Numerical simulation of installation operations for offshore wind farms

Pierre-Yves Wuillaume

UNIVERSITE BRETAGNE SCIENCES LOIRE POUR L'INGENIEUR

CENTRALE NANTES

THESE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 602
Sciences pour l'Ingénieur
Spécialité : Mécanique des Milieux Fluides

Par
Pierre-Yves WUILLAUME

Simulation numérique des opérations d'installation pour les fermes d'éoliennes offshore

Thèse présentée et soutenue à Nantes, le 15 janvier 2019.
Unité de recherche : Laboratoire d'Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA), CNRS UMR 6598

Rapporteurs avant soutenance :

René H. M. Huijsmans	Professeur, TU Delft (Pays-Bas)
Julien Salomon	Directeur de recherche, INRIA (France)

Composition du Jury :

Président :	René H. M. Huijsmans	Professeur, TU Delft (Pays-Bas)
Examinateurs :	Julien Salomon	Directeur de recherche, INRIA (France)
	Erin Bachynski	Associate Professor, NTNU (Norvège)
	Boyin Ding	Research Fellow, University of Adelaide (Australie)
	Aurélien Babarit	Ingénieur de recherche HDR, Ecole Centrale de Nantes (France)
Dir. de thèse :	Pierre Ferrant	Professeur, Ecole Centrale de Nantes (France)

Invité

Mattias Lynch	Technical Lead Ocean Engineering, INNOSEA (France)

À ma mère, pour ce qu'elle a fait pour ma grand-mère et pour moi.

To my mother, for what she did for my grandmother and me.

Remerciements - Acknowledgement

En premier lieu, je souhaite remercier Messieurs René Huijsmann et Julien Salomon pour avoir accepté d'être les rapporteurs de cette thèse de doctorat. Leurs commentaires sur mon travail m'ont été précieux. Je remercie également Madame Erin Bachynski et Monsieur Boyin Ding pour avoir fait partie du Jury de cette thèse.

J'ai bénéficié tout au long de ce doctorat d'un excellent encadrement au sein du *LHEEA*. Je suis reconnaissant à Pierre Ferrant, directeur de thèse, ainsi qu'à Aurélien Babarit et François Rongère, tous deux encadrants, d'y avoir fortement contribué. Je tiens à remercier tout particulièrement Pierre Ferrant et Aurélien Babarit pour m'avoir permis de terminer ma thèse dans de bonnes conditions. Je sais gré à Aurélien Babarit d'avoir toujours relu mon travail avec précision et rapidité, de m'avoir donné l'opportunité de faire une visite dans un laboratoire japonais et d'avoir eu l'idée des essais en bassin. Je remercie François Rongère pour ses conseils, toujours enrichissants, notamment en mécanique multicorps et en informatique.

Ce doctorat a été financé via une convention *CIFRE* avec l'entreprise *INNOSEA*. Aussi, je remercie Hakim Mouslim et Maxime Philippe de m'avoir choisi pour réaliser cette thèse. Maxime Philippe puis Mattias Lynch ont encadré ce travail chez *INNOSEA* et je leur suis reconnaissant pour leurs conseils et leurs points de vue industriels ainsi que pour m'avoir laissé toute la liberté nécessaire pour mener à bien ce projet.

Je souhaite également remercier toutes les personnes extérieures à l'encadrement qui m'ont apporté leur aide à divers moments au cours de ce doctorat. Je pense en particulier à Lucas Letournel et à Camille Chauvigné pour leur expertise sur le code *WS_CN* et pour leurs réponses à mes (nombreuses) questions. Je suis très reconnaissant à Félicien Bonnefoy pour avoir encadré et fortement contribué au projet des essais en bassin. Je remercie également Messieurs Alexandre Simos et Rafael Watai pour leur aide précieuse pendant la validation de l'extension de *WS_CN* aux simulations multicorps. J'adresse mes remerciements à Monsieur Motohiko Murai et aux étudiants de l'université nationale de Yokohama que j'ai pu rencontrer pour leur accueil chaleureux, leur aide pendant mes premiers jours et pour m'avoir fait découvrir leur beau pays, le Japon. J'exprime ma gratitude à Adrien Grellier, Vincent Degat ainsi qu'à Samuel Girardin pour avoir résolu mes problèmes informatiques en tout genre. Enfin, je remercie Anne Lagatu et Sonia Lambert pour avoir simplifié toutes les démarches administratives rencontrées pendant ces trois ans et demi.

Un grand merci aux membres du bureau 2016/2017 de l'*ACDC* avec qui j'ai vécu une expérience très enrichissante : Siyimane Moyaine, Julie Hemmer, Hélène Clémot, Laetitia Pernod et Reda Jaafri. Je remercie également Monsieur Jean-Yves Hascoët et Madame Annie Barreau de la Direction de la Recherche pour leur aide et leurs conseils pendant ce mandat.

J'adresse mes remerciements à Charles Spraul et à Chunmei Xie pour la très bonne ambiance qui régna dans notre bureau. Merci à l'ensemble des doctorants que j'ai pu

rencontrer au *LHEEA* ainsi qu'à Violette et à Quentin. Pour finir, je remercie mes parents pour m'avoir soutenu pendant ces dix années d'études qui s'achèvent par ces mots.

Résumé

L'éolien offshore est l'énergie marine la plus avancée et utilisée dans le monde. Afin d'accroître l'énergie extraite du vent, les dimensions des éoliennes deviennent plus importantes et les parcs éoliens sont installés de plus en plus loin des côtes, où les mers sont agitées et les vents forts. De fait, les opérations marines sont plus complexes et plus chères et les fenêtres météo sont écourtées et se raréfient.

Dans le cadre de cette thèse, un logiciel de simulation numérique des opérations marines est développé, en particulier pour des applications de descentes et de remontées de colis lourds. L'Algorithme aux Corps Rigides Composites, implémenté dans le logiciel *InWave*, est utilisé pour modéliser le système multicorps. Un modèle de câble et de treuil est développé, suivant la théorie multicorps utilisée, et comparé à la théorie câble classique dite «*lumped mass*». Les efforts hydrodynamiques ainsi que les interactions hydrodynamiques sont modélisés par une théorie d'écoulement potentiel instationnaire satisfaisant l'hypothèse de faible perturbation, dite «*weak-scatterer*». L'approche «*weak-scatterer*» du logiciel *WS_CN* est étendue aux simulations multi-flotteurs et validée par comparaison avec des données expérimentales.

InWave et *WS_CN* sont couplés afin de résoudre l'interaction houle-structure pour des systèmes multicorps articulés en mer. Un couplage fort est adopté pour sa robustesse. L'équation de couplage est établie et validée via des comparaisons avec *WS_CN*. Le logiciel ainsi créé se nomme *InWaveS_CN* et utilise un code d'intégration en Python.

Une nouvelle stratégie de maillage, basée sur un algorithme de découpe de maillages et une méthode par avance de front, est développée dans *WS_CN*.

Enfin, des essais en bassin d'une opération de redressement ont été menés à l'*École Centrale de Nantes*. La comparaison entre les simulations numériques et les données expérimentales offre une première et prometteuse validation d'*InWaveS_CN*.

Mots-clés : Opérations marines, Dynamique multicorps, Dynamique de câbles, Écoulement potentiel, *Weak-scatterer*, Interaction fluide-structure, Découpe de maillages.

Abstract

Offshore wind represents the most advanced and used marine energy in the world. To increase the wind power extraction, turbines grow in size and wind farms are installed further offshore in presence of rough seas and strong winds. Marine operations become more challenging and expensive, weather windows are shorter and less frequent.

This PhD work focuses on the development of a numerical tool to simulate marine operations with consistency, in particular lowering and lifting operations. The Composite-Rigid-Body Algorithm, implemented in the numerical tool *InWave*, is used to model multibody systems. A cable model and a winch model are developed following this multibody approach and compared to the classical low-order lumped mass theory. Hydrodynamic loads and hydrodynamic interactions are simulated using an unsteady potential flow theory based on the weak-scatterer hypothesis, implemented in the numerical tool *WS_ CN*. This approach is extended to multibody simulations and validated with comparisons to experimental data.

InWave and *WS_ CN* are coupled to solve wave-structure interaction for articulated multibody systems with large relative motions in waves. A tight coupling is selected for its robustness. The coupling equation is derived and validated from comparisons with *WS_ CN*. This leads to the creation of a new numerical tool, *InWaveS_ CN*, using Python as glue code language.

A new mesh strategy, based on the coupling between a panel cutting method and an advance front method, is developed in *WS_ CN*.

Experiments of an upending operation were conducted at *Ecole Centrale de Nantes*. The comparison between the numerical simulations and the experimental data leads to a first and promising validation of *InWaveS_ CN*.

Keywords: Marine operations, Multibody dynamics, Cable dynamics, Potential flow theory, Weak-scatterer, Fluid-structure interaction, Panel cutting method.

Contents

List of Figures

List of Tables

Introduction

Contents

1 Marine renewable energies

Marine renewable energies (*MRE*) are the renewable energies that can be extracted from the ocean. Their main use is electricity generation. They are expected to contribute to the mitigation of climate change. Marine renewable energies encompass various kinds of energies:

- Wave energy;
- Tidal energy;
- Offshore wind energy;
- Ocean thermal energy;
- Osmotic energy.

In 2017, 42.6 GWh were generated from these *MRE* devices. This represents 0.17 % of the global electricity production[1]. 19 GW capacity have been installed so far[2]. Offshore wind contributed to 41.6 GWh of the electricity generation and 18.7 MW of the installed capacity, namely 98 % of the total. Thus, offshore wind is the most advanced and used marine energy. Between 2016 and 2017, the offshore wind installed capacity grew by 30 %. Industrial offshore wind farms are currently operational while other marine energies are still in their earliest stage.

Two types of offshore wind turbines exist (Figure 1):

- Bottom-fixed turbines, mainly used in shallow waters (up to 40 m) for near-shore sites. The foundations include three main designs:
  - Monopiles;
  - Jackets;
  - Gravity-based.
- Floating turbines, for wind farms in deep water. The advantages of such a technology are the more powerful and constant winds found further offshore, the increase in available sites and the reduction of visual impact. Turbines may be horizontal-axis or vertical-axis, this latter enabling to lower the center of gravity of the structure and becoming insensitive to wind direction. Only few floating wind turbines have been installed so far. Three main different floaters are considered:
  - Semi-submersibles;
  - Spars;
  - Tension-leg platforms.

To increase the wind power extraction, turbines get taller and blades become longer. Furthermore, wind farms are further offshore and so, subject to rough seas and strong winds. Consequently, offshore installations are more challenging and weather windows for marine operations are shorter and less frequent. Marine operations costs (installation, operations and maintenance) are significant in offshore wind and have a large impact on the cost of electricity. This latter is split into two expenditures:

[1] https://yearbook.enerdata.net
[2] http://www.irena.org

- Capital expenditure or *CAPEX*: purchase or improvement of assets (turbine, foundation, sub-station, cables, installation, transportation to the site, etc.);
- Operational expenditure or *OPEX*: ongoing costs to operate a product (land rental, insurance, taxes, operation, maintenance, etc.).

According to Crabtree et al. [1], *CAPEX* accounts for 70 % of the cost of electricity, while *OPEX* represents 30 %. The reduction of both *CAPEX* and *OPEX* is necessary to reach the cost of the onshore wind industry and make the marine renewable energies competitive with other energy sources (coal, oil, gas, etc.). Including installation, the marine operations for offshore wind account for 30 % of the cost of electricity. The reduction of these costs goes through the improvement and the optimization of the marine operations, such as installations, maintenance and decommissioning operations [2]. The reduction of their risks, their duration and the increase of the available weather windows are also necessary. That is why, the study of marine operations (theoretically, experimentally and numerically) is a very important topic for the development of marine renewable energies.

(a) Bottom-fixed[3] **(b)** Floating[4]

Figure 1 – Offshore wind turbines

2 Marine operations

Marine operations are all the activities conducted at sea to transport and install offshore structures, from the organization and the planning to the realisation of the projects. These operations needs to be conducted within the operation limits of sea states (or weather windows) [3]. These limits are defined to ensure the safety of both personnel and assets. Different types of marine operations exist:

- Towing operations: transport of subsea or floating structures to site using a barge or a towing vessel (Figure 2);
- Up-ending operations: controlled rotation of a structure from its horizontal position to its upright position by means of a ballasting system and/or a floating

[3] https://stoprust.com
[4] https://www.ideol-offshore.com

crane and setting it down on the sea bottom, its foundation or leaving at its stable-floating position (Figure 3);

- Offloading operations: two vessels (for instance a *FPSO*[5] and a shuttle tanker) are connected to each other (to transfer the oil from the *FPSO* to the shuttle tanker) in tandem (in front of each other) (Figure 4) or side-by-side;
- Floatover installation: a vessel transports the topsides (upper parts of offshore structures) to site, then the vessel places the topsides over the fixed or floating substructures and lowers the topsides while sustaining the vessel position (Figure 5);
- Loadout operations: transferring the cargo from the quay onto a barge using cranes, skid rails, trailer, etc. (Figure 6);
- Laying operations: deployment of cables, umbilicals or power cables from a pipe-laying vessel equipped with a chute (Figure 7);
- Piling operations: offshore monopiles pushed into the seabed (Figure 8);
- Lowering and lifting operations;
- Etc.

The lowering and lifting operations are the goal of this PhD work, consequently they are presented in details in the next section.

[5] Floating, Production, Storage and Offloading

Figure 2 – Towing of the floating wind turbine *WindFloat*[6]

Figure 3 – Upending of a monopile[7]

[6] http://www.rechargenews.com
[7] https://ocean-energyresources.com

Figure 4 – Tandem offloading operation between a *FPSO* and a shuttle tanker[8]

Figure 5 – Floatover installation of a topside[9]

[8] http://www.colindelarue.com
[9] http://www.offshoretechllc.com

Figure 6 – Loadout of a tension-leg platform using skid rails[10]

Figure 7 – Power cable laying[11]. The buoys indicate the cable position.

[10] https://www.ynfpublishers.com
[11] http://www.emr-paysdelaloire.fr

Figure 8 – Piling operation of a monopile[12]

3 Lowering and lifting operations

Lowering and lifting operations are used to lower or lift an object into the sea. The lifting equipment is made of:

- A vessel;
- A crane on top of the vessel, with possibly several booms;
- A hoisting cable connecting the crane to the hook;
- A winch to wind or unwind the hoisting cable;
- Slings (loops of material around the payload and connected to the hook) or riggings (ropes or chains connecting the payload to the hook);
- The payload to be lowered or lifted.

Vessels may be jack-up vessels (Figure 9) or floating crane vessels (Figure 10) (monohull, semi-submersible or catamaran). Jack-up vessels are self-elevating platforms providing a stable position for installation operations.

Payloads may be:

- A subsea template [4];

[12] https://www.delta.tudelft.nl

- A subsea manifold [5];
- A ship;
- A *ROV* (Remotely Operated Vehicle);
- A wind turbine component (blade, nacelle, foundation) [6];
- A pre-installed wind turbine [7];
- A tidal turbine (Figure 10);
- A wave energy converter;
- An anchor;
- Etc.

An example of lifting operation is displayed in Figure 10 with a crane-mounted ship, a hoisting cable, riggings and a payload which is a tidal turbine.

A lowering operation is split into four steps [5, 8]:

- The payload is lifted off from the deck of the vessel and manoeuvred in the air before being lowered. Snap loads (high tension in the hoisting cable) or pendulum motion involving risks of collision may occur.
- The hoisting cable is unwound and the cargo is lowered through the wave zone. Hydrodynamic loads appear, in particular slamming loads (due to the hydrodynamic impact) which can cause snap loads. In case of lifting operation, slamming loads become water exit loads.
- The payload is lowered deeply into the sea. The increase of the cable length involves a modification of the natural frequencies of the system. Vertical oscillations of the payload due to the vessel motion induced by the waves may be significant. Current (with possibly both time-dependent velocity and direction) needs to be taken into account and leads to a horizontal offset. The drift motion of the vessel may occur if a dynamic positioning system is not used.
- Once the payload is close enough to the seabed, the final step is the landing operation. In case of lifting operation, it is a retrieval operation. The impact of the object on the seabed must not lead to any damage. The accuracy of the payload position must be guaranteed.

Thus, before starting any lowering or lifting operation, every lifting equipment has to be checked to avoid overloads. This includes the crane capacity, the rigging design, the structural strength of the cargo or the seakeeping of the vessel. This analysis can be efficiently support by models. Different branches of the physics are necessary:

- Multibody dynamics: the floating crane forms an articulated body system (vessel, crane, cable, payload) in mechanical interactions (the motion of the vessel involves a motion of the payload through the cables, and conversely);
- Cable dynamics;

- Hydrodynamics: several bodies are subject to hydrodynamic loads (vessel, payload), hydrodynamic interactions and slamming loads;
- Fluid-structure interaction: waves induce the motion of the vessel and the payload, in return the motion of those bodies modifies the wave field;
- Control systems: for reducing the motion of the payload by controlling its position and the tension in the hoisting cable.

A sketch of these several physical fields is presented in Figure 11.

Figure 9 – Jack-up vessel[13]

[13] http://www.heavyliftspecialist.com

Figure 10 – Lifting operation of the *Sabella D10* tidal turbine[14]

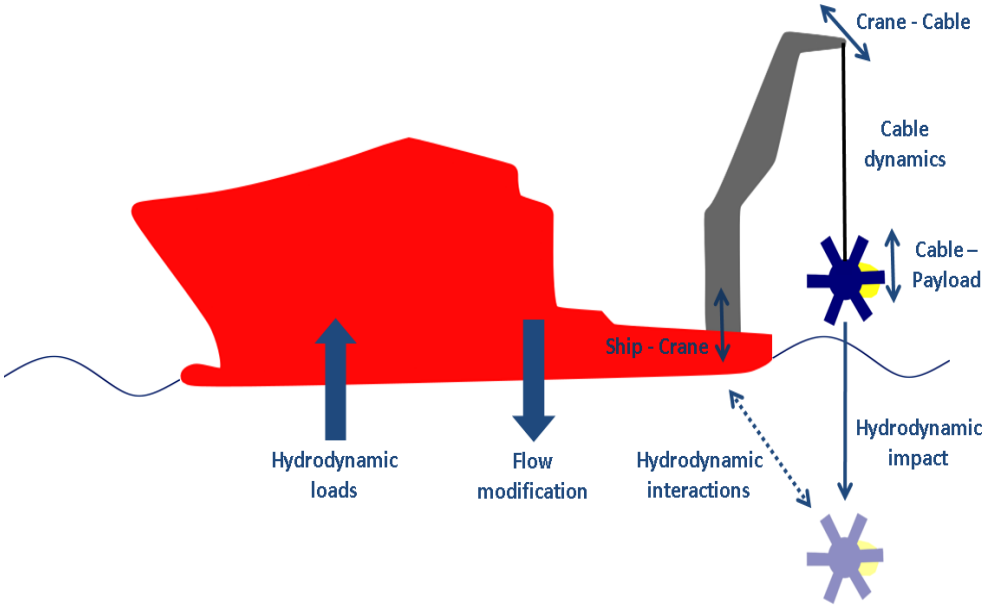


Figure 11 – Modeling of a lowering operation. The hoisting cable is in black. The thick arrows represent the fluid-structure coupling and the double-headed thin arrows denote the mechanical interactions between bodies.

[14] http://www.sabella-d10.bzh

Note

Other methods exist to lower a payload without using a floating crane. For example, the pendulous installation method requires two vessels: a transportation vessel and an installation vessel, separated one from the other by a distance roughly equal to the water depth. The payload is hung to the transportation vessel by a cable which the other extremity is fixed to the installation vessel. Then the payload is detached from the transportation vessel and lowers following a pendulous motion [9].

Another possibility is based on the pencil buoy method [10]. The payload is immersed inshore, then towed to site suspended from a pencil shaped buoy and finally lowered offshore using a winch.

4 State of the art of the numerical simulation of lowering or lifting operations

To schedule the operations, establish the operational conditions (weather windows and uptime levels), design the equipments or predict the body motions to reduce risks, collisions or any injury to the workers, numerical simulation has become an essential tool.

The numerical simulation of lowering or lifting operations at sea has been extensively studied in the literature. These works can be gathered in three domains:

- Structural analysis;
- Hydrodynamic analysis;
- Motion control analysis.

4.1 Structural analysis

Offshore lifting or lowering operations require to model an articulated multibody system. The first structural studies were based on simplified approaches. For instance, by assumption, the crane tip follows a prescribed motion. This mechanical model uncouples the dynamics of the payload from the dynamics of the floating crane. It assumes that the payload does not influence the crane and the supporting vessel. This hypothesis is justified in case of small ratio of the payload mass over the vessel mass [11]. Elling and McClinton [12] and Chin et al. [13] computed the dynamics of a single concentrated mass, such as a pendulum, linked to a crane tip by a rigid cable with a prescribed harmonic displacement of the boom. This approach allows studying the motion of the lifted object subject to a parametric excitation. For instance, Elling and McClinton [12] highlighted a resonance phenomenon when the frequency of the harmonic imposed crane tip motion matches the pendulum natural frequency. Nevertheless mechanical coupling effects can be important [14].

It is also possible to linearize the motion equation at the system's equilibrium position. The purpose of a linearized model is to perform a frequency-domain analysis and compute the natural frequencies of the system. Comparisons between nonlinear and linearized models along with experimental measurements showed a good agreement for small amplitude motions but not near the resonance [15, 16].

Another simplification is to consider a two-dimensional problem. Ellermann et al. [17, 18] simulated a floating crane linked to a suspended lumped mass by assuming a motion in a vertical plane. Then, this model was simplified in order to study the

influence of different parameters using a multiple scales method and a path-following technique to perform a bifurcation analysis.

Uncoupled or linearized or two-dimensional models suffer from a lack of generality. Payload dynamics may influence the motion of the floating crane in case of heavy lifting operations, nonlinearities are present in the physics (mechanics, mooring, hoisting cable dynamics, etc.) and a two-dimensional model restricts the simulation to planar motions. Thus, more complex coupled models were necessary. Lots of studies (Schellin et al. [16], Van Den Boom et al. [19], Witz [20], Malenica et al. [21], etc.) were devoted to the simulation of a floating crane with a payload hanging in the air. The problem is in three dimensions and the cable is either rigid or elastic. Only few studies consider the payload in the water. The dynamics of a submerged payload with a constrained motion using a rigid cable was studied by Bai et al. [22]. Hannan and Bai [23] performed a dynamical analysis of the same constrained payload near a floating barge. The barge is motionless, which simplifies the mechanical problem. They used phase trajectories, Poincaré maps and bifurcation diagrams to analyse the payload motion in waves. Bashir et al. [24] and Nam et al. [5] did a numerical study with both a floating crane and a payload in water.

Since Cha et al. [25], mechanical models have started being based on fully coupled multibody dynamics. These authors used constraint equations to include the internal degrees of freedom of their multibody system. Using this approach, they could simulate two rigid bodies with six degrees of freedom each and linked by an elastic cable (a floating crane with a lifted heavy cargo). This model was also used by Ku and Roh [7] in order to simulate the dynamic response of an offshore wind turbine suspended in the air by a floating crane.

So far, motion equations were derived manually. In case of a change in the multibody system, they had to be derived again. Hence, it became convenient to use multibody dynamic algorithms to write automatically the motion equation for most or any multibody systems. Ku and Ha [26] used multibody dynamic algorithms based on the Newton-Euler equation to model a heavy load lifting operation with multi-cranes. These algorithms are based on a recursive formulation and can be applied to all multibody systems. Ham et al. [27] preferred the Euler-Lagrange equation. They formulated it with constraint equations. An application was done with a dynamic analysis of a floating crane with two booms and a heavy load in the air.

Hitherto, only rigid bodies were considered. According to Ren et al. [28], rigid crane boom can be used in case of short crane booms and small payload-to-ship ratios, otherwise the crane boom flexibility should be taken into account. Some studies considers a flexible crane boom. Ren et al. [28] simulated a flexible boom floating crane with a point mass payload linked by a rigid rope by assuming a two-dimensional motion. The authors showed the flexibility involves high-frequency vibrations in the surge vessel motion along with a reduction of the payload-swing angle. Park et al. [29] modeled a flexible boom floating crane in three dimensions using a finite-element method and two six-degree-of-freedom rigid bodies for the crane and the payload. Surge and pitch motions were increased in case of the flexibility. The high-frequency vibrations found in [28] are not reported.

Thus, the mechanical model of the lifting and lowering operations becomes more complex with time, from uncoupled bi-dimensional approaches to fully coupled nonlinear dynamics handling mechanical interactions. The same applies for cable models, from a

single rigid cable to possibly several elastic cables (slings).

4.2 Hydrodynamic and aerodynamic analysis

The main external loads acting on a floating crane and the payload are:

- Weight;
- Wind loads;
- Mooring loads;
- Hydrodynamic loads.

Wind loads are not always considered in the numerical simulations of lowering or lifting operations. They may induce large motions in case of blade installations [3] or light payloads [8]. When they are computed, the wind loads are mainly expressed following a drag equation [19, 26, 30]:

$$\mathbf{F}^{Wind} = \frac{1}{2} C_D^{Wind} \rho_{air} A \mathbf{V}_{Wind}^2 \tag{1}$$

with:

- $\mathbf{F}^{Wind}$ the wind loads;
- C_D^{Wind} the dimensionless drag coefficient;
- ρ_{air} the air density;
- A the projected area of the bodies which contact with wind;
- $\mathbf{V}_{Wind}$ the wind velocity.

More complex models using a time-varying wind velocity [31] and taking into account wind turbulence [6] are also developed. No wind loads will be considered in this PhD work.

Regarding mooring loads, different theories are used: quasi-static [31], linear [20] or nonlinear [16]. These models are presented in chapter II.

Hydrodynamic loads are split into several components:

- Hydrostatic loads;
- Hydrodynamic loads which include loads due to:
  - Wave-structure interaction effects;
  - Current;
  - Viscous effects.

When the current and viscous loads are computed, they follow a drag equation [16, 30, 31]. Hydrostatic loads may be either linear or nonlinear. A linear approach is used when the hydrostatic loads are computed based on the mean wetted surface of the bodies. They are expressed with a linear hydrostatic matrix [5, 16, 19]:

$$\mathbf{F}^{Hydrostatic} = \mathbf{F}^{B} - \mathbf{K}^{Hydrostatic}\mathbf{X} \tag{2}$$

where $\mathbf{F}^{Hydrostatic}$, $\mathbf{F}^{B}$, $\mathbf{K}^{Hydrostatic}$ and $\mathbf{X}$ represent the hydrostatic loads, the buoyancy force, the hydrostatic stiffness matrix and the position vector of the body, respectively.

When hydrostatic loads are nonlinear, the pressure is integrated over the exact wetted body surface [7, 25]. Lee et al. [32] compared the use of linear and nonlinear hydrostatics in case of a lowering operation and showed that nonlinearities in the hydrostatic loads increase the dynamic motion.

Concerning the wave-structure interaction, most of the studies are based on the linear frequency-domain potential flow theory [7, 16, 19, 21, 25]. A linear time-domain potential flow theory is used by Ku and Ha [26]. Linear theory (both in frequency and time domains) assumes a small amplitude motion of the bodies and a small steepness of the waves. It is valid when the payload is in the air as it is the case in the majority of the works. When the payload is in the water and is lowered or lifted, the linear potential flow theory is not applicable with consistency. The presence of two bodies close to each other leads to hydrodynamic interactions and the lowering or the lifting of the payload involves a large relative amplitude motion. That is why, Hannan [11] used a fully nonlinear potential flow theory to simulate such marine operations in his PhD work. In this case, the assumptions of the linear theory do not have to be satisfied. One part of his work focused on the motion of a fully immersed cylinder in waves and subject to a constrained pendulum motion with a lowering velocity [22]. He also studied the hydrodynamic interactions between a fixed floating barge and the same cylindrical payload [23]. The presence of the barge close to the payload modified the motion of this latter due to the hydrodynamic interactions.

More details about the different hydrodynamic approaches are given in chapter III.

Thus, as for the mechanical models, the wave-structure interaction models become more complex with time. Most of the studies are based on the linear frequency-domain potential flow theory which involves two strong assumptions on the body motion and the wave steepness. A whole lowering operation cannot be consistently simulated with this theory. The use of an unsteady potential flow based solver to simulate the floating crane and the payload in waves gains in importance.

4.3 Motion control analysis

In order to attenuate the heave motion of the lifted object and reduce the tension variation in the hoisting cable, heave compensation systems are used. Their final goal is to decouple the payload motion from the wave-induced motion of the vessel, especially during a deepwater lowering or lifting operation. Three families of heave compensation exist [5, 33]:

- Passive heave compensation: a spring-damper system is placed along the hoisting line to shift the heave natural frequency of the lifting equipment. These systems do not require any energy to operate.
- Active heave compensation: sensors and controlled actuators (winches, pistons) are used to oppose the heave motion. These systems require energy to operate and are more complex, expensive and efficient than passive heave compensators.
- Hybrid active-passive heave compensation: combination of features of passive and active heave compensation systems.

Other motion compensation systems are possible such as an active tugger line force control for single blade installation [34]. Motion compensation systems will not be considered in this PhD work.

4.4 Norms

Det Norske Veritas (*DNV*) provides guidelines about the modelling and the analysis of marine operations [8]. Three models are proposed for estimating the loads acting on a lowered payload through the wave zone:

- A simplified method;
- A regular design wave approach;
- A time-domain approach.

The simplified method assumes:

- The horizontal extent of the payload is small compared to the wave length;
- The vertical motion of the payload follows the crane tip motion;
- The vertical acting loads are dominant.

This method gives simple conservative estimates of the loads acting on the lifted object. Only characteristic values are included in the computation. For example, the characteristic wave amplitude η_a is:

$$\eta_a = 0.9 H_S \tag{3}$$

with H_S the significant wave height.

The characteristic hydrodynamic force may be evaluated by:

$$F_{Hyd} = \sqrt{(F_D + F_{Slam})^2 + (F_M - F_\rho)^2} \tag{4}$$

with

- F_D the characteristic hydrodynamic drag force;
- F_{Slam} the characteristic slamming impact force;
- F_M the characteristic hydrodynamic mass force;
- F_ρ the characteristic varying buoyancy force.

In the regular design wave approach, a single regular wave is applied to the payload which is subject to a prescribed harmonic motion. The design wave amplitude η is:

$$\eta = \begin{cases} 0.9H_S & \text{for operations performed within 30 min} \\ H_S & \text{for operations performed in more than 30 min} \end{cases} \quad (5)$$

The time-domain approach of an object lowered into the wave zone is based on the motion equation for a vertical motion only. No horizontal displacement is considered. The payload is subject to the following forces: hoisting cable force, weight, buoyancy force, steady force due to current, inertia force, wave damping force, drag force, wave excitation force and slamming or water exit force.

Regarding the simulation of both the floating crane and the payload, *DNV* guidelines split the problem into two categories:

- Light lifts, where the mass of the payload is less than 2 % of the floating crane displacement;
- Heavy lifts, where the mass of the payload is more than 2 % of the floating crane displacement.

In the first case, it is assumed that the vessel motion is not affected by the payload motion and the mechanical model is uncoupled. In the second case, the coupled dynamics is required.

The coupled approach of *DNV* is a nine degree-of-freedom model (six for the crane and three for the payload). The two bodies are linked by a spring. The hydrodynamic loads are evaluated for the crane and the payload using a hydrodynamic database provided by a linear potential flow based solver. Hydrostatic loads are linear too. The mechanical coupling is ensured by a coupling stiffness matrix. Thus, this model takes into account both hydrodynamic and mechanical interactions.

In case of a deepwater lowering, *DNV* norms provide a more accurate approach which models the control of the vertical motion of the lifted object and the horizontal offset due to the current. Additional recommendations are given in case of landing on seabed or retrieval.

DNV norms have been compared to commercial numerical tools (cf. subsection 4.5). Jacobsen and Leira [4] simulated a submerged towing operation. The simplified method and the numerical results of *SIMO* were compared to experiments. It was shown that *DNV* guidelines did not provide an accurate response estimation and should only be used for feasibility studies at an early design stage. Kimiaei et al. [35] used *DNV* guidelines released in 1996 and 2008 and compared them to *OrcaFlex* on the installation of a subsea frame through the wave zone. They concluded that the two versions of the norms overestimates the numerical results of *OrcaFlex*, the earliest guidelines being less conservative.

4.5 Software packages

Several commercial time-domain numerical tools are available for performing simulations of marine operations:

- *SIMO*[15] (SImulation of Marine Operations) developed by *SINTEF Ocean* and *DNV* [36];

[15] https://www.dnvgl.com/services/complex-multibody-calculations-simo-2311

- *aNySIM*[16] developed by *MARIN* [37];
- *OrcaFlex*[17] developed by *Orcina* [38];
- *DeepLines*[18] developed by *Principia* and *IFP Energies Nouvelles* [39];
- *ProteusDS*[19] developed by *DSA* [40];
- *ARIANE*[20] developed by *Bureau Veritas* [41];
- *ANSYS Aqwa*[21] developed by *ANSYS*;
- *FryDoM*[22] (Flexible and Rigid body Dynamic modeling for Marine operations) developed by *D-ICE ENGINEERING* and *Ecole Centrale de Nantes*;
- *MOSES* [23] (Multi Operational Structural Engineering Simulator) developed by *Bentley Systems* [42].

These software packages have the same features overall:

- Dynamical computations are multibody;
- Bodies are rigid with six degrees of freedom;
- Mechanical interactions between bodies are taken into account;
- Wave loads are computed using a frequency-domain potential flow theory;
- Hydrostatic loads are based on a linear approach;
- Wave-structure interaction between floating bodies are taken into account;
- Additional loads (wind, current, mooring, second-order wave loads, etc.) may be specified.

Nevertheless, there are differences in the details. For example, *OrcaFlex* and *ProteusDS* use a high-order lumped mass cable theory while *DeepLines* and *FryDoM* use the finite element method (cf. section II.1). Nonlinear hydrostatic calculations are available in *DeepLines* and *FRyDoM*. Kinematic loops are supported by *FRyDoM* (cf. section I.1).

4.6 Conclusions of the state of the art

From this review of the state of the art of the numerical simulation of lowering and lifting operations, the following main conclusions arise:

- Dynamical solvers have been improved over the years until being able to simulate multibody systems including cables and mechanical interactions;

[16] https://www.marin.nl/publication/anysim-a-versatile-hydrodynamics-engineering-tool
[17] https://www.orcina.com/
[18] http://www.principia-group.com/blog/product/produit-deeplines/
[19] https://dsa-ltd.ca
[20] https://www.bureauveritas.com/home/about-us/our-business/marine-and-offshore/offshore/our-solutions/software
[21] https://www.ansys.com/fr-fr/products/structures/ansys-aqwa
[22] https://frydom.org/
[23] https://www.bentley.com/en/products/brands/moses

- These dynamical solvers have been coupled to wave-structure interaction solvers, primarily using the linear potential flow theory, enabling the simulation of floating cranes with a payload in the air only;
- Hydrodynamic interactions between the floating crane and the payload are rarely studied and when it is done, it is at the price of a too simple mechanical model;
- The use of a linear frequency-domain potential flow based solver is not consistent when the payload is lowered or lifted due to the violation of the small amplitude motion hypothesis;
- Similarly, the steadiness of the wave-structure interaction due to the linear potential flow theory is not consistent when floating and immersed bodies have a large relative motion;
- Commercial software packages are based on the same approach: a multibody dynamical solver coupled with a linear potential flow based solver and suffer of the aforementioned conclusions when lowering and lifting operations are simulated;
- Norms present guidelines based models which can be too simple.

That is why a new approach is required for performing numerical simulations of lowering and lifting operations. It is necessary to couple both a dynamical solver which would be able to handle multibody numerical computations involving several bodies, articulations, cables and a winch and a wave-structure interaction solver which would be consistent enough to evaluate the unsteady hydrodynamic loads and the hydrodynamic interactions between bodies with a large relative amplitude motion. This analysis is the motivation for this PhD project.

5 Context of the project

This PhD work was supported by a *CIFRE*[24] scholarship through a partnership between the *LHEEA*[25] laboratory of *Ecole Centrale de Nantes* and *INNOSEA*.

5.1 The *LHEEA* laboratory of *Ecole Centrale de Nantes*

The *LHEEA* laboratory of *Ecole Centrale de Nantes* is an internationally well-known laboratory specialized in numerical simulations and experiments at model scale and *in situ* in four themes:

- Free surface hydrodynamics;
- Fluid-structure interaction;
- Dynamics of the atmosphere;
- Systems approach for ground and marine propulsion systems.

[24] Convention Industrielle de Formation par la Recherche or Industrial Convention of Formation by Research

[25] Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphèrique

One research topic, dedicated to the Ocean Energy and Ocean Waves Group of the lab., is the study of the marine renewable energy devices (offshore wind turbines, wave energy converters, tidal turbines) by numerical simulation of the resource (wave, wind), the motion of the systems and their performance.

This PhD is the continuation of several works of the *LHEEA*. For example Rongère and Clément [43] presented a methodology to model and simulate multibody offshore structures. To do so, a dynamic algorithm was coupled with the linear potential flow theory in case of a single floating body (a wave energy converter in the paper).

The laboratory has also developed a potential flow based solver using the weak-scatterer hypothesis [44, 45], named *WS_CN*. This numerical tool computes the unsteady hydrodynamic loads without doing any assumption on the amplitude of the body motion nor on the wave steepness. So far, the application of this tool has been the simulation of single wave energy converters to evaluate more accurately their performance. This solver will be used in our work.

Thus, the interests of the *LHEEA* for this PhD are the pursuit of research about the numerical simulation of multibody offshore structures, the continuation of the development of an unsteady potential flow based solver, the quantification of its interest and its application in a new field: marine operations.

5.2 *INNOSEA*

INNOSEA is an engineering company specialized in marine renewable energy projects and is a spin-off company from *Ecole Centrale de Nantes*. *INNOSEA*'s main areas of expertise include hydrodynamic, aeroelastic, structural and metocean analyses. The company is also an editor of software tools for the numerical simulation of marine renewable energy devices.

This PhD is part of the research and development strategy of *INNOSEA* to create a numerical tool able to simulate multibody offshore structures. To do so, the software *InWave* has been developed by *INNOSEA* and the *LHEEA* to simulate articulated marine energy devices [46]. This tool is coupled with the linear potential flow based solver *Nemoh* [47].

The interests of *INNOSEA* for this PhD are the extension of *InWave* to handle the simulation of marine operations and the coupling with an unsteady potential flow based solver when the use of *Nemoh* could not be consistent and accurate enough.

6 Objectives

In order to fill the knowledge gaps exposed in subsection 4.6 and bring a contribution to the numerical simulation of lowering and lifting operations, the main objectives of the PhD thesis are:

- The implementation of cable and winch modelling capabilities in the existing multibody dynamic numerical tool (*InWave*);
- The extension of the existing hydrodynamic numerical tool (*WS_CN*) to perform simulations involving several immersed or surface-piercing bodies;
- The coupling between these two solvers to simulate lowering and lifting operations;

– The comparison of the classical approach to model lowering operations using the linear potential flow theory and the developed numerical tool to quantify the effects of the wave-structure interaction theory used in *WS_ CN*.

7 Layout

The first chapter of the thesis introduces the multibody theory used in *InWave*. After a state of the art of the multibody system dynamics, the modelling approach based on relative coordinates along with the dynamic algorithm, named the Composite Rigid-Body Algorithm or *CRBA*, are presented in details. The motion equation of the multibody systems is established.

The second chapter is devoted to cable dynamics. A state of the art of the different cable modelling approaches is exposed. Two approaches are used to simulate cables, one is based on the *CRBA* while the second method uses the low-order lumped mass theory. The latter is developed into a solver named *CableDyn*, validated from comparisons with experiments. Comparisons are done between the two approaches. A winch model is also presented to wind or unwind a cable by modifying the number of cable elements during the numerical simulations.

The third chapter is dedicated to the hydrodynamic theory. A state of the art of the different potential flow approach is presented. The theory of the model used in this PhD, based on the weak-scatterer hypothesis, is presented. Its numerical implementation through the solver *WS_ CN* is introduced for the solving of the hydrodynamic problem as well as for the mesh generator. This numerical tool has been extended to multibody simulations and validated from comparisons with measurements in case of forced motions and with a linear potential-flow based solver for free motions. Other developments such as the free surface remeshing, the parallelization of the computation of the influence coefficients or the use of a sparse linear system solver are also presented.

The fourth chapter introduces the fluid-structure coupling between the multibody solver *InWave* and the hydrodynamic solver *WS_ CN*. A state of the art of the fluid-structure interaction is provided. Then, the theory of the tight coupling is explained. The language binding between the two existing numerical tools is detailed as *InWave* is implemented in C++ and *WS_ CN* in Fortran. The tight coupling is checked from comparisons with the hydrodynamic solver alone. Other coupling strategies are also studied. The coupling between *InWave* and *CableDyn* is presented.

The fifth chapter proposes a new mesh strategy to overcome the lack of robustness of the initial mesh generator in *WS_ CN* originally. The explanation of this choice along with a state of the art of the mesh generation are provided. The development of each step of this new mesh generator and its comparison with the initial one are given.

The last chapter is dedicated to the experiments of an upending operation of a spar conducted in the wave basin of *Ecole Centrale de Nantes*. The experimental device is presented, together with the measurements obtained. These results are used for comparison with the numerical results coming from the coupling between *InWave*, *CableDyn*, *WS_ CN* and the new mesh generator.

Chapter I

Multibody System Dynamics

Contents

The simulation of marine operations requires solving the dynamics of a multibody system, constituted, at least, of a floating crane, a cable and a payload in the case of a lowering operation. This chapter introduces a state of the art of the multibody system dynamics which leads to the presentation of the different methods to model a multibody system. Several dynamic algorithms are exposed. The Composite-Rigid-Body Algorithm, which is used through this PhD thesis, is presented in details.

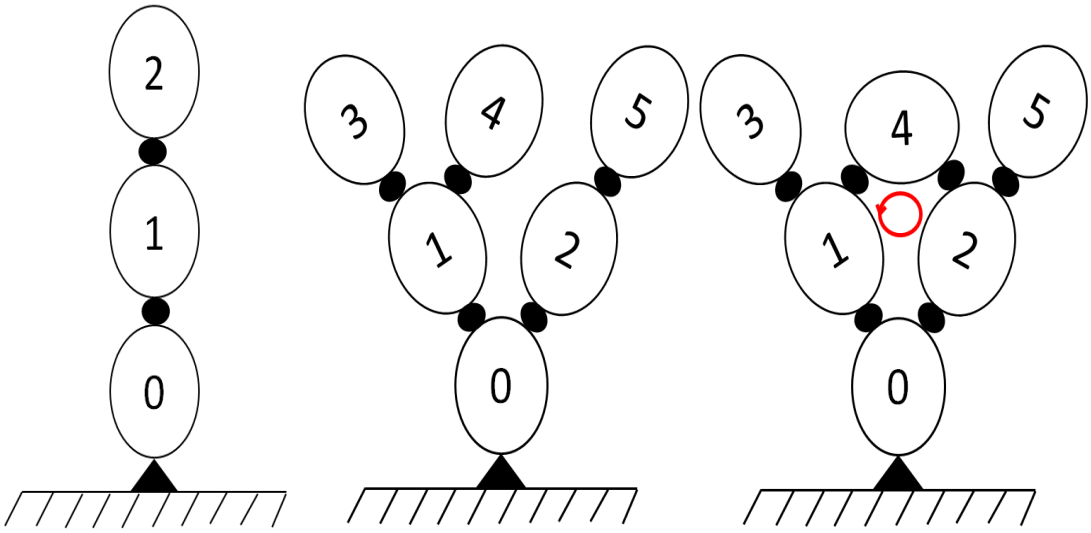
I.1 State of the art

I.1.1 Classification of multibody systems

A multibody system is an assembly of connected rigid or flexible bodies. Each connection between two bodies is called a joint. Only rigid bodies are considered in this work. Three families of multibody systems exist:

- Open-loop or open-chain systems;
- Kinematic trees or open chain with multiple branches;
- Closed-loop systems.

Open-loop systems are unbranched kinematic trees and are composed of bodies which have exactly one ancestor and one successor. With kinematic trees, each body may have several successors but always a single ancestor. The consequence of this definition is the appearance of branches in the multibody system. Finally, closed-loop systems form the most general case. Each body may have several successors and ancestors, leading to kinematic loops. One example of each family of multibody systems is shown in Figure I.1.



(a) Open-loop system **(b)** Kinematic tree **(c)** Closed-loop system

Figure I.1 – Sketches of the different multibody system families. Black circle: body, black point: joint and red arrow: kinematic loop.

I.1.2 Dynamic approaches to multibody simulation

Multibody dynamics is composed of:

- Forward or direct dynamics: the acceleration is evaluated from the knowledge of the loads acting on the multibody system;
- Inverse dynamics: the loads produced by a given acceleration are computed.

Direct dynamics is used for the simulation of the free body motion. Inverse dynamics is required for the motion control or the trajectory planning. Both approaches are based on the motion equation of the multibody system but used differently. The mathematical model of the forward dynamics (FD) and the inverse dynamics (ID) can be written as follows [48]:

$$\ddot{\mathbf{q}} = \mathrm{FD}(model, \mathbf{q}, \dot{\mathbf{q}}, \mathbf{\Gamma}) \tag{I.1}$$

and

$$\mathbf{\Gamma} = \mathrm{ID}(model, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tag{I.2}$$

with:

- $\mathbf{q}$ the position vector;
- $\dot{\mathbf{q}}$ the velocity vector;
- $\ddot{\mathbf{q}}$ the acceleration vector;
- $\mathbf{\Gamma}$ the vector of the internal loads;
- *model* the input characteristics of the multibody system (mass, inertia, external loads, etc.).

Only forward dynamics will be used in this PhD thesis.

I.1.3 Direct dynamics algorithms

Direct dynamics algorithms can mainly be formulated according to four different approaches [31, 49]:

- The Articulated-Body Algorithm (*ABA*);
- The Composite-Rigid-Body Algorithm (*CRBA*);
- The Augmented Formulation;
- The Discrete Euler-Lagrange Equation.

I.1.3.1 The *ABA* and the *CRBA*

Both the Articulated-Body Algorithm and the Composite-Rigid-Body Algorithm are part of the embedding technique as defined by Shabana [50]. Instead of using the classical Cartesian coordinates (also named the redundant coordinate set [51]), these algorithms are based on a joint coordinate set (also named a reduced or relative coordinate set). With the Cartesian coordinates, each body is located from a unique global frame whereas by using joint coordinates, each body is located from one of its ancestors. The Articulated-Body Algorithm and the Composite-Rigid-Body Algorithm

are directly applicable to kinematic trees (which include open-chain systems). Both of them are based on the Newton-Euler equation, written in a recursive way. But they differ in the way they compute the accelerations. The Articulated-Body Algorithm propagates the constraints from one body to the next one and calculates the acceleration of one body at a time. The Composite-Rigid-Body Algorithm is more general, because it consists in writing the motion equation of the whole system to find the accelerations of all bodies at one time [48].

The *ABA* has been applied by Rongère and Clément [43] to simulate the motion of a single wave energy converter. To evaluate the hydrodynamic loads, the Articulated-Body Algorithm was coupled with the linear potential flow-based theory. Within this theory, one part of the hydrodynamic loads depends on the acceleration of the body: the added-mass loads. The *ABA* enables the simulation of a single body subject to hydrodynamic loads. But the simulation of several floating or immerged bodies in hydrodynamic interactions is more challenging and requires the use of the Composite-Rigid-Body Algorithm. Indeed, in this case, the added-mass loads of one body depend not only on the acceleration of the body itself, but also on the acceleration of all other bodies subject to hydrodynamic loads. As the *ABA* computes the acceleration of one body at a time, it fails to simulate hydrodynamic interactions, hence the need to use the *CRBA*. Combourieu et al. [46] coupled the *CRBA* with a linear potential flow-based theory to simulate a complex wave energy converter with several moving bodies in hydrodynamic interactions.

Closed-loop systems can be modeled by the *ABA* and the *CRBA* but with additional developments compared to kinematic trees. Closed-loops involve joint variables which are not independent anymore. The main strategy to deal with such multibody systems is to make cuts in the closed-loop system in order to find a kinematic tree, named a spanning tree. At the cuts, a constraint equation is used based on Lagrangian multipliers [48].

I.1.3.2 The Augmented Formulation

Lagrangian multipliers form the key point of the Augmented Formulation. Within this approach, the redundant Cartesian coordinates are used to locate each body of the multibody system. The motion equation of the whole multibody system is established assuming each body has six degrees of freedom (*dof*). As the redundancy of the chosen coordinate system is due to the geometrical constraints in the multibody system, it is necessary to add constraint equations to take into account these kinematic constraints. For the sake of clarity, let us consider two six-*dof* bodies linked by a revolute joint. There are seventeen unknowns in total: the six degrees of freedom of each body and five components of the internal loads in the joint. But only twelve scalar motion equations are available. Nevertheless, an algebraic equation may be written to ensure the contact between the two bodies despite their relative rotational motion. This constraint equation allows to express the internal loads in the joint and then to close the system of equations.

Constraint equations and inequations (represented by the function ϕ) are sorted in several categories (Figure I.2). Equality kinematic constraints represent permanent physical contacts between bodies, whereas inequality constraints arise when bodies can make contact or impact and be separated [52].

Constraint equations mainly depend on time and positions only (holonomic constraints) or are velocity-dependent (nonholonomic constraints). Nonholonomic constraints appear for rolling contact whereas holonomic constraints occur for sliding con-

tact. Holonomic constraint group is split into two categories according to the time dependency. Rheonomic constraints are time-dependent and used to add prescribed motions (excitation) in joints. Scleronomic constraints are time-independent and form the most common category of constraints.

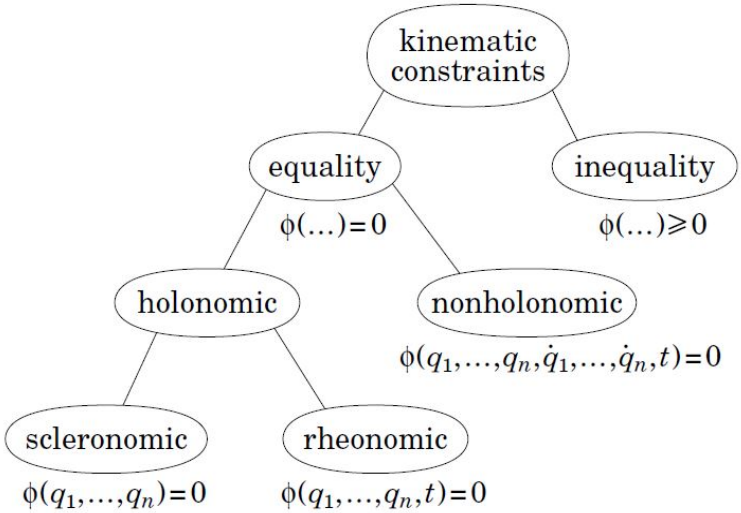


Figure I.2 – Classification of constraints (from [48])

For instance, considering a two-dimensional pendulum made of a punctual mass and an inextensible rod of negligible mass (Figure I.3), the scleronomic constraint equation is:

$$\phi(x,y) = x^2 + y^2 - l^2 = 0 \tag{I.3}$$

with:

- x the vertical position of the mass;
- y the horizontal position of the mass;
- l the rod length.

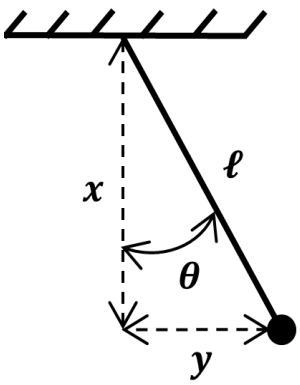


Figure I.3 – Simple pendulum

If this example was modelled using a reduced coordinate set, the unique variable in two dimensions would be θ, the angular position of the mass. With the Augmented

Formulation, the Newton-Euler motion equation of the whole multibody system is:

$$\mathbf{M}\ddot{\mathbf{X}} = \mathbf{F_c} + \mathbf{F_{ext}} \tag{I.4}$$

with:

- $\mathbf{M}$ the whole system mass matrix;
- $\ddot{\mathbf{X}}$ the vector of the Cartesian accelerations of all the bodies;
- $\mathbf{F_c}$ the internal loads in the whole system;
- $\mathbf{F_{ext}}$ the external loads acting on the whole system;

Both $\mathbf{F_c}$ and $\ddot{\mathbf{X}}$ are unknown in a direct dynamics algorithm. Constraint equations are added to close the system of equations:

$$\boldsymbol{\phi}(\mathbf{X}, t) = 0 \tag{I.5}$$

Internal loads are linked to the constraint equations by the use of Lagrange multipliers $\boldsymbol{\lambda}$:

$$\mathbf{F_c} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}}^T \boldsymbol{\lambda} \tag{I.6}$$

Two time-differentiations of (I.5) allow to couple the constraint equations to the whole system motion equation (I.4):

$$\begin{pmatrix} \mathbf{M} & -\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}}^T \\ \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{X}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{F_{ext}} \\ \frac{\partial^2 \boldsymbol{\phi}}{\partial t \partial \mathbf{X}} \dot{\mathbf{X}} \end{pmatrix} \tag{I.7}$$

Equations (I.4) and (I.5) form a differential-algebraic system of equations (*DAE*) of index 3. The index is given by the number of time-differentiations required to transform a *DAE* into a first-order system of ordinary differential equations (*ODE*). Two time-differentiations were necessary to write (I.7). In order to obtain the time-derivation of the Lagrange multipliers, a third time-differentiation is applied and leads to a *DAE* of index 3. The index also indicates the numerical difficulty to solve the *DAE* [53].

Several techniques exist to solve index-3 *DAE*. The direct integration of index-3 *DAE* (Equations (I.4) and (I.5)) being subject to numerical instabilities, other strategies are preferred [54]. One approach is to use a stabilization method. Equation I.7 is integrated in time. Doing so, the validity of (I.5) is not ensured and a drift motion appears because of the time integration. This is why a stabilization method is used to take into account the violation of (I.5). It is based on an alteration of the internal loads to compensate the errors of the direct integration [53, 55]. Another possible method is based on the reduction of the *DAE*-index to 1 or 2 [54]. Additional Lagrange multipliers (reduction to index 2) or new variables which match the same time-derivative of the Lagrange multipliers (reduction to index 1) are added to the system of equations.

The use of a relative coordinate set with the *ABA* or the *CRBA* to model kinematic trees always leads to an *ODE* system which is greatly easier to solve numerically.

I.1.3.3 The Discrete Euler-Lagrange equation

The last main direct dynamics algorithm formulation is the use of the Euler-Lagrange equation instead of the Newton-Euler equation. Ham et al. [27] discretized the Euler-Lagrange equation in presence of kinematic constraints, different kinds of joints and closed-loops. This method also leads to an automatic writing of the multibody motion equations.

I.1.3.4 Discussion

Each formulation of the direct dynamics algorithms enables the simulation of all multibody systems but with different numerical difficulties. Both the Articulated-Body-Algorithm and the Composite-Rigid-Body-Algorithm are efficient and straightforward for computing the dynamics of kinematic trees but need much more attention on how the input data are carried out due to the use of a reduced coordinate set. The extension to closed-loop systems can be challenging too. Regarding the Augmented Formulation, no distinction is done between kinematic trees and closed-loop systems as constraint equations are always required. The solving of the kinematic constraints coupled with the motion equations is a difficult numerical task.

I.1.4 *InWave*

The multibody dynamics solver used through this PhD thesis is *InWave*. It has been the result of a joint development between the *LHEEA* laboratory of *Ecole Centrale de Nantes* and *INNOSEA* since 2014. The direct dynamics algorithm of *InWave* is the Composite-Rigid-Body Algorithm for kinematic trees. It is implemented in C++. This algorithm has been coupled with the linear potential flow-based solver *NEMOH* [47]. In its early developments, *InWave* was a numerical tool purely dedicated to the modeling of Wave Energy Converters (*WEC*) [46]. Since 2015, three PhD projects have been launched to extend the capabilities of *InWave*.

The work of Vincent Leroy, PhD student at the *LHEEA* laboratory and *INNOSEA*, is about the unsteady behaviour of offshore floating wind turbines. *InWave* has been coupled to a free-vortex wake theory-based unsteady aerodynamic solver [56]. This coupling was firstly used to investigate the behaviour of floating horizontal-axis wind turbines and was compared to state-of-the-art steady aerodynamic models. Eventually, a steady double multiple streamtube theory-based solver has been implemented and coupled to *InWave* to study floating vertical-axis wind turbines. The impact of the aerodynamic solver on the behaviour at sea of such systems has been investigated with a comparison to the unsteady aerodynamic coupling results [57].

The work of David Ogden, EngD student at University of Edinburgh (UK) and *INNOSEA* focuses on the numerical simulation of complex wave energy converters with kinematic loops, flexible bodies, etc. To do so, *InWave* has been coupled to the multibody numerical tool *HotInt*. This has opened new features: complex joints between bodies, closed-loops or ropes. The developed coupling is being validated with a series of *WECs* [58].

The last axis of development of *InWave* is the aim of this PhD thesis: the simulation of marine operations.

I.2 Modeling of multibody systems

I.2.1 Numbering of the bodies

A multibody system is composed of $n+1$ bodies. As *InWave* only considers kinematic trees, the number of joints is n. If the number of joints is higher than n, there is at least one kinematic loop. In Figure I.1b, $n = 5$ and is equal to the number of joints, whereas in Figure I.1c, $n = 5$ but the number of joints is 6 and one closed-loop is present.

A number is given to each body. The body numbered 0 is the base of the multibody system. It is the unique body for which Cartesian coordinates with respect to the global earth-fixed inertial frame are used . The base can be either fixed or floating (moving). Other bodies are numbered with increasing order from the base. Their position is based on the modified Denavit-Hartenberg parameters (mDH). The numbering of the bodies defines an antecedence relationship through the multibody system. Mathematically, if the i-th body is the ancestor of the j-th body, then $i < j$ and the antecedence relationship is written:

$$i = a_j \tag{I.8}$$

By definition, the base is the single body without ancestor and the terminal bodies, considered as the leaves of the kinematic tree, are the bodies without successor. Joints are numbered such as the joint j connects the body a_j to the body j .

I.2.2 The modified Denavit-Hartenberg parameters

InWave uses a reduced set of coordinates to locate each body of the multibody system. It is different from the classical Cartesian coordinates. It follows the geometric configuration created by Denavit and Hartenberg for open chain systems and extended to any multibody system (with or without closed-loops) by Khalil and Kleinfinger [59]. The advantages of these coordinates are that they allow to describe any kind of multibody systems with a minimum of parameters without ambiguities.

Only one degree-of-freedom joints are considered, either revolute or prismatic. A revolute joint, respectively a prismatic joint, grants for a rotation, respectively a translation, between two bodies. To produce more complex joints, virtual massless bodies are added to ensure the necessary degrees of freedom. For instance, in case of a ball joint linking two physical bodies, two virtual massless bodies are required to create three concurrent-axis and orthogonal to each other revolute joints. The articular variable defining the rotation around the revolute axis joint or the translation along the prismatic axis joint is q_j. The set of the articular variables, written $\mathbf{q}$, is the articular position vector of the multibody system.

Each body has a local frame, Σ_j, of origin $\mathbf{O}_j$ and orthonormal basis $(\mathbf{x}_j, \mathbf{y}_j, \mathbf{z}_j)$. An extra unit vector, $\mathbf{u}_j$ is also built to be able to define the mDH parameters in every situation. The local base frame, Σ_0 is defined with respect to the global inertial earth fixed reference frame Σ_e, of origin $\mathbf{O}_e$ and orthonormal basis $(\mathbf{x}_e, \mathbf{y}_e, \mathbf{z}_e)$. $\mathbf{O}_e, \mathbf{x}_e$ and $\mathbf{y}_e$ lie on the mean water free surface and $\mathbf{z}_e$ points upwards.

The step-by-step construction of these local frames is based on the following rules [43, 60]:

- $\mathbf{z}_j$ is along the axis of the joint j;
- $\mathbf{x}_j$ is along the mutual perpendicular to $\mathbf{z}_j$ and one of the succeeding joint axis $\mathbf{z}_k$ such as $a_k = j$. This definition leads to three possible cases:
  - If $\mathbf{x}_{a_j}$ is not perpendicular to $\mathbf{z}_j$, $\mathbf{u}_j$ is defined along the mutual perpendicular to $\mathbf{z}_{a_j}$ and $\mathbf{z}_j$.
  - If $\mathbf{x}_{a_j}$ is perpendicular to $\mathbf{z}_j$, then $\mathbf{u}_j$ is equal to $\mathbf{x}_{a_j}$;
  - If the body j is a terminal body, $\mathbf{x}_j$ can be defined arbitrarily.
- $\mathbf{y}_j$ completes the basis such as $(\mathbf{x}_j, \mathbf{y}_j, \mathbf{z}_j)$ forms an orthonormal basis.

In practice, $\mathbf{z}_j$-axes are first defined for all bodies, then both $\mathbf{x}_j$ and $\mathbf{u}_j$ and finally $\mathbf{y}_j$.

Once the local frames are established, the six modified Denavit-Hartenberg parameters are defined such as:

- $i = a_j$;
- γ_j the angle between $\mathbf{x}_i$ and $\mathbf{u}_j$ around $\mathbf{z}_i$;
- b_j the distance between $\mathbf{x}_i$ and $\mathbf{u}_j$ along $\mathbf{z}_i$;
- α_j the angle between $\mathbf{z}_i$ and $\mathbf{z}_j$ around $\mathbf{u}_j$;
- d_j the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$ along $\mathbf{u}_j$;
- θ_j the angle between $\mathbf{u}_j$ and $\mathbf{x}_j$ around $\mathbf{z}_j$;
- r_j the distance between $\mathbf{u}_j$ and $\mathbf{x}_j$ along $\mathbf{z}_j$.

If $\mathbf{u}_j = \mathbf{x}_{a_j}$, then $\gamma_j = 0$ and $b_j = 0$, so only four parameters are necessary. Otherwise, the six parameters are useful. For a given multibody system, the set of *mDH* parameters is not unique. It depends on the numbering of the bodies, the order of the joints and the orientation of the local frame unit-vectors. Figure I.4 presents a sketch of the local frames and the modified Denavit-Hartenberg parameters in case of a set of three bodies of a kinematic tree.

The six modified Denavit-Hartenberg parameters $(\gamma_j, b_j, \alpha_j, d_j, \theta_j, r_j)$ form the reduced coordinate set used in *InWave*. If the joint j is revolute, then $q_j = \theta_j$, otherwise $q_j = r_j$ for a prismatic joint. The general definition of the articular variable becomes:

$$q_j = \bar{\sigma}_j \theta_j + \sigma_j r_j \tag{I.9}$$

with:

- $\sigma_j = 0$ if the joint j is revolute;
- $\sigma_j = 1$ if the joint j is prismatic;
- $\bar{\sigma}_j = 1 - \sigma_j$.

Other *mDH* parameters stay constant during the simulation.

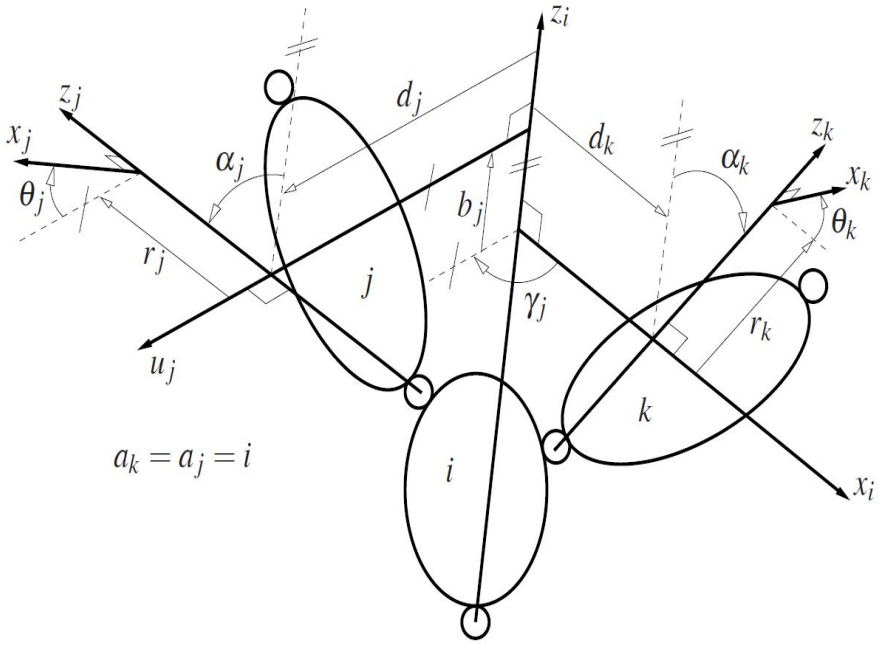


Figure I.4 – Local frames and modified Denavit-Hartenberg parameters in case of a kinematic tree (from [43])

From the *mDH* parameters, it is possible to define the rotation matrix from Σ_j to the ancestor local frame Σ_{a_j} [43]:

$$ {}^{a_j}\mathbf{R}_j = \begin{pmatrix} C_{\gamma_j}C_{\theta_j} - S_{\gamma_j}C_{\alpha_j}S_{\theta_j} & -C_{\gamma_j}S_{\theta_j} - S_{\gamma_j}C_{\alpha_j}C_{\theta_j} & S_{\gamma_j}S_{\alpha_j} \\ S_{\gamma_j}C_{\theta_j} + C_{\gamma_j}C_{\alpha_j}S_{\theta_j} & -S_{\gamma_j}S_{\theta_j} + C_{\gamma_j}C_{\alpha_j}C_{\theta_j} & -C_{\gamma_j}S_{\alpha_j} \\ -S_{\alpha_j}S_{\theta_j} & S_{\alpha_j}C_{\theta_j} & C_{\alpha_j} \end{pmatrix} \tag{I.10} $$

with:

- $C_* = \cos(*)$;
- $S_* = \sin(*)$.

