and Jasak, H. (2016), Technical and Economic Readiness Review of CFD-Based Numerical Wave Basin for Offshore Floater Design, *in* '*Proceedings of the Offshore Technology Conference (OTC)*'.

Kim, S.-H., Yamashiro, M. and Yoshida, A. (2010), 'A simple two-way coupling method of BEM and VOF model for random wave calculations', *Coastal Engineering* **57**(11), 1018 – 1028.

Kumar, N. (2016), A rigidBodyDynamics tutorial with demonstrations, *in* 'In Proceedings of CFD with OpenSource Software'. Edited by Nilsson. H.

Lachaume, C., Biausser, B., Grilli, S., Fraunié, P. and Guignard, S. (2003), Modeling of Breaking and Post-breaking Waves on Slopes by Coupling of BEM and VOF Meth-

ods, International Society of Offshore and Polar Engineers. Honolulu, Hawau, USA, May 25-30.

Lawson, M., Yu, Y.-H., Nelessen, A., Ruehl, K. and Michelen, C. (2014), Implementing nonlinear buoyancy and excitation forces in the WEC-Sim wave energy converter modeling tool, *in* '*Proceedings of the 33rd International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*', Vol. 9B: Ocean Renewable Energy. San Francisco, CA, USA, 8-13 June 2014.

Le Boulluec, M., Ohana, J., Martin, A. and Houmard, A. (2013), Tank testing of a new concept of floating offshore wind turbine, *in* '*Proceedings of the 32nd international conference on Ocean, Offshore and Arctic Engineering (OMAE)*', Vol. 8: Ocean Renewable Energy, American Society of Mechanical Engineers. Nantes, France, 9-14 June.

Le Méhauté, B. (1976), *An introduction to hydrodynamics and water waves*, Springer-Verlag Berlin Heidelberg New York.

Lee, C. and Newman, J. (2013), 'WAMIT user manual, version 7.0', *WAMIT, Inc., Chestnut Hill, MA* .

Li, Q., Wang, J., Yan, S., Gong, J. and Ma, Q. (2018), 'A zonal hybrid approach coupling FNPT with OpenFOAM for modelling wave-structure interactions with action of current', *Ocean Systems Engineering* **8**(4), 381–407.

Lin, P. and Liu, P. L.-F. (1998), 'A numerical study of breaking waves in the surf zone', *Journal of fluid mechanics* **359**, 239–264.

Liu, X. and García, M. H. (2008), 'Three-dimensional numerical model with free water surface and mesh deformation for local sediment scour', *Journal of Waterway, Port, Coastal, and Ocean Engineering* **134**(4), 203–217.

Magagna, D. and Uihlein, A. (2015), 'Ocean energy development in Europe : Current status and future perspectives', *International Journal of Marine Energy* **11**, 84–104.

Nadaoka, K., Hino, M. and Koyano, Y. (1989), 'Structure of the turbulent flow field under breaking waves in the surf zone', *Journal of Fluid Mechanics* **204**, 359–387.

Ning, D., Zang, J., Liu, S., Taylor, R. E., Teng, B. and Taylor, P. (2009), 'Free-surface evolution and wave kinematics for nonlinear uni-directional focused wave groups', *Ocean Engineering* **36**(15-16), 1226–1243.

NORSOK (2007), 'Actions and Action Effects (N-003)'. Standards Norway: Oslo, Norway.

O'Neill, L., Fakas, E. and Cassidy, M. (2006), 'A methodology to simulate floating offshore operations using a design wave theory', *Journal of offshore mechanics and Arctic engineering* **128**(4), 304–313.

OpenFOAM C++ Source Code v4.1 (2019). https://cpp.openfoam.org/v4/, Online accessed Mar-2019.

OpenFOAM ESI (2019). https://www.openfoam.com/documentation/user-guide/fvSchemes.php#x23-870273 Online accessed Mar-2019.

OpenFOAM-WIKI (2019), 'The PIMPLE algorithm in OpenFOAM'. Online accessed Mar-2019.

Palm, J., Eskilsson, C., Paredes, G. M. and Bergdahl, L. (2016), 'Coupled mooring analysis for floating wave energy converters using cfd: Formulation and validation', *International Journal of Marine Energy* **16**, 83–99.

Pedersen, J. R., Eltard-Larsen, B., Bredmose, H. and Jasak, H. (2017), A new volume-of-fluid method in openfoam, *in* P. Q. M. Visonneau and D. L. touzé, eds, 'Proceedings of the VII International Conference on Computational Methods in Marine Engineering (MARINE)'. Nantes, France, 15-17 Mai 2017.

Pelamis (1998). http://www.emec.org.uk/about-us/wave-clients/pelamis-wave-power/, Online accessed Mar-2019.

Penalba, M., Giorgi, G. and Ringwood, J. V. (2017), 'Mathematical modelling of wave energy converters: a review of nonlinear approaches', *Renewable and Sustainable Energy Reviews* **78**, 1188–1207.

Peskin, C. S. (1972), 'Flow patterns around heart valves: a numerical method', *Journal of computational physics* **10**(2), 252–271.

Pico (2019). http://www.pico-owc.net/, Online accessed July-2019.

Pierson Jr, W. J. and Moskowitz, L. (1964), 'A proposed spectral form for fully developed wind seas based on the similarity theory of SA Kitaigorodskii', *Journal of geophysical research* **69**(24), 5181–5190.

ProjectCHRONO (2019). An Open Source Multi-physics Simulation Engine, Source code.

Proteus (2019). A Python toolkit for computational methods and simulation, Source code.

Qian, L., Causon, D., Mingham, C. and Ingram, D. (2006), 'A free-surface capturing method for two fluid flows with moving bodies', *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* **462**.

Rafiee, A. and Fiévez, J. (2015), Numerical prediction of extreme loads on the CETO wave energy converter, *in* '*Proceedings of the 11th European Wave and Tidal Energy Conference (EWTEC)*'. Nantes, France, 6-11th September, 2015.

Rafiee, A., Wolgamot, H., Draper, S., Orszaghova, J., Fiévez, J. and Sawyer, T. (2016), Identifying the design wave group for the extreme response of a point absorber wave energy converter, *in* '*Proceedings of the Asian Wave and Tidal Energy Conference (AWTEC)*'. Singapore, 24-28 October.

Ransley, E., Greaves, D., Raby, A., Simmonds, D. and Hann, M. (2017), 'Survivability of wave energy converters using CFD', *Renewable Energy* **109**, 235–247.

Ransley, E. J. (2015), Survivability of wave energy converter and mooring coupled system using CFD, PhD thesis.

Ransley, E., Yan, S., Brown, S., Mai, T., Graham, D., Ma, Q., Musiedlak, P., Engsig-Karup, E., Eskilsson, C., Li, Q., Wang, J., Xie, Z., Sriram, V., Stoesser, T., Zhuang, Y., Wan, D., Chen, G., Chen, H., Qian, L., Ma, Z., Mingham, C., Causon, D., Gatin, I., Jasak, H., Vukcevic, V., Downie, S., Higuera, P., Buldakov, E., Stagonas, D., Chen, Q., Zang, J. and Greaves, D. (2019), 'A blind comparative study of focused wave interactions with a fixed fpso-like structure (ccp-wsi blind test series 1)', *International Journal of Offshore and Polar Engineering (IJOPE)* .

Rapp, R. J. and Melville, W. K. (1990), 'Laboratory measurements of deep-water breaking waves', *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* **331**(1622), 735–800.

Renewable Economy (2014), 'CETO wave energy machine swept away in cyclone, report says, online accessed mar-2019'.

Rhie, C. and Chow, W. L. (1983), 'Numerical study of the turbulent flow past an airfoil with trailing edge separation', *AIAA journal* **21**(11), 1525–1532.

Ribes, A. and Caremoli, C. (2007), Salome platform component model for numerical simulation, *in* '*31st Annual International Computer Software and Applications Conference (COMPSAC)*', Vol. 2, pp. 553–564. Beijing, China.

Rusche, H. (2003), Computational fluid dynamics of dispersed two-phase flows at high phase fractions, PhD thesis, Imperial College London (University of London).

Schmitt, P. and Elsaesser, B. (2015), A review of wave makers for 3D numerical simulations, *in* '*Proceedings of the Marine 2015, 6th International Conference on Computational Methods in Marine Engineering*', pp. 15–17. Rome, Italy.

SETIS (2013), 'Ocean Energy: Technology Inforation Sheet', *European Commission, Strategic Energy Technologies Information System (SETIS)* .

Simonetti, I., Cappietti, L., El Safti, H., Manfrida, G., Matthies, H. and Oumeraci, H. (2015), The use of openfoam as a virtual laboratory to simulate oscillating water column wave energy converters, *in* '*Proceedings of the VI International Conference on Computational Methods in Marine Engineering*'.

Sjökvist, L., Wu, J., Ransley, E., Engström, J., Eriksson, M. and Göteman, M. (2017), 'Numerical models for the motion and forces of point-absorbing wave energy converters in extreme waves', *Ocean Engineering* **145**, 1–14.

Sriram, V., Ma, Q. and Schlurmann, T. (2014), 'A hybrid method for modelling two dimensional non-breaking and breaking waves', *Journal of computational physics* **272**, 429–454.

Stagonas, D., Higuera, P. and Buldakov, E. (2018), 'Simulating Breaking Focused Waves in CFD: Methodology for Controlled Generation of First and Second Order', *Journal of Waterway, Port, Coastal, and Ocean Engineering* **144**(2), 06017005.

Stokes, G. G. (1847), 'On the theory of oscillatory waves', *Transactions of the Cambridge Philosophical Society* **8**, 441–455.

Sumer, B. M. et al. (2006), *Hydrodynamics around cylindrical strucures*, Vol. 26, World scientific.

Taylor, P. H., Jonathan, P. and Harland, L. A. (1997), 'Time domain simulation of jack-up dynamics with the extremes of a Gaussian process', *Journal of vibration and acoustics* **119**(4), 624–628.

The OpenFOAM Foundation (2019). https://openfoam.org/, Online accessed Mar-2019.

Ting, F. C. and Kirby, J. T. (1994), 'Observation of undertow and turbulence in a laboratory surf zone', *Coastal Engineering* **24**(1-2), 51–80.

Tromans, P. S., Anaturk, A. R. and Hagemeijer, P. (1991), A new model for the kinematics of large ocean waves-application as a design wave, *in* '*Proceedings of the First*

International Offshore and Polar Engineering Conference (ISOPE)', International Society of Offshore and Polar Engineers, pp. 64–71. Edinburgh, UK, 11-16 August.

U.S. Army Corps of Engineers (2007), *Coastal Engineering Manual, Engineer Manual 1110-2-1100*, U.S. Army Corps of Engineers, Washington, D.C.

Verbrugghe, T., Domínguez, J. M., Crespo, A. J., Altomare, C., Stratigaki, V., Troch, P. and Kortenhaus, A. (2018), 'Coupling methodology for smoothed particle hydrodynamics modelling of non-linear wave-structure interactions', *Coastal Engineering* **138**, 184–198.

Versteeg, H. K. and Malalasekera, W. (2007), *An introduction to computational fluid dynamics: the finite volume method*, 2nd edn, Pearson Education Ltd.

Vyzikas, T., Greaves, D., Simmonds, D., Maisondieu, C., Smith, H. and Radford, L. (2014 - b), Application of numerical models and codes, Technical report, Marine Energy in Far Peripheral and Island Communities (MERiFIC).

Vyzikas, T., Ransley, E., Hann, M., Magagna, D., Greaves, D., Simmonds, D., Magar, V. and Conley, D. (2013), Integrated numerical modelling system for extreme wave events at the wave hub site, *in* '*Proceedings of ICE Conference on Coasts, Marine Structures and Breakwaters*'.

Vyzikas, T., Stagonas, D., Buldakov, E. and Greaves, D. (2014), On the simulation of focused waves with openfoam & waves2foam, *in* '*Proceedins of Coastlab14: 5th Int Conf on the Application of Physical Modelling to Port and Coastal Protection,* Varna, Bulgaria', Vol. 2, pp. 237–282.

Wadhah, S., Kaiss, A. and Tadrist, L. (2004), Calculation of unsteady incompressible free surface flows by a fractional-step finite method, *in* '*Proceedings of the International Conference on advances in mechanical engineering (ICAME-2004)*'.

Walker, D. A. G., Taylor, P. H. and Taylor, R. E. (2004), 'The shape of large surface

waves on the open sea and the Draupner New Year wave', *Applied Ocean Research* **26**(3–4), 73 – 83.

