NTNU

Norwegian University of
Science and Technology

# Open Source Simulation for Virtual Maritime Operations

Sergi Escamilla i Miquel

Master in Ship Design

Submission date: June 2019

Supervisors: Henrique M. Gaspar
Icaro A. Fonseca

Norwegian University of Science and Technology

Department of Ocean Operations and Civil Engineering

# Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. **Failure to complete the statement does not excuse students from their responsibility.**

| | | |
|---|---|---|
| *Please complete the mandatory statement by placing a mark **in each box** for statements 1-6 below.* | | |
| **1.** | **I/we hereby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than is mentioned in the paper/assignment.** | ☒ |
| **2.** | **I/we herby declare that this paper**<br>1. Has not been used in any other exam at another department/university/university college<br>2. Is not referring to the work of others without acknowledgement<br>3. Is not referring to my/our previous work without acknowledgement<br>4. Has acknowledged all sources of literature in the text and in the list of references<br>5. Is not a copy, duplicate or transcript of other work | Mark each box:<br>1. ☒<br>2. ☒<br>3. ☒<br>4. ☒<br>5. ☒ |
| **3.** | **I am/we are aware that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations at NTNU.** | ☒ |
| **4.** | **I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check.** | ☒ |
| **5.** | **I am/we are aware that The Norwegian University of Science and Technology (NTNU) will handle all cases of suspected cheating according to prevailing guidelines.** | ☒ |
| **6.** | **I/we are aware of the University's rules and regulations for using sources.** | ☒ |

# Publication agreement

**ECTS credits: 30**

**Supervisor: Henrique M. Gaspar, NTNU**

**Co-supervisor: Icaro A. Fonseca, NTNU**

## Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).

All thesis fulfilling the requirements will be registered and published in Brage with the approval of the author(s).

**I/we hereby give NTNU the right to, free of**

**charge, make the thesis available for electronic publication:** ☒**yes** □**no**

**Is there an agreement of confidentiality?** □**yes** ☒**no**

(A supplementary confidentiality agreement must be filled in)

**Date: 11/06/2019**

Scientific Theory and Methods: MSc Project Proposal

NTNU – Ålesund
Norwegian University of
Science and Technology

**Date. 04/12/2018**

**Department of Ocean Operations and Civil Engineering**

# Open Source Simulation for Virtual Maritime Operations

## Introduction

Marine operations are rapidly coming in demand over de past years, as more and more the ocean is used to place what takes crucial space from land. In order to successfully carry or improve a current maritime operation within the estimated timeframe and costs, simulations are commonly run by designers and personnel before moving to a real domain, allowing the user to estimate what can go wrong for the different variables and contexts. This offers the possibility to create a wide variety of marine simulations within different sub-fields and goals considering various external disturbances given by the simulation requirements that can approximate the simulation to a real-life model.

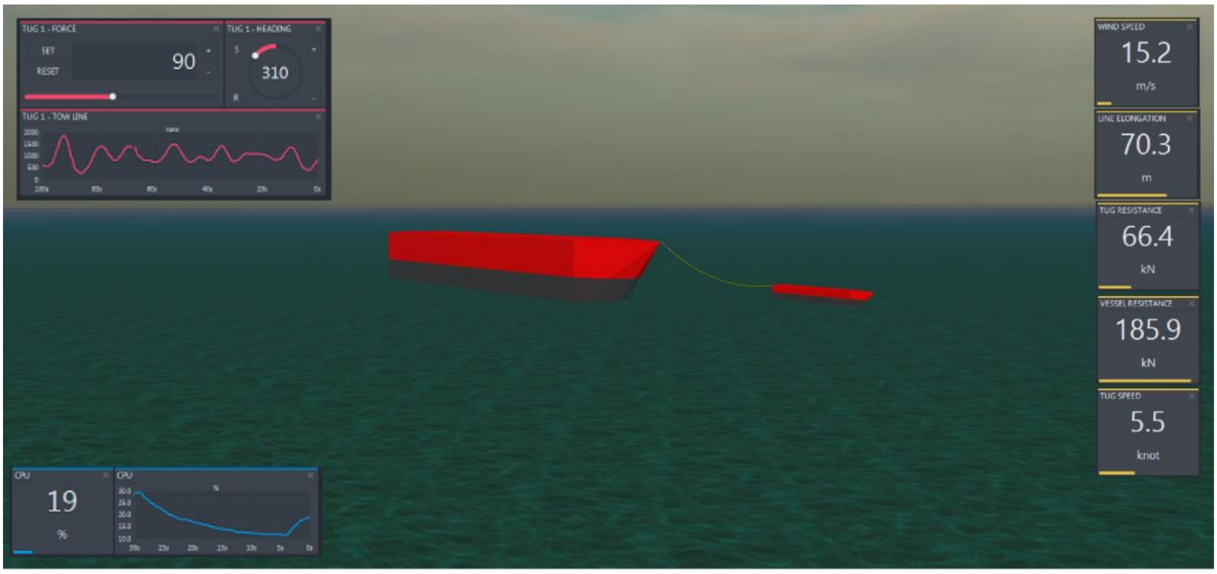


Figure 1 – Simulation of a towing operation with real-time data.

## Motivation

Open source simulation through a web-based design opens many windows for reinventing what nowadays is known as proprietary simulation software. This study offers the possibility to expand this technology to a wide range of potential users that do not know about the advantages of open source simulation by convincing them that reliable operations can be reproduced in a web-based design environment, which is widely compatible with everyone's computers and can be improved and modified by any developer who wants to contribute in the process.

## Scope

The scope of the project is to reuse and develop new objects in an existing open-source environment with structures and methods for vessel and marine operation simulation through a web-based platform by using JavaScript, an object-oriented programming language. The study will use these tools to build up and develop the current conceptual ship design library by modelling interaction between single bodies and creating new multibody models able to output different responses according to a range of various inputs, considering the different requirements and limitations for maritime operations.

## Objective / Research questions

The objective of this study is to develop simulations of multibody interaction in the maritime operation domain by using a taxonomy for simulation of maritime operations in product collaborative design. The research questions are:

- Is there technology for web-based simulation that will allow different maritime operations such as multibody interaction with applied disturbances?
- How to take the most advantage of the Vessel.js library in order to reuse blocks and simulate multibody operations?
- How can this technology be extended in order to be useful and have considerable benefits for the maritime industry?

## Methodology

By using a taxonomy for maritime operations simulations, different operations are chosen to simulate in an open source environment. In order to create case studies, these simulations are decomposed in single elements in order to understand their behaviour separately before making them interact with other elements to create a multibody. In the process, different disturbances are applied to the bodies according to the needs of the simulations and the level of complexity. The simulations are coded in JavaScript and visualized in a web environment in order to assess them. A discussion for the efficiency of the method and the utility of these simulations is carried.

## Milestones:

1. Identification of operations to develop
2. Literature review on theoretical background on operations, virtual prototyping and Vessel.js library.
3. Methodology
   3.1. Define entity models to simulate.
   3.2. Define state models along with the conditions to simulate the operations with.
   3.3. Define process models from the previous models.
   3.4. Establish the connection between models in order to create multibody.
4. Develop case studies of time domain simulations for maritime operations.
5. Analyse and discuss the results.
6. Write the report.
7. Read the report and evaluate.

## Schedule

| Task | Predecessors | 15 Jan | 1 Feb | 15 Feb | 1 Mar | 15 Mar | 1 Apr | 15 Apr | 1 May | 15 May | 1 Jun |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | | ■ | | | | | | | | | |
| 2. | | ■ | ■ | ■ | ■ | | | | | | |
| 3.1 | 2. | | ■ | | | | | | | | |
| 3.2 | 3.1 | | | ■ | | | | | | | |
| 3.3 | 3.2 | | | | ■ | ■ | | | | | |
| 3.4 | 3.3 | | | | | | ■ | ■ | | | |
| 4. | 3.1 | | | | | | | ■ | | | |
| 5 | 4. | | | | | | | | ■ | ■ | |
| 6 | 1. | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| 7 | 6. | | | | | | | | | | ■ |

The work scope may prove to be different than initially anticipated. Subject to approval from the supervisor, topics may be added or deleted from the list above or reduce in extent.

The thesis shall be written as a research report, following the template given in Inspera. During preparation of the text, the candidate should make efforts to create a well-arranged and well-written report. To ease the evaluation of the thesis, it is important to cross-reference text, tables and figures. For evaluation of the work a thorough discussion of results is needed. Discussion of research method, validation and generalization of results is also appreciated.

The thesis shall be submitted in electronic version according to standard procedures (.PDF or .ZIP files). Instructions are found on the NTNU website (Inspera) and on Blackboard. In addition to the specified tasks, an A3 poster should be prepared and delivered together with this proposal, and a conference paper will be handled at the end of the research.

After finalizing and delivering the thesis, it must be sent a copy to the supervisor(s).

**Deliveries:**

Preliminary Thesis (31st March)
Final Thesis + Article (11th June)

Sergi Escamilla Miquel

Sergi Escamilla i Miquel
Student – MSc Ship Design
sergie@stud.ntnu.no

Henrique Murilo Gaspar
Supervisor – IHB
henrique.gaspar@ntnu.no

Ícaro Aragão Fonseca
Co-supervisor - PhD Student
icaroa@stud.ntnu.no

# PREFACE

This thesis is the closure of my Master of Science degree in Ship Design imparted by the Department of Ocean Operations and Civil Engineering at the Norwegian University of Science and Technology (NTNU). The work deals with some topics learnt during my master's degree as well as knowledge acquired during a collaboration with researchers from the Numerical Offshore Tank (TPN) at the University of São Paulo (USP). The thesis is also directly connected to the Ship Design and Operation Lab at NTNU in Ålesund with additional support provided by the EDIS Project, in cooperation with the Research Council of Norway and Ulstein International AS.

Ålesund, June 2019

Sergi Escamilla Miquel

Sergi Escamilla Miquel

## ACKNOWLEDGMENTS

Various people have contributed to this thesis, either academically or morally. Therefore, I would like to acknowledge the time, knowledge and funding provided by my head supervisor Henrique Gaspar, as well as for all the dedication that my co-supervisor Icaro Fonseca has invested in me and this work.

I would also like to thank the post-doc researcher Daniel P. Vieira and all the people in the University of São Paulo who took the time to teach me some of the methods used in this work and provided me with useful material in order to develop a good thesis.

Also, a huge gratitude to all the collaborators in the Vessel.js library for developing the tools that allowed me to reuse and build up my thesis, and also for the ones who spend some of their time helping me with my simulations as well as providing me with some insight and code.

Finally, I would like to thank my family and friends for all the support and patience given during the months that I have been working on this thesis. They have been my most important pillar for moral support.

All my gratitude,

Sergi Escamilla Miquel

# SUMMARY

The maritime industry is rapidly growing over time, rocketing the number operations carried over seas every day. Simulations are nowadays carried before bringing one of these operations to the real domain, and so a great window is open for developing new open source simulators dealing with maritime operations as an alternative to the current proprietary software. This last faces a challenge regarding accessibility, compatibility and the ability to share results, which is quickly solved in a web browser simulator, that is open to everyone for interaction, modification and improvement.

This thesis contributes into developing multibody operations that can operate in a simple web browser by using methods embedded in a current conceptual ship design library, at the same time that new tools are developed in order to increase the potential of this library. To do this, a method for categorizing the elements composing the simulation is followed, along with a new approach to calculate the equations of motions in respect to the current methods used in the library mentioned. This new method allows to obtain more accuracy when dealing with more than one floating body, for instance in a side-by-side offloading operation.

The method is applied to different cases with a progression that allows the user to keep track of the addition of complexity. A simple case is simulated and new bodies with new behaviours are added to the simulation until a maritime operation is configured. Finally, a discussion looks at the different cases and evaluates the impact of the simulations over the real purpose of the thesis. It also describes a way on how the readers can reuse the work in this thesis to make their own maritime and subsea operations.

# Table of contents

# List of figures

# List of tables

# Chapter 1

## 1 INTRODUCTION

The fact that seventy percent of the Earth planet is nowadays covered by water, makes the ocean space the perfect candidate to be the motor of the future global economy by providing a great potential to overtake some of the challenges that this planet is facing today, as it supplies food, minerals, energy, transportation, leisure and a wide range of other options desired by society (See Figure 1.1).

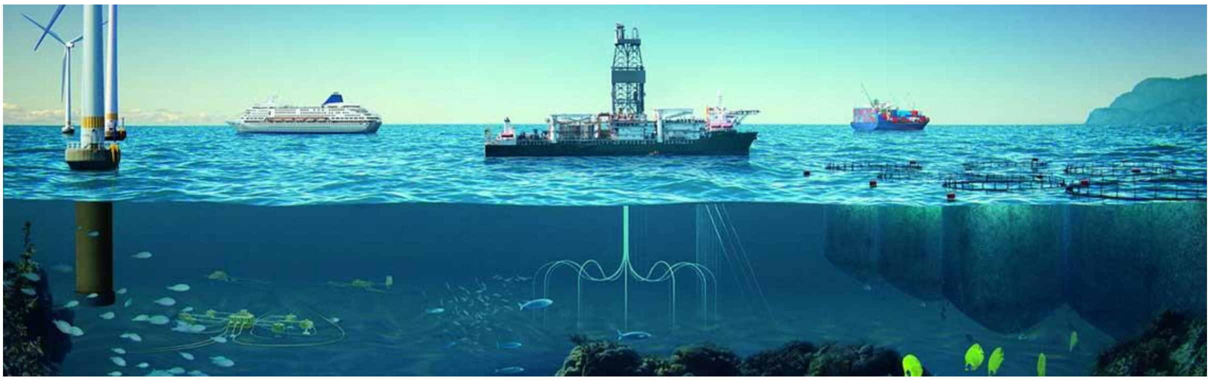


Figure 1.1. Ocean space displaying maritime activity (DNV-GL 2018)

Thanks to this opportunity, maritime operations are rapidly coming in demand while making this sector to lead the way in exploiting as much as possible this mother nature’s gift at the cost of constantly innovating, as well as optimizing the current operations happening every day in our seas, efficiency and cost wise. This means that the potential for development of new marine operations seeking for ocean space opportunities keeps growing exponentially and new windows for designers and engineers are appearing.

### 1.1 REAL-TIME MULTIBODY SIMULATION

One way to successfully plan or improve the procedure of a current maritime operation within the estimated timeframe and costs is to run real-time simulations so one can estimate in advance what can go wrong for the different variables and contexts in order to make a good planification and design according to it. These simulations are commonly run by designers and crew before moving to a real domain and are used either to verify calculations and share results or to train crew members that are going board a ship without compromising their integrity.

There is consequently a boom on the demand of users using multibody simulations in order to represent on a virtual environment one of the wide ranges of maritime operations that are rocketing our economies. Reached this point, one realises that these kinds of simulations are the key to represent any imaginable maritime operation by using as many single objects as needed, which combined will form a multibody configuration. Furthermore, if these single objects, from now on called entities, are assigned an activity and this activity is displayed over time, we obtain a proper maritime operation in one or various scenarios where the user is able to decide where to place a boat, the subsea equipment, the risers and the mooring lines for later obtaining a response out of that information.

Currently, this is possible by using tools included in proprietary software, which means that one can access freely to trail periods, but if the user wants to use extra functionalities, then they will be asked to pay, and sometimes the licensing is not affordable for everyone. Additionally, even if licensing is not a problem and the user obtains some results with the aim to share them, a couple hours could go by before the user has copied and pasted different graphs, tables and data in a PowerPoint presentation. This is normally due to the lack of compatibility of this specialized software, that cannot be displayed or run outside its environment. Also, current competitivity among software producers force the user to learn new different software that end up doing the same tasks, leading to a lack of compatibility between them. With this problem, arises the need to introduce the “open” concept in order to reveal what is behind the door, an accessible, standard and easy to use option to simulate and visualize these multibody operations.

Up until some years ago, there were not strong and reliable open source simulator platforms for marine and subsea operations, and this fact had inevitably brought users to use paid software in order to obtain the desired results. As this concept is rather new, as of today, we can say that only a few initiatives taken by researchers, forward-looking businesses, professors and students are starting to come up with ideas such as the library this project is involved with, Vessel.js (http://www.vesseljs.org), as well as other software initiatives to make CFD analysis such as OpenFOAM (https://www.openfoam.com). However, the options are still nowadays limited and therefore a big window is presented for developing these new tools in a way that are attractive to users by giving them features such as reliability, accessibility and ease of use. This need is to be fulfilled at the same time it is open for manipulation and improvement, promoting this way a collaborative community, where the knowledge can be spread and revised by others.

## 1.2 OBJECTIVES

It is the objective of this thesis then, to tackle these problems through open source web-based technology. Working in a web-based environment makes an easier experience for the user, as all the points mentioned previously can be taken care amended through all the advantages that the combination of JavaScript and open source libraries provide. Taking this direction, many simulations can be run on a maritime and subsea domain, while monitoring in detail what happens in every moment to every single entity, for instance, the plot of the tension in real time of a rope that is towing a vessel, the motion of this last one, or even the real time data of how the vessels react to different wave inputs over time.

Another issue that this study addresses is related to the reliability of the data in these kinds of simulations. What usually happens in these early stage libraries is that the data is sometimes not the most accurate, as in this stage of the design phase, the developer tends to prioritize other features such as displaying a good visualization in order to catch the attention of the user instead of focusing too much on the physics. This is why even though the simulations can visually look reliably enough, when someone wants to obtain some accurate data, some simulators cannot be entirely trusted, as they take approximations to describe the physics. This study will focus part of its attention to obtain the data from one of the most reliable tools for analysing wave interactions with vessels and offshore platforms, which is WAMIT.

Finally, the aim is also to take the user to a more understanding graphical user interface (GUI) when dealing with WAMIT output files. These are documents full of text that need some post-processing through one of the traditional MATLAB or Python scripts in order to start seeing some charts about what happens in those results. However, a user lacking programming skills will not be able to make such interface. This is why this study tries to bring the user to a more user-friendly interaction with WAMIT results. Furthermore, this will also allow to validate the physical experiments by obtaining graphs and output results such as displaying the motion in graphs, and other characteristics of interest.

## 1.3 SCOPE

The study intends to reuse and develop new objects in an existing open-source environment (Vessel.js) with structures and methods for vessel and marine operation simulation through a web-based platform by using JavaScript, an object-oriented programming language. These simulations aim to be the first open source simulations considering the real motions of multiple ships interacting with mooring lines and hawsers. The study will use these tools from various libraries to build up and develop the current Vessel.js library by modelling interactions between single entities and creating new multibody models that will be able to output motions obtained from WAMIT runs, considering the different requirements and limitations for maritime operations.

In order to work on the topics described so far and as described in Figure 1.2, this study will comprise two different areas that are commonly identified separately. These will be brought together with the aim to bring insight into the typical hydrodynamic calculations through a more interactive way.

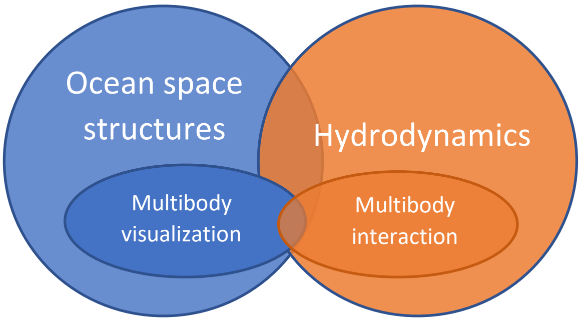


Figure 1.2. Scope of the study as a Venn diagram

On one side there is the vast field of ocean space structures simulation and visualization. This side will deal with the multibody simulation and visualization of these structures belonging to an ocean space from the hand of open source technology, which will address to the current library (Vessel.js) and web-based programming (JavaScript).

Vessel.js will provide maritime-related tools and JavaScript will give a platform where to work from, by developing these maritime operations. The code in the library is meant to be used and reused as much as possible for this study. This side of the scope provides the project with an openness and a collaboration spirit, while for the web-based programming, JavaScript will be used as a universal web language in order to ease the coding requirements. This provides this study with all the functionalities needed for the simulations.

On the other side, there is the long-known field of hydrodynamics. This side will deal with the hydrodynamics of single and multiple floating bodies in order to understand what is behind the simulations carried in WAMIT software. The aim is to study and simulate the interaction between entities and how their response influences other entities. As seen in the response amplitude operations from the cases displayed in Appendix B, the hydrodynamics applied to one vessel is not the same as the hydrodynamics applied to various vessels side to side, due to the so-called shadow effect.

These two areas are rarely studied together, as the open source tag is a limitation. Nowadays, companies have developed their own software or are using paid software to carry maritime simulations, but when it comes to representing WAMIT, simple Phyton scrips give some insufficient data of what happens in the simulations. By combining these two concepts, several limitations will emerge, mainly due to the simplified mathematical background for the first version of these simulations, and also because of the lack of work done previously with these two fields together, which will derive into other limitations, such as constraints following the methods and procedures, but they will be stated and proposed as future work.

## 1.4 STRUCTURE OF THE THESIS

*Chapter 2* presents a literature review for the topics related to the project objectives: an overview of the mathematical background for the objects to simulate, the interaction between them as well as an introduction to the open source concept in engineering simulations along with the use of JavaScript as a web-based programming language.

*Chapter 3* proposes a methodology followed to develop all the simulations along with the limitations, constraints and boundary conditions that the simulations are facing.

*Chapter 4* presents the simulations studied, starting with simple examples and building up to more complex ones.

*Chapter 5* evaluates and discusses the results obtained from the simulations.

*Chapter 6* works on the concluding remarks and the possible future works for the development of these operations.

# Chapter 2

## 2 LITERATURE REVIEW

### 2.1 HYDRODYNAMICS OF MULTIPLE BODIES

Hydrodynamics is a field that has been studied for centuries and is used in the maritime field in order to explain and calculate how bodies behave in water. Throughout the years, different scientists came up with dissimilar ways in which to calculate different theories in the hydrodynamics field such as ship motion for both diffraction and radiation problem. In this chapter the methods used in the Vessel.js library to provide motion to the 3D entities are defined from less to more complex.

#### 2.1.1 Estimation of ship motions with closed-form expressions

The first theory method (Jensen et al., 2004) is a semi-analytical approach to derive frequency response functions for the wave induced motions of monohull ships. This approach was developed to obtain a quick and close approximation of the wave-induced motions and accelerations in the conceptual design phase and mainly relies on the few parameters known on this stage of the design such as length, breadth, draught, block coefficient, water plane area coefficient, heading and speed.

In his study, Jensen compare the expressions obtained with detailed seakeeping analysis made with strip theories (SGO and Shipmo) as well as with experimental data obtained from model tests. The results obtained after this comparison are demonstrated to be reasonably close to the theories with which it has been compared with, as seen in Figure 2.1. In this figure one can see the validation of the expressions made upon four vessels with different methods for heave and pitch motions with various wave headings, together with the correspondent vertical acceleration for the sake of comparison. These expressions are well stable for long-term predictions when considering the operation profile and the wave scatter diagram.

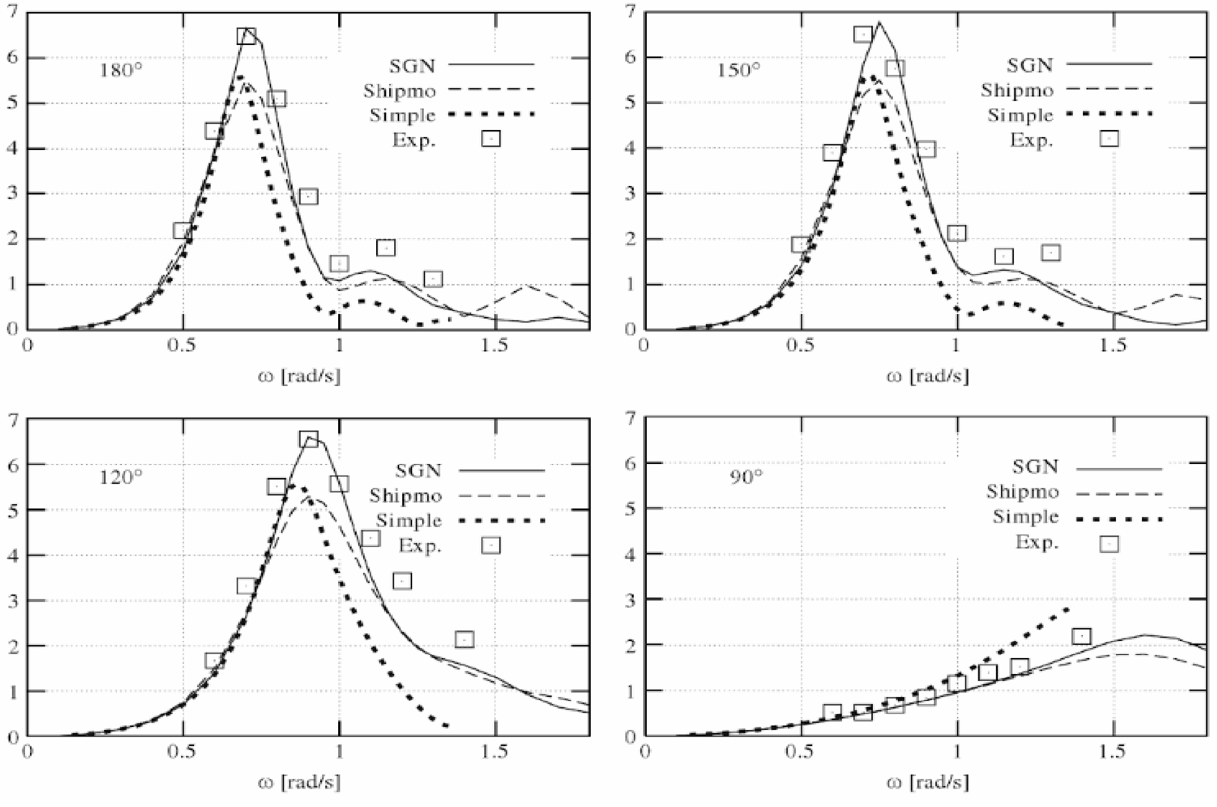


Figure 2.1. Comparison of the expressions with model tests and various linear strip theory results for the vertical acceleration at forward perpendicular for a monohull ferry as function of wave frequency for various headings. (Jensen, 2004)

Despite of this accuracy, there are some conditions under which the expressions face some limitations such as when:

- Heave is too small for $\lambda/L > 1$
- Pitch is too large around $\lambda/L = 1$ for Froude numbers larger than 0.2
- Roll is too large around the resonance frequency.

Jensen's paper only considers the motions for a box-shaped vessel and only accounts for the vertical motion and roll, by deriving analytically the linear strip theory (Gerritsma et alt., 1964). For the vertical motions the coupling terms between heave and pitch are neglected and assumed a constant added mass which is equal to the displaced water. The equations of motion for these two motions are:

$$2\frac{kT}{\omega^2}\ddot{w} + \frac{A^2}{kB\alpha^3\omega}\dot{w} + w = aFcos(\omega t) \tag{2.1}$$

$$2\frac{kT}{\omega^2}\ddot{\theta} + \frac{A^2}{kB\alpha^3\omega}\dot{\theta} + \theta = aGsin(\omega t) \tag{2.2}$$

Being $w$ the position for the heave equation and $\theta$ for the pitch equation of motion, K being the wave number, $\omega$ is the wave frequency, B, T are the bread and draft of the box respectively and alpha is defined as:

$$\alpha = 1 - Fn\sqrt{kLcos\beta} \tag{2.3}$$

For the roll motion, assuming to be decoupled from the other transverse motions, the equation of motion in regular waves is the following:

$$\left(\frac{T_N}{2\pi}\right)^2 C_{44}\ddot{\varphi} + B_{44}\dot{\varphi} + C_{44}\varphi = M \tag{2.4}$$

Where $T_N$ is the natural period for roll, which replaces the mass moment of inertia and the added mass in the equation of motion. M is the roll exciting moment, which is found with the Haskind relation. $C_{44}$ is the restoring moment coefficient and finally $B_{44}$ the hydrodynamic damping of the ship. This restoring coefficient can be obtained with:

$$C_{44} = g\Delta GM_T \tag{2.5}$$

### 2.1.2 Estimation of ship motions with differential equations

Any stochastic process can be analysed in frequency and time domain. This is important to consider because every method uses a different way to solve the equations of motion. What will mainly change is that if the system is linear, which means that it behaves linearly in respect to its displacement, velocity and acceleration, then this system can be studied in the frequency domain, where the ship's motions are considered as low amplitude sinusoidal motions. In real life though, linearity is rare, as viscous damping, forces and moments due to currents, waves, wind, anchoring loads and second order wave loads in the system are encountered. Instead, the solution of the equations of motion must be dependent on time.

For the first option the motion amplitude is solved among a range of different frequencies whereas for the time domain, the motion in a given frequency is represented through time. In this study, the main focus is to know how the response of these motions during the operations develops in time for a given frequency or frequencies, so the focus is centred into developing these equations of motions in time domain.

For this estimation the equations are solved by direct numerical integration of non-linear equations by integrating the acceleration and velocity curves in the time domain. Being done step by step, numerical integration provides several advantages in the studied case, where a follow-up can be made through time, and non-linear effects, such as the non-restoring moment and viscous damping are accounted. Additionally, the boundary conditions can be changed in every time step. This will be useful for the study, as the conditions are different in every time step of the simulation. The approximation for any simulation studied will be nonlinear, and in order to do that, a paper exposing nonlinear modelling of marine vehicles in six degrees of freedom (6DOF) is followed (Fossen, 1995).

***Equations of motion***

Applying the Newton's second law, the motions of a barge in still water can be represented by a mass-damping-spring system, as shown in Figure 2.2, where a disturbance occurs, or an initial state is set. Going back to a ship, if this force is totally perpendicular to the vessel in z direction, it will create heave displacement, which is calculated with the following equation of motion.

$$M\ddot{x} + C\dot{x} + kx = \sum F\,(t) \tag{2.6}$$

Where

M is the mass of the system, also called hydrodynamic added mass.

B is the damping coefficient associated with the fluid.

C is the "spring" constant that acts like a restorative force.

x is the position along the axis in which we are analyzing the equation of motion.

$\dot{x}$ is the velocity in the same direction, which is the time derivative of the position

$\ddot{x}$ is the acceleration in the same direction, being the derivative of the velocity.

F(t) are the forces acting on the ship in the desired direction at a desired time.

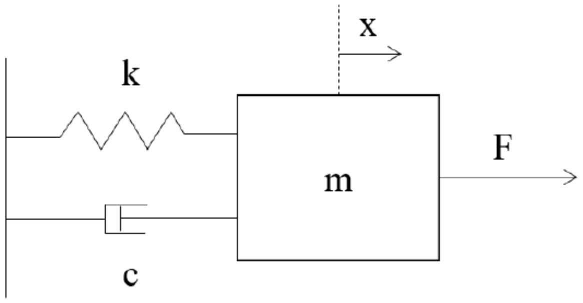


Figure 2.2. Scheme of Modelled Oscillator

Having in mind this well-known system, a floating barge is forced downwards by an external force F(t), which is applied coinciding with the movement of the mass, so it will maintain the system oscillation and the system can be in resonance. This force creates a displacement of z(t), and for then the buoyancy force (ρg∇) is larger than the constant gravitational force (mg). There is now the displaced volume ∇ in addition of how much the displaced volume is increased (δ∇). This creates a restoring force which is equivalent to a spring with stiffness K= $-\rho g A_{wl}$.

If this “spring” is compressed or placed in tension it will create a restoring force when trying to return the mass to its original location. This magnitude is proportional to the magnitude of the displacement. But the mass coming back to its equilibrium position will have some inertia which will place the mass to the other side of its equilibrium position. Then, the spring will create another restoring force in the other direction in order to restore the mass to its original position.

This motion will go repeatedly until the effects of the damper dissipate to zero the energy that the system has been accumulating because of the oscillations. This damping effect is equivalent to the damper in the classical example of mass-spring-damper as $D = -F_{damp}/v$. During heave motion for instance, the energy used to create waves is used as a damping coefficient, which quickly dissipates the energy of the system.

Once known an approximation of how the barge is going to behave, rigid body motions must be explained in order to understand the motions in all degrees of freedom of a single entity in calm waters. For this, surge ($\eta_1$), sway ($\eta_2$) and heave ($\eta_3$) are defined as translational motions and roll ($\eta_4$), pitch ($\eta_5$) and yaw ($\eta_6$) as oscillatory angular motions as seen in Figure 2.3.

Also, in the figure, when analysing the motions of a vessel in 6 DOF, it is very convenient to establish two earth-centred coordinate frames, an inertial one or earth-fixed and a body-fixed coordinate system. The first one rotates with the earth around its spin axis, whereas the second one, it is directly defined on the vessel itself.

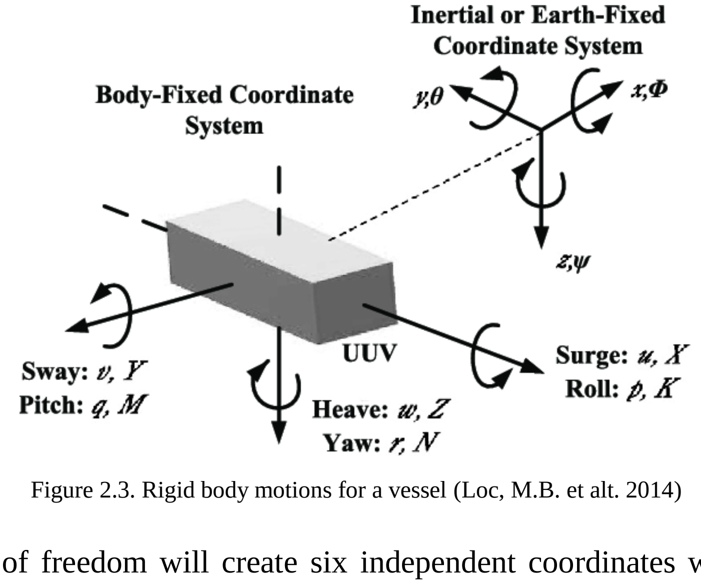


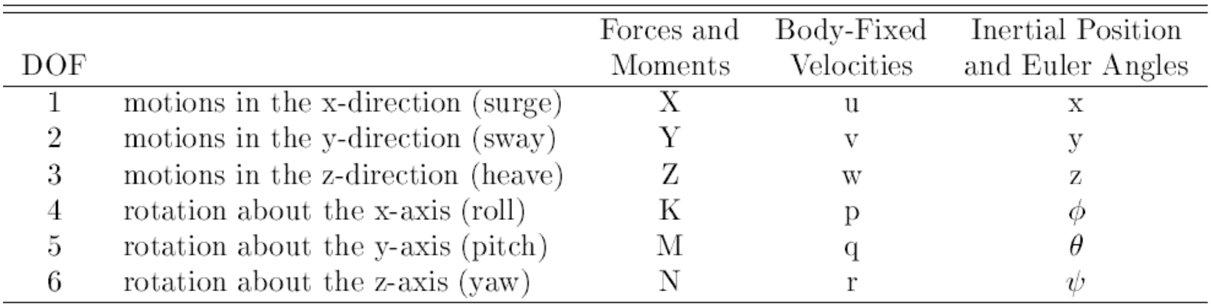
Figure 2.3. Rigid body motions for a vessel (Loc, M.B. et alt. 2014)

These six degrees of freedom will create six independent coordinates which will represent translational and angular positions. Note that vector ($v$) of linear and angular velocities is determined in the body fixed axis and the vector ($\eta$) of positions and orientations is defined in an Earth fixed axis system.

$$\begin{aligned}
&\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T]^T; \qquad &&\boldsymbol{\eta}_1 = [x, y, z]^T; \qquad &&\boldsymbol{\eta}_2 = [\phi, \theta, \psi]^T \\
&\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T, \boldsymbol{\nu}_2^T]^T; \qquad &&\boldsymbol{\nu}_1 = [u, v, w]^T; \qquad &&\boldsymbol{\nu}_2 = [p, q, r]^T \\
&\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T, \boldsymbol{\tau}_2^T]^T; \qquad &&\boldsymbol{\tau}_1 = [X, Y, Z]^T; \qquad &&\boldsymbol{\tau}_2 = [K, M, N]^T
\end{aligned}$$

To understand this, the position and orientation of the body relative to the inertial frame can be represented by a vector $\eta = [\eta_1^T, \eta_2^T]^T$ where $\eta_1$ represents the Cartesian coordinates of the origin of coordinates as measured in the inertial frame and $\eta_2$ specifies the rotation angles. The translational and angular velocities of the body relative to the inertial frame and expressed in the body-fixed frame, are denoted as $v_1$ and $v_2$.

Table 2.1. Definition of the physical characteristics of motions in every DOF. (Fossen, T.I. 2011)



| DOF | | Forces and Moments | Body-Fixed Velocities | Inertial Position and Euler Angles |
|---|---|---|---|---|
| 1 | motions in the x-direction (surge) | X | u | x |
| 2 | motions in the y-direction (sway) | Y | v | y |
| 3 | motions in the z-direction (heave) | Z | w | z |
| 4 | rotation about the x-axis (roll) | K | p | $\phi$ |
| 5 | rotation about the y-axis (pitch) | M | q | $\theta$ |
| 6 | rotation about the z-axis (yaw) | N | r | $\psi$ |

In order to create an accurate model of the block motions, the connection between the vector ($\eta$) and the vector ($v$) must be defined. This can be expressed with the following Jacobian matrix.

$$\dot{\eta} = J(\eta)v \tag{2.7}$$

Being $J(\eta)$ the rotation matrix that converts the vector described in the absolute coordinate system to the actual vector for the body coordinate system. The *c*, *s* and *t* notations stand for cos, sin and tan respectively. The obtention of this matrix is explained in Annex A.

$$\boldsymbol{J}(\boldsymbol{\eta}) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta & 0 & 0 & 0 \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi & 0 & 0 & 0 \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & 0 & 0 & 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & 0 & 0 & 0 & \mathrm{s}\phi/\mathrm{c}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix}$$

To derive the equations of motion for a vessel, the motion of a rigid body along with its hydrodynamics and hydrostatics is defined. According to (Fossen, 1991), the rigid body dynamics can be expressed as:

$$M_{RB}\dot{v} + C_{RB}(v)v = \tau_{RB} \tag{2.8}$$

The equations of motions will be derived from a vector parametrization of the inertia, Coriolis and centrifugal force, as well as the hydrodynamic added mass forces, with $M_{RB}$ representing the rigid body mass matrix, and $C_{RB}(v)$ the rigid body Coriolis and centripetal matrix due to the rotation of the body frame about the inertial frame. Finally, $\tau_{RB} = [X, Y, Z, K, M, N]^T$, represents a generalized vector of moments and external forces expressed in the body reference, also as:

$$\tau_{RB} = \tau_{hydrodynamic} + \tau_{hydrostatic} + \tau_{wind} + \tau_{wave} + \tau_{control} \tag{2.9}$$

Then, Newton's second law is used to express the motion of the barge in an ordinary differential equation. For a barge in the body fixed frame (most common), Fossen states that the mathematical model for the equation of motion is given with the following differential matrix equation:

$$(M_A + M_{RB})\dot{v} + (C_{RB}(v) + C_A(v))v + B(v)v + g(\eta) + g_0 = \tau_{ext} + \tau_{wind} + \tau_{wave} \tag{2.10}$$

Where

$M_{RB}$ is the rigid body inertia matrix (6x6)
$M_A$ is the inertia of the added mass (6x6)
$C_{RB}(v)$ is the rigid body Coriolis and centripetal matrix (6x6)
$C_A(v)$ is the Hydrodynamic added Coriolis and centripetal matrix (6x6)
B(v) is the hydrodynamic damping matrix (6x6)
$g_0$ are the restoring forces and moments due to ballast and systems. (6x1)
g($\eta$) is the vector for generalized gravitational and buoyancy forces. (6x1)
$\tau_{ext}$ is the vector of excitation forces (such as propulsion forces) (6x1)

The content of these matrices can be found in Annex A.

With the combination of the equation of motion given by Fossen (Equation 2.10), the rotational transformation (Equation 2.6) and considering that the force vector is decomposed by the excitation force, the damping force, the restoring force and the gravitational force, the equation to solve is:

$$\dot{v} = \frac{[J(\eta) * \tau_{wave} - (C_{RB}(v) + C_A(v)) * v - B * v - J(\eta) * C * \eta - J(\eta) * g]}{M_{RB} + M_A} \quad (2.11)$$

Numerical integration methods are required to solve the equation of motion (Equation 2.11), such as Dormand and Prince (1980) with its Dormand and Prince (4,5). The advantage about the numerical integration is that it will create an array for a solution in every time step, so the size of the array is different for all kinds of different input.

### 2.1.3 Estimation of ship motions with potential theory

WaveAnalysisMIT (WAMIT) was developed at MIT in 1987 and it is a complete tool to solve the diffraction/radiation problem in order to analyse the interaction between waves and structures and is widely used because of its high degree of accuracy and efficiency. This software solves the velocity potential in the wetted surface of the structure, and it is based on linear second-order potential theory which solves the problem by using the Boundary Element Method (BEM) with three-dimensional panel elements.

It is a very complete software which provides the user with hydrodynamic data such as added mass A(w), potential damping coefficients B(w), restoring terms, wave exciting force (calculated via Haskind's Relation), loads, pressure velocity in the fluid domain, first- and second-order wave forces, mean drift forces and structure motions for a given geometry in a specific period and wave direction. It is also a very versatile tool to develop all sorts of different simulations, allowing as many maritime configurations as one can imagine. Wamit gives the results for 6xN DOF, being N the number of bodies in the simulation. The processed data can be used in real-time simulation of marine vessels in all these degrees of freedom.

So far only 2D potential theory has been considered with the closed-form expressions or the differential equations. Also, some well-known software in ship design such as ShipX (Veres) are using this Strip Theory, which approximates the motion of the fluid as two-dimensional and carries the calculation by dividing the submerged part of the vessel into a finite number of strips. Then the 2D hydrodynamic coefficients are calculated with the boundary element method (BEM). These methods with 2D potential theory tend to predict motions of multibody systems simplifying the hydrodynamic (wave and current) interactions between bodies.

The use of 3D potential theory through panel methods is a lot different than 2D potential theory, as the software offers to calculate the retardation functions that describe the fluid memory effects, while solving a boundary value problem for zero and infinite added mass. Using this theory, the flow is assumed to be potential, with no separation nor lifting effects, as well as considering the free surface and body boundary conditions as linear. With this theory one also makes sure to have a dynamic performance, as it accounts the mass and the forces acting upon the vessels and also ensure to evaluate correctly the wave loads in a multibody system, since it considers the radiation and diffraction of waves due to the motion of each body and the influence these have over the others.

When applying 3D potential theory, in order to allow the application of panels, the integrals over the fluid domain can be converted to integrals over the boundaries of the fluid domain. This panel method will divide the surface of the ship into discrete elements or panels. In every panel a system of sources and sinks is defined in order to comply with the Laplace equation, allowing this way to find the strength of the distributions by identifying the potential. These potential theory programs normally compute added mass A(w), potential damping coefficients B(w), restoring terms, wave exciting force (calculated via Haskind's Relation), first and second order wave induced forces and motions (amplitudes and phases) for a specific geometry in various wave frequencies and wave directions.  An overview of the calculation process used on this study is shown in Figure 2.4, where WAMIT is integrated with a TPN-Petrobras software in order to simulate side-by-side operations.

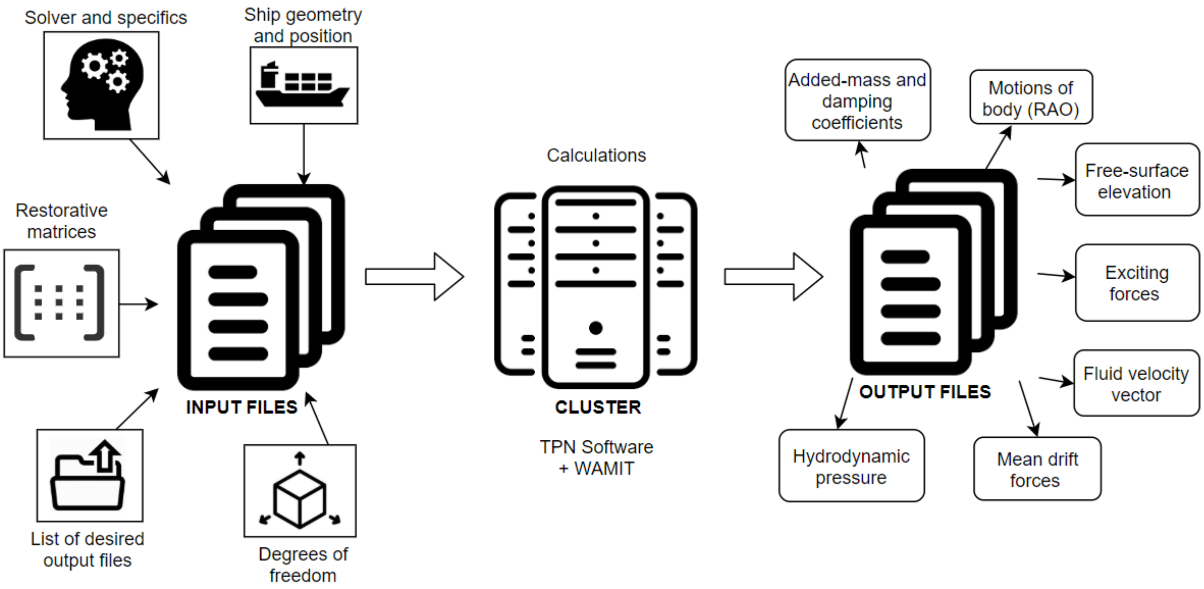


Figure 2.4. Process followed to run the cases on this study with the merged TPN-WAMIT software.

The typical application of WAMIT will start by preparing the input files and run WAMIT for after obtaining the output required, which can be printed and post-processed. The software deals with two main subprograms, the POTEN which solves the velocity potential of the body and the FORCE which evaluates the physical parameters such as the force and motion coefficients, fluid pressure, velocity and free-surface elevation.

Reached this point, the only thing to consider as a drawback when using WAMIT to this study is that it requires certain time to computationally run a problem, making it incompatible with web-based simulation, as when working in a web environment, the user expects to obtain an immediate response after providing some input. It is for this reason that this study goes after an open source simulator with pre-loaded calculations for operations that can help to visualize WAMIT output files provided with several parameters and characteristics.

