THE UNIVERSITY *of* EDINBURGH

A Numerical Modelling Tool for Multibody Wave Energy Converters

Engineering Doctorate (EngD) Thesis

David Ogden

Innosea Ltd.

Declaration

I confirm that this thesis, presented for the degree of Engineering Doctorate, has:

i. Been composed entirely by myself

ii. Been solely the result of my own work except where explicitly stated otherwise by reference or acknowledgment

iii. Not been submitted for any other degree or professional qualification

.....

David Ogden

Date: .....

Abstract

Numerical models of wave energy converters (WECs) have been successfully used since the 1970s to understand a device's characteristics and improve its performance before advancing to costlier, higher-risk stages of development such as tank testing and sea trials. In the last decade several software packages have become available to the industry specifically for time-domain multibody WEC modelling using potential flow theory. One of these tools is InWave, developed by Innosea, which is based on a reduced-coordinate multibody dynamics solver. However, one of the main challenges in developing a WEC modelling tool is the fact that the wave energy sector has not yet converged on a particular technology and there are many different designs currently in development, featuring a wide range of working principles. This thesis presents a novel WEC modelling tool: InWave-HOTINT, which uses a third-party multibody dynamics code (HOTINT) based on a redundant coordinate multibody dynamics method. This approach enables InWave-HOTINT to model a much wider range of mechanical topologies - including WECs featuring closed mechanical loops, multi-DoF PTOs and net mooring systems. The thesis describes the development and verification of the tool, as well as a demonstration of some of the new capabilities via a model based on the Albatern S12 WEC.

Acknowledgements

I have been very fortunate to work with some wonderful people over the past few years and I'm extremely grateful to everybody who has supported me during this EngD. I would like express my sincere gratitude to my primary academic supervisor - Dr. David Forehand, for his support and advice throughout this project. Special thanks to my industrial supervisors: Dr. Maxime Philippe, Adrien Combourieu and Dr. Rémy Pascal for their technical guidance and encouragement. I have greatly enjoyed visiting the Innosea office in Nantes - thanks to everyone who welcomed me there, especially to Samuel Girardin for his programming advice. My secondary academic supervisor, Prof. Lars Johanning, has also provided excellent guidance throughout this project. I would also like to thank Prof. David Ingram, Dr. Morten Kramer and David Findlay for a pleasant viva and for their much appreciated comments.

I would like to acknowledge the excellent work of everybody involved in the development of the open source code HOTINT - thanks in particular to Alexander Dorninger and Simon Weitzhofer for their technical support. Thanks also to Anthony McDonald for his advice on modelling the Albatern devices.

I would also like to acknowledge the efforts of everybody involved in IDCORE - for putting together such a brilliant program and for supporting me to attend the EWTEC 2015 and ICOE 2018 conferences. Thanks also to all the friends I've made along the way, especially the 2014 IDCORE cohort, for making this EngD experience so enjoyable.

I am extremely grateful to my parents, family and friends for the love and encouragement they've given me during this EngD and especially to my wife, Becky, for always believing in me and for being incredibly supportive during this time. Thank you!

Contents

List of Figures

List of Tables

Acronyms

BEM Boundary Element Method

CFD Computational Fluid Dynamics

CAD Computer-Aided Design

CGI Computer-Generated Imagery

CoG Centre of Gravity

DAE Differential-Algebraic Equation

DoF Degree of Freedom

EMEC European Marine Energy Centre

GUI Graphical User Interface

HDB Hydrodynamic Database

IEA International Energy Association

IPCC Intergovernmental Panel on Climate Change

IRF Impulse Response Function

LCoE Levelised Cost of Energy

ODE Ordinary Differential Equations

OES Ocean Energy Systems

OWC Oscillating Water Column

OWSC Oscillating Wave Surge Converter

PTO Power Take-Off

RAO Response Amplitude Operator

RIRF Radiation Impulse Response Function

SPH Smoothed-Particle Hydrodynamics

TCP/IP Transmission Control Protocol/Internet Protocol

WEC Wave Energy Converter

Chapter 1

Introduction

Wave energy is one of the most difficult forms of renewable energy to capture. The ocean is a harsh environment to operate in and the most effective method of extracting energy from the waves is yet to be determined. Numerical modelling is a powerful tool for WEC designers that can help mitigate some of the costs and risks associated with WEC development. This thesis aims to advance an existing numerical modelling tool (InWave) to enable WEC designers to model a much wider range of features - such as complex multibody mechanisms. This chapter explains the context and importance of the work, as well as the main limitations in the field, the project aims and the structure of the thesis.

1.1 Climate Change, Energy & the Potential of Wave Energy

Modern research into wave energy is primarily driven by industrial society's need to rapidly decarbonize its energy supply in response to climatic change. At the 2015 United Nations Climate Change Conference (COP 21) in Paris, world leaders made an historic agreement to "keep a global temperature rise this century well below 2°C above pre-industrial levels and to pursue efforts to limit the temperature increase to 1.5°C" (United Nations FCCC 2015). The recent Intergovernmental Panel on Climate

Change (IPCC) special report explains that limiting global warming to $1.5°C$ will require cutting CO_2 emissions by approximately 45% by 2030 (compared to 2010 levels) - reaching net zero around 2050 (IPCC 2018).

In 2017, 85% of energy[1] consumed in the world was produced from fossil fuels[2] (BP 2018). Hence, holding to $1.5°C$ will require (among other actions) an immediate, large-scale and rapid deployment of renewable energy infrastructure to decarbonize energy production systems (IPCC 2018).

UK energy consumption estimates range from 1640 - 4700 TWh/year (MacKay 2009; Waters 2018), whereas the total practical energy available from all of the UK's renewable resources is estimated to be just 258 - 585 TWh/year (Sustainable Development Commission 2006). Hence, there is a strong argument to be made that countries like the UK should be exploring all of their renewable energy options.

Waves contain large amounts of kinetic and potential energy - the global resource is estimated to be between 18 - 32 PWh/year (Gunn and Stock-Williams 2012; Mørk et al. 2010). It is thought that between 5 - 25% of this energy could be extracted by large arrays of WECs (Cruz 2008; Gunn and Stock-Williams 2012). Hence, wave energy could potentially provide up to 8 PWh/year - about 5% of the world's current energy demand (BP 2018).

The UK has one of the largest wave resources in the world - its Atlantic waves contain an average power of 40 kW per metre of wave crest. With an estimated 670km of suitable seafront, the UK's raw Atlantic wave power resource is roughly 26 GW, or 230 TWh/year (Carbon Trust 2012). Capturing 25% of this energy via large arrays of WECs would produce 60 TWh/year - enough to meet 20% of the UK's electricity demand, or 3.5% of its total energy demand (Department for Business Energy & Industrial Strategy 2017; Waters 2018).

The potential of wave energy has long been recognized. But technical, economical and political hurdles have combined to prevent any devices being deployed at scale: at present there are no wave farms anywhere in the world. Recent R&D programmes have emphasized the importance of revisiting fundamental concepts and working principles

[1] 2017 world energy consumption = 13511.2Mtoe, or 157Pwh. 85% of 157PWh =134PWh.
[2] oil: 34%, coal: 28%, natural gas: 23%.

in order to achieve step-change reductions in the levelised cost of energy (LCOE) (Hurst 2015; Vaughan 2017).

Because testing WECs at sea is expensive and risky, developers commonly use models (both *physical scale models* and *numerical models*) to explore and test new concepts/designs. The modelling of WECs will be critical in delivering the performance step-changes required by current technology development programmes, but there are some significant limitations of current modelling tools which need to be addressed: the following section will explore WEC modelling approaches in more detail.

1.2 Numerical Modelling of Wave Energy Converters (WECs)

1.2.1 Overview

WEC modelling can broadly be split into two categories: *physical scale modelling* (i.e. building a scale model of a WEC and testing it in a wave tank) and *numerical modelling* (i.e. simulating a WEC on a computer by programming the relevant physics). Physical scale modelling and numerical modelling are generally complimentary strands of research in WEC development (Retzler 2015b). Sometimes physical models are the only viable way to investigating highly non-linear behaviour (e.g. extreme waves), which can be extremely computationally expensive with numerical models. But even in typical operational conditions, physical models may reveal unexpected phenomena that can improve a WEC developer's understanding of their device and improve the accuracy of their numerical models (Henry et al. 2014; Lamont-Kane et al. 2015).

However, physical scale modelling can be expensive: typical day-rates for test tanks vary in the range of £2.5-10k/day, with campaigns normally running for 5-10 days (Retzler 2015a). Building the actual model can also be expensive - and although some retrofits may be applied, a physical model is typically more difficult and expensive to modify than a numerical model (Josset et al. 2007). Furthermore, physical modelling suffers from scaling issues and measurement uncertainties which numerical modelling avoids. Hence, numerical modelling can be a valuable tool for WEC developers to test different designs and operating conditions quickly and cheaply (Cruz 2008; Day et al.

2015; Topper and Ingram 2011). Numerical models of WECs have been commonly used for several decades in a variety of applications - from optimizing geometries (Garcia-Tereul and Forehand 2018; Pizer 1994) to designing control systems (Anderlini 2017; Nambiar et al. 2015) and evaluating the performance of different concepts (Babarit, Hals, et al. 2011).

WEC numerical modelling can cover a wide range of topics (such as cost modelling, O&M modelling, physics modelling) in a combined techno-economic model that may be used to optimize a WEC design (Pecher and Kofoed 2017). In this thesis, only the physics modelling of the device will be considered, for which there are a range of different hydrodynamic approaches - with varying computational expense and accuracy (Figure 1.1). Henceforth, the expression 'WEC modelling' will be used to refer to the physics-related numerical modelling of WECs.

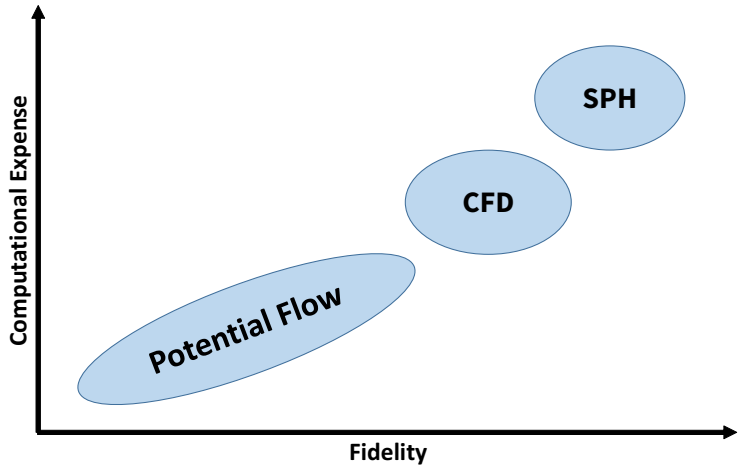


Figure 1.1: WEC hydrodynamic modelling approaches - the speed/accuracy trade-off.

The potential flow section shown in Figure 1.1 is elongated to represent the fact that there are a variety of different potential flow models:

- **Linear potential flow:**[3] the fluid is assumed to be inviscid and the incident flow irrotational and incompressible. In the frequency domain formulation, the fluid forces and WEC characteristics are all fully linear. In the time-domain formulation, mechanical nonlinearities in the WEC system can be included.

- **Partially nonlinear potential flow:** nonlinear Froude-Krylov and hydrostatic

[3]This approach is described in more detail in Chapter 3

forces are included by integrating the fliud pressure over the instantaneous wetted surface of the body.

- **Weakly nonlinear potential flow:** dynamic quantities such as velocity potential and body displacements are expressed as series expansions in order to incorporate higher-order wave-structure interactions.

- **Fully nonlinear potential flow:** the fully nonlinear free-surface and body boundary conditions are satisfied on the exact and instantaneous positions of these boundaries. The boundaries are tracked during the simulation and a new Boundary Value Problem (BVP) is formulated at each time-step.

The different WEC modelling approaches are commonly divided into the following 3 categories: frequency-domain (linear potential flow), time-domain (linear and nonlinear potential flow) and higher-order (weakly/fully nonlinear potential flow, CFD & SPH), as shown in Table 1.1. Folley (2016) states that just selecting the most appropriate approach for a particular WEC concept and modelling objective is one of the key challenges in the field. Linear potential flow theory is the most commonly used approach (Folley estimates that 90% of numerical models of WECs have been based on linear potential flow theory) - popular for its relative accuracy, reliability and speed. Linear potential flow theory has been successfully used to model a wide range of operating conditions for many different WECs (Day et al. 2015; Quoceant Ltd. 2016b; Topper and Ingram 2011) but the modeller must be aware of the limitations of the approach - these will be explained in detail in Chapter 3.

Method	**Applications**	**Available Tools**
Frequency domain	•Large parametric studies	•NEMOH •WAMIT
Time domain	•Power production •Drivetrain/control/mooring design •Structural loads	•InWave •ProteusDS •WaveDyn •WEC-Sim
Higher order	•More accurate power predictions •Validation •System identification •Survivability	•OpenFOAM •SPHysics •Proteus

Table 1.1: Overview of the main categories of WEC model, with typical applications and available tools.

In recent years, several software packages have been developed for the wave energy sector that are dedicated to time-domain WEC numerical modelling and are based on linear potential flow theory, namely:

- InWave (Innosea, France/UK)[4]
- ProteusDS (Dynamic Systems Analysis Ltd., Canada)[5]
- WaveDyn (DNV GL, UK)[6]
- WEC-Sim (National Renewable Energy Laboratory (NREL) and Sandia National Laboratories (SNL), USA)[7]

Access to well-validated, versatile, off-the-shelf modelling codes can help developers to focus on improving WEC designs, rather than devoting resources to developing and verifying in-house numerical modelling software. However, due to limitations of the existing WEC numerical modelling tools, many developers have been unable to use them. This problem could lead to developers having to devote resources to developing their own bespoke modelling software (possibly repeating some work that others have already done), or indeed performing no numerical modelling at all - potentially incurring greater risks at more advanced stages of development (e.g. due to sub-optimal or poorly-understood designs). Given the ambitious targets of the wave energy sector, understanding and overcoming the limitations of existing numerical modelling software packages is critical and will be explored further in the following section.

1.2.2 Current Limitations

The existing time-domain WEC modelling codes described in Section 1.2.1 have so-far focused on modelling simple, open-chain mechanisms comprised of rigid bodies connected by prismatic or revolute joints (Combourieu, Lawson, et al. 2015; Cruz, Mackay, et al. 2013; Lawson et al. 2015; Lucas et al. 2012; Nicoll et al. 2012; Tom, Lawson, and Yu 2015). Most of these codes have inherent limitations in their methodologies, which

[4]Which this thesis advances: `http://www.innosea.co.uk/products/`
[5]ProteusDS webpage: `https://dsa-ltd.ca/proteusds/overview/`
[6]WaveDyn webpage: `https://www.dnvgl.com/services/wave-and-tidal-design-support-5653`
[7]WEC-Sim on github: `https://wec-sim.github.io/WEC-Sim/`

makes it difficult (if not impossible) to model more complex rigid body joints and mechanisms - potentially excluding some WEC developers from using them.

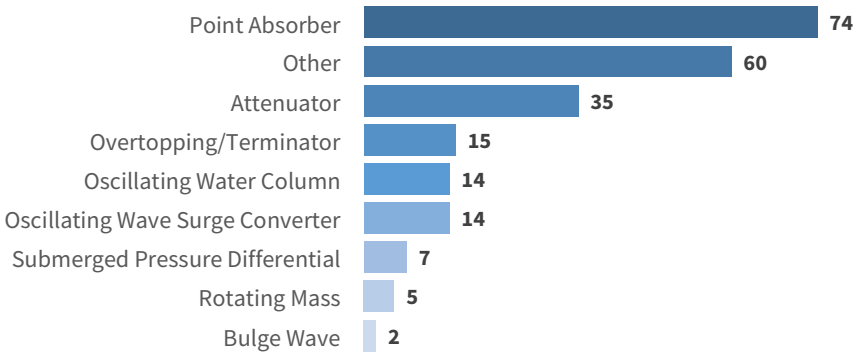


Figure 1.2: WECs currently in development around the world - as classified by the European Marine Energy Centre (EMEC 2017)[8].

InWave, ProteusDS and WaveDyn all use *reduced-coordinate* multibody dynamics algorithms (aka ordinary differential equations/ODE methods) (Combourieu, Lawson, et al. 2015; Lucas et al. 2012; McMillan et al. 1995b; Rongère and Clément 2013), which are known for their computational efficiency, but not their versatility (Baraff 1996). Reduced-coordinate multibody dynamics approaches inherently lack the capability to model non-holonomic constraints (i.e. velocity-dependent constraints such as gears), they do not allow arbitrary sets of constraints to be combined (Baraff 1996), and they do not typically support the modelling of flexible bodies or closed kinematic loops (Shabana 2013) although modified algorithms can be used to model these features (Featherstone 2008; Mukherjee and Anderson 2007). For many WEC developers these are critical features.

An alternative multibody dynamics approach is the *redundant coordinate* method (aka differential-algebraic equations/DAE, redundant-coordinate, absolute coordinate or constraint-based methods). Redundant coordinate multibody dynamics algorithms are widespread across mechanical engineering and computer animation (Erleben 2005; Schiehlen 1997), as they grant the user the freedom to model a wider range of constraints and to combine different kinds of constraints within a system (Baraff 1996). Many general-purpose, industrial multibody dynamics packages such as ADAMS, Chrono, HOTINT, MBDyn use *redundant coordinate* methods (Gerstmayr, Dorninger,

[8]Of the 'Other' category, many devices feature working principles that are awkward to characterize, but some devices categorized as 'Other' simply have not disclosed their working principles. Hence, these numbers should only be considered indicative - more research on active WEC developers is needed.

et al. 2013; Masarati, Morandini, et al. 2014; Schiehlen 1990; Tasora et al. 2015; Wood and Kennedy 2003).

A redundant coordinate multibody dynamics algorithm has been employed by WEC-Sim (via the closed-source MATLAB multibody dynamics software package - Simscape Multibody). However, many redundant coordinate algorithms (including Simscape Multibody) utilize sparse matrix methods when factorizing the system's mass matrix. In other words, Simscape Multibody does not permit the full ($n_{dof} \times n_{dof}$) mass matrix to be modified at each time-step. This becomes problematic for coupling a time-domain potential flow solver[9], where added mass at infinite frequency, $\mathbf{A}(\infty)$, must be added to the system mass matrix at each time-step and is dense; containing off-diagonal terms. WEC-Sim has implemented a workaround to this problem that has been successful for many users, but some modellers have experienced unstable results for certain test conditions.

Furthermore, existing time-domain WEC modelling software packages have tended to develop a tight coupling between the multibody dynamics solver and the hydrodynamic solver, making it difficult to extend the codebases (for example, to include more advanced power take-off (PTO) models (Crozier and Mueller 2017)). Given the existing lack of design convergence within the wave energy sector (as shown in Figure 1.2), and the fact that current technology development programmes are seeking step-change reductions in the levelized cost of energy, LCOE (which may require radical new designs), there is a strong argument that the sector's numerical modelling tools need to be as versatile as possible and use modular, extensible designs.

In summary, several time-domain WEC modelling software packages based on linear potential flow theory have been developed in recent years and have been successfully applied to a range of WEC concepts. But the following limitations still exist:

- With the exception of WEC-Sim, time-domain WEC codes have tended to focus on reduced-coordinate multibody dynamics algorithms, which are inherently incapable of:
  - Combining arbitrary sets of constraints in a multibody model.

[9] **Solver:** a generic term for some software that solves a mathematical problem (such as systems of ordinary differential equations, ODEs, or systems of differential-algebraic equations, DAEs).

  - Modelling non-holonomic (velocity-dependent) constraints (such as gears).
  - Modelling closed kinematic loops (without significant modifications to the algorithms).

- Due to the methodology used by Simscape Multibody, WEC-Sim has encountered issues with including added mass at infinite frequency in the system's mass matrix, and has had to implement a workaround to deal with this problem.
- WEC codes have mainly focused on prismatic and revolute joints - excluding WEC developers whose device contains more complex mechanisms.
- WEC codes have focused on tight couplings, making it difficult to extend their capabilities (e.g. to model more complex sub-systems).

The goal of this thesis is to overcome these limitations by advancing the functionality and capabilities of the InWave code.

Another important limitation of current WEC modelling tools is the inability to model deformable bodies. This is an active area of research with high potential, but in this thesis only rigid-body WECs are considered.

1.3 Aims of the EngD

Main aim: To advance the capabilities of Innosea's in-house WEC modelling software (InWave) and enable the modelling of a wider range of WEC working principles and subsystems.

Supporting objectives:

- Research alternative multibody dynamics modelling approaches,
- Develop new WEC modelling software with the capability to model the following features:
  - A wider range of joint types,
  - Closed kinematic loops,

  - More complex mooring configurations featuring rope-to-rope connections,
  - More complex power take off and control systems.

- Verify the accuracy of the new software against existing codes, using existing test-cases,
- Create new models that demonstrate the new capabilities of the software.

1.4 Overview of the Thesis

1.4.1 Flow Chart

The structure of the thesis is basis on the standard 'IMRAD' structure and is shown in Figure 1.3 to aid the reader.

1.4.2 Methodology Statement

A major component of this thesis is the development of a novel software platform (InWave-HOTINT) for simulating WECs, which should advance the Innosea software package, InWave. At the commencement of the project, InWave had some limitations in modelling complex rigid body mechanisms (e.g. closed kinematic loops), which was attributed to the multibody dynamics solver. However, at the time it was not fully understood by Innosea why this was the case or how to best overcome the limitation. After researching alternative multibody dynamics approaches and determining an appropriate methodology (the redundant coordinate method), there were two options for implementation:

1. Develop and validate a new redundant coordinate-based solver in-house from scratch within the existing InWave framework.
2. Utilize an existing third party, open-source multibody dynamics code (based on the redundant coordinate method) and couple this code to InWave.

Initially, Innosea favoured the first approach, as it was thought that this would lead to

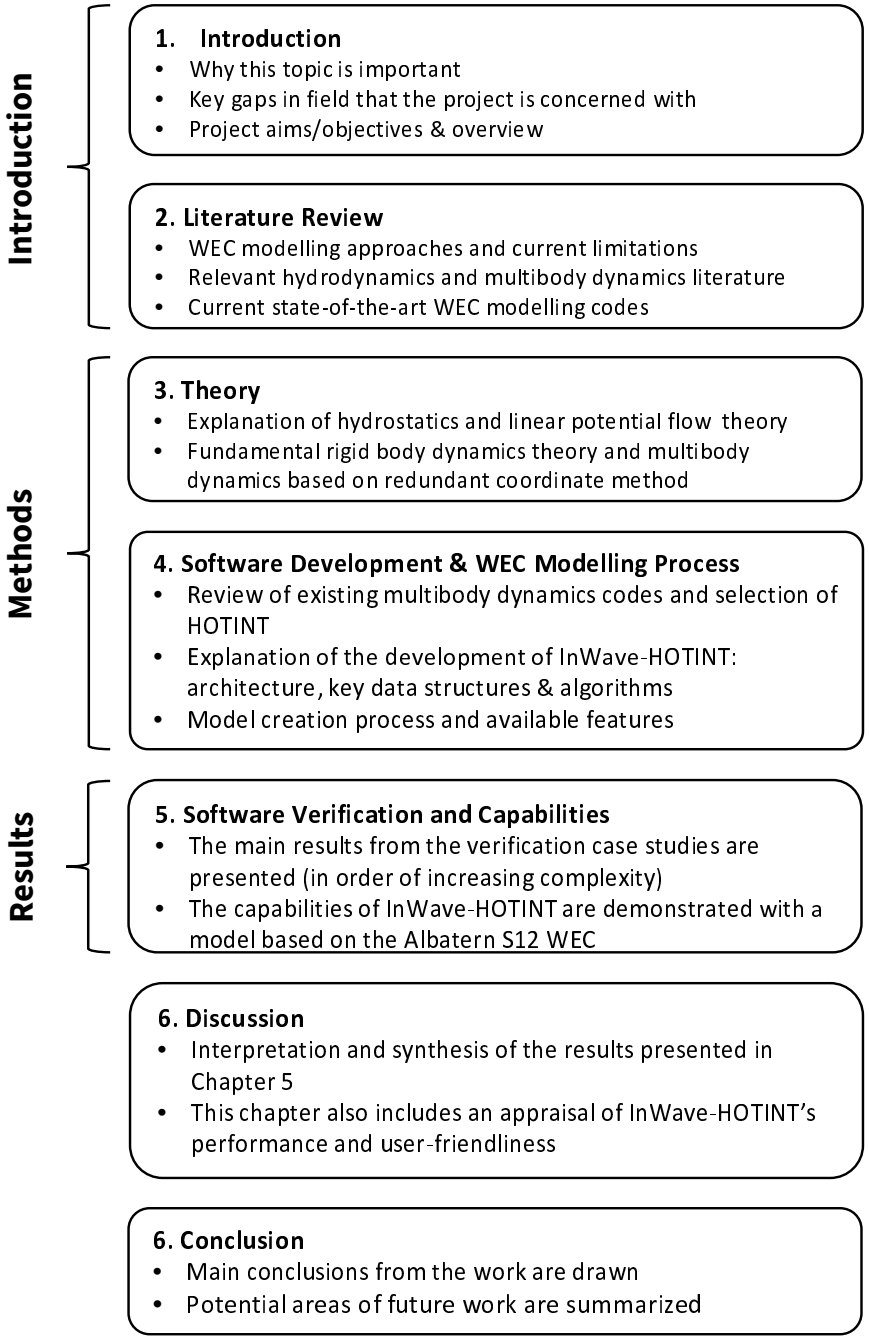


Figure 1.3: Structure of the thesis.

a fuller understanding and higher degree of control of the software, as well as a more coherent product. Hence, initial efforts (from September 2015) focused on developing a redundant coordinate multibody dynamics solver in-house. However, the redundant coordinate method inherently suffers from 'drifting' errors - where small integration errors build up over time and cause rigid bodies in a multibody system to drift part. A correction module was implemented to prevent this from happening, which worked well for simple multibody systems but led to instabilities in the total energy[10] of more complex systems. From the literature, it was identified that the most accurate solution was to use a more advanced integration routine (Such as those developed by Gerstmayr and Stangl (2004), Masarati, Lanz, et al. (2001), and Negrut (1998)), but it was clear that this would take time to implement, and that there were existing open-source codes available that had already successfully implemented these approaches. Furthermore, many of the open-source codes had already developed libraries of elements, constraints and controllers that matched well with Innosea's future ambitions. As a result, by April 2016 Innosea were persuaded (with supporting evidence-based arguments) to focus on utilizing an existing third-party, open-source multibody dynamics solver - not only to avoid repeating existing work but also to capitalize on many developer-years of stability and performance improvements.

Another significant component of the thesis is the verification of the software; checking the accuracy of the results. Although presented as two separate chapters, this was actually performed in parallel with the development of the software. Previously-tested models based on a point absorber, an oscillating wave surge converter (OWSC), and a floating OWSC were turned into test cases that could be quickly checked every time the code underwent significant development/modification. The user-friendliness of the code was also continuously checked and improved - early incarnations relied on hard-coding data directly in the C++ source files. User-generated text input files were gradually phased in instead, which were used to perform automated batch runs.

Once all of the test-cases were successfully verified, some of the new capabilities of the code were explored. A new model based on the Albatern S12 WaveNET WEC (McDonald et al. 2017) was developed to demonstrate the InWave-HOTINT modelling procedure, and some of the tool's capabilities, including:

[10] i.e. the Lagrangian; the difference between total kinetic and total potential energy in the system.

- Closed kinematic loops.
- Combination of constraint types (to model PTO as a mixture of ball joint with rotational spring-dampers).
- A net mooring system (using interconnected linear rope spring-dampers).

This model has not been validated against experimental results. Indeed, this would not be possible using only the available published data as there are too many unknown parameters (such as dimensions, centre of gravity locations, rope and PTO stiffness/-damping, etc.). However, some non-dimensionalized data published by McDonald et al. has served as a reference point to qualitatively compare general characteristics of the physical and numerical models, such as relationships between the relative RAOs of certain bodies in the system, and some natural frequencies. By comparing the same RAOs for a range of different system variations, it has also been possible to identify some characteristics of the system. However, it has not been the aim of the project to analyze the device in depth, particularly given that the device's parameters have been estimated. The main purpose of this model is to serve as an example for future InWave-HOTINT users, and to illustrate what is possible with the tool.

Chapter 2

Literature Review

This chapter reviews some of the key contributions to WEC numerical modelling, and how the theoretical approaches first developed in the 1970s have formed the basis of modern academic and industrial WEC numerical modelling codes. These codes have been successfully validated for a range of devices but still have some significant limitations. A fundamental component of each industrial software package is the multibody dynamics solver, hence this topic is reviewed in more detail. The chapter is separated into the following sections:

1. **WEC numerical modelling:** focusing on fluid-structure interaction and the establishment of linear potential flow theory as the most popular approach used in WEC modelling (this justifies the focus on linear potential flow methods throughout the rest of the chapter).

2. **Multibody dynamics:** the field is reviewed more generally, taking into consideration approaches used in other engineering fields.

3. **Modelling mooring systems:** the main approaches (quasi-static and dynamic) are presented, as well as an explanation of how a linear approximation can be justified under certain conditions.

4. **Time-domain WEC numerical modelling:** state-of-the-art software packages are reviewed, and a more detailed review of InWave is provided. Models from Maynooth University (providing a comparison of different multibody dynamics approaches) and Pelamis Wave Power (demonstrating how WEC numerical models have been used in industry) are also reviewed despite not being publicly available.

2.1 WEC Numerical Modelling

This section focuses on the fluid-structure interaction (FSI) aspect of WEC numerical modelling. There are several strategies available, with varying computational expense and accuracy (as shown in Figure 1.1).

To the author's knowledge, linear potential flow theory was the only approach used in WEC numerical modelling from the 1970s until 1997, when the first fully non-linear potential flow WEC model was developed (Clément 1997; Folley 2016). Increasing computational power has made non-linear hydrodynamic approaches (such as fully non-linear potential flow theory, CFD and SPH) more feasible for WEC modellers in recent years[1]. However, linear potential flow theory is still the most commonly used approach in WEC numerical modelling. Folley (2016) estimates that over 90% of WEC numerical models have been based on linear potential flow theory.

Linear potential flow theory is also the standard approach used in time-domain industrial WEC modelling tools, which typically utilize frequency-domain hydrodynamic coefficients generated by a boundary element method (BEM) code as an input. The key developments that have led to this paradigm are reviewed in the following section. The theory will be explained in more detail in Chapter 3.

2.1.1 The Establishment of Potential Flow Theory in WEC Numerical Modelling

Shortly after Salter (1974) prompted renewed interest in wave energy, the fundamental theory of wave energy conversion was developed independently by several researchers

[1]The release of open-source software such as the CFD code, OpenFOAM (Jasak et al. 2007) has also made high-fidelity models more feasible.

(Folley 2016). These early approaches established the use linear wave theory[2] (Figure 2.1) to analyze a device's performance over a range of frequencies (Evans 1976). Early experimental validation of frequency-domain linear potential flow models are presented by Count (1978) and Mynett et al. (1979), who compared numerical results against physical model results of the Edinburgh Duck - demonstrating the accuracy of the linear potential flow approach over a range of frequencies. Numerical methods then exploited rapidly increasing computational power over the following decades.

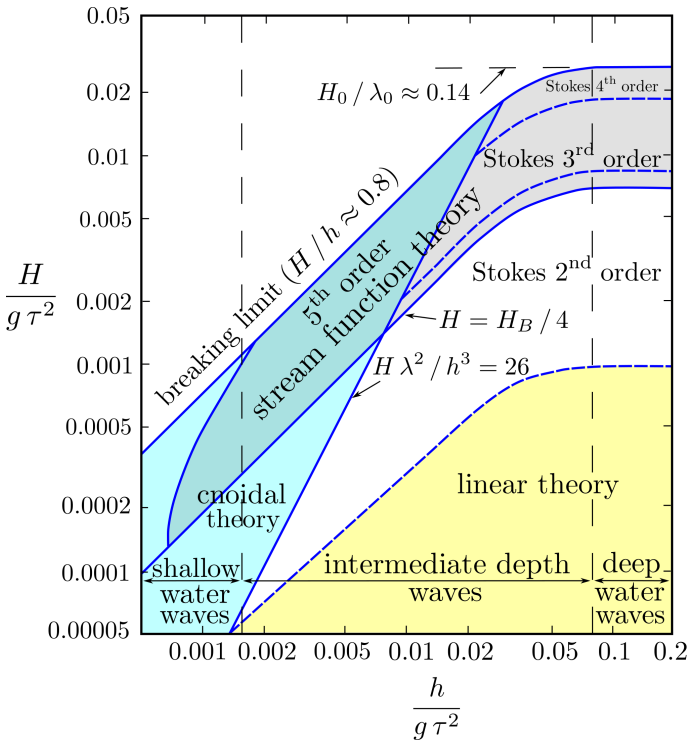


Figure 2.1: Limits of various wave theories (Le Méhauté 1976)[3]. H: wave height. τ: wave period. h: water depth.

Pizer (1994) demonstrated the power of linear frequency domain modelling for optimizing WEC designs by performing parametric geometrical variations of the Edinburgh Duck model[4]. Pizer developed this code as part of his doctoral studies - around the same time that MIT researchers Newman (1985) and Lee (1988) also developed numerical methods for linear potential flow theory, which ultimately led to the development of WAMIT (Wave Analysis at Massachusetts Institute of Technology); a free

[2] Also known as Airy wave theory.

[3] Represented here in the popular adapted form created by Wikipedia user, Kraaiennest.

[4] Pizer states that for a duck model with 320 panels, computing hydrodynamic coefficients for 30 wave frequencies and 19 wave angles for one model would take 8 hours (on a 15 MIPS computer).

surface radiation/diffraction code[5] first released in 1987. Panel methods address two problems: the *diffraction* problem describes the way a body distorts the wave field by its presence and the *radiation* problem describes the waves generated by a body as it moves in the fluid. Each of the body's 6 degrees of freedom (surge, heave, sway, roll, pitch, yaw) has an associated radiation potential (McCabe 2004).

The basis of the panel method is a form of Green's theorem where the velocity potential at any point in the fluid is represented by surface distributions of singularities over the boundary surfaces (Lee and Newman 2005). The submerged surface of a floating or submerged body is represented by panels and each panel represents a fluid source (singularity), which contributes to the flow over the surface. The Green functions determine the influence of a particular panel on the rest of the panels. Essentially, BEM codes produce two types of coefficients to describe the hydrodynamic forces on each body. Firstly, added mass and damping coefficients represent wave radiation effects. Secondly, wave excitation force and moments represent wave diffraction effects. These coefficients and forces can then be used to determine the motion of the body - typically with a response amplitude operator, RAO (McCabe 2004), which is effectively a transfer function that describes the response of the body in a given DoF for a particular wave frequency.

The importance of time-domain models for WECs was recognized as early as Meir (1978), who discussed the potential of a time-domain model to capture some non-linear characteristics of an oscillating water column device. Jefferys (1984) states that non-linear power conversion subsystems such as air turbines and hydraulic pumps necessitate the use of time-domain models, and explains how well-established methods from naval architecture (Cummins 1962) can be used to represent hydrodynamic forces in a time-domain model. Jefferys (1984) explains the method of deriving radiation impulse response functions (RIRFs) from frequency domain radiation damping coefficients, and convolving the resulting RIRF time series with the WEC system's velocity histories to obtain a radiation damping force in the time-domain. This approach is applied to a single degree of freedom (dof) oscillating water column (OWC) model by Jefferys, who also goes on to develop an alternative technique - replacing the convolution integral with an approximate ordinary differential equation (ODE), to reduce

[5] Also known as a 'BIEM/BEM (Boundary Integral Equation Method/Boundary Element Method) code' or 'panel method' code.

computation time to $1/6^{th}$. This is one of the first instances of a *state-space* approach being used to model hydrodynamic forces in a WEC model (Kurniawan et al. 2011) - it has become a valuable approach to reducing computation times in WEC modelling (Folley 2016; Forehand et al. 2016; Tom, Lawson, and Yu 2015).

The panel method is now widely used in many ocean engineering applications to determine the hydrodynamics of complex offshore structures. The commercial BEM code WAMIT has proved very popular in the wave energy community - either for solely performing frequency-domain analysis of a device (e.g. Cruz and Salter (2006) and Payne et al. (2008)), or as a pre-processor for time-domain models (e.g. Forehand et al. (2016) and Mackay et al. (2012)). The success of WAMIT has also led to similar BEM codes being used to model WECs, such as:

- AQWA: e.g. a point absorber WEC by Pastor and Liu (2014)
- NEMOH: e.g. a deformable WEC by Babarit, Singh, et al. (2017)
- WADAM: e.g. a two-body heave WEC by De Andrés et al. (2013)

Penalba et al. (2017) provide an overview of several BEM codes, and a more detailed comparison of WAMIT and NEMOH in the context of WEC modelling. WAMIT is shown to have superior performance, but NEMOH (and the NEMOH-based OpenWARP) is the currently the only open-source BEM code available.

2.1.2 Multibody Hydrodynamics with Linear Potential Flow Theory

Wave interactions between multiple floating bodies is an area of interest across several fields of science and engineering and extends back at least to Ohkusu (1969), who was motivated to understand the behaviour of catamarans and floating stations. One of the first papers investigating multibody hydrodynamic interactions for wave energy was Srokosz and Evans (1979), who expanded the cylinder theory presented by Evans (1976). Newman (2001) discusses some of the first papers to use panel methods to analyze two independent bodies (i.e. a WEC array); stating that computational limits around 1980 restricted the work to simple shapes with coarse meshes. As computational power increased, the panel method was applied to more complex multibody

problems - such as floating offshore bases (Lee and Newman 2000). In WAMIT and NEMOH, the radiation potential is decomposed into components that correspond to a particular mode of one body, while the other bodies in the system are kept stationary (Lee 1995). In this way, the total radiation potential consists of $6 \times n_{body}$ components (i.e. the total number of DoFs in the system). As each DoF interacts with each other DoF in the system, there is a total of $(6 \times n_{body})^2$ radiation interactions in the system. The numerical complexity of this approach is a function of the square of the number of unknowns[6] and has been highlighted as a potential limitating factor in the applicability of panel methods towards modelling WEC systems consisting of many bodies (Flavia 2017; Penalba et al. 2017).

Although more efficient methods have been developed for computing wave interactions within arrays of WECs (Flavia 2017; Folley et al. 2012; McNatt 2015), the direct panel method is still the most common approach for modelling multibody wave interactions for individual multibody WECs and small WEC arrays (i.e. < 10 bodies). Examples of the panel method being used to model multibody WECs can be seen in Beatty et al. (2015), Cândido and Justino (2011), Forehand et al. (2016), Ó'Catháin, Leira, and Ringwood (2007), Paparella (2017), and Taghipour et al. (2008) - in these cases using WAMIT.

2.1.3 Mesh Creation for Linear Potential Flow Solvers

Mesh creation is a fundamental aspect of many areas of computer-aided engineering and an extensive area of research - Frey and George (2008) provide a comprehensive overview of different mesh generation methods. For WEC modelling with BEM codes, the mesh is a critical component - providing a discrete representation of the surface of each rigid body[7] in the system. As the mesh is an approximation of the body's surface, there is a trade off between accuracy and computational expense that must be taken into consideration by the modeller.

Fundamentally, the WEC's mesh is created by representing each body's surface as a grid of coordinate points (nodes), and joining these points together with lines to create

[6] In NEMOH the number of unknowns is equal to the number of panels used in the model.
[7] As only the body's surface is meshed, 3D mesh elements will not be considered in this section.

a mesh. Grids can be 'structured' or 'unstructured' - a detailed explanation of the two approaches is available in Peyret (1996) and Frey and George (2008), and a representative illustration is included in Figure 2.2.

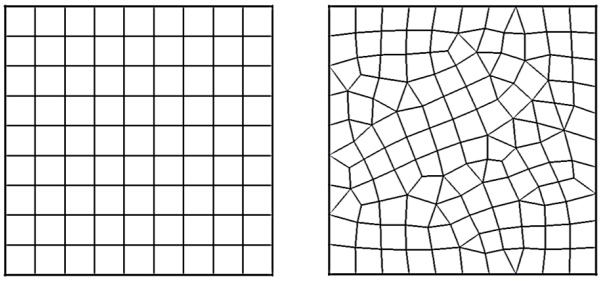


Figure 2.2: **Left:** example of structured mesh. **Right:** example of an unstructured mesh (created in Gmsh).

Unstructured grids are typically better suited to modelling complex geometries, where it is difficult to apply a regular grid to the body's shape (Frey and George 2008). However, unstructured grid based solvers typically use data-dependent memory access patterns (as nodes may be arbitrarily distributed in the grid), whereas structured grid-based solvers can exploit regular/fixed memory access patterns to reduce memory latency (Corrigan 2009; Corrigan et al. 2011). Although structured grid based solvers using this technique may have improved performance, it is unclear if WAMIT and NEMOH support this functionality. Nonetheless, the code's main developers typically favour the use of structured grids (Babarit, Hals, et al. 2011; Newman and Lee 1992).

To improve the accuracy of a mesh, the most straightforward way is to simply increase the number of panels - Babarit, Hals, et al. (2011) demonstrate the effect of this approach on the convergence of hydrodynamic coefficients for a range of WECs (Figure 2.3).

However, increasing the number of panels in the mesh increases the computation time. Newman and Lee (1992) show how a WAMIT model using a structured grid with *cosine spacing* between nodes can improve accuracy with less impact on performance by only refining the mesh near weak singularities in the mesh, such as sharp corners (See Figure 2.4).

For creating surface meshes to be used with the panel method, there are two main kinds

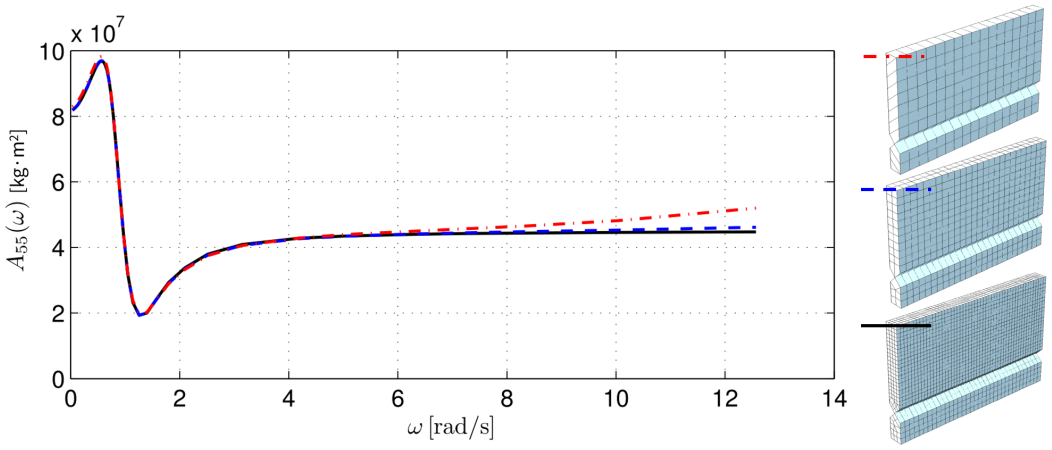


Figure 2.3: Example of the how the number of mesh panels (n_{panels}) can impact the convergence of hydrodynamic coefficients in a BEM model. Example shown is of the added mass in pitch (A_{55}) for an oscillating wave surge converter (Babarit, Hals, et al. 2011).

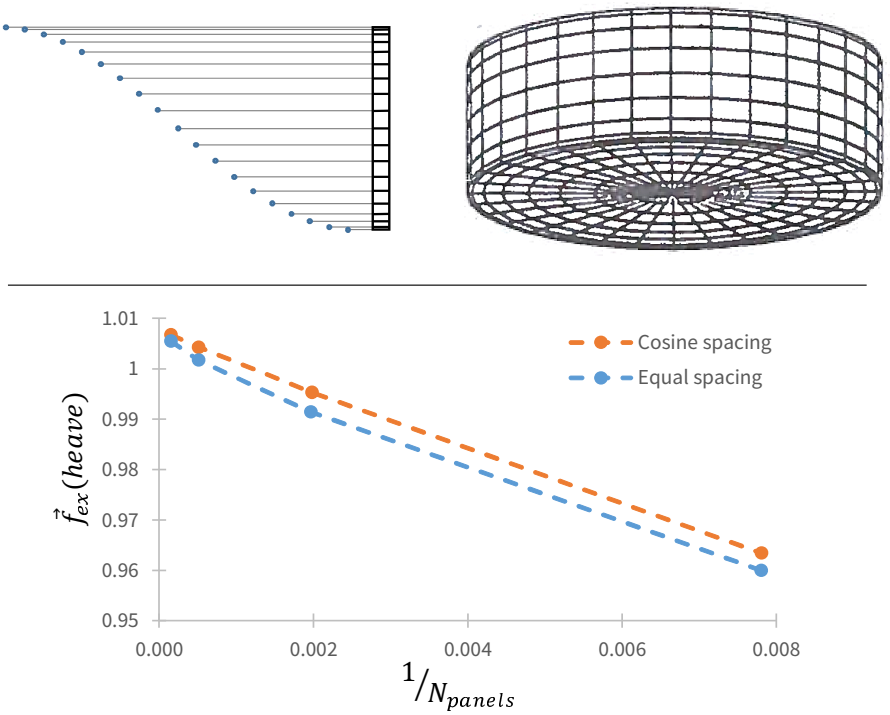


Figure 2.4: **Top left:** illustrative example of how nodes are distributed in a structured mesh using cosine spacing. **Top right:** example of a structured mesh using cosine spacing. **Bottom:** non-dimensional heave excitation force acting on a cylinder - one mesh using cosine spacing near the cylinder's edge and the free surface, the other using equal spacing (with computation points at $N_{panels} = 128$, 512, 2048 and 8192). Adapted from Newman and Lee (1992).

of 2D elements that are of interest: triangular and quadrilateral elements. Quadrilateral elements are arguably more widely supported - indeed, NEMOH only supports quadrilateral elements[8] (Babarit 2017a). NEMOH supports the node orientation shown in Figure 2.5, where the panel is viewed from inside the fluid domain (Babarit 2017a). The correct order of the nodes can be confirmed using the right-hand rule; the normal of each mesh element should point away from the body (i.e. into the fluid domain).

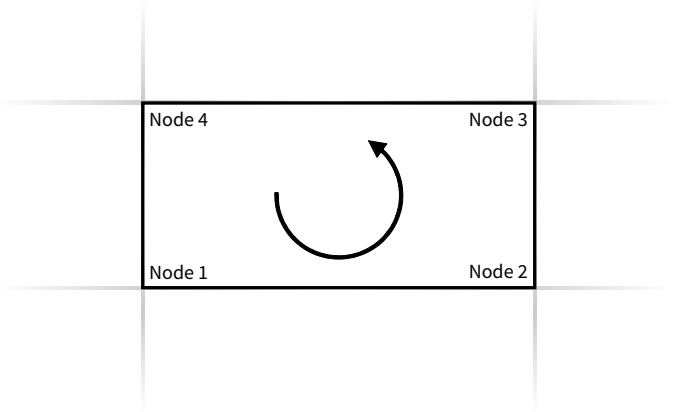


Figure 2.5: A 2D quadrilateral mesh element showing the node ordering convention (when viewd from inside the fluid domain) supported by NEMOH (Babarit 2017a).

The panel method only requires the mesh of the submerged part of the body - for NEMOH, the mesh must be 'cut' at the waterline before running the main program (Babarit 2017a). This can be performed easily with MeshMagick (part of the the NEMOH suite), which can also be used to determine the body's displacement and inertia tensor, $\mathbf{I}$, about a defined centre of gravity (CoG) position, as well as the hydrostatic equilibrium position ($\vec{s}_{eq}$) and hydrostatic stiffness matrix ($\mathbf{K}_H$) (Babarit 2017b). Another useful purpose of the mesh is simply for visualizing the system, which can potentially reveal useful information from time-domain simulations (Quoceant Ltd. 2016a) - as well as enabling obvious errors to be detected and resolved quickly as the model is being developed (Crozier and Mueller 2017).

[8]However, triangular elements can be represented as quadrilateral elements by placing 2 vertices in the same position (known as a degenerate mesh) - this can be a useful workaround for using STL meshes in NEMOH.

2.2 Multibody Dynamics

Multibody dynamics is a broad, active area of research into systems of rigid and/or flexible bodies with constraints (e.g. joints, hydraulics, gears), and has applications across many fields of engineering - significant contributions have come from the vehicle, space and robotics sectors (García de Jalón and Bayo 1994). Some critical theoretical aspects of multibody dynamics are explained in Chapter 3, but a brief review of the main approaches and developments is included here.

2.2.1 Fundamentals

Modern multibody dynamics solvers trace their roots to the work of Lagrange, who established two approaches for the systematic analysis of multibody systems (Lagrange 1811): Lagrange's equations of the *first* and *second* kind.

Lagrange's *equations of the first kind* are a set of differential-algebraic equations (DAEs) - where each rigid body in the system retains 6 degrees of freedom regardless of how it is constrained[9], and a set of algebraic equations describe the constraints in the system (Strauch 2009):

$$\frac{\partial L}{\partial \vec{s}_i} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\vec{s}}_i} + \sum_{i=k}^{n_C} \lambda_k \frac{\partial C_k}{\partial \vec{s}_i} = 0 \qquad \text{for } i = 1, \ldots, n_{body} \tag{2.1}$$

Where i labels each body in the system, and k labels each constraint. L is the 'Lagrangian'; the difference between kinetic energy, T, and potential energy, V ($L = T - V$). $\vec{s}$ is the multibody system's position vector. There is a Lagrange multiplier, λ_k, for each constraint, C_k, in the system (where n_C is the number of constraint equations). Typically, the DAEs of a multibody system are index-3, which means that the algebraic constraints equations have been formulated at the position level[10] (Führer and Leimkuhler 1991; Laulusa and Bauchau 2008).

[9] Hence, this approach is also commonly referred to as a 'redundant coordinate' approach.

[10] i.e. They describe the joints in the system in terms of their connection points on each body.

Lagrange's *equations of the second kind* reduce the equations of motion to the smallest possible number by using generalized coordinates to express the system's degrees of freedom[11]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_j} = \frac{\partial L}{\partial q_j} \tag{2.2}$$

Where q denotes a set of generalized coordinates, which describe the multibody system in terms of the minimum amount of degrees of freedom required (i.e. there are no 'redundant' degrees of freedom considered in this approach). This leads to a set of ordinary-differential equations (ODEs) describing the dynamics of the multibody system (Führer and Leimkuhler 1991; Rosenberg 1977; Strauch 2009).

As Section 2.2.2 illustrates; a large variety of multibody dynamics formalisms have been developed since the work of Lagrange, but as Baraff (1996) argues: "ultimately we are faced with a basic choice. Either we model constraints by reducing the number of coordinates needed to describe the system's state, or we introduce additional forces to maintain the constraints".