Wang, J., Ma, Q. and Yan, S. (2016), 'A hybrid model for simulating rogue waves in random seas on a large temporal and spatial scale', *Journal of Computational Physics* **313**, 279–309.

Wang, J., Ma, Q. and Yan, S. (2017), 'On quantitative errors of two simplified unsteady models for simulating unidirectional nonlinear random waves on large scale in deep sea', *Physics of Fluids* **29**(6).

Wave Hub (2019). http://www.wavehub.co.uk, Online accessed Mar-2019.

WaveNet (2019). http://wavenet.cefas.co.uk/Map, Online accessed Mar-2019.

WaveRoller (2019). http://aw-energy.com/, Online accessed Mar-2019.

waves2foam Code Repository (2019). waves2foam Library.

Wei, Y., Abadie, T., Henry, A. and Dias, F. (2016), 'Wave interaction with an oscillating wave surge converter. Part II: Slamming', *Ocean Engineering* **113**, 319–334.

Westphalen, J., Greaves, D. M., Williams, C. K., Taylor, P. H., Causon, D. M., Mingham, C. G., Hu, Z. Z., Stansby, P. K., Rogers, B. D. and Omidvar, P. (2009), Extreme Wave Loading on Offshore Wave Energy Devices using CFD : a Hierarchical Team Approach, *in* '*Proceedings of the 8th European Wave and Tidal Energy Conference (EWTEC)*'. Uppsala, Sweden.

Windt, C., Davidson, J. and Ringwood, J. V. (2018), 'High-fidelity numerical modelling of ocean wave energy systems: A review of computational fluid dynamics-based numerical wave tanks', *Renewable and Sustainable Energy Reviews* **93**, 610–630.

Yabe, T. and Aoki, T. (1991), 'A universal solver for hyperbolic equations by cubic-polynomial interpolation i. one-dimensional solver', *Computer physics communications* **66**(2-3), 219–232.

Yan, S. (2006), 'Numerical simulation of nonlinear response of moored floating structures to steep waves'.

YAN, S., YAMASHIRO, M., YOSHIDA, A., AYUKAWA, K. and IRIE, I. (2003), 'DEVELOPMENT OF A COMBINED METHOD OF BEM AND VOF FOR ANALYZING WAVE MOTION', *PROCEEDINGS OF CIVIL ENGINEERING IN THE OCEAN* **19**, 83–88.

Zervos, A., Lins, C. and Muth, J. (2010), *Re-thinking 2050: A 100% Renewable Energy Vision for the European Union*, European Renewable Energy Council (EREC).

Zhao, X.-z., Hu, C.-h. and Sun, Z.-c. (2010), 'Numerical simulation of extreme wave generation using vof method', *Journal of hydrodynamics, ser. B* **22**(4), 466–477.

Bound copies of published papers

Investigation of Model Validity for Numerical Survivability Testing of WECs

Pierre-Henri Musiedlak
School of Engineering, Plymouth University
Reynolds Building, Drakes Circus,
PL4 8AA Plymouth, United Kingdom
pierre-henri.musiedlak@plymouth.ac.uk

Edward Jack Ransley, Deborah Greaves,
Martyn Hann, Gregorio Iglesias
School of Engineering, Plymouth University
PL4 8AA Plymouth, United Kingdom
edward.ransley@plymouth.ac.uk

Benjamin Child
DNV-GL,
One Linear Park, Avon Street,
Bristol BS2 0PS, United kingdom
benjamin.child@dnvgl.com

***Abstract*—This paper investigates the applicability of two numerical models to assess the survivability of Wave Energy Converters (WECs). Simulations using both a fully nonlinear Navier-Stokes solver (based on OpenFOAM) and WaveDyn (a linear time-domain model for multi-body interactions) are compared with physical experiments involving a free-floating buoy with a single mooring line. Events in which survivability is a concern are modelled using the focus wave-group NewWave. Two wave-groups (one steeper than the other) are used to identify the validity of each numerical model as a function of wave steepness. By taking into account the CPU cost and model validity, the range of applicability for both models is discussed. This constitutes the first step in future work: coupling the two numerical models to form an efficient modelling tool that benefits from the computational efficiency of WaveDyn while including the fidelity of a Navier-Stokes solver when required; therefore providing valuable information for WEC developers.**

***Index Terms*—Floating body, OpenFOAM, NewWave, experimental validation, wave steepness**

I. Introduction

Wave Energy Converter (WEC) developers consider reliability and survivability as key challenges in the design of their device. Existing research and standards concerning wave-structure-interaction (from oil and gas or offshore wind industries) seem unadapted to WEC design. Specifically, in the case of a point-absorber-type device, the structure cannot be considered to be fixed and, unlike traditional floating structures, the motion must not only be controlled to avoid damage but accentuated to generate power [1].

Present survivability design processes are based upon extremes, typically represented by single extreme wave events. Despite the characterization of these events being crucial, several mechanisms for their generation have been proposed (e.g. dispersive focusing or superposition) and a consensus on the description of an extreme event has yet to be found [2]. Furthermore, the peak loads on a WEC are not always the result of an extreme event, but can occur as a consequence of a particular series of smaller waves or, due to the motion history of the device [3].

Both numerical and physical modelling are widely used across engineering design [4]. The reliability of physical models is well established, and presently, the design and optimisation of WECs relies heavily upon them [5]. However, tank testing and physical experiments can be expensive and are typically limited to small scales (especially in the case of survivability studies). Numerical modelling is becoming increasingly important in the development of the offshore industry and WEC systems, where CFD-based Numerical Wave Tanks (NWT) have started to be recognized as design tools for survivability studies [6]. However, although a large number of design methods, and models, exist (with a wide range of fidelity), the limits and capacities of each are still unknown making selection of an appropriate model unclear [2].

Assessing the validity of a numerical code improves its reliability, as it defines a range of simulations and representations where the model can be used, and provides developers with certification for their models. 'Application of numerical models and codes' [7] classifies codes typically used in WEC development by physical process and the code capacity to accurately represent it by discerning a mark. However, precise measurements of code accuracy (or inaccuracies), using parametric criteria representing wave non-linearities - such as wave-steepness - instead of case-specific ones, are lacking. Also, WEC developers wish to perform accurate simulations with the least amount of CPU work [2]. Therefore, defining a code range of use improves efficiency as expensive codes will only be used to undertake survival testing, for example, whereas cheaper models will be used in more sedate cases, such as operational conditions. Also to ascertain the limit of use between those two numerical models, their validity is assessed against physical reference, with experiments that are representative of survival conditions.

The aim of this work is, therefore, to identify the validity of two numerical models, with different underlying physics, as a function of the wave steepness. The two software packages under investigation are:

- WaveDyn – a linear time-domain model for multi-body dynamics developed by DNV-GL [8], and;
- OpenFOAM - an open-source fully non-linear Computational Fluid Dynamics (CFD) code.

For each code, a numerical mirror of the Ocean Basin in the COAST Laboratory at Plymouth University [9] is generated. Simulations are performed to reproduce physical experiments involving the interaction of survival conditions - a focused

wave event - with a simplified WEC system consisting of a floating buoy and a single taut mooring [3]. The accuracy and speed of the simulations are then discuss to identify the applicability of each numerical model as a function of wave steepness. This research is part of an overall project aiming to couple both numerical models, to provide an efficient numerical tool. It will take advantage of WaveDyn low computational cost to solve any cases, while being able to swap to OpenFOAM at any instant to assess local survivability event.

II. Reference models

As part of an engineering design study for survivability, this study uses several models: a wave-model or design-wave, NewWave; a physical model, wave-tank plus buoy; and two numericals models, CFD with OpenFOAM and linear time-domain with WaveDyn.

A. Wave-model: NewWave

This work is based on the generation of a focused wave group using NewWave theory. Introduced by [10], NewWave theory produces, for a given sea state, the average shape of the highest wave with a specified exceedance probability [11]. It is often used as a design wave across marine sector in both physical and numerical analysis: [12] compared loads using NewWave description with on-site measurement of a North Sea oil platform; [13] used NewWave to study over-topping of embankments; and in the WEC sector, [5] identified their design wave as similar to a NewWave one, and used this description for numerical simulations.

At first order the surface elevation η of the generated focused wave is given by the addition of each wave component [14]:

$$\eta = \eta^{(1)} = \sum_{i=1}^{N} a_i cos[k_i(x - x_0) - \omega_i(t - t_0) + \epsilon_i] \quad (1)$$

where a_i, k_i, ω_i and ϵ_i are the amplitude, wave number, wave frequency and phase of the i^{th} component respectively. N is the total number of wave components. Using NewWave theory, the amplitude of each wave component, a_i, are defined according to the spectral energy $S(\omega)$, and the amplitude A of the main crest of the generated NewWave,

$$a_i = A\frac{S(\omega_i)\Delta\omega}{\sum_{p=1}^{N} S(\omega_p)\Delta f} \quad (2)$$

where A is defined using the zeroth moment of the spectrum $m_0 = (H_s/4)^2$, giving:

$$A = \sqrt{2m_0 ln(N)} \quad (3)$$

An example of a NewWave wave focused group at the focus location, generated using a Pierson-Moskowitz spectrum, can be found in Figure 1.

The wave steepness is characterized by kA, where the wave-number k corresponds to the peak frequency of the resulting wave groups spectrum assuming linear theory [3].

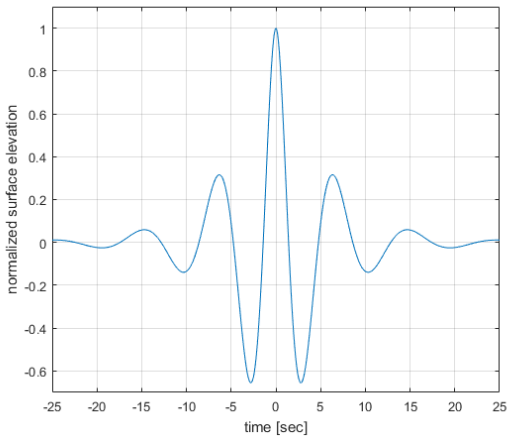


Fig. 1. Theoretical NewWave at focus location, generated by a Pierson-Moskowitz spectrum

B. Physical Model: OCEAN wave-tank

This study uses the scale model of a 'generic point-absorber WEC' realised during the EPSRC X-MED project [3]. The absence of a PTO system removes more complexity, assures the model to be as generic as possible, and can also be considered as a WEC in survivability mode (PTO is off), therefore making this research meaningful to a wide panel of WECs. The overall validity of this study is assured by those simplifications, as attention if focus on validity of the two numerical model to represent motion. Also it constitutes a first step in an incremental investigation, where models can be made more complex in the future. The simple mooring line assures proportionality between motions and moorings loads; hence the second ones will be not be represented here.

The model consisted of a 0.5m diameter hemispherical with 0.25m high cylinder on top. The total dry mass of the model is 43.2kg. It is moored to the wave-tank ground using an universal joint to assure multi-directional movement. The mooring line consists of the succession, from top to bottom, of a 35kN/m stiff rope, with a 66.3N/m stiff spring, and with a load cell. It connects the model bottom to the universal joint. At resting position (a representation can be found in Figure 2), the spring is extended by 0.27m.

Tests were conducted in the 35mx15.5mx2.8m Ocean Basin at Plymouth University's COAST laboratory. 11 probes measure the surface-elevation at 128Hz, upstream of the model. An optical tracking system was used to record the 6 degree-of-freedom motion of the model, Figure 2.

C. Numerical Models

Numerical models are used extensively throughout the wave energy sector, for almost every step in the development of WECs, from engineering to finance. Each can be very specific to a certain task or element of the WEC system. In the engineering design, for example, the estimation of mechanical loads on the structure alone, a wide panel of models is available for WEC developers. Capacities, limitations and

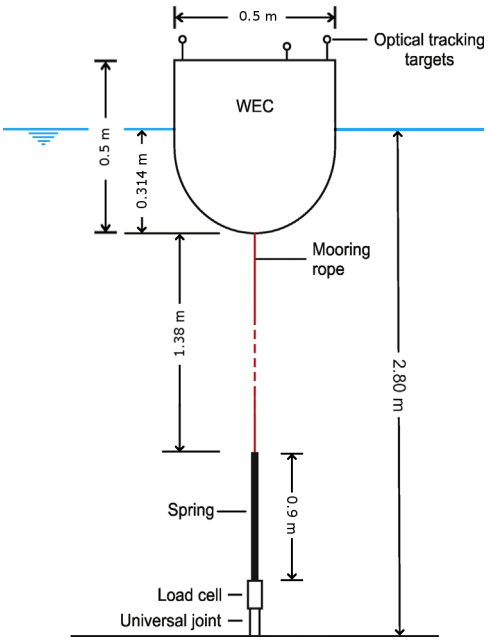


Fig. 2. X-MED model set-up and instrumentations

fidelity of those models are mostly unknown, even if the report realised through MERiFIC project started this investigation [7], however of great importance according to WEC developers [2].