#### *2.1.3.1 Hydrodynamics for side by side vessels*

Having a vessel in an open space is not the same as having a vessel with another one next to it, even if these are not in direct contact. When experiencing incoming waves from any possible angle, the motions of these vessels will be affected by the vessel next to them. This is the so-called shadow effect. To account for this, (Chen et alt., 2018) describe the motion of each vessel with a similar equation of motion from a rigid body, but accounting twelve degrees of freedom instead of six, in order to include both ships.

$$[M + a(\infty)]\{\ddot{\xi}\} + Df(\{\dot{\xi}\}) + K\{\xi\} + \int_0^t [h(t-\tau)]\{\dot{\xi}\} = F_{wave} + F_{curr} + F_{wind} + F_{ext} \tag{2.12}$$

For the vessel's inertia tensor and the added mass matrix, which gives the total inertia of the ship, changes the dimension of the matrix from 6x6 to a 12x12 matrix, as it considers the new vessel.

| **6 DOF** | **12 DOF** |
|---|---|
| $[M + a(\infty)]$ | $\begin{bmatrix} [M + a(\infty)]_{i,i} & a(\infty)_{i,j} \\ a(\infty)_{j,i} & [M + a(\infty)]_{j,j} \end{bmatrix}$ |

Where the i and j subscripts refer to the different vessels. Note that there are two matrices with the total inertia for each ship, and two new 6x6 matrices, that dictates the effects on the added mass matrix from one body to the other one, being *i,j* the influence that has ship *i* over ship *j*, and *i,i* the ship itself without interaction. The same applies for the damping (D), restoring matrices (K) and coupled retardarion matrix ($h(t-\tau)$). Also, due to symmetric properties, these equations can be combined alternatively.

$$a(\infty)_{i,j} = a(\infty)_{j,i}\,, h(t-\tau)_{i,j} = h(t-\tau)_{j,i}$$

Summarizing, the coupled motion equation for two bodies can be expressed with a set of 12 coupled equations:

$$\begin{bmatrix} [M+a(\infty)]_{i,i} & a(\infty)_{i,j} \\ a(\infty)_{j,i} & [M+a(\infty)]_{j,j} \end{bmatrix} \begin{Bmatrix} \ddot{\xi}_i \\ \ddot{\xi}_j \end{Bmatrix} + \begin{bmatrix} D_{i,i} & D_{i,j} \\ D_{j,i} & D_{j,j} \end{bmatrix} \begin{Bmatrix} \dot{\xi}_i \\ \dot{\xi}_j \end{Bmatrix} f\left( \begin{Bmatrix} \dot{\xi}_i \\ \dot{\xi}_j \end{Bmatrix} \right) +$$

$$\begin{bmatrix} K_{i,i} & K_{i,j} \\ K_{j,i} & K_{j,j} \end{bmatrix} \begin{Bmatrix} \xi_i \\ \xi_j \end{Bmatrix} + \int_0^t \begin{bmatrix} h(t-\tau)_{i,i} & h(t-\tau)_{i,j} \\ h(t-\tau)_{j,i} & h(t-\tau)_{j,j} \end{bmatrix} \begin{Bmatrix} \dot{\xi}_i \\ \dot{\xi}_j \end{Bmatrix} d\tau = \begin{Bmatrix} F_i \\ F_j \end{Bmatrix} \quad (2.13)$$

Note that the hydrodynamic interaction effects on the added mass and the damping forces are accounted in the coupled added mass as well as in the coupled retardation functions at an infinite frequency.

Furthermore, when evaluating the responses of multibody side-by-side configurations, it is very important to consider other aspects that can interfere on the interaction such as the mechanical coupling between the bodies and some slender members composing the coupling such as fenders, mooring lines, risers and hawkers. The coupling effects with the hull must also be considered.

***Mooring line theory for visualization purposes***

A catenary system is the most used configuration for mooring systems when it comes to shallow water. The catenary term refers to the free hanging line due to the effect of gravity and produces a restoring force due to the suspended weight of the catenary, which increases with the depth, and the change of the vessel motion, which means that the vessel is trying to lift the mooring lines.

This catenary system will terminate with a horizontal line in the seabed, meaning that the anchor point only depends on the horizontal forces happening on the seabed. A catenary mooring line is always composed of two parts, a half free hanging catenary and a line laying in the seabed (Figure 2.5 and 2.6 respectively). Using H.M. Irvine (2000) equations for mooring lines, some properties are given during the case studies, and then geometry can be represented as a function of all the forces that are acting at the end of the line, as seen in Equations 2.13 and 2.14.

$$x(s) = \frac{H}{\omega}\{\ln\left[\frac{V_a+\omega s}{H} + \sqrt{1+\left(\frac{V_a+\omega s}{H}\right)^2}\right] - \ln\left[\frac{V_a}{H} + \sqrt{1+\left(\frac{V_a}{H}\right)^2}\right] + \frac{H_s}{EA} \quad (2.13)$$

$$z(s) = \frac{H}{\omega}\left[\sqrt{1+\left(\frac{V_a+\omega s}{H}\right)^2} - \sqrt{1+\left(\frac{V_a}{H}\right)^2}\right] + \frac{1}{EA}\left(V_a s + \frac{\omega s^2}{2}\right) \quad (2.14)$$

Where $\omega = gA(\rho_{cable} - \rho_{water})$ . These two equations describe the catenary profile given any input on the right-hand side of the equations.

However, H and V are not normally known, so the following equivalences are used in order to simplify the equations. $H_a = H\ , V_a = V - \omega L$. Assuming this means that the decrease in the vertical anchor force component is proportional to the mass of the suspended line.

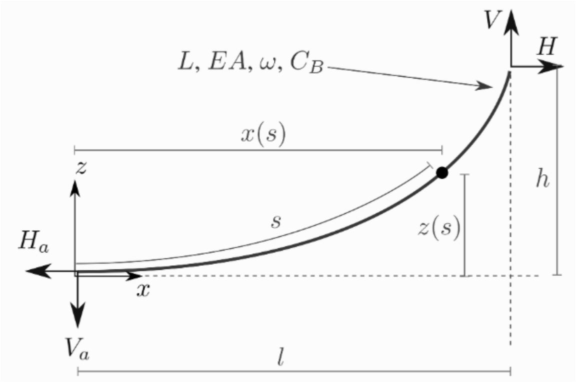


*Figure 2.5. Free hanging catenary. (Cable structures, H.M. Irvin, 2000)*

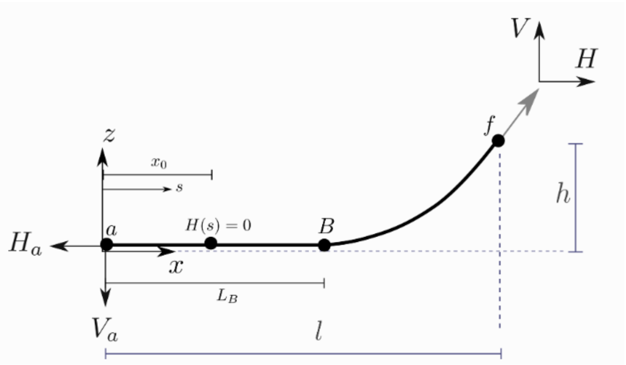


*Figure 2.6. Catenary mooring line. (Cable structures, H.M. Irvin, 2000)*

Going back to Equations 2.13 and 2.14 and knowing the values for *l* and *h*, the values V and H can be found by solving both equations simultaneously.

$$l = \frac{H}{\omega}\left[\ln\left(\frac{V}{H} + \sqrt{1 + \left(\frac{V}{H}\right)^2}\right) - \ln\left(\frac{V-\omega L}{H} + \sqrt{1 + \left(\frac{V-\omega L}{H}\right)^2}\right)\right] + \frac{HL}{EA} \tag{2.15}$$

$$h = \frac{H}{\omega}\left[\sqrt{1 + \left(\frac{V}{H}\right)^2} - \sqrt{1 + \left(\frac{V-\omega L}{H}\right)^2}\right] + \frac{1}{EA}\left(VL - \frac{\omega L^2}{2}\right)$$

(2.16)

Once defined the free hanging catenary, the rest is about obtaining an expression fort the resting cable on the seabed. This part of the cable will behave depending on the free catenary line motion. The solution of this problem is found by continuing Equation 2.13 and 2.14 beyond the point where the line enters in contact with the seabed. In order to guarantee the continuity of the boundary conditions, integration constants are introduced.

$$x(s) = \left\{\begin{array}{cc} s & if\ 0 \le s \le x_0 \\ s + \frac{C_B\omega}{2EA}\left[s^2 - 2x_0 s + x_0\lambda\right] & if\ x_0 < s \le L_B \\ L_B + \frac{H}{\omega}\ln\left[\frac{\omega(s-L_B)}{H} + \sqrt{1 + \left(\frac{w(s-L_B)}{H}\right)^2}\right] + \frac{H_S}{EA} + \frac{C_B\omega}{2EA}\left[x_0\lambda - L_B^2\right] & if\ L_B < s \le L \end{array}\right\} \tag{2.17}$$

Being $\lambda = L_B - \frac{H}{C_B\omega}$ for $x_0 > 0$ or $\lambda = 0$ if it is not. Considering when $0 < s \le L_B$, the vertical height is null due to the line resting on the seabed. So, the forces will only happen parallel to the horizontal plane.

$$z(s) = \left\{ \begin{array}{cc} 0 & if\ x_0 < s \le L_B \\ \frac{H}{\omega}\left[\sqrt{1+\left(\frac{w(s-L_B)}{H}\right)^2} - 1\right] + \frac{\omega(s-L_B)^2}{2EA} & if\ L_B < s \le L \end{array} \right\} \tag{2.18}$$

Equations 2.17 and 2.18 define the geometry of the mooring line as a function of the free hanging catenary length ($s$). Then a closed-form expression for $l$ and $h$ that solves simultaneously $H$ and $V$ is defined by substituting s = L. Solving Equations 2.19 and 2.20 one can finally find all the parameters for the problem and solve the catenary mooring line.

$$l = L_B + \left(\frac{H}{\omega}\right)\ln\left[\frac{V}{H} + \sqrt{1+\left(\frac{V}{H}\right)^2}\right] + \frac{HL}{EA} + \frac{C_B\omega}{2EA}\left[x_0\lambda - L_B^2\right] \tag{2.19}$$

$$h = \frac{H}{\omega}\left[\sqrt{1+\left(\frac{V}{H}\right)^2} - 1\right] + \frac{V^2}{2EA\omega} \tag{2.20}$$

***Hawsers line theory for visualization purposes***

The hawsers will have a similar behaviour to the previously explained catenary line, but in this case two of them will be joined together to simulate a whole asymmetric catenary, as the ends of the hawsers will not always be at the same height and depth. These two catenary ropes (A to C and C to B in Figure 2.7) will be ruled by Equations 2.13 and 2.14 but without knowing the position of the C point.

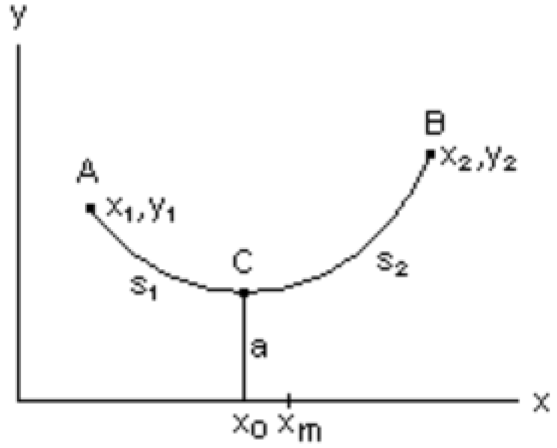


Figure 2.7. Asymmetric catenary problem. A. L. Nelson (1952)

So, the first step to solve the problem is to find the distance $a$ for later finding C point coordinates. Once this is found, two independent catenary problems are left to solve. Finding the $a$ value is an iterative process quickly done in JavaScript and it is explained with the pertinent equations below. Assuming that two ends of the whole catenary are known as well as the length of the rope, $a$ values are assumed in Equation 2.23 in order to find a value for $(x_m - x_0)/a$, where $x_m = (x_1 + x_2)/2$ is the midpoint and $\Delta x = (x_2 - x_1)/2$ is half span.

$$\frac{y_2 - y_1}{a} = 2\sinh\left(\frac{(x_m - x_0)}{a}\right)\sinh\left(\frac{\Delta x}{a}\right) \tag{2.23}$$

This expression is derived from the catenary line expression:

$$y = a\cosh\left(\frac{x}{a}\right) \tag{2.24}$$

The value $a$ is iterated on Equation 2.23 giving value to the expression $(x_m - x_0)/a$, which at the same time is applied on Equation 2.25. When the value of $s$ gives as output the real length of the rope, is when the $a$ value is correct.

$$s = \frac{y_2 - y_1}{\tanh\left[\frac{x_m - x_0}{a}\right]} \tag{2.25}$$

After that, the value of the expression $(x_m - x_0)/a$ is used to obtain the $x_0$, as the last incognita remaining. Knowing the first coordinate of the C point, Equation 2.26 is applied to obtain $y_0$, and simple trigonometry will provide the value of $z_0$.

$$y_1 = acosh\frac{x_1 - x_0}{a} + y_o \tag{2.26}$$

***Mooring line theory for motion calculation purposes***

The motion of the vessel will experience some external restoring forces when lines such as mooring lines and hawsers are part of a maritime operation. Solving this problem is not a trivial task as many factors are influencing these lines such as current, wave forces and the catenary effect with its own weight. In mooring lines, the inertia forces are proportional to the acceleration, consisting in rigid body mass and added mass. Also, if the load frequency is higher than the natural frequency, then the systems gains some inertia.

In order to determine the stiffness of the mooring line, these depend on geometrical and elastic stiffness (Faltinsen, 1990) acting like springs in series giving an overall effective stiffness.

$$\frac{1}{k_{Effective}} = \left(\frac{1}{k_{Geometric}} + \frac{1}{k_{Elastic}}\right) \tag{2.21}$$

If $k_{Geometric}$ tends to infinite, the effective stiffness is mainly elastic elongation, meaning that the mooring line is totally stretched. On the other hand, if $k_{Elastic}$ tends do infinite, the effective stiffness is mainly due to geometrical properties. This will be non-linear geometrical stiffness which is one of the most important effects on mooring lines and needs to be considered when doing a mooring analysis.

The geometric coefficient comes from the equilibrium of moments of the catenary mooring line. As mentioned before, a part of the mooring line will be laying on the seabed, and eventually, due to the motion of the vessel or block, this will move away from its equilibrium position, increasing the suspended length, the total force and the moment arm. These effects will end up creating a restoring force to the equilibrium position. On the other hand, the elastic coefficient comes from the elastic behaviour of the mooring line, which acts like a common spring. This means that when it is taken away from its equilibrium position, it will restore back to its original place.

So, the ultimate mooring force can be considered as:

$$\begin{aligned} &Mooring\ force \\ &\qquad = steady\ force\ (independent\ on\ stiffness) \\ &\qquad + slow\ drift\ mooring\ force\ (depending\ on\ stiffness) \\ &\qquad + wave\ frequency\ motion\ (depending\ on\ stiffness) \end{aligned}$$

And its displacement is understood as:

$$Displacement\ =\ Steady\ force\ /\ stiffness$$

These restoring forces will be distributed in a linear restoring matrix for the ship with various $k_{Effective}$ coefficients which will depend on the combination of motion for every degree of freedom. A simplification for this problem, assuming that the lines are symmetrical, and the system is in equilibrium, can be carried by using Matrix 2.22 (Khair, 2017). A similar approach is considered for the external restoring matrix for the hawsers.

Restoring matrix for mooring line Restoring matrix for hawsers

$$K_m = \begin{bmatrix} K_{11} & 0 & 0 & 0 & K_{15} & 0 \\ 0 & K_{22} & 0 & K_{24} & 0 & 0 \\ 0 & 0 & K_{33} & 0 & 0 & 0 \\ 0 & K_{42} & 0 & K_{44} & 0 & 0 \\ K_{51} & 0 & 0 & 0 & K_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{66} \end{bmatrix} \qquad K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{66} \end{bmatrix} \qquad (2.22)$$

However, given that these restoring coefficients are extremely complex to calculate due to the many factors they depend on, these calculations are made by the external software (MAP++)

developed by the National Renewable Energy Laboratory (NWTC). This software uses the theory for catenary lines previously explained based on a Peyrot et alt. (1978) paper to determine an external linear restorative matrix produced by mooring lines and hawsers. These mooring lines will act as spring for the ship, slightly soothing the motion of it and restraining the motion to certain degrees of freedom.

## 2.2 OCEAN VIRTUAL PROTOTYPING

He et al. (2015), propose a useful tool to help designers build virtual models which involve the characteristics of physical products through an object-oriented design resource management for virtual prototyping in collaborative design.

These technics work through a process of building, simulating and improving the models regarding requirements for function, appearance and other aspects of the products. According to this process, the design resource model can be classified in three basic types.

- Entity model (EM), which defines the physical product to be simulated, including design specification data and information about the product, as well as 2D and 3D models.
- State model (SM), which represents the EM exposed to internal and external state constrains, such as models under the ultimate work position in the kinematics simulation. This analyses the entity by assigning it a state.
- Process model (PM), which is an accumulation of the SMs, representing the model behaviour over time, from the initial to the final state. This model can also be obtained by subjecting the EM to a dynamic constraint.

An adaptation of this taxonomy was adapted for ship design related studies in order to apply it to the virtual prototyping of maritime operations, (Fonseca and Gaspar, 2015; Fonseca et al., 2018), as seen in Figure 2.8.

With this adaptation, entity models can be represented as any maritime system with all levels of detailing, such as well-defined 3D model of a ship with component specification or the characteristics and visualization of a mooring line. As for the state models of a ship, these can be represented as the entity models subjected to internal or external disturbances such as a wide range of environmental conditions, resistance of a vessel at certain speed with certain loading conditions at a given time.

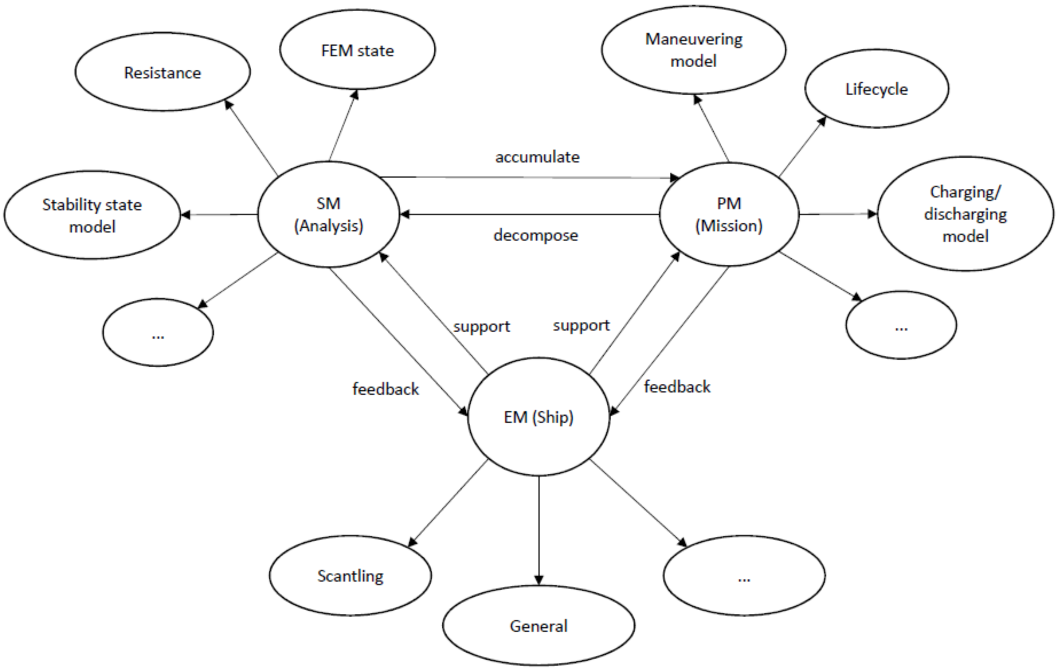
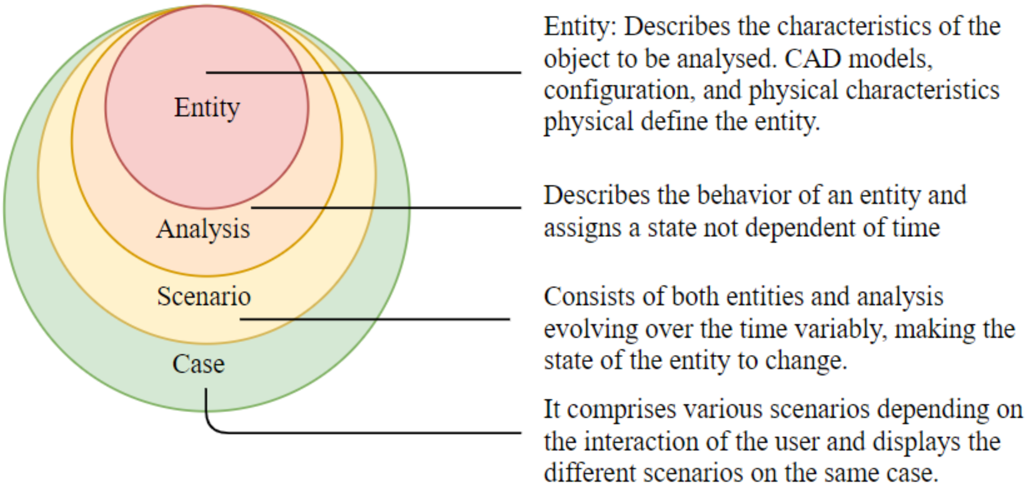


Figure 2.8. Virtual prototyping applied to ship design (Fonseca and Gaspar (2015), He et al. 2015)

Finally, for process models, a sequence of states that change over time offering a dynamic simulation is displayed. This can be changes on draft, speed or weather conditions, as well as the coupling with wave motion, leaving the user with a range of infinite possible scenarios. The user can change the scenarios simply by using a slider and then the conditions of the scenario change. All these different scenarios are comprised in a case study as shown in Figure 2.9.



Entity: Describes the characteristics of the object to be analysed. CAD models, configuration, and physical characteristics physical define the entity.

Describes the behavior of an entity and assigns a state not dependent of time

Consists of both entities and analysis evolving over the time variably, making the state of the entity to change.

It comprises various scenarios depending on the interaction of the user and displays the different scenarios on the same case.

Figure 2.9. Configuration of the model, from a single entity to an entire case study.

### 2.2.1 Open source web-based simulation with Vessel.js

The concept of web-based simulation, as its name points out, offers the possibility to run all kinds of simulations in a web environment, which makes it available to every potential user demanding very few software and hardware requirements, only a web browser, which can be opened from anywhere in the world. This interactivity brings insight to the open source concept, as such operations are difficult to describe in paper and this way, the user can play with different parameters and their interactivity.

#### *2.2.1.1 Web based design*

A web-based application depends on three main languages, HTML, which deals with the presentation, CSS which takes care of the style of the website and JavaScript, that gives the tools to develop open and interactive applications in this web environment. These three languages help to cover one of the main problems stated before for conventional simulators, compatibility, one of the keys to success of this open source concept. Hence, a web based implemented programming language like JavaScript is used in this study to modify and develop new models for the simulations in the Vessel.js library.

JavaScript is today the most widely used programming language in the world, according to GitHub statistics and it is the best one when it comes to dynamic control of websites. Using a universal language such as JavaScript provides a great potential for the current open source platform by making it something fast, reliable and that encourages a collaborative community, among other benefits. In addition, if we also combine functional programming with object-oriented, we create an even more powerful tool to work with (Wagner, 2016). Using this technology, the user can define and organize in objects thousands of different inputs and outputs within a neat and understandable interface.

For the current application, one can see how after spending a reasonable time developing a JavaScript program, solutions are obtained right after every new input is inserted in a web browser. This means that calculations are developed in the browser instantly in order to give a response to the user. On the other hand, doing the same process in some of the most popular software used by engineers such as Excel and MATLAB, could take more time and with results that are not easy to read, as they need to be run again for every iteration.

Being the most used programming language also has the advantage of having a wide range of JavaScript online libraries, information and resources online. For this study, in order to simulate graphics, several already built libraries will be used such as Bootsrap and jQuery for the sake of presenting a structured website, also Zinchart or Chart.js in order to show different graphs and Three.js along with WebGL to create and render 2D and 3D graphics.

#### *2.2.1.2 Open source concept*

Simulations in maritime operations and other domains are carried nowadays repeatedly around the world in current proprietary software, and although some software packages are starting to release their online free versions, getting some extra functionalities in them is normally a paid option. Therefore, the need for an open platform given by an open source simulation environment arises in order to see the full potential for ship design and all sorts of maritime operations and interactions.

Publishing in open source, one makes the code available for its use or modification by other users or developers. This way, it does not belong to a proprietary software company and it maintains its openness to everyone with regular updates if there in an ongoing development. This encourages a collaborative community to the steady growth of this old but yet quite recently harnessed technology, with the aim to make it freely available. To make sure that all the open source developers are in the same page, the Open Source Initiative (OSI), whom holds the certification mark for Open Source, establishes some basic terms when using this concept.

#### *2.2.1.3 Vessel.js library*

All the cases studied in this project will be developed in the Vessel.js library environment, which has most of its functionalities based on an object-oriented approach, where the code is developed as a collection of different objects, that can be put together with the aim to accomplish a task. These objects work as tools that can be pulled from different scripts in order to be used in a new one. They can handle new information, call methods, functions, make calculations and various other operations

This conceptual ship design library is made of several tools that help the user to represent a ship as a JavaScript object allowing to simulate different functionalities and behaviours for this object, which can be seen as a block with certain properties fulfilling certain requirements, such as dimensions, location and different attributes as for instance weight, density, or number of installed engines.

Every system of the vessel can be represented in an object and can be simulated as a block, obtaining as a result, a scrwhole ship composed by all the systems necessary to work. This is based in the System based ship design methodology (Lavender, 2012) which follows a bottom up approach, starting with all the payload and ship functions, assigning them an area and a volume and building up a whole vessel. Other applications are also made with the available tools.

The process is so handy that it would allow to encapsulate ships by creating a parent that would give a base to create different ships with some common characteristics. Every block has a state that collects the positions and changing characteristics of it. Giving these characteristics to all the blocks, the library is able to calculate the weight for every block (considering the lightweight and the filling ratio of the tanks) and therefore the overall displacement of the ship if all blocks are considered. Also, small angles of trim, draft, hydrostatic and stability coefficients can be calculated as part of the state of the ship (Fonseca, 2019).

The versatility of the library allows to make as many different simulations as the mind can think of, so nowadays some examples are developed in the library website (http://www.vesseljs.org). Some of these include simulations about the vessels motion using different approaches such as closed-form expressions (Jensen, 1995) or with differential equations (Fossen, 2004). Also, some examples started to work on multi-entity configurations by including some mooring lines that are following the motion of the ship, thus getting updated instantly and allowing the user to configure de lines with different input characteristics.

Also, the first example with additional bodies on the ship deck having independent behaviours such a pendulum is displayed on the library. This particular example for instance is able to render the 3D model while calculating the ship motion with closed-form expressions on real time in the browser, at the same time that the motion of the pendulum is calculated with Lagrangian equations (Fonseca, 2018), according to the input of the GUI.

These are displayed in a full set environment with textures resembling the water, sky and even the incoming light is set according to the sun displayed in the textures. Interaction is fairly easy through a GUI.

### 2.2.2 Simulation environments

These maritime simulations are run in small scale in a browser but aim to resemble in some terms the maritime, offshore and subsea operations that are carried every day in the world first class simulator centres such as the Numerical Basin Tank (TPN) in the University of São Paulo (USP), Brazil and at the Offshore Simulation Centre (OSC) in Ålesund, Norway. These centres are using simulations for either training, product development or research purposes.

#### *2.2.2.1 Tanque de Provas Numérico (TPN)*

In addition to provide the user with all kind of realistic simulations, TPN offers the possibility to test models in the wave tank seen on Figure 2.10, capable to produce all kind of wave patterns from any angle given by the user. This helps researchers to run experimental tests in the basin, analyse and compare them with their mathematical models that are calculated through a very powerful cluster. These can be later displayed on the screens of the simulators.

After many of these experimental tests and simulations, a collaboration with various Brazilian universities and the oil company PETROBRAS has developed a software that performs non-linear numerical simulations by using diverse methods and algorithms able to make a coupled analysis of different offshore systems, as well as lines and bodies together. This software includes the waves analysis software WAMIT to help solve the problem and adapts it to run into a parallel computing cluster.  It is currently used to verify and design complex offshore systems that are easier to compute than to test in a basin. Since this simulator is running, it has been performing several experiments that helped to validate the results for real case scenarios.

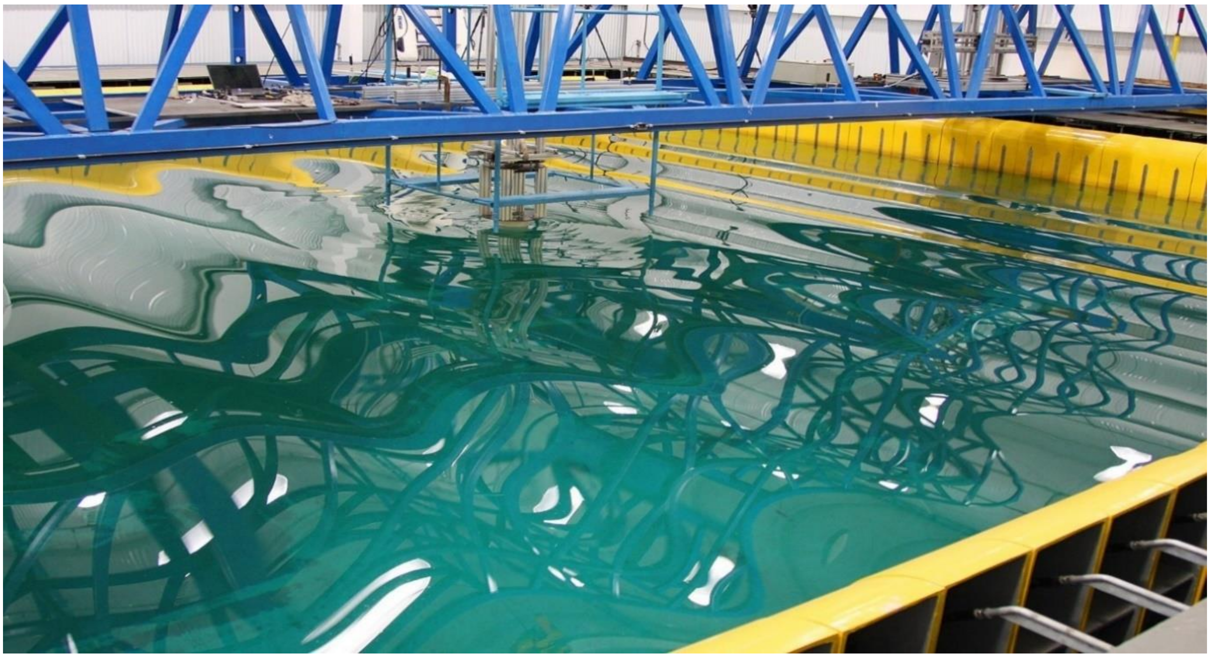


Figure 2.10. Wave tank in TPN (Tanque de Provas Numérico 2018)

It is specialized for offshore and side-by-side operations as when running simulations with lines. These are modelled with Finite Element Method (FEM) which requires a high amount of computational power. This is why TPN currently has two clusters. One of them with 1.536 cores and 15 Tflops of processing capacity and the other one of 256 cores and 2 TFlops of general processing capacity. This parallelization in TPN makes individual processes for the FEM analysis of each mooring line and hawser line, in addition to the main process that evaluates the dynamical simulations of each geometry and evaluates the environmental forces, which are processed and distributed in nodes.

The cases in this study are run with pre-loaded WAMIT runs that have been configured to run in the TPN cluster. The process as seen in Figure 2.4, consists in providing certain inputs such as geometries, dimensions, mass matrix, hydrodynamic coefficients (potential damping, added mass, and first and second order wave force coefficients), environmental conditions (wave and wind spectra and current) and the characteristics of mooring lines, risers, thrusters or DP systems, for later running it in the cluster with the mentioned software and obtaining the desired output data.

This runs in a parallel processing computer cluster that deals with the inputs in order to generate the output files which describe the motions of the geometries simulated in six degrees of freedom, the tensions on the mooring lines and the hawsers.

#### *2.2.2.2 Offshore Simulator Centre (OSC)*

OSC is one of the most advanced providers for simulators on offshore marine operations around the world. It was created on 2005 by private companies and educational institutions in order to develop a training program for demanding offshore operations. It is known that most accidents at sea are caused by human error, so this centre offers nowadays one of the best facilities to provide operation personnel with sophisticated training regarding lifting, anchor handling and platform supply operations (PSO). This training aims to teach them how to operate safely and cooperatively the different vessels and equipment on board of a vessel. By doing this, the personnel is trained to avoid accidents, or to cause them before moving to a real domain, where a misused of machinery can have catastrophic consequences.

The simulators are also used as prototypes for new developments of complex maritime and subsea operations. The operations are created according to the clients need and strictly following real physics. For this reason, they provide to the user a high-quality 3D graphics that make the simulator experience incredibly realistic with accurate textures and real equipment. These simulators, as for example the one simulating a bridge, can alter the vessel's environment such as the waves, the current, the wind, and even the daylight. Some of the crane simulations even surpass IMO standards and comply higher ones such as DNV Class A, NORSOK or IMACA.

Since the opening of OSC, they have been developing the most advanced simulators, being most of them the first ones to be developed in the world. Nowadays it has over 40 simulators with examples such as two complete anchor handling and supply vessels, a crane, a winch, a machine room and a dynamic positioning training centre. As seen in Figure 2.11, one of the simulators consist of a ship bridge with all relevant technical systems, such as maps, navigation, manoeuvring systems, radar, radio or even winch control system. The interface used in OSC links all kinds of elements in order to make the simulation as real as possible. It connects maritime physical equipment, such as consoles, handles, chairs and seats with control panels for user interaction. This interaction is brought into physical and mathematical models in order to calculate the physics behind it and it is brought back to the user through screens resembling what the user would see with their eyes in real life.

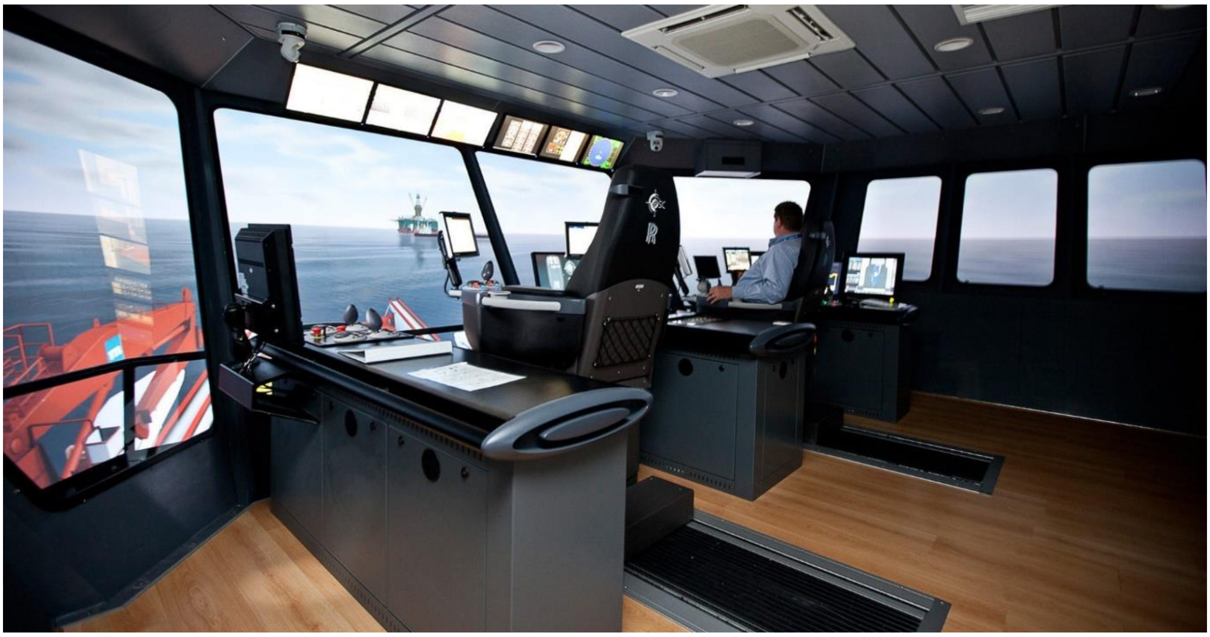


Figure 2.11. Offshore Simulator Centre in Ålesund (OSC 2017)

Furthermore, the last incorporations are for virtual/augmented reality technology, in order to make interaction even more realistic and effective. The aim for this technology is to be applied in design and development as well as being used for verification procedures, prediction of behaviour and training.

However, when it comes to open source, and collaborative coding, it is more complicated with OSC, as they provide very stiff simulators, because each crane, anchor handling or PSO models are controlled with a closed algorithm extremely difficult to modify, so it is difficult to switch between alternative control methods.

# Chapter 3

## 3 METHODOLOGY

This chapter explains the method used to tackle the proposed problem by going through a whole set of steps. Figure 3.1 shows these in two main stages, first a definition phase, where the problem is formulated and all the necessary preliminary calculations are performed in order to get into the second phase, named "web based" which will host the simulations, perform calculations and interact with the user. This aims to show that a simulation on a browser is not just several lines of code, but there is a work behind all simulations that are based on scientific concepts. Finally, the chapter closes with a simple example to show the methodology process.

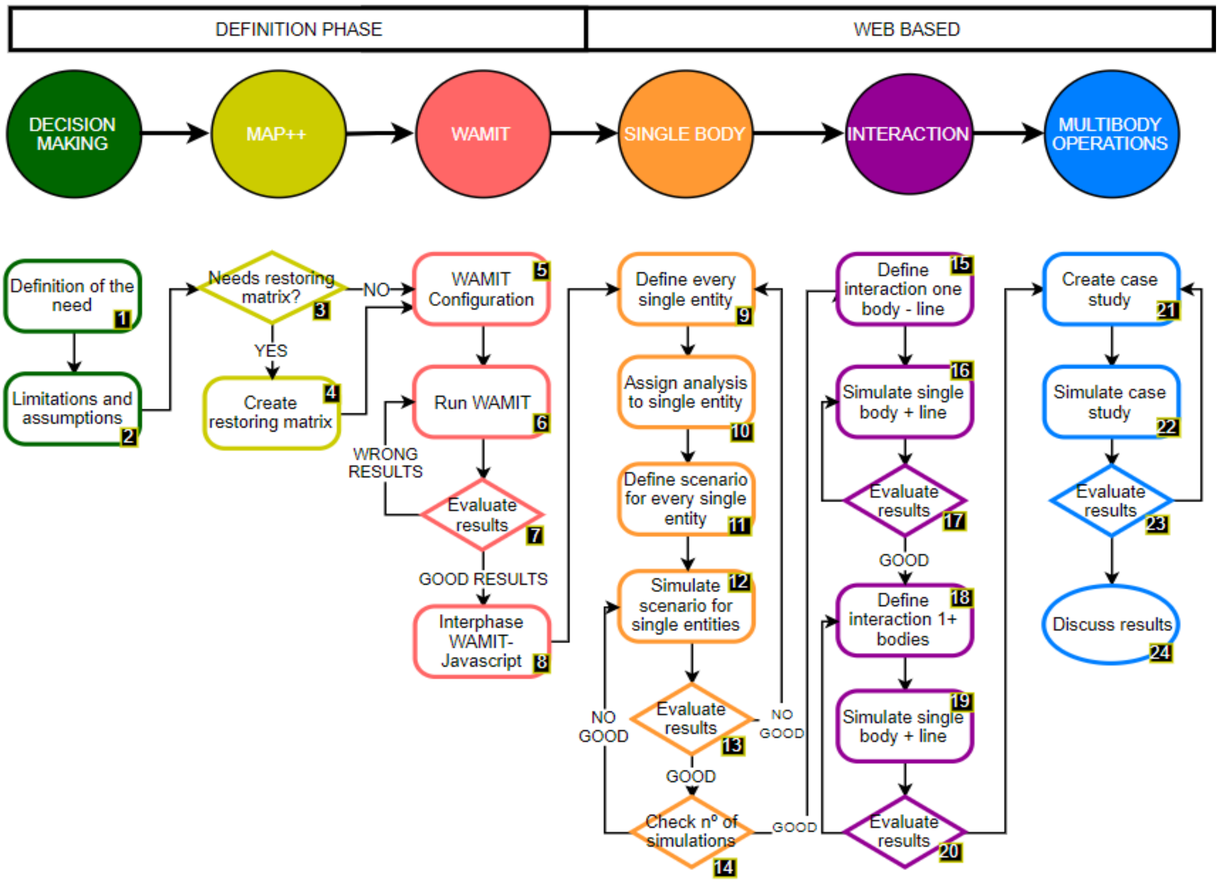

DEFINITION PHASE
WEB BASED
DECISION MAKING
MAP++
WAMIT
SINGLE BODY
INTERACTION
MULTIBODY OPERATIONS
Definition of the need 1
Limitations and assumptions 2
Needs restoring matrix? 3
NO
YES
Create restoring matrix 4
WAMIT Configuration 5
Run WAMIT 6
WRONG RESULTS
Evaluate results 7
GOOD RESULTS
Interphase WAMIT-Javascript 8
Define every single entity 9
Assign analysis to single entity 10
Define scenario for every single entity 11
Simulate scenario for single entities 12
Evaluate results 13
NO GOOD
NO GOOD
GOOD
Check nº of simulations 14
GOOD
Define interaction one body - line 15
Simulate single body + line 16
Evaluate results 17
GOOD
Define interaction 1+ bodies 18
Simulate single body + line 19
Evaluate results 20
Create case study 21
Simulate case study 22
Evaluate results 23
Discuss results 24

Figure 3.1. Flowchart of the methodology used.

Along the explanation of the methodology in Section 3, bracketed values "[*value*]" will be referring to the numbers stated on Figure 3.1 for each one of the steps, so the reader can obtain a good understanding of the method. Also, as seen in Figure 3.1, there are six major topics that will be presented on this chapter.

## 3.1 DECISION MAKING

When developing simulations for demanding maritime operations, it is important to find cases that are of interest [1], and this will be determined by finding a balance between satisfying the demand of maritime simulations and the feasibility of doing this simulation in JavaScript with the current functionalities available on the Vessej.js library.

The complexity of the simulation is also defined by what the user expects to obtain as a result, what is the input they provide and where the data is obtained from, because as it will be further seen in this study, obtaining data from WAMIT and bringing it to JavaScript code is not trivial.

It will also be important to set and identify the requirements and limitations that the different models to simulate before running any simulation [2]. These parameters are directly connected to the scope of the study being:

1) The **simulations will be dynamic**, as it is the only simulation that allows to simulate the time-dependent behaviour of systems. This kind of simulations maintain an internal memory of the combination of previous inputs, variables and conditions applied and outputs. This means that dynamic simulations will involve differential equations in order to be solved. Simulations will only be considered in a **time domain**. This is due to the need to know what happens on the system in every second and be able to simulate it visually. In this domain, external excitations such as forces depending on time or waves with given periods and heights are applied to the system over time.

2) In order to make the first simulation (radiation problem) be as close as possible to a real case and account for non-linear parameters, the equations of motion will be derived from **non-linear approaches**. These non-linearities can only be simulated in time domain, due to its time dependency, unlike analysis made in frequency domain, which only considers linear theory.

3) The equations of motion are strictly obtained by WAMIT from the RAO's amplitudes and phases provided for every degree of freedom, period, and heading. Any **misbehaviour** is due to limitations with the WAMIT simulator.

4) Decide how many **degrees of freedom** the components forming the operation will have for every single entity. For instance, the focus when working with a rope will be put only in a 3DOF motion, as some of the rotations are not of interest in the current study.

5) There must be a good **definition of** what is the **output** intended to obtain for every single model and what is going to be the initial **input** and limitations. The mathematical background should be strong enough to be able to ask the user as few parameters as possible. Certain parameters will have to be introduced to the equation on the beginning.

6) Determine the **limitations of the simulation** by identifying in which conditions the operation will not be able to be performed as a consequence of a conflict between the input and the mathematical background for the calculations, as for instance the length of the cable being too short for the distance between vessels. Warnings are designed in these conditions to tell the user that the input of the simulation is out of range.

7) Even though the effort to make this simulation more accurate, it is difficult make lots of calculations in a web-based simulator, as user computers are not powerful enough to handle so much data processing. This limits the quality of the simulations for certain users, as for instance, turning the supposed catenary effects on hawsers to linear lines. Also, in order to avoid an **excess of data processing**, in this study the mean drift forces and moments, as well as first-order and second-order wave forces are not accounted.

8) The last requirement is that the simulations made in this study must provide with a **well-structured code** in order to be understandable for other users that might be interacting and improving the code in the future. The code must also be hosted in an open source platform in order to be accessible and encourage a collaborative community.

## 3.2 RESTORING MATRICES

Before running an analysis in WAMIT, a consideration has to be made in order to assess whether the motion of the ship will be independent, or it is going to be influenced by third parties [3]. In the case of physical interactions, the ones where two entities are physically connected, there will always be some sort of a restoring matrix involved in order to keep them together, as seen in Figure 3.2, where the moored ship needs the combination of the ship motion and the restoring matrix in order to represent the appropriate motion for the ship.

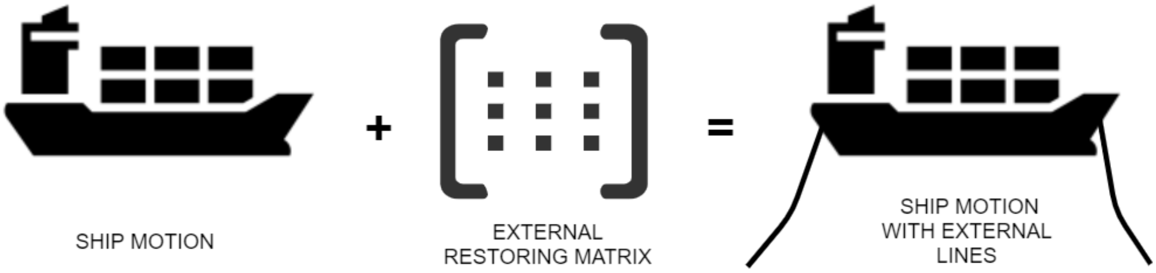
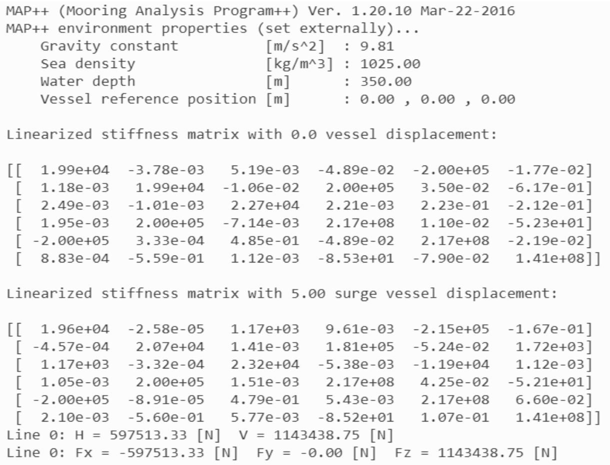


Figure 3.2. Analysis composing a moored vessel.

The link between two or more bodies is considered as a media with which these bodies are connected, therefore affecting one another. The most straightforward interaction is with ropes going from one vessel to the other which will act as simple springs damping the movement of the vessel. When considering this restoring matrix in this study, only the ones that are produced by the addition of ropes are considered, as could be mooring lines, hawsers and risers as well as a towing line. Other possible external restoring matrices such as the fenders or the interaction with the dock are disregarded and are open for future work.