2.2.2 Brief History of Multibody Dynamics Developments

García de Jalón and Bayo (1994) explains how Lagrange's equations of motion were limited to the analysis of very simple systems until the 1950s, at which point scientists and engineers working in the space and robotics industries (typically working on open-chain/tree-like systems) started to harness digital computing power. The following decades saw the development of several reduced coordinate algorithms including Denavit and Hartenberg (1955), Featherstone (1983), and Liegeois (1977). As explained by Featherstone (2008), reduced coordinate algorithms typically have $O(n)$ complexity (i.e. computational time grows linearly with the number of bodies in the system), which makes them well suited to modelling systems with a large number of bodies (Featherstone 2008). Featherstone algorithms have also been applied to offshore applications - being used to model underwater robots by Hosseini et al. (2006) and McMillan

[11] As this approach leads to a minimum set of set of equations, algorithms based on this approach are also referred to as 'reduced coordinate' or 'relative coordinate'.

et al. (1995a), which has arguably increased confidence in their applicability for WEC applications (see Section 2.4).

Although reduced coordinate solvers have been popular for modelling multibody systems since the 1950s, García de Jalón and Bayo (1994) explain how the 1980s saw growing demand from the vehicle industry to model systems with closed loops (e.g. suspension) and non-holonomic (velocity-dependent) constraints (e.g. gears), which is difficult to do with conventional reduced coordinate methods (Baraff 1996; Featherstone 2008). As computational power was increasing rapidly, interest in DAE approaches started to grow - Schiehlen (1990) shows 3 codes (out of a review of 20 codes) based on Lagrange's equations of the first kind. One of these codes is ADAMS, which has since become the most widely used multibody dynamics solver in the world today (MSC 2019), whereas most of the other codes in the Schiehlen review have been discontinued. The versatility of the DAE approach may have contributed towards ADAMS' success as a general-purpose multibody dynamics code. However, early implementations of the DAE approach suffered from a performance penalty as the algorithms had $O(n^3)$ complexity; the computational effort grows cubicly with the number of bodies in the system[12]. Featherstone (2008) compares the performance of $O(n)$ algorithms (articulated body algorithm, ABA and recursive Newton-Euler algorithm, RNEA to an $O(n^3)$ algorithm (composite rigid body algorithm, CRBA - see Figure 2.6).

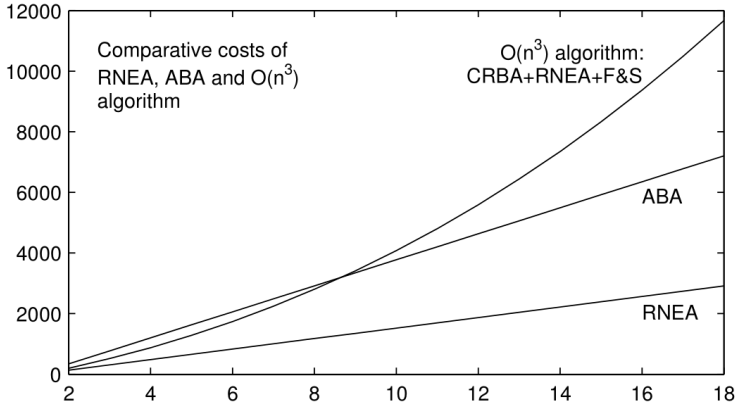


Figure 2.6: Number of floating point operations versus number of bodies for various multibody dynamics algorithms (Featherstone 2008).

The growth of the visual effects industry in the 1990s further enhanced the popularity

[12]More accurately, these algorithms have $O(nd^2)$ complexity, where d is the depth of the connectivity tree (Featherstone 2008).

of DAE approaches, with Baraff (1996), Barzel and Barr (1988), and Witkin et al. (1990) favouring DAE approaches not just for their greater functionality, but for their ability to bypass the awkward (sometimes impossible) process of parameterizing a multibody system's degrees of freedom. Baraff (1996) in particular made a significant contribution to the field: demonstrating that DAE approaches can also achieve $O(n)$ complexity by exploiting the fact that the system's matrices are sparse.

Accuracy is more critical in scientific and engineering applications than in the visual effects industry. But as discussed by Ascher et al. (1995) and Negrut, Rampalli, et al. (2007), index-3 DAEs are very difficult to solve directly, which can affect the accuracy of the results. Small numerical errors may accumulate over each time step and manifest as 'drift' in the simulation results - where either the bodies in the multibody system literally drift away from each other, or the total energy in the system becomes unstable. Bauchau and Laulusa (2007) and Laulusa and Bauchau (2008) explain several common solution strategies, including:

- Maggi's formulation
- Index reduction
- Constraint violation stabilization
- Constraint violation elimination

The most common method encountered in the literature is 'index reduction', and this will be described in more detail in Chapter 3[13]. Index reduction techniques are typically combined with constraint violation stabilization techniques (Ascher et al. 1995; Baumgarte 1972) and/or constraint violation elimination techniques (Flores 2015; Nikravesh 2007) within the same solver in order to overcome the problem of numerical drift. Schmitt et al. (2005) present a comparison of several solution strategies for index-3 DAEs, concluding that reducing to index-2 DAE formulations (i.e. formulating the algebraic constraints equations at the velocity level) and using high-order explicit integration methods can dramatically reduce the effects of positional and energy drift. The effectiveness of these strategies is improved if they are used in conjunction with accurate integration methods Schmitt et al. (2005).

[13] For succint explanations of the other solution strategies, the reader is referred to Laulusa and Bauchau (2008)

Key contributions to the integration of index-3 DAEs for engineering multibody dynamics applications have been made by in recent decades by Masarati, Lanz, et al. (2001), Gerstmayr and Stangl (2004) and Negrut, Haug, et al. (2003), who investigated high-order integration routines and whose work has led to the development and release of the popular open-source multibody dynamics codes MBDyn, HOTINT and Chrono, respectively. Furthermore, Gerstmayr (2009) states that the redundant coordinate approach used by HOTINT can achieve the same performance as a reduced coordinate $O(n)$ code for open-chain multibody systems, without having to factorize the system's mass matrix.

2.3 Modelling Mooring Systems

The mooring system of a floating WEC is a critical subsystem, the design of which is beyond the scope of this project. Dedicated mooring design software exists, but it is important that WEC numerical modelling tools can interface with these packages, or else be extensible enough to provide the same functionality in future iterations/releases. Hence, this section will review the main approaches to modelling mooring systems, of which there are 3 categories (Davidson and Ringwood 2017):

- **Passive mooring**: station-keeping only; limited influence on the WEC's power extraction (e.g. Slack moored devices such as Pelamis).
- **Active mooring**: station-keeping *and* has significant influence on the WEC's power extraction (e.g. the Desalination Duck device).
- **Reactive mooring**: provides reaction force for the WEC to extract power from the waves. Especially suited when the PTO absorbs power from relative movement between WEC and seabed (e.g. the Laminaria device).

Mooring systems are typically nonlinear - as illustrated in Figure 2.7 by the load excursion curve for a single catenary mooring line (Fitzgerald and Bergdahl 2008). The figure shows how the line tension can increase dramatically if the catenary lifts off the sea bed and becomes fully taut.

A review of the different modelling approaches used for WEC mooring systems has

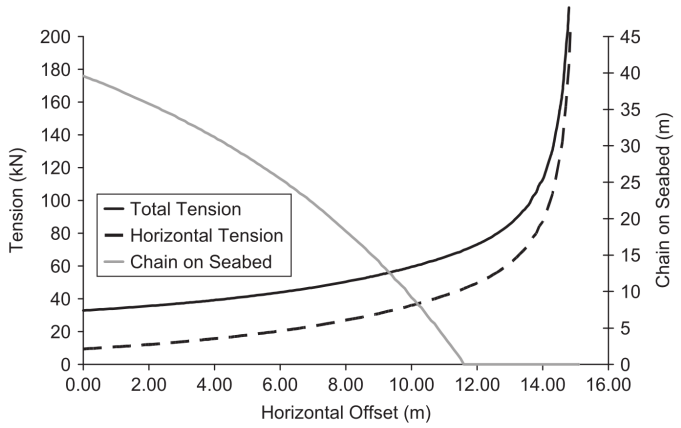


Figure 2.7: Example of a catenary mooring line's nonlinear tension with respect to horizontal offset (Fitzgerald and Bergdahl 2008).

been undertaken by Davidson and Ringwood (2017). To capture the characteristics shown in Figure 2.7, it is typically necessary to use a fully *dynamic* mooring model. However, under some conditions, a *quasi-static* or *linear* model may be justified. The model's purpose and requirements must be taken into account when determining which approach to use, as the computational expense varies significantly.

2.3.1 Dynamic Mooring Models

To capture the nonlinear dynamic characteristics of the mooring, and the hydrodynamic/inertia loads on the line, a fully dynamic mooring model is typically required (Johanning et al. 2006) for more accurate performance estimates and for mooring design. The most common approaches used in dynamic mooring line models are lumped mass and finite element. Figure 2.8 shows a segment of a lumped mass model, with weight ($\vec{f_w}$) and buoyancy ($\vec{f_b}$) acting on the point mass, as well as transverse and tangential drag loads ($\vec{f_{dp}}$ and $\vec{f_{dq}}$, respectively).

Weight and buoyancy are computed for each rope segment, and the drag forces can be computed using Morison's equation; treating each rope segment as a slender cylinder (represented by the dashed line in Figure 2.8) (Hall and Goupee 2015). A point mass element only has 3 (translational) degrees of freedom - its orientation and inertia are not considered. Hence, a lumped-mass mooring line model cannot accurately model twist in the rope. To capture this behaviour, finite element approaches have been used (with increasing computational expense) (Davidson and Ringwood 2017).

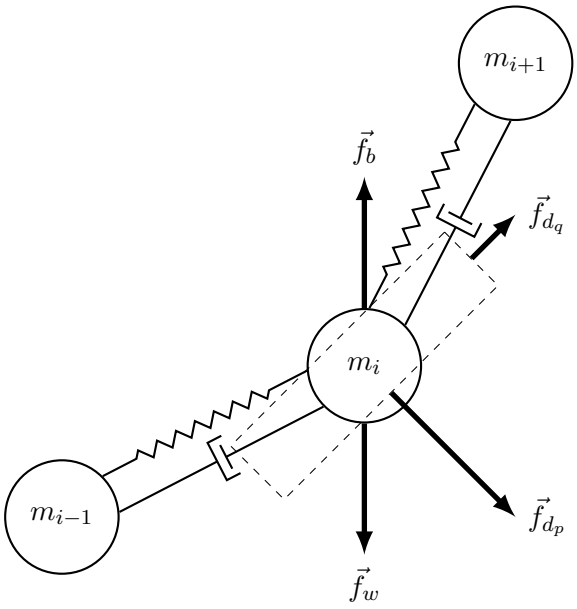


Figure 2.8: A lumped-mass mooring line segment.

2.3.2 Quasi-Static Mooring Models

Under some assumptions, a less computational expensive 'quasi-static' approach may be used in the early design phase to identify suitable materials and estimate line dimensions. This information can then be fed into more advanced dynamic mooring models (Johanning et al. 2006).

In a quasi-static approach, the tension in the line is derived from the catenary formulations and assumes that the line is in static equilibrium during each time step (i.e. the loads are assumed constant over the time step) (Davidson and Ringwood 2017). Hence, hydrodynamic and inertial forces on the line are neglected in this approach and the line is assumed to retain a catenary form throughout the simulation. Figure 2.9 demonstrates how a quasi-static mooring model can be suitable in some WEC operating conditions (where the mooring line does not become fully taut).

2.3.3 Linear Mooring Models

For evaluating a WEC's performance over a wide range of sea states, it is common to use an even less computationally expensive approach and approximate the mooring force as fully linear (Babarit, Hals, et al. 2011; Davidson and Ringwood 2017; Yu et al.

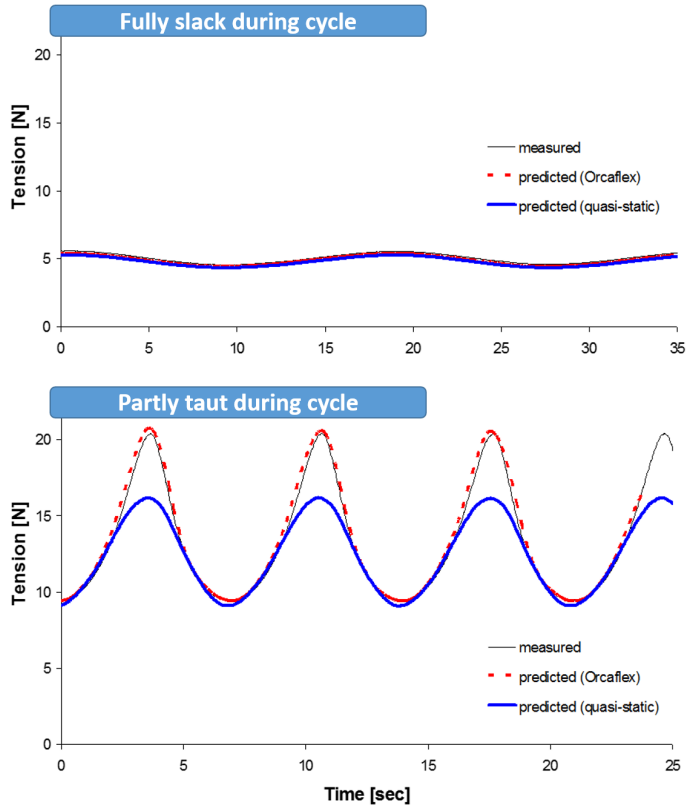


Figure 2.9: Comparison of axial tension time series for experimental, quasi-static and dynamic (Orcaflex) results. **Top:** mooring line remains fully slack during each cyle. **Bottom:** mooring line becomes partly taut during each cyle. (adapted from Johanning et al. (2007)).

2014). Figure 2.10 is based on the nonlinear catenary mooring line curve shown in Figure 2.7, and demonstrates how the line's restoring force can be approximated as linear close to the system's equilibrium state[14] (Fitzgerald and Bergdahl 2008).

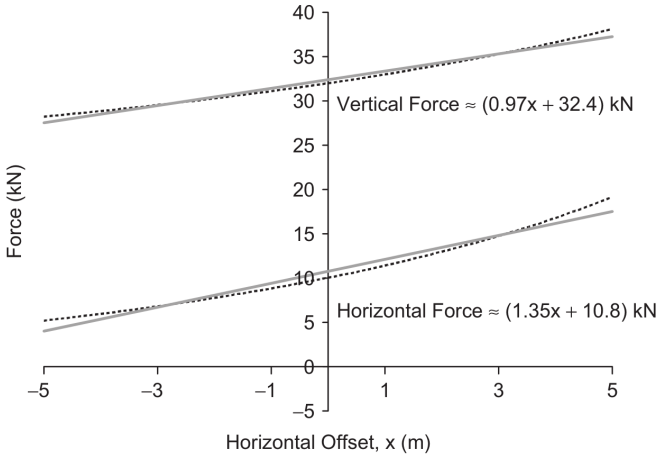


Figure 2.10: Example of how catenary mooring line restoring force behaves approximately linear close to the equilibrium state - based on Figure 2.7 (Fitzgerald and Bergdahl 2008).

This approach is commonly used in fully linear frequency domain models (Babarit, Hals, et al. 2011; Fitzgerald and Bergdahl 2008), but has also been used in time domain models (Combourieu, Lawson, et al. 2015; Ruehl, Michelen, Bosma, et al. 2016). In practice, a linear model is typically implemented with an $(n_{dof} \times n_{dof})$ matrix of stiffness coefficients and an $(n_{dof} \times n_{dof})$ matrix of damping coefficients which correspond to generalized DoFs in the system and can be multiplied with the system's displaced position to apply a restoring force to the appropriate degree of freedom in the WEC:

$$\vec{f}_m(t) = \mathbf{K}_m(\vec{s}(t) - \vec{s}_{eq}) + \mathbf{C}_m(\dot{\vec{s}}(t)) \tag{2.3}$$

2.3.4 Mooring Software Packages

Davidson and Ringwood (2017) present a summary of available software packages for modelling mooring systems (adapted in Table 2.1):

Two open source options are MAP++ and MoorDyn, which use quasi-static and dynamic approaches, respectively. Hence, these packages could potentially be used at

[14] Which is often valid due to the small amplitude motion assumed by linear potential flow theory.

Code	License	Quasi-Static	Dynamic	WEC Dynamics
AQWA	Proprietary	✓	✓	✓
DNV Sesam	Proprietary		✓	✓
FLEXCOM	Proprietary	✓	✓	✓
MAP++	Apache	✓		
MoorDyn	GPL v3		✓	
OrcaFlex	Proprietary		✓	✓
ProteusDS	Proprietary		✓	✓

Table 2.1: Overview of software packages available for modelling mooring systems (adapted from Davidson and Ringwood (2017)).

different stages of the mooring design process.

2.4 Time-Domain WEC Modelling

2.4.1 Bespoke Multibody WEC Models

There are many time domain WEC modelling codes that have been developed - as evidenced by the 25 different participants in the International Energy Association (IEA) Ocean Energy Systems (OES) Task 10 WEC modelling verification project. However, this section will focus on two codes in particular - developed by Pelamis Wave Power and Maynooth University to model specific multibody WECs. The PELs suite provides very useful insights into how state-of-the-art numerical modelling tools have been applied in industry. The Maynooth models investigate both ODE and DAE multibody dynamics approaches and provide some useful insights on the pros and cons of each approach.

2.4.1.1 PELs Suite

Arguably one of the most sophisticated WEC modelling software packages was developed by Pelamis Wave Power over the 17 year lifetime of the company. The software includes non-linear 2D hydrodynamics, 3D multibody diffraction and radiation, finite element mooring models, and detailed power take-off and control models. A sum-

mary of each module, with approximate CPU run times[15] and typical applications is shown in Figure 2.11. The nonlinear time domain program (pel_nltime) features a non-linear multibody dynamics solver but its theoretical details have not been published (Quoceant Ltd. 2016a).

Main Programmes	Body Dynamics	Hydrodynamics	Control	CPU	Applications
Linear freq domain	linear	3D diffraction 2D strip theory	linear	1s	Large parametric studies
Linear time domain	linear	3D diffraction 2D strip theory	non-linear	30 mins	Control optimization and power absorption
Non-linear time domain	non-linear	2D strip theory	non-linear	2 to 20 days	Power absorption and survivability

Figure 2.11: Overview of the main programs included in the PELs suite (Quoceant Ltd. 2016a).

The PELs suite predominantly used the 'Strathclyde panel code' originally developed by Pizer (1994) to compute hydrodynamic coefficients, and later included support for using WAMIT databases of frequency domain hydrodynamic coefficients. WAMIT has some advantages over other codes such as automatic avoidance of irregular frequencies[16] (Quoceant Ltd. 2016a). PELs can also exploit the slenderness of the Pelamis WEC's geometry by using 2D hydrodynamic coefficients, which can be generated with the Strathclyde code or with PDStrip (TU Hamburg). The 2D approach is faster and therefore well-suited to computationally intensive tasks such as geometry optimization.

Quoceant Ltd. (2016a) show how the PELs suite was used throughout the development of the Pelamis WEC. For example, animations of the device were created to improve understanding of the device's behaviour and characteristics. Predicted time series of loads on the structure were used in the structural design process and the numerical model was also used to optimize the device's control system (Figure 2.12).

The PELs suite has been extensively compared with data from physical model testing carried out by Pelamis, and against full scale data obtained from the P2 prototypes tested in Orkney in 2014. Quoceant Ltd. (2016a) explain how the PELs total power results had better agreement with the full scale results than the physical modelling

[15] For a set of 100 wave spectra representing sea states for an average year, not including computation of hydrodynamic coefficients.

[16] Frequencies at which the boundary integral equations have non-unique solutions, manifesting (incorrectly) as resonance in the system (Liu and Falzarano 2017; McCabe 2004).

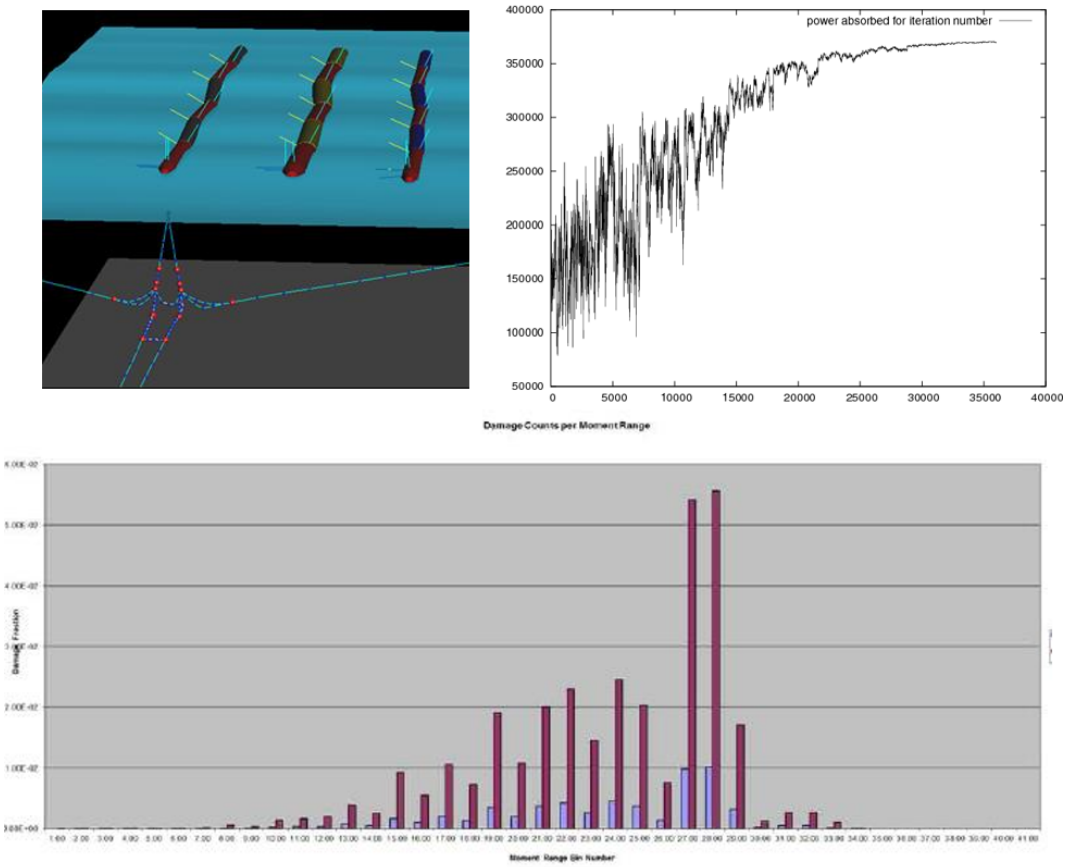


Figure 2.12: Examples of how the PELs suite has been used. **Top left:** device visualization. **Top right:** control optimization based on simulated annealing method. **Bottom:** load histogram (generated from time series) for use in structural design process. (Quoceant Ltd. 2016a).

results - despite the greater complexity of the full scale system. The source of disagreement has not been determined (despite many possible explanations ruled out by dedicated studies) and demonstrates the difficulty of validating numerical models and understanding the results and discrepancies with real-world systems.

2.4.1.2 Maynooth Models

Multibody WEC numerical models have also been developed at Maynooth University - Ó'Catháin, Leira, Ringwood, and Gilloteaux (2008) describe the development of a model using a reduced coordinate multibody dynamics method (Newton-Euler with eliminated constraint forces). For this model, WAMIT is used to compute the frequency domain hydrodynamic coefficients, with a state-space approximation of the convolution integral used to compute radiation forces in the time domain. The model

is validated against physical model data for a 2-body hinged barge.

Paparella et al. (2015) build on the work of Ó'Catháin, Leira, Ringwood, and Gilloteaux - presenting two multibody dynamics approaches: a DAE (redundant coordinates) approach and an ODE (reduced coordinates) approach. Pseudo-spectral methods are applied to both formulations in order to obtain the solution for the dynamics of the 2-body hinged barge system, using hydrodynamic coefficients produced by WAMIT. Paparella et al. conclude that the DAE approach is superior to the ODE approach in terms of accuracy of the computed motion of the hinged barge device, but the ODE approach is faster. Paparella (2017) explains that the DAE approach also provides information regarding the joint reaction forces (the ODE approach does not), which could provide useful inputs to the joint design process.

2.4.2 Off-the-shelf Time-domain WEC Modelling Software Packages

2.4.2.1 ProteusDS

ProteusDS (developed by Dynamic Systems Analysis Ltd.) is a dynamic analysis software package capable of simulating vessels, structures, mooring lines, and other technologies in marine environments (Dynamic Systems Analysis Ltd 2018). In addition to simulating WECs, ProteusDS has also been used to model floating wind turbines (Hall, Buckham, et al. 2011) and tidal energy converters (Bivol et al. 2017) and a range of other offshore systems across the defence, aquaculture and oil and gas industries.[17]

The ProteusDS multibody dynamics solver is based on Featherstone's articulated body algorithm - selected for its efficiency and its previous use in offshore applications (McMillan et al. 1995b; Nicoll et al. 2012). The hydrodynamics solver is based on time-domain linear potential flow theory and must import pre-computed frequency-domain coefficients from a BEM solver such as WAMIT, NEMOH or SHIPMO3D. Non-linear Froude-Krylov and hydrostatic forces are also available.

ProteusDS includes a finite element solver for creating fully dynamic mooring models. Control systems can be modelled either using built-in elements or via Simulink cosim-

[17] A complete list of ProteusDS publications is available at `https://dsa-ltd.ca/about-dsa/publications/`

ulation with the application programming interface (Dynamic Systems Analysis Ltd 2018).

ProteuDS has been validated against WEC physical model results for several devices including the SurfPower WEC (Nicoll et al. 2012) and two concepts developed by the University of Victoria's West Coast Wave Initiative inspired by the WaveBob and OPT PowerBuoy devices (Beatty et al. 2015). Code-to-code verification has also been performed as part of the WEC^3 project (Combourieu, Lawson, et al. 2015) and the IEA OES Task 10 project (Wendt et al. 2017).

2.4.2.2 WaveDyn

WaveDyn (developed by DNV-GL) is a simulation tool designed specifically for time-domain, multibody WEC modelling and performance evaluation. WaveDyn's multibody dynamics solver is based on a proprietary algorithm, and as such has not been described in detail in the literature. However, the method of building up a structural model in WaveDyn describes starting a multibody system with a floating body 'to allow 6 DoF motions of the next body' (Mackay et al. 2012), before constructing the rest of the system by connecting joint elements, which may be translational or rotational (Cruz, Mackay, et al. 2013; Lucas et al. 2012; Mackay et al. 2012). Hence, the approach described has the characteristics of a reduced-coordinate multibody dynamics solver.

WaveDyn imports frequency-domain hydrodynamic coefficients from an external BEM solver (such as WAMIT or AQWA), which are converted to time-domain hydrodynamic loads on the WEC by proprietary time-domain linear potential flow equations. Non-linear Froude-Krylov and hydrostatic forces are also available. In addition to linear mooring models, WaveDyn supports quasi-static mooring models using look-up tables (typically derived from an external analysis/software), where the mooring force is a function of the WEC's position and velocity (Mackay et al. 2012). Similarly, PTO forces can be applied using look-up tables, as polynomial functions of the joint's displacement and velocity, or by creating a fully dynamic PTO model (Mackay et al. 2012).

WaveDyn has been validated against WEC physical modelling results for several devices including Columbia Power (Cruz, Livingstone, et al. 2011), Pelamis and WaveBob

(Mackay et al. 2012). Code-to-code verification has been performed as part of the WEC[3] project (Combourieu, Lawson, et al. 2015).

2.4.2.3 WEC-Sim

WEC-Sim (developed by Sandia National Laboratories and the National Renewable Energy Laboratory, with support from the US Department of Energy) is an open-source WEC simulation tool developed in MATLAB & Simulink using The MathWorks' proprietary multibody dynamics package Simscape Multibody (formerly SimMechanics). An overview of the code is shown in Figure 2.13.

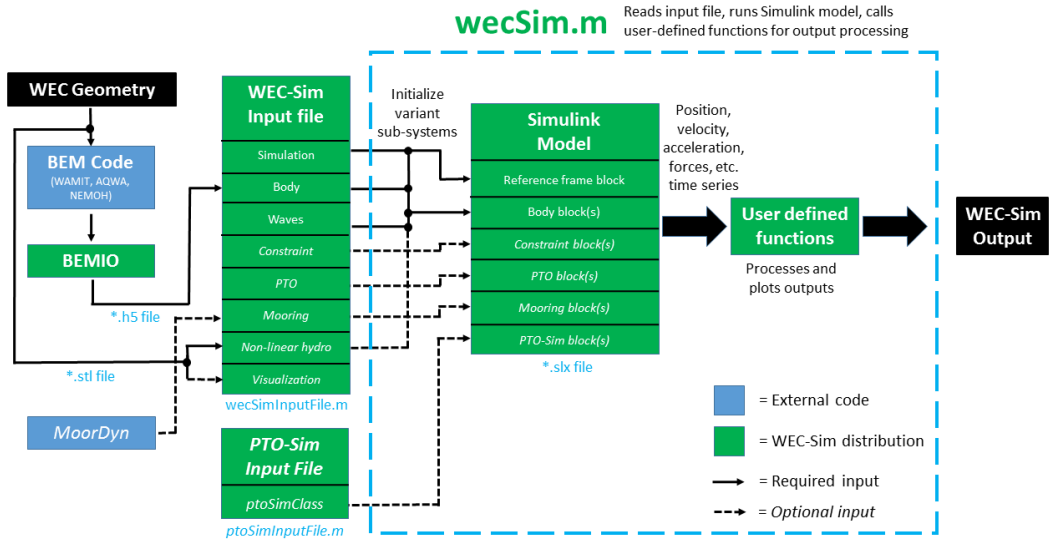


Figure 2.13: WEC-Sim workflow diagram (NREL and Sandia 2015).

Simscape Multibody does not provide official theory documentation, but the original white paper and patent describe the development of both ODE and DAE approaches in the code (Wood 2011; Wood and Kennedy 2003), to offer efficiency and versatility in the same package. Hence, WEC-Sim supports a range of constraint and control elements available in Simscape Multibody and Simulink.

As discussed in Chapter 1, Simscape Multibody exploits the typically sparse nature of a multibody system's mass matrix to improve the solver's performance. The matrix factorization technique is described by Wood (2011). Although beneficial to the solver's overall performance, the technique has presented some difficulties for including the added mass at infinite frequency matrix, $\mathbf{A}_\infty$ (size $n_{dof} \times n_{dof}$) in the system's mass

matrix, **M**. WEC-Sim has implemented a workaround by combining the added mass in each translational dof into one term and adding the combined added mass to the body's total mass at the start of the simulation. The added inertia terms are included in the system's inertia tensor at each time-step[18]. However, this workaround has been found to cause instability for some systems under certain test conditions[19].

In addition to linear mooring models, WEC-Sim also supports dynamic mooring modelling via the lumped mass mooring line package, MoorDyn (Hall and Goupee 2015). WEC-Sim's hydrodynamics solver is based on time-domain linear potential flow theory and supports state-space approximations of the convolution integral for improved speed computing the radiation forces experienced by the WEC. WEC-Sim also includes the 'BEMIO' module for reading/converting pre-computed databases of frequency-domain hydrodynamic coefficients from WAMIT, NEMOH and AQWA. Non-linear Froude-Krylov and hydrostatic forces are also available (Combourieu, Lawson, et al. 2015).

WEC-Sim has been validated against WEC physical modelling results for several devices including the Reference Model 3 (RM3) developed by the US DOE (Ruehl, Michelen, Kanner, et al. 2014), an oscillating surge WEC based on Oyster (Yu et al. 2014) and a floating oscillating surge WEC (Ruehl, Michelen, Bosma, et al. 2016). Code-to-code verification has been performed as part of the WEC^3 project (Combourieu, Lawson, et al. 2015) and the IEA OES Task 10 project (Wendt et al. 2017).

2.4.3 InWave

The underlying theory of InWave was orginally developed at Ecole Centrale de Nantes and is based on the coupling of a Featherstone multibody dynamics algorithm with a hydrodynamic solver to model multibody WECs (Rongère and Clément 2013). Rongère and Clément explain that by using a multibody dynamics formalism, there is no need to derive and hard-code the WEC's equations of motion - which can be a very time consuming and error prone process. Rongère and Clément select a recursive Newton-Euler algorithm (RNEA) for the implementation, based on the performance

[18] See: WEC-Sim GitHub Issue #161: 'Why adjustMassMatrix?'
[19] Personal correspondance.

advantages over other multibody dynamics algorithms (particularly $O(n^3)$ algorithms) explained by Featherstone (2008) (Figure 2.6).

Combourieu, Philippe, Rongère, et al. (2014) describe the development of the methodology described by Rongère and Clément into the commercial software package InWave. The code has been implemented in C++ using an object-oriented design. The main components of InWave are shown in Figure 2.14 and include incident wave generation and a post-processing/visualization environment (later developed into a full GUI in InWave v1.0) to improve user-friendliness.

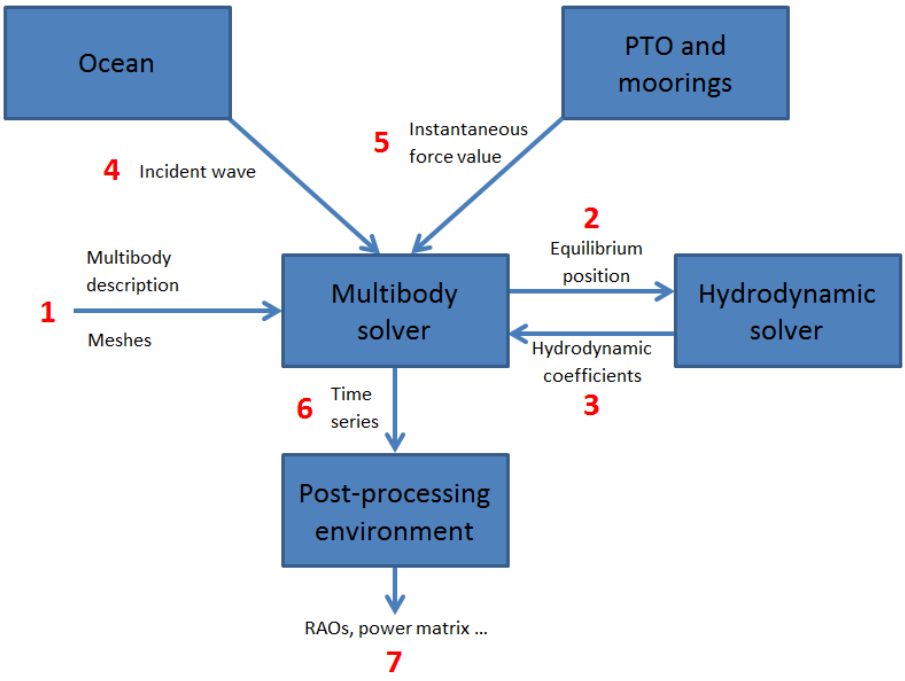


Figure 2.14: Overview of InWave's architecture & main components (Combourieu, Philippe, Rongère, et al. 2014).

The open source BEM solver NEMOH is fully integrated in InWave, using a generalized degrees of freedom approach to minimize the number of radiation damping problems that have to be solved by NEMOH. Combourieu, Philippe, Rongère, et al. (2014) explain: "The multibody solver makes the structure move with a unit velocity along each of the DoF independently. When a body moves along its DoF, the following bodies move with it (Figure 2.15)...the velocity of each face centre along the face normal are recorded by the multibody solver and given as a boundary condition to NEMOH for radiation problems. Concerning diffraction problems, the boundary conditions are simply deduced from the fluid velocity on each face centre of the structure

at equilibrium (dotted outline in Figure 2.15)."

The wave excitation force can be modelled either by linear or nonlinear Froude-Krylov force. In the case of linear Froude-Krylov force, the incident wave loads are directly added to the diffraction loads in the frequency domain. Nonlinear hydrostatic forces are also available for multibody systems - linear hydrostatics are only available for single body systems (Combourieu, Lawson, et al. 2015). Viscous drag can also be modelled with Morison's equation - using the relative velocity between the bodies and fluid. Mooring and PTO systems can be modelled in InWave with either user-defined functions or by using constant stiffness and damping coefficients.

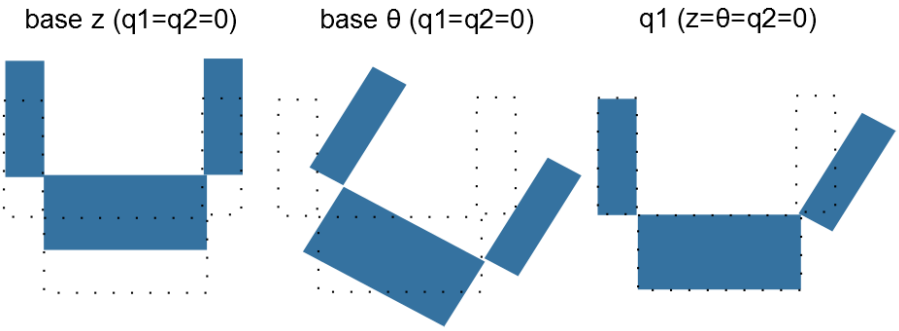


Figure 2.15: Example of a multibody structure's motion for defining radiation problems (Combourieu, Philippe, Rongère, et al. 2014).

Code to code verification of InWave has been performed using data from the NumWEC and WEC^3 projects (Combourieu, Lawson, et al. 2015; Combourieu, Philippe, Rongère, et al. 2014; Leroy, Combourieu, et al. 2014). Experimental validation has been performed using data from physical modelling campaigns carried out by Innosea with SEACAP and Langlee (Combourieu, Philippe, Larivain, et al. 2015). A summary of the published verification and validation test cases used by InWave is shown in Figure 2.16.

In addition to building on InWave's WEC modelling capabilities, recent R&D efforts have focused on developing new features and couplings to enable the modelling of floating wind turbines (Leroy, Gilloteaux, et al. 2017) and marine operations (Wuillaume et al. 2018) with InWave.

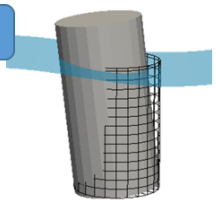
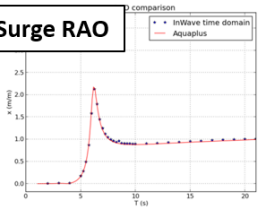
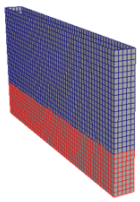
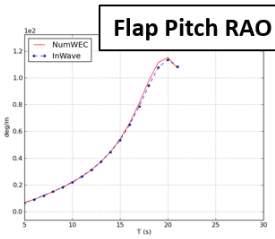
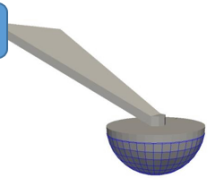
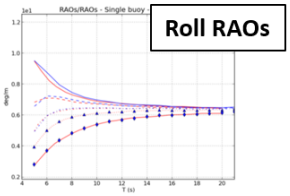
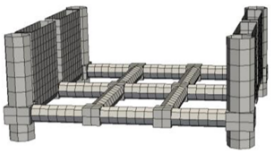
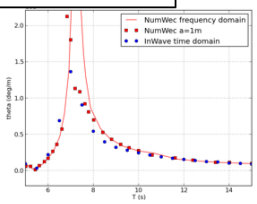
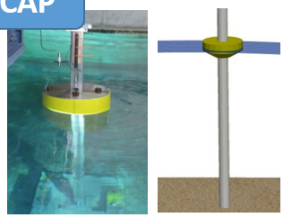
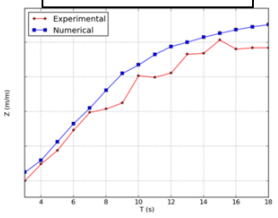
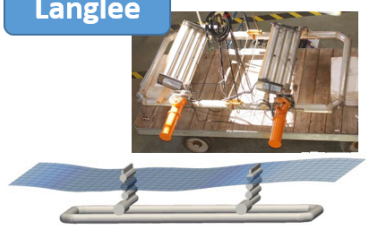
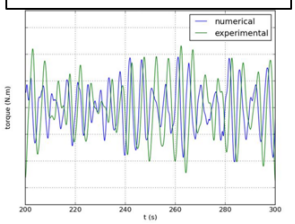
	Model	Notes	Example Data
Code to Code Verification	Cylinder 	• 1 rigid body (6 dofs) • Results verified against ECN's Aquaplus frequency domain code • Combourieu, Philippe et al (2014)	Surge RAO 
	BOF 	• 2 rigid bodies (base is fixed, flap can pitch) • Results verified against NumWEC data - Babarit (2011) • Leroy, Combourieu et al (2014)	Flap Pitch RAO 
	BHBA 	• 1 rigid body (1 dof: roll) • Results verified against NumWEC data - Babarit (2011) • Leroy, Combourieu et al (2014)	Roll RAOs 
	F3OF 	• 3 rigid bodies (base: 6 dofs, flaps: pitch) • Results verified against NumWEC data - Babarit (2011) • Combourieu, Philippe et al (2014) • Combourieu, Lawson et al (2015)	Base Pitch RAO 
Experimental Validation	SEACAP 	• 2 rigid bodies (base: fixed, float: heave) • Results validated against data from experimental campaign carried out by Innosea and Hydrocap • Combourieu, Philippe et al (2015)	Float Heave RAO 
	Langlee 	• 3 rigid bodies (base: 6 dofs, flaps: pitch) • Results validated against data from experimental campaign carried out by Innosea and Langlee • Combourieu, Philippe et al (2015)	Flap Torque Time Series 

Figure 2.16: Summary of published InWave verification and validation cases.

Chapter 3

Theory

This chapter details some of the hydrodynamics and multibody dynamics theory introduced in Chapter 2. Key resources used to write this chapter include Folley (2016) for numerical modelling of WECs, Le Méhauté (1976), Mei et al. (2005), and Newman (1977) for hydrodynamics and Shabana (2013) for multibody dynamics.

3.1 Hydrodynamics

3.1.1 Hydrostatics

Hydrostatics is the most fundamental topic of fluid mechanics, with a long history dating back to Archimedes of Syracuse (287 - 212 BC) at least. The hydrostatic force $\vec{f}_{HS}$ is a critical part of virtually all WEC models. In this chapter $\vec{f}$ is used to denote 6 DoF combined force and moment vectors.

A body immersed in a fluid (either fully or partially) experiences a buoyant force equal and opposite to the weight of the fluid it displaces:

$$W = -mg = -\rho_{body} V g \tag{3.1}$$

$$f_b = \rho_{water} g \forbar{V} \tag{3.2}$$

Where V is the total volume of the body and $\forbar{V}$ is the body's submerged volume. If a floating body is displaced from its equilibrium position by a linear vertical translation or any rotation it will experience an additional 'restoring' hydrostatic force and moment(s). For small displacements this can be modelled as a linear force and moment:

$$\vec{f_{hs}} = \mathbf{K}_{hs} \Delta \vec{s} \tag{3.3}$$

Where $\mathbf{K}_{hs}$ is the matrix of hydrostatic restoring coefficients and $\Delta \vec{s}$ is the change from the body's hydrostatic equilibrium position. For 1 floating body:

$$\mathbf{K}_{hs} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33} & K_{34} & K_{35} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} & 0 \\ 0 & 0 & K_{53} & K_{54} & K_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.4}$$

The geometry of a floating body influences it's hydrostatic stiffness properties, hence the hydrostatic stiffness coefficients can be derived from the second moments of the waterplane area, S:

$$K_{33} = \rho g S_o \tag{3.5}$$

$$K_{34} = \rho g \int\int_{S_0} y dS \tag{3.6}$$

$$K_{35} = -\rho g \int\int_{S_0} x dS \tag{3.7}$$

$$K_{44} = \rho g \int\int_{S_0} y^2 dS - \rho g \forbidden{\forall} z_g + \rho g \forbidden{\forall} z_b \tag{3.8}$$

$$K_{45} = \rho g \int\int_{S_0} xy dS \tag{3.9}$$

$$K_{55} = \rho g \int\int_{S_0} x^2 dS - \rho g \forbidden{\forall} z_g + \rho g \forbidden{\forall} z_b \tag{3.10}$$

Where S_0 is the waterplane area in the static condition. If the planes xOz and yOz are planes of symmetry, then $\mathbf{K}_{hs}$ is diagonal and $K_{34} = K_{35} = K_{45} = 0$.

3.1.2 Hydrodynamics

Fluid-structure interactions are typically complex, non-linear phenomena: capturing all of physical effects in a numerical model can lead to prohibitively long computation times for WEC designers. As mentioned in Chapters 1 and 2, linear potential flow theory has been successfully used to model many WECs in a wide range of operating conditions. This approach neglects some properties of the fluid in order to simplify the equations of motion - thus enabling faster solutions. However, the limitations of the approach much be understood and respected. This section will explain the assumptions made in linear potential flow theory, and the how the time-domain equations are derived.

3.1.2.1 Linear Potential Flow Theory

In order to model a fluid in motion, we essentially need to know the velocity vector, $\vec{v}(x, y, z, t)$, of a fluid particle at the point, $\vec{x} = (x, y, z)$ in the flow:

$$\vec{v} = \frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.11}$$

However, there are some properties of the fluid other than velocity which may also vary throughout the flow. To simplify the model we can assume that these remain constant. We can assume that the density of the fluid remains constant (i.e. the fluid is incompressible):

$$\rho = const. \tag{3.12}$$

This assumption means that the divergence of the flow's velocity must be zero. Hence, the continuity (conservation of mass) equation is simply:

$$\nabla \cdot \vec{v} = 0 \tag{3.13}$$

or,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{3.14}$$

Shear stresses can also occur in a moving fluid (also referred to as viscous effects) and are very important fluid properties which can be critical for acurrately modelling certain conditions. However, linear potential flow theory neglects viscous effects and treats the water as an *ideal fluid* (the flow is assumed to be inviscid). Because the fluid is modelled as inviscid from $t = 0s$, the vorticity must be zero everywhere - hence the fluid is also irrotational - for this reason regions of potential flow are also referred to simply as regions of irrotational flow.

Under these assumptions, the velocity vector $\vec{v}(x, y, z, t)$ may be replaced with a mathematical abstraction; the gradient of a scalar potential, $\phi(x, y, z, t)$:

$$\vec{v} = \nabla\phi = \begin{bmatrix} \frac{\partial\phi}{\partial x} \\ \\ \frac{\partial\phi}{\partial y} \\ \\ \frac{\partial\phi}{\partial z} \end{bmatrix} \tag{3.15}$$

This enables us to replace the 3 unknown scalar components of $\vec{v}$ with 1 unknown scalar: the velocity potential ϕ, from which all 3 velocity components may be computed.

The validity of the irrotational flow assumption may be checked by considering a particle travelling from point A to B within a potential flow (Figure 3.1).

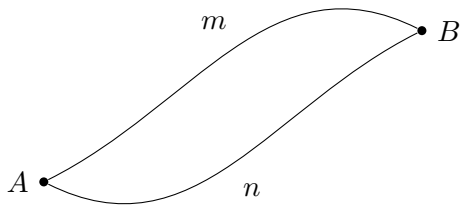


Figure 3.1: Different paths in the potential field.

The particle's velocity potential at B depends only on its position relative to A, and is independent of the path taken. Therefore, the following integral:

$$\phi = \int_A^B v_s \, \mathrm{d}s \tag{3.16}$$

is independent of the path taken (v_s is the velocity component tangential to the particle's path, s). Hence,

$$\int_A^B v_m \, \mathrm{d}s = \int_A^B v_n \, \mathrm{d}s \tag{3.17}$$

This condition can only be satisfied if the flow is irrotational; it would not be valid for a particle flowing in a closed loop[1]. Hence,

[1] Although a circular vortex could be irrotational flow if the particles do not rotate about their own axes.

$$\nabla \times \vec{v} = 0 \tag{3.18}$$

A further proof that the potential flow can be obtained by substituting 3.15 into 3.18. Substituting 3.15 into the continuity equation 3.14 yields:

$$\nabla \cdot \nabla \phi = 0 \tag{3.19}$$

$$\frac{\partial}{\partial x_i}\frac{\partial \phi}{\partial x_i} = \frac{\partial^2 \phi}{\partial x_i \partial x_i} = 0 \tag{3.20}$$

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \tag{3.21}$$

Equation 3.21 is the Laplace equation, which expresses conservation of mass for potential flows and provides the governing second-order, partial differential equation to be solved for ϕ.

Summary of the main linear potential flow assumptions:

- The fluid is **incompressible**.
- The fluid is **inviscid**.
- The flow is **irrotational**.

How these assumptions typically restrict a model:

- The amplitude of the waves must be small compared to their wavelength (Figure 3.2 (a)).
- The amplitude of the WEC motion must be small compared to the device's characteristic dimension.
- The bathymetry must be flat (or the fluid is assumed to be infinitely deep).

- Surface tension effects are negligible.

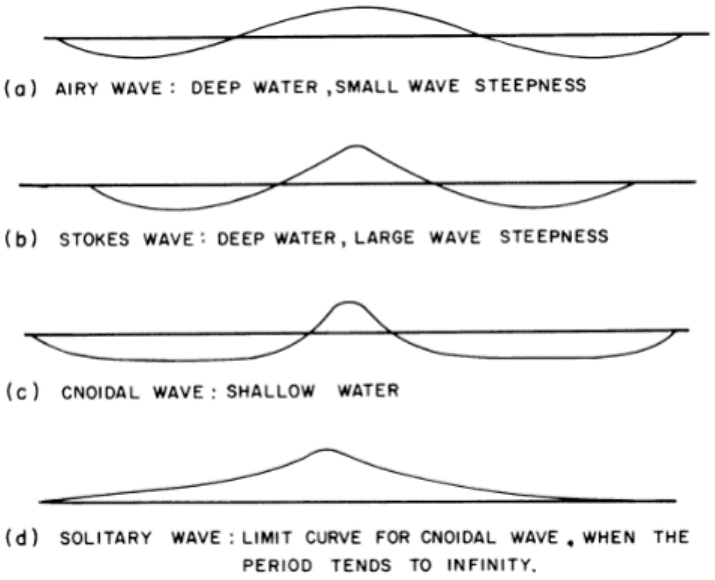


Figure 3.2: Illustration of various wave profiles (Le Méhauté 1976).

In practice, these limitations make linear potential flow theory inapplicable to modelling extreme waves, where viscous effects may be large, and non-linear wave regimes. Secondly, many WECs are designed to resonate in the wave spectrum, which may violate the small amplitude motion assumption. Thirdly, surface tension effects can be significant in physical model campaigns - potentially causing discrepancies with linear potential flow-based results). In situations where linear potential flow theory cannot be applied, higher-fidelity approaches may be more suitable (e.g. CFD, SPH).