The composition of rotation matrices ensure:

$$ {}^{j}\mathbf{R}_i = {}^{j}\mathbf{R}_k \, {}^{k}\mathbf{R}_i \tag{I.11} $$

Regarding their inverse, the well-known formula is:

$$ {}^{j}\mathbf{R}_i^{-1} = {}^{j}\mathbf{R}_i^{T} \tag{I.12} $$

The position of the origin of Σ_j from the origin of Σ_{a_j} and expressed in Σ_{a_j} is:

$$ {}^{a_j}\mathbf{P}_j = \begin{pmatrix} d_jC_{\gamma_j} + r_jS_{\gamma_j}C_{\alpha_j} \\ d_jS_{\gamma_j} - r_jC_{\gamma_j}S_{\alpha_j} \\ r_jC_{\alpha_j} + b_j \end{pmatrix} \tag{I.13} $$

In other words, if ${}^{e}\mathbf{O}_{a_j}\mathbf{O}_j$ represents the position of the frame Σ_j with respect to the frame Σ_{a_j} and expressed in Σ_e, it comes:

$$ {}^{a_j}\mathbf{P}_j = {}^{a_j}\mathbf{R}_e \, {}^{e}\mathbf{O}_{a_j}\mathbf{O}_j \tag{I.14} $$

So far, local frames have been built except for the base. The base frame is directly defined with respect to the earth-fixed inertial frame Σ_e. Σ_0 is obtained after a translation of vector $\mathbf{O}_e\mathbf{O}_0$ and, then, three rotations based on the definition of the Cardan angles. Their construction follows the next steps:

- Translation of vector $\mathbf{O}_e\mathbf{O}_0 = {}^e\mathbf{P}_0$ to move Σ_e from $\mathbf{O}_e$ to $\mathbf{O}_0$;
- Rotation from Σ_e to Σ_{ψ_0} around $\mathbf{z}_e$ through an angle ψ_0. Σ_{ψ_0} is defined by the orthonormal basis $(\mathbf{x}_{\psi_0}, \mathbf{y}_{\psi_0}, \mathbf{z}_{\psi_0})$ where $\mathbf{z}_{\psi_0} = \mathbf{z}_e$. The associated rotation matrix is:

$$ {}^e\mathbf{R}_{\psi_0} = \begin{pmatrix} C_{\psi_0} & -S_{\psi_0} & 0 \\ S_{\psi_0} & C_{\psi_0} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{I.15} $$

- Rotation from Σ_{ψ_0} to Σ_{θ_0} around $\mathbf{y}_{\psi_0}$ through an angle θ_0. Σ_{θ_0} is defined by the orthonormal basis $(\mathbf{x}_{\theta_0}, \mathbf{y}_{\theta_0}, \mathbf{z}_{\theta_0})$ where $\mathbf{y}_{\theta_0} = \mathbf{y}_{\psi_0}$. The associated rotation matrix is:

$$ {}^{\psi_0}\mathbf{R}_{\theta_0} = \begin{pmatrix} C_{\theta_0} & 0 & S_{\theta_0} \\ 0 & 1 & 0 \\ -S_{\theta_0} & 0 & C_{\theta_0} \end{pmatrix} \tag{I.16} $$

- Rotation from Σ_{θ_0} to $\Sigma_{\varphi_0} = \Sigma_0$ around $\mathbf{x}_{\theta_0}$ through an angle φ_0. Σ_{φ_0} is defined by the orthonormal basis $(\mathbf{x}_{\varphi_0}, \mathbf{y}_{\varphi_0}, \mathbf{z}_{\varphi_0})$ where $\mathbf{x}_{\varphi_0} = \mathbf{x}_{\theta_0}$. The associated rotation matrix is:

$$ {}^{\theta_0}\mathbf{R}_{\varphi_0} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_{\varphi_0} & -S_{\varphi_0} \\ 0 & S_{\varphi_0} & C_{\varphi_0} \end{pmatrix} \tag{I.17} $$

The six-component generalized position vector to locate Σ_0 with respect to Σ_e in term of position and orientation is:

$$ \boldsymbol{\eta}_0 = \begin{pmatrix} \boldsymbol{\eta}_0^{trans} \\ \boldsymbol{\eta}_0^{rot} \end{pmatrix} \tag{I.18} $$

with:

- $\boldsymbol{\eta}_0^{trans} = \begin{pmatrix} x_0 & y_0 & z_0 \end{pmatrix}^T = {}^e\mathbf{P}_0$ the Cartesian position of $\mathbf{O}_0$ from $\mathbf{O}_e$;
- $\boldsymbol{\eta}_0^{rot} = \begin{pmatrix} \varphi_0 & \theta_0 & \psi_0 \end{pmatrix}^T$ the angular position using the Cardan angles.

$x_0, y_0, z_0, \varphi_0, \theta_0$ and ψ_0 represent the surge, sway, heave, roll, pitch and yaw of the base body, respectively. If the base is fixed, then $\Sigma_0 = \Sigma_e$ and $\dot{\boldsymbol{\eta}}_0 = \mathbf{0}_{6\times 1}$.

The rotation matrix from Σ_0 to Σ_e becomes [43]:

$$ {}^e\mathbf{R}_0 = {}^e\mathbf{R}_{\psi_0}\, {}^{\psi_0}\mathbf{R}_{\theta_0}\, {}^{\theta_0}\mathbf{R}_{\varphi_0} \tag{I.19} $$

$$ = \begin{pmatrix} C_{\psi_0}C_{\theta_0} & -S_{\psi_0}C_{\varphi_0} + C_{\psi_0}S_{\theta_0}S_{\varphi_0} & S_{\psi_0}S_{\varphi_0} + C_{\psi_0}C_{\varphi_0}S_{\theta_0} \\ S_{\psi_0}C_{\theta_0} & C_{\psi_0}C_{\varphi_0} + S_{\varphi_0}S_{\theta_0}S_{\psi_0} & -C_{\psi_0}S_{\varphi_0} + S_{\psi_0}C_{\varphi_0}S_{\theta_0} \\ -S_{\theta_0} & C_{\theta_0}S_{\varphi_0} & C_{\theta_0}C_{\varphi_0} \end{pmatrix} \tag{I.20} $$

The results of this section allows to locate every point of each body in all local frames Σ_j and in the global inertial frame Σ_e. To perform automatic dynamic computations, it is necessary to figure out the kinematic recursive relationships for the velocities and the accelerations of each body. This is the goal of the next section.

I.3 Kinematic recursive equations

The six-component generalized velocity vector of body j with respect to Σ_e and expressed in Σ_j is:

$$ {}^j\mathbf{V}_j = \begin{pmatrix} {}^j\mathbf{v}_j \\ {}^j\boldsymbol{\omega}_j \end{pmatrix} \tag{I.21} $$

with:

- ${}^j\mathbf{v}_j$ the linear velocity of body j with respect to Σ_e at the point $\mathbf{O}_j$ and expressed in Σ_j.
- ${}^j\boldsymbol{\omega}_j$ the angular velocity of body j with respect to Σ_e and expressed in Σ_j.

To establish the kinematic recursive equation for the velocities, let us introduce two extra notations to manipulate more easily both linear and angular velocities:

- ${}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_e)$ the velocity of the point $\mathbf{O}_j$ of Σ_j with respect to Σ_e and expressed in Σ_e. The definition of ${}^j\mathbf{v}_j$ involes:

$$ {}^j\mathbf{v}_j = {}^j\mathbf{R}_e \, {}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_e) \tag{I.22} $$

- ${}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e)$ the angular velocity of Σ_j with respect to Σ_e and expressed in Σ_e. The definition of ${}^j\boldsymbol{\omega}_j$ involes:

$$ {}^j\boldsymbol{\omega}_j = {}^j\mathbf{R}_e \, {}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e) \tag{I.23} $$

In this section, it is assumed that $i = a_j$.

The transport of the linear velocity from $\mathbf{O}_j$ of Σ_j to $\mathbf{O}_i$ of Σ_i provides:

$$ {}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_e) = {}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_i) + {}^e\mathbf{v}(\mathbf{O}_i, \Sigma_i/\Sigma_e) + {}^e\boldsymbol{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j \tag{I.24} $$

with:

- ${}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_i)$ the relative linear velocity between the frames Σ_j and Σ_i. It only depends on the nature of the joint j:

$$ {}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_i) = \sigma_j \dot{q}_j \, {}^e\mathbf{z}_j \tag{I.25} $$

- ${}^e\mathbf{z}_j$ the joint axis of joint j in Σ_e. Its value in the local frame of the body j is:

$$ {}^j\mathbf{z}_j = {}^j\mathbf{R}_e \, {}^e\mathbf{z}_j = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T \tag{I.26} $$

The transport of the angular velocity from Σ_j to Σ_i gives:

$$ {}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e) = {}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_i) + {}^e\boldsymbol{\Omega}(\Sigma_i/\Sigma_e) \tag{I.27} $$

with ${}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_i)$ the relative angular velocity between the frames Σ_j and Σ_i. It only depends on the nature of the joint j:

$$ {}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_i) = \bar{\sigma}_j \dot{q}_j \, {}^e\mathbf{z}_j \tag{I.28} $$

Equations (I.24) and (I.27) are now projected in the local frame of body j using (I.14), (I.22) and (I.23):

$$ {}^j\mathbf{v}_j = {}^j\mathbf{R}_i \, {}^i\mathbf{v}_i - {}^j\mathbf{R}_i \mathbf{S}({}^i\mathbf{P}_j) \, {}^i\boldsymbol{\omega}_i + \sigma_j \dot{q}_j \, {}^j\mathbf{z}_j \tag{I.29} $$

$$ {}^j\boldsymbol{\omega}_j = {}^j\mathbf{R}_i \, {}^i\boldsymbol{\omega}_i + \bar{\sigma}_j \dot{q}_j \, {}^j\mathbf{z}_j \tag{I.30} $$

with:

– $\mathbf{S}$ the skew-symmetric matrix of the cross product associated with the vector $\mathbf{u}$ such as:

$$\mathbf{u} \times \mathbf{v} = \mathbf{S}(\mathbf{u})\mathbf{v} \tag{I.31}$$

The expression of $\mathbf{S}$ for a vector $\mathbf{u} = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix}^T$ is:

$$\mathbf{S}(\mathbf{u}) = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \tag{I.32}$$

The adoption of a matrix format for (I.29) and (I.30) allows to write the final kinematic recursive equation for the velocities:

$${}^j\mathbf{V}_j = {}^j\mathbf{T}_i \, {}^i\mathbf{V}_i + \dot{q}_j \, {}^j\mathbf{a}_j \tag{I.33}$$

with:

– ${}^j\mathbf{T}_i$ the transformation matrix between the six-component generalized velocity vectors:

$${}^j\mathbf{T}_i = \begin{pmatrix} {}^j\mathbf{R}_i & -{}^j\mathbf{R}_i \mathbf{S}({}^i\mathbf{P}_j) \\ \mathbf{0}_{3\times3} & {}^j\mathbf{R}_i \end{pmatrix} \tag{I.34}$$

These transformation matrices follow the same rule for their composition as the rotation matrices:

$${}^p\mathbf{T}_q = {}^p\mathbf{T}_k \, {}^k\mathbf{T}_q \tag{I.35}$$

– ${}^j\mathbf{a}_j$ the six-component generalized joint axis:

$${}^j\mathbf{a}_j = \begin{pmatrix} \sigma_j \, {}^j\mathbf{z}_j^T & \bar{\sigma}_j \, {}^j\mathbf{z}_j^T \end{pmatrix}^T \tag{I.36}$$

As the base has no ancestor, (I.33) is not valid for $j = 0$. ${}^0\mathbf{V}_0$ is directly expressed from the time-derivative of $\boldsymbol{\eta}_0$. By definition:

$$\dot{\boldsymbol{\eta}}_0^{trans} = {}^e\mathbf{R}_0 \, {}^0\mathbf{v}_0 \tag{I.37}$$

The angular velocity of the base depends on the time-differentiation of the Cardan angles:

$$\boldsymbol{\omega}_0 = \dot{\psi}_0 \, \mathbf{z}_e + \dot{\theta}_0 \, \mathbf{y}_{\psi_0} + \dot{\varphi}_0 \, \mathbf{x}_{\theta_0} \tag{I.38}$$

The three terms of $\boldsymbol{\omega}_0$ are written in three different frames, consequently it is necessary to express this vector is the same unique frame. Two choices are possible:

– In Σ_e [45]:

$${}^e\boldsymbol{\omega}_0 = \begin{pmatrix} 0 \\ 0 \\ \dot{\psi}_0 \end{pmatrix} + {}^e\mathbf{R}_{\psi_0} \begin{pmatrix} 0 \\ \dot{\theta}_0 \\ 0 \end{pmatrix} + {}^e\mathbf{R}_{\psi_0} \, {}^{\psi_0}\mathbf{R}_{\theta_0} \begin{pmatrix} \dot{\varphi}_0 \\ 0 \\ 0 \end{pmatrix} \tag{I.39}$$

$$= \mathbf{S}_0 \dot{\boldsymbol{\eta}}_0^{rot} \tag{I.40}$$

$\mathbf{S}_0$ denotes the transformation matrix between the intermediate Cardan frames and Σ_e:

$$\mathbf{S}_0 = \begin{pmatrix} C_{\theta_0} C_{\psi_0} & -S_{\psi_0} & 0 \\ C_{\theta_0} S_{\psi_0} & C_{\psi_0} & 0 \\ -S_{\theta_0} & 0 & 1 \end{pmatrix} \tag{I.41}$$

This quantity is not used in the Composite-Rigid-Body Algorithm but is used in the weakly nonlinear potential flow-based theory which is presented in chapter III.

– In Σ_0 [43]:

$$
\begin{aligned}
{}^0\boldsymbol{\omega}_0 &= {}^{\varphi_0}\mathbf{R}_{\theta_0}\,{}^{\theta_0}\mathbf{R}_{\psi_0}\begin{pmatrix}0\\0\\\dot{\psi}_0\end{pmatrix} + {}^{\varphi_0}\mathbf{R}_{\theta_0}\begin{pmatrix}0\\\dot{\theta}_0\\0\end{pmatrix} + \begin{pmatrix}\dot{\varphi}_0\\0\\0\end{pmatrix} && \text{(I.42)}\\
&= {}^e\boldsymbol{\Omega}_0^{-1}\,\dot{\boldsymbol{\eta}}_0^{rot} && \text{(I.43)}
\end{aligned}
$$

${}^e\boldsymbol{\Omega}_0$ notifies the transformation matrix between Σ_0 and the intermediate Cardan frames:

$$
{}^e\boldsymbol{\Omega}_0 = \begin{pmatrix}1 & S_{\varphi_0}T_{\theta_0} & C_{\varphi_0}T_{\theta_0}\\ 0 & C_{\varphi_0} & -S_{\varphi_0}\\ 0 & \dfrac{S_{\varphi_0}}{C_{\theta_0}} & \dfrac{C_{\varphi_0}}{C_{\theta_0}}\end{pmatrix} \tag{I.44}
$$

with $T_* = \tan(*)$.

${}^e\boldsymbol{\Omega}_0$ and $\mathbf{S}_0$ are linked to ${}^e\mathbf{R}_0$ by the following formula:

$$
\mathbf{S}_0\,{}^e\boldsymbol{\Omega}_0 = {}^e\mathbf{R}_0 \tag{I.45}
$$

Finally the relationship between ${}^0\mathbf{V}_0$ and $\dot{\boldsymbol{\eta}}_0$ becomes:

$$
\dot{\boldsymbol{\eta}}_0 = {}^e\mathbf{J}_0\,{}^0\mathbf{V}_0 \tag{I.46}
$$

${}^e\mathbf{J}_0$ denotes the Jacobian matrix between Σ_0 and Σ_e:

$$
{}^e\mathbf{J}_0 = \begin{pmatrix}{}^e\mathbf{R}_0 & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & {}^e\boldsymbol{\Omega}_0\end{pmatrix} \tag{I.47}
$$

The kinematic recursive equation for the accelerations is derived using the same method as for the velocities (cf. Appendix A):

$$
{}^j\dot{\mathbf{V}}_j = {}^j\mathbf{T}_i\,{}^i\dot{\mathbf{V}}_i + {}^j\boldsymbol{\gamma}_j + \ddot{q}_j\,{}^j\mathbf{a}_j \tag{I.48}
$$

with:

– ${}^j\dot{\mathbf{V}}_j$ the projection in Σ_j of the generalized acceleration vector of body j with respect to Σ_e:

$$
{}^j\mathbf{V}_j = \begin{pmatrix}{}^j\dot{\mathbf{v}}_j\\ {}^j\dot{\boldsymbol{\omega}}_j\end{pmatrix} = \begin{pmatrix}{}^j\mathbf{R}_e\,{}^e\dot{\mathbf{v}}_j\\ {}^j\mathbf{R}_e\,{}^e\dot{\boldsymbol{\omega}}_j\end{pmatrix} \tag{I.49}
$$

– ${}^j\boldsymbol{\gamma}_j$ the Coriolis and relative angular and linear accelerations:

$$
{}^j\boldsymbol{\gamma}_j = \begin{pmatrix}{}^j\mathbf{R}_i\,\mathbf{S}({}^i\boldsymbol{\omega}_i)\mathbf{S}({}^i\boldsymbol{\omega}_i)\,{}^i\mathbf{P}_j + 2\sigma_j\dot{q}_j\mathbf{S}({}^j\mathbf{R}_i\,{}^i\boldsymbol{\omega}_i)\,{}^j\mathbf{z}_j\\ \bar{\sigma}_j\dot{q}_j\mathbf{S}({}^j\mathbf{R}_i\,{}^i\boldsymbol{\omega}_i)\,{}^j\mathbf{z}_j\end{pmatrix} \tag{I.50}
$$

Regarding the linear acceleration, ${}^j\dot{\mathbf{v}}_j$ does not match the time-differentiation of ${}^j\mathbf{v}_j = {}^j\mathbf{R}_e\,{}^e\mathbf{v}_j$. A correction is required. Two results about the time-derivation of rotation matrices needs to be reminded:

$$
\left(\frac{d\,{}^e\mathbf{R}_j}{dt}\right)_{/\Sigma_e} = \mathbf{S}({}^e\boldsymbol{\omega}_j)\,{}^e\mathbf{R}_j \tag{I.51}
$$

$$
\left(\frac{d\,{}^e\mathbf{R}_j^{-1}}{dt}\right)_{/\Sigma_e} = -\,{}^e\mathbf{R}_j^{-1}\left(\frac{d\,{}^e\mathbf{R}_j}{dt}\right)_{/\Sigma_e}{}^e\mathbf{R}_j^{-1} \tag{I.52}
$$

Furthermore, for any rotation matrix $\mathbf{M}$ and any 3-component vectors $\mathbf{a}$ and $\mathbf{b}$, the cross product follows the formula:

$$(\mathbf{Ma}) \times (\mathbf{Mb}) = \mathbf{M}(\mathbf{a} \times \mathbf{b}) \tag{I.53}$$

Or using the cross product matrix $\mathbf{S}$:

$$\mathbf{S}(\mathbf{Ma})\mathbf{Mb} = \mathbf{MS}(\mathbf{a})\mathbf{b} \tag{I.54}$$

The time-differentiation of ${}^{j}\mathbf{v}_j$ leads to:

$$\left(\frac{d^{j}\mathbf{v}_j}{dt}\right)_{/\Sigma_e} = \left(\frac{d^{j}\mathbf{R}_e\, {}^{e}\mathbf{v}_j}{dt}\right)_{/\Sigma_e} \tag{I.55}$$

$$= \left(\frac{d^{e}\mathbf{R}_j^{-1}}{dt}\right)_{/\Sigma_e} {}^{e}\mathbf{v}_j + {}^{j}\mathbf{R}_e\, {}^{e}\dot{\mathbf{v}}_j \tag{I.56}$$

And using (I.51) and (I.52):

$$\tag{I.57}$$

$$\left(\frac{d^{j}\mathbf{v}_j}{dt}\right)_{/\Sigma_e} = -{}^{j}\mathbf{R}_e\, \mathbf{S}({}^{e}\boldsymbol{\omega}_j)\, {}^{e}\mathbf{R}_j\, {}^{j}\mathbf{v}_j + {}^{j}\dot{\mathbf{v}}_j \tag{I.58}$$

Making ${}^{e}\mathbf{R}_j$ appear in the cross product matrix $\mathbf{S}$ and using (I.53), it comes:

$$\left(\frac{d^{j}\mathbf{v}_j}{dt}\right)_{/\Sigma_e} = {}^{j}\dot{\mathbf{v}}_j - \mathbf{S}({}^{j}\boldsymbol{\omega}_j)\, {}^{j}\mathbf{v}_j \tag{I.59}$$

Regarding the angular acceleration:

$$\left(\frac{d^{j}\boldsymbol{\omega}_j}{dt}\right)_{/\Sigma_e} = {}^{j}\dot{\boldsymbol{\omega}}_j \tag{I.60}$$

Let us define $\boldsymbol{\nu}_j$ as equal to ${}^{j}\mathbf{V}_j$. The application of (I.59) and (I.60) to the base body, by using the six-component generalized accelerations, gives:

$$\dot{\boldsymbol{\nu}}_0 = {}^{0}\dot{\mathbf{V}}_0 - \begin{pmatrix} \mathbf{S}({}^{0}\boldsymbol{\omega}_0)\, {}^{0}\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{I.61}$$

This section established the kinematic recursive equations for the velocities (I.33) and the accelerations (I.48). These relationships are at the base of the Composite-Rigid-Body Algorithm, presented in details in the next section.

I.4 The Composite-Rigid-Body Algorithm

The Newton-Euler motion equation of the whole multibody system is:

$$\mathbf{H}(\mathbf{q}) \begin{pmatrix} {}^{0}\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{\Gamma} \end{pmatrix} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \tag{I.62}$$

with:

- $\mathbf{H} = \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{pmatrix}$ the $(6+n)\times(6+n)$ generalized mass matrix of the multibody system;
- $\mathbf{H}_{11}$ the (6×6) mass matrix of the whole multibody system seen as a single rigid body;
- $\mathbf{H}_{22}$ the $(n\times n)$ mass matrix of the multibody system when the base is fixed;
- $\mathbf{H}_{12}$ the $(6\times n)$ coupled mass matrix representing the inertial effects caused by the articulations on the base;
- $\mathbf{H}_{21} = \mathbf{H}_{12}^T$ the $(n\times 6)$ coupled mass matrix representing the inertial effects caused by the base on the articulations;
- $\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix}$ the $(6+n)$ vector of the external loads and the Coriolis and relative accelerations on the base ($\mathbf{C}_1$) and the articulations ($\mathbf{C}_2$). This vector depends neither on the base acceleration ${}^0\dot{\mathbf{V}}_0$ nor on the articular articulation $\ddot{\mathbf{q}}$.
- $\mathbf{\Gamma}$ the $(6+n)$ vector of the internal loads along or around the joint axes;
- ${}^0\dot{\mathbf{V}}_0$ the acceleration of the base with respect to the global inertial frame and expressed in the base local frame;
- $\ddot{\mathbf{q}}$ the vector of the articular accelerations.

The solving of (I.62) requires the knowledge of $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{\Gamma}$. This latter is defined by the user of the multibody solver. For instance, for a wave energy converter, $\mathbf{\Gamma}$ represents the power take-off force (PTO) of the device. Spring-damper systems are often used [61]. In that case, the internal load in a joint takes the following form:

$$\Gamma_j = -k_{PTO} q_j - B_{PTO}\dot{q}_j \tag{I.63}$$

with:

- q_j the articular variable;
- k_{PTO} the power take-off stiffness;
- B_{PTO} the power take-off damping.

The Composite-Rigid-Body Algorithm aims at computing $\mathbf{H}$ and $\mathbf{C}$. To do so, the preliminary step is the writing of the motion equation for each body. The motion equation of body j at $\mathbf{O}_j$ expressed in its local frame Σ_j is:

$${}^j\mathbf{\Phi}_j = {}^j\mathbf{F}_{Tj} \tag{I.64}$$

with:

- ${}^j\mathbf{\Phi}_j$ the time-differentiation of the linear and angular momentum;
- ${}^j\mathbf{F}_{Tj}$ the sum of external and internal loads acting on body j.

The time-differentiation of the linear and angular momenta at $\mathbf{O}_j$ projected in Σ_j is (cf. Appendix A):

$$ {}^j\mathbf{\Phi}_j = {}^j\mathbf{M}_j\,{}^j\dot{\mathbf{V}}_j + \begin{pmatrix} m_j\mathbf{S}({}^j\boldsymbol{\omega}_j)\mathbf{S}({}^j\boldsymbol{\omega}_j)\,{}^j\mathbf{S}_j \\ \mathbf{S}({}^j\boldsymbol{\omega}_j)\,{}^j\mathbf{I}_{\mathbf{O}_j}\,{}^j\boldsymbol{\omega}_j \end{pmatrix} \tag{I.65} $$

with:

- ${}^j\mathbf{M}_j$ the (6×6) mass matrix of body j expressed in Σ_j:

$$ {}^j\mathbf{M}_j = \begin{pmatrix} m_j\mathbf{I}_3 & -m_j\mathbf{S}({}^j\mathbf{S}_j) \\ m_j\mathbf{S}({}^j\mathbf{S}_j) & {}^j\mathbf{I}_{\mathbf{O}_j} \end{pmatrix} \tag{I.66} $$

- m_j the mass of body j;
- ${}^j\mathbf{I}_{\mathbf{O}_j}$ the (3×3) inertia matrix of body j reduced at $\mathbf{O}_j$ with respect to Σ_j;
- ${}^j\mathbf{S}_j$ the position of the center of gravity of body j with respect to Σ_j. In other words, if ${}^j\mathbf{O}_j\mathbf{G}_j$ denotes the position of the center of gravity of body j with respect to the origin of Σ_j and expressed in Σ_j, then:

$$ {}^j\mathbf{S}_j = {}^j\mathbf{O}_j\mathbf{G}_j \tag{I.67} $$

- $\mathbf{I}_3$ the (3×3) identity matrix.

The loads applied to the body j are:

- The external loads at $\mathbf{O}_j$ expressed in Σ_j such as weight, hydrodynamic loads, aerodynamic loads, mooring loads, etc.: ${}^j\mathbf{F}_{ej}$;
- The internal loads due to the unique ancestor, at $\mathbf{O}_j$ and expressed in Σ_j: ${}^j\mathbf{F}_j$;
- The internal loads ${}^j\mathbf{F}_k$ due to the successor bodies k such as $a_k = j$, at $\mathbf{O}_j$ and expressed in Σ_j: $-\sum_{k/a_k=j} {}^j\mathbf{F}_k$.

After the transport of ${}^j\mathbf{F}_k$ to $\mathbf{O}_k$ and in Σ_k, it yields:

$$ {}^j\mathbf{F}_k = {}^k\mathbf{T}_j^T\,{}^k\mathbf{F}_k \tag{I.68} $$

Finally:

$$ {}^j\mathbf{F}_{Tj} = {}^j\mathbf{F}_{ej} + {}^j\mathbf{F}_j - \sum_{k/a_k=j} {}^k\mathbf{T}_j^T\,{}^k\mathbf{F}_k \tag{I.69} $$

Hence, (I.64) becomes:

$$ {}^j\mathbf{M}_j\,{}^j\dot{\mathbf{V}}_j + \begin{pmatrix} m_j\mathbf{S}({}^j\boldsymbol{\omega}_j)\mathbf{S}({}^j\boldsymbol{\omega}_j)\,{}^j\mathbf{S}_j \\ \mathbf{S}({}^j\boldsymbol{\omega}_j)\,{}^j\mathbf{I}_{\mathbf{O}_j}\,{}^j\boldsymbol{\omega}_j \end{pmatrix} = {}^j\mathbf{F}_{ej} + {}^j\mathbf{F}_j - \sum_{k/a_k=j} {}^k\mathbf{T}_j^T\,{}^k\mathbf{F}_k \tag{I.70} $$

The weight is an example of external loads applied to each body. Its expression in the local frame Σ_j is:

$$ {}^j\mathbf{F}_{ej}^{Weight} = -\begin{pmatrix} m_j g\,{}^j\mathbf{R}_e\,\mathbf{z}_e \\ m_j g\mathbf{S}({}^j\mathbf{S}_j)\,{}^j\mathbf{R}_e\,\mathbf{z}_e \end{pmatrix} \tag{I.71} $$

where g is the gravity constant.

The Coriolis and relative accelerations along with the external loads are grouped together to give the generalized loads acting on body j:

$$ {}^{j}\boldsymbol{\beta}_j = -{}^{j}\mathbf{F}_{ej} + \begin{pmatrix} m_j \mathbf{S}({}^{j}\boldsymbol{\omega}_j)\mathbf{S}({}^{j}\boldsymbol{\omega}_j)\,{}^{j}\mathbf{S}_j \\ \mathbf{S}({}^{j}\boldsymbol{\omega}_j)\,{}^{j}\mathbf{I}_{\mathbf{O}_j}\,{}^{j}\boldsymbol{\omega}_j \end{pmatrix} \tag{I.72} $$

Finally, Equation I.70 is simplified using (I.72):

$$ {}^{j}\mathbf{M}_j\,{}^{j}\dot{\mathbf{V}}_j = {}^{j}\mathbf{F}_j - \sum_{k/a_k=j} {}^{k}\mathbf{T}_j^T\,{}^{k}\mathbf{F}_k - {}^{j}\boldsymbol{\beta}_j \tag{I.73} $$

The Composite-Rigid-Body Algorithm unfolds in three stages:

- A forward loop through the bodies $\forall j \in [\![1\ ;\ n]\!]$;
- A backward loop through the bodies $\forall j \in [\![n\ ;\ 0]\!]$;
- The construction of the matrix $\mathbf{H}$ and the vector $\mathbf{C}$.

The state vector of this algorithm is:

$$ \mathbf{Y}_{CRBA} = \begin{pmatrix} \boldsymbol{\eta}_0 \\ \boldsymbol{\nu}_0 \\ \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} \tag{I.74} $$

I.4.1 The forward loop

The aim of this loop is the computation of all quantities which depend neither on the articular accelerations nor on the base acceleration nor on the internal loads. That is to say, $\forall j \in [\![1\ ;\ n]\!]$:

- $i = a_j$;
- ${}^{i}\mathbf{R}_j$ from (I.10);
- ${}^{i}\mathbf{P}_j$ from (I.13);
- ${}^{j}\mathbf{T}_i$ from (I.34);
- ${}^{j}\mathbf{V}_j$ from (I.33);
- ${}^{j}\boldsymbol{\gamma}_j$ from (I.50);
- ${}^{j}\boldsymbol{\beta}_j$ from (I.72).

The initialization of this loop is achieved using the state vector (I.74) to calculate:

- ${}^{e}\mathbf{R}_0$ from (I.20);
- $\dot{\boldsymbol{\eta}}_0$ from (I.46).

I.4.2 The backward loop

The aim of this loop is the computation of the (6×6) mass matrix and the generalized loads of the composite bodies. The composite body j is composed of the body j and all its successors until the terminal bodies of the branch, whose the branch-root body is body j. For instance, in Figure I.1b, the composite body 1 is made of the bodies 1, 3 and 4. By definition, a composite body has no successor. The composite body of a terminal body is the terminal body itself. The composite body of the base is the whole multibody system. The use of composite bodies justifies the name of this direct dynamics algorithm.

The motion equation of the composite body j, using (I.73), is:

$$ {}^{j}\mathbf{M}_j^c \, {}^{j}\dot{\mathbf{V}}_j = {}^{j}\mathbf{F}_j - {}^{j}\boldsymbol{\beta}_j^c \tag{I.75} $$

with:

- ${}^{j}\mathbf{M}_j^c$ the (6×6) mass matrix of the composite body j expressed in Σ_j;
- ${}^{j}\boldsymbol{\beta}_j^c$ the generalized loads (external and inertial) acting on the composite body j and expressed in Σ_j;
- ${}^{j}\mathbf{F}_j$ the internal loads due to the unique ancestor of the composite body j, at $\mathbf{O}_j$ and expressed in Σ_j.

Applying (I.73) to the ancestor of body j (the body i such as $i = a_j$) and by introducing (I.75) and then (I.48), it comes:

$$ {}^{i}\mathbf{M}_i \, {}^{i}\dot{\mathbf{V}}_i = {}^{i}\mathbf{F}_i - \sum_{k/a_k=i} {}^{k}\mathbf{T}_i^T \left[{}^{k}\mathbf{M}_k^c \left({}^{k}\mathbf{T}_i \, {}^{i}\dot{\mathbf{V}}_i + {}^{k}\boldsymbol{\gamma}_k + \ddot{q}_k \, {}^{k}\mathbf{a}_k \right) + {}^{k}\boldsymbol{\beta}_k^c \right] - {}^{i}\boldsymbol{\beta}_i \tag{I.76} $$

After the identification of the terms of (I.76) with (I.75), the following recursive equations arise:

$$ {}^{j}\mathbf{M}_j^c = {}^{j}\mathbf{M}_j + \sum_{k/a_k=j} {}^{k}\mathbf{T}_j^T \, {}^{k}\mathbf{M}_k^c \, {}^{k}\mathbf{T}_j \tag{I.77} $$

$$ {}^{j}\boldsymbol{\beta}_j^c = {}^{j}\boldsymbol{\beta}_j + \sum_{k/a_k=j} {}^{k}\mathbf{T}_j^T \left[{}^{k}\mathbf{M}_k^c \left({}^{k}\boldsymbol{\gamma}_k + \ddot{q}_k \, {}^{k}\mathbf{a}_k \right) + {}^{k}\boldsymbol{\beta}_k^c \right] \tag{I.78} $$

I.4.3 Construction of H and C

The first row of (I.62) is:

$$ \mathbf{H}_{11} \, {}^{0}\dot{\mathbf{V}}_0 + \mathbf{H}_{12}\ddot{\mathbf{q}} + \mathbf{C}_1 = \mathbf{0}_{6\times 1} \tag{I.79} $$

By definition, the base has no ancestor therefore ${}^{0}\mathbf{F}_0 = \mathbf{0}_{6\times 1}$. Thus, (I.75) for $j = 0$ becomes:

$$ {}^{0}\mathbf{M}_0^c \, {}^{0}\dot{\mathbf{V}}_0 = - {}^{0}\boldsymbol{\beta}_0^c \tag{I.80} $$

${}^{0}\boldsymbol{\beta}_0^c$ is obtained by developing recursively ${}^{j}\boldsymbol{\beta}_j^c$ from n to 0 using (I.78):

$$ {}^{0}\boldsymbol{\beta}_0^c = {}^{0}\boldsymbol{\beta}_0 + \sum_{k=1}^{n} {}^{k}\mathbf{T}_0^T \, {}^{k}\mathbf{M}_k^c \, {}^{k}\mathbf{a}_k \, \ddot{q}_k + \sum_{k=1}^{n} {}^{k}\mathbf{T}_0^T \left({}^{k}\boldsymbol{\beta}_k + {}^{k}\mathbf{M}_k^c \, {}^{k}\boldsymbol{\gamma}_k \right) \tag{I.81} $$

The identification of the terms of (I.80) and (I.81) with (I.79) gives [62]:

$$\mathbf{H}_{11} = {}^{0}\mathbf{M}_{0}^{c} \tag{I.82}$$

$$col_q(\mathbf{H}_{12}) = {}^{q}\mathbf{T}_{0}^{T}\,{}^{q}\mathbf{M}_{q}^{c}\,{}^{q}\mathbf{a}_{q} \quad \forall q \in [\![1\ ;\ n]\!] \tag{I.83}$$

$$\mathbf{C}_1 = {}^{0}\boldsymbol{\beta}_0 + \sum_{k=1}^{n} {}^{k}\mathbf{T}_{0}^{T}\left({}^{k}\boldsymbol{\beta}_k + {}^{k}\mathbf{M}_{k}^{c}\,{}^{k}\boldsymbol{\gamma}_k\right) \tag{I.84}$$

$col_q(\mathbf{H}_{12})$ denotes the q-th column of the matrix $\mathbf{H}_{12}$.

The second row of (I.62) is:

$$\mathbf{H}_{21}\,{}^{0}\dot{\mathbf{V}}_0 + \mathbf{H}_{22}\ddot{\mathbf{q}} = \boldsymbol{\Gamma} - \mathbf{C}_2 \tag{I.85}$$

Γ_j denotes the j-th component of $\boldsymbol{\Gamma}$ and by definition of $\boldsymbol{\Gamma}$:

$$\Gamma_j = {}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\mathbf{F}_j \tag{I.86}$$

The left-multiplication of (I.75) by ${}^{j}\mathbf{a}_j$ allows to write:

$${}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\mathbf{M}_{j}^{c}\,{}^{j}\dot{\mathbf{V}}_j = \Gamma_j - {}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\boldsymbol{\beta}_{j}^{c} \tag{I.87}$$

Let $\beta(j)$ be the set of the bodies on the direct branch between the base (not included) and the body j (included). The iterative introduction of (I.48) in (I.87) leads to:

$${}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\mathbf{M}_{j}^{c}\,{}^{j}\mathbf{T}_0\,{}^{0}\dot{\mathbf{V}}_0 + {}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\mathbf{M}_{j}^{c}\sum_{k\in\beta(j)} {}^{j}\mathbf{T}_k\,{}^{k}\mathbf{a}_k\,\ddot{q}_k = \Gamma_j - {}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\mathbf{M}_{j}^{c}\sum_{k\in\beta(j)} {}^{j}\mathbf{T}_k\,{}^{k}\boldsymbol{\gamma}_k - {}^{j}\mathbf{a}_{j}^{T}\,{}^{j}\boldsymbol{\beta}_{j}^{c} \tag{I.88}$$

Equation I.85 involves that if ${}^{0}\dot{\mathbf{V}}_0 = \mathbf{0}_{6\times1}$ and $\ddot{\mathbf{q}} = \mathbf{0}_{n\times1}$ then $\boldsymbol{\Gamma} = \mathbf{C}_2$. By denoting ${}^{j}\tilde{\boldsymbol{\beta}}_{j}^{c} = {}^{j}\boldsymbol{\beta}_{j}^{c}\Big|_{\substack{{}^{0}\dot{\mathbf{V}}_0=\mathbf{0}_{6\times1}\\ \ddot{\mathbf{q}}=\mathbf{0}_{n\times1}}}$, (I.78) becomes:

$${}^{j}\tilde{\boldsymbol{\beta}}_{j}^{c} = {}^{j}\boldsymbol{\beta}_j + \sum_{k/a_k=j} {}^{k}\mathbf{T}_{j}^{T}\left[{}^{k}\mathbf{M}_{k}^{c}\,{}^{k}\boldsymbol{\gamma}_k + {}^{k}\tilde{\boldsymbol{\beta}}_{k}^{c}\right] \tag{I.89}$$

The identification of the terms of (I.88) and (I.89) with (I.85) gives [62]:

$$row_p(col_q(\mathbf{H}_{22})) = {}^{p}\mathbf{a}_{p}^{T}\,{}^{p}\mathbf{M}_{p}^{c}\,{}^{p}\mathbf{T}_q\,{}^{q}\mathbf{a}_q \quad \forall (p,q) \in [\![1\ ;\ n]\!]^2 \text{ such as } p \geqslant q \tag{I.90}$$

The upper triangle of $\mathbf{H}$ is filled by symmetry.

$$\mathbf{H}_{21} = \mathbf{H}_{12}^{T} \tag{I.91}$$

$$row_p(\mathbf{C}_2) = {}^{p}\mathbf{a}_{p}^{T}\left[{}^{p}\mathbf{M}_{p}^{c}\sum_{k\in\beta(p)} {}^{p}\mathbf{T}_k\,{}^{k}\boldsymbol{\gamma}_k + {}^{p}\tilde{\boldsymbol{\beta}}_{p}^{c}\right] \tag{I.92}$$

$row_p(\mathbf{H}_{22})$ denotes the p-th row of the matrix $\mathbf{H}_{22}$.

Eventually, the backward loop may be summed up in the following way: $\forall j \in [\![n\ ;\ 0]\!]$:

- $i = a_j$;
- ${}^{i}\tilde{\boldsymbol{\beta}}_{i}^{c}$ is computed from (I.89);

– ${}^{i}\mathbf{M}_{i}^{c}$ is computed from (I.77).

The computation of $\mathbf{H}$ is achieved using:

– (I.82) for $\mathbf{H}_{11}$;

– (I.83) for $\mathbf{H}_{12}$;

– (I.91) for $\mathbf{H}_{21}$;

– (I.90) for $\mathbf{H}_{22}$.

The calculation of $\mathbf{C}$ is based on:

– (I.84) for $\mathbf{C}_1$;

– (I.92) for $\mathbf{C}_2$.

Once the motion equation of the whole multibody system is formed, it remains to solve this linear system of size $(6+n)$:

$$\begin{pmatrix} {}^{0}\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} = \mathbf{H}^{-1}\left[\begin{pmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{\Gamma} \end{pmatrix} - \mathbf{C}\right] \tag{I.93}$$

A LU decomposition method is used.

Degrees of freedom can be blocked by removing columns and rows from $\mathbf{H}$, $\mathbf{C}$ and $\boldsymbol{\Gamma}$ in order to delete the interactions between the blocked and active *dof*. Hence, the corresponding components of ${}^{0}\dot{\mathbf{V}}_0$ and/or $\ddot{\mathbf{q}}$ are zeroed. Much attention is necessary in case of base-*dof* blocking because ${}^{0}\dot{\mathbf{V}}_0 \neq \ddot{\boldsymbol{\eta}}_0$. For example, if the fourth *dof* of the base is blocked then ${}^{0}\dot{\boldsymbol{\omega}}_0(1) = 0$ which is not always equivalent to $\ddot{\varphi}_0 = 0$.

I.5 Time integration

The final set of governing equations leads to an ordinary differential equation system such as $\dot{\mathbf{Y}} = f(\mathbf{Y}, t)$. The time-differentiation of the state vector (I.74) is obtained using:

– (I.46) for $\dot{\boldsymbol{\eta}}_0$;

– (I.93) and (I.61) for $\dot{\boldsymbol{\nu}}_0$;

– (I.93) for $\ddot{\mathbf{q}}$

In *InWave*, the state vector is time-stepped using either a fourth-order explicit Runge-Kutta (*RK4*) scheme or an adaptive Adams-Moulton scheme. Only the *RK4* scheme will be used in this work. We remind the classical time-step equation of the *RK4* scheme:

$$\mathbf{Y}(t+dt) = \mathbf{Y}(t) + \frac{dt}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{I.94}$$

with:

– dt the fixed time step of the *RK4* scheme;

- $\mathbf{k}_1 = f(\mathbf{Y}, t)$;
- $\mathbf{k}_2 = f(\mathbf{Y} + \frac{dt}{2}\mathbf{k}_1, t + \frac{dt}{2})$;
- $\mathbf{k}_3 = f(\mathbf{Y} + \frac{dt}{2}\mathbf{k}_2, t + \frac{dt}{2})$;
- $\mathbf{k}_4 = f(\mathbf{Y} + dt\mathbf{k}_3, t + dt)$.

I.6 Conclusion

This chapter presented a review of the general concepts used for multibody dynamics. Several multibody modelings and direct dynamics algorithms were exhibited. In this PhD thesis, bodies are modeled using the modified Denavit-Hartenberg parameters. The details of the Composite-Rigid-Body Algorithm were also given and demonstrated. This algorithm is based on few hypotheses: multibody systems are limited to kinematic trees of rigid bodies linked by one-*dof* joints. The *CRBA* unfolds in three steps which lead to the construction and the solving of the motion equation for the whole multibody system. The numerical tool *InWave* relies on this algorithm and is used in this PhD to perform time-domain simulations of marine operations. The next chapter focus on cable dynamics, presents the application of the *CRBA* to model cables and its comparison to a classical cable dynamic theory.

Chapter II

Cable Dynamics

Contents

The simulation of marine operations involves the use of cables and thus solving the cable dynamics. This chapter introduces a state of the art of cable dynamics. Two models are presented in details: the low-order lumped mass model and the multibody model. This latter is based on the Composite-Rigid-Body Algorithm presented in the last chapter. A validation of these theories is exposed for each model.

This chapter also presents a state of the art of the winch modeling to wind or unwind a cable. The chosen winch approach is detailed and an example of its use is given.

II.1 State of the art

Cables are extensively used in offshore engineering. Mooring lines maintain the floating bodies on station by preventing their drift motion. Mooring cables are also the key point of the stability of tension-leg platforms for instance. Power cables convey the energy production of a marine renewable energy device to the fixed power-grid connections resting on the seabed [63]. A wide range of marine operations require the use of cables. For example, towing operations need towlines and lowering / lifting operations use hoisting lines. This is why it is necessary to compute the cable dynamics to achieve a marine operation simulation.

Cables are subject to three types of deformation (Figure II.1):

- An axial deformation: the stretching;
- A curvature deformation: the bending;
- A twisting deformation: the torsion.

If the cable is not considered as rigid, the stretching is of first importance. Bending and torsion effects are significant in case of low-tension cables where internal loads dominate the cable dynamics [64]. But, the modeling of the bending and torsion loads increase markedly the complexity of the numerical models.

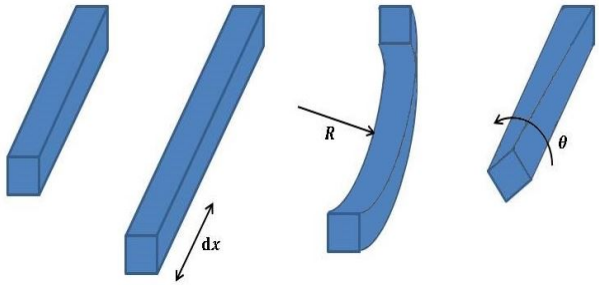


Figure II.1 – Deformations of a cable element: stretching of length dx, bending with a radius of curvature R and torsion by an angle θ

A physical effect must be avoided in any cable: snap loads. Snap or snatch load is a spike of very short duration in the cable tension caused when a line becomes slack and then suddenly taut [65]. The amplitude of these loads is much higher than the maximum of the dynamic loads. Snapping phenomenon produces shocks in the line

material and reduces the fatigue life [66]. *DNV* defines a criterion, in case of a lowering operation, to avoid snap loads using a 10 % margin for safety [8]:

$$F_{hyd} \leqslant 0.9 F_{static} \tag{II.1}$$

with:

- F_{hyd} the characteristic hydrodynamic force, from (4);
- F_{static} the static weight of the payload (difference between the weight and the buoyancy force).

II.1.1 Mathematical modellings

Six different mathematical models exist to evaluate cable loads:

- Quasi-static approach;
- Force-displacement-velocity model;
- Lumped mass models;
- Finite-element method;
- Finite-difference method;
- Multibody approach.

II.1.1.1 Quasi-static approach

The quasi-static theory computes the line position and the tension from the catenary equation for a continuous slack cable. Cable dynamics, internal axial damping loads and hydrodynamic loads are omitted. This model assumes each cable element is only subject to the internal axial tension, the weight and the buoyancy. The line is permanently in static equilibrium during the simulation and the tension only depends on the position of its two end points (the fairlead for the extremity attached to the vessel and the anchor for the other extremity, connected to the seabed). This approach is widely used to describe a mooring system due to its quickness and its ease of implementation [67]. But the important assumptions lead to large possible discrepancies [65]. Nevertheless this method can also be used to initialize the position of a slack cable in order to perform dynamic computations [65]. The initialization of the position of a taut cable is figured out by assuming a straight line between the extremities of the cable.

The motion equation for a cable element in quasi-static equilibrium, in the plane defined by these two end points (Figure II.2), gives:

- Along the normal axis of the cable element:

$$T d\phi = (W - B)\cos(\phi) + D_n \tag{II.2}$$

with:

- T the tension;
- W the weight;

- B the buoyancy;
- D_n the normal drag force;
- ϕ the angle between the cable element and the horizontal axis;
- $d\phi$ a small variation of ϕ.

– Along the tangential axis of the cable element:

$$dT = (W - B)\sin(\phi) - D_t \tag{II.3}$$

with:

- D_t the tangential drag force;
- dT a small variation of the tension T.

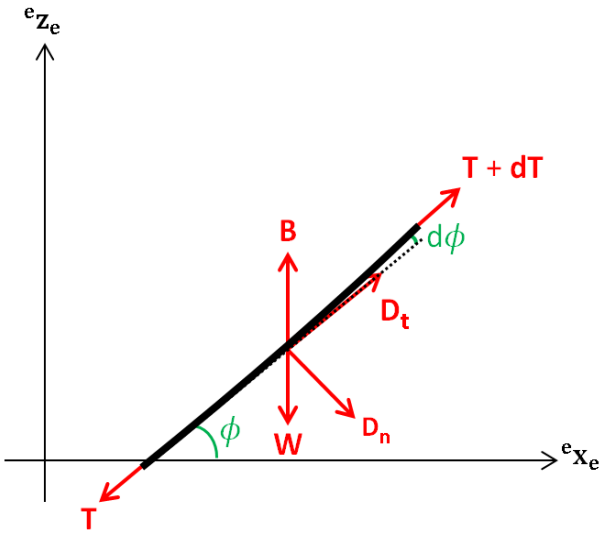


Figure II.2 – Loads on a cable element in quasi-static equilibrium

The quasi-static hypothesis assumes gravity forces (weight and buoyancy) are predominant on drag loads, leading to:

$$Td\phi = (W - B)\cos(\phi) \tag{II.4}$$
$$dT = (W - B)\sin(\phi) \tag{II.5}$$

These equations implies that the horizontal tension is constant throughout the line and the vertical tension follows the formula [67]:

$$V = \mu s + V_A \tag{II.6}$$

with:

– V the vertical tension;

– μ the linear weight such as $\mu = (\rho - \rho_{water})Ag$;

– ρ the cable density;

– ρ_{water} the water density;

- A the cross sectional area of the cable;
- V_A the vertical tension at the anchor;
- s the curvilinear coordinate along the cable such as $s = 0$ at the anchor and $s = L_u$ at the fairlead;
- L_u the unstretched length of the cable.

The tension at any point of the cable is given by:

$$T(s) = \sqrt{H^2 + (\mu s + V_A)^2} \tag{II.7}$$

where H denotes the (constant) horizontal tension in the cable.

The cable position is described by the catenary equation [67]:

$$\begin{cases} x(s) &= \dfrac{H}{\mu}\left[sinh^{-1}\left(\dfrac{\mu s + V_A}{H}\right) - sinh^{-1}\left(\dfrac{V_A}{H}\right)\right] + \dfrac{Hs}{EA} \\ z(s) &= \dfrac{H}{\mu}\left[\sqrt{1 + \left(\dfrac{\mu s + V_A}{H}\right)^2} - \sqrt{1 + \left(\dfrac{V_A}{H}\right)^2}\right] + \dfrac{1}{EA}\left(\dfrac{\mu s^2}{2} + V_A s\right) \end{cases} \tag{II.8}$$

E is the cable Young's modulus.

Equation II.8 only depends on the horizontal and vertical tensions at the anchor (or at the fairlead). Their computation is achieved by using a Newton-Raphson algorithm to zero the vector $\mathbf{F}$, defined by:

$$\mathbf{F}(H, V_A) = \begin{pmatrix} x(L_u) - x_{input}(L_u) \\ z(L_u) - z_{input}(L_u) \end{pmatrix} \tag{II.9}$$

$x_{input}(L_u)$ and $z_{input}(L_u)$ are the horizontal and vertical displacements between the two end points of the cable.

II.1.1.2 Force-displacement-velocity model

Another simple approach is the use of a force-displacement-velocity model based on cable stiffness and damping matrices [68]:

$$\mathbf{F}^{cable} = \mathbf{F}_0 - \mathbf{K}(\mathbf{x})\mathbf{x} - \mathbf{C}(\mathbf{x})\dot{\mathbf{x}} \tag{II.10}$$

with:

- $\mathbf{F}^{cable}$ the cable loads;
- $\mathbf{F}_0$ a constant vertical force representing the mooring system weight, the cable pretension, etc.;
- $\mathbf{K}$ the mooring stiffness matrix;
- $\mathbf{C}$ the mooring damping matrix
- $\mathbf{x}$ the body position.

If $\mathbf{K}$ and $\mathbf{C}$ are constant, the model is linear, otherwise it is nonlinear.

The cable loads computed in this way only represent the global mooring system. There is no representation of the individual lines. But this approach is easy to implement and leads to fast computations as soon as the two matrices $\mathbf{K}$ and $\mathbf{C}$ are known. Slack cables cannot adequately be simulated like this, except in case of small amplitude motions. An example of the use of this forced-displacement-velocity model can be found in [69].

The four other approaches are dynamic models. Inertia effects and hydrodynamic loads are taken into account. Thus, a dynamic model is more accurate for predicting the loads in a cable.

II.1.1.3 Lumped mass models

In the lumped mass theory, the cable is discretized into $N_{Elements}$ cable elements and $N_{Elements} + 1$ nodes where the mass is concentrated (or lumped) (Figure II.3). Nodes are connected by spring-damper systems and located with respect to a global inertial frame. Then, the motion equation is written at the position of the nodes [70]:

$$\forall i \in [\![0\ ;\ N]\!]\ (\mathbf{M}_i + \mathbf{A}_i)\ddot{\mathbf{r}}_i = \mathbf{W}_i + \mathbf{B}_i + \mathbf{T}_i + \mathbf{C}_i + \mathbf{D}_i \tag{II.11}$$

with, at the node i:

- $\mathbf{M}_i$ the 3×3 mass matrix;
- $\mathbf{A}_i$ the 3×3 added-mass matrix;
- $\mathbf{W}_i$ the weight;
- $\mathbf{B}_i$ the buoyancy force;
- $\mathbf{T}_i$ the axial tension due to the adjacent nodes;
- $\mathbf{C}_i$ the axial damping loads due to the adjacent nodes;
- $\mathbf{D}_i$ the hydrodynamic loads;
- $\mathbf{r}_i$ the position vector.

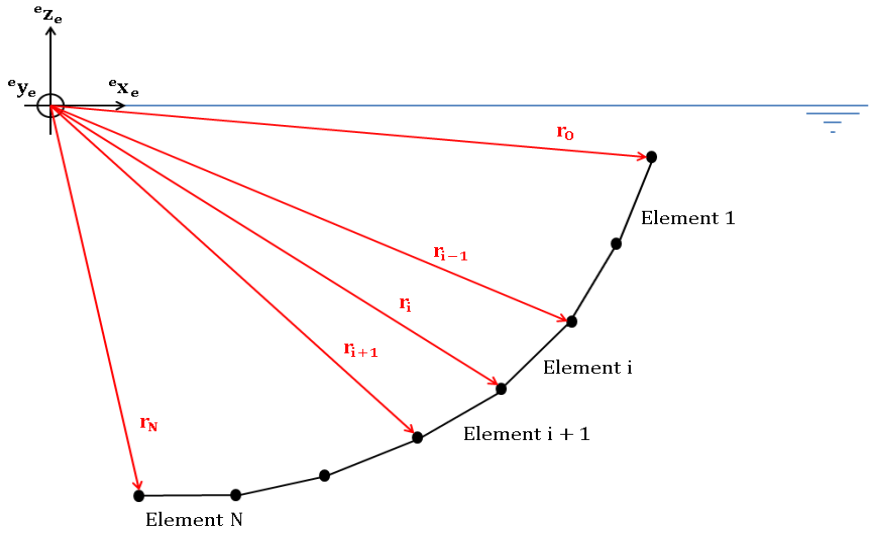


Figure II.3 – Spatial discretization of a cable. The position vector of the nodes are in red. The nodes are the black points and the cable elements the black lines.

The main advantage of the lumped mass theory lies in the strictly diagonal total mass matrix (including mass and added-mass matrices). No matrix inversion is required in this approach. By doing that, the coupling terms of the total mass matrix are neglected. Cases where lines are connected between them, such as nets, may lead to a loss of accuracy [71]. Equation II.11 does not include either bending or torsion effects, deriving a low-order lumped mass formulation. Buckham [64] developed a high-order lumped mass formulation including bending and torsion effects. Bending and torsion loads are computed based on the continuous cable equation (II.12) and a finite-element model which approximates the cable by a cubic spline. The total mass matrix is maintained strictly diagonal.

$$\begin{cases} \mathbf{M}\ddot{\mathbf{r}} & = \mathbf{F}' + \mathbf{h} + \mathbf{w} \\ \mathbf{J}\dot{\boldsymbol{\omega}} & = \mathbf{N}' + \mathbf{r}' \times \mathbf{F} + \mathbf{n} \end{cases} \tag{II.12}$$

with:

- $\mathbf{M}$ the total mass matrix per cable unit length;
- $\mathbf{r}$ the position vector;
- $\mathbf{F}$ the internal forces per cable unit length;
- $\mathbf{h}$ the hydrodynamic loads per cable unit length;
- $\mathbf{w}$ the weight and buoyancy per cable unit length;
- $\mathbf{J}$ the inertia matrix per cable unit length;
- $\boldsymbol{\omega}$ the angular velocity;
- $\mathbf{N}$ the internal moments per cable unit length;
- $\mathbf{n}$ the external moments per cable unit length;
- $(\,)'$ the spatial differentiation.

The lumped mass theory is widespread and several examples of its use can be found in the literature [70, 72, 73]. The low-order lumped mass theory is detailed in section II.2.

II.1.1.4 Finite-element method

The finite-element method uses the same cable discretization as the lumped mass theory but using the integral form of (II.12). The motion equation is derived from the virtual work principle [74]. The mass distribution is considered as continuous compared to the lumped mass approach where the mass is discretized at the position of the nodes. It results in a total mass matrix with off-diagonal coefficients expressing the inertial coupling terms. The finite-element method is known to require less cable elements compared to the lumped mass approach but requires a complex mathematical formulation and larger computational time [65].

II.1.1.5 Finite-difference method

The lumped mass and finite-element formulation only discretize the cable in space. The time-stepping is achieved by an external integrator scheme. Time derivatives are considered as continuous. The finite-difference method discretizes (II.12) in both space and time. The time-step dt becomes inherent to the discretization as the cable discretization in nodes was inherent in the previous mathematical modellings. Thus, a time-dependent variable x may be discretized, for instance, using a forward, backward or centred scheme [65]:

$$\frac{\partial x}{\partial t} \approx \frac{x_{i+1} - x_i}{dt} \approx \frac{x_i - x_{i-1}}{dt} \approx \frac{x_{i+1} - x_{i-1}}{2dt} \tag{II.13}$$

This approach is easier to implement than a finite-element theory but is prone to numerical difficulties [71]. Details on the discretization of the motion equations can be found in [75, 76].

II.1.1.6 Multibody approach

A multibody approach, based on the Augmented Formulation presented in section I.1, is another way to model cables. It uses constraint equations to define a cable joint. Servin and Lacoursière [77] use the following constraint equation to simulate rigid body cables for virtual environments (visual simulations):

$$\boldsymbol{\phi} = \begin{pmatrix} \mathbf{p}_{i+1} - \mathbf{p}_i \\ \theta_i \\ \Omega_i \end{pmatrix} = \mathbf{0}_{5\times 1} \tag{II.14}$$

Where $\mathbf{p}_i$, θ_i and Ω_i denote the position of the body i, its bend and twist angles, respectively. The stretching (elasticity), bending and torsion coefficients are introduced in the system through constraint regularization [77, 78]. This approach was also used by Ham et al. [27] to model a floating crane with a heavy load suspended with a constraint-based wire rope.

II.1.2 Time integration and numerical damping

Regarding the numerical integration of cable dynamics, several schemes are possible. The fourth-order Runge-Kutta scheme, presented in section I.5, remains a popular

technique. This time-stepper is robust but requires a small time step due to the stiffness of the system of equations [64]. That is why another approach is often chosen: the use of an implicit time integrator which includes numerical damping to attenuate high-frequency waves in cables and more generally in structural dynamics. Most of them are based on the Newmark's scheme [79]. It integrates numerically second-order ordinary differential equations whereas a *RK4* scheme deals with first-order ordinary differential equations. Let us consider the displacement $\mathbf{u}_n$, the velocity $\dot{\mathbf{u}}_n$ and the acceleration $\ddot{\mathbf{u}}_n$ at the time t_n, the second-order *ODE* to solve is:

$$\mathbf{M}\ddot{\mathbf{u}}_n = \mathbf{F}(t_n, \mathbf{u}_n, \dot{\mathbf{u}}_n) \tag{II.15}$$

Integration algorithms are compared in terms of accuracy, stability and numerical dissipation. Accuracy depends on the local truncation error to compute the time-dependent variables. An algorithm is accurate to n-order if the error accumulated is $\mathcal{O}(dt^n)$. The stability is the property of an algorithm, under any initial conditions, to limit the numerical error during a simulation. An algorithm is said *unconditionally stable* if the convergence of the solution is independent of the time step dt, otherwise it is *conditionally* stable for a range of values of dt [80]. As all explicit methods, the *RK4* scheme is conditionally stable. The stability relies on the spectral radius of the numerical amplification matrix, $\mathbf{A}$, defined by:

$$\begin{pmatrix} \mathbf{u}_{n+1} \\ \dot{\mathbf{u}}_{n+1} \\ \ddot{\mathbf{u}}_{n+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{u}_n \\ \dot{\mathbf{u}}_n \\ \ddot{\mathbf{u}}_n \end{pmatrix} \tag{II.16}$$

The spectral radius of $\mathbf{A}$, notified $\rho(\mathbf{A})$, is the largest absolute value of its eigenvalues. To ensure the stability of an algorithm, it must satisfy [80]:

$$\rho(\mathbf{A}) \leqslant 1 \tag{II.17}$$

The numerical dissipation is the capacity of an algorithm to eliminate spurious high-frequency oscillations. The spectral radius of the amplification matrix also measures the dissipation. If $\rho(\mathbf{A}) = 1$ then no dissipation is present but if $\rho(\mathbf{A})$ is lower, the numerical damping increases.

The Newmark's scheme time-steps the displacement and the velocity by the following equations [79]:

$$\begin{cases} \mathbf{u}_{n+1} &= \mathbf{u}_n + dt\dot{\mathbf{u}}_n + \left(\frac{1}{2} - \beta\right) dt^2 \ddot{\mathbf{u}}_n + \beta dt^2 \ddot{\mathbf{u}}_{n+1} \\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + (1-\gamma) dt \ddot{\mathbf{u}}_n + \gamma \ddot{\mathbf{u}}_{n+1} \end{cases} \tag{II.18}$$

Where β and γ are the Newmark's parameters. This scheme is unconditionally stable if $2\beta \geqslant \gamma \geqslant \frac{1}{2}$. Numerical damping is introduced if $\gamma > \frac{1}{2}$ and $\beta \geqslant \frac{1}{4}\left(\gamma + \frac{1}{2}\right)$ but the accuracy is only of first order [81].

This scheme has been improved to reach a second-order accuracy. Hilber et al. [82] proposed to weight the internal loads between two time steps by adding a third parameter, this is the *HHT-α* scheme, whereas Wood et al. [83] weighted the inertial loads, obtaining the *WBZ* scheme. Finally, Chung and Hulbert [84] decided to weight both internal and inertial loads, leading to the *Generalized-α* method [85]. The equations

of the Newmark's scheme (II.18) are used along with a variant of the motion equation solved at each time step:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{u}}_{n+1-\alpha_M} & = \mathbf{F}(t_{n+1-\alpha_F}, \mathbf{u}_{n+1-\alpha_F}, \dot{\mathbf{u}}_{n+1-\alpha_F}) \\ (*)_{n+1-\alpha_M} & = (1-\alpha_M)(*)_{n+1} + \alpha_M (*)_n \\ (*)_{n+1-\alpha_F} & = (1-\alpha_F)(*)_{n+1} + \alpha_F (*)_n \end{cases} \tag{II.19}$$

The *Generalized-α* method includes the *HHT-α* scheme ($\alpha_M = 0$), the *WBZ* scheme ($\alpha_F = 0$) and the Newmark's scheme ($\alpha_M = \alpha_F = 0$). This method is second-order accurate if $\gamma = \frac{1}{2} - \alpha_M + \alpha_F$ and unconditionally stable if $\alpha_M \leqslant \alpha_F \leqslant \frac{1}{2}$ and $\beta \geqslant \frac{1}{4} + \frac{1}{2}(\alpha_F - \alpha_M)$. The optimal high-frequency dissipation with minimal low-frequency impact is achieved by using [84]:

$$\beta = \frac{1}{4}(1 - \alpha_M + \alpha_F)^2 \tag{II.20}$$

$$\alpha_M = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \tag{II.21}$$

$$\alpha_F = \frac{\rho_\infty}{\rho_\infty + 1} \tag{II.22}$$

where ρ_∞ is the desirable value of spectral radius at infinite frequency, leading to a controllable numerical dissipation.

The *Generalized-α* method is used, for example, by Buckham [64] with a high-order lumped mass theory and by Gobat [76] with a finite-difference approach.

II.1.3 Cable software packages

Several commercial or off-the-shelf numerical tools dedicated to cable dynamics exist. Some of them, considered as the most common, are listed in Table II.1.

Software package	Status	Cable theory	Time-stepper
MAP++ [67]	Open-source	Quasi-static	ø
MoorDyn [70]	Open-source	Low-order lumped mass	*RK2*[1]
OrcaFlex [38]	Commercial	High-order lumped mass	*Generalized-α*[2]
DeepLines [39]	Commercial	Finite elements	Newmark
ProteusDS [40]	Commercial	High-order lumped mass	*RK4*

Table II.1 – Cable theory and time-stepper in different software packages

MAP++[3] models mooring lines based on a multi-segmented quasi-static approach. It handles seabed contact and friction. It is released as an open-source library under the

[1] A constant-time-step second order Runge-Kutta integration scheme.

[2] For the implicit scheme only, an explicit integration scheme based on a semi-implicit Euler method is also available.

[3] https://map-plus-plus.readthedocs.io/en/latest

Apache License by *NREL*[4] and can be run with Python, C, C++ or Fortran programs. Another open-source software package is *MoorDyn*[5] [86]. It uses the low-order lumped mass theory and handles line interconnections, clump weights, floats and seabed friction [87]. It is released under a GPLv3 license for its C++-based version and under an Apache License for its Fortran-based version.

Regarding the commercial numerical tools, *OrcaFlex* and *ProteusDS* use the high-order lumped mass theory whereas *DeepLines* relies on a finite-element approach.

II.2 The low-order lumped mass theory

II.2.1 The need of an in-house cable solver

The simulation of lowering or lifting operations involves the computation of cable dynamics. Based on the review of the previous section, it appears that a quasi-static or force-displacement-velocity approaches are not sufficient for lowering or lifting operations as neither inertial loads nor hydrodynamic loads are evaluated. Bending and torsion effects may be significant for slack cables whereas hoisting cables are taut. They also lead to higher computational cost [70]. Consequently, a dynamic model with only the effects of the stretching seems enough in our case. Finite-element and finite-difference approaches require both complex developments and a coupling between these theories and the Composite-Rigid-Body Algorithm. This leads to two potential solutions for the problem of interest in this PhD: the lumped mass theory and the multibody approach. The former leads to simple developments if bending and torsion are omitted but the coupling between the lumped mass theory and the *CRBA* remains. The latter leads to the simulation of cables within the multibody system theory. At the beginning of this PhD work, *InWave* could not simulate cable dynamics. Therefore, the decision was taken to use both theories.

As seen in subsection II.1.3, *MoorDyn* is an open-source low-order lumped mass theory-based numerical tool which matches our specifications. This software package is available with a C++-based version and a Fortran-based version. As *InWave* is implemented in C++, the Fortran-based version involves overcoming the non-interoperability of the two programming languages. Regarding the C++-based version of *MoorDyn*, it is released under the GPLv3 license. Therefore, if this numerical tool is included in *InWave*, the total source code must be released. This is not possible for *INNOSEA* as *InWave* is their in-house numerical tool.

That is why, it was decided to develop an in-house low-order lumped mass-based cable solver. This solution allows to reach multiple goals:

- The comparison between this cable theory and the *CRBA*;
- In case of failure of the simulation of cables by the *CRBA*, this solver could be used as a third-party module;
- There is no problem of license;
- It can also be used as a dynamic mooring solver.

[4] https://www.nrel.gov
[5] http://www.matt-hall.ca/moordyn.html

II.2.2 Theoretical developments

As explained in subsubsection II.1.1.3, the cable is discretized into $N_{Elements}$ elements and $N_{Elements}+1$ nodes. In case of a mooring cable, the node 0 represents the fairlead whereas the node $N_{Elements}$ is the anchor. The motion equation at the position of the nodes and in the global inertial frame is:

$$\forall i \in [\![0\ ;\ N_{Elements}]\!]\ (\mathbf{M}_i + \mathbf{A}_i)\ddot{\mathbf{r}}_i = \mathbf{W}_i + \mathbf{B}_i + \mathbf{T}_i + \mathbf{C}_i + \mathbf{D}_i \tag{II.23}$$

The expressions of each load component can be found in [70]. For a better understanding of the physics of the low-order lumped mass theory, they are reminded below.