Ropes are entities that do not develop an activity on their own but depend on an external source for the change of their state, which in a vessel’s case, is its motion. The barge entities will be feeding a linear 6DOF motion to the ropes, by assuming an initial point to anchor the rope and updating this point according to the barge motion. This update will be direct for the translational motions, surge, sway and heave, which will be summed on top of the initial point, and for the rotational motions, a Jacobian Matrix is used in order to find the variation of the anchoring point coordinates.

Seen this, the interaction in Vessel.js is trivial by following the motion of the ship. However, this motion is constrained by these lines, so first an analysis on how the motion of the ship is constrained by the lines is made. This influence is hard to calculate as it depends on the coupled motions of two ships with different tonnage and loading conditions. Therefore, this restraining motion through hawsers and mooring lines will be simulated with Moring Analysis Program (MAP++) software [4]. An input file with the characteristics of the mooring lines and their position in the vessel is required. This will describe parameters, such as position of the start and end of the rope, material, length, as well as determining the motion of the vessel. This software comes in a Python script and once it is run, a text file looking like Figure 3.3 with the solution is generated as well as a 3D plot of the input introduced, as seen in Figure 3.4, so one can check that the analysis is correctly configured.



```
MAP++ (Mooring Analysis Program++) Ver. 1.20.10 Mar-22-2016
MAP++ environment properties (set externally)...
    Gravity constant          [m/s^2]  : 9.81
    Sea density               [kg/m^3] : 1025.00
    Water depth               [m]      : 350.00
    Vessel reference position [m]      : 0.00 , 0.00 , 0.00

Linearized stiffness matrix with 0.0 vessel displacement:

[[  1.99e+04  -3.78e-03   5.19e-03  -4.89e-02  -2.00e+05  -1.77e-02]
 [  1.18e-03   1.99e+04  -1.06e-02   2.00e+05   3.50e-02  -6.17e-01]
 [  2.49e-03  -1.01e-03   2.27e+04   2.21e-03   2.23e-01  -2.12e-01]
 [  1.95e-03   2.00e+05  -7.14e-03   2.17e+08   1.10e-02  -5.23e+01]
 [ -2.00e+05   3.33e-04   4.85e-01  -4.89e-02   2.17e+08  -2.19e-02]
 [  8.83e-04  -5.59e-01   1.12e-03  -8.53e+01  -7.90e-02   1.41e+08]]

Linearized stiffness matrix with 5.00 surge vessel displacement:

[[  1.96e+04  -2.58e-05   1.17e+03   9.61e-03  -2.15e+05  -1.67e-01]
 [ -4.57e-04   2.07e+04   1.41e-03   1.81e+05  -5.24e-02   1.72e+03]
 [  1.17e+03  -3.32e-04   2.32e+04  -5.38e-03  -1.19e+04   1.12e-03]
 [  1.05e-03   2.00e+05   1.51e-03   2.17e+08   4.25e-02  -5.21e+01]
 [ -2.00e+05  -8.91e-05   4.79e-01   5.43e-03   2.17e+08   6.60e-02]
 [  2.10e-03  -5.60e-01   5.77e-03  -8.52e+01   1.07e-01   1.41e+08]]
Line 0: H = 597513.33 [N]  V = 1143438.75 [N]
Line 0: Fx = -597513.33 [N]  Fy = -0.00 [N]  Fz = 1143438.75 [N]
```

Figure 3.3. Restoring matrices obtained for the input and output required in MAP++

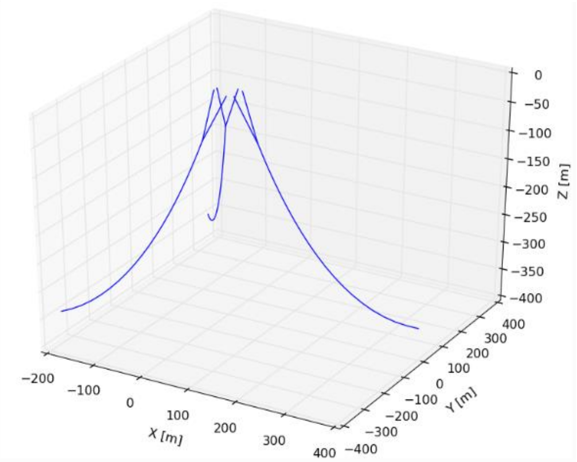


Figure 3.4. Mooring lines example in MAP++

In maritime operations, another more complex way to consider an interaction is when two bodies are affecting each other through the waves. This means that the waves reaching one of the vessels will be damped by the presence of the other vessel in proximity, directly having an impact on the second's vessel motion. This problem which considers the shadow effect is considered automatically by WAMIT when it is configured properly.

## 3.3 GENERATION OF MOTION

The motion in this study is mainly generated with WAMIT software [5], which is run in an interface not more advanced than a console. Thus, it is important to get familiarized with how the software works as it depends on several input files that will define the type of simulation to run, the geometries involved, and the forces applied to them. In addition to other less relevant files, these are the main ones used for this study.

**General Configuration file:** The most important characteristics for the simulation are stated in this file. The user can choose between a wide range of options. For the simulations that are carried in this project, some of the options have to be controlled, such as the type of solver (direct or iterative), the accuracy of the integration panels, the integration method, the use of high or low order method, panel size and number of CPU used, among other less important options. On this stage, the user can choose how to calculate some of the input files that will be required such as calculating the RAO's by using the Haskind exciting force or the diffraction exciting force. It is also possible to choose only certain specified radiation modes.

All the analysis for this study are made with the low order method, that requires less computational time, although the results are not as precise in most cases as when compared to the high order method. In the high order method, the body geometry can be represented in many different ways such as flat panels, B-splines approximations, geometrical models developed in other software, etc, whereas with low order, the geometry is approximated with quadrilateral panels and the solutions for the velocity potential are also approximated by piecewise constant values for each panel.

**General force file:** This is the file that describes the input regarding the body dynamics and where the ships to analyse are specified. Also, the kind of output that the user wants to obtain from the run is assigned in this file and it is chosen among thirteen option from which only two of them are going to be used in the study, the added-mass and damping coefficients, and the motions of the body, which provides the response amplitude operators (RAO). These last ones will provide an amplitude and a phase for each one of the degrees of freedom, periods and headings specified. Other options would be to obtain mean drift forces, fluid velocity vector, hydrodynamic pressure on the body or exciting forces. However, for the sake of simplification on the visualization of these results, these applications only account the RAO amplitude and phases, leaving the integration of mean drift and wave forces for future work. These results still seem fairly correct, but some tolerance has to be accounted.

**Ship configuration file**: This file specifies the added mass matrix, degrees of freedom, linear damping and linear external restorative matrix for external forces applied upon the vessel as obtained in Section 3.2. As for the mass matrix, it is calculated by knowing the displacement and the $I_{44}$, $I_{55}$ and $I_{66}$ coefficients with the data obtained from the CAD ship geometry.

$$\mathbf{M} = \begin{bmatrix} M & 0 & 0 & 0 & 0 \\ 0 & M & 0 & 0 & 0 \\ 0 & 0 & I_{44} & 0 & 0 \\ 0 & 0 & 0 & I_{55} & 0 \\ 0 & 0 & 0 & 0 & I_{66} \end{bmatrix}$$

where $I_{44} = M(0.35 * B)^2$ and $I_{55} = I_{66}\, M(0.25 * L)^2$

**Ship Geometry file**: The form file is modelled with Rhinoceros and a mesh is created in order to use its cells as panels. In this form file, it is very important to set the position of the ship according to the global system coordinate, because this will determine the loading condition of the ship and consequently its draft. The distance between the floating line and the centre of the global system is accounted in the ship configuration file.

**Potential file.** This file defines the problems to solve. One can choose whether to solve the radiation, diffraction or both problems. It also states incident depth, wave periods, wave heading angles, ship geometries to use as well as their positions and degrees of freedom to analyse.

### 3.3.1 Runs with multibody

When more than one body is simulated in WAMIT, an additional body consisting of an artificial rectangular damping lid is placed on the free surface between the two vessels. This gap lid is placed in order to cover enough portion between the two vessels so that it is physically possible to damp the important resonant modes. So, the lid does not have to match precisely the actual gap size, as the gap width will be variable accordingly with the ship motions.

In addition to the lid, what is different than single body WAMIT simulations, is that when multiple floating bodies are in a certain proximity, the behaviour of every ship tends to differ from its normal performance due to the large elevations of the free surface creating a resonance in the gap between the bodies. Other factors such as the shadow effect happening between these bodies will also influence their performances. So, by placing this lid, WAMIT considers the resonance between this gap and the shadow effect. What it does not consider are the appendices physically attached to the ship such as fenders or hawsers, that must be added through external linearized restorative matrices.

When the configuration is ready, the WAMIT simulation is run in TPN cluster taking an amount of time that will depend on the number of nodes used, the number of entities analysed, and the type of run carried. In the cases studied, there is a noticeable difference between analysing one single vessel or three [6].

Then a decision is made according to whether the simulation presents errors, or it can be trusted with valid results [7]. If the simulation went through, then a Python script can be run in order to quickly read the output of the simulation and plot the motions of the ship. This will tell in advance if the simulation was developed correctly only by examining the RAO.

### 3.3.2 Converting data from WAMIT to a web interface [8]

In the simulations for this study, one of the goals is to make an interphase WAMIT-web in order to bring the calculated data to the simulator. A procedure on how to implement all this theory into this programming language along with looking for similar examples that might help to develop a model must be defined in this stage of the process. There are two ways introduced in this study.

**1)** A market-oriented version where the user finds a simple “Explore file” button on the web browser in order to introduce the motion output file from WAMIT with the RAO’s. As seen in Figure 3.5, a JavaScript program reads the data from the file and turns it into a useful JavaScript object containing the RAOs with the pertinent amplitude and phase, separating them in the different degrees of freedom. The text file is turned into a string, for after converting it into values separated by a comma and then they will be gathered in a response for a single wave condition on each line. An empty object is created in parallel with the empty spots required. These will contain “x” number of cases, “y” number of degrees of freedom (according to the number of ships), 30 different periods, and 25 heading for each one of these periods. Finally, the responses are taken individually from the first object and placed in the right hierarchical order to be easily called.

Then, the user will introduce some conditions, a given wave period and a wave heading, and the JavaScript code will start to display the motion for those conditions, taking data from this last object in order to pass it into the vessel. This is a very stiff case and can only simulate two vessels side by side, and the user must be aware that the input file introduced must be of two vessels side by side in order to display good results.

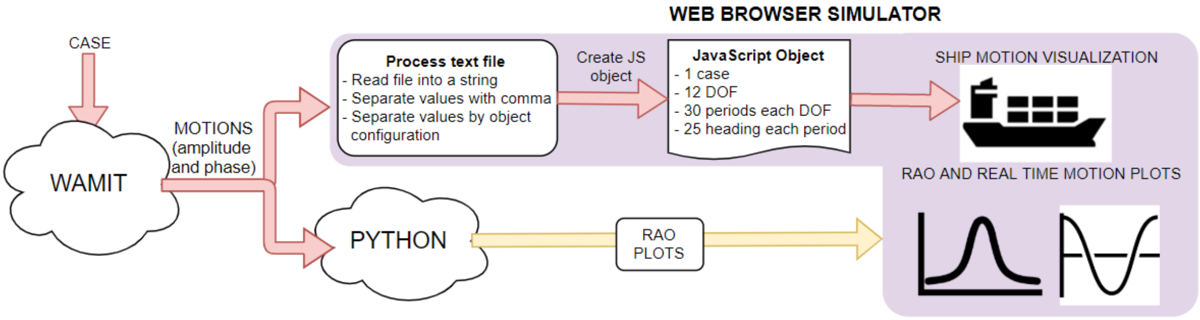

WEB BROWSER SIMULATOR
CASE
WAMIT
MOTIONS (amplitude and phase)
Process text file
- Read file into a string
- Separate values with comma
- Separate values by object configuration
Create JS object
JavaScript Object
- 1 case
- 12 DOF
- 30 periods each DOF
- 25 heading each period
SHIP MOTION VISUALIZATION
RAO AND REAL TIME MOTION PLOTS
PYTHON
RAO PLOTS

Figure 3.5. First case process, from the case definition to the visualization.

**2)** An educational oriented version with ten cases built in. In this case, there is a pre-loaded data file with all the results from the ten different runs so then the user can simply change the configuration of the simulation in order to choose one of these ten cases. This configuration will depend on the number of ships, and whether to use mooring lines and hawsers or not. The interface works the same way but this time the user does not have to provide a file, as these are pre-loaded with a JSON file as seen in Figure 3.6.

The data from the JSON text file now converted into separate responses will allow to simulate the entities motion and real time graphs so the user can compare with the RAO’s obtained from the WAMIT results.

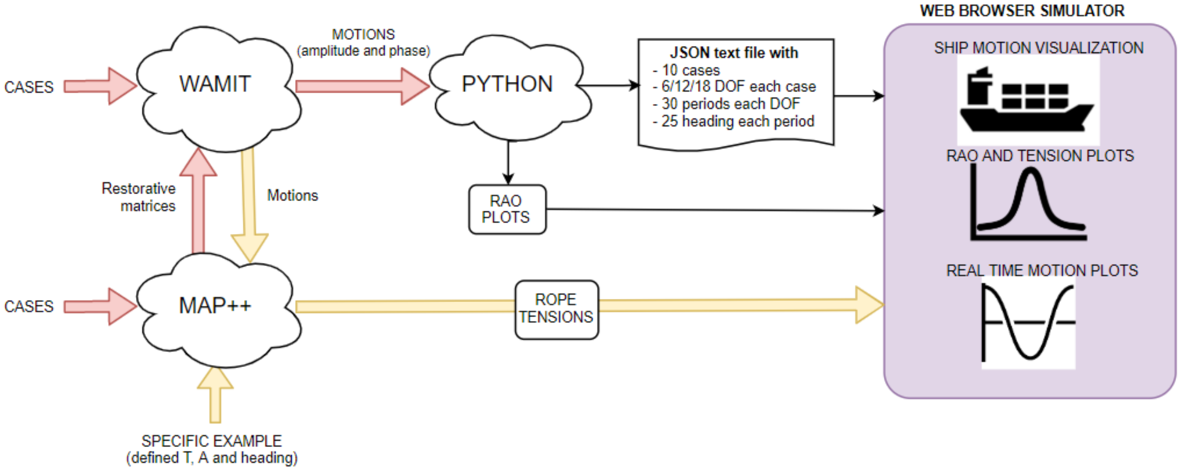


Figure 3.6. Second case process, from the cases definition to the visualization.

## 3.4 Single body simulation

The process to create a simulation starts by understanding the behaviour of the single entities for later combining them with other entities in order to create more complex cases. The first step for the simulation of a single body is to create the entity itself, which comes defined by its intrinsic characteristics, then applying a state to the object, which can depend on the body itself or on external factors. At last, this defined body and its interactions are subjected to the time variable in order to obtain a simulation that evolves over time with a dynamic character.

### 3.4.1 Generation of every entity model [9]

The entity models for the simulations will come defined by every element that separately works as a whole with a dependent behaviour. The single entities on this study will include an ocean, a seabed, a ship, a line, and additional objects placed on the seabed or elsewhere. These will be created using Three.js library and the standard methods to do so, just as Table 3.1 shows. However, for the ship, the created “box” will have additional properties and functions in additions to the ones a simple boxGeometry has, as mentioned on Section 2.2 when describing the Vessel.js library.

This level of detailing for the ship entity can include the definition of the shape of the ship due to being modularized. So, the whole entity can be created with “derived objects”, which will have a parent called “base object”. These derived objects placed in the right coordinates can create an approximation to the vessel. (Fonseca et alt., 2018). Each one of these modules inside the ship entity can be assigned with individual characteristics such as weight distribution, fullness, colour, density, dimensions, etc.

Table 3.1. Code for the visualization of different entity models used in the case studies.

| Ocean entity | Simple line entity |
|---|---|
| ocean = new Ocean({<br><br>parentGUI: gui,<br>sunDir: sun.position.clone().normalize(),<br>size: oSize,<br>segments: 127<br><br>}); | var material = new THREE.LineBasicMaterial({<br>color: 0x0000ff<br>});<br>var geometry = new THREE.Geometry();<br>geometry.vertices.push(<br>new THREE.Vectror3( -10, 0, 0),<br>new THREE.Vectror3( 0, 10, 0),<br>);<br>var line = new THREE.Line(geometry, material); |
| **Ship entity with Vessel.js** | **Simple cuboid** |
| ship = new Vessel.Ship(spec); // Vessel with defined spec<br><br>var ship3D = new Ship3D(ship, {<br>shipState: states,<br>stlPath: "../data/STL files",<br>upperColor: 0x33aa33,<br>lowerColor: 0xaa3333,<br>hullOpacity: 1,<br>objectOpacity: 1<br>});<br>zUpCont.add(ship3D); | var geometry = new THREE.BoxGeometry(700, 700, 700, 10, 10, 10);<br><br>var material = new THREE.MeshBasicMaterial({<br>color: 0xfffff,<br>wireframe: true<br>})<br><br>var cube = new THREE.Mesh(geometry, material); |

Also, a more complex way to display ropes, if they need to be acting as a flexible rope and not as a rigid bar is to divide the line with small segments following some physical properties, in this case, a catenary system or a mooring line with seabed-touching characteristics. For this, a simple line geometry will be created with a definite number of vectors with empty positions.

```
var geometry = new THREE.Geometry();

    for (var j = 0; j < 99; j++) {
    geometry.vertices.push(new THREE.Vector3(0,0,0));}
```

A calculation in JavaScript is required in order to know the position for the vertices of every catenary segment for two given points and the length which will be stored in three matrices (x,y,z directions). Finally, the vertices for the 3D line are pulled from those matrices. So, when updating the motion on the two points (or one in case of a mooring line), new segment positions are calulated, requiring the vertices in the 3D visualization to update.

```
for (var m = 0; m < 99; m++) {
        LineGeometry.geometry.vertices[m].x = x[m];
        LineGeometry.geometry.vertices[m].y = y[m];
        LineGeometry.geometry.vertices[m].z = z[m];
       LineGeometry.geometry.verticesNeedUpdate = true;
}
```

### 3.4.2 Generation of an analysis over every entity [10]

Following the taxonomy proposed in Section 2.2, the so-called state model incorporates an activity over an entity model previously defined. This model will handle all the states that an entity can take in a static way, since this activity does not involve evolving over time, being just a caption of the entity model state at a given time. These states are the response that every entity takes after experiencing a physical change, normally categorized as internal or external.

- The **internal stimuli** deal with forces happening inside the entity itself. These will be only defined in the ship and represent stimuli such as the change of the centre of gravity when a body is lifted from deck with a crane, the change on loading conditions or the process of calculating hydrostatic and stability coefficients. For the ship object, these are calculated with methods embedded in the Vessel.js library and for the other entities, physics laws apply depending on the kind of stimuli.

- The **external stimuli** deal with forces produced by external factors. Regarding a ship, examples would be the force produced by the mooring lines, hawsers or by the environmental forces acting upon it such as waves, current or wind. Also, other external stimuli are considered such as third parties placing an object to the deck or applying a force to the vessel, either a pushing or a pulling force. These external stimuli will be divided into discrete and continuous stimuli, depending on whether these stimuli depend on the progression of time. The external stimuli used in the current simulation in the form of mooring and hawsers will be introduced in WAMIT simulation through linear external restorative matrices.  Regarding other entities, the external stimuli will be for instance the motion of the ship over a line entity.

### 3.4.3 Generation of a scenario for every single entity [11]

So far, only static behaviours have been considered, therefore in order to determine what happens on the simulation over time, the process model is defined. This represents a dynamic simulation that adds the time variation over the two previous models.

In the taxonomy proposed before two types of simulations changing over time, discrete and continuous, being the first one, the kind of simulations that consider states to change abruptly over defined events or the second one tracking the system states at every time step, by using differential equations (Fonseca, 2018). For this current study, most of the operations will be considered to be continuous such as the change of the states for the mooring lines and hawsers as well as the ship motion itself. As the main driver for the study is the motion of the vessels and the other entities depend on this motion, single entity scenarios will mainly be created when Equation 3.1 is used to define the motion in every degree of freedom for the vessel. This will also be derived over other entities such as lines attached to a certain point of the vessel.

$$\eta_x = R \cdot a \cdot \cos(\omega t - \varphi) \tag{3.1}$$

Where

$\eta_x$ is vessel displacement (metres for surge, sway, heave and degree for roll, pitch, yaw)
R, $\varphi$ are the RAO amplitude and phase respectively.
$a$,$\omega$ are wave amplitude (meters) and the wave frequency (radians/second) respectively
t is time variable (in seconds).

### 3.4.4 Simulation of single objects separately [12]

With all the theoretical background obtained, the mathematical formulation of this problem must be linked to the 3D model to visualize. A solver in Vessel.js is configured to calculate the change of the vessel position for every different input, which in this case will vary with every time step and will be updated instantly on the vessel position in order to make a real-time simulation.

The simulation will be run until it is precise and ready to be simulated in a more complex case, where it will interact with other bodies. During the simulations, the process will be monitored, the inputs will be changed and tested again, in order to obtain different outputs for further evaluation [13]. For the evaluation of the single model various parameters are considered. If any of these are not up to the standards, the process should go back to define the entity, state and process models, collect more information to make the model as accurate as possible and go through other necessary steps. Thus, when evaluating the simulation, the aspects to consider are to:

- Check if the simulation works.
- Check if it simulates the operation properly.
- Check if the results from the simulation make sense with the input introduced.
- Check if the results are compatible with third party’s interaction.
- Check if the results are effective and will be useful when simulations multibody interaction.

A decision is also made depending on whether all single entities are simulated or not. After the evaluation turns out to be correct, all other scenarios for every single entity are completed [14].

## 3.5 INTERACTION BETWEEN BODIES

Defining the interactions comes as one of the main cores of this study, as these will link the multiple bodies in order to create a proper and reliable simulation of maritime operations. The theory seen in Section 2.1.3 will be applied in this section in order to apply it to Vessel.js and obtain a real-time visualization of the mathematical mode.

The obtention of single body models has allowed to analyse their behaviour individually, however these behaviours themselves will not tell the overall behaviour of the coupled system. A multibody then will be defined as a set of bodies that have different response when acting separately as single bodies than when these do it as a joined body by some sort of interaction.

In the example studied by Ormberg and Larsen in 1998, these interactions were studied in two different cases in order to compare them afterwards. The first case seen in Figure 3.7 is a separated analysis, one for the vessel and another one for the mooring line. The second case is a coupled analysis, which evaluates the whole system at once.

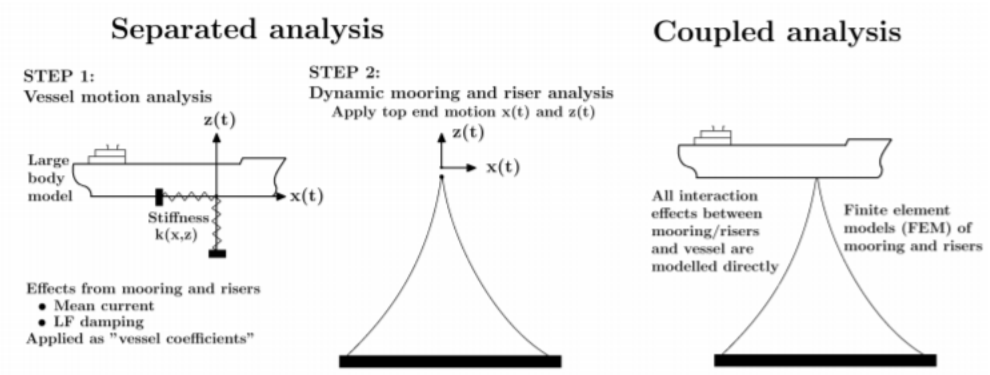

Separated analysis
STEP 1:
Vessel motion analysis
z(t)
Large body model
x(t)
Stiffness k(x,z)
Effects from mooring and risers
• Mean current
• LF damping
Applied as "vessel coefficients"
STEP 2:
Dynamic mooring and riser analysis
Apply top end motion x(t) and z(t)
z(t)
x(t)
Coupled analysis
All interaction effects between mooring/risers and vessel are modelled directly
Finite element models (FEM) of mooring and risers

Figure 3.7. Coupled and decoupled approach (Ormberg and Larsen, 1998)

They found that "the turret motions estimated by a separated analysis also compare well with both coupled analysis and experiments if mean current loads and low frequency damping from moorings and risers are included in an accurate manner. Otherwise, the use of separate analysis will severely underpredict the mean offset and overpredict low frequency motions." (Omberg et al., 1998).

For the simplification of the present study and being aware of the limitations imposed by studying multibody separately, the analysis carried in this report will be done with a decoupled approach. This leads to assume that the low frequency motions might be overpredicted and the mean offset underpredicted. So, in this case, a multibody dynamic system will be studied, which is defined as the study of the dynamic behaviour of interconnected solid bodies, which can lead to translational or rotational motions. These are connected to each other by join points (kinematic constrains) that control and restrict the relative motion. The study of multibody dynamics systems will describe how single bodies will respond to the influence of forces caused by another body.

A distinction must be made between rigid and flexible bodies. In the studied case, a vessel will be considered as a solid body (a rigid or elastic part of a mechanical body). In the case of a rope, it will be considered as a flexible and elastic body. The link between two or more bodies, will be defined by kinematical constrains which will restrict the motion of the bodies.

Multibody motion analysis, which is described by its kinematic behaviour (Esefeld, B. 2013), is relevant due to the need to understand how multiple moving bodies interact with each other and their environment, and what will happen when third party entities apply disturbances to the bodies, generally loads that are usually not easy to predict. The points where the interaction will be happening also need to be defined, and the number of independent kinematical directions one object can move. So, the number of degrees of freedom of a body will be determined by the minimum number of parameters required to define the position of an object. A free body will have six degrees of freedom when it comes to spatial motion. However, for the sake of simplification, some of the bodies will be considered to have only 3, such as the ropes, where the rotations will be considered zero for the cases studied.

Then constraint conditions must also be considered which will imply a restriction in the kinematical degrees of freedom for one or more bodies. These will define the relative translation or rotation between two or more bodies. Bilateral constrains have to be evaluated, which are ones used in fixed points or joins, and unilateral constrains, that account for friction. These constrains will be created upon a body from another body in most of the cases, in exception on there a mooring line is anchored to the seabed, which is the only moment with a fully fixed constrain on the earth axis.

### 3.5.1 Definition of the interactions [18]

The theoretical principles for the interaction between entities have been previously set in Section 2.2.3 for the simple cases studied. Further interactions will be defined for other operations with a potential to be developed.

So far, a simple rope has been defined and simulated, but when it has to be moved according to the motion there are two things to take into consideration:

- A rope in motion will not be straight and will have a catenary effect. This means that Equation 2.24 must be put in JavaScript code in a way that it gets updated with the motion. This will be done by splitting the rope in various segments that will get the vertices updated with the motion. The end of one vertex will be the start of another one.
- The first of all the vertices will be given by the anchoring point on the sea bottom or the mooring point on a vessel. Following **(**Oliveira, F.F, 2018) work, these points will be defined by default anchoring points for the rope when the ship has a steady position. Then, according to the motion of the ship, these points will get automatically updated. These get directly updated with the sway, surge and heave motion, as they are pure linear motions. However, with roll, pitch and yaw, the variation of position is calculated with a transformation using Euler angles.

Ideally, a state should be taken over time from all the bodies that need interaction in order to correlate their motions, forces and other characteristics. Then this state should be used as initial condition for another body in order to work in synchrony.

Adding hawsers derives into adding a linear external restorative matrix in both ships. In this case it causes a fully coupled stiffness matrix, as the displacement of one of the bodies produces a pulling force over the other body. The way to calculate the stiffness matrix in this case is by assuming that the displacement of the attaching point from one vessel is the same as the negative displacement of the attaching point of the other body. This produces a 12 x 12 stiffness matrix, with 6 degrees of freedom for each body.

### 3.5.2 Simulation of the interactions [19]

Simulating physical interactions such as ropes with Three.js depends on the vertices of the rope being updated. Taking the example of a catenary rope as a single entity, the vertices of the line need to be updated constantly with the motion on a certain point in the vessel, which will be called anchor point on the ship.

The motions on these points are determined by a relative position calculated with Jacobian expression and Euler angles, which help to determine the x, y, z components of the rotational motions. As for the other motions, surge, sway and heave, the displacement can be taken directly from them.

```
anchorPointOnShip = [pos[0] + motion.surge, pos[2] + motion.heave, pos[1] - motion.sway];
```

where “pos” is the relative position calculated for the rotational motions.

Once the connection point is set and simulated the results are evaluated with the same criteria previously described [20], only in this case, note that the rope segments need to be updated continuously over time.

## 3.6 MULTIBODY SIMULATION

For the sake of simplification, this study will start with low-profile simulations and will keep increasing the complexity of the models. It will start by having an initial condition that will be assumed to be a body in calm water, which lacks motion. To start having some motion, a disturbance will be applied to the body. After done this basic simulation, motion will be given, as well as introducing multibody by joining the bodies with ropes, mooring lines and giving them constrains. The object studied might be free, moored, in dynamic positioning, or being propelled.

With all the single bodies producing the desired results, the core of this study is reached, which is to simulate multibody operations in an open environment. This point evaluates all the theoretical background described so far in order to perform the desired operation by combining these single entities with other entities by using the interactions described before [21]. In this point it will be very important to decide the points of interaction and to properly define additional constrains and boundary conditions that apply to the whole system but were not studied in the single body case.

Having this clear, the code in JavaScript will be written in a structured and understandable manner. The help of various libraries will be used in order to achieve the proposed goals such as numericjs for numerical calculations, Three.js to plot the graphics in 3D, plotly to plot graphs and diagrams and others used for the application aesthetics [22]. Also, visualization solutions must be found for multibody in order to be able to run on a browser as many times as needed until obtained the desired results. Different inputs will be introduced in order to check that the application has been made correctly and also fake input will be tried, in order to assess the reliability of the simulation. If the input bumps into a simulation limitation, a warning should be display and the rest of the tests will be made in this stage.

The intend of this step is to take the simulations done and test if they are correct by calculating the same values obtained by the simulation in an external software such as Excel. This is time-costing, and despite this being the reason why the web application is made, it is necessary in order to guarantee the reliability of the simulation.

The results will be assessed also according to the requirements specified on the beginning, whether the results make sense for the introduced input and whether the results can be used effectively to approach the behaviour of the operation in the real world [23]. It is also the moment to decide whether the simulation conditions are enough, or the state model needs to be defined more accurately, either by applying different internal or external stimuli.

## 3.7 SIMPLE EXAMPLE OF THE METHODOLOGY

Following the steps described in the previous section, a simple example is developed in order to show the methodology used. Consider the simplest configuration of a barge attached to a solid rope and a fixed platform. In order to simplify the example even more, this platform is going to be a fixed body away enough from the barge, so it does not influence it through any shadow effects. This example pursues to visualize the coupled motion between the moving body and the rope. The interaction between the rope and the fixed body will not be studied.

Once defined the problem, the assumptions and the limitations, the next step is to identify all the entities to be simulated, the analysis to apply upon them and the scenarios to consider. As seen in Figure 3.8, all the entities are represented and simulated separately, and when these are all put together, the first possible scenario with a calm sea, and no motions is obtained.

Starting by adding the state model to these entities, the required entities will obtain different new states. For instance, the sea will go from a steady state to a regular sea state with a wide range of different options for amplitude, period and wave heading. As a consequence of this, the barge will need to be provided with linear motions in 6 DOF, which will cause the rope to respond to this motion and move accordingly to this one moving end, while the other remains fixed to the platform. Finally, in the case of the platform, no state model is provided as it is assumed to be fixed from the beginning and there are no parameters changing over time. These characteristics will modify the motion of the vessel in 3.750 different scenarios instead of the only one obtained at the beginning with no states in the entity models.

### *Addition of waves*

The first state to be introduced to the entity models will be the waves incorporated to the calm sea entity. Starting with this entity will then allow to consider the motion on the barge entity. For this analysis, an already built library is used, as this is not the scope of this study. This library creates a mesh that resembles a regular sea state with different parameters to play with. This is done with a graphical user interface (GUI) that allows the user to choose between different scenarios of the wave state.

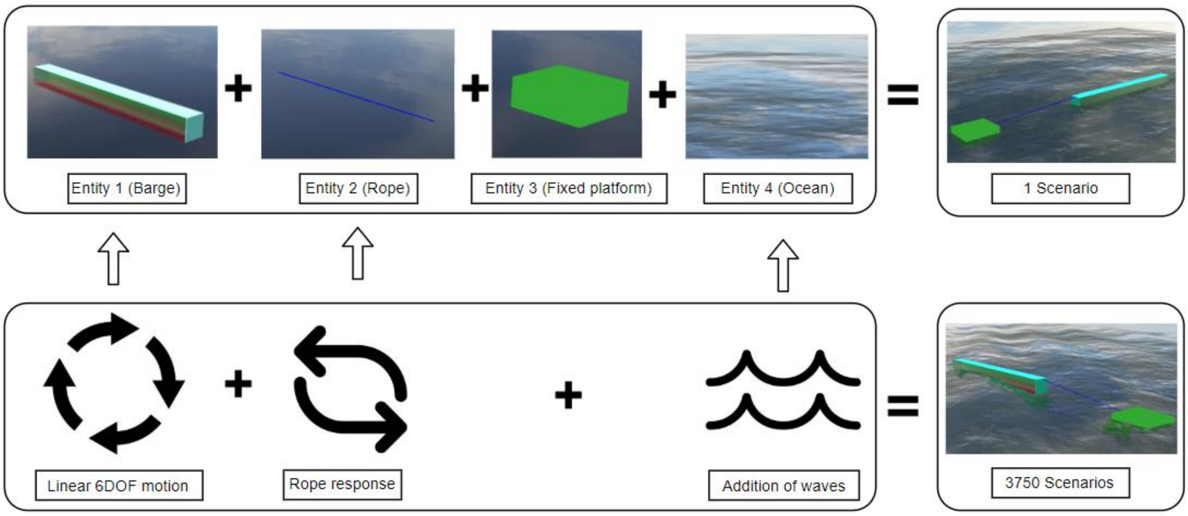


Figure 3.8. Definition of the entity, state and process models for this example.

### *Linear 6DOF motion of the barge*

The process of defining the motion of the barge comes first into defining the entity model, which is the barge itself. This body will be assumed as a 40 meters long barge, 10 meters breadth, and 4 meters depth with 1.5 meters for the draft according to local loading conditions. The entity in Figure 3.9 is sketched in Rhinoceros and meshed with the characteristics described in order to export the geometry to be used in WAMIT as a gdf file.

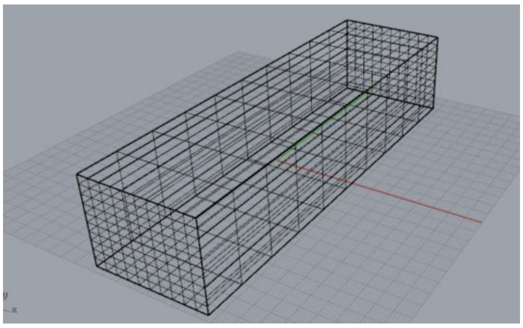


Figure 3.9. GDF file in Rhinoceros

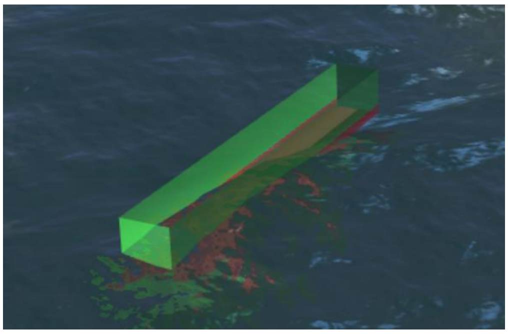


Figure 3.10. 3D entity in the browser with Vessel.js

Radiation and diffraction problems are solved for this run and only the motions are required as output for the selected solver, as only the RAO’s will be needed for the motion and by doing this it cuts computational time. As a part of the process, all the entities will be simulated as a single body in order to obtain the proper behaviour, so the barge is simulated in WAMIT first on its own and then with an external restorative matrix that represents the rope being restrained by the fixed platform. As for the rest of parameters, they are solved with default solvers and configurations for a single vessel. WAMIT is run in the TPN cluster and results are obtained.

The output will be worked simultaneously on Python and on JavaScript, in order to obtain the plots for the motion and to bring the motions to the web browser application respectively. This application is developed by setting up a vessel with the given characteristics. The motion obtained from WAMIT is also introduced to the system through an "Explore file" button and read instantly to convert it in an object of 6 DOF containing 30 periods for each DOF and 25 headings for each period.

Once all the data is collected and ready to use, the model in JavaScript must be defined. The main characteristics for the entity model are previously stated and defined. The visualization for the barge will be done in a 3D box as in Figure 3.10 with the given characteristics, and finally the motions defining the state model will come from the object previously defined.

An evaluation of the motion for the barge is done, in order to see if it matches with the RAO's obtained in Figure 3.11 and if the motion has been transferred correctly to the body (or if the motion looks accurate enough, because some scaling adjustments need to be done).

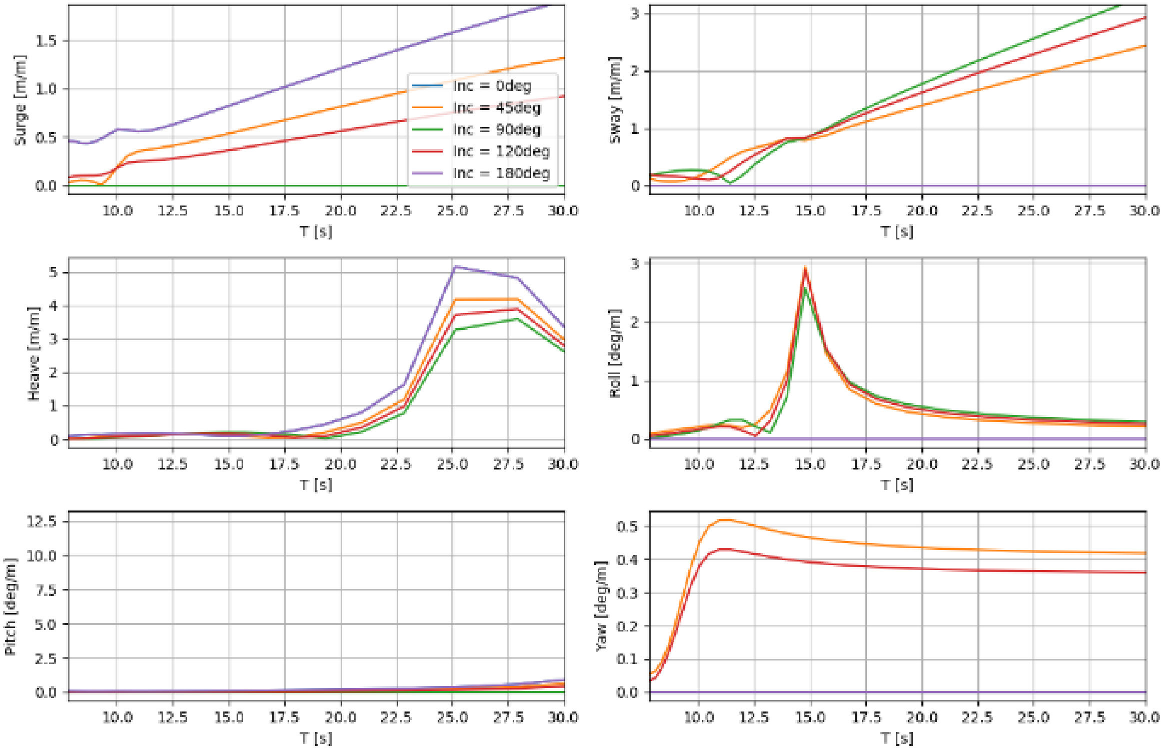


Figure 3.11. RAO for the entity model simulation in 6 DOF.

***Rope response***

As for the rope, the same conditions from the single barge are applied. In this case though, by being tested separately as a single object, does not obtain any motion from any side, as the vessel is the source of movement. So, this entity is considered dependent on other entities for its state mode and so this is what will define the first interaction of a multibody.

The motion of the rope will be set from an initial point. This point will be the attachment point of the rope to the vessel in a steady condition. Then, on top of that initial point, the rope will have to be following the ship motion as explained in Section 3.5.2. With this point changing its position constantly, the rope segments will be switching position following the explanation given in Section 3.4.1. On the other hand, the interaction with the fixed body is a static condition, so this part simplifies the problem considerably.

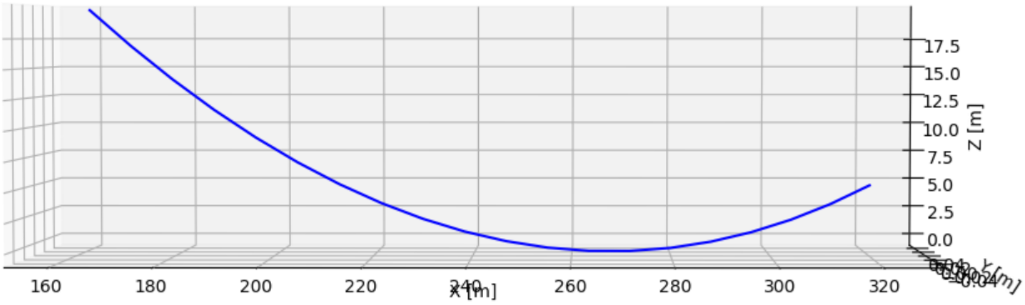
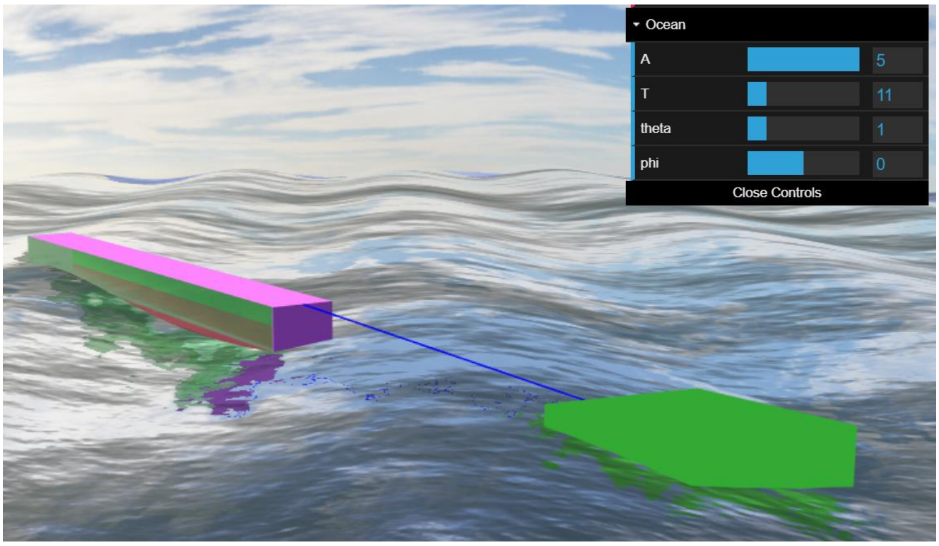


Figure 3.12. 3D visualization of the rope entity in MAP++

***Simulation of multibody***

Joining all three entities together, the state model of the simple barge will be considering the external stimuli obtained by the rope, that will be constraining the motion. There is not a consideration for internal stimuli at this stage of the work, such as tank fullness, etc. These external stimuli are considered only on the pre-processing data run in WAMIT as a restorative matrix that will be obtained through the MAP++ software (Figure 3.12). Going back to Figure 3.8 all the state models are put together when representing the multibody scenario.



▾ Ocean

A 5

T 11

theta 1

phi 0

Figure 3.13. Multibody simulation for this example

Once all the interactions are defined and all the motions prepared, a case study is created by joining all the pieces together to form the multibody simulation with a good visualization as in Figure 3.13. In addition to the visualization, the RAO plots are obtained through Python in order to be able to compare the plots with the ones obtained earlier with the single barge. These RAO plots representing the multibody simulations are displayed in Figure 3.14.

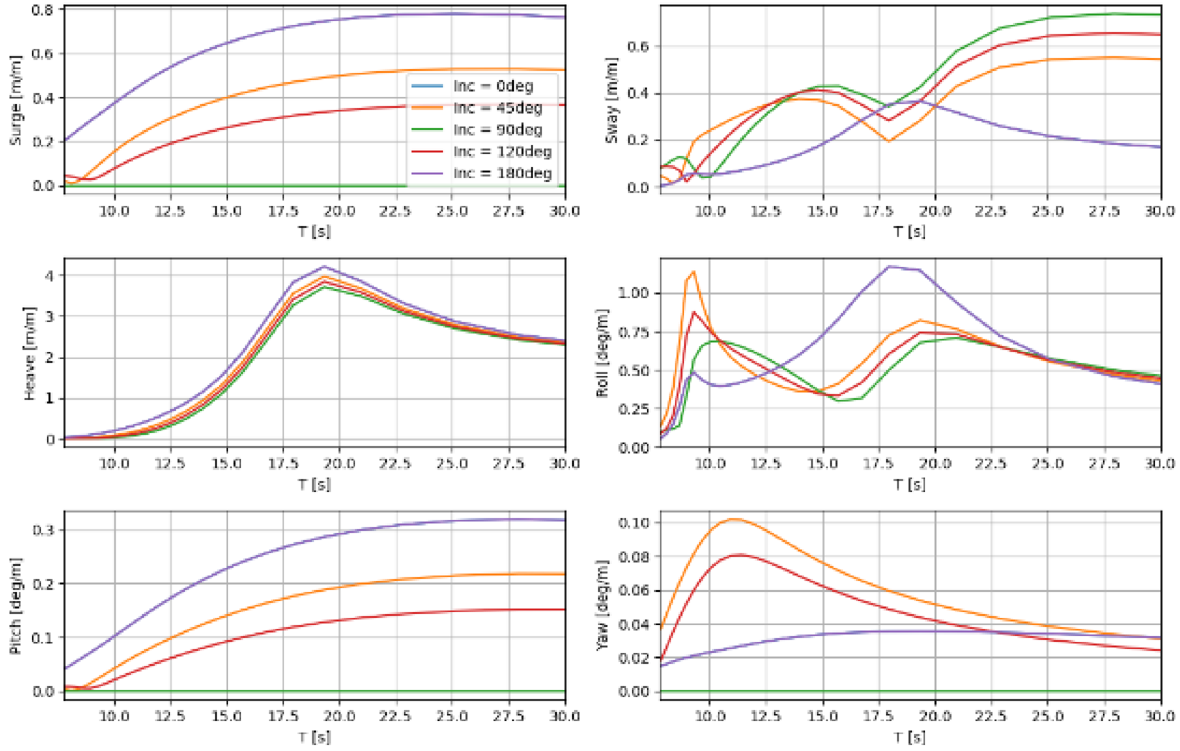
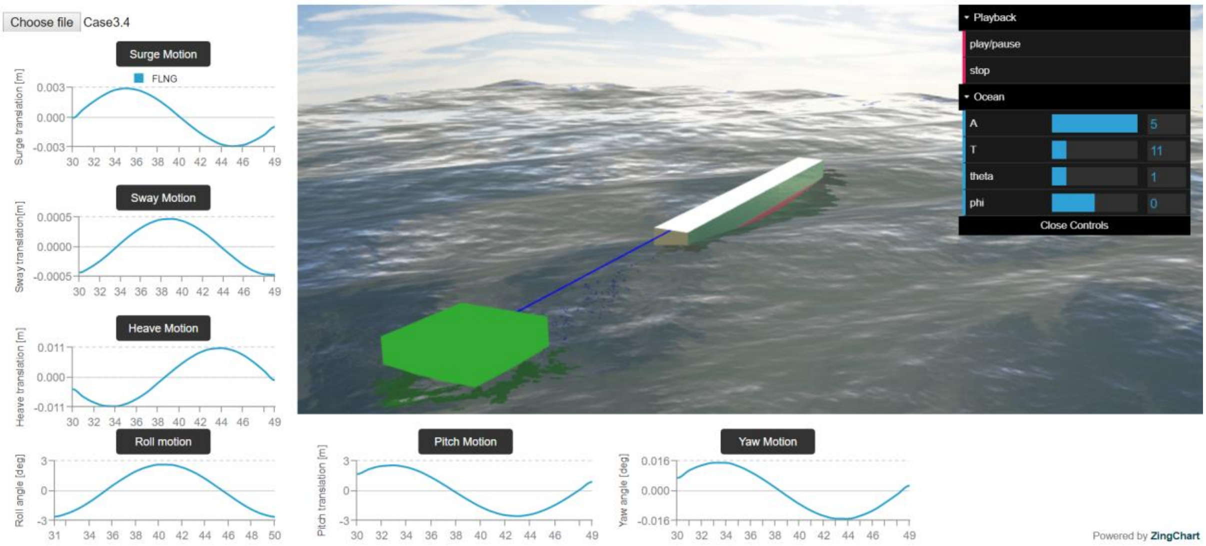


Figure 3.14. RAO for the multibody simulation in 6 DOF.