3.1.2.2 Boundary Conditions

The flow governed by equation 3.21 must satisfy a set of boundary conditions (at the free surface, seabed and on the WEC's wetted surface(s)). The water's free surface cannot withstand pressure differences. This condition can be applied via Bernoulli's equation:

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}(\nabla \phi)^2 + \frac{p_0}{\rho} + g\eta = C \qquad on \qquad z = -\eta(x, y, t) \tag{3.22}$$

Where η is the free surface elevation. p_0 is typically atmospheric air pressure and $C = \frac{p_0}{\rho}$ so that:

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}(\nabla \phi)^2 + g\eta = 0 \qquad on \qquad z = -\eta(x, y, t) \tag{3.23}$$

The free surface is also considered impermeable, hence the fluid velocity normal to the surface must equal the surface velocity:

$$\frac{\partial \eta}{\partial t} + \frac{\partial \phi}{\partial x}\frac{\partial \eta}{\partial x} + \frac{\partial \phi}{\partial y}\frac{\partial \eta}{\partial y} + \frac{\partial \phi}{\partial z} = 0 \qquad on \qquad z = -\eta(x, y, t) \tag{3.24}$$

The second order terms in Equations 3.23 and 3.24 can be assumed small if the wavelength is much larger than the wave amplitude. Applying the boundary conditions at the free surface's undisturbed position (i.e. still water), the linearization of Equations 3.23 and 3.24 yields:

$$\frac{\partial \phi}{\partial t} + g\eta = 0 \qquad on \qquad z = 0 \tag{3.25}$$

And

$$\frac{\partial \eta}{\partial t} + \frac{\partial \phi}{\partial z} = 0 \qquad on \qquad z = 0 \tag{3.26}$$

In the general form (obtained by introducing Equation 3.26 into the time derivative of Equation 3.25) the free surface boundary condition is:

$$\frac{\partial^2 \eta}{\partial t^2} + g\frac{\partial \phi}{\partial z} = 0 \qquad on \qquad z = 0 \tag{3.27}$$

For modelling WECs, it can be assumed that the boundary at the wetted surface of the WEC's bodies is impermeable, hence the fluid velocity normal to the body surface, u_n, must be equal to the body velocity in the direction normal to its surface:

$$\frac{\partial \phi}{\partial n} = u_n \tag{3.28}$$

For the seabed boundary condition, assuming that the seabed is flat (either at depth of $h\ m$ or ∞) and impermeable, there is must be no vertical velocity component:

$$\frac{\partial \phi}{\partial z} = 0 \qquad \text{on z = -h} \tag{3.29}$$

The final condition that must be satisfied is that radiated waves must decay as the distance from the body increases:

$$\phi \propto (kr)^{1/2} e^{ikr} \qquad \text{as } r \to \infty \tag{3.30}$$

Where r is the radial distance and k is the wave number:

$$\frac{\omega^2}{g} = k tanh(kh) \tag{3.31}$$

3.1.2.3 Hydrodynamic Forces

The potential flow hydrodynamic force is obtained by integrating the dynamic pressure on the mean wetted body surface, S_b:

$$\vec{f}_{hd} = \rho \int_{S_b} \frac{\partial \phi}{\partial t} n dS_b \tag{3.32}$$

Where n is the unit vector normal to the body surface. Under the linearity assumption,

the velocity potential can be decomposed into 2 components describing the diffracted wave field and the radiated wave field:

$$\phi = \phi_D + \phi_{rad} \tag{3.33}$$

Where the diffracted wave potential, ϕ_D is:

$$\phi_D = \phi_0 + \phi_s \tag{3.34}$$

Where ϕ_0 represents the incident wave potential, and ϕ_s represents the scattered wave potential.

Hence, equation 3.32 can be expressed in terms of the complex amplitudes of the hydrodynamic force:

$$\hat{f}_{hd} = \hat{f}_{ex} + \hat{f}_{rad} \tag{3.35}$$

Where f_{ex} is the excitation force, f_{rad} is the radiation force and the hat symbol ($\hat{}$) is used to denote a complex vector; i.e. the excitation forces have both amplitude and phase in the frequency domain.

Wave Excitation Force

In the linearized approach presented here, only the first-order wave excitation force is considered[2], which has two components:

- The *Froude-Krylov* force, f_{FK}, associated with the undisturbed incident wave (i.e. as if there was no body present).
- The *scatter* force, f_s, which results from waves interacting with the body and diffracting/scattering across the free surface.

[2]Other forces that may have a significant impact on the WEC are the steady mean drift force and slow drift force. These are second order wave excitation forces and are not considered here.

$$f_{ex} = f_{FK} + f_s \tag{3.36}$$

The scatter term f_s may be considered as a 'correction' to the Froude-Krylov force, due to the presence of a body and the resulting disturbance it causes to the incident wave field.

Wave Excitation Force: Frequency-Domain

The Froude-Krylov force is derived from the incident wave potential, considering the pressure distribution over the mean wetted surface of the motionless body:

$$\hat{f}_{FK}(\omega) = i\omega\rho \int_{S_b} \hat{\phi}_0 n dS_b \tag{3.37}$$

Similarly, the scattering component of the excitation force is derived from integrating the scattered wave potential over the mean wetted surface:

$$\hat{f}_s(\omega) = i\omega\rho \int_{S_b} \hat{\phi}_s n dS_b \tag{3.38}$$

Wave Excitation Force: Time-Domain

To obtain the excitation force in the time-domain, the excitation force impulse response function (IRF) ($\mathbf{K}_{ex}$) can be convolved with the wave elevation, η.

$$\mathbf{K}_{ex}(t) = \mathbf{K}_{FK}(t) + \mathbf{K}_s(t) \tag{3.39}$$

The individual IRFs:

$$\mathbf{K}_{FK}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}_{FK}(\omega) e^{i\omega t} d\omega \tag{3.40}$$

$$\mathbf{K}_s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}_s(\omega)e^{i\omega t}d\omega \tag{3.41}$$

The time-domain equation for $\vec{f}_{ex}(t)$ is therefore:

$$\vec{f}_{ex}(t) = \mathbf{K}_{ex}(t) * \eta(t) = \int_{-\infty}^{+\infty} \mathbf{K}_{ex}(\tau)\eta(x_0, y_0, t-\tau)d\tau \tag{3.42}$$

3.1.2.4 Radiation Force

As a body oscillates in a fluid and radiates waves, there is some damping due to the radiation force. There is also an 'added mass' effect due to the body also moving some of the fluid in its vicinity. These phenomena are analagous to impedance (resistance and reactance) in eletrical circuits, and can be expressed as real and imaginary parts:

Radiation Force: Frequency-Domain

$$\hat{f}_{rad}(\omega) = -i\omega Z(\omega)\hat{\vec{s}}(\omega) \tag{3.43}$$

Where $Z(\omega)$ represents impedance:

$$Z(\omega) = -i\omega\rho\int_{S_b} \Phi n dS_b = \mathbf{B}(\omega) + i\omega\mathbf{A}(\omega) \tag{3.44}$$

$\mathbf{B}(\omega)$ is known as the hydrodynamic *damping coefficient*, and $\mathbf{A}(\omega)$ is the added mass coefficient (due to an increase in inertia as water in the vicinity of the body moves with it).

Hence,

$$\hat{f}_{rad}(\omega) = -i\omega\mathbf{B}(\omega)\hat{\vec{s}}(\omega) + \omega^2\mathbf{A}(\omega)\hat{\vec{s}}(\omega) \tag{3.45}$$

Where the damping coefficient **B** can be derived from the excitation force using the Haskind relation:

$$\mathbf{B}(\omega) = \frac{\omega k}{4k\rho g^2 D(kh)|\mathbf{A}(\omega)|} \int_{-\pi}^{\pi} \hat{f}_{ex}(\beta)\hat{f}^*_{ex}(\beta) \tag{3.46}$$

Where * denotes the complex conjugate, and β represents the phase angle of the incident wave. $D(kh)$ is a depth function, given by:

$$D(kh) = \left[1 + \frac{2kh}{\sinh(2kh)}\right] \tanh(kh) \tag{3.47}$$

Where h is the water depth and k is the 'wave number', which is linked to the dispersion relation (Equation 3.31):

The damping coefficient, $\mathbf{B}(\omega)$ is also related to the added mass coefficient $\mathbf{A}(\omega)$ via the 'Kramers-Kronig' relation (Folley et al. 2012):

$$\mathbf{B}(\omega) = \frac{2\omega^2}{\pi} \int_0^{\infty} \frac{\mathbf{A}(\omega) - \mathbf{A}(\infty)}{\omega^2 - y^2} dy \tag{3.48}$$

and

$$\mathbf{A}(\omega) - \mathbf{A}(\infty) = \frac{2}{\pi} \int_0^{\infty} \frac{-\mathbf{B}(y)}{\omega^2 - y^2} dy \tag{3.49}$$

Where y is a dimensionless parameter, discussed by Kotik and Mangulis (1962).

Radiation Force: Time-Domain

The radiation force in the time-domain can then be obtained by convolution:

$$\vec{f}_{rad}(t) = -\mathbf{A}(\infty)\ddot{\vec{s}}(t) + \mathbf{K}_{rad}(t) * \dot{\vec{s}}(t) \tag{3.50}$$

$$\mathbf{K}_{rad}(t) * \dot{\vec{s}}(t) = \int_{-\infty}^{t} \mathbf{K}_{rad}(t-\tau)\dot{\vec{s}}(\tau)d\tau \tag{3.51}$$

The radiation IRF (RIRF), $\mathbf{K}_{rad}(t)$ is given by:

$$\mathbf{K}_{rad}(t) = \frac{2}{\pi}\int_{0}^{\infty} \mathbf{B}(\omega)cos(\omega t)d\omega \tag{3.52}$$

In this approach, the added mass matrix at infinite frequency, $\mathbf{A}(\infty)$, must be added directly to the system's mass matrix:

$$(\mathbf{M} + \mathbf{A}(\infty))\ddot{\vec{s}}(t) = \int_{-\infty}^{t} \mathbf{K}_{rad}(t-\tau)\dot{\vec{s}}(\tau)d\tau + \vec{f}_{other} \tag{3.53}$$

Where $\vec{f}_{other}$ includes excitation, hydrostatic and any other forces present in the system. Solving the convolution integral by directly integrating it can be computationally expensive - to improve performance, several alternative solution methods exist - such as the Prony identification method and state-space methods. While representing the convolution integral as a state-space model is generally regarded to be much faster than direct integration, it requires the identification of 4 unknown vectors. This system-identification problem (or 'realization problem') may require additional numerical procedures in order to correctly set up the model and can also introduce errors compared to direct numerical integration.

3.2 Multibody Dynamics

There are several different approaches to solving the dynamic response of a multibody system - this section will begin by explaining some of the fundamental concepts of rigid body dynamics, and how this can be expanded to treat a multibody system.

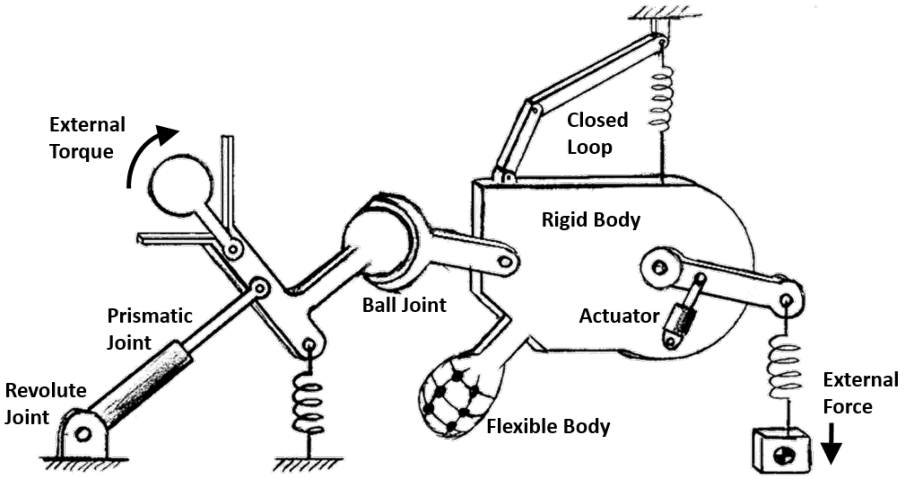


Figure 3.3: Some possible features that can be seen in multibody systems.

3.2.1 Rigid Body Kinematics

Fundamental to any multibody dynamics formalism is a systematic approach to describing the translations and rotations of rigid bodies in a fixed coordinate system. So to not assume any characteristics of the system, a three dimensional Cartesian coordinate system is used.

For a rigid body in space, we're interested in tracking its 6 DoFs: 3 translations (surge, sway, heave) and 3 rotations (roll, pitch, yaw) (Figure 3.4). These are typically combined in a 6-dimensional vector (sometimes called a 'state vector'):

$$\vec{s} = \begin{bmatrix} surge \\ sway \\ heave \\ roll \\ pitch \\ yaw \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{3.54}$$

Although roll, pitch and yaw are useful ways of thinking about a system when setting up a model or analysing the results, when computing the state vector its important to consider the order of rotations, which affects the final orientation of the body. There are 3 predominant approaches to handling rigid body rotations:

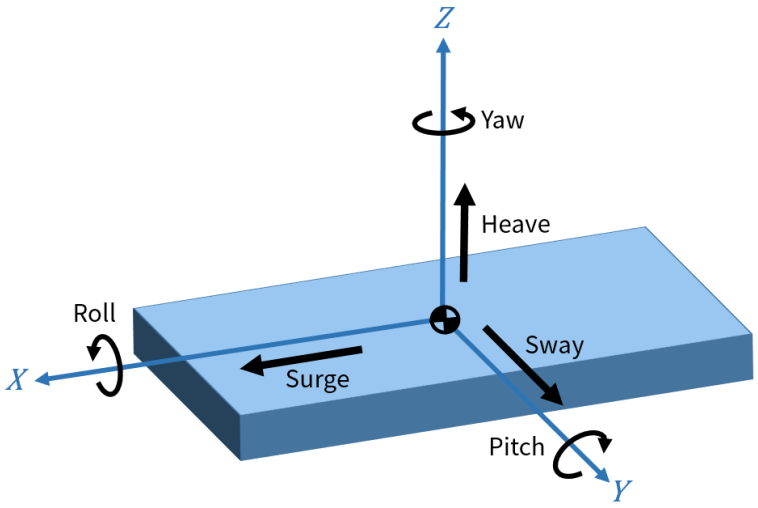


Figure 3.4: The 6 DoFs of a rigid body in space with naming conventions.

- Euler angles
- Rotation matrices
- Quaternions

Euler angles are commonly used for their simplicity, but this is not the most robust method as it can suffer from gimbal lock. Rotation matrices are also straightforward to use, but are less efficient as they require use of 9 numbers to track 3 variables. Quaternions are a popular compromise between efficiency and reliability: using 4 numbers but not suffering from the same stability issues as Euler angles[3]. The pros and cons of each method mean that it can be useful to simply select the most suitable approach for the particular operation. Hence, the details of each approach will be explained further.

3.2.1.1 Euler/Tait-Bryan Angles

Euler's rotation theorem states: *In 3D space, any displacement of a rigid body with one point fixed is a rotation about some axis.* Hence, a vector with direction matching the rotation axis, and with magnitude equal to the rotation angle could be used to describe the rotation of a rigid body about a fixed point. However, a combination of rotations cannot be expressed using this approach.

Euler introduced the concept of using three successive, elemental rotations (i.e. rota-

[3]For translations there are no issues with using a simple 3D vector as the final position of the body does not depend on the order of the translations: a 3D vector contains all of the necessary information.

tions about the axes of a coordinate system) to describe the orientation of a rigid body. The rotations may be extrinsic (about the axes of the inertial frame - see Figure 3.5) or intrinsic (about the axes of the moving frame, which is fixed to the rigid body and changes orientation with each elemental rotation).

For many engineering applications, its more intuitive to use *Tait-Bryan* angles (aka *Cardan* or *nautical* angles), for which there are 6 possible rotation sequences (x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z).

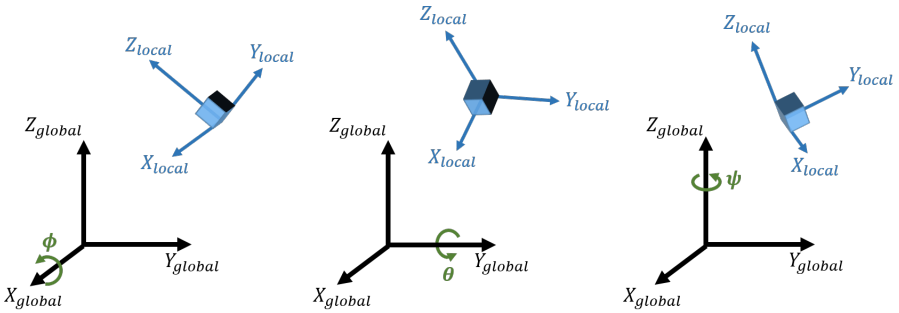


Figure 3.5: A rotation represented by ϕ, θ & ψ using extrinsic rotations and x-y-z sequence.

The rotation sequence shown in Figure 3.5 is often expressed using explicit notation in order to avoid any confusion about the convention that has been used (i.e. which rotation sequence, and whether rotations are intrinsic or extrinsic). However, this notation can be cumbersome. If we assume all rotations to be extrinsic and using the x-y-z sequence, we can express the orientation as a vector:

$$\vec{\Theta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{3.55}$$

Which can be combined with the translational vector to give the state vector in Equation 3.54.

Differentiating $\vec{\Theta}$ with respect to time gives the rigid body's angular velocity vector, $\vec{\omega}$. Differentiating extrinsic rotations gives the angular velocities in the global frame.

If two axes end up coinciding after a rotation, this leads to the rigid body losing one of it's degrees of freedom - a problem known as gimbal lock[4]. However, this problem tends to occur after a rotation of $90°$ (or an integer multiple of $90°$) - hence, if rotations are typically smaller than this, gimbal lock should not occur.

3.2.1.2 Rotation Matrices

A rotation in 3D space can also be represented as a 3×3 matrix. The following matrices describe can be use to describe the elemental rotations about each Cartesian axis:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.56}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.57}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.58}$$

To represent these elemental rotations as an extrinsic rotation in the order x-y-z, the matrices can be multiplied together as so:

$$\mathbf{R} = \mathbf{R}_x(\psi)\mathbf{R}_y(\theta)\mathbf{R}_z(\phi) \tag{3.59}$$

[4]The Achilles' heel of Euler angles!

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta \\ (\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi) & (\cos\phi\cos\psi - \sin\phi\sin\theta\sin\psi) & \sin\phi\cos\theta \\ (\sin\phi\sin\psi - \cos\phi\cos\psi\sin\theta) & (\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) & \cos\phi\cos\theta \end{bmatrix} \tag{3.60}$$

It is also straightforward to obtain the x-y-z extrinsic Tait-Bryan angles from a rotation matrix formulated this way:

$$\phi = \text{atan2}(\mathbf{R}_{23}, \mathbf{R}_{33}) \tag{3.61}$$

$$\theta = \text{atan2}(-\mathbf{R}_{31}, \sqrt{\mathbf{R}_{23}^2 + \mathbf{R}_{33}^2}) \tag{3.62}$$

$$\psi = \text{atan2}(\mathbf{R}_{12}, \mathbf{R}_{11}) \tag{3.63}$$

It is frequently necessary to express the movement of a rigid body in different frames (e.g. determine a rigid body's motion relative to another body in the system). To determine the rotation matrix corresponding to the relative angles between two bodies:

$$\mathbf{R}_{AB} = \mathbf{R}_A^{-1}\mathbf{R}_B \tag{3.64}$$

To express a vector in a frame that has been rotated relative to the original frame:

$$\vec{v}_{rotated} = \mathbf{R}\vec{v} \tag{3.65}$$

Rotation matrices can be very useful for a number of dynamics applications, particularly for post-processing kinematic data. However, the extra floating point operations results in a higher computational effort, which can make them impractical for multibody dynamics simulation. Moreover, rotation matrices are not particularly intuitive - conversions to Tait-Bryan angles are typically required to make sense of the rotation data.

3.2.1.3 Rotation Quaternions

Quaternions are an extension of complex numbers. Although first derived by Hamilton in 1843, they have become very common in recent decades - particularly for CGI applications. Quaternions are generally expressed in the form:

$$w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{3.66}$$

Or,

$$\vec{q} = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \tag{3.67}$$

Where w, x, y, z are real numbers and i, j, k are symbols with the following properties:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{3.68}$$

$$\mathbf{ij} = k = -\mathbf{ji} \tag{3.69}$$

$$\mathbf{ki} = j = -\mathbf{ik} \tag{3.70}$$

$$\mathbf{jk} = i = -\mathbf{kj} \tag{3.71}$$

A unit quaternion (a versor) has a magnitude of 1:

$$\vec{q} = \sqrt{w^2 + x^2 + y^2 + z^2} = 1 \tag{3.72}$$

A unit quaternion can be applied to 3D rotation problems by building on Euler's rotation theorem and representing a rotation with four numbers. The rotation quaternion

essentially expands on Euler's formula:

$$e^{ix} = \cos x + i \sin x \tag{3.73}$$

Hence, a rotation θ about about an axis given by unit vector, $\vec{u} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, can be expressed by the following quaternion:

$$\vec{q} = e^{(x\mathbf{i}+y\mathbf{j}+z\mathbf{k})\frac{\theta}{2}} = \cos\frac{\theta}{2} + (x\mathbf{i} + y\mathbf{j} + z\mathbf{k})\sin\frac{\theta}{2} \tag{3.74}$$

Which can also be expressed as a 4D vector:

$$\vec{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right)\vec{u} \end{bmatrix} \tag{3.75}$$

A useful identity that can be derived from the quaternion $\vec{q}$ is the *transformation matrix*, $\mathbf{G}$:

$$\mathbf{G} = \begin{bmatrix} -x & w & -z & y \\ -y & z & w & -x \\ -z & -y & x & w \end{bmatrix} \tag{3.76}$$

Using extrinsic rotations and obtaining ω in world space, the quaternion's time derivative, $\dot{\vec{q}}$, is given by:

$$\dot{\vec{q}} = \frac{1}{2}\vec{\omega} * \vec{q} \tag{3.77}$$

Where the symbol '$*$' represents a quaternion multiplication. Using matrix multiplication instead, $\dot{\vec{q}}$ can also be obtained with the quaternion transformation matrix:

$$\dot{\vec{q}} = \frac{1}{2}\mathbf{G}^{\mathbf{T}} * \vec{\omega} \tag{3.78}$$

The ability to describe a rotation using 4 floating points, but not have to worry about gimbal lock, makes quaternions a popular approach in dynamics applications. However, there are some issues with using quaternions in multibody dynamics: using a 7D state vector (instead of 6D) typically requires the system mass matrix to be converted from 6×6 to 7×7, using the transformation matrix, $\mathbf{G}$. This can become expensive in hydrodynamics applications using linear potential flow theory, where added mass at infinite frequency (a dense matrix with $ndof \times ndof$ dimensions) needs to be transformed and included at each time step (this is explained further in Section 4.5).

3.2.2 Rigid Body Dynamics

A rigid body is one which cannot change shape or mass. In reality a body typically deforms under the application of a force, but the rigid body idealization allows many systems to be accurately simulated (where body deformations are small) with much lower computational expense.

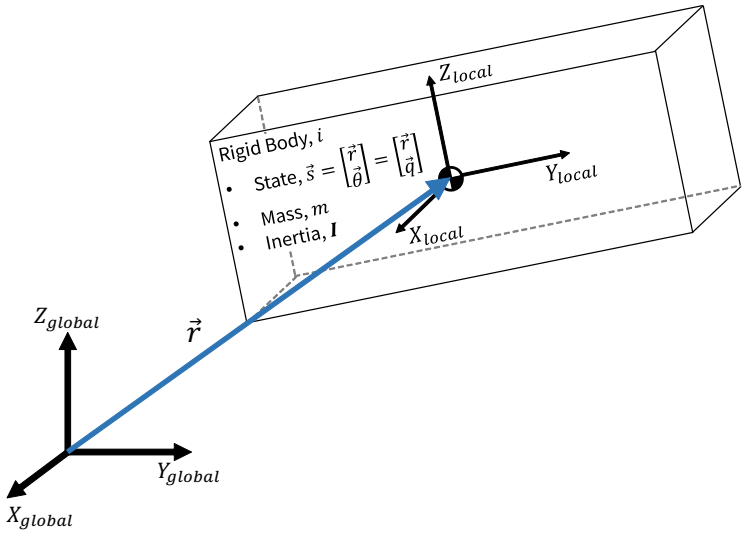


Figure 3.6: Some fundamental characteristics of a single rigid body in space.

By considering a single rigid body (Figure 3.6), we can introduce some fundamental characteristics and notations.

- A state vector, $\vec{s}$ (either 6×1 or 7×1 depending on method used for rotations):

$$\vec{s_i} = \begin{bmatrix} \vec{r_i} \\ \vec{\Theta}_i \end{bmatrix} = \begin{bmatrix} \vec{r_i} \\ \vec{q_i} \end{bmatrix} \tag{3.79}$$

- A velocity vector, $\vec{u}$ (either 6×1 or 7×1):

$$\dot{\vec{s}} = \vec{u_i} = \begin{bmatrix} \vec{v_i} \\ \vec{\omega_i} \end{bmatrix} = \begin{bmatrix} \vec{v_i} \\ \dot{\vec{q}_i} \end{bmatrix} \tag{3.80}$$

- A mass matrix, $\mathbf{M}$ (either 6×6 or 7×7):

$$\mathbf{M_i} = \begin{bmatrix} m_i \mathbf{1}_{3\times3} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_i} \end{bmatrix} = \begin{bmatrix} m_i \mathbf{1}_{3\times3} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_i^*} \end{bmatrix} \tag{3.81}$$

  Where $\mathbf{1}_{3\times3}$ is a 3×3 identity matrix, $\mathbf{I_i}$ is the body's inertia matrix and $\mathbf{I_i^*}$ is the 4×4 transformed inertia matrix, for use in a quaternion-based approach:

$$\mathbf{I_i^*} = \mathbf{G_i^T I_i G_i} \tag{3.82}$$

The equations of motion that determine the motion of this rigid body under the application of a force and/or torque were discovered by Newton and Euler during the scientific revolution:

$$\begin{bmatrix} \vec{f_i} \\ \vec{\tau_i} \end{bmatrix} = \begin{bmatrix} m_i \mathbf{1}_{3\times3} & 0 \\ 0 & \mathbf{I_i} \end{bmatrix} \begin{bmatrix} \vec{a_i} \\ \vec{\alpha_i} \end{bmatrix} + \begin{bmatrix} 0 \\ \vec{\omega_i} \times \mathbf{I_i} \vec{\omega_i} \end{bmatrix} \tag{3.83}$$

Where $\vec{a_i} = \dot{\vec{v_i}}$ and $\vec{\alpha_i} = \dot{\vec{\omega}}_i$. $\left[\vec{\omega_i} \times \mathbf{I_i} \vec{\omega_i}\right]$ is the Coriolis term.

3.2.3 Systems of Rigid Bodies

This section will describe a Lagrange-multiplier approach to multibody dynamics, in which each body retains 6 DoFs, regardless of how it is constrained in the system. Formalisms taking this approach are also called 'redundant-coordinate' formalisms, and the total number of DoFs in the system is equal to $n_{bod} \times 6$.

For a system of rigid bodies, the individual mass matrices, state vectors and force vectors can simply be combined:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{M}_n \end{bmatrix}, \vec{u} = \begin{bmatrix} \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_n \end{bmatrix}, \vec{f}_{ext} = \begin{bmatrix} \vec{f}_1 \\ \vec{f}_2 \\ \vdots \\ \vec{f}_n \end{bmatrix} \tag{3.84}$$

The governing ODE of the system is simply an extension of $f = ma$:

$$\mathbf{M}\dot{\vec{u}}(t) = \vec{f}_{ext}(t) \tag{3.85}$$

However, just solving this equation alone would consider the rigid bodies as independent from each other. To account for bodies affecting one another via joints, a separate set of kinematic constraint equations must be introduced, which are typically formulated at the position-level. For example, a ball joint (Figure 3.7) can be represented by a vector equation:

$$\vec{C}_k = (\vec{r}_i + \vec{p}_i) - (\vec{r}_j + \vec{p}_j) = \vec{0} \tag{3.86}$$

$$\vec{C}_k = \vec{r}_{pi} - \vec{r}_{pj} = \vec{0} \tag{3.87}$$

Differentiating position-based constraint equations leads to velocity-based and acceleration-

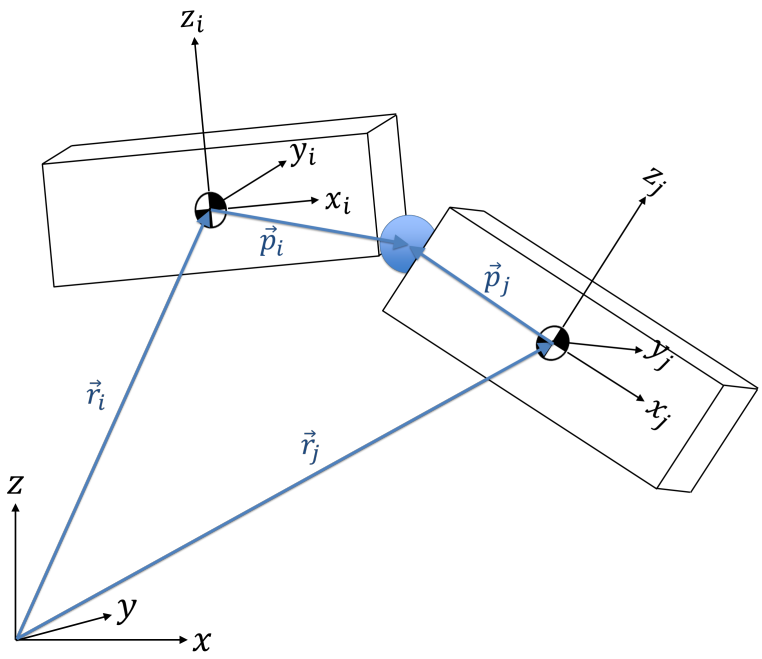


Figure 3.7: Illustration of two rigid bodies connected via a ball joint.

based constraint equations:

$$\dot{\vec{C}}_k = \frac{\mathrm{d}C_k}{\mathrm{d}t} = \frac{\partial C_k}{\partial \vec{s}}\dot{\vec{s}} = \mathbf{J}_k\vec{u} \tag{3.88}$$

$$\ddot{\vec{C}}_k = \mathbf{J}_k\dot{\vec{u}} + \dot{\mathbf{J}}_k\vec{u} \tag{3.89}$$

Deriving the acceleration-based constraint equations can be useful for novel or unusual contraints, but for most standard joints the Jacobian, $\mathbf{J}$ is well-known - e.g. for a ball joint:

$$\mathbf{J}_{ball} = \left[\mathbf{J}^i_{lin}, \mathbf{J}^j_{lin}, \mathbf{J}^i_{ang}, \mathbf{J}^j_{ang}\right] \tag{3.90}$$

Where,

$$\mathbf{J}^i_{lin} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.91}$$

$$\mathbf{J}^j_{lin} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{3.92}$$

$$\mathbf{J}^i_{ang} = -skew(\vec{p}_i) \tag{3.93}$$

$$\mathbf{J}^j_{ang} = -skew(\vec{p}_j) \tag{3.94}$$

Where $skew$ denotes skew-symmetric matrix. A large system of constraints can be represented by one combined Jacobian:

$$\mathbf{J} = \frac{\partial \vec{C}}{\partial \vec{s}} = \begin{bmatrix} \frac{\partial \vec{C}_1}{\partial \vec{s}_1} & \frac{\partial \vec{C}_1}{\partial \vec{s}_2} & \cdots & \frac{\partial \vec{C}_1}{\partial \vec{s}_n} \\ \frac{\partial \vec{C}_2}{\partial \vec{s}_1} & \frac{\partial \vec{C}_2}{\partial \vec{s}_2} & \cdots & \frac{\partial \vec{C}_2}{\partial \vec{s}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \vec{C}_2}{\partial \vec{s}_1} & \frac{\partial \vec{C}_n}{\partial \vec{s}_2} & \cdots & \frac{\partial \vec{C}_n}{\partial \vec{s}_n} \end{bmatrix} \tag{3.95}$$

To enforce a system of constraints, a separate 'constraint-force' vector, $\vec{f_c}$, must to be computed and included in the system's equation of motion:

$$\mathbf{M}\dot{\vec{u}}(t) = \vec{f_c}(t) + \vec{f}_{ext}(t) \tag{3.96}$$

Where,

$$\vec{f_c} = \mathbf{J}^T \vec{\lambda} \tag{3.97}$$

J determines the directions of the constraint force, while the Lagrange multipliers, $\vec{\lambda}$ determine the magnitude. The main objective of the Lagrange Multiplier Method is to determine the vector $\vec{\lambda}$ so that the constraint force vector can be computed and applied

to the system.

The mixture of ordinary differential equations (ODEs) describing the motion of rigid bodies, and algebraic equations describing the constraints leads to a set of *differential-algebraic equations* (DAEs), which are commonly expressed in the form:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\vec{u}} \\ \vec{\lambda} \end{bmatrix} = \begin{bmatrix} \vec{f}_{ext} \\ -\vec{c} \end{bmatrix} \tag{3.98}$$

Where $\vec{c} = \dot{\mathbf{J}}\vec{u}$.

To solve the multibody system's DAEs, the vector $\vec{\lambda}$ must be found so that $\vec{f}_c$ combined with $\vec{f}_{ext}$ and applied to the system results in motion that does not violate the system's constraint equations. A common approach to this problem is the '$\mathbf{JM^{-1}J^T}$ Approach':

Rearranging the system's equation of motion for $\dot{\vec{u}}$ gives:

$$\dot{\vec{u}} = \mathbf{M}^{-1}\mathbf{J}^T\vec{\lambda} + \mathbf{M}^{-1}\vec{f}_{ext} \tag{3.99}$$

Inserting $\dot{\vec{u}}$ into Equation 3.89 yields:

$$\ddot{\vec{C}} = \mathbf{J}(\mathbf{M}^{-1}\mathbf{J}^T\vec{\lambda} + \mathbf{M}^{-1}\vec{f}_{ext}) + \vec{c} = \vec{0} \tag{3.100}$$

Expanding:

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\vec{\lambda} + \mathbf{J}\mathbf{M}^{-1}\vec{f}_{ext} + \vec{c} = \vec{0} \tag{3.101}$$

Equation 3.101 can be simplified to:

$$\mathbf{A}\vec{\lambda} + \vec{b} = \vec{0} \tag{3.102}$$

Where,

$$\mathbf{A} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \tag{3.103}$$

$$\vec{b} = \mathbf{J}\mathbf{M}^{-1}\vec{f}_{ext} + \vec{c} \tag{3.104}$$

The fundamental aspects of redundant coordinate multibody dynamics has been described here and sufficient information is included to develop a basic redundant coordinate solver. For further information on advanced integration methods used to solve Lagrange's equations of the first kind with improved accuracy, speed and stability, the reader is referred to Negrut (1998) and Gerstmayr and Stangl (2004). These works describe the theoretical foundations of the state-of-the-art open-source multibody dynamics codes Chrono and HOTINT, respectively.

Chapter 4

Software Development & WEC Modelling Process

This chapter describes the development of a new WEC modelling software platform: InWave-HOTINT, which couples the multibody dynamics code HOTINT to a time-domain hydrodynamic solver (based on InWave), which itself utilizes another third-party code (NEMOH) as a pre-processor.

As there are many open-source multibody dynamics codes available, the first part of this chapter will describe the selection criteria and process that led to HOTINT being chosen and explain why this was favoured over other codes.

The second part of the chapter will explain the InWave-HOTINT coupling strategy, and overall architecture of the code. The remaining sections will describe the code in more detail: the process of creating a WEC model and some details of the algorithms.

Although presented as separate chapters, the development of InWave-HOTINT was conducted in parallel with the verification of the code - primarily via direct comparison of kinematic results (e.g. decay test time-series and response amplitude operators (RAOs) generated from time-domain results).

4.1 Multibody Dynamics Code Review

As discussed in Chapter 1, the utilization of a third-party multibody dynamics code was eventually preferred over developing a multibody code in-house, to avoid repeating work and to capitalize on many developer-years of effort that has contributed to improved speed, stability, accuracy and libraries of elements. However, there are many multibody dynamics codes available. Hence, to determine the most suitable code, a review of available multibody codes was undertaken in April 2016. As well as fundamental information about the codes, certain capabilities of interest were also investigated, such as the ability to model flexible bodies and cables (however, the extent of each code's capabilities in these categories and the theoretical basis has not been investigated in depth). The assessment criteria is summarized as follows:

- The code's fundamental information:
  - Name of the code and lead developer
  - Target application (i.e. visual effects or science & engineering)
  - License and current project status (i.e. date of last update)
  - Programming language(s) used
  - Methodology (i.e. redundant or reduced coordinates?)
- Some details of the code - can it model:
  - Closed kinematic loops
  - Flexible bodies and cables (typically via finite element method, but this category may also include lower-order methods)
  - Drivetrains (e.g. shafts, gearboxes)
  - Control systems (does the code have built-in control elements to take certain variables in the model as inputs, perform some mathematical operations on them and return a 'controlled' variable?)
  - Can the code utilize threading/parallelization to improve its performace?

Furthermore, the ability to interface with the industry-standard control systems modelling software, Simulink. The information obtained for each multibody code found is presented in Table 4.1[1].

[1]This research was also used to make an evidence-based case for third-party code adoption, as explained in Chapter 1.

Review of Multibody Dynamics Software Packages (April 2016).

Name	Developer	Target Application	License/Status	Last Updated	Source Code Language	Methodology	Closed-loops	Flexible Bodies	Cables	Drivetrain	Control Lib	Simulink	Threading/Parallelization
3d_mec	Public University of Navarra, Spain	Engineering	GNU GPL	29/01/2010	C++	Lagrange Multiplier	Y						
ADAMS	MSC Software, USA	Engineering	Commercial Software	01/10/2014	Closed	Lagrange Multiplier	Y	Y	Y	Y	Y	Y	Y
Alaska	Institut für Mechatronik e.V.	Engineering	Test version available for noncommercial applications	01/12/2015	C++	Lagrange Multiplier & Reduced-Coordinate	Y	Y		Y		Y	
APEngine	APEngine	Visual Effects	Not yet released										
AUTOLEV			Terminated as of December 31, 2010										
AVL EXCITE	AVL	Engineering	Commercial Software	01/01/2016	Closed	Lagrange Multiplier	Y	Y		Y	Y	Y	Y
BEPU physics		Visual Effects	Apache										
Box2D	Erin Catto	Visual Effects	zlib license	01/01/2016	C++	Lagrange Multiplier	Y						
Bullet	Erwin Coumans	Visual Effects	zlib license	01/04/2016	C++	Lagrange Multiplier & Reduced-Coordinate	Y	Y	Y				Y
Camel-View	iXtronics	Engineering	Terminated										
Chrono	Project Chrono	Engineering	BSD-3	01/04/2016	C++	Lagrange Multiplier	Y			Y		Y	Y
Chipmunk2D	Howling Moon Software	Visual Effects	liberal MIT license	01/01/2016	C	Lagrange Multiplier	Y						
Compamm		Engineering	Terminated										
Concurrent Dynamics	Concurrent Dynamics	Engineering	Contact support@concurrent-dynamics.com for free license		C++						Y	Y	
Darwin2K	Carnegie Mellon University	Engineering	GNU GPLv2	15/03/2003	C++	Reduced coordinate	Y			Y	Y		
daVinciCode			Terminated										
DigitalRune	DigitalRune	Visual Effects	Commercial Software	10/03/2016	C#	Lagrange Multiplier	Y						Y
Dymola	DSS Catia	Engineering	Commercial Software	25/11/2015	Modelica		Y			Y	Y	Y	Y

Name	Developer	Target Application	License/Status	Last Updated	Source Code Language	Methodology	Closed-loops	Flexible Bodies	Cables	Drivetrain	Control Lib	Simulink	Threading/Parallelization
DYMORE	Olivier A. Bauchau, University of Maryland, USA	Engineering	GNU GPL	01/01/2014		Lagrange Multiplier	Y	Y	Y		Y	Y	
DynaFlex			Terminated: now MapleSim										
DynaMechs	The Ohio State University	Engineering	GNU GPLv2	18/07/2001	C++	Reduced coordinate	Y				Y		
DynaMo			Terminated										
DynaSym			Terminated: Incorporated into DSS Catia/Dymola										
EasyDyn			Terminated										
EOM	University of Windsor	Engineering	GNU GPL	27/08/2015	Octave			Y					
Farseer Physics Engine	Farseer Physics	Visual Effects	Ms-PL	26/08/2013	C++	Lagrange Multiplier	Y				Y		
freeCAD	AR-CAD	Engineering	No license	14/05/2007	Smalltalk	Lagrange Multiplier	Y			Y			
Havok Pysics	Havok	Visual Effects				Lagrange Multiplier					Y		Y
Henge 3D		Visual Effects	MIT License										
HOTINT	Johannes Kepler University, Linz, Austria	Engineering	BSD-3	23/12/2013	C++	Lagrange Multiplier	Y	Y	Y		Y	Y	
HyperMatter			Commercial Software										
IBDS/PBD	Jan Bender	Visual Effects	MIT License	01/01/2015	C++	Lagrange Multiplier	Y						
Icarus		Visual Effects	zlib license										
JigLib	Rowlhouse	Visual Effects	MIT License	21/09/2012	Java	Lagrange Multiplier							
jinngine		Visual Effects	GPL										
Jitter		Visual Effects	MIT License										
KinSyth			Terminated										
LiquidFun	Google	Visual Effects	zlib license	01/01/2016	C++	Lagrange Multiplier	Y						
LMS Virtual.Lab	Siemens	Engineering	Commercial Software		C++	Lagrange Multiplier	Y	Y	Y	Y	Y	Y	Y

Name	Developer	Target Application	License/Status	Last Updated	Source Code Language	Methodology	Closed-loops	Flexible Bodies	Cables	Drivetrain	Control Lib	Simulink	Threading/Parallelization
MapleSim	Maplesoft	Engineering	Commercial Software	01/02/2015	Closed	Lagrange Multiplier	Y	Y		Y	Y	Y	Y
Matali		Visual Effects	Commercial Software										
MBDyn	Politecnico di Milano	Engineering	GNU GPLv2	30/10/2015	C++	Lagrange Multiplier	Y	Y					
MBS3D			Terminated										
MBSoft			Terminated										
MECANO			Terminated: Incorporated into LMS Virtual.Lab										
MechDesigner	PS Motion	Engineering	Commercial Software	22/03/2016			Y			Y	Y		
MechXML			Terminated										
Meqon			Terminated										
Microsoft robotics developer studio			Commercial Software										
Modelica/Multibody	Modelica Association/Martin Otter	Engineering	Used with Modelica - Modelica License 2 - use in commercial applications is permitted										
Motion Genesis Kane	Motion Genesis	Engineering	Commercial Software	01/01/2016	MATLAB/C/Fortran	Lagrange Multiplier	Y				Y		
MotionSolve	Altair Hyperworks	Engineering	Commercial Software	01/03/2016	C++	Lagrange Multiplier	Y	Y		Y	Y	Y	Y
Multiflex	University of Glasgow	Engineering	Mathematica code	01/01/2004	Mathematica	Reduced-coordinate		Y					
Nape Physics Engine		Visual Effects	GNU GPL										
Natural Motion		Visual Effects	Commercial Software										
Neweul-M²	University of Stuttgart	Engineering	Open source	17/11/2015	Matlab	Lagrange Multiplier	Y	Y			Y	Y	
Newton Dynamics		Visual Effects	zlib license	22/09/2014	C++								

Name	Developer	Target Application	License/Status	Last Updated	Source Code Language	Methodology	Closed-loops	Flexible Bodies	Cables	Drivetrain	Control Lib	Simulink	Threading/Parallelization
ODE	Russel Smith	Visual Effects	BSD	04/02/2014	C/C++	Lagrange Multiplier	Y						
ofxBox2d	Erin Catto/Todd…	Visual Effects	zlib license	01/01/2016	C++	Lagrange Multiplier	Y						
OMD	Enlighten Engineering	Engineering	Open source	01/03/2016	C++	Lagrange Multiplier & Reduced-Coordinate	Y					Y	
Oops! Framework													
Open Physics Library			Terminated										
OpenModelica		Engineering											
OpenTissue		Visual Effects	zlib license	08/01/2012	C++	Lagrange Multiplier							
Unity		Visual Effects											
PhysicsAndMathLibrary	game-physics-engine	Visual Effects	Open source	01/01/2008	C++	Lagrange Multiplier	Y	Y	Y				
Moby	Physsim	Engineering	GNU GPLv2	15/02/2010	C++	Lagrange Multiplier & Reduced-Coordinate							
PhysX		Visual Effects	Commercial Software										
Pixelux		Visual Effects											
PyDy	PyDy	Engineering	BSD	01/04/2016	Python	Reduced-coordinate							
RecurDyn	Function Bay	Engineering	Commercial Software	01/01/2016		Lagrange Multiplier	Y	Y	Y	Y	Y	Y	
ReDySim	IIIT Hyderabad	Engineering	Open source	01/01/2014	Matlab	Reduced-coordinate	Y						
RoboAnalyzer	IIT Delhi												
RobotBuilder													
Robotran	Universite Catholique de Louvain	Engineering	Closed source	01/01/2014	Matlab/ Python/ C/C++	Reduced-coordinate	Y						
SAM	Artas	Engineering	Commercial Software	01/01/2015			Y	Y					
SC Motion	AR-CAD	Engineering	Commercial Software	01/01/2015			Y			Y			
SD/FAST	PTC	Engineering	Closed source	19/04/2001									
Silux		Engineering		01/01/2002									
Simbody	SimTK	Engineering	Apache	15/06/2015	C++	Reduced-coordinate							

Name	Developer	Target Application	License/Status	Last Updated	Source Code Language	Methodology	Closed-loops	Flexible Bodies	Cables	Drivetrain	Control Lib	Simulink	Threading/Parallelization
SimCreator	Realtime technologies inc	Engineering	Commercial Software	01/01/2013			Y	Y		Y	Y		
Simscape Multibody	MathWorks	Engineering	Commercial Software	01/01/2016	Matlab		Y			Y	Y	Y	
SIMPACK	DSS Simulia	Engineering	Commercial Software				Y	Y		Y	Y	Y	
Simple Physics Engine		Visual Effects	Commercial Software	06/08/2009									
Simul-X													
SimulationX	ITI	Engineering	Commercial Software				Y	Y		Y	Y		
SOFA	CIMIT, Boston	Engineering	Open source	01/01/2016	C++	Lagrange Multiplier	Y	Y					
SPACAR	University of Twente	Engineering	Open source	27/02/2016	Matlab	Lagrange Multiplier	Y	Y				Y	
SystemModeler	Wolfram	Engineering	Commercial Software	01/01/2016			Y			Y	Y		
Tokamak			Terminated										
True Axis	True Axis	Visual Effects	Open source	01/01/2009	C++								
TRUE World	TRUE - Temporal Reasoning Universal Elaboration	Engineering		25/06/2014							Y		
UMDHeliUM			Terminated										
Universal Mechanism	Bryansk State Technical University, Russia	Engineering	Commercial Software	01/01/2013			Y					Y	
Working Model	Design Simulation Technologies	Engineering	Commercial Software	01/01/2016			Y	Y		Y		Y	

A total of 95 multibody dynamics codes were discovered, 77 of which were under active development or management. The majority of active codes discovered (48; 62%) have been developed for scientific & engineering applications - the remainder have been developed for visual effects applications such as films and games.

Although similar theoretical approaches are used in both sectors, the main difference is typically the time integration scheme: engineering codes focus on accuracy whereas visual effects codes focus on speed. For example, the engineering code HOTINT uses implicit Runge-Kutta (IRK) integration methods up to an order of 20. On the other hand, the visual effects code Open Dynamics Engine utilizes a first-order explicit Euler integration scheme, and it's documentation states: "Open Dynamics Engine's (ODE) current integrator is very stable, but not particularly accurate unless the step size is small. For most uses of ODE this is not a problem – ODE's behavior still looks perfectly physical in almost all cases. However, ODE should not be used for quantitative engineering [at present]".

4.1.1 Selection of HOTINT

An open-source code was desired in order to allow full access to the multibody dynamics equations for their modification and coupling to a hydrodynamic solver. Furthermore, a Lagrange multiplier (redundant-coordinate/DAE) approach was sought in order to overcome the limitations encountered by InWave's reduced-coordinate solver, as discussed in Chapters 1 and 2.

Of the 48 active engineering codes, 19 were open-source, with a fairly even split between the two methodologies (Figure 4.1). The reduced-coordinate engineering codes discovered (e.g. DynaMechs, Robotran, ReDySim) were typically more focused on the modelling of open-chain robotic systems such as robotic arms/manipulators.

Eleven open-source engineering codes were discovered that utilized the Lagrange multiplier method - the breakdown according to programming language and license is shown in Table 4.2.

As the existing InWave code has been implemented in C++, a multibody dynamics code implemented in the same language would make the option of a monolithic cou-

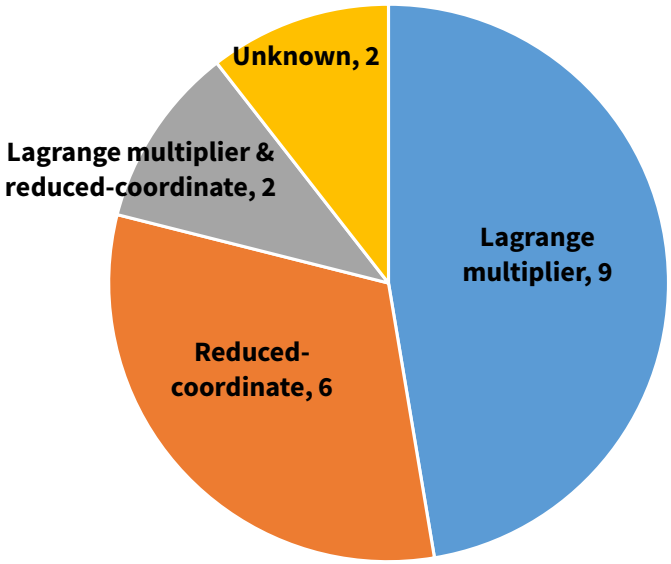


Figure 4.1: Breakdown of methodologies used by the 19 multibod dynamics codes discovered that were open-source and engineering-focused (as of April 2016).

Table 4.2: Programming language & license details for open-source, engineering-focused, Lagrange multiplier multibody dynamics codes.

Language \ License	BSD-3	GNU GPL	GNU GPLv2	Unknown License
C++	•Chrono •HOTINT	•3d_mec •DYMORE	•MBDyn •Moby	•OMD •SOFA
Matlab				•Neweul-M² •SPACAR
Smalltalk				•freeCAD

pling strategy more straightforward. Of the C++ codes shown in Table 4.2, Chrono and HOTINT were published under the BSD-3 license[2] - one of the most permissive open-source software licenses, which would give Innosea more options in the future with regards to further InWave development, licensing and usage in a commercial environment.