Even if physical models are still required in the near future for WEC design [5], numerical models remain the solution to extrapolate and interpolate physical testing results, to full scale models or non-executable experiments (e.g. multi-directional waves, waves with different currents...).

1) OpenFOAM - Numerical Wave Tank:

OpenFOAM is an open-source CFD code gaining popularity due to its range of applications, its possibility of being modified easily as it is written in the object oriented programming language C++, its active community, and of course its absence of licence fees. Also, some solvers solve the Navier-Stokes equations using the Finite Volume Method, which is the well-established technique also used by main commercial CFD codes [15]. This makes it attractive for both university research and industries. A well-made description of OpenFOAM solving WEC fluid structure interaction problem was realised for previous EWTEC conference [16], where the realisation of a NWT for WEC is explained in much details.

In this study, OpenFOAM (version 4.1) solves the Reynolds-Averaged Navier-Stokes (RANS) equations for two incompressible, isothermal, immiscible fluids (water and air) [17]. It uses a Volume Of Fluid (VoF) based method [15], to capture the interface. The movement of a solid body and its resulting mesh deformation is calculated using the internal libraries *rigidBodyDynamics* and *rigidBodyMeshMotion* respectively. These apply an interpolation of movement as function of distance to the object surface [18], using translation vector and rotation quaternion. The wave-generation toolbox *waves2Foam* [19] is adapted to the solver (*waveDyMFOAM*).

Based on its physical reference one, COAST, and in a similar manner, a Numerical Wave Tank (NWT) is realised and made of three regions (a schematic representation can be found in Figure 3):

- 1: the wave-maker: an extended boundary generating waves in the left-to-right direction, and absorbing in the right-to-left direction
- 2: the working-section: ruled by the solver
- 3: the relaxation-zone or beach: an extended boundary absorbing waves in the left-to-right direction

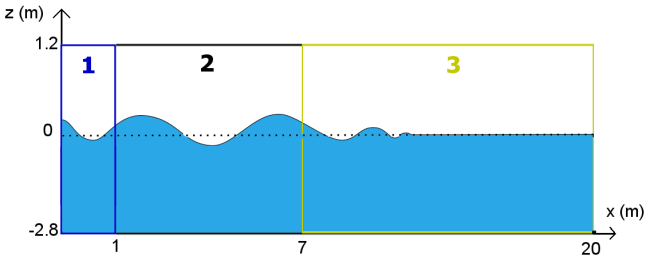


Fig. 3. The NWT schematic representation

The NWT uses a probe surface-elevation time-series as input for its wave-maker (1), by superposing linear wave components (Stokes 1^{st}) obtained by a FFT decomposition of this signal. Then, the wave group spreads through the working-section (2) according to the solver. The beach (3) absorbs the incoming wave by applying a gradually increasing damping function.

Issues relative to the realisation of a non-case specific, in term of domain geometry, 2D-NWT with the ability to generate and absorb different wave conditions, such as NewWave focus wave-group are: reflected waves, grid refinement according to wave-height, laboratory experiments comparisons, and CPU cost [20]. A NWT with a non-specific domain geometry allows the user to run several tests in a similar manner than a physical one. Also, the best representation of the underlying physics was found using a fully squared cell grid [1]. As based on the COAST one, the working-section is set up to the distance from the model maximum surge to the first probe, 6m. A compromise on previous issues, results in a 20m long NWT with a 13m long beach, and with a resolution of at least three cells per wave-height.

Previous NWT is expanded to three dimensions, and the model is included in the mesh within a movement adaptive mesh area. Also comparatively to 2D simulations, 3D ones are computationally expensive. Square cells are conserved, but the mesh is now refined around the mean-water area, [-0.5m,0.5m], to save CPU, and on the model surface to improve its resolution. In order to represent COAST specifics behaviours, such as wave reflection from the side walls, the NWT wide is set to COAST one (15.5m) without the use of relaxation-zone on sides. Using the heave decay-test, explained

later, a grid convergence study is realised, resulting in a 3 millions cells grid for the 20mx15.5mx2.8m (a cut along the length of this mesh can be seen on Figure 4).

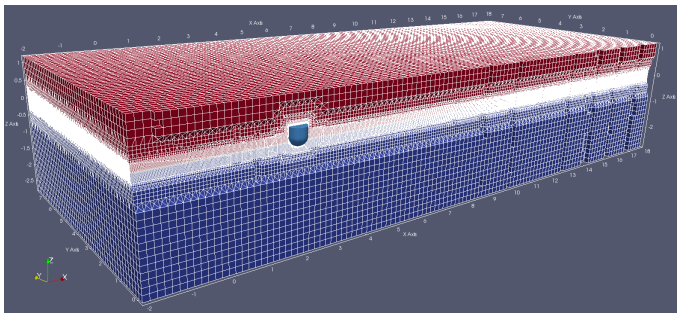


Fig. 4. OpenFOAM half-grid view – water is in blue and the wave spreads from left to right

The buoy model is defined by its mass, its centre of gravity, and its inertia matrix. The mooring line is represented by a 2.486m long spring with a 66.3N/m stiffness (rope stiffness influence is neglected).

2) WaveDyn:

WaveDyn is a performance and loading calculations tool for a range of WEC, developed by DNV-GL in Bristol [8]. It allows simulations of single or arrays of WEC. A device is constructed using a representative model made of specific bodies linked together with mass-less rigid links and adjustable joints.

Wave-structure interaction (WSI), in WaveDyn, is based on the Boundary Element Method (BEM). Each component is assigned with hydrodynamics properties, coming from a flow solver (AQWA or WAMIT). Body kinematics computes diffraction, radiation and buoyancy forces. This approach as a multi-body arrangement of Cummins equation, [21]:

$$(m_m + m_r(\infty))\ddot{x}(t) + f_{hs}(t) + \int_{-\infty}^{t} k(t-\tau)\dot{x}(\tau)\partial\tau = f_e(t) + f_{ext}(...) \quad (4)$$

Where x is the body displacement from its equilibrium position, m_m is the physical body mass, $m_r(\infty)$ is the theoretical added mass due to radiation force at infinite wave frequency, $f_{hs}(x)$ is the buoyancy force, the convolution-integral is the radiation force where $k(t)$ is the body impulse response function, $f_e(t)$ is the excitation force due to incident waves, and $f_{ext}(...)$ represents all additional non-hydrodynamics applied forces such as those due to moorings or Power-Take-Off (PTO).

X-MED WaveDyn model, Figure 5, is composed of three bodies (from seabed to the buoy):

- Seabed, the fixed datum,
- A slider - green diamond - representing the mooring line,
- the Buoy, where hydrodynamics forces comes from a WAMIT solver.

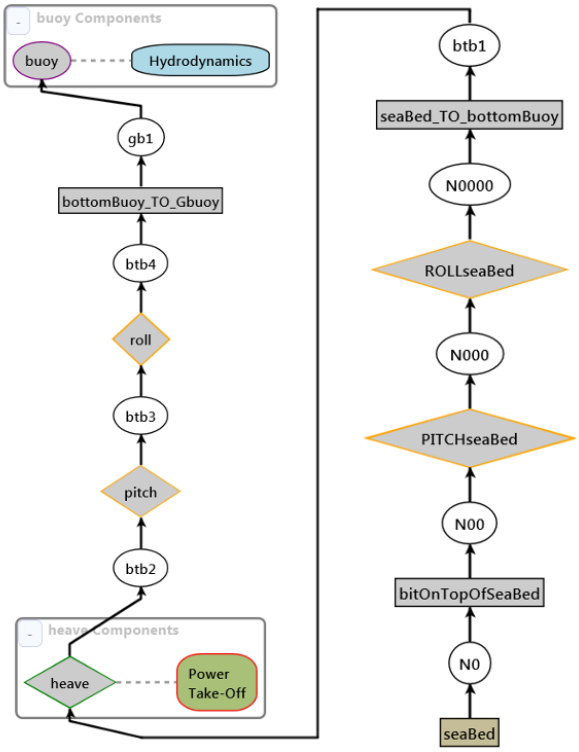


Fig. 5. X-MED WaveDyn model

Bodies are connected to each other with hinges - yellow diamonds -, and mass-less rigid links to represent the distance between two.

A wave spectrum described the decomposition into linear components of a sea-state using the probe at model location.

III. Test Plan

A heave decay test was performed in which the buoy was released from 0.204m from its resting moored position and the resonance frequency measured as 0.93Hz. This test is used for initial validation of the two numerical models.

Interaction of the taught-moored buoy in focused waves was then investigated. This study uses a Pierson-Moskowitz spectrum from a 100 year storm using hindcast data from the Wave Hub site ($T_z = 14.1$ s , $H_s = 14.4$ m, [22] p19). With this spectrum, a NewWave wave is defined, and generated at 50th scale with 1000 waves - a 3h sea-state [13]. Using Eqn.3, the largest crest amplitude is: $A = 0.267m$; and the first order wave components are found using Eqn.1 and in accordance to the COAST range of waves generation. This wave-group is defined as the reference case.

In order to assess the effect of wave-steepness on WEC movement, a steeper wave-group was created by increasing the reference peak frequency with a 1.09 proportional factor. This technique avoid the extra heave motion due to a steeper wave obtained by another technique which consists in increasing the crest amplitude while fixing the spectrum peak frequency [3].

But, please note, that the second wave-group can no longer be considered as a NewWave group.

Non-linear wave effects tend to shift the focus location [23] from its theoretical position. So a trial and error process was used during the experiments in order to focus wave groups where required. Waves groups were repeated three times to assess repeatability, where steepness and amplitude were measured. Table I sum up their characteristics, and figure 6 shows the surface-elevation, measured during the experiment, at focus location for both wave groups. The symmetry was considered when the two draught were at same depth.

TABLE I
CHARACTERISTICS MEASURED OF THE TWO WAVE-GROUPS

Case	Measured steepness	Measured Amplitude (m)
ST1	0.167	0.285
ST2	0.189	0.302

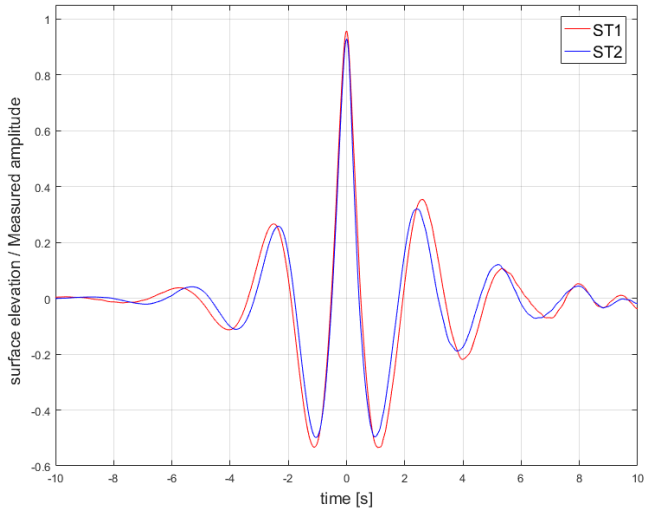


Fig. 6. Surface-elevation measured at focus location during the experiment

For the two wave-cases, surge and heave motion are compared, as they are of main importance for loads on the mooring line. Data are obtained from a Qualysis motion tracking system.