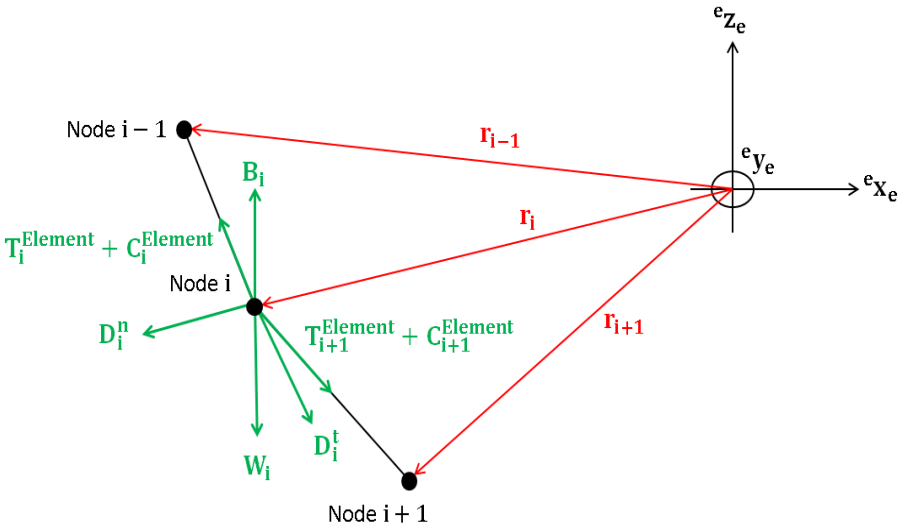


Figure II.4 – Loads at the node i

Each force at the node i is due to the two adjacent cable elements i and $i+1$. Thus, the weight $\mathbf{W}_i$ is the result of the weight of both the half element i and the half element $i+1$:

$$\mathbf{W}_i = \mathbf{W}_i^{Element} + \mathbf{W}_{i+1}^{Element} \tag{II.24}$$

with:

$$\mathbf{W}_i^{Element} = \frac{1}{2} A L_u \rho_c \mathbf{g} \tag{II.25}$$

where L_u, ρ_c and $\mathbf{g}$ represent, respectively, the unstretched length of the cable element, the cable density and the gravity acceleration vector. L_u is assumed to be constant through the cable.

The expression of the buoyancy force is given by:

$$\mathbf{B}_i = \mathbf{B}_i^{Element} + \mathbf{B}_{i+1}^{Element} \tag{II.26}$$

with:

$$\mathbf{B}_i^{Element} = -\frac{1}{2} A L_u \rho \mathbf{g} \tag{II.27}$$

where ρ is the water density.

The axial tension due to the cable element i is:

$$\mathbf{T}_i^{Element} = \begin{cases} -\dfrac{EA}{L_u}(\|\mathbf{r}_i - \mathbf{r}_{i-1}\| - L_u)\left(\dfrac{\mathbf{r}_i - \mathbf{r}_{i-1}}{\|\mathbf{r}_i - \mathbf{r}_{i-1}\|}\right) & \text{if } \|\mathbf{r}_i - \mathbf{r}_{i-1}\| \geqslant L_u \\ \mathbf{0}_{3\times 1} & \text{otherwise} \end{cases} \tag{II.28}$$

The second condition means there is no compression of the cable.

The axial tension at the node i is expressed by:

$$\mathbf{T}_i = \mathbf{T}_i^{Element} - \mathbf{T}_{i+1}^{Element} \tag{II.29}$$

The friction between the strands of the cable dissipates energy and is modeled by a linear viscous damper [64]:

$$\mathbf{C}_i^{Element} = -\frac{CA}{L_u}\left[\frac{\partial}{\partial t}(\|\mathbf{r}_i - \mathbf{r}_{i-1}\| - L_u)\right]\left(\frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{\|\mathbf{r}_i - \mathbf{r}_{i-1}\|}\right) \tag{II.30}$$

with:

$$\frac{\partial}{\partial t}(\|\mathbf{r}_i - \mathbf{r}_{i-1}\| - L_u) = \frac{1}{\|\mathbf{r}_i - \mathbf{r}_{i-1}\|}\sum_{j=1}^{3}(r_i^j - r_{i-1}^j)(\dot{r}_i^j - \dot{r}_{i-1}^j) \tag{II.31}$$

r_i^j is the j^{th} component of the position vector $\mathbf{r}_i$. C is the internal viscous damping coefficient of the cable. Finally, the axial damping force at the node i is:

$$\mathbf{C}_i = \mathbf{C}_i^{Element} - \mathbf{C}_{i+1}^{Element} \tag{II.32}$$

The drag loads are computed from Morison's equation [88] as the cable can be seen as a slender cylinder. By neglecting the wave kinematics, the relative velocity at the node i is $-\dot{\mathbf{r}}_i$. Its tangential component is $(-\dot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i$. $\mathbf{q}_i$ is the tangent unit-vector at the node i, defined by:

$$\mathbf{q}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{\|\mathbf{r}_{i+1} - \mathbf{r}_{i-1}\|} \tag{II.33}$$

The normal and tangential drag loads, $\mathbf{D}_i^n$ and $\mathbf{D}_i^t$, are:

$$\mathbf{D}_i^n = \frac{1}{2}\rho C_{dn} d L_u \|(\dot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i - \dot{\mathbf{r}}_i\|[(\dot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i - \dot{\mathbf{r}}_i] \tag{II.34}$$

$$\mathbf{D}_i^t = \frac{1}{2}\rho C_{dt} d L_u \|(-\dot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i\|(-\dot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i \tag{II.35}$$

where C_{dn}, C_{dt} and d are the normal and tangential drag coefficients and the cable diameter. Hence:

$$\mathbf{D}_i = \mathbf{D}_i^n + \mathbf{D}_i^t \tag{II.36}$$

With the same reasoning, the normal and tangential added-mass loads, $\mathbf{A}_i^n\ddot{\mathbf{r}}_i$ and $\mathbf{A}_i^t\ddot{\mathbf{r}}_i$, are:

$$\mathbf{A}_i^n\ddot{\mathbf{r}}_i = \rho C_{an} A L_u[(\ddot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i - \ddot{\mathbf{r}}_i] \tag{II.37}$$

$$\mathbf{A}_i^t\ddot{\mathbf{r}}_i = \rho C_{at} A L_u[(-\ddot{\mathbf{r}}_i \cdot \mathbf{q}_i)\mathbf{q}_i] \tag{II.38}$$

where C_{an} and C_{at} are the normal and tangential added-mass coefficients. The tangential added-mass force may be used, for example, in case of chains [70]. Finally, it comes:

$$\mathbf{A}_i = \mathbf{A}_i^n + \mathbf{A}_i^t \tag{II.39}$$

$$= \rho A L_u\big[C_{an}(\mathbf{I}_{3\times 3} - \mathbf{q}_i\mathbf{q}_i^T) + C_{at}(\mathbf{q}_i\mathbf{q}_i^T)\big] \tag{II.40}$$

Thus, once all the loads are computed, the acceleration of every node may be evaluated:

$$\forall i \in [\![0\ ;\ N_{Elements}]\!]\ \ddot{\mathbf{r}}_i = (\mathbf{M}_i + \mathbf{A}_i)^{-1}(\mathbf{W}_i + \mathbf{B}_i + \mathbf{T}_i + \mathbf{C}_i + \mathbf{D}_i) \tag{II.41}$$

II.2.3 State vector and time integration

The state vector of the low-order lumped mass theory (LM) is:

$$\mathbf{Y}_{LM} = \begin{pmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{pmatrix} \tag{II.42}$$

where $\mathbf{r}$ and $\dot{\mathbf{r}}$ gather the positions and the velocities of all nodes in following the numbering of Figure II.3.

The time-derivative of the state vector is obtained from (II.41). The time-stepping is achieved using a fourth-order explicit Runge-Kutta scheme with a fixed time step (I.94).

These theoretical developments have been implemented in a Python numerical tool named *CableDyn*.

II.2.4 Validation

Buckham [64] presented a validation of his low-order lumped mass-based numerical tool using two test cases considering a single towline. A kinematic boundary condition is applied at the towpoint, located at the node 0 of the cable and at the mean sea level ($r_0^2 = r_0^3 = 0$). The first maneuver consists in a horizontal acceleration of this point while the second one is a horizontal deceleration. Comparisons are made with the numerical results of Buckham and experimental data of Vaz and Patel [89]. The characteristics of the 6-element cable are listed in Table II.2. This cable represents an armoured optical fibre cable.

Parameter	Value
Equivalent diameter (m)	0.0332
Cable density (kg/m^3)	3121
Young's modulus (GPa)	77.5
Damping coefficient (N s/m^2)	11 551
Total unstretched length (m)	300
Normal drag coefficient	1.649
Tangential drag coefficient	0
Normal added-mass coefficient	1
Tangential added-mass coefficient	0

Table II.2 – Cable mechanical and hydrodynamic properties

The water density, ρ, is 1025 kg /m^3 and the time step is 0.1 s.

In the acceleration test case, the cable is towed at a velocity of 0.566 m/s for a 1000 s period so that it reaches a steady state, then the towpoint is accelerated at a constant acceleration of 0.011 15 m/s^2 over a 60 s interval and achieves a constant speed of 1.235 m/s. The kinematic boundary condition at the towpoint is:

$$r_0^1 = \begin{cases} 0.566t & \text{if } t \leqslant 1000 \text{ s} \\ 566 + 0.566(t-1000) + 0.005575(t-1000)^2 & \text{if } t > 1000 \text{ s and } t \leqslant 1060 \text{ s} \\ 620.03 + 1.235(t-1060) & \text{if } t > 1060 \text{ s} \end{cases} \tag{II.43}$$

Regarding the deceleration test case, the cable is towed at a velocity of 1.286 m/s for the same period of 1000 s. Then the towpoint is decelerated at a constant acceleration of -0.0129 m/s^2 over a 60 s interval and achieves a constant speed of 0.514 m/s. The kinematic boundary condition at the towpoint in this test case is:

$$r_0^1 = \begin{cases} 1.286t & \text{if } t \leqslant 1000 \text{ s} \\ 1286 + 1.286(t-1000) - 0.00643(t-1000)^2 & \text{if } t > 1000 \text{ s and } t \leqslant 1060 \text{ s} \\ 1340.012 + 0.514(t-1060) & \text{if } t > 1060 \text{ s} \end{cases} \tag{II.44}$$

The comparisons between the experimental data provided by Vaz and Patel [89], the numerical results of Buckham [64] and *CableDyn* are shown in Figure II.5 for the acceleration test case and in Figure II.6 for the deceleration test case. A fair agreement is observed in both cases. Nevertheless, numerical results obtained from *CableDyn* and from Buckham do not exactly match. This may be explained by the different formulation of the hydrodynamic loads (drag and added-mass loads) in the two low-order lumped mass theories compared here. For example, Buckham [64] used a loading function in the expression of the drag loads for accounting for the non-linear breakup between the normal and tangential components.

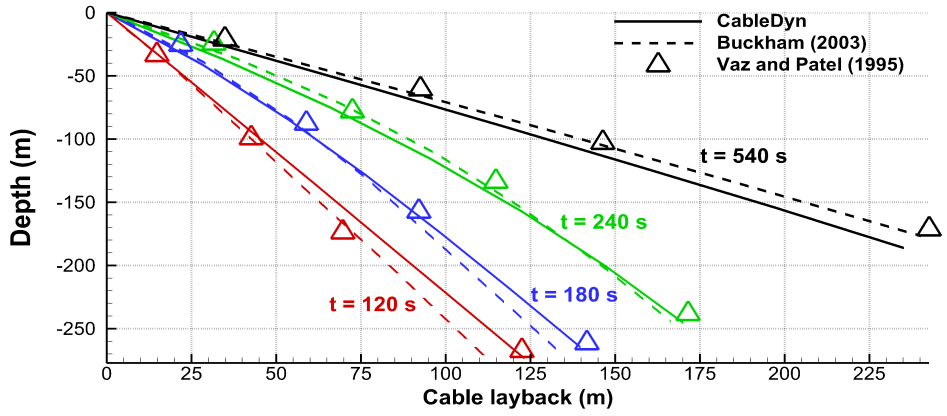


Figure II.5 – Cable profiles during the acceleration test case at different times: in red at t =120 s, in blue at t =180 s, in green at t =240 s and in black at t =540 s.

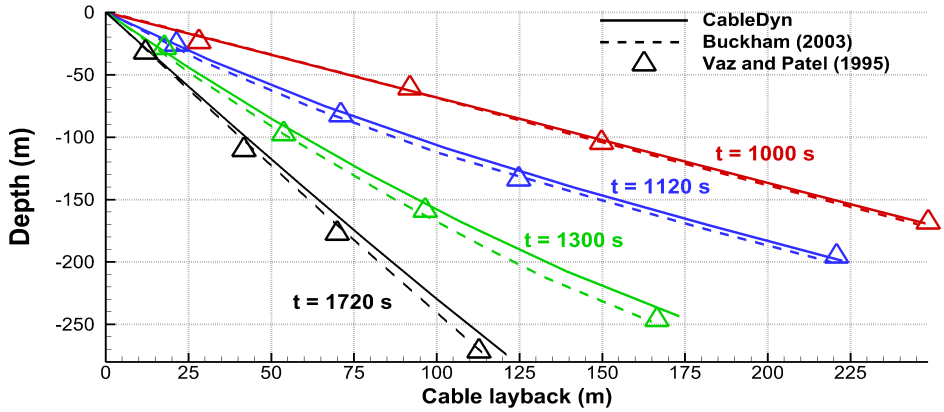


Figure II.6 – Cable profiles during the deceleration test case at different times: in red at t =1000 s, in blue at t =1120 s, in green at t =1300 s and in black at t =1720 s.

II.3 Cable dynamics using the *CRBA*

II.3.1 Theoretical developments

In the low-order lumped mass theory, each node has three degrees of freedom. Within the multibody system modeling, each node needs three joints to ensure two rotations at the position of the node and one translation to take into account the stretching along the cable element linked to the node. As the theory of the Composite-Rigid-Body Algorithm, presented in section I.4, only used single-degree-of-freedom joints, three bodies are necessary, including two massless bodies and one physical body. To sum-up, every cable element j is made of:

- A massless body $3j-2$;
- A revolute joint $3j-2$ around $\mathbf{y}_{3(j-1)} = \mathbf{z}_{3j-2}$;
- A massless body $3j-1$;
- A revolute joint $3j-1$ around $\mathbf{y}_{3j-2} = \mathbf{z}_{3j-1}$;
- A physical body $3j$ where the mass is lumped;
- A prismatic joint $3j$ along $\mathbf{y}_{3j-1} = \mathbf{z}_{3j}$.

The numbering of the bodies, given in this list for indication, assumes the cable starts at the body 1 which, obviously, is not always the case. The local frames are defined in Figure II.7. For each body $\mathbf{u}_j = \mathbf{x}_i$ with $i = a_j$.

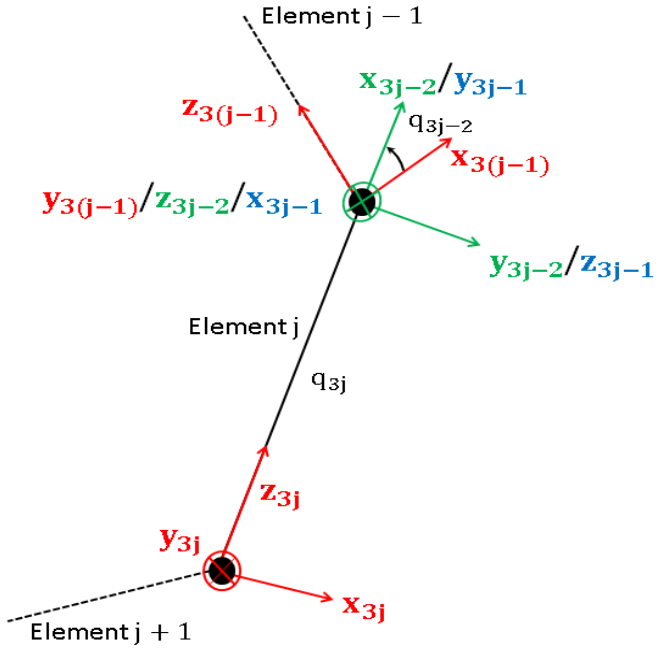


Figure II.7 – Local frames for the cable element j in assuming $q_{3j-1} = 0$. The red color is dedicated to the prismatic joints ($3(j-1)$ and $3j$) while the blue and green colors are used for the revolute joins $3j-1$ and $3j-2$ respectively.

The modified Denavit-Hartenberg parameters, defined in subsection I.2.2, for these three bodies are given in Table II.3. Based on the construction of the local frames, q_{3j} is negative. In case of a straight vertical cable, $q_{3j-2} = q_{3j-1} = -\frac{\pi}{2}$ rad.

	σ_j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
Body $3j-2$	0	0	0	$-\frac{\pi}{2}$	0	q_{3j-2}	0
Body $3j-1$	0	0	0	$-\frac{\pi}{2}$	0	q_{3j-1}	0
Body $3j$	1	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	q_{3j}

Table II.3 – Modified Denavit-Hartenberg parameters for the cable element j

To ensure the internal loads in the cable, articular torques and forces must be applied to match (II.29) and (II.32). They are defined by [90]:

$$\Gamma_{3j-2} = 0 \tag{II.45}$$

$$\Gamma_{3j-1} = 0 \tag{II.46}$$

$$\Gamma_{3j} = \begin{cases} -\frac{EA}{L_u}(q_{3j} + L_u) - \frac{CA}{L_u}\dot{q}_{3j} & \text{if } |q_{3j}| \geqslant L_u \\ -\frac{CA}{L_u}\dot{q}_{3j} & \text{otherwise} \end{cases} \tag{II.47}$$

Equations (II.45) and (II.46) involve that no internal torque is added in the revolute joints as neither bending nor torsion effect are modeled. Equation II.47 ensures the stretching effects due to the elasticity and the viscous damping. As with the low-order lumped mass theory, no compression of the cable is modeled.

Regarding the hydrodynamic loads, they are only applied on the physical body $3j$. The expression of the buoyancy force is:

$$ {}^{3j}\mathbf{F}_{e3j}^{Buoyancy} = \begin{pmatrix} \rho A L_u g \, {}^{3j}\mathbf{R}_e \, \mathbf{z}_e \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{II.48} $$

${}^{3j}\mathbf{v}_{3j}$ is the velocity of the physical body $3j$ in its own local frame. Its two first components, ${}^{3j}\mathbf{v}_{3j}(1)$ and ${}^{3j}\mathbf{v}_{3j}(2)$, represent the normal velocity while the third component, ${}^{3j}\mathbf{v}_{3j}(3)$, is the tangential velocity of the cable element. So the drag force at the body $3j$ is expressed by:

$$ {}^{3j}\mathbf{F}_{e3j}^{Drag} = -\begin{pmatrix} \frac{1}{2}\rho d L_u \left[C_{dn} \left\| \begin{pmatrix} {}^{3j}\mathbf{v}_{3j}(1) \\ {}^{3j}\mathbf{v}_{3j}(2) \\ 0 \end{pmatrix} \right\| \begin{pmatrix} {}^{3j}\mathbf{v}_{3j}(1) \\ {}^{3j}\mathbf{v}_{3j}(2) \\ 0 \end{pmatrix} + \pi C_{dt} |{}^{3j}\mathbf{v}_{3j}(3)| \begin{pmatrix} 0 \\ 0 \\ {}^{3j}\mathbf{v}_{3j}(3) \end{pmatrix} \right] \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{II.49} $$

By the same reasoning, the added-mass matrix arises:

$$ {}^{3j}\mathbf{M}_{3j}^{Added-mass} = \begin{pmatrix} {}^{3j}\mathbf{A}_{3j} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \end{pmatrix} \tag{II.50} $$

$$ {}^{3j}\mathbf{A}_{3j} = \rho A L_u \begin{pmatrix} C_{an} & 0 & \\ 0 & C_{an} & 0 \\ 0 & 0 & C_{at} \end{pmatrix} \tag{II.51} $$

These developments have been implemented in *InWave*. Furthermore, as nothing in this section is specific to the Composite-Rigid-Body Algorithm, this model can also be applied to the Articulated-Body Algorithm.

II.3.2 Validation

The multibody approach for computing cable dynamics, based on the Composite-Rigid-Body Algorithm, is now compared to the low-order lumped mass theory. The two test cases presented in subsection II.2.4 require the control of the towpoint position. The cable modeling described in the previous section cannot be used immediately for this purpose. Extra joints should be added to enable the translational motion of the towpoint. For the sake of understanding, a more suitable test case is defined. A 3-element cable is considered with the same mechanical properties as defined in Table II.2, except for:

- The Young's modulus: 77.5 MPa;
- The damping coefficient: $\frac{10^5}{A}\,\mathrm{N\,s/m^2}$;
- The total unstretched length: 10 m.

The time step is 0.001 s and the total duration of the simulation is 10 s.

The node 0 is kept at the position $\mathbf{r}_0 = \mathbf{0}_{3\times 1}$ during the simulation while the starting position of the node 3 is $\mathbf{r}_3 = (-2, 0, -10)^T$. The two internal nodes are along the straight line between the nodes 0 and 3. In the multibody modeling, 10 bodies are necessary: 9 due to the three cable elements and one more for the base. The cable motion is in the the plane $(\mathbf{O}_e, \mathbf{x}_e, \mathbf{z}_e)$ so $q_{3j-1} = -\frac{\pi}{2}\,\mathrm{rad}\ \forall j \in [\![1\ ;\ 3]\!]$. As the cable is

straight, $q_4 = q_7 = -\frac{\pi}{2}$ rad at the starting time. Finally, the initial angle is imposed by fixing $q_1 = -1.3734$ rad.

Different cable profiles are presented in Figure II.8.

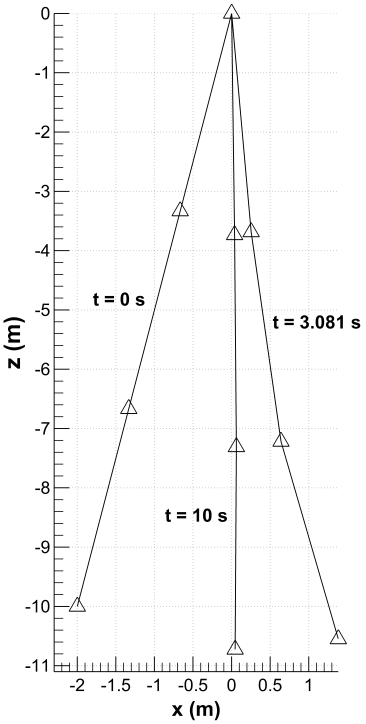


Figure II.8 – Profiles of an oscillating 3-element cable. The triangles represent the position of the nodes. The black lines denote the cable elements.

The relative length, respectively the relative velocity, of the third cable element (matching the articular variable q_9, respectively $\dot{q}_9$) obtained with the *CRBA* and the lumped mass theory (*LM*) are compared in Figure II.9, respectively Figure II.10. A very good agreement is observed. Some differences are observed, due to the different formulation of the hydrodynamic loads. For example, with the lumped mass approach, to define the drag loads, the tangent direction is given by Equation II.33 and is the approximated direction of a line passing between the two adjacent node points, whereas with the *CRBA*, $\mathbf{q}_j = {}^e\mathbf{z}_j$, the direction of the cable element j.

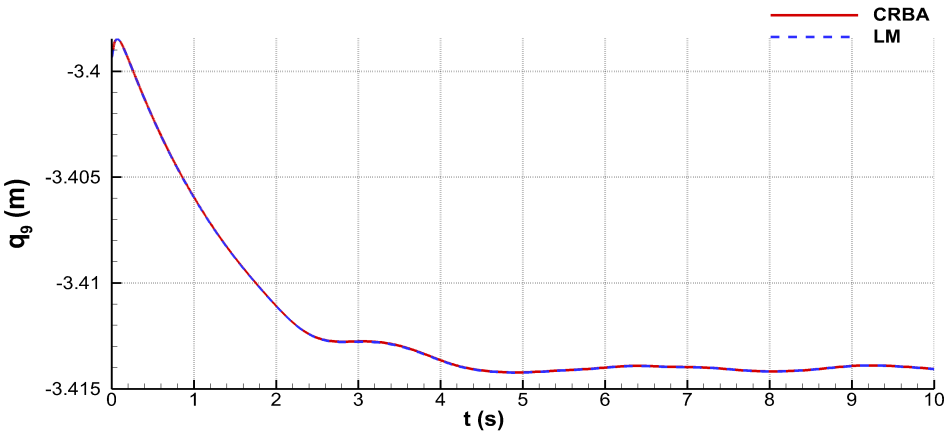


Figure II.9 – Comparison of q_9 from numerical results of the *CRBA* and the lumped mass theory (*LM*)

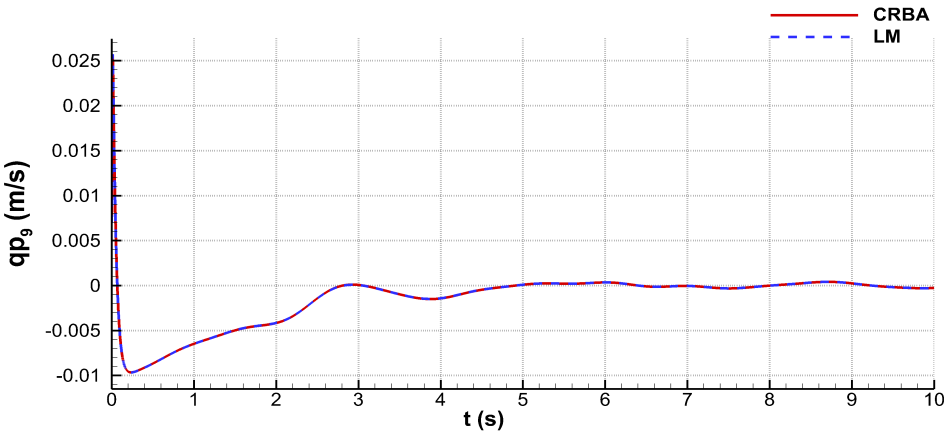


Figure II.10 – Comparison of $\dot{q}_9$ from numerical results of the *CRBA* and the lumped mass theory (*LM*)

II.4 Cable joints in the *CRBA*

II.4.1 Theoretical developments

Each cable element results in three bodies in the multibody system and two of them are massless. This leads to useless computations. In order to speed up the *CRBA*, instead of using single-degree-of-freedom, cable joints are created. They are formed of the same three bodies presented in the last section.

The forward loop of the Composite-Rigid-Body Algorithm stays identical, because all elementary physical quantities must still be computed. Regarding the backward loop of the *CRBA*, (I.77) and (I.89) become [90]:

$$ {}^{3(j-1)}\mathbf{M}^c_{3(j-1)} = {}^{3(j-1)}\mathbf{M}_{3(j-1)} + {}^{3j}\mathbf{T}^T_{3(j-1)}\ {}^{3j}\mathbf{M}^c_{3j}\ {}^{3j}\mathbf{T}_{3(j-1)} \tag{II.52} $$

$$ {}^{3(j-1)}\tilde{\boldsymbol{\beta}}^c_{3(j-1)} = {}^{3(j-1)}\boldsymbol{\beta}_{3(j-1)} + {}^{3j}\mathbf{T}^T_{3(j-1)}\left[{}^{3j}\mathbf{M}^c_{3j}\left(\sum_{k=0}^{2}{}^{3j}\mathbf{T}_{3j-k}\ {}^{3j-k}\boldsymbol{\gamma}_{3j-k}\right) + {}^{3j}\tilde{\boldsymbol{\beta}}^c_{3j}\right] \tag{II.53} $$

Regarding the construction of the matrix $\mathbf{H}$ and the vector $\mathbf{C}$, (I.83), (I.90) and (I.92) turn to:
$\forall q \in [\![1\ ;\ N_{Elements}]\!]$:

$$col_{[3q-2,3q-1,3q]}(\mathbf{H}_{12}) = {}^{3q}\mathbf{T}_0^T\ {}^{3q}\mathbf{M}_{3q}^c\ {}^{3q}\mathbf{P}_q \tag{II.54}$$

$\forall (p,q) \in [\![1\ ;\ N_{Elements}]\!]^2$ such as $p \geqslant q$:

$$row_{[3p-2,3p-1,3p]}(col_{[3q-2,3q-1,3q]}(\mathbf{H}_{22})) = {}^{3p}\mathbf{P}_p^T\ {}^{3p}\mathbf{M}_{3p}^c\ {}^{3p}\mathbf{T}_{3q}\ {}^{3q}\mathbf{P}_q \tag{II.55}$$

$\forall p \in [\![1\ ;\ N_{Elements}]\!]$:

$$row_{[3p-2,3p-1,3p]}(\mathbf{C}_2) = {}^{3p}\mathbf{P}_p^T \left[{}^{3p}\mathbf{M}_{3p}^c \sum_{i \in \beta(3p)} {}^{3p}\mathbf{T}_{3i} \left(\sum_{k=0}^{2} {}^{3i}\mathbf{T}_{3i-k}\ {}^{3i-k}\boldsymbol{\gamma}_{3i-k} \right) + {}^{3p}\tilde{\boldsymbol{\beta}}_{3p}^c \right] \tag{II.56}$$

$col_{[3q-2,3q-1,3q]}(\mathbf{A})$, respectively $row_{[3q-2,3q-1,3q]}(\mathbf{A})$, denotes the $(3q-2)$-th, $(3q-1)$-th and $3q$-th columns, respectively rows, of $\mathbf{A}$. $N_{Elements}$ is the number of cable elements.

${}^{3j}\mathbf{P}_j$ represents the (6×3) generalized projection matrix of the cable joint j expressed in the local frame of the body $3j$, equivalent of ${}^{j}\mathbf{a}_j$ for the single-*dof* joints, and defined by:

$${}^{3j}\mathbf{P}_j = \begin{pmatrix} {}^{3j}\mathbf{T}_{3j-2}\ {}^{3j-2}\mathbf{a}_{3j-2} & {}^{3j}\mathbf{T}_{3j-1}\ {}^{3j-1}\mathbf{a}_{3j-1} & {}^{3j}\mathbf{a}_{3j} \end{pmatrix} \tag{II.57}$$

II.4.2 Validation

The test case presented in subsection II.3.2 now serves to compare the cable modeling with the *CRBA* using the single-*dof* joint formulation and the cable joint formulation. Numerical results are displayed in Figure II.11. The two formulations match perfectly.

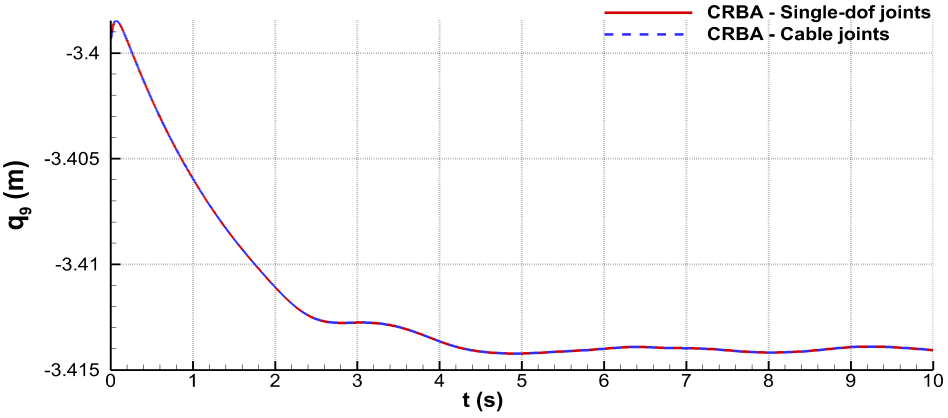


Figure II.11 – Comparison of q_9 from numerical results of the *CRBA* with both the single-*dof* formulation and the cable joint formulation

The CPU-time comparison of the different cable approaches can only be done with the same programming language. But *InWave* is implemented in C++ and *CableDyn* in Python. To achieve the CPU-time comparison, *InWave* is substituted for *DirectDyn*, a Python numerical tool where the Articulated-Body-Algorithm along with the Composite-Body-Algorithm with both single-*dof* and cable joint formulations are implemented. The mean CPU-time over 10 runs of the pendulum test case for each approach

is presented in Table II.4. The *ABA* used here has a single-*dof* formulation. Obviously, the implementation of cable joints could also be achieved in this direct dynamics algorithm.

ABA	Single-*dof* *CRBA*	Cable joint *CRBA*	LM
160.3 s	175.1 s	89.0 s	35.2 s

Table II.4 – CPU-time of different cable dynamics theories applied to the pendulum test case

The use of cable joints in the *CRBA* involves a time reduction of 49.2 % compared to the use of single-*dof* joints. The forward loop is not affected by this reduction because every elementary quantity (velocities, Coriolis accelerations, external loads, etc.) always has to be evaluated for each body. The backward loop, which now iterates the number of cable elements instead of the number of bodies, and the construction of the generalized mass matrix $\mathbf{H}$ and the load vector $\mathbf{C}$, previously coefficient-by-coefficient and now by sets of 3×3 matrices for $\mathbf{H}$ and 3×1 vectors for $\mathbf{C}$, are at the origin of the CPU-time reduction.

As expected, the *ABA* is less time-consuming than the *CRBA* when both of them are based on single-*dof* joints. Indeed, the Composite-Body-Algorithm needs the inversion of $\mathbf{H}$ whereas the Articulated-Body Algorithm does not. It is interesting to notice that the *CRBA* with a cable joint formulation is less cumbersome than the *ABA* with single-*dof* joints.

The time gap between the multibody approaches and the lumped-mass theory originates from the mechanical interactions which are not taken into account by the latter theory. Only one loop is achieved over the nodes of the cable per Runge-Kutta step whereas several loops are made through the multibody system in both the *CRBA* and the *ABA*.

II.5 Winch modeling

In a lowering or lifting operation, the hoisting cable is unwound or wound using a winch. A winch is also used in case of a towed body [91] or a tethered underwater vehicle system [92]. Therefore, a winch model has to be added to our numerical tool to perform marine operation simulations.

II.5.1 State of the art

Although lot of works exist about the simulation of the dynamics of fixed length cables, only a few deal with variable length cables [92]. Banerjee and Do [93] developed an underwater cable dynamics model using a cable controller to achieve the cable deployment or retrieval at the ship. The cable deployment control law was:

$$\dot{L} = K_V(L_C - L) \tag{II.58}$$

with:

- L is the cable scope;
- L_C denotes the controlled cable length;

– K_V represents the gain.

Wang et al. [94] used a finite-element method to model a time-varying length cable. The cable length $L(t)$ followed the formula:

$$L = L_0 + v_d t \tag{II.59}$$

where L_0 is the initial cable length, v_d the deployment cable velocity and t the time.

Kamman and Huston [91] presented a model of variable length towed cable based on a lumped mass theory. They assumed the number of cable elements to be held constant during the simulation. Thus, when a cable was paid out, respectively reel-in, cable elements were expanded, respectively shortened, one after the other. The thrust force created by the mass flux between the vessel and the cable through the winch was not accounted for. The mass flux is created by the increase or the decrease of the cable mass due its variable length. Prabhakar and Buckham [92] developed a model in which the topmost cable element length became a time varying quantity while the rest of the cables remained constituted of constant length elements. The force induced by the winch was added to the motion equation of the topmost element.

When the element connected to the winch is too long, it is split into a fixed length element and a new variable length element. On the contrary, when the topmost element is too small, it is merged with the subsequent downstream element. This model was used by Zand et al. [95] too.

In the software *OrcaFlex*, the unstretched length of the cable element linked to the winch is controlled and time-dependent [38].

A winch model based on the work of Zand et al. [95] and the theory of *OrcaFlex* has been developed for both the low-order lumped mass theory and the multibody theory.

II.5.2 Thrust force and cable loads

The deployment or the retrieval of a cable by a winch involves a variation of the mass of the cable hanging from the winch. The force due to the momentum flux at the winch boundary, which matches the position of the node 0 is [95]:

$$\mathbf{F}_0^{Winch} = v_d \left(\frac{\mathbf{r}_1 - \mathbf{r}_0}{\|\mathbf{r}_1 - \mathbf{r}_0\|} \right) \frac{d\mathbf{M}_0}{dt} \tag{II.60}$$

where v_d is the deployment velocity of the winch, positive for a payout, negative for a retrieval.

The unstretched length of the first cable element is given by:

$$L_u(t) = L_u(0) + \int_0^t v_d(\tau)\, d\tau \tag{II.61}$$

Thus, the mass flux follows the expression:

$$\frac{d\mathbf{M}_0}{dt} = A\rho_c v_d \tag{II.62}$$

Equation II.23 for the first node becomes:

$$(\mathbf{M}_0 + \mathbf{A}_0)\ddot{\mathbf{r}}_0 = \mathbf{W}_0 + \mathbf{B}_0 + \mathbf{T}_0 + \mathbf{C}_0 + \mathbf{D}_0 + \mathbf{F}_0^{Winch} \tag{II.63}$$

The axial tension and damping due to the top cable element are updated with the time-varying unstretched length.

$$\mathbf{T}_1^{Element} = \begin{cases} -\dfrac{EA}{L_u(t)}(\|\mathbf{r}_1 - \mathbf{r}_0\| - L_u(t))\left(\dfrac{\mathbf{r}_1 - \mathbf{r}_0}{\|\mathbf{r}_1 - \mathbf{r}_0\|}\right) & \text{if } \|\mathbf{r}_1 - \mathbf{r}_0\| \geqslant L_u(t) \\ \mathbf{0}_{3\times 1} & \text{otherwise} \end{cases} \tag{II.64}$$

$$\mathbf{C}_1^{Element} = -\frac{CA}{L_u(t)}\left[\frac{\partial}{\partial t}(\|\mathbf{r}_1 - \mathbf{r}_0\| - L_u(t))\right]\left(\frac{\mathbf{r}_1 - \mathbf{r}_0}{\|\mathbf{r}_1 - \mathbf{r}_0\|}\right) \tag{II.65}$$

with:

$$\frac{\partial}{\partial t}(\|\mathbf{r}_1 - \mathbf{r}_0\| - L_u(t)) = \frac{1}{\|\mathbf{r}_1 - \mathbf{r}_0\|}\sum_{j=1}^{3}(r_1^j - r_0^j)(\dot{r}_1^j - \dot{r}_0^j) - v_d(t) \tag{II.66}$$

Regarding the multibody theory, the cable loads are expressed by:

$$\Gamma_3 = \begin{cases} -\dfrac{EA}{L_u(t)}(q_3 + L_u(t)) - \dfrac{CA}{L_u(t)}(\dot{q}_3 + v_d(t)) + v_d(t)\dfrac{d\mathbf{M}_0}{dt} & \text{if } |q_3| \geqslant L_u(t) \\ -\dfrac{CA}{L_u(t)}(\dot{q}_3 + v_d(t)) + v_d(t)\dfrac{d\mathbf{M}_0}{dt} & \text{otherwise} \end{cases} \tag{II.67}$$

II.5.3 Cable deployment

During a pay-out, the unstretched length of the cable element connecting to the winch is increased using (II.61). Once the cable length is too long, the first element is divided into two elements. The one linked to the winch is a variable-length element while the other ones are constant-length elements of unstretched length $L_u(0)$. The splitting criterion is:

$$L_u(t) = \alpha L_u(0) \tag{II.68}$$

with $\alpha > 1$.

If α is too small, the new variable-length element is too small and numerical errors may appear. If α is too big, large elements arise which could also lead to numerical errors. A good compromise is to create a new element whose its length is the half of the initial cable element size, so:

$$\alpha = \frac{3}{2} \tag{II.69}$$

II.5.3.1 Low-order lumped mass model

Figure II.12 shows the scheme of the cable splitting using the lumped mass approach. It consists of adding a new node between the two top nodes.

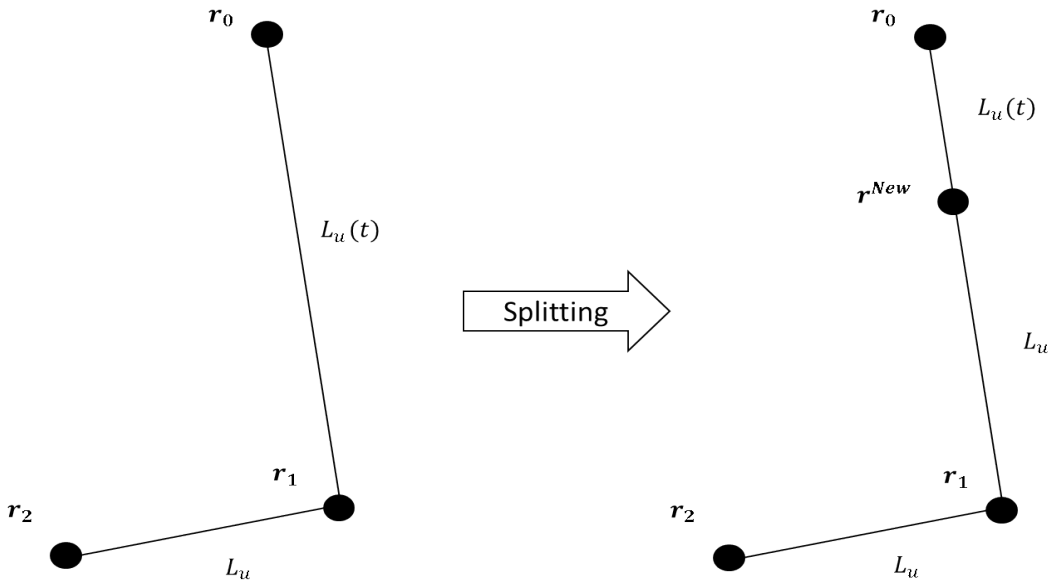


Figure II.12 – Cable splitting with the low-order lumped mass theory

The position and the velocity of the new node are:

$$\mathbf{r}^{New} = \mathbf{r}_0 + \frac{1}{3}(\mathbf{r}_1 - \mathbf{r}_0) \tag{II.70}$$

$$\dot{\mathbf{r}}^{New} = \dot{\mathbf{r}}_0 + \frac{1}{3}(\dot{\mathbf{r}}_1 - \dot{\mathbf{r}}_0) \tag{II.71}$$

The mass distribution is updated, which gives for the new node:

$$\mathbf{M}_{New} = \frac{1}{2}A\rho_c(L_u(t) + L_u) \tag{II.72}$$

where $L_u(t)$ represents the length of the new first cable element.

II.5.3.2 Multibody model

Figure II.13 displays the scheme of the cable splitting using the multibody approach. A cable element, made of three bodies and joints as explained in subsection II.3.1, is added. It requires to initialize the prismatic joint of the new cable element and to update the relative coordinates of the succeeding cable element.

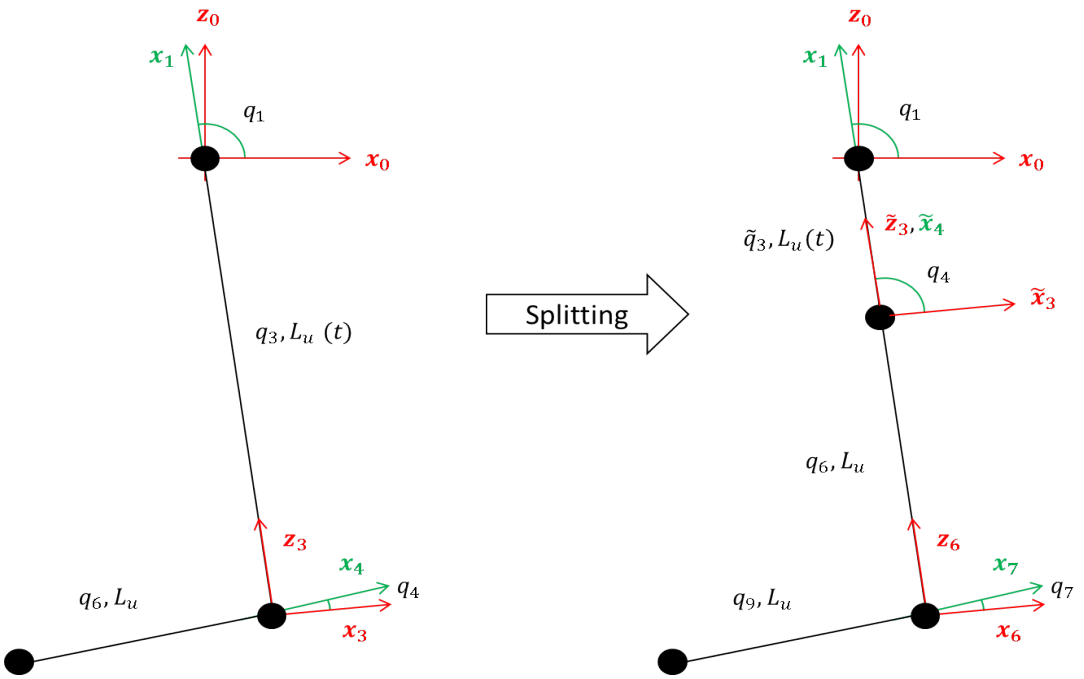


Figure II.13 – Cable slitting with the multibody theory

The new articular positions and velocities are:

$$\tilde{q}_3 = \frac{1}{3} q_3 \tag{II.73}$$

$$\dot{\tilde{q}}_3 = \frac{1}{3} \dot{q}_3 \tag{II.74}$$

$$q_4 = -\frac{\pi}{2} \tag{II.75}$$

$$\dot{q}_4 = 0 \tag{II.76}$$

$$q_5 = -\frac{\pi}{2} \tag{II.77}$$

$$\dot{q}_5 = 0 \tag{II.78}$$

$$q_6 = q_3 - \tilde{q}_3 \tag{II.79}$$

$$\dot{q}_6 = \dot{q}_3 - \dot{\tilde{q}}_3 \tag{II.80}$$

The mass distribution distribution is updated as for the low-order lumped mass model.

II.5.3.3 Comparison

These two models, for the lumped mass theory and the multibody theory, are compared using the test case presented in subsection II.3.2. The lowering velocity is fixed to 0.5 m/s between 8 s and 44 s. No ramp is used on the lowering velocity.

Figure II.14 shows the length of the topmost cable element or the articular variable of the first prismatic joint. The discontinuity of the length indicates the addition of a new cable element while the straight lines indicate the increase of the unstretched length. The cut-off length for adding a new element, equal to $\frac{3}{2} L_u$ is time-varying due to the elasticity of the cable, q_3 denoting the real length of the cable element and not its

unstretched length. The two curves match perfectly, demonstrating that both Cartesian and multibody approaches allow to unwind a cable. Figure II.15 presents the angle of the top cable element with respect to the vertical axis. Contrary to the previous graph, this curve is continuous despite the creation of the new cable elements. As a reminder, at the starting time, the cable has an angle of $-78.69°$ with the vertical. Then the cable reaches a vertical position, leading to a final angle of $q_1 = \frac{-\pi}{2}$ rad. Figure II.16 features the cable profile at different time steps.

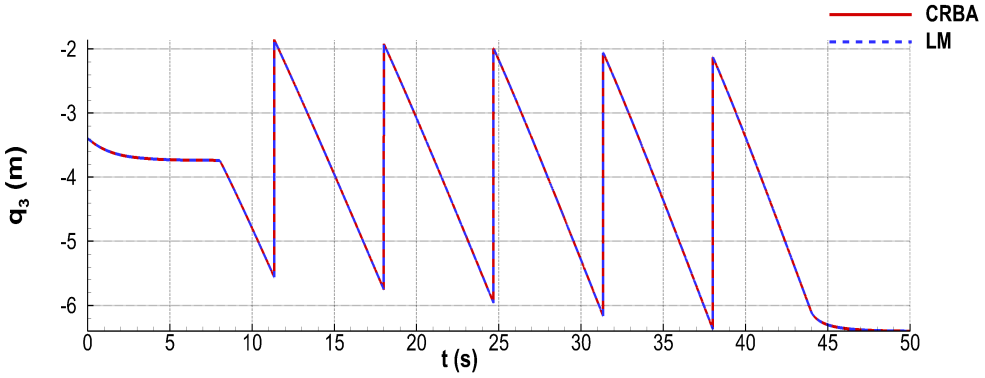


Figure II.14 – Comparison of the length of the topmost cable element from numerical results of the *CRBA* and the lumped mass theory (*LM*)

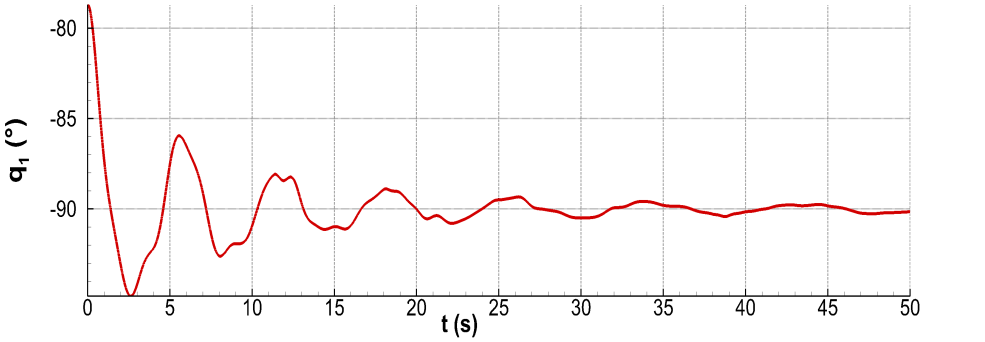
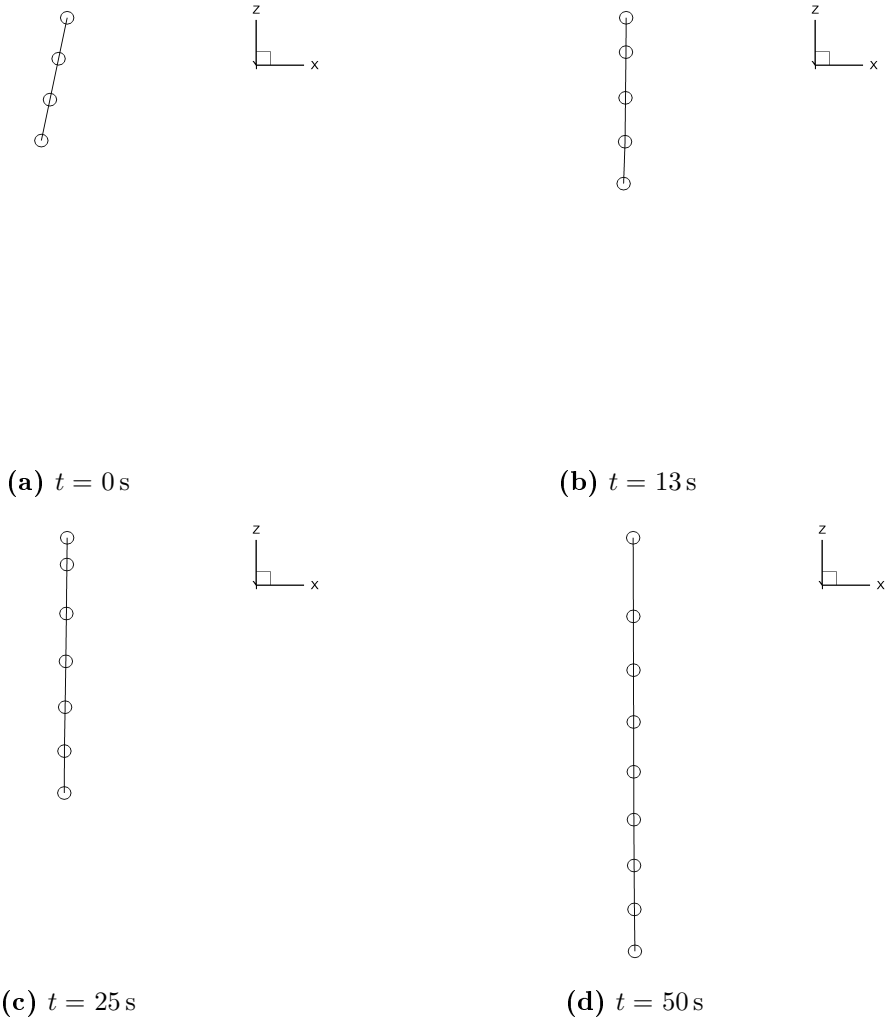


Figure II.15 – Angle between the topmost cable element and the vertical axis from numerical results of the *CRBA*



(a) $t = 0\,\mathrm{s}$ **(b)** $t = 13\,\mathrm{s}$

(c) $t = 25\,\mathrm{s}$ **(d)** $t = 50\,\mathrm{s}$

Figure II.16 – Cable profile during a lowering operation at different time steps

II.5.4 Cable retrieval

During a pay-in, the unstretched length of the cable element connected to the winch is decreased using (II.61). Once the cable length is too small, the two first elements are merged. The resulting element is a variable-length element. The merging criterion is:

$$L_u(t) = \beta L_u(0) \tag{II.81}$$

with $\beta < 1$.

For the same reasons as given in subsection II.5.3, a good compromise is to merge the elements when:

$$\beta = \frac{1}{2} \tag{II.82}$$

II.5.4.1 Low-order lumped mass model

Figure II.17 displays the scheme of the cable merging using the lumped mass approach. The second node is simply deleted. Then, the first node is directly connected to the third one.

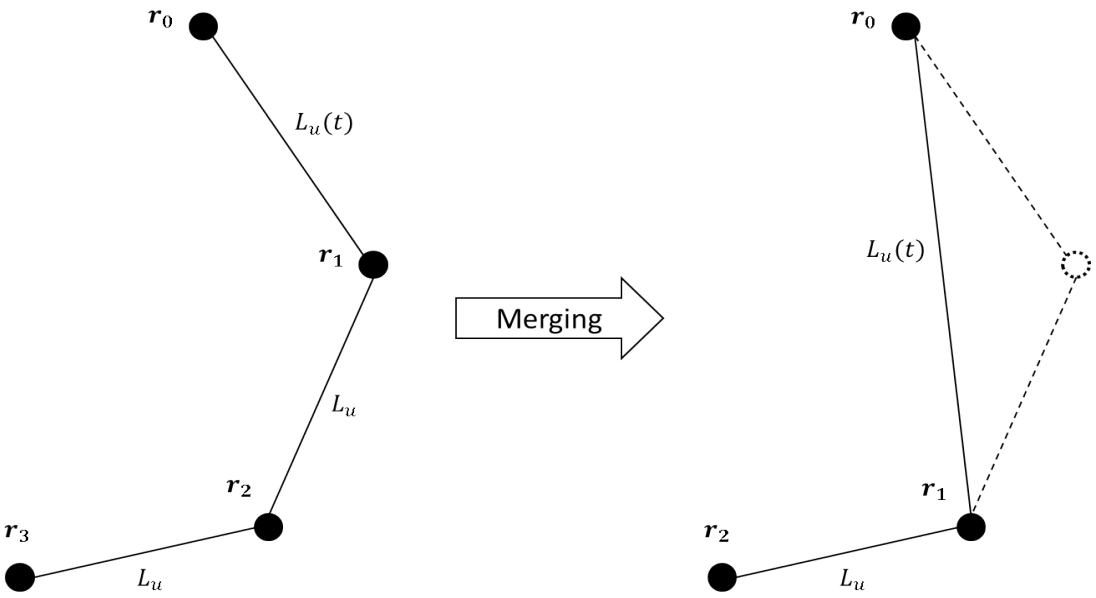


Figure II.17 – Cable merging with the low-order lumped mass theory

The mass distribution is updated, which gives for the node 1:

$$\mathbf{M}_1 = \frac{1}{2} A \rho_c (L_u(t) + L_u) \tag{II.83}$$

where $L_u(t)$ represents the length of the new first cable element.

II.5.4.2 Multibody model

Figure II.18 shows the scheme of the cable merging using the multibody approach. A cable node is deleted, which involves the deletion of two cable elements and the creation of a new one. The impact of this transformation also concerns the initial third cable element.

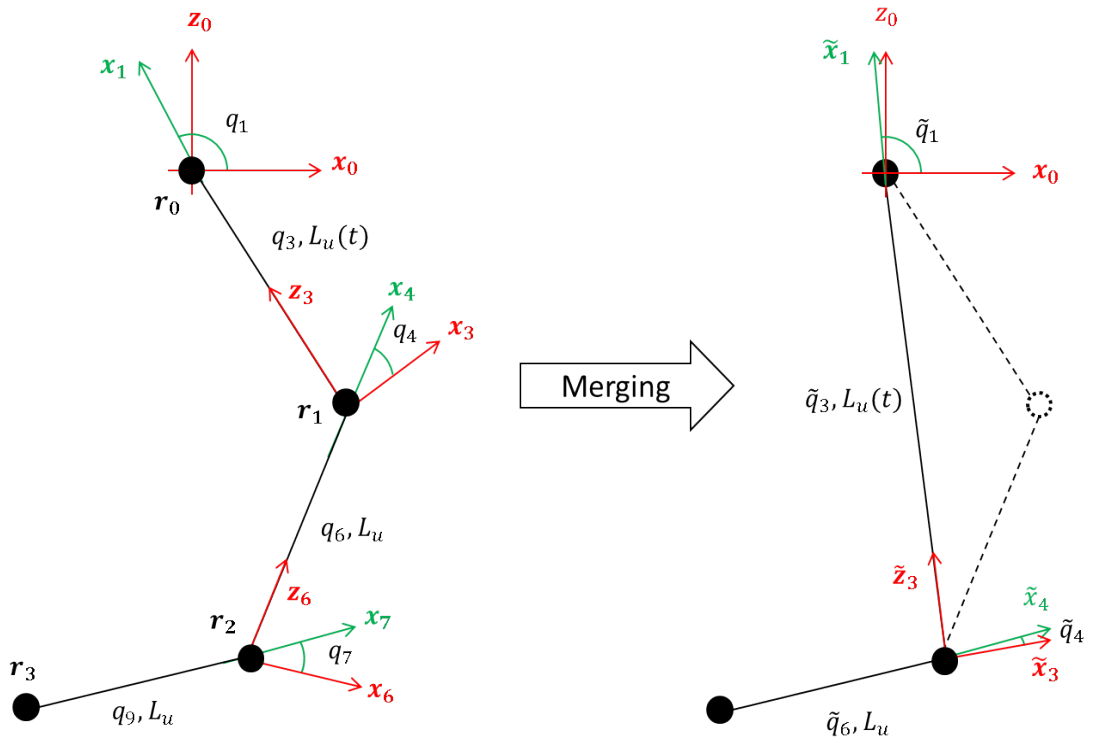


Figure II.18 – Cable merging with the multibody theory

The initialization of the articular positions and velocities requires the use of the Cartesian positions and velocities. The Cartesian positions are computed from the homogeneous transformation matrices ${}^{j}\mathbf{h}_{i}$, defined for the body j with $i = a_j$, by:

$$ {}^{j}\mathbf{h}_{i} = \begin{pmatrix} {}^{j}\mathbf{R}_{i} & {}^{i}\mathbf{P}_{j} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{II.84} $$

And for the base, by:

$$ {}^{0}\mathbf{h}_{e} = \begin{pmatrix} {}^{0}\mathbf{R}_{e} & {}^{0}\mathbf{R}_{e}\,\boldsymbol{\eta}_{0} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{II.85} $$

The recursive equation for the homogeneous matrices is:

$$ {}^{j}\mathbf{h}_{e} = {}^{j}\mathbf{h}_{i}\,{}^{i}\mathbf{h}_{e} \tag{II.86} $$

Finally, the Cartesian position of the body j, ${}^{e}\mathbf{P}_{j}$, is evaluated by:

$$ {}^{e}\mathbf{h}_{j} = {}^{j}\mathbf{h}_{e}^{-1} = \begin{pmatrix} {}^{e}\mathbf{R}_{j} & {}^{e}\mathbf{P}_{j} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{II.87} $$

Regarding the Cartesian velocities, they are computed from:

$$ {}^{e}\mathbf{V}_{j} = {}^{e}\mathbf{R}_{j}\,{}^{j}\mathbf{V}_{j} \tag{II.88} $$

Finally, the articular positions and velocities are [64]:

$$\tilde{q}_1 = -\frac{\pi}{2} + arctan\left(\frac{r_2^1 - r_0^1}{r_2^3 - r_0^3}\right) \tag{II.89}$$

$$\dot{\tilde{q}}_1 = \frac{(\dot{r}_2^1 - \dot{r}_0^1)(r_2^3 - r_0^3) - (\dot{r}_2^3 - \dot{r}_0^3)(r_2^1 - r_0^1)}{(r_2^1 - r_0^1)^2 + (r_2^3 - r_0^3)^2} \tag{II.90}$$

$$\tilde{q}_2 = \begin{cases} -\frac{\pi}{2} + arctan\left(\frac{-(r_2^2 - r_0^2)cos\left(\tilde{q}_1 + \frac{\pi}{2}\right)}{(r_2^3 - r_0^3)}\right) & \text{if } cos\left(\tilde{q}_1 + \frac{\pi}{2}\right) \geqslant sin\left(\tilde{q}_1 + \frac{\pi}{2}\right); \\ -\frac{\pi}{2} + arctan\left(\frac{-(r_2^2 - r_0^2)sin\left(\tilde{q}_1 + \frac{\pi}{2}\right)}{(r_2^1 - r_0^1)}\right) & \text{otherwise.} \end{cases} \tag{II.91}$$

$$\dot{\tilde{q}}_2 = \frac{-\left[\left((\dot{r}_2^2 - \dot{r}_0^2)cos\left(\tilde{q}_1 + \frac{\pi}{2}\right) - \dot{\tilde{q}}_1 sin\left(\tilde{q}_1 + \frac{\pi}{2}\right)(r_2^2 - r_0^2)\right)(r_2^3 - r_0^3) - (\dot{r}_2^3 - \dot{r}_0^3)(r_2^2 - r_0^2)cos\left(\tilde{q}_1 + \frac{\pi}{2}\right)\right]}{(r_2^3 - r_0^3)^2\left[1 + \left(\frac{-(r_2^2 - r_0^2)cos\left(\tilde{q}_1 + \frac{\pi}{2}\right)}{(r_2^3 - r_0^3)}\right)^2\right]} \tag{II.92}$$

$$\tilde{q}_3 = -\|\mathbf{r}_2 - \mathbf{r}_0\| \tag{II.93}$$

$$\dot{\tilde{q}}_3 = \frac{1}{\|\mathbf{r}_2 - \mathbf{r}_0\|}\sum_{j=1}^{3}(r_2^j - r_0^j)(\dot{r}_2^j - \dot{r}_0^j) \tag{II.94}$$

$$\tilde{q}_4 = -\frac{\pi}{2} + arctan\left(\frac{r_3^1 - r_2^1}{r_3^3 - r_2^3}\right) - \tilde{q}_1 \tag{II.95}$$

$$\dot{\tilde{q}}_4 = \frac{(\dot{r}_3^1 - \dot{r}_2^1)(r_3^3 - r_2^3) - (\dot{r}_3^3 - \dot{r}_2^3)(r_3^1 - r_2^1)}{(r_3^1 - r_2^1)^2 + (r_3^3 - r_2^3)^2} - \dot{\tilde{q}}_1 \tag{II.96}$$

$$\tilde{q}_5 = \begin{cases} -\pi + arctan\left(\frac{-(r_3^2 - r_2^2)cos\left(\tilde{q}_4 + \frac{\pi}{2}\right)}{(r_3^3 - r_2^3)}\right) - \tilde{q}_2 & \text{if } cos\left(\tilde{q}_4 + \frac{\pi}{2}\right) \geqslant sin\left(\tilde{q}_4 + \frac{\pi}{2}\right); \\ -\pi + arctan\left(\frac{-(r_3^2 - r_2^2)sin\left(\tilde{q}_4 + \frac{\pi}{2}\right)}{(r_3^1 - r_2^1)}\right) - \tilde{q}_2 & \text{otherwise.} \end{cases} \tag{II.97}$$

$$\dot{\tilde{q}}_5 = \frac{-\left[\left((\dot{r}_3^2 - \dot{r}_2^2)cos\left(\tilde{q}_4 + \frac{\pi}{2}\right) - \dot{\tilde{q}}_4 sin\left(\tilde{q}_4 + \frac{\pi}{2}\right)(r_3^2 - r_2^2)\right)(r_3^3 - r_2^3) - (\dot{r}_3^3 - \dot{r}_2^3)(r_3^2 - r_2^2)cos\left(\tilde{q}_4 + \frac{\pi}{2}\right)\right]}{(r_3^3 - r_2^3)^2\left[1 + \left(\frac{-(r_3^2 - r_2^2)cos\left(\tilde{q}_4 + \frac{\pi}{2}\right)}{(r_3^3 - r_2^3)}\right)^2\right]} \tag{II.98}$$

$$\tilde{q}_6 = q_9 \tag{II.99}$$

$$\dot{\tilde{q}}_6 = \dot{q}_9 \tag{II.100}$$

where r_i^j is the j^{th} component of the position vector $\mathbf{r}_i$ of the node i.

The mass distribution distribution is updated as for the low-order lumped mass model.

II.5.4.3 Comparison

These two models of reel-in are compared from a modified version of the test case presented in subsection II.3.2. The initial angle of $-78.69°$ is kept but the cable is three times longer and split into six elements instead of three. The node 0 stays at $\mathbf{r}_0 = \mathbf{0}_{3\times1}$ but the other extremity is now at $\mathbf{r}_6 = (-6, 0, -30)^T$. The lifting velocity is fixed to 0.5 m/s between 8 s and 44 s. No ramp is used on the lifting velocity.

Figure II.19 shows the length of the topmost cable element or the articular variable of the first prismatic joint. The discontinuity of the length outlines the deletion of a new cable element while the straight lines indicate the decrease of the unstretched length. The two curves match perfectly, demonstrating both Cartesian and multibody approaches allow to model the winding of a cable. Figure II.20 presents the angle of the top cable element with respect to the vertical axis. Contrary to the reel-in test case, some discontinuities were expected. Indeed, the suppression of a node involves an angular error close to the winch as it can be seen in Figure II.18. In the present test case, the cable remains almost vertical so this error is very small. Figure II.21 displays the cable profile at different time steps.