Hence, Chrono and HOTINT were investigated in more detail - an overview of the two codes is shown in Table 4.3. The codes are very similar - both offer a wide range of joints (and other types of multibody constraints) as well as support for modelling flexible bodies and performing co-simulation with Simulink - all important features for Innosea's WEC modelling ambitions. However, several factors made HOTINT the preferred choice over Chrono:

- Most critically, the ability to modify entire system mass matrix, $\mathbf{M}$, at each time step (necessary to include the added mass at infinite frequency matrix, $\mathbf{A}(\infty)$ - see Chapter 3).
- An extensive library of control elements:
  - Sensors for obtaining data from variables within the multibody system.
  - Various operators (e.g. transfer functions, custom mathematical functions).
  - Data modifiers for updating system variables/return control force.
- Extensive documentation, with dedicated developer's documentation[3].
- Dedicated support.

HOTINT permits the modification of the entire mass matrix because it uses implicit Runge-Kutta time integration methods, which avoid factorizing the multibody system's mass matrix. As explained in Chapter 2, many other Lagrange multiplier approaches factorize the mass matrix when integrating the equations of motion, and exploit the fact that a multibody system's mass matrix typically has a sparse, block-diagonal structure. HOTINT also includes sparse matrix methods, which can improve

[2] https://opensource.org/licenses/BSD-3-Clause

[3] It should be noted that since this code review was undertaken in 2016, Chrono's documentation has been significantly expanded.

the computational efficiency of the solver. However, for InWave-HOTINT, the non-sparse matrix methods must be used in order to include off-diagonal added mass terms in the system's mass matrix.

Table 4.3: Overview of Chrono and HOTINT (as of April 2016).

	Chrono	**HOTINT**
First release	2013	2013
No. of developers	9	6
Lead developers	University of Wisconsin-Madison, USA University of Parma, Italy	Johannes Kepler University, Austria[4]
Language	C++	C++
Notes	•Library of powertrain elements •Supports use of the GPU	•Library of control elements •Mass matrix is not factorized

4.2 InWave-HOTINT Coupling

4.2.1 Main InWave-HOTINT Coupling Strategy

HOTINT offers the option to link to an external load calculator via TCP/IP[5] - this approach has been utilized to develop a *non-monolithic* coupling between separately compiled codes (HOTINT and the SPH code LIGGGHTS) by Schörgenhumer (2012). Alternatively, the relevant hydrodynamic load solvers could be included in the HOTINT source code and compiled together as one program (a *monolithic* coupling). Both approaches require the creation of an interface - to transfer the relevant system information to the hydrodynamic load solver, and to return the updated forces to the multibody solver. In a non-monolithic structure using TCP/IP, this process is typically more complicated and slower than a monolithic code (Schörgenhumer 2012), which although can become difficult to maintain is typically more efficient as it can make better use of the main memory.

In this project a monolithic approach has been taken - incorporating the hydrodynamic solvers within the HOTINT codebase. However, the additions have been implemented in a modular way to minimize the amount of modifications made to the existing HOTINT code - only including the necessary interfaces to the HOTINT kernel. Hence, as HOTINT is updated it should be relatively straightforward to transfer

[4] As of March 2019, HOTINT is being developed by the Linz Center of Mechatronics GmbH, Austria.

[5] Transmission Control Protocol/Internet Protocol: a suite of communication protocols used for transferring data on the internet and other computer networks.

the hydrodynamics modules and include them in the updated HOTINT code. Similarly, if Innosea need to utilize an alternative multibody dynamics solver in the future, it should be relatively straightforward to transfer the hydrodynamics modules and develop an interface to pass the relevant kinematic data to the hydrodynamics module.

The overall InWave-HOTINT structure is shown in Figure 4.2 and will be presented in more detail in the following sections. The HOTINT modules are shown in red and NEMOH in blue. Fundamentally, the main goal of the InWave-HOTINT coupling is to be able to create multibody models with all the features that HOTINT offers (rigid bodies, flexible bodies, joints, ropes, controllers, Simulink cosimulation etc.), while also being able to apply hydrodynamic loads to selected rigid bodies in the system using the time-domain potential flow approach originally developed in InWave. This will allow the WEC modeller to explore a range of design options and topologies and analyze each design iteration in terms of the power produced, the loads experienced by the device, and other variables in the system, such as kinematics[6].

NEMOH has not been fully integrated in InWave-HOTINT (as it was in the original InWave) - instead, NEMOH is run separately and the hydrodynamics coefficients (hydrodynamics database/HDB) are saved to disk. The HDB location is then specified in the `.hhi` input file, and read at the beginning of a time-domain simulation. Although this approach can only be used with NEMOH formatted databases at the moment, it should be easy to extend it to other formats such as WAMIT.

[6] At present, electrical subsystems have not been considered in InWave-HOTINT.

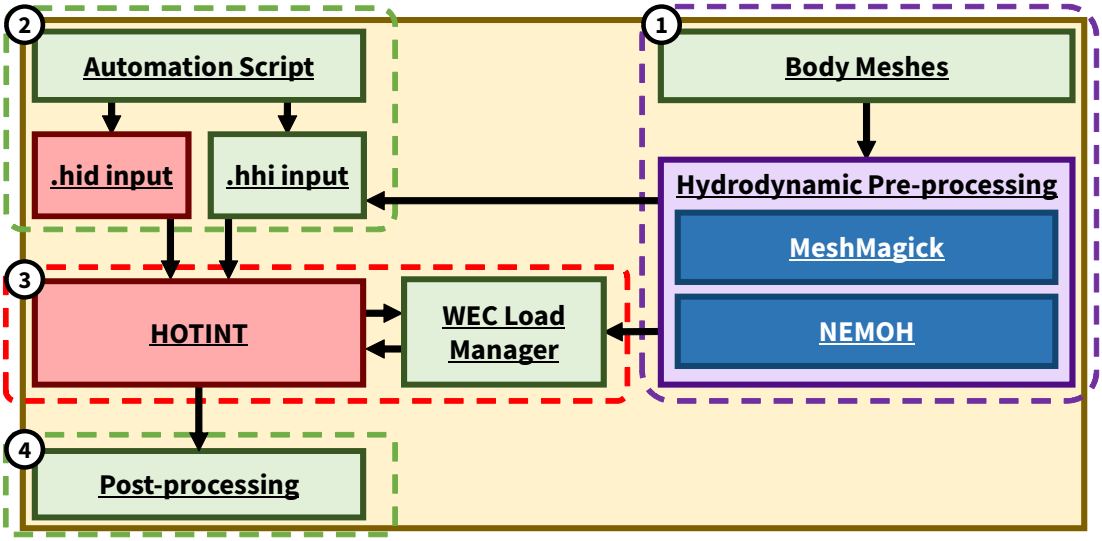


Figure 4.2: Overview of the InWave-HOTINT software structure, using the NEMOH BEM code to compute the hydrodynamic coefficients (commonly referred to as hydrodynamic database or HDB).

4.2.2 InWave-HOTINT WEC Modelling Process

As shown in Figure 4.3 there are 3 separate input files (`.hid` `.hhi` and `.cal` - all written in plain text) required from the user: the standard HOTINT input file (`.hid`) has been modified to include the location of the 'HOTINT hydro input' file (`.hhi`), which includes various extra inputs such as the hydrostatic stiffness matrix ($\mathbf{K}_H$) and environmental conditions. This modular approach was preferred over including the `.hhi` inputs directly within the `.hid` file. The `.hhi` file also contains the location of the HDB to be used, which is generated using the standard NEMOH process. It is possible to create automated batch runs using a Python script to modify the `.hid` and `.hhi` input files and call the HOTINT program via the command line.

Figure 4.3 shows that before time-domain simulations can be performed, the body's mesh must undergo some pre-processing (i.e. ensuring that it has been cropped at the waterline and that the body is in hydrostatic equilibrium and - see Section 4.3). The mesh, and the centre of gravity (CoG) positions of each body are then passed to NEMOH. The CoG positions and inertia tensor $\mathbf{I}$ are required by the `.hid` input file. $\mathbf{K}_H$ is required by the `.hhi` input file.

Key aspects of the time-domain simulation process (the coupling between HOTINT and the WEC load manager in Figure 4.2), are presented in Section 4.5. As Figure 4.3 shows, there are two approaches for analyzing the data. The raw data from each simulation (the position and velocity data of all degrees of freedom in the system, plus any sensors specified by the user to record other variables in the system - such as rope tension or joint damping force) can be visualized directly in the HOTINT GUI. For more WEC-specific analysis, a number of Python routines are available for computing relative motion between bodies and common metrics such as RAOs and power matrices.

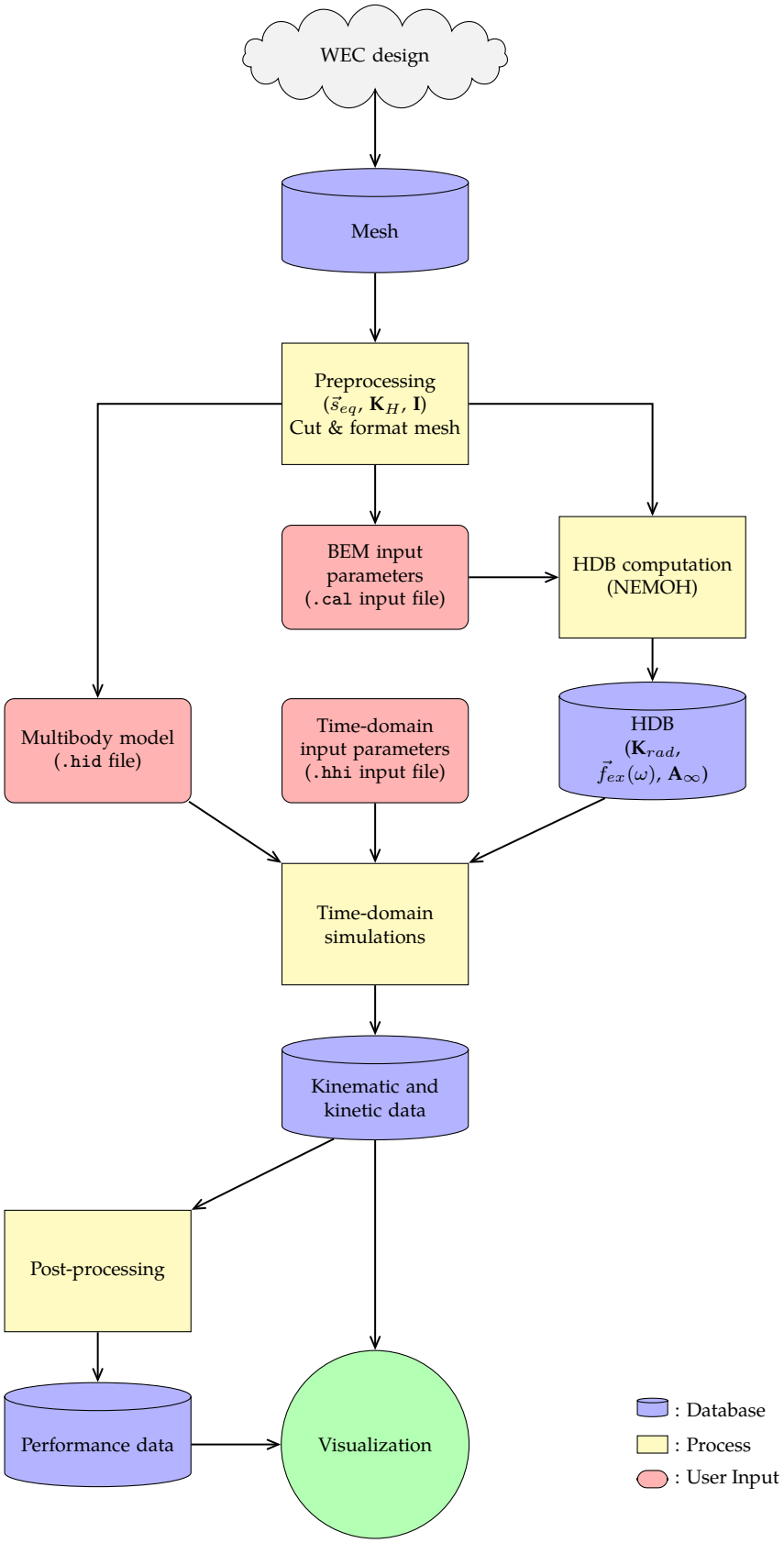


Figure 4.3: Overview of the WEC modelling workflow used by InWave-HOTINT.

4.2.3 Overview of an InWave-HOTINT WEC Model

An overview of a typical WEC model that can be created with InWave-HOTINT, with all its constituent subsystems and other inputs is shown in Figure 4.4. The details of these parts will be described in more detail in Sections 4.3 and 4.4.

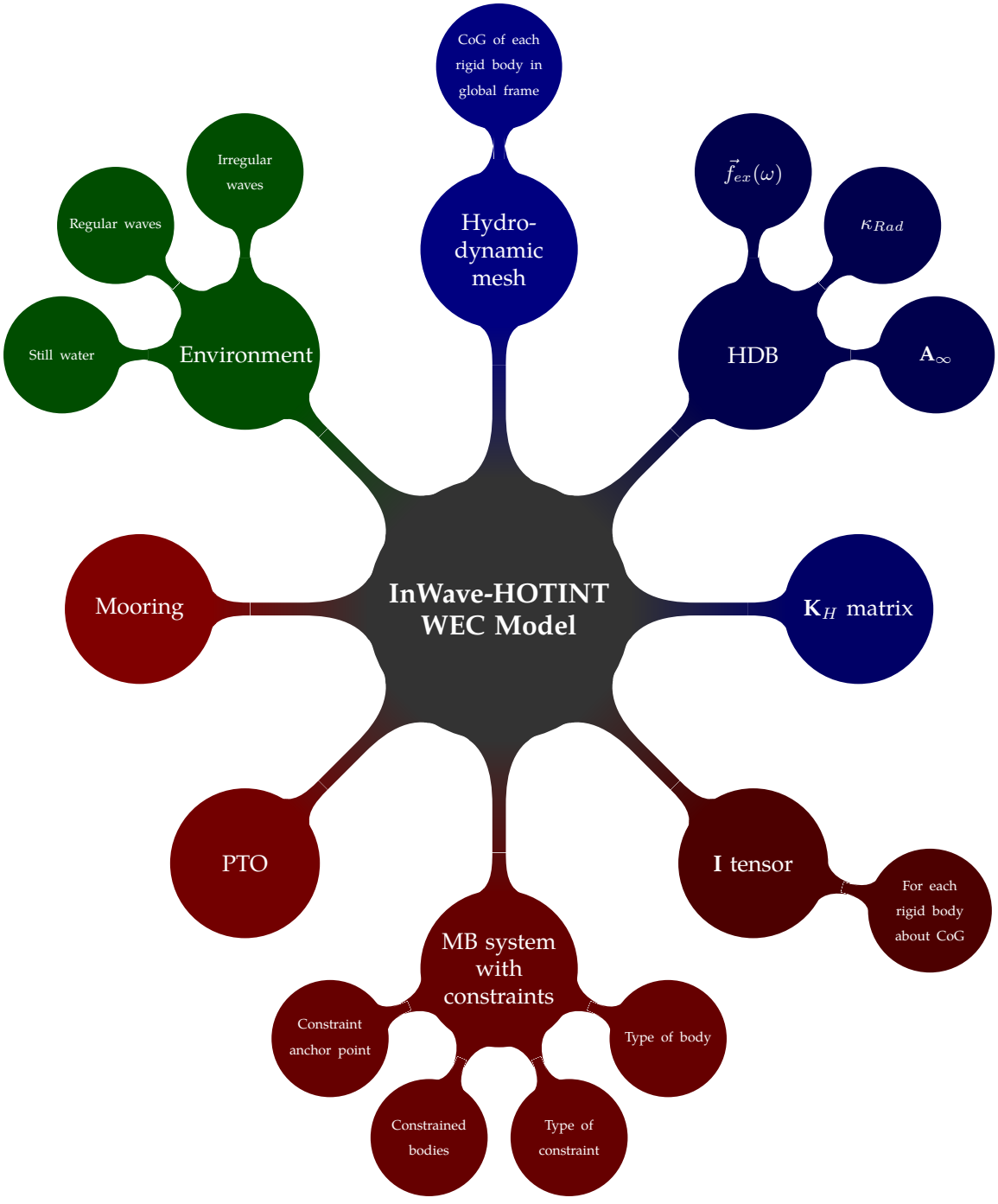


Figure 4.4: Overview of the different components of a typical WEC model created with InWave-HOTINT.

4.3 Creating a Model I: BEM Model

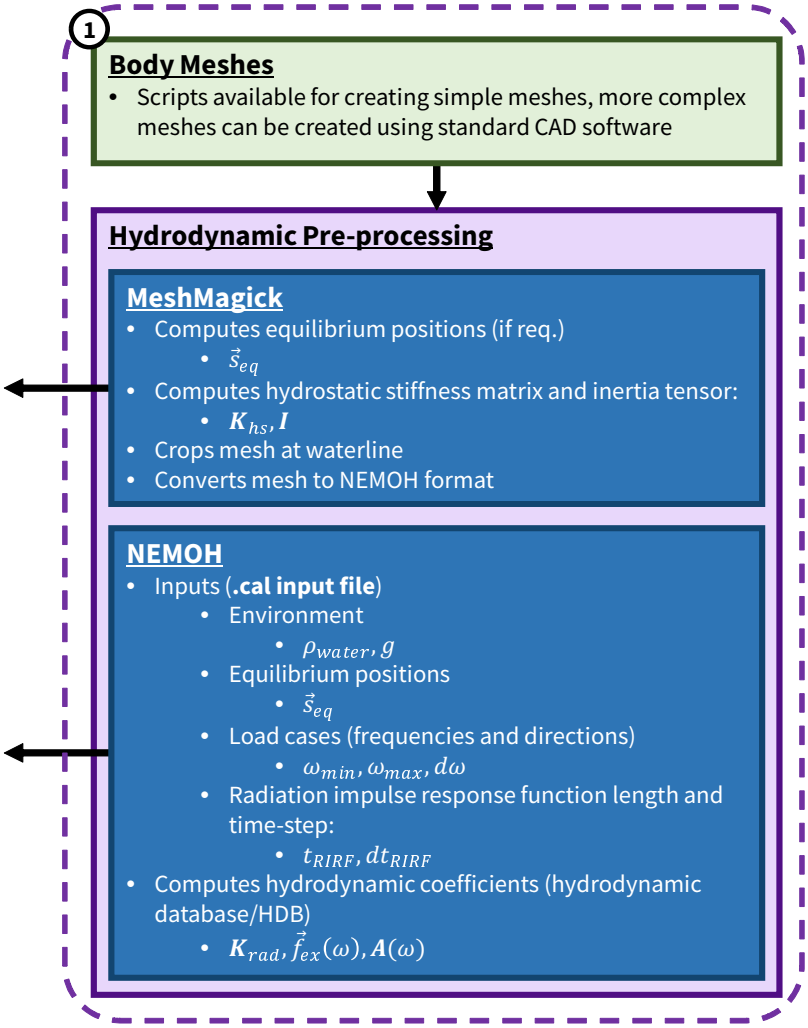


Figure 4.5: Hydrodynamic pre-processing.

4.3.1 Mesh Creation Process

For a multibody WEC, an individual mesh of each wetted rigid body in the system must be created. There are many CAD packages (e.g. Rhinoceros, SolidWorks) available for modelling rigid bodies, which can easily export surface meshes. These meshes can be converted into a format compatible with NEMOH by using MeshMagick. Third-party CAD packages (and MeshMagick) can also be used to determine a body's centre of gravity, which can be especially useful for bodies that have complex geometries and/or non-uniform weight distributions. The CoGs are required for the next stages

of modelling.

4.3.1.1 Hydrostatic Equilibrium Position

Before being used by NEMOH, the mesh must be positioned so that it represents the WEC in hydrostatic equilibrium (i.e. the position at which the system's buoyancy force vector is equal to its weight and any other forces which may affect its equilibrium position, such as mooring line weight, taut mooring lines or foundations). For simple geometries this position may be determined analytically. However, more complex systems may require a numerical solution. Both MeshMagick and InWave offer a solution to this problem: a time-domain simulation can be performed with no hydrodynamic loads - only weight, buoyancy and moorings, to determine the system's equilibrium position. However, MeshMagick can only perform this calculation for single body systems, whereas InWave can compute the equilbrium position of a multibody system. The same routine has not yet been required in InWave-HOTINT, but can be included in the future if required.

Using flat 2D panels to model a curved surface leads to errors when calculating the volume of the mesh. In turn, this can lead to small errors in the body's hydrostatic equilibrium position, as the buoyancy computed from the mesh volume is different to the 3D body/the physical model. Hence, care must be taken in the modelling process to ensure that the numerical model's centre of gravity position at equilibrium is consistent with the hydrostatic equilibrium results (either computed from a mesh or known a priori). This is particularly important when comparing results to other codes or physical model data. In many instances, this requires bypassing the numerical solution of the equilibrium positions and assigning their values directly (see Section 5.3 for further information and an example).

Once the hydrostatic equilibrium position has been determined, the mesh must be clipped at the undisturbed water surface, so that only the submerged part of the body is meshed. Panels above the waterline are not required by BEM solvers, but NEMOH does not automatically clip the mesh - this must be done beforehand with MeshMagick.

4.3.2 Computation of Hydrostatic Stiffness Matrices (K_H) and Inertia Tensors (I)

The hydrostatic stiffness matrix can be calculated according to the theory presented in Section 3.1.1. Although $\mathbf{K}_H$ can be determined by hand for simple geometries, it is more robust to use a program to compute the matrix. This can help to avoid human error and is not restricted by complex geometries. MeshMagick has been used in InWave-HOTINT to calculate $\mathbf{K}_H$ for individual bodies in the system and can also be used to compute the inertia tensor, **I**, of a rigid body (with uniform density), if required. For InWave-HOTINT, **I** must be computed about the body's centre of gravity and expressed in the global frame.

4.3.3 Computing the Hydrodynamic Coefficients

Once the mesh of each body in the system has been created and prepared correctly, it can then be used by NEMOH to compute the hydrodynamic coefficients (added mass $\mathbf{A}(\omega)$, radiation damping $\mathbf{B}(\omega)$, diffraction $\vec{f}_{diff}(\omega)$ and Froude-Krylov $\vec{f}_{FK}(\omega)$) over a range of different wave frequencies.

Each body (complete with individual mesh at its equilibrium position and centre of gravity location in global frame) must be defined in the NEMOH model. Joints, moorings and other mechanical features are not required at this stage. The density of the water must be specified (typically $1025kg/m^3$ for sea water and $1000kg/m^3$ for tank water) - making sure that this is consistent with the water density used to compute the hydrostatic equilibrium position.

In theory the hydrodynamic coefficients should be evaluated up to infinite frequency, but as $\mathbf{B}(\omega)$ values converge relatively quickly, the data may be truncated. The highest frequency that may be computed depends on the average panel size in comparison to the wave length. Experience with InWave has determined that for many full scale WEC numerical models, a range of $0.02 \leq \omega \leq 6.0rad/s$ (i.e. $T_P \approx 1.0 \rightarrow 30.0s$) is sufficient, and a frequency step of $\omega = 0.02rad/s$ usually provides sufficient detail to enable accurate integration of the radiation damping. The hydrodynamic pre-processing routine is shown in Figure 4.6.

NEMOH also provides the option to compute the radiation impulse response functions, $\mathbf{K}_{rad}$ by cosine transformation of the radiation damping coefficients (see Section 3.1). At present, the length and time-step of the RIRFs must be determined manually and are system-dependent. The length of the RIRFs can be critical: integrating the radiation damping using trapezoidal methods results in a periodical RIRF with a period equal to the inverse of the frequency spacing: $T = {}^{2\pi}/_{\delta\omega}$ and must be truncated at a point where the functions have sufficiently decayed. As explained in Section 4.5, the time-step used for $\mathbf{K}_{rad}$ must be the same as the time-step used in the time-domain solver.

Input:
Mesh and mass of each body in the multibody system
Environment information (ρ_{water}, d_{water})
Number of waves and range of frequencies ($n\omega$, ω_{min}, ω_{max})
Radiation impulse response function (RIRF) length & time-step (t_{RIRF}, dt_{RIRF})
Output:
Hydrostatic properties ($\vec{s}_{eq}$, $\mathbf{K}_H$)
HDB ($\vec{f}_{ex}(\omega)$, $\mathbf{A}(\infty)$, $\mathbf{B}(\omega)$, $\mathbf{K}_{rad}$)
foreach *body* $\in$ $system$ **do**
  if floating body is not in hydrostatic equilbrium **then**
    Determine equilibrium position
    Compute hydrostatic stiffness matrix, $\mathbf{K}_H$
  end
  Cut mesh in equilibrium position at the waterline
end
for $system$ **do**
  Compute $\vec{f}_{ex}(\omega)$
  Compute added mass at infinite frequency ($\mathbf{A}(\infty)$)
  Compute radiation damping ($\mathbf{B}(\omega)$)
  Compute RIRFs $\mathbf{K}_{rad}[ndof, ndof, t_{RIRF}]$
end

Figure 4.6: Pre-processing routine for computing multibody system's hydrostatic properties (MeshMagick) and hydrodynamic coefficients (NEMOH).

4.4 Creating a Model II: Multibody Model

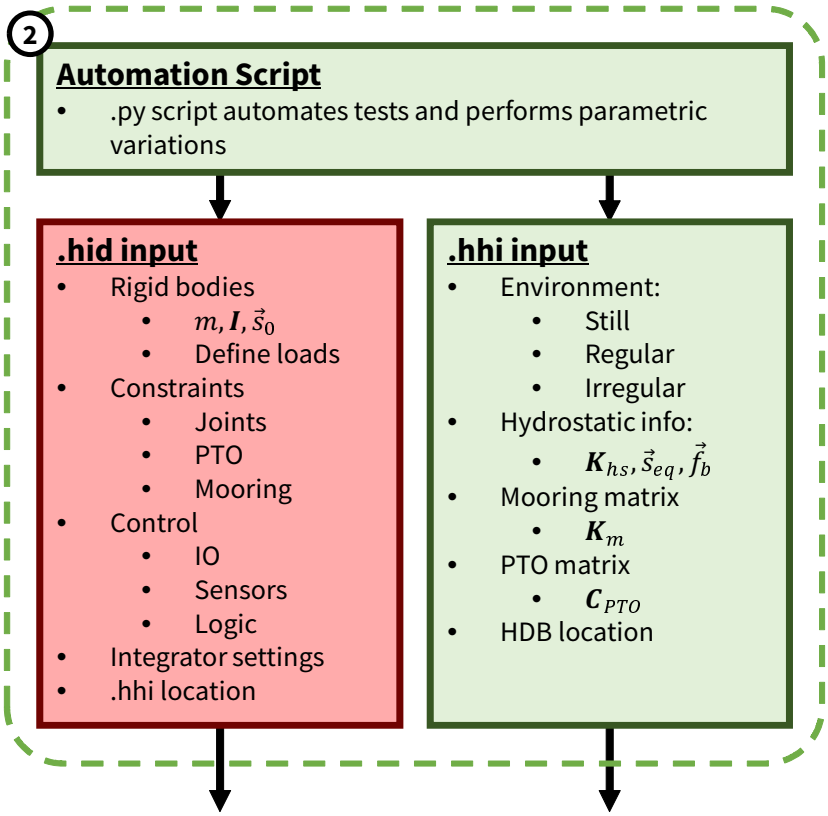


Figure 4.7: Time-domain model inputs.

4.4.1 Setting up the Multibody Model with Constraints

The multibody model consists of the rigid bodies included in the NEMOH model, along with any additional bodies (which are not under the application of wave loads), joints, PTOs, mooring ropes and control systems. The multibody model is created in HOTINT, using its built-in scripting language. Some of the available elements are shown in Figure 4.8. A more complete list is presented in the HOTINT documentation (Gerstmayr, Aigner, et al. 2013).

As mentioned previously, hydrodynamic loads can only be applied to *rigid bodies* at present. These bodies must be defined with the same centre of gravity positions used in the NEMOH model. There are two options available for rigid bodies: `Rigid3DKardan` (sic), which expresses orientation with Tait-Bryan angles and `Rigid3D`, which expresses

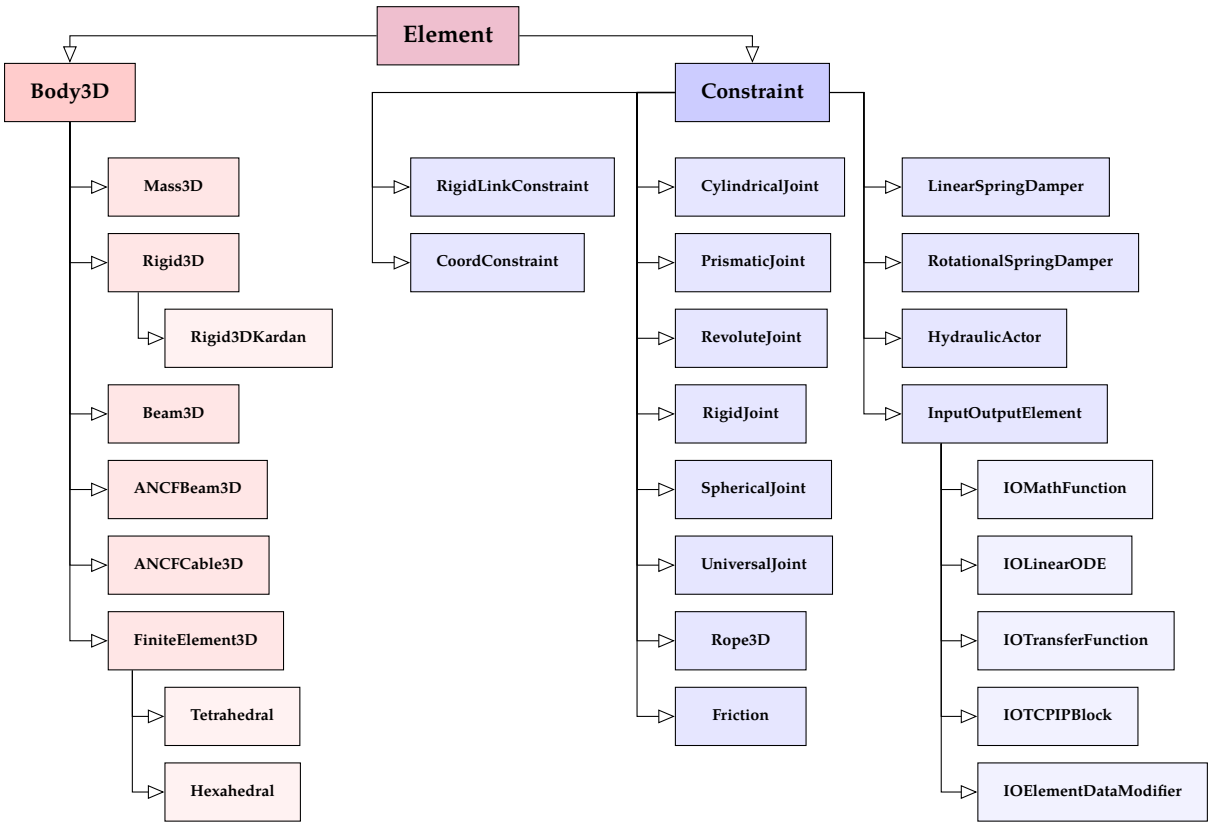


Figure 4.8: HOTINT class diagram showing selected elements that can be used to create a multibody WEC model (adapted from Gerstmayr, Aigner, et al. (2013)).

orientation with quaternions. However, `Rigid3D` elements are only available for 1-body systems at present.

Any of the constraint elements can be included in the model, and constraints can be combined freely in the system (for example, a linear spring-damper can be added on top of a prismatic joint to model a linear translational PTO). Full descriptions of each element's characteristics are available in the HOTINT documentation (Gerstmayr, Aigner, et al. 2013). `CoordConstraint` can be useful in a WEC model to eliminate a degree of freedom (either global or relative between bodies). This can be used to reduce numerical drift errors in unconstrained DoFs (for example, a buoy's sway dof in waves with a zero heading angle).

`InputOutputElement` blocks can be used to model more complex sub-systems in the model, such as non-linear PTOs, control systems and active moorings. For example, `Rope3D` elements, in addition to some fixed stiffness and damping properties, include

a variable amount of coiled (or spooled) rope; this variable can be modified at each time step using `IOElementDataModifier`. Additionally, sensors can be included in the model for a wide range of different parameters, and can send data to other `InputOutputElement` blocks in order to implement some control logic. For more complex control systems, `IOTCPIPBlock` can be used to pass system variables to external software (e.g. Simulink), to perform some additional computations and return a control force.

4.4.2 Modelling Power Take-Off (PTO) and Control Systems

In InWave-HOTINT, there are three options available for modelling a WEC's PTO. Firstly, the standard method already used in InWave can be used - specifying a 'PTO matrix', $\mathbf{C}_{PTO}$ in the `.hhi` file, which contains a constant damping value for a particular degree of freedom and is multiplied with the system's velocity vector to compute a linear PTO damping force vector at each time-step:

$$\vec{f}_{PTO}(t) = \mathbf{C}_{PTO} \cdot \dot{\vec{s}}(t) \tag{4.1}$$

However, this approach must be used with caution as the velocity vector $\dot{\vec{s}}(t)$ contains the 6 velocity components of each body's centre of gravity *in the global frame*. In other words, the relative velocity between bodies is not accounted for in this approach, which is typically the information required to compute the damping force in a PTO connecting 2 bodies. Hence, this approach can only be utilized when the relative velocity between bodies is equivalent to one of the body's velocities in the global frame. Despite this limitation, the approach can be useful in InWave-HOTINT for verifying results against some existing InWave test cases such as heaving buoys (one body, only the heave motion is considered) and bottom fixed oscillating wave surge converters.

A more robust approach is to model the PTO using standard HOTINT elements defined in the .hid input file, as explained in Section 4.4.1. If this approach is taken, there is no need to specify a PTO matrix in the `.hhi` input file. Basic linear prismatic or rotational PTOs can be modelled by applying `PrismaticJoint` or `RevoluteJoint` to the connected bodies - this applies an exact kinematic constraint using the Lagrange multiplier method, and then adding `LinearSpringDamper` or `RotationalSpringDamper` in

addition to the kinematic constraint.

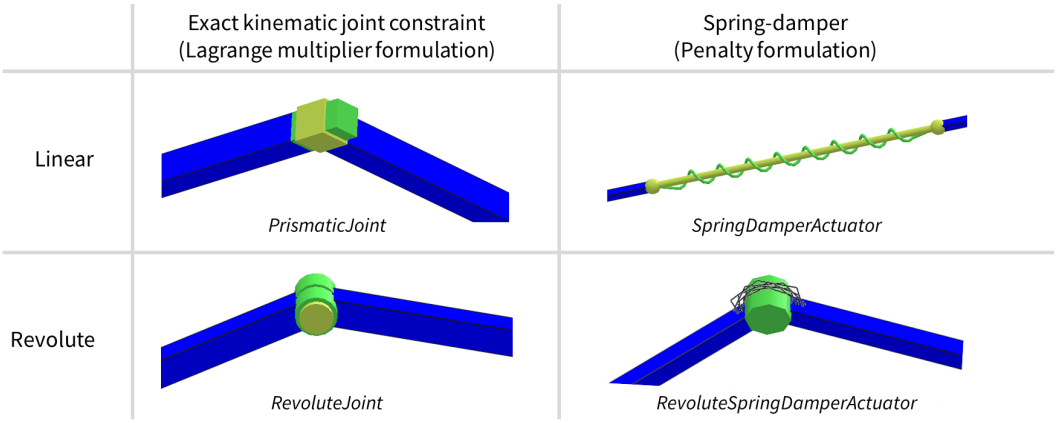


Figure 4.9: Overview of some key HOTINT elements that can be used to model common PTOs (pictures from Gerstmayr, Aigner, et al. (2013)).

The spring-damper elements are realized in a penalty formulation and apply a spring and/or damping force to the constrained bodies, based on their relative motion. In addition, an actuator force ($\vec{f_a}(a)$) can be included to apply a control force. For the linear spring-damper element (constant stiffness and damping):

$$f_{PTO}(t) = K_{PTO} \cdot \Delta x + C_{PTO} \cdot v + \vec{f}(a) \tag{4.2}$$

Where,

$$\Delta x = l - l_0 = (\vec{p}^{(1)} - \vec{p}^{(2)})^T \vec{dir} - l_0 \tag{4.3}$$

Where $\vec{p}^j$ denotes the position vectors of the joint attachment points. l_0 is the spring length, $\vec{dir}$ is the unit vector from $vecp^{(2)}$ to $vecp^{(1)}$ and v is the spring velocity:

$$v = (\dot{\vec{p}}^{(1)} - \dot{\vec{p}}^{(2)})^T \vec{dir} \tag{4.4}$$

$$\vec{dir} = \frac{\vec{p}^{(1)} - \vec{p}^{(2)}}{\sqrt{(p_x^{(1)} - p_x^{(2)})^2 + (p_y^{(1)} - p_y^{(2)})^2 + (p_z^{(1)} - p_z^{(2)})^2}} \tag{4.5}$$

For non-linear PTOs, $K_{PTO,i}$ and $C_{PTO,i}$ can be expressed as functions of position and

velocity, respectively:

$$\vec{f}_{PTO,i}(t) = K_{PTO,i}(\Delta x) \cdot \Delta x + C_{PTO,i}(v) \cdot v + \vec{f}(b) \tag{4.6}$$

Similarly, the spring and damping force components may be expressed as functions of system variables available at the current time-step, t (in other words, the `SpringDamper` element cannot access a variable's complete time history). This approach can be used to model fully non-linear behaviour:

$$\vec{f}_{PTO,i}(t) = \vec{f}_{K,i}(t, \vec{s}_i, \ldots) + \vec{f}_{C,i}(t, \dot{\vec{s}}_i, \ldots) \tag{4.7}$$

Multiple `SpringDamper` elements can also be combined with ball joints and universal joints to model more complex PTOs that have multiple degrees of freedom (see Section 5.4). Stiction and friction can also be applied to any degree of freedom in the system by using the `FrictionConstraint` element.

For modelling control systems, HOTINT's `Sensors` can be used to track variables throughout the system and provide inputs to a controller model. HOTINT's `IOElement` blocks can be used to model controllers and return a control force. It is also possible to use `IOElement` blocks to directly modify a position or velocity variable in the system.

For more advanced PTO and control systems modelling (or to incorporate existing PTO/control systems that have been modelled in a separate program such as Simulink), `IOTCPIPBlock` elements can be used to pass data to an external program via TCP/IP connection to perform co-simulations[7].

4.4.3 Modelling Mooring Systems

At present there are two approaches to modelling mooring systems with InWave-HOTINT. The constant mooring matrix approach originally used in InWave can be

[7]The HOTINT documentation states that this feature also opens up the possibility of performing hardware-in-the-loop simulations, which could be a useful feature for WEC developers.

taken with InWave-HOTINT, by defining the mooring matrix $\mathbf{K}_M$ in the `.hhi` input file. As with the PTO matrix approach described in the previous section, this approach is especially useful for comparing model results between InWave-HOTINT and InWave (and other codes/models that have used a constant mooring matrix approach) and for simulating decay tests, as the initial (displaced) position and the equilibrium position can both be defined explicitly:

$$\vec{f}_M(t) = \mathbf{K}_M \cdot (\vec{s}(t) - \vec{s}_{eq}) \tag{4.8}$$

It is also possible to model moorings using standard HOTINT elements directly in the `.hid` input file, as explained in Section 4.4.1. `Rope3D` elements can be used to model elastic ropes that are always under tension. They use a penalty formulation, similar to the spring-damper described by Equation 4.2 (without the additional actuator force, $\vec{f}_a$). This element can be attached to rigid bodies and pulleys (as shown in Figure 4.10), and also provides an option to spool the rope (i.e. to model a winch).

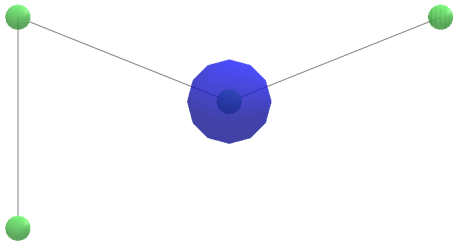


Figure 4.10: A HOTINT `Rope3D` element, demonstrating pulley (green) and point mass (blue) connections (pictures from Gerstmayr, Aigner, et al. (2013)).

Multiple `Rope3D` elements can be connected together with point masses (`Mass3D`) to create more complex net mooring configurations (demonstrated in Section 5.4).

`Rope3D` elements do not allow the equilibrium length of the rope to be defined explicitly, which can make decay tests more awkward to perform. Essentially, in the displaced position at $t = 0s$, the rope will not experience a restoring force to return it to its equilibrium position (it effectively assumes that the user-defined initial position is the equilibrium position). This problem can be overcome by starting the simulation with the bodies in their equilibrium positions, and then applying a controlled displacement force to the desired bodies to recreate the decay test.

To date, only linear mooring models have been considered in InWave-HOTINT. However, `Rope3D` elements could be used to discretize lengths of mooring line rope in a higher fidelity mooring model - such as the lumped mass approach described in Section 2.3.1[8].

4.5 Time-domain Algorithms

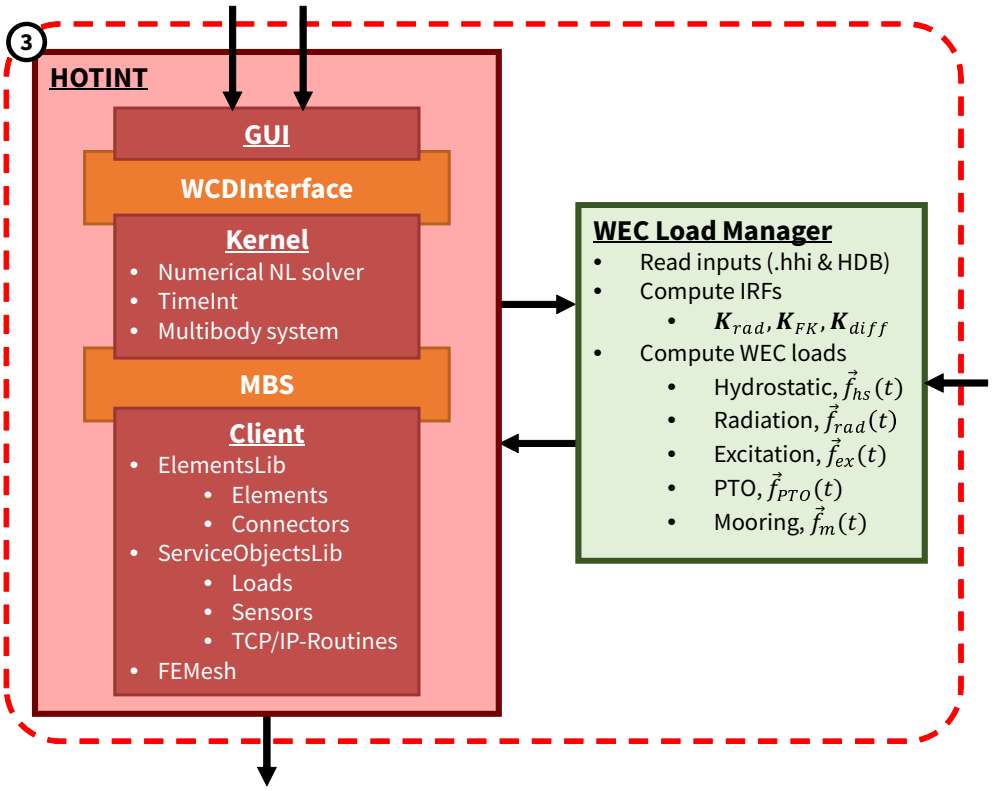


Figure 4.11: Time-domain algorithms.

When a `.hid` model is opened via the HOTINT GUI, the location of the `.hhi` file is immediately read and stored in memory. When the simulation is started via the GUI, the HDB and other data are read and stored in memory[9]. The GUI interfaces with the rest of the HOTINT code via the WCDInterface module (`WinCompDriverInterface`).

The main access point in the HOTINT codebase is the kernel, where the time integrator calls two key functions: `EvalM()` and `EvalF2()`, which update the system's mass

[8]For fully dynamic mooring models with torsion in the rope HOTINT's `ANCFCable3D` elements could be potential option in the future.

[9]This may take a few minutes for large databases, which can be several gigabytes.

matrix and second-order force vector, respectively. HOTINT's implicit Runge-Kutta integration method calls these functions several times within a single time-step as the algorithm converges on a solution. However, the hydrodynamic forces are functions of the kinematic data from the previous time-step, and therefore only need to be called by `EvalF2()` once - at the start of each time step (at the first iteration of the integration algorithm).

The full added mass at infinite frequency matrix, $\mathbf{A}(\infty)$, can be included directly in the system's multibody mass matrix by modifying the `EvalM()` function to perform $\mathbf{M} + \mathbf{A}(\infty)$ at each time-step. As discussed in Chapter 3, the mass matrix of each rigid body may either be 6×6 or 7×7 depending on whether the orientations are expressed using Tait-Bryan angles or quaternions. As NEMOH uses Tait-Bryan angles to express rotations, the added mass matrix of a floating body has dimensions 6×6. Therefore, if quaternions are used in the multibody model, $\mathbf{A}(\infty)$ must be converted from 6×6 to 7×7 using the quaternion transformation matrix $\mathbf{G}$ (introduced in Equation 3.76):

$$\mathbf{A}(\infty)^*_{01} = \mathbf{A}(\infty)_{01} \cdot \mathbf{G} \tag{4.9}$$

$$\mathbf{A}(\infty)^*_{10} = \mathbf{G}^T \cdot \mathbf{A}(\infty)_{10} \tag{4.10}$$

$$\mathbf{A}(\infty)^*_{11} = \mathbf{G}^T \cdot \mathbf{A}(\infty)_{10} \cdot \mathbf{G} \tag{4.11}$$

This process is illustrated in Figure 4.12.

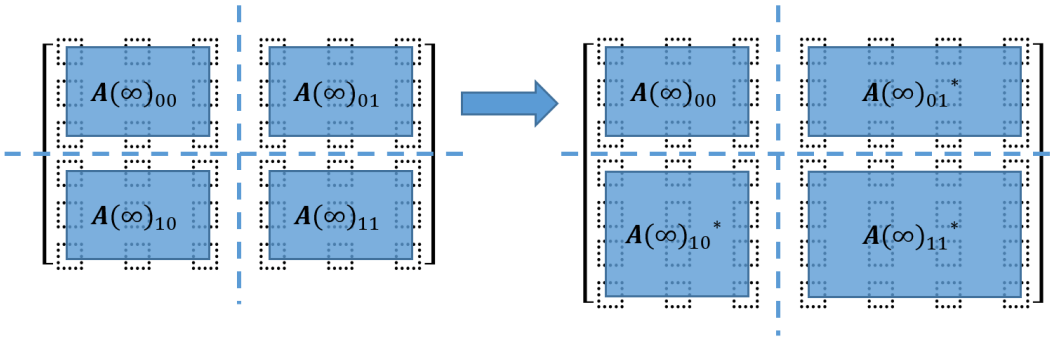


Figure 4.12: Conversion of $[6 \times 6]$ $\mathbf{A}_\infty$ matrix to $[7 \times 7]$ quaternion-based format.

At present, this feature has only been implemented for systems with one floating body. Although quaternions have some advantages over Tait-Bryan angles (most importantly

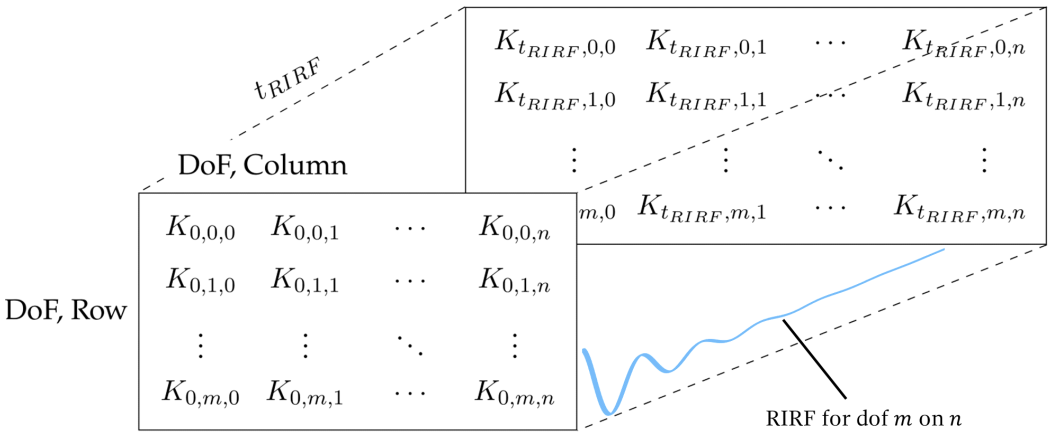


Figure 4.13: Illustration of the radiation impulse response function's (RIRF) 3D matrix structure.

the avoidance of gimbal lock), there is additional computational expense involved in converting the added mass matrix to a quaternion-based format. As linear potential flow theory assumes small amplitude motion, the risk of gimbal lock is very small. However, the quaternion option might become useful in the future if gimbal lock becomes a problem. It should also be noted that HOTINT allows rigid bodies described with Tait-Bryan angles and quaternions to be combined within the same system[10]. Hence, it is possible to have a system of 'dry' rigid bodies (attached to a floating rigid body) expressed with quaternions - this could be a useful approach in the future if InWave-HOTINT is used for modelling offshore systems other than WECs, such as floating cranes and wind turbines, where the risk of gimbal lock occurring is potentially higher due to the large ranges of motion experienced by some of the 'dry' bodies in the system.

As explained in Section 4.3.3, NEMOH also provides the option to compute the radiation impulse response functions, $\mathbf{K}_{rad}$ by cosine transformation of the radiation damping. If this option is taken, $\mathbf{K}_{rad}$ is written in plain text format by NEMOH, but must be read into the 3D matrix format shown in Figure 4.13 for use with the convolution algorithm. Alternatively, $\mathbf{K}_{rad}$ can be computed by the WEC load manager at $t = 0s$, using the same approach.

At present the $\mathbf{K}_{rad}$ time-step must equal the time-step that will be used in the time-domain simulation, as there is currently no function to interpolate between $\mathbf{K}_{rad}$ values

[10]HOTINT refers to these elements as 'Rigid3DKardan' (sic) and 'Rigid3D' elements, respectively.

[11].

The system's velocity vector must be stored at each time-step in order to successfully perform the convolution integral. The velocity 'history' time series is stored in a 2D matrix, with dimensions $[t_{sim}/dt, n_{dofs}]$, and is initialized at $t = 0s$.

[11] Although this can be changed by interpolating the RIRFs during the computation of $\vec{f}_{rad}(t)$ at each time-step.