By comparing RAOs from Figure 3.11 to Figure 3.14, one can see that for the single body, surge and sway tend to keep growing with the incident wave period. However, as seen in the multibody simulation now the surge and sway motions are restrained by the fixed platform and the rope.

The output parameters of the simulation can be checked through the web-browser, besides judging the visual simulation, whether the values obtained make sense or not. Another option for this console is to print the value of a variable and it will be changing over time, so it is possible to check the results.

However, when using a simulator, the user expects a more user friendly interface and they are expecting, in addition to this physics data happening behind the scenes, a visual idea of what is happening, in order to corroborate the credibility of the software and obtain some information to share and work with. For this purpose, several plots can be obtained from the motion data from the results calculated instantly in the browser with the aid of some external JavaScript libraries such as ZingChart, as seen in Figure 3.15 for some of the motions. This will provide plots of the motions in real time. These will represent the amplitude for every degree of freedom in relation to the neutral position over time.


Choose file Case3.4

Playback: play/pause, stop

Ocean

A 5

T 11

theta 1

phi 0

Close Controls

Figure 3.15. Addition of real time motion plots to the example.

In addition to the RAO motions, the tension on the rope is also plotted to obtain this magnitude over time. These are not obtained in real-time and due to the several possible combinations, a harsh sea configuration will be pre-set in order to simulate the tension in the rope in an unfavourable situation. A wave amplitude of 5 m, wave period of 20 seconds and wave heading of 45$^o$ are taken as an example to plot Figure 3.16.

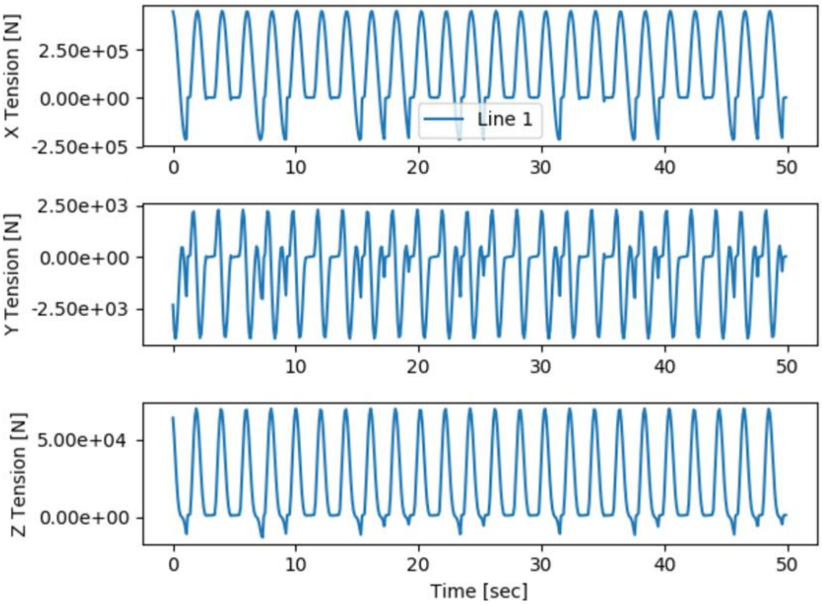


Figure 3.16. Tension of the line in every dimension calculated with MAP++

The motion of the vessel is fed into the software MAP++ and the position has been established as well as the line properties in order to obtain the total tension. Finally, the module of the tension is plotted in Figure 3.17.

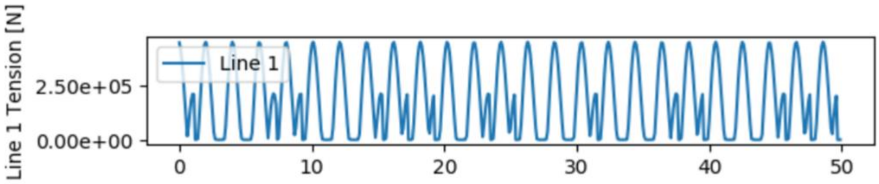


Figure 3.17. Module of the tension obtained.

# Chapter 4

## 4 OCEAN SPACE SIMULATIONS

The ocean space is comprised of multiple bodies with infinite possible configurations. By following the methodology detailed in Section 3, a progression of marine applications going from a very basic model to more complex examples is followed as in Figure 4.1. Simulations are carried in a web-application able to provide some output of interest after the user sets some initial conditions. The different case studies and their results are described in detail in the following paragraphs.

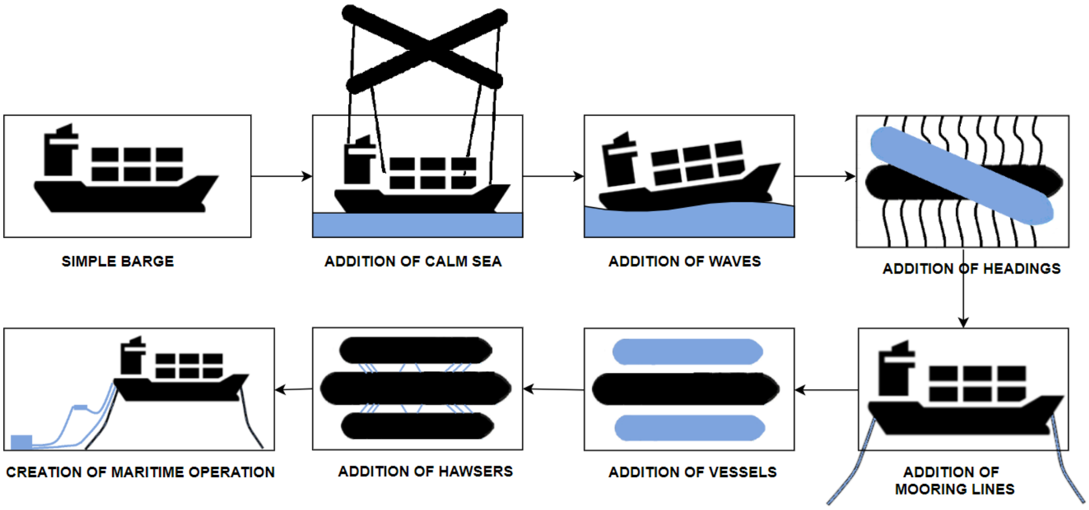


Figure 4.1. Progression of marine applications developed on this study

Also, due to the need to interpret the WAMIT output in a better way, this thesis is developed in order to give the user an easy transition from unreadable text files to visual simulations in a web browser. An interface has been built in a way that the user can choose how many ships to simulate (from one to three), if these are moored through a catenary line at the bottom, and in the case that there were two or more ships, if hawsers are added or the ships remain free.

For market interests, the main simulations will be carried with an FLNG tanker when displaying only one ship, and some additional Suezmax vessels when simulating an offloading side-by-side operation. There are many operations that do not require mooring on land and it is often much better to do it on open in order to avoid going to a port, at the same time that one saves up costs and shortens commercial routes.

Considering that the calculations for the motion are not instantly carried in the browser but in an external environment with pre-loaded files, it is difficult that future applications can resize the vessels on the web-based simulation. Therefore, common commercial characteristics for these vessels have been picked and are shown in Table 4.1.

Table 4.1. Vessels characteristics for FLNG and Suezmax

| **FLNG** | |
|---|---|
| **Length overall** | 316.5 m |
| **Breadth** | 29 m |
| **Depth** | 23 m |
| **Draft** | 8 m |
| **Displacement** | 290.000 T |
| **Loading condition** | Ballast |

| **Suezmax** | |
|---|---|
| **Length overall** | 264.75 m |
| **Breadth** | 24 m |
| **Depth** | 18 m |
| **Draft** | 5 m |
| **Displacement** | 150.000 T |
| **Loading condition** | Ballast |

As for the number of cases developed in this study, the FLNG will have the option to be moored or free, as well as attached to a second and third vessel or detached. The Suezmax vessels will not be moored but only attached to the FLNG. Therefore, the FLNG will have a sum of restoring matrices as a consequence of the mooring lines and hawsers coming from other ships, and for the Suezmax vessels, only restoring forces coming from the hawsers attached to the FLNG are allowed as will be seen in Section 4.6.

## 4.1 CASE 1: SIMULATION OF THE RADIATION PROBLEM

The first case considered will be dealing with the radiation problem. This will go through all the process of setting the radiation problem onto the simplest scenario with the least entities (a barge and the ocean). This will provide real time graphs that will tell the user what the displacement of the motion is in every time step.

For starters, a breakdown of the scenario to simulate is necessary. As shown in Figure 4.2, the entities included in this simple simulation will be just a box-shaped barge. The analysis that will be performed upon this entity will be the linear 6 degrees of freedom response for finally obtaining this single scenario of a barge responding to several different initial states of motion.

Additionally, in order to place this box somewhere in the space, the ocean entity is placed cutting the draft line of the barge. This entity is merely a flat plane resembling a still water condition. The problem in this case requires the sea to produce some excitation waves when the barge moves. However, these will not be displayed in the visualization phase due to high complexity and incompatibility with the current ocean entity in Vessel.js.

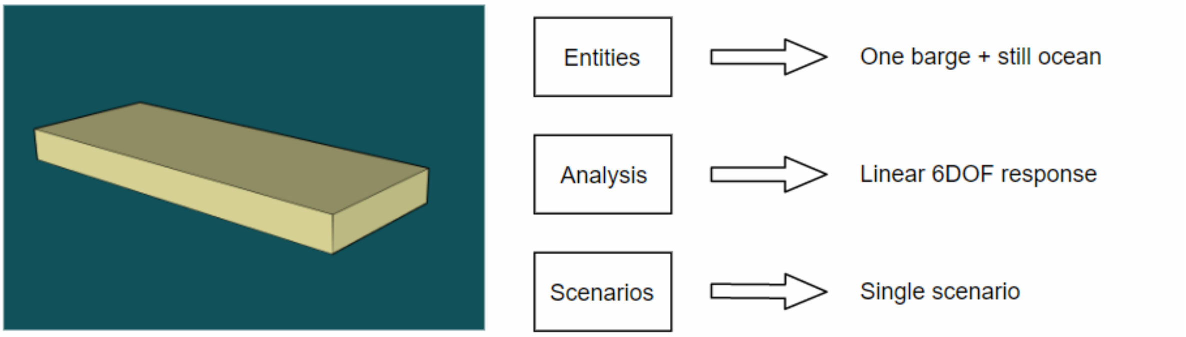


Figure 4.2. Description of the model with the entities, analysis and scenarios

This first case will be an attempt to introduce WAMIT upon the Vessel.js library. This is a transition from the actual library to a more advanced method to calculate the ship motions. The example of a dynamically moved barge will be taken from the Vessel.js library and the hydrodynamic coefficients will be calculated in WAMIT instead of with empirical formulas and by asking the user to input some damping parameters which are not included by default.

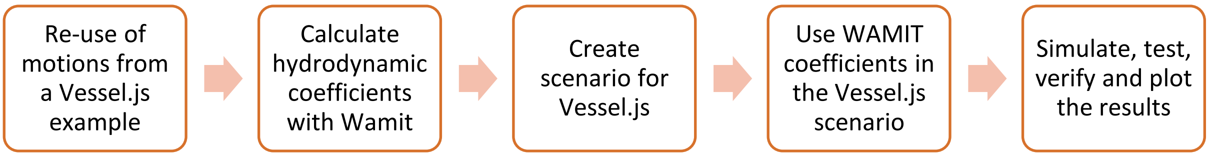
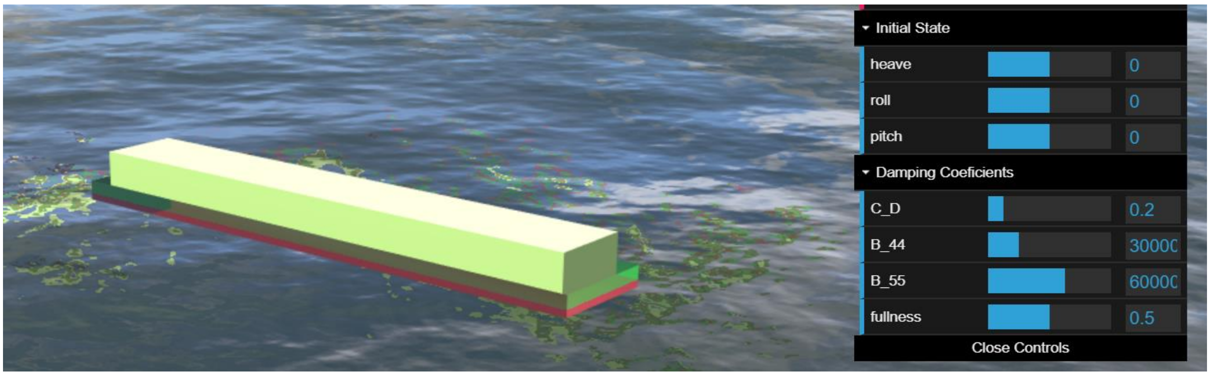

Re-use of motions from a Vessel.js example
Calculate hydrodynamic coefficients with Wamit
Create scenario for Vessel.js
Use WAMIT coefficients in the Vessel.js scenario
Simulate, test, verify and plot the results

Figure 4.3. Process to create the current case study.

### 4.1.1 Reuse of motions from Vessel.js library

This case will be of a simple barge in calm water and a disturbance will be applied following the radiation problem. WAMIT cannot provide equations of motion that do not exist and an interface with all kinds of disturbances possible would be impracticable. Therefore, a current example from the Vessel.js library of a vessel that follows motions obtained with differential equations based on the ones exposed in Section 2.1.2, has been used for this simple example (Ferrari et al., 2018).

This example has a drawback and it is that some restoring and damping coefficients need to be introduced manually by the user as seen in Figure 4.4, whom sometimes does not know. Also, this example is fed with empirical mass matrix, added mass, damping and restoring coefficients. The contribution with this case study will be to substitute these with the coefficients obtained with a WAMIT run resembling the radiation problem.


Initial State

heave 0

roll 0

pitch 0

Damping Coeficients

C_D 0.2

B_44 30000

B_55 60000

fullness 0.5

Figure 4.4. Dynamical motion example in Vessel.js

The forced motion over the body will create waves which will derive into wave forces as well as inertial forces. This will be considered when calculating the added mass and damping coefficients in WAMIT, as it will affect the motion of the barge. By using the example from Vessel.js library, one can set an initial state in three different degrees of freedom, so a motion can be provided according to the user needs.

In order to carry a radiation problem in WAMIT there is an option that allows the user to choose which degrees of freedom will be examined for this problem. On the contrary, the diffraction problem has to be set null, which lets WAMIT know the desire of only running radiation problem. It is important to set the number of periods and wave heading to zero in order to ensure a calm sea. Also, following the depth configured in the simulator, this will be of 55 meters. For the force file (FRC) the files to receive should be equal to zero for file 2 to 4, as linear hydrodynamic coefficients that deal with the radiation problem such as added-mass and damping are obtained from an output file called “Force.1”.

The output from the WAMIT file Force.1, is added manually to the arrays with the added-mass and damping coefficients, which follow the methodology explained in Section 2.1.2. But this time with real hydrodynamic coefficients that will allow to build a good added mas, damping and restoring matrices. This is the only time in this study that motion calculations will be performed in real-time on the browser depending on the user’s input. This can be done by using the numeric.js library that helps solving differential equations.

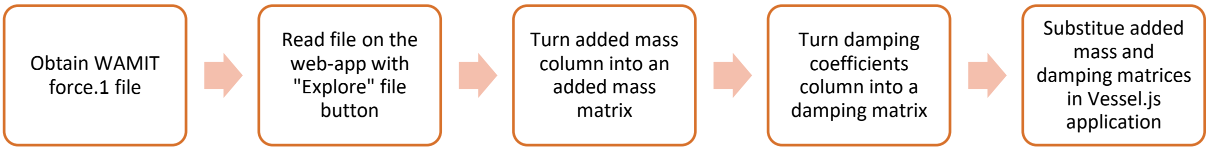
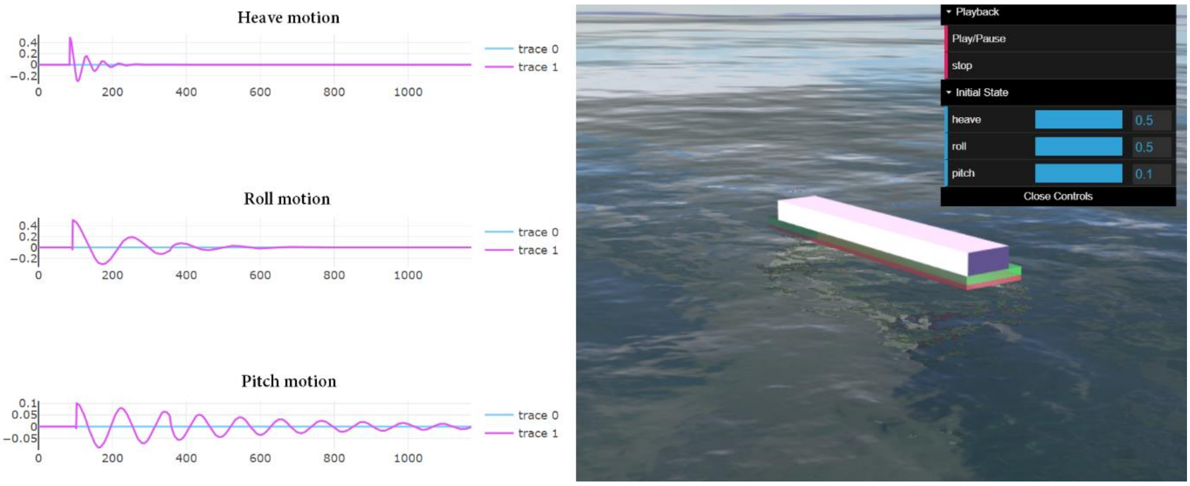

Obtain WAMIT force.1 file
Read file on the web-app with "Explore" file button
Turn added mass column into an added mass matrix
Turn damping coefficients column into a damping matrix
Substitue added mass and damping matrices in Vessel.js application

Figure 4.5. Process to bring the motion from a WAMIT output file to the web-browser application.

### 4.1.2 Obtention of the results

In this case, a continuous observation is carried in the web browser console in order check the real time motion and compare evaluate wheter the results might be valid or not. These are heave, roll and pitch. By changing the input on the GUI, new plot situations are displayed over time. This interactivity brings insight that is difficult to transmit in paper but through a web application is easy to describe.

To test an example, initial conditions for heave, roll and pitch are given suddenly to a Mississippi barge in order to see how the motion behaves with the damping coefficients and added mass provided for this case. As seen in Figure 4.6, and as expected, the motion starts when the initial state is set with the sliders. This can be seen as an abrupt increase of the motions in the plots. Then the motion is damped as fast as the damping coefficients allow. For instance, the damping of the pitch motion takes longer than the damping of the heave motion



heave 0.5
roll 0.5
pitch 0.1

Figure 4.6. Vessel.js representation of the radiation problem case with real-time plots for heave, roll and pitch.

The motion in this case can be considered to be more accurate than the example on the Vessel.js library, where the user can introduce some damping and restoring coefficients manually with the help of some sliders. These coefficients normally depend on several factors, some of them involving the ship hydrodynamics and stability, so they are not as precise as should if the user estimates them.

## 4.2 CASE 2. WAMIT MOTIONS AND WAVE ADDITION

This second case will present the entity barge placed on the entity ocean, but now there will be an incoming regular wave which will be expected to give some motion to the barge. This will be plotted in order to see and evaluate the results for this motion and compare with other studies to see if the motion represented is right.

The previous case is taken as a base in order to build up and introduce new entities and analysis to the simulations. This time, as seen in Figure 4.7, the analysis is focused onto adding incoming waves from one single heading to the ocean entity. This will also force the simulation to analyse the barge entity according to these waves. However, this time the linear 6DOF responses for the barge are entirely obtained from WAMIT and translated to the web-application. The expected output for this simulation is for 150 different scenarios, as the user can choose among 5 amplitudes and 30 different wave periods.

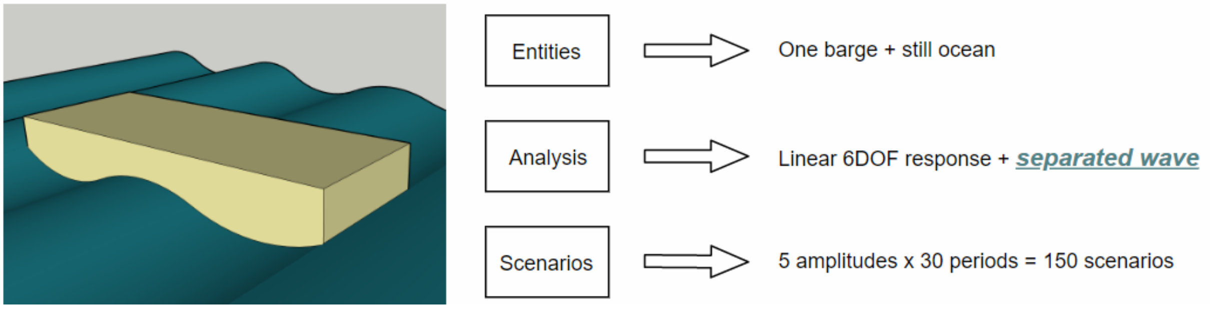

Entities
One barge + still ocean
Analysis
Linear 6DOF response + separated wave
Scenarios
5 amplitudes x 30 periods = 150 scenarios

Figure 4.7. Description of the model with its entities, analysis and scenarios. Addition of waves

This case is the first app run entirely with WAMIT motions, without depending of external ways to calculate the ship motions. In order to calculate this response in WAMIT, a single heading of 150 degrees is chosen, which is one giving a good response for visualization purposes. The radiation and diffraction problems will both be activated when running the WAMIT software.

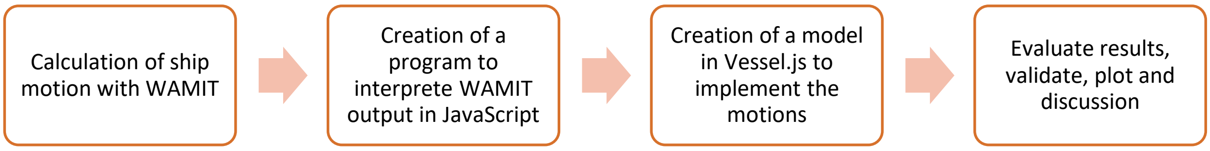
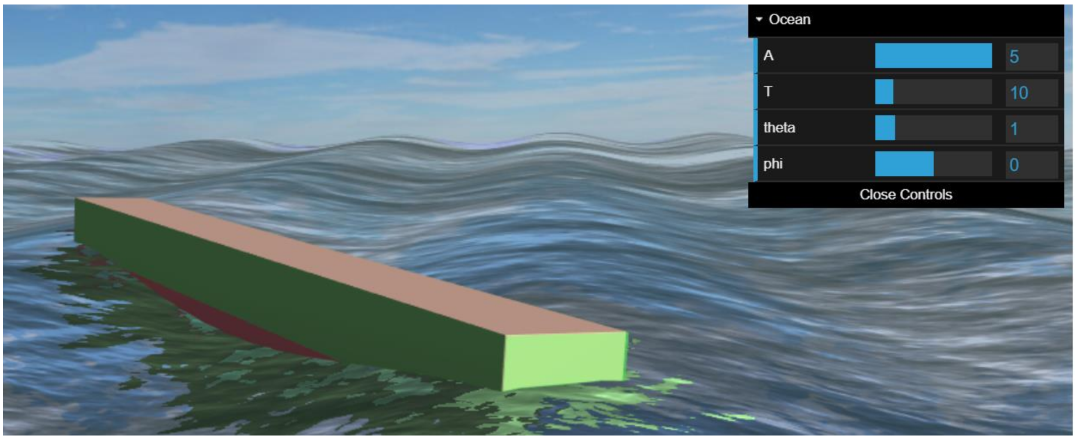

Calculation of ship motion with WAMIT
Creation of a program to interprete WAMIT output in JavaScript
Creation of a model in Vessel.js to implement the motions
Evaluate results, validate, plot and discussion

Figure 4.8. Process for the creation of the current case.

For this simulation a library is used to display the regular waves. This will allow the simulation to identify whether the motion of the ship is correct or not. This will also give the chance to have a graphical user interface that will help the user to choose the desired wave period (0-30 seconds) as well as the wave amplitude (0-5 meters). In this case, the phase will come already defined by WAMIT, so it is not necessary for the user to introduce it. The library provides a phase slider that changed the phase of the waves, but no the phase of the vessel. Even when this slider is used, the motion is displayed correctly but both the crests and valleys of the waves do not match the motion anymore. So, this slider, although provided by the library, does not add any insight on this thesis.


Ocean

A 5

T 10

theta 1

phi 0

Close Controls

Figure 4.9. Visualization of this case with one given heading for all the possible periods and amplitudes.

When the background is defined, the vessel with the given specifications is created and put into the water with the corresponding draft. The motions obtained from WAMIT will be read from the created matrix and the ship will start behaving accordingly.

Once the simulation is done, the RAO plots are obtained from a Python script that is able to interpret the WAMIT output and turn it into easily readable data. This will provide a plot with a single line as seen in Figure 4.10, which identifies the single heading of 150 degrees. This way it is possible to check if the motions obtained make sense and are following the WAMIT output. If it was not the case, we should check the interface that turns WAMIT output files into ready-use motion arrays.

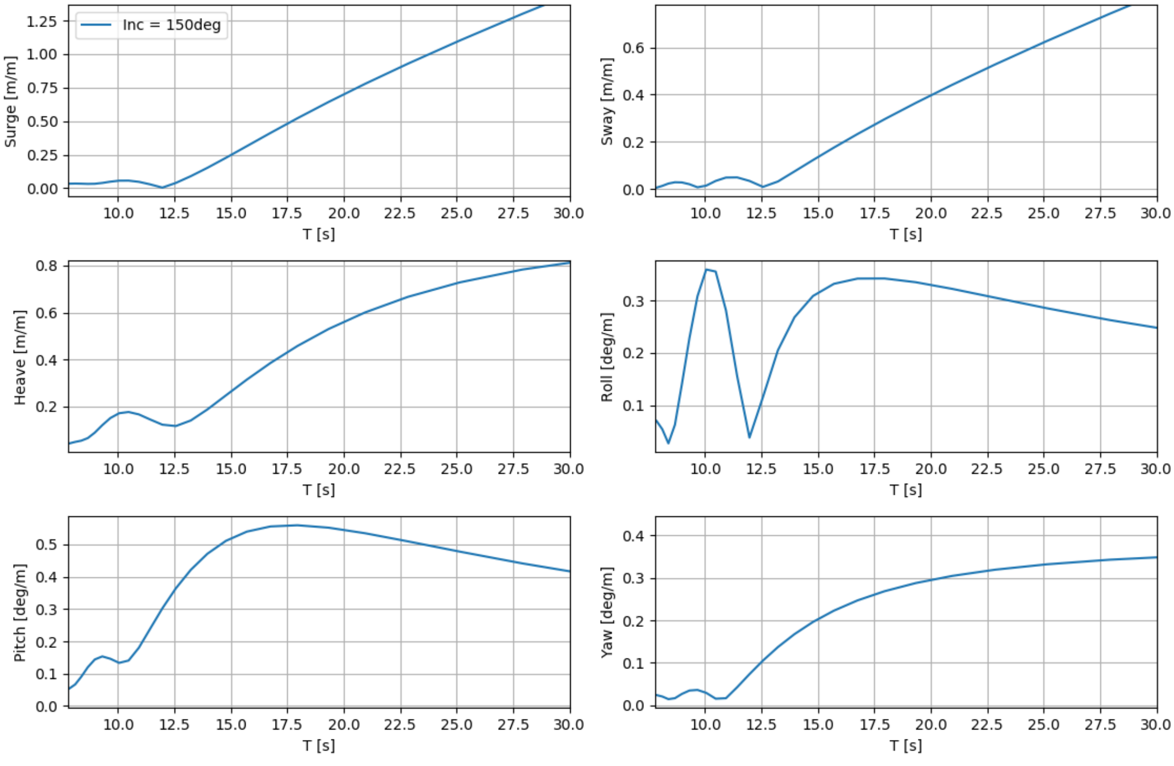


Figure 4.10. RAO amplitude for the studied case in all six degrees of freedom.

### 4.2.1 Comparison with other studies

Hu et alt (2016) experimented with the motions of a FLNG with similar dimensions to the ones analysed in this study, but with the exception that the model used by Hu was almost twice as wide as the model studied on this study. The researchers also compared the experimental study with the motions obtained in SESAM as one can see in Figure 4.11.

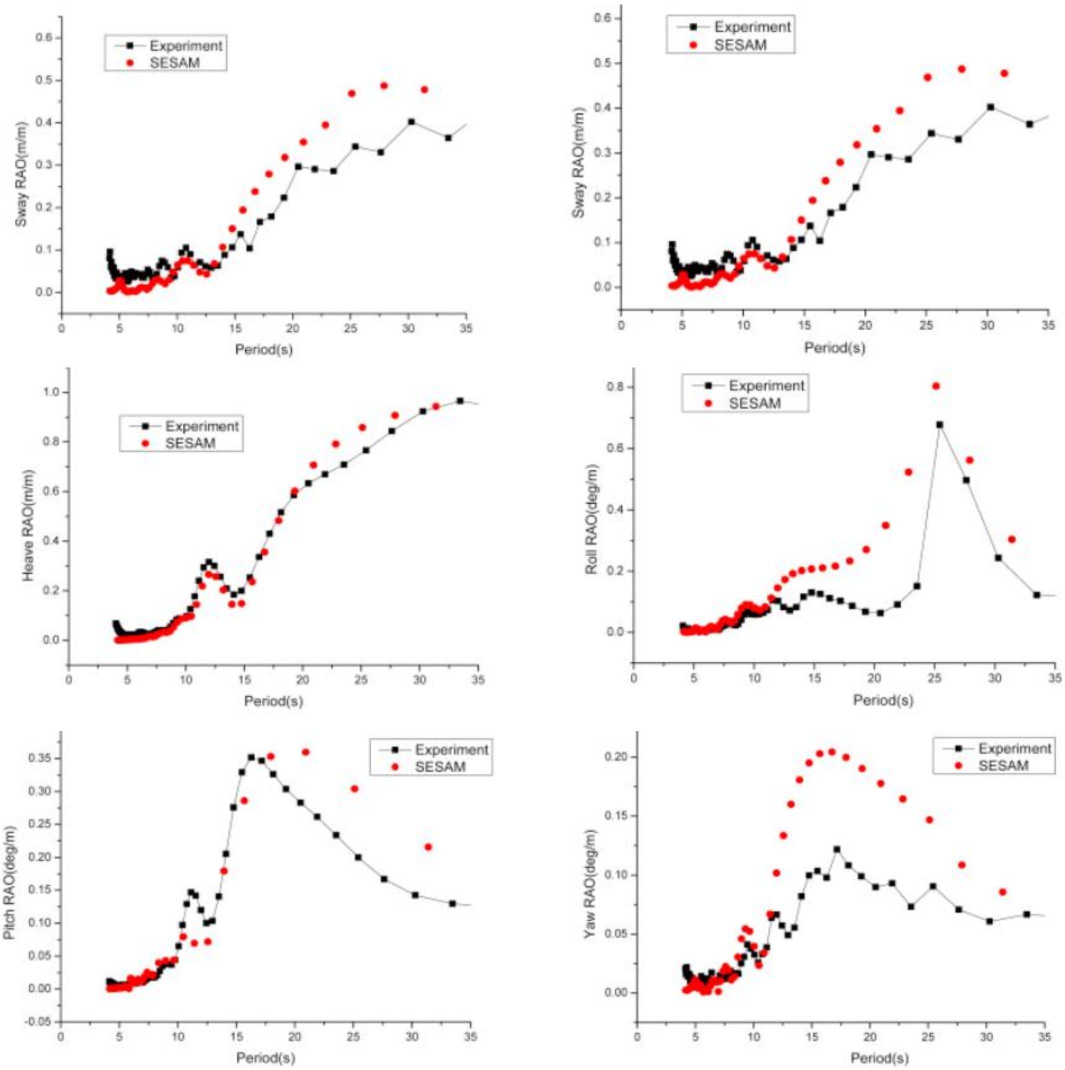


Figure 4.11. RAO amplitude plots for 6 degrees of freedom of an FLNG with similar characteristics (Hu et alt., 2016)

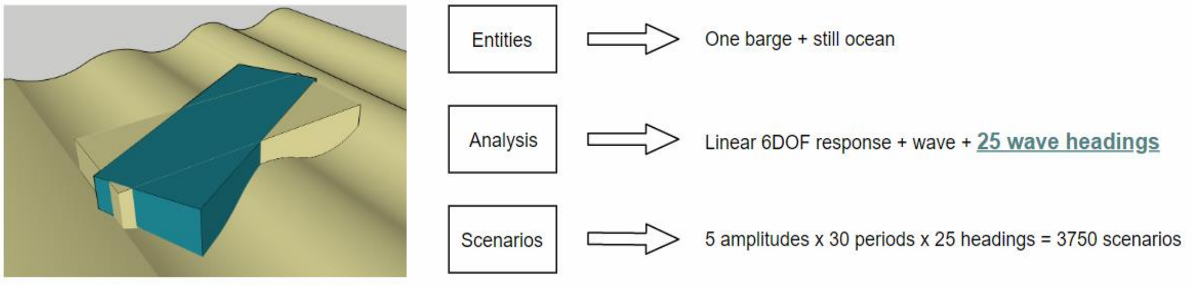
In order to validate the motion data obtained in this Case 2, a comparison between the SESAM results of this study (Figure 4.11) and the results obtained in this case (Figure 4.10) is made. The same conditions as in the study being compared are taken when simulating the current model, with 150 degrees wave heading, around 30 periods and a 25% loading condition.

Looking at the results, the fact that the model from the experiment is wider, still produces similar results but as it is expected, the roll motion for the simulation in the current case is more pronounced for lower periods, which is not the case in the study made by Hu. The same applies with less severity with the pitch. As for the other degrees of freedom, the amplitudes are quite close to the values obtained in WAMIT.

## 4.3 Case 3. Addition of different wave headings

Building up from Case 2, this will add to the ocean entity the capacity to analyse the simulation with 25 different wave headings. It is expected to obtain the same results as in the previous case but in this case with several headings. 25 heading have been chosen in order to represent a heading every 15º until a 360º circle has been completed.

In order to do that in WAMIT, it is enough to run the same simulation as run in the previous case but adding the 24 remaining wave headings to be able analyse the barge entity with them. All degrees of freedom are chosen to be calculated, and only Force.4 output file is demanded, which will provide the motion in a RAO form.


Entities → One barge + still ocean

Analysis → Linear 6DOF response + wave + 25 wave headings

Scenarios → 5 amplitudes x 30 periods x 25 headings = 3750 scenarios

Figure 4.12. Description of the model with its entities, analysis and scenarios. Addition of 25 headings.

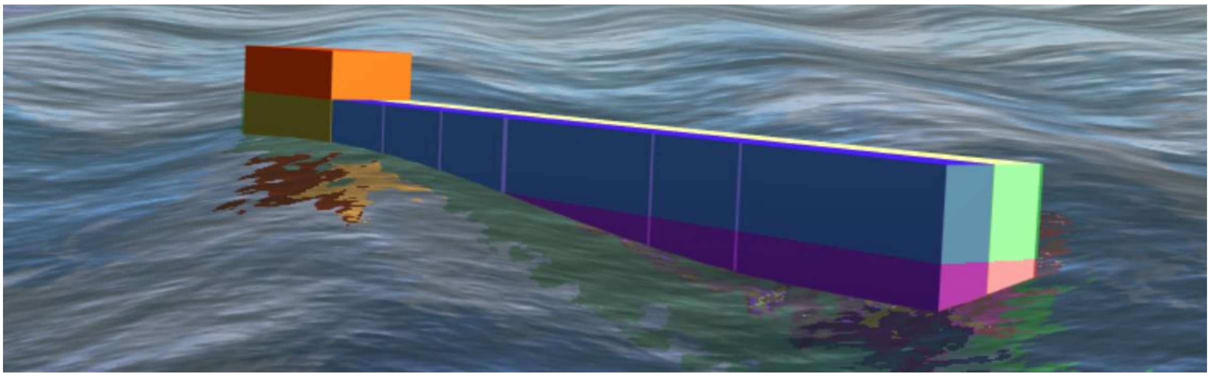


Figure 4.13. Visualization of this case in a regular sea with a given wave amplitude, period and heading.

For the web-based simulation, the "Regular sea" library in Vessel.js already has a build-up option that lets the user choose the different headings through the same GUI used to choose the amplitude and the wave periods. With these the results obtained are the same as in the previous case, but with more headings in different colours according to the legend in Figure 4.14.

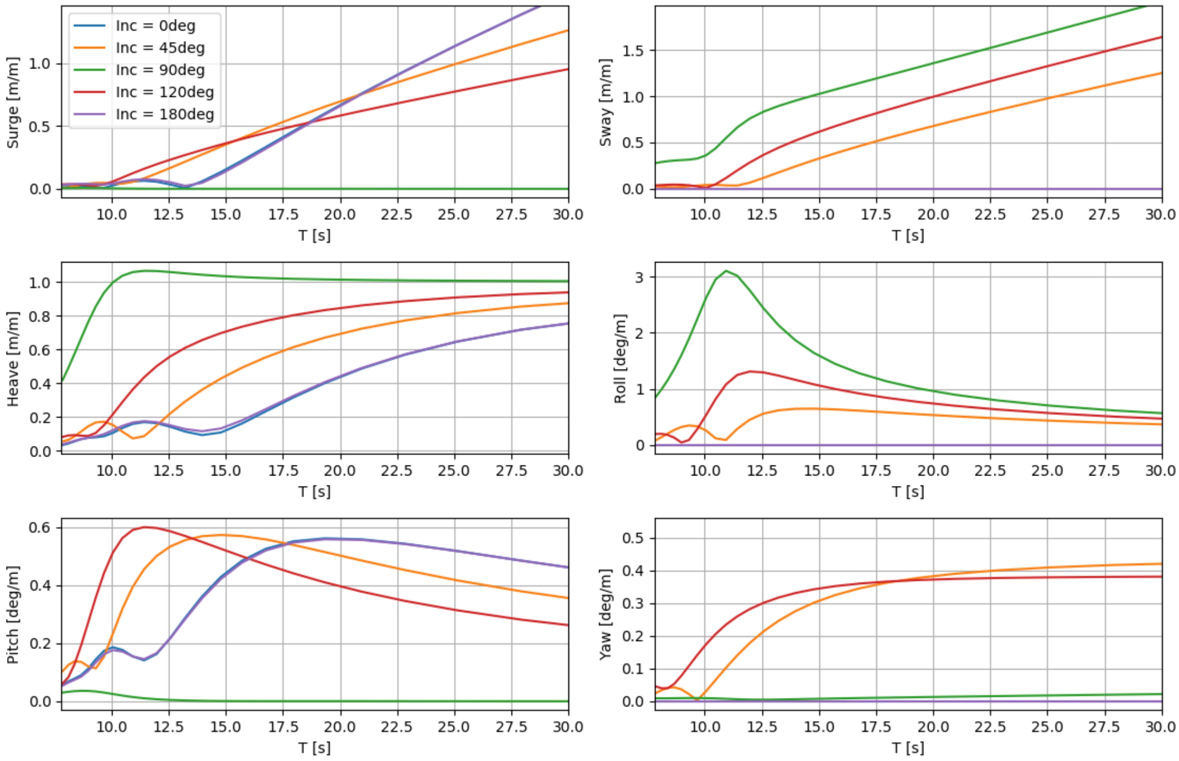
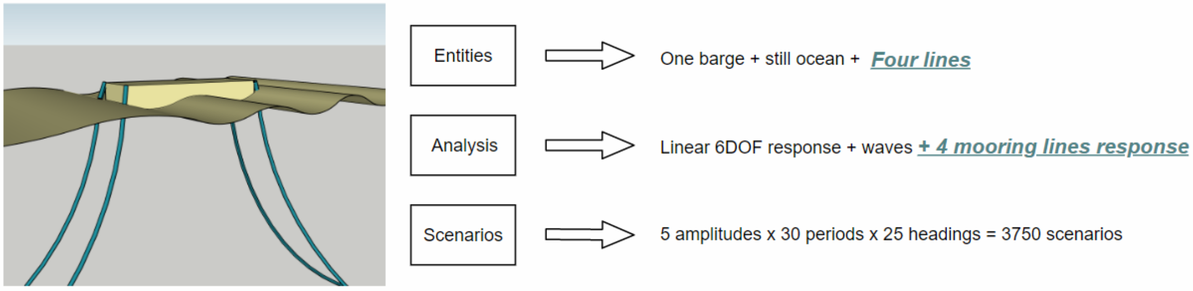


Figure 4.14. RAO plots for the FLNG with various headings

## 4.4 CASE 4. ADDITION OF MOORING LINES

This will be the first case that will include simulation of multibody in the form of a mooring line, as seen in Figure 4.15. These are known to restrain the motion of the ship in order to maintain position and in some case to carry different kinds of maritime operations. Some of them so precise that the use of taut mooring line is used in order to have an even more steady vessel.



Entities → One barge + still ocean + ***Four lines***

Analysis → Linear 6DOF response + waves ***+ 4 mooring lines response***

Scenarios → 5 amplitudes x 30 periods x 25 headings = 3750 scenarios

Figure 4.15. Model definition by its entities, analysis and scenarios.

The results for this case are expected to be different than from the previous case, as the motions, especially the ones acting in the ocean plane and the heave will be restrained and damped. In addition to the previous case, this one will have four extra entities in the form of mooring lines which will be restraining the motion of the barge entity.

When defining the rope entity in JavaScript, this will have to be divided into several segments in order to be able to represent the catenary effect as defined in Section 3.4.1. This means that the motion coming from the vessel will force the rope segments to take different positions which will be defined as a new state for every time step. On the other end of the rope there will be a fixed constrain, as it is supposed to be anchored to the seabed. This also forces to introduce the seabed as a new entity for the simulation.

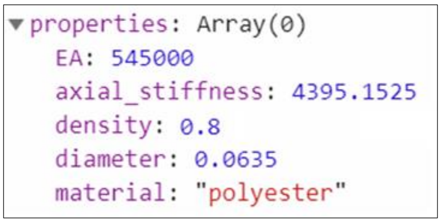
For this simulation, the obtention of the restoring matrices is the first step to take. However, as mentioned previously, there is no easy way to obtain these matrices and this is why MAP++ is used to simulate the behaviour of the mooring lines. This analysis is made only once and then the restorative matrices obtained are used in all the simulations where mooring lines are required. The input for the MAP++ is the anchoring position on the seabed and the anchoring points on the ship as seen in **Appendix C**. Then, these matrices will be combined accordingly to each situation.

### 4.4.1 External restorative matrices

The simulation in WAMIT is run the same way as in the previous case, but now the ship has an additional linear external matrix that needs to be introduced in the ship WAMIT input file. Once done this, the analysis will take care of the matrix as if it were four mooring lines with the specified data from Table 4.1. These parameters will have to match the browser simulation by giving the rope the adequate properties. Each one of these ropes can be given different characteristics that can be used later in order to calculate tensions, elongation and fatigue.  In order to give these characteristics to the entities created on the web simulator, the same properties are defined as an object inside each line entity.

Table 4.2. Characteristics used to calculate the restoring matrix

| **Mooring line properties** | |
|---|---|
| Material | Polyester |
| Density | 0.8 Kg/m |
| Diameter (D) | 0.0635 meters |
| Elastic modulus | 126 kN |
| Axial stiffness | 1090000 x D$^2$ |



```
▼properties: Array(0)
   EA: 545000
   axial_stiffness: 4395.1525
   density: 0.8
   diameter: 0.0635
   material: "polyester"
```

Figure 4.16. Object in JavaScript representing the characteristics of the mooring lines

The restoring external matrix will be calculated with MAP++ and then introduced to WAMIT. As the cases studied in this thesis will all have the same vessels with same dimensions, draft and loading conditions, one calculation will be enough for the mooring lines, which will always be attached to the FLNG (always with the same characteristics) and the hawsers will only interact between the FLNG and the Suezmax, no matter the number of vessels. If more vessels are added, the restorative matrix will be modified accordingly.

The distribution of the mooring lines in the simulation, will be done by reusing an example in Vessel.js dealing with mooring lines (Ferrari, 2018). These mooring lines will be acting like a resting catenary line. This means that the mooring line performs a certain catenary effect due to its weight while some of it rests on the seabed, so the vessel has a certain ability to move freely in a radius given by this line. The start points of the lines are defined at the anchoring point of the vessel (the four corners of the barge) with the same characteristics that will be entered in the Vessel.js application (*mooringAngle* for the angle measured to the z axis with which the mooring line is anchored to the seabed, *radialDistance* to determine how far from the boat the anchoring point on the seabed is and *anchorLenght* in order to define the length of the mooring line). Once defined all the characteristics, MAP++ is run in order to obtain the restoring matrices needed shown below.