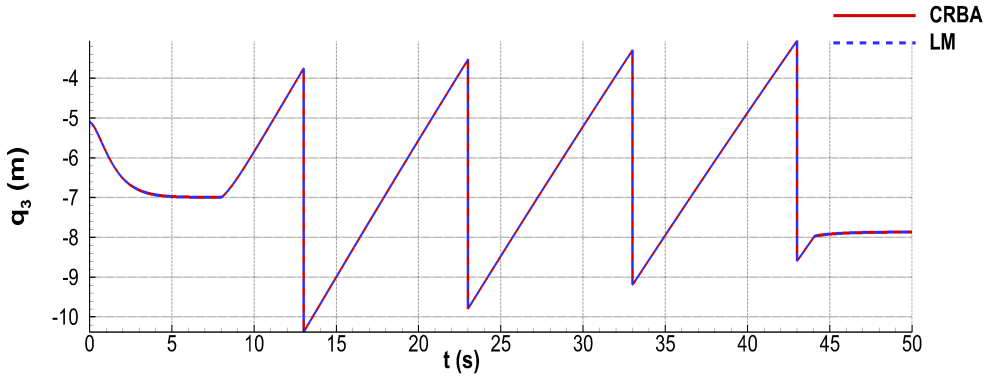


Figure II.19 – Comparison of the length of the topmost cable element from numerical results of the *CRBA* and the lumped mass theory (*LM*)

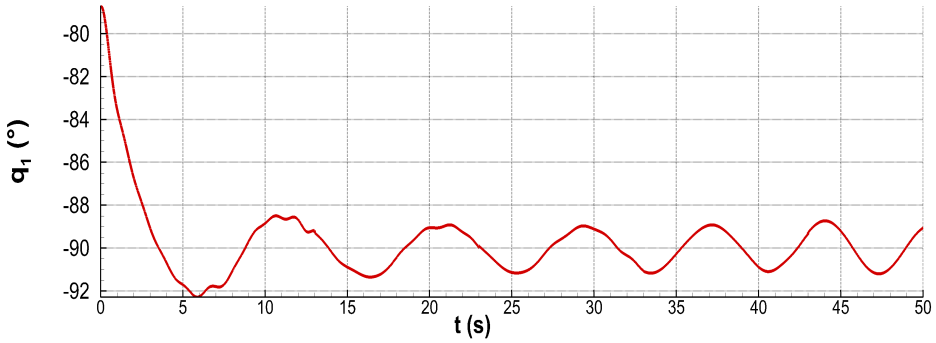
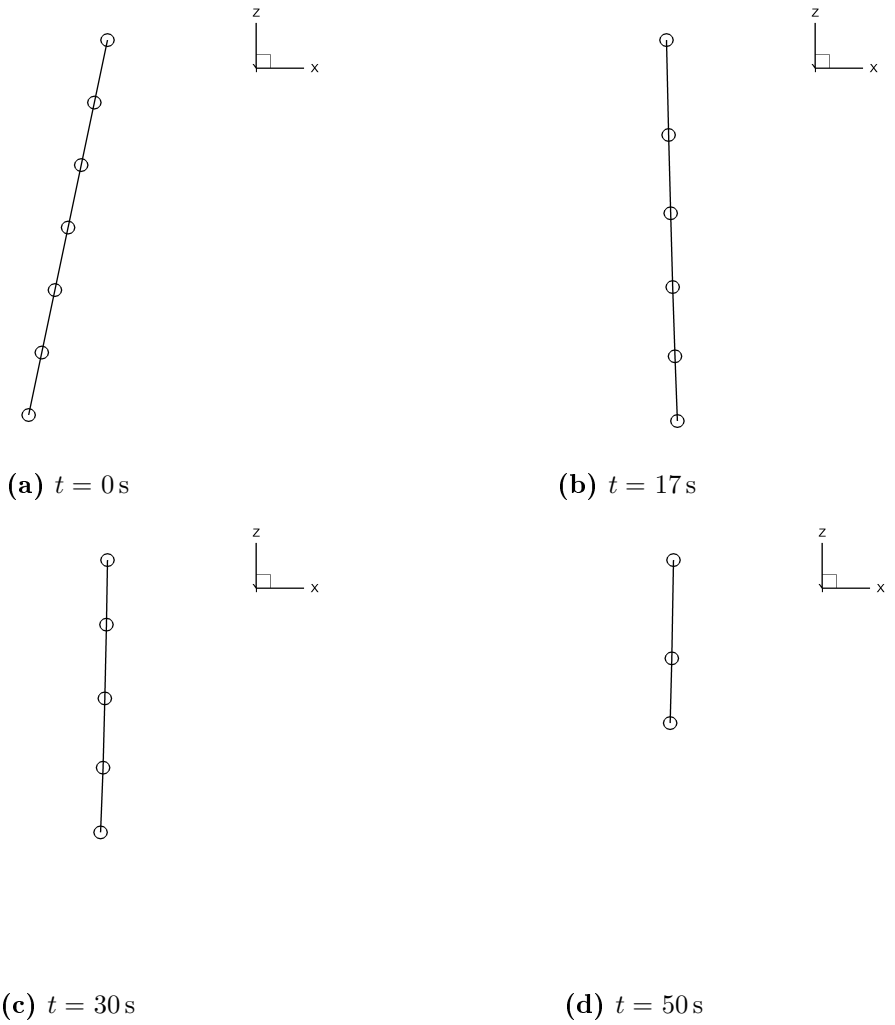


Figure II.20 – Angle between the topmost cable element and the vertical axis from numerical results of the *CRBA*



(a) $t = 0\,\mathrm{s}$ **(b)** $t = 17\,\mathrm{s}$

(c) $t = 30\,\mathrm{s}$ **(d)** $t = 50\,\mathrm{s}$

Figure II.21 – Cable profile during a lifting operation at different time steps

II.6 Conclusion

This chapter presented a review of the different theories to model cables. Two models were detailed and compared: the low-order lumped mass theory implementing in *CableDyn* and a multibody approach using the *CRBA* of *InWave*. These two theories model the stretching of the cable but neither the bending nor the torsion. The mass is discretized at the nodes of the cable. First, the low-order lumped mass theory has been validated with two test cases using numerical results and experimental data of the literature. A good agreement is observed. Then, a comparison was done between the two theories of cable dynamics and showed a perfect agreement. A cable joint has also been developed in the *CRBA* to speed-up the numerical simulations.

A state of the art of the winch models was exposed in this chapter. The cable element connecting to the winch becomes variable-unstretched length. The spatial discretization of the cable changes to ensure the elements are not too long or too short. This model may be used with both the low-order lumped mass theory and the multibody theory. Its implementation in the low-order lumped mass theory and the multibody theory gave the same results.

The next chapter presents the hydrodynamic theory used in this PhD work.

Chapter III

Hydrodynamics

Contents

The simulation of marine operations may involve the presence of two bodies subject to hydrodynamic loads which have a large relative amplitude motion. Then, the classical frequency-domain linear potential flow approach is not applicable anymore. After presenting a review of the hydrodynamic theories, this chapter details the potential flow theory based on the weak-scatterer hypothesis, used in the numerical tool WS_CN. The developments done during this PhD are exposed: the extension to multibody simulation, the free surface remeshing and the parallelization of the computation of the influence coefficients.

III.1 State of the art

III.1.1 Potential flow theory

Three hypotheses are done:

- The flow is irrotational (*H1*);
- The flow is incompressible (*H2*);
- The fluid is inviscid (*H3*).

The flows based on these assumptions are named the *potential flows*. Viscosity and turbulence are neglected. This approximation is valid in case of wave propagation for instance. When a body is present, viscous effects may be important, depending on the body shape. Viscous loads are less important in case of slender body shapes. Nevertheless, the potential flow theory is widely used, shows some good agreements in case of seakeeping, manoeuvring and wave propagation between the theory and the experiments and presents a good compromise between the accuracy and the CPU-time. This theory is applied here in hydrodynamics but it may be also used in aerodynamics [96].

III.1.1.1 Laplace's equation

(*H1*) involves:

$$\nabla \times \mathbf{v} = \mathbf{0}_{3\times 1} \tag{III.1}$$

where $\mathbf{v}$ is the fluid velocity.

Equation III.1 leads to:

$$\mathbf{v} = \nabla\phi \tag{III.2}$$

where ϕ is the velocity potential. Thus, only one scalar function is necessary to compute the three dimensional fluid velocity, reducing the number of unknowns.

The incompressibility of the flow (*H2*) allows to write:

$$\nabla \cdot \mathbf{v} = 0 \tag{III.3}$$

From (III.3), the Laplace's equation in the fluid domain $\mathcal{D}$ arises:

$$\Delta\phi = 0 \tag{III.4}$$

Consequently, the velocity potential is a harmonic function.

III.1.1.2 Bernoulli's equation

The Bernoulli's equation is derived from the momentum conservation using the three hypotheses of the potential flow theory. Its expression is:

$$p = -\rho\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}\nabla\phi \cdot \nabla\phi + gz\right) \tag{III.5}$$

where p denotes the total pressure, ρ the fluid density, g the gravity constant and z the vertical position. $\frac{\partial *}{\partial t}$ represents the Eulerian (or partial) derivative.

The last term of the right-hand side of (III.5) is the hydrostatic pressure:

$$p^{Static} = -\rho g z \tag{III.6}$$

III.1.1.3 Boundary conditions

Two types of boundaries have to be considered:

- Free surface;
- Solid surfaces (immersed or floating bodies, surfaces of the numerical tank walls and the sea bottom).

Free surface boundary conditions

The free surface is defined by the function η named the free surface elevation or wave elevation. It is assumed that η is a single-valued function. In this condition, the wave breaking cannot be modeled. The equation of the free surface is:

$$z = \eta(x, y, t) \tag{III.7}$$

The kinematic free surface boundary condition reflects that a particle on the free surface always remains part of the free surface:

$$\frac{\partial \eta}{\partial t} = -\nabla\phi \cdot \nabla\eta + \frac{\partial \phi}{\partial z} \quad \text{at } z = \eta(x, y, t) \tag{III.8}$$

The dynamic free surface boundary condition ensures the continuity of the pressure at the interface. The atmospheric pressure being taken equal to 0, it yields from the Bernoulli's equation:

$$\frac{\partial \phi}{\partial t} = -\frac{1}{2}\nabla\phi \cdot \nabla\phi - g\eta \quad \text{at } z = \eta(x, y, t) \tag{III.9}$$

Solid surface boundary condition

The inviscidness of the fluid (*H3*) involves a slip condition on solid boundaries of normal $\mathbf{n}$:

$$\mathbf{v} \cdot \mathbf{n} = \frac{\partial \phi}{\partial n} = \mathbf{v}^{Solid} \cdot \mathbf{n} \tag{III.10}$$

This body condition ensures the impermeability of the surface. The equality of the tangential components would lead to the creation of a boundary layer which is not consistent with the hypothesis of inviscidness.

III.1.1.4 Hydrodynamic loads

By integration of the total pressure over the instantaneous wetted surface of the body j, $S_{B_j}(t)$, of normal vector $\mathbf{n}$, pointing outwards the fluid domain, the hydrodynamic loads at the center of gravity $\mathbf{G_j}$ arise:

$$ {}^{e}\mathbf{F}_j^{Hydro} = \begin{pmatrix} \iint_{S_{B_j}(t)} p\mathbf{n}\, dS \\ \iint_{S_{B_j}(t)} p({}^{e}\mathbf{G}_j\mathbf{M}) \times \mathbf{n}\, dS \end{pmatrix} \tag{III.11} $$

The hydrostatic loads are obtained by the integration of the hydrostatic pressure (III.6):

$$ {}^{e}\mathbf{F}_j^{Static} = \begin{pmatrix} \iint_{S_{B_j}(t)} p^{Static}\mathbf{n}\, dS \\ \iint_{S_{B_j}(t)} p^{Static}({}^{e}\mathbf{G}_j\mathbf{M}) \times \mathbf{n}\, dS \end{pmatrix} \tag{III.12} $$

III.1.2 Numerical simulation of potential flows

Even if the potential flow approximation is the result of many assumptions, the problem stays complex and several nonlinearities are present:

- The free surface nonlinearities: the free surface boundary equations needs to be fulfilled on an unknown surface $z = \eta(x, y, t)$;
- The slip condition is applied on the instantaneous position of the bodies;
- The hydrodynamic loads are nonlinear and coupled with the unknown motion of the bodies.

Thus, several classes of potential flow models exist and their differences lie in the treatment of these nonlinearities:

- Fully nonlinear models;
- Weakly nonlinear models based on the weak-scatterer hypothesis;
- Body-exact models;
- Linear models.

III.1.2.1 Fully nonlinear models

When the potential flow theory presented above is directly applied, it leads to fully nonlinear models. The bodies and the free surface are meshed at their real position and the pressure is integrated over the instantaneous wetted surface as shown in Figure III.1. It is the most accurate method based on the potential flow approximation. It is widely used to simulate wave propagation [97].

But, the fluid-structure simulation makes the difficulties of the method appear. Indeed, the boundary conditions on the bodies and the free surface are expressed at the exact positions. Thus, the regridding of the mesh is a necessity: firstly, to update the position of the boundaries, and secondly, to keep a good quality mesh which is

deformed by the waves and the bodies. In case of surface-piercing bodies, sawtooth instabilities can appear and smoothing techniques are required. The computation of the intersection curves between the free surface and the bodies for any translational and rotational motion is a difficult task. To avoid the reflection of the perturbed waves at the numerical boundaries, artificial damping coefficients are used on the free surface. Wave breaking may occur in such a description of the free surface and causes the stop of the simulation. Finally, the space discretization of the mesh must be small enough to simulate all the perturbed waves generated by the presence of the bodies. All these characteristics make the fully nonlinear models cumbersome and numerically challenging, despite their accuracy.

Nevertheless, fully nonlinear models are subject of an important number of publications [98, 99]. To reduce the complexity of the fully nonlinear approach, other models have been developed.

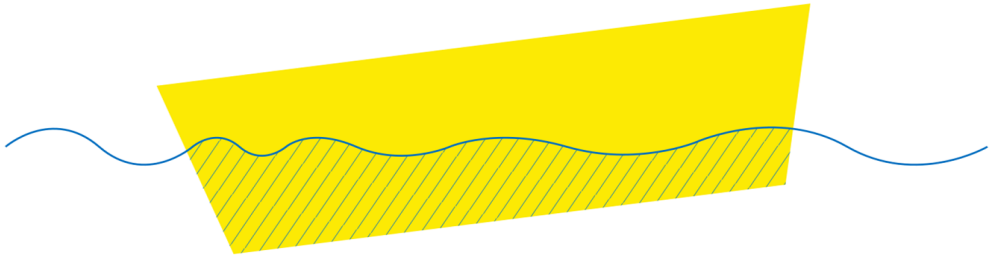


Figure III.1 – Fully nonlinear model of a floating body (yellow) in waves at their exact elevation (solid blue line) and its wetted surface (hatched area)

III.1.2.2 Weakly nonlinear models based on the weak-scatterer hypothesis

In most of the other models, both the velocity potential and the wave elevation are decomposed into an incident component and a perturbed component. This latter may also be decomposed into several elementary quantities as it will be presented further. For the moment:

$$\begin{cases} \phi &= \phi^I + \phi^P \\ \eta &= \eta^I + \eta^P \end{cases} \tag{III.13}$$

with:

- $(*)^I$ the incident component;
- $(*)^P$ the perturbed component.

The incident component is assumed to be known while the perturbed part is unknown. This decomposition allows to apply everywhere in the domain any kind of incident waves (regular or irregular). The incident wave does not have to be propagated, hence no wave maker model is necessary.

The weak-scatterer hypothesis assumes the perturbed quantities have to be small compared to the incident quantity:

$$\begin{cases} \phi^P \ll \phi^I \\ \eta^P \ll \eta^I \end{cases} \tag{III.14}$$

Doing so, the free surface boundary equations are linearized around the incident free surface elevation $z = \eta^I(x, y, t)$. The free surface nonlinearities are simplified. Contrary to the fully nonlinear approach, this surface is known. Thus, it is not necessary to mesh the perturbed waves, only the incident waves needs to be meshed. The pressure is integrated over the instantaneous wetted body surface delimited by the incident wave elevation as presented in Figure III.2. The weak-scatterer hypothesis involves the possibility to use a larger spatial discretization and therefore reduces the CPU-time. But the fulfilment of this hypothesis is required, which could limit the application of this theory. No assumption has been done about the body nonlinearities. As it will be presented in subsection III.1.3, this method is particularly adapted to slender surface-piercing bodies with or without forward speed and immersed bodies.

This model is the hydrodynamic theory used in this PhD work. More details and explanations about this approach are given in the rest of this chapter.

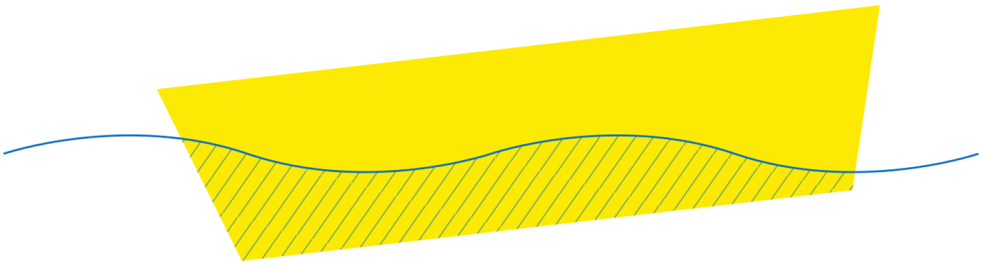


Figure III.2 – Weakly nonlinear model based on the weak-scatterer hypothesis of a floating body (yellow) in waves at their incident elevation (solid blue line) and its wetted surface (hatched area)

III.1.2.3 Body-exact models

A further simplification may be applied to the free surface conditions. These conditions can be linearized around the mean free surface elevation $z = 0$. Thus, the free surface mesh remains planar (Figure III.3), which enables a faster mesh convergence and the reduction of computing time. In that case, the pressure is integrated over the wetted body surface delimited by the mean wave elevation. This method is only consistent if small steepness waves are present. The wave steepness, ϵ, for a regular wave of amplitude A and wave number k, is defined by:

$$\epsilon = kA \tag{III.15}$$

This method is named the *body-exact approximation*. An example of such an approach is given in [100].

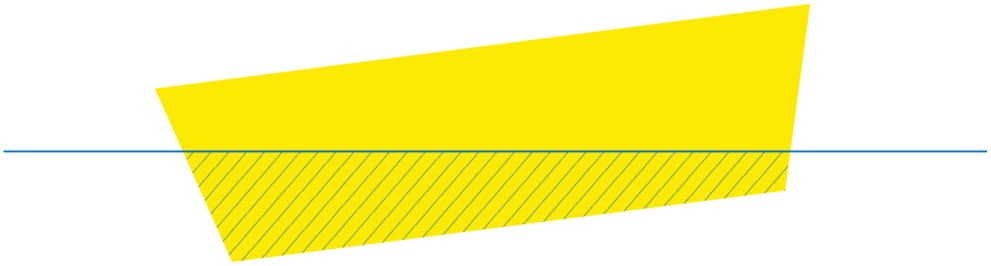


Figure III.3 – Body exact model of a floating body (yellow) in waves at their mean elevation (solid blue line) and its wetted surface (hatched area)

III.1.2.4 Linear models

In addition to the simplification of the free surface conditions, the body nonlinearities may be also simplified. To do so, bodies are assumed to have a small amplitude motion. Body meshes are fixed at the equilibrium position of the bodies (Figure III.4). It results a linear model. The perturbed components are expressed using a Stokes' series expansion based on the perturbation technique (here up to an order of magnitude of two):

$$\begin{cases} \phi^P &= \phi^{P(0)} + \epsilon\phi^{P(1)} + \epsilon^2\phi^{P(2)} + \mathcal{O}(\epsilon^3) \\ \eta^P &= \eta^{P(0)} + \epsilon\eta^{P(1)} + \epsilon^2\eta^{P(2)} + \mathcal{O}(\epsilon^3) \end{cases} \tag{III.16}$$

with $(*)^{P(j)}$ the j^{th} order perturbed component.

This decomposition is introduced in each equation of the potential flow theory. Terms of the same order of magnitude are collected, defining several problems of different order. Each one depends on the problems of smaller order of magnitude. The zeroth-order matches the hydrostatics. The first-order problem corresponds to the fully linear problem while the second-order problem bring quadratic terms. In these models, the body boundary conditions are linearized on the mean position of the bodies and the pressure is integrated over this fixed mean wetted body surface. Hence, the mesh of the bodies is fixed. The linearization of the free surface boundary conditions is also applied, leading to a totally still mesh. The CPU-time is much lower than for the aforementioned models. An example of such an approach for the first-order problem is also given in [100].

Figure III.4 – Linear model of a floating body (yellow) in waves at their mean elevation (solid blue line) and its wetted surface (hatched area)

Another decomposition is classically applied. Based on the superposition principle, the velocity potential is written:

$$\phi^P = \phi^D + \phi^R \tag{III.17}$$

ϕ^D represents the velocity potential of diffraction, when the body is fixed and the incoming wave is present. ϕ^R denotes the total radiation velocity potential, when the body is moving with a forced motion without incoming wave. This latter velocity potential is decomposed into elementary quantities:

$$\phi^R = \sum_{j=1}^{6} \phi_j^R \tag{III.18}$$

where ϕ_j^R is the velocity of radiation for the j^{th} degree of freedom. The elementary problems for these velocity potentials are solved and the global solution is obtained by superposition.

In order to add some nonlinearities, some works integrate the incident and hydrostatic pressure over the instantaneous wetted body surface delimited by the incident free surface elevation while the dynamic pressure is computed over the mean body surface. This is the so-called *nonlinear Froude-Krylov* approximation. The Froude-Krylov loads are defined by:

$${}^e\mathbf{F}_j^{FK} = \begin{pmatrix} \iint_{S_{B_j}(t)} p^I \mathbf{n}\, dS \\ \iint_{S_{B_j}(t)} p^I ({}^e\mathbf{G}_j \mathbf{M}) \times \mathbf{n}\, dS \end{pmatrix} \tag{III.19}$$

where p^I represents the pressure due to the incident velocity potential ϕ^I:

$$p^I = -\rho \left(\frac{\partial \phi^I}{\partial t} + \frac{1}{2} \nabla \phi^I \cdot \nabla \phi^I \right) \tag{III.20}$$

Although this method can show good results, it suffers from an inconsistency as the hydrodynamic loads are not assessed over the same surfaces. An example of application of this approach is given by Gilloteaux [101].

The linear models can also be solved in frequency domain. In this case, the body motion is assumed steady and all time-dependent quantities $f(x, y, z, t)$ are written as follows:

$$f(x, y, z, t) = \mathrm{Re}\left(\overline{f}(x, y, z) e^{-i\omega t}\right) \tag{III.21}$$

where i is the imaginary unit, ω the wave frequency and $\overline{f}$ the complex amplitude of f.

The radiation problems are considered for an harmonic motion and for each of the wave frequencies and directions of interest. Once these hydrodynamic problems are solved, a hydrodynamic database is created with the added-mass and damping coefficients and the exciting forces [47]. These frequency-dependent quantities serve to compute the impulse responses and then to achieve time-domain simulations using Cummin's equation [102] (cf. Appendix B).

Based on the Stokes' series expansion (III.16), a second-order problem may also be defined and leads to a new set of frequency-dependent coefficients, named the *quadratic transfer functions* or *QTF* [103]. Even if some extra physical phenomenon can be taken into account by this way (mean drift, resonance due to the low-frequency and high-frequency second-order wave loads), the assumptions of the theory always have to be fulfilled: small steepness waves and small amplitude body motion. The first-order and second-order frequency-domain models are among the most used hydrodynamic theories both in the industry and in the laboratories, for ship resistance, manoeuvrability, design

optimization as well as seakeeping of marine renewable energy devices. Examples of such solvers are: *Nemoh*[1] [47], *WAMIT*[2], *Diodore*[3], *HydroSTAR*[4], *ANSYS Aqwa*[5], etc.

The small amplitude body motion condition may often be very restrictive especially in multibody simulations. This is why some researchers working with a frequency-domain model update their hydrodynamic database during their simulation when the relative motion between the bodies becomes too important. Nevertheless, such an approach is inconsistent due to the retardation functions, taking into account the flow memory, which are not computed at the same position anymore [104].

III.1.2.5 Selected model

Table III.1 summarizes the main differences between the aforementioned potential flow theories.

As highlighted in subsection 4.2, so far, only three approaches have been used to compute the hydrodynamic loads in marine operation simulations: a frequency-domain model [7, 16, 25], a linear time-domain model [26] and a fully nonlinear potential flow model [105]. The hydrodynamic theory used in this PhD is a weakly nonlinear model based on the weak-scatterer hypothesis. This approach is consistent with the simulation of marine operations since they often involve several bodies with, possibly, a large relative motion. Therefore, the full linearization of the body condition made in a linear potential-flow based solver is likely too restrictive for this application. Moreover, a frequency-domain model assumes a steady state motion. Using a weakly nonlinear theory, no hypothesis is done about the body motion amplitude, the unsteadiness of the flow and the wave steepness. Compared to a fully nonlinear approach, it is expected to lead to more stable simulations. But some nonlinear effects (wave drift, hydrodynamic impact) may be underestimated. No comparison has been made between the weakly nonlinear model based on the weak-scatterer hypothesis and a classical frequency-domain model in the case of a marine operation. Thus, it is of interest to compare them and quantify their differences.

[1] https://lheea.ec-nantes.fr/logiciels-et-brevets/nemoh-presentation-192863.kjsp
[2] https://www.wamit.com
[3] http://www.principia-group.com/blog/product/diodore
[4] https://www.veristar.com/portal/veristarinfo/detail/software/SeakeepingandMooringAnalysis/HYDROSTAR/Hydros
[5] https://www.ansys.com/fr-fr/products/structures/ansys-aqwa

	NL	WSC	BE	NFK	LTD	FD
Decomposition	ϕ or $\phi = \phi^I + \phi^P$	$\phi = \phi^I + \phi^P$	$\phi = \phi^I + \phi^P$	$\phi = \phi^I + \phi^D + \phi^R$	$\phi = \phi^I + \phi^P$ or $\phi = \phi^I + \phi^D + \phi^R$	$\phi = \phi^I + \phi^D + \phi^R$
Assumptions	ø	$\phi^P \ll \phi^I$	$\epsilon \ll 1$	$\begin{cases} \epsilon \ll 1 \\ A_m \ll 1 \end{cases}$	$\begin{cases} \epsilon \ll 1 \\ A_m \ll 1 \end{cases}$	$\begin{cases} \epsilon \ll 1 \\ A_m \ll 1 \\ \text{Steady state} \end{cases}$
Meshed free surface	$z = \eta(x, y, t)$	$z = \eta^I(x, y, t)$	$z = 0$	$\begin{cases} z = \eta^I(x, y, t) & \text{for HS and FK;} \\ z = 0 & \text{otherwise.} \end{cases}$	$z = 0$	$z = 0$
Mesh body surface	$S_B(t)$	$S_B(t)$	$S_B(t)$	$\begin{cases} S_B(t) & \text{for HS and FK;} \\ S_B(0) & \text{otherwise.} \end{cases}$	$S_B(0)$	$S_B(0)$
Time	$\mathcal{O}$(1 week)	$\mathcal{O}$(Days)	$\mathcal{O}$(1 day)	$\mathcal{O}$(1 h)	$\mathcal{O}$(1 h)	$\mathcal{O}$(1 h)

Table III.1 – Main differences between several potential flow theories. NL: Fully nonlinear, WSC: Weakly nonlinear based on the weak-scatterer hypothesis, BE: Body exact approximation, NFK: Nonlinear Froude-Krylov, LTD: Linear time-domain, FD: Frequency-domain. A_m denotes the body motion amplitude. HS and FK represent the hydrostatic and Froude-Krylov loads.

III.1.3 State of the art of the potential flow theory based on the weak-scatterer hypothesis

The weak-scatterer hypothesis was introduced by Pawlowski and Bass [106] in 1991 to perform time-domain computations of large ship motions in heavy seas. They described their assumption by the words:

> "The disturbance induced by the moving ship in the wave flow is considered to be of smaller magnitude than the wave flow quantities which are proportional to the wave height, but at least of the same magnitude as the wave flow quantities proportional to the square of the wave height. This assumption, explained here in simple terms, is called the weak scatterer hypothesis."

Thus, the scattered wave components are assumed much smaller than the incident wave components. The scope of application of this hypothesis is [106]:

> "In the theoretical model described [...], the oncoming wave which interacts with the ship, is assumed to be high and steep, so that the quantities proportional to the square of its height cannot be neglected. The motions of the ship, induced by the wave, are also considered to be large, of a magnitude proportional to the wave height."

This theory, according to Pawlowski and Bass [106], should be applied in case of:

- Large body motions;
- Steep waves.

If the weak-scatterer hypothesis is not fulfilled [107]:

> "[...] this theory still gives a non-linear time domain formulation of the scattering problem, albeit not consistent in its non-linear part. In other words, the applicability of the weak scatterer hypothesis is necessary to construct a consistent non-linear scattering theory in which the linear free surface condition is used."

This method was applied by Pawlowski in [106, 107, 108]. Comparisons were made for a Series 60 and a trawler between numerical results and experimental data in small amplitude and steep regular waves. A good agreement was observed which confirmed the validity of the hypothesis. Only few research studies have been published so far.

Lin et al. [109] developed a time-domain solver based on the weak-scatterer hypothesis, called *LAMP-4*. This numerical tool is part of the program LAMP (*Large Amplitude Motions Program*) for the computation of the motions and loads of a ship operating in extreme sea conditions. It exists four versions of the solver LAMP [110]:

- *LAMP-1* follows a linear approach;
- *LAMP-2* is based on a nonlinear Froude-Krylov model;
- *LAMP-3* is similar to *LAMP-2* but allows large lateral displacements of the ship;
- *LAMP-4* applies the weak-scatterer method.

Kring et al. [111] presented another multi-level potential flow based numerical tools: *SWAN*, short of *Ship Wave ANanlysis*, developed at the Massachusetts Institute of Technology. It includes:

- *SWAN-1*: a linear frequency-domain code;
- *SWAN-2*: a linear time-domain code with linear or nonlinear Froude-Krylov loads;
- *SWAN-4*: a time-domain code based on the weak-scatterer hypothesis.

SWAN-4 was used with a Series 60 and a Snowdrift moving in head seas and compared to *SWAN-2*. Both the linear theory and the weak-scatterer approach were in good agreement with the experimental data. The weak-scatterer simulations only brought a small correction [111]. The extension of *SWAN* to the weak-scatterer hypothesis was the PhD work of Huang [112]. In his thesis, other comparisons were made between the linear and weak-scatterer theories with a containership in head seas. A significant improvement was noticed using *SWAN-4* compared to *SWAN-2* especially at the resonance. Moreover, for small wave slopes, the weak-scatterer method converged to the linear theory results. But, the steeper the incoming waves were, the larger the differences between the weak-scatterer method and the linear theory were.

SWAN-1, *SWAN-2* and *SWAN-4* were compared by Grigoropoulos et al. [113] to experimental results for several kinds of ships: a Series 60, a reefer and a ROPAX. Both *SWAN-1* and *SWAN-2* gave robust numerical results for every test cases, the use of nonlinear Froude-Krylov loads added accuracy to the predictions. But, *SWAN-4* gave some unreasonable results which could be very different from the experiments. No explanations were given of these surprising results.

A third numerical tool, named *WISH* (Wave-Induced loads and Ship motion), of the Seoul National University, uses the weak-scatterer hypothesis. As for *LAMP* and *SWAN*, *WISH* exists with several hydrodynamic theories [114]:

- *WISH-1*: a linear time-domain code;
- *WISH-2*: a linear time-domain code with nonlinear Froude-Krylov loads;
- *WISH-3*: a time-domain code based on the weak-scatterer hypothesis.

Comparisons between *SWAN-4*, *WISH-3* and experimental data were achieved in [115] for a Series 60 and a containership in head seas and showed a good agreement albeit the formulations were slightly different. Other comparisons to experimental data and the three versions of *WISH* in case of a containership in head seas showed an overall good agreement of every method. Best agreements were obtained using the weak-scatterer formulation, especially in rough seas. *WISH* has been coupled to a flexible-body mechanical solver, giving *WISH-FLEX*.

The weak-scatterer theory has only been applied to a non-ship like body by Bretl [116]. His PhD work focused on the study of a wave energy converter made of a floating rigid hemisphere coupled with a planar pendulum.

To sum up, it stems from this state of the art that the weak-scatterer approach has been mainly applied to the computation of the motions and the hydrodynamic loads of a moving slender ship in steep head regular waves. Large deformations of the mesh, for instance due to a lateral motion, have not been encountered. Irregular trains of waves

have not been tested either. Since 2012, a potential flow based solver following the weak-scatterer hypothesis has been under development at *LHEEA* laboratory of *Ecole Centrale de Nantes*. It is named *WS_ CN*.

Letournel [44] developed a first version of *WS_ CN* dealing with a single submerged body with translational motions. Several validation cases were presented, all of them consider a submerged sphere. First, the weak-scatterer theory was compared to the linear approach in the case of the diffraction of regular waves by a fixed sphere. Small steepness regular waves involved the matching of the hydrodynamic loads from the two theories for every frequency, whereas high steepness regular waves caused important differences. Then, radiation cases due to a heave harmonic motion of the sphere were treated. The weak-scatterer approach was compared to several other theories: linear, body-exact and fully nonlinear. A good agreement between all theories was observed for the harmonic coefficients in case of a small amplitude motion. Afterwards, three decay tests were displayed with different initial positions. The motion of the sphere from the linear and weak-scatterer approaches matched in every scenario. These results constituted the first validation of the weak-scatterer approach compared to other fluid potential theory for a non-ship like body. An application was performed by Letournel with *CETO*-type wave energy converter with a spring-damper power-take-off. Numerical comparisons were achieved using *NEMOH* and *WS_ CN*. Once again, the two models showed a good agreement for small amplitude motions and small steepness waves, but some differences appeared otherwise. The mean absorbed power by the *WEC* was modified in this case. Another submerged *WEC* has been simulated using the weak-scatterer approach: an oscillating wave surge converter or wave roller-type *WEC* [117]. The same conclusions were obtained as with the *CETO*-type *WEC*. Letournel also showed the weak-scatterer hypothesis could be not verified punctually when the heaving sphere got close to the free surface.

Chauvigné [45] extended *WS_ CN* to a single surface-piercing body with an arbitrary motion. An advance front method was used to generate the total mesh or regenerate the body mesh. Two mesh morphing methods were also adopted to avoid the remeshing in case of small deformations: a linear spring analogy on the body surface and an interpolation scheme based on a radial basis function on the free surface. Several validation cases were presented by Chauvigné. The diffraction of a regular wave by either a bottom-fixed or truncated surface-piercing vertical cylinder was studied. Comparisons were made with respect to linear and fully nonlinear theories and experimental data for the maximum run-up around the cylinder. This case was highly nonlinear, such as a linear approach was not accurate enough. In the two configurations (fixed-bottom or truncated cylinder), the weak-scatterer model showed a good agreement with the fully nonlinear theory and experimental data on the bow side of the cylinder but overestimated the maximum run-up on the lee side, probably due to important free surface nonlinearities. Then the radiation of a truncated cylinder in surge or pitch motion was evaluated and compared to a nonlinear approach. In both cases, a very good agreement between the two approaches was observed in the time series of the maximum run-up. In details, the relative error on the first harmonic was very low but larger discrepancies were observed for higher harmonics. A free motion validation test with a truncated cylinder linked to its equilibrium position by a spring was presented. As with Letournel [44], numerical results from *WS_ CN* matched the linear theory in case of small amplitude regular wave but differences appear with a higher amplitude wave. Finally,

an application with the *WaveStar* wave energy converter was featured. Comparisons were done with the offshore numerical tool *DeepLines* and the experimental data. The numerical results with the weak-scatterer method for a diffraction case were conclusive and matched *DeepLines* and the experimental measures. But, the free motion of the *WaveStar* device was not accurately simulated by the two numerical tools due to an improper modeling of the power take off. Nevertheless, the linear and weak-scatterer results matched again. As explained by Chauvigné [45], the weak-scatterer hypothesis was not always validated in this latter application. The disturbed wave elevation could be roughly of the same order of magnitude as the incident component.

These two PhDs have brought the weak-scatterer theory in a new direction: the simulation of a single submerged or floating wave energy converter. Intensive comparisons with the linear and nonlinear potential flow theories were achieved and demonstrated the capacity of the method to perform seakeeping computations. As said by Huang [112], non-slender body shapes may induce important wave perturbations and this phenomenon was observed in [44, 45], leading to an unvalidated weak-scatterer hypothesis. Regarding the computation time, Letournel et al. [117] found the weak-scatterer approach is one order of magnitude faster than a fully nonlinear approach and two orders of magnitude greater than the real time (in sequential calculations). It appears from these works that the weak-scatterer theory has been compared to a body-exact approach only once. Yet, this latter method could be efficient if the steepness of the waves is not too large. In the case of still water, the two theories match.

Regarding the CPU-time reduction, several types of symmetries were implemented by Letournel et al. [117] in *WS_CN*: vertical along the (xOz) plane and horizontal on the flat sea bottom. Simulations in open domain are also possible when the lateral numerical tank surfaces are far enough. Chauvigné [45] have allowed a partial calculation of the influence coefficients when sub-domains are motionless.

At the beginning of this PhD work, the simulations using *WS_CN* dealt with a single submerged or floating body in sequential computations. Surface-piercing body simulations have been performed by Chauvigné [45] but only with a small amplitude motion of the body and so with only small deformations of the free surface mesh. Therefore, simulations with a large deformation of the free surface mesh were not possible, as well as multibody simulations.

Note

In the literature, the term *weakly nonlinear* covers a variety of situations. For example, in [114, 115] and [118], the so-called weakly nonlinear approach matched the nonlinear Froude-Krylov approach. Greco and Lugni [119] defined their potential flow based solver relied on the weak-scatterer hypothesis as weakly nonlinear but their hydrodynamic loads were estimated as linear in frequency domain while the Froude-Krylov and hydrostatic loads were nonlinear. Ruggeri et al. [120] presented some weakly nonlinear computations but their time-domain solver was fully linear using a high-order panel method and computing second-order mean drift forces. Thus, the expression *weakly nonlinear* includes different types of approximations and is definitely unclear. In this PhD thesis, following the denominations of Letournel et al. [117] and Chauvigné [45], there is an equivalence between the terms *weak-scatterer* and *weakly nonlinear*.

III.2 Theory of the potential flow based on the weak-scatterer hypothesis

III.2.1 Assumptions and governing equations

The hydrodynamic theory used in this PhD assumes:

- A tri-dimensional, unsteady and potential flow;
- A flat sea bottom;
- A single-valued free surface elevation;
- The unawareness of the surface tension;
- The knowledge of the incoming wave field;
- The nullity of the pressure above the free surface;
- The rigidity of the floating and immersed bodies;
- The validity of the weak-scatterer hypothesis.

The fluid domain, $\mathcal{D}$, has a boundary, $\mathcal{S}$, constituted of the free surface FS, the wetted body surfaces S_B and the numerical tank surfaces (including the sea bottom) T. The reference frame is Σ_e, as defined in subsection I.2.2. The number of hydrodynamic bodies is N^{WSC}.

From the assumptions, the potential flow theory is applied. The governing equations have already been demonstrated:

- (III.4) in the flow;
- (III.8) and (III.9) for the free surface;
- (III.10) for the solid boundaries (bodies and sea bottom);
- A permeability condition on the numerical tank walls:

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi^I}{\partial n} \Rightarrow \frac{\partial \phi^P}{\partial n} = 0 \tag{III.22}$$

- A radiation condition, the wave field far from the bodies matches the incident wave field:

$$\begin{cases} \phi \xrightarrow[r \to +\infty]{} \phi^I \\ \eta \xrightarrow[r \to +\infty]{} \eta^I \end{cases} \tag{III.23}$$

The use of the decomposition (III.13) gives:

$$\begin{cases} \Delta\phi^P = -\Delta\phi^I \quad \text{in the fluid domain } \mathcal{D} \\ \dfrac{\partial\eta^P}{\partial t} = -\dfrac{\partial\eta^I}{\partial t} - \nabla\phi^I\cdot\nabla\eta^I - \nabla\phi^I\cdot\nabla\eta^P - \nabla\phi^P\cdot\nabla\eta^I - \nabla\phi^P\cdot\nabla\eta^P \\ +\dfrac{\partial\phi^I}{\partial z} + \dfrac{\partial\phi^P}{\partial z} \quad \text{at } z=\eta(x,y,t) \\ \dfrac{\partial\phi^P}{\partial t} = -\dfrac{\partial\phi^I}{\partial t} - \dfrac{1}{2}\nabla\phi^I\cdot\nabla\phi^I - \nabla\phi^I\cdot\nabla\phi^P - \dfrac{1}{2}\nabla\phi^P\cdot\nabla\phi^P - g(\eta^I+\eta^P) \quad \text{at } z=\eta(x,y,t) \\ \dfrac{\partial\phi^P}{\partial n} = -\dfrac{\partial\phi^I}{\partial n} + \mathbf{v}^{Solid}\cdot\mathbf{n} \quad \text{on } S_B(t) \text{ and the seabed} \\ \dfrac{\partial\phi^P}{\partial n} = 0 \quad \text{on the numerical tank walls} \\ \phi^P \xrightarrow[r\to+\infty]{} 0 \\ \eta^P \xrightarrow[r\to+\infty]{} 0 \end{cases} \tag{III.24}$$

The incoming wave field is computed following a potential flow approximation, consequently:

$$\Delta\phi^I = 0 \quad \text{in the fluid domain } \mathcal{D} \tag{III.25}$$

The weak-scatterer approximation enables the linearization of the free surface boundary equations at the incident wave elevation. By application of the Taylor's theorem for a function f at the point $z=\eta^I(x,y,t)$, it yields:

$$f(x,y,z,t)\Big|_{z=\eta(x,y,t)} = f(x,y,z,t)\Big|_{z=\eta^I(x,y,t)} + (\eta-\eta^I)\frac{\partial f(x,y,z,t)}{\partial z}\Big|_{z=\eta^I(x,y,t)} + \mathcal{O}(\phi^{P2}) + \mathcal{O}(\eta^{P2}) \tag{III.26}$$

Equation III.26 is used with the free surface boundary equations of (III.24) and, after applying the weak-scatterer hypothesis (III.14):

$$\begin{cases} \dfrac{\partial\eta^P}{\partial t} = -\dfrac{\partial\eta^I}{\partial t} - \nabla\phi^I\cdot\nabla\eta^I - \nabla\phi^I\cdot\nabla\eta^P - \nabla\phi^P\cdot\nabla\eta^I + \dfrac{\partial\phi^I}{\partial z} + \dfrac{\partial\phi^P}{\partial z} \\ +\eta^P\left(\dfrac{\partial^2\phi^I}{\partial z^2} - \nabla\eta^I\cdot\dfrac{\partial\nabla\phi^I}{\partial z}\right) \quad \text{at } z=\eta^I(x,y,t) \\ \dfrac{\partial\phi^P}{\partial t} = -\dfrac{\partial\phi^I}{\partial t} - \dfrac{1}{2}\nabla\phi^I\cdot\nabla\phi^I - \nabla\phi^I\cdot\nabla\phi^P - g(\eta^I+\eta^P) \\ -\eta^P\left(\dfrac{\partial^2\phi^I}{\partial z\partial t} + \nabla\phi^I\cdot\dfrac{\partial\nabla\phi^I}{\partial z}\right) \quad \text{at } z=\eta^I(x,y,t) \end{cases} \tag{III.27}$$

III.2.2 Arbitrary Lagrangian-Eulerian description

The solving of the fluid governing equations involves the use of a computational mesh. There are three possible descriptions of the motion in a continuum media:

- The Lagrangian description: the nodes of the mesh follow the material points, so that the boundaries can be easily followed in case of small deformations. If large deformations occur, the quality of the panels geometry is not guaranteed and the mesh has to be regenerated. This approach is mainly used in structural mechanics or in the Smooth Particle Hydrodynamics method [121] for instance.

- The Eulerian description: the nodes of the mesh are fixed. This approach is easy to implement, robust and large deformations of the mesh may be applied. The mesh quality is preserved, but tracking the boundaries of the domain requires a fine mesh.
- The arbitrary Lagrangian-Eulerian (ALE) description combines the better characteristics of the two previous approaches. The boundary equations are solved at the material nodes taking into account the arbitrary motion of the mesh and ensuring its good quality. This approach is preferred in case of free surface flows.

In order to track the physical quantities at the nodes of the mesh under an arbitrary motion, the ALE description is preferred. Thus, the free surface boundary equations need to be updated at the nodes of the mesh moving with an arbitrary motion.

Let $\Omega_a(t)$ be an arbitrary control volume moving at the velocity $\mathbf{v}_a$. Then the derivative of a quantity $\mathbf{f}$ of $\Omega_a(t)$ is:

$$\frac{\delta \mathbf{f}}{\delta t} = \frac{\partial \mathbf{f}}{\partial t} + (\mathbf{v}_a \cdot \nabla)\mathbf{f} \tag{III.28}$$

where $\dfrac{\delta *}{\delta t}$ represents the derivative with respect to an ALE description.

In our case, $\mathbf{v}_a$ is the velocity of the nodes of the mesh, denoted $\mathbf{v}_{mesh}$. The free surface boundary equations become:

$$\begin{cases} \dfrac{\delta \eta^P}{\delta t} = -\dfrac{\partial \eta^I}{\partial t} - \nabla\phi^I \cdot \nabla\eta^I - (\nabla\phi^I - \mathbf{v}_{mesh}) \cdot \nabla\eta^P - \nabla\phi^P \cdot \nabla\eta^I + \dfrac{\partial \phi^I}{\partial z} + \dfrac{\partial \phi^P}{\partial z} \\ +\eta^P \left(\dfrac{\partial^2 \phi^I}{\partial z^2} - \nabla\eta^I \cdot \dfrac{\partial \nabla\phi^I}{\partial z} \right) \quad \text{at } z = \eta^I(x,y,t) \\ \dfrac{\delta \phi^P}{\delta t} = -\dfrac{\partial \phi^I}{\partial t} - \dfrac{1}{2}\nabla\phi^I \cdot \nabla\phi^I - (\nabla\phi^I - \mathbf{v}_{mesh}) \cdot \nabla\phi^P - g(\eta^I + \eta^P) \\ -\eta^P \left(\dfrac{\partial^2 \phi^I}{\partial z \partial t} + \nabla\phi^I \cdot \dfrac{\partial \nabla\phi^I}{\partial z} \right) \quad \text{at } z = \eta^I(x,y,t) \end{cases} \tag{III.29}$$

III.2.3 The boundary value problem

Eventually, the boundary value problem (BVP) to solve is:

$$\begin{cases} \Delta\phi^P = 0 \quad \text{in the fluid domain } \mathcal{D} \\ \dfrac{\delta\eta^P}{\delta t} = -\dfrac{\partial\eta^I}{\partial t} - \nabla\phi^I \cdot \nabla\eta^I - (\nabla\phi^I - \mathbf{v}_{mesh}) \cdot \nabla\eta^P - \nabla\phi^P \cdot \nabla\eta^I + \dfrac{\partial\phi^I}{\partial z} + \dfrac{\partial\phi^P}{\partial z} \\ +\eta^P \left(\dfrac{\partial^2\phi^I}{\partial z^2} - \nabla\eta^I \cdot \dfrac{\partial\nabla\phi^I}{\partial z}) \right) \quad \text{at } z = \eta^I(x,y,t) \\ \dfrac{\delta\phi^P}{\delta t} = -\dfrac{\partial\phi^I}{\partial t} - \dfrac{1}{2}\nabla\phi^I \cdot \nabla\phi^I - (\nabla\phi^I - \mathbf{v}_{mesh}) \cdot \nabla\phi^P - g(\eta^I + \eta^P) \\ -\eta^P \left(\dfrac{\partial^2\phi^I}{\partial z \partial t} + \nabla\phi^I \cdot \dfrac{\partial\nabla\phi^I}{\partial z} \right) \quad \text{at } z = \eta^I(x,y,t) \\ \dfrac{\partial\phi^P}{\partial n} = -\dfrac{\partial\phi^I}{\partial n} + \mathbf{v}^{Solid} \cdot \mathbf{n} \quad \text{on } S_B(t) \text{ and the seabed} \\ \dfrac{\partial\phi^P}{\partial n} = 0 \quad \text{on the numerical tank walls} \\ \phi^P \xrightarrow[r\to+\infty]{} 0 \\ \eta^P \xrightarrow[r\to+\infty]{} 0 \end{cases} \tag{III.30}$$

Different methods exist to solve this problem, for instance:

- The finite-difference method;
- The finite-element method;
- The boundary element method (BEM).

Apart from their different formulations, these methods mainly differ on the mesh requirement and the sparsity of the linear systems they involve. The finite-difference method and the finite-element method require volume meshes whereas the boundary element methods need surface meshes and so less unknowns and smaller linear systems. But, these linear systems are dense in case of BEM and sparse otherwise. Sparse matrices require less memory to store their coefficients and efficient algorithms exist to solve large sparse linear systems. Nevertheless, the easier mesh management offered by the BEM makes its method widely used. *WS_CN* is based on the boundary element method.

III.2.4 The boundary element method

III.2.4.1 The boundary integral equation

Note

So far the normal unit vector $\mathbf{n}$ pointed *outward* the fluid, from now on, the normal unit vector points *inward* the fluid.

The boundary element method is based on the Green's second identity:

Theorem 1. *Let $\Omega(t)$ be a volume of $\mathbb{R}^3$, $\partial\Omega(t)$ its boundary and $\mathbf{n}$ the inward unit normal to this boundary. Let dV and dS be an elementary volume and surface of $\Omega(t)$*

and $\partial\Omega(t)$, respectively. Let f and g be both twice continuously differentiable scalar functions. Then:

$$\iiint_{\Omega(t)} (f\Delta g - g\Delta f)\, dV = -\iint_{\partial\Omega(t)} \left(f\frac{\partial g}{\partial n} - g\frac{\partial f}{\partial n} \right) dS \tag{III.31}$$

where $\dfrac{\partial f}{\partial n}$ represents the normal derivative:

$$\frac{\partial f}{\partial n} = \nabla f \cdot \mathbf{n} = f_n \tag{III.32}$$

The functions f and g are chosen such as:

$$\begin{cases} f = \phi^P \\ g = G \end{cases} \tag{III.33}$$

where G represents the Green's function, solution of the equation:

$$\Delta G(\mathbf{x}_i, \mathbf{x}_j) = \delta(\mathbf{x}_i - \mathbf{x}_j) \tag{III.34}$$

δ denotes the Dirac delta function, $\mathbf{x}_i$ and $\mathbf{x}_j$ are two points of $\mathbb{R}^3$.

Basically, it exists two ways to find the Green's function. Either G only satisfies (III.34) and becomes a *Rankine source Green's function* or G also satisfies the free surface boundary equations and the radiation condition and becomes a *Kelvin source Green's function* or *free surface Green's function*. In *WS_ CN*, a Rankine source Green's function is used, that is to say:

$$G(\mathbf{x}_i, \mathbf{x}_j) = -\frac{1}{4\pi\|\mathbf{x}_i - \mathbf{x}_j\|} = -\frac{1}{4\pi\|\mathbf{r}_{ij}\|} \tag{III.35}$$

In *WS_ CN*, with no loss of generality, the Green's function is simply defined by:

$$G(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\|\mathbf{r}_{ij}\|} \tag{III.36}$$

Its normal derivative is:

$$\frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j) = -\frac{\mathbf{r}_{ij} \cdot \mathbf{n}_j}{\|\mathbf{r}_{ij}\|^3} \tag{III.37}$$

$\dfrac{\partial *}{\partial n_j}$ is the normal derivative with respect to $\mathbf{n}_j$.

Applying (III.31) over the whole fluid domain $\mathcal{D}$ bounded by the surface $\mathcal{S}$ at the point $\mathbf{x}_i$ gives:

$$\iiint_{\mathcal{D}} \left(\phi^P(\mathbf{x}_j)\Delta G(\mathbf{x}_i, \mathbf{x}_j) - G(\mathbf{x}_i, \mathbf{x}_j)\Delta\phi^P(\mathbf{x}_j)\right) dV = -\iint_{\mathcal{S}} \left(\phi^P(\mathbf{x}_j)\frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j) - G(\mathbf{x}_i, \mathbf{x}_j)\frac{\partial \phi^P}{\partial n}(\mathbf{x}_j) \right) dS \tag{III.38}$$

Here $\mathbf{x}_j$ is the variable of integration.

The integral of the Laplacian of the Green's function may be expressed by:

$$\iiint_{\mathcal{D}} \Delta G(\mathbf{x}_i, \mathbf{x}_j)\, dV = \iiint_{\mathcal{D}} \Delta\left(\frac{1}{\|\mathbf{r}\|}\right) dV \tag{III.39}$$

$$= \iiint_{\mathcal{D}} \nabla \cdot \left(-\frac{\mathbf{r}}{\|\mathbf{r}\|^3}\right) dV \tag{III.40}$$

$$= \iint_{\mathcal{S}} \frac{\mathbf{r}\cdot\mathbf{n}}{\|\mathbf{r}\|^3}\, dS \tag{III.41}$$

$$= -\iint_{\mathcal{S}} \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS \tag{III.42}$$

$$= \Omega(\mathbf{x}_i) \tag{III.43}$$

$\Omega(\mathbf{x}_i)$ may be seen as the solid angle at the point $\mathbf{x}_i$. The generalization of this latter expression yields:

$$\iiint_{\mathcal{D}} \phi^P(\mathbf{x}_j)\Delta G(\mathbf{x}_i, \mathbf{x}_j)\, dV = -\phi^P(\mathbf{x}_i) \iint_{\mathcal{S}} \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS = \phi^P(\mathbf{x}_i)\Omega(\mathbf{x}_i) \tag{III.44}$$

The first term of the left hand side of (III.38) is zeroed:

$$\Delta\phi^P(\mathbf{x}_j) = 0 \text{ in } \mathcal{D} \Rightarrow \iiint_{\mathcal{D}} G(\mathbf{x}_i, \mathbf{x}_j)\Delta\phi^P(\mathbf{x}_j)\, dV = 0 \tag{III.45}$$

Finally (III.38) becomes:

$$-\phi^P(\mathbf{x}_i)\Omega(\mathbf{x}_i) - \iint_{\mathcal{S}} \phi^P(\mathbf{x}_j)\frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS + \iint_{\mathcal{S}} G(\mathbf{x}_i, \mathbf{x}_j)\frac{\partial \phi^P}{\partial n}(\mathbf{x}_j)\, dS = 0 \tag{III.46}$$

Equation III.46 is the boundary integral equation (*BIE*) in the fluid. Only two surface integrals are present so the computational domain is the boundary $\mathcal{S}$ of the fluid domain $\mathcal{D}$: the free surface FS, the wetted body surfaces S_B and the numerical tank surfaces (including the sea bottom) T:

$$\mathcal{S} = FS \cup S_B \cup T \tag{III.47}$$

Where S_B is composed of the wetted surface S_{B_j} of each body j:

$$S_B = \bigcup_{j=1}^{N^{WSC}} S_{B_j} \tag{III.48}$$

Two terms of (III.46) need to be clarified:

$$\iint_{\mathcal{S}} \phi^P(\mathbf{x}_j)\frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS \quad \text{and} \quad \iint_{\mathcal{S}} G(\mathbf{x}_i, \mathbf{x}_j)\frac{\partial \phi^P}{\partial n}(\mathbf{x}_j)\, dS \tag{III.49}$$

III.2.4.2 The influence coefficients

The boundary integral equation is solved using the collocation method. The boundary surface of the fluid domain is discretized into N_p panels of surface $\mathcal{S}_p$ and N nodes:

$$\mathcal{S} = \bigcup_{p=1}^{N_p} \mathcal{S}_p \tag{III.50}$$

The integral equation is written at each node i of the mesh:

$$-\phi^P(\mathbf{x}_i)\Omega(\mathbf{x}_i) - \sum_{p=1}^{N_p} \iint_{\mathcal{S}_p} \phi^P(\mathbf{x}_j) \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS + \sum_{p=1}^{N_p} \iint_{\mathcal{S}_p} G(\mathbf{x}_i, \mathbf{x}_j) \frac{\partial \phi^P}{\partial n}(\mathbf{x}_j)\, dS = 0 \tag{III.51}$$

In *WS_ CN*, panels are triangular. This ensures their flatness and it is also a common mesh format. Each triangle p is defined by its three vertices $(\mathbf{x}_1^p, \mathbf{x}_2^p, \mathbf{x}_3^p)$. Its centre of gravity is at $\mathbf{x}_G^p$. Its contour is denoted $\mathcal{C}_p$. The unknowns are at the nodes. The computation of the surface integrals is achieved analytically, assuming a linear approximation of the physical quantities over the panels. For instance, for a quantity f over the panel p:

$$f(\mathbf{x}) = f(\mathbf{x}_G^p) + \nabla_S(f) \cdot (\mathbf{x} - \mathbf{x}_G^p) \tag{III.52}$$

$\nabla_S(f)$ is the surface gradient, defined by:

$$\nabla_S(f) = \mathbf{\Sigma} \begin{pmatrix} f(\mathbf{x}_1^p) \\ f(\mathbf{x}_2^p) \\ f(\mathbf{x}_3^p) \end{pmatrix} \tag{III.53}$$

with:

$$\mathbf{\Sigma} = \frac{1}{\mathbf{\Delta}} \begin{bmatrix} -(\mathbf{A}+\mathbf{B}) & \mathbf{A} & \mathbf{B} \end{bmatrix} \tag{III.54}$$

$$\mathbf{\Delta} = \|\mathbf{x}_2^p - \mathbf{x}_1^p\|^2 \|\mathbf{x}_3^p - \mathbf{x}_1^p\|^2 - [(\mathbf{x}_2^p - \mathbf{x}_1^p) \cdot (\mathbf{x}_3^p - \mathbf{x}_1^p)]^2 \tag{III.55}$$

$$\mathbf{A} = \|\mathbf{x}_3^p - \mathbf{x}_1^p\|^2 (\mathbf{x}_2^p - \mathbf{x}_1^p) - [(\mathbf{x}_2^p - \mathbf{x}_1^p) \cdot (\mathbf{x}_3^p - \mathbf{x}_1^p)](\mathbf{x}_3^p - \mathbf{x}_1^p) \tag{III.56}$$

$$\mathbf{B} = -[(\mathbf{x}_2^p - \mathbf{x}_1^p) \cdot (\mathbf{x}_3^p - \mathbf{x}_1^p)](\mathbf{x}_2^p - \mathbf{x}_1^p) + \|\mathbf{x}_2^p - \mathbf{x}_1^p\|^2 (\mathbf{x}_3^p - \mathbf{x}_1^p) \tag{III.57}$$

Letournel [44] gives the expressions of the two surface integrals:

$$\iint_{\mathcal{S}_p} \phi^P(\mathbf{x}_j) \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS = \left[\widetilde{\mathbf{\Delta}} \iint_{\mathcal{S}_p} \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS - \left(\oint_{\mathcal{C}_p} G(\mathbf{x}_i, \mathbf{x}_j)(\mathbf{x}_j - \mathbf{x}_i) \times \mathbf{dl} \right)^T \mathbf{\Sigma} \right] \begin{pmatrix} \phi^P(\mathbf{x}_1^p) \\ \phi^P(\mathbf{x}_2^p) \\ \phi^P(\mathbf{x}_3^p) \end{pmatrix} \tag{III.58}$$

$$= \mathbf{C}_p^i \begin{pmatrix} \phi^P(\mathbf{x}_1^p) \\ \phi^P(\mathbf{x}_2^p) \\ \phi^P(\mathbf{x}_3^p) \end{pmatrix} \tag{III.59}$$

$$\iint_{\mathcal{S}_p} G(\mathbf{x}_i, \mathbf{x}_j) \frac{\partial \phi^P}{\partial n}(\mathbf{x}_j)\, dS = \left[\widetilde{\mathbf{\Delta}} \iint_{\mathcal{S}_p} G(\mathbf{x}_i, \mathbf{x}_j)\, dS - \left(\oint_{\mathcal{C}_p} \frac{\mathbf{n}_p \times \mathbf{dl}}{G(\mathbf{x}_i, \mathbf{x}_j)} \right)^T \mathbf{\Sigma} \right] \begin{pmatrix} \frac{\partial \phi^P}{\partial n}(\mathbf{x}_1^p) \\ \frac{\partial \phi^P}{\partial n}(\mathbf{x}_2^p) \\ \frac{\partial \phi^P}{\partial n}(\mathbf{x}_3^p) \end{pmatrix} \tag{III.60}$$

$$= \mathbf{D}_p^i \begin{pmatrix} \frac{\partial \phi^P}{\partial n}(\mathbf{x}_1^p) \\ \frac{\partial \phi^P}{\partial n}(\mathbf{x}_2^p) \\ \frac{\partial \phi^P}{\partial n}(\mathbf{x}_3^p) \end{pmatrix} \tag{III.61}$$

with:

$$\widetilde{\mathbf{\Delta}} = \frac{1}{3}\begin{pmatrix}1 & 1 & 1\end{pmatrix} + (\mathbf{x}_i - \mathbf{x}_G^p)^T \mathbf{\Sigma} \tag{III.62}$$

$\mathbf{n}_p$ represents the normal of the panel p.

Equation III.51 expresses the boundary integral equation with respect to the panels of the mesh, but the use of the collocation method involves writing this equation with respect to the nodes of the mesh:

$$-\phi^P(\mathbf{x}_i)\Omega(\mathbf{x}_i) - \sum_{j=1}^{N} G_{ij}\phi^P(\mathbf{x}_j) + \sum_{j=1}^{N} H_{ij}\frac{\partial \phi^P}{\partial n}(\mathbf{x}_j) = 0 \tag{III.63}$$

where $\mathbf{G}$ and $\mathbf{H}$ are the matrices of the influence coefficients or influence matrices. They represent the influence of every node on every node. Their elements are:

$$G_{ij} = \sum_{p \in \mathcal{P}(j)} C_p^i(\delta_p^j) \tag{III.64}$$

$$H_{ij} = \sum_{p \in \mathcal{P}(j)} D_p^i(\delta_p^j) \tag{III.65}$$

$$\delta_p^j = q \in \{1,2,3\} \text{ such as } \mathbf{x}_j = \mathbf{x}_q^p \tag{III.66}$$

$\mathcal{P}(j)$ represents the set of the adjacent panels of which the node j is a vertex. $C_p^i(k)$ and $D_p^i(k)$ denote the k^{th} component of the row vectors $\mathbf{C}_p^i$ and $\mathbf{D}_p^i$.

Regarding the solid angle, Equations (III.63) and (III.44) give:

$$\phi^P(\mathbf{x}_i)\Omega(\mathbf{x}_i) = -\phi^P(\mathbf{x}_i) \iint_{\mathcal{S}} \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS \tag{III.67}$$

$$= -\phi^P(\mathbf{x}_i) \sum_{j=1}^{N} G_{ij} \tag{III.68}$$

leading to:

$$\sum_{j=1}^{N} G_{ij}\phi^P(\mathbf{x}_j) = \sum_{j=1}^{N} H_{ij}\frac{\partial \phi^P}{\partial n}(\mathbf{x}_j) \tag{III.69}$$

with:

$$G_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ij} \tag{III.70}$$

Finally the boundary integral equation can be written with a matrix format:

$$\mathbf{G}\boldsymbol{\phi}^P = \mathbf{H}\boldsymbol{\phi}_n^P \tag{III.71}$$

$\boldsymbol{\phi}^P$ and $\boldsymbol{\phi}_n^P$ are the velocity potential and normal velocity vectors of size N. The i^{th} component of $\boldsymbol{\phi}^P$, respectively $\boldsymbol{\phi}_n^P$, is $\phi(\mathbf{x}_i)$, respectively $\phi_n(\mathbf{x}_i)$.

Note

As the unknowns are at the nodes, there is a singularity at the interfaces free surface / bodies and free surface / numerical tank because both the free surface boundary

equations and the slip condition are applied. Moreover, at the sharp edges in the body meshes, it arises a discontinuity of the normal vectors so the slip condition is not well defined. Consequently multiple node technique is used at the interfaces and at the sharp edges. Doing so, several boundary conditions are satisfied at the same location. This method increases the number of nodes in the mesh but ensures that all the boundary equations are treated properly. The continuity of the velocity potential at the multiple nodes is also checked during the simulation.

III.2.4.3 The linear system

In our problem, the velocity potential is known on the free surface and its normal derivative is known on the solid surfaces (bodies, sea bottom, numerical tank walls). The goal of the *BEM* is to compute the missing quantity on each surface. Thus, the velocity potential will be evaluated on the solid surface and its normal derivative will be found on the free surface.

Mathematically, it means the boundary integral equation (III.71) is decomposed into block matrices and vectors to separate the known quantities from the unknown ones. Consequently, the influence matrices are written:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}(FS) & \mathbf{G}(T) & \cdots & \mathbf{G}(B_j) & \cdots \end{pmatrix} \tag{III.72}$$

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}(FS) & \mathbf{H}(T) & \cdots & \mathbf{H}(B_j) & \cdots \end{pmatrix} \tag{III.73}$$

FS, T and B_j represents respectively the free surface, the numerical tank and the body j. Each block is a rectangular matrix of size the number of nodes in the whole mesh times the number of the nodes in the considered part of the mesh. For instance, $\mathbf{G}(B_j)$ is of size $N \times N(B_j)$ with N the number of nodes in the mesh and $N(B_j)$ the number of nodes in the mesh of the body j.

Equation III.71 becomes a linear system $\mathbf{AX} = \mathbf{B}$ of size N such as:

$$\mathbf{A} = \begin{pmatrix} \mathbf{H}(FS) & -\mathbf{G}(T) & \cdots & -\mathbf{G}(B_j) & \cdots \end{pmatrix} \tag{III.74}$$

$$\mathbf{X} = \begin{pmatrix} \phi_n^P(FS) \\ \phi^P(T) \\ \vdots \\ \phi^P(B_j) \\ \vdots \end{pmatrix} \tag{III.75}$$

$$\mathbf{B} = \begin{pmatrix} -\mathbf{H}(T) & \cdots & -\mathbf{H}(B_j) & \cdots \mathbf{G}(FS) \end{pmatrix} \begin{pmatrix} \phi_n^P(T) \\ \vdots \\ \phi_n^P(B_j) \\ \vdots \\ \phi_n^P(FS) \end{pmatrix} \tag{III.76}$$

The matrix $\mathbf{A}$ is dense, as explained in subsection III.2.3. The solving of this linear system is done using a generalised minimal residual iterative scheme (*GMRES*) with a diagonal preconditioner.

III.2.5 Gradients

Once the velocity potential is known at every node of the mesh, it remains to compute the gradient of both the velocity potential and the free surface elevation to

apply the free surface boundary equation (III.29). Regarding wave elevation $\eta(x, y)$, its gradient is given by:

$$\nabla\eta^P = \begin{pmatrix} \dfrac{\partial\eta^P}{\partial x} \\ \dfrac{\partial\eta^P}{\partial y} \\ 0 \end{pmatrix} \tag{III.77}$$

As the velocity potential also depends on z, its gradient is split into a surface component ∇_S and a normal component $\dfrac{\partial}{\partial n}$ such as:

$$\nabla\phi^P = \nabla_S\phi^P + \frac{\partial\phi^P}{\partial n}\mathbf{n} \tag{III.78}$$

The normal component on the bodies comes from the slip condition (III.30) and on the free surface from the boundary element method (III.76). The horizontal derivatives of the wave elevation and the surface gradient of the velocity potential are computed using a polyharmonic spline approximation [44]. The main results are reminded hereinafter. The approximation of the function s at the node $\mathbf{x}_i$ is:

$$s(\mathbf{x}_i) = \sum_{j=0}^{N_v^i} \alpha_j \psi(\|\mathbf{x}_i - \mathbf{x}_j\|) + p(\mathbf{x}_i) \tag{III.79}$$

with:

- ψ the radial basis function;
- p the unknown polynomial;
- N_v^i the number of first-order and second-order neighboring nodes of the node i;
- $\mathbf{x}_j$ the neighboring nodes;
- α_j the unknown coefficients.

Note

The first-order neighboring nodes are defined as the immediate neighboring nodes while the second-order neighboring nodes are defined as the neighboring nodes of the first-order neighboring nodes, as shown in Figure III.5.

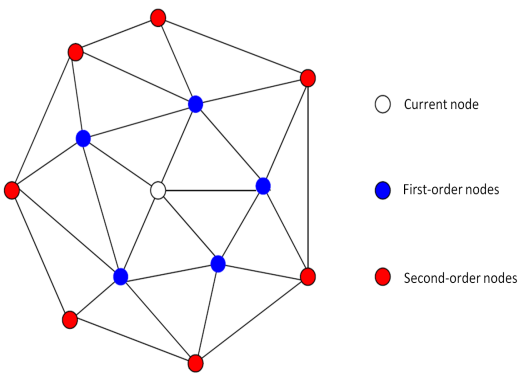


Figure III.5 – Example of the neighborhood (from [44])

In WS_CN, a third-order polynomial spline is used so the kernel ψ and the polynomial p are:

$$\psi(x) = x^3 \tag{III.80}$$

$$p(\mathbf{x}) = \alpha_{N_v^i+1} + \alpha_{N_v^i+2}x + \alpha_{N_v^i+3}y + \alpha_{N_v^i+4}xy + \alpha_{N_v^i+5}x^2 + \alpha_{N_v^i+6}y^2 \tag{III.81}$$

α_j and p are determined by the interpolation conditions on the nodes $\mathbf{x}_j$, neighbors of the node $\mathbf{x}_i$:

$$s(\mathbf{x}_j) = s_j \tag{III.82}$$

$$\sum_{j=1}^{N_v^i} \alpha_j q(\mathbf{x}_j) = 0 \tag{III.83}$$

where s_j is the known velocity potential or wave elevation of the node j and q denotes all polynomials with a degree less or equal than two.

Finally, these equations form a linear system of size $N_v^i + 7$:

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0}_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{s} \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{III.84}$$

with:

- M the matrix containing the evaluation of the radial basis function:

$$M_{ij} = \psi(\|\mathbf{x}_i - \mathbf{x}_j\|) \tag{III.85}$$

- P the matrix with row j is given by the vector $\begin{pmatrix} 1 & x_j & y_j & x_j y_j & x_j^2 & y_j^2 \end{pmatrix}$;
- $\boldsymbol{\alpha}$ the vector of the coefficients α_j, including the coefficients of p.