Chapter 5

Software Verification & Capabilities

This Chapter presents the main results obtained with InWave-HOTINT. Sections 5.1 to 5.3 present the results of 3 case-studies used throughout the development of InWave-HOTINT to continuously verify the accuracy of the code against other WEC modelling codes. The other codes that have been used for comparison include:

- InWave
- ProteusDS
- WaveDyn
- WEC-Sim

Each of these codes has been validated independently using experimental data from WEC physical modelling campaigns (see Section 2.4.2), which has helped to improve confidence in these tools. Code to code comparisons are still extremely useful though: helping to reveal discrepancies in the results that may arise from incorrect user inputs or errors in the code. Having access to numerical data permits a wide range of variables to be compared (e.g. individual $\vec{f}_{rad}$ or $\vec{f}_{hs}$ components) that may be difficult or impossible to record from physical experiments. Having access to the original InWave code has been especially useful in this regard.

Throughout the verification process, the user-friendliness of InWave-HOTINT has been continuously assessed and improved: from hard-coding some variables in the initial prototyping, to reading model input parameters from a text input file, to finally providing support for Python-generated batch input files and Python post-processing routines.

The 3 verification case-studies in this chapter are presented in order of increasing complexity: from the single body heaving sphere, to a 2-body oscillating wave surge converter (OWSC) with fixed base, to a 3-body floating OWSC (Table 5.1).

Table 5.1: Overview of the InWave-HOTINT verification case-studies.

Case-study	**Compared with**	**Reference**
Sphere	28 different codes from around the world	Wendt et al. (2017)
B-OF	InWave, NumWEC	Babarit, Hals, et al. (2011)
F-3OF	InWave, WaveDyn, ProteusDS, WEC-Sim	Combourieu, Lawson, et al. (2015)

In each of these cases an existing mesh has been used; the same mesh as used by the other codes[1]. This is done to remove the mesh as a possible source of discrepancy between the codes when comparing their results.

In addition to checking the accuracy of InWave-HOTINT and improving the user-friendliness, the verification process exposed some common errors that can be encountered in the WEC modelling process (independent of the software package used). This has enabled the identification of critical model parameters that must be checked before running time-domain simulations.

Section 5.4 describes the InWave-HOTINT modelling process and demonstrates some of the new capabilities by modelling a device featuring closed mechanical loops, a net mooring system and bi-directional power take-offs. The model is inspired by the Albatern WEC but has not been experimentally validated for reasons explained in Chap-

[1] Except for the B-OF case: the same mesh as InWave is used, but the original NumWEC mesh is slightly different, featuring chamfers close to the hinge joint (Figure 2.3).

ter 1.

This chapter focuses on presenting the results: Chapter 6 provides a more detailed interpretation and synthesis of these results.

5.1 Case-Study 1: Sphere

5.1.1 Background

The Sphere test case-was recently used in the International Energy Association's (IEA) Ocean Energy Systems (OES) project: "Task 10: Wave Energy Converters Modelling Verification and Validation" (Wendt et al. 2017), with the aim that a simple system would make code-to-code comparison more straightforward[2] and provide a platform to compare and verify more complex models in the future. A very fine mesh of half of the Sphere's submerged surface was provided to the project's participants (Figure 5.1).

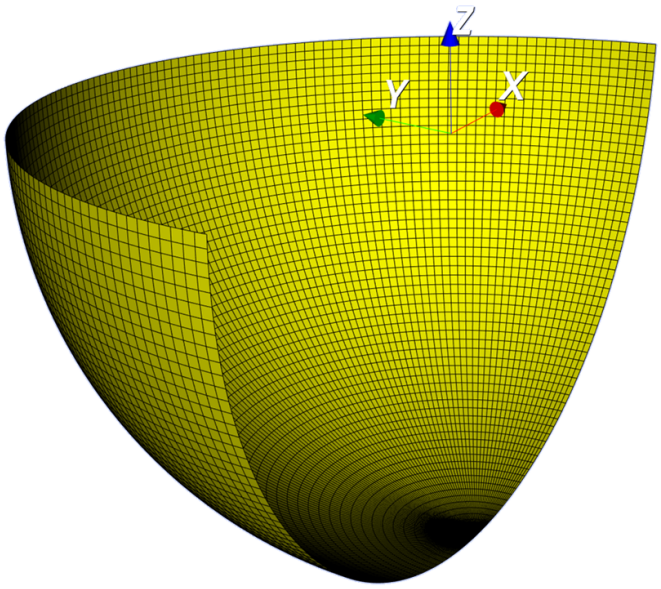


Figure 5.1: Half-mesh of the sphere's submerged surface (7200 panels).

Key properties required to create the model are shown in Table 5.2. The Sphere is only permitted to heave, hence because it cannot rotate, the inertia tensor is not required.

Of the 28 codes compared by Wendt et al. (2017), 4 have been selected for ease of com-

[2]by mitigating potential issues related to communication, data exchange and uncertainties in model definition.

Table 5.2: Properties of the Sphere.

General		
Water depth	∞	m
Water density	1000	kg/m^3
Sphere		
Mass	$261.8 \cdot 10^3$	kg
Radius	5	m
Origin	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	m
CoG	$\begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}$	m

parison: InWave, ProteusDS, WEC-Sim and aNySIM. Further information on the first three of these codes is available in Section 2.4.2. aNySIM is developed by the Maritime Research Institute Netherlands (MARIN).

The WEC-Sim and aNySIM results both utilize non-linear hydrostatic and Froude-Krylov forces based on the instantaneous body position and wave elevation.

5.1.2 Development of the InWave-HOTINT Model

5.1.2.1 Hydrodynamic Pre-processing

Hydrodynamic coefficients were computed for 300 different frequencies over the range $0.02 \leq \omega \leq 6.0$ (rad/s) (with NEMOH). The length of the radiation impulse response functions, t_{RIRF} is $40.0s$.

The hydrostatic stiffness matrix of the Sphere, K_H, has been computed using Meshmagick (note that the rotational stiffness terms are not actually required in the computation due to the Sphere being only permitted to heave):

$$K_H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 770215 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5130589 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5130589 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} {}^{N}/_{m} \\ and \\ {}^{Nm}/_{rad} \end{matrix} \tag{5.1}$$

The mass of the Sphere has been provided as $261.8 \cdot 10^3$ kg (4 s.f.). This is based on the theoretical volume of the Sphere:

$$V = \frac{4}{3}\pi r^3 \tag{5.2}$$

Hence with a $5m$ radius, the submerged volume of the Sphere is $\forbidden{V}_{Sphere,th} = 261.799m^3$ (6 s.f.). However, the mesh of the Sphere, despite being very fine still contains small imperfections. Therefore, the computed volume of the Sphere is $\forbidden{V}_{Sphere,comp} = 261.724m^3$ (6 s.f.). In InWave-HOTINT the *theoretical* values (rounded to 4 s.f.) have been used to determine the Sphere's buoyancy vector:

$$\vec{f}_{buoyancy} = \rho \cdot g \cdot \forbidden{V} \tag{5.3}$$

$$\vec{f}_{buoyancy} = 1000 \cdot 9.81 \cdot 261.8 \tag{5.4}$$

$$\vec{f}_{buoyancy} = 2568258.0N \tag{5.5}$$

5.1.3 Decay Tests

The results of two free-decay tests (with no power take off or mooring) are presented here: from initial displacements of $+1m$ and $+5m$ in the heave direction.

The 1m decay test results are presented in Figure 5.2 and show good agreement be-

tween the participants. To investigate small discrepancies, a close up of the $30 - 40s$ region is shown in Figure 5.3. This figure reveals some variation in natural period, amplitude and equilibrium position. Although the differences are very small, they could be useful for identifying possible errors (either in the codes or the inputs) and potential improvements.

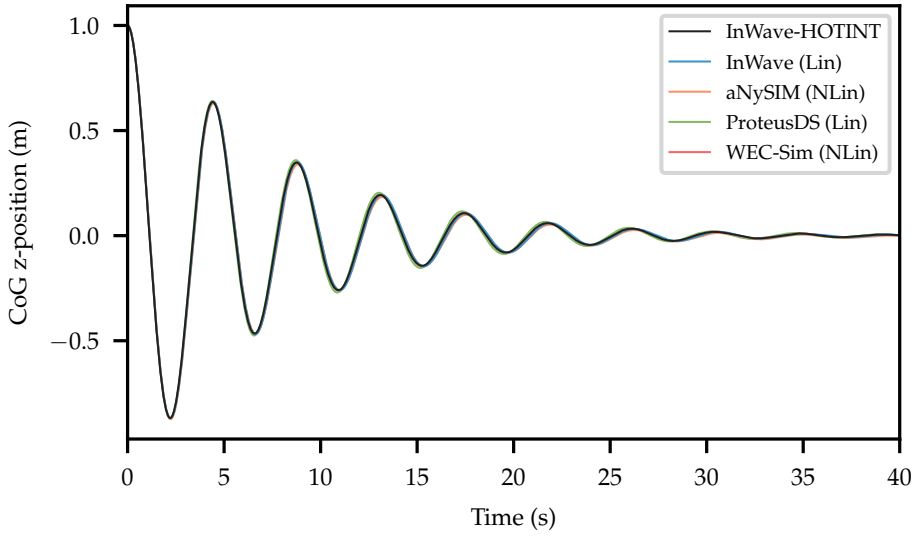


Figure 5.2: Sphere decay test (1m displacement in heave) - comparison of results.

As the radius of the Sphere is $5m$, the second decay test 'drops' the Sphere from a position where it is initially just touching the water surface. Hence, non-linear hydrostatic effects are much more significant in this test and the difference between the linear and weakly non-linear codes becomes more apparent (Figures 5.4 and 5.5).

The results of the non-linear codes show a longer natural period, which could be an indication that the Sphere's stiffness in heave is reduced due to the wetted surface area decreasing away from equilibrium (opposed to the linear codes, which assume a constant hydrostatic stiffness throughout the range of motion).

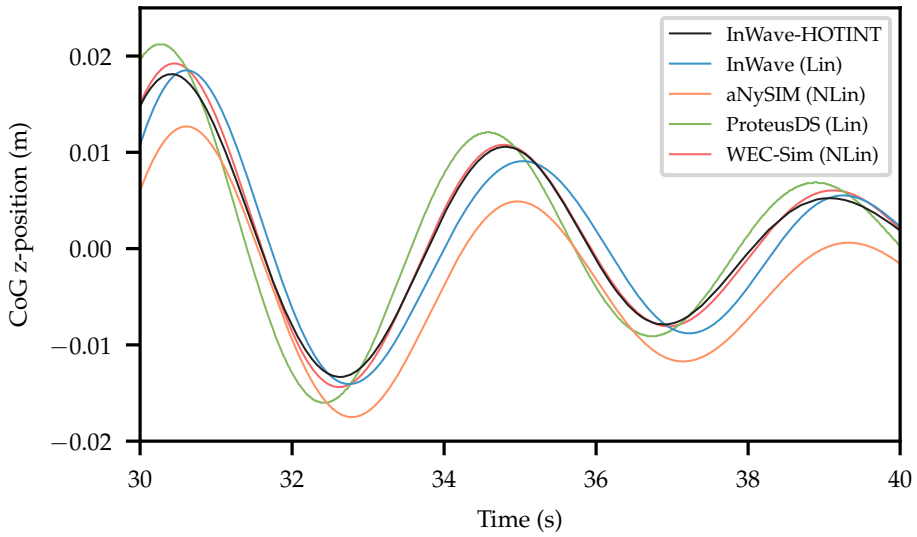


Figure 5.3: Sphere decay test (1m displacement in heave) - comparison of results (30-40s region).

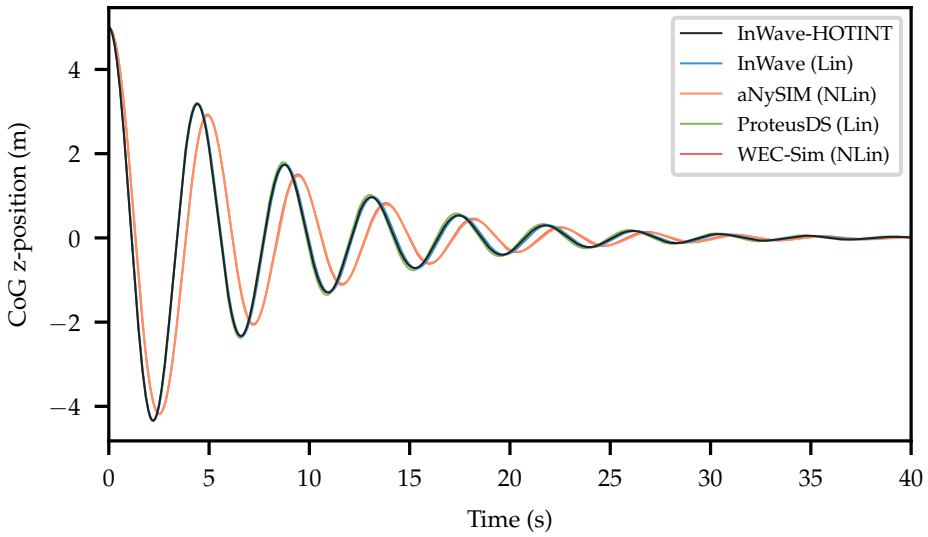


Figure 5.4: Sphere decay test (5m displacement in heave) - comparison of results.

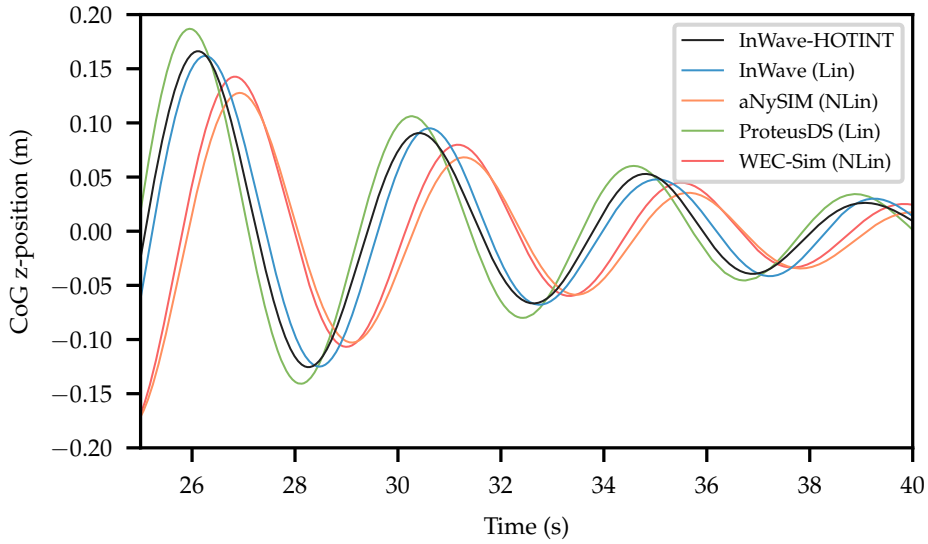


Figure 5.5: Sphere decay test (5m displacement in heave) - comparison of results (25-40s region).

5.1.4 RAOs

To investigate non-linear effects, the Sphere's response was computed over 10 different wave periods (T_p) for 3 levels of wave steepness ($S = 0.0005$, 0.002 and 0.01) and for 3 different PTO settings. Hence, 90 regular wave tests were performed in total. The wave height, H, has been calculated for deep water conditions according to the wave period and steepness:

$$S = \frac{H}{g \cdot T_p^2} \tag{5.6}$$

$$H = S \cdot g \cdot T_p^2 \tag{5.7}$$

The 3 different PTO configurations tested for the Sphere are as follows:

1. No PTO
2. Optimal PTO damping[3]:

[3]Based on linear theory (Tom, Lawson, Yu, and Wright 2016; Wendt et al. 2017)

$$C_{PTO,Opt} = B_{33}(\omega)\sqrt{1 + \left(\frac{K_{33} - \omega^2(m + A_{33})}{\omega \cdot B_{33}(\omega)}\right)^2}$$

Where B_{33}, K_{33} and A_{33} are radiation damping, hydrostatic stiffness and added mass in heave. m is the mass of the Sphere and ω is the wave frequency.

3. Fixed

The T_p, H and $C_{PTO,Opt}$ values used by each participant are shown in Table 5.3, each given to 3 d.p.

Table 5.3: Summary of regular wave conditions tested for the Sphere.

	H (m)			
T_p (s)	$S = 0.0005$	$S = 0.002$	$S = 0.01$	C_{opt} $^{Ns}/_{m}$
3.0	0.044	0.177	0.883	398736.034
4.0	0.078	0.314	1.570	118149.758
4.4	0.095	0.380	1.899	90080.857
5.0	0.123	0.491	2.453	161048.558
6.0	0.177	0.706	3.532	322292.419
7.0	0.240	0.961	4.807	479668.979
8.0	0.314	1.256	6.278	633979.761
9.0	0.397	1.589	7.946	784083.286
10.0	0.491	1.962	9.810	932117.647
11.0	0.594	2.374	11.870	1077123.445

Figures 5.6 to 5.8 show selected RAOs and illustrate that there is generally good agreement between each participant.

The clearest discrepancy arises in steep wave conditions with optimal PTO damping (Figure 5.8). Here there is a separation between the linear and non-linear codes at the longer periods (from $T_p > 7s$), which is not observed in the 'no PTO' results (Figure 5.7). In long waves, with no PTO, the Sphere behaves as a wave follower, which mitigates non-linear effects arising from the instantaneous position of the Sphere and the water surface. The presence of a PTO restricts the motion of the Sphere; making it more difficult to 'follow' the longer waves and hence more susceptible to non-linear effects.

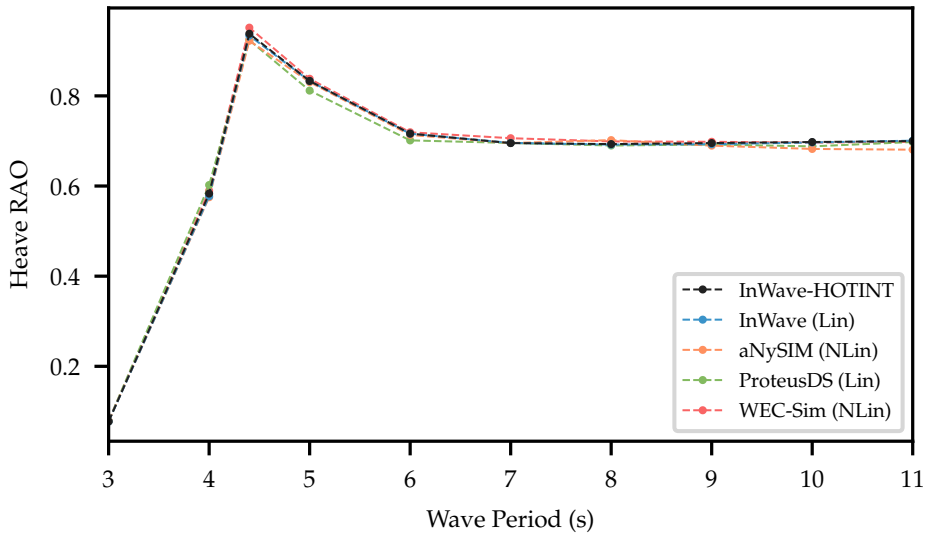


Figure 5.6: Sphere RAO comparison (optimal PTO damping, $S = 0.002$).

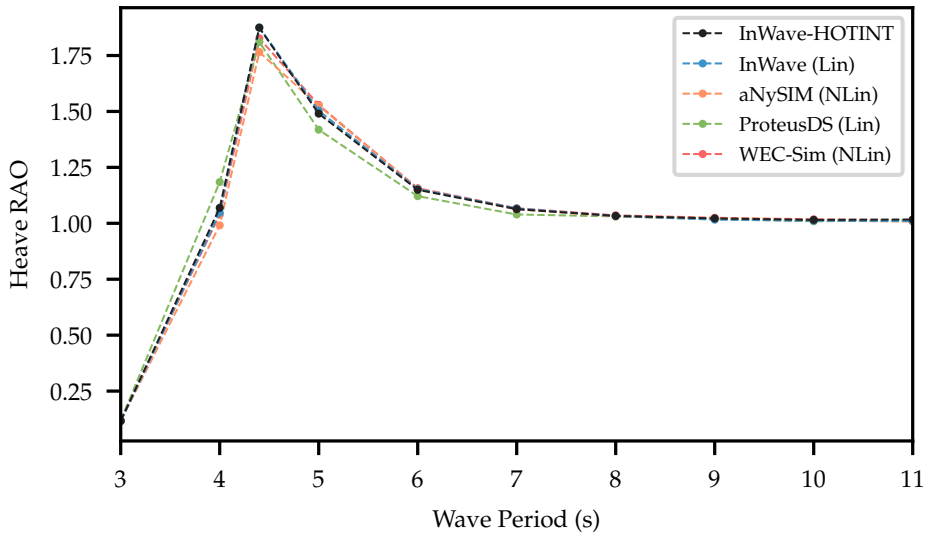


Figure 5.7: Sphere RAO comparison (no PTO, $S = 0.01$).

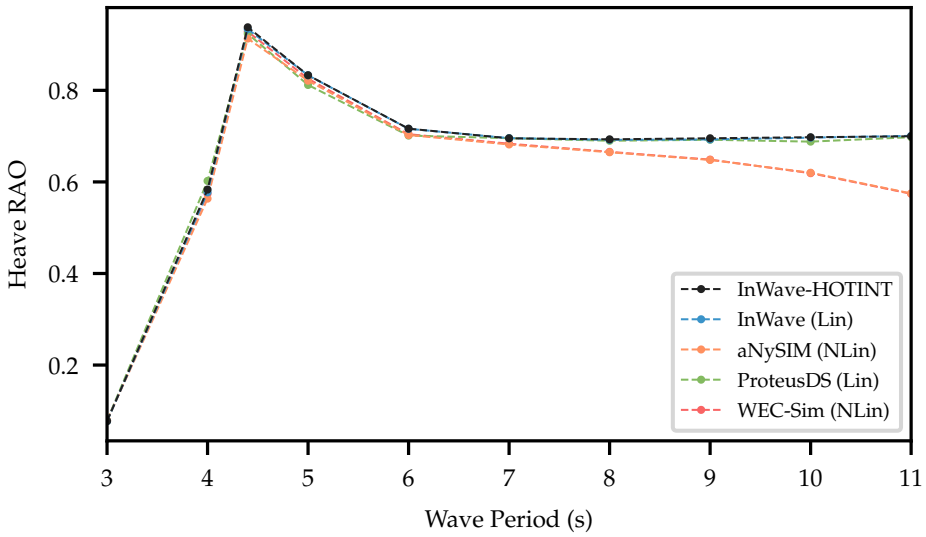


Figure 5.8: Sphere RAO comparison (optimal PTO damping, $S = 0.01$).

5.2 Case-Study 2: B-OF

5.2.1 Background

The B-OF ("bottom-fixed oscillating flap") test case was originally developed by Babarit, Hals, et al. (2011) to assess the performance of a generic oscillating wave surge converter (OWSC). It is loosely based on Aquamarine's Oyster 2 device (Figure 5.9). The model features 2 bodies, although one of the bodies represents the WEC's base and is fixed to the seabed.

Figure 5.9: Aquamarine Oyster800 WEC (picture from Babarit, Hals, et al. (2011)).

The original mesh developed by Babarit, Hals, et al. (2011) featured chamfers on both bodies at the hinge joint. However, the InWave test case neglected these chamfers and modelled both bodies as cuboids (Figure 5.10). For verifying InWave-HOTINT, the same mesh as InWave has been used. Although it is relatively coarse and has not been refined near the edges/free surface, Figure 2.3 shows that the hydrodynamic coefficients computed for the coarser B-OF meshes only begin to diverge from around $6rad/s$.

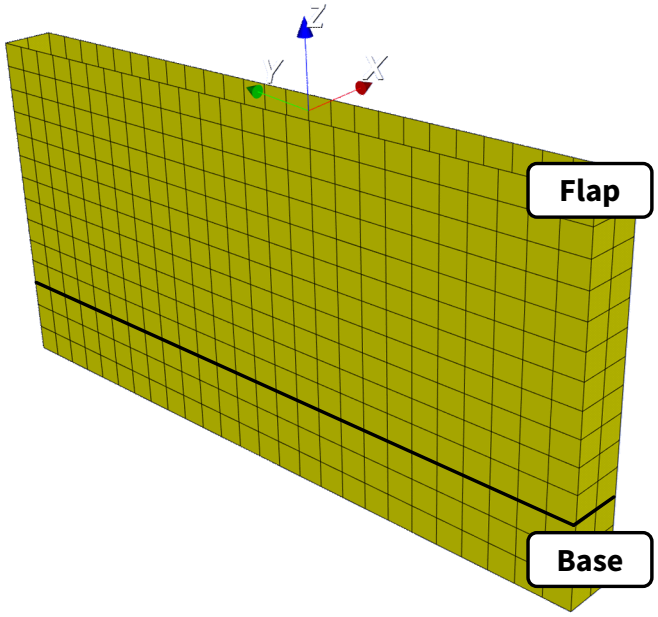


Figure 5.10: Submerged mesh of the B-OF (flap: 560 panels, base: 224 panels).

Unlike the Sphere and F-3OF, the B-OF model has not been used in any formal code-to-code comparison projects, and therefore the InWave-HOTINT results can only be verified against NumWEC and InWave for this case-study:

- NumWEC: linear hydrostatic and excitation force, viscous damping effects not included in the time-domain RAO results.
- InWave: non-linear hydrostatic solver is used to compute $\vec{f}_{hs}$. Linear excitation force, no viscous damping.

Hence, the main difference in the models is the presence of non-linear hydrostatics in InWave: the hydrostatic force is updated at each time-step according to the instantaneous position of the mesh and the free surface. This cannot be turned off for multi-

body systems in InWave: linear hydrostatics is only available for single body systems. Hence, to minimize non-linear effects, very small amplitude waves were used to create the RAOs.

Each code has created the B-OF model based on the properties summarized in Table 5.4.

Table 5.4: Properties of the B-OF.

General	
Water depth	$13\ m$
Water density	$1025\ {}^{kg}/_{m^3}$
Base	
Mass	$3 \cdot 10^6\ kg$
Thickness (x)	$2\ m$
Width (y)	$26\ m$
Height (z)	$4\ m$
CoG (wrt origin)	$[0, 0, -9]\ m$
Flap	
Mass	$150 \cdot 10^3\ kg$
Thickness (x)	$2.0\ m$
Width (y)	$26.0\ m$
Height (z)	$9.0\ m$
CoG (wrt origin)	$[0, 0, -2.4]\ m$
I_{yy} (about CoG)	$2.24 \cdot 10^6\ kg \cdot m^2$

5.2.2 Development of the InWave-HOTINT Model

Hydrodynamic coefficients were computed for 300 wave frequencies over a range of $0.02 \leq \omega \leq 6\ (rad/s)$, which should mitigate any issues arising from the coarseness of the mesh. The length of the impulse response functions, t_{RIRF} is $30.0s$.

The submerged volume of the flap was computed by hand ($\forall = 2 \cdot 26 \cdot 9 = 468m^3$), which was used to determine the buoyancy vector of the flap:

$$\vec{f_b} = 1025 \cdot 9.81 \cdot 468 \tag{5.8}$$

$$\vec{f_b} = 4705857N \tag{5.9}$$

MeshMagick was used to compute the flap's hydrostatic stiffness matrix. Because the base if fixed to the ground, no hydrostatic forces are applied to this body, resulting in the following $\mathbf{K}_H$ matrix for the system:

$$\mathbf{K}_H = \begin{bmatrix} 0_{6\times6} & & & & & & 0_{6\times6} & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\ & & 0 & 0 & 0 & 0 & 0 & 0 & & & & \\ 0_{6\times6} & & 0 & 0 & 522873 & 0 & 0 & 0 & & & & \\ & & 0 & 0 & 0 & 19522638 & 0 & 0 & & & & \\ & & 0 & 0 & 0 & 0 & -9708009 & 0 & & & & \\ & & 0 & 0 & 0 & 0 & 0 & 0 & & & & \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ {}^{N}/_{m} \\ \text{and} \\ {}^{N \cdot m}/_{rad} \\ \\ \\ \\ \\ \\ \end{matrix} \tag{5.10}$$

The HOTINT model uses 2 `Rigid3DKardan` elements to model the bodies, the base is

fixed to the seabed using a `RigidJoint` constraint. A `RevoluteJoint` constraint has been used to model the kinematic constraint between the bodies, but to model the PTO, the damping matrix method has been used instead of a HOTINT spring-damper element. This is generally not recommended, as the matrix method applied a damping force to a particular degree of freedom in the global frame (whereas it is actually desired to applied a damping force to the relative degree of freedom between the bodies). However, in this case the base is fixed, and the relative pitch between bodies is equivalent to the flap's pitch in the global frame. By using the matrix method in this instance, it was hoped that comparisons with InWave would be more straightforward.

5.2.3 Decay Tests

For the free decay test, the B-OF's flap was given a $+10\ degree$ ($0.174533\ rads$) initial displacement. There is very good agreement between InWave and InWave-HOTINT in these results (Figure 5.11). A very small difference in natural period most likely occurs due to InWave-HOTINT using a linear hydrostatic solver and InWave using a non-linear hydrostatic solver[4].

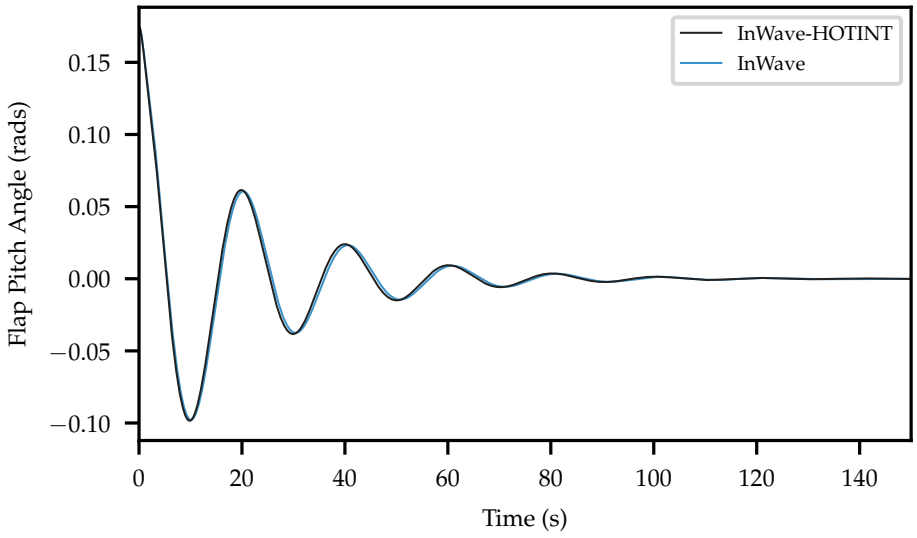


Figure 5.11: B-OF decay test: 10 degree displacement.

[4]At the time of writing, only a linear hydrostatic solver has been implemented in InWave-HOTINT, and InWave's linear hydrostatic solver is only available for 1-body systems.

5.2.4 RAOs

To overcome the obstacle of different hydrostatic solvers, very small amplitude ($A = 0.001m$) regular waves were tested, so that non-linear hydrostatic effects were negligible. Regular waves were tested over a range of $T_P = 4$ to $21s$, at $2s$ intervals. There is very good agreement between InWave and InWave-HOTINT - the time series produced at the resonant period ($20s$) are shown in Figures 5.12 and 5.13). The NumWEC results have been extracted from published data using plot digitizer (which will have introduced a small amount of error). There is a small discrepancy between NumWEC and InWave/InWave-HOTINT however at the resonant period which is discussed in the following chapter.

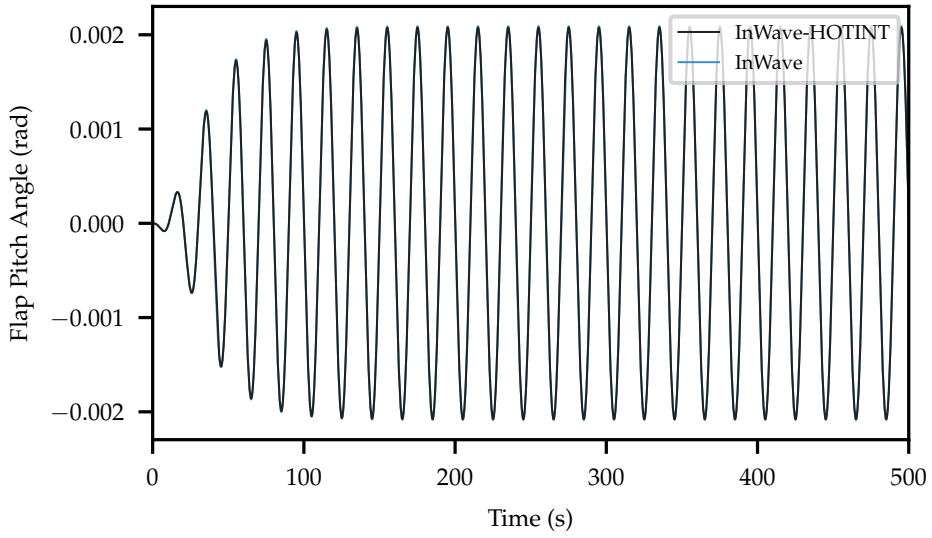


Figure 5.12: B-OF regular wave time series comparison ($A = 0.001m$, $T_p = 20s$).

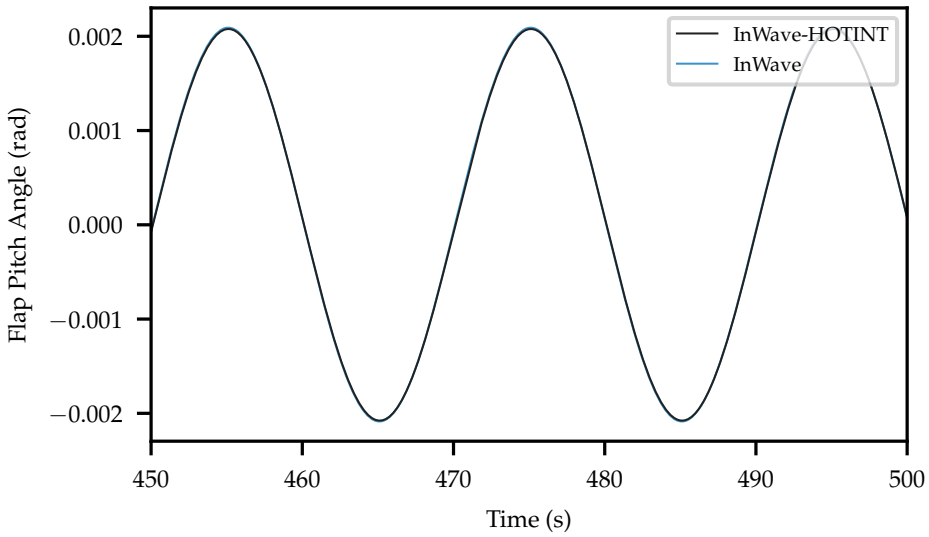


Figure 5.13: B-OF regular wave time series comparison ($A = 0.001m$, $T_p = 20s$, 450-500s region).

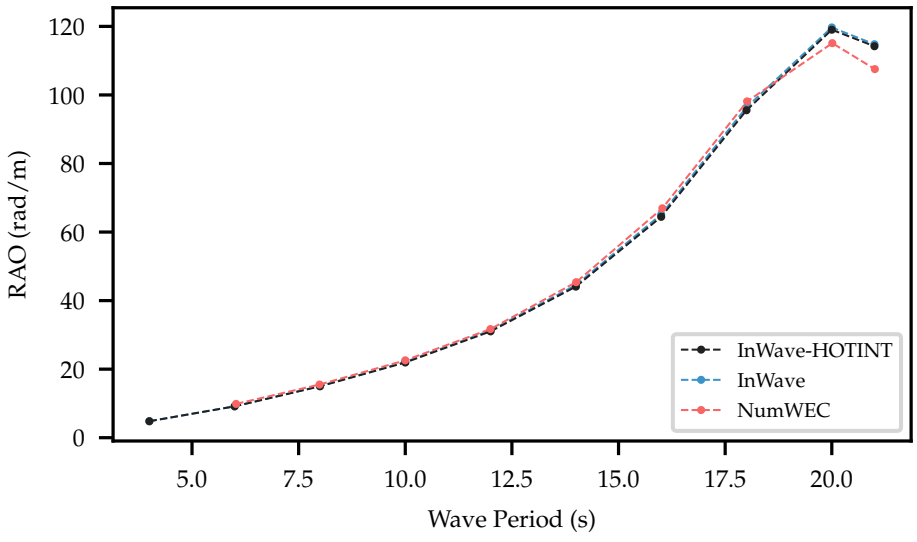


Figure 5.14: B-OF RAO comparison.

5.3 Case-Study 3: F-3OF

5.3.1 Background

The F-3OF ("floating 3-body oscillating flap") test was also originally developed by Babarit, Hals, et al. (2011) as part of the NumWEC project and is based on the Langlee WEC (Figure 5.15). The model features 3 floating bodies: the base can move in 6 DoFs, and the two flaps are attached to the base via hinge joints.

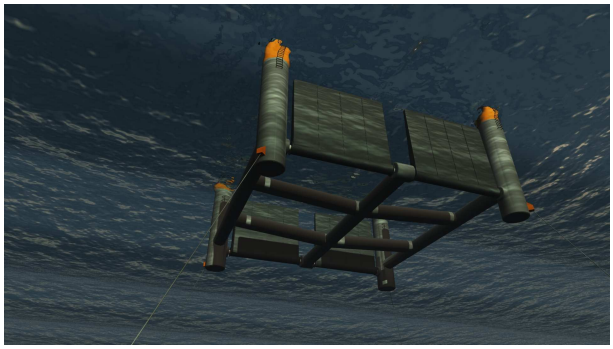


Figure 5.15: Langlee WEC (Babarit, Hals, et al. 2011).

The F-3OF model was selected WEC^3 code comparison project Combourieu, Lawson, et al. (2015) to compare time domain results (decay tests) between Inwave, ProteusDS, WaveDyn and WEC-Sim. More information about these codes has been included in Section 2.4.2. The F-3OF was chosen for the WEC^3 project specifically because of its unusual topology, with significant hydrodynamic interactions between the bodies determining the characteristics of the system. Because the hydrodynamic interactions between bodies are critical, the F-3OF is more challenging to model compared to the other verification case studies presented in this chapter. However, the original data from the WEC^3 project has been made available and includes some data that were not published but reveal some important details. Furthermore, access to InWave has been extremely helpful for modelling and verifying the F-3OF, as it has enabled the direct comparison of specific components (i.e. $\vec{f}_{rad}(t)$ for a given DoF).

The original mesh developed by Babarit, Hals, et al. (2011) has been used by all of the WEC^3 participants, and is shown in Figure 5.16. The mesh is relatively coarse and does not feature any refinement close to key features and edges. However, as each participant is using the same mesh, and the mesh is only used for code-to-code comparison

(rather than actually analyzing the WEC), any potential errors arising from the mesh are non-critical and should be present across the results from all the participants. In other words, the fact that everybody is using the same mesh is the important point, regardless of whether or not the mesh is sub-optimal.

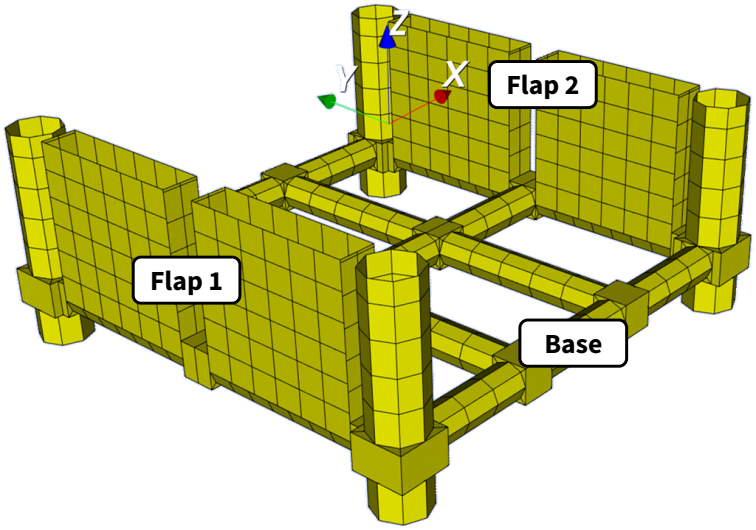


Figure 5.16: Mesh of the submerged surfaces of each body in the F-3OF system (Base: 976 panels, Flaps: 248 panels each).

As the WEC[3] project only focused on comparing decay test data, it has only been possible to compare regular wave data with InWave and NumWEC.

The F-3OF system properties are shown in Table 5.5, where each body's CoG is defined in the global frame (origin at $[0, 0, 0]\ m$), and the hinge positions and inertias are defined in the local frame (origin at the body's CoG).

5.3.2 Development of the InWave-HOTINT Model

Hydrodynamic coefficients were computed for 300 wave frequencies between $0 \leq \omega \leq 6\ (rad/s)$. The length of the impulse response functions, t_{RIRF} was originally chosen as $200s$ to match the InWave model. Because the meshes feature some openings (at the hinges), it is not possible to compute their submerged volumes. Instead, the buoyancy vector has been calculated using the weight of each body. For the base:

$$\vec{f}_{b,base} = 1.089825 \cdot 10^6 \cdot 9.81 \tag{5.11}$$

Table 5.5: Properties of the F-3OF.

General	
Water depth	$200\ m$
Water density	$1025\ {}^{kg}/_{m^3}$
Base	
Mass	$1.089825 \cdot 10^6\ kg$
Length (x)	$28\ m$
Width (y)	$28\ m$
Draft (z)	$12\ m$
CoG (wrt global origin)	$[0, 0, -9]\ m$
I_{yy} (about CoG)	$76.3 \cdot 10^6\ kg \cdot m^2$
Flap 1 hinge location (wrt local frame's origin @ CoG)	$[-12.5, 0, 0]\ m$
Flap 2 hinge location (wrt local frame's origin @ CoG)	$[12.5, 0, 0]\ m$
Mooring Stiffness (in the x direction)	$100 \cdot 10^3\ {}^{N}/_{m}$
Flaps	
Mass	$179.25 \cdot 10^3\ kg$
Thickness (x)	$2.0\ m$
Width (y)	$21.0\ m$
Draft (z)	$10.0\ m$
I_{yy} (about CoG)	$1.3 \cdot 10^6\ kg \cdot m^2$
Hinge location (wrt local frame's origin @ CoG)	$[0, 0, -3.5]\ m$
PTO damping	$40 \cdot 10^6\ {}^{Ns}/_{rad}$
CoG of Flap 1 (wrt global origin)	$[-12.5, 0, -5.5]\ m$
CoG of Flap 2 (wrt global origin)	$[12.5, 0, -5.5]\ m$

$$\vec{f}_{b,base} = 10691183.25N \tag{5.12}$$

For each flap:

$$\vec{f}_{b,flap} = 179.25 \cdot 10^3 \cdot 9.81 \tag{5.13}$$

$$\vec{f}_{b,flap} = 1758442.5N \tag{5.14}$$

The hydrostatic stiffness matrix of each body has been computed with MeshMagick and assembled as a block diagonal matrix, $\mathbf{K}_H$, corresponding to the whole system.

The HOTINT model utilizes `Rigid3DKardan` elements to represent each rigid body in the system. The joints are modelled using `RevoluteJoint` elements for the kinematic constraint, and `RotatorySpringDamperActuator` elements to model the PTO damping at the hinges. To model the mooring, the matrix method has been used (instead of using HOTINT's rope elements). This was chosen in order to reduce potential sources of discrepancy between the InWave and InWave-HOTINT results, but both methods should produce the same results.

Some of the decay tests have required certain bodies in the system to be 'locked'. To achieve this with InWave-HOTINT, the `RigidJoint` element has been used (either to lock the flaps, or to fix the base in space, depending on the requirements of the test).

$$\mathbf{K}_H = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & \\
0 & 0 & 0.256 & 0 & 0 & 0 & & 0_{6\times6} & & & 0_{6\times6} & \\
0 & 0 & 0 & 49.3 & 0 & 0 & & & & & & \\
0 & 0 & 0 & 0 & 49.3 & 0 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & \\
 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0_{6\times6} & & & & & 0 & 0 & 0.382 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & 24.7 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & 0 & 9.34 & 0 \\
 & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & \\
 & 0_{6\times6} & & & & & & 0_{6\times6} & & & 0_{6\times6} & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & &
\end{bmatrix} \cdot 10^6 \begin{matrix} N/m \\ \text{and} \\ N \cdot m/rad \end{matrix} \tag{5.15}$$

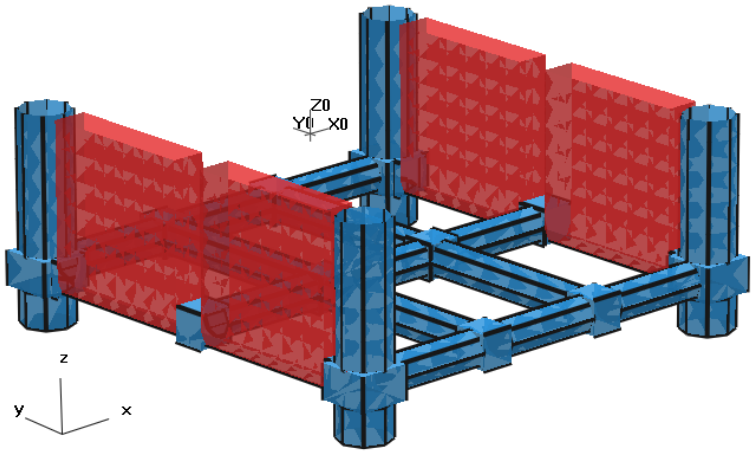


Figure 5.17: Visualization of the F-3OF model in the HOTINT GUI.

5.3.2.1 Description of the tests

The verification tests performed for the F-3OF model are shown in Table 5.6

Table 5.6: List of verification tests for the F-3OF model.

Type of test	Description	Case	Initial Conditions	Configuration
Decay Tests	Surge	DT1	System is given +5 m displacement in surge	Flaps are locked
	Pitch	DT2	Whole system is rotated by +10 deg in pitch	Flaps are locked
	Flap 1	DT3	Flap 1 is rotated by +10 deg in pitch	Base is locked
Regular Waves	Small waves ($A = 0.01m$) for computing RAOs	REG1-20	Equilibrium position	PTO ON

5.3.3 Decay Tests

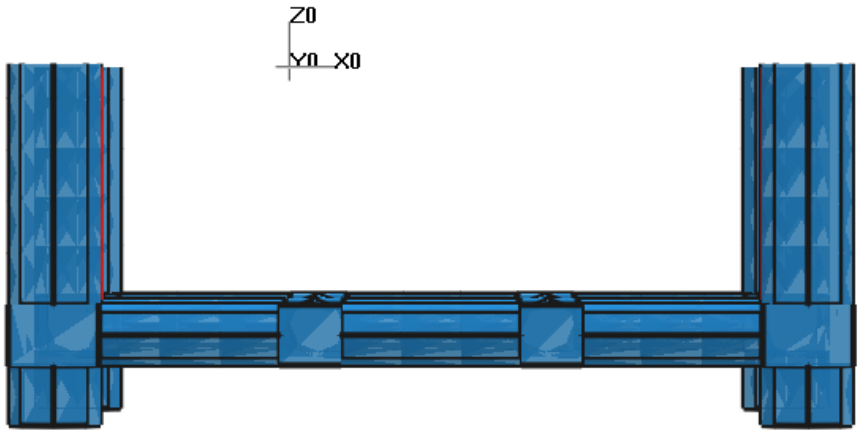


Figure 5.18: Initial position of the F-3OF for decay test 1 (DT1).

Decay Test 1 (DT1): Surge, 5m From the surge decay test (Figure 5.18), we can observe some problematic results: InWave, InWave-HOTINT and to a lesser extent, ProteusDS, display instability from $t = 200s$ (the same length as the RIRF used in InWave and InWave-HOTINT). Up to 200s there is a slight decay in the results, before the system's oscillations become unstable. This transition from convergence to divergence at $t = t_{RIRF}$ suggests a possible RIRF truncation error, and indeed, the surge RIRF of each body (Figure 5.20) reveals that the RIRF length is too long, causing $\vec{f}_{rad}$ to become unstable (Figure 5.21).

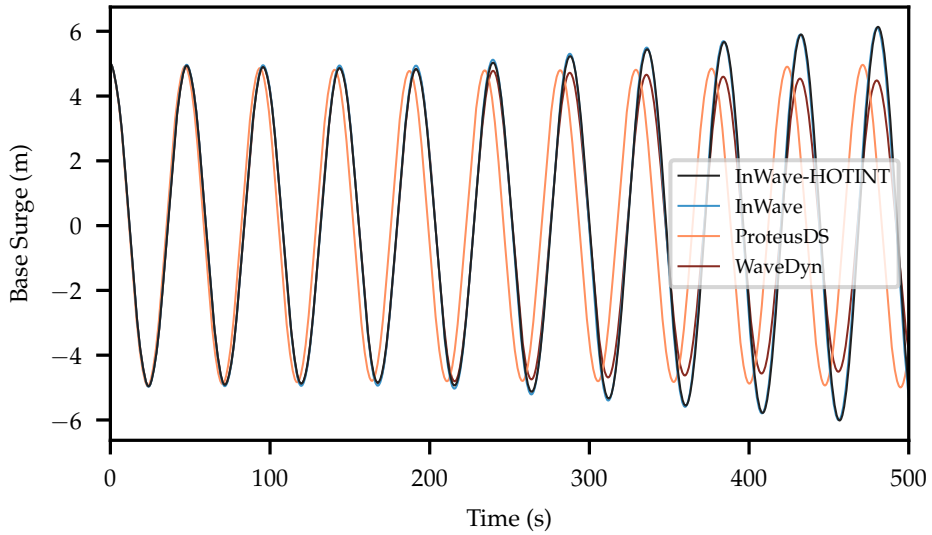


Figure 5.19: F-3OF decay test 1 (DT1) results - comparison of the base's position in surge (using $t_{RIRF} = 200s$ for InWave-HOTINT and InWave).