Mooring external restorative matrix

$$\begin{bmatrix} 8.9e{+}04 & 0 & 0 & 0 & 7.5e{+}06 & 0 \\ 0 & 2.0e{+}05 & 0 & -6.1e{+}05 & 0 & 0 \\ 0 & 0 & 2.2e{+}04 & 0 & 0 & 0 \\ 0 & -6.1e{+}05 & 0 & 7.4e{+}06 & 0 & 0 \\ 7.5e{+}06 & 0 & 0 & 0 & 9.6e{+}08 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.1e{+}08 \end{bmatrix}$$

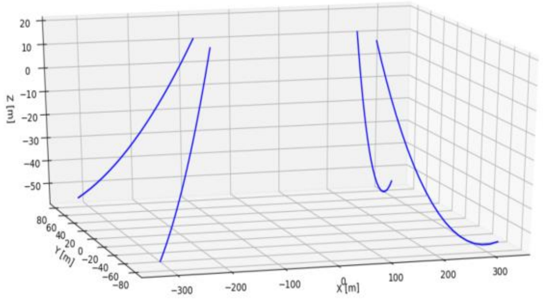


Figure 4.17. Visualization of the mooring lines in MAP++

Once obtained the restoring matrices, a new run can be done in WAMIT and the same process as in Case 3 is followed for visualization purposes, but this time with a motion that corresponds to the moored vessel. In order to display the mooring lines, an application developed in Vessel.js is reused (Ferrari, 2018), which follows the mooring line theory stated in Section 2.1.3. The result can be seen in Figure 4.18.

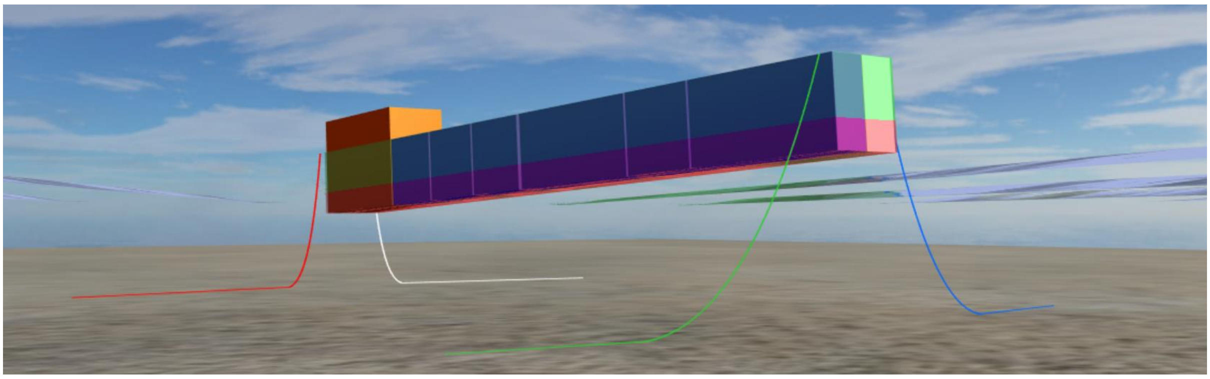
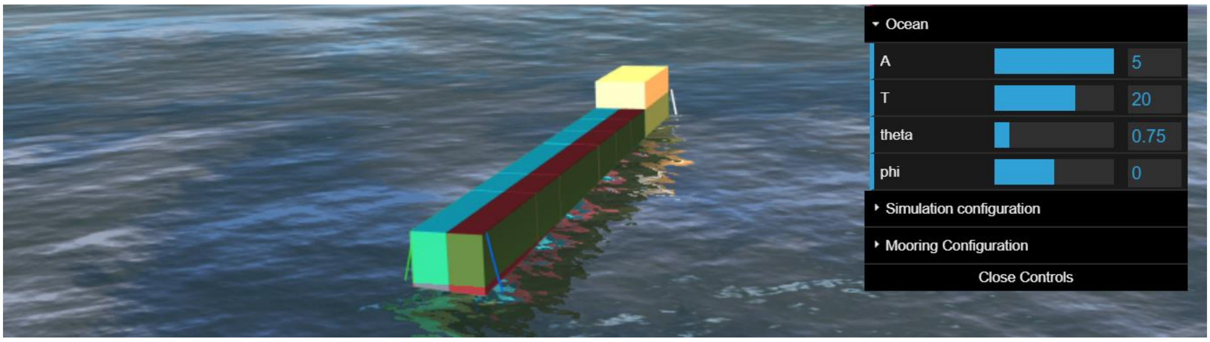


Figure 4.18. Visualisation of the case in Vessel.js

By comparing the RAOs obtained from the results in this case with the ones from the previous case without the mooring lines (Appendix B), there is an attenuation of the amplitude for the ship motions, which means that the mooring lines are restraining the ship that normally responds to larger motions.

### 4.4.2 Estimation of the tension in the mooring lines

The mooring lines are exposed to fatigue for their constant dynamic movement. In this section an example of a specific harsh sea condition is shown in order to analyze the tension results before bringing a real-time application to the web browser. This is just a simple example of a given condition, as plotting the tensions for the four mooring lines with all the different periods, headings and amplitudes results in thousands of different possibilities. Then, a large amplitude of 5 m is set, with a period of 20 seconds and a wave heading of 45º as in Figure 4.19.



A 5

T 20

theta 0.75

phi 0

Figure 4.19. Case example for calculating the tension on the mooring lines

The tensions are calculated with MAP++, the same software providing the mooring lines which runs through a Python script. The software uses the input given by the user (wave conditions specified before) and the theory also mentiod in order to calculate the tension. Considering this input, the software will plot the amplitude of the motions over time as seen in Figure 4.20.

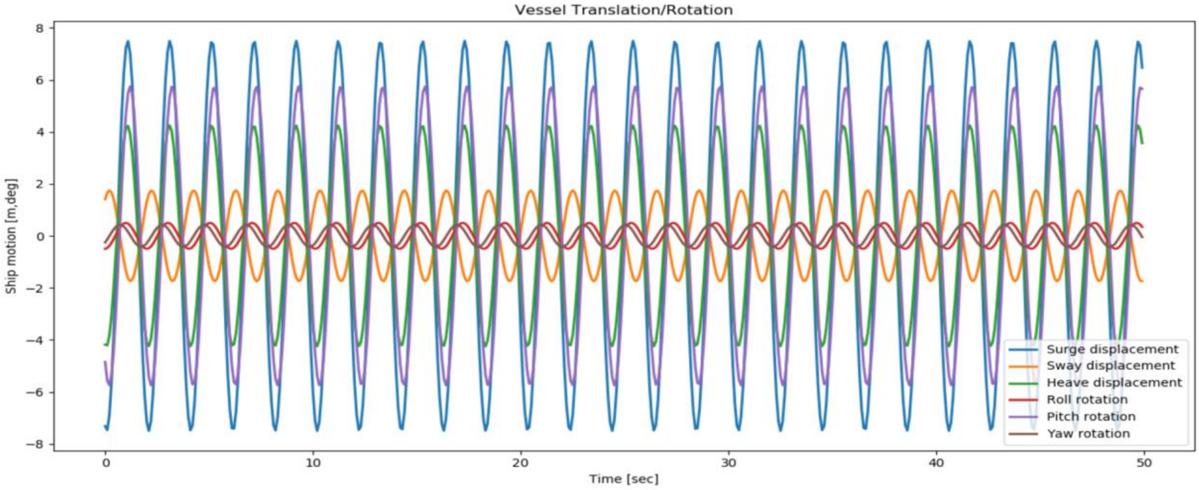


Figure 4.20. Ship motion for all 6 degrees of freedom introduced in MAP++

By providing the software all sort of information such as setting the anchoring points on seabed, the attaching points on the ship and giving the appropriate characteristics, in this case polyester rope properties, the software is able to calculate the tensions in every coordinate direction (x,y,z) for each one of the lines. The software considers the line to be in an open area, with no external forces applied, such as current, waves or even the buoyancy of the rope itself and outputs a result shaped in a text file as well as the plots seen in Figure 4.21.

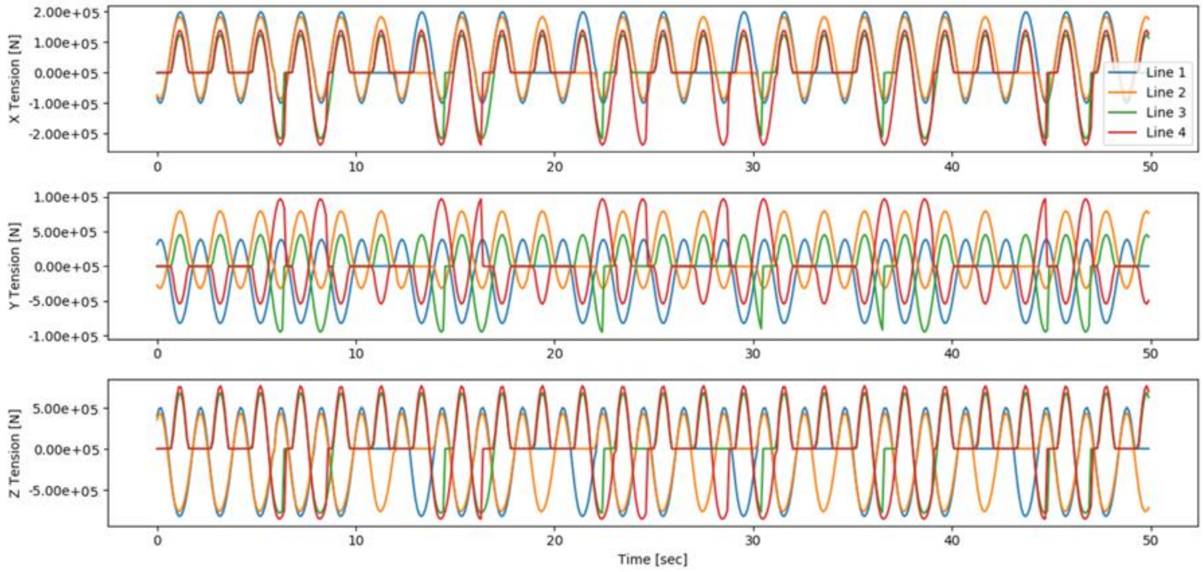


Figure 4.21. Tensions in X, Y and Z direction for the four mooring lines studied.

This will further allow to calculate the module of these tensions and obtain the resultant force for each one of the lines as seen in Figure 4.22.

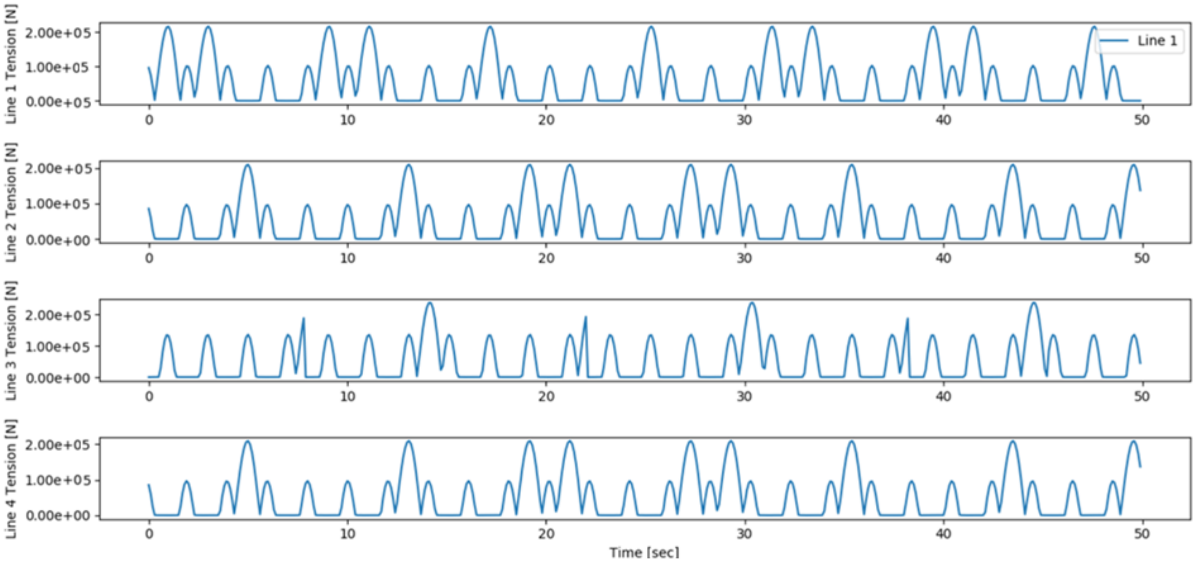


Figure 4.22. Module of the tension of each separate mooring line studied on this example.

By simplifying the rope to a linear spring problem and knowing the rope properties, such as the axial stiffness and diameter, one can easily plot the elongation $\Delta l$ for the different ropes by using Equation 4.1.

$$\Delta l = \frac{T}{k} \tag{4.1}$$

Where $T$ is the module of the tension in each rope, and $k$ is the axial stiffness coefficient found in Table 4.3 for different types of material on the ropes used as mooring lines. These coefficients are obtained from simple statistical techniques and depend on the diameter of the rope. For the current case, polyester ropes with a diameter of 0.635 meters are chosen, so the elongation over time can be displayed in single line plots as in Figure 4.23.

Table 4.3. Specifics of mooring lines with different materials

| | **Axial stiffness (kN)** | **Weight in water (kg/m)** |
|---|---|---|
| Nylon ropes | $1.18x10^5D^2$ | 36.66 $D$ |
| Polyester ropes | $1.09x10^6D^2$ | 13.33 $D$ |
| Polypropylene ropes | $1.06x10^6D^2$ | 11.33 $D$ |
| Chain lines | $4.13x10^4D^2$ | 110 |
| Wire line | $7.32x10^4D^2$ | 520 D |

When choosing a material for the mooring lines, the material will be determined by the depth that the mooring line needs to reach, since the deeper the seabed is, the heavier the line gets. For this reason, synthetic ropes are used in order to overcome the weight in deep water and this way compensate for a large restoring matrix. On the contrary, other types of heavier materials might be used in shallower waters.

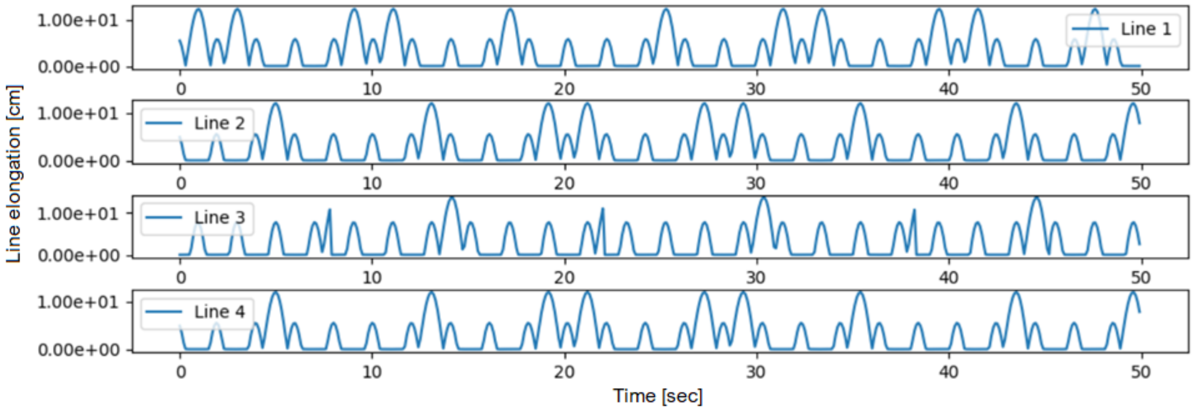


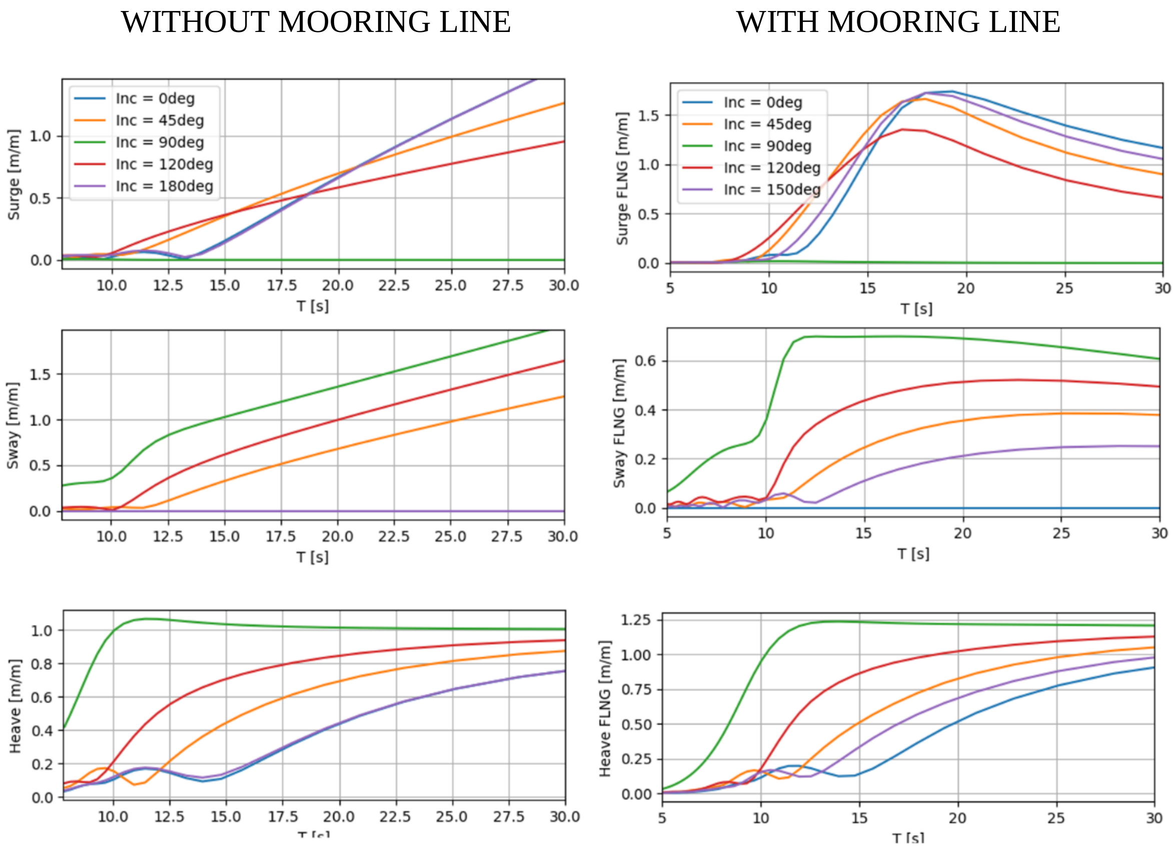
Figure 4.23. Elongation of the mooring lines in the current example.

### 4.4.3 Real time calculation of the tensions

Once the simple case is established, the same principles can be applied and calculated automatically on the web browser at the same time that the simulation is running, so the user can know the real-time tension of each mooring rope. This feature has been originally developed as an example of the Vessel.js library (Kramel, 2019), and has been adapted to the current case. To do that, Kramel uses Equations 2.19 and 2.20 to calculate the tension on the mooring lines at every time step during the simulation. For further applications these lines could even change colours depending on the tension in order to visually aid the user to identify what lines are handling more tension. An example of this real time tension is further displayed in Figure 4.35.

### 4.4.4 Comparison of the motion with and without mooring lines

When obtaining the RAO plots for this new case, one can compare the results with the same entity without the mooring lines, as seen in Figure 4.24. The motion happening parallel to the ocean plane is the most affected by the mooring lines, such as surge, sway and yaw, which significantly restrain the motion. A good way to interpret these results is by looking at the surge motion where one can see that without the mooring lines, the amplitude kept growing linearly and with them, the amplitude reaches a maximum until decreasing again for most of the headings described.


WITHOUT MOORING LINE

WITH MOORING LINE

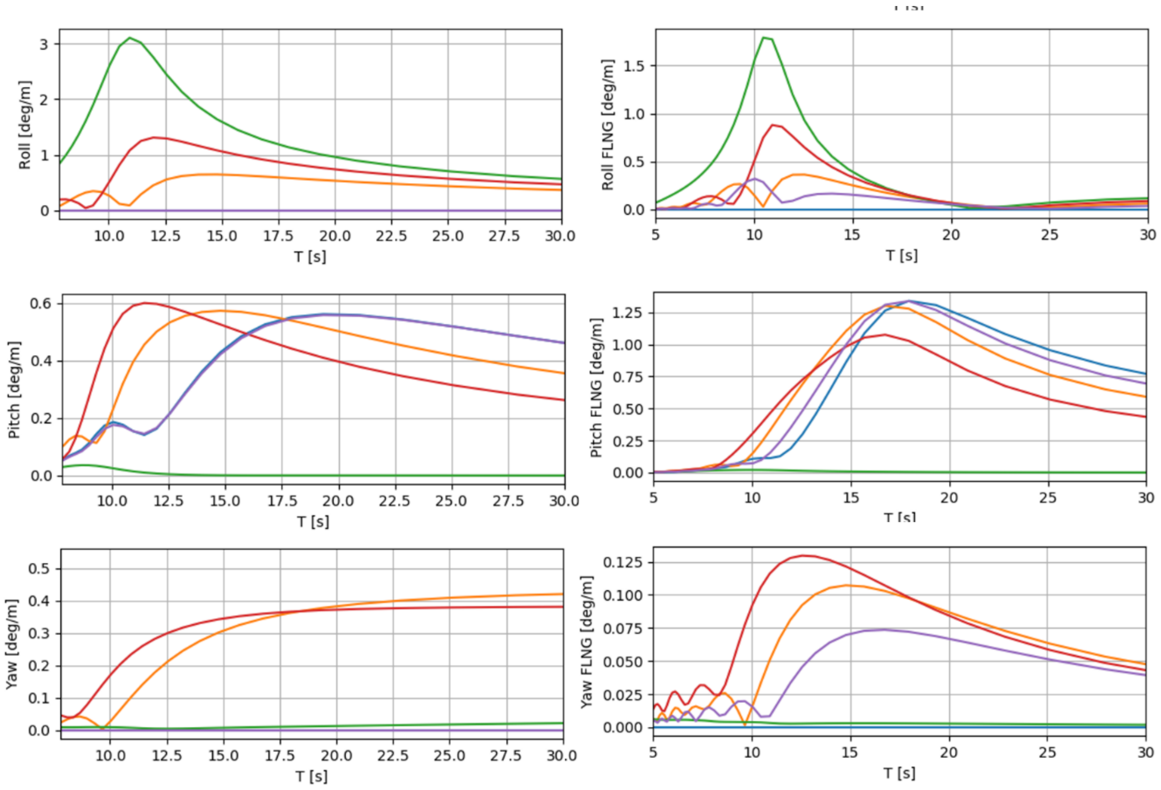
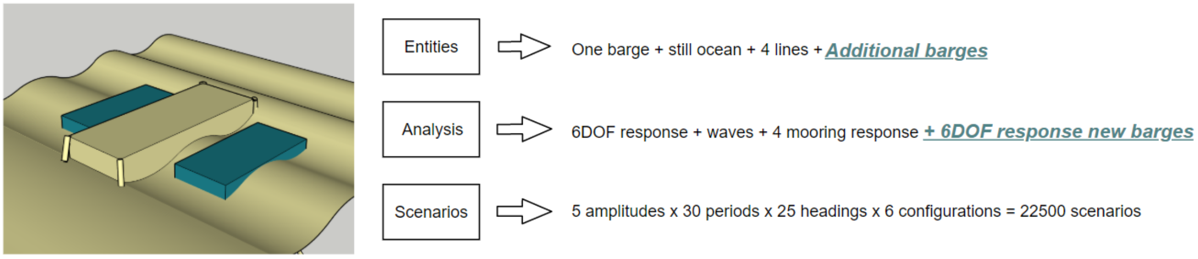


Figure 4.24. Comparison of RAO with and without mooring lines for 6DOF

## 4.5 Case 5. Additional vessels

This case goes a step further with multibody and aims to work with more than a single vessel, meaning that as stated on the literature review, the behaviour of all the additional vessels will be affected by the others around. Therefore, for this case different vessel motions will be expected for the FLNG than the ones previously stated. Also, a separate motion will be assigned to the extra bodies added next to the FLNG, which will be either one or two Suezmax vessels.



Entities ⇨ One barge + still ocean + 4 lines + ***Additional barges***

Analysis ⇨ 6DOF response + waves + 4 mooring response ***+ 6DOF response new barges***

Scenarios ⇨ 5 amplitudes x 30 periods x 25 headings x 6 configurations = 22500 scenarios

Figure 4.25. Model definition by its entities, analysis and scenarios.

What is also implemented are some additional sliders on the web browser, so that the user can choose the amount of ships in an offloading operation from a FLNG. Realistically, there can be a maximum of three ships in this operation, one being the feeder and two being the receivers.

The receivers can be involved to the FLNG in different operational conditions, so an analysis is made for when the ships are side by side and still not moored to each other. Every additional ship will require a new WAMIT calculation, and on top that, these cases can be simulated while the FLNG in the middle is moored to the seabed or not. As seen in Figure 4.26, the user is given the choice between these four possible scenarios with a simple slider that will activate the number of ships and the option to include or not the mooring lines.

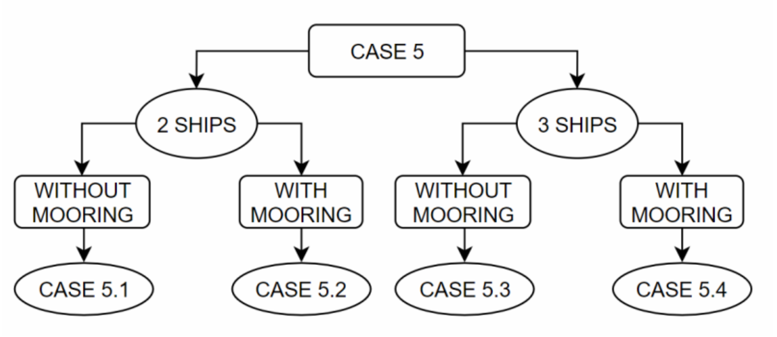


Figure 4.26. Possible scenarios with the current case

The process to perform this simulation slightly changes in respect to the previous cases, as introducing new bodies also introduce new states and ways to interact between them. The interface must be defined in order to understand the influence that one vessel has upon another one as seen in Figure 4.27.

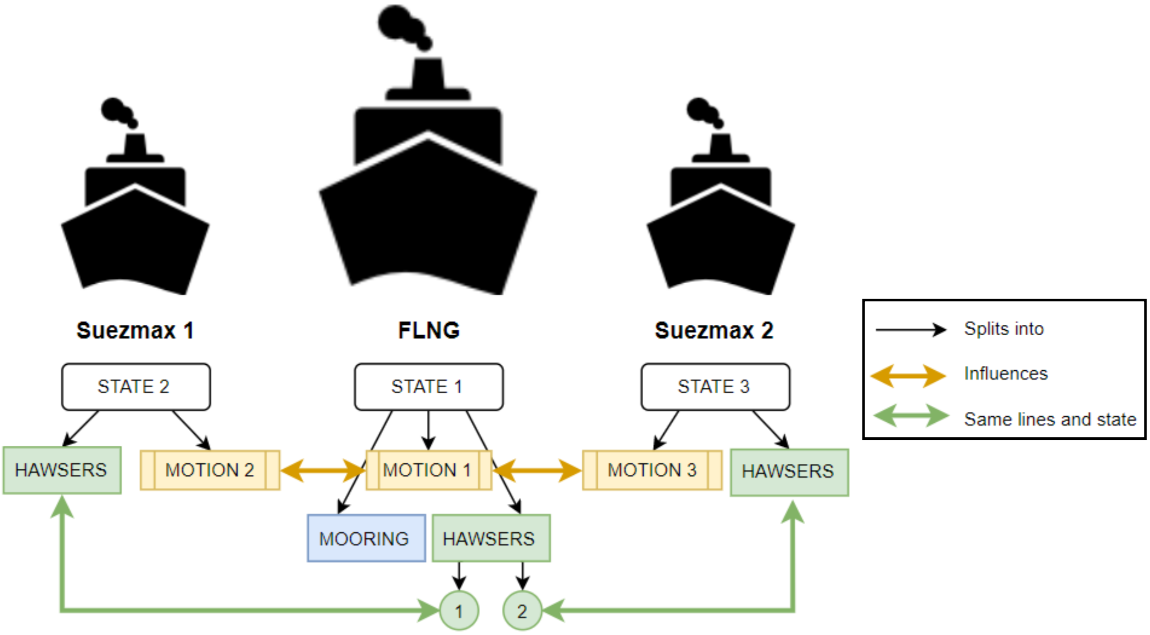


Figure 4.27. Definition of independent states and how these are correlated.

The process followed for the simulation keeps being the same as in the previous examples. The restoring matrix is the same for the FLNG. However, in this case, the configuration in WAMIT is set with side by side multibody as explained in section 2.1.3. New vessels are added in

WAMIT in order to simulate the motions for all the four cases in Figure 4.28. As mentioned in Section 3.3.1, WAMIT requires to introduce a lid on the gap between two vessels.

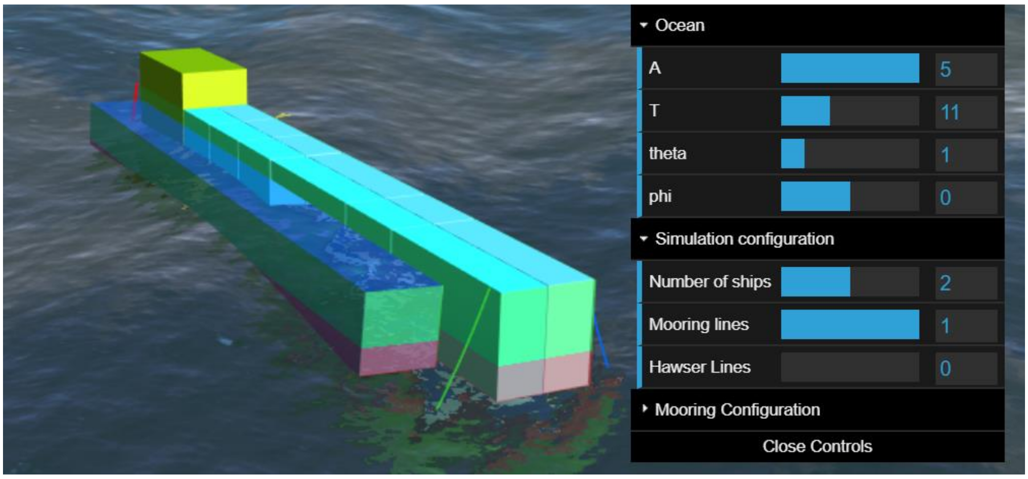

Ocean
A 5
T 11
theta 1
phi 0
Simulation configuration
Number of ships 2
Mooring lines 1
Hawser Lines 0

Figure 4.28. Visualization of this case with an addition of a Suezmax.

The mass matrices for both vessels are calculated and introduced to WAMIT as input as well as an additional gap lid placed between the two bodies in order to account for the resonance occurring on the gap between the vessels.

Mass matrix of FLNG

$$\begin{bmatrix} 1.43e+05 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.43e+05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.43e+05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.93e+07 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9.97e+08 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.05e+09 \end{bmatrix}$$

Mass matrix of Suezmax

$$\begin{bmatrix} 76983 & 0 & 0 & 0 & 0 & 0 \\ 0 & 76983 & 0 & 0 & 0 & 0 \\ 0 & 0 & 76983 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.18e+07 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.94e+08 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.16e+08 \end{bmatrix}$$

### 4.5.1 WAMIT-JavaScript interface for multiple vessels

So far, when there was only one vessel, a single “Explore file” button could read a file and put it directly into a matrix. However, when several simulations must be represented in the same web-based interface, it requires more than one WAMIT output file in order to update the motions in the simulator correctly and let the user interact with different ocean space configurations. For this, it is not easy to read the files separately and obtain different matrices for every file. To solve this problem that has more than one case to be stored, an object is an ideal concept to nest the cases and their motions, so a large JavaScript Object Notation (JSON) file with the output for each one of the cases is created.

This JSON file is created with a Python script that automatically transforms the data from the Force.4 obtained in WAMIT into useful and nested objects. This way it is also possible to plot the RAO shown in the previous cases and in Appendix B. This JSON file is composed initially of the four cases studied, but later on a compilation of all the cases on this study with ten of them is made. Each one of these cases nests the pertinent degrees of freedom depending on the number of ships. Then, for each degree of freedom there are 30 possible periods inside and for each one of these there are 25 different headings ranging from 0º to 180º every 15 degrees. For the ultimate ten cases file, it leaves a file with 7.500 possible combinations to simulate on the web browser, a big enough database to give an accurate idea of how the simulator works.

Regarding the visualization on the web browser, the addition of extra vessel entities will require creating a new ship object with the Vessel.js library, assigning it a new state and applying the motion obtained by WAMIT that correlates the side-by-side case in order to obtain a result like the one shown in Figure 4.29. The plots for these 4 cases are shown in **Appendix B.**

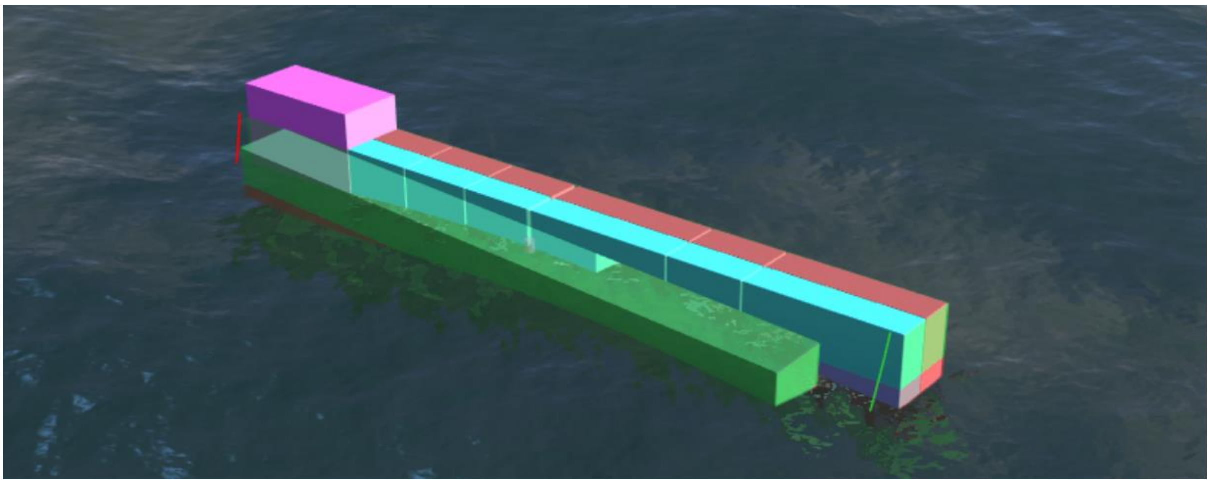


Figure 4.29. Visualization of response of the current case with a moored FLNG and one Suezmax on the side.

### 4.5.2 Comparison of the motion with one and two bodies

It is already mentioned that adding an extra body next to an original one will cause some disturbance to this, even if they are not physically attached. This is the so-called shadow effect and to see the difference between these two cases, Figure 4.30 is presented, where one can see that although the RAO plots are not experiencing a big difference, slight changes for some heading in particular are to be considered. This might seem insignificant, but it is crucial when one has to know with exactitude the motions for surgery-work operations.

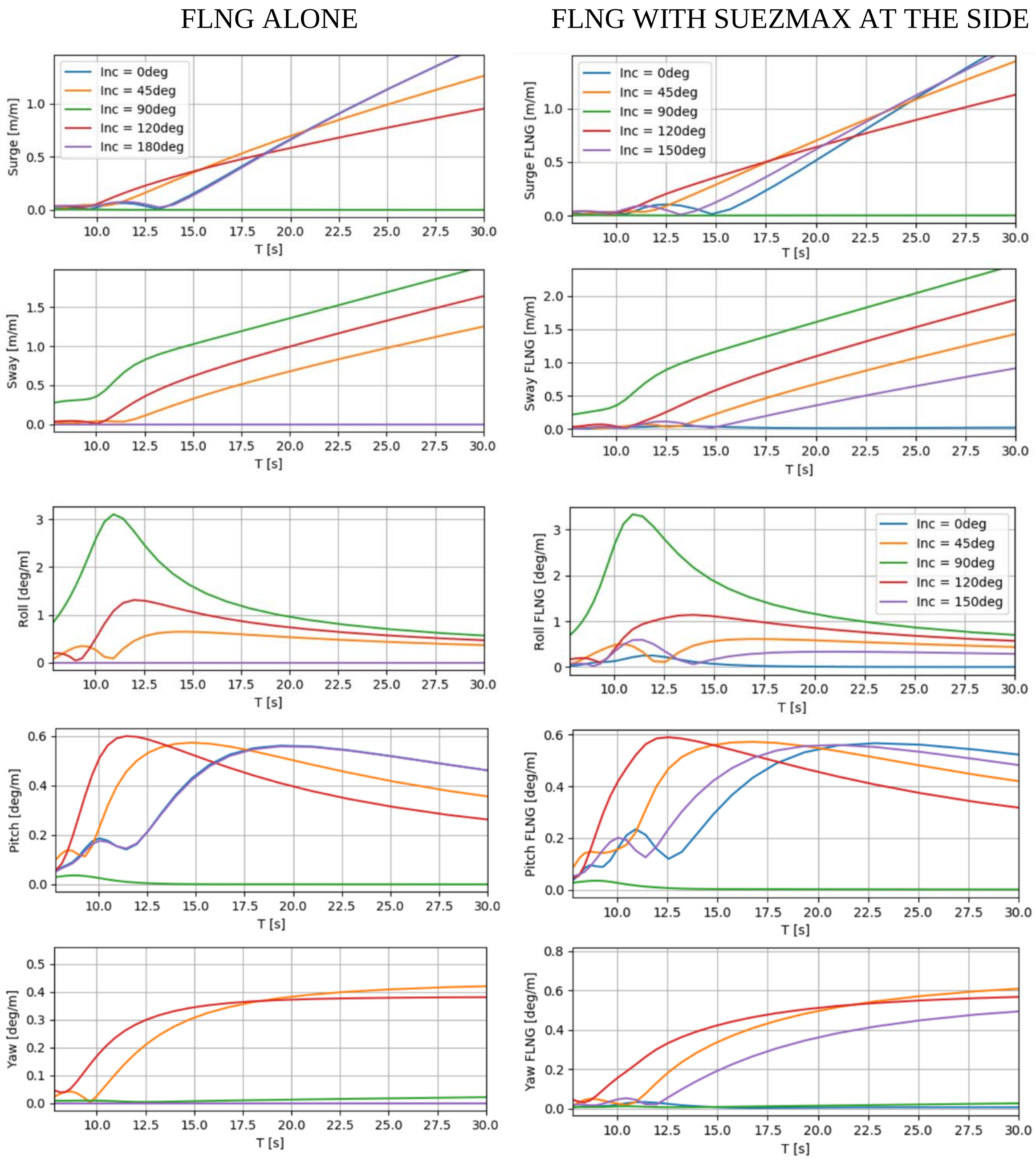


Figure 4.30. RAO comparison between an FLNG along at sea (left column) and an FLNG with an additional vessel at the side (right column)

## 4.6 CASE 6. HAWSERS ADDITION. OFFLOADING SIDE BY SIDE OPERATION

Building up from the previous cases, the last entity will be introduced. By adding this entity, we can already consider this configuration a current marine operation, which happens when one or two vessels are moored with hawsers to an FLNG, which is moored to the seabed in order to download LNG to these other vessels.

It is not the same to have side by side vessels with or without hawsers, as although roll, heave and pitch motions will not change much, the ones on the ocean plane will be restrained by the other vessel, especially if this second vessel is moored. So, the results expected in this section are very similar results for ships with hawsers and no hawsers, but with slight variations.

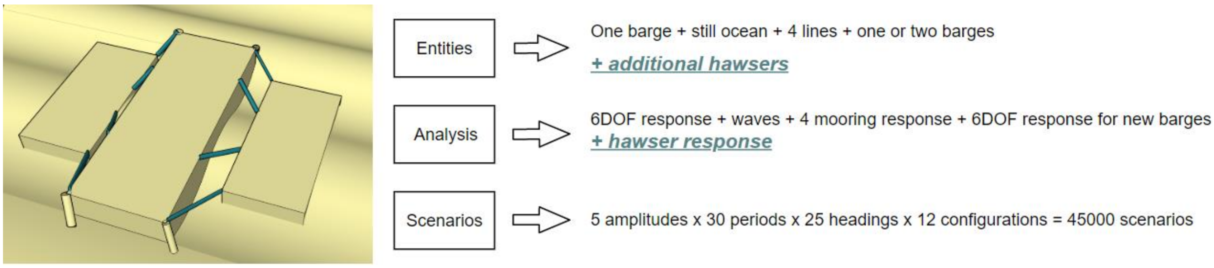

Entities
One barge + still ocean + 4 lines + one or two barges
+ additional hawsers
Analysis
6DOF response + waves + 4 mooring response + 6DOF response for new barges
+ hawser response
Scenarios
5 amplitudes x 30 periods x 25 headings x 12 configurations = 45000 scenarios

Figure 4.31. Model definition by its entities, analysis and scenarios.

As done in the previous case, the mooring external restorative matrix is obtained for this new simulation that considers the hawsers as restraining elements. This is obtained again with MAP++ for later being run in WAMIT and obtaining the correspondent motions for the case. The restoring matrix evaluated and the visualization of the input lines are shown in Figure 4.32.

Hawsers external restorative matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2.7e+06 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7.7e+6 & 0 \\ 0 & 0 & 0 & 3e+03 & 3e+06 & 5.1e+08 \end{bmatrix}$$

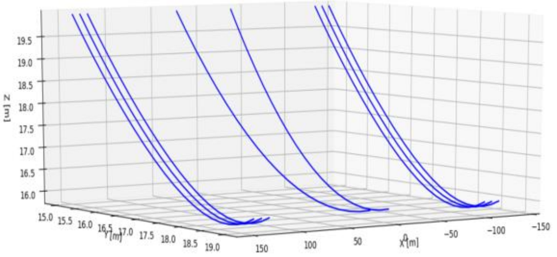


Figure 4.32. Visualization of the hawser lines in MAP++

This case will sum some additional configurations when comparing to the three previous cases by adding hawsers in each one of the four configurations mention in Case 5, as in Figure 4.33.

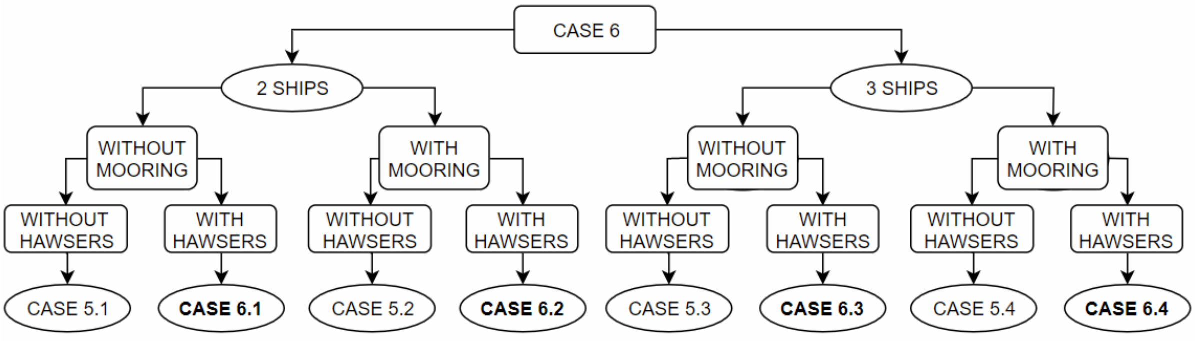


Figure 4.33. Additional configurations for the current case

As previously described in Figure 4.27, the process in this case is slightly different as the ships are influenced by each other, not only by the shadow effect but now also directly by the hawsers joining the vessels. A comparison between the two cases regarding motion influence will be later performed.

### 4.6.1 Usage of restorative matrices

The use of these matrices will come depending on the case and the characteristics of the ship. The adequate combination and sign must be applied in order to obtain the desired result, as when two ships are attached the restorative matrix due to the hawsers will be the same for both ships but in contrary signs, as when one whip is pulling, the other will be pulled. The combination of restorative matrices for the cases developed up to this point can be seen in Table 4.4, where *MM* stands for mooring restoring matrix which is found in Section 4.4, and *MH* for hawsers restoring matrix found in Section 4.6.

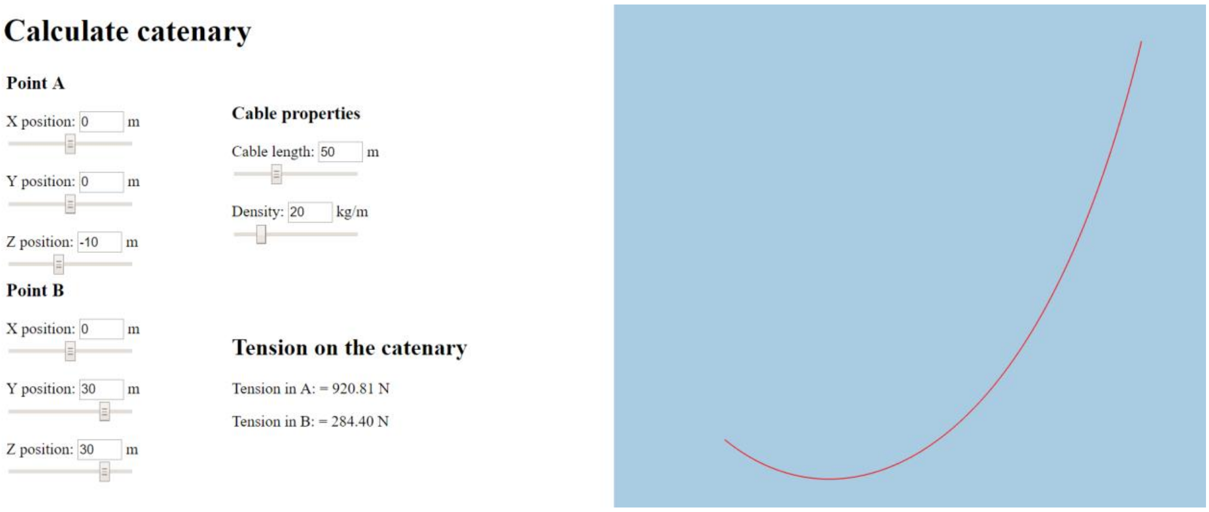
Table 4.4. Usage of restoring matrices in the different cases studied so far.

| | Vessel 1<br>FLNG | Vessel 2<br>Suezmax 1 | Vessel 3<br>Suezmax 2 |
|---|---|---|---|
| Case 3 | Free | X | X |
| Case 4 | -MM | X | X |
| Case 5.1 | -MM | Free | X |
| Case 5.2 | Free | Free | X |
| Case 5.3 | -MM | Free | Free |
| Case 5.4 | Free | Free | Free |
| Case 6.1 | -MM + MH | -MH | X |
| Case 6.2 | MH | -MH | X |
| Case 6.3 | -MM + MH + MH | -MH | -MH |
| Case 6.4 | MH | -MH | -MH |

### 4.6.2 Catenary development

In the current case, the use of catenary ropes is meant to be displayed on the hawsers. However, nowadays the Vessel.js library only provides the user with catenary lines adapted to mooring conditions, meaning that when the catenary reaches its lowest point it turns into a horizontal line, as it is touching the seabed. This also means that the user can control where end of the rope is located, but not the other one, as it depends on many factors. However, there is no such a rope simulating a catenary from beginning to end. So, in order to have full control of where the catenary starts and ends, a new app has been developed with new equations as in Figure 4.34. This takes the user to a single interface where to play with the parameters of the rope such as its length, start and end point as well as the density of the rope, which will directly affect the tension on the edges of the rope, which are also displayed.