The spatial derivatives of the wave elevation arise:

$$\begin{cases} \dfrac{\partial \eta^P}{\partial x}(\mathbf{x}_i) = \displaystyle\sum_{j=0}^{N_v^i} \alpha_j \frac{\partial \psi}{\partial x}(\|\mathbf{x}_i - \mathbf{x}_j\|) + \frac{\partial p}{\partial x}(\mathbf{x}_i) \\ \dfrac{\partial \eta^P}{\partial y}(\mathbf{x}_i) = \displaystyle\sum_{j=0}^{N_v^i} \alpha_j \frac{\partial \psi}{\partial y}(\|\mathbf{x}_i - \mathbf{x}_j\|) + \frac{\partial p}{\partial y}(\mathbf{x}_i) \end{cases} \tag{III.86}$$

Concerning the gradient of the velocity potential, a change of variables is necessary to take into account the z-dependency [44].

To ensure the continuity of the gradient of the velocity potential at the multiple nodes in case of intersection curve or sharp edges, the following system is solved (here for a double node):

$$\begin{cases} \nabla\phi^P \cdot \mathbf{n}_{D1} = \dfrac{\partial \phi^P}{\partial n_1} \\ \nabla\phi^P \cdot \mathbf{n}_{D2} = \dfrac{\partial \phi^P}{\partial n_2} \\ \nabla\phi^P \cdot (\mathbf{n}_{D1} \times \mathbf{n}_{D2}) = \nabla\phi^P \cdot (\mathbf{n}_{D1} \times \mathbf{n}_{D2}) \end{cases} \tag{III.87}$$

where $\mathbf{n}_{D1}$, $\mathbf{n}_{D2}$, $\dfrac{\partial \phi^P}{\partial n_1}$ and $\dfrac{\partial \phi^P}{\partial n_2}$ represent the normals and the normal derivative of the velocity potential of the two intersecting surfaces.

If $\mathbf{n}_{D1} \approx \mathbf{n}_{D2}$, the two first equations of the linear system are identical, a new system has to be defined:

$$\begin{cases} \nabla \phi^P \cdot \mathbf{n}_{D1} = \dfrac{\partial \phi^P}{\partial n_1} \\ \nabla \phi^P \cdot (\mathbf{n}_{D1} \times \mathbf{n}_{D2}) = \nabla \phi^P \cdot (\mathbf{n}_{D1} \times \mathbf{n}_{D2}) \\ \nabla \phi^P \cdot [\mathbf{n}_{D2} \times (\mathbf{n}_{D1} \times \mathbf{n}_{D2})] = \nabla \phi^P \cdot [\mathbf{n}_{D2} \times (\mathbf{n}_{D1} \times \mathbf{n}_{D2})] \end{cases} \tag{III.88}$$

III.2.6 Fluid-structure interaction

If the floating and immersed bodies are still or following a prescribed motion, the hydrodynamic problem is completely described by the previous sections. But, in case of free body motions, the motion equation of each body needs to be solved. In WS_CN, bodies are assumed not to be in *mechanical* interactions but only in *hydrodynamic* interactions, so that the motion equation for the body j at its centre of gravity $\mathbf{G}_j$ takes the following form:

$$^e\mathbf{M}_j^{Hydro} \ddot{\boldsymbol{\eta}}_j = {}^e\mathbf{F}_j^{WSC} + {}^e\mathbf{F}_j^{Inertia} + {}^e\mathbf{F}_j^{Other} \tag{III.89}$$

with:

- $^e\mathbf{M}_j^{Hydro}$ the mass matrix used in the hydrodynamic theory:

$$^e\mathbf{M}_j^{Hydro} = \begin{pmatrix} m_j \mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & {}^e\mathbf{I}_{\mathbf{G}_j} \mathbf{S}_j \end{pmatrix} \tag{III.90}$$

  This matrix is different from $^e\mathbf{M}_j$ used in section I.4 in expression and in point of computation ($\mathbf{G}_j$ for $^e\mathbf{M}_j^{Hydro}$, the center of the joint $\mathbf{O}_j$ for $^e\mathbf{M}_j$). $\mathbf{S}_j$ comes from (I.41) and $^e\mathbf{I}_{\mathbf{G}_j}$ is the (3×3) inertia matrix of body j reduced at $\mathbf{G}_j$ with respect to Σ_e.

- $^e\mathbf{F}_j^{Inertia}$ the inertial loads:

$$^e\mathbf{F}_j^{Inertia} = \begin{pmatrix} \mathbf{0}_{3\times 3} \\ -{}^e\mathbf{I}_{\mathbf{G}_j} \dot{\mathbf{S}}_j \mathbf{S}_j^{-1} {}^e\boldsymbol{\omega}_j - \mathbf{S}({}^e\boldsymbol{\omega}_j) {}^e\mathbf{I}_{\mathbf{G}_j} {}^e\boldsymbol{\omega}_j \end{pmatrix} \tag{III.91}$$

- $^e\mathbf{F}_j^{WSC}$ the hydrodynamic loads based on the weak-scatterer hypothesis;

- $^e\mathbf{F}_j^{Other}$ the other external loads (weight, aerodynamic loads, mooring loads, power take-off loads, etc.);

- $\boldsymbol{\eta}_j$ the generalized position vector of the body j:

$$\boldsymbol{\eta}_j = \begin{pmatrix} x_j \\ y_j \\ z_j \\ \phi_j \\ \theta_j \\ \psi_j \end{pmatrix} \tag{III.92}$$

  where these coordinates represent, respectively, the surge, sway, heave, roll, pitch and yaw of the body j.

Note

In [45, 117], the expression of the inertial loads is different:

$$ {}^{e}\mathbf{F}_{j}^{Inertia} = \begin{pmatrix} \mathbf{0}_{3\times 3} \\ -\left({}^{e}\mathbf{I}_{\mathbf{G}_j}\dot{\mathbf{S}}_j + {}^{e}\dot{\mathbf{I}}_{\mathbf{G}_j}\mathbf{S}_j \right)\dot{\boldsymbol{\eta}}_j \end{pmatrix} \tag{III.93} $$

This latter expression matches (III.91).

The hydrodynamic loads are obtained by the integration of the pressure from Bernoulli's equation. To be consistent with the hydrodynamic theory developed here, the decomposition (III.13) and the weak-scatterer hypothesis (III.14) are applied in (III.5), leading to the following expression of the pressure:

$$ p^{WSC} = -\rho\left(\frac{\partial \phi^I}{\partial t} + \frac{\partial \phi^P}{\partial t} + \frac{1}{2}\nabla\phi^I \cdot \nabla\phi^I + \nabla\phi^I \cdot \nabla\phi^P + gz \right) \tag{III.94} $$

Then, the expression of the hydrodynamic loads of the body j at its center of gravity is:

$$ {}^{e}\mathbf{F}_{j}^{WSC} = \begin{pmatrix} -\iint_{S_{B_j}(t)} p^{WSC}\mathbf{n}\, dS \\ -\iint_{S_{B_j}(t)} p^{WSC}(\mathbf{x} - {}^{e}\mathbf{S}_j) \times \mathbf{n}\, dS \end{pmatrix} \tag{III.95} $$

In the expression of the pressure:

- z is known from the node positions;
- ϕ^I, $\nabla\phi^I$ and $\dfrac{\partial \phi^I}{\partial t}$ are known from the incoming wave field which is assumed to be known;
- ϕ^P is known from the boundary value problem subsection III.2.3 and the solving of the boundary integral equation (III.2.4.3);
- $\nabla\phi^P$ is known using the B-spline approximation subsection III.2.5;
- $\dfrac{\partial \phi^P}{\partial t}$ is *unknown*.

Thereby, (III.89) involves two unknowns: the acceleration $\ddot{\boldsymbol{\eta}}_j$ and the time-differentiation of the velocity potential $\dfrac{\partial \phi^P}{\partial t}$. The fluid-structure interaction appears here.

III.2.6.1 State of the art of the computation of the time-differentiation of the velocity potential

Five methods exist to solve this fluid-structure interaction in case of an unsteady potential flow theory [122, 123, 124]:

- The backward finite difference method;
- The iterative method;
- The mode-decomposition method;
- The indirect method;

– The implicit boundary method.

The backward finite difference method is the most straightforward approach due to its simplicity. The time-differentiation of the velocity potential is obtained from the velocity potential at the current and previous time steps:

$$\frac{D\phi}{Dt} = \frac{\phi(t) - \phi(t - dt)}{dt} \tag{III.96}$$

where $\frac{D*}{Dt}$ denotes the Lagrangian derivative. This method is known to be numerically unstable [122].

The iterative method adds a retroactive loop along with a predictor-corrector to converge on the values of the body accelerations and the time-derivative of the velocity potential. This method is used by Cao et al. [125] for example.

The mode-decomposition method splits the time-differentiation of the velocity potential into seven elementary functions (or modes) which correspond to an unit acceleration for each degree of freedom (ψ_j) and the acceleration due to the velocity field (ψ_7):

$$\frac{\partial\phi}{\partial t} = \sum_{j=1}^{6} a_j\psi_j + \psi_7 \tag{III.97}$$

Each mode is solved using a boundary integral equation. Once these modes are found, the body acceleration can be determined. This method was applied by Koo and Kim [122] for instance.

The indirect method does not compute directly $\frac{\partial\phi}{\partial t}$ but only its integration over the body surface. To do so, auxiliary functions are introduced and the use of the Green's second identity leads to an expression of the hydrodynamic forces from these auxiliary functions. Therefore the boundary integral equation of the time-derivative of the velocity potential is not solved. Nevertheless, the body-surface pressure cannot be obtained with this approach. This method has been proposed by Wu and Eatock Taylor [126] and has been widely used so far [11, 127, 128].

The implicit boundary method couples the motion equation to the boundary integral equation satisfied by $\frac{\partial\phi}{\partial t}$. Then, this latter quantity is integrated over the body surface to evaluate the hydrodynamic loads. This approach requires to develop the time-derivative of the body boundary condition, which is a difficult task. Two expressions of this body condition were derived:

– One developed by Cointe et al. [129] in 2D and by Van Daalen [130] in 3D using local normal and tangential components of body and fluid velocities;

– The second proposed by Tanizawa [131] in 2D and extended in 3D by Berkvens [132] based on the acceleration of a fluid particle sliding on the body surface.

The proof of the equivalence of these two expressions is demonstrated by Letournel et al. [133] and a unified expression is given.

In *WS_ CN*, the implicit boundary method was preferred for its efficiency in terms of stability and time-consuming to the other methods and follows the formalism developed by Letournel [44].

III.2.6.2 The implicit boundary method

The method consists in solving a new boundary value problem. The Laplacian of the $\frac{\partial \phi^P}{\partial t}$ is found using (III.30):

$$\Delta \frac{\partial \phi^P}{\partial t} = \frac{\partial \Delta \phi^P}{\partial t} = 0 \quad \text{in the fluid domain } \mathcal{D} \tag{III.98}$$

Note

The time-derivative of the velocity potential, $\frac{\partial \phi}{\partial t}$, is named *the acceleration potential* sometimes but it is not equal to the Lagrangian time-differentiation of the velocity:

$$\frac{D\mathbf{v}}{Dt} = \frac{D\nabla\phi}{Dt} = \frac{\partial \nabla \phi}{\partial t} + (\nabla\phi)\cdot\nabla\phi = \nabla\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}\nabla\phi\cdot\nabla\phi\right) \tag{III.99}$$

Because for all vectors $\mathbf{a}$:

$$\frac{1}{2}\nabla(\mathbf{a}\cdot\mathbf{a}) = (\mathbf{a}\cdot\nabla)\mathbf{a} + \mathbf{a}\times(\nabla\times\mathbf{a}) \tag{III.100}$$

$\frac{\partial \phi}{\partial t} + \frac{1}{2}\nabla\phi\cdot\nabla\phi$ does not satisfy Laplace's equation because of the velocity-squared term [123]. The terms *time-derivative* or *time-differentiation* of the velocity potential are preferred.

The free surface boundary equation is given by (III.28):

$$\frac{\partial \phi^P}{\partial t} = \frac{\delta \phi^P}{\delta t} - +(\mathbf{v}_{mesh}\cdot\nabla)\phi^P \quad \text{at } z = \eta^I(x,y,t) \tag{III.101}$$

The slip condition needs to use the normal derivative of the time-differentiation of the velocity potential, defined by:

$$\frac{\partial^2 \phi^P}{\partial t \partial n} = \frac{\partial^2 \phi^P}{\partial n \partial t} = \nabla\left(\frac{\partial \phi^P}{\partial t}\right)\cdot\mathbf{n} \tag{III.102}$$

On the seabed and the numerical tank walls, this quantity is:

$$\frac{\partial^2 \phi^P}{\partial t \partial n} = -\frac{\partial^2 \phi^I}{\partial t \partial n} \qquad \text{on the seabed} \tag{III.103}$$

$$\frac{\partial^2 \phi^P}{\partial t \partial n} = 0 \qquad \text{on the numerical tank walls} \tag{III.104}$$

On the wetted body surfaces for the body j, the slip condition is given by Letournel et al. [133] in case of translational motions and extended to rotational motions by Chauvigné [45], here at the node $\mathbf{x}_i$:

$$\frac{\partial^2 \phi^P}{\partial t \partial n} = -\frac{\partial^2 \phi^I}{\partial t \partial n} + \begin{pmatrix} \mathbf{n} \\ [(\mathbf{x}_i - {}^e\mathbf{S}_j)\times\mathbf{n}]^T\mathbf{S}_j \end{pmatrix}\cdot\ddot{\boldsymbol{\eta}}_j + q_j \tag{III.105}$$

q_j is given by:

$$\begin{aligned} q_j =& {}^e\boldsymbol{\omega}_j \cdot \mathbf{s}_1 \left(\frac{\partial \phi^P}{\partial s_2} - 2(\dot{\boldsymbol{\eta}}_j^{trans} \cdot \mathbf{s}_2) \right) - {}^e\boldsymbol{\omega}_j \cdot \mathbf{s}_2 \left(\frac{\partial \phi^P}{\partial s_1} - 2(\dot{\boldsymbol{\eta}}_j^{trans} \cdot \mathbf{s}_1) \right) \\ &+ \frac{(\dot{\boldsymbol{\eta}}_j^{trans} \cdot \mathbf{s}_1)}{R_1} \left(\frac{\partial \phi^P}{\partial s_1} - (\boldsymbol{\eta}_j^{trans} \cdot \mathbf{s}_1) \right) + \frac{(\dot{\boldsymbol{\eta}}_j^{trans} \cdot \mathbf{s}_2)}{R_2} \left(\frac{\partial \phi^P}{\partial s_2} - (\boldsymbol{\eta}_j^{trans} \cdot \mathbf{s}_2) \right) \\ &+ (\dot{\boldsymbol{\eta}}_j^{trans} \cdot \mathbf{n}) \left[\frac{\partial^2 \phi^P}{\partial s_1^2} + \frac{\partial^2 \phi^P}{\partial s_2^2} + \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \frac{\partial \phi^P}{\partial n} \right] \\ &+ \left[(\dot{\mathbf{S}}_j \dot{\boldsymbol{\eta}}_j^{rot}) \times (\mathbf{x}_i - {}^e\mathbf{S}_j) + {}^e\boldsymbol{\omega}_j \times ({}^e\boldsymbol{\omega}_j \times (\mathbf{x}_i - {}^e\mathbf{S}_j)) \right] \cdot \mathbf{n} \end{aligned} \tag{III.106}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are the two local tangent vectors and R_1 and R_2 denote the local curvature associated to these vectors.

The boundary integral equation used with the velocity potential (III.46) is also valid with its time-derivative, here written at the point $\mathbf{x}_i$:

$$-\frac{\partial \phi^P}{\partial t}(\mathbf{x}_i)\Omega(\mathbf{x}_i) - \iint_{\mathcal{S}} \frac{\partial \phi^P}{\partial t}(\mathbf{x}_j) \frac{\partial G}{\partial n_j}(\mathbf{x}_i, \mathbf{x}_j)\, dS + \iint_{\mathcal{S}} G(\mathbf{x}_i, \mathbf{x}_j) \frac{\partial^2 \phi^P}{\partial n \partial t}(\mathbf{x}_j)\, dS = 0 \tag{III.107}$$

III.2.6.3 The discretization

Note

The time-differentiation is written either $\dfrac{\partial *}{\partial t}$ or $\dot{(*)}$.

As the boundary integral equation is the same as in the first boundary value problem, except that the velocity potential is substituted for its time-differentiation, then the influence matrices $\mathbf{G}$ and $\mathbf{H}$ stay identical. Therefore, (III.71) becomes:

$$\mathbf{G}\dot{\boldsymbol{\phi}} = \mathbf{H}\dot{\boldsymbol{\phi}}_n \tag{III.108}$$

Regarding the slip condition of the body j (III.105), its discretization gives:

$$\dot{\boldsymbol{\phi}}_n^P(B_j) = -\dot{\boldsymbol{\phi}}_n^I(B_j) + \mathbf{CK}_j \ddot{\boldsymbol{\eta}}_j + \mathbf{Q}_j \tag{III.109}$$

$\mathbf{CK}_j$ is a rectangular matrix of size $N(B_j) \times 6$ and $\mathbf{Q}_j$ is a vector of size $N(B_j)$.

The hydrodynamic loads of the body j (III.95) becomes:

$${}^e\mathbf{F}_j^{WSC} = {}^e\mathbf{CT}_j \dot{\boldsymbol{\phi}}^P(B_j) + {}^e\mathbf{T}_j^{WSC} \tag{III.110}$$

where ${}^e\mathbf{CT}_j$ is a $6 \times N(B_j)$ matrix and ${}^e\mathbf{T}_j^{WSC}$ is a vector of size 6 including the known components of the hydrodynamic loads.

Thus, the discretized motion equation arises:

$${}^e\mathbf{M}_j^{Hydro} \ddot{\boldsymbol{\eta}}_j - {}^e\mathbf{CT}_j \dot{\boldsymbol{\phi}}^P(B_j) = {}^e\mathbf{T}_j^{WSC} + {}^e\mathbf{F}_j^{Inertia} + {}^e\mathbf{F}_j^{Other} \tag{III.111}$$

III.2.6.4 The linear system

The set of ordinary differential equations to solve is:

$$\begin{cases} \mathbf{G}\dot{\boldsymbol{\phi}} = \mathbf{H}\dot{\boldsymbol{\phi}}_n \\ {}^{e}\mathbf{M}_j^{Hydro}\ddot{\boldsymbol{\eta}}_j - {}^{e}\mathbf{C}\mathbf{T}_j\dot{\boldsymbol{\phi}}^P(B_j) = {}^{e}\mathbf{T}_j^{WSC} + {}^{e}\mathbf{F}_j^{Inertia} + {}^{e}\mathbf{F}_j^{Other} \quad \forall j \in [\![1\ ;\ N^{WSC}]\!] \\ \dot{\boldsymbol{\phi}}_n^P(B_j) = -\dot{\boldsymbol{\phi}}_n^I(B_j) + \mathbf{C}\mathbf{K}_j\ddot{\boldsymbol{\eta}}_j + \mathbf{Q}_j \quad \forall j \in [\![1\ ;\ N^{WSC}]\!] \end{cases} \tag{III.112}$$

Equation III.112 forms a linear system $\mathbf{AX} = \mathbf{B}$ of size $N + \sum_{j=1}^{N^{WSC}} N(B_j) + 6N^{WSC}$. The matrix $\mathbf{A}$ is dense so the solving of this linear system is achieved using a *GMRES* method with a diagonal preconditioner. The vector $\mathbf{X}$ is:

$$\mathbf{X} = \begin{pmatrix} \dot{\boldsymbol{\phi}}_n^P(FS) \\ \dot{\boldsymbol{\phi}}^P(T) \\ \vdots \\ \dot{\boldsymbol{\phi}}^P(B_j) \\ \vdots \\ \dot{\boldsymbol{\phi}}_n^P(B_j) \\ \vdots \\ \ddot{\boldsymbol{\eta}}_j \\ \vdots \end{pmatrix} \tag{III.113}$$

III.3 Mesh generation

The solving of the two boundary value problems requires the use of a surface mesh on the whole domain: free surface, wetted body surfaces and numerical tank walls. To avoid numerical discrepancies, a good quality triangulated mesh is necessary, in terms of size and shape. Using an unsteady hydrodynamic solver, the mesh changes with time. Consequently, not only a mesh has to be created at the initial stage, but also at any time during the simulation, independently of the wave field and the body motions.

In his PhD work, Letournel [44] only performed simulations with a single submerged body so the surface mesh of the domain was easy to generate. Indeed, in this case, the body mesh and the free surface do not intersect each other. Then, Chauvigné [45] has extended the code to the simulation of a single surface-piercing body. This involves the tracking of the intersection curve and the continuity of the mesh at this interface. The (free-)surface-to-(body-)surface intersection is seeked using a marching method which starts from a known starting point on the intersection curve and then steps along it in a direction prescribed by the curve local geometry. Regarding the mesh generation, an advance front method has been developed by Chauvigné [45] and applied for both the body and the free surface meshes.

III.3.1 Intersection curve tracking

III.3.1.1 Initial point

Each body is formed of a set of parametric surfaces $\mathbf{S}(u, v)$. For example, a cylinder is made of two discs and a cylindrical body. Its parametric equation is:

$$\mathbf{S}^{Cylinder}(u, v) = \begin{pmatrix} R\cos(u) \\ R\sin(u) \\ v \end{pmatrix} \tag{III.114}$$

A cube is formed of six planes. The parametric equation of a plane is:

$$\mathbf{S}^{Plane}(u, v) = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} \tag{III.115}$$

The initial point on the intersection curve is searched along an isoparametric curve Figure III.6 and found using a dichotomy method.

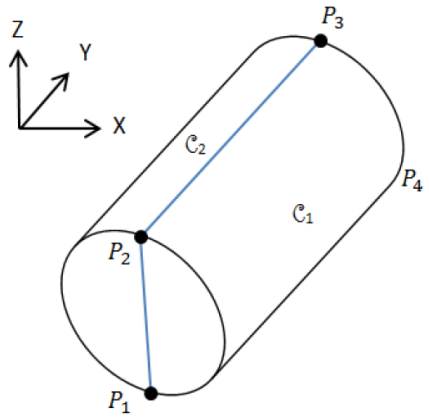


Figure III.6 – Example of an isoparametric line for a cylinder (from [45])

III.3.1.2 Marching method

Once the initial point $\mathbf{P}_0 = \mathbf{S}(u_0, v_0)$ of the intersection line is found, the rest of the curve is tracked using a marching method. The intersection curve $\mathbf{C}(s) = \mathbf{S}(u_C(s), v_C(s))$ is defined by the equation:

$$\mathbf{T}(s) \cdot \mathbf{n}_{FS} = 0 \tag{III.116}$$

where:

– $\mathbf{T}$ is the local tangent vector of the surface $\mathbf{S}(u, v)$:

$$\mathbf{T}(s) = \mathbf{S}_u \frac{\partial u}{\partial s} + \mathbf{S}_v \frac{\partial v}{\partial s} \tag{III.117}$$

with:

$$\begin{cases} \mathbf{S}_u = \dfrac{\partial \mathbf{S}}{\partial u} \\ \mathbf{S}_v = \dfrac{\partial \mathbf{S}}{\partial v} \end{cases} \tag{III.118}$$

- $\mathbf{n}_{FS}$ is the free surface normal vector;
- s is the curvilinear coordinate.

The rates of change of (u_C, v_C) along the intersection are solutions of the following system of differential equation:

$$\begin{cases} \dfrac{du_C}{ds} = \dfrac{\mathbf{S}_v \cdot \mathbf{n}_{FS}}{\sqrt{E(\mathbf{S}_v \cdot \mathbf{n}_{FS})^2 - 2F(\mathbf{S}_v \cdot \mathbf{n}_{FS})(\mathbf{S}_u \cdot \mathbf{n}_{FS}) + G(\mathbf{S}_u \cdot \mathbf{n}_{FS})^2}} \\ \dfrac{dv_C}{ds} = \dfrac{\mathbf{S}_u \cdot \mathbf{n}_{FS}}{\sqrt{E(\mathbf{S}_v \cdot \mathbf{n}_{FS})^2 - 2F(\mathbf{S}_v \cdot \mathbf{n}_{FS})(\mathbf{S}_u \cdot \mathbf{n}_{FS}) + G(\mathbf{S}_u \cdot \mathbf{n}_{FS})^2}} \end{cases} \tag{III.119}$$

where:

$$\begin{cases} E = \mathbf{S}_u \cdot \mathbf{S}_u \\ F = \mathbf{S}_u \cdot \mathbf{S}_v \\ G = \mathbf{S}_v \cdot \mathbf{S}_v \end{cases} \tag{III.120}$$

As the initial point on the intersection curve is known, the other points are found from the integration of (III.119):

$$\mathbf{P} = \mathbf{S}(u_0 + \delta_u, v_0 + \delta_v) \tag{III.121}$$

with:

$$\begin{cases} \delta_u = \displaystyle\int_{s_0}^{s_0+ds} du \\ \delta_v = \displaystyle\int_{s_0}^{s_0+ds} dv \end{cases} \tag{III.122}$$

A fourth-order explicit Runge-Kutta scheme with a fixed spatial step ds is used (I.94).

III.3.2 Grid generation

As soon as the intersection curve is known (Figure III.7b) from the parametric surfaces of the bodies (Figure III.7a), the mesh generation process may start. It unfolds in five steps:

- The extremities of the edges of the geometries are tracked (for instance, the vertices of a cube);
- The intersection curves are meshed (Figure III.7c);
- The edges of all geometries are meshed, the multiple nodes are defined (Figures III.7c and III.7d);
- The wetted body surfaces are meshed (Figure III.7e);
- The free surface and the numerical tank walls are meshed and connected to the wetted body meshes (Figure III.7f).

The grid generation is based on the advance front method. The mesh front is initialized by the meshed edges of a surface. Then, new triangular panels are created successively, which updates the front of the mesh, until the whole surface is meshed. A new panel is generated from three existing nodes of the front or, if it is not possible, from two existing nodes of the same edge of the front and a new node. The position of this new node $\mathbf{P}_{new}$ satisfies the following equation:

$$\mathbf{MP}_{new} = \sqrt{\delta_1^2 - \left(\frac{\delta_{ref}}{2}\right)^2}\,\mathbf{n}_{Front} \tag{III.123}$$

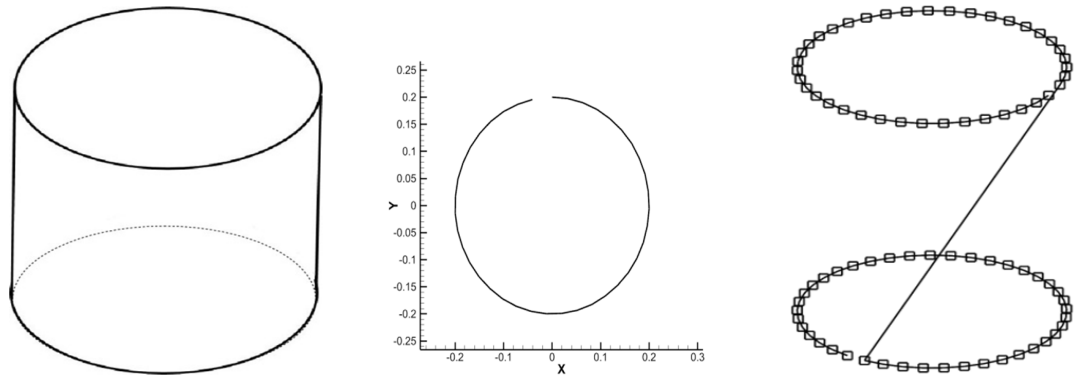
with:

- $\mathbf{n}_{Front}$ the normal vector oriented from the mesh part of the surface to the unmeshed part;
- $\mathbf{M}$ the midpoint of the edge of the front;
- δ_{ref} the reference panel size;
- δ_1 a distance defined by:

$$\delta_1 = \begin{cases} \delta_{ref} & \text{if } 0.5\ell < \delta_{ref} \text{ and } \ell > \delta_{ref} \\ 0.55\delta_{ref} & \text{if } 0.55\ell > \delta_{ref} \\ 2\delta_{ref} & \text{if } 2\ell > \delta_{ref} \end{cases} \tag{III.124}$$

- ℓ the length of the edge.

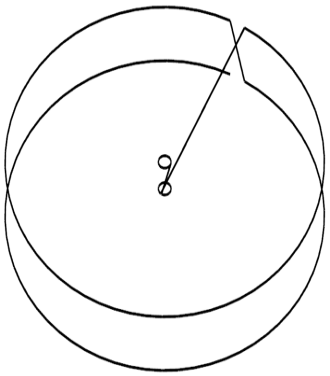
Intersection tests are used to avoid the creation of forbidden or degenerate panels. The advance front method leads to an unstructured mesh.



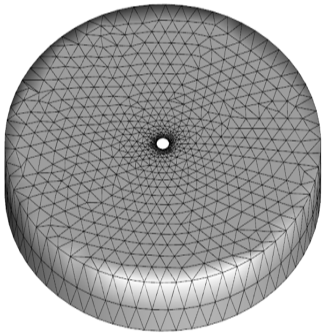
(a) Parametric surfaces of a cylinder

(b) Intersection curve

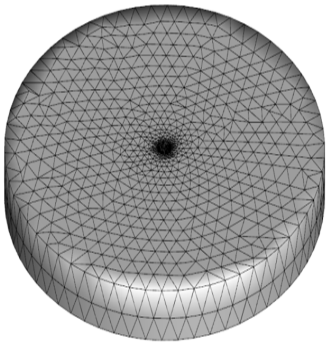
(c) Mesh of the edges of a cylinder



(d) Mesh of all edges of the domain



(e) Mesh of the free surface and the numerical tank walls



(f) Final mesh including the cylinder

Figure III.7 – The steps of the mesh generation of the whole domain with a floating vertical cylinder

III.3.3 Reference panel size

The reference panel size δ_{ref} is often constant over a surface but not on the free surface. Indeed, to avoid the creation of meshes with too many panels, the free surface mesh is finer close to the surface-piercing bodies than far from the bodies. The evolution of δ_{ref} over the free surface is figured out by solving the following problem [45]:

$$\begin{cases} \Delta\delta_{ref} & = 0 \text{ on the free surface} \\ \delta_{ref} & = \delta_{input} \text{ at the boundaries of the free surface} \end{cases} \tag{III.125}$$

The boundaries of the free surface are the intersection curves with the surface-piercing bodies and the intersection between the free surface and the numerical tank walls. δ_{input} is the input panel size fixed by the user.

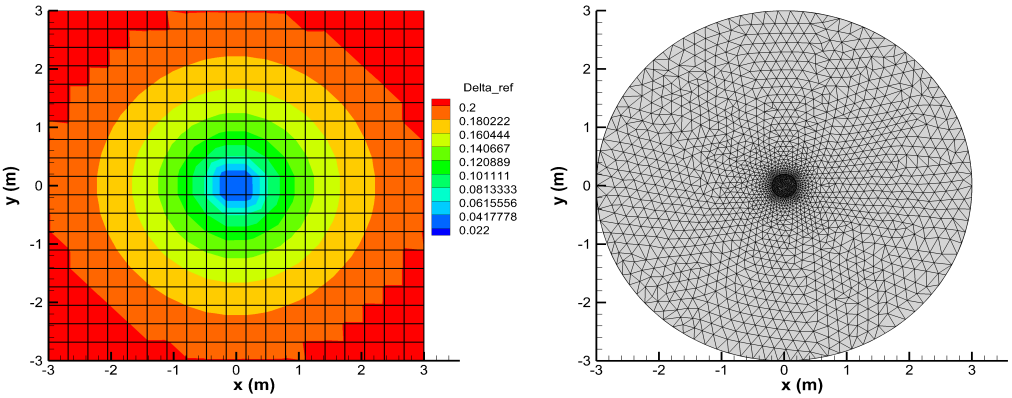
These equations are discretized over a *first* free surface mesh matching the plane $z = 0$. The mesh is structured with a constant spatial discretization h_{ref} in both x and y directions. To make the difference with the final unstructured mesh, the structured mesh is named the *Cartesian grid*. The solution is searched at the center of every cell of this grid. A second-order central finite-difference scheme is used to solve the Laplace's

equation. And after discretization:

$$\frac{\delta_{ref}^{i+1,j} - 2\delta_{ref}^{i,j} + \delta_{ref}^{i-1,j}}{h_{ref}^2} + \frac{\delta_{ref}^{i,j+1} - 2\delta_{ref}^{i,j} + \delta_{ref}^{i,j-1}}{h_{ref}^2} = 0 \tag{III.126}$$

$\delta_{ref}^{i,j}$ represents the value of δ_{ref} in the cell (i, j) of the Cartesian grid. Equation III.126 leads to a linear system $\mathbf{AX} = \mathbf{B}$ of size N_{ref}, the number of cells in the grid. A LU factorization is used. The value of δ_{ref} at the position of the created point is found from a bilinear interpolation.

Figure III.8a shows the values of δ_{ref} over the free surface for a domain of radius 3 m with $\delta_u = 0.2$ m with a surface-piercing floating cylinder of radius 0.2 m and $\delta_u = 0.022$ m. Figure III.8b displays the final mesh with high quality panels and a smooth transition between the finest area to the coarsest area.

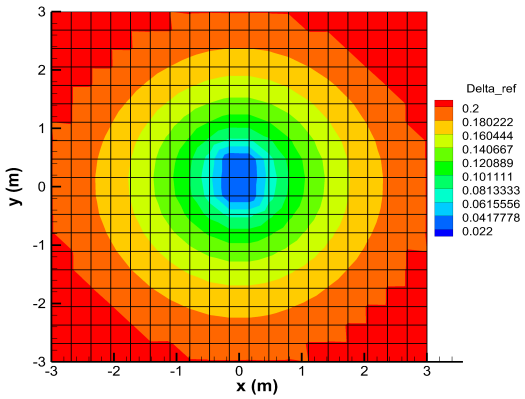


(a) δ_{ref} over the free surface, the black lines represent the Cartesian grid.

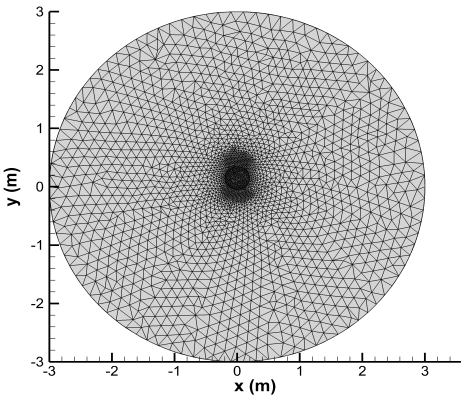
(b) Final mesh

Figure III.8 – $h_{ref} = 0.3$ m - $y = 0$ m

In case of a large amplitude motion, the mesh can drastically lose its quality. For instance in Figure III.9, the cylinder is moved 0.15 m along y-axis. An important discontinuity of the mesh discretization appears which leads to numerical errors, especially when local interpolations are used (cf. section III.8).



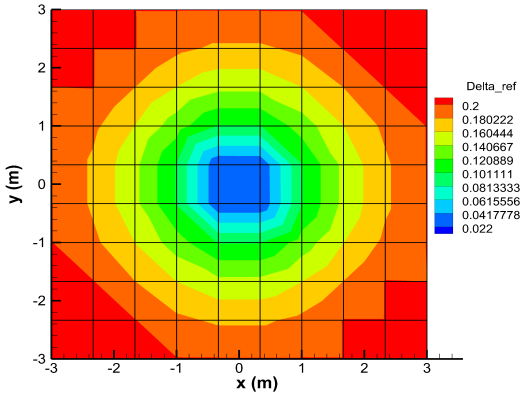
(a) δ_{ref} over the free surface, the black lines represent the Cartesian grid.



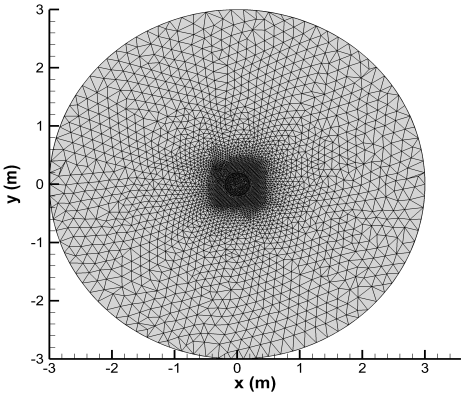
(b) Final mesh

Figure III.9 – $h_{ref} = 0.3$ m - $y = 0.15$ m

The discontinuity can be more important if a larger h_{ref} is used. In Figure III.10, the whole cell containing the cylinder has a finer discretization than the rest of the mesh.



(a) δ_{ref} over the free surface, the black lines represent the Cartesian grid.



(b) Final mesh

Figure III.10 – $h_{ref} = 0.6$ m - $y = 0$ m

Consequently, the structured mesh, necessary to solve (III.125), has to be fine enough to ensure a high quality mesh for any arbitrary motion of the bodies, especially in case of multibody simulations with large relative motions. This involves choosing a small h_{ref} and so increase significantly the size of the linear system to solve. In the two cases presented above, $N_{ref} = 400$ and 100 for $h_{ref} = 0.3$ m and 0.6 m. If $h_{ref} = 0.01$ m, then $N_{ref} = 360000$ (*sic*), which requires a tremendous quantity of memory if the linear system is considered as dense. Nevertheless, the linear system is not dense: (III.126) involves a large sparse linear system. In his PhD work, Chauvigné [45] did not use the sparsity of the linear system to solve it and it was not possible to compute δ_{ref} in case of fine meshes. The quality of the mesh could not be maintained during the simulation if δ_{ref} was not small enough. Thus, a new linear system solver has been

coupled with *WS_ CN* in order to improve the memory efficiency: *PARDISO*[6] (Parallel Sparse Direct And Multi-Recursive Iterative Linear Solvers). A LU decomposition is always performed but the method is optimized for sparse linear systems. The non-zero coefficients of the matrix $\mathbf{A}$ are stored using a Compressed Sparse Row format (or *CSR*), based on three arrays:

- **aa**, containing the non-zero coefficients;
- **ja**, storing the column indices of the non-zero coefficients in increasing order;
- **ia**, listing the first column indices of the previous array for every row.

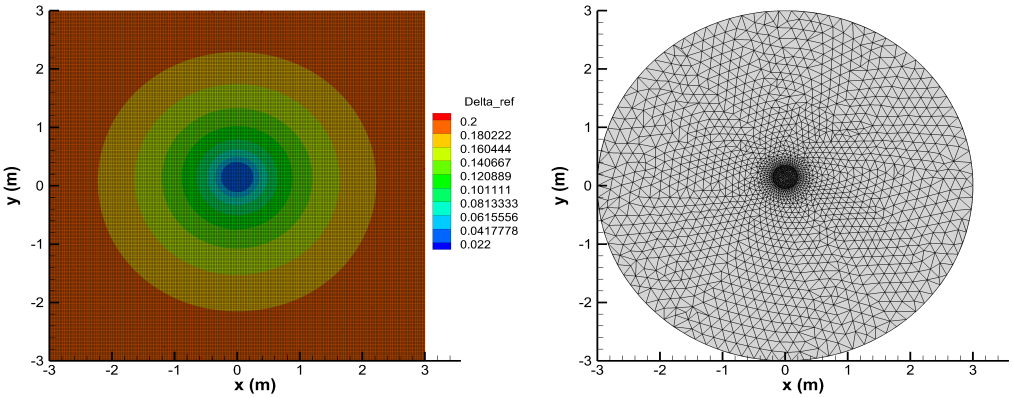
For example, if the matrix $\mathbf{A}$ is [7]:

$$\mathbf{A} = \begin{pmatrix} 4 & * & * & 7 & * \\ * & * & 6 & * & * \\ -3 & * & * & -2 & 9 \\ * & 1 & * & 3 & * \\ * & * & 8 & * & 6 \end{pmatrix} \tag{III.127}$$

Then, the *CSR* format of $\mathbf{A}$ is:

$$\begin{cases} \mathbf{aa} &= \begin{pmatrix} 4 & 7 & 5 & -3 & -2 & 9 & 1 & 3 & 8 & 6 \end{pmatrix} \\ \mathbf{ja} &= \begin{pmatrix} 1 & 4 & 3 & 1 & 4 & 5 & 2 & 4 & 3 & 5 \end{pmatrix} \\ \mathbf{ia} &= \begin{pmatrix} 1 & 3 & 4 & 7 & 9 \end{pmatrix} \end{cases} \tag{III.128}$$

The use of this storage automatically allows to deal with very large linear system and therefore small discretization step h_{ref}. The case presented in Figure III.9 is now displayed again but with $h_{ref} = 0.01$ m in Figure III.11 and show a very good quality mesh.



(a) δ_{ref} over the free surface, the black lines represent the Cartesian grid.

(b) Final mesh

Figure III.11 – $h_{ref} = 0.01$ m - $y = 0.15$ m

[6] https://www.pardiso-project.org
[7] $*$ denotes a zero.

Table III.2 presents the CPU-time for solving (III.125) in the present test case by considering the sparsity of the linear system (*Sparse*) or not (*Dense*). As it can be seen, not only the use of *PARDISO* allows to deal with larger linear systems but also it is quite faster. As a reminder, a mesh counts roughly between 5000 and 15 000 nodes, giving the size of the linear system to solve for the first boundary value problem. Hence, the computation of the δ_{ref} with a fine Cartesian grid can easily lead to larger linear systems to solve than the boundary element methods.

h_{ref} (m)	N_{ref}	Density (%)	CPU time (s)	
			Dense	Sparse
0.6	100	2.92	0.001	0.094
0.3	400	9.45×10^{-1}	0.004	0.094
0.1	3600	1.29×10^{-1}	0.525	0.125
0.07	7225	6.59×10^{-2}	2.494	0.140
0.06	10 000	4.80×10^{-2}	5.785	0.187
0.05	14 400	3.36×10^{-2}	ø	0.234
0.01	360 000	1.38×10^{-3}	ø	4.227
0.005	1 440 000	3.46×10^{-6}	ø	ø

Table III.2 – CPU-times for different sizes of linear system using or not its sparsity

The density of a linear system is the number of non-zero coefficients in $\mathbf{A}$ divided by the total number of coefficients, *i.e.* N_{ref}^2. ø indicates that the calculation was not possible because of the lack of memory on the local computer used during this PhD work.

III.4 Mesh morphing

Once the mesh is generated, it has to be updated at each time step of the simulation to take into account the motion of both the bodies and the free surface. Two choices are possible. Either the generation of a new mesh or the deformation of the existing mesh. The first method ensures a good mesh quality through the time-domain simulation but involves the interpolation of the physical quantities on the free surface (cf. section III.8) and is time-consuming. Nevertheless this first method can handle large deformations. The second approach is possible when small deformations occur and requires a smaller CPU time.

To avoid too many remeshings, two mesh morphing algorithms are implemented in *WS_CN* [45]:

- A spring analogy method for the mesh of the bodies;
- A mesh deformation method using radial basis functions for the free surface mesh.

When the deformations are too large, a remeshing process is applied.

III.4.1 Body mesh morphing

The spring analogy method consists of a physical analogy which replaces every edge of the mesh by a fictitious spring connecting two vertices. The method was introduced by Batina [134] for unstructured meshes in case of an unsteady aerodynamic analysis of oscillating airfoils. But it is also used in hydrodynamics, for example by Leroyer [135] and Jacquin [136] with a viscous flow solver. The spring analogy method is applied in case of [137]:

- Moving boundary problems;
- Unstructured mesh smoothing.

In this latter application, the nodes are moved inside the mesh to obtain better shape panels and a higher quality mesh. This method will be used in section V.1. An example of such a mesh optimization may be found in [138]. The spring analogy method used in the present hydrodynamic solver enables to deform the mesh subject to moving boundaries.

The force exerted on the node i by its neighboring nodes follows Hooke's law [137]:

$$\mathbf{F}_i = \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{x}_j - \mathbf{x}_i - \mathbf{d}_{ij}) \tag{III.129}$$

where:

- $\mathbf{x}_i$ is the position of the node i;
- $\mathbf{d}_{ij}$ is the equilibrium vector between the node i and j;
- k_{ij} denotes the spring stiffness;
- N_v^i represents the number of **first-order** neighboring nodes of the node i.

Two different linear spring analogy exist, depending on the expression of the equilibrium vector:

- The *segment spring analogy* when the equilibrium lengths are equal to the initial lengths of the edges: $\mathbf{d}_{ij} = \mathbf{x}_j^{old} - \mathbf{x}_i^{old}$. The expression of the spring loads becomes:

$$\mathbf{F}_i = \sum_{j=1}^{N_v^i} k_{ij}(\boldsymbol{\delta}_j - \boldsymbol{\delta}_i) \tag{III.130}$$

  with $\boldsymbol{\delta}_i = \mathbf{x}_i - \mathbf{x}_i^{old}$, the displacement of the node i. The mesh is only deformed if its boundaries are deformed themselves. Consequently this approach is chosen to solve moving boundary problems.

- The *vertex spring analogy* when the equilibrium lengths are zeroed: $\mathbf{d}_{ij} = \mathbf{0}_{3\times1}$. Doing so, the mesh can be deformed even if its boundaries are still, which is the case for a mesh smoothing.

Here, the mesh deformation algorithm is used to update the mesh after a modification of the position of the bodies, thus the segment spring analogy is chosen.

In order to prevent the colliding of nodes and the interpenetration of neighboring panels, the spring stiffness is inversely proportional to the square of the edge length. Thus, if two vertices get closer, the spring becomes stiffer and conversely.

$$k_{ij} = \frac{1}{\|\mathbf{x}_j - \mathbf{x}_i\|^2} \tag{III.131}$$

The static equilibrium of the mesh leads to:

$$\mathbf{F}_i = \mathbf{0}_{3\times 1} \tag{III.132}$$

This equation may be solved iteratively for every node i or the whole linear system can be written and solved for all the nodes at a time.

Torsional springs also exist to handle larger deformations [139, 140]. Other modifications of the linear spring analogy are available too [141].

To ensure the spring loads are along the edge axis, Chauvigné [45] preferred the following expression:

$$\mathbf{F}_i = \sum_{j=1}^{N_v^i} k_{ij}(\boldsymbol{\delta}_j \cdot \mathbf{n}_{ij} - \boldsymbol{\delta}_i \cdot \mathbf{n}_{ij}) \tag{III.133}$$

with $\mathbf{n}_{ij}$ the tangential unit vector along the edge formed by the nodes i and j.

The displacement is written in the local direct orthonormal base $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{n}_i)$ of the node i:

$$\boldsymbol{\delta}_i = d_u^i \mathbf{u}_i + d_v^i \mathbf{v}_i + d_w^i \mathbf{n}_i \tag{III.134}$$

where:

- $\mathbf{n}_i$ is the inward normal vector of the body surfaces;
- $\mathbf{u}_i$ and $\mathbf{v}_i$ are the two tangential vectors.

The slip condition on the body j at the node i involves:

$$d_w^i = [{}^e\mathbf{v}_j + {}^e\boldsymbol{\omega}_j \times (\mathbf{x}_i - {}^e\mathbf{S}_j)] \cdot \mathbf{n}_i dt \tag{III.135}$$

On the intersection curve or at the sharp edges of the body meshes, a node on this intersection has to remain on it. Thus, $\mathbf{n}_i$ is normal to one of the surfaces, $\mathbf{u}_i$ is along the intersection line and $\mathbf{v}_i$ is chosen such as the local basis is orthonormal and direct. In that case, d_v is defined by:

$$d_v^i = \begin{cases} \dfrac{\partial \phi^I}{\partial n_{FS}} \mathbf{n}_{FS} \cdot \mathbf{v}_i dt & \text{on the intersection curve} \\ [{}^e\mathbf{v}_j + {}^e\boldsymbol{\omega}_j \times (\mathbf{x}_i - {}^e\mathbf{S}_j)] \cdot \mathbf{v}_i dt & \text{at the sharp edges} \end{cases} \tag{III.136}$$

$\mathbf{n}_{FS}$ is the normal vector of the double node linked to the free surface on the intersection curve.

If three surfaces or more intersect each other at the same node i, then the condition is imposed on d_u^i:

$$d_u^i = [{}^e\mathbf{v}_j + {}^e\boldsymbol{\omega}_j \times (\mathbf{x}_i - {}^e\mathbf{S}_j)] \cdot \mathbf{u}_i dt \tag{III.137}$$

The use of the slip condition at every node makes the problem bi-dimensional, so the static equilibrium of this segment spring analogy method is for every node i:

$$\begin{cases} \mathbf{F}_i \cdot \mathbf{u}_i & = 0 \\ \mathbf{F}_i \cdot \mathbf{v}_i & = 0 \end{cases} \tag{III.138}$$

After mathematical developments, the system of equations becomes for a node i on a smooth surface:

$$\begin{cases} \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_j \cdot \mathbf{n}_{ij})(\mathbf{u}_i \cdot \mathbf{n}_{ij}) d_u^j - \left[\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_i \cdot \mathbf{n}_{ij})^2\right] d_u^i + \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{v}_j \cdot \mathbf{n}_{ij})(\mathbf{u}_i \cdot \mathbf{n}_{ij}) d_v^j \\ -\left[\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_i \cdot \mathbf{n}_{ij})(\mathbf{v}_i \cdot \mathbf{n}_{ij})\right] d_v^i = \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{n}_j \cdot \mathbf{n}_{ij})(\mathbf{u}_i \cdot \mathbf{n}_{ij}) d_w^j - \left[\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{n}_i \cdot \mathbf{n}_{ij})(\mathbf{u}_i \cdot \mathbf{n}_{ij})\right] d_w^i \\ \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_j \cdot \mathbf{n}_{ij})(\mathbf{v}_i \cdot \mathbf{n}_{ij}) d_u^j - \left[\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_i \cdot \mathbf{n}_{ij})(\mathbf{v}_i \cdot \mathbf{n}_{ij})\right] d_u^i + \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{v}_j \cdot \mathbf{n}_{ij})(\mathbf{v}_i \cdot \mathbf{n}_{ij}) d_v^j \\ -\left[\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{v}_i \cdot \mathbf{n}_{ij})^2\right] d_v^i = \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{n}_j \cdot \mathbf{n}_{ij})(\mathbf{v}_i \cdot \mathbf{n}_{ij}) d_w^j - \left[\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{n}_i \cdot \mathbf{n}_{ij})(\mathbf{v}_i \cdot \mathbf{n}_{ij})\right] d_w^i \end{cases} \tag{III.139}$$

Equation III.139 leads to a sparse linear system of size twice the number of nodes in the body mesh $\mathbf{AX} = \mathbf{B}$ with:

$$\mathbf{X} = \begin{pmatrix} \vdots \\ d_u^i \\ d_v^i \\ \vdots \end{pmatrix} \tag{III.140}$$

The solution is achieved using a LU factorization. This algorithm enables the definition of the node velocity in the body meshes:

$$\mathbf{v}_{mesh} = \frac{d_u^i}{dt}\mathbf{u}_i + \frac{d_v^i}{dt}\mathbf{v}_i + \frac{d_w^i}{dt}\mathbf{n}_i \tag{III.141}$$

III.4.2 Free surface mesh morphing

Regarding the free surface mesh morphing, the spring analogy method is not performed. It would lead to important CPU times, consequently another approach is chosen: an interpolation scheme of the free surface node displacement based on radial basis functions (RBF) [45, 142].

The interpolation function s is expressed by:

$$s(\mathbf{x}) = \sum_{j=1}^{N_c} \alpha_j \psi(\left\|\mathbf{x} - \mathbf{x}_j^c\right\|) + p(\mathbf{x}) \tag{III.142}$$

with:

– ψ the radial basis function defined by:

$$\psi(x) = x^2 log(x) \tag{III.143}$$

- p an unknown first degree polynomial;
- $\mathbf{x}_j^c$ the control nodes where the displacements are known;
- N_c the number of control nodes;
- α_j the unknown coefficients.

α_j and p are determined by the interpolation conditions:

$$s(\mathbf{x}_j^c) = d_j^c \tag{III.144}$$

$$\sum_{j=1}^{N_c} \alpha_j q(\mathbf{x}_j^c) = 0 \tag{III.145}$$

where d_j^c is the known displacement of the node j and q denotes all polynomials with a degree less or equal than that of p.

Finally, these equations form a linear system of size $N_c + 3$:

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0}_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{d}^c \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{III.146}$$

with M the matrix containing the evaluation of the radial basis function such as $M_{ij} = \psi(\left\|\mathbf{x}_i^c - \mathbf{x}_j^c\right\|)$ and P is the matrix with row j is given by the vector $\left(1, x_j^c, y_j^c\right)$. $\boldsymbol{\alpha}$ is the vector of the coefficients α_j and $\boldsymbol{\beta}$ the vector of the coefficients of p.

The coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are computed twice: a first time for the displacements along the x-axis and a second time along the y-axis. Two interpolation functions are obtained: s^x and s^y. Then, the node velocity in the free surface mesh is figured out by solving the following system of equations:

$$\begin{cases} \mathbf{v}_{mesh} \cdot \mathbf{n}_{FS} & = \dfrac{\partial \phi}{\partial n_{FS}} \\ \mathbf{v}_{mesh} \cdot {}^e\mathbf{x}_e & = \dfrac{s^x(\mathbf{x}_i)}{dt} \\ \mathbf{v}_{mesh} \cdot {}^e\mathbf{y}_e & = \dfrac{s^y(\mathbf{x}_i)}{dt} \end{cases} \tag{III.147}$$

III.5 Incident wave models

In the decomposition (III.13), the incident components are assumed to be known. Two models of incoming wave fields are used in *WS_ CN*:

- The Airy wave theory:

$$\phi^I(\mathbf{x}, t) = \sum_{j=1}^{N_{waves}} \frac{A_j g}{\omega_j} \frac{\cosh[k(z+D)]}{\cosh(kD)} \sin(\mathbf{k}_j \cdot \mathbf{x} - \omega_j t + \phi_j) \tag{III.148}$$

  where A_j, ω_j, $\mathbf{k}_j$ and ϕ_j represent the wave amplitude, the wave frequency, the wave vector and the phase of the j^{th} Airy wave. N_{waves} is the total number of waves.

– The stream function theory of Rienecker and Fenton [143]:

$$\phi^I(\mathbf{x},t) = \sum_{j=1}^{N_{RF}} B_j \frac{\cosh[jk(z+D)]}{\cosh(jkD)} \sin[j(\mathbf{k}_{RF}\cdot\mathbf{x} - \omega_{RF}t + \phi_{RF})] \tag{III.149}$$

where B_j and N_{RF} are the coefficients and the order of the Fourier series. ω_{RF}, $\mathbf{k}_{RF}$ and ϕ_{RF} represent the wave frequency, the wave vector and the phase of the Rienecker and Fenton's wave.

The Airy wave model is linear whereas the Rienecker and Fenton's wave theory is nonlinear and therefore more consistent with the weakly nonlinear hydrodynamic solver used in this PhD work.

The dispersion relation for a wave of frequency ω and wave number is:

$$\omega^2 = gk\tanh(kh) \tag{III.150}$$

where h is the water depth and g the gravity constant.

The incident wave field is present from the starting time of the simulation. To avoid the generation of non-physical and spurious waves due to the abrupt appearance of the bodies in the flow, a ramp function is applied on the body conditions (III.30) and (III.105):

$$f(t) = \begin{cases} 0 & \text{when } t \leqslant T_1 \\ 1 & \text{when } t \geqslant T_2 \\ \frac{1}{2}\left[1 - \cos\left(\pi\frac{t-T_1}{T_2-T_1}\right)\right] & \text{otherwise} \end{cases} \tag{III.151}$$

with T_1 and T_2 the starting and final time of application of the ramp function.

III.6 Absorbing numerical beach

To ensure the permeability condition on the numerical tank walls (III.22) and to avoid the wave reflection, a so-called absorbing numerical beach is applied. Its length is often equal to the greatest wave length of the incident wave field. A damping coefficient ν is added in the free surface boundary conditions:

$$\begin{cases} \dfrac{\delta\eta^P}{\delta t} = -\dfrac{\partial\eta^I}{\partial t} - \nabla\phi^I\cdot\nabla\eta^I - (\nabla\phi^I - \mathbf{v}_{mesh})\cdot\nabla\eta^P - \nabla\phi^P\cdot\nabla\eta^I + \dfrac{\partial\phi^I}{\partial z} + \dfrac{\partial\phi^P}{\partial z} \\ +\eta^P\left(\dfrac{\partial^2\phi^I}{\partial z^2} - \nabla\eta^I\cdot\dfrac{\partial\nabla\phi^I}{\partial z})\right) - \nu\eta^P \quad \text{at } z = \eta^I(x,y,t) \\ \dfrac{\delta\phi^P}{\delta t} = -\dfrac{\partial\phi^I}{\partial t} - \dfrac{1}{2}\nabla\phi^I\cdot\nabla\phi^I - (\nabla\phi^I - \mathbf{v}_{mesh})\cdot\nabla\phi^P - g(\eta^I + \eta^P) \\ -\eta^P\left(\dfrac{\partial^2\phi^I}{\partial z\partial t} + \nabla\phi^I\cdot\dfrac{\partial\nabla\phi^I}{\partial z}\right) - \nu\phi^P \quad \text{at } z = \eta^I(x,y,t) \end{cases} \tag{III.152}$$

The expression of the damping coefficient is:

$$\nu(r) = \begin{cases} 0 & \text{when } r \leqslant r_0 \\ 1 & \text{when } r \geqslant R \\ \frac{1}{2}\left[1 - \cos\left(\pi\frac{r-r_0}{R-r_0}\right)\right] & \text{otherwise} \end{cases} \tag{III.153}$$

with r the distance of a node with respect to the center of the domain, R the radius of the cylindrical domain and $r_0 = R - L_{abs}$ with L_{abs} the length of the absorbing numerical beach.

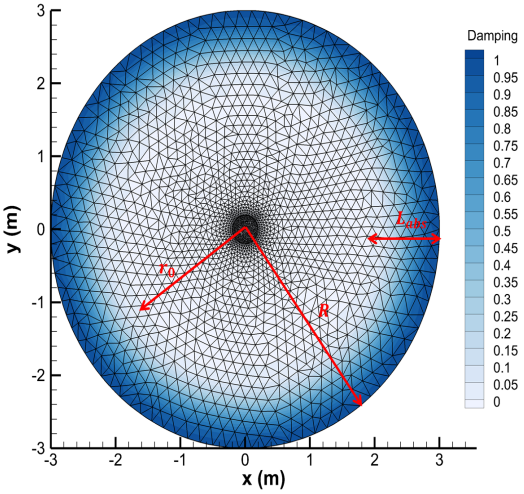


Figure III.12 – Example of a damping numerical beach with $L_{abs} = 1$ m and $R = 3$ m

III.7 State vector and time integration

The state vector of the weakly nonlinear potential flow theory based on the weak-scatterer hypothesis is:

$$\mathbf{Y}_{WSC} = \begin{pmatrix} \phi^P(FS) \\ \boldsymbol{\eta}^P \\ \mathbf{P}_{mesh} \\ \vdots \\ \boldsymbol{\eta}_j \\ \vdots \\ \dot{\boldsymbol{\eta}}_j \\ \vdots \end{pmatrix} \tag{III.154}$$

$\mathbf{P}_{mesh}$ represents the position of all the nodes of the mesh. Its time-derivative is $\mathbf{v}_{mesh}$ of

The time-differentiation of $\mathbf{Y}_{WSC}$ is obtained using:

- (III.29) for $\dot{\boldsymbol{\phi}}^P(FS)$ and $\dot{\boldsymbol{\eta}}^P$;
- (III.141) and (III.147) for $\mathbf{v}_{mesh}$;
- (III.113) for $\ddot{\boldsymbol{\eta}}_j$.

The time-stepping is achieved using a fourth-order explicit Runge-Kutta scheme with a fixed time step (I.94).

At the initial time $t = 0s$, the flow is assumed to be at rest so the scattered quantities are zeroed:

$$\boldsymbol{\phi}^P(FS) = \mathbf{0}_{N_{FS}\times 1} \tag{III.155}$$
$$\dot{\boldsymbol{\phi}}^P(FS) = \mathbf{0}_{N_{FS}\times 1} \tag{III.156}$$
$$\boldsymbol{\eta}^P(FS) = \mathbf{0}_{N_{FS}\times 1} \tag{III.157}$$
$$\dot{\boldsymbol{\eta}}^P(FS) = \mathbf{0}_{N_{FS}\times 1} \tag{III.158}$$

where N_{FS} is the number of nodes of the free surface mesh.

III.8 Free surface remeshing

The simulations of floating bodies with large amplitude motions involve important deformations of the mesh and the appearance of degenerated panels even if mesh morphing methods are available (cf. Figure III.13). This leads to numerical errors. Therefore, it becomes necessary to regenerate the mesh of the domain when the panel shape quality decreases. The regeneration of the wetted body surface meshes only requires the call to the mesh generator as presented in section III.3. The hydrodynamic solver is not affected by this change. Chauvigné [45] developed the remeshing of the bodies in his PhD work. In contrast, the remeshing of the free surface mesh has an impact on the hydrodynamic simulation. Indeed, two quantities of the state vector (III.154) are located at the free surface nodes: the perturbed velocity potential $\boldsymbol{\phi}^P(FS)$ and the scattered wave elevation $\boldsymbol{\eta}^P$. These quantities are known for the current free surface mesh but unknown for the new free surface mesh. An interpolation scheme based on third-order polynomial splines used for the computation of the surface gradients (cf. subsection III.2.5) is applied to evaluate $\boldsymbol{\phi}^P(FS)$ and $\boldsymbol{\eta}^P$ on the new free surface mesh from the current one. The free surface remeshing process unfolds in three steps:

- A *new* free surface mesh $(\mathbf{x}_i^{New})_{i\in[\![1\ ;\ N_{FS}^{New}]\!]}$ is created using the advance front method, the former (*old*) mesh $(\mathbf{x}_i^{Old})_{i\in[\![1\ ;\ N_{FS}^{Old}]\!]}$ is saved (Figure III.14a);
- For each node of this new mesh, the nearest node of the former mesh is searched (Figure III.14b);
- A third-order polynomial spline interpolation is applied to evaluate the perturbed velocity potential and the scattered wave elevation at the location of the new nodes, based on the nearest node and its neighboring nodes in the former mesh (Figure III.14c).

For example, for the wave elevation, the spline approximation is:

$$\eta^P(\mathbf{x}_i^{New}) = \sum_{j=0}^{N_v^{i,Old}+1} \alpha_j \psi\left(\left\|\mathbf{x}_i^{New} - \mathbf{x}_j^{Old}\right\|\right) + p(\mathbf{x}_i^{New}) \tag{III.159}$$

$N_v^{i,Old}$ refers to the number of first-order and second-order neighboring nodes of the nearest point in the former mesh of $\mathbf{x}_i^{New}$. $+1$ signifies that the nearest point of $\mathbf{x}_i^{New}$ in the former mesh is included. α_j and p are determined by the interpolation conditions

on the nodes $\mathbf{x}_j^{Old}$, neighbors of the nearest node in the former mesh of the node $\mathbf{x}_i^{New}$ in the new mesh:

$$s(\mathbf{x}_j^{Old}) = \eta^P(\mathbf{x}_j^{Old}) \tag{III.160}$$

$$\sum_{j=1}^{N_v^{i,Old}+1} \alpha_j q(\mathbf{x}_j^{Old}) = 0 \tag{III.161}$$

The same notations are used here as in subsection III.2.5.

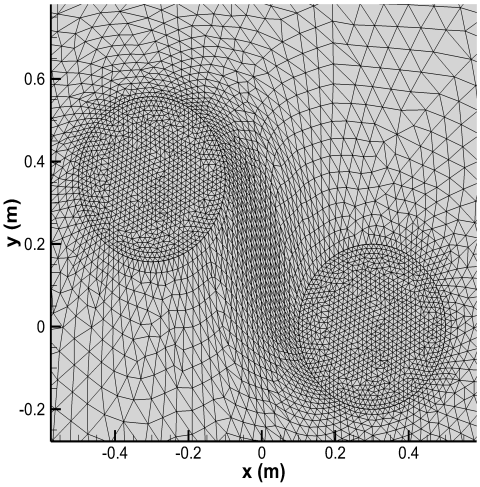
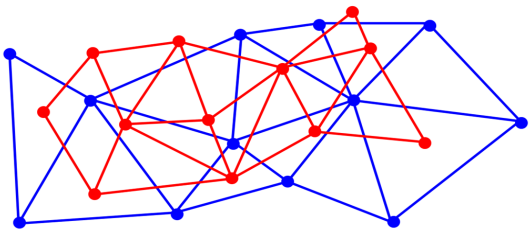
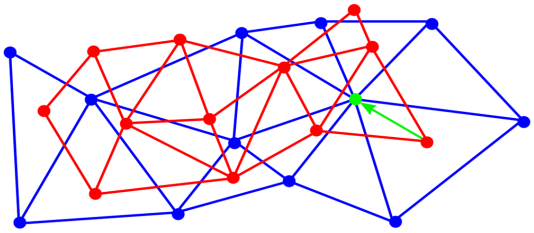


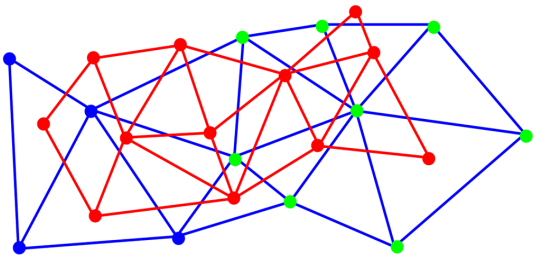
Figure III.13 – Mesh distortion due to the translation of a cylinder along the y-axis (left). The second cylinder is fixed (right).



(a) Former mesh (blue) and new mesh (red)



(b) One node in the new mesh (arrow tail) and its nearest point in green in the former mesh (arrow head)



(c) Nodes in the former mesh used for the interpolation (in green)

Figure III.14 – The three steps of the interpolation scheme on the free surface mesh

The case of a cylinder in waves with a heaving motion is used for the validation of the free surface remeshing. Its mass is 12.88 kg, its radius 0.2 m, its length 0.2 m and its draft 0.1 m. An incoming regular wave of amplitude 0.005 m and wave frequency 8 rad/s is used. A wave probe is located at the position $(0.4, 0, 0)$. The mesh and time step convergences were checked and lead to a mesh of 10 000 panels and a time step of 0.01 s. Two sets of numerical results are compared based on when the free surface remeshing process is called:

- When it is necessary: the size or the shape of a panel on the free surface is too poor (20 remeshings for 20 s);
- At every time step: the remeshing is forced (2000 remeshings in total).

Figure III.15 presents the comparison of the time series of the cylinder motion and the wave elevation at the wave probe. The numerical results are similar even if few differences are observed. They are caused by the interpolations during the free surface remeshing algorithm which involves numerical errors.