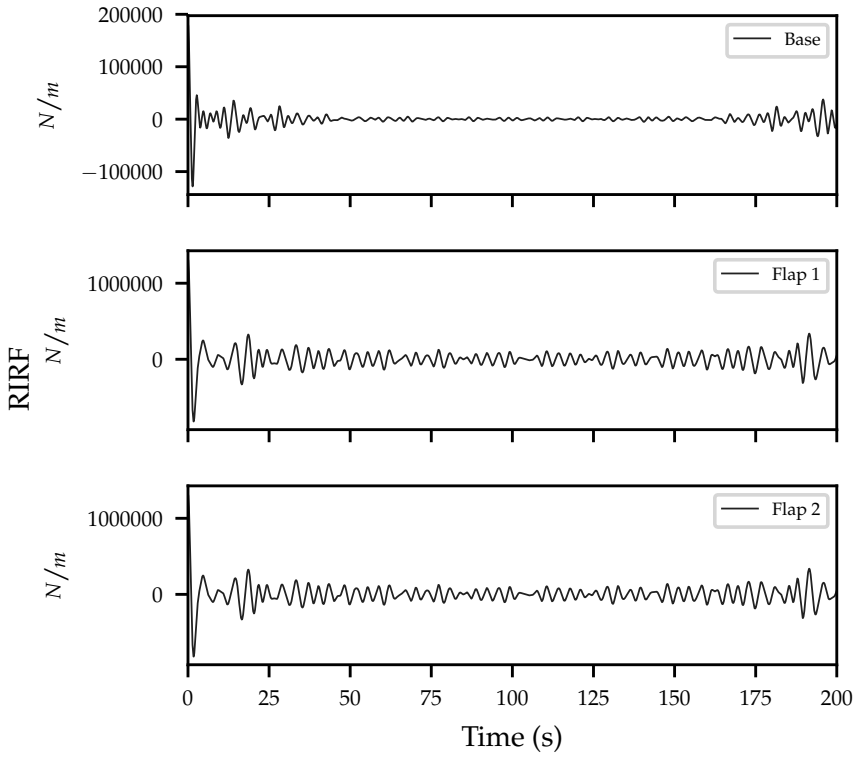


Figure 5.20: F-3OF RIRF results for each body in surge-surge (computed with NEMOH).

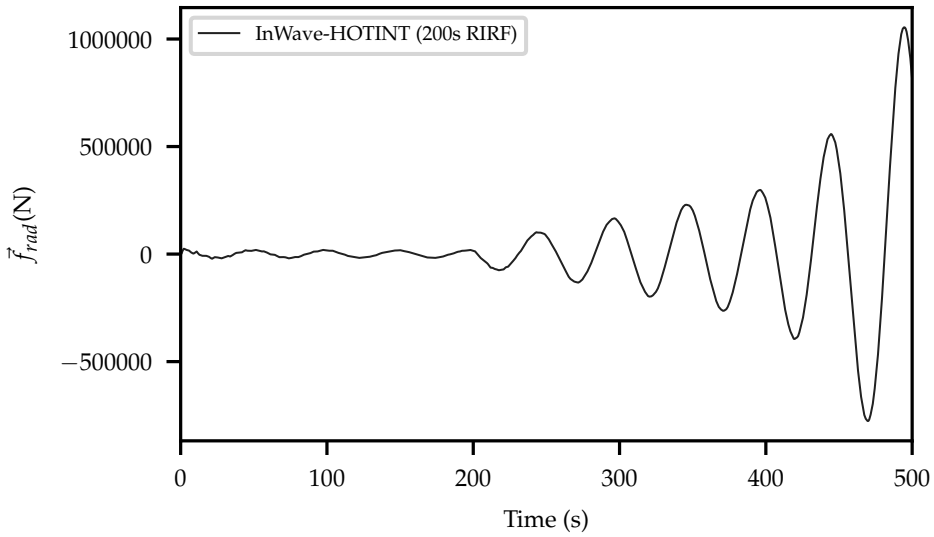


Figure 5.21: F-3OF $\vec{f}_{rad}$ instability observed when using $t_{RIRF} = 200s$.

Based on Figure 5.20, the selection of a shorter RIRF length of $125s$ can be justified to produce stable results (Figures 5.22 and 5.23).

Although the shorter RIRF length fixes the stability issue in InWave-HOTINT, there are still some significant discrepancies with the other codes, both in decay rate and to a lesser extent natural period (with ProteusDS having a slightly shorter natural period, and WaveDyn & InWave both having slightly longer natural periods compared to InWave-HOTINT). Without knowing what values were selected by the other participants for the $d\omega$ and t_{RIRF} parameters, it is difficult to determine if this is also a source of error in the other code's results. However, comparing the base's heave results for the same decay test (Figure 5.24) reveals another possible source of error. As observed in some of the Sphere case-study results, there are small offsets present in each code's results for the base equilibrium position: ProteusDS has a positive offset, and WaveDyn & InWave both have a negative offset. Athough small, it is possible that this error would have an effect on the system's natural periods. It is also possible that this error could affect the radiation damping force; as the bodies are oscillating about different positions from those defined in the hydrodynamic preprocessor.

Decay Test 2 (DT2) Decay test 2 sees the F3OF's flaps 'locked' and the whole device rotated by $+10$ degrees in pitch (Figure 5.25).

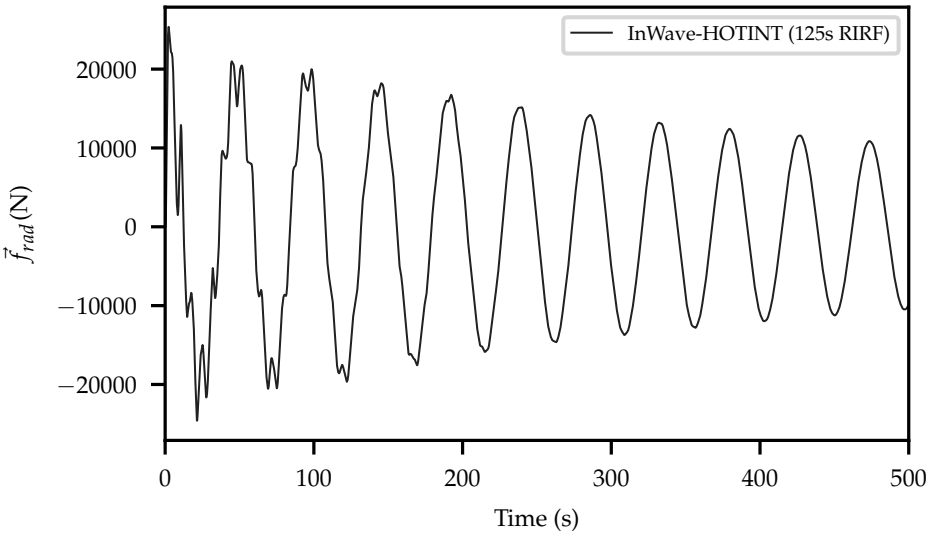


Figure 5.22: F-3OF $\vec{f}_{rad}$ decay observed when using $t_{RIRF} = 125s$.

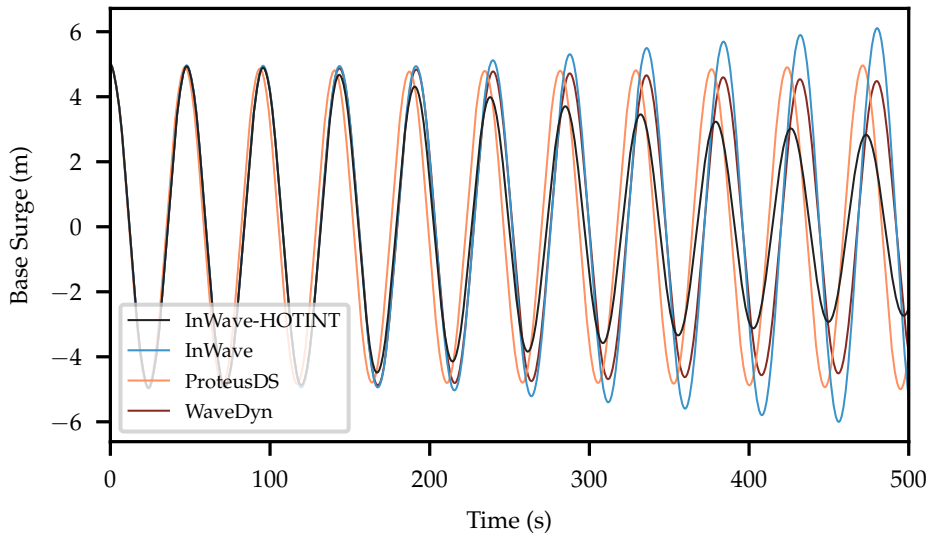


Figure 5.23: F-3OF decay test 1 (DT1) results - comparison of the base's position in surge (using $t_{RIRF} = 125s$ for InWave-HOTINT).

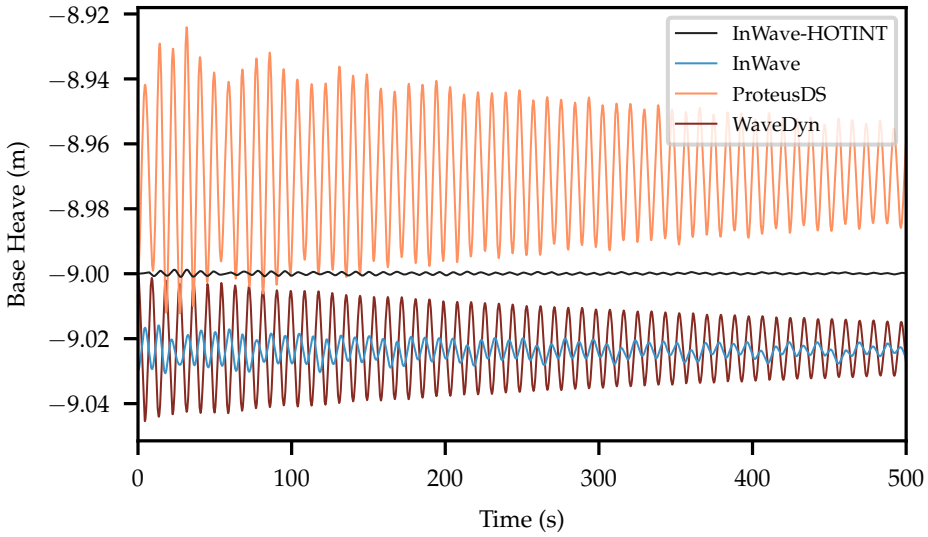


Figure 5.24: F-3OF decay test 1 (DT1) results - comparison of the base position in heave.

Decay test 2 shows the best agreement between the WEC[3] participants, both in natural period and rate of decay (Figure 5.26). Figure 5.27 again reveals small offsets in the system's equilibrium positions in heave, which could potentially explain the small discrepancies observed in Figure 5.26.

InWave is utilizing a non-linear hydrostatic solver in these results, which could also explain the discrepancies. However, the decay test was from an initial displacement of $10°$; the same as the B-OF decay test, which was too small to produce non-linear effects.

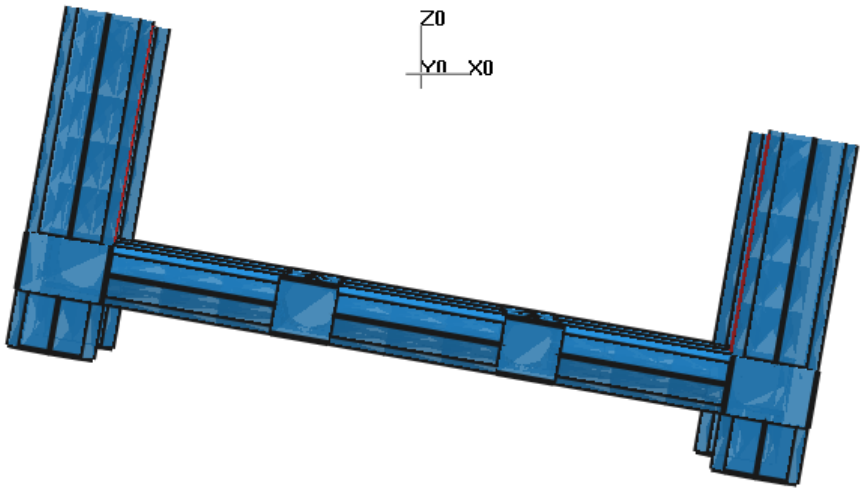


Figure 5.25: Initial position of the F-3OF for decay test 2 (DT2).

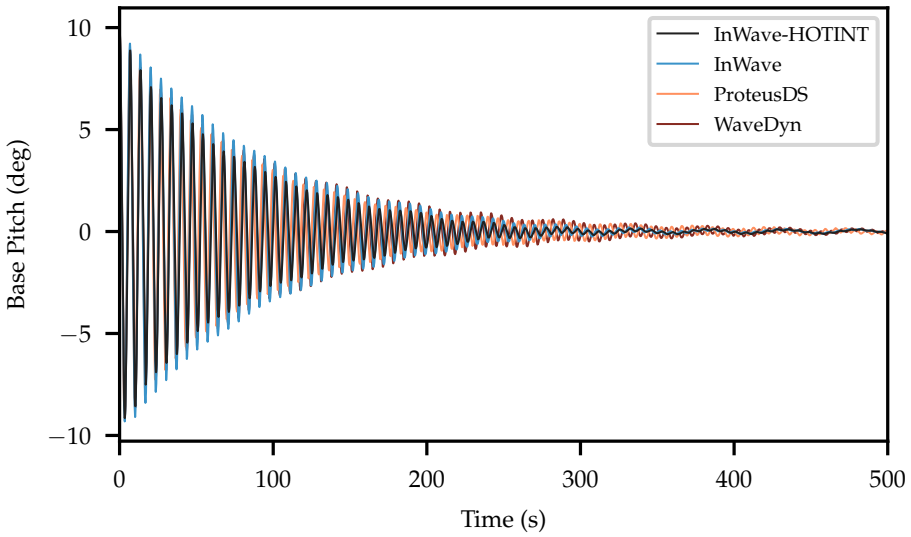


Figure 5.26: F-3OF decay test 2 (DT2) results - comparison of the base's position in pitch.

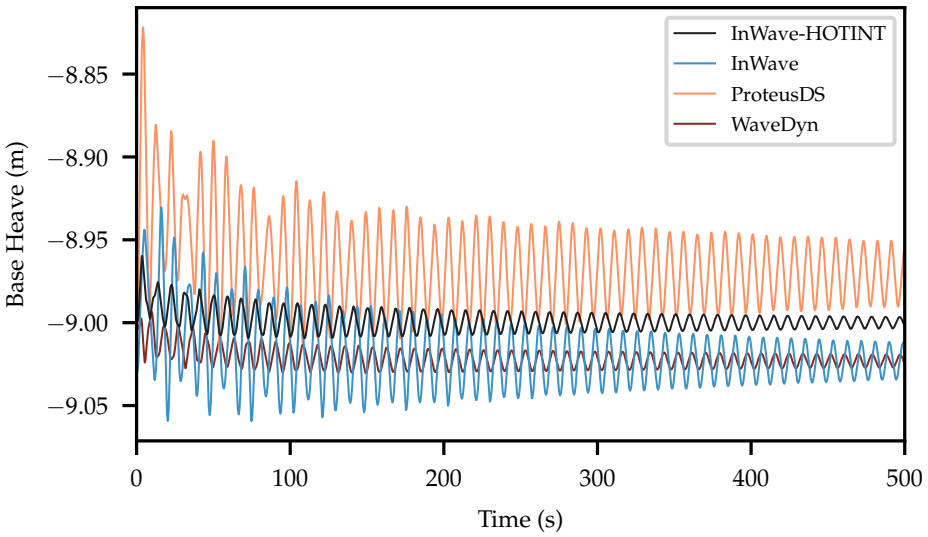


Figure 5.27: F-3OF decay test 2 (DT2) results - comparison of the base's position in heave.

Decay Test 3 (DT3) Decay test 3 tests the hydrodynamic interactions between the F3OF's flaps. The base is fixed and flap 1 is given an initial displacement of $+10$ degrees (Figure 5.28). As flap 1 decays it radiates waves, which excite flap 2 and cause it to start oscillating at the same frequency (Figures 5.29 and 5.30).

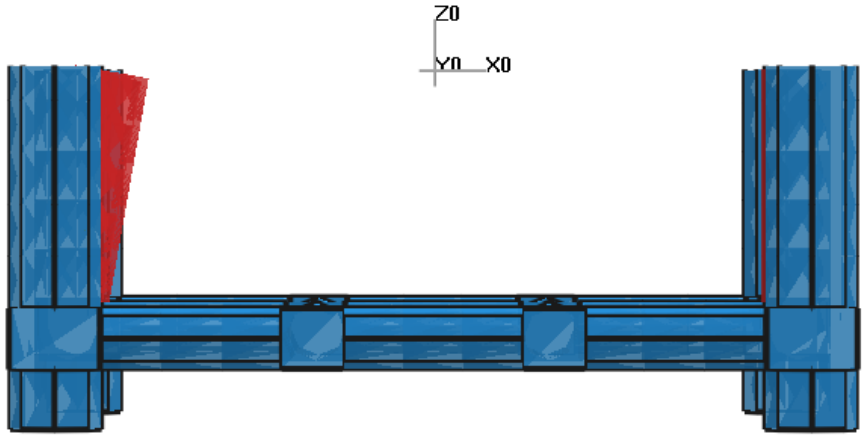


Figure 5.28: Initial position of the F-3OF for decay test 3 (DT3).

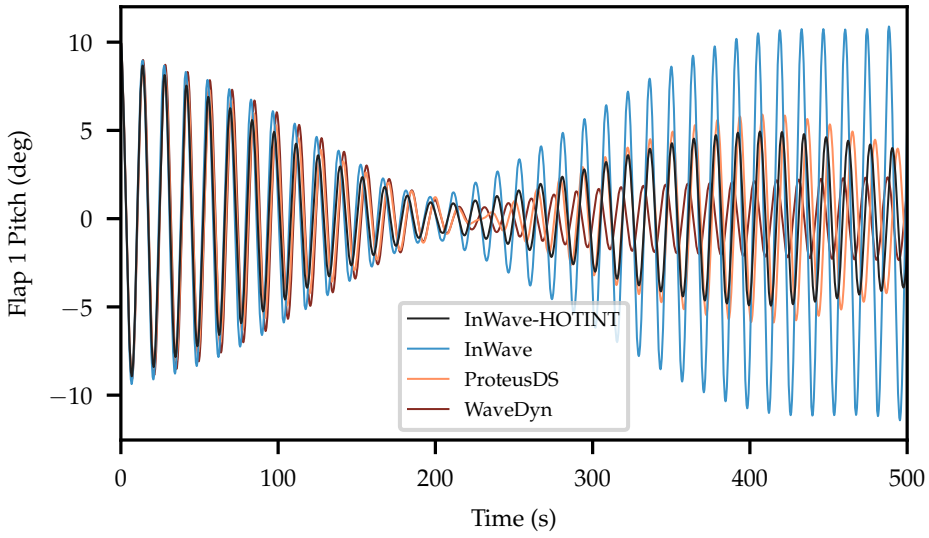


Figure 5.29: F-3OF decay test 3 (DT3) results - comparison of flap 1's position in pitch.

Another characteristic of this test which makes it interesting is that the base was locked, and therefore the equilibrium offset errors that were present in decay tests 1 and 2 do not manifest in these results (although there may still be discrepancies in $\mathbf{K}_H$ and $\vec{f}_b$ between the different codes - it is not possible to say as this data has not been shared). Close-ups of the first 80s of the time series (Figures 5.31) show quite good agreement

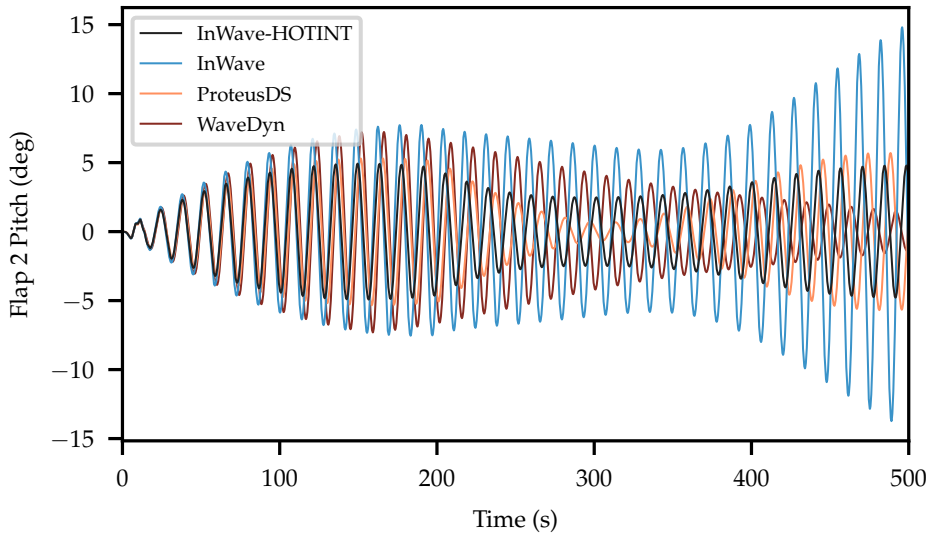


Figure 5.30: F-3OF decay test 3 (DT3) results - comparison of flap 2's position in pitch.

between the different codes. However, after this initial phase, the results differ significantly. Most noticeable are the InWave results, which eventually became unstable. The reason for this has already been determined from decay test 1: a poorly truncated RIRF does not decay properly within the defined RIRF length (t_{RIRF}), and causes $\vec{f}_{rad}$ to become unstable around $t = t_{RIRF}$. It is possible that the radiation force is also the source of discrepancies with the other codes: it was noted in decay test 1 that after correcting t_{RIRF} in InWave-HOTINT, the results were much more reasonable, with a smooth decay, and yet did not match the other codes' results. However, without access to these variables from the other codes it is not possible to be certain about the source of the discrepancies. Indeed, it is not said whether the other codes computed $\vec{f}_{rad}$ by direct integration of the convolution integral or by another method (e.g. state-space approximation).

5.3.4 RAOs

Small amplitude waves ($A = 0.01m$) were tested for computing the RAOs, in order to reduce non-linear effects (as the InWave hydrostatic solver is non-linear).

Despite the differences in decay test results, there was very good agreement between

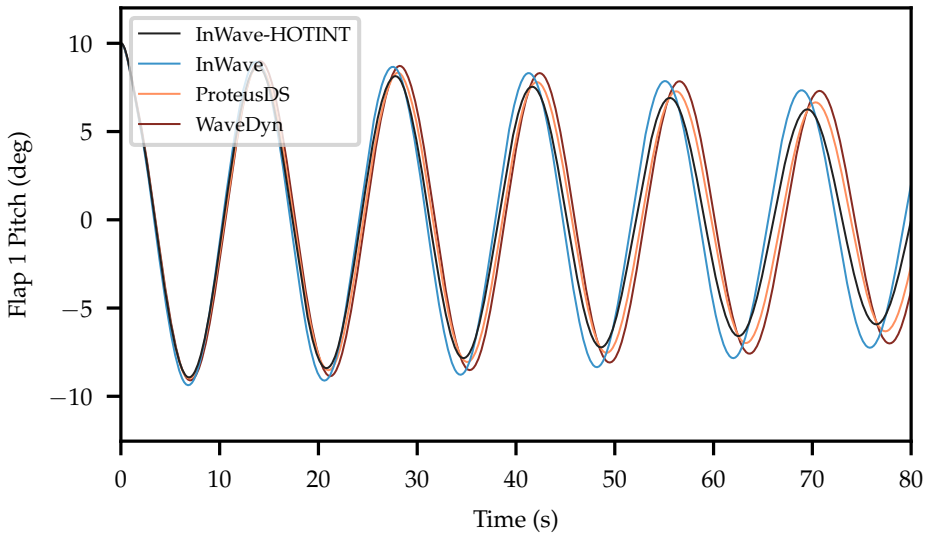


Figure 5.31: F-3OF decay test 3 (DT3) results - comparison of flap 1's position in pitch (0-80s region).

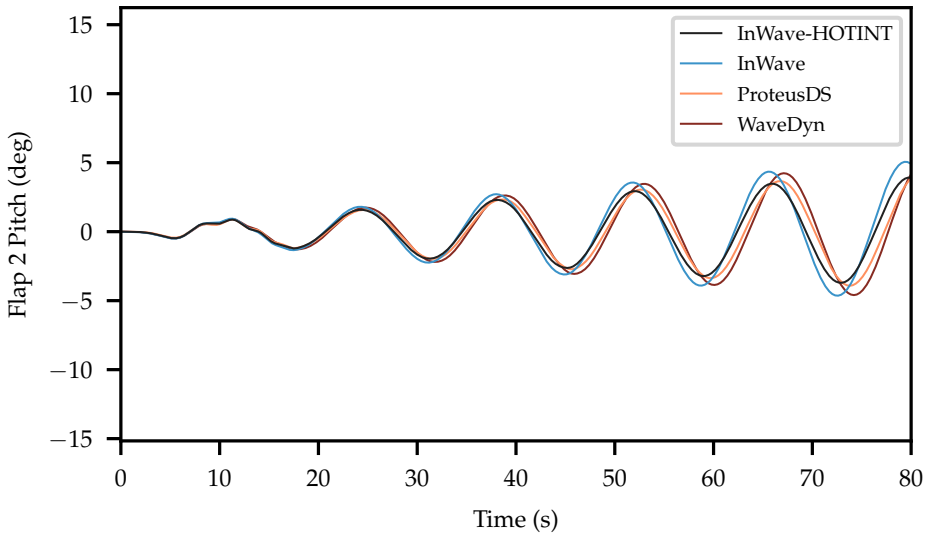


Figure 5.32: F-3OF decay test 3 (DT3) results - comparison of flap 2's position in pitch (0-80s).

InWave and InWave-HOTINT in the RAO results, suggesting that errors arising in still water conditions can potentially be masked in the presence of waves. However, there are significant discrepancies with the NumWEC results. The NumWEC time-domain results include Morison's equation, which explain the more responses in surge and heave. Although Morison's equation has been implemented in InWave, it has not been utilized in these results.

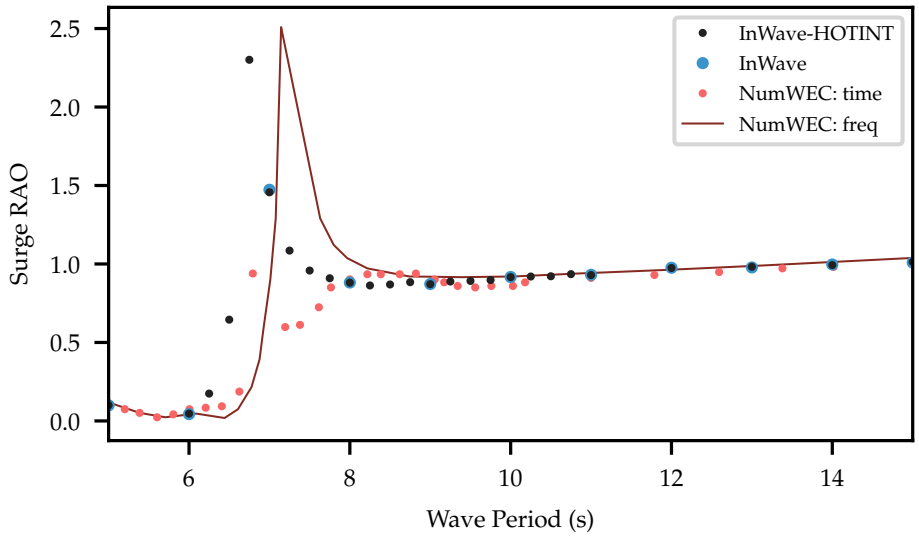


Figure 5.33: F3OF RAO results - comparison of the base's position in surge.

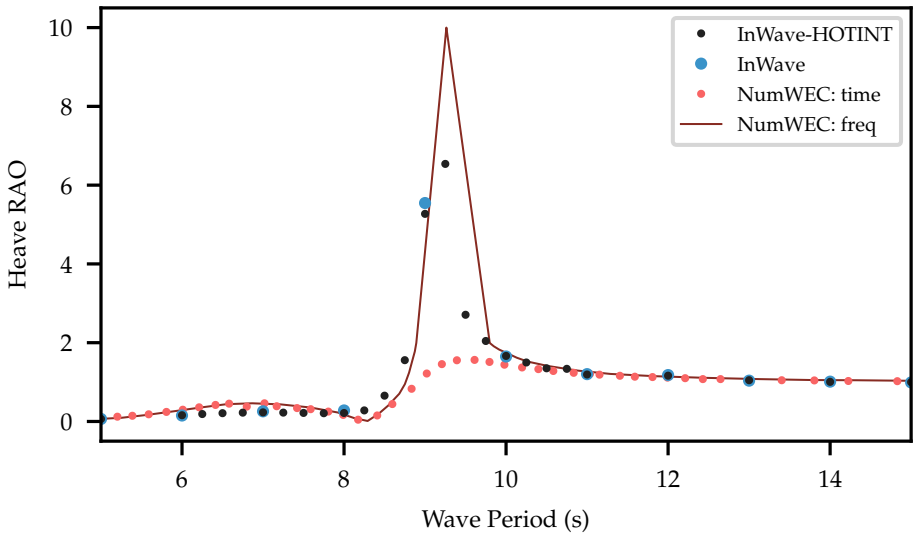


Figure 5.34: F3OF RAO results - comparison of the base's position in heave.

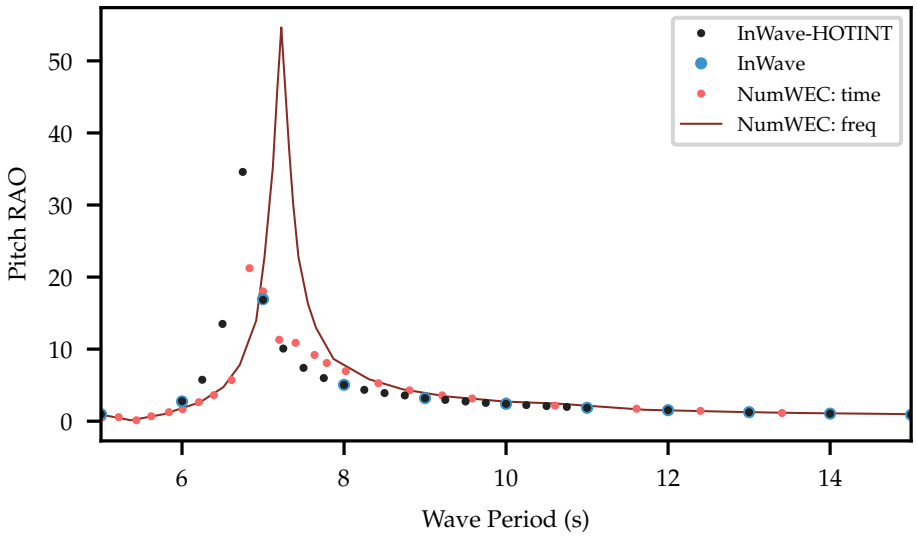


Figure 5.35: F3OF RAO results - comparison of the base's position in pitch.

5.4 InWave-HOTINT Capabilities & Modelling Process

This section details the InWave-HOTINT modelling process by developing a novel WEC model from scratch. The model also demonstrates some of InWave-HOTINT's capabilities - including the ability to model closed kinematic loops, multiple degree of freedom PTO systems and net mooring systems.

5.4.1 Albatern S12: Overview

Albatern Ltd.'s WaveNET device is arguably one of the most kinematically complex WECs currently in development. The concept features modular 'Squid' units, made up 3 buoyant nodes connected to a central node by 3 horizontal link arms, via hydraulic joints which absorb power from the relative pitch and yaw motion between the bodies. The central node in this system has 6 individual PTO units connected to it: absorbing power from the relative pitch and yaw between the node and the 3 link arms. The other nodes have 2 PTOs in addition to their 2 mooring straps (Figure 5.36).

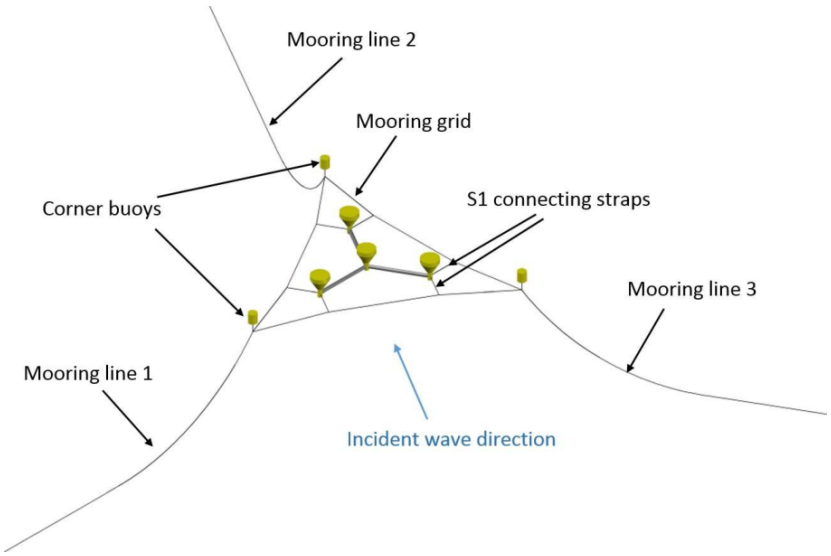


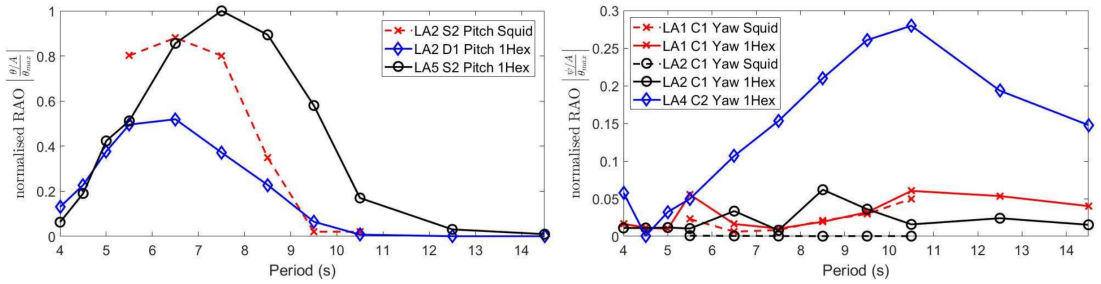
Figure 5.36: Labelled illustration of an individual Squid unit (McDonald et al. 2017).

The Squids can be connected together to create arbitrarily large arrays of interconnected floats. Connecting these rigid bodies together in an array creates closed kinematic loop mechanisms. The 'Hex' device consists of 3 Squids connected together in a hexagonal arrangement.

The mooring system is similar for both the Squid and Hex device and is a net consisting of 3 catenary mooring lines attached to a triangular grid. The grid has straps that attach to each node.

Hence, the Albatern device has several features that are challenging to capture in a numerical model - making it an ideal demonstration case for InWave-HOTINT. However, experimental data have not been accessed and the model's parameters have been estimated rather than measured directly from a physical model: therefore no experimental validation of the results is attempted in this section.

In Autumn 2016, Albatern conducted physical model testing of 1:18 scale models of the 12S Squid and Hex devices at the FloWave Ocean Energy Research Facility, Edinburgh, as part of the Wave Energy Scotland (WES) Novel WEC stage 1 (NWEC 1) development programme. Key information about the devices has been estimated from the public report submitted to WES (Albatern Ltd. 2017), and the related conference papers (Barker-Ewart 2017; McDonald et al. 2017). Furthermore, non-dimensionalized experimental data published by McDonald et al. (2017) has been used as a reference point to explore the effect of various design choices on certain degrees of freedom in the system (Figure 5.37). Hence, the same Squid and Hex RAOs compared by McDonald et al. (2017) are plotted later in this section, where it is shown that the results can be influenced by a wide range of parameters in the system.



(b) Pitch (θ) (a) Yaw (ψ)

Figure 5.37: Selected undamped articulation RAOs (normalised by maximum pitch response amplitude (θ_{max}) from the individual Squid and the Hex WaveNET array tests (McDonald et al. 2017).

In order to create a representative mesh of the S12 system, its key dimensions were estimated from the photographs in Figure 5.38. Figure 5.39 shows the estimated dimensions of the S12 Squid physical model. The waterline is estimated to be $85mm$ below the top of the node. Each node's centre of gravity is estimated to be $300mm$ be-

low the waterline. The link arms are estimated to attach at the nodes $380mm$ below the waterline (i.e. $100mm$ below the node's centre of gravity). It has also been assumed that the mooring 'straps' attach to the nodes at the same position as the link arms.

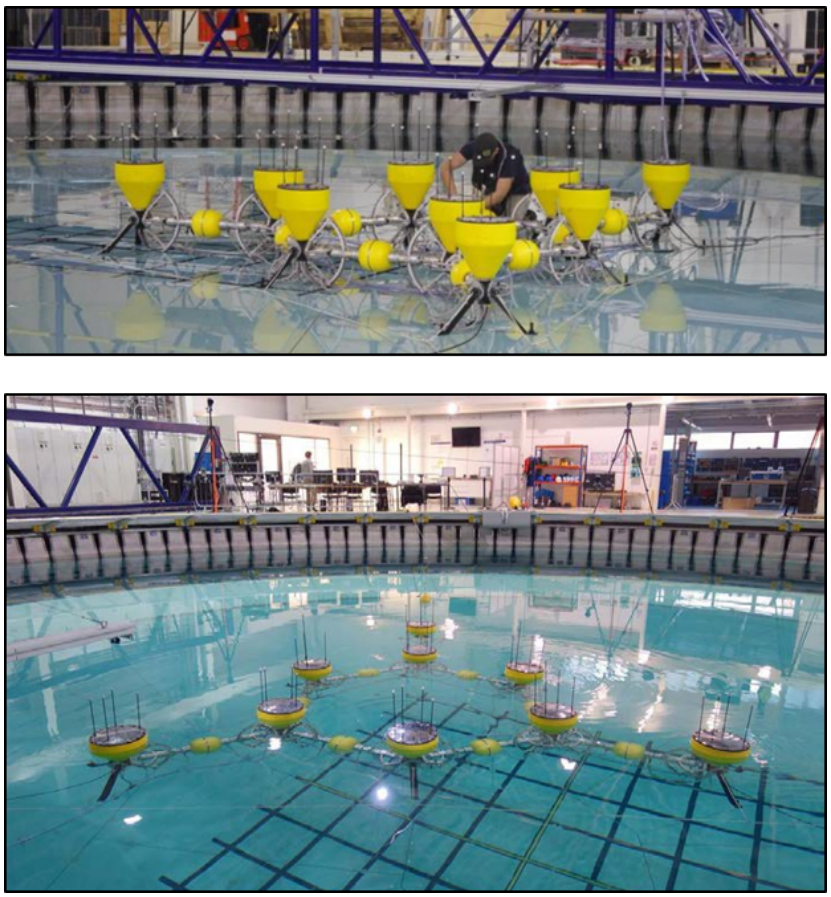


Figure 5.38: Photographs of the S12 device taken during tank testing at FloWave (Albatern Ltd. 2017).

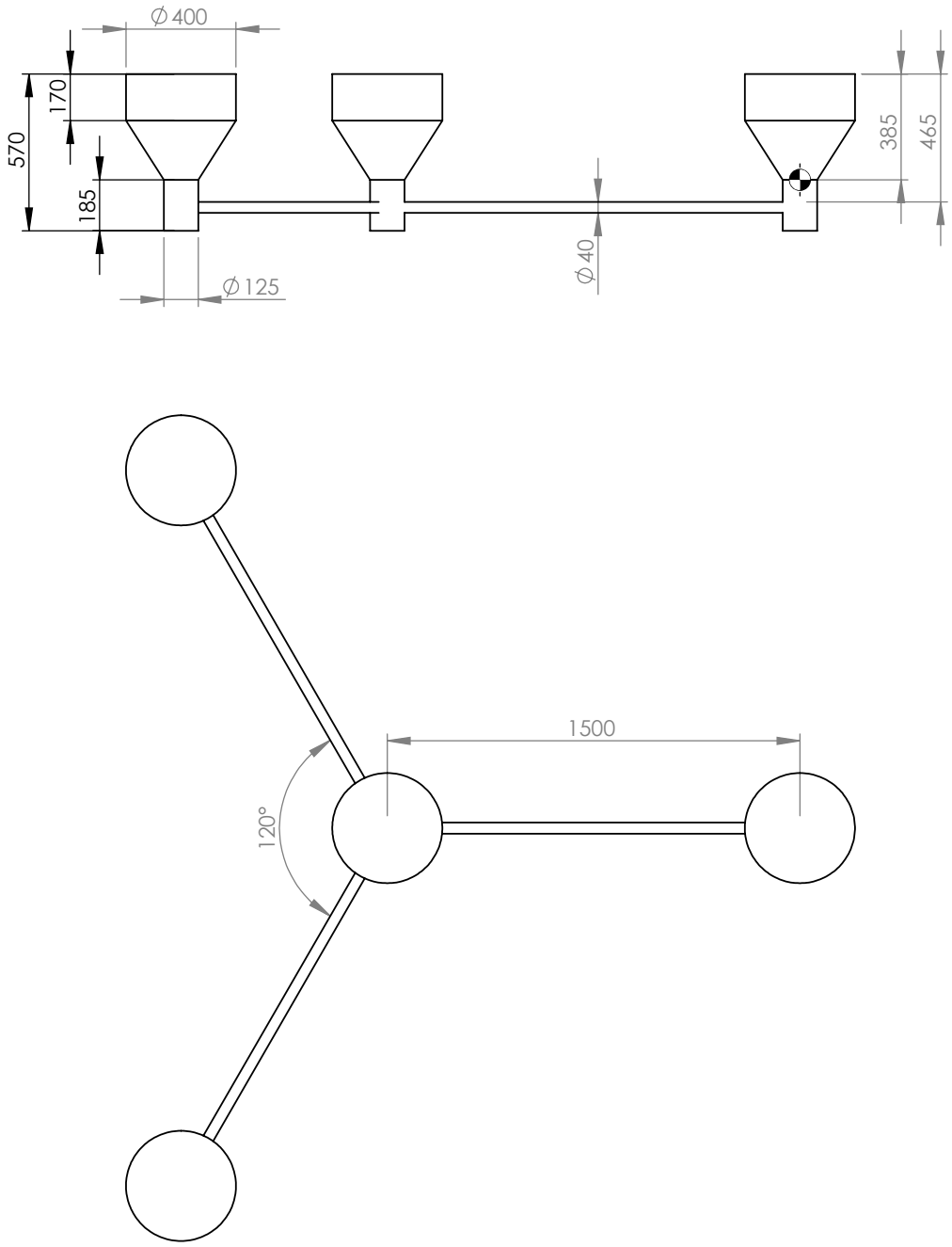


Figure 5.39: Dimensions (in mm) of the S12 Squid physical model - estimated from the photographs shown in Figure 5.38.

5.4.2 Creating the S12 Meshes

5.4.2.1 Meshing the Node

Based on the estimated node dimensions shown in Figure 5.39, a surface mesh was created by defining points along the edge profile of the submerged part of the node and revolving these points about the node's central axis at discrete intervals (Figure 5.40). Cosine spacing was used to define the points along each face in order to improve accuracy around the node's edges and close to the waterline.

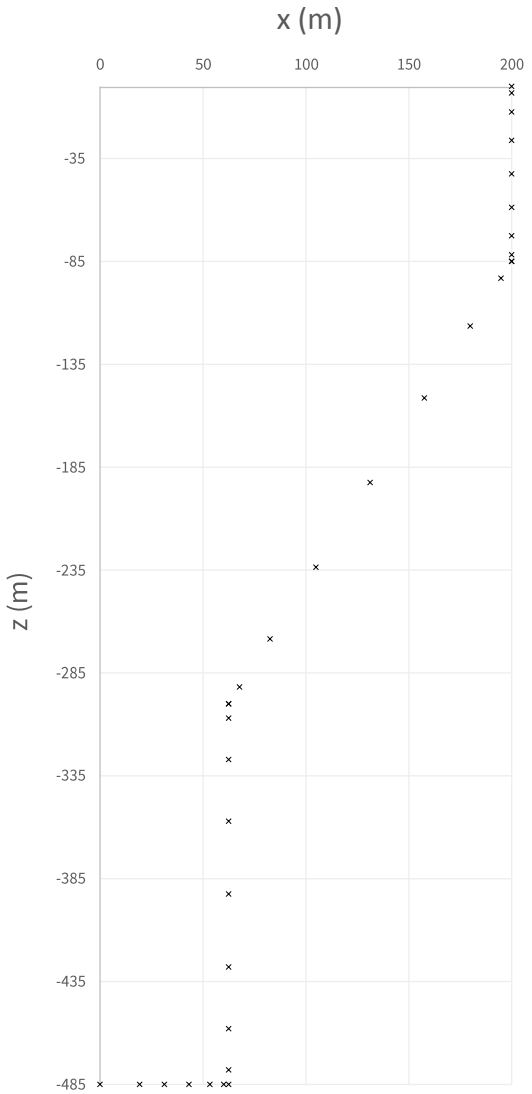


Figure 5.40: The edge profile used to create the S12 node mesh consisting of 888 panels; cosine spacing is used along each face.

The example shown in Figure 5.40 is of a medium fidelity mesh used, consisting of 888 panels. Initially, a finer, more accurate mesh consisting of 2176 panels (Figure 5.41) was used to compute the hydrodynamic coefficients of this mesh. It was determined that the results begin to asymptote from $\omega = 24 rad/s$, hence a frequency range of $0.02 < \omega \leq 24 rad/s$ was selected for further computations (Figures 5.42 and 5.43).

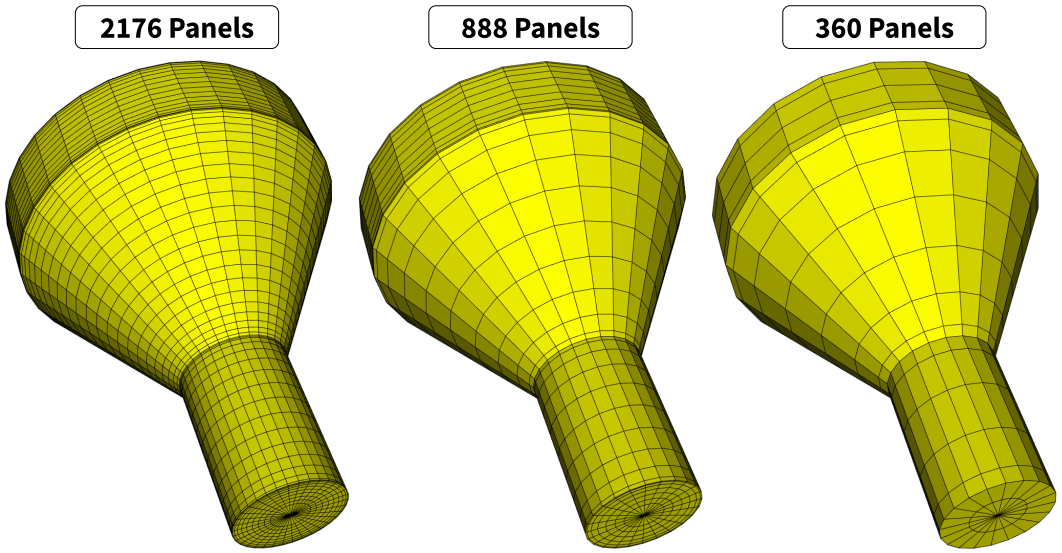


Figure 5.41: S12 Node meshes with varying number of panels.

The results obtained for each mesh are compared in Figures 5.42 and 5.43. The close agreement between each mesh supports the used of the coarser mesh (360 panels) in the modelling of the Squid and Hex arrays. The spikes observed at higher frequencies represent 'irregular frequencies': errors due to the hydrodynamic solver, NEMOH[5].

Meshmagick was used to determine the Node's inertial and hydrostatic stiffness properties (Table 5.7).

[5] Although the hydrodynamic results shown here include irregular frequencies, recent work at Innosea has focusing on developing new methods in NEMOH to overcome this problem

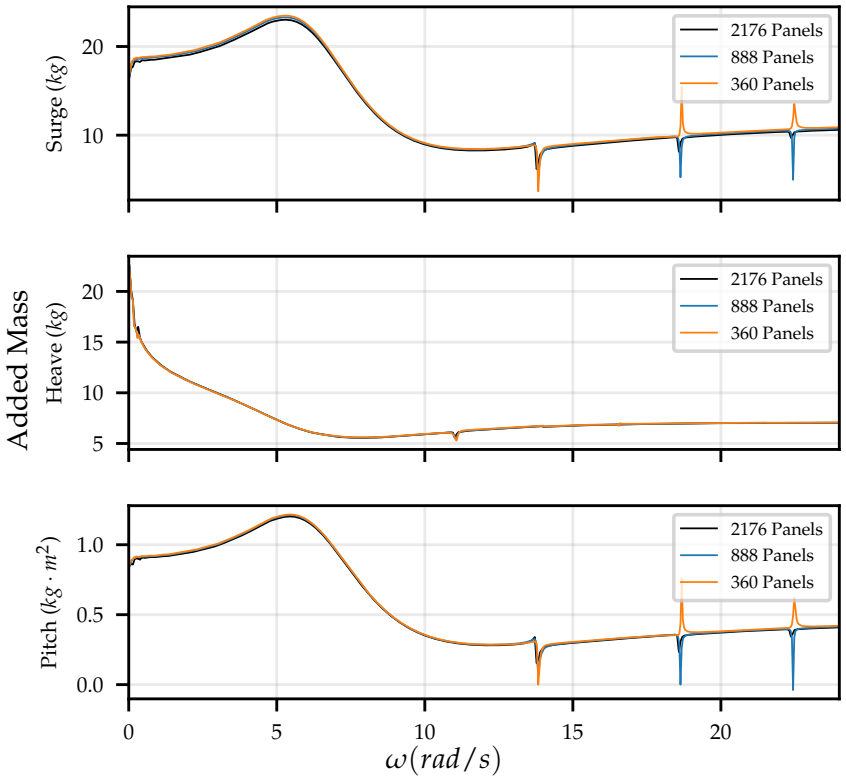


Figure 5.42: S12 Node: added mass coefficients for different meshes.

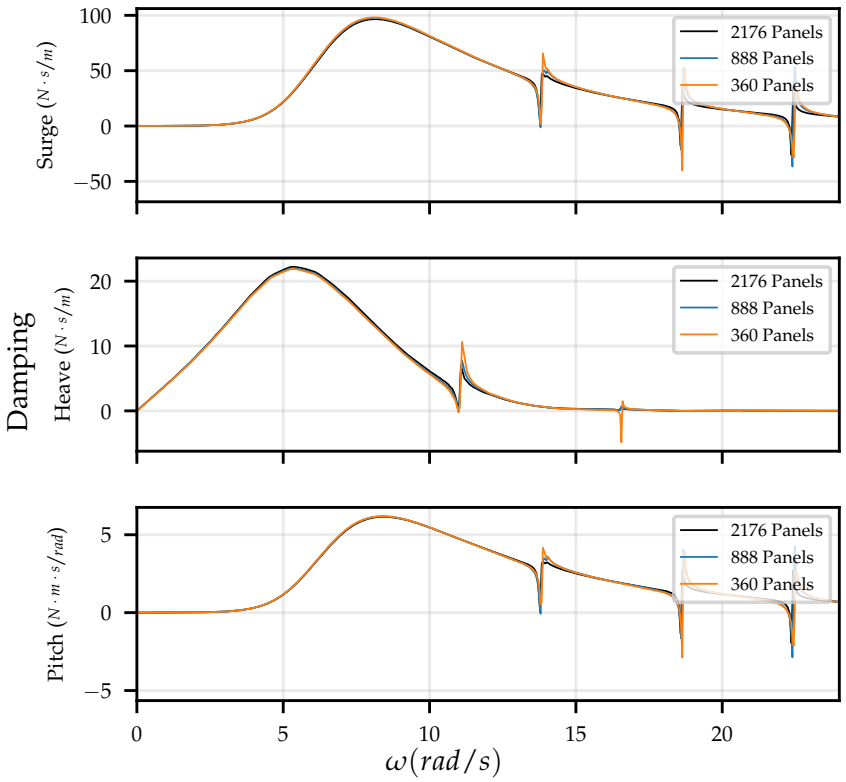


Figure 5.43: S12 Node: damping coefficients for different meshes.