This catenary is supposed to be brought to the new case in order to display the hawsers in a catenary way. However, reached this point, the web app is becoming quite powerful and it already requires a large computation effort to read the motion files and calculate the mooring lines and plots in real time. By adding the calculations for the catenary lines of eight different hawsers, the computational effort is too large for some computers running the web app. This limitation obligates to use straight lines for the visualization and coding as seen in Figure 4.34. Numerically wise, this will be accounted in WAMIT with an additional external restoring matrix for both vessels, so the motion will be defined correctly in the simulator, but not the shape of these hawsers.



Calculate catenary

Point A

X position: 0 m

Y position: 0 m

Z position: -10 m

Point B

X position: 0 m

Y position: 30 m

Z position: 30 m

Cable properties

Cable length: 50 m

Density: 20 kg/m

Tension on the catenary

Tension in A: = 920.81 N

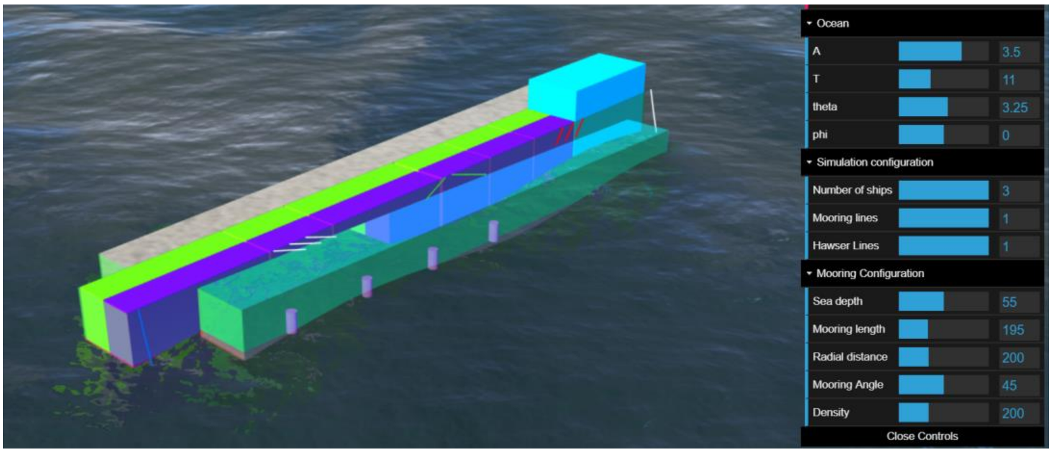
Tension in B: = 284.40 N

Figure 4.34. Catenary app developed to simulate an asymmetrical catenary from point A to B with a given length.

### 4.6.3 User interface for the ten studied cases

Once understood how to implement WAMIT into Vessel.js, a more complete interface for this simulation has been developed. This is slightly more intuitive and gives the user more options to play with the simulator. A graphical user interface (GUI) is added in this Case in order to let the user choose between the ten different possible scenarios, consisting of the eight described in Case 5 and 6, and two additional considering Case 3 and Case 4. With this GUI seen in Figure 4.35, additional interactivity is provided, which will help the user to understand the process better than in paper.

This case also adds a feature to see the motion of the ships in real time, so one can compare with factual data if the motions are having a fair behaviour, in addition to see them in the visualization, which is more difficult to corroborate results. The data for these plots is the same data for the ship motions calculated with the motion equations from the matrices pre-loaded with the raos and phases from WAMIT simulation and can be seen in Figure 4.36.



Ocean

| | |
|---|---|
| A | 3.5 |
| T | 11 |
| theta | 3.25 |
| phi | 0 |

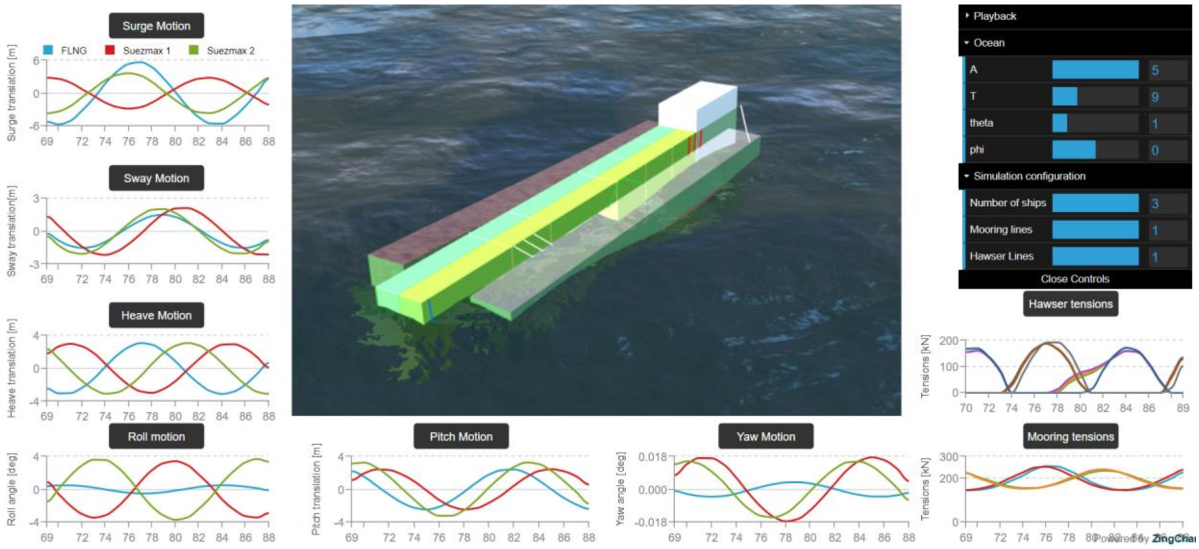
Simulation configuration

| | |
|---|---|
| Number of ships | 3 |
| Mooring lines | 1 |
| Hawser Lines | 1 |

Mooring Configuration

| | |
|---|---|
| Sea depth | 55 |
| Mooring length | 195 |
| Radial distance | 200 |
| Mooring Angle | 45 |
| Density | 200 |

Figure 4.35. Visualization of the current case displaying the maximum possible entities.



Ocean

| | |
|---|---|
| A | 5 |
| T | 9 |
| theta | 1 |
| phi | 0 |

Simulation configuration

| | |
|---|---|
| Number of ships | 3 |
| Mooring lines | 1 |
| Hawser Lines | 1 |

Figure 4.36. Visualization of the current case with extra plots showing the motion of the three vessels in real time.

At this stage of the project, the tensions for the mooring lines are already calculated from Case 4. However, the tensions for the new introduced hawsers are not calculated and might be of some interest by the user. It is for this reason that these are also calculated automatically in the browser with a new formulation. In this case, these are meant to be catenary lines, despite the fact that they are displayed as straight lines.

As for the hawser tensions, an improvement with respect to the other cases has been made and now the tensions are calculated instantly in the browser. Assuming that the motion is displayed according to WAMIT calculations considering the restraint of the hawsers, the positions of the points of attachments will be used to calculate the variation of the length of the hawsers when in tension. This increment of the length will be used to calculate the tension with the properties given by a polyester rope of 0.0635 m mentioned in Table 4.2.

### 4.6.4 Comparison of the motion with and without the hawsers

After running several simulations with and without the hawsers between the vessels, similar ship motions have been observed for these both cases. The RAO’s have been compared and overlapped, which have been found to be very close to one another, leaving only slight variations of centimetres in some of the motions that are considered insufficient for its further study. These are however considered for the visualization as straight lines, as it brings some insight and gives a better understanding of the interaction between vessels.

This slight variation is assumed to be because of the rope configuration when calculating the restoring matrix, when a rope with a high elasticity has been chosen, so it does not influence the motion too much.

## 4.7 CASE 7. SUBSEA OPERATION WITH STEEL LAZY WAVE-RISER CABLE

For the current case, the goal for this study is reached, which is by now being able to combine the entities and analyses described so far in order to configure many other scenarios representing operations to simulate. This is shown in this case as an example to provide another operation with the tools already used on the previous cases. Later on, on Section 5, the process to create new configurations using the tools created so far is explained.

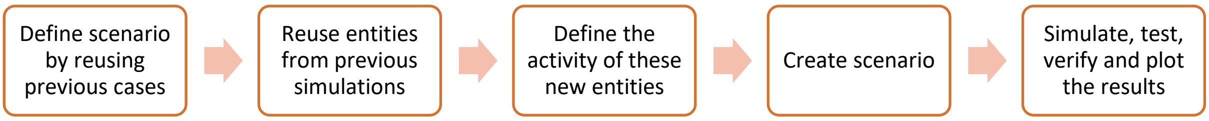

Define scenario by reusing previous cases
Reuse entities from previous simulations
Define the activity of these new entities
Create scenario
Simulate, test, verify and plot the results

Figure 4.37. Process for the creation of the current case.

### 4.7.1 Definition of the operation

Using the progression built on the previous examples, a subsea operation is considered for development. With all the tools already available, a different configuration of the ship, mooring lines and ropes can be carried in order to simulate this subsea operation. The motions of the vessels can be reused for this case (always modifying the external restoring matrix) and can be used as input for the ropes involved in this type of subsea operation.

The operation is inspired on the latest operations of oil extraction through a pre-salt layer made by PETROBRAS on the Brazilian coast of Santos. This operation includes non-structural lines such as electrical, umbilical and steel lazy wave-riser (SLWR) cable going down to the seabed to feed some subsea equipment while the vessel remains moored.

When it comes to defining the lines, as seen in Figure 4.40 there will be three cables configuring this problem.

The *umbilical cable* will be the one supplying consumables to the drilling equipment. This will be quite thick but will not be offering much resistance for the motion of the ship, as it will be made of an elastic material in order to guarantee the integrity of the material. One side will be attached to the anchoring point on the ship, which will give a motion to the line. The other end will be attached to the box on the bottom, so it will be fixed. It will act as a catenary line with some of the cable laying on the seabed.

The *electrical cable* that will supply the necessary power to the drilling equipment. This will also be acting as a catenary line with some cable laying in the bottom. However, this time, being an electrical cable, the elastic modulus will be much lower, and it will not be accounted in the additional external restoring matrix, for its negligible weight.

The *Steel Lazy Wave Riser* will be composed of two segments of catenary lines as shown in Figure 4.38. In order to simplify the problem, the motion of the distributed buoyancy module will be considered fixed due to being very small, as the first catenary segment will damp the movement. To this buoy the end of the first segment and the beginning of the second segment will be attached and have a limited motion. The first line going from the vessel to the buoy will act as a pure catenary cable, and the second one, connecting the buoy with the drilling equipment will be provided with some extra length laying on the seabed, in order to guarantee that there is only tension created by the weight of the cable, not because the ship is pulling the cables. This is why a good mooring system will be introduced too in order to maintain the vessel in position.

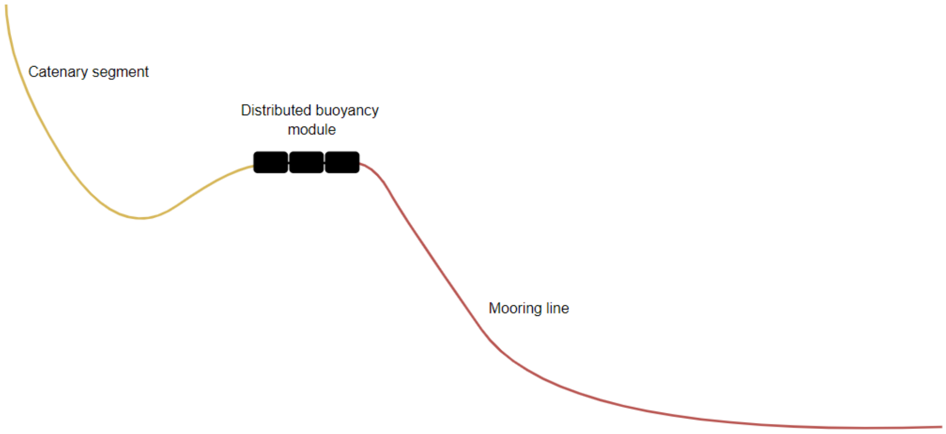


Figure 4.38. Configuration of the SLWR

The suspended length of any cable gets longer the deeper the seabed is, so the weight also increases proportionally, leaving the tension in the vessel being higher, the deeper it goes. It is for this reason that these buoys are also installed, to take some of the weight of this riser. These buoyance modules are less dense than sea water, which produce a net upward force into the cable. These are used to shape the steel lazy cable in order to decouple the dynamic motion produced by the vessel. This way, the hanging catenary absorbs most of the motion and none is transmitted to the seabed.

The diameter of the riser and the wall thickness will come determined by the hydraulic purposes and the sea pressure that it has to withstand in order to avoid collapse or bursting. For this reason, this cable might be much heavier than the others, affecting somehow differently the ship, but as mentioned before, as a simplification, no restoring matrix will be added respecting the cables added in this example.

### 4.7.2 Definition of the model

Figure 4.39 points out all the single entities that are needed for this simulation. All of these are known already from previous cases. Also, their behaviour has been studied previously and now these are put all together as seen in Figure 4.35, in order to create these different scenarios with the current configuration. Each one of these entities will have a state applied to it, which will force the web browser to perform an analysis over them.

Once joined all together a scenario is created as in Figure 4.40, but if in addition, the user can play with the 5 different amplitudes, 30 periods and 25 headings, the number of scenarios become 3.750.

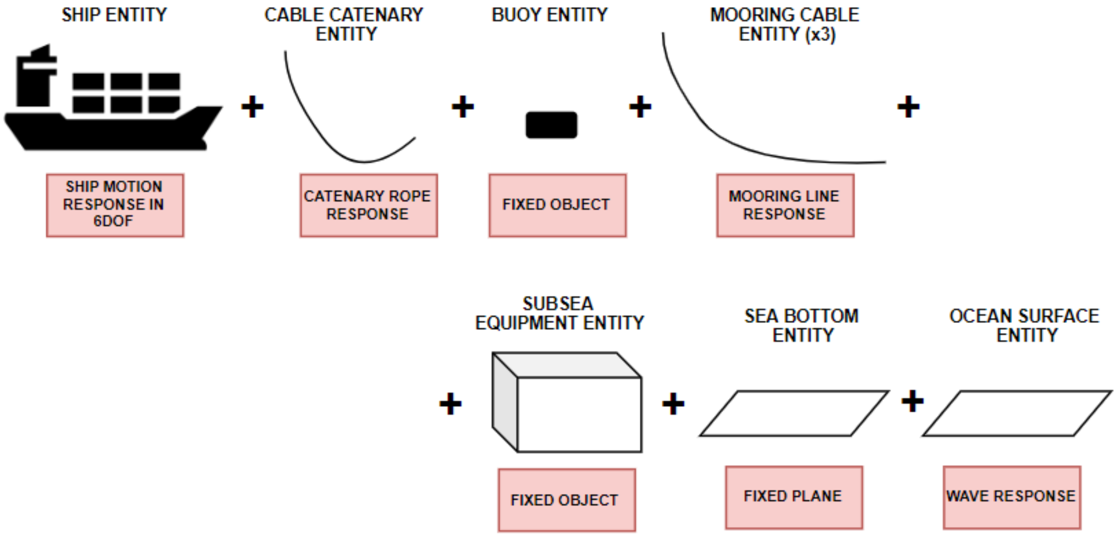


Figure 4.39. Single entities and analysis applied to each one for the current simulation.

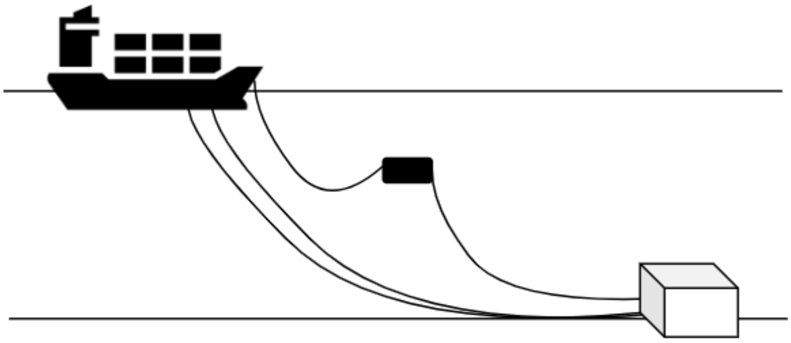
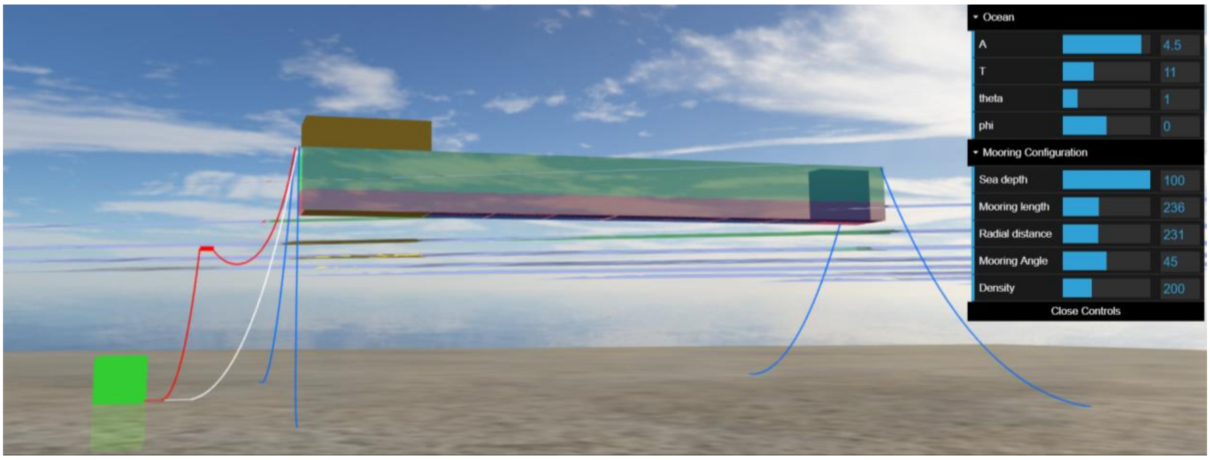


Figure 4.40. Subsea operation with SLWR, umbilical and electric cable.

### 4.7.3 Simulation and visualization

This scenario will consist of a single vessel moored (Case 4), with additional lines (Case 5 and Case 6). In order to show the easiest way to reuse the previous simulations, the cables will be considered to be light enough, so they do not influence the motion of the vessel. The SLWR will be considered the heaviest, but since it has a set of buoys holding most part of the weight, it will also be neglected. This way no additional WAMIT runs will have to be run, as no additional restoring matrices are added. If these cables had to be considered, another WAMIT run would be required. So, the results for the motion will be the same as in Case 4, but the tensions of the additional elements can be calculated automatically on the browser by reusing the from previous cases and giving new data due to the different position of the lines.

A similar simulation to the one in Case 5 will be performed, but there will be four extra cables feeding the subsea operation in addition to the mooring lines. As previously done, these lines will be assigned some properties stored in objects in the simulation, so one can eventually take this data and turn it into useful calculations such as rope tensions.



A 4.5
T 11
theta 1
phi 0
Sea depth 100
Mooring length 236
Radial distance 231
Mooring Angle 45
Density 200

Figure 4.41. Visualization the current case.

The code to display the entities is reused for every single entity. The only variations in this simulation is the position of the additional entities, the creation of new boxes to simulate the buoys and the subsea equipment and the connection between them. As for the analysis of these entities, the theory for mooring lines is used for the umbilical and electric cables.

As for the SLWR the buoy will be considered static, as its function is to damp the motion of the ship simulating the seabed. Taking into account that currents on the sea are not considered, there is no reason to find the mathematical model for the motion of this buoy, as its movement will be almost null if there is no current.

### 4.7.4 Evaluation

The representation of this maritime operation is carried successfully after putting all the elements together. It was expected to simulate an operation that could connect the vessel with the subsea equipment while damping the motion of the lines arriving to the blocks on the seabed, in order not to spoil the cables and the equipment itself. So, when simulating, one can see that the motion of the lines reaching the block at the bottom is practically null. This is the result of placing this subsea equipment far from the vessel and adapting the length of the cables so that there would be some resting catenary line on the seabed.

At the same time, the level of complexity for the development of this operation was not too elevated, as we only had to place the new elements on the desired position, and give motion to the elements requiring it, such as the end of the ropes attached to the vessel or the buoy that experiences a little heave motion from the catenary line. For this reason, we can conclude that new operations created by any reader can be easily created using the tools and cases created on this study.

# Chapter 5

## 5 DISCUSSION AND FUTURE IMPLEMENTATION

This chapter discusses the cases from Chapter 4 by evaluating the results obtained in the different cases as well as compares the difference between cases in order to verify the reliability of the method. The chapter finishes opening a window to the readers to help them develop their own maritime operations by reusing the code used and developed in these case studies. The cases studied on the previous chapters have been accurately defined in order to follow a progression of complexity, which has allowed to analyse with more detail every little change on the simulations and thus evaluate at every step if the obtained results were successful.

### 5.1 EVALUATION OF THE MOTION RESULTS

From a motion calculation perspective, the results calculated are fairly within what was expected. However, used to the idea of the motion in one single vessel, having a side by side configuration derives into a variation of the motion a vessel is supposed to have, so a different interpretation of the RAOs has to be made. This gives an idea that the WAMIT simulations are performed correctly.

Relating the results to the literature review, we can see that for Case 1, the behaviour of the ship is strongly related to the Vessel.js example for a radiation problem developed with differential equations. For the following examples, WAMIT is the core of the simulation, as all the dynamics happening are previously calculated in this software, so one can assume that the analysis of the hydrodynamics of the vessels follows an accurate potential theory. Also, the use of the taxonomy is preserved through the whole study, helping to organize the code in a coherent and comprehensive way, allowing the interpretation of the motion data correctly.

The major findings of this project have been to introduce a new way to define the motions in Vessel.js through 3D potential theory using a software developed by TPN and Petrobras which introduces WAMIT to calculate offshore systems configurations. This has allowed to overcome the limitations of single body motion displaying, as WAMIT takes cares of bodies that are in close proximity and acts accordingly. This way of calculating the motions comes in handy when several operations are pre-calculated, or one wants to analyse the behaviour of a single operation that is going to be done repeatedly. If for example a ship design office needs to make several WAMIT runs for the same vessel with different loading conditions, in order to visualize and obtain real-time data, a web-browser application is ideal, as one can just upload the WAMIT output file and obtain an instant idea of what is happening to the simulation.

However, using this method to calculate the motions faces some limitations, as obtaining the motion from an external software that takes a long time to run is not the ideal solution for a web browser application. If the user were looking for an instant response as well as being able to scale the ship and have more freedom when modifying code, simpler ways to calculate the vessel responses need to be adopted such as the ones that are currently running with closed-form expressions or differential equations. Also, a new architecture could be proposed where a web interface would be linked directly to the TPN program. This last case will be suggested as future work.

Also, some challenges have been experienced while trying to read the input of more than one text file at the same time, so a simulator that allows the user to load various cases at the same time came off the table, obligating to create pre-loaded motions and leaving the previous task for future implementations.

## 5.2 EVALUATION OF THE METHOD

From a method perspective the results obtained prove the ability to achieve good results following the procedure proposed as well as achieving the objectives stated on Section 1, which are mainly to have been able to simulate correctly multibody and its interaction with several numbers of constrains comprised within different cases. The way to divide the cases in the taxonomy provides the user the facility to see how it really works and that a simulation is simple if it is properly understood. Each element will have a state separately which will be influenced by others when put together as a multibody.

Following the method described in Section 3, this study has been able to produce multibody cases with all kind of combinations and interactions, which is a massive step forward in comparison to the one ship entity and a mooring line system of the current Vessel.js library.

All the other entities, analysis, processes and their combination on this study are entirely developed by the author with the aim to simulate multibody subsea and maritime operations with a different approach for the ship motions. These multiple bodies are able to have different states interacting with a link entity. These are elastic ropes able to behave according to the states of two different vessels at every time step, which will depend on motions that are calculated with a different approach compared to the previous examples made in Vessel.js.

Looking at the results, one can say that almost all cases are relevant enough to be studied, as the behaviour of the entities is somewhat different when adding some of the entities such extra bodies and mooring lines. However, for the four subcases in Case 6 considering the introduction of hawsers, the mathematical background is not strictly necessary for this early stage of development of a simulator, as the difference of the simulation with and without them is not significant enough to be considered, as seen in Section 4.6.4. Also, for future studies, a less elastic rope can be analysed to see a more pronounced difference.

In a way, we can call the study a success because when a WAMIT run is made, there is no way to accurately know if the motions of a vessel are correct or not. By making this interface, we have been able to see the performance on real time of the vessels involved either visually or with real time plots. This has allowed to decide on many occasions that the motions were not displaying correctly and forced to rerun the solver in order to correct them to the final result.

All the unsettled points in the results are going to be discussed in the future work.

## 5.3 PRACTICAL USE OF THE SIMULATIONS

The contribution of this study is to be able to start from a very simple radiation problem to a subsea operation only by adding entities and analysis to these while using some tools from an open source library. The analysis area is focused on the hydrodynamics as well as into the visualization, as a difference from other studies that promote more the visualization or are more focused on the states of the entities.

This section wants to show how to build a marine operation only by adding entities used in these cases along with their respective analysis. This work is important because from here, anyone can take these case studies and build up for any further application, as the vessels and their motion will be already given.

In order to create a new multibody operation, the steps shown in Figure 5.1 are required. First the last version of Vessel.js library needs to be downloaded (http://vessel.js.org), where one will find some of the examples from this study. On the examples it is possible to take the most basic case providing all functionalities which is based on WAMIT motions (Case 3), where there is a simple barge on a regular sea, with no additional entities. The user can then go through the different cases reusing the piece of code needed for constructing a maritime operation. For instance, if an additional vessel and its response are needed, the code for that can be found on Case 5. If in addition some ropes are needed, we can find that in Case 6 and so on.

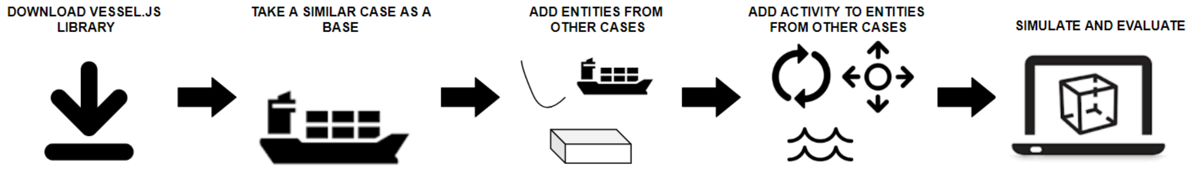

DOWNLOAD VESSEL.JS LIBRARY
TAKE A SIMILAR CASE AS A BASE
ADD ENTITIES FROM OTHER CASES
ADD ACTIVITY TO ENTITIES FROM OTHER CASES
SIMULATE AND EVALUATE

Figure 5.1. Process to create a multibody maritime operation

This process though is limited, as the motion of the ship depends on these additional entities added. In some cases, as seen in Figure 5.2, a new WAMIT run should be calculated before taking as accurate the motions on the ships, as someone taking a basic example with one ship and four mooring lines and after adding ten more lines limiting the motion of the ship, cannot expect the motion to be the same. This process is only valid for additional light lines that will not affect much the motion of the ship, or also for the addition of vessels far from the original vessel, as otherwise, the motion should also be affected and a new WAMIT run is required.

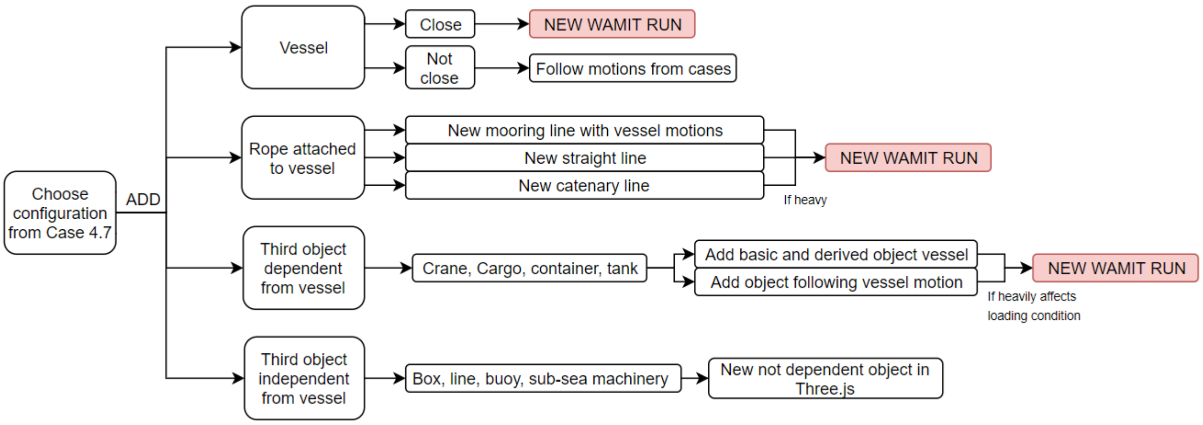

Choose configuration from Case 4.7
ADD
Vessel
Close
NEW WAMIT RUN
Not close
Follow motions from cases
Rope attached to vessel
New mooring line with vessel motions
New straight line
New catenary line
NEW WAMIT RUN
If heavy
Third object dependent from vessel
Crane, Cargo, container, tank
Add basic and derived object vessel
Add object following vessel motion
NEW WAMIT RUN
If heavily affects loading condition
Third object independent from vessel
Box, line, buoy, sub-sea machinery
New not dependent object in Three.js

Figure 5.2. Ways on how to add more entities and analysis for new operations formation.

As for adding more configurations, there is no need to add anything else, the simple user interface designed in Case 7 deals with all different kinds of scenarios only by moving a slider. If some additional functions want to be added, the sliders must be adapted to every new case created by the user.

# Chapter 6

## 6 CONCLUDING REMARKS

This chapter gives a conclusion to the study based on the case study performed and the discussion made in the previous chapter. The general points are stated in this chapter. After, several ideas are proposed as future work that could be done on top of this study are suggested.

### 6.1 CONCLUSIONS

This study proposes the use of WAMIT calculations to feed an open source library dedicated to the simulation and visualisation of marine related operations and configurations. The project contributes to the library by adding multibody operations such as side-by-side offloading or subsea. The work uses a taxonomy that organizes the simulations by entities, states and scenarios, which allow to treat each entity as a separate simulation, being are later put together using link entities in order to simulate an operation. It starts by simulating simple cases that keep growing in complexity, adding entities and analysis to the simulation at a time.

Easy tools are provided in order to read WAMIT output files and to simulate the various cases presented in an easy-to-use interface with different sliders that change from one case to another while reading pre-loaded WAMIT results as motions. These simulations are done in a web browser for a specific reason, and this is because of the interactivity, openness and compatibility that JavaScript provides as a programming language to carry these kind of simulations, which supports all kinds of additional libraries that allow to make the visualization a lot easier by providing 3D rendering, plots and other functionalities.

Finally, some guidelines are given to the users allowing them the opportunity to create their own maritime or subsea operations with the reusage of the cases done in this project. As stated on the objective, the cases are done with a collaborative library where everyone can use the code freely, improve it and create new operations, and it is for this reason that this work encourages as much as possible to do so.

## 6.2 Recommendations for future work.

Considering what limitations and challenges were faced on this study, some topics came out as possible candidates to be developed in the future by other researchers.

At this stage of the study one of the limitations is that despite the accuracy and good results that WAMIT provides, it is not suitable for real time simulation as it is now. It gives certain flexibility to the user to create new operations, but not full control, because calculations are done outside the browser. An idea to work in the future would be to develop a way to introduce WAMIT runs in the web browser, even if that took some time to run the calculations. Someone could develop an architecture where to read the input of the user through a GUI in the browser and link it directly to the TPN cluster, run it and bring back the results to the webpage, without the need of uploading any files. This way, proper potential theory could be used to solve any case the user could request. The level of complexity is very high, but it could be a good topic for a future master thesis.

In the case that this new way of calculating the motions could be performed, it would be a good idea to also go a step further and start accounting for what is not accounted in this study such as additional elements like fenders or dock proximity with these elements. Also wave forces and mean drift forces could be considered in a more accurate and professional simulation environment, but for this work it is believed to be enough with just displaying the motion of the ship only with the RAO amplitude and phase.

Finally, a good addition to the library would be to start working with rigid body interaction. This would be a totally different concept, as the motions would not have to be calculated anymore as they are done now, trying to match the regular sea library. They would be a real response of the input waves. There should be a code specifying how to deal with a ship in waves, but then, no matter the condition, the ship would behave like a rigid body in a viscous fluid, just like in real life. Also, working with rigid bodies would allow for instance to have two side by side vessels with fenders and simulate the collisions. There are nowadays some libraries simulating rigid bodies with collision using the Three.js environment and it could be a good addition with a different approach to the Vessel.js library.

# BIBLIOGRAPHY

AASE, Ø. (2018), *Modelling and control of catenary mooring systems for vessels, oil rigs and barges,* Master's thesis, NTNU, Trondheim.

DHAVALIKAR, S.; NEGI, A.; DOSHI, R. (2009), *Stability and motion analysis for barges*, National Conf. on Computer Aided Modelling and Simulation in Computational Mechanics, Mumbai.

ESEFELD, B.; SCHINDLER, T.; ULBRICH, H. (2013), *A coupling approach for the numerical integration of non-smooth multibody systems*. 4th Canadian Conference on Nonlinear Solid Mechanics, Montreal

FALTINSEN, O. M. (1990), *Sea loads on ships and offshore structures.* Cambridge University Press, Cambridge, UK

FILHO, A.N.Q; TANNURI, E.A. (2009), *DP offloading operation: a numerical evaluation of wave shielding effect.* 8th IFAC International Conference on Manoeuvring and control of marine craft. Guarujá, Brazil.

FONSECA, Í.A.; GASPAR, H.M. (2015), *An object-oriented approach for virtual prototyping in conceptual ship design*, 29th European Conference on Modelling and Simulation (ECMS) Conf., Varna

FONSECA, Í.A; OLIVEIRA, F.F, GASPAR, H.M. (2019), *Virtual prototyping and simulations of multibody marine operations using web-based technologies.* ASME 2019 38th International and Conference on Ocean, Offshore and Arctic Engineering, Glasgow, Scotland.

FONSECA, Í.A. (2018), *An Open and Collaborative Object-Oriented Taxonomy for Simulation of Marine Operations*, Master's Thesis, NTNU, Trondheim

FOSSEN, T.I. (2011), *Handbook of Marine Craft Hydrodynamics and Motion Control*, Trondheim, Wiley.

FOSSEN, T.I.; FJELLSTAD, O. (1995), *Nonlinear modelling of marine vehicles in 6 degrees of freedom*, Journal of Mathematical Modelling of Systems, Vol. 1, No. 1, 1995, Trondheim

GASPAR, H.M. (2017), *JavaScript Applied to Maritime Design and Engineering*, 16th Conf. Computer and IT Applications in the Maritime Industries (COMPIT), Cardiff

GASPAR, H.M. (2018), *Vessel.js: an open and collaborative ship design object-oriented library*, Int. Marine Design Conf. (IMDC), Helsinki

GEMILANG, G.M. (2015). *Feasibility study of selected riser concept in deep water and harsh environment.* Master thesis, University of Stavanger, Stavanger.

HE, B.; WANG, Y.; SONG, W.; TANG, W. (2015), *Design resource management for virtual prototyping in product collaborative design*, Proc. Institution of Mechanical Engineers, Part B: J. Eng. Manufacture 229

HU, Z; WANG, S; CHEN, G; CHAI, S; JIN, Y- (2016), *The effects of LNG-tank sloshing on the global motions of FLNG system.* International Journal of Naval Architecture and Ocean Engineering 9 (2017) 114-125

IRVINE, H.M. (1992). *Cable Structures. Volume 5.* Dover Publications, New York, NY.

JENSEN, J.J.; MANSOUR, A.E; OLSEN, A.S. (2004), *Estimation of ship motions using closed-form expressions.* Ocean Engineering 31(1):61-85

KIM, Y.; KIM, K.; KIM, J.; KIM, T.; SEO, M.; KIM, Y. (2011). *Time-domain analysis of nonlinear motion responses and structural loads on ships and offshore structures: development of WISH programs*, Inter J Nav Archit Oc Engng (2011), Seoul.

LEE, C; Newman, J. (2006). *WAMIT user Manual*. Chestnut Hill, MA, USA.

LEVANDER, K. (2012), *System based ship design*, NTNU Marine Technology, Trondheim

LEWIS, E.V. (1989), *Principles of Naval Architecture*, Jersey City, The society of naval architects and marine engineers.

LOC, M.B. (2014), *Time optimal trajectory design for unmanned underwater vehicle.* Ocean engineering 89 (2014) 69-81

MONTEIRO, T.G (2016), A knowledge-based approach for an open object-oriented library in ship design. Master's Thesis, NTNU, Ålesund.

NATVIG, B.J.; PENDERED, J.W. (1980), *Motion response of floating structures on regular waves*, Aker engineering A/S, Oslo

NEWMAN, J.N. (1935), *Marine hydrodynamics.* The MIT press. Massachusetts.

OLIVEIRA, F.F. (2019). *Implementation of Open Source Code for 6 Degrees of Freedom Simulations in Maritime Applications.* Research Report, Ship Design and Operation Lab, NTNU.

OLIVIER, N. (2013), *Dynamic analysis and design of mooring lines*, Master's Thesis, NTNU, Trondheim

OMBERG, H.; LARSEN, K. (1998), *Coupled analysis of a floater motion and mooring dynamics for a turret-moored ship*. Statoil Research centre and MARINTEK, Trondheim.

SANFILIPO, F.; HATLEDAL, L.I.; STYVE, A.; PETTERSEN, K.Y.; ZHANG, H. (2015). *Integrated Flexible Maritime Crane Architecture for the Offshore Simulation Centre (OSC).* July 2015. IEEE Journal Oceanic Engineering. DOI: 10.1109/JOE.2015.2439891

WAGNER, G. (2016), *Introduction to simulation using JavaScript*, 2016 Winter simulation conference, Cottbus.

WINGEREI, K. (2008), *Dynamics and damping in mooring lines*, Master's Thesis, University of Oslo, Oslo

WITTENBURG, G. (1977), *Dynamics of multibody systems*, Teubner, Springer.

# APPENDIX A

## Additional calculations for the calculation of motion with differential equations

In order to find the transformation matrix, this diagonal matrix is used $J(\eta) = \begin{bmatrix} J_1(\eta) & 0 \\ 0 & J_2(\eta) \end{bmatrix}$ where the matrices $J_1(\eta)$ and $J_2(\eta)$ forming the diagonal are the so-called Euler angle transformations (Fossen, 2011).

$$\begin{aligned} \boldsymbol{J}_1(\boldsymbol{\eta}) &= \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \\ \boldsymbol{J}_2(\boldsymbol{\eta}) &= \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi/\mathrm{c}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix} \end{aligned}$$

The parameters used in the equation of motion are defined as follows. $M_{RB}$ can be obtained from (Fossen, 1995) by using the Lagrange approach. (Salvensen, 1970)

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & -m\boldsymbol{S}(\boldsymbol{r}_G) \\ m\boldsymbol{S}(\boldsymbol{r}_G) & \boldsymbol{I}_0 \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{xz} \\ mz_G & 0 & -mx_G & -I_{yx} & I_y & -I_{yz} \\ -my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix}$$

Being $I_0$ the inertia tensor composed of $I_x, I_y$ $and$ $I_z$ which are the moments of inertia about the X, Y and Z axes and the $I_{xy} = I_{yx}$, $I_{zx} = I_{xz}$ $and$ $I_{zy} = I_{yz}$ the products of inertia.