IV. RESULTS AND DISCUSSION

A. Decay test: results and discussion

Decay results are shown in Figure 7 and Figure 8, where the heave motion of the buoy is plotted against time as predicted by the two numerical modes and as measured in the laboratory experiment, using two different time scales. There is generally good agreement between the numerical predictions and the experiment data shown in both Figures. In term of resonance frequency, when considering only the first periods (Figure 7), OpenFOAM and WaveDyn find the same resonance frequency, f=0.91Hz, which is very similar to that measured in the

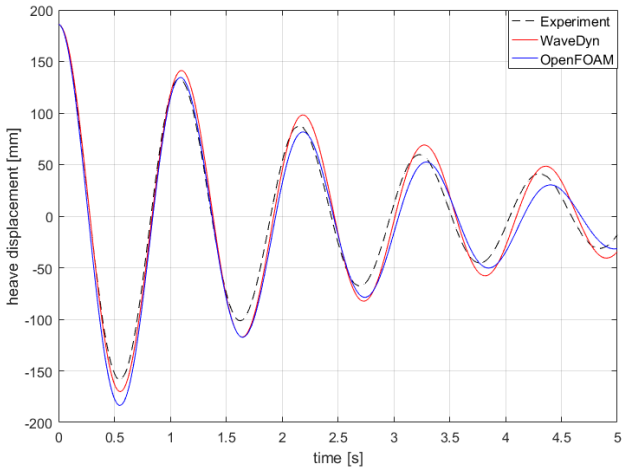


Fig. 7. First 5s for Heave decay test for moored X-MED

experiment, f=0.93Hz. At later wave periods, the OpenFOAM prediction appears to deviate from the experiment and the period of oscillation lengthens.

In terms of amplitude, both models over-estimate the heave motion in the first 5s, Figure 7, in both crest and trough. This over-estimation is seen throughout the WaveDyn simulation, whereas the amplitude of motion appears to be damped over time in the OpenFOAM simulation. OpenFOAM over-estimates mainly the trough amplitude, whereas WaveDyn over-estimates the amplitude in both crest and trough.

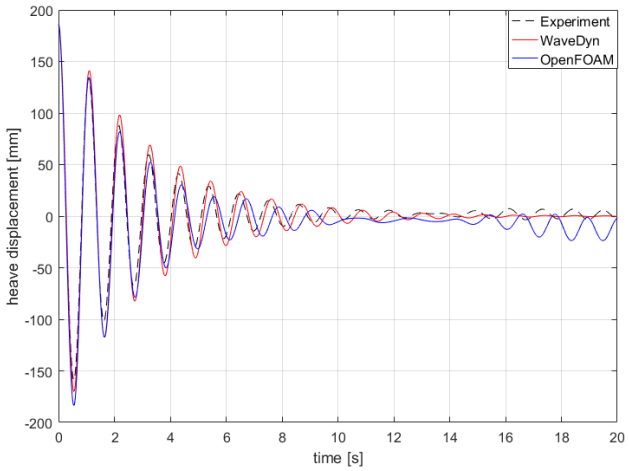


Fig. 8. Heave decay test for moored X-MED

After 14s, in Figure 8, experiment and OpenFOAM amplitudes of motion are amplifying again, whereas WaveDyn keeps following its normal decrease. This effect is probably due to waves generated by the buoy first oscillations, which are reflected by the side walls and come back at the model. As the WaveDyn BEM model do not represent the sides walls (the buoy evolves in an infinitely wide tank), those reflections effects could not be captured. Whereas the OpenFOAM NWT

width was chosen accordingly to the physical one so to capture those effects; therefore this constitutes a success in representation. This side-reflection issue in WaveDyn model will not influenced futures results as reflections in the experiment are considered to happen after the main crest.

In comparing the CPU requirement of the two numerical models; WaveDyn requires less than a minute to simulate $25s$ on a desktop computer, whereas OpenFOAM requires 13h on the high-performance computing facility ARCHER, using 24 processors in parallel.

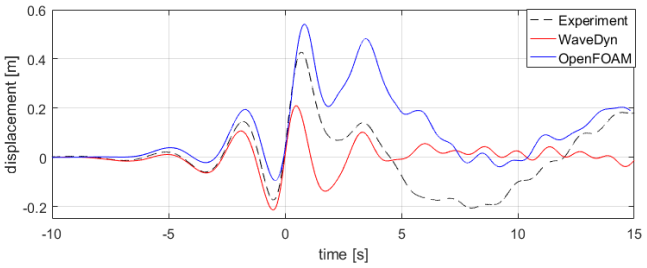
B. Wave-cases: results and discussion

Viscosity is likely to have an influence on the results in situations where turbulence and flow separation are important. WaveDyn is a linearised model and is likely to predict poorly situations in which non-linear interactions between waves and structures occur. Thus we would expect to see greater difference for the steeper wave case.

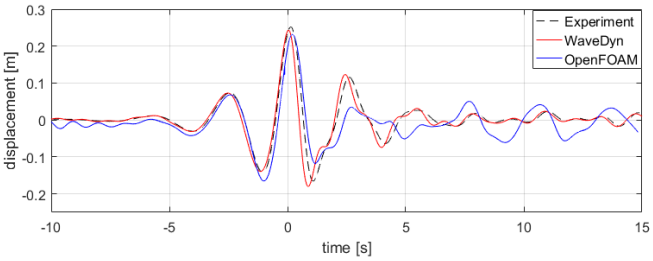
According to the experiments, shown as the black dotted line on Figures 9 and 10 - where both heave and surge motion of the buoy from the three models are plotted against time - the buoy movement can be decomposed into several steps, which appear to be correlated to the NewWave shape 1 (note that backward motion means towards the wave-maker, and forward motion is towards the beach):

1. Buoy is pushed forward and up - first NewWave peak
2. Buoy moves backwards and down - first trough of NewWave
3. Buoy is pushed up and forwards - Main crest left hand side
4. Buoy starts to regain its resting position - Main crest right hand side and second trough of NewWave
5. Buoy is briefly pushed forwards and up - second NewWave peak
6. Buoy regains its resting position with oscillations - after $5s$

Both numerical models manage to reproduce the general behaviour, and the heave motion is particularly well captured for both numerical models as shown in Figures 9(b) and 10(b). In heave, WaveDyn appears to predict more accurately step 5, which is in both cases under-estimated by OpenFOAM. This success is likely to be explained by the difference in surface-elevation generation: WaveDyn uses the surface elevation measured at the focus location during the experiment as input, and therefore it does not represent the propagation of the wave, but assures a perfect (at first order) representation of the free-surface. Whereas in OpenFOAM, the wave is propagating from its inlet boundary (left hand side of the tank), which is defined using an upstream wave gauge, towards the tank end. So at the inlet, the free-surface description is perfect - as in WaveDyn, as it uses also a sum of wave components - ; but at the focus location, the wave is the result of the wave-group spread, therefore inducing errors in the free-surface descriptions. Those are probably due to numerical diffusion, or due to the use of a linear decomposition for the description of a non-linear input.

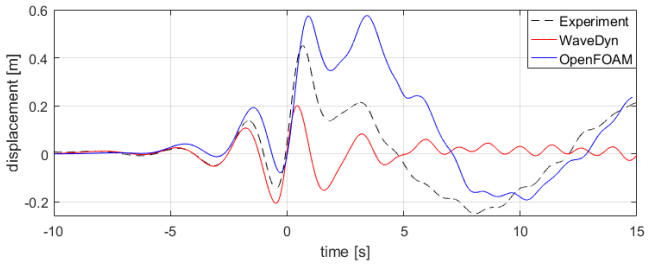


(a) Surge

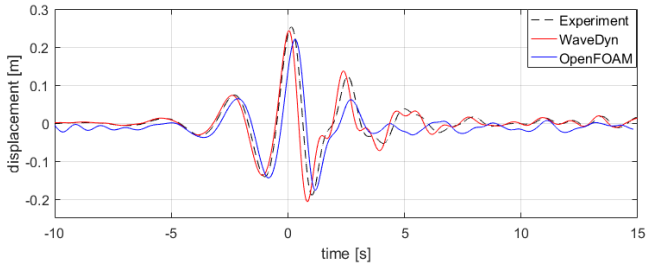


(b) Heave

Fig. 9. ST1 Experiment and numerical models comparison



(a) Surge



(b) Heave

Fig. 10. ST2 Experiment and numerical models comparison

In surge, figures 9(a) and 10(a), OpenFOAM over-estimated, whereas WaveDyn under-estimated the main surge peak. This motion - steps 3 and 4 - results from the push felt by the model due to the wave. Interestingly, step 5 is reproduced by both models, and it is overestimated in both cases. Step 6 in surge is not predicted by WaveDyn, possibly because the entire surge motion was underestimated. OpenFOAM manages to predict this pattern of oscillations, and captures even smaller ones. OpenFOAM represents the WSI in a fully-coupled way,

where fields have a direct consequence on the buoy 6-DoF, and vice-versa. All buoy's degrees of freedom are coupled as well, so that the each influences the other. WaveDyn assumes linear hydrodynamics for WSI (diffraction, radiation and hydrostatic force), where no viscosity is taken into account, and movement are considered as small. Surge motion is appears a consequence of the wave passage, and therefore the viscosity plays a key role on its description; hence explaining OpenFOAM success comparatively to WaveDyn. Also, in a similar fashion to a surfer waiting for its wave, the buoy is carried by the heave motion resulting from the wave. This coupling between degree of freedom appears important for the surge description, and explains models differences.

whereas WaveDyn assumes linear theory for the water-column description as well. This difference in fluid motion under the free-surface is likely to explain those one in surge motion. But unfortunately, neither model managed to capture, at a same level of accuracy as heave, the surge motion.

C. Model validity

Previous work on the X-MED buoy using NewWave, was published by Ransley [24], and shows similar behaviour in both heave and surge, which gives confidence to the OpenFOAM results presented here.

Carnegie [5] found their design wave as a combination of both maximum surge and heave motion. Therefore, the discrepancy found in the prediction of the surge motion by WaveDyn, even for the first wave group (ST1), might make the model inappropriate for wave groups of greater steepness for surge representation.

On the other hand, the OpenFOAM model still needs some improvements as there are some differences evident in the motion prediction for surge, which were unexpected. In the aforementioned study [24], surge motion was capture with a better accuracy by its OpenFOAM model. Therefore, some further development is required on the OpenFOAM model. For example, turbulences were not taken into account by the solver as the flow was considered as laminar. Turbulences models usually generates a damping of the motion as they decrease the fluid flow energy. Therefore, a possible development on the OpenFOAM model is to take into account those phenomenons by comparing turbulent models, which is likely to reduce surge motion.

V. Conclusion and future development

The two numerical models presented here are capable of reproducing movement of a floating moored buoy under two design-waves based on a NewWave representation for a 100 year event at WaveHub. However, both models shows some lacks compare to the physical model representation, used as reference.

Due to its linear wave generation representation (directly at focus location), WaveDyn represents very accurately the buoy heave motion, but lacks in accuracy for the surge motion, which seems to be more influenced by wave propagation and its influence on the water column description.

OpenFOAM appears to succeed to represent more accurately overall buoy behaviour, but with concerning inaccuracies. As previously stated and proposed by E.J. Ransley in [24], using linear superposition to generate highly non-linear waves is concerning. A higher order decomposition for NWT wave-maker boundary is required for development, and will hopefully over-come heave motion representation. Also, OpenFOAM model might be incomplete as no turbulences model was used, which might greatly helped to overcome the over-estimation of the surge motion representation. An OpenFOAM models comparison between the similar study [24], achieving better results, will constitute the first step of a future development.

Unfortunately, no concrete difference between the two wave group were found in this study, therefore limiting the investigations over wave-steepness. But the lack in accuracy of WaveDyn model in surge might suggest the use of a less steep wave group case for future investigations, as WaveDyn surge representation can be considered as inaccurate. Steeper wave group might also be useful to find an influence on heave representation. But due to WaveDyn wave generation, heave representation is expected to succeed as the long as buoy heave motion is similar in shape to the wave.

But in terms of the time allocated for design by WEC developers, OpenFOAM CPU cost [a week] can be put into debate as WaveDyn simulations [couple of minutes] are comparatively all but instantaneous. WaveDyn shows some great success in representing behaviour previous to the main event part, or considering heave motion only. Also, OpenFOAM has proven its capacities in many studies, such as [6], to represent well highly non-linear fluid-structure interaction (i.e. wave-breaking, large motion, over-turning surface...), to take into account turbulence models representing flow separation, and being able to handle geometries with non-linearities; different physical phenomenon present in extreme events that WaveDyn cannot represent.

Therefore, it is expected that there is a possible optimisation - if considering that OpenFOAM model can better succeed-, in terms of both CPU effort and accuracy, in which the appropriate model is selected according to the non-linearity present. This idea is also approved by B.F.M. Child [25], which research is linked to this work. Future work will consist of a coupling of these two models to realise such an optimisation and significantly reduce the computational overheads associated with survivability modelling of WECs. The idea is to use both software advantages: WaveDyn speed in weakly non-linear events; and OpenFOAM accuracy out of WaveDyn range of capacity (for highly non-linear events). The coupling is likely to be tight with WaveDyn having the supremacy. Simulation runs on WaveDyn until an out-of-range event (highly non-linear) occurs. At this moment, simulation swaps to OpenFOAM, so that this event can be represented with accuracy. It swaps back to WaveDyn once the event is finished, and in the range of WaveDyn capacity. This coupling benefits in term of CPU and accuracy, using best capacities of the two models. Therefore, accurate range of model validity,

and a trigger for the swapping, are keys points for the coupling effectiveness.