Table 5.7: Properties of the S12 Node.

Water density, ρ_{water}:	1000	kg/m^3
Displacement volume, $\forbidden{V}$:	0.025	m^3
Displacement mass, m:	25	kg
Inertia matrix, $\mathbf{I}$:	$\begin{bmatrix} 474.8 & 0 & 0 \\ 0 & 474.8 & 0 \\ 0 & 0 & 362.6 \end{bmatrix}$	$kg \cdot m^2$
Hydrostatic stiffness matrix, $\mathbf{K}_H$:	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1212.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 53.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 53.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	N/m and $N \cdot m/rad$

From the computed displacement volume the node's buoyancy vector can be determined:

$$\vec{f_b} = \rho g \forbidden{V} \tag{5.16}$$

$$\vec{f_b} = 1000 \cdot 9.81 \cdot 0.025 \tag{5.17}$$

$$\vec{f_b} = 245.25N \tag{5.18}$$

5.4.2.2 Meshing the Squid and Hex

For both the Squid and Hex, it has been assumed that the nodes dominate the systems' hydrodynamic response. Hence, the link arms have not been included in the Squid and Hex hydrodynamic meshes (Figures 5.44 and 5.46).

Squid The naming convention used to label each body in the Squid system is shown in Figure 5.45, and the location of each node's centre of gravity has been calculated by trigonometry, according to a link arm length of $1.5m$ (results presented in Table 5.8).

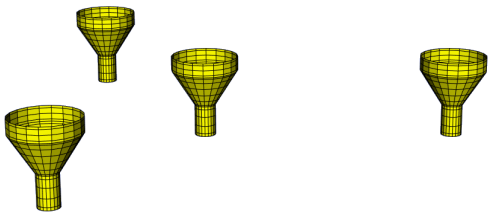


Figure 5.44: Hydrodynamic mesh of the Squid device used for hydrodynamic preprocessing.

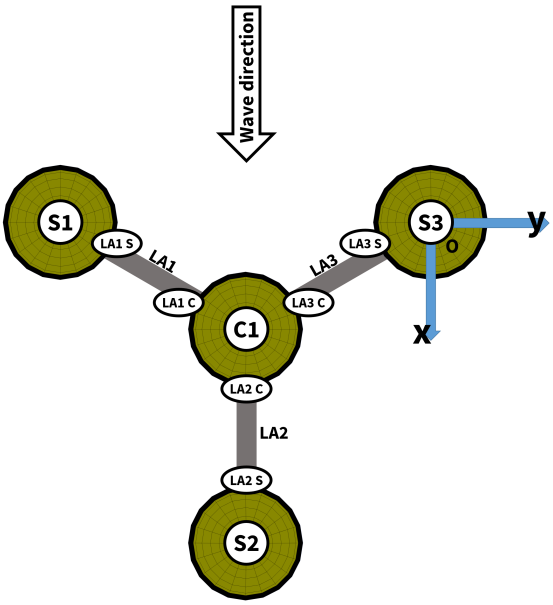


Figure 5.45: Naming convention for the Squid system (plan view). Adapted from McDonald et al. (2017).

Table 5.8: Centre of Gravity position of each node in the Squid device.

		CoG		
		x (m)	y (m)	z (m)
Node	S3	0.000	0.000	-0.300
	C1	0.750	-1.299	-0.300
	S1	0.000	-2.598	-0.300
	S2	2.250	-1.299	-0.300

Hex The naming convention used for the Hex system is shown in Figure 5.47 and the calculated position of each node's centre of gravity presented in Table 5.9.

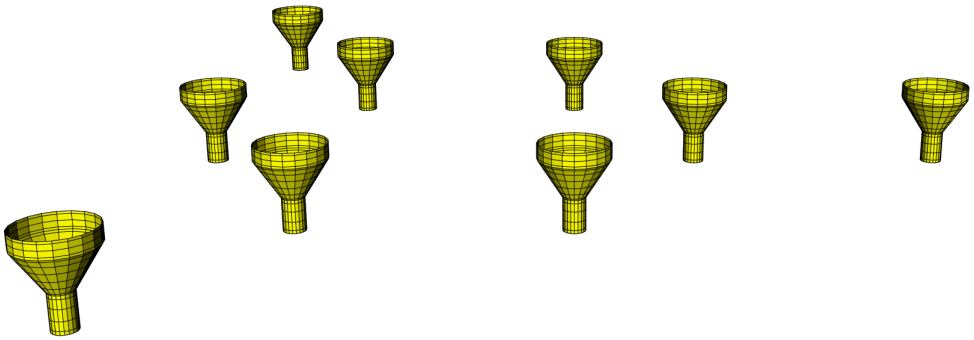


Figure 5.46: Hydrodynamic mesh of the Hex device used for hydrodynamic preprocessing.

Table 5.9: Centre of Gravity position of each node in the Hex device.

		CoG		
		x (m)	y (m)	z (m)
Node	**D3**	0.000	0.000	-0.300
	C1	0.750	-1.300	-0.300
	S1	0.000	-2.600	-0.300
	D1	2.250	-1.300	-0.300
	C3	0.750	1.299	-0.300
	S3	0.000	2.598	-0.300
	D2	2.250	1.299	-0.300
	C2	3.000	0.000	-0.300
	S2	4.500	0.000	-0.300

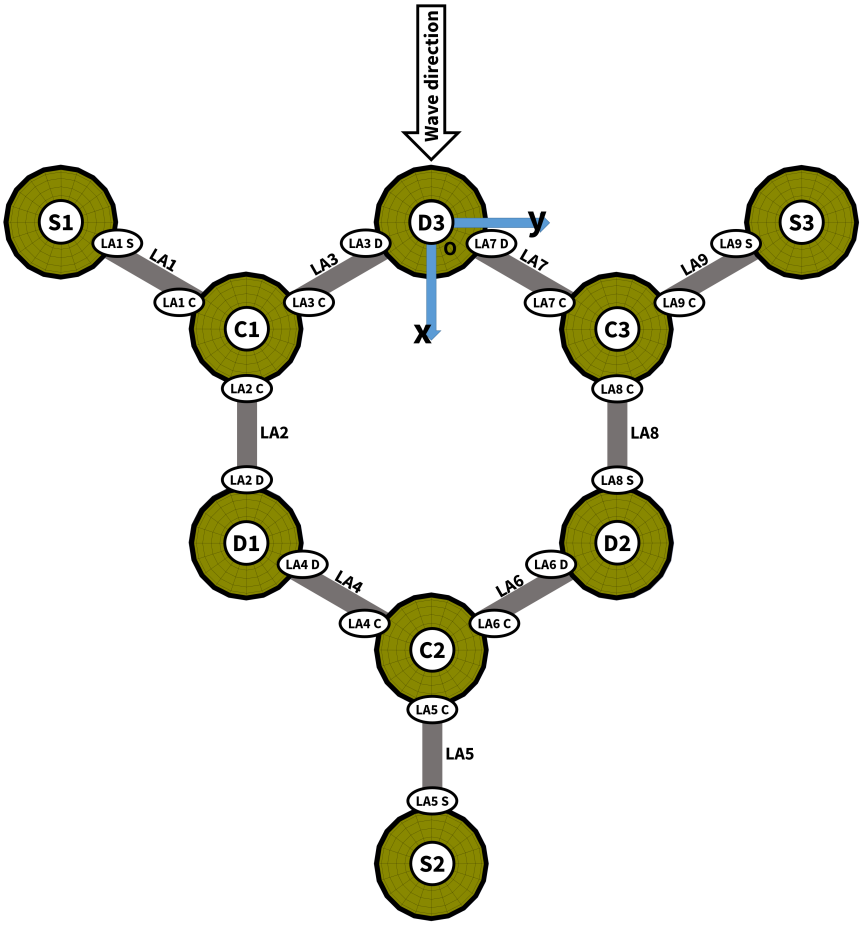


Figure 5.47: Naming convention for the Hex system (plan view). Adapted from McDonald et al. (2017).

5.4.2.3 Radiation Impulse Response Functions

Hydrodynamic coefficients were computed for 480 different frequencies over the range $0.02 \leq \omega \leq 24.0\ rad/s$ (i.e. $d\omega = 0.05\ rad/s$). To check the hydrodynamic interactions between the bodies, selected radiation impulse response functions (RIRFs) corresponding to significant degrees of freedom (such as the interactions in the Hex between nodes C1-D1 and C2-S2) were checked to ensure they had decayed (Figures 5.48 and 5.49).

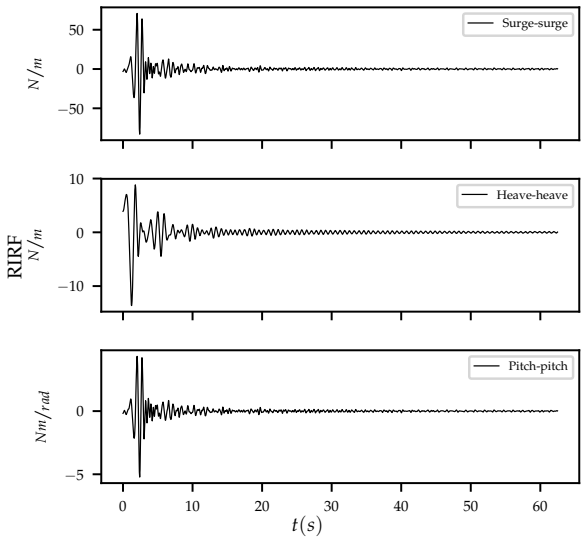


Figure 5.48: Hex radiation impulse response functions (RIRFs) for selected degrees of freedom between nodes C1 and D1.

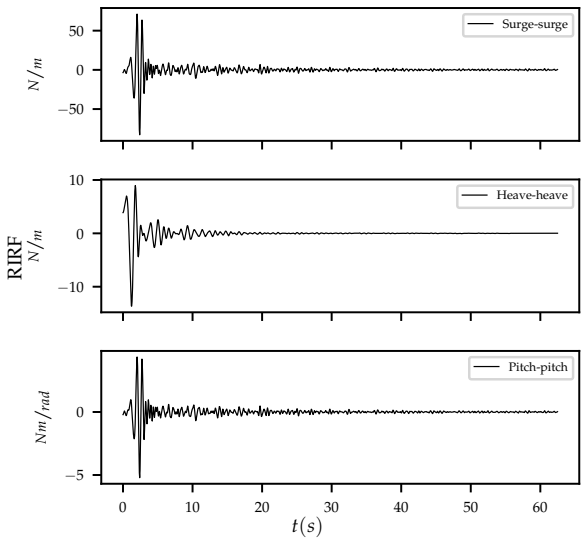


Figure 5.49: Hex radiation impulse response functions (RIRFs) for selected degrees of freedom between nodes C2 and S2.

5.4.3 Time-domain Models of the Squid and Hex

The Albatern machines are modelled in HOTINT using the standard `.hid` input file. Each node is modelled as a 6 DoF `Rigid3DKardan` (as is required for bodies in the system that experience hydrodynamic loads). The link arms are also modelled using `Rigid3DKardan` elements, although they do not experience hydrodynamic loads and are assumed to be neutrally buoyant. An example of a `Rigid3DKardan` element is shown below, for the node `D2`:

```
D2
{
  element_type = "Rigid3DKardan"
  name= "D2"
  loads = [nGravityF]

  Physics
  {
    mass = m_node
    Moment_of_inertia = [Ixx_node, 0, 0; 0, Iyy_node, 0; 0, 0, Izz_node]
  }

  Initialization.Initial_position = [(l_la*cos(pi/3))+l_la, (l_la*sin(pi/3)), zcog_node]
  Initialization.Initial_rotation = [0, 0, 0]

  Graphics.geom_elements = [nGeom1]
  Graphics.use_alternative_shape = 1
}
nD2 = AddElement(D2)
```

Note that the initial position of the node is defined algebraically, using the link arm length `l_la` as the variable (instead of defining the position numerically). The entire system is defined in this fashion; enabling changes to the whole system to be made simply by modifying a small number of variables.

The nodes and link arms are connected to each other using `PointJoint` elements, which are based on a Lagrange multiplier formulation to represent the joint as an exact kinematic constraint. An example of the ball joint connecting node `S1` to link arm `LA1` is shown below:

```
LA1_S1
{
```

```
  element_type = "PointJoint"
  Position1
  {
   element_number= nLA1 %number of constrained element
   position= [-l_la/2, 0, 0] %local position
  }
  Position2
  {
   element_number= nS1 %number of constrained element
   position= [0, 0, -zcogjoint_node] %local position
  }
}
AddConnector(LA1_S1)
```

Where the variable `zcogjoint_node` is the vertical distance between the node's centre of gravity and the link arm's attachment point. Note that in the model, the link arm attaches at $x = y = 0m$ i.e. on the node's central axis. The bodies are essentially permitted to move 'through' each other: collisions are ignored.

The `PointJoint` constraint handles the kinematic constraint between the bodies, and ensures that each body can only rotate relative to the other (no translations), but to model the stiffness and damping at the joint introduced by the PTO, further constraints are required. Two `CoordinateConstraint` elements are applied to each joint to apply the PTO's stiffness and damping to the relative pitch and yaw between the bodies:

```
LA2_C1_pitch
{
  element_type= "CoordinateConstraint"
  Coordinate1
  {
    element_number = nLA2 %element number for coordinate 1
    local_coordinate = 5 %local coordinate of element 1
  }
  Coordinate2
  {
    element_number = nC1 %element number for coordinate 2
    local_coordinate = 5 %local coordinate of element 2
  }
  Physics
  {
   use_penalty_formulation = 1
   Penalty.spring_stiffness = k_joint
   Penalty.damping = c_joint
```

```
  }
  Graphics
  {
   drawing_size = 0.01
  }
}
AddConnector(LA2_C1_pitch)
```

This constraint has also been used to constrain the relative roll between the bodies. Again, the stiffness and damping are defined algebraically by the variables `k_joint` and `c_joint`, which has enabled batch runs to be performed without re-generating the whole `.hid` file - instead just the first few lines (where the variables are defined) are updated at each iteration.

To model the Squid and Hex's net mooring systems, `Rope3D` (linear spring-damper) elements have been used to model the mooring straps, each side of the triangular grid and the catenary lines. The ropes can be attached to each other via 6 DoF `Rigid3DKardan` elements (denoted 'rope connectors' or `RC` for short), and to the nodes:

```
Strap_D1_RopePort
{
  element_type= "Rope3D"
  name= "Strap_D1_RopePort" %name of the element
  Graphics.draw_size = 0.04
  Physics
  {
    Penalty
    {
      rope_stiffness = k_mooring
      damping = c_mooring
    }
  }
  Geometry
  {
    element_numbers= [nD1, nRC_D1_RopePort] %element numbers of the suspension points
    positions= [0, 0, zcogjoint_node ; 0, 0, 0]
  }
}
nStrap_D1_RopePort = AddConnector(Strap_D1_RopePort)
```

As shown in the example for node D1's mooring strap, the 'fore/aft/port/starboard' naming convention has been used to define the mooring components of the mooring

system. As with the PTO elements, the stiffness and damping properties are defined by the variables `k_mooring` and `c_mooring`, which can easily be modified in the `.hid` file header for performing batch runs with parametric variations.

The complete `.hid` input file for the Hex system with net mooring configuration consists of 3550 lines of text; a visualization of the system is shown in Figure 5.50.

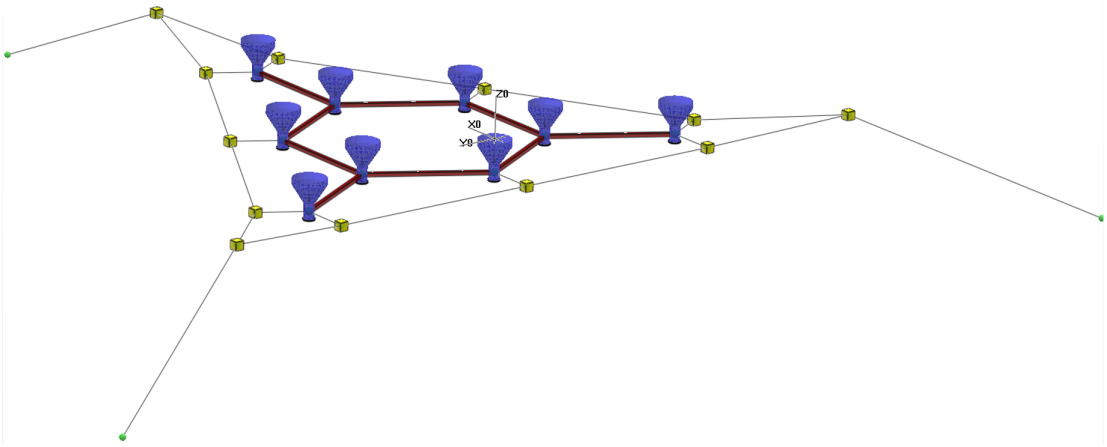


Figure 5.50: Visualization of the Hex HOTINT model with net mooring system.

5.4.4 Mooring Variations

This section demonstrates the range of mooring variations that can be tested in the Albatern models. The same relative RAOs considered by McDonald et al. (2017) (Figure 5.37) are compared in order to observe how changes to the model affect these particular degrees of freedom.

5.4.4.1 Net/simplified Mooring Systems

The model of the net mooring system (Figure 5.50) closely resembles the physical model's mooring system, and could potentially produce the most accurate results. However, the mooring system is complex and HOTINT's `Rope3D` elements do not permit a pre-tension to be applied by default. In early iterations of the model, the pre-tension was set by applying a buoyancy force to the corner buoys in the mooring system, to allow the net to 'rise up' and create a pre-tension in the ropes dynamically.

However, this could induce a transient response in the system that took time to stabilize; thus increasing the amount of time required to generate RAOs for batch runs with parametric variations.

A potential solution to this issue is to simplify the mooring system, so that the ends of the mooring strap are fixed in space, instead of being attached to the triangular mooring grid (Figure 5.51).

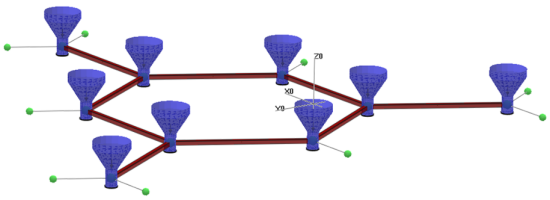


Figure 5.51: Visualization of the Hex HOTINT model with 'simple' mooring system ($k_{rope} = 300^{N}/_{m}$, $c_{rope} = 100^{Ns}/_{m}$ and $T_{rope,0} = 0N$).

The results of the simplified mooring system are compared with the net mooring system in Figures 5.52 and 5.53. No pre-tension is applied in this configuration. The rope stiffness, $k_{rope} = 300^{N}/_{m}$ at model-scale ('ms') corresponds to a stiffness of $97100^{N}/_{m}$ at full-scale ('fs'):

$$k_{rope,fs} = \frac{k_{rope,ms}}{(1/18)^2} \tag{5.19}$$

$$\frac{300}{(1/18)^2} = 97.1 \cdot 10^3 {}^{N}/_{m} \tag{5.20}$$

This is a similar rope stiffness used in the F-3OF models - $100 \cdot 10^3 {}^{N}/_{m}$) (Section 5.3). For the damping, a value of $100^{Ns}/_{m}$ was used; comparable to the maximum surge radiation damping value determined for a single node (Figure 5.43).

From Figures 5.52 and 5.53 we can see that the choice of mooring topology has a much more significant effect on the yaw RAOs than the pitch RAOs (which are virtually unaffected by the change in mooring configuration).

In the simplified mooring configuration, the response of the relative yaw RAOs increases.

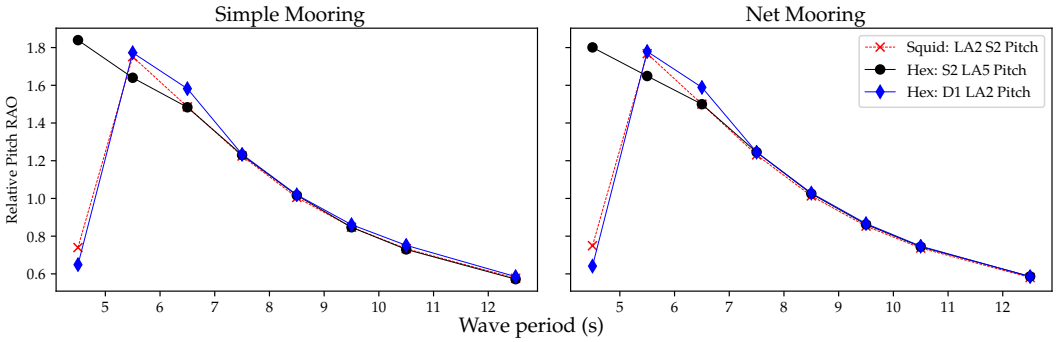


Figure 5.52: RAOs of the relative pitch between selected bodies in the Squid and Hex systems for different mooring systems ($k_{rope} = 300^{N}/_{m}$, $c_{rope} = 100^{Ns}/_{m}$ and $T_{rope,0} = 0N$).

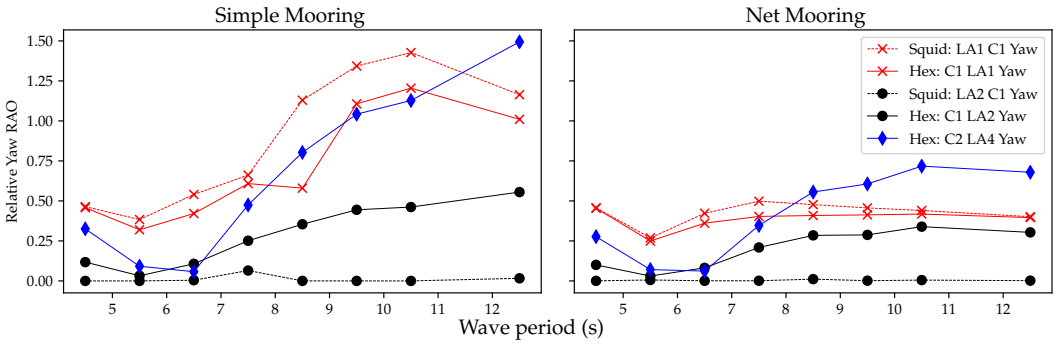


Figure 5.53: RAOs of the relative yaw between selected bodies in the Squid and Hex systems for different mooring systems ($k_{rope} = 300N/m$, $c_{rope} = 100Ns/m$ and $T_{rope,0} = 0N$).

5.4.4.2 Pre-tension in the Mooring Straps

Different pretension values have been applied to the Squid and Hex mooring straps in the simple mooring configuration (Figure 5.51). A value of $250N$ was selected to provide a similar effort to the buoyancy of each node, with $100N$ an intermediate value.

Figures 5.54 and 5.55 show that the amount of pre-tension in the mooring straps can have a significant effect on both the pitch and yaw RAOs. The response of the pitch RAOs is reduced for larger pre-tensions, and the resonant frequency shifts towards longer periods. For the yaw RAOs, the response is less affected and the resonant frequency shifts in the opposite direction: towards the shorter periods.

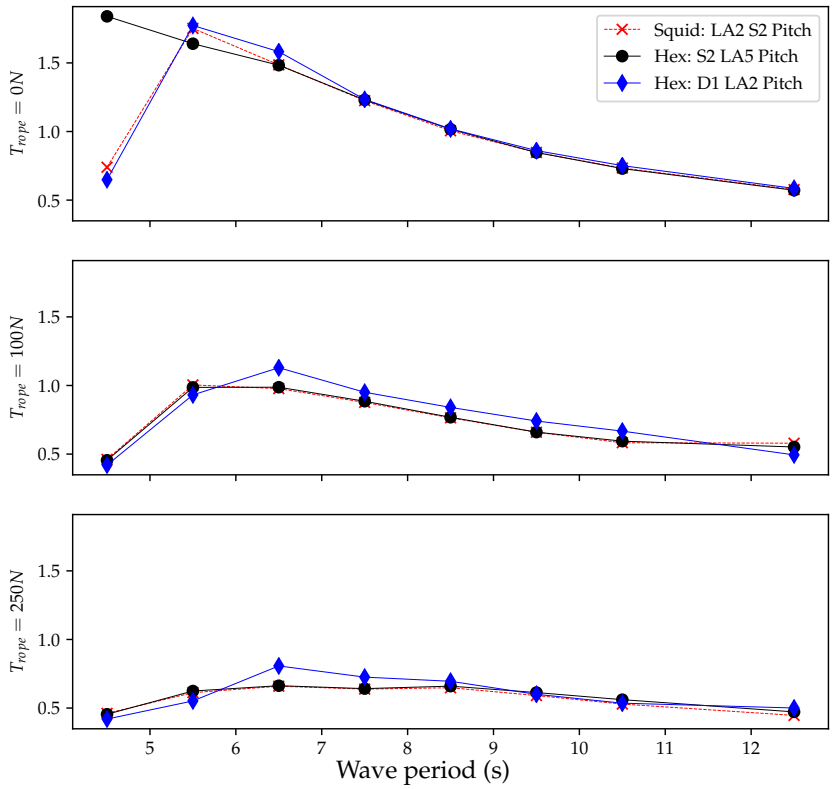


Figure 5.54: RAOs of the relative pitch between selected bodies in the Squid and Hex systems for different pre-tensions in the ropes ($k_{rope} = 300N/m$, $c_{rope} = 100Ns/m$).

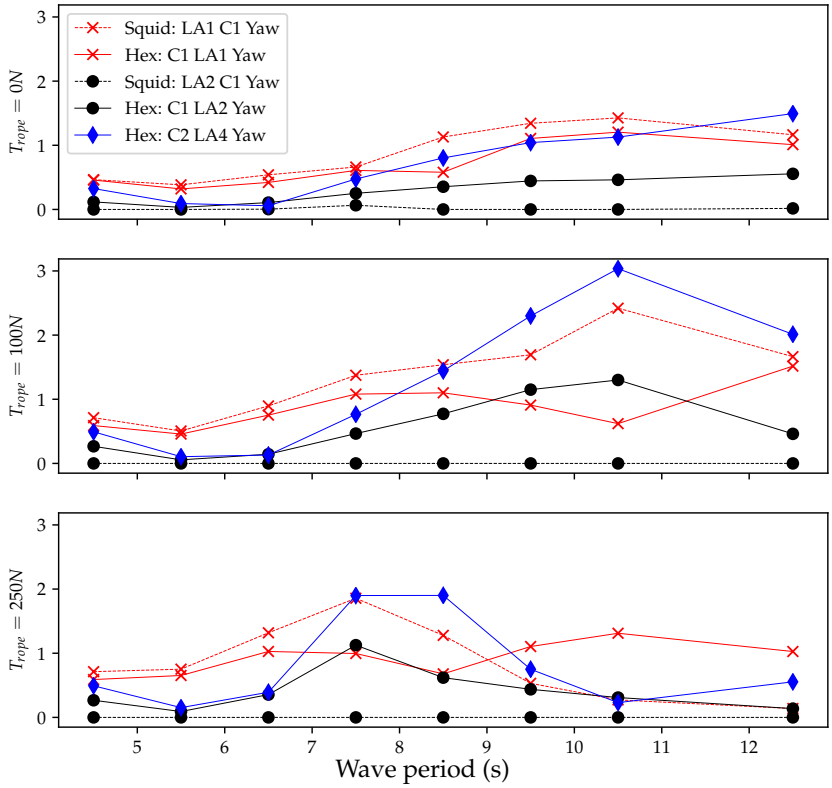


Figure 5.55: RAOs of the relative yaw between selected bodies in the Squid and Hex systems for different pre-tensions in the ropes ($k_{rope} = 300N/m$, $c_{rope} = 100Ns/m$.

5.4.4.3 Variations in Mooring Strap Stiffness & Damping

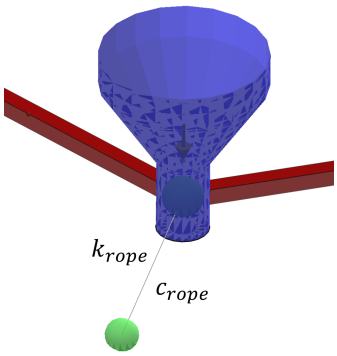


Figure 5.56: Illustration of a mooring strap in HOTINT, with stiffness and damping properties.

The Hex and Squid models with 'simple' mooring configurations were selected for batch processing, with varying mooring strap stiffness (k_{rope}) and damping (c_{rope}) properties at each iteration. Values either side of the initial stiffness ($300^{N}/_{m}$) and damping ($100^{Ns}/_{m}$) values were tested, to observe increasing and decreasing the rope stiffness and damping properties affects the response of the selected degrees of freedom.

Figure 5.57 shows that the pitch RAOs are relatively unaffected by the change in rope properties, whereas Figure 5.58 shows that the yaw RAOs are much more influenced by the mooring strap's properties.

The changes observed in the yaw RAOs seem reasonable: increasing the stiffness of the mooring straps lowers the resonant frequency, whereas increasing the damping of the mooring straps flattens the response. The pitch RAOs could be less influenced by the changes if the pre-tension value is much higher than the dynamic loads.

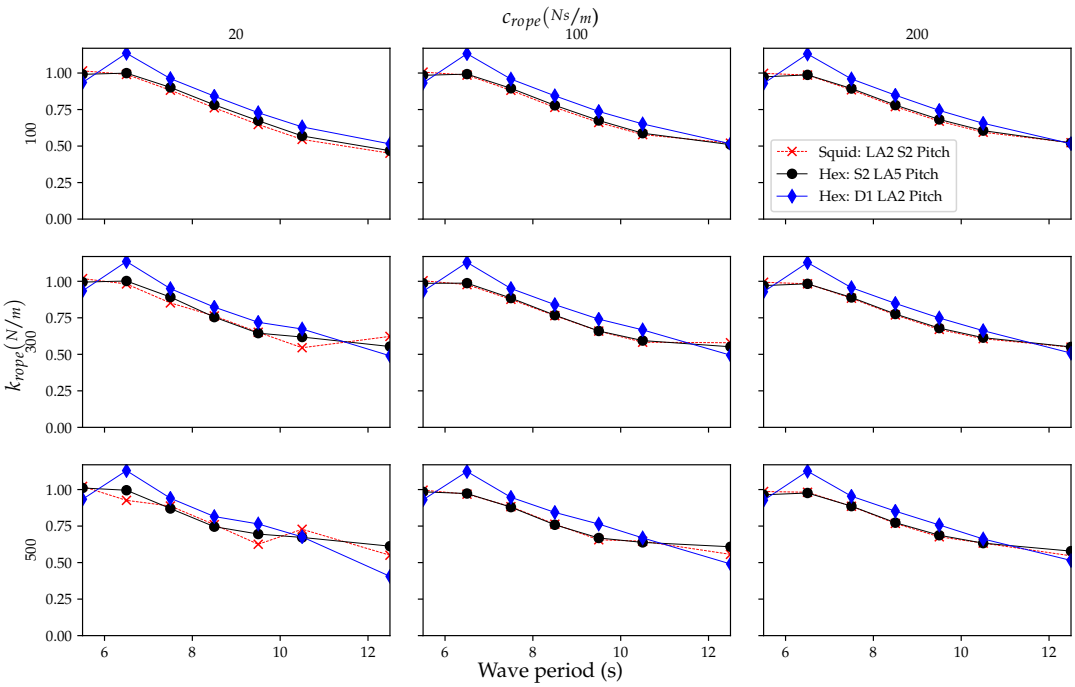


Figure 5.57: RAOs of the relative pitch between selected bodies in the Squid and Hex systems (k_{PTO} and $c_{PTO} = 0$).

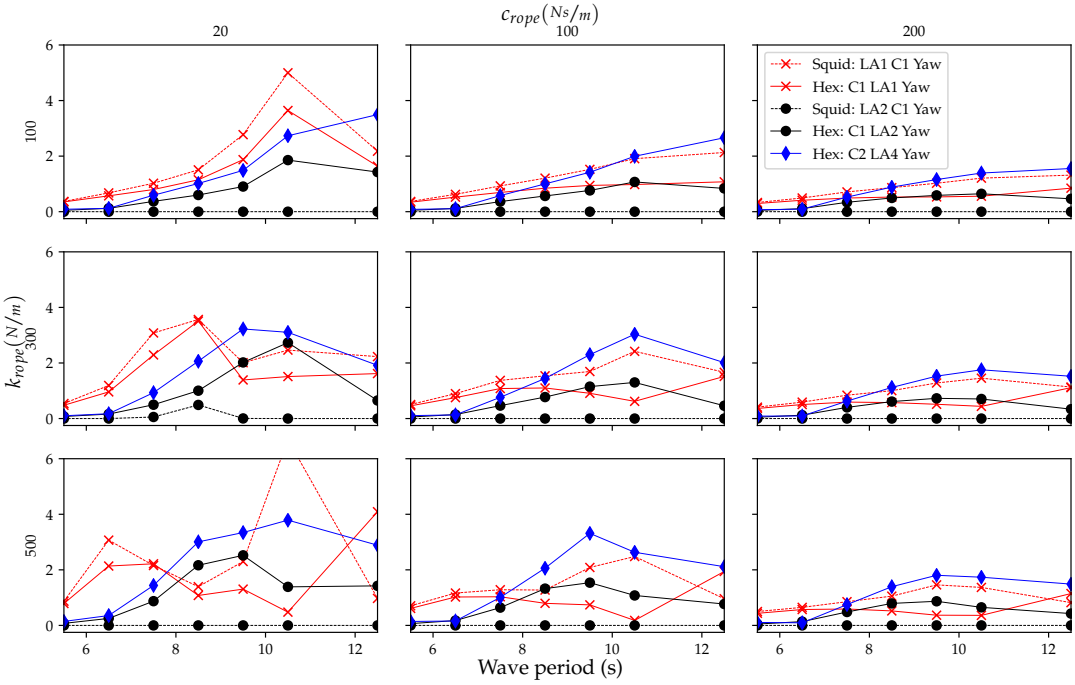


Figure 5.58: RAOs of the relative yaw between selected bodies in the Squid and Hex systems (k_{PTO} and $c_{PTO} = 0$).

5.4.5 PTO Variations

Again, the Hex and Squid models with 'simple' mooring configurations were selected for batch processing; in this instance varying the PTO stiffness and damping at the system's joints (k_{joint} and c_{joint}) at each iteration.

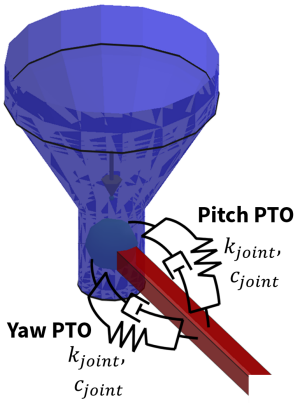


Figure 5.59: Illustration of the stiffness and damping applied to the relative yaw and pitch between each node and link arm.

A joint stiffness of $50^{Nm}/_{rad}$ was selected to provide a similar effort to the hydrostatic force on the node in pitch and roll (Table 5.7). A higher value of $500^{Nm}/_{rad}$ has also been selected, which corresponds to $52.5 \cdot 10^{6\,Nm}/_{rad}$ at full-scale; similar to a joint (machinery) stiffness selected by Babarit, Hals, et al. (2011) for the B-OF device.

A joint damping value of $6^{Nms}/_{rad}$ was selected to provide a similar effort to the maximum radiation damping observed in pitch for an individual node (Figure 5.43). Two values either side of this ($3^{Nms}/_{rad}$ and $9^{Nms}/_{rad}$) were also tested to help understand how varying the amount of damping in the joints affects the selected RAOs.

Figure 5.60 shows that the pitch RAOs are relatively unaffected by the changes in joint damping. Increasing the joint stiffness flattens the response, but not seem to affect the resonant period.

Figure 5.61 shows again that that yaw RAOs are more influenced by the joint stiffness and damping changes than the pitch RAOs. Increasing the damping flattens the response, and increasing the stiffness from $0^{Nm}/_{rad}$ to $50^{Nm}/_{rad}$ shifts the resonant periods towards shorter wave periods.

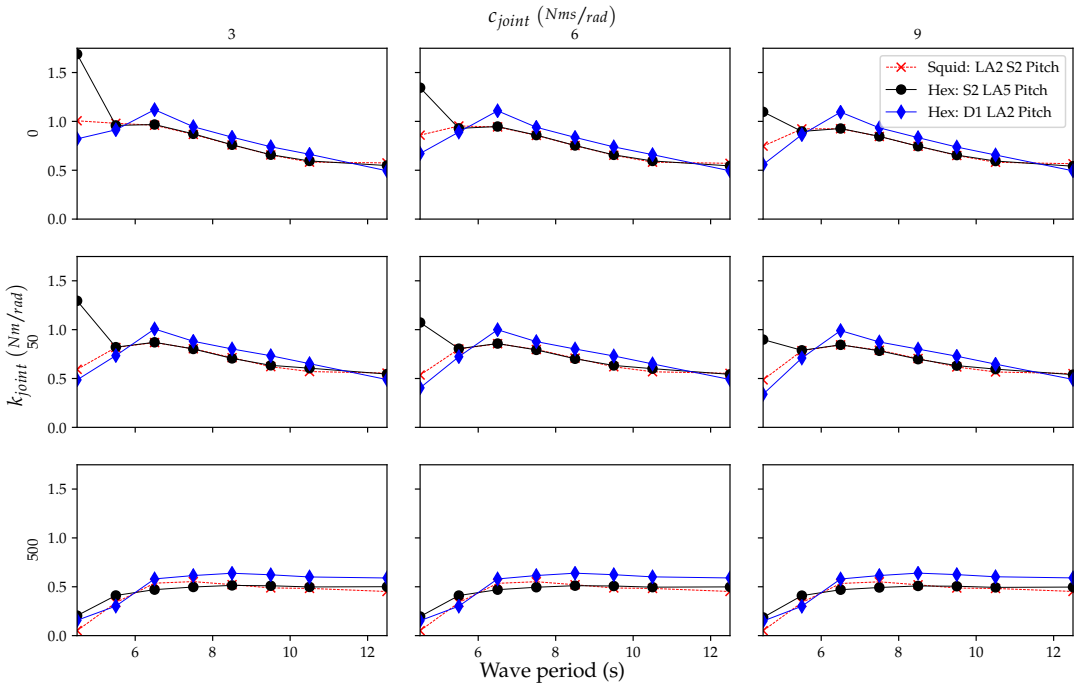


Figure 5.60: RAOs of the relative pitch between selected bodies in the Squid and Hex systems ($k_{rope} = 300N/m$ and $c_{rope} = 100Ns/m$).

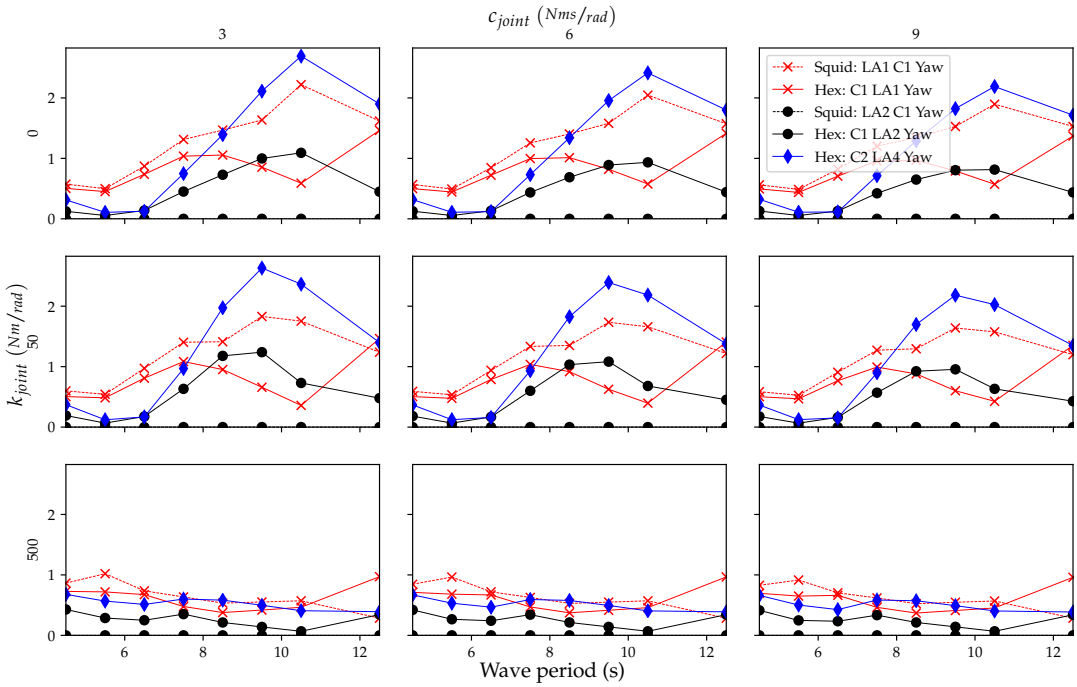


Figure 5.61: RAOs of the relative yaw between selected bodies in the Squid and Hex systems ($k_{rope} = 300^{N}/_{m}$ and $c_{rope} = 100^{Ns}/_{m}$).

5.4.6 Link Arm Length Variations

The HOTINT models of the Squid and Hex have been defined algebraically (both in the net mooring and simplified mooring case). Hence, the positions of all the elements in the system can be modified for different array sizes simply by changing the link arm length variable, l_{la}. NEMOH must be also be re-run for each l_{la} value in order to compute the correct hydrodynamic coefficients.

The link arm length is an important characteristic of the Squid and Hex devices. Figure 5.63 shows that the pitch RAOs natural frequency can be modified by changing l_{la} and that for the longest l_{la} there is a significant difference between the RAOs of the Hex and Squid. However, as only 3 relative DoFs in the system are being considered, there is not enough evidence to draw conclusions about the effect of coupling individual Squid units together. To fully understand the effect of increasing the link arm length, it is necessary to consider the whole array, which may require analyzing the combined power output.

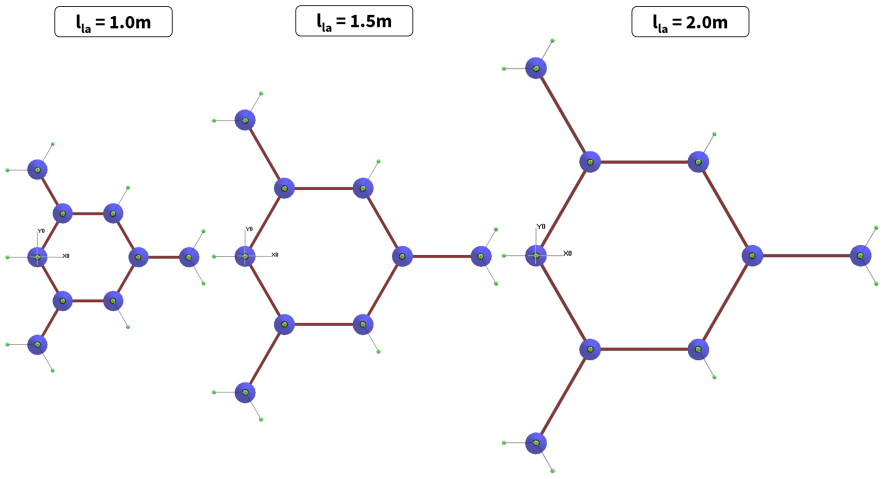


Figure 5.62: The different Hex models with simplified mooring system and varying link arm length.

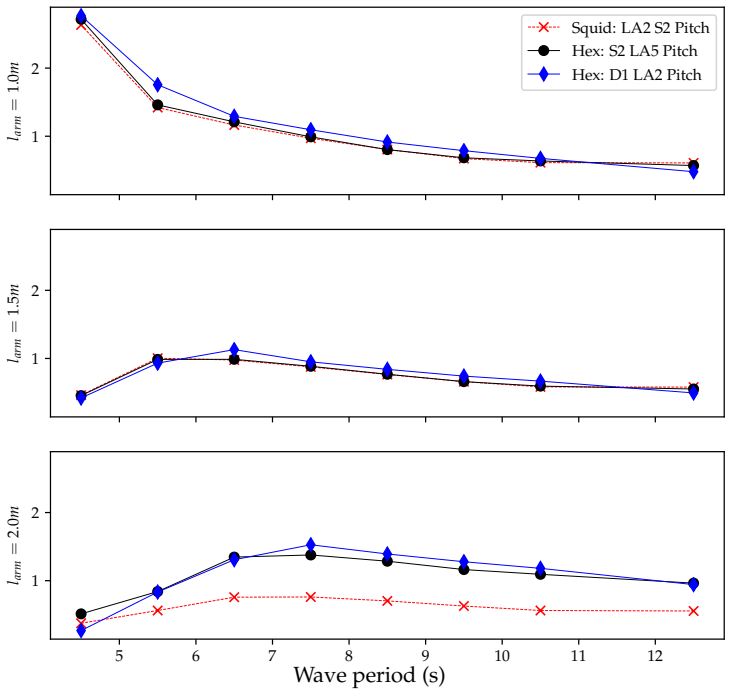


Figure 5.63: RAOs of the relative pitch between selected bodies in the Squid and Hex systems for different link arm lengths ($k_{rope} = 300N/m$, $c_{rope} = 100Ns/m$) and $T_{rope} = 100N$.

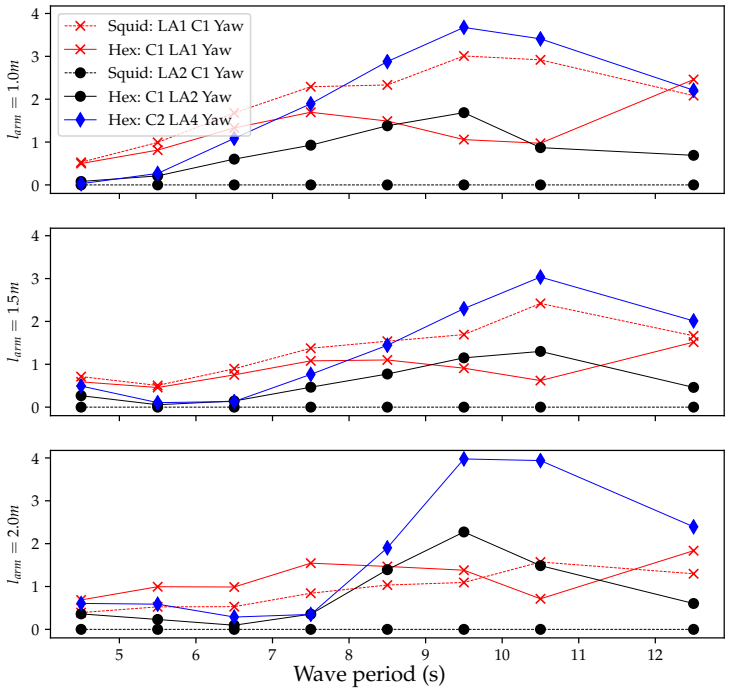


Figure 5.64: RAOs of the relative yaw between selected bodies in the Squid and Hex systems for different link arm lengths ($k_{rope} = 300N/m$, $c_{rope} = 100Ns/m$) and $T_{rope} = 100N$.

5.4.7 Irregular Waves and Power Production

As identified in the previous subsection, RAOs of individual DoFs in the system can reveal very useful information, but is insufficient to assess the performance of the whole device. A useful way to analyze the whole system could be to compute the total power output, of all the DoFs. This is demonstrated for the Squid device with net mooring configuration (Figure 5.65).

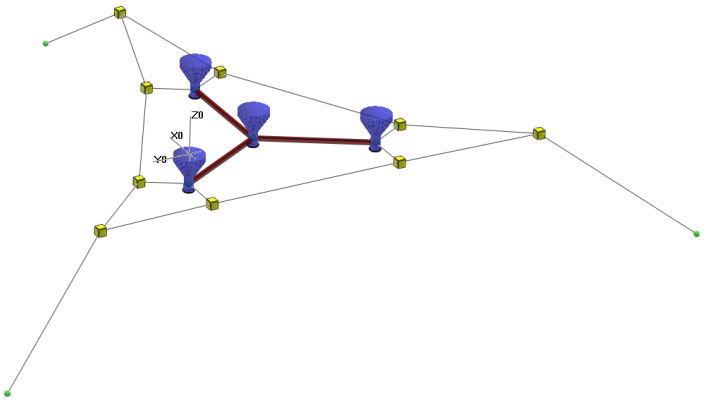


Figure 5.65: Illustration of the Squid model with net mooring configuration used to demonstrated the computation of combined power output.

To compute the combined power output, the HOTINT source code was modified directly, to multiply the damping force computed by each `CoordinateConstraint` by the relative velocity corresponding to that PTO. The data was stored in an object during the simulation and printed to a `.txt` file once the simulation had completed.

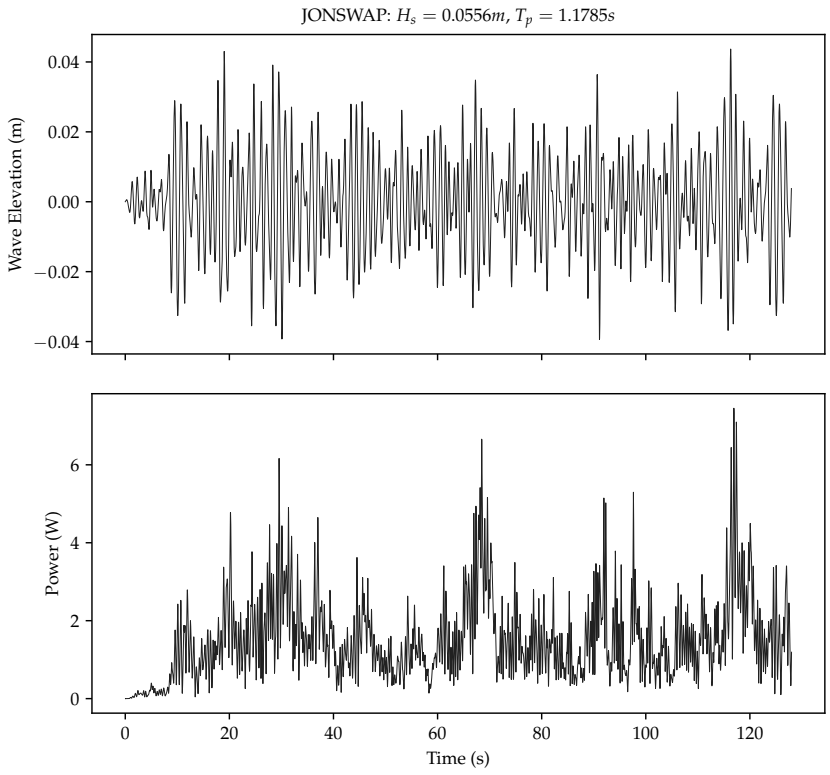


Figure 5.66: Total power output for the 1:18 scale Squid model in irregular waves, $k_{joint} = 50^{Nm}/_{rad}$, $c_{joint} = 6^{Nms}/_{rad}$.