Also $C_{RB}(v)$ can be obtained according to Fossen, 1995 using the Newton approach:

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} m\boldsymbol{S}(\boldsymbol{\nu}_2) & -m\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{S}(\boldsymbol{r}_G) \\ m\boldsymbol{S}(\boldsymbol{r}_G)\boldsymbol{S}(\boldsymbol{\nu}_2) & -\boldsymbol{S}(\boldsymbol{I}_0\,\boldsymbol{\nu}_2) \end{bmatrix} = \begin{bmatrix} C_{11}(v) & C_{12}(v) \\ C_{21}(v) & C_{22}(v) \end{bmatrix}$$

Where $r_G = [x_G, y_G, z_G]^T$ is the centre of gravity and the C values.

$$C_{11}(v) = \begin{bmatrix} 0 & -mv_{r,z} & mv_{r,y} \\ mv_{r,z} & 0 & -mv_{r,x} \\ -mv_{r,y} & mv_{r,x} & 0 \end{bmatrix}$$

$$C_{12}(v) = \begin{bmatrix} v_{r,z}p_{G,b,z} + v_{r,y}p_{G,b,y} & -v_{r,y}p_{G,b,x} & -v_{r,z}p_{G,b,x} \\ -v_{r,x}p_{G,b,y} & v_{r,z}p_{G,b,z} + v_{r,x}p_{G,b,x} & -v_{r,z}p_{G,b,y} \\ -v_{r,x}p_{G,b,z} & -v_{r,y}p_{G,b,z} & v_{r,y}p_{G,b,y} + v_{r,x}p_{G,b,x} \end{bmatrix}$$

$$C_{21}(v) = \begin{bmatrix} -v_{r,z}p_{G,b,z} - v_{r,y}p_{G,b,y} & v_{r,x}p_{G,b,y} & v_{r,x}p_{G,b,z} \\ v_{r,y}p_{G,b,x} & -v_{r,z}p_{G,b,z} - v_{r,x}p_{G,b,x} & v_{r,y}p_{G,b,z} \\ v_{r,z}p_{G,b,x} & v_{r,z}p_{G,b,y} & -v_{r,y}p_{G,b,y} - v_{r,x}p_{G,b,x} \end{bmatrix}$$

$$C_{22}(v) = \begin{bmatrix} 0 & (I_{b,xz}v_{r,x} + I_{b,yz}v_{r,y} + I_{b,zz}v_{r,z}) & -(I_{b,xy}v_{r,x} + I_{b,yy}v_{r,y} + I_{b,yz}v_{r,z}) \\ -(I_{b,xz}v_{r,x} + I_{b,yz}v_{r,y} + I_{b,zz}v_{r,z}) & 0 & (I_{b,xx}v_{r,x} + I_{b,xy}v_{r,y} + I_{b,xz}v_{r,z}) \\ (I_{b,xy}v_{r,x} + I_{b,yy}v_{r,y} + I_{b,yz}v_{r,z}) & -(I_{b,xx}v_{r,x} + I_{b,xy}v_{r,y} + I_{b,xz}v_{r,z}) & 0 \end{bmatrix}$$

Where the added mass matrix is (Lewis, 1988)

$$M_A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{bmatrix}$$

The damping matrix is:

$$B = \begin{bmatrix} B_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho L T C_D & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho A_W C_D & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & B_{66} \end{bmatrix}$$

And the restoring matrix is:

$$C = \rho g \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_W & 0 & -A_W x_F & 0 \\ 0 & 0 & 0 & \nabla \overline{GM_T} & 0 & 0 \\ 0 & 0 & -A_w x_F & 0 & \nabla \overline{GM_L} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

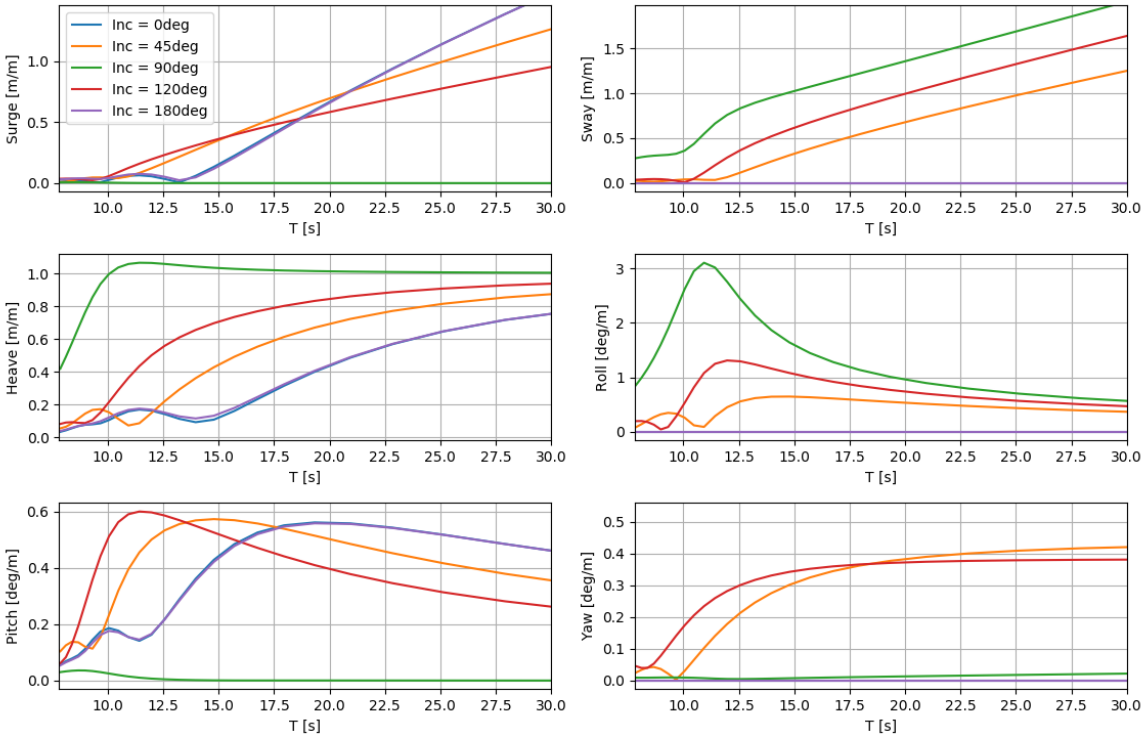
Where $x_F$ is the location of the centre of gravity with respect to the origin in the body frame. For free floating bodies, restoring ‘springs’ are only considered in heave, roll and pitch motions. While the gravity force is expressed as:

$$g = [0, 0, -g, 0, 0, 0]^T$$

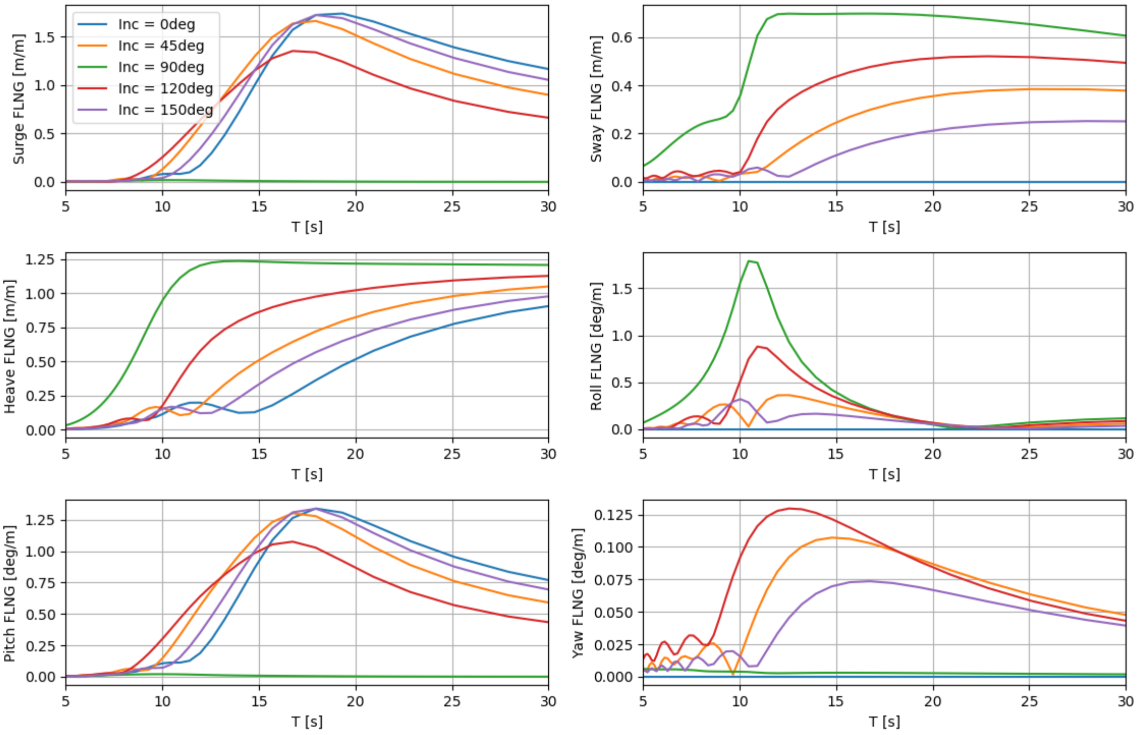
# APPENDIX B

## RAOs for the cases studied

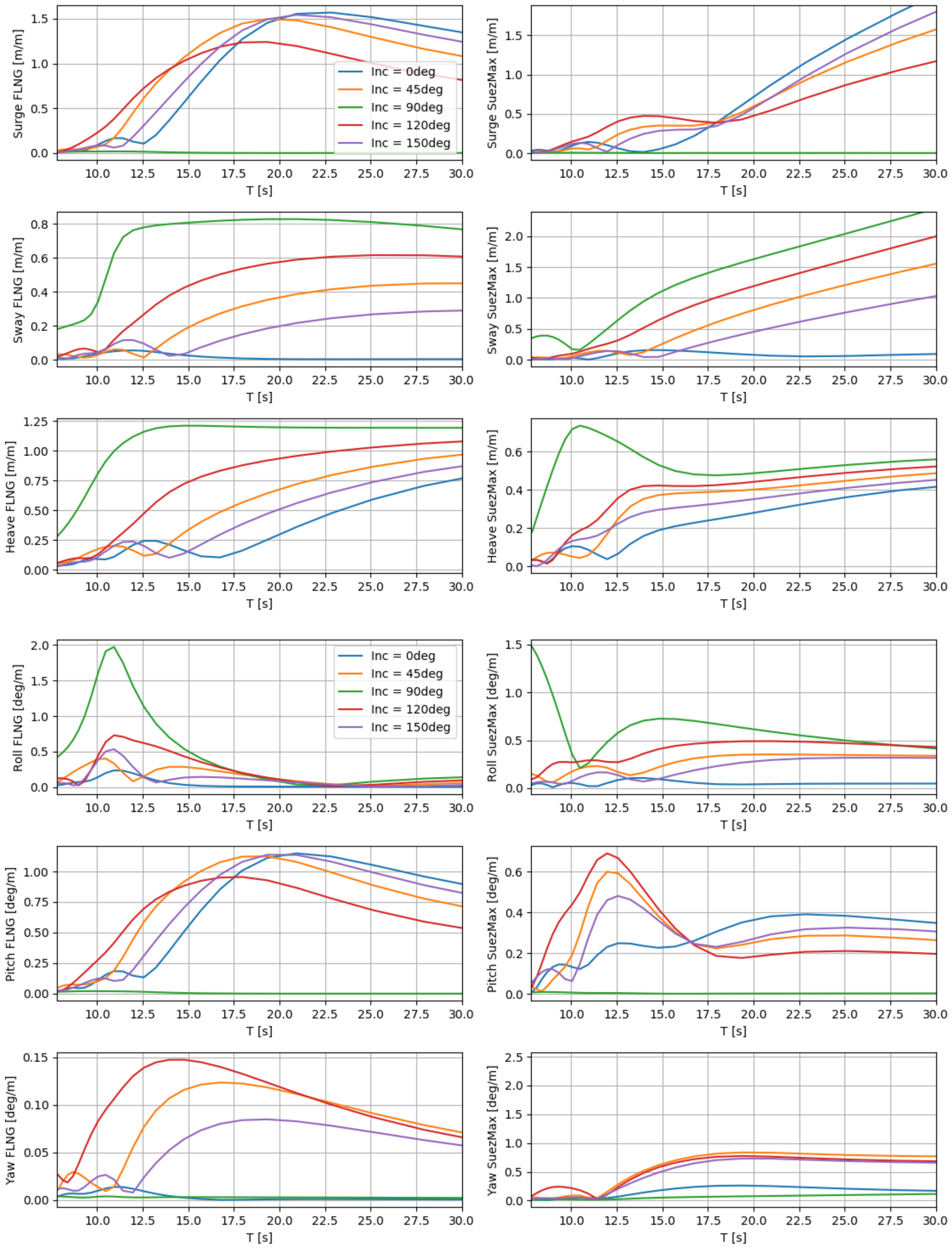
Case 3. One ship and no mooring



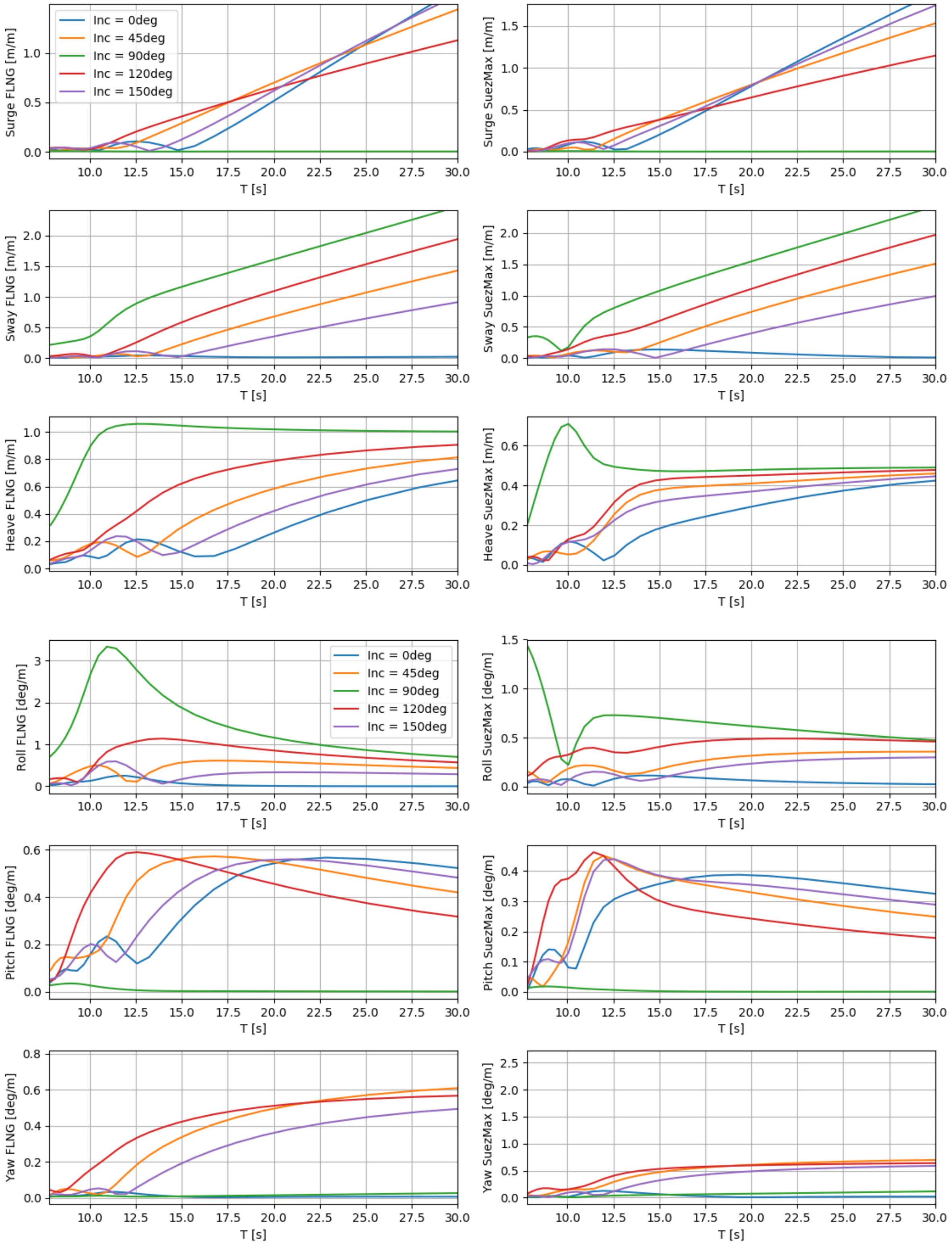
Case4. One ship with mooring



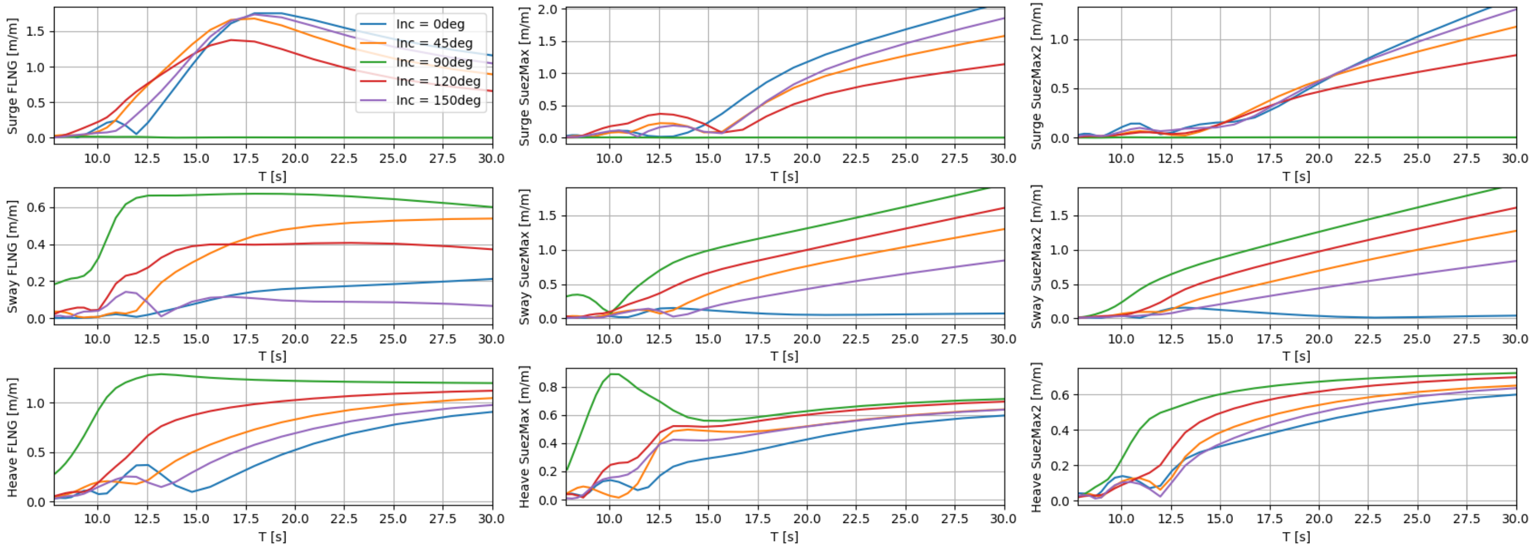
Case 5.1. Multibody with two ships, mooring and no hawsers

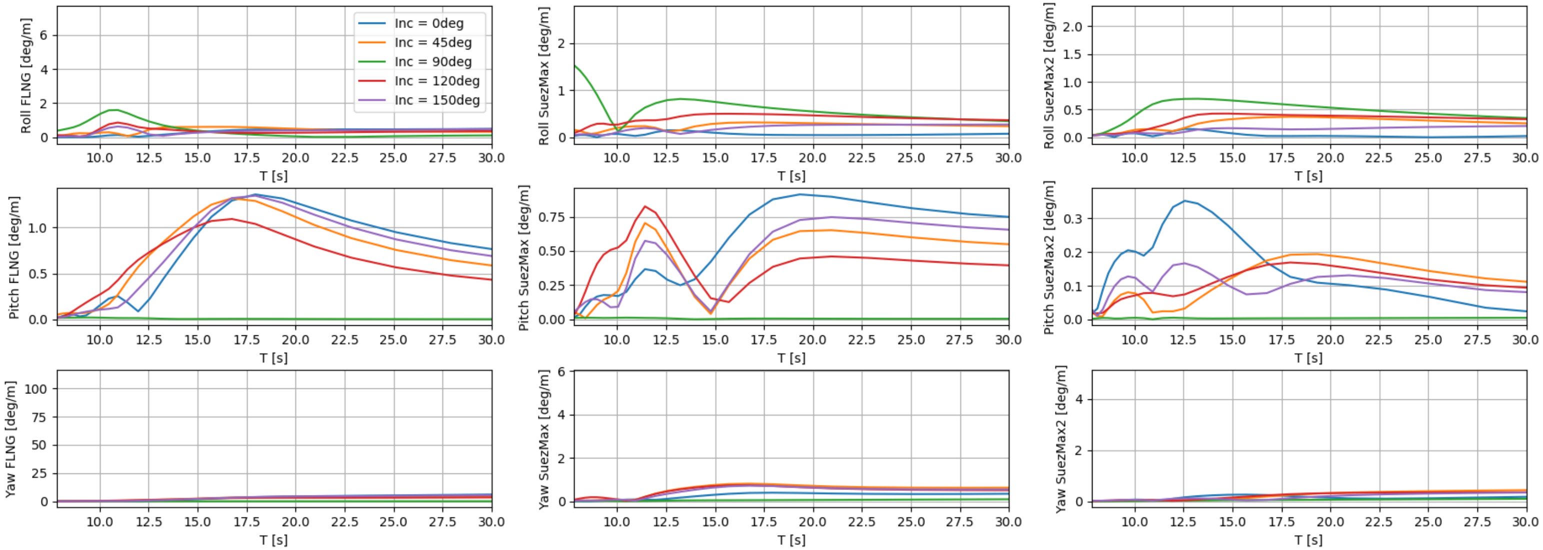


Case 5.2. Multibody with two ships, no mooring and no hawsers

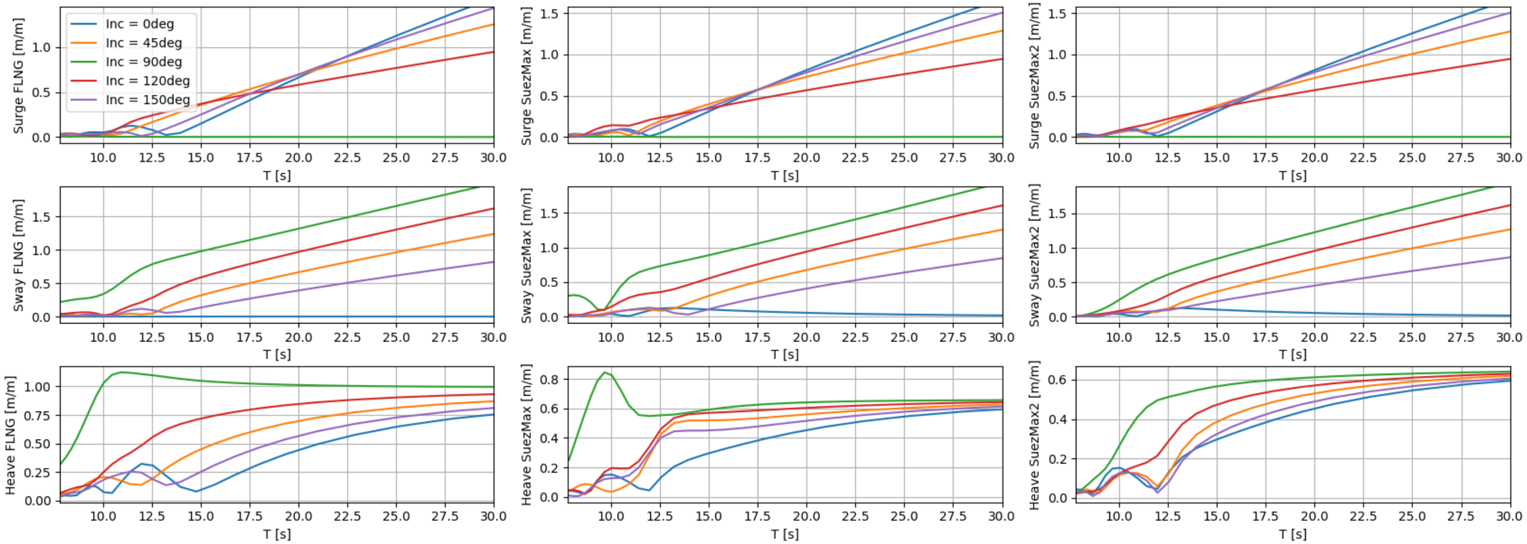


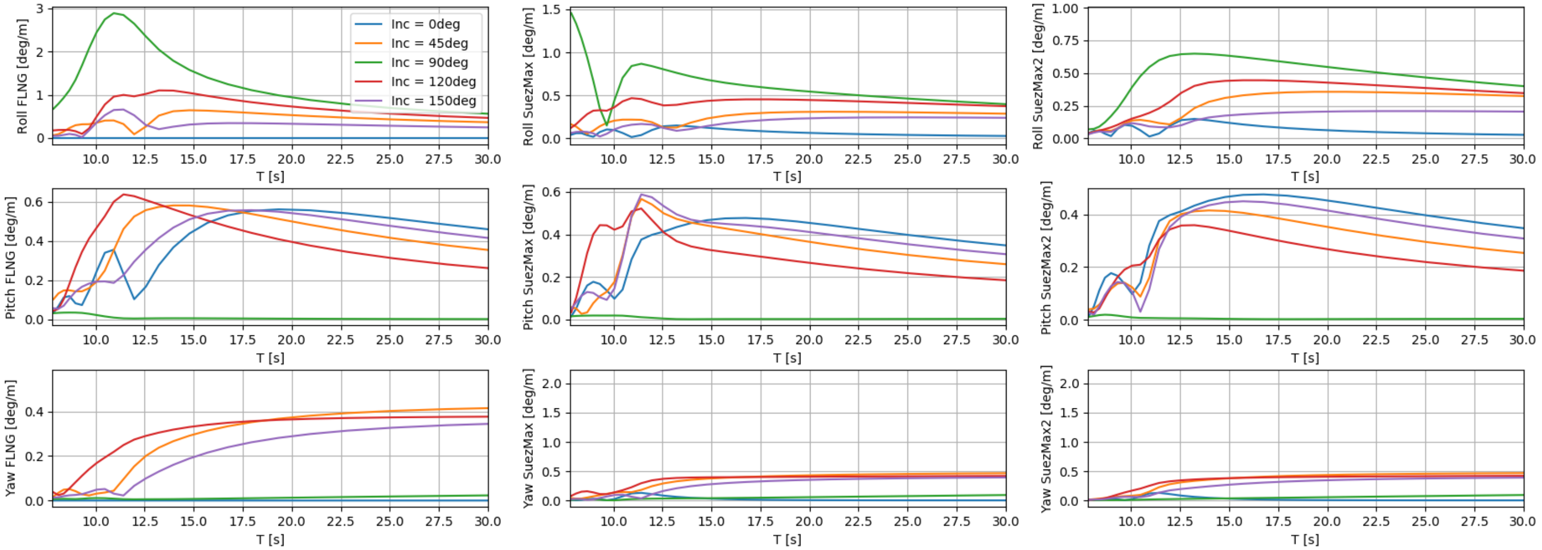
Case 5.3. Multibody with two ships, mooring and no hawsers



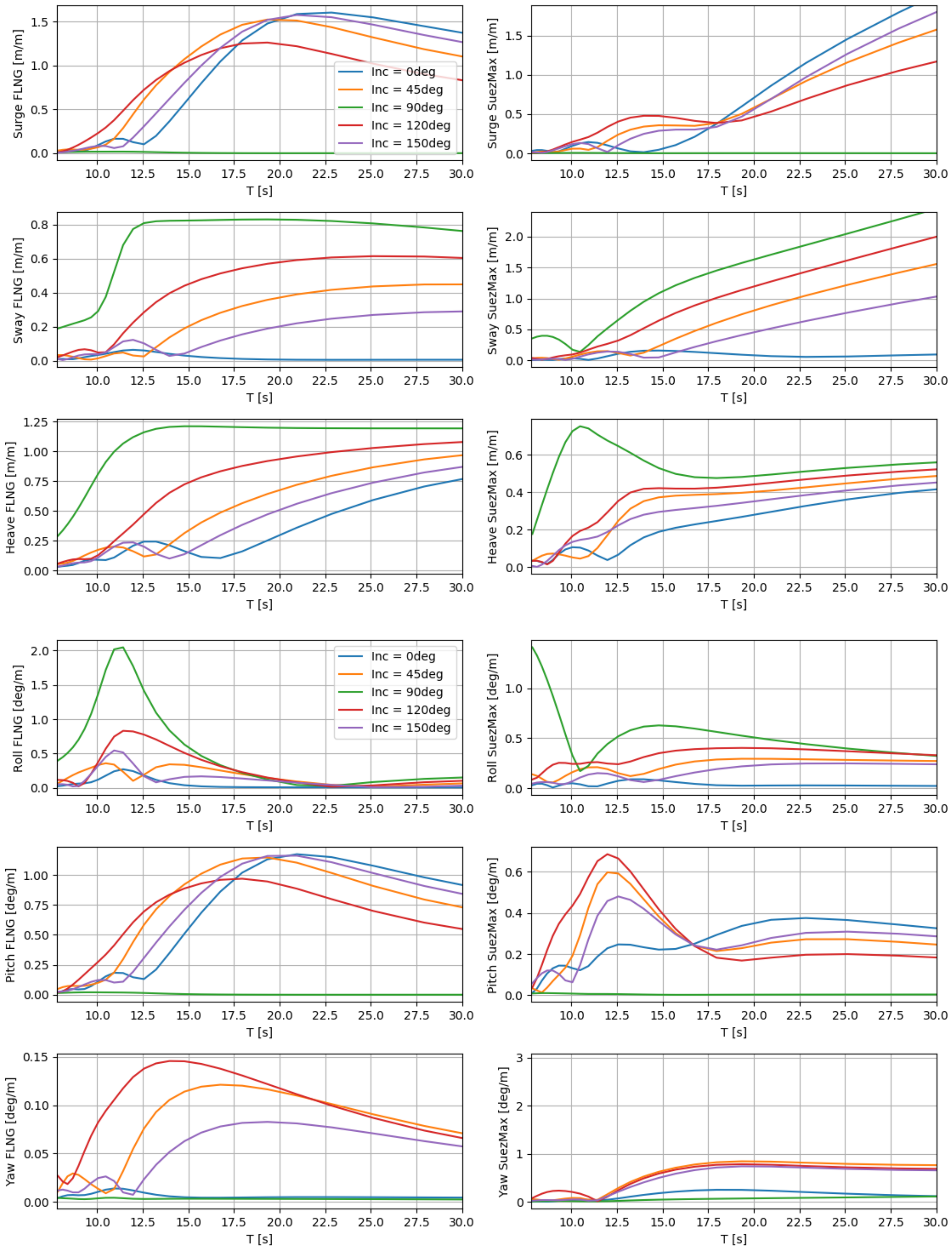


Case 5.4. Multibody with two ships, no mooring and no hawsers

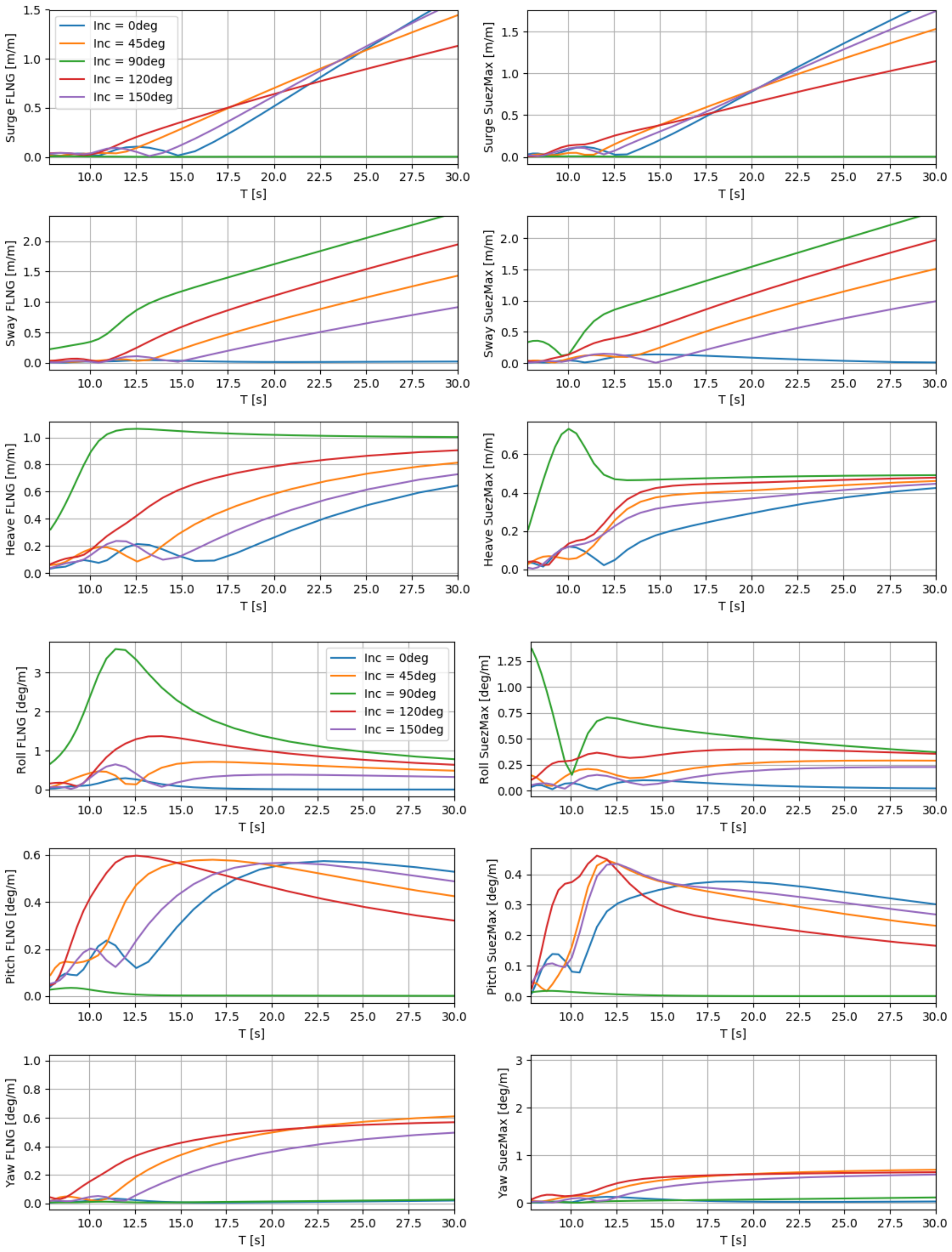




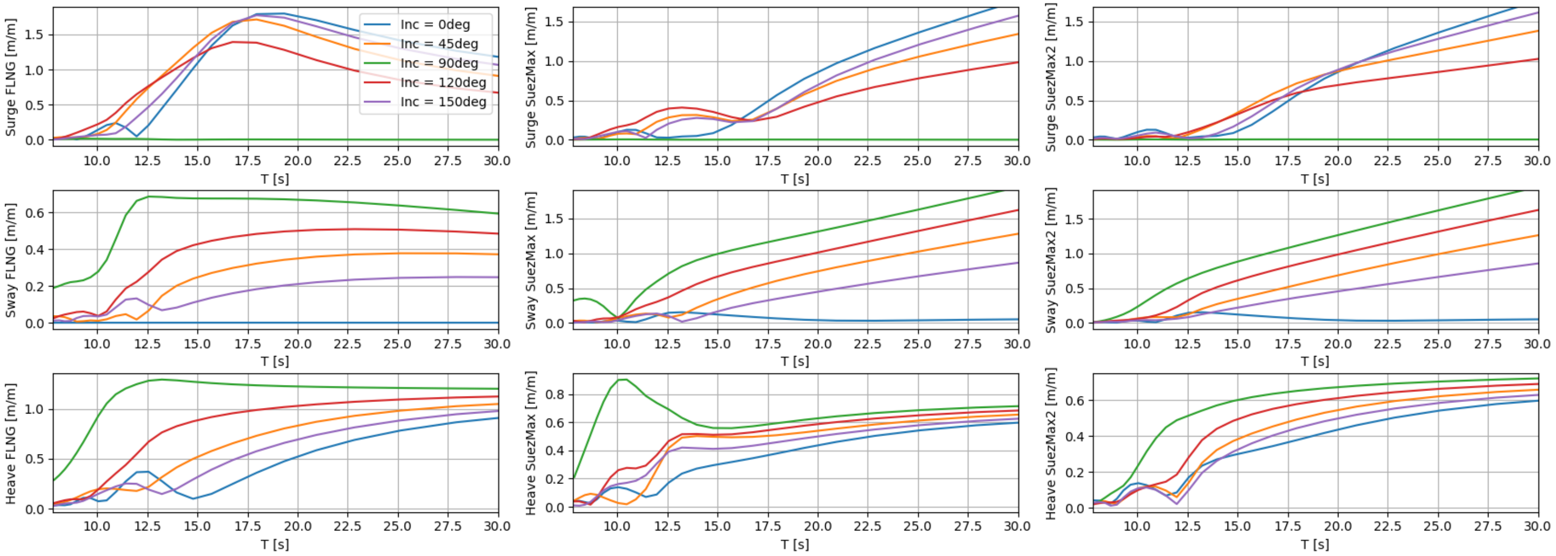
Case 6.1. Multibody with two ships, mooring and hawsers

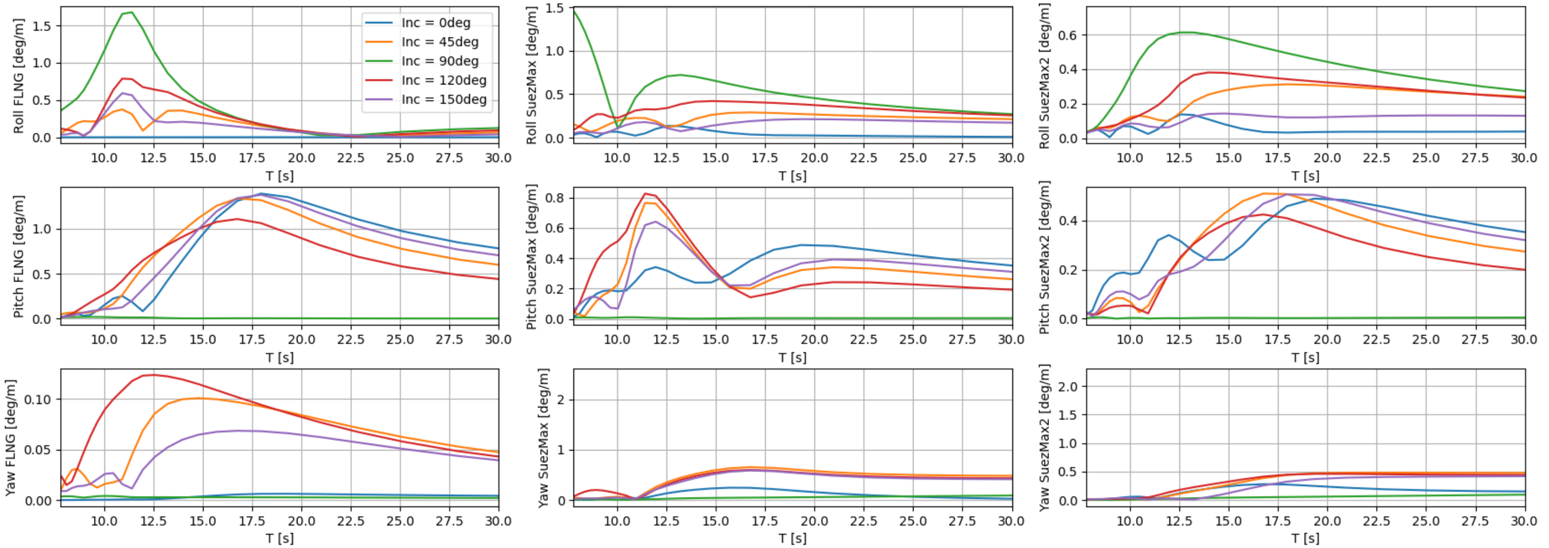


Case 6.2. Multibody with two ships, no mooring and hawse

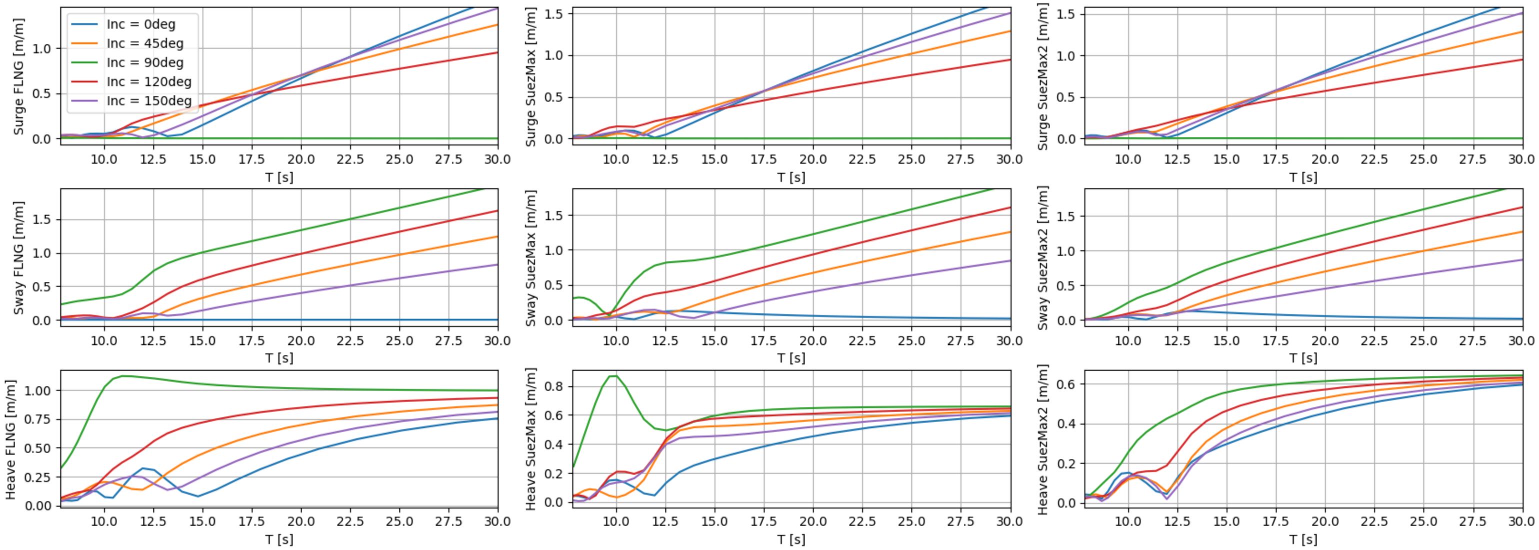


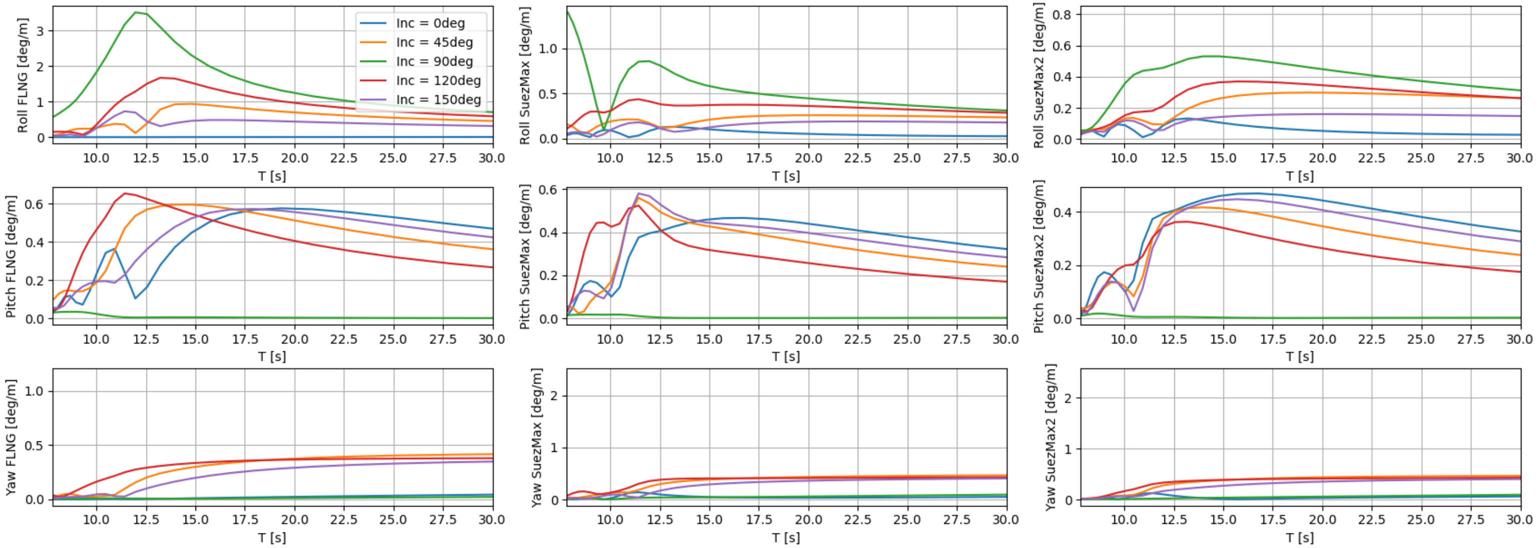
Case 6.3. Multibody with two ships, mooring and hawsers





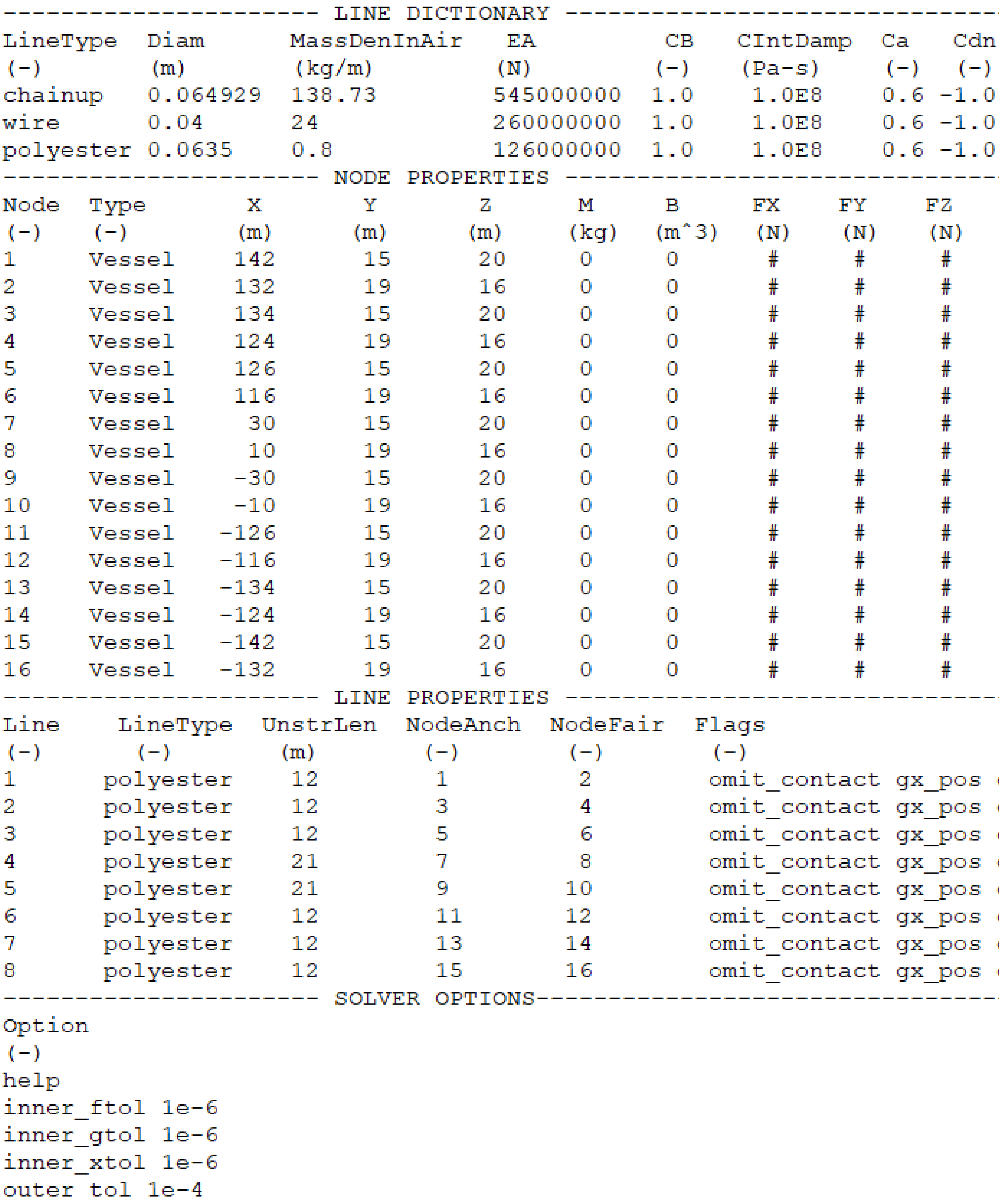
## Case 6.4. Multibody with two ships, no mooring and hawsers