ACKNOWLEDGEMENT

This work used the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk). The author would like to thank his supervision team for their everyday help and advices. The X-MED experiments was part of EPSRC project and constitutes a very useful database. Also, the author would like to thank all people involved in OpenFOAM development, from small codes to on-line tutorial and forums.

REFERENCES

[1] E. J. Ransley, "Survivability of Wave Energy Converter and Mooring Coupled System using CFD," Ph.D. Thesis, Plymouth University, 2015.

[2] CCP-WSI, "Wave Structure Interaction Computation and Experiment Roadmap," Produced by Plymouth University, Tech. Rep. 1, September 2016.

[3] M. Hann, D. Greaves, and A. Raby, "Snatch loading of a single taut moored floating wave energy converter due to focussed wave groups," *Ocean Engineering*, vol. 96, pp. 258–271, 2015.

[4] D. Magagna and A. Uihlein, "Ocean energy development in Europe : Current status and future perspectives," *International Journal of Marine Energy*, vol. 11, pp. 84–104, 2015.

[5] A. Rafiee, H. Wolgamot, S. Draper, J. Orzaghova, J. Fievez, and T. Sawyer, "Identifying the design wave group for the extreme response of a point absorber wave energy converter." AWTEC, 2016.

[6] C. Paper, J. Whan, K. Technipfmc, H. Jang, S. Lee, and S. Heavy, "Technical and Economic Readiness Review of CFD-Based Numerical Wave Basin for Offshore Floater Design," no. January, 2016.

[7] T. Vyzikas, D. Greaves, D. Simmonds, C. Maisondieu, H. Smith, and L. Radford, "Application of numerical models and codes," MERiFiC, Tech. Rep., 2014.

[8] DNV-GL, "WaveDyn presentation and documentation," 2016.

[9] "COAST Plymouth University Marine Engineering Laboratory," 2016.

[10] P. S. Tromans, A. R. Anaturk, and P. Hagemeijer, "A new model for the kinematics of large ocean waves application as a design wave," no. 1, January 1991.

[11] L. Xu, N. Barltrop, and B. Okan, "Bow impact loading on FPSOs 1– Experimental investigation," *Ocean Engineering*, vol. 35, no. 11-12, pp. 1148 – 1157, 2008.

[12] J. B. Rozario, P. S. Tromans, P. H. Taylor, and M. Efthymiou, *Comparison of Loads Predicted Using "Newwave" and Other Wave Models with Measurements on the Tern Structure*. In: Wave Kinematics and Environmental Forces. Advances in Underwater Technology, Ocean Science and Offshore Engineering, 1993, vol. 29, Springer, Dordrecht.

[13] A. C. Hunt-Raby, A. G. Borthwick, P. K. Stansby, and P. H. Taylor, "Experimental measurement of focused wave group and solitary wave overtopping," *Journal of Hydraulic Research*, vol. 49, no. 4, 2011.

[14] D. Z. Ning, J. Zang, S. X. Liu, R. E. Taylor, B. Teng, and P. H. Taylor, "Free-surface evolution and wave kinematics for nonlinear uni-directional focused wave groups," *Ocean Engineering*, vol. 36, no. 15-16, pp. 1226–1243, 2009.

[15] H. Versteeg and W. Malalasekara, *An introduction to computational fluid dynamics - The Finite Volume Method*, 2nd ed. Pearson Education Ltd., 2007.

[16] J. Davidson, M. Cathelain, L. Guillemet, T. L. Huec, and J. Ringwood, "Implementation of an OpenFOAM Numerical Wave Tank for Wave Energy Experiments," *Proceedings of the 11th European Wave and Tidal Energy Conference*, 2015.

[17] E. J. Ransley, M. Hann, D. Greaves, A. Raby, and D. Simmonds, "Numerical and physical modeling of extreme waves at Wave Hub," *Journal of Coastal Research*, vol. 2, Special Issue, no. 65, 2013.

[18] "OpenFOAM-4.1 rigidBodyMeshMotion Class Reference," Accessed July 2017.

[19] N. G. Jacobsen, D. R., Fuhrman, and J. Fredsøe, "A wave generation toolbox for the open-source CFD library: OpenFoam®," *International Journal for Numerical Methods in Fluids*, no. 70, pp. 1073–1088, October 2011.

[20] P.-H. Musiedlak, D. Greaves, B. Child, G. Iglesias, M. Hann, and E. Ransley, "Geometric parameters for a 2D-Numerical Wave Tank," International Network for Offshore Renewable Energy (INORE) 10th European Symposium, 12-18 June 2016, Nantes, France; and PRIMaRe 2016, University of Bath UK.

[21] W. Cummins, "The impulse response function and ship motions," *DAVID TAYLOR MODEL BASIN WASHINGTON DC*, 1962.

[22] Halcrow and South West of England Regional Development Agency, "Wave Hub Development and Design Phase, Coastal Processes Study Report," 2006.

[23] T. E. Baldock, C. Swan, and P. H. Taylor, "A Laboratory Study of Nonlinear Surface Waves on Water," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 354, no. 1707, pp. 649–676, 1996.

[24] E. J. Ransley, D. Greaves, A. Raby, D. Simmonds, and M. Hann, "Survivability of wave energy converters using CFD," *Renewable Energy*, vol. 109, pp. 235 – 247, 2017.

[25] B. F. M. Child, J. Oberhagemann, M. Hann, , D. Greaves, and A. Raby, "A methodology for identification and simulation of extreme design load cases for wave energy converters," in *METS*, 2017.

6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
1115 June 2018, Glasgow, UK

STARTING A WAVE STRUCTURE INTERACTION CFD SIMULATION FROM AN ADVANCE TIME: HOT-START

MUSIEDLAK P.-H.[1], RANSLEY E.[1], BROWN S.A.[1], CHILD B.F.M.[2], HANN M.[1], IGLESIAS G.[1] AND GREAVES D.[1]

[1] School of Engineering, Plymouth University
Reynolds Building, Drakes Circus, PL4 8AA Plymouth, United Kingdom
pierre-henri.musiedlak@plymouth.ac.uk

[2] DNV-GL
One Linear Park, Avon Street, Bristol BS2 0PS, United kingdom
benjamin.child@dnvgl.com

Key words: Hot-start, CFD, wave-structure interaction, Mesh-deformation, Coupling

Abstract. This paper details the reduction of an OpenFOAM CFD WSI simulation to its major event, by starting it at an advance time: a new procedure named hot-start, which is a first step towards a future coupling development. The investigations concern the fluid flow and a structure motion hot-start, taken separately, and are restricted to numerical comparison. Four design waves based upon the NewWave theory are simulated with increasing starting times - where the wave field is initialized as the sum of the linear components of the considered wave - and compare to conventional ones - which are initialized with still water. The structure motion hot-start is assessed using a heave decay test: a conventional heave decay simulation is compared against several ones where the motion, velocity and acceleration of the structure are assigned for several time-steps. An initial mesh-deformation library is specifically created to assign the structure at the hot-start position by incrementally deforming the mesh towards the right structure position. Independent to the non-linearity of the case, a start 4 s prior to the main event is found to be enough to accurately represent the wave field. The motion of the structure was found to require at least 5 time-steps in order to converge to the reference one. Those results aim to be usable for other CFD WSI applications.

1 INTRODUCTION

Numerical modelling is widely used in offshore and coastal engineering to assess wave-structure interaction (WSI) since it gives increased understanding of processes such as: the evolution of the coast line; the manoeuvrability of ships; the mechanical design of floating oil and gas platforms; or of wind turbines. Many offshore standards are based upon numerical modelling, and often adapted with the experience gathered by success and failures. Marine Renewable Energy (MRE) developers use the offshore oil and gas industry standards. But, nowadays the numerous failures of the different MRE devices

has proven those standards to be misfits to the sector [1]. Often a MRE - especially Wave Energy Converters (WEC) - device motion needs to be accentuated to generate power [2], unlike traditional oil and gas floating structure which are designed to controlled and limit their motion. Hence, this dynamic behaviour of MRE devices requires models which are capable of accurately simulating large motions. Also, a MRE device is often composed of multiple components which interact with each other, resulting in complex, and often highly non-linear, device motion which depends heavily on past events [3].

Therefore, these industries require a more complex numerical model which is able to assess such levels of physical complexity. Computational Fluid Dynamics (CFD) simulations solve the full Navier-Stokes equations with limited simplifications. Their use as a design tool is growing in several industries, where the offshore industry starts to recognize their reliability compare to empirical methods use in industry standards. But this increase in complexity induces an increase - sometimes drastic - in Central Processing Unit (CPU) effort. This is a major issue if such methods want to be utilized in routine design processes [1], since it limits the use of CFD to very case specific physics representation, or research bases cases.

In WSI, CFD simulations are mainly used for mechanical design of extreme loads. Typically, the engineering method uses a design wave, which hits the structure, hence allowing prediction of the loads. Using a Numerical Wave Tank (NWT) starting from still water, the CFD simulation generates the wave at the inlet, and then propagates it towards the outlet. The period of time necessary to create the wave from still water is required to build-up of fully non-linear fluid flow. This is the common set-up of a WSI CFD simulation: a similar example to this study is the WEC developer Carnegie who simulated in a NWT, the dynamic response of their device under extreme events using the NewWave description, where the simulations started from still water [4].

However, the main interest of a WSI simulation is the impact of the wave on the structure, rather than the propagation of the wave itself, and this constitutes only a small amount of the full CFD simulation. Therefore, this paper presents a novel approach that limits WSI CFD simulations to the times of interest: the simulation will start slightly before the impact - this strategy is termed 'hot-start'. It is expected to result in significant CPU savings without substantially compromising the accuracy of the results. In the case of use for WSI problems, and to maintain the accuracy of the results, a hot-started simulation requires consideration two mains issues (taken separately in this study): 1) the wave field reproduction, and; 2) the hot-start assignment of the motion of the structure.

This study is a first step in the development of a coupling between an industry standards based numerical model - WaveDyn, developed by DNV-GL in Bristol UK , and the open-source, CFD code OpenFOAM. To maximize efficiency, the coupling strategy utilises the computationally efficient method, WaveDyn, preferentially reserving the expensive NS solver for instances in which the linear assumptions of WaveDyn are violated [5]. The coupling strategy is outside of the scope of this paper. This study focuses on the achieving the hot-start for a CFD simulation in a purely numerical approach. No comparison with experimental data will be conducted. The study objective is to prove the feasibility of a CFD hot-start for the wave field, and for a rigid body motion.

2 METHODOLOGY

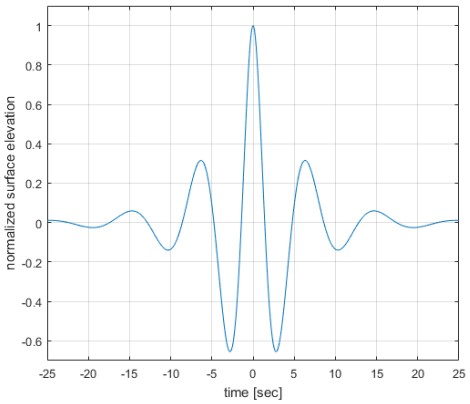
2.1 Initial conditions for the wave field variables

The objective of the following method is to be able to accurately hot-start a $2D$ wave-only simulation using a design wave. It aims to be adaptable for other wave design. Investigations are compared against CFD simulations using the usual strategy, i.e. starting from still water at time $t = 0$.

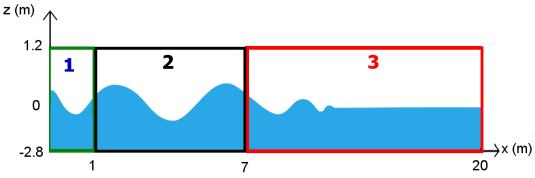
2.1.1 The reference NWT and design wave set-up

Design waves are typically used to assess the survivability of a structure in extreme wave events, [4], [2]. They aim to generate the maximum loads the structure would be exposed to during its design lifetime. This engineering method has been subjected to debate: indeed, for structures subject to large motions, more extreme loads can be found outside the scope of this extreme representation depending on the historic of the device motion [6]. But, this method is still widely used, and is used in this study.