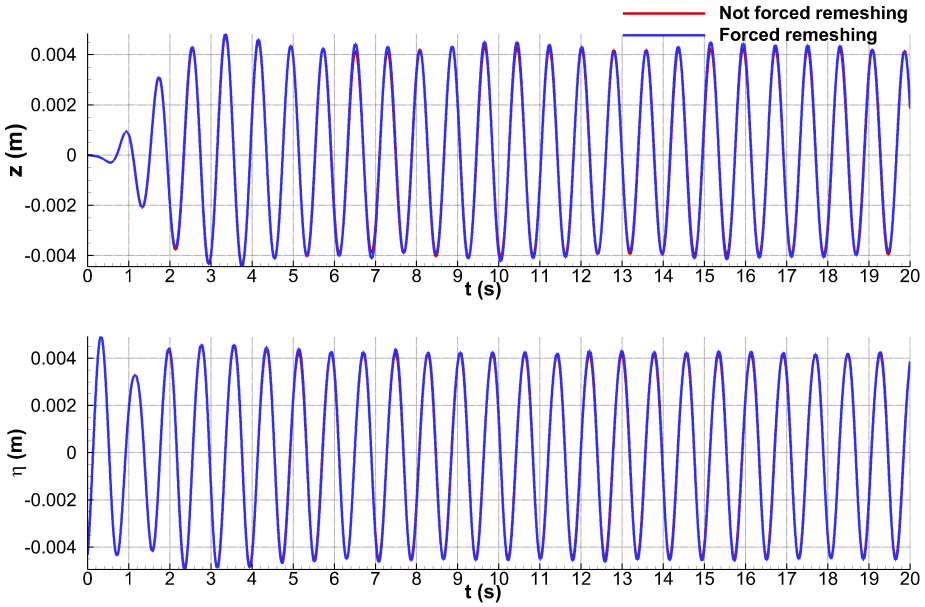


Figure III.15 – Comparison of time series of the heave motion (top) and the wave elevation (bottom) from numerical results with and without forced free surface remeshing processes

III.9 Gaussian filter

The free surface remeshing is not able to remove all the numerical errors on the free surface mesh. It is necessary to use a filter to remove the high-frequency non-physical waves or saw-tooth waves. Letournel [144] implemented a Gaussian filter on the perturbed wave elevation in *WS_CN*. The filter is classically used every five time steps. Its formula is:

$$\eta^P(\mathbf{x}_i) = \frac{1}{\alpha}\int_S \eta^P(\mathbf{x}) G_{filter}(\mathbf{x}_i, \mathbf{x})\, dS \tag{III.162}$$

with

$$\alpha = \int_S G_{filter}(\mathbf{x}_i, \mathbf{x})\, dS \tag{III.163}$$

$$G_{filter}(\mathbf{x}_i, \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}\right) \tag{III.164}$$

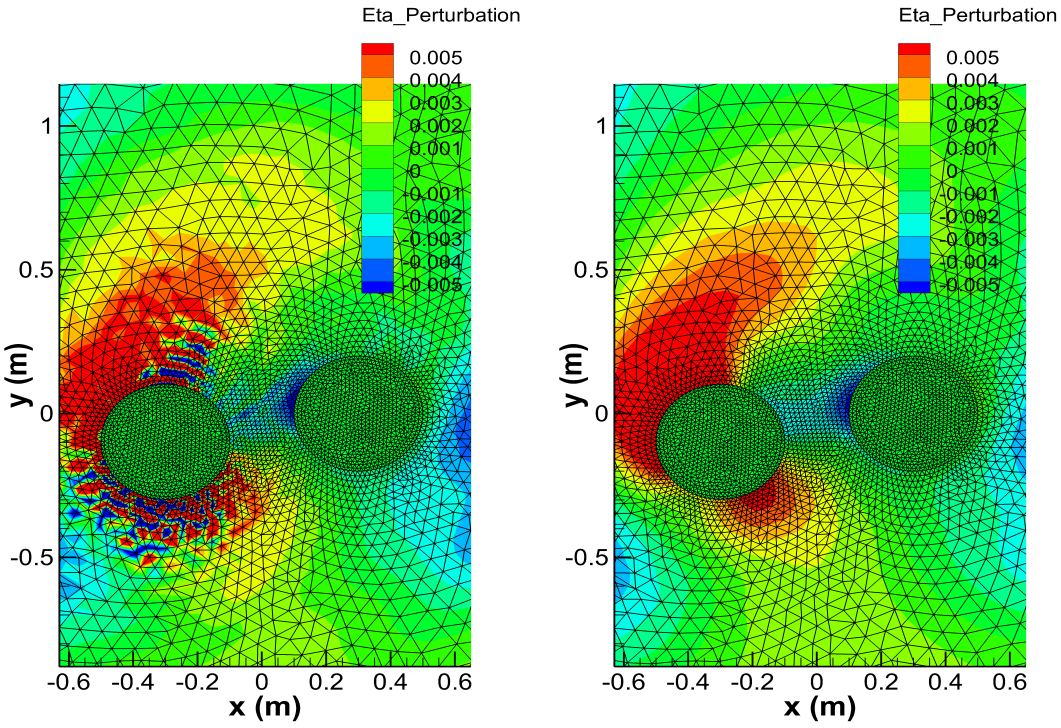
The standard deviation σ is fixed to $\frac{1}{2}\sqrt{\mathcal{A}_i}(1+\nu)$ where $\mathcal{A}_i$ is the area associated with the node i and ν the damping of the absorbing numerical beach (III.153).

After discretization, (III.162) becomes:

$$\eta^P(\mathbf{x}_i) = \frac{1}{\alpha}\sum_{j=1}^{N_v^i} \eta^P(\mathbf{x}_j) G_{filter}(\mathbf{x}_i, \mathbf{x}_j)\mathcal{A}_j \tag{III.165}$$

where N_v^i is the number of first-order and second-order neighboring nodes of the node i.

The effect of the filtering is outlined in Figure III.16. The free surface mesh is regenerated when it is necessary, consequently the mesh is of good quality. Nevertheless, the sawtooth waves are clearly visible in Figure III.16a. The Gaussian filter removes them as it can bee seen in Figure III.16b.



(a) Without filtering

(b) Filtering every five time steps

Figure III.16 – Effect of the filtering of the scattered wave elevation on a simulation with a translation cylinder along the y-axis (left) and a fixed cylinder (right).

III.10 Parallelization

Several techniques are used to decrease the CPU time of *WS_ CN*. In the PhD work of Letournel [44]:

- The (xOz) plane is used as symmetry plane to divide the number of panels by two if the physical problem is symmetric;
- The flatness of the sea bottom allows not to mesh this surface because it is used as a symmetry plane;
- A far-field approximation of the influence coefficients enables to speed-up their computation;

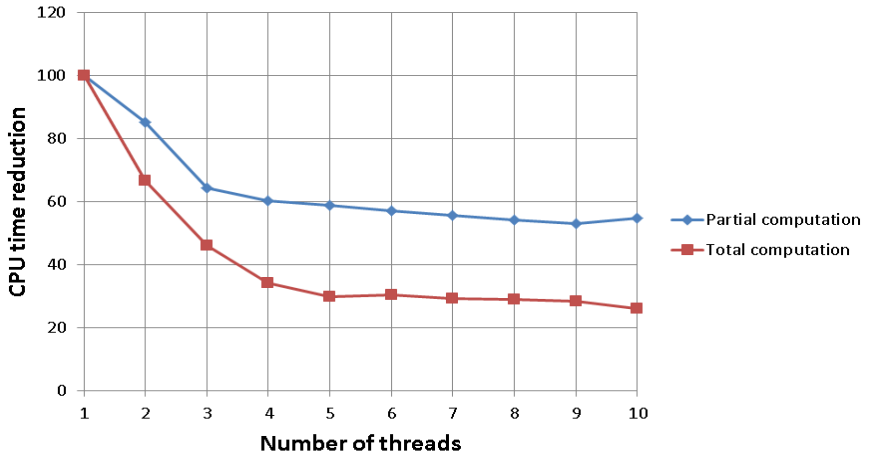
Chauvigné [45] developed a partial computation of the influence coefficients. The parts of the mesh which only have a small relative motion between each other keep the same influence coefficients. For example the panels of the numerical tank wall are almost still during a simulation, so their influence on themselves is constant and does not need to be evaluated again.

Three new developments were added:

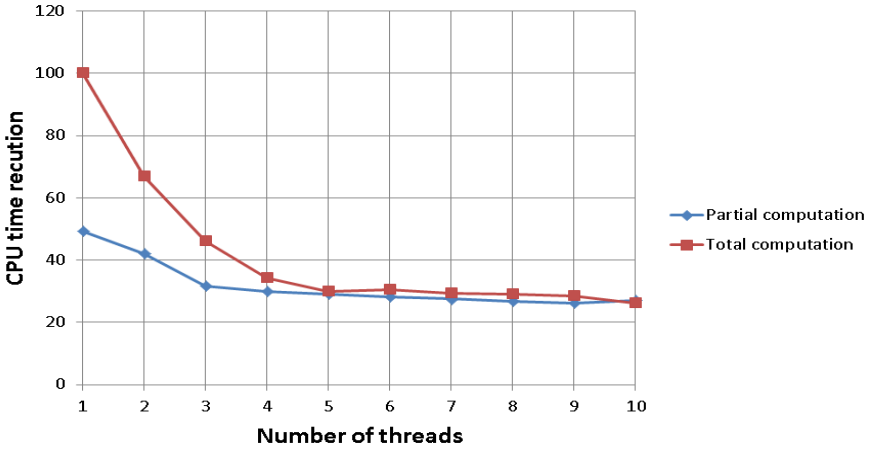
- The use of the sparse linear system solver *PARDISO* in the body mesh morphing algorithm (cf. subsection III.4.1), as it was done in subsection III.3.3;
- The respect of the column-major order because *WS_ CN* is implemented in Fortran;
- The parallelization using *OpenMP*[8] (Open Multi-Processing) of the computation of the influence coefficients.

Figure III.17 presents the results of the parallelization for a simulation with two surface-piercing cylinders and a mesh of 8300 panels and 4550 nodes and without free surface remeshing. The top figure shows that the time saving is more important if the influence coefficients are completely evaluated. The interest of the partial computation of the influence coefficients depends on the relative velocity between the mesh parts and so of the dynamics of the problem. The bottom figure shows that the same final time is reached with the two methods once the main effects of the parallelization are obtained. From 5 threads, the interest of the partial computation of the influence coefficients is negligible. The partial computation is interesting for a sequential computation as it lowers the simulation time of 50 %.

[8] https://www.openmp.org



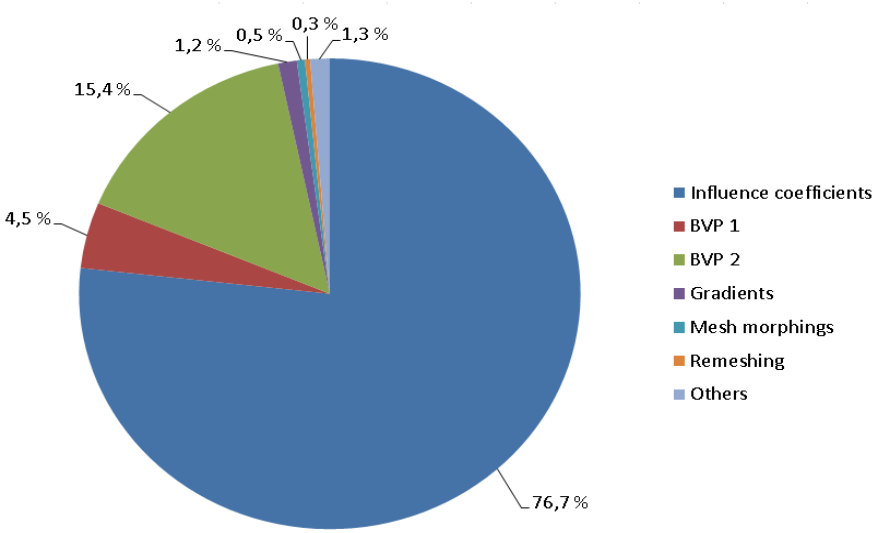
(a) Value of 100 for one thread



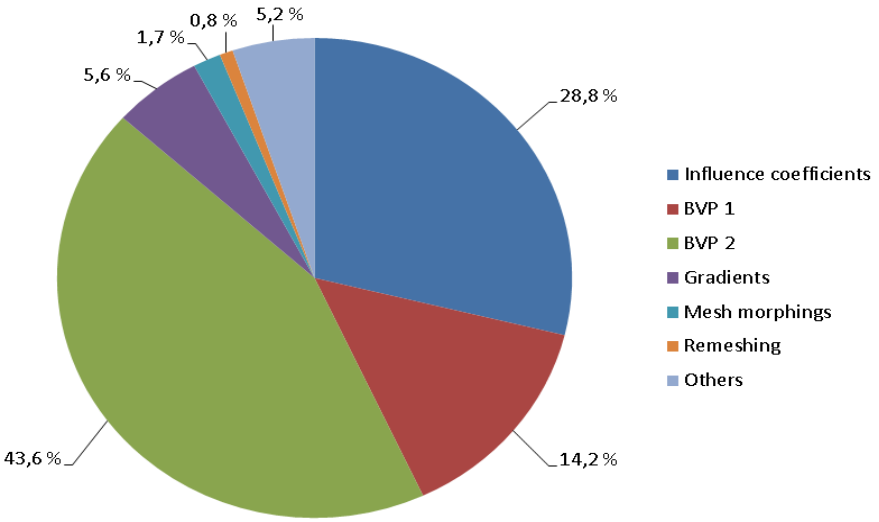
(b) Value of 100 for one thread and for the total computation

Figure III.17 – Influence of the parallelization on the CPU time reduction for a partial and total calculation.

The following pie charts outline the profiling of WS_CN with one and ten threads. The reduction of the share of the influence coefficient computation is compensated by the inversion of the linear systems for the first and second boundary value problems. Hence, further efforts towards computational time reduction should focus on the algorithms for solving the BVP problems.



(a) One thread



(b) Ten threads

Figure III.18 – Profiling of WS_CN

The same validation case as in III.8 is used to compare the numerical results obtained with and without the parallelization. Figure III.19 presents the comparison of the time series of the cylinder motion and the wave elevation at the wave probe using a sequential computation (nth = 1) and a parallel computation with 5 threads (nth = 5). The results are identical.

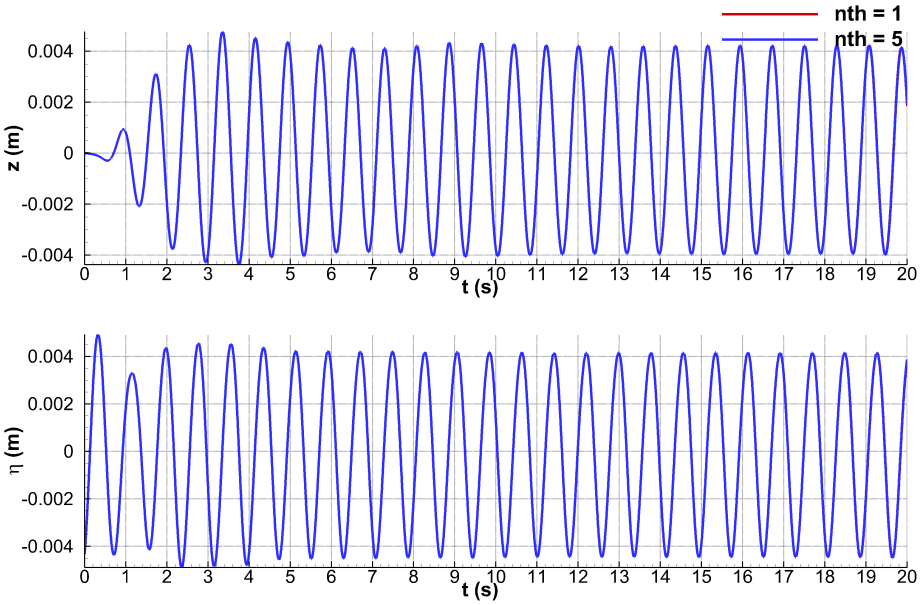


Figure III.19 – Comparison of time series of the heave motion (top) and the wave elevation (bottom) from numerical results using a sequential (nth = 1) and parallel (nth = 5) computation

III.11 Extension to multibody simulations

Letournel [44] and Chauvigné [45] only run single body simulations with *WS_CN*. The extension to multibody simulations was necessary to perform marine operation simulations involving at least two bodies subject to hydrodynamic loads. It unfolds in two steps:

- The extension of the mesh generator;
- The extension of the hydrodynamic solver.

These developments were directly presented by taking into account a multibody format for writing the equations in the previous sections. It stays to validate them both in forced and free motions.

III.11.1 Forced motion

The test cases presented by Watai et al. [145] are used for the validation of the capacity of *WS_CN* to deal with multiple interacting floating bodies.

III.11.1.1 Presentation of the test cases

Watai et al. [145] conducted experimental tests at the University of Sao Paulo (Brazil) with two cylinders in regular waves:

- One fixed cylinder, named *Body 1*;
- One moving cylinder, named *Body 2*, with a large prescribed harmonic motion along the x-axis.

The characteristics of these cylinders are listed in Table III.3.

Diameter (m)	0.4
Height (m)	0.36
Draft (m)	0.2
Position of *Body 1* (m)	$(0, 0.3, 0)$
Initial position of *Body 2* (m)	$(0, -0.3, 0)$

Table III.3 – Characteristics of the two cylinders

Three wave probes (*WP1*, *WP2*, *WP3*) are also used. Their position is given in Table III.4.

WP1 (m)	$(0, 0.7, 0)$
WP2 (m)	$(0, 0, 0)$
WP3 (m)	$(0, -0.7, 0)$

Table III.4 – Wave probe locations

The incident wave field coming from the negative y, so *WP1* is upstream and *WP3* downstream. The positions of every device are highlighted in Figure III.20.

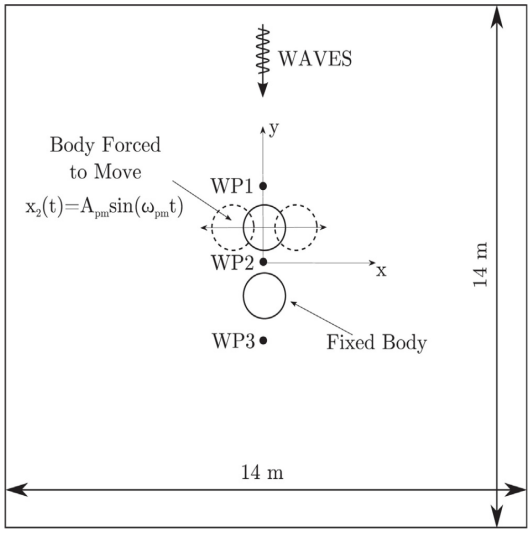


Figure III.20 – Sketch of the top view of the experimental set-up (from [145])

Watai et al. [145] considered four regular waves, listed in Table III.5.

For each regular wave, three wave frequencies, ω_{pm}, of the harmonic prescribed motion of *Body 2* were selected. The motion amplitude, A_{pm} is fixed at 0.37 m. Table III.6 presents the twelve test cases.

III.11.1.2 Results

A cylindrical domain is used for the simulations, with a radius of two wave lengths (2λ). One wave length is used for the numerical absorbing beach. Bodies are meshed

ID	ω (rad/s)	λ (m)	A (m)
Reg 1	6.400	1.506	0.0115
Reg 2	6.800	1.330	0.0100
Reg 3	7.000	1.259	0.0095
Reg 4	7.200	1.190	0.0090

Table III.5 – Regular wave characteristics. λ and A represent the wave length and the wave amplitude.

Case	Wave	ω_{pm} (rad/s)	A_{pm} (m)
1	Reg 1	0.427	0.37
2	Reg 1	0.213	0.37
3	Reg 1	0.107	0.37
4	Reg 2	0.453	0.37
5	Reg 2	0.227	0.37
6	Reg 2	0.113	0.37
7	Reg 3	0.467	0.37
8	Reg 3	0.233	0.37
9	Reg 3	0.117	0.37
10	Reg 4	0.480	0.37
11	Reg 4	0.240	0.37
12	Reg 4	0.120	0.37

Table III.6 – Characteristics of the twelve test cases

at real scale. The ramp function (III.151) is applied for a wave period from the start of the simulation.

The mesh convergence for Case 1 is presented in Figures III.21 and III.22. It is noticed that the mesh convergence is slow along the x axis of the hydrodynamic loads compared to the two other directions. A mesh of 20 000 panels is used. The time step convergence is presented in Figures III.23 and III.24. It is also slow along the x-axis. A time step of 0.005 s is used hereinafter. Figures III.25 and III.26 show the mesh of the domain for Case 1.

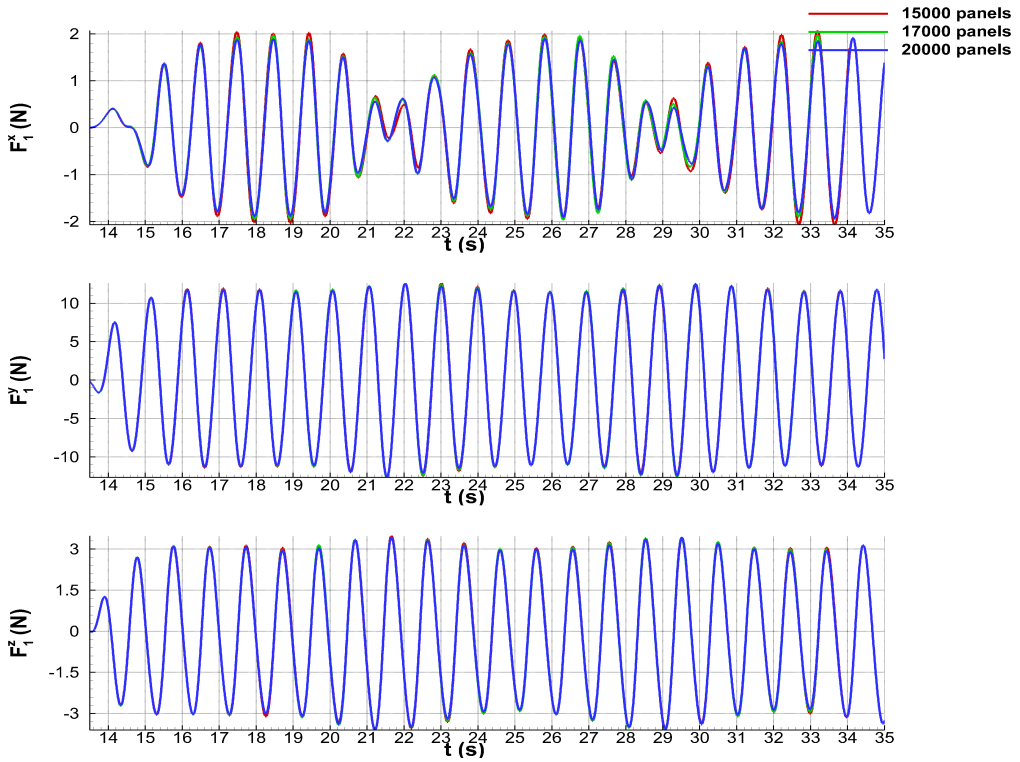


Figure III.21 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical results for different meshes for Case 1

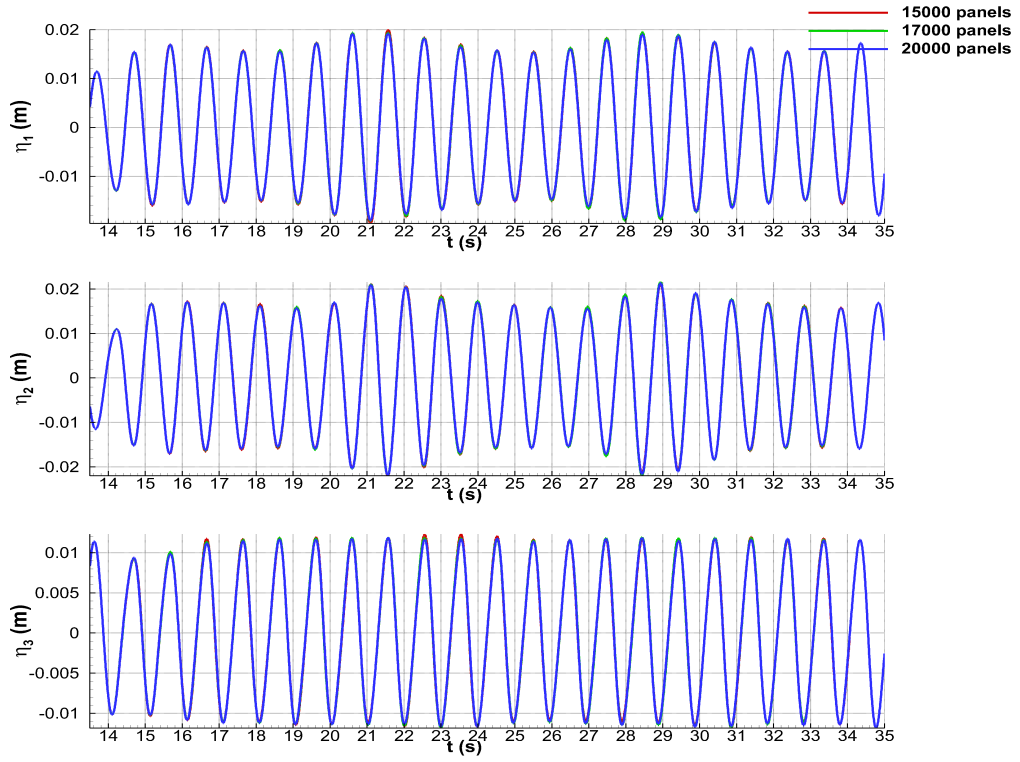


Figure III.22 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical results for different meshes for Case 1

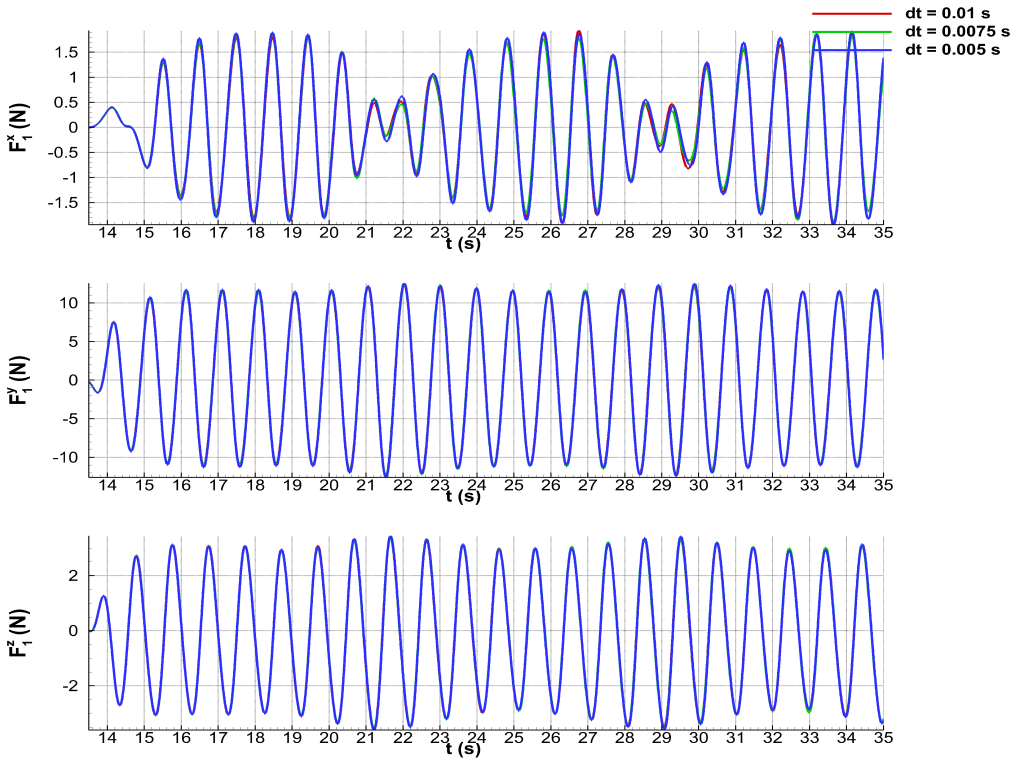


Figure III.23 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical results for different time steps for Case 1

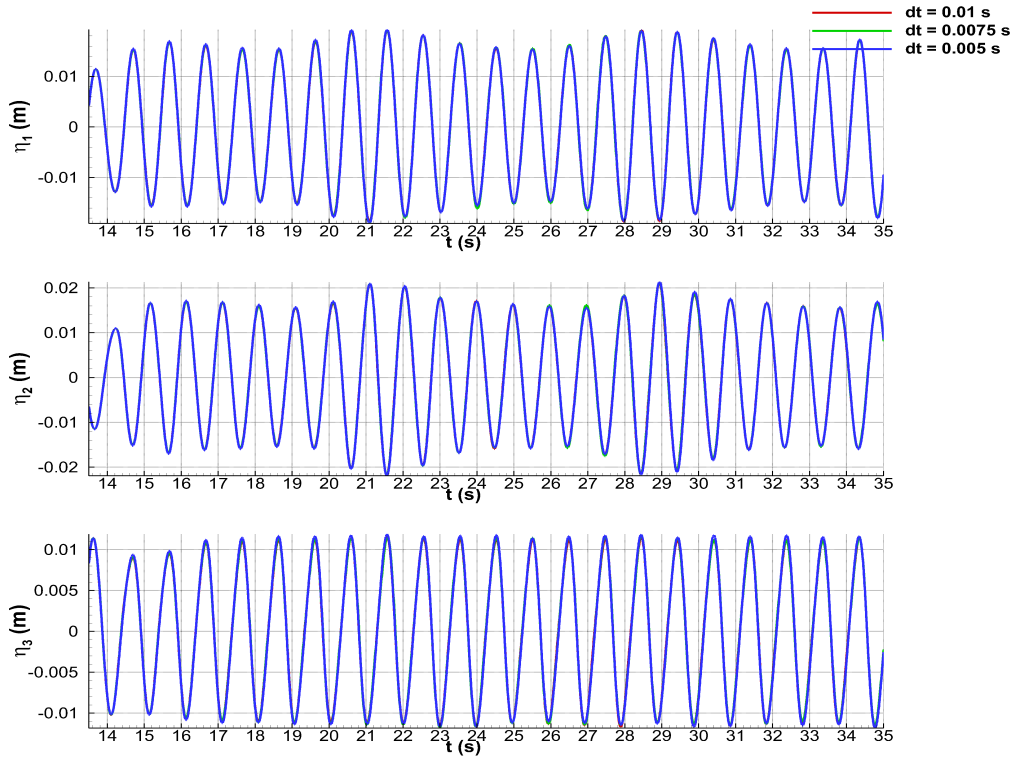


Figure III.24 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical results for different time steps for Case 1

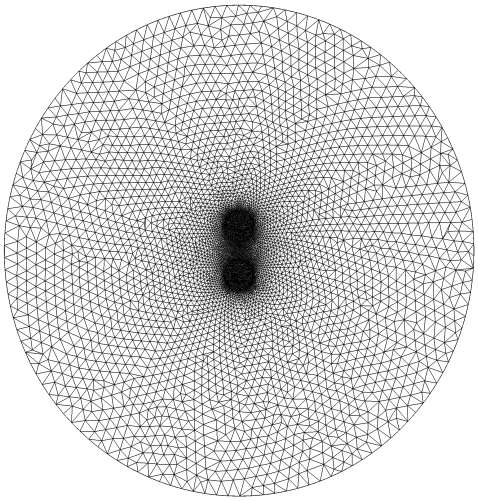


Figure III.25 – Top view of the mesh for Case 1

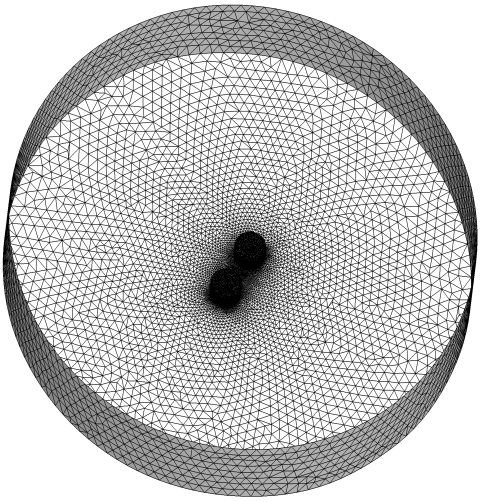


Figure III.26 – Bottom view of the mesh for Case 1

The prescribed harmonic motion of *Body 2* for Case 1 is shown in Figure III.27.

Figure III.27 – Prescribed harmonic motion of *Body 2* for Case 1

Numerical results of *WS_ CN* and experimental data of Watai et al. [145] are compared by using the wave elevations at the three wave probes and the hydrodynamic loads

on *Body 1* after removal of the hydrostatic part of the Bernoulli's equation (III.94). The results of Case 1 are presented in Figure III.28 for the loads on *Body 1*, Figure III.29 for the wave elevations and Figure III.30 for the loads on *Body 2*. Case 5 and 10 were also studied and comparisons are presented after digitization of the data of Watai et al. [145]. Hydrodynamic loads on *Body 1* are presented for Case 5 in Figure III.31 and for Case 10 in Figure III.32. Experimental and numerical signals were synchronized *a posteriori*, as the incident wave signals are not provided in [145].

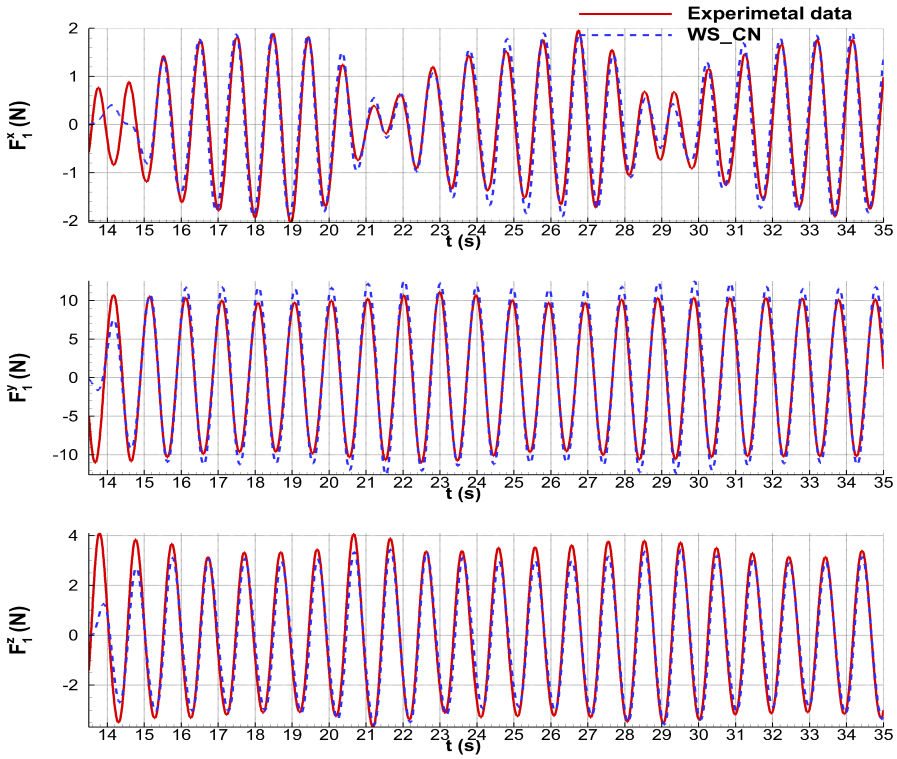


Figure III.28 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 1

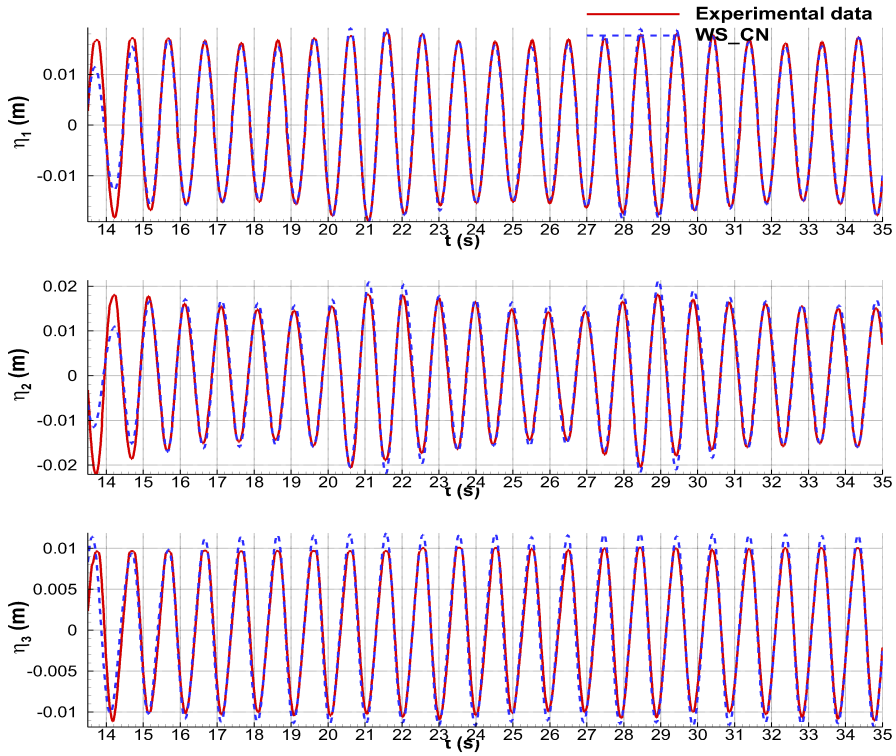


Figure III.29 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical and experimental results for Case 1

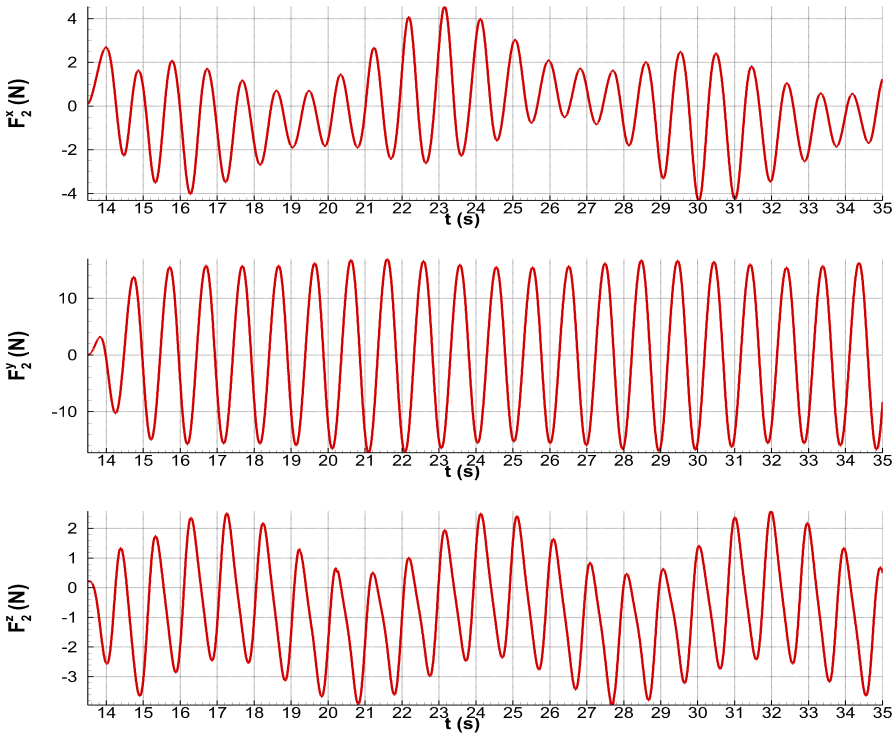


Figure III.30 – Time series of the hydrodynamic loads on *Body 2* from numerical results for Case 1

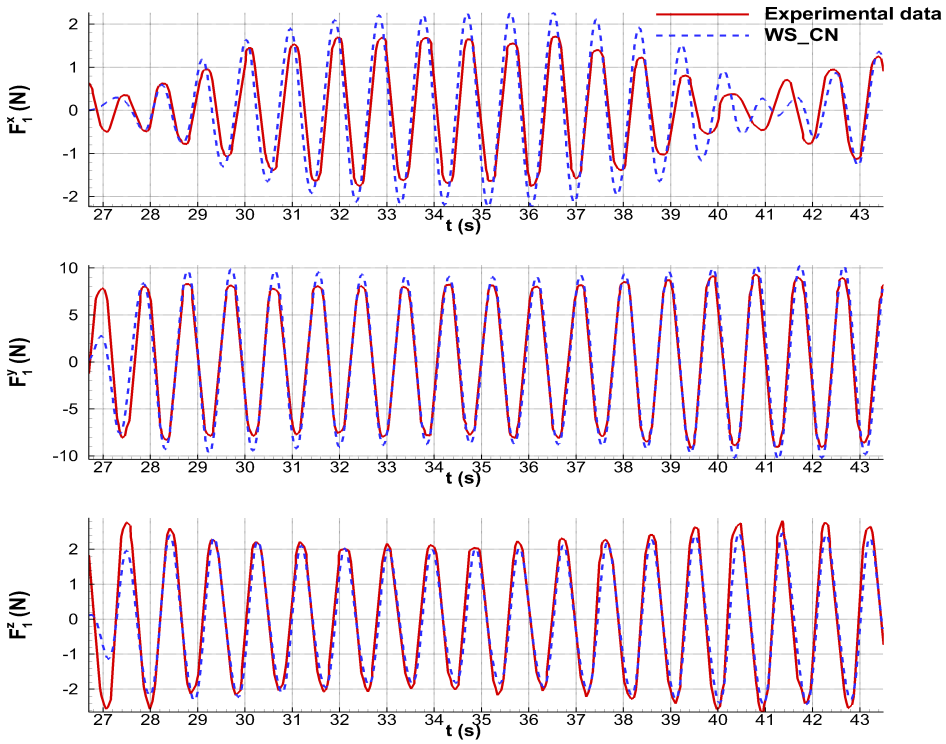


Figure III.31 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 5

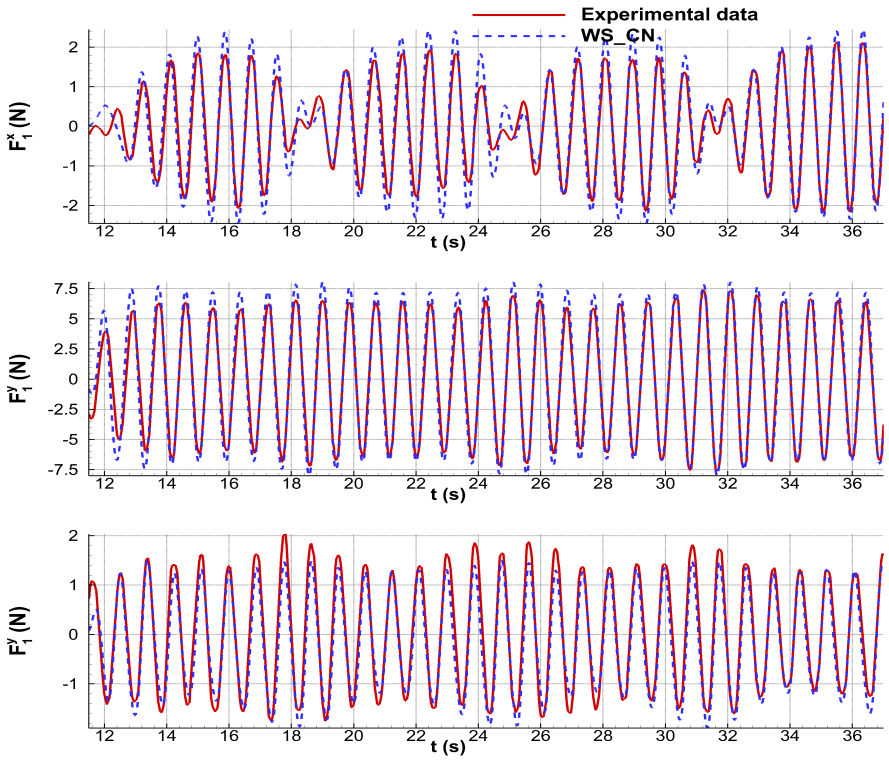


Figure III.32 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 10

The time series of the hydrodynamic loads show an overall good agreement between the experimental data and *WS_CN*. The modulation of the amplitude of F_1^x in Figures III.28, III.31 and III.32 is well captured. The low frequency oscillations match the harmonic motion of *Body 2*. In the three test cases, one can note that *WS_CN* appears to overestimate the amplitude of the loads, especially for F_1^x and F_1^y while an underestimation is observed for F_1^z. Regarding the wave elevation, the modulations in η_1 and η_2 in Figure III.29 is observed in the numerical computations. The same overestimations, as the hydrodynamic loads, are noticed.

Overestimations are also present in the work of Watai et al. [145] and are likely due to the main hypotheses of the fluid solver (inviscid fluid, irrotational flow). Nevertheless Watai et al. [145] needed much less panels (4010 panels in total for Case 1) than the present fluid method (around 20 000 on the free surface mesh for Case 1). These authors used a body-exact approximation (using a linearized free surface at $z = 0$) whereas a weak-scatterer hypothesis is applied here, which could explain the need of many more panels and so a slower mesh convergence. The use of many remeshing processes, because of the large relative motion, leads to interpolations of the physical quantity on the free surface and could deteriorate the accuracy. In Case 1, the total mesh is regenerated around 200 times during the simulations (5 % of the number of times steps). Figure III.33 shows the effect of a forced remeshing process at every time step. It is only noticeable along the x-axis. The remeshing processes cannot be deleted otherwise large numerical error arise. Therefore, the effect of the required 200 remeshing processes cannot be quantified. Another possible explanation of the differences between the numerical results and the experimental data is the use of the Gaussian filtering. In our simulations, the filtering is applied every five time steps. If the filtering is not used, sawtooth waves appear (cf. Figure III.16) which lead to numerical errors. The effect of the modification of the frequency of call of the Gaussian filtering is shown in Figure III.34. Only few differences are present.

For the sake of illustration, the perturbed wave pattern for Case 1 at $t = 24.595$ s is displayed in Figure III.35.

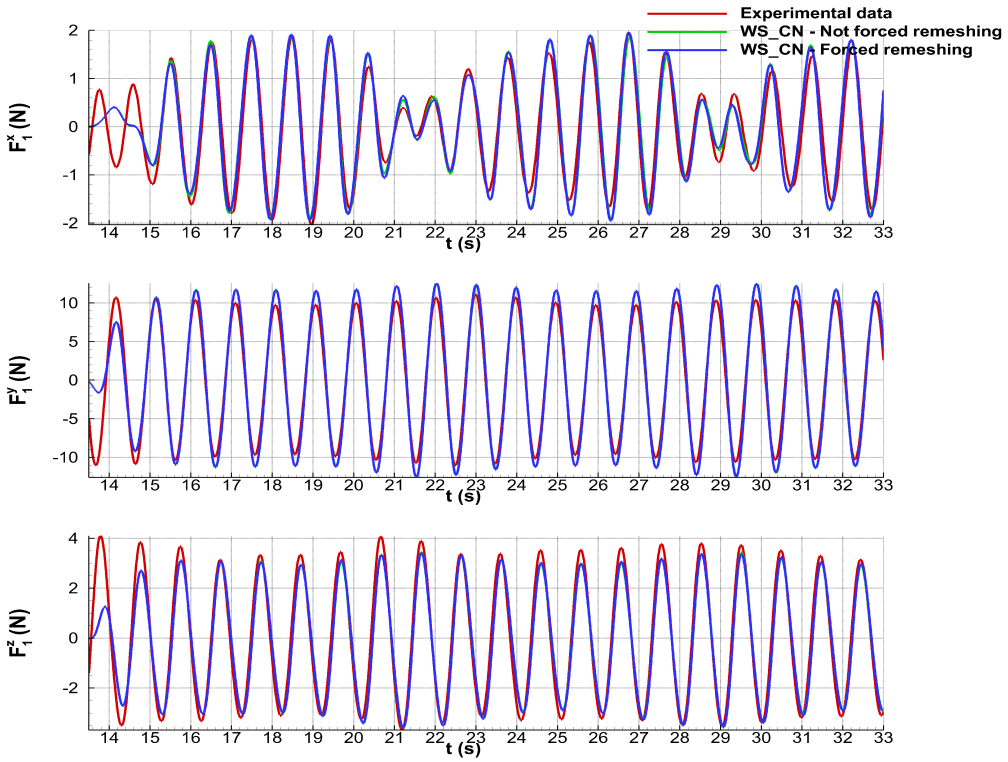


Figure III.33 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 1 with and without forced remeshing processes

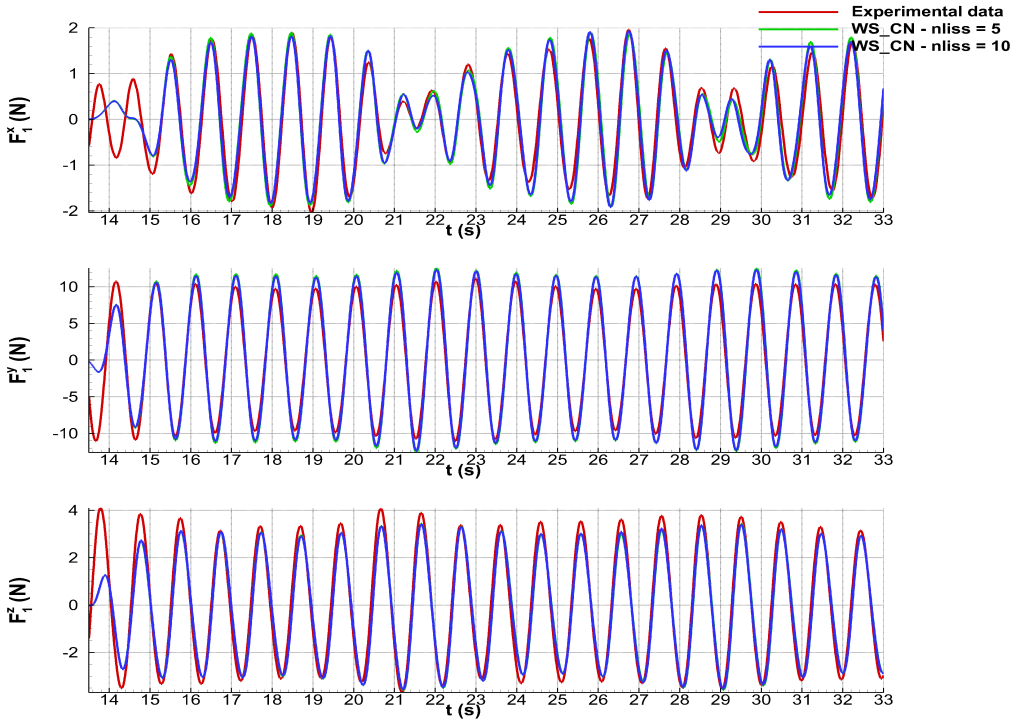


Figure III.34 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 1 for different frequencies of call of the Gaussian filtering (*nliss*)

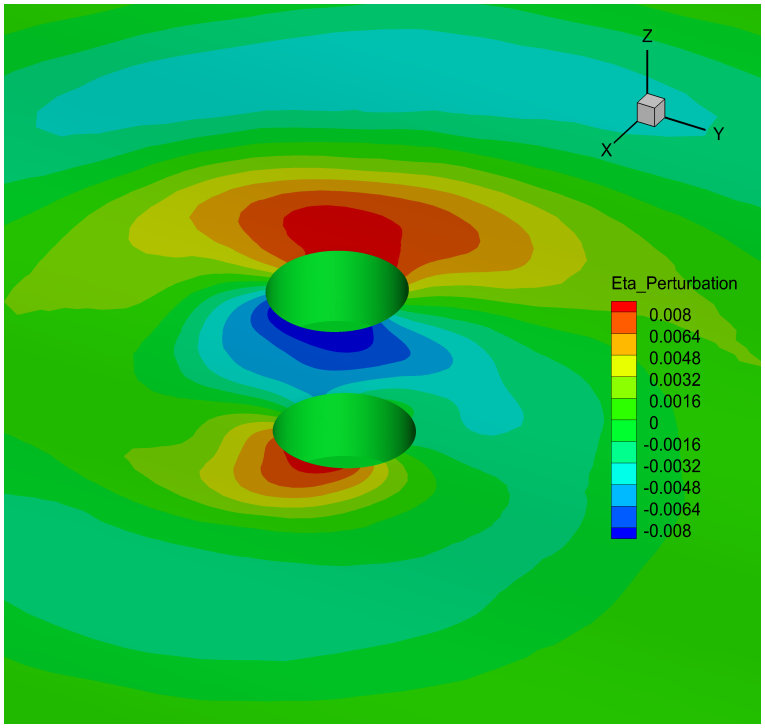


Figure III.35 – Perturbed component of the wave pattern (η^P) at $t = 24.595$ s

III.11.1.3 Impact of the weak-scatterer hypothesis and influence of the free surface nonlinearities

Figure III.36 presents the incident and perturbed components of the wave elevation at the three wave probe locations. The weak-scatterer hypothesis (III.14) is not fulfilled at every moment and everywhere even if the scattered component is lower than the incident one. However, the incident wave field is always of small amplitude and steepness ($\epsilon \leqslant 1\%$) and the use of a body-exact approximation would be fully consistent. Watai et al. [145] used this approach.

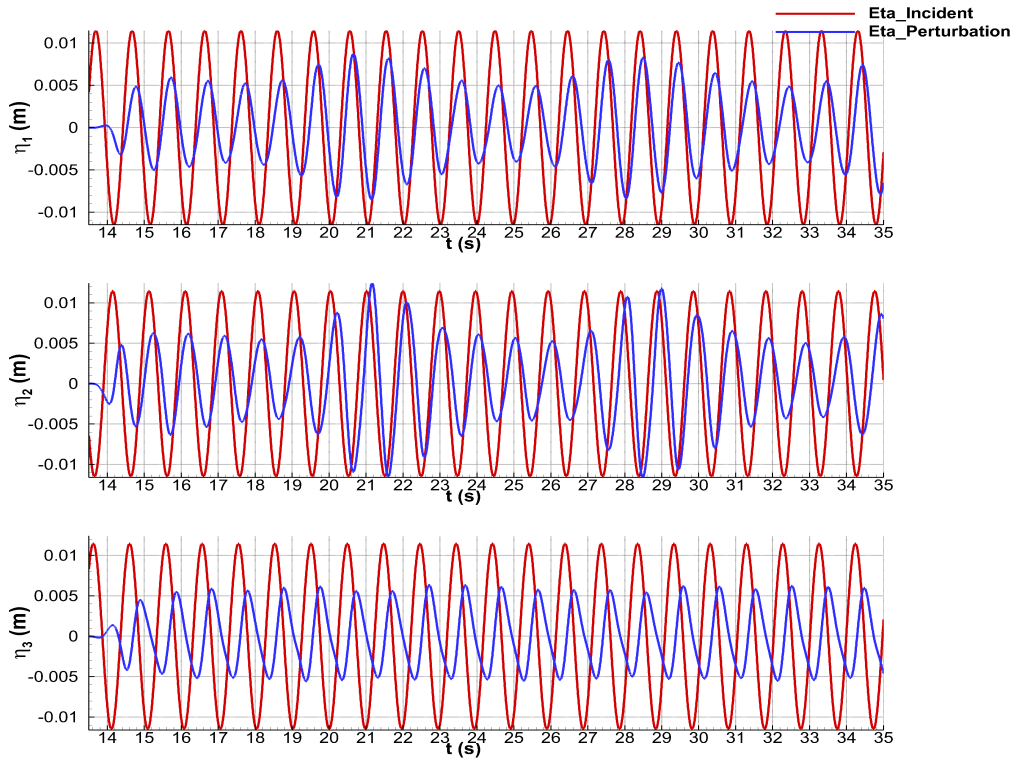


Figure III.36 – Comparison of time series of the incident and scattered components of the wave elevation at *WP1*, *WP2* and *WP3* from numerical results for Case 1

WS_CN may be switched to a body-exact approximation in order to evaluate the interest of the weak-scatterer method for Case 1. The mesh convergence in case of a body exact approximation is presented in Figures III.37 and III.38. The mesh with 12 000 panels is chosen. The time step is 0.005 s. The body-exact method requires less panels than the weak-scatterer method (with 20 000 panels). Figures III.39 and III.40 present the comparison between the experimental data and both the weak-scatterer and body-exact approximations. Only small differences are noticed between the weak-scatterer and the body-exact approximations.

Finally, the weak-scatterer method seems to be slightly more accurate than the body-exact method. This confirms that the weak-scatterer method leads to a body-exact approximation when the incident wave stiffness and the free surface nonlinearities are small as required by the body-exact approximation.

If the wave steepness (III.15) is increased, and so the free surface nonlinearities too, then the differences between the two methods increase. Numerical results are showed in Figures III.41 and III.42 for a wave amplitude twice higher than for Case 1 and in Figures III.43 and III.44 for a wave amplitude three times bigger than Case 1. The wave steepness for Case 1 is $\epsilon = 0.048$. The wave length is kept constant, only the wave amplitude is modified.

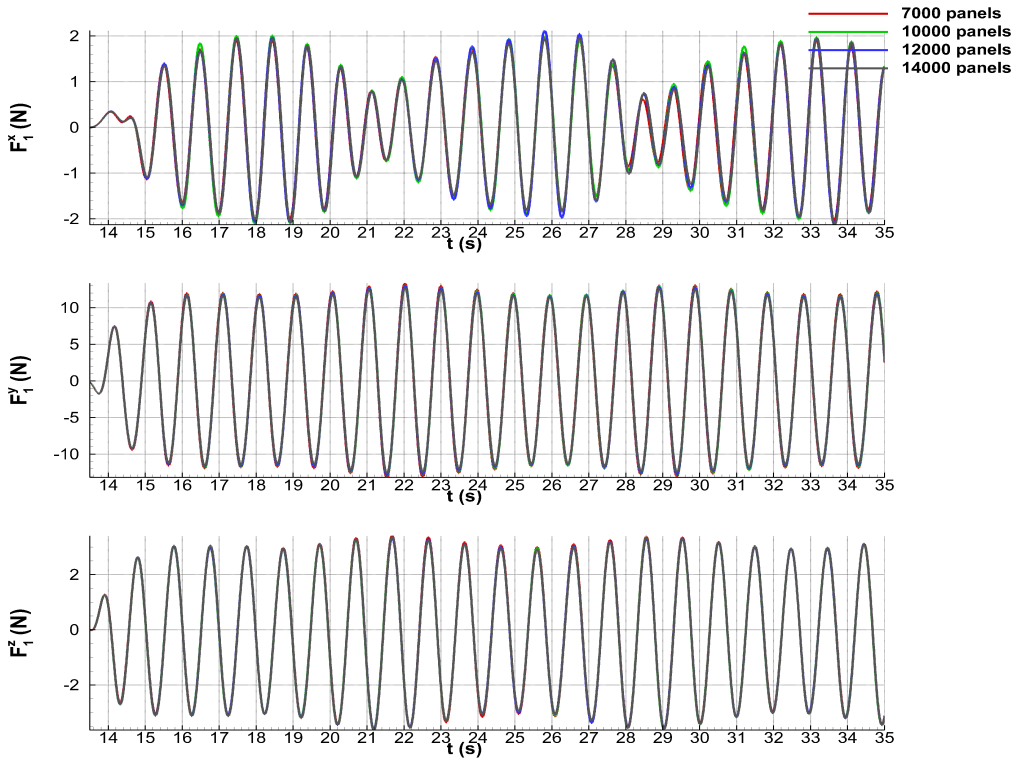


Figure III.37 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical results for Case 1 with a body-exact approximation and several meshes

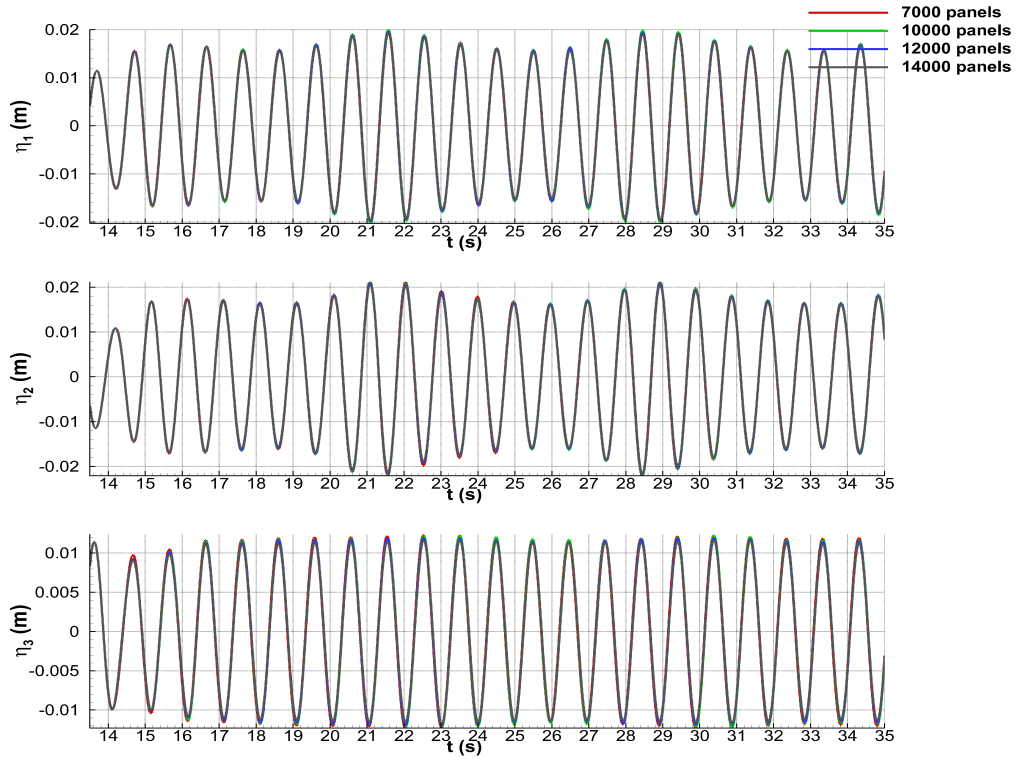


Figure III.38 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical results for Case 1 with a body-exact approximation and several meshes

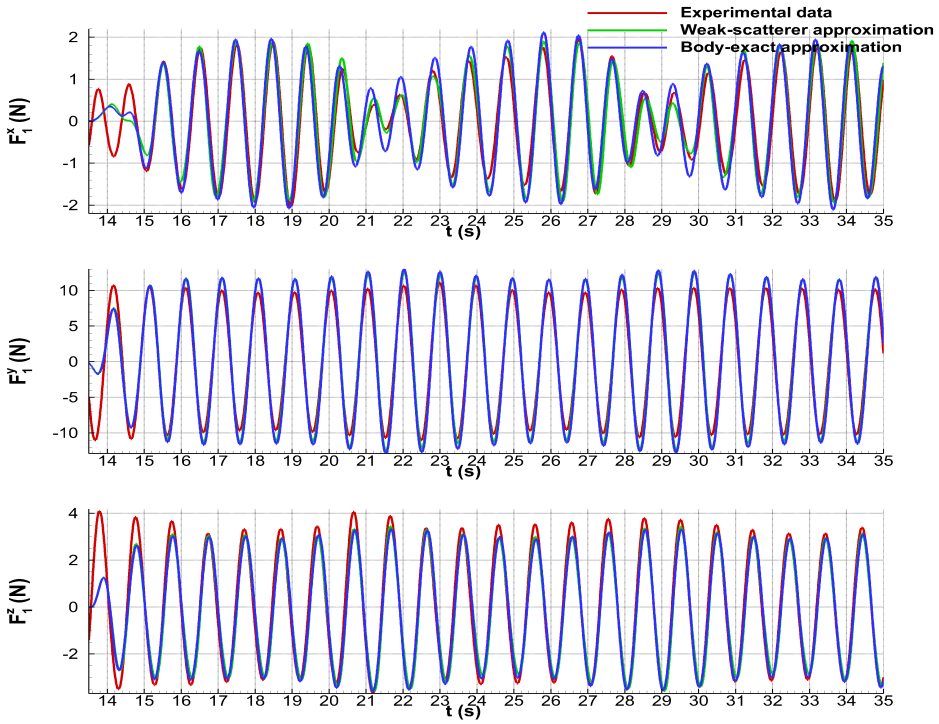


Figure III.39 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 1 with two hydrodynamic approximations

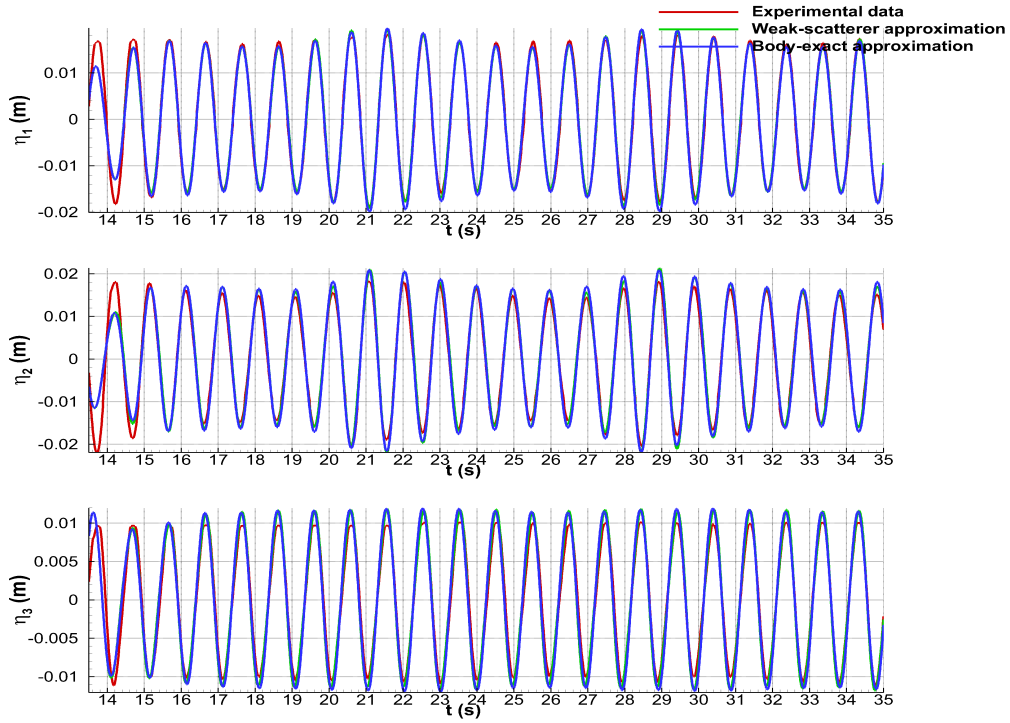


Figure III.40 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical and experimental results for Case 1 with two hydrodynamic approximations

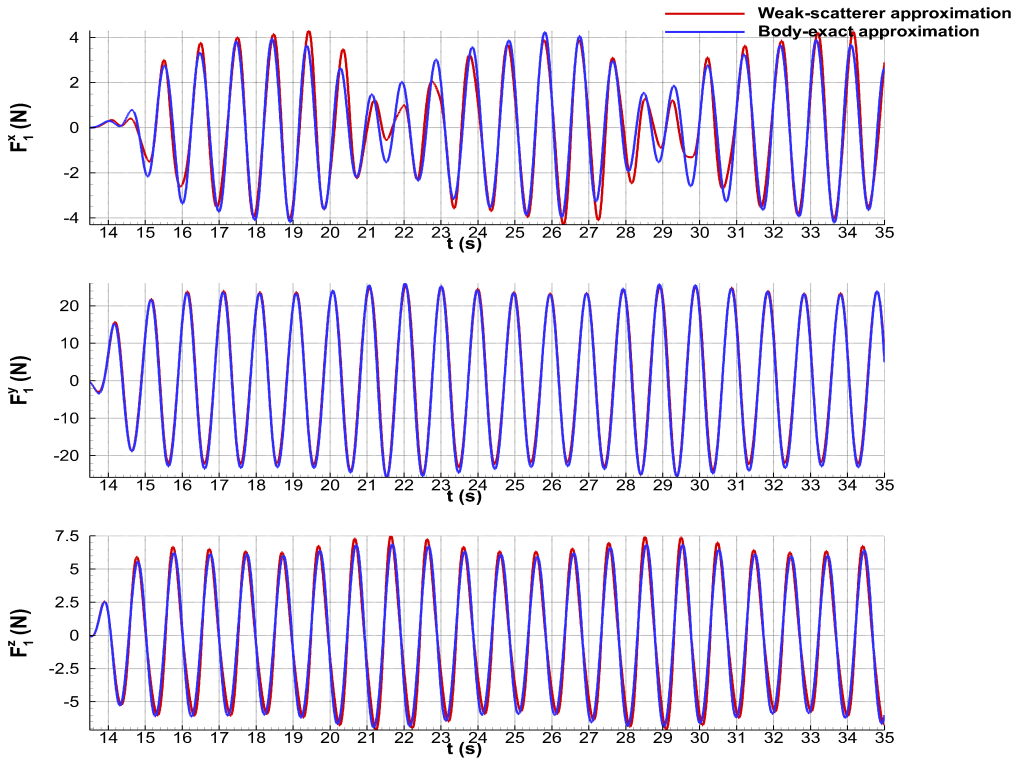


Figure III.41 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical results for $\epsilon = 0.096$

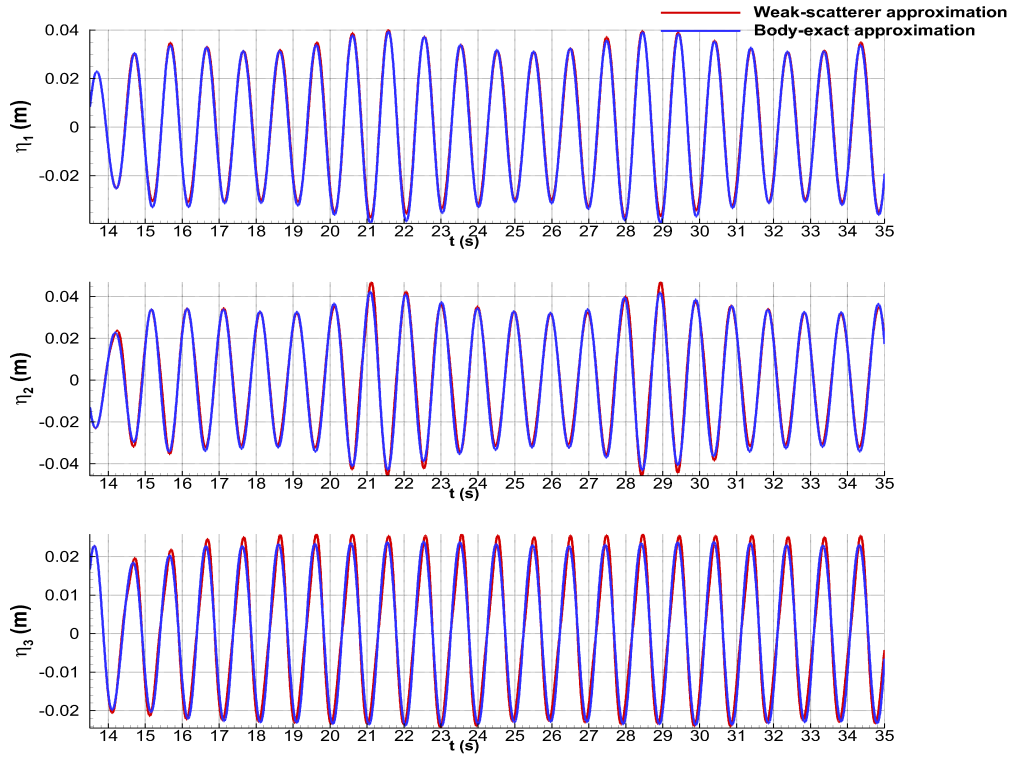


Figure III.42 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical results for $\epsilon = 0.096$

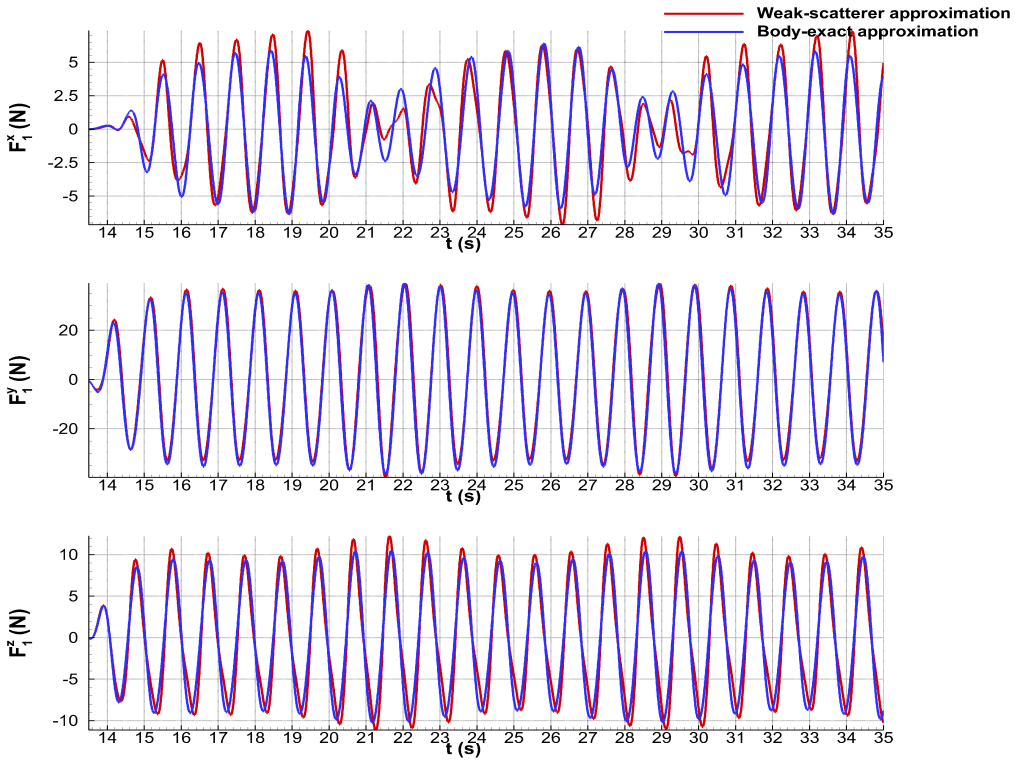


Figure III.43 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical results for $\epsilon = 0.144$

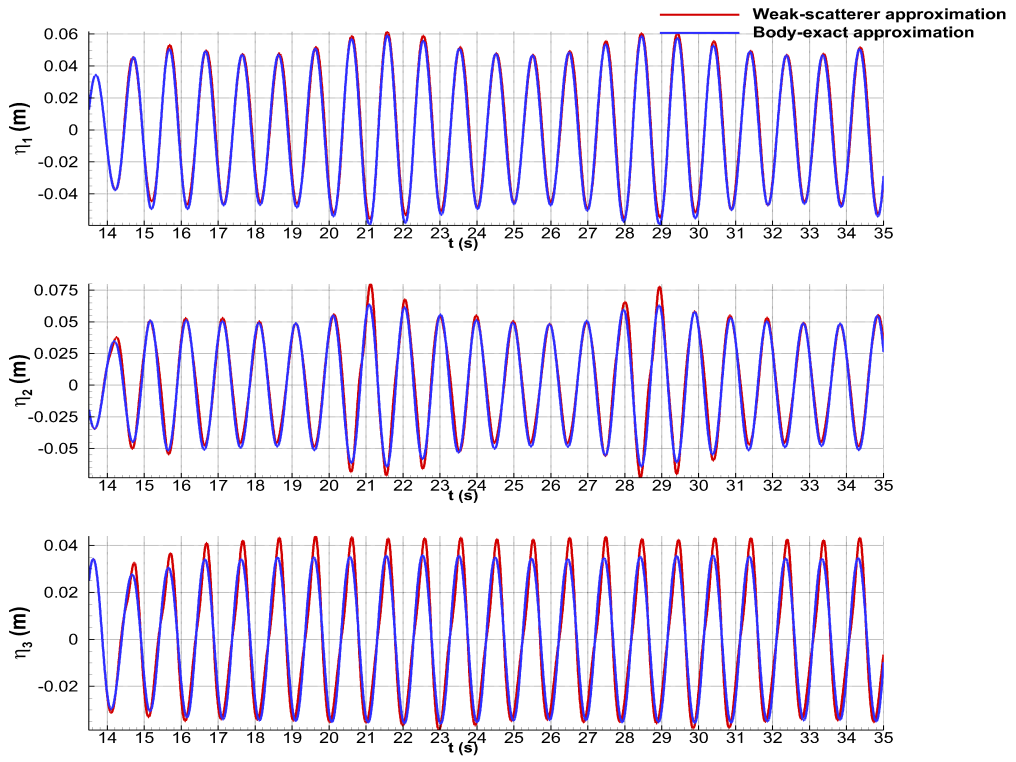


Figure III.44 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical results for $\epsilon = 0.144$

III.11.1.4 Influence of the body nonlinearities

WS_ CN may also be switched to a fully linear approximation. In this case, the free surface is linearized at its mean position ($z = 0$) and the body conditions are linearized at the equilibrium position of the bodies (initial position here). The mesh convergence is shown in Figures III.45 and III.46. The mesh with 9000 panels is chosen. The time step is 0.005 s. Less panels are required compared to the weak-scatterer and body-exact approaches. Figures III.47 and III.48 present the comparison between the experimental data, the weak-scatterer, the body-exact and the fully linear approximations. The amplitude and the frequency of F_1^y and F_1^z are good but the modulation of the amplitude is not captured. The hydrodynamic load along the x-axis, F_1^x, does not match the experimental data and the other numerical methods at all. Regarding the wave elevations, the modulation is not captured either. Larger differences are observed for η_1 and η_2 because the hydrodynamic interactions are more important than for η_3. Indeed, the wave probe 3 is located in a sheltered area provided by *Body 1* [145], which explains the better agreement of the linear model with the weak-scatterer and body-exact models.