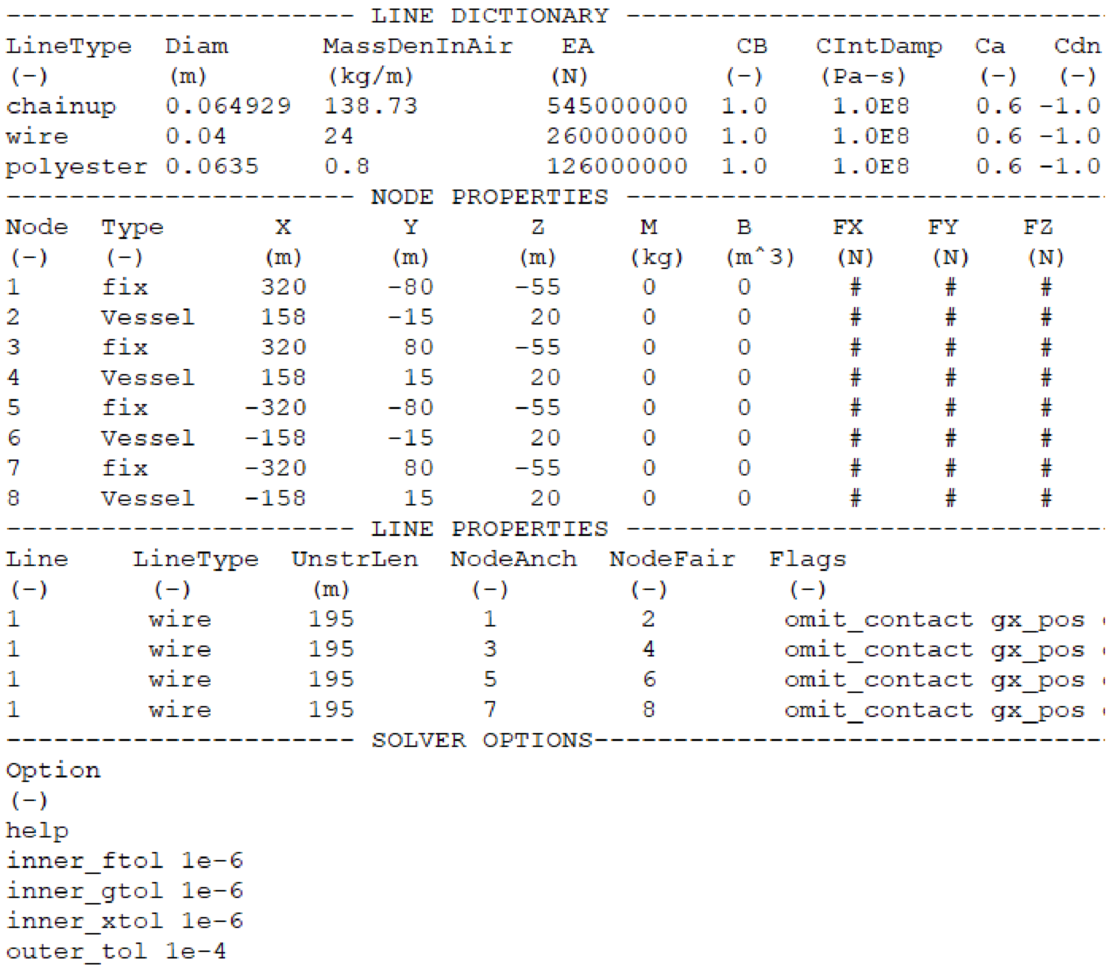
# APPENDIX C

## Hawsers input for MAP++



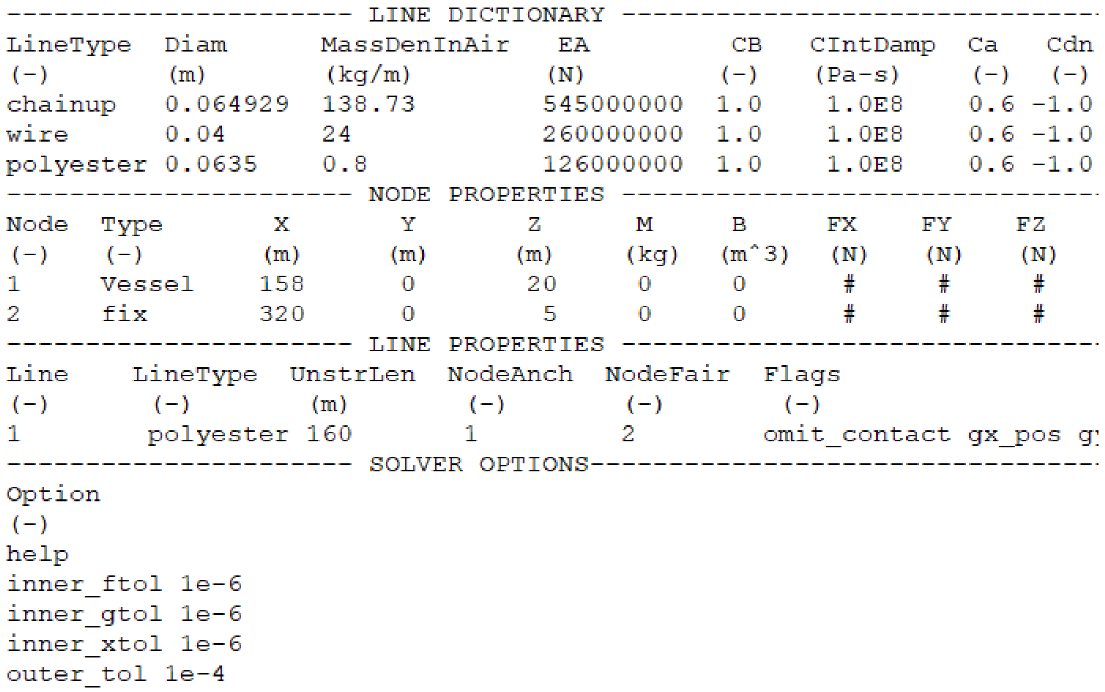
```
---------------------- LINE DICTIONARY -------------------------------
LineType  Diam      MassDenInAir  EA          CB    CIntDamp  Ca   Cdn
(-)       (m)       (kg/m)        (N)         (-)   (Pa-s)    (-)  (-)
chainup   0.064929  138.73        545000000   1.0    1.0E8    0.6 -1.0
wire      0.04      24            260000000   1.0    1.0E8    0.6 -1.0
polyester 0.0635    0.8           126000000   1.0    1.0E8    0.6 -1.0
---------------------- NODE PROPERTIES -------------------------------
Node  Type        X       Y       Z      M     B      FX    FY    FZ
(-)   (-)        (m)     (m)     (m)    (kg)  (m^3)  (N)   (N)   (N)
1     Vessel     142      15      20     0     0      #     #     #
2     Vessel     132      19      16     0     0      #     #     #
3     Vessel     134      15      20     0     0      #     #     #
4     Vessel     124      19      16     0     0      #     #     #
5     Vessel     126      15      20     0     0      #     #     #
6     Vessel     116      19      16     0     0      #     #     #
7     Vessel      30      15      20     0     0      #     #     #
8     Vessel      10      19      16     0     0      #     #     #
9     Vessel     -30      15      20     0     0      #     #     #
10    Vessel     -10      19      16     0     0      #     #     #
11    Vessel    -126      15      20     0     0      #     #     #
12    Vessel    -116      19      16     0     0      #     #     #
13    Vessel    -134      15      20     0     0      #     #     #
14    Vessel    -124      19      16     0     0      #     #     #
15    Vessel    -142      15      20     0     0      #     #     #
16    Vessel    -132      19      16     0     0      #     #     #
---------------------- LINE PROPERTIES -------------------------------
Line    LineType  UnstrLen  NodeAnch  NodeFair  Flags
(-)      (-)       (m)       (-)       (-)       (-)
1      polyester    12        1         2         omit_contact gx_pos
2      polyester    12        3         4         omit_contact gx_pos
3      polyester    12        5         6         omit_contact gx_pos
4      polyester    21        7         8         omit_contact gx_pos
5      polyester    21        9        10         omit_contact gx_pos
6      polyester    12        11       12         omit_contact gx_pos
7      polyester    12        13       14         omit_contact gx_pos
8      polyester    12        15       16         omit_contact gx_pos
---------------------- SOLVER OPTIONS---------------------------------
Option
(-)
help
inner_ftol 1e-6
inner_gtol 1e-6
inner_xtol 1e-6
outer tol 1e-4
```

## Mooring input for MAP++



```
---------------------- LINE DICTIONARY ---------------------------------
LineType  Diam      MassDenInAir  EA            CB    CIntDamp   Ca   Cdn
(-)       (m)       (kg/m)        (N)           (-)   (Pa-s)     (-)  (-)
chainup   0.064929  138.73        545000000     1.0    1.0E8     0.6 -1.0
wire      0.04      24            260000000     1.0    1.0E8     0.6 -1.0
polyester 0.0635    0.8           126000000     1.0    1.0E8     0.6 -1.0
---------------------- NODE PROPERTIES ---------------------------------
Node   Type      X       Y       Z      M     B     FX     FY     FZ
(-)    (-)      (m)     (m)     (m)    (kg)  (m^3)  (N)    (N)    (N)
1      fix      320     -80     -55     0     0      #      #      #
2      Vessel   158     -15      20     0     0      #      #      #
3      fix      320      80     -55     0     0      #      #      #
4      Vessel   158      15      20     0     0      #      #      #
5      fix     -320     -80     -55     0     0      #      #      #
6      Vessel  -158     -15      20     0     0      #      #      #
7      fix     -320      80     -55     0     0      #      #      #
8      Vessel  -158      15      20     0     0      #      #      #
---------------------- LINE PROPERTIES ---------------------------------
Line     LineType  UnstrLen  NodeAnch  NodeFair  Flags
(-)       (-)        (m)       (-)       (-)      (-)
1          wire      195        1         2         omit_contact gx_pos
1          wire      195        3         4         omit_contact gx_pos
1          wire      195        5         6         omit_contact gx_pos
1          wire      195        7         8         omit_contact gx_pos
---------------------- SOLVER OPTIONS-----------------------------------
Option
(-)
help
inner_ftol 1e-6
inner_gtol 1e-6
inner_xtol 1e-6
outer_tol 1e-4
```

## Example input for MAP++



```
---------------------- LINE DICTIONARY ---------------------------------
LineType  Diam      MassDenInAir  EA            CB    CIntDamp   Ca   Cdn
(-)       (m)       (kg/m)        (N)           (-)   (Pa-s)     (-)  (-)
chainup   0.064929  138.73        545000000     1.0    1.0E8     0.6 -1.0
wire      0.04      24            260000000     1.0    1.0E8     0.6 -1.0
polyester 0.0635    0.8           126000000     1.0    1.0E8     0.6 -1.0
---------------------- NODE PROPERTIES ---------------------------------
Node   Type      X       Y       Z      M     B     FX     FY     FZ
(-)    (-)      (m)     (m)     (m)    (kg)  (m^3)  (N)    (N)    (N)
1      Vessel   158      0       20     0     0      #      #      #
2      fix      320      0        5     0     0      #      #      #
---------------------- LINE PROPERTIES ---------------------------------
Line     LineType  UnstrLen  NodeAnch  NodeFair  Flags
(-)       (-)        (m)       (-)       (-)      (-)
1          polyester 160       1         2        omit_contact gx_pos g
---------------------- SOLVER OPTIONS-----------------------------------
Option
(-)
help
inner_ftol 1e-6
inner_gtol 1e-6
inner_xtol 1e-6
outer_tol 1e-4
```

# APPENDIX D

## Digital appendix

Some of the examples developed on this thesis will be published in the Vessel.js library (http://www.vesseljs.org) in order to keep being developed by other users and so the reader can see a taste of what this thesis is about. The cases published on the Vessel.js Github page will allow the user to reuse the code developed on these cases and create the maritime and subsea operations mentioned on Section 5.

# APPENDIX E

## Conference paper draft

The following conference paper is a draft developed during this work. This paper synthetizes the work done in the thesis in the most relevant paragraphs in order to let the reader understand the overall meaning of this thesis in a shorter period of time. This is written as a requirement for the delivery of this current study.

# Open Source Simulation for Virtual Maritime Operations

**Sergi E. Miquel, NTNU,** Alesund/Norway, sergiescamilla@gmail.com
**Henrique M. Gaspar,** NTNU, Alesund/Norway, henrique.gaspar@ntnu.no
**Icaro A. Fonseca,** NTNU, Alesund/Norway, icaroaragao@outlook.com
**Daniel P. Vieira,** USP, Sao Paulo/Brazil, dpratavieira@gmail.com

## Abstract

Marine operations are rapidly coming in demand over de past years, and simulations are commonly run in order to estimate the system's response in diverse scenarios before moving to a real domain. Open source simulation through a web-based design opens many windows for reinventing what nowadays is known as proprietary simulation software. This environment brings advantages such as rapid and collaborative development, wide compatibility with diverse operational systems, no need for powerful computers, friendly user interfaces, and it provides results quick to obtain and easy to share.

The scope of the project is to reuse and develop new objects in an existing open-source environment with structures and methods for vessel and marine operation simulation through a web-based platform by using JavaScript, an object-oriented programming language. The study will use these tools to build up and develop the current conceptual ship design library by modelling interaction between single bodies and creating new multibody models able to output different responses according to a range of various inputs, considering the different requirements and limitations for maritime operations.

By using a taxonomy for maritime operations simulations, different operations are chosen to simulate in an open source environment. In order to create case studies, these simulations are decomposed in single elements in order to understand their behaviour separately before making them interact with other elements to create a multibody. In the process, different disturbances are applied to the bodies according to the needs of the simulations and the level of complexity. The simulations are coded in JavaScript and visualized in a web environment in order to assess them. A discussion for the efficiency of the method and the utility of these simulations is carried.

## 1. Maritime Operations in Open Source

In order to successfully carry or improve a current maritime operation within the estimated timeframe and costs, simulations are commonly run by designers and personnel before moving to a real domain, allowing the user to estimate what can go wrong for the different variables and contexts. This offers the possibility to create a wide variety of marine simulations within different sub-fields and goals considering various external disturbances given by the simulation requirements that can approximate the simulation to a real-life model.

Open source simulation through a web-based design opens many windows for reinventing what nowadays is known as proprietary simulation software, which deals with expensive licensing not affordable for everyone. Additionally, even if licensing is not a problem and the user obtains some results with the aim to share them, a couple hours could go by before the user has copied and pasted different graphs, tables and data in a PowerPoint presentation. This is normally due to the lack of compatibility of this specialized software, that cannot be displayed or run outside its environment.

The need of developing these new tools in a way that are attractive to users by giving them features such as reliability, accessibility and ease of use needs to be fulfilled at the same time it is open for manipulation and improvement, promoting this way a collaborative community, where the knowledge can be spread and revised by others. This study offers the possibility to expand this technology to a wide range of potential users that do not know about the advantages of open source simulation by convincing them that reliable operations can be reproduced in a web-based design environment, which is widely compatible with everyone's computers and can be improved and modified by any developer who wants to contribute in the process.

For all these reasons, there is consequently a boom on the demand of users using multibody simulations in order to represent on a virtual environment one of the wide ranges of maritime operations that are rocketing our economies. Reached this point, one realises that these kinds of simulations are the key to represent any imaginable maritime operation by using as many single objects as needed, which combined will form a multibody configuration. Furthermore, if these single objects, from now on called entities, are assigned an activity and this activity is displayed over time, a proper maritime operation is obtained in one or various scenarios where the user is able to decide where to place a boat, the subsea equipment, the risers and the mooring lines for later obtaining a response out of that information.

Another issue that this study addresses is related to the reliability of the data in these kinds of simulations. What usually happens in these early stage libraries is that the data is sometimes not the most accurate, as in this stage of the design phase, the developer tends to prioritize other features such as displaying a good visualization in order to catch the attention of the user instead of focusing too much on the physics. This is why even though the simulations can visually look reliably enough, when someone wants to obtain some accurate data, some simulators cannot be entirely trusted, as they take approximations to describe the physics. This study will focus part of its attention to obtain the data from one of the most reliable tools for analysing wave interactions with vessels and offshore platforms, which is WAMIT. Finally, the aim is also to take the user to a more understanding graphical user interface (GUI) when dealing with WAMIT output file.

Given this problem, a solution if proposed in order to integrate multibody operations in an open source environment using the most precise way found to calculate the motions and reusing parts of a conceptual ship design library, Vessel.js, which has most of its functionalities based on an object-oriented approach, where the code is developed as a collection of different objects, that can be put together with the aim to accomplish a task and represent as close as possible characteristics and functionalities of a ship. These objects work as tools that can be pulled from different scripts in order to be used in a new one. They can handle new information, call methods, functions, make calculations and various other operations.

Every system of the vessel can be represented in an object and can be simulated as a block, obtaining as a result, a whole ship composed by all the systems necessary to work. This is based in the System based ship design methodology (Lavender, 2012) which follows a bottom up approach, starting with all the payload and ship functions, assigning them an area and a volume and building up a whole vessel. Other applications are also made with the available tools.

## 2. Obtention of the equations of motion

The motions for the simulations on this study are obtained differently to the examples currently working on the Vessel.js library. For this case, and considering that multibody operations are on steak, the methods used in the current library are not valid, as the motion of two vessels side by side are influenced by each other. Also, linear external restoring matrices are applied in order to account for mooring lines and hawsers. These operations have a higher level of complexity and are difficult to run in a web browser. It is for this reason that WaveAnalysisMIT (WAMIT) is used. This software is a complete tool to solve the diffraction/radiation problem in order to analyse the interaction between waves and structures and is widely used because of its high degree of accuracy and efficiency. This software solves the velocity potential in the wetted surface of the structure, and it is based on linear second-order potential theory which solves the problem by using the Boundary Element Method (BEM) with three-dimensional panel elements.

The complexity added on this analysis is that the equations of motion are not evaluated individually, and each vessel depends on coefficients generated by the vessel on its side. Therefore, a new equation of motion is expressed as seen in Equation 1.

$$\begin{bmatrix} [M+a(\infty)]_{i,i} & a(\infty)_{i,j} \\ a(\infty)_{j,i} & [M+a(\infty)]_{j,j} \end{bmatrix} \begin{Bmatrix} \ddot{\xi}_i \\ \ddot{\xi}_j \end{Bmatrix} + \begin{bmatrix} D_{i,i} & D_{i,j} \\ D_{j,i} & D_{j,j} \end{bmatrix} \begin{Bmatrix} \dot{\xi}_i \\ \dot{\xi}_j \end{Bmatrix} f\left( \begin{Bmatrix} \dot{\xi}_i \\ \dot{\xi}_j \end{Bmatrix} \right) + \\ \begin{bmatrix} K_{i,i} & K_{i,j} \\ K_{j,i} & K_{j,j} \end{bmatrix} \begin{Bmatrix} \xi_i \\ \xi_j \end{Bmatrix} + \int_0^t \begin{bmatrix} h(t-\tau)_{i,i} & h(t-\tau)_{i,j} \\ h(t-\tau)_{j,i} & h(t-\tau)_{j,j} \end{bmatrix} \begin{Bmatrix} \dot{\xi}_i \\ \dot{\xi}_j \end{Bmatrix} d\tau = \begin{Bmatrix} F_i \\ F_j \end{Bmatrix} \qquad (1)$$

Where the subscripts *i* and *j* correspond to one and another vessel respectively, being *i*,*j* the influence that vessel *i* makes upon vessel *j*, M the mass matrix, a(∞), the added mass, damping (D), restoring matrices (K) and coupled retardarion matrix ($h(t-\tau)$), as well as F for external forces.

The influence of mooring lines and hawsers will also make of this operation a special one, as these need to be considered, as the motion is constrained by these lines. This influence is hard to calculate as it depends on the coupled motions of two ships with different tonnage and loading conditions. Therefore, this restraining motion through hawsers and mooring lines is simulated with Moring Analysis Program (MAP++) software. An input file with the characteristics of the mooring lines and their position in the vessel is required before obtaining the restoring matrices needed.

Finally, the last feature when configuring a WAMIT run for side by side interaction an additional body consisting of an artificial rectangular damping lid is placed on the free surface between the two vessels. This gap lid is placed in order to cover enough portion between the two vessels so that it is physically possible to damp the important resonant modes. So, by placing this lid, WAMIT considers the resonance between this gap and the shadow effect made by the coupled motions of both bodies. What it does not consider are the appendices physically attached to the ship such as fenders or hawsers, that must be added through external linearized restorative matrices.

In order to obtain these motions that require a high computational power, the simulations are run with a software developed by a collaboration with various Brazilian universities and the oil company PETROBRAS. This software incorporates WAMIT and uses diverse methods and algorithms able to make a coupled analysis of different offshore systems, as well as lines and bodies together. This software is run in a powerful cluster located at the Numerical Offshore Basin (USP) in order to obtain enough computational power for the simulations required. Figure 1 shows an overall of the process required to obtain the vessels motions.

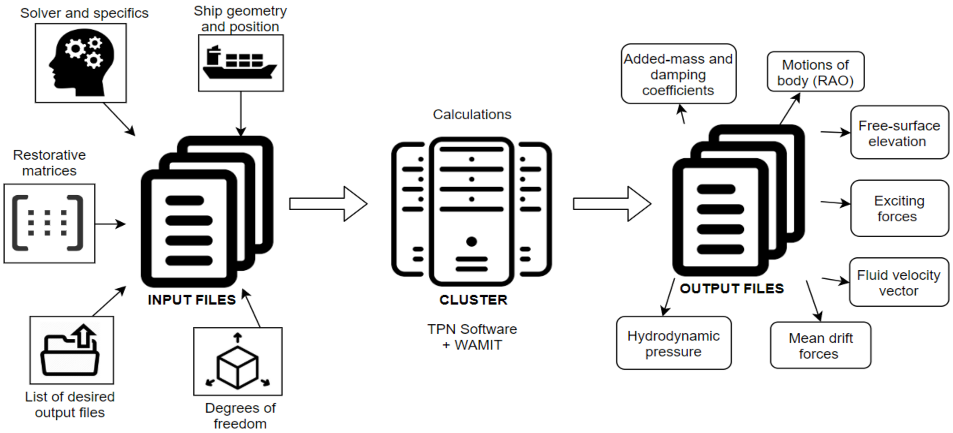


Figure 3. Process to obtain the vessels motions through a WAMIT run.

### 3. Virtual prototyping of the case studies.

An adaptation of He et al. (2015) taxonomy is adapted for ship design related studies in order to apply it to the virtual prototyping of maritime operations, (Fonseca and Gaspar, 2015; Fonseca et al., 2018). This is a useful tool to help designers build virtual models which involve the characteristics of physical products through an object-oriented design resource management for virtual prototyping in collaborative design. These technics work through a process of building, simulating and improving the models regarding requirements for function, appearance and other aspects of the products. According to this process, the design resource model can be classified in three basic types.

- Entity model (EM), which defines the physical product to be simulated, including design specification data and information about the product, as well as 2D and 3D models.
- State model (SM), which represents the EM exposed to internal and external state constrains and analyses the entity by assigning it a state.
- Process model (PM), which is an accumulation of the SMs, representing the model behaviour over time, from the initial to the final state and subjecting the EM to a dynamic constraint.

With this adaptation, entity models can be represented as any maritime system with all levels of detailing, such as well-defined 3D model of a ship with component specification or the characteristics and visualization of a mooring line. As for the state models of a ship, these can be represented as the entity models subjected to internal or external disturbances such as a wide range of environmental conditions, resistance of a vessel at certain speed with certain loading conditions at a given time. All these different scenarios are comprised in a case study as shown in Figure 2.

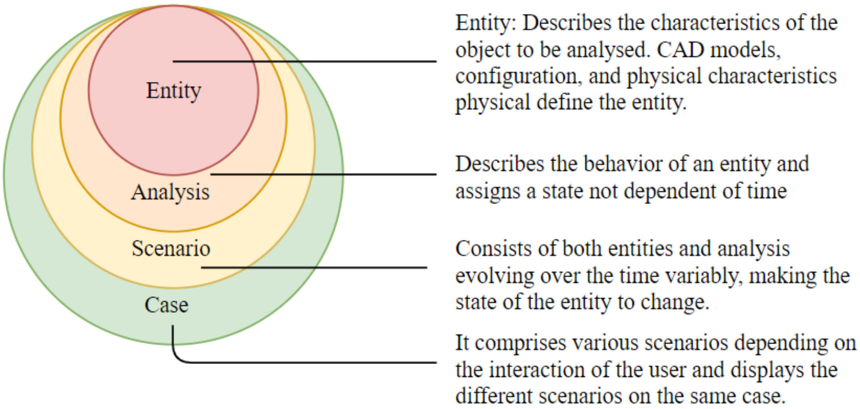

Entity
Analysis
Scenario
Case
Entity: Describes the characteristics of the object to be analysed. CAD models, configuration, and physical characteristics physical define the entity.
Describes the behavior of an entity and assigns a state not dependent of time
Consists of both entities and analysis evolving over the time variably, making the state of the entity to change.
It comprises various scenarios depending on the interaction of the user and displays the different scenarios on the same case.

Figure 4. Configuration of the model, from a single entity to an entire case study.

The entity models for the simulations will come defined by every element that separately works as a whole with a dependent behaviour. The single entities on this study will include an ocean, a seabed, a ship, a line, and additional objects placed on the seabed or elsewhere. These will be created using Three.js library. This level of detailing for the ship entity can include the definition of the shape of the ship due to being modularized. So, the whole entity can be created with "derived objects", which will have a parent called "base object". These derived objects placed in the right coordinates can create an approximation to the vessel. (Fonseca et alt., 2018).

A bit more complex is to simulate ropes which are composed by small segments following some physical properties, in this case, a catenary system or a mooring line with seabed-touching characteristics. A calculation in JavaScript is required in order to know the position for the vertices of the ropes (Oliveira F.F, 2018 et al.)

The way to provide motion to these entities is by obtaining it from the WAMIT runs done previously. To bring these WAMIT output text files into the web browser, the user finds a simple “Explore file” button on the web browser in order to introduce the motion output file from WAMIT with the RAO’s. As seen in Figure 3, a JavaScript program reads the data from the file, processes it and turns it into a useful JavaScript object containing the RAOs with the pertinent amplitude and phase, separating them in the different degrees of freedom.

This object contains “x” number of cases, “y” number of degrees of freedom (according to the number of ships), 30 different periods, and 25 heading for each one of these periods. Finally, the responses are taken individually from the first object and placed in the right hierarchical order to be easily called. Then, the user will introduce some conditions, a given wave period and a wave heading, and the JavaScript code will start to display the motion for those conditions, taking data from this last object in order to pass it into the vessel.

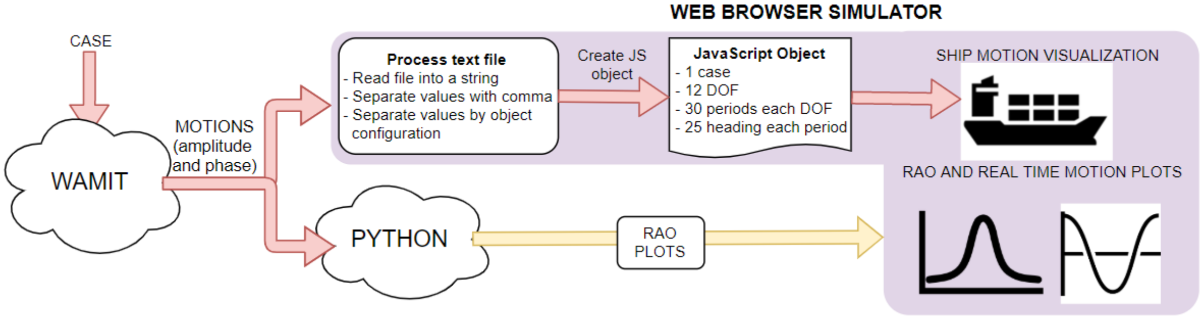

WEB BROWSER SIMULATOR
CASE
WAMIT
MOTIONS (amplitude and phase)
Process text file
- Read file into a string
- Separate values with comma
- Separate values by object configuration
Create JS object
JavaScript Object
- 1 case
- 12 DOF
- 30 periods each DOF
- 25 heading each period
SHIP MOTION VISUALIZATION
RAO AND REAL TIME MOTION PLOTS
PYTHON
RAO PLOTS

Figure 5. Interface from the WAMIT output motions to the web browser simulator.

Reached this point, the web simulator allows to provide the first analysis to every entity. Starting by giving motion to all the ships present in the simulation, being the source of movement for other entities attached to the ship. Until this point, only static behaviours have been considered, therefore in order to determine what happens on the simulation over time, the process model is defined. This represents a dynamic simulation that adds the time variation over the two previous models.

$$\eta_x = R \cdot a \cdot \cos(\omega t - \varphi) \tag{3.1}$$

Where

$\eta_x$ is vessel displacement (metres for surge, sway, heave and degree for roll, pitch, yaw)
R, $\varphi$ are the RAO amplitude and phase respectively.
$a$,$\omega$ are wave amplitude (meters) and the wave frequency (radians/second) respectively
t is time variable (in seconds).

The interactions such as rope attached to the bodies, will be following the attachment point set in the vessel by calculating the motion on that given point. On the other side, the same applies if a hawser is set or else if it is a mooring line, a fixed attachment point on the seabed is set. The motions displayed by the vessels will be already considering the hawsers and mooring lines that are going to be attached, as this is considered previously before running the motions software.

Once the entities, analysis and scenarios have been defined they are tested individually, each entity with its analysis performing a single scenario, for later joining all the entities necessary to create a multibody operation. As seen in Figure 4, this operation considering a side by side operation consists of three vessels with three different states interacting with each other. One can see the influences that the vessels have upon the other vessels and how their attachments affect the motions of the neighbouring vessels.

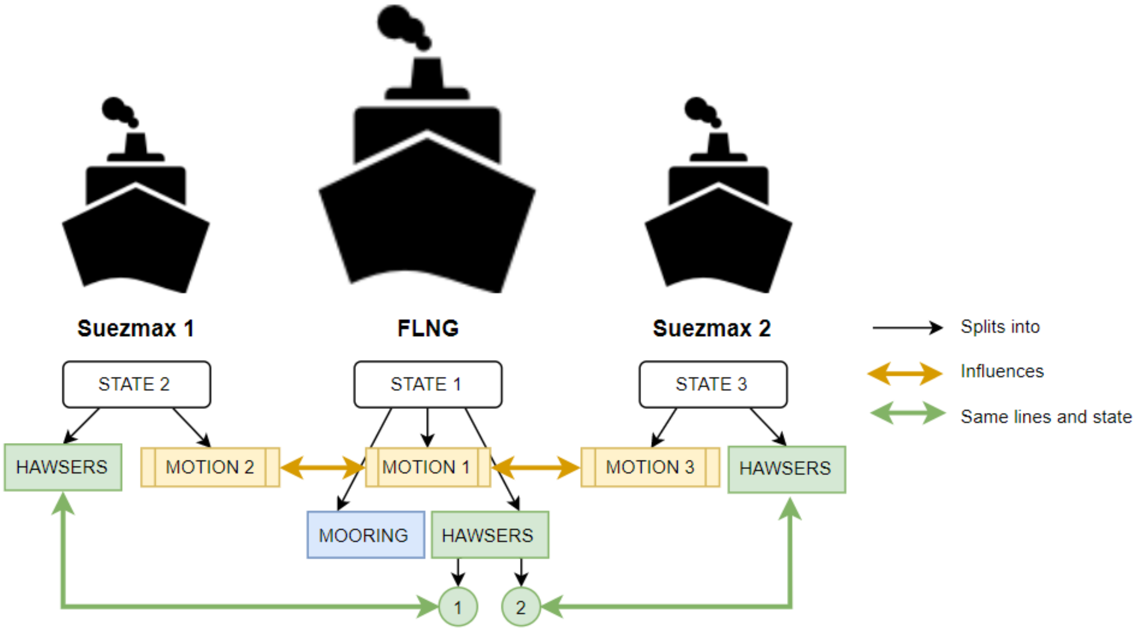


Figure 6. Definition of independent states and how these are correlated.

## 4. Simulation of multibody operations maritime and subsea

Defined all the entities, analysis, and possible scenarios, every case study can be defined. To do that, the number of ships to be studied hast to be chosen as well as the distribution of the operation. Also, the lines attached to each vessel, which will have a crucial role when calculating the motion of the ship. Several cases are run progressively as seen in Figure 5, starting with a simple case studio with a barge entity, adding later an ocean entity and simulating the radiation problem. The next step is to add a wave pattern analysis to the ocean entity in order to provide some 6 degrees of freedom linear motion to the ship, as a response of the wave analysis. After that, other behaviours such as adding new headings are performed. Finally, the addition of more entities such as hawsers and new vessels will bring the user to a very complete side by side operation.

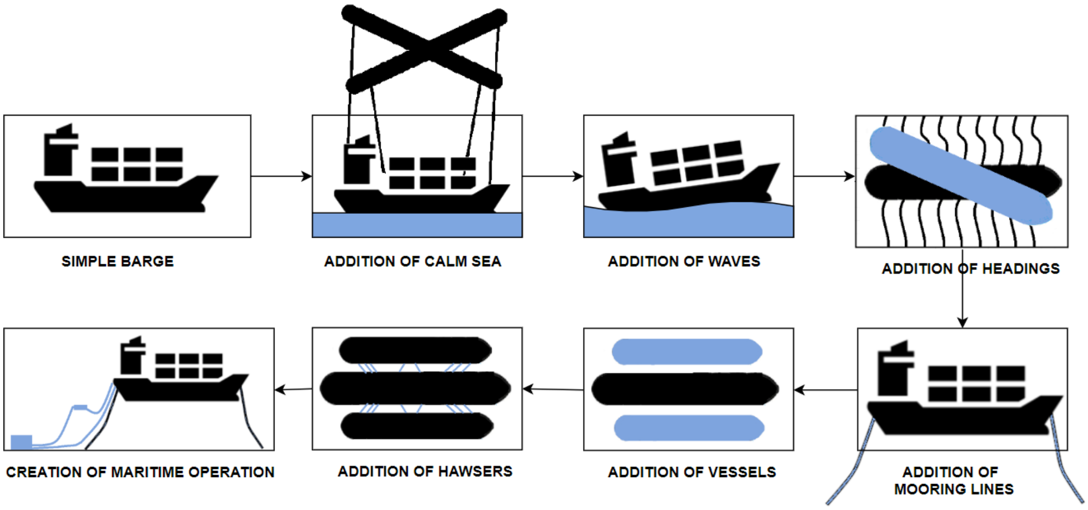
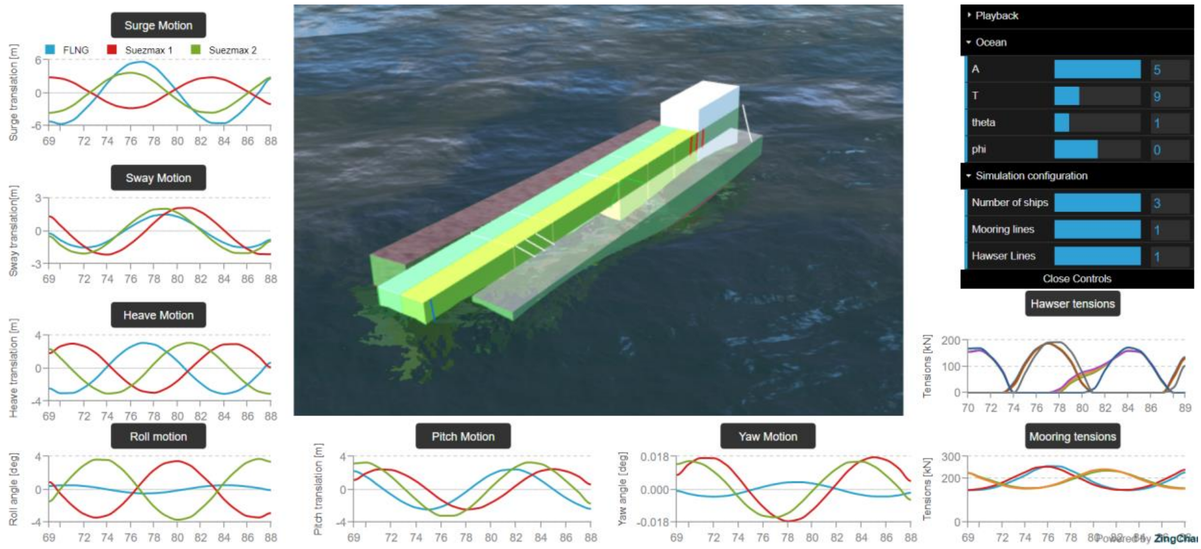


Figure 7. Progression of the case studies.

A side by side operation as the one displayed in Figure 6 is performed after adding all the possible entities studied in this thesis. Also, a graphical user interphase (GUI) is added in order to provide the user with several options such as adding or removing ships, and same wise with the mooring lines and hawsers. Finally, the interface allows the user to modify the wave analysis by providing different sea conditions that will respond by displaying the motions calculated in WAMIT and placed into the web browser through a JavaScript object.

It is also important in a web simulation to obtain real-time data as seen in Figure 6. This will help the user identify some magnitudes for the motions and also the tension on the mooring lines and hawsers. This can be very helpful for educational purposes or also for a company that needs to quickly check the result of WAMIT runs on a browser and see how the results behave.



A 5
T 9
theta 1
phi 0
Number of ships 3
Mooring lines 1
Hawser Lines 1

Figure 8. Case study with a friendly graphical user interface and real-time data displaying.

## 5. Tool to create maritime and subsea operations

Having worked progressively allows later to build maritime operations only by adding entities used in these cases along with their respective analysis. This work is important because from here, anyone can take these case studies and build up for any further application, as the vessels and their motion will be already given.

In order to create a new multibody operation, the steps shown in Figure 5.1 are required. First the last version of Vessel.js library needs to be downloaded (http://vessel.js.org), where one will find some of the examples from this study. On the examples it is possible to take the most basic case providing all functionalities which is based on WAMIT motions, where there is a simple barge on a regular sea, with no additional entities. The user can then go through the different cases reusing the piece of code needed for constructing a maritime operation as seen in Figure 7.

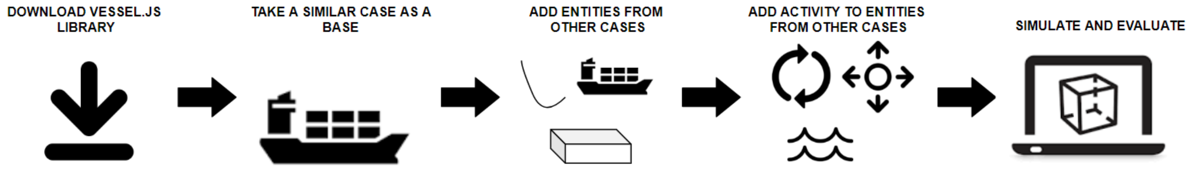

DOWNLOAD VESSEL.JS LIBRARY
TAKE A SIMILAR CASE AS A BASE
ADD ENTITIES FROM OTHER CASES
ADD ACTIVITY TO ENTITIES FROM OTHER CASES
SIMULATE AND EVALUATE

Figure 9. Process to create a multibody maritime operation

This process though is limited, as the motion of the ship depends on these additional entities added. In some cases, as seen in Figure 8, a new WAMIT run should be calculated before taking as accurate the motions on the ships, as someone taking a basic example with one ship and four mooring lines and after adding ten more lines limiting the motion of the ship, cannot expect the motion to be the same. This process is only valid for additional light lines that will not affect much the motion of the ship, or also for the addition of vessels far from the original vessel, as otherwise, the motion should also be affected and a new WAMIT run is required.

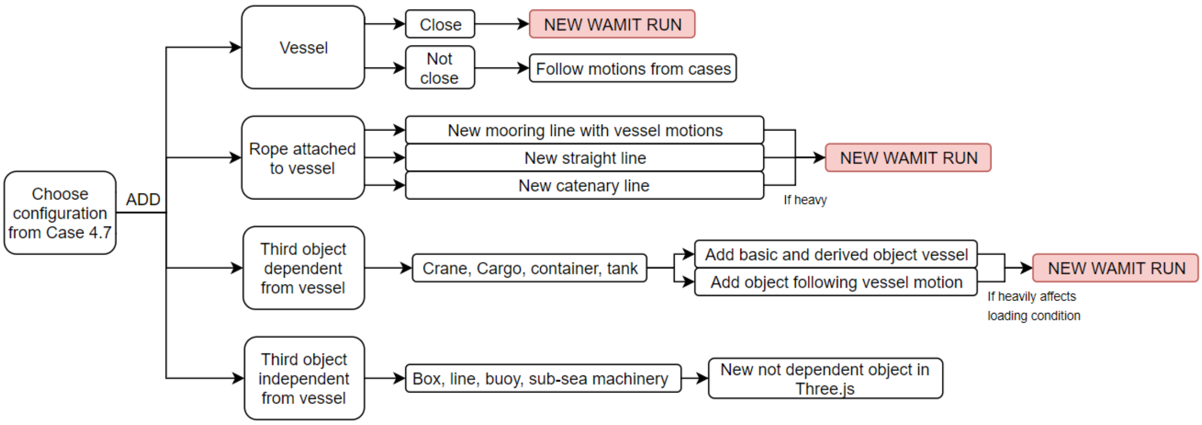

Choose configuration from Case 4.7
ADD
Vessel
Close
NEW WAMIT RUN
Not close
Follow motions from cases
Rope attached to vessel
New mooring line with vessel motions
New straight line
New catenary line
NEW WAMIT RUN
If heavy
Third object dependent from vessel
Crane, Cargo, container, tank
Add basic and derived object vessel
Add object following vessel motion
NEW WAMIT RUN
If heavily affects loading condition
Third object independent from vessel
Box, line, buoy, sub-sea machinery
New not dependent object in Three.js

Figure 10. Ways on how to add more entities and analysis for new operations formation.

## 6. Result evaluation

The major findings of this project have been to introduce a new way to define the motions in Vessel.js through 3D potential theory using a software developed by TPN and Petrobras which introduces WAMIT to calculate offshore systems configurations. Exploring multibody simulations on the Vessel.js library has also been a great contribution as this study has been able to go from one barge with some added lines to any kind of imaginable configuration using the interaction and taxonomy followed. This has allowed to overcome the limitations of single body motion displaying, as WAMIT takes cares of bodies that are in close proximity and acts accordingly.

From a motion calculation perspective, the results calculated are fairly within what was expected. However, used to the idea of the motion in one single vessel, having a side by side configuration derives into a variation of the motion a vessel is supposed to have, so a different interpretation of the RAOs has to be made. This gives an idea that the WAMIT simulations are performed correctly.

From a method perspective the results obtained prove the ability to achieve good results following the procedure proposed as well as achieving the objectives stated for the project, which are mainly to have been able to simulate correctly multibody and its interaction with several numbers of constrains comprised within different cases. The way to divide the cases in the taxonomy provides the user the facility to see how it really works and that a simulation is simple if it is properly understood. Also able to produce multibody cases with all kind of combinations and interactions, which is a massive step forward in comparison to the one ship entity and a mooring line system of the current Vessel.js library. All the other entities, analysis, processes and their combination on this study are entirely developed by the author with the aim to simulate multibody subsea and maritime operations with a different approach for the ship motions.

In a way, we can call the study a success because when a WAMIT run is made, there is no way to accurately know if the motions of a vessel are correct or not. By making this interface, we have been able to see the performance on real time of the vessels involved either visually or with real time plots. This has allowed to decide on many occasions that the motions were not displaying correctly and forced to rerun the solver in order to correct them to the final result.

## 7. Future work

Considering what limitations and challenges were faced on this study, some topics came out as possible candidates to be developed in the future by other researchers.

At this stage of the study one of the limitations is that despite the accuracy and good results that WAMIT provides, it is not suitable for real time simulation as it is now. An idea to work in the future would be to develop a way to introduce WAMIT runs in the web browser. An architecture could be developed where to read the input of the user through a GUI in the browser and link it directly to the TPN cluster, run it and bring back the results to the webpage, without the need of uploading any files.

In the case that this new way of calculating the motions could be performed, it would be a good idea to also go a step further and start accounting for what is not accounted in this study such as wave forces and mean drift forces could be considered in a more accurate and professional simulation environment, but for this work it is believed to be enough with just displaying the motion of the ship only with the RAO amplitude and phase.

Finally, a good addition to the library would be to start working with rigid body interaction. It would be a real response of the input waves. There should be a code specifying how to deal with a ship in waves, but then, no matter the condition, the ship would behave like a rigid body in a viscous fluid, just like in real life. Also, working with rigid bodies would allow to work with collisions, that nowadays are not handled, as the visualization does not make the bodies solid. There are nowadays some libraries simulating rigid bodies with collision using the Three.js environment and it could be a good addition with a different approach to the Vessel.js library.

## Acknowledgments

I would like to acknowledge the time, knowledge and funding provided by my head supervisor Henrique Gaspar, as well as for all the dedication that my co-supervisor Icaro Fonseca has invested in me and this work.

I would also like to thank the post-doc researcher Daniel P. Vieira and all the people in the University of São Paulo who took the time to teach me some of the methods used in this work and provided me with useful material in order to develop a good thesis.

Also, a huge gratitude to all the collaborators in the Vessel.js library for developing the tools that allowed me to reuse and build up my thesis, and also for the ones who spend some of their time helping me with my simulations as well as providing me with some insight and code.

## References

AASE, Ø. (2018), *Modelling and control of catenary mooring systems for vessels, oil rigs and barges,* Master’s thesis, NTNU, Trondheim.

DHAVALIKAR, S.; NEGI, A.; DOSHI, R. (2009), *Stability and motion analysis for barges*, National Conf. on Computer Aided Modelling and Simulation in Computational Mechanics, Mumbai.

ESEFELD, B.; SCHINDLER, T.; ULBRICH, H. (2013), *A coupling approach for the numerical integration of non-smooth multibody systems*. 4th Canadian Conference on Nonlinear Solid Mechanics, Montreal

FALTINSEN, O. M. (1990), *Sea loads on ships and offshore structures.* Cambridge University Press, Cambridge, UK

FILHO, A.N.Q; TANNURI, E.A. (2009), *DP offloading operation: a numerical evaluation of wave shielding effect.* 8th IFAC International Conference on Manoeuvring and control of marine craft. Guarujá, Brazil.

FONSECA, Í.A.; GASPAR, H.M. (2015), *An object-oriented approach for virtual prototyping in* *conceptual ship design*, 29th European Conference on Modelling and Simulation (ECMS) Conf., Varna

FONSECA, Í.A; OLIVEIRA, F.F, GASPAR, H.M. (2019), *Virtual prototyping and simulations of multibody marine operations using web-based technologies.* ASME 2019 38th International and Conference on Ocean, Offshore and Arctic Engineering, Glasgow, Scotland.

FONSECA, Í.A. (2018), *An Open and Collaborative Object-Oriented Taxonomy for Simulation of Marine Operations*, Master's Thesis, NTNU, Trondheim

FOSSEN, T.I. (2011), *Handbook of Marine Craft Hydrodynamics and Motion Control*, Trondheim, Wiley.

FOSSEN, T.I.; FJELLSTAD, O. (1995), *Nonlinear modelling of marine vehicles in 6 degrees of freedom*, Journal of Mathematical Modelling of Systems, Vol. 1, No. 1, 1995, Trondheim

GASPAR, H.M. (2017), *JavaScript Applied to Maritime Design and Engineering*, 16th Conf. Computer and IT Applications in the Maritime Industries (COMPIT), Cardiff

GASPAR, H.M. (2018), *Vessel.js: an open and collaborative ship design object-oriented library,* Int. Marine Design Conf. (IMDC), Helsinki

GEMILANG, G.M. (2015). *Feasibility study of selected riser concept in deep water and harsh environment.* Master thesis, University of Stavanger, Stavanger.

HE, B.; WANG, Y.; SONG, W.; TANG, W. (2015), *Design resource management for virtual prototyping in product collaborative design*, Proc. Institution of Mechanical Engineers, Part B: J. Eng. Manufacture 229

HU, Z; WANG, S; CHEN, G; CHAI, S; JIN, Y- (2016), *The effects of LNG-tank sloshing on the global motions of FLNG system.* International Journal of Naval Architecture and Ocean Engineering 9 (2017) 114-125

IRVINE, H.M. (1992). *Cable Structures. Volume 5.* Dover Publications, New York, NY.

JENSEN, J.J.; MANSOUR, A.E; OLSEN, A.S. (2004), *Estimation of ship motions using closed-form expressions.* Ocean Engineering 31(1):61-85

KIM, Y.; KIM, K.; KIM, J.; KIM, T.; SEO, M.; KIM, Y. (2011). *Time-domain analysis of nonlinear motion responses and structural loads on ships and offshore structures: development of WISH programs,* Inter J Nav Archit Oc Engng (2011), Seoul.

LEE, C; Newman, J. (2006). *WAMIT user Manual*. Chestnut Hill, MA, USA.

LEVANDER, K. (2012), *System based ship design*, NTNU Marine Technology, Trondheim

LEWIS, E.V. (1989), *Principles of Naval Architecture*, Jersey City, The society of naval architects and marine engineers.

LOC, M.B. (2014), *Time optimal trajectory design for unmanned underwater vehicle.* Ocean engineering 89 (2014) 69-81

MONTEIRO, T.G (2016), A knowledge-based approach for an open object-oriented library in ship design. Master's Thesis, NTNU, Ålesund.

NATVIG, B.J.; PENDERED, J.W. (1980), *Motion response of floating structures on regular waves*, Aker engineering A/S, Oslo

NEWMAN, J.N. (1935), *Marine hydrodynamics.* The MIT press. Massachusetts.

OLIVEIRA, F.F. (2019). *Implementation of Open Source Code for 6 Degrees of Freedom Simulations in Maritime Applications.* Research Report, Ship Design and Operation Lab, NTNU.

OLIVIER, N. (2013), *Dynamic analysis and design of mooring lines*, Master's Thesis, NTNU, Trondheim

OMBERG, H.; LARSEN, K. (1998), *Coupled analysis of a floater motion and mooring dynamics for a turret-moored ship*. Statoil Research centre and MARINTEK, Trondheim.

SANFILIPO, F.; HATLEDAL, L.I.; STYVE, A.; PETTERSEN, K.Y.; ZHANG, H. (2015). *Integrated Flexible Maritime Crane Architecture for the Offshore Simulation Centre (OSC).* July 2015. IEEE Journal Oceanic Engineering. DOI: 10.1109/JOE.2015.2439891

WAGNER, G. (2016), *Introduction to simulation using JavaScript*, 2016 Winter simulation conference, Cottbus.

WINGEREI, K. (2008), *Dynamics and damping in mooring lines*, Master's Thesis, University of Oslo, Oslo

WITTENBURG, G. (1977), *Dynamics of multibody systems*, Teubner, Springer.