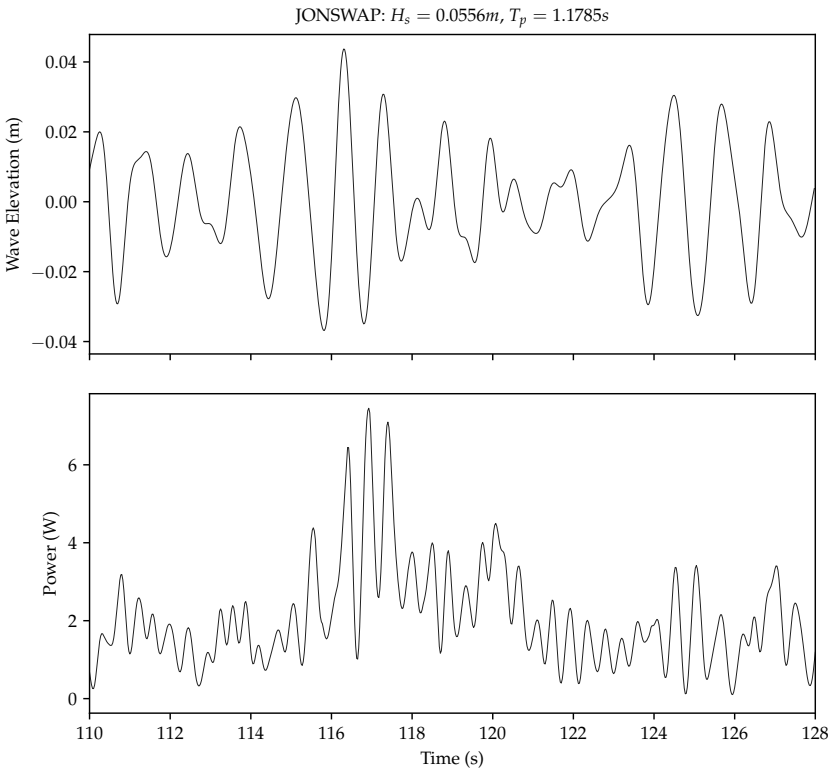


Figure 5.67: Total power output for the 1:18 scale Squid model in irregular waves, $k_{joint} = 50^{Nm}/_{rad}$, $c_{joint} = 6^{Nms}/_{rad}$ (110-128s region).

Chapter 6

Discussion

The IEA OES Task 10 project has enabled the comparison of 29 different WEC modelling codes - to our knowledge this is the biggest WEC code comparison project to date. The project chose a simple model (the Sphere) in order to make code to code comparisons more straightforward. Despite its simplicity, the Sphere is a very useful test case that has revealed some important discrepancies between different codes. Only 4 of the participants from the original dataset have been selected here (to make visual comparisons easier on the eye), however the publication by Wendt et al. (2017), which compares the time series and RAOs of all of the codes, reveals small differences between each code. Hence, each participant's results fit into a wider 'band' of results, which can also be seen (to a lesser extent) in this chapter. This 'band' can be regarded as a degree of uncertainty to users of the codes. Because the discrepancies are quite small, the causes were not investigated in the original Task 10 project, but having access to InWave and InWave-HOTINT was very useful for identifying sources of error, as a range of different variables could be compared directly.

It was surprising to observe discrepancies between the InWave and InWave-HOTINT results, as it was initial thought that the same input parameters had been used for both models. The difference in natural period between InWave and InWave-HOTINT results made the the hydrostatic force, $\vec{f}_{hs}$ a likely source of the discrepancy. Indeed,

examination of the hydrostatic stiffness matrix, $\mathbf{K}_H$, used in InWave reveals some small errors:

$$\mathbf{K}_{H(InWave)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 768650 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5122142 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5122142 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} {}^{N}/_{m} \\ \text{and} \\ {}^{N \cdot m}/_{rad} \end{matrix} \tag{6.1}$$

Note that because the sphere is only allowed to heave, $K_{H,33}$ is the only relevant term here, which is slightly (0.2%) smaller than the $770215^{N}/_{m}$ value used in InWave-HOTINT. This is enough to cause the discrepancies observed in the time-domain results.

It has been determined that InWave $\mathbf{K}_H$ results are different because they have been computed for the Sphere with a slightly lower draft: the position of the Sphere's CoG is slightly lower - at $[0, 0, -2.00617]m$. In other words, the Sphere is sitting $6mm$ lower in the water column in InWave, when at equilibrium. However, the Sphere's CoG position was defined a priori, and should be $z_{CoG} = -2.0m$.

This problem occurs because InWave first recomputes the body's equilibrium position (via a time-domain simulation) according to the body's submerged volume, and the resulting weight and buoyancy vectors. Mesh imperfections result in a slightly different volume to the volume calculated analytically. It is possible that a similar error is present in some of the results of the other codes. A workaround for this specific problem has been developed in InWave (although not used here), which only allows the equilibrium computation to perform a single time-step - effectively forcing this routine to return the original equilibrium position.

The equilibrium-finding algorithm is a very useful part of the InWave code - especially when working with complex geometries. However the routine has not yet been required in InWave-HOTINT, and should therefore be included in the future. The Sphere plots show some results that include nonlinear hydrostatic and Froude-Krylov fources (the aNySIM and WEC-Sim codes). Again, these options are available in InWave (al-

though not used in this project) and should be included in InWave-HOTINT in the future. As these modifications are made to the InWave-HOTINT codebase, these verification checks should be made frequently - the Sphere is an excellent model for that.

The B-OF model results show the best agreement between InWave and InWave-HOTINT out of all the test cases. This is despite InWave only having a nonlinear hydrostatic solver for multibody systems, and InWave-HOTINT only having a linear hydrostatic solver. Hence, the nonlinear hydrostatic effects can be successfully avoided by performing decay tests with very small displacements and computing RAOs with very small amplitude waves. The good agreement is most likely due to the fact that the B-OF device is not freely floating, as it is attached to the (fixed) foundation body via a hinge joint. Hence, it is practically impossible to compute the system's equilibrium position incorrectly.

When compared to the NumWEC results, a small discrepancy was observed at the resonant period. As the equilibrium position error cannot be a factor in this case, it is possible that the radiation input parameters (i.e. RIRF length, t_{RIRF}) are different in both cases. However, a more likely error is that the NumWEC results are based on a B-OF mesh that features chamfers at the device's hinge. This would almost certainly affect hydrostatic properties, as the calculated volume (and therefore buoyancy vector) must be reduced due to the presence of the chamfer.

Using simple test cases has been invaluable in the development of InWave-HOTINT, enabling simple bugs and errors to be identified quickly. Although they do not demonstrate the tool's full capabilities, it is highly recommended that simple test cases are continued to be used in future development efforts.

The F-3OF test is more complex than the Sphere and B-OF cases, and to date there has not been a satisfactory level of agreement between code-to-code comparisons for this test case. Nonetheless, during the development of InWave-HOTINT, the F-3OF model revealed some important sources of error which be useful for future code-to-code verification work based on this model.

Despite the relatively simple topology of the F-3OF model, there has been significant discrepancies in the results of this model between the different codes - even in the results of simple decay tests. Thanks to having full access to InWave and InWave-

HOTINT, it has been determined that some of these discrepancies have been due to the poor selection of certain model inputs

Chapter 5 discussed some of these errors in details; as they helped guide the improvement of InWave-HOTINT's results. In summary:

- RIRFs computed using trapezoidal integration methods results in a periodical function: if the length of the RIRF used in the time-domain simulations is too long, it may include more than one cycle and produce unstable results.

- As with the Sphere, care needs to be taken to ensure that the defined equilibrium position of any floating bodies is consistent with the hydrostatic/buoyancy properties of the floating body (and hence its volume)

Although at this moment it is difficult to say whether or not the other participant's results have been influenced by the length of t_{RIRF} or not, it is certain that all of the participants' results suffer from small offsets in hydrostatic equilibrium positions, which will affect natural period and potentially the radiation damping force too. This reinforces the point that a great deal of care must be taken in setting up the numerical model of the WEC - even with regards to the relatively simpler aspects of the model. In future code-to-code comparison projects, the inclusion of an additional 'static' test (i.e. still water, each body is set to its equilibrium position at $t = 0s$) could help identify any hydrostatic problems in the models, and remove this as a possible source of error before moving on to more complex tests.

As the redundant coordinate multibody dynamics method must compute all of the redundant DoFs in the system, and the hydrodynamic solver in this approach computes all of the hydrodynamic interactions between redundant DoFs in the system, the performance of InWave-HOTINT was a topic of interest for Innosea from the beginning of its development. However, it is usually difficult to give an accurate and fair comparison of different software packages as there are many variables that can affect the results. Hence, any performance comparison should be considered indicative only. With this in mind, some computation times are provided for InWave and InWave-HOTINT for the different case studies presented in Chapter 5.

To compare the performance of InWave and InWave-HOTINT, a timed free decay test

was repeated 10 times for the Sphere, B-OF and F-3OF models. At the time of testing, a linear hydrostatic force was not available for multibody systems in InWave, hence the B-OF and F-3OF models include a nonlinear hydrostatic force (whereas InWave-HOTINT computes a linear hydrostatic force for all models). All other parameters ($d\omega$, t_{RIRF}, dt, t) and force models are the same in both codes.

Table 6.1: Comparison of InWave and InWave-HOTINT's computation times for each case study.

Model	Statistic	InWave	InWave-HOTINT
Sphere	mean computation time (s)	13.41	7.62
	standard deviation (s)	2.16	0.16
B-OF	mean computation time (s)	139.27	90.98
	standard deviation (s)	21.39	1.33
F-3OF	mean computation time (s)	226.12	106.16
	standard deviation (s)	102.57	1.15

Table 6.1 shows that the InWave-HOTINT computation times were significantly lower than InWave's across all tests. The case with the highest number of bodies (F-3OF) was also the case where InWave-HOTINT was most faster than InWave - the opposite of what was expected. It is unclear how much the nonlinear hydrostatic solver impacts InWave's performance in this case, hence further work is needed in order to fully understand the differences.

It seems logical to expect that as the number of bodies in the WEC system grows, at some point the InWave computation times would start to become lower than InWave-HOTINT. To determine the point at which this occurs, a model could be created (e.g. based on the Pelamis device) that can be easily extended so that new bodies can be added to the chain and the point at which it is more beneficial to use the reduced-coordinate algorithm can be identified.

The standard deviation values calculated for the different data reveal that InWave-HOTINT had a much smaller spread of results compared to InWave. Again, the reasons for this are not fully understood, but it could be useful when performing batch runs in InWave-HOTINT, as the computation time estimates will be much more reliable than InWave's.

One of the main goals of InWave-HOTINT was to offer the full range of HOTINT's ca-

pabilities, combined with the ability to apply hydrodynamic loads to selected rigid bodies in the system. This goal has been achieved, and has been demonstrated in the Albatern models, where the full system (nodes, linkarms, joints, PTO, mooring) is modelled in HOTINT, but only the nodes are modelled in NEMOH. Hence, in the HOTINT model, only the nodes experience hydrodynamic loads.

Despite estimating the Albatern S12 dimensions from photographs, some of the RAO results have actually been encouraging. Many of the pitch RAOs presented show resonant frequencies around $6.5s$ - similar to the RAOs presented by McDonald et al. (2017). Furthermore, the yaw RAOs have shown similar resonant frequencies to the data presented by McDonald et al., and similar relationships between each relative DoF that has been compared (i.e. the Hex C2 LA4 yaw typically produces the largest response, whereas the Squid LA2 C1 yaw is negligible). Hence, the InWave-HOTINT model could potentially provide a useful basis for experimental validation work in the future using the Albatern data recorded at FloWave.

One of the main differences between the InWave-HOTINT and physical model results is that the yaw RAOs typically produced a much higher response than the pitch RAOs: the opposite to what was observed in the physical model results. However, the initial comparison of the net and simplified mooring system (Figure 5.53) shows that the net mooring configuration produces a much lower yaw response, and is actually quite similar to the physical model results. This makes sense as the net mooring configuration is a closer representation of the physical model's mooring, and it should be expected that this configuration is more compliant.

Nonetheless, as experimental validation was not the goal with this model, and as the simplified mooring model had some benefits of avoiding transient responses, which can reduce the simulation time required to reach a steady-state (necessary for RAO computation), the simplified mooring model was used for the subsequent testing[1]. Nonetheless, the difference between the results of both mooring models highlights the importance of having a flexible WEC modelling tool: as features like this can have a significant impact on the numerical results.

[1]It should be mentioned that since conducting this work, an alternative method of applying pre-tension to the `Rope3D` elements has been developed, which could help to reduce the transient response when working with mooring pre-tension

6.1 Summary

In general, the InWave-HOTINT results have compared very well against similar WEC modelling codes. Where discrepancies have been observed, likely sources of error have been identified and in some complex cases where the discrepancies have been significant and difficult to explain (e.g. F-3OF decay tests 1 & 3), the InWave-HOTINT results look more reasonable than the other codes, which have displayed small amounts of instability and/or incorrect equilibrium positions.

Significant discrepancies can occur between codes due to the selection of t_{RIRF}, with poor selection easily resulting in unstable responses in decay tests. It is therefore important not only to have a robust checking method to ensure that the RIRF has converged for all degrees of freedom, but for code-to-code comparisons to share this parameter that it is consistent across models to eliminate uncertainty.

Despite its perceived (relative) simplicity, the hydrostatic force, $\vec{f}_{hs}$ has been a common error source, with many codes displaying slightly offset heave equilibrium positions in the time-domain results. This can affect the hydrostatic stiffness matrix, $\mathbf{K}_H$ - resulting in different natural periods in the time domain. Furthermore, the different drafts of the models can affect the hydrodynamic interactions between bodies in a multibody system.

The issues explained here are not necessarily programming errors within the actual codes, but are more likely to be modelling errors which can be improved by adopting a more robust modelling strategy. Hence, for future code comparison projects, the following changes could enable easier, fairer comparisons of the codes:

- The use of the same frequency step ($d\omega$) for HDB computation.
- The selection of the same impulse response length, t_{RIRF}, ensuring that the RIRFs have converged for all DoFs.
- An additional time-domain test (before performing decay tests) setting the system at its equilibrium position and (ideally) observing no movements. This can be a useful check to ensure that $\mathbf{K}_H$ has been computed consistently and accurately across different platforms.

The adoption of these steps in future code-to-code comparison projects should not only help to reduce the errors in between codes, but as participants move on to comparing more complex models (with stronger non-linearities, viscous drag forces and control systems), the results should become easier to compare (and the cause of any discrepancy easier to identify).

Chapter 7

Conclusion

The main aim of this research was to advance the capabilities of Innosea's in-house WEC modelling software package (InWave) to enable the modelling of a wider range of WEC devices. This thesis has presented the research, development, verification and demonstration of a new WEC modelling platform: InWave-HOTINT, which overcomes some of the most critical limitations of InWave's mechanical solver and significantly advances Innosea's WEC modelling capabilities. To re-cap the original InWave mechanical solver's characteristics:

- InWave's original multibody dynamics solver is based on a reduced-coordinate approach ($O(n)$ complexity); redundant degrees of freedom are eliminated from the system's equations of motion with the aim of improving the solver's efficiency. This multibody dynamics approach (and therefore InWave) is optimized for modelling open-chain multibody systems.

- Closed kinematic loops are not possible in the original version of InWave and only revolute and prismatic joints are available to the user.

- Mooring and PTO forces in the original version of InWave are applied to specific generalized degrees of freedom in the system, which makes it difficult to model multi-DoF PTOs and net mooring configurations.

Reduced-coordinate multibody dynamics algorithms have historically been more commonplace in the wave energy sector (for their perceived efficiency advantages). But with InWave-HOTINT, this thesis demonstrates that a redundant-coordinate multibody dynamics approach can offer a much wider range of capabilities compared to reduced-coordinate algorithms. This is critical in the wave energy sector, where there has been practically no design convergence across the industry and a variety of working principles and mechanical features are currently being investigated. The versatility of redundant-coordinate multibody dynamics algorithms could also be advantageous to *individual* developers during design iterations; enabling them to explore a wide range of design options and to test significant changes to the mechanical topology of their machine.

Redundant-coordinate multibody dynamics codes are commonplace across many industries (such as robotics, vehicles and video games) - popular for their ability to model arbitrary multibody systems on the fly and to combine a wide range of different types of constraints within a model. Indeed, the use of a redundant coordinate approach in a WEC modelling code is not novel: the WEC-Sim code has also utilized such an algorithm - via MATLAB's closed-source solver, SimScape Multibody. However, this multibody dynamics solver utilizes sparse matrix methods (for performance improvements), which has prevented WEC-Sim from directly including off-diagonal added mass terms in the system's mass matrix. Instead, WEC-Sim uses a workaround to solve the WEC's equations of motion in the time-domain, which can be problematic in certain test conditions. Hence, there were two principle concerns in regards to implementing a redundant-coordinate solver in InWave:

- That the performance would be significantly worse than the original InWave code because the approach would inherently need to compute the mechanical system's redundant degrees of freedom *and* the hydrodynamic interactions between redundant degrees of freedom in the system.

- That stability/accuracy could be compromised if it was not possible to include the full added mass at infinite frequency matrix in the system's equations of motion.

with InWave-HOTINT, Both of these concerns have been allayed in this thesis. As part

of this research, 95 different multibody dynamics codes were assessed, which led to the selection of HOTINT; an open-source, redundant coordinate multibody dynamics code that avoids factorizing the system's mass matrix. HOTINT's algorithm is able to achieve comparable performance with reduced-coordinate algorithms and because the system's mass matrix is not factorized, it is possible to include the complete added mass at infinite frequency matrix at each time-step. Therefore, the potential flow time-domain equations can be implemented in full in InWave-HOTINT.

This thesis has demonstrated that InWave-HOTINT can achieve computation times that are comparable to, if not faster than InWave for multibody systems up to $n_{body} = 3$. However, it has not been possible to determine the exact reasons for this, as at the moment InWave can only use a nonlinear hydrostatic solver for multibody (> 1 body) systems and InWave-HOTINT can only use a linear hydrostatic solver. Hence, the only 'fair' comparison is for the Sphere test case, but the contribution of each force component towards each code's total simulation time has not actually been compared. In order to do this, both codes must be profiled: this should be a priority action before attempting any optimization/performance improvements, to ensure that any further development of the code(s) is based on evidence rather than speculation. Further work is also needed in order determine any scalability issues and to determine if there is a 'cross-over' point, where n_{body} starts to have a significant impact on the performance of the codes. This information could be used to help identify the comparative advantages of both InWave and InWave-HOTINT.

HOTINT's performance has no doubt benefitted from the 15 years of continuous development work underpinning the code - resulting in a well-organized, efficient solver. This work has also resulted in extensive libraries of elements (of bodies, constraints, controllers, etc.). Hence, the main InWave-HOTINT development goal has been to ensure that the WEC modeller has access to the full range of HOTINT's capabilities, combined with the ability to apply hydrodynamic loads to any selected rigid bodies in their model. Therefore, very little modification has been made to the original HOTINT codebase. The most significant modifications to the HOTINT code are primarily interfaces, which:

- Transfer kinematic data to the InWave hydrodynamic solvers (which have been extracted from the tight hydro-mechanical coupling in the original InWave code

and adapted for use with a redundant-coordinate algorithm).

- Transfer hydrodynamic forces back to HOTINT's multibody solver.
- Handle additional data structures & I/O routines to support the first two objectives.

Furthermore, the additional inputs required to perform the time-domain simulations are provided in a separate (`.hhi`) file, so that the modification to the HOTINT input file (`.hid`) is minimal (simply one extra line is required to define the location of the `.hhi` file). The InWave-HOTINT code is compiled together to create a single `.exe` program (i.e. the coupling is 'monolithic'). But the hydrodynamic solvers are self-contained `.cpp` and `.h` files. This coupling strategy should make it straightforward to transfer the hydrodynamics solver to an updated HOTINT codebase - the code is actively being developed and new releases are anticipated in the future. Although this modular structure has many benefits, it is not particularly user-friendly to have 3 separate input files (including NEMOH's `.cal` file), and this is an important future area of improvement for making InWave-HOTINT simpler and safer to use.

In parallel with the development of InWave-HOTINT, the code has been continuously verified against the original InWave code and against other WEC numerical modelling codes. Free decay tests have proved to be one of the most useful tests for comparing numerical results - and have revealed several errors commonly made in the field. Decay tests make it straightforward to identify discrepancies between the user-defined equilibrium position of body and the computed equilibrium position, which typically manifest as offsets in the time series data. This type of error can arise due to mesh imperfections resulting in inconsistencies between computed submerged volumes (and therefore buoyancy vectors) and theoretical values. This problem can also affect a body's hydrostatic stiffness properties as well as its draft - resulting in small discrepancies in natural periods. Decay tests can also reveal instabilities in the model - which can be related to poorly-defined equilibrium positions, or more likely poorly-truncated radiation impulse response functions (RIRFs). RIRFs that have been computed via a trapezoidal integration scheme typically produce a periodical function - if the RIRF is too long, it may not decay within the specified time period (or more precisely, it may decay and then grow again towards the end of the time series). A more advanced integration

routine that avoids producing a periodical RIRF could be implemented in the future, but for now a series of simple checks are used when modelling with InWave-HOTINT to ensure these common pitfalls are avoided. Further work could potentially make this process more user-friendly/safer - perhaps by automating some of the checking procedures.

The new capabilities of InWave-HOTINT have been demonstrated by creating a novel model based on the Albatern WEC. This model has several features that are difficult to model: a closed kinematic loop, multi-DoF PTOs attached to universal joints and a net mooring system. Each of these features is impossible to model with the original InWave code. Several examples of how the model could be used to perform data-driven design are presented, with RAOs for the Squid taking 25 minutes to produce, and the more complex Hex device typically taking around 2 hours to produce. Whether the net or simple mooring configuration is used has little affect on the computation time: indicating that the computational expense is largely dominated by the hydrodynamic terms.

7.1 Summary of Further Work

- Confidence in InWave-HOTINT is currently largely based on the verification of results against other WEC numerical modelling codes (all of which have been experimentally validated). In order to improve confidence in InWave-HOTINT, validation of the Albatern S12 model could be undertaken: either using physical model or CFD data.
- The hydrodynamics modelling options included in InWave-HOTINT could be expanded in several key areas:
  - Nonlinear hydrostatic force
  - Nonlinear Froude-Krylov force
  - Morison drag force

  All of these features are currently available in the latest version of InWave;

it should therefore be relatively straightforward to include them in InWave-HOTINT.

- After including nonlinear hydrostatics in InWave-HOTINT, a comprehensive performance comparison against InWave should be conducted, profiling both codes.
  - This performance comparison should be used to explore the scalability of both codes and identify the most critical parts of the codes influencing performance.
  - This work could be used to provide evidence-based guidance for future performance improvements (e.g. replacing the convolution integral with a state-space approximation method).

The user-friendliness of InWave-HOTINT could be improved in several key areas:

- At the moment 3 input files are required by the user, with some information repeated across files (e.g. CoG position is required in NEMOH `.cal` and HOTINT `.hid` files). This cross-over is a potential source of errors and some consolidation towards creating a single user input file should make future versions of InWave-HOTINT safer.
- At the moment, data analysis is primarily performed in Emacs org-mode, which permits the embedding of python scripts within a markdown (`.org`) report, which can then be exported to LaTeX format. This method of reporting aims to offer the best of both worlds: high quality data analysis scripting (via Python) and high quality reporting (via LaTeX), combined with full traceability for all results presented in reports. However, as Emacs is not very commonly used in industry, an alternative post-processing interface could be included with InWave-HOTINT in future releases.

7.1.1 Potential Future Reseach Directions

- After Morison drag loads have been included in the code, improved mooring modelling can be provided: using HOTINT features to create lumped mass or

finite element cable models. Significant verification and validation work for these models may be required.

- Recent work with InWave has focused on the coupling with an aerodynamic solver for modelling offshore wind turbines. A similar approach could also be included in InWave-HOTINT to expand its commercial range.

With InWave-HOTINT, it has been demonstrated that complex WECs comprised of rigid bodies can now be modelled. Although further work is still required it could be argued that there is now good coverage in the WEC industry for a wide range of different working principles for machines comprised of rigid bodies. However, WECs featuring deformable bodies have significant potential and should not be overlooked by current WEC modelling tools. Future work on InWave-HOTINT could investigate modelling this WEC type, potentially utilizing the finite element solver already included in HOTINT.

Bibliography

Albatern Ltd. (2017). *WaveNet Series 12 - WES Novel Wave Energy Converter Stage 1 Project - Public Report*. Tech. rep. Wave Energy Scotland.

Anderlini, E. (2017). "Control of wave energy converters using machine learning strategies". PhD thesis. University of Edinburgh.

Ascher, U. M. et al. (1995). "Stabilization of Constrained Mechanical Systems with DAEs and Invariant Manifolds". In: *Mechanics of Structures and Machines* 23.2, pp. 135–157.

Babarit, A. (2017a). *NEMOH-Mesh*. URL: `https://lheea.ec-nantes.fr/logiciels-et-brevets/nemoh-mesh-192932.kjsp?RH=1489593406974` (visited on 20/11/2018).

– (2017b). *NEMOH-Toolbox*. URL: `https://lheea.ec-nantes.fr/logiciels-et-brevets/nemoh-toolbox-192933.kjsp?RH=1489859622433` (visited on 20/11/2018).

Babarit, A., J. Hals, et al. (2011). *The NumWEC Project*. Tech. rep.

Babarit, A., J. Singh, et al. (2017). "A linear numerical model for analysing the hydroelastic response of a flexible electroactive wave energy converter". In: *Journal of Fluids and Structures* 74, pp. 356–384.

Baraff, D. (1996). "Linear-Time Dynamics using Lagrange Multipliers". In: *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 96)*. New Orleans, USA, pp. 137–146.

Barker-Ewart, L. (2017). "Optimising structural loading and power production for floating wave energy converters". In: *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*. Cork, Ireland.

Barzel, R. and A. H. Barr (1988). "A modeling system based on dynamic constraints". In: *ACM SIGGRAPH Computer Graphics*. Vol. 22. 4. ACM, pp. 179–188.

Bauchau, O. A. and A. Laulusa (2007). "Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems". In: *Journal of Computational and Nonlinear Dynamics* 3.1.

Baumgarte, J. (1972). "Stabilization of constraints and integrals of motion in dynamical systems". In: *Computer methods in applied mechanics and engineering* 1.1.

Beatty, S. J. et al. (2015). "Experimental and numerical comparisons of self-reacting point absorber wave energy converters in regular waves". In: *Ocean Engineering* 104, pp. 370–386.

Bivol, I. et al. (2017). "PLAT-O#2 at FloWave: A tank-scale validation of ProteusDS dynamic analysis tool for floating tidal". In: *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*. Cork, Ireland.

BP (2018). *BP Statistical Review of World Energy*. Tech. rep. London, UK: BP plc.

Cândido, J. J. and P. A. P. S. Justino (2011). "Modelling, control and Pontryagin Maximum Principle for a two-body wave energy device". In: *Renewable Energy* 36.5, pp. 1545–1557.

Carbon Trust (2012). *UK Wave Energy Resource*. Tech. rep.

Clément, A. H. (1997). "Dynamic Nonlinear Response of OWC Wave Energy Devices". In: *International Journal of Offshore and Polar Engineering* 7.2.

Combourieu, A., M. Lawson, et al. (2015). "WEC3 : Wave Energy Converter Code Comparison Project". In: *Proceedings of the 11th European Wave and Tidal Conference (EWTEC)*. Nantes, France.

Combourieu, A., M. Philippe, A. Larivain, et al. (2015). "Experimental Validation of InWave, a Numerical Design Tool for WECs". In: *Proceedings of the 11th European Wave and Tidal Conference (EWTEC)*. Nantes, France.

Combourieu, A., M. Philippe, F. Rongère, et al. (2014). "InWave : A New Flexible Design Tool Dedicated to Wave Energy Converters". In: *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*. San Francisco, USA.

Corrigan, A. (2009). "Kernel-Based Meshless Methods". PhD Thesis. George Mason University.

Corrigan, A. et al. (2011). "Running unstructured grid-based CFD solvers on modern graphics hardware". In: *International Journal for Numerical Methods in Fluids* 66.2, pp. 221–229.

Count, B. M. (1978). "On the dynamics of wave-power converters". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 363.1715, pp. 559–579.

Crozier, R. and M. Mueller (2017). "Development of a Multi-Rate Wave-to-Wire Modelling Tool". In: *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*. Cork, Ireland.

Cruz, J., ed. (2008). *Ocean Wave Energy*. Green Energy and Technology (Virtual Series). Springer Berlin Heidelberg.

Cruz, J., M. Livingstone, and K. Rhinefrank (2011). "Validation of a New Wave Energy Converter (WEC) Modelling Tool: Application to the Columbia Power WEC". In: *Proceedings of the ASME 30th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*, pp. 679–699.

Cruz, J., E. Mackay, et al. (2013). "Validation of Design and Planning Tools for Wave Energy Converters (WECs)". In: *1st Marine Energy Technology Symposium (METS)*. Washington, D. C.

Cruz, J. and S. H. Salter (2006). "Numerical and experimental modelling of a modified version of the Edinburgh Duck wave energy device". In: *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment* 220.3, pp. 129–147.

Cummins, W. E. (1962). *The Impulse Response Function and Ship Motions*. Tech. rep. David Taylor Model Basin. Hydrodynamics Laboratory, USA.

Davidson, J. and J. V. Ringwood (2017). "Mathematical Modelling of Mooring Systems for Wave Energy Converters—A Review". In: *Energies* 10.5.

Day, A. H. et al. (2015). "Hydrodynamic modelling of marine renewable energy devices: A state of the art review". In: *Ocean Engineering* 108, pp. 46–69.

De Andrés, A. D. et al. (2013). "Time domain model for a two-body heave converter: Model and applications". In: *Ocean Engineering* 72, pp. 116–123.

Denavit, J. and R. S. Hartenberg (1955). "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices". In: *Trans. of the ASME. Journal of Applied Mechanics*. Vol. 22, pp. 215–221.

Department for Business Energy & Industrial Strategy (2017). *Digest of UK energy statistics 2017*. Tech. rep.

Dynamic Systems Analysis Ltd (2018). *ProteusDS*. URL: `https://dsa-ltd.ca/proteusds/overview/` (visited on 02/20/2019).

EMEC (2017). *Wave developers*. URL: `http://www.emec.org.uk/marine-energy/wave-developers/`.

Erleben, K. (2005). "Stable, Robust, and Versatile Multibody Dynamics Animation". PhD Thesis. University of Copenhagen, Denmark.

Evans, D. V. (1976). "A theory for wave power absorption by oscillating bodies". In: *Journal of Fluid Mechanics* 77.

Featherstone, R. (1983). "The Calculation of Robot Dynamics Using Articulated-Body Inertias". In: *The International Journal of Robotics Research* 2.1, pp. 13–30.

– (2008). *Rigid Body Dynamics Algorithms*. Boston, MA: Springer US.

Fitzgerald, J. and L. Bergdahl (2008). "Including moorings in the assessment of a generic offshore wave energy converter: A frequency domain approach". In: *Marine Structures* 21.1, pp. 23–46.

Flavia, F. F. (2017). "A Numerical Tool for the Frequency Domain Simulation of Large Clusters of Wave Energy Converters". PhD Thesis. Ecole Centrale de Nantes.

Flores, P. (2015). *Concepts and Formulations for Spatial Multibody Dynamics*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing.

Folley, M., ed. (2016). *Numerical Modelling of Wave Energy Converters: State-of-the-Art Techniques for Single Devices and Arrays*. Elsevier.

Folley, M. et al. (2012). "A Review of Numerical Modelling of Wave Energy Converter Arrays". In: *Proceedings of the ASME 31st International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. Rio de Janeiro, Brazil.

Forehand, D.I.M. et al. (2016). "A fully coupled wave-to-wire model of an array of wave energy converters". In: *IEEE Transactions on Sustainable Energy* 7.1, pp. 118–128.

Frey, Pascal Jean and Paul-Louis George (2008). *Mesh Generation*. 2nd ed. London, UK: ISTE.

Führer, C. and B. J. Leimkuhler (1991). "Numerical solution of differential-algebraic equations for constrained mechanical motion". In: *Numerische Mathematik* 59.1, pp. 55–69.

García de Jalón, J. and E. Bayo (1994). *Kinematic and Dynamic Simulation of Multibody Systems*. Mechanical Engineering Series. New York, NY: Springer New York.

Garcia-Tereul, A. and D.I.M. Forehand (2018). "Optimal wave energy converter geometry for different modes of motion". In: *Proceedings of the 3rd International Conference on Renewable Energies Offshore (Renew 2018)*. Lisbon, Portugal, pp. 299–305.

Gerstmayr, J. (2009). "Hotint – A C++ Environment for the Simulation of Multibody Dynamics Systems and Finite Elements". In: *Multibody Dynamics 2009, ECCOMAS Thematic Conference*. Warsaw, Poland.

Gerstmayr, J., L. Aigner, et al. (2013). *Hotint Documentation*. Tech. rep. Linz, Austria: Linz Center of Mechatronics GmbH.

Gerstmayr, J., A. Dorninger, et al. (2013). "HOTINT: A Script Language Based Framework for the Simulation of Multibody Dynamics Systems". In: *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Gerstmayr, J. and M. Stangl (2004). "High-order implicit Runge–Kutta methods for discontinuous mechatronical systems". In: *Proceedings of the 32nd International Summer School Conference - Advanced Problems in Mechanics (APM)*.

Gunn, K. and C. Stock-Williams (2012). "Quantifying the global wave power resource". In: *Renewable Energy* 44, pp. 296–304.

Hall, M., B. Buckham, et al. (2011). "The importance of mooring line model fidelity in floating wind turbine simulations". In: *OCEANS'11 MTS/IEEE KONA*. Waikoloa, HI, USA.

Hall, M. and A. Goupee (2015). "Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data". In: *Ocean Engineering* 104, pp. 590–603.

Henry, A. et al. (2014). "The Characteristics of Wave Impacts on an Oscillating Wave Surge Converter". In: *Journal of Ocean and Wind Energy* 1.2, pp. 101–110.

Hosseini, M.K. et al. (2006). "A composite rigid body algorithm for modeling and simulation of an underwater vehicle equipped with manipulator arms". In: *Journal of offshore mechanics and Arctic engineering* 128.2, pp. 119–132.

Hurst, T. (2015). "Wave Energy Scotland". In: *All Energy*. Glasgow.

IPCC (2018). *Global Warming of 1.5 °C - Summary for Policymakers*. Tech. rep. Incheon, Republic of Korea: Intergovernmental Panel on Climate Change (IPCC).

Jasak, H., A. Jemcov, and Z. Tukovic (2007). "OpenFOAM : A C ++ Library for Complex Physics Simulations". In: *International Workshop on Coupled Methods in Numerical Dynamics*. Dubrovnik, Croatia.

Jefferys, E. R. (1984). "Simulation of wave power devices". In: *Applied Ocean Research* 6.1, pp. 31–39.

Johanning, L., G. H. Smith, and J. Wolfram (2006). "Mooring design approach for wave energy converters". In: *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 220.4, pp. 159–174.

– (2007). "Measurements of static and dynamic mooring line damping and their importance for floating WEC devices". In: *Ocean Engineering* 34.14-15, pp. 1918–1934.

Josset, C., A. Babarit, and A. H. Clément (2007). "A wave-to-wire model of the SEAREV wave energy converter". In: *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment* 221.2, pp. 81–93.

Kotik, J. and V. Mangulis (1962). "On the Kramers-Kronig Relations for Ship Motions". In: *International Shipbuilding Progress* 9.97, pp. 361–368.

Kurniawan, A., J. Hals, and T. Moan (2011). "Assessment of Time-Domain Models of Wave Energy Conversion Systems". In: *Proceedings of the 9th European Wave and Tidal Energy Conference (EWTEC)*. Southampton, UK.

Lagrange, J. L. (1811). *Analytical Mechanics*. Ed. by A. Boissonnade and V. N. Vagliente. Springer Netherlands.

Lamont-Kane, P. et al. (2015). "Extreme Loading on an Oscillating Wave Surge Converter". In: *Proceedings of the 11th European Wave and Tidal Energy Conference (EWTEC)*. Nantes, France.

Laulusa, A. and O. A. Bauchau (2008). "Review of Classical Approaches for Constraint Enforcement in Multibody Systems". In: *Journal of Computational and Nonlinear Dynamics* 3.1.

Lawson, M. et al. (2015). "Improving and Validating the WEC-Sim Wave Energy Converter Modeling Code". In: *Proceedings of the 3rd Marine Energy Technology Symposium (METS)*. Washington DC, USA.

Le Méhauté, B. (1976). *An Introduction to Hydrodynamics and Water Waves*. 1st ed. Springer-Verlag Berlin Heidelberg.

Lee, C. -H. (1988). "Numerical Methods for Boundary Integral Equations in Wave Body Interactions". PhD thesis. Massachusetts Institute of Technology.

– (1995). *WAMIT Theory Manual*. Tech. rep. Cambridge, Massachusetts: Massachusettes Institute of Technology.

Lee, C. -H. and J. N. Newman (2000). "An Assessment of Hydroelasticity for Very Large Hinged Vessels". In: *Journal of Fluids and Structures* 14, pp. 957–970.

– (2005). "Computation of wave effects using the panel method". In: *Numerical Models in Fluid-Structure Interaction*. Vol. 18. WIT Press, pp. 211–251.

Leroy, V., A. Combourieu, et al. (2014). "Benchmarking of the new design tool InWave on a selection of wave energy converters from NumWEC project". In: *2nd Asian Wave and Tidal Energy Conference (AWTEC)*. Tokyo, Japan.

Leroy, V., J. C. Gilloteaux, et al. (2017). "Development of a Simulation Tool Coupling Hydrodynamics and Unsteady Aerodynamics to Study Floating Wind Turbines". In: *Volume 10: Ocean Renewable Energy*. Trondheim, Norway.

Liegeois, A. (1977). "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.12, pp. 868–871.

Liu, Y. and J. M. Falzarano (2017). "Irregular frequency removal methods: theory and applications in hydrodynamics". In: *Marine Systems & Ocean Technology* 12.2, pp. 49–64.

Lucas, J. et al. (2012). "Development of a wave energy converter (WEC) design tool–application to the WaveRoller WEC including validation of numerical estimates". In: *Proceedings of the 4th International Conference on Ocean Energy (ICOE)*. Dublin, Ireland.

MacKay, D. J. C. (2009). *Sustainable Energy - without the hot air*. UIT Cambridge Ltd.

Mackay, E. et al. (2012). "Validation of a new wave energy converter design tool with large scale single machine experiments". In: *Proceedings of the 1st Asian Wave and Tidal Conference Series (AWTEC)*. Jeju, Korea.

Masarati, P., M. Lanz, and P. Mantegazza (2001). "Multistep Integration of Ordinary, Stiff and Differential-Algebraic Problems for Multibody Dynamics Applications". In: *XVI Congresso Nazionale AIDAA*. Palermo, Italia.

Masarati, P., M. Morandini, and P. Mantegazza (2014). "An Efficient Formulation for General-Purpose Multibody/Multiphysics Analysis". In: *Journal of Computational and Nonlinear Dynamics* 9.4.

McCabe, A. P. (2004). *An Appraisal of a Range of Fluid Modelling Software*. Tech. rep. Department of Engineering, Lancaster University.

McDonald, A. et al. (2017). "Experimental Investigation of Array Effects for a Mechanically Coupled WEC Array". In: *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*. Cork, Ireland.

McMillan, S., D. E. Orin, and R. B. McGhee (1995a). "DynaMechs: An Object Oriented Software Package for Efficient Dynamic Simulation of Underwater Robotic Vehicles". In: *Underwater Robotic Vehicles: Design and Control*. Ed. by J. Yuh, pp. 73–98.

– (1995b). "Efficient dynamic simulation of an underwater vehicle with a robotic manipulator". In: *IEEE Transactions on Systems, Man, and Cybernetics* 25.8.

McNatt, J. C. (2015). "Cylindrical Linear Water Waves and their Application to the Wave-body Problem". PhD thesis. The University of Edinburgh.

Mei, C.C., M. Stiassnie, and D.K.P. Yue (2005). *Theory and Applications of Ocean Surface Waves: Linear aspects*. Advanced series on ocean engineering. World Scientific.

Meir, R. (1978). "The Development of the Oscillating Water Column". In: *Proceedings of the Wave Energy Conference*. Ed. by P. Quarrell. London-Heathrow, UK, pp. 35–44.

Mørk, G. et al. (2010). "Assessing the Global Wave Energy Potential". In: *Proceedings of the ASME 29th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*, pp. 447–454.

MSC (2019). *Adams: The Multibody Dynamics Software*. URL: `https://www.mscsoftware.com/product/adams` (visited on 06/03/2019).

Mukherjee, R. M. and K. S. Anderson (2007). "A Logarithmic Complexity Divide-and-Conquer Algorithm for Multi-flexible Articulated Body Dynamics". In: *Journal of Computational and Nonlinear Dynamics* 2.1.

Mynett, A. E., D. D. Serman, and C. C. Mei (1979). "Characteristics of Salter's cam for extracting energy from ocean waves". In: *Applied Ocean Research* 1.1, pp. 13–20.

Nambiar, A. J. et al. (2015). "Effects of hydrodynamic interactions and control within a point absorber array on electrical output". In: *International Journal of Marine Energy* 9, pp. 20–40.

Negrut, D. (1998). "On the implicit integration of differential-algebraic equations of multibody dynamics". PhD thesis. University of Iowa.

Negrut, D., E. J. Haug, and H. C. German (2003). "An Implicit Runge – Kutta Method for Integration of Differential Algebraic Equations of Multibody Dynamics". In: *Multibody System Dynamics* 9, pp. 121–142.

Negrut, D., R. Rampalli, et al. (2007). "On an Implementation of the Hilber-Hughes-Taylor Method in the Context of Index 3 Differential-Algebraic Equations of Multibody Dynamics". In: *Journal of Computational and Nonlinear Dynamics* 2.1.

Newman, J. N. (1977). *Marine Hydrodynamics*. Cambridge, Massachusetts: The MIT Press, p. 402.

– (1985). "Algorithms for the free-surface Green function". In: *Journal of Engineering Mathematics* 19, pp. 57–67.

– (2001). "Wave Effects on Multiple Bodies". In: *Hydrodynamics in Ship and Ocean Engineering*. Ed. by M. Kashiwagi. Kyushu University, Japan, pp. 3–26.

Newman, J. N. and C. -H. Lee (1992). "Sensitivity of wave loads to the discretization of bodies". In: *6th International Conference on the Behaviour of Offshore Structures*. London, UK.

Nicoll, R. S., C. F. Wood, and A. R. Roy (2012). "Comparison of Physical Model Tests With a Time Domain Simulation Model of a Wave Energy Converter". In: *Proceedings of the ASME 31st International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. Rio de Janeiro, Brazil.

Nikravesh, P. E. (2007). "Initial condition correction in multibody dynamics". In: *Multibody System Dynamics* 18.1, pp. 107–115.

NREL and Sandia (2015). *WEC-Sim*. URL: `http://wec-sim.github.io/WEC-Sim/` (visited on 21/02/2019).

Ó'Catháin, M., B. J. Leira, and J. V. Ringwood (2007). "Modelling of Multibody Marine Systems with Application to Wave-Energy Devices". In: *Proceedings of the 7th European Wave and Tidal Energy Conference (EWTEC)*. Porto, Portugal.

Ó'Catháin, M., B. J. Leira, J. V. Ringwood, and J. C. Gilloteaux (2008). "A modelling methodology for multi-body systems with application to wave-energy devices". In: *Ocean Engineering* 35.13, pp. 1381–1387.

Ohkusu, M. (1969). "On the Heaving Motion of Two Circular Cylinders on the Surface of a Fluid". In: *Reports of Research Institute for Applied Mechanics* 17.58, pp. 167–185.

Paparella, F. (2017). "Modeling and Control of a Multibody Hinge-Barge Wave Energy Converter". PhD thesis. Maynooth University.

Paparella, F. et al. (2015). "On the solution of multi-body wave energy converter motions using pseudo-spectral methods". In: *Proceedings of the 11th European Wave and Tidal Energy Conference (EWTEC)*. Nantes, France.

Pastor, J. and Y. Liu (2014). "Power Absorption Modeling and Optimization of a Point Absorbing Wave Energy Converter Using Numerical Method". In: *Journal of Energy Resources Technology* 136.2.

Payne, G. S. et al. (2008). "Assessment of boundary-element method for modelling a free-floating sloped wave energy device. Part 2: Experimental validation". In: *Ocean Engineering* 35.3, pp. 342–357.

Pecher, A. and J. P. Kofoed, eds. (2017). *Handbook of Ocean Wave Energy*. Vol. 7. Ocean Engineering & Oceanography. Springer International Publishing.

Penalba, M., T. Kelly, and J. V. Ringwood (2017). "Using NEMOH for modelling wave energy converters: A comparative study with WAMIT". In: *Proceedings of the 12th European Wave and Tidal Energy Conference (EWTEC)*. Cork, Ireland.

Peyret, R., ed. (1996). *Handbook of Computational Fluid Mechanics*. Elsevier Ltd.

Pizer, D. (1994). *Numerical Modelling of Wave Energy Absorbers*. Tech. rep. Harwell Laboratory, Energy Technology Support Unit.

Quoceant Ltd. (2016a). *Description & Critique of PELs Suite*. Tech. rep. Edinburgh: Wave Energy Scotland.

– (2016b). *Modelling R&D Recommendations*. Tech. rep. Edinburgh: Wave Energy Scotland.

Retzler, C. H. (2015a). *Critique of potential future work*. Tech. rep. Edinburgh: Wave Energy Scotland.

– (2015b). *Using Pelamis Modelling Tools to Advance the Industry State-of-the-Art*. Tech. rep. Wave Energy Scotland.

Rongère, F. and A. H. Clément (2013). "Systematic Dynamic Modeling and Simulation of Multibody Offshore Structures: Application to Wave Energy Converters". In: *Proceedings of the ASME 32nd International Conference on Ocean, Offshore and Artic Engineering (OMAE)*. Nantes, France.

Rosenberg, R. M. (1977). *Analytical Dynamics of Discrete Systems*. Vol. 24. 9. Boston, MA: Springer US, pp. 2568–2577.

Ruehl, K., C. Michelen, B. Bosma, et al. (2016). "WEC-Sim Phase 1 Validation Testing: Numerical Modeling of Experiments". In: *Proceedings of the ASME 35th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. Busan, South Korea.

Ruehl, K., C. Michelen, S. Kanner, et al. (2014). "Preliminary verification and validation of WEC-Sim, an open-source wave energy converter design tool". In: *Proceedings of the ASME 33rd International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. San Francisco, USA.

Salter, S. H. (1974). "Wave power". In: *Nature* 249, pp. 720–724.

Schiehlen, W., ed. (1990). *Multibody Systems Handbook*. Berlin, Heidelberg: Springer Berlin Heidelberg.

– (1997). "Multibody System Dynamics: Roots and Perspectives". In: *Multibody System Dynamics* 1.2, pp. 149–188.

Schmitt, A., J. S. Bender, and H. Prautzsch (2005). *Impulse-Based Dynamic Simulation of Higher Order and Numerical Results*. Tech. rep. Fakultät für Informatik, Universität Karlsruhe.

Schörgenhumer, M. (2012). "Fluid-Structure Interaction – Coupling of flexible multibody dynamics with particle-based fluid mechanics". Master's Thesis. Linz Center of Mechatronics.

Shabana, A. A. (2013). *Dynamics of Multibody Systems*. 4th ed. New York, NY: Cambridge University Press.

Srokosz, M. A. and D. V. Evans (1979). "A theory for wave-power absorption by two independently oscillating bodies". In: *Journal of Fluid Mechanics* 90.2, pp. 337–362.

Strauch, D. (2009). *Classical Mechanics: An Introduction*. 1st ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Sustainable Development Commission (2006). *Paper 2: Reducing CO2 emissions - nuclear and the alternatives*. Tech. rep.

Taghipour, R. et al. (2008). "Structural Analysis of a Multibody Wave Energy Converter in the Frequency Domain by Interfacing WAMIT and AQAQUS". In: *Proceedings of the ASME 27th International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*. Estoril, Portugal.

Tasora, A. et al. (2015). "High Performance Computing in Science and Engineering". In: *High Performance Computing in Science and Engineering (HPCSE)*. Vol. 9611. Lecture Notes in Computer Science. Garching/Munich: Springer, Cham, pp. 19–49.

Tom, N., M. Lawson, and Y. Yu (2015). "Demonstration of the Recent Additions in Modeling Capabilities for the WEC-Sim Wave Energy Converter Design Tool". In: *Proceedings of the ASME 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. St. John's, Newfoundland, Canada.

Tom, N., M. Lawson, Y. Yu, and A. Wright (2016). "Development of a nearshore oscillating surge wave energy converter with variable geometry". In: *Renewable Energy* 96, pp. 410–424.

Topper, M. B. R. and D. M. Ingram (2011). "Identifying the Frontier of Knowledge for Marine Renewable Energy Research". In: *Proceedings of the Ninth European Wave and Tidal Energy Conference (EWTEC)*. Southampton, UK.

United Nations FCCC (2015). "Paris Agreement". In: *21st Conference of the Parties*. Paris: United Nations.

Vaughan, A. (2017). *UK wave power far too costly, warns energy research body*. URL: `https://www.theguardian.com/environment/2017/jan/16/uk-wave-power-far-too-costly-warns-energy-research-body`.

Waters, L. (2018). *Energy Consumption in the UK*. Tech. rep. Department for Business, Energy & Industrial Strategy.

Wendt, F. et al. (2017). "OES Task 10 WEC Modelling Verification and Validation". In: *Proceedings of the Twelfth European Wave and Tidal Energy Conference*. Cork, Ireland.

Witkin, A., M. Gleicher, and W. Welch (1990). "Interactive Dynamics". In: *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics (I3D)*. Snowbird, Utah, USA, pp. 11–21.

Wood, G. D. (2011). *Simulation of Constrained Systems*. United States Patent No. US 7,904,280 B2.

Wood, G. D. and D. C. Kennedy (2003). *Simulating Mechanical Systems in Simulink with SimMechanics*. Tech. rep. The Mathworks.

Wuillaume, P.-Y. et al. (2018). "Comparison Between Experiments and a Multibody Weakly Nonlinear Potential Flow Approach for Modeling of Marine Operations". In: *Proceeding of the ASME 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. Madrid, Spain.

Yu, Y. et al. (2014). "Design and Analysis for a Floating Oscillating Surge Wave Energy Converter". In: *Proceedings of the ASME 33rd International Conference on Ocean, Offshore and Arctic Engineering (OMAE)*. San Francisco, USA.