Finally, the linear model is not accurate enough to simulate the hydrodynamic interactions in the present case. The free surface nonlinearities are small, which enables the use of the body-exact approximation but the body nonlinearities are too important to use a linear approximation. This confirms that an unsteady potential flow based solver is required in case of large relative motions.

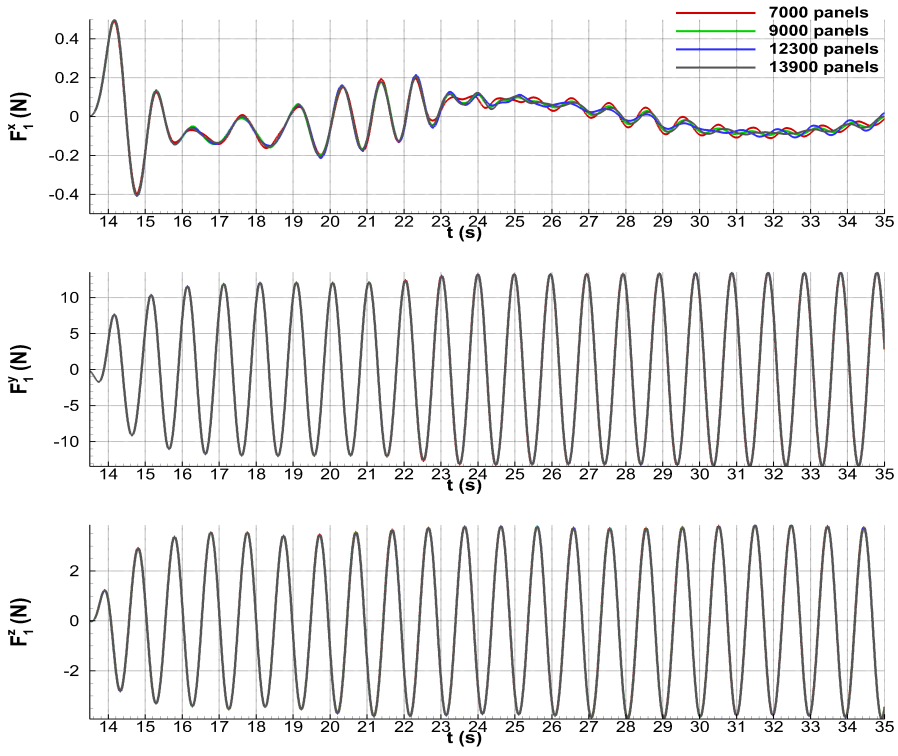


Figure III.45 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical results for Case 1 with a linear approximation and several meshes

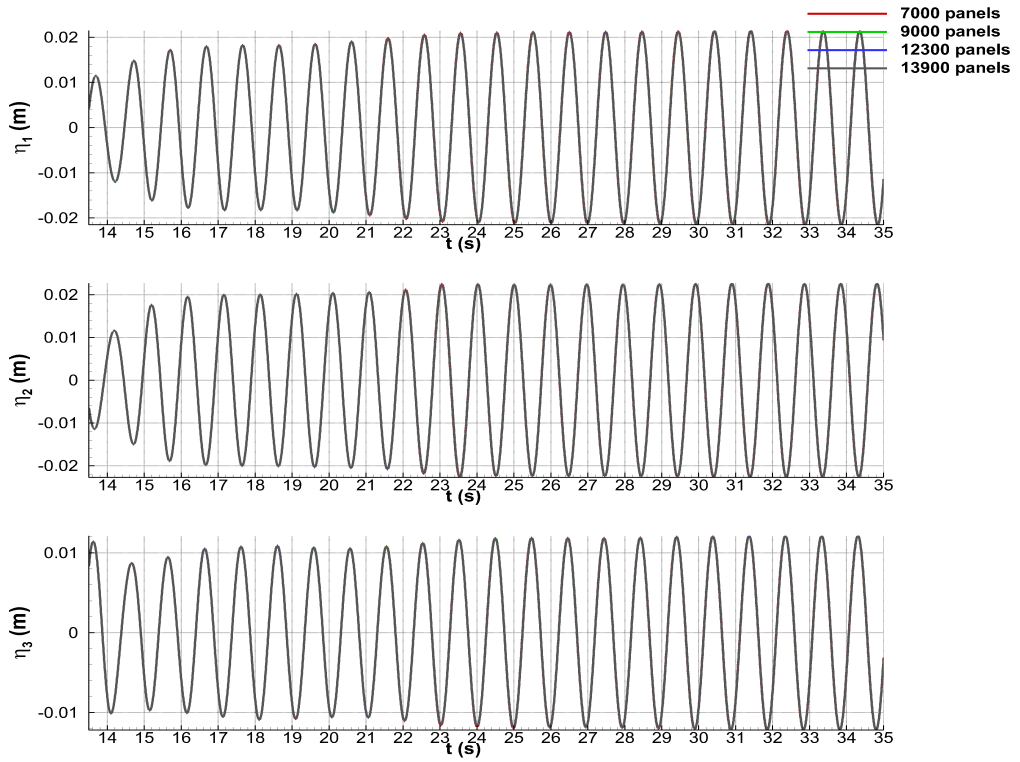


Figure III.46 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical results for Case 1 with a linear approximation and several meshes

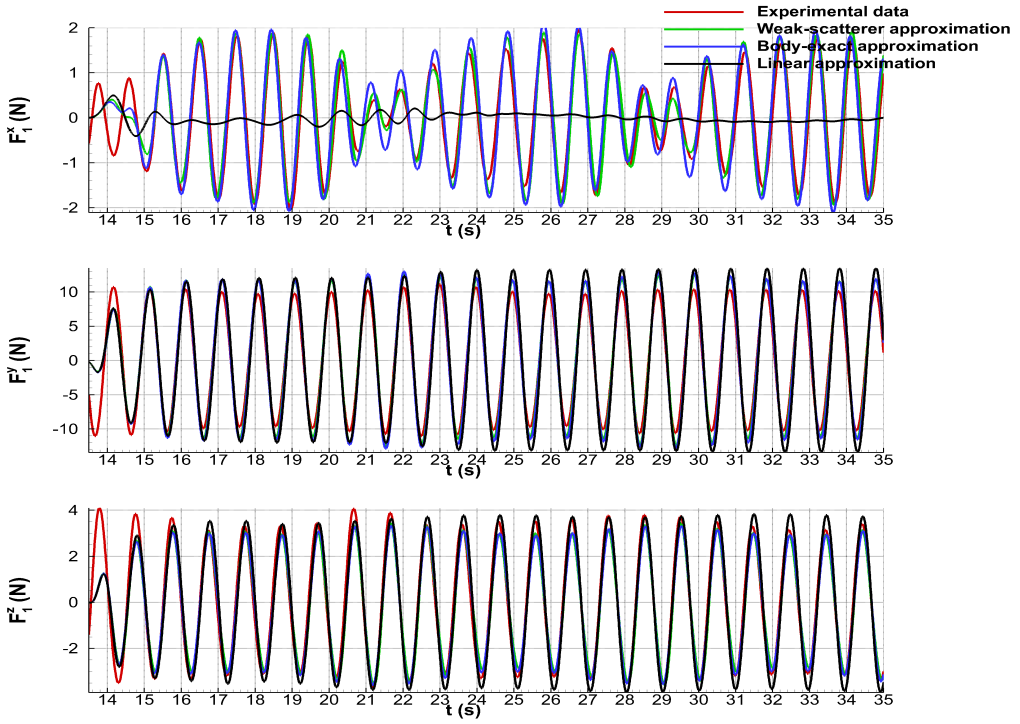


Figure III.47 – Comparison of time series of the hydrodynamic loads on *Body 1* from numerical and experimental results for Case 1 with three hydrodynamic approximations

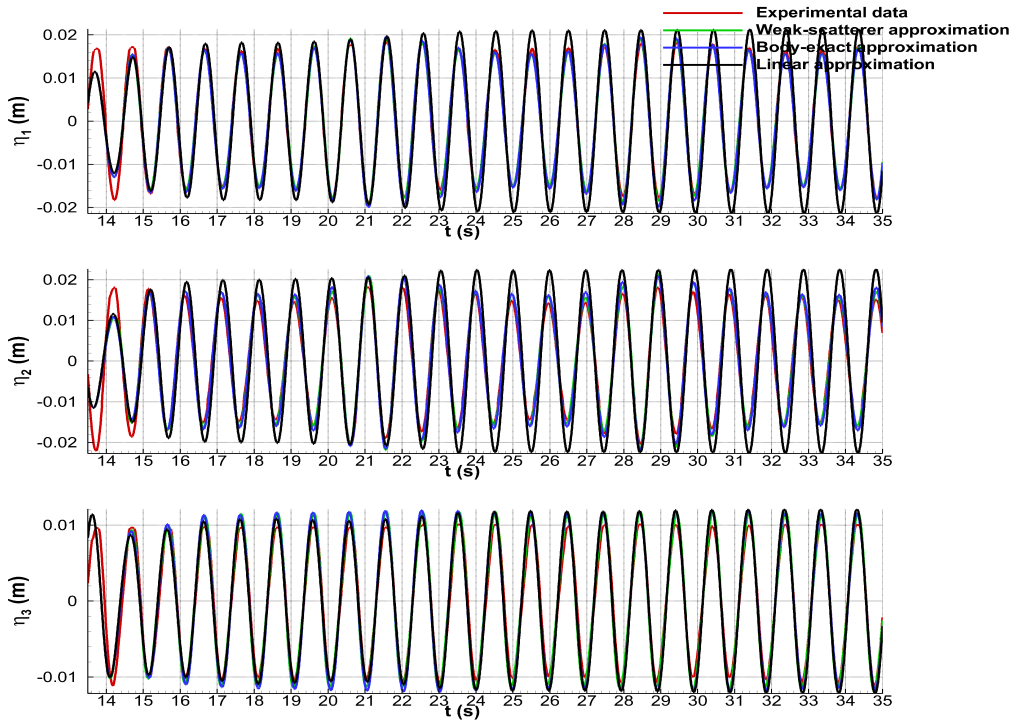


Figure III.48 – Comparison of time series of the wave elevations at *WP1*, *WP2* and *WP3* from numerical and experimental results for Case 1 with three hydrodynamic approximations

III.11.2 Free motion

In this section, the extension of *WS_ CN* to multibody simulations is verified for several bodies in free motion. A comparison is done between *WS_ CN* and the diffraction-radiation code *Nemoh* [47] coupled with *InWave*.

Two identical vertical cylinders (named *Cylinder 1* and *Cylinder 2*) of radius 0.2 m, length 0.2 m and draft 0.1 m are considered. The initial positions of *Cylinder 1* and *Cylinder 2* are $(0, -0.4, 0)$ and $(0, 0.4, 0)$.

Their mass is 12.88 kg. An incoming regular wave of amplitude 0.001 m and wave frequency 8 rad/s is used. A wave probe is present at the position $(0.4, 0, 0)$. The cylinders only move in heave. The multibody system used in *InWave-Nemoh* is made of two bodies linked by a prismatic joint. *Cylinder 1* is the base while *Cylinder 2* is the body 1. The set of modified Denavit-Hartenberg coefficients is given in Table III.7.

j	σ_j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	1	$\frac{\pi}{2}$	0	0	0.8	$-\frac{\pi}{2}$	0

Table III.7 – *mDH* parameters for the free motion test case in *InWave-Nemoh*

The mesh convergence using *WS_ CN* is shown in Figure III.49 while the time step convergence for a mesh of 15 000 panels is presented in Figure III.50. From these results, a mesh of 15 000 panels with a time step of 0.005 s are selected in *WS_ CN*.

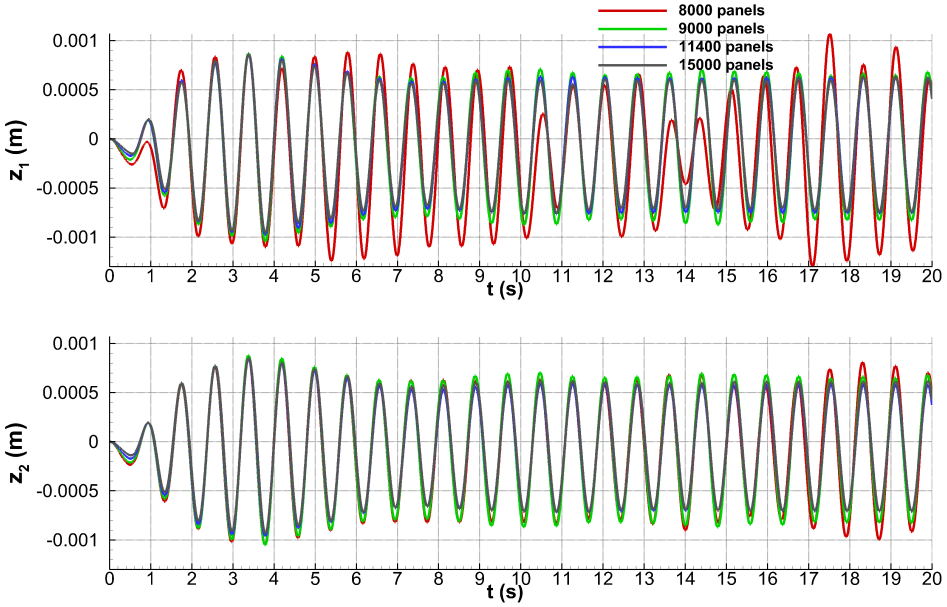


Figure III.49 – Mesh convergence using *WS_CN* for the heave motion of *Cylinder 1* (top) and *Cylinder 2* (bottom)

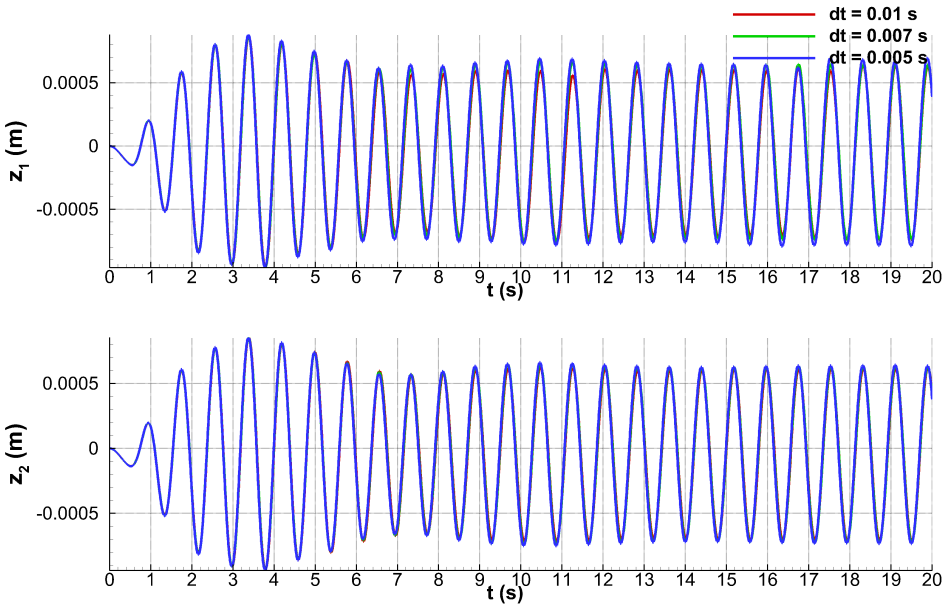
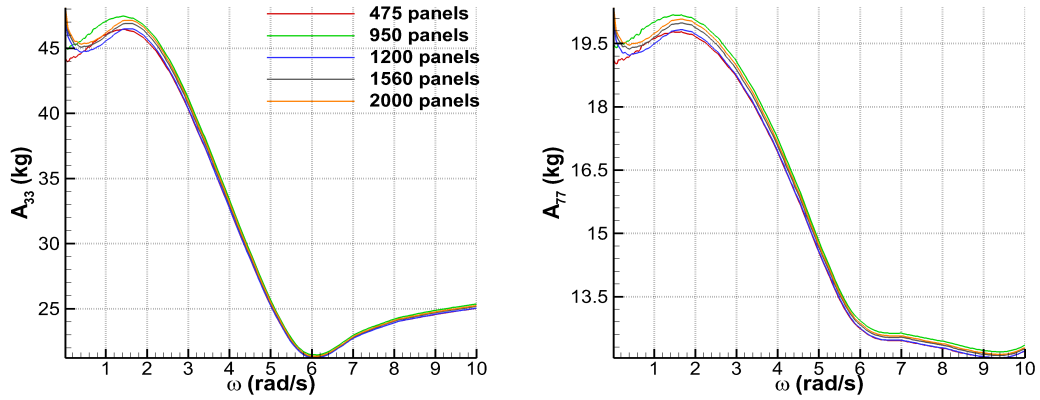


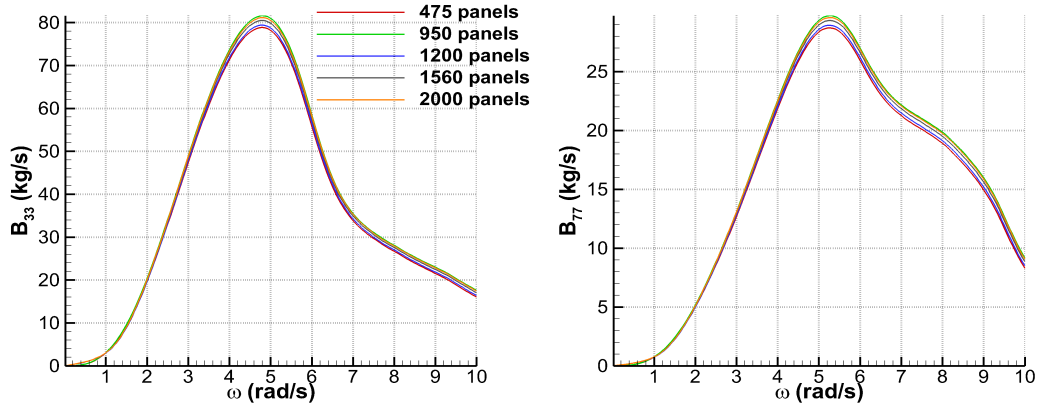
Figure III.50 – Time step convergence using *WS_CN* for the heave motion of *Cylinder 1* (top) and *Cylinder 2* (bottom)

The mesh convergence of the hydrodynamic coefficients (added mass, hydrodynamic damping, exciting loads) using *Nemoh* is presented in Figure III.51. A_{33} and A_{77} represent the added mass coefficients in heave of the whole multibody system and Cylinder 2 due to a unit heave acceleration of the whole multibody system and Cylinder 2, respectively. B_{33} and B_{77} are the damping coefficients in heave of the whole multibody system and Cylinder 2 due to a unit heave velocity of the whole multibody system and Cylinder 2, respectively. F_3 and F_7 denote the exciting forces in heave of the whole multibody system and Cylinder 2, respectively. Based on the results of Figure III.51,

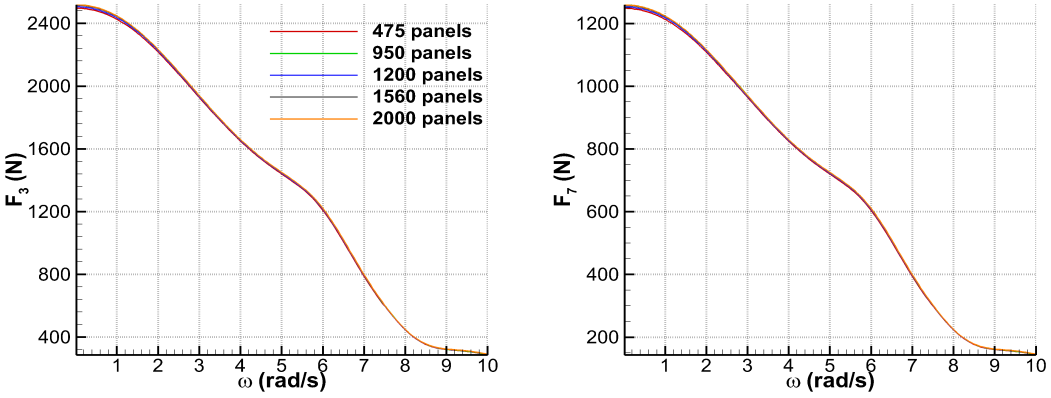
the mesh with 2000 panels per cylinder is chosen.



(a) Added mass coefficients



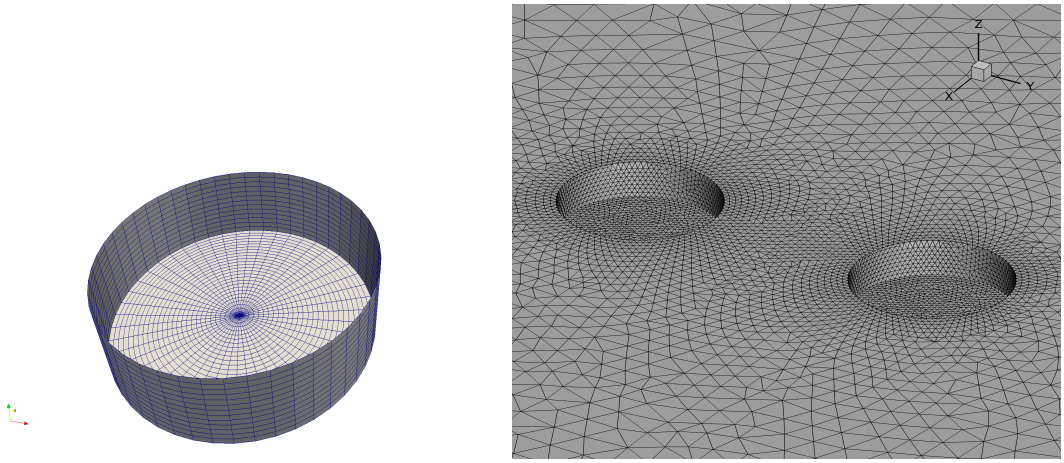
(b) Hydrodynamic damping coefficients



(c) Exciting forces

Figure III.51 – Mesh convergence of the hydrodynamic coefficients using *Nemoh*. The number of panels is given for one cylinder.

The final meshes for every hydrodynamic solver are displayed in Figure III.52.



(a) *Nemoh* **(b)** *WS_ CN*

Figure III.52 – Final meshes

In the linear model, once the hydrodynamic database is computed from *Nemoh*, time-domain simulations are performed based on Cummins' equation [102] (cf. Appendix B) using *InWave*. The time step convergence is shown in Figure III.53. A time step of 0.01 s is enough. It can be seen that the linear potential flow approach requires a smaller time step than the weak-scatterer approach. Regarding the number of panels, the mesh of 15 000 panels used in *WS_ CN* represents a mesh of 3300 panels per cylinder. A smaller number of panels is required by the linear approach to reach the convergence of the numerical results.

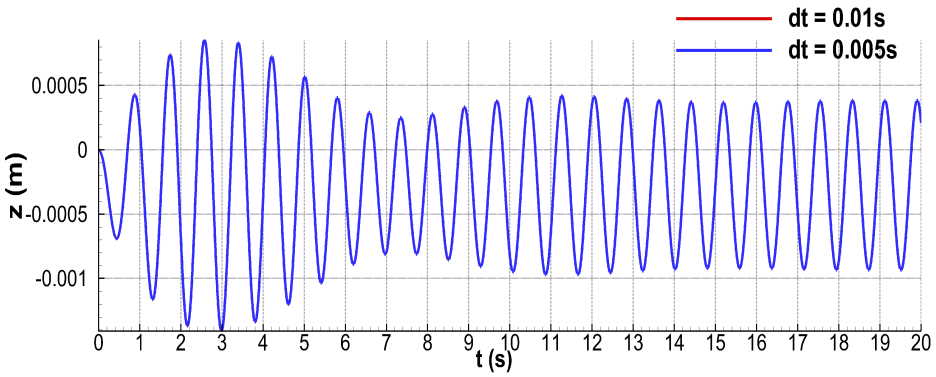


Figure III.53 – Time step convergence using *InWave* for the heave motion of *Cylinder 1*

The comparison between *InWave-Nemoh* and *WS_ CN* is presented in Figure III.54. The agreement is good between the two different hydrodynamic models. The differences observed at the beginning of the simulation are due to the effects of the ramp for smoothing the appearance of the waves.

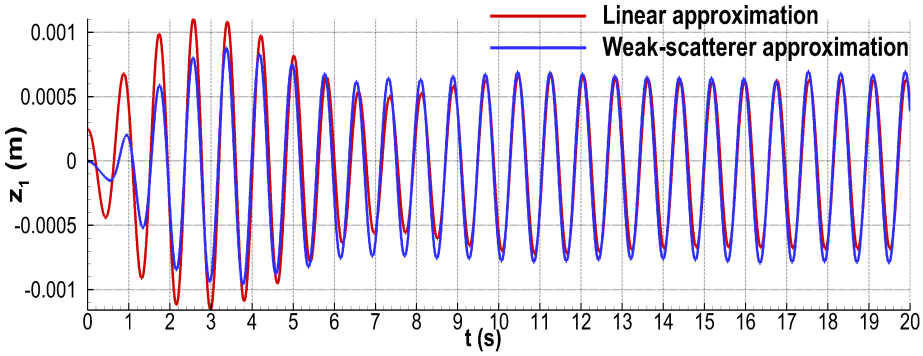


Figure III.54 – Comparison of time series of the heave motion of *Cylinder 1* from numerical results of *InWave-Nemoh* and *WS_CN*

The comparison between the motion of the two cylinders is shown in Figure III.55 for the weakly nonlinear solver and in Figure III.56 for the linear solver. q_1 represents the relative motion between the two bodies of the multibody system. For every method, the two cylinders have the same motion, which is expected because the problem is symmetrical.

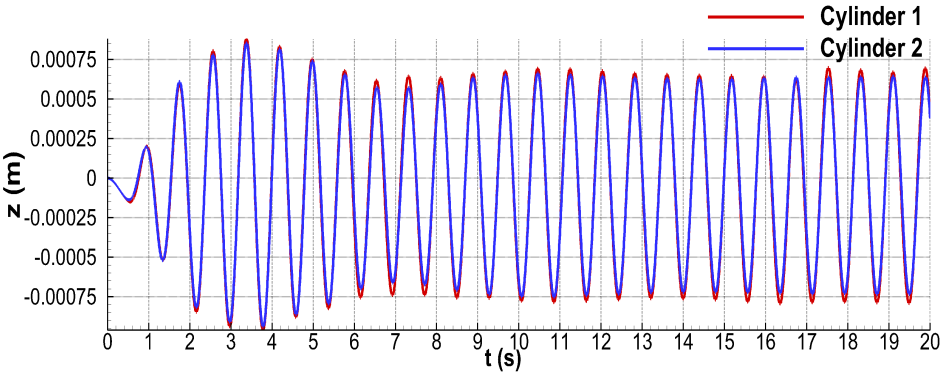


Figure III.55 – Comparison of time series of the heave motion of *Cylinder 1* and *Cylinder 2* from numerical results of *WS_CN*

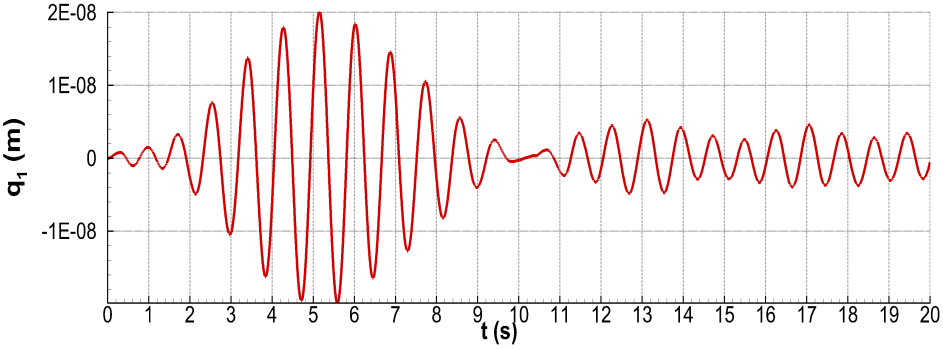


Figure III.56 – Comparison of time series of the relative heave motion between *Cylinder 1* and *Cylinder 2* from numerical results of *Nemoh*

III.12 Conclusion

This chapter presented the hydrodynamic theory used in this PhD work in *WS_CN*: the potential flow approach based on the weak-scatterer hypothesis. A review of the different approximations of the potential flow theory was provided along with a state of the art of the use of the weak-scatterer approximation. In order to simulate marine operations involving several bodies subject to hydrodynamic loads, *WS_CN* has been extended to multibody simulations. A validation in forced motion was presented using experimental data. The comparison between the experimental data and the weak-scatterer model gave good results for both the hydrodynamic loads and the wave elevations. Hydrodynamic interactions between the floating bodies were well captured. An analysis of the weak-scatterer hypothesis was done through the comparison with a body-exact approximation and a fully linear approximation. As the wave steepness was small, the body-exact model also gave good results compared to the experiments but with fewer panels in the mesh. Nevertheless, when the wave steepness and the free surface nonlinearities were increased, the weak-scatterer and body-exact approaches started being different. The linear approximation gave insufficiently accurate results compared to the experimental data.

These results have proved the interest of an unsteady potential flow theory in case of large relative motions. Furthermore, the weak-scatterer approximation is fully relevant when the free surface nonlinearities are important, otherwise a body-exact could be enough. If a body-exact approximation is used, in order to reduce the size of the mesh and therefore the CPU time, a free surface Green's function (or Kelvin source Green's function) could be preferred to the Rankine source Green's function used in *WS_CN*. The free surface would not be meshed.

The extension to multibody simulations in free motion was validated by the comparison with the frequency-domain potential flow-based solver *Nemoh* coupled with *InWave*.

The stability of *WS_CN* has been improved by the implementation of the free surface remeshing process. The CPU time has been decreased by the parallelization of the computation of the influence coefficients using *OpenMP*. The memory footprint has been reduced by the use of the sparse linear system solver *PARDISO*.

The coupling of the potential flow theory based on the weak-scatterer hypothesis of *WS_CN* with the Composite-Rigid-Body Algorithm of *InWave* is the goal of the next chapter.

Chapter IV

Fluid-structure interaction

Contents

The previous chapters presented the main technical requirements to simulate marine operations: multibody dynamics, cable dynamics and hydrodynamics. The goal of this chapter is to introduce the coupling between these models. First, a state of the art of the multiphysics couplings is presented. Then, a coupling strategy, between the potential flow theory based on the weak-scatterer hypothesis of WS_ CN and the Composite-Rigid-Body Algorithm of InWave, is proposed and detailed. A validation of these developments along with a comparison with other coupling strategies are presented.

IV.1 State of the art

IV.1.1 Multiphysics problems

Multiphysics problems are met in many fields [146, 147]:

- Civil engineering: wind-structure interaction, soil dynamics, etc.;
- Ocean engineering: fluid-structure interaction, cavitation, acoustics, etc.;
- Aerospace, nuclear or automotive engineering: fluid-structure interaction, thermal-structure interaction, etc.;
- Biological engineering: fluid-structure interaction, etc.;
- Other.

These multiphysics problems can be represented by partial differential equations in space and time coupled at their physical boundaries. Software modules are generally available to solve one domain of the physics (named a field). For coupled problems, it is necessary to couple several software packages. Park et al. [148] published one of the first papers on fluid-structure interaction (finite-element methods for the structure, boundary integral technique for the fluid). Since the early works, three schemes to simulate multiphysics problems have arisen [146, 149]:

- Field elimination;
- Monolithic treatment;
- Partitioned treatment.

IV.1.1.1 Field elimination

The field elimination method eliminates one or more fields in the coupled equations. The other fields are time-stepped simultaneously. For instance, in a fluid-structure problem, this method implies to introduce the fluid equations into the motion equation, so that this latter is the only equation to solve. The advantage of this method is that it reduces the number of variables to time-step. But it leads to the increase in the complexity of the multiphysics problem to solve (high-order time-derivatives can appear) and the loss of mathematical properties [146]. This approach is rarely used.

IV.1.1.2 Monolithic treatment

With a monolithic treatment, the governing differential equations are time-stepped simultaneously. This system of differential equations is called the coupling equation. The advantages of a monolithic treatment are the robustness and the accuracy as the governing equations of the system are solved synchronously. But this approach involves a complex assembling and solving of the set of equations. Furthermore, the temporal discretization needs to be the same (although several time integrators may be used [150]). Fixed time-step through the field solvers are inconvenient if one of them requires a much smaller time step compared to the other ones. An example of monolithic treatment is given by Blom [151] with a monolithic fluid-structure coupling using Euler's equation for the fluid and the motion equation for a rigid body. A scheme of a monolithic coupling is shown in Figure IV.1.

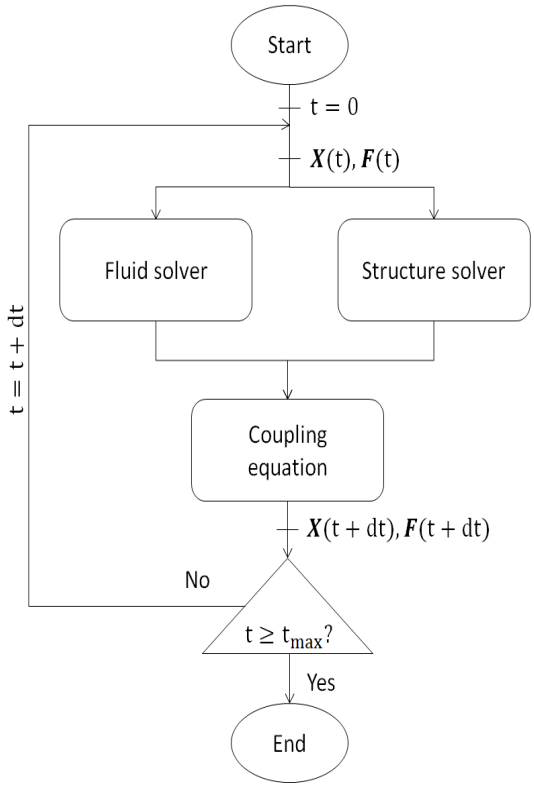


Figure IV.1 – Monolithic approach in case of a fluid-structure coupling. **X** and **F** represent the motion of the structure and the hydrodynamic loads.

IV.1.1.3 Partitioned treatment

With a partitioned treatment, each field of the physics is treated in isolation and separately stepped in time. Several strategies exist to do so. Since 1980s, a general theory of partitioned procedures for second-order systems has been developed by Park et al. [152, 153]. In case of sequential execution of the field solvers, partitioned approaches are named staggered solutions. This case was studied in detail by Felippa and Park [146]. Different algorithms were developed for the staggered approach [154, 155, 156]. The positive aspects of a partitioned treatment are the inherent modularity along with the possibility to fit the time step, the time integrator and more generally the physical model for every field solver. Thereby, each field solver is independent from the others.

For complex coupled problems, this approach is more software-wise and less challenging than the monolithic treatment. The drawbacks are the numerical errors and the stability problems.

As the solvers are called asynchronously, it always exists a time-lag between the time integrations of the fields. For instance, the exact position of a fluid-structure interface is not necessarily the same in every solver because the spatial discretizations are different. Therefore, the slip condition for an inviscid fluid or the no-slip condition for a viscous fluid is not exactly enforced. The result of this is an artificial increase or decrease of the energy in the system which leads to a loss of accuracy. The conservation of the energy at the fluid-structure interface allows to improve the robustness and accuracy of such partitioned approaches [157]. The energy conservation has also to be ensured with a monolithic treatment when several spatial discretizations (meshes) are used [158]. The use of a predictor can reduce the effects of the time-lag [157].

Amongst the staggered methods or loose couplings, there are two classes: the purely sequential methods or explicit couplings and the iterative methods or implicit couplings [156, 159]. Each method can also be divided into two versions: serial or parallel, leading to four classical staggered coupling algorithms [160] (Figure IV.2):

- The explicit couplings:
  - The conventional serial staggered method (*CSS*) (Figure IV.3);
  - The conventional parallel staggered method (*CPS*).
- The implicit couplings:
  - The block Gauss-Seidel method (Figure IV.4);
  - The block Jacobi method.

The only advantage of the parallel version of a coupling method is the possibility of parallelization. The numerical properties are better for the serial version [121]. By using a retroactive loop, an implicit coupling gives the same results as a monolithic treatment, other things being equal [156]. But, the solvers are called more than once per time step. Consequently implicit couplings lead to an increase of the CPU time compared to explicit couplings.

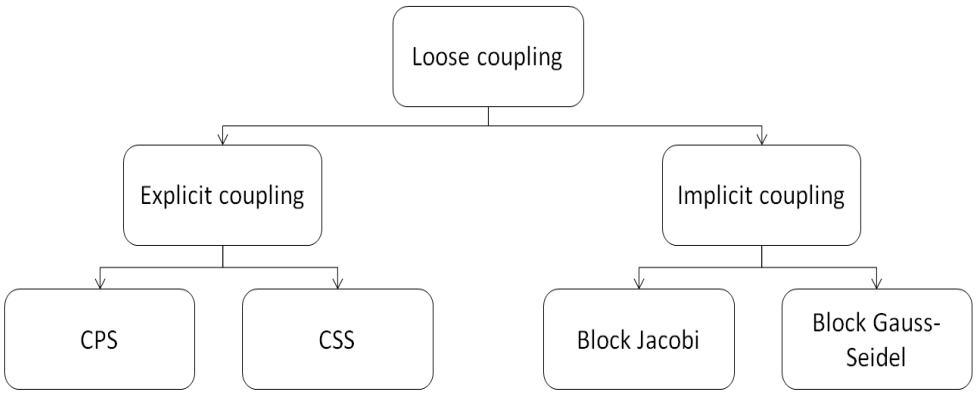
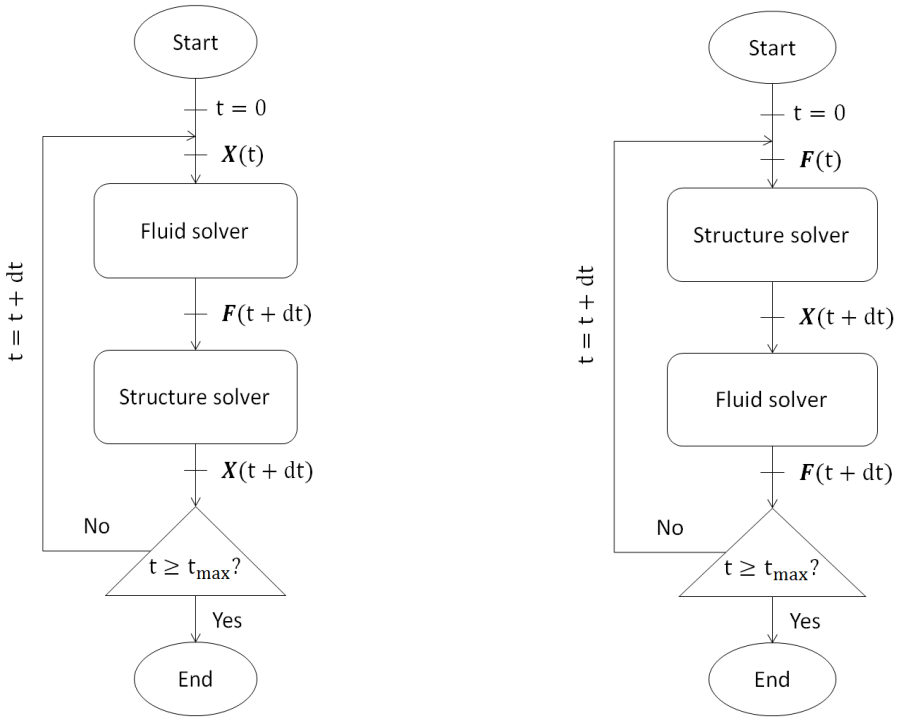


Figure IV.2 – Main loose coupling algorithms



(a) Fluid formulation **(b)** Mechanical formulation

Figure IV.3 – CSS algorithms. **X** and **F** represent the motion of the structure and the hydrodynamic loads.

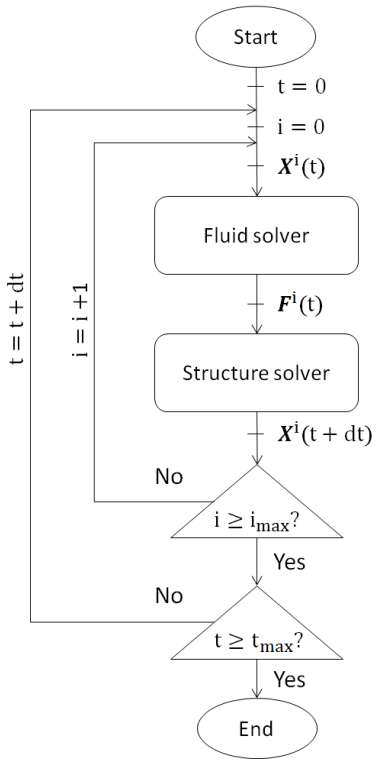


Figure IV.4 – Block Gauss-Seidel algorithm in case of a fluid-structure coupling. **X** and **F** represent the motion of the structure and the hydrodynamic loads.

Other algorithms exist. For instance, subcyles can be added if a field needs much smaller time step, couplings can be asynchronous, the exit condition in a iterative loop may be enhanced by using a Newton or Newton-like method in order to accelerate the implicit coupling [160], etc.

In 1995, Belanger et al. [161] encountered numerical instabilities in their staggered fluid-structure coupling when the fluid added-mass became larger than the structural mass. They proposed a method to avoid these numerical discrepancies by adding a virtual added-mass in the motion equation to stabilize their simulation [161]. An explanation was given using a theoretical model [162]. It showed that the ratio of the fluid density over the structural density is critical for the stability of the staggered coupling. Moreover, when decreasing the time step, instabilities become more important. Förster et al. [163] showed that this problem does not depend on the time integration scheme. The added-mass effect was also studied for compressible flows by van Brummelen [164]. Yvin [160] used a relaxation method to reduce the numerical instabilities without modifications of the fluid or structure solvers. As this added-mass effect depends on the fluid density, it does not appear when the fluid is the air, because its density is too small. This numerical problem is mostly encountered in partitioned water-structure coupling.

IV.1.2 Computer programming

In the previous section, coupling schemes were presented from a physical and numerical point of view. The software architecture of multiphysics numerical tools is an important consideration too. Two options are possible to solve a coupled problem: either a new numerical tool is created and dedicated to the multiphysics problem or different existing numerical tools are coupled.

Partitioned approaches are modular by their nature. The use of legacy codes is possible. The independence of the implementation of the different solvers is guaranteed. For the monolithic treatment of a coupled problem, it is not straightforward. This is why it is necessary to make a distinction between a *monolithic formulation*, where a new numerical tool is created *ex nihilo*, including all the solvers, and a *tight coupling* in which different independent subsystems solving a unique field of the physics are linked through a coupling equation and time-stepped simultaneously. This latter coupling is considered as a partitioned approach [159]. It gathers the numerical advantages of a monolithic treatment (robustness, accuracy) and the modularity and the software-wise approach of a partitioned treatment. Loose couplings are distinct of tight couplings by the separate time-stepping of the solvers. For example, Li [150] used a tight coupling approach with a Smooth-Particle-Hydrodynamics model and a finite-element method.

Finally, the different coupling strategies are summarized in Figure IV.5.

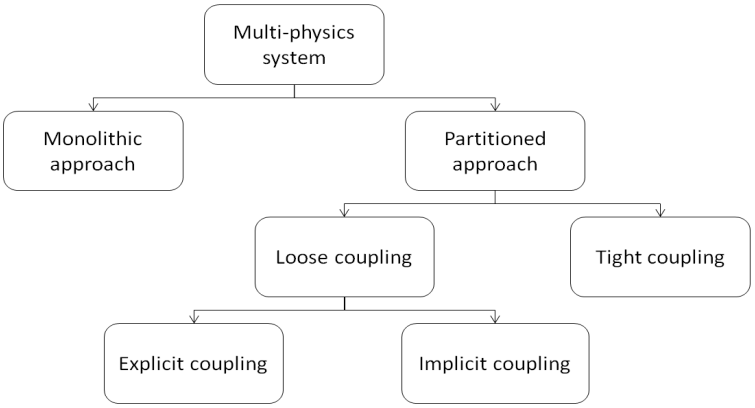


Figure IV.5 – Main coupling strategies

IV.1.3 Language binding

In partitioned approaches, the implementation of the solvers is independent from the others. Consequently, different programming languages may be used, for instance to take advantage of the speed of low-level languages and the ease of implementation of high-level languages. Doing so input-output communication between solver may become challenging. Different methods exist to overcome this difficulty.

The simplest approach relies on temporary text files. Each matrix, vector or scalar required by a solver is written in a temporary file. In case of an important bulk of data, the time of writing and reading makes this method prohibitive.

The use of a *glue code* is another possibility [165]. All modules are interfaced with one high-level program which manages the input-output relationships for the whole code via *DLLs* (Dynamic Link Library) or the direct coupling of the source codes. This latter option needs the use of a specific programming language dedicated to the interfacing with other existing applications. Python[1] is an example of such a language. Different tools exist to compile low-level languages with Python. For example, *Cython*[2] allows linking scripts written in C or C++ with Python [166], *f2py*[3] does the same between Fortran and Python [167], etc. The availability of the source codes restrains clearly this method (for instance when commercial tools are used).

Finally the use of message passing between solvers is another possibility. For instance, MPI communication (Message Passing Interface) enabled the coupling between the fluid solver *SPH-flow*[4] and the structural solver *Code_Aster*[5] [121]. Yvin [160] preferred employing network sockets to link the multibody dynamics software package *MBDyn*[6] and the hydrodynamic solver *ISIS-CFD*[7].

[1] https://www.python.org
[2] http://cython.org
[3] https://docs.scipy.org/doc/numpy/f2py
[4] http://www.sph-flow.com
[5] https://www.code-aster.org
[6] https://www.mbdyn.org
[7] https://lheea.ec-nantes.fr/software-and-patents/isis-cfd-193387.kjsp

IV.2 Coupling between the *CRBA* and the weakly nonlinear potential flow theory

IV.2.1 The original coupling in *WS_CN*

The terms to define the different methods to solve the fluid-structure interaction, presented in subsubsection III.2.6.1, do not match those presented in section IV.1. Each field (fields of the potential flow theory and multiphysics coupling theory) has defined its own terms. Nevertheless, an equivalence is possible and is presented in Table IV.1. The number of boundary value problems to solve for computing the time-differentiation of the velocity potential is given for each method. The boundary value problem for solving the velocity potential is not included.

Potential flow theory	Multiphysics coupling theory	N_{BIE}
Backward finite difference	Loose explicit	0
Iterative	Loose implicit	1 + i
Mode decomposition	Monolithic	7
Indirect	Monolithic	6
Implicit boundary	Monolithic	1

Table IV.1 – Equivalence of the terms to define the fluid-structure couplings used in the potential flow theory and in the multiphysics coupling theory. N_{BIE} denotes the number of boundary integral equations to solve at each time step to compute $\frac{\partial \phi}{\partial t}$. i represents the number of iterations to reach the convergence at each time step.

The mode decomposition and indirect methods are two monolithic couplings because the differential equations are time-stepped simultaneously. But they are specific to the potential flow modelling because of the mode decomposition and the use of the Green's second identity. The implicit boundary method is strictly a monolithic coupling, as the differential equations of the potential flow theory and the motion equations are solved at the same time.

IV.2.2 The choice of a coupling strategy

As presented in the previous sections, several coupling strategies are possible. *InWave* and *WS_CN* already existed before this PhD work, therefore a monolithic coupling, which leads to a new implementation of both solvers was not a meaningful option. An implicit loose coupling would lead to a higher CPU time cost than a tight coupling. An explicit loose coupling would only give a small reduction of the computation time, as the second Boundary Value Problem always needs to be figured out, for the price of a decrease of the accuracy and the stability of the simulation.

An advantage of a loose coupling approach is the possibility to fit the time step for each solver. The mechanical solver needs to simulate cables and leads to a stiff system of differential equations, so small time steps are expected. The suitable order of magnitude of the fluid solver time step is not predictable with accuracy. But the subcycling of the mechanical solver in a tight coupling could fix this problem.

This is why, in order to maintain the robustness, the stability and the accuracy, without increasing significantly the CPU time while creating a modular architecture, a

tight coupling between the Composite-Rigid-Body Algorithm of *InWave* and the weakly nonlinear potential flow theory of *WS_CN* is preferred.

IV.2.3 Theoretical developments

The tight coupling between the Composite-Rigid-Body Algorithm and the weakly nonlinear potential flow theory involves solving both the mechanical and fluid governing equations in the same time. It requires the introduction of the hydrodynamic loads in the multibody motion equation (I.62) and the expression of the slip condition (III.105) with respect to the articular accelerations.

IV.2.3.1 Multibody motion equation

Two methods are possible for taking into account the hydrodynamic loads in the multibody motion equation. The first method involves to include the hydrodynamic loads in the external load vector ${}^{j}\mathbf{F}_{ej}$ (*Method 1*). For example, the weight is added in the multibody motion equation following this method (cf. section I.4). The second method requires the projection of the hydrodynamic loads in the articular space using the articular variables (*Method 2*). For instance, the vector of the internal loads along the joint axes $\mathbf{\Gamma}$ (I.62) follows this approach.

In the case of the weight, *Method 1* is straight forward. But, *Method 2* could also be applied. Regarding the hydrodynamic loads, *Method 1* is not as feasible. As seen in subsection III.2.6, the hydrodynamic loads bring a new unknown: the time-differentiation of the velocity potential. This unknown must be accessible and isolated to form the linear system of the fluid-structure coupling. *Method 1* does not enable that. That is why, *Method 2* is preferred.

The application of *Method 2* to include the hydrodynamics loads in (I.62) gives:

$$\mathbf{H}\begin{pmatrix} {}^{0}\dot{\mathbf{V}}_{0} \\ \ddot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{6\times 1} \\ \mathbf{\Gamma} \end{pmatrix} - \mathbf{C} + \begin{pmatrix} {}^{0}\mathbf{F}^{WSC} \\ \mathbf{\Gamma}^{WSC} \end{pmatrix} \tag{IV.1}$$

${}^{0}\mathbf{F}^{WSC}$ is the sum of all hydrodynamic loads acting on the whole multibody system at $\mathbf{O}_0$ and projected in the base frame:

$${}^{0}\mathbf{F}^{WSC} = \sum_{j\in\mathcal{F}^{WSC}} {}^{0}\mathbf{N}_{e}^{j}\, {}^{e}\mathbf{G}_{j}\, {}^{e}\mathbf{F}_{j}^{WSC} \tag{IV.2}$$

with:

- $\mathcal{F}^{WSC}$ the set of the bodies subject to the hydrodynamic loads. Each hydrodynamic body has a number in the multibody system and a number in $\mathcal{F}^{WSC}$. There is:
$$card(\mathcal{F}^{WSC}) = N^{WSC} \tag{IV.3}$$
$card(*)$ represents the number of elements of a set.
- ${}^{e}\mathbf{F}_{j}^{WSC}$ the hydrodynamic loads acting on the body j at $\mathbf{G}_j$ in Σ_e given by (III.95);
- ${}^{e}\mathbf{G}_{j}$ the (6×6) matrix to change the point of computation of the hydrodynamic loads from $\mathbf{G}_j$ to $\mathbf{O}_j$:
$${}^{e}\mathbf{G}_{j} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{S}({}^{e}\mathbf{R}_{j}\, {}^{j}\mathbf{S}_{j}) & \mathbf{I}_3 \end{pmatrix} \tag{IV.4}$$

– ${}^0\mathbf{N}_e^j$ the (6×6) matrix to project the hydrodynamic loads into the base frame from $\mathbf{O}_j$ to $\mathbf{O}_0$:

$$ {}^0\mathbf{N}_e^j = \begin{pmatrix} {}^0\mathbf{R}_e & \mathbf{0}_{3\times3} \\ \mathbf{S}({}^0\mathbf{P}_j)\,{}^0\mathbf{R}_e & {}^0\mathbf{R}_e \end{pmatrix} \tag{IV.5} $$

$\mathbf{\Gamma}^{WSC}$ is the sum of the hydrodynamic articular loads:

$$ \mathbf{\Gamma}^{WSC} = \sum_{j \in \mathcal{F}^{WSC} \setminus \{0\}} \mathbf{\Gamma}_j^{WSC} \tag{IV.6} $$

where $\mathbf{\Gamma}_j^{WSC}$ is the projection of the hydrodynamic loads acting on the body j into the articular space [48]:

$$ \mathbf{\Gamma}_j^{WSC} = {}^j\mathbf{J}_j^T\, {}^j\boldsymbol{\rho}_e\, {}^e\mathbf{G}_j\, {}^e\mathbf{F}_j^{WSC} \tag{IV.7} $$

with:

– ${}^j\boldsymbol{\rho}_e$ the (6×6) projection matrix of the hydrodynamic loads from Σ_e to Σ_j:

$$ {}^j\boldsymbol{\rho}_e = \begin{pmatrix} {}^j\mathbf{R}_e & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & {}^j\mathbf{R}_e \end{pmatrix} \tag{IV.8} $$

– ${}^j\mathbf{J}_j$ the $(6 \times n)$ Jacobian matrix of the body j, defined by:

$$ {}^j\mathbf{V}_j = {}^j\mathbf{T}_0\, {}^0\mathbf{V}_0 + {}^j\mathbf{J}_j\, \dot{\mathbf{q}} \tag{IV.9} $$

The Jacobian matrix represents the mapping between the operating space (using Cartesian coordinates) and the articular space (using articular variables). If the base is fixed, the k^{th} column of ${}^j\mathbf{J}_j$ represents the contribution of the k^{th} articular velocity to the Cartesian velocity ${}^j\mathbf{V}_j$. Following this definition, the Jacobian matrix does not exist for the base $(j = 0)$.

The hydrodynamic loads are expressed by (III.110). Equation IV.1 becomes:

$$ \begin{aligned} \mathbf{H}\begin{pmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} - \begin{pmatrix} \sum\limits_{j \in \mathcal{F}^{WSC}} {}^0\mathbf{N}_e^j\, {}^e\mathbf{G}_j\, {}^e\mathbf{C}\mathbf{T}_j\, \dot{\boldsymbol{\phi}}^P(B_j) \\ \sum\limits_{j \in \mathcal{F}^{WSC} \setminus \{0\}} {}^j\mathbf{J}_j^T\, {}^j\boldsymbol{\rho}_e\, {}^e\mathbf{G}_j\, {}^e\mathbf{C}\mathbf{T}_j\, \dot{\boldsymbol{\phi}}^P(B_j) \end{pmatrix} \\ = \begin{pmatrix} \mathbf{0}_{6\times1} \\ \mathbf{\Gamma} \end{pmatrix} - \mathbf{C} + \begin{pmatrix} \sum\limits_{j \in \mathcal{F}^{WSC}} {}^0\mathbf{N}_e^j\, {}^e\mathbf{G}_j\, {}^e\mathbf{T}_j^{WSC} \\ \sum\limits_{j \in \mathcal{F}^{WSC} \setminus \{0\}} {}^j\mathbf{J}_j^T\, {}^j\boldsymbol{\rho}_e\, {}^e\mathbf{G}_j\, {}^e\mathbf{T}_j^{WSC} \end{pmatrix} \end{aligned} \tag{IV.10} $$

IV.2.3.2 Slip conditions

The slip conditions (III.109) need to be expressed with respect to the articular accelerations instead of the Cartesian accelerations. The first step is to write the Cartesian velocity with respect to the velocity of the base and the articular velocities. From (I.40), one has:

$$ {}^e\boldsymbol{\omega}_j = \mathbf{S}_j \begin{pmatrix} \dot{\varphi}_j \\ \dot{\theta}_j \\ \dot{\psi}_j \end{pmatrix} \tag{IV.11} $$

with:

$$\mathbf{S}_j = \begin{pmatrix} C_{\theta_j} C_{\psi_j} & -S_{\psi_j} & 0 \\ C_{\theta_j} S_{\psi_j} & C_{\psi_j} & 0 \\ -S_{\theta_j} & 0 & 1 \end{pmatrix} \tag{IV.12}$$

${}^e\mathbf{V}_j$ is the Cartesian velocity of body j at $\mathbf{O}_j$ with respect to Σ_e. $\dot{\boldsymbol{\eta}}_j$ is defined by (I.18) for the body 0 but its definition is extended for every body j. This velocity is expressed at $\mathbf{G}_j$ and using the Cardan angles. ${}^e\mathbf{V}_j$ and $\dot{\boldsymbol{\eta}}_j$ are related by:

$$\dot{\boldsymbol{\eta}}_j = \boldsymbol{\tau}_j^{-1\,e}\mathbf{V}_j \tag{IV.13}$$

with:

$$\boldsymbol{\tau}_j^{-1} = \begin{pmatrix} \mathbf{I}_3 & -\mathbf{S}({}^e\mathbf{R}_j\,{}^j\mathbf{S}_j) \\ \mathbf{0}_{3\times 3} & \mathbf{S}_j^{-1} \end{pmatrix} \tag{IV.14}$$

The $(6\times(n+6))$ generalized Jacobian matrix ${}^e\boldsymbol{\Lambda}_j$ allows to relate the (6×1) velocity vector of body j in Σ_e to the base and articular velocities:

$${}^e\mathbf{V}_j = {}^e\boldsymbol{\Lambda}_j \begin{pmatrix} {}^0\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} \tag{IV.15}$$

The kth column of ${}^e\boldsymbol{\Lambda}_j$ represents the contribution of the kth component of the base velocity ${}^0\mathbf{V}_0$, if $k \leqslant 6$, or the kth, if $k > 6$, articular velocity to the Cartesian velocity ${}^e\mathbf{V}_j$.