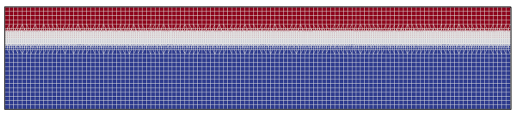
This study uses a Pierson-Moskowitz spectrum from a 100 year storm hindcast data obtained at the Wave Hub site ($Tz = 14\,\mathrm{s}$, $Hs = 4.4\,\mathrm{m}$, [7]). The design wave is defined using the NewWave [8] wave representation, which produces, for a given sea-state, the average shape of the highest wave with a specified exceedance probability [9]. This shape is a focus event which occurs at a specific position in time and space, as shown in Figure 1a. The mathematical description at first order is defined by a sum of linear waves; thus it is easy to implement at the NWT inlet. Four NewWave type focus events of increasing steepness are obtained from this hindcast [6].



(a) Theoretical shape of the surface-elevation of a NewWave event at focus location, generated by a Pierson-Moskowitz spectrum



(b) A schematic representation of the $2D$-NWT: 1 is the inlet; 2 is the working region; and 3 is the relaxation zone



(c) The domain is a $20 \times 0.1 \times 4\,\mathrm{m}$ cuboid consisting of cubic background of 6 cells per meter, refined to level 3 around the mean-water line

Figure 1: The reference design wave and the $2D$-NWT set-up

The four focus event are reproduced in the 2 dimensional (2D) NWT represented in

Figure 1b, which dimensions are based upon a previous study [10]. The $waves2Foam$ library is used for the wave generation and absorption methods [11]. The wave is generated at the inlet, which is the region number 1 in Figure 1b. At the inlet, a superposition of linear wave components, obtained using a Fast Fourier Transform (FFT) of the experiment wave gauge located the furthest upstream, is used to generate the wave. The components are selected by incrementally adding waves in order of magnitude (largest first) until a user prescribed precision is achieved [5]. This optimization of the selection of the number of components was found to save Random Access Memory (RAM) and CPU effort [12]. The wave then propagates in a fully non-linear manner along the NWT in the working region, which is the region number 2 in Figure 1b. The outlet, or relaxation zone, is the region 3 in Figure 1b, where the wave is absorbed. To assure the fully non-linear propagation of the four waves cases, the reference CFD simulations are starting from still water at time $t = 0$; the conventional CFD set-up.

The $2D$-NWT mesh shown in Figure 1c is generated using the commands $blockMesh$ and $snappyHexMesh$. The background mesh is made of 6 square cells per meter, as square meshes were found to converge more rapidly than one of increasing ratios [2], and to more accurately reproduce the physics. This background mesh is then refined three times using the octree refinement strategy [13] around the mean water line along the full length of the tank (as simulations were found to be slower and different from a fully square mesh if a shorter refinement region was used).

The experimental equivalent position of the structure (here the X-MED buoy [6]) is considered as the focus space location, abscissa $x = 5.58\,\mathrm{m}$. The focus time, t_{focus}, is chosen as the time where the highest surface-elevation is observed at this location. The flow field characteristics are measured at this position: the surface-elevation as a function of time; the water velocities and pressure along the water-column. The surface elevation is measured during the simulation by the library $waveGaugesNProbes$ [11]. The velocity and pressure at time $t = t_{focus}$ along the water column are post-processed using Paraview by slicing the domain at the structure position, and then extracting the two profiles.

2.1.2 Method

A hot-start CFD NWT set-up only differs from a conventional one by its starting time, t_{hot}, and the initial set-up of the wave fields. The wave field is described at the inlet by the same sum of linear components in both cases, but for a hot-start simulation because the starting time is different than 0, the description applies to the full NWT length at the hot-start time, t_{hot}. Therefore, compare to a conventional CFD simulation where the fluid has been propagating in a fully non-linear manner across the NWT until the hot-start time, a hot-started simulation at this starting time, i.e. t_{hot}, will lack in accuracy.

But the differences are expected to reduce as the two simulations run, as the hot-started simulation build-up to a fully non-linear description. In other words, in order to reproduce a non-linear event, it is expected that a period of time is required for a hot-started simulation to converge to the reference one. So, the time of the focus extreme event is considered as the location in time where the simulation needs to be hot-started:

$t_{focus} = t_{hot}$. Additionally, a period of time, t_{minus}, is subtracted to the hot-start time so that the hot-started simulation can build up to the solution. It is required that this parameter is smaller than the hot-start time; otherwise the conventional CFD set-up would be used.

Therefore, the method consists of running hot-started simulations with increasing t_{minus}, from 0 to t_{hot}. As t_{minus} increases, the hot-started simulation have more time to build up towards the reference one, and by comparing the reproduction of the focus event, a convergence is expected. The surface elevation, velocity and pressure fields predicted by the hot-started simulations are benchmarked against the predictions from the reference simulation, which used the conventional setup. To investigate the dependency of the hot-start with the non-linearity of the event, the four focus event of increasing steepness are used. The convergence of the hot-started simulation are expected to be depend on the non-linearity of the case.

2.2 Positioning the structure and assigning initial motion state

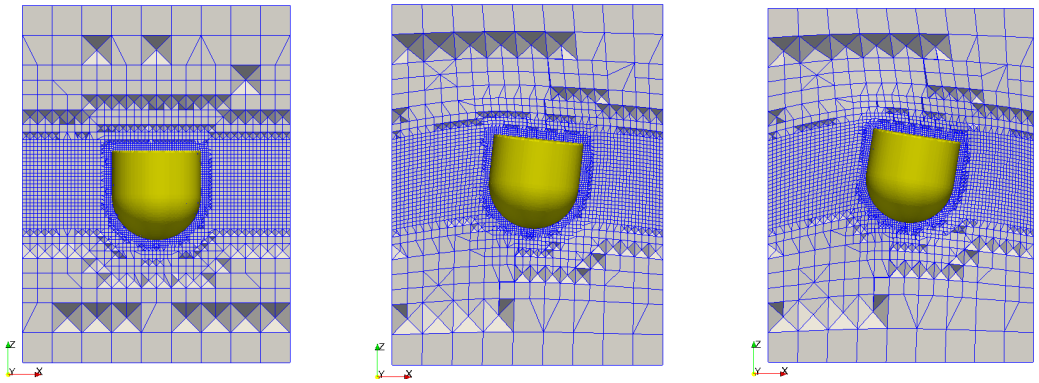
In this second sub-section, the study focuses on the hot-start for a rigid-body only, trying to avoid the influence of the fluid. Its objective is to accurately hot-start a $3D$ simulation involving a simple motion of a rigid-body to provide the proof of the hot-start concept for WSI applications.

2.2.1 The initial mesh deformation - $deformDyMMesh$

As the structure position at the hot-start is supposed to be known, the usual approach is to generate a new mesh with the up-to-date geometry file (*.stl* using a Computer Aided Design software (CAD)) from this new position. The final mesh is undeformed, and any new deformation due to the movement of the structure, will deform the mesh. Also, if the hot-start position is different from the structure's equilibrium, the deformation of the mesh will increase as the structure returns to its equilibrium. This might led to a mesh of lower quality at the equilibrium which can generate some instabilities. This pre-process step can be quite time-consuming, and, in the future use of this study for a software coupling, the structure position could be different for each simulated cases. And, this would require to generate a different geometry file each time, or to make this automatically, which means another coupling process with the CAD software.

This study uses a new approach, where only the mesh with its structure at equilibrium is first needed. By deforming this mesh, the structure is moved to its position at the initial hot-start time. One advantage of this method is that only one geometry model is required, and that any structure position can be obtained simply be deforming the mesh.

For this purpose, a new library was created based on the $waveDyMFoam$ solver and the $rigidBodyDynamics$ library from OpenFOAM-4.1; named $deformDyMMesh$. Using as input an offset from the structure position and a number of iterations, it moves the structure by the amount defined by the amplitude of the offset divided by the number of iterations. The mesh is then updated, and this results in a new deformed mesh. By repeating this process the number of iterations, the structure ends at the wanted the



(a) Original mesh (b) Half-deformed mesh (c) Mesh fully deformed

Figure 2: Steps of the $deformDyMMesh$ use

position, and the mesh is deformed accordingly. The larger the offset, the more iterations are required to ensure the mesh quality of each iteration. A poor mesh quality results often in squeezed cells, and an inability of using the deformed mesh for any further simulations.

An example of the capacity of this library is shown in Figure 2, where a structure, here the X-MED buoy [6], is successfully moved from its initial position using the $deformDyMMesh$ library, and that it results in a deformed mesh of quality.

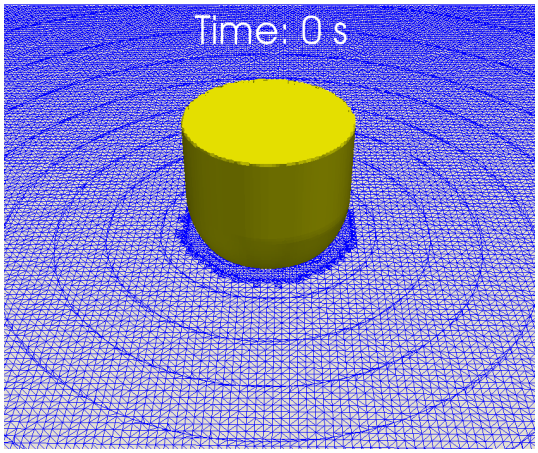
The hardware used for this study is the Viglen Genie computer, equipped with Intel Xeon E5-1680 v4 at 3.40 GHz with 16 processors, where the 64 version of Ubuntu is directly installed. All the commands are run in serial mode.

The execution of the original mesh in Figure 2a takes 14 s for this hardware: $blockMesh$ of a $1.5 \times 1.5 \times 2$ m box with 6 cells per meter, refined to level 3 between $[-0.25; 0.25]$, and to level 4 at the structure surface; a total of 160306 cells. The deformation done in this example moves the structure by: 0.06 m in surge, 0.14 m in heave, a 10 degree angle in pitch, and none in sway, roll and yaw. Depending on the number of iteration required for the deformation, $deformDyMMesh$ takes between 50 s for 50 iterations, and drops to 10.2 s for 15 iterations, which is the minimum number of iterations found for this amplitude of deformation for this case.
However the proven quality of the mesh generated by the $deformDyMMesh$ library, no proof of its ability not to influence the results has been done so far.

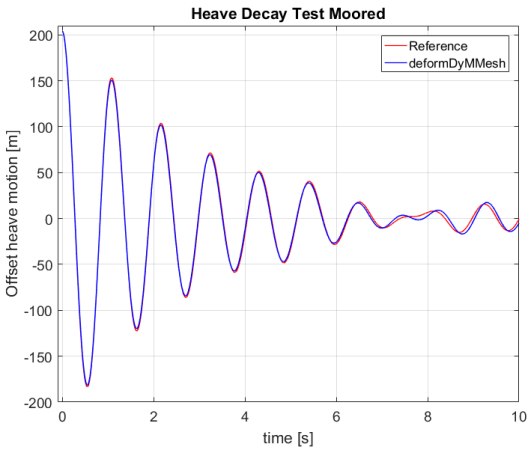
2.2.2 Proof of use of $deformDyMMesh$

To prove the use of this library, a heave decay test is performed. The simulation reference is carried out using a geometry file updated according to the heave decay release position to insure the initial mesh to be undeformed. And, it is opposed to a simulation starting with a mesh deformed by the $deformDymMesh$ library from the structure at its equilibrium position to the release position. The flow fields are set as still water. No hot-start are considered here, and the structure used for the proof of $deformDyMMesh$ is the X-MED buoy, which has no velocity nor acceleration set at the starting time for

both cases.



(a) The case using $deformDyMMesh$ with a cut at the mean water line at time $t = 0\,\mathrm{s}$



(b) The heave displacement of a conventional simulation compared to one using $deformDyMMesh$ library

Figure 3: A decay test to prove the use of the $deformDyMMesh$ library

Figure 3b shows the results of the simulations by comparing the two structures heave displacement as an offset from the structure equilibrium position. Very slight differences are found before time $t = 6\,\mathrm{s}$. And, those are expected to be due to the small difference in the initial flow fields as the deformed mesh generates small circular like waves, which can be seen in Figure 3b. After time $t = 6\,\mathrm{s}$, the small are growing due to the radiated waves generated by the structure motion and reflecting on the walls. But, those differences are considered as negligible, where the circular waves and the slight differences are expected to be sensitive to the heave decay test only. Therefore, this proves the use of the $deformDyMMesh$ library for the structure initial position assignment without influencing the simulation.

2.2.3 The initial structure velocity and acceleration

However, for an hot-started simulation, the assignment of the position expected to be not sufficient on its own, and that the initial velocity and acceleration of the structure are of importance, and therefore, are required as additional initial conditions on the structure motion.

The importance of specifying the structure initial velocity and acceleration for a hot-start is investigated using the same heave decay test, with the same reference case, and compared against hot-starting simulations of different initial velocity and acceleration set-up. In order to investigate only the initial structure motion conditions, it is necessary to avoid, or at least reduce, the effects of the fluid. So, the hot-started simulations are starting from an early time, $t_{hot} = 0.1\,\mathrm{s}$, where the influence of radiated waves due to

the structure drop can be neglected, but where the structure motion is significant enough not to be neglected. Using the previous conclusion, the hot-started simulations use the $deformDyMMesh$ library to assign the structure position at the hot-start time. Firstly, a simulation will not specify the velocity and acceleration to prove its necessity for an hot-started case. Secondly, several simulations will specify the velocity and acceleration found by the reference simulation for an increasing number of time-step, before releasing the structure.

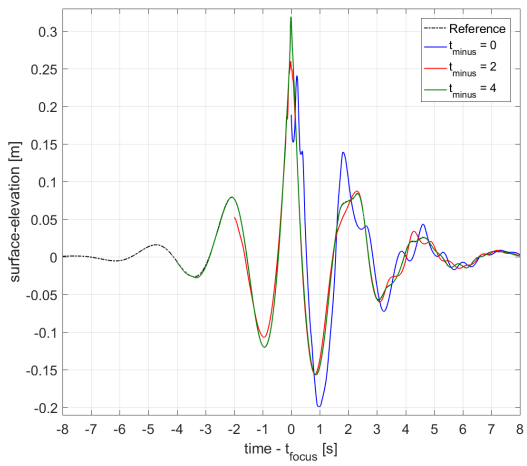
The comparison is limited to the first second of simulation to avoid reflections, and because the motion is expected to converge towards the reference one. An investigation on the number of corrective time-steps required for the convergence is also conducted.

3 DISCUSSION

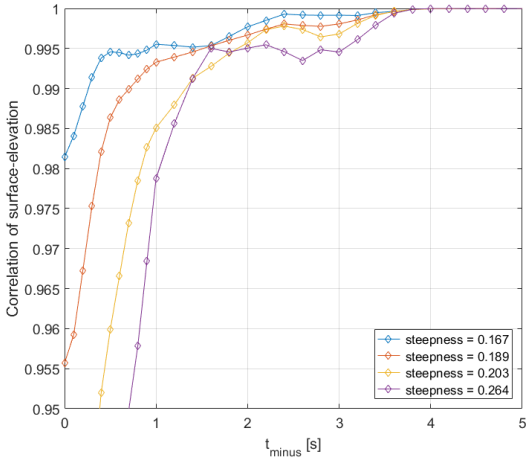
3.1 Initial conditions for the wave field variables

3.1.1 Surface-elevation

In Figure 4a the surface-elevation of $2D$ wave-only simulations of the steepest case are plotted in colour against the reference simulation, the dotted line. Figure 4b presents the correlation between a hot-started simulation and the reference one, as a function of the t_{minus} used for each simulation. The wave cases are numbered by their experimental measured steepness [6].



(a) The surface-elevation of three different hot-started $2D$ wave-only simulations, for the steepest case compared with the reference



(b) Correlation comparison of the surface-elevation of 2D simulations starting at a specific time against the reference one

Figure 4: The surface-elevation hot-start results

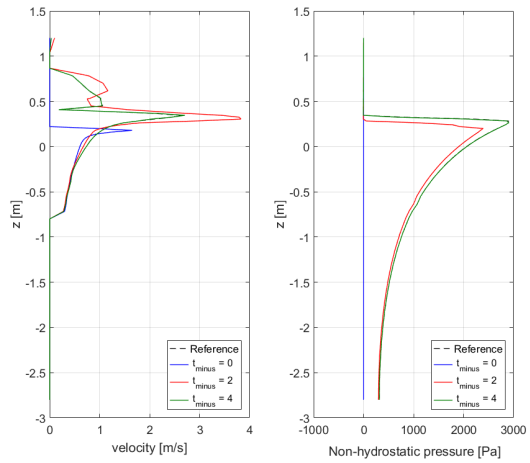
The surface-elevation of the hot-started simulation with $t_{minus} = 4$ is the green line in Figure 4a, and is on top of the reference one. Indeed, in Figure 4b, $t_{minus} = 4$ clearly appears as the first converged solution for the surface-elevation representation. And,

unexpectedly, this result is valid for the four cases, hence not depending on the steepness or the non-linearity of the wave. But, the convergence of the solution is slower as the non-linearity of the wave increases. Consequently, if a lower correlation criterion would be used, a lower t_{minus} could be used for the less steep cases.

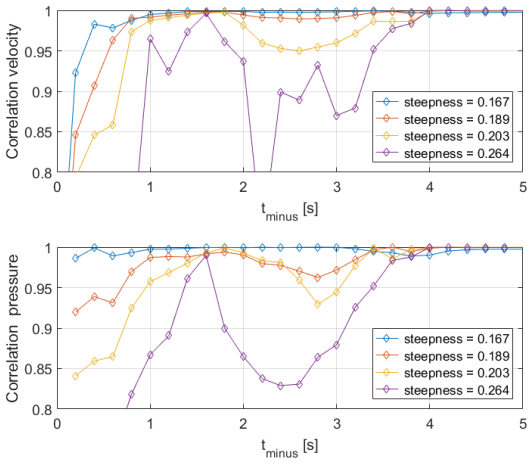
As expected, when no build-up period is allowed, for $t_{minus} = 0$, the sum of linear component is set as the initial condition across the NWT, which results in significant differences, as shown by the blue curve in Figure 4a.

3.1.2 Velocity and Pressure

Figure 5a presents the fluid velocity and pressure profiles at focus time, at the tank location $x = 5.58$m, for the steepest case. The black dotted line is the reference case, and three t_{minus} hot-start simulations are plotted on top. For the four waves cases, the correlation between a hot-started simulation and the reference one, is plotted against t_{minus} in Figure 5b.



(a) Velocity and pressure profile at focus time, at the tank location $x = 5.58$m, for the steepest case



(b) Correlation comparison of the velocity (top) and pressure (bottom) profiles of 2D simulations starting at a specific time against the reference one

Figure 5: The fluid velocity and pressure hot-start results

As for the surface-elevation, the solution converges at $t_{minus} = 4$ (Figure 5b), where the two profiles are exactly on top of the reference in Figure 5a. Therefore, all the results and behaviour previously obtained with the surface-elevation comparison are valid for the velocity and pressure profiles comparison.

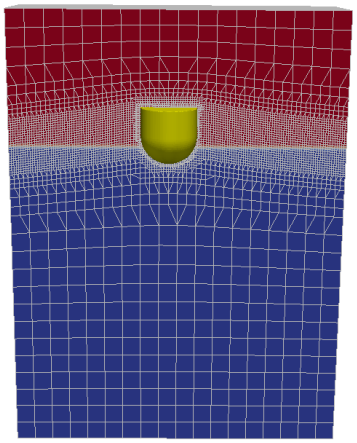
But it could be noted that for the steepest case, there is a real need of using $t_{minus} = 4$ as the correlation drops down for t_{minus} values between 2 and 3; purple curves on Figures 5b top and bottom. A less significant reduction can be observed for less steep cases, with a minor amplitude in the correlation of the surface-elevation in Figure 5b.

Using a $t_{minus} = 4$ hot-start simulation compare to a conventional one allows a reduction of 25% of CPU for the least steep case, and of 12% for the steepest one.

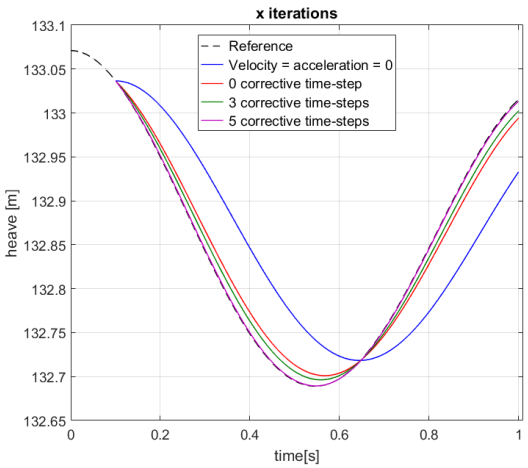
3.2 Positioning the structure and assigning initial motion state

3.2.1 Hot-start heave decay test

In Figure 6a, the initial position of the structure is set, thus the mesh is deformed. The flow fields are set as flat water across the domain (water is in blue, air in red). This initial set-up is used for the four different hot-started cases presented in Figure 6b, where several rigid body hot-start tries are compared. The referent heave decay is plotted in black dotted line.



(a) The mesh deform using the *deformDyMMesh* library, at the hot-start time $t = 0.1$ s for a heave decay test



(b) Comparison of different initial motion-state procedure for the 0.1s heave decay hot-started simulation: when the velocity and acceleration are or are not specified; and when additional corrective time-steps are used

Figure 6: The structure motion hot-start results

If the velocity and acceleration of the structure are not specified - hence zero - in the initial set-up, then the simulation results as a decay test released from a different height; the blue curve on Figure 6b. However, once the velocity and acceleration in the initial hot-start set-up are specified, significant improvements can be observed; the red curve on Figure 6b. Finally, the addition of corrective time-steps causes the simulation to converge towards the reference solution; Figure 6b shows that, the assignment of an initial motion can be done using at least 5 corrective time-steps.

4 CONCLUSIONS

- This paper investigates the starting of a WSI CFD simulation from an advance time, named hot-start. It aims to reduce the CFD simulation strictly around the non-linear event, by reducing the build-up time usually used in CFD to launch a simulation. It also constitutes an important piece of a future coupling procedure. In this study, the hot-start investigations are restricted to the wave field and the structure motion, separately.

- Compared to a non hot-started CFD simulation, the representation of the focus event was found to be accurately reproduced if started $4s$ before the main event; hence not requiring more than $6s$ of simulation. This conclusion was found to be independent on the non-linearity of the wave, and also confirmed through the three flow field: surface-elevation, water column velocity and pressure.

- A new library - $deformDyMMesh$ - was achieved in order to deform the mesh according to the structure position at hot-start, and its use was proven to have no influence on the results of the simulation.

- The assignment of motion of a structure was found to require at least 5 corrective time-steps - where each position and motion found by the previous time-step were corrected using the reference case - before converging to the referent solution.

- Therefore, this study proves the use and possibility of an advance start for CFD simulations in WSI cases based on a focus event. The method and the results are expected to be adaptive to other WSI in different CFD applications.

5 ACKNOWLEDGEMENTS

The code presented here is available through the Collaborative Computational Project in Wave Structure Interaction (CCP-WSI) [EP/M022382/1], which brings together the WSI community to share CFD developments in an open source environment (https://www.ccp-wsi.ac.uk) [14].

The author would like to thank his supervision team for their everyday help and advices. Thanks to S.A. Brown for its reliable help in OpenFOAM coding. Also, the author would like to thank all people involved in OpenFOAM development, from small codes to online tutorial and forums.

REFERENCES

[1] CCP-WSI Working Group, *Wave Structure Interaction Computation and Experiment Roadmap - Part 1: A Report on the 1st CCP-WSI Focus Group Workshop*, 2016

[2] Ransley, E (2015). "Survivability of wave energy converter and mooring coupled system using CFD", *PhD thesis, School of Marine Science and Engineering, University of Plymouth*

[3] Pinna, R., Cassidy, M., Dynamic analysis of a monopod platform using constrained NewWave, International Conference on Offshore Mechanics and Arctic Engineering, *23rd International Conference on Offshore Mechanics and Arctic Engineering, Volume 2*. (pp. 141-148). doi:10.1115/OMAE2004-51158.

[4] Rafiee, A.; Wolgamot, H.; Draper, S.; Orszaghova, J.; Fivez, J.; Sawyer, T. Identifying the design wave group for the extreme response of a point absorber wave energy converter. *In Proceedings of the Asian Wave and Tidal Energy Conference (AWTEC), Singapore, 2428 October 2016*

[5] Musiedlak, P-H, Ransley, E, Greaves, D, Child, B, Hann,Mand Iglesias, G (2017). Investigations of model validity for numerical survivability testing of WECs , *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*, Cork, Ireland.

[6] Hann, M., Greaves, D. & Raby, A. (2015), 'Snatch loading of a single taut moored floating wave energy converter due to focussed wave groups', *Ocean Engineering 96*, 258271. URL: http://dx.doi.org/10.1016/j.oceaneng.2014.11.011

[7] Halcrow (2006), South West of England Regional Development Agency, Wave Hub development and design phase, coastal processes study report, pp19, 2006.

[8] Chaplin, J. R. (1996), On frequency-focusing unidirectional waves, International Journal of Offshore and Polar Engineering, 6(2), 131137.

[9] Xu,L., Barltrop,N., Okan, B.,(2008). Bow impact loading on FPSOs 1 - experimental investigation, Ocean Engineering 35, 11481157, https://doi.org/10.1016/j.oceaneng.2008.04.013

[10] Musiedlak PH., Greaves D., Child B., Iglesias G., Hann M., and Ransley E.J., Geometric parameters for a 2D-Numerical Wave Tank, *International Network for Offshore Renewable Energy*, 12-18 June 2016, Nantes, France; and *PRIMaRe* 2016, University of Bath UK.

[11] Jacobsen, N, Fuhrman, D and Fredse, J (2012). A wave generation toolbox for the open-source CFD library: OpenFOAM®, *International Journal for Numerical Methods in Fluids*, Vol. 70, pp. 10731088

[12] Brown S.A., Musiedlak P.H., Ransley E.J. and Greaves D.," Numerical simulation of focused wave interactions with a fixed FPSO using OpenFOAM 4.1", *ISOPE* (2018)

[13] Greaves, D., Borthwick, A., 1999. Hierarchical tree-based finite element mesh generation. Int. J. Numer. Meth. Eng. 45, 447471.

[14] A Collaborative Computational Project in Wave Structure Interaction http://www.ccp-wsi.ac.uk/

Bound copies of poster presentation

INORE 10th European Symposium 12-18 June 2016 Nantes, France

Geometric parameters for a 2D-Numerical Wave Tank

Pierre-Henri Musiedlak // pierre-henri.musiedlak@plymouth.ac.uk // Plymouth University, UK

Deborah Greaves, Ben Child (DNV-GL) Gregorio Iglesias, Martyn Hann, Edward Ransley

Aim

This poster aims to be a storyboard on the realisation of a 2D-Numerical Wave Tank (2D-NWT), simulating numerically a wave-tank and its experiments.

Context

In the atmosphere of after the COP21 and its 2°C agreement, ORE shows even more its potential. But, engineering design and survivability of devices are still critical, and consist of a key point to insure the sector development.

Tools

➡ **OpenFOAM**, an open-source software made of a set of C++ libraries written on text files, allowing customization

➡ **Waves2FOAM**, a library allowing generation and absorption of waves.

Physics and equations

➡ **Physics:** two incompressible viscous fluids

➡ **Mathematical model:**
Navier-Stokes for incompressible fluids

Continuity equation $\nabla \cdot \mathbf{U} = 0$

Mass conservation

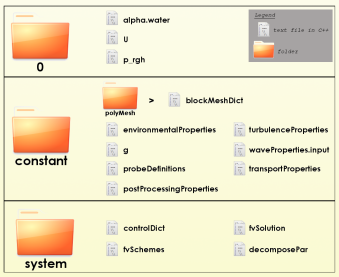
$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) - \nabla \cdot (\mu_{eff} \nabla \mathbf{U})$$
$$= -\nabla p^* - g * X \nabla \rho + \nabla \mathbf{U} \cdot \nabla \mu_{eff} + \sigma \kappa \nabla \alpha$$

➡ **Numerical method:** Finite Volume Method for space and time discretisation

➡ **Interface air/water:** Volume of Fluid (VoF) Method - a fraction $\alpha \in [0,1]$ at each cell (0 is pure air, 1 is pure water)

➡ **Unknowns:**
`U` (velocity)
`p` (pressure - is `p/rho*g*h` in OpenFOAM)
`alpha.water` (water proportion)

OpenFOAM architecture



Case directory

OpenFOAM is set up with lots of different classes - `C++ special names` - use to define different parameters

Using the terminal in the case directory:
➡ `blockMesh` : generate the mesh according to blockMeshDict
➡ `waveGaugesNProbes` : generate wave-gauges
➡ `setWaveParameters`: set wave input
➡ `setWaveFields`: set wave and fields (alpha.water, U, p_rgh)
➡ `waveFoam` : run solver
➡ `paraFoam`: visualise results

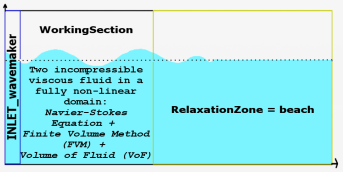
Setup

Like a Physical Wave Tank (PWT), the 2D-NWT is composed of three parts:

➡ **INLET_wavemaker**
non-physically-realistic area where waves2Foam allows different wave types to be generated (regular, irregular, combined) through a `relaxationZone`.

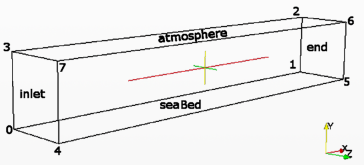
➡ **WorkingSection**
where simulation matches experiment

➡ **BEACH**
non-physically-realistic area where waves2Foam absorbs the incoming wave and prevents from reflection.



Boundaries

2D-NWT = 3D-NWT with one-cell thickness in the third direction, plus two boundaries type `empty` for both left and right sides.



Using classes, each boundary is then defined

	Geometry		Boundaries		
Name	points	*blockMesh*	*alpha.water*	*U*	*p_rgh*
inlet	(0,4,7,3)	patch	waveAlpha	waveVelocity	zeroGradient
seaBed	(0,1,5,4)	wall	zeroGradient	fixedValue (0,0,0)	zeroGradient
end	(1,5,6,2)	patch	zeroGradient	fixedValue (0,0,0)	zeroGradient
atmosphere	(3,7,6,2)	patch	inletOutlet	pressureInletOutletVelocity	totalPressure
left	(0,1,2,3)	empty	empty	empty	empty
right	(4,5,6,7)	empty	empty	empty	empty

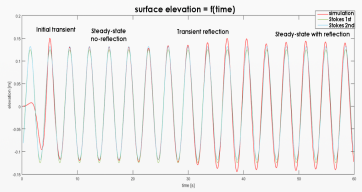
Table 1: Geometry and Boundaries definition

waves2Foam defines the INLET_wavemaker by the keywords: `waveAlpha` and `waveVelocity` allowing wave generation.

Reflection issue

Even in NWT and whatever the length of the beach, some reflections will happen!

A simulation in a 60m long non-optimised NWT shows four distinct behaviours:
➡ Initial transient
➡ Steady-state no-reflection
➡ Transient reflection
➡ Steady-state with reflection



Simulation in a non-optimsed 60m NWT

➡ In this NWT, simulations with differents beach length show that the reflection phenomenon always takes place at the same time (21s). So the reflection occurs at the end of the NWT.

➡ But it goes faster - (2*60-6)/21=5.4m/s - for a 4.3m/s input waves.

Process

The depth is fixed to 2.8m according to Plymouth University Ocean Basin, and the three parameters left to define are:
➡ NWT length
➡ Beach length
➡ Cell-size

First, the cell-size, as it has miror influence on both other parameters.

INLET is fixed to 1m according to previous work - [3] and [4].

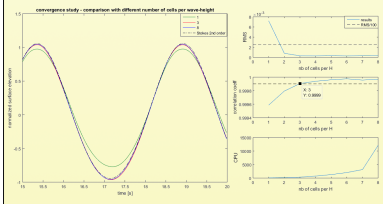
WorkingSection is here fixed to 6m, for the future experiments.

Cell-size & Mesh

➡ Full square cells mesh: better physical representation in both directions

➡ No reflection effect: 60m NWT with a 53m long beach, and only the 20s will be considered (before reflexion occurs).

➡ Stokes 2nd order, T = 3.56s H=0.25m (for a bigger wave)



Comparison of simulations using an increasing number of cells per wave-height

$$RMS < \frac{H}{100} \;\&\; corrcoeff > 0.9999$$

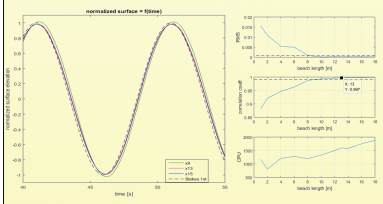
➡ **Cell-size = 3 cells per wave-height**

Beach

The longer the input wave-length, the longer the beach should be. Plymouth University Ocean Basin generates waves from 0.1Hz to 2Hz

➡ **Most restrictive**: 10s period wave.

➡ **Process:** Increase NWT_length with beach_length, to reach compromise



Comparison of simulations using an increasing beach length

$$RMS < \frac{H}{100} \;\&\; corrcoeff > 0.999 \;\&\; \frac{d(corrcoeff)}{d(x)} < 0.001$$

➡ **Beach_length = 13m**

Final NWT

INLET [m]	Working Section [m]	Beach [m]	cells per wave-height
1	7	13	3

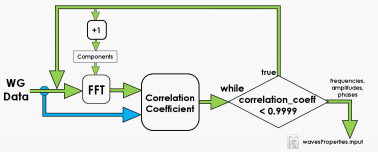
Table 2: NWT characteristics

This poster is tagged with augmented content
To access additional content on your mobile device:
- ensure you have a strong internet connection (Wi-Fi).
- download the **Aurasma**® *app onto your mobile device.*
- Skip making an account.
- Press the **A** *at the bottom centre.*
- search for **pmusi** *using the*
- follow **pmusi's Public Auras**
- Return to the viewfinder by pressing the
- frame images tagged with Aurasma's logo in the app viewfinder.

Application

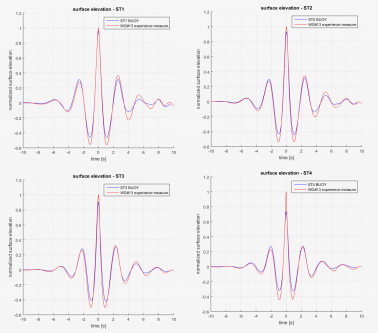
Plymouth University Ocean Basin data:
➡ Time-series of surface-elevation at different position
➡ Extreme events (NewWave)
➡ Time-series of loads
➡ Time-series of model movement
➡ Several models or without

4 cases with increasing steepness and non-linearity were simulated using a wave-gauge (WG) time-series.



From raw experimental data to readable input for INLET_waveMaker (`combinedWave`)

`combinedWave` in `wavesProperties.input` allows a linear addition of Stokes 1st order.



Surface elevation with increasing steepness time-series

The NWT proves its ability to generate any experiment using the surface-elevation time-series as input.

Future development

This NWT is indeed a solid basis for future development:
➡ Go 3D
➡ Fixed structure
➡ Single taunt moored buoy
➡ Multi-body

And as part of WaveDyn development:
➡ Comparison with WaveDyn
➡ Coupling with WaveDyn

OpenFOAM community and its development will be very helpful: a moored buoy, a multi-body floating device, a water-turbine...

http://www.ccp-wsi.ac.uk/

References

[1] N. G. Jacobsen, D. R. Fuhrman and J. Fredsøe, A wave generation toolbox for the open-source CFD library: OpenFoam®, Int. J. Numer. Meth. Fluids, 2012; 70:1073–1088.
[3] T. Vyzikas, E. Ransley, M. Hann, D. Magagna, D. Simmonds, V. Magar, and D. Conley. Integrated Numerical Modelling System for Extreme Wave Events at the Wave Hub Site, in Proceedings of Institue of Civil Engineering (ICE): Coasts, Marine structures and Breakwaters, 18-20 September 2013: Edinburgh, UK.
[4] E. Ransley, Survivability of Wave Energy Converter and Mooring Coupled System using CFD. Phd Thesis, September 2012: Plymouth University, UK.
[5] M. Hann, D. Greaves, and A. Raby. Snatch loading of a single taut moored floating wave energy converter due to focussed wave groups, Ocean Engineering, 2015; 96:258–271.

Acronyms

RMS : Root Mean Square difference between results and theory
corrcoeff : correlation coefficient - MATLAB script
NWT: Numerical Wave Tank
PWT: Physical Wave Tank