After time-differentiation of (IV.15) and (IV.13) and the introduction of (I.61), the Cartesian acceleration, $\ddot{\boldsymbol{\eta}}_j$, can be expressed as function of the base and articular accelerations:

$$\begin{aligned} \ddot{\boldsymbol{\eta}}_j = \dot{\boldsymbol{\tau}}_j^{-1\,e}\boldsymbol{\Lambda}_j \begin{pmatrix} {}^0\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} + \boldsymbol{\tau}_j^{-1\,e}\dot{\boldsymbol{\Lambda}}_j \begin{pmatrix} {}^0\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} \\ + \boldsymbol{\tau}_j^{-1\,e}\boldsymbol{\Lambda}_j \left[\begin{pmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} - \begin{pmatrix} \begin{pmatrix} \mathbf{S}({}^0\boldsymbol{\omega}_0)\,{}^0\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} \\ \mathbf{0}_{n\times 1} \end{pmatrix} \right] \end{aligned} \tag{IV.16}$$

The time-derivation of $\boldsymbol{\tau}_j^{-1}$ is expressed by:

$$\dot{\boldsymbol{\tau}}_j^{-1} = \begin{pmatrix} \mathbf{0}_{3\times 3} & -{}^e\mathbf{R}_j\,\mathbf{S}({}^j\boldsymbol{\omega}_j)\mathbf{S}({}^j\mathbf{S}_j)\,{}^j\mathbf{R}_e \\ \mathbf{0}_{3\times 3} & -\mathbf{S}_j^{-1}\dot{\mathbf{S}}_j\mathbf{S}_j^{-1} \end{pmatrix} \tag{IV.17}$$

Finally, the introduction of (IV.16) in (III.109) yields:

$$\begin{aligned} \dot{\boldsymbol{\phi}}_n^P(B_j) = -\dot{\boldsymbol{\phi}}_n^I(B_j) + \mathbf{C}\mathbf{K}_j\boldsymbol{\tau}_j^{-1\,e}\boldsymbol{\Lambda}_j \begin{pmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} \\ + \mathbf{C}\mathbf{K}_j \left[(\dot{\boldsymbol{\tau}}_j^{-1\,e}\boldsymbol{\Lambda}_j + \boldsymbol{\tau}_j^{-1\,e}\dot{\boldsymbol{\Lambda}}_j) \begin{pmatrix} {}^0\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} - \boldsymbol{\tau}_j^{-1\,e}\boldsymbol{\Lambda}_j \begin{pmatrix} \begin{pmatrix} \mathbf{S}({}^0\boldsymbol{\omega}_0)\,{}^0\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} \\ \mathbf{0}_{n\times 1} \end{pmatrix} \right] + \mathbf{Q}_j \end{aligned} \tag{IV.18}$$

It remains to clarify the expressions of the Jacobian matrix ${}^j\mathbf{J}_j$, the generalized Jacobian matrix ${}^e\boldsymbol{\Lambda}_j$ and its time-differentiation ${}^e\dot{\boldsymbol{\Lambda}}_j$.

IV.2.3.3 Jacobian matrices

From the definition of the Jacobian matrices given in IV.2.3.1, let us consider the kth articulation of a multibody system and a body j such as $j > k$. The base is assumed motionless. The articular velocity $\dot{q}_k$ leads to a velocity vector ${}^k\mathbf{V}_j^k$ for the body j:

$$ {}^k\mathbf{V}_j^k = \begin{pmatrix} \left[\sigma_k\, {}^k\mathbf{z}_k + \bar{\sigma}_k({}^k\mathbf{z}_k \times {}^k\mathbf{P}_j)\right]\dot{q}_k \\ \bar{\sigma}_k\, {}^k\mathbf{z}_k\, \dot{q}_k \end{pmatrix} \tag{IV.19} $$

By summing the contributions of all the joints of the direct branch $\beta(j)$ (as defined in subsection I.4.3), one can have:

$$ \mathbf{V}_j = \sum_{k\in\beta(j)} {}^k\mathbf{V}_j^k \tag{IV.20} $$

$$ = \begin{pmatrix} \sum\limits_{k\in\beta(j)} \left[\sigma_k\, {}^k\mathbf{z}_k + \bar{\sigma}_k({}^k\mathbf{z}_k \times {}^k\mathbf{P}_j)\right]\dot{q}_k \\ \sum\limits_{k\in\beta(j)} \bar{\sigma}_k\, {}^k\mathbf{z}_k\, \dot{q}_k \end{pmatrix} \tag{IV.21} $$

It is necessary to define a common frame of projection, for instance Σ_j:

$$ {}^j\mathbf{V}_j = \begin{pmatrix} \sum\limits_{k\in\beta(j)} \left[\sigma_k\, {}^j\mathbf{z}_k + \bar{\sigma}_k\mathbf{S}({}^j\mathbf{z}_k)\, {}^j\mathbf{R}_k\, {}^k\mathbf{P}_j)\right]\dot{q}_k \\ \sum\limits_{k\in\beta(j)} \bar{\sigma}_k\, {}^j\mathbf{z}_k\, \dot{q}_k \end{pmatrix} \tag{IV.22} $$

Thus, following the matrix format of (IV.9), the expression of ${}^j\mathbf{J}_j$ is:

$$ col_k({}^j\mathbf{J}_j) = \begin{cases} \begin{pmatrix} \left[\sigma_k\, {}^j\mathbf{z}_k + \bar{\sigma}_k\mathbf{S}({}^j\mathbf{z}_k)\, {}^j\mathbf{R}_k\, {}^k\mathbf{P}_j)\right]\dot{q}_k \\ \bar{\sigma}_k\, {}^j\mathbf{z}_k\, \dot{q}_k \end{pmatrix} & \text{if } k \in \beta(j) \\ \mathbf{0}_{6\times 1} & \text{if } k \notin \beta(j) \end{cases} \tag{IV.23} $$

The change of frames is achieved using the projection matrix (IV.8), for instance in the base frame:

$$ {}^0\mathbf{J}_j = {}^0\boldsymbol{\rho}_j\, {}^j\mathbf{J}_j \tag{IV.24} $$

IV.2.3.4 Expression of ${}^e\mathbf{\Lambda}_j$

The projection in Σ_e of (IV.9) leads to:

$$ {}^e\mathbf{V}_j = {}^e\boldsymbol{\rho}_j\, {}^j\mathbf{V}_j \tag{IV.25} $$

$$ = {}^e\boldsymbol{\rho}_j\, {}^j\mathbf{T}_0\, {}^0\mathbf{V}_0 + {}^e\boldsymbol{\rho}_j\, {}^j\mathbf{J}_j\, \dot{\mathbf{q}} \tag{IV.26} $$

$$ = {}^e\boldsymbol{\rho}_0\, {}^0\mathbf{L}_j\, {}^0\mathbf{V}_0 + {}^e\boldsymbol{\rho}_0\, {}^0\mathbf{J}_j\, \dot{\mathbf{q}} \tag{IV.27} $$

$$ = {}^e\boldsymbol{\rho}_0 \left({}^0\mathbf{L}_j\, {}^0\mathbf{V}_0 + {}^0\mathbf{J}_j\, \dot{\mathbf{q}}\right) \tag{IV.28} $$

where:

– ${}^0\mathbf{L}_j$ changes the point of computation of ${}^0\mathbf{V}_0$ from $\mathbf{O}_0$ to $\mathbf{O}_j$:

$$ {}^0\mathbf{L}_j = \begin{pmatrix} \mathbf{I}_3 & -\mathbf{S}({}^0\mathbf{P}_j) \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{pmatrix} \tag{IV.29} $$

– the expression of ${}^0\mathbf{J}_j$ is given by (IV.24).

Based on (IV.15), the expression of the generalized Jacobian matrix is:

$$ {}^e\mathbf{\Lambda}_j = \begin{cases} {}^e\boldsymbol{\rho}_0 \begin{pmatrix} {}^0\mathbf{L}_j & {}^0\mathbf{J}_j \end{pmatrix} & \text{if } j \neq 0 \\ {}^e\boldsymbol{\rho}_0 \begin{pmatrix} {}^0\mathbf{L}_j & \mathbf{0}_{6\times n} \end{pmatrix} & \text{if } j = 0 \end{cases} \tag{IV.30} $$

IV.2.3.5 Expression of ${}^{e}\dot{\mathbf{\Lambda}}_j$

The time-differentiation of (IV.30) is challenging because of the Jacobian matrix ${}^{0}\mathbf{J}_j$. Two methods exist to compute the time-derivation of such a Jacobian matrix. The first one is a direct calculation as presented by Hourtash [168]. The drawback of this approach is its complexity. The second method is based on a recursive algorithm which computes ${}^{e}\dot{\mathbf{\Lambda}}_j \begin{pmatrix} {}^{0}\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix}$ instead of ${}^{e}\dot{\mathbf{\Lambda}}_j$ directly [169].

The time-derivative of (IV.15) is:

$$ {}^{e}\dot{\mathbf{V}}_j = {}^{e}\dot{\mathbf{\Lambda}}_j \begin{pmatrix} {}^{0}\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} + {}^{e}\mathbf{\Lambda}_j \left[\begin{pmatrix} {}^{0}\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} - \begin{pmatrix} \begin{pmatrix} \mathbf{S}({}^{0}\boldsymbol{\omega}_0)\,{}^{0}\mathbf{v}_0 \end{pmatrix} \\ \mathbf{0}_{3\times 1} \\ \mathbf{0}_{n\times 1} \end{pmatrix} \right] \tag{IV.31} $$

Assuming ${}^{0}\dot{\mathbf{V}}_0 = \mathbf{0}_{6\times 1}$ and $\ddot{q} = \mathbf{0}_{n\times 1}$ and applying (IV.30):

$$ {}^{e}\dot{\mathbf{V}}_j^{*} = {}^{e}\dot{\mathbf{\Lambda}}_j \begin{pmatrix} {}^{0}\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} - {}^{e}\boldsymbol{\rho}_0 \begin{pmatrix} \mathbf{S}({}^{0}\boldsymbol{\omega}_0)\,{}^{0}\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{IV.32} $$

By denoting:

$$ {}^{e}\dot{\mathbf{V}}_j^{*} = {}^{e}\dot{\mathbf{V}}_j \Big|_{\substack{{}^{0}\dot{\mathbf{V}}_0 = \mathbf{0}_{6\times 1} \\ \ddot{\mathbf{q}} = \mathbf{0}_{n\times 1}}} \tag{IV.33} $$

${}^{e}\dot{\mathbf{V}}_j^{*}$ can also be expressed by:

$$ {}^{e}\dot{\mathbf{V}}_j^{*} = {}^{e}\boldsymbol{\rho}_j\, {}^{j}\dot{\mathbf{V}}_j^{*} \tag{IV.34} $$

The introduction of (IV.34) in (IV.32) gives:

$$ {}^{e}\dot{\mathbf{\Lambda}}_j \begin{pmatrix} {}^{0}\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} = {}^{e}\boldsymbol{\rho}_j\, {}^{j}\dot{\mathbf{V}}_j^{*} + {}^{e}\boldsymbol{\rho}_0 \begin{pmatrix} \mathbf{S}({}^{0}\boldsymbol{\omega}_0)\,{}^{0}\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{IV.35} $$

Using the assumption $\ddot{q} = \mathbf{0}_{n\times 1}$ in (I.48) allows writing for $i = a_j$:

$$ {}^{j}\dot{\mathbf{V}}_j^{*} = {}^{j}\mathbf{T}_i\, {}^{i}\dot{\mathbf{V}}_i^{*} + {}^{j}\boldsymbol{\gamma}_j \tag{IV.36} $$

While the hypothesis ${}^{0}\dot{\mathbf{V}}_0 = \mathbf{0}_{6\times 1}$ initializes this recursive equation:

$$ {}^{0}\dot{\mathbf{V}}_0^{*} = \mathbf{0}_{6\times 1} \tag{IV.37} $$

Using (IV.36) and (IV.37), ${}^{j}\dot{\mathbf{V}}_j^{*}$ can be computed for each body during the Composite-Rigid-Body Algorithm. Then, the quantity ${}^{e}\dot{\mathbf{\Lambda}}_j \begin{pmatrix} {}^{0}\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix}$ can be calculated from (IV.35).

Thereby:

$$ {}^{e}\dot{\mathbf{\Lambda}}_j \begin{pmatrix} {}^{0}\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{cases} \left\{ \begin{array}{l} {}^{e}\boldsymbol{\rho}_j\, {}^{j}\dot{\mathbf{V}}_j^{*} + {}^{e}\boldsymbol{\rho}_0 \begin{pmatrix} \mathbf{S}({}^{0}\boldsymbol{\omega}_0)\,{}^{0}\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} \\ \text{with:} \\ {}^{j}\dot{\mathbf{V}}_j^{*} = {}^{j}\mathbf{T}_i\, {}^{i}\dot{\mathbf{V}}_i^{*} + {}^{j}\boldsymbol{\gamma}_j \text{ for } i = a_j \\ {}^{0}\dot{\mathbf{V}}_0^{*} = \mathbf{0}_{6\times 1} \end{array} \right\} & \text{if } j \neq 0 \\ {}^{e}\boldsymbol{\rho}_0 \begin{pmatrix} \mathbf{S}({}^{0}\boldsymbol{\omega}_0)\,{}^{0}\mathbf{v}_0 \\ \mathbf{0}_{3\times 1} \end{pmatrix} & \text{if } j = 0 \end{cases} \tag{IV.38} $$

IV.2.3.6 Linear system and time integration

The tight fluid-structure coupling between the Composite-Rigid-Body Algorithm and the weakly nonlinear potential flow theory is formed by (III.108), (IV.10) and (IV.18):

$$\begin{cases} \mathbf{G}\dot{\boldsymbol{\phi}} = \mathbf{H}\dot{\boldsymbol{\phi}}_n \\ \mathbf{H}\begin{pmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} - \begin{pmatrix} \sum\limits_{j\in\mathcal{F}^{WSC}} {}^0\mathbf{N}_e^j\, {}^e\mathbf{G}_j\, {}^e\mathbf{C}\mathbf{T}_j\, \dot{\boldsymbol{\phi}}^P(B_j) \\ \sum\limits_{j\in\mathcal{F}^{WSC}\setminus\{0\}} {}^j\mathbf{J}_j^T\, {}^j\boldsymbol{\rho}_e\, {}^e\mathbf{G}_j\, {}^e\mathbf{C}\mathbf{T}_j\, \dot{\boldsymbol{\phi}}^P(B_j) \end{pmatrix} = \\ \qquad\qquad \begin{pmatrix} \mathbf{0}_{6\times1} \\ \boldsymbol{\Gamma} \end{pmatrix} - \mathbf{C} + \begin{pmatrix} \sum\limits_{j\in\mathcal{F}^{WSC}} {}^0\mathbf{N}_e^j\, {}^e\mathbf{G}_j\, {}^e\mathbf{T}_j^{WSC} \\ \sum\limits_{j\in\mathcal{F}^{WSC}\setminus\{0\}} {}^j\mathbf{J}_j^T\, {}^j\boldsymbol{\rho}_e\, {}^e\mathbf{G}_j\, {}^e\mathbf{T}_j^{WSC} \end{pmatrix} \\ \dot{\boldsymbol{\phi}}_n^P(B_j) - \mathbf{C}\mathbf{K}_j\boldsymbol{\tau}_j^{-1\,e}\boldsymbol{\Lambda}_j \begin{pmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} = \\ -\dot{\boldsymbol{\phi}}_n^I(B_j) + \mathbf{C}\mathbf{K}_j \left[(\dot{\boldsymbol{\tau}}_j^{-1\,e}\boldsymbol{\Lambda}_j + \boldsymbol{\tau}_j^{-1\,e}\dot{\boldsymbol{\Lambda}}_j) \begin{pmatrix} {}^0\mathbf{V}_0 \\ \dot{\mathbf{q}} \end{pmatrix} - \boldsymbol{\tau}_j^{-1\,e}\boldsymbol{\Lambda}_j \begin{pmatrix} \begin{pmatrix} \mathbf{S}({}^0\boldsymbol{\omega}_0)\,{}^0\mathbf{v}_0 \\ \mathbf{0}_{3\times1} \end{pmatrix} \\ \mathbf{0}_{n\times1} \end{pmatrix} \right] + \mathbf{Q}_j \;\forall j \in \mathcal{F}^{WSC} \end{cases} \tag{IV.39}$$

Equation IV.39 may be re-written as a $(N + \sum\limits_{j\in\mathcal{F}^{WSC}} N(B_j) + n + 6)$ linear system: $\mathbf{AX} = \mathbf{B}$. N is the total number of nodes in the mesh, $N(B_j)$ is the number of nodes in the mesh of the body j and n is the number of degrees of freedom in the multibody system. It is solved by using a *GMRES* method with a diagonal preconditioner. Doing so, the vector $\mathbf{X}$ is known:

$$\mathbf{X} = \begin{pmatrix} \dot{\boldsymbol{\phi}}_n^P(FS) \\ \dot{\boldsymbol{\phi}}^P(T) \\ \vdots \\ \dot{\boldsymbol{\phi}}^P(B_j) \\ \vdots \\ \dot{\boldsymbol{\phi}}_n^P(B_j) \\ \vdots \\ {}^0\dot{\mathbf{V}}_0 \\ \ddot{\boldsymbol{q}} \end{pmatrix} \tag{IV.40}$$

The state vector of this coupling ($CRBA - WSC$) is:

$$\mathbf{Y}_{CRBA-WSC} = \begin{pmatrix} \boldsymbol{\phi}^P(FS) \\ \boldsymbol{\eta}^P \\ \mathbf{P}_{mesh} \\ \vdots \\ \boldsymbol{\eta}_j \\ \vdots \\ \dot{\boldsymbol{\eta}}_j \\ \vdots \\ \boldsymbol{\nu}_0 \\ \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} \tag{IV.41}$$

Its time-differentiation is obtained using:

- (III.29) for $\dot{\boldsymbol{\phi}}^P(FS)$ and $\dot{\boldsymbol{\eta}}^P$;
- (III.141) and (III.147) for $\mathbf{v}_{mesh}$;
- (IV.16) for $\ddot{\boldsymbol{\eta}}_j$;
- (I.93) and (I.61) for $\dot{\boldsymbol{\nu}}_0$;
- (I.93) for $\ddot{\mathbf{q}}$.

The time-stepping is achieved using a fourth-order explicit Runge-Kutta scheme with a fixed time step (I.94).

IV.2.4 Language binding

The Composite-Rigid-Body Algorithm is implemented in C++ in the numerical tool *InWave* while the weakly nonlinear potential flow theory is implemented in Fortran 90 in the solver *WS_CN*. Thus, a language binding is mandatory because of the two different programming languages. As explained in subsection IV.1.3, different possibilities of language binding exist. The source code of both *InWave* and *WS_CN* are available so the use of a glue code to guarantee the modularity of the numerical fluid-structure coupling is favoured. The co-simulation is driven in a Python environment. *Cython* makes the language binding between the C++ source code and the Python glue code whereas *f2py* does the same for the Fortran 90 source code. These tools compile the source code and create a Python library (.pyd) for each solver which can be called in a Python environment.

The creation and the solving of the tight fluid-structure coupling along with the temporal loop of the co-simulation are achieved by a Python numerical tool named *InWaveS_CN*. Figure IV.6 presents the relationships between *InWave*, *WS_CN* and *InWaveS_CN*.

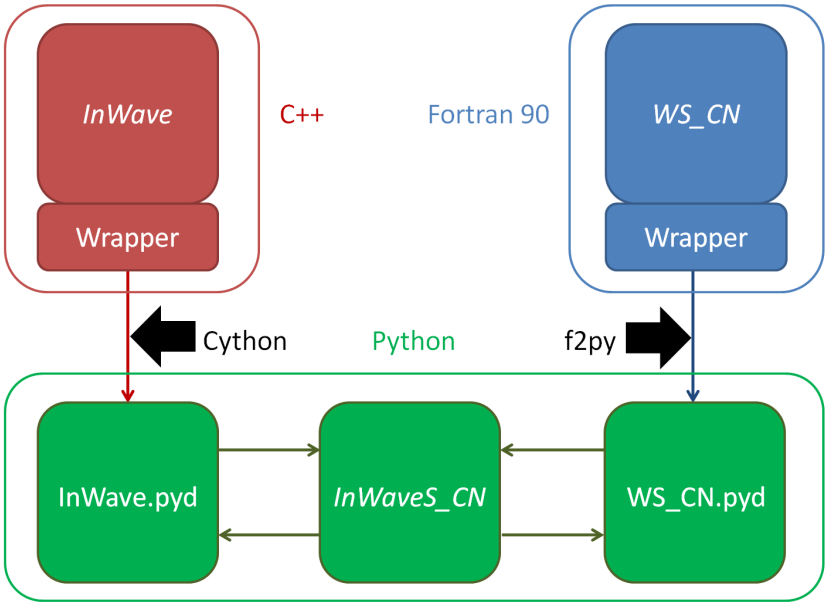


Figure IV.6 – Language binding between *InWave* and *WS_CN* using Python as glue code language. The red, green and blue colors represent the C++, Python and Fortran languages.

IV.2.5 Implementation

The implementation of the tight coupling between *InWave* and *WS_CN* unfolds in seven steps:

1. *InWave* performs the three stages of the Composite-Rigid-Body Algorithm while *WS_CN* computes the influence coefficients and solves the first boundary value problem;
2. The quantities, required for solving the fluid-structure interaction, are sent to *InWaveS_CN*;
3. The linear system (IV.39) is formed;
4. The linear operator, the right-hand side along with the initialization of the solution are sent to *WS_CN*;
5. The iterative method *GMRES* is applied to solve the linear system;
6. The solution of the linear system is sent to *InWaveS_CN*;
7. The solution is assigned to the fluid and mechanical solvers, the multibody acceleration is expressed into the Cartesian space. The state vector of each solver is time-stepped.

Figure IV.7 presents the sketch of the seven different steps. Only one stage of the fourth-order Runge-Kutta method is represented.

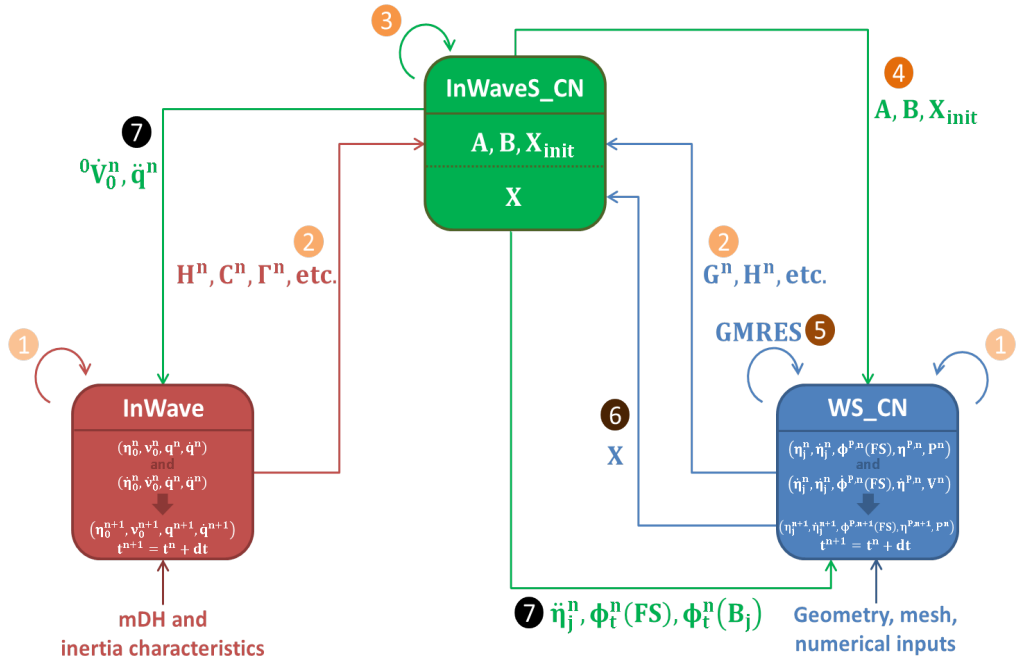


Figure IV.7 – The seven steps of the tight coupling between *InWave* and *WS_CN*. The red, green and blue colors represent the C++, Python and Fortran languages. The arrows denote the communication between the modules. The color of the arrows depends on the programming language which sends the data. The superscript n denotes the n^{th} time step.

IV.3 Validation

The fluid-structure coupling presented in this chapter is now compared to an academic test case which can also be run by the weakly nonlinear potential flow-based solver alone.

Two identical surface-piercing floating circular cylinders are considered. Both floating bodies (named *Cylinder 1* and *Cylinder 2*) have a radius of 0.2 m, a height of 1 m and a draft of 0.5 m. The position of the centers of gravity along with inertia characteristics are given in Table IV.2. At the starting time, the two cylinders are aligned along the **y**-axis and separated by 1 m. The incident wave is regular, with an amplitude of 0.01 m, a frequency of 8 rad/s and propagating towards the positive x. A cylindrical domain of radius 2λ is used with an absorbing beach length of λ. The initial mesh, with 10 650 panels, is showed in Figure IV.8. The time step is 0.01 s.

Each body has three degrees of freedom; surge, heave and pitch. Regarding the multibody modelling, *Cylinder 1* is considered as the base of the multibody system. Three joints are necessary to ensure the three *dof* of *Cylinder 2*. The two first joints are prismatic while the last one is revolute. The modified Denavit-Hartenberg parameters of the multibody system are given in Table IV.3. d_1 represents the relative distance between the two cylinders.

The time series of the motions of both *Cylinder 1* and *Cylinder 2* are presented in Figures IV.9 and IV.10. Numerical results coming from *WS_CN* are based on (III.112) while those obtained with *InWaveS_CN* use (IV.39). They show a perfect

agreement between the two formulations which validates the implementation of the present coupling.

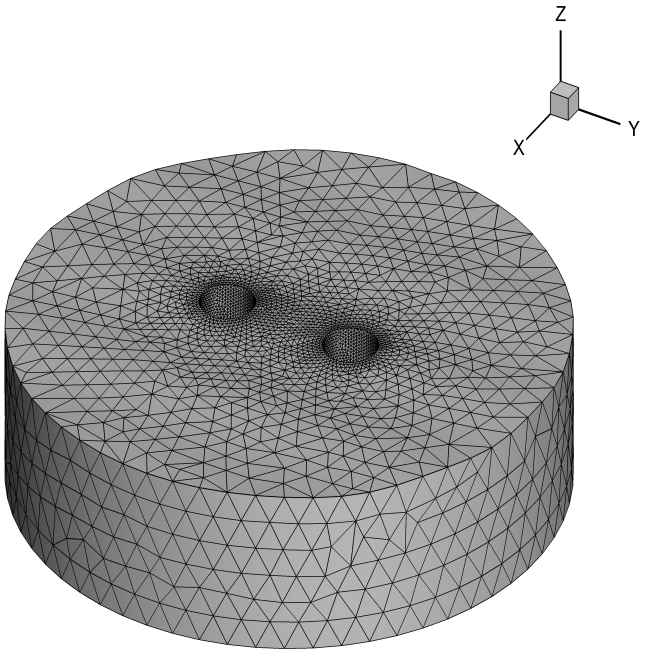


Figure IV.8 – Initial mesh of the comparison test case between *WS_CN* and *InWaveS_CN*

CoG of Cylinder 1 (m)	$(0, -0.5, 0)$
CoG of Cylinder 2 (m)	$(0, 0.5, 0)$
Mass (kg)	64.4
I^G_{yy} (kg m^2)	10

Table IV.2 – Inertia characteristics and initial position of the centres of gravity of *Cylinder 1* and *2*

j	σ_j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	1	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	1	π	0
2	1	0	0	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0
3	0	0	0	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	0

Table IV.3 – *mDH* parameters associated with the multibody system

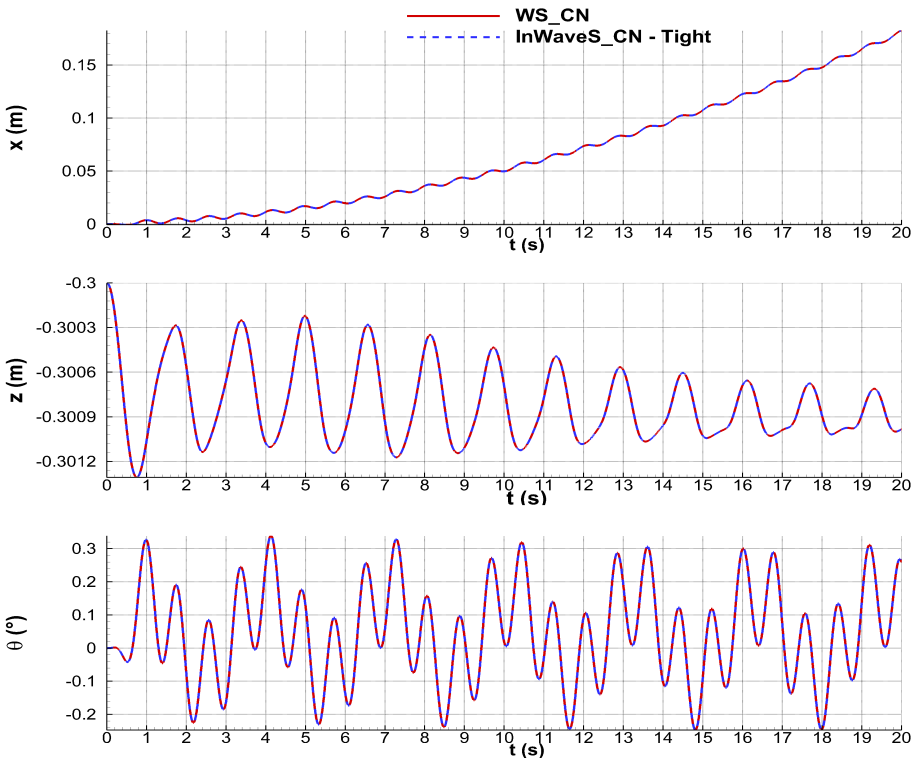


Figure IV.9 – Comparison of time series of the motions of *Cylinder 1* from numerical results of *InWaveS_ CN* and *WS_ CN*

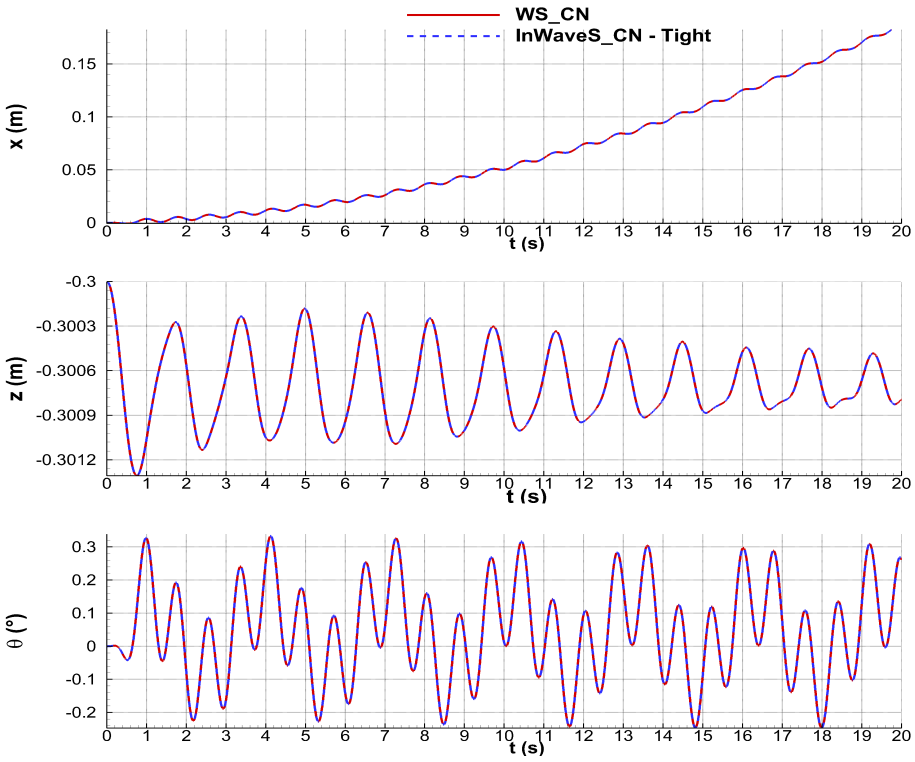


Figure IV.10 – Comparison of time series of the motions of *Cylinder 2* from numerical results of *InWaveS_ CN* and *WS_ CN*

IV.4 Other coupling strategies

Using the Python environment, loose coupling strategies are easily feasible and may be compared to the tight coupling presented above. A tight coupling solves (IV.39) whereas a loose coupling solves Equations (III.108) and (III.109) first and then (IV.1) or *vice versa*. In case of an iterative loose coupling (or implicit loose coupling) a test condition is necessary within the internal loop. It is given by (IV.42). The triplet (p, n, i) denotes the p^{th} component of the vector, the n^{th} time step and the i^{th} internal step.

$$\begin{cases} \left| {}^0\dot{\mathbf{V}}_0(p,n,i+1) - {}^0\dot{\mathbf{V}}_0(p,n,i) \right| & \leqslant \epsilon \ \forall p \in [\![1\ ;\ 6]\!] \\ |\ddot{\mathbf{q}}(p,n,i+1) - \ddot{\mathbf{q}}(p,n,i)| & \leqslant \epsilon \ \forall p \in [\![1\ ;\ n]\!] \\ max\left(\left| \dot{\boldsymbol{\phi}}^P(B_j,n,i+1) - \dot{\boldsymbol{\phi}}^P(B_j,n,i) \right| \right) & \leqslant \epsilon \ \forall j \in [\![1\ ;\ N]\!] \end{cases} \tag{IV.42}$$

ϵ represents the tolerance of the *GMRES* method. For the sake of speed, a maximum of 30 internal steps is fixed.

Three following couplings are considered:

- Tight;
- Explicit loose with a fluid formulation;
- Implicit loose with a fluid formulation.

They are compared using the test case presented in section IV.3. Figure IV.11 and Figure IV.12 show the surge, heave and pitch motions for the Cylinders 1 and 2 based on a tight, loose explicit and loose implicit couplings. Loose couplings use a fluid formulation. As expected, the tight and implicit loose couplings match perfectly while the explicit loose coupling shows some differences. This is due to the time lag between the computations of the hydrodynamic loads and the accelerations. Regarding the implicit loose coupling, this approach needs between 15 and 25 internal iterations per computation. The addition of a predictor and/or corrector within the internal loop could reduce the number of iterations. Table IV.4 presents the CPU time for each coupling strategy without using the parallelization of the influence coefficients. The explicit loose coupling is quicker than the tight coupling but it is less accurate. The implicit loose coupling has an opposite behaviour. Finally, the tight coupling is a good compromise between the CPU cost, the robustness and the accuracy. The increase of the CPU time when the tight coupling is used is a consequence of the communication time between the different modules and the different languages.

The level of differences between a tight and a loose explicit coupling depends on the case (incident wave, hydrodynamic interactions, etc.). For instance, if the center-to-center distance the two cylinders is 0.6 m instead of 1.0 m ($d_1 = 0.6$), the differences in heave motion increase because of the more important hydrodynamic interactions between the bodies (Figure IV.13).

Moreover, the tight and implicit loose couplings of *InWaveS_CN* along with the weakly nonlinear potential flow-based solver *WS_CN* lead to the same results whereas three different sets of differential equations need to be solved. This fact ensures the validation of the different fluid-structure couplings presented in this chapter.

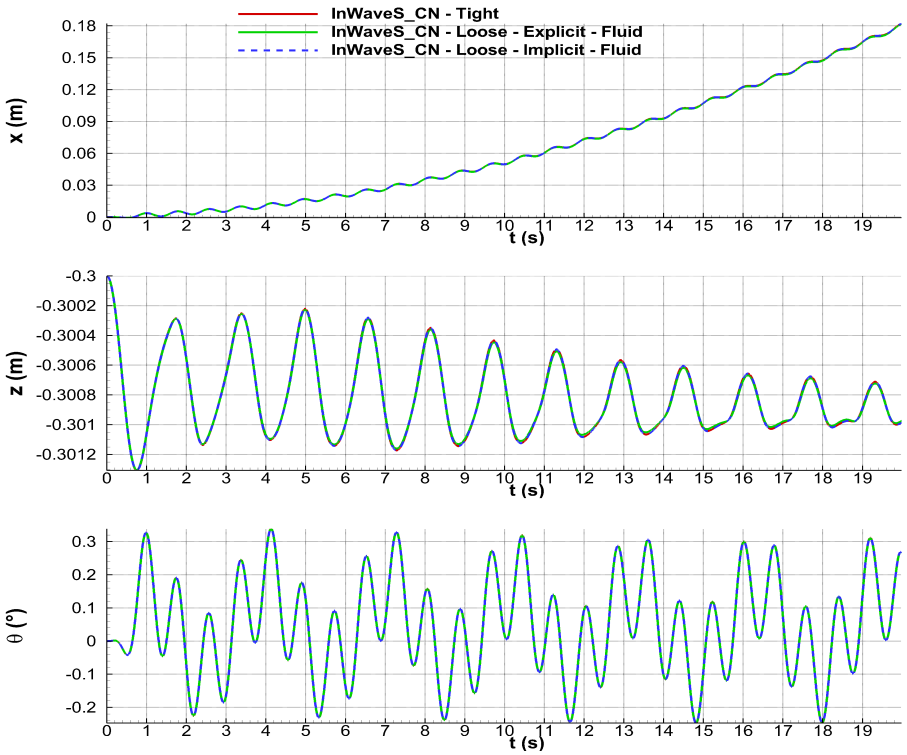


Figure IV.11 – Comparison of time series of the motions of *Cylinder 1* from numerical results of *InWaveS_ CN* for a tight, loose explicit and implicit couplings

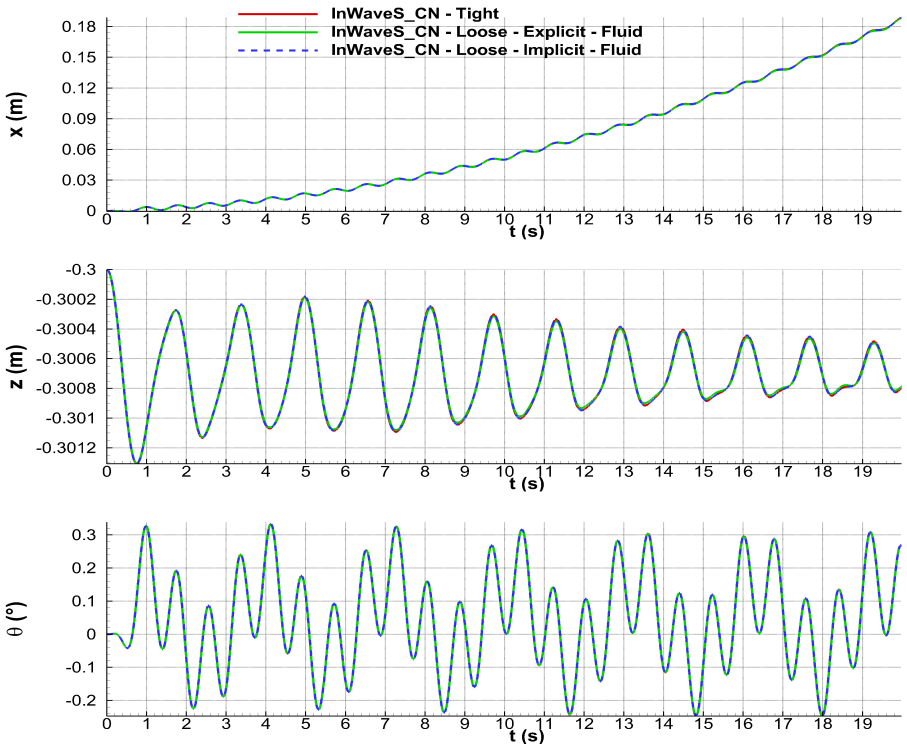


Figure IV.12 – Comparison of time series of the motions of *Cylinder 2* from numerical results of *InWaveS_ CN* for a tight, loose explicit and implicit couplings

Coupling strategy	Numerical tool	CPU time (h)
Monolithic	*WS_ CN*	40.7
Tight	*InWaveS_ CN*	49.2
Explicit loose	*InWaveS_ CN*	33.6
Implicit loose	*InWaveS_ CN*	67.0

Table IV.4 – Comparison between different coupling strategies

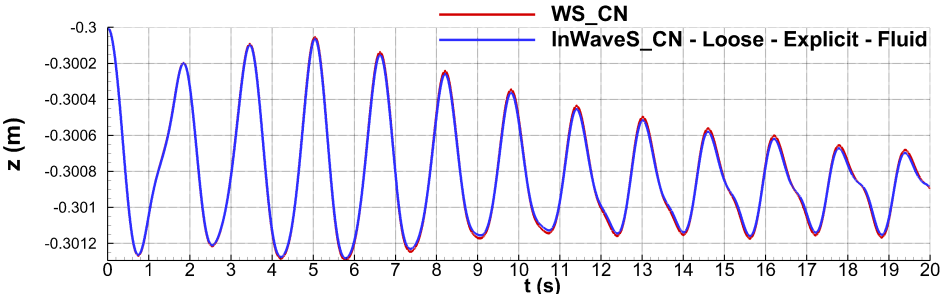


Figure IV.13 – Comparison of the heave motion of *Cylinder 1* with $d_1 = 0.6$ from numerical results of *WS_ CN* and *InWaveS_ CN* with a loose explicit coupling

IV.5 Coupling between the *CRBA* and the low-order lumped mass theory

The coupling between the *InWave* and *WS_ CN* may be used to performed a lowering operation including a vessel, a hoisting cable and a payload. But, as soon as there is a cable with one extremity connected to the multibody system and the other extremity fixed in the Cartesian frame, the *CRBA* cannot be used. This is the case, for example, for the mooring cables where the endpoints are fixed to the vessel and the seabed or the hoisting cable used in chapter VI where the position of the winch and the rotation axis of the buoy are fixed in the global frame. In the direct dynamics algorithms of *InWave*, it is only possible to prescribe the position of the base in Σ_e, the other bodies being located using relative coordinates. The body positions may be prescribed when inverse dynamics algorithms are used. Consequently, two choices are possible to figure out this problem. Either developing some inverse dynamics algorithms or simulating cables using the Cartesian coordinates. This latter approach is preferred because, as explained in section II.2, a low-order lumped mass theory based solver, named *CableDyn*, has been developed and can be easily coupled with *InWaveS_ CN*.

IV.5.1 Coupling strategy

The low-order lumped mass theory and the Composite-Rigid-Body algorithm are coupled using an explicit loose coupling for the sake of ease of development (cf. subsubsection IV.1.1.3). In other words, at each time step, the spring loads are evaluated from the position of the multibody system at the previous time step. Then, these loads are added to the multibody motion equation (IV.1). A tight coupling would have been more complex to implement as it would require to solve both the multibody motion equation

(IV.1) and the motion equation for each cable node (II.23) simultaneously. Furthermore, the low-order lumped mass theory is based on Cartesian coordinates whereas the multibody theory uses relative coordinates, therefore a change of coordinate system for one of the two solvers would be mandatory. A loose coupling between a floating body and cables (for example a mooring system) provides good results compared to a tight coupling as shown by Jacob et al. [170].

One advantage of the loose coupling with the cable solver is the possibility to fit the time step, by the use of the subcycling technique. Indeed, the time step to solve the cable dynamics is usually much smaller than for the fluid-structure dynamics. It is also possible to use another time integration scheme than the fourth-order Runge-Kutta scheme of *InWave* to add some numerical damping in the cable solver (cf. subsection II.1.2). If necessary, it is easy to transform the present explicit loose coupling in an implicit loose coupling with the addition of an retroactive loop.

Finally, the fluid-structure coupling is tight while the cable-structure coupling is loose and explicit. Figure IV.14 shows the algorithm of the coupling between the three solvers: *InWave*, *WS_CN* and *CableDyn*.

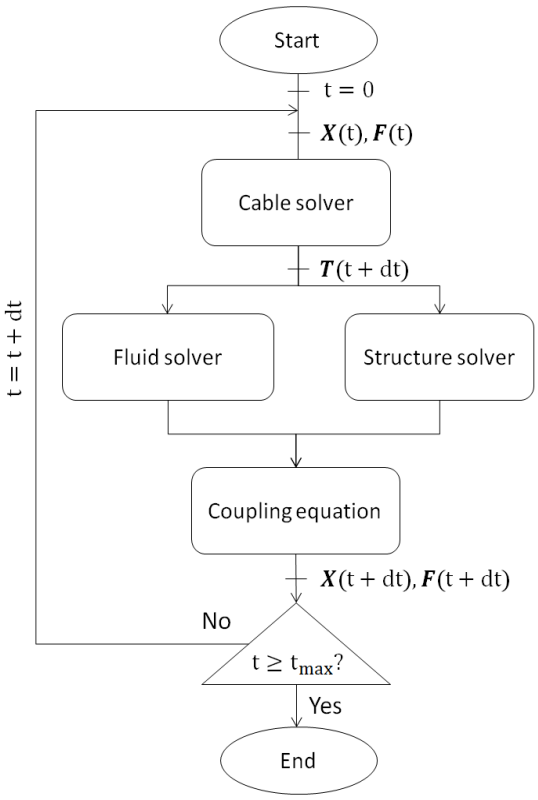


Figure IV.14 – Tight-loose coupling between *InWave* (structure solver), *WS_CN* (fluid solver) and *CableDyn* (cable solver). **X**, **F** and **T** represent the motion of the structure, the hydrodynamic loads and the cable loads.

IV.5.2 Theoretical developments

Let us consider a cable of which one extremity, the *anchor*, is position-controlled and the other extremity, the *fairlead*, is fixed to the j^{th} body of the multibody system at the point $\mathbf{A}_j$. The notations defined in chapter I are used in the rest of this section. Following the method used in subsubsection IV.2.3.1 to apply the hydrodynamic loads,

the cable loads are directly added to the multibody motion equation (IV.1):

$$\mathbf{H}\begin{pmatrix} {}^{0}\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{\Gamma} \end{pmatrix} - \mathbf{C} + \begin{pmatrix} {}^{0}\mathbf{F}^{WSC} \\ \boldsymbol{\Gamma}^{WSC} \end{pmatrix} + \begin{pmatrix} {}^{0}\mathbf{F}^{Cable} \\ \boldsymbol{\Gamma}^{Cable} \end{pmatrix} \tag{IV.43}$$

${}^{0}\mathbf{F}^{Cable}$ is the sum of all cable loads acting on the whole multibody system at $\mathbf{O}_0$ and projected in the base frame:

$${}^{0}\mathbf{F}^{Cable} = \sum_{j\in\mathcal{F}^{Cable}} {}^{0}\mathbf{N}_e^j \, {}^{e}\mathbf{G}_j^{Cable} \, {}^{e}\mathbf{F}_j^{Cable} \tag{IV.44}$$

with:

- $\mathcal{F}^{Cable}$ the set of the bodies subject to cable loads;
- ${}^{e}\mathbf{F}_j^{Cable}$ the cable loads acting on the body j at $\mathbf{A}_j$ in Σ_e and expressed by:

$${}^{e}\mathbf{F}_j^{Cable} = \mathbf{T}_0 + \mathbf{C}_0 \tag{IV.45}$$

  The subscript 0 refers to the node 0 of the cable (cf. subsection II.2.2), which is the fairlead by definition. The axial tension $\mathbf{T}_0$ and the axial damping force $\mathbf{C}_0$ are evaluated from (II.29) and (II.32), for each cable.

- ${}^{e}\mathbf{G}_j^{Cable}$ the (6×6) matrix to change the point of computation of the cable loads from $\mathbf{A}_j$ to $\mathbf{O}_j$:

$${}^{e}\mathbf{G}_j^{Cable} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{S}({}^{e}\mathbf{R}_j \, {}^{j}\mathbf{O}_j\mathbf{A}_j) & \mathbf{I}_3 \end{pmatrix} \tag{IV.46}$$

- ${}^{j}\mathbf{O}_j\mathbf{A}_j$ the position of the fairlead connected to the body j with respect to Σ_j;
- ${}^{0}\mathbf{N}_e^j$ given by (IV.5).

$\boldsymbol{\Gamma}^{Cable}$ is the sum of the cable articular loads:

$$\boldsymbol{\Gamma}^{Cable} = \sum_{j\in\mathcal{F}^{Cable}\setminus\{0\}} \boldsymbol{\Gamma}_j^{Cable} \tag{IV.47}$$

where $\boldsymbol{\Gamma}_j^{Cable}$ is the projection of the cable loads acting on the body j into the articular space:

$$\boldsymbol{\Gamma}_j^{Cable} = {}^{j}\mathbf{J}_j^T \, {}^{j}\boldsymbol{\rho}_e \, {}^{e}\mathbf{G}_j^{Cable} \, {}^{e}\mathbf{F}_j^{Cable} \tag{IV.48}$$

where:

- ${}^{j}\boldsymbol{\rho}_e$ is obtained from (IV.8);
- ${}^{j}\mathbf{J}_j$ results of (IV.23).

IV.5.3 Time integration

The time integration of the whole coupling is achieved using a fourth-order explicit Runge-Kutta scheme. The positions and the velocities of the fairleads are updated from (IV.49) and (IV.50).

$${}^{e}\mathbf{OA}_j = {}^{e}\mathbf{P}_j + {}^{e}\mathbf{R}_j \, {}^{j}\mathbf{O}_j\mathbf{A}_j \tag{IV.49}$$

$${}^{e}\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_e) = {}^{e}\mathbf{R}_j \left[{}^{j}\mathbf{v}_j + \mathbf{S}({}^{j}\boldsymbol{\omega}_j) \, {}^{j}\mathbf{O}_j\mathbf{A}_j \right] \tag{IV.50}$$

IV.5.4 Language binding

CableDyn is implemented in Python, so it becomes a new module of the existing coupling between *InWave* and *WS_CN*. Figure IV.15 presents the updated scheme of the coupling.

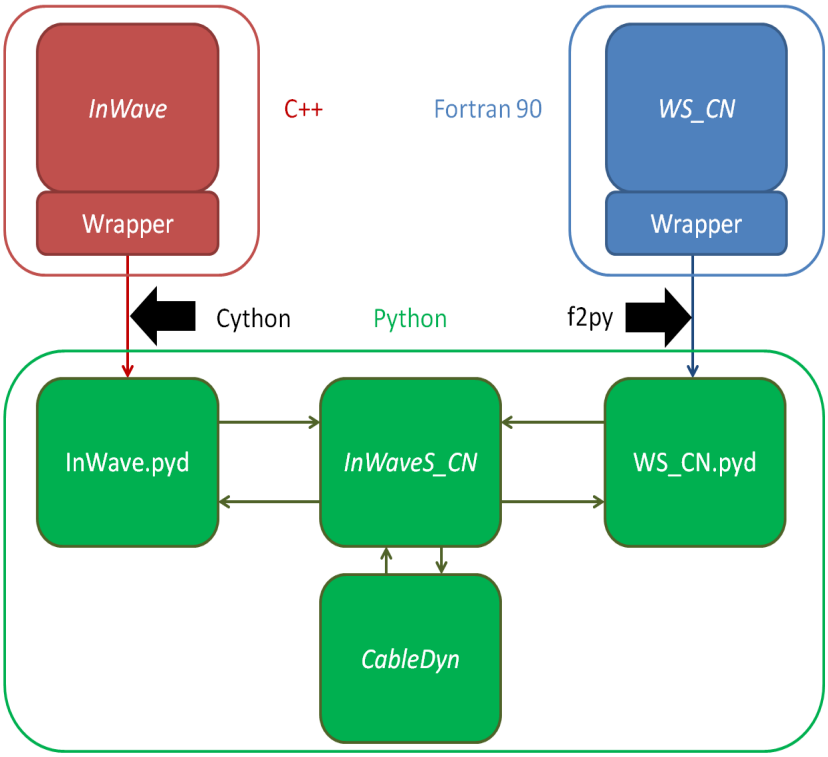


Figure IV.15 – Language binding between *InWave*, *WS_CN* and *CableDyn* using Python as glue code language. The red, green and blue colors represent the C++, Python and Fortran languages.

IV.6 Conclusion

This chapter presented the theory, the implementation and the language binding of a tight coupling between the *CRBA* of *InWave* and the potential flow theory based on the weak-scatterer of *WS_CN*. The Python numerical tool dedicated to this coupling has been named *InWaveS_CN*. A validation of its development was achieved by comparison with *WS_CN* alone and showed a perfect agreement. The tight coupling was also compared to different loose coupling strategies. The implicit loose coupling gave the same results as the tight coupling but required a larger CPU time while the explicit loose coupling presented some differences which could be more important if the case was changed. The tight coupling represents a robust compromise between accuracy and CPU time.

CableDyn, the cable dynamics solver based on the low-order lumped mass theory has been included into *InWaveS_CN* as a new module to perform cable simulations. The theory of this coupling was presented.

The next chapter presents a new strategy for generating meshes in *WS_CN*, which will be used in the last chapter of this PhD work to validate *InWaveS_CN* with experimental data.

Chapter V

Development of a new mesh generator

Contents

The mesh generator of WS_CN, which was implemented before the start of this PhD, presents a lack of robustness when the surface-piercing body is not analytic and, so far, has only been used with academic geometries (sphere, vertical cylinder, cube, Wigley hull). Consequently a new mesh generator is required. This is the aim of this chapter. A state of the art of the mesh generation and the intersection curve tracking is presented. Then, all the steps of the new mesh strategy are detailed. Two validation test cases and a study of the CPU time are exposed.

V.1 The need of a new mesh generator

The mesh generator presented in section III.3 is able to deal with academic surface-piercing bodies (vertical cylinder, sphere, cube, Wigley hull) or immersed bodies. However, it suffers of a lack of robustness for more complex shapes. For example, the simulation of the experiments, detailed in chapter VI, requires to deal with a horizontal floating cylinder and the mesh generator fails to properly mesh it. The problem comes from the intersection curve tracking (cf. subsection III.3.1) which is not robust enough to find the interface in this case. As pointed out by Ko et al. [171], the marching method, as used in *WS_CN*, becomes problematic when the shape of the body is complicated and the configuration of the intersection with the free surface is complex. With a horizontal cylinder, sharp edges are present on the intersection curve and two types of surfaces intersect the free surface: the cylindrical surface and the discs.

Following the observation of the lack of robustness of the present mesh generator, two choices are possible. Either trying to enhance it or launching the development of a new mesh generator which could be more robust, universal and usable for industrial applications. We chose the second option for four reasons:

- The implementation of the initial mesh generator makes the task of enhancement hard and the success uncertain;
- The improvement of the initial mesh generator involves the development of a robust geometric modeller for any floating body, for instance based on NURBS [45]. This requires lots of work and means the development of a computer-aided design tool from scratch, whereas some tools already exist and some of them are free and open-source.
- The initial mesh generator suffers of important memory leaks which deeply affect the performance of the code when long simulations are run;
- The development of a new mesh generator gives us the opportunity to test a new strategy of grid generation.

V.2 Functional specification

The development of a new approach could solve some problems of the initial mesh generator. Its functional specification is:

- *Quality*: the new mesh generator has to create good quality meshes able to be used in an unsteady potential flow based solver;
- *Non-regression*: the new mesh generator has to give the same or better results in terms of stability and accuracy as the initial mesh generator;

- *Robustness*: the new mesh generator has to be able to deal with any floating body with a single intersection curve with the free surface and for any wave model;
- *Modularity*: the new mesh generator has to retain the modularity of the coupling between *InWave* and *WS_ CN*;
- *User-friendliness*: the new mesh generator does not have to require any extra coding when a new geometry is used. This also means the solver *WS_ CN* may be run as a black box by a user who is not a developer.
- *Memory allocation*: the new mesh generator does not have to suffer of memory leaks;
- *Process time*: the new mesh generator does not have to involve an important increase of the CPU-time compared to the initial mesh generator.

V.3 State of the art

A mesh generation algorithm needs to know the shape of the bodies to compute their intersection with waves. This is achieved using an interface tracking method. The body surfaces can be modeled using several formats [172]:

- Analytical expressions of the parametric surfaces when the geometries are simple (cylinder, sphere, cube, axisymmetric geometry, etc.);
- B-splines [171, 173];
- NURBS (Non-uniform rational basis splines), as it is commonly done in the computer-aided design tools [45].

Several methods exist to compute the intersection curve between a body and the free surface, that is to say a surface-to-surface intersection [45, 101, 172, 174]:

- The analytical methods where the exact solution is seeked;
- The lattice methods where the surface-to-surface intersection is transformed into a curve-to-surface intersection;
- The subdivision methods where the problem is decomposed recursively into sub-problems easier to solve;
- The marching methods, which start from a known starting point on the intersection curve and then step along it in a direction prescribed by the curve local geometry.

Once the intersection curve is obtained, the mesh generation follows. Several algorithms are available, for instance:

- The quadtree method where the domain is enclosed into a bounding box which is recursively subdivided into smaller panels by taking into account the presence of the bodies. The positioning of the nodes is not optimal but the method is simple and robust.

- The advance front method where the boundaries (including the intersection curves) are meshed and form the initial front of the mesh. Then, this front is updated by creating new nodes and panels until all the surfaces are covered. The panels have a good shape but the convergence is not always ensured.

- The Delaunay triangulation method creates panels from a given set of nodes such as no point is present in the circumcircle of any triangular panel. This method requires in a first step the creation of the nodes. This approach may be time consuming.

In *WS_CN*, as explained in subsection III.3.1 and subsection III.3.2, a marching method and an advance front method are used, respectively, for the intersection curve tracking and the mesh generation.

The main drawback of this approach (Parametric curves - Intersection curves - Mesh generation) is the mandatory knowledge of a parametric representation of the bodies. In most hydrodynamic solvers (*StarCCM+*[1], *OpenFOAM*[2], *FINE/Marine*[3], *Nemoh*, *WAMIT*, etc.), the mesh of the bodies is an input of the numerical tool, using an external mesh generator (*BlockMesh*[4], *HEXPRESS*[5], *Gmsh*[6], *Rhinoceros 3D*[7], *SALOME*[8], etc.). Then, the mesh is created by the solver. So, another approach is possible if the mesh of the bodies is known initially. The initial mesh is cut at the known free surface and then the the free surface mesh is generated from the intersection curves. This approach is called the panel cutting method and has been implemented at the end of this PhD.

So far, the panel cutting method has been used in two cases:

- The computation of the nonlinear hydrostatic and Froude-Krylov loads;

- The solving of the steady nonlinear wave resistance problem.

Examples of the computation of the nonlinear hydrostatic and Froude-Krylov loads based on a panel cutting method may be found in [31, 175, 176]. The original mesh is recursively subdivided on the intersection curve using a quadtree process. Then the adjacent underwater subpanels are agglomerated to form bigger panels and coarse meshes. Horel et al. [177] and Sengupta et al. [178] applied directly the panel cutting method by clipping the panels of the initial mesh to fit the incoming waves. Lee and Lee [179] used the panel cutting method in case of hydrostatic calculations with flexible structures and non-matching meshes.

Regarding the solution of the steady nonlinear wave resistance problem, this was achieved by Choi et al. [180]. The panels were cut at the real wave elevation and then the mesh generated was used to perform a hydrodynamic computation and not only the hydrostatic and the Froude-Krylov calculations. Nevertheless, due to the steadiness of the problem, an iterative method was used to compute the velocity potential and

[1] https://mdx.plm.automation.siemens.com/star-ccm-plus
[2] https://openfoam.org
[3] https://www.numeca.com/product/finemarine
[4] https://cfd.direct/openfoam/user-guide/v6-blockMesh
[5] https://www.numeca.com/product/hexpress
[6] http://gmsh.info
[7] https://www.rhino3d.com
[8] https://www.salome-platform.org

the wave elevation such as the nonlinear steady free surface boundary conditions were satisfied. Between 5 and 25 iterations are usually necessary in such problems according to Raven [181]. 25 computations represent less than 7 time steps when a fourth-order Runge-Kutta integration scheme is used (with four solvings of the hydrodynamic problem per time step). For comparison, a time-domain method needs several hundreds or even thousands of time steps per numerical simulation. Consequently, the mesh quality required for a time-domain simulation is much higher than for an iterative method. Moreover, Choi et al. [180] used quadrangular panels which were approximately rectilinear with the intersection curve, therefore the panel cutting method did not generate lots of tiny panels which could lead to numerical errors. The panels smaller than 10 % of the smallest panel of the initial mesh were eliminated. Nevertheless, in another configuration, with triangular panels, the number of tiny panels could be much more important and would lead to poor quality meshes. Finally, the panel cutting method has only been used by Choi et al. [180] in case of ships, that is to say with smooth surfaces without sharp edges.

Note

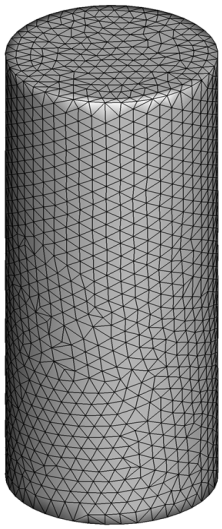
In *CFD* solvers[9], the cutting method is also used and is called the cut-cell method (because volume meshes, made of cells, are required instead of surface meshes, made of panels) [182]. The cells are cut to fit the body geometry. When poorly formed cells are generated, a cell merging method is applied to delete them [183]. As it will be described in subsection V.4.2, this problem is also present with surface meshes.

V.4 Coupling between the panel cutting method and the advance front method

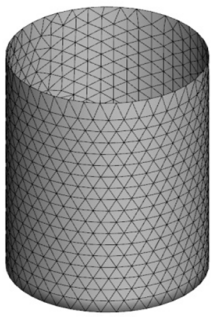
The originality of the proposed method is the generation of meshes adapted for a time-domain unsteady potential flow approach requiring high quality connected nodes surface meshes using triangular panels with possibly sharp edges for thousands of time steps. This method assumes the initial mesh is of best quality. It unfolds in five steps:

- An initial mesh is obtained from an external mesh generator (Figure V.1a);
- The mesh is cut at the incident wave field (Figure V.1b);
- The intersection curve is tracked from the cut mesh (Figure V.1c);
- The free surface and the numerical tank walls are meshed using the same advance front method as in subsection III.3.2 (Figure V.1d);
- The cut mesh is connected with the free surface mesh, leading to the final mesh (Figure V.1e).

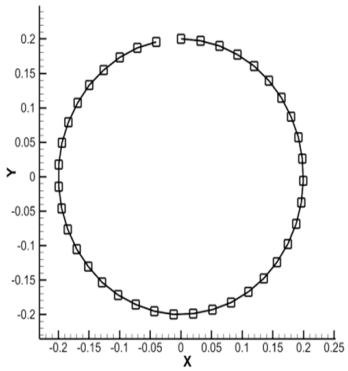
[9] *Computational Fluid Dynamics*, for the simulation of viscous and turbulent flows.



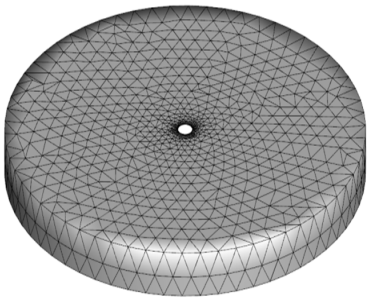
(a) Initial mesh of a cylinder



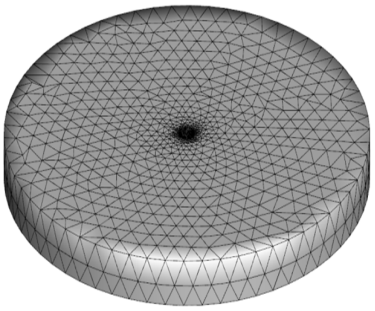
(b) Cut mesh obtained from the panel cutting method



(c) Intersection curve



(d) Mesh of the free surface and the numerical tank walls



(e) Final mesh including the cylinder

Figure V.1 – The steps of the mesh generation of the whole domain with a floating vertical cylinder using the panel cutting method

V.4.1 The mesh clipping

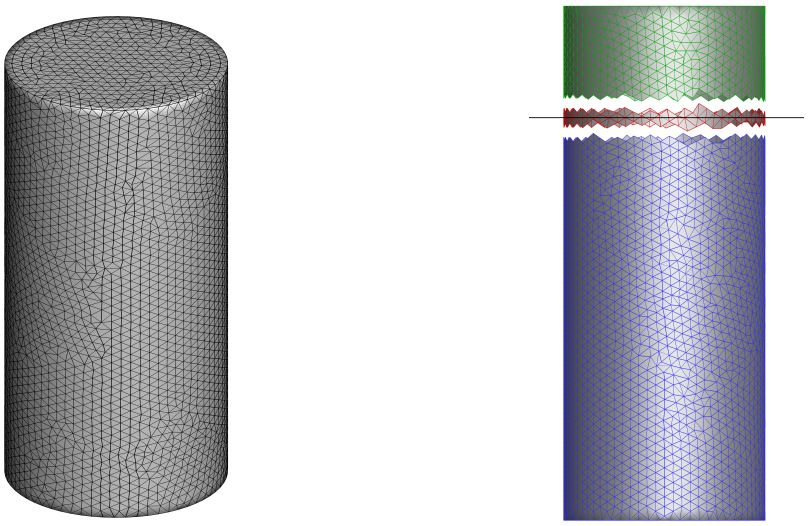
The panel cutting method comes from the numerical tool: *Meshmagick*[10]. This software package has been developed at *Ecole Centrale de Nantes* by François Rongère. It enables the management of surface meshes encountered in the potential flow theory:

- The conversion between major file formats for hydrodynamic computations tools (*Nemoh*, *WAMIT*, *HydroSTAR* or *Diodore*);
- Elementary transformations such as translation, rotation, scaling, symmetry, normals orientation verification, etc.;
- Hydrostatics computations: stiffness matrix, position of the center of buoyancy, displacement, etc.;
- The clipping of the mesh by a **plane**.

[10] https://github.com/LHEEA/meshmagick

This tool is implemented in Python and released under the GPLv3 license.

The mesh clipping starts with a partition of the mesh: for instance in Figure V.2 for a cylinder of radius 0.2 m and height 1 m. The panels which are strictly above the incident wave fields are not taking into account while those strictly below the incident wave fields are automatically kept. Regarding the panels where some vertices are above the incident free surface and some others below, they form the *crown mesh*. The panel cutting method is only applied to this part of the mesh.

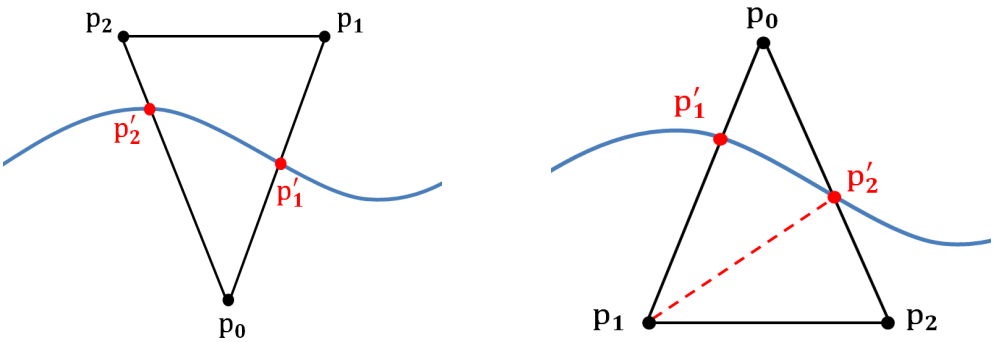


(a) Initial mesh of a cylinder

(b) Partition of the mesh. Blue: lower mesh, green: upper mesh, red: crown mesh. The black line represents the mean sea level. The space between the parts is added for the visualization.

Figure V.2 – Partition of a mesh in *Meshmagick*

Figure V.3 presents the two main cases of panel cutting: when one or two vertices are above the sea level. Other cases appear when one or two nodes are exactly on the intersection curve. As a reminder, *WS_ CN* only deals with triangular panels.



(a) $(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2)$ becomes $(\mathbf{p}_0\mathbf{p}'_1\mathbf{p}'_2)$

(b) $(\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2)$ becomes $(\mathbf{p}_1\mathbf{p}'_2\mathbf{p}'_1)$ and $(\mathbf{p}_1\mathbf{p}_2\mathbf{p}'_2)$

Figure V.3 – Examples of panel clipping

Originally, only the clipping with planes was considered in *Meshmagick*. The trian-

gular panel - plane intersection was figured out analytically. Indeed, the equation of a plane is:

$$\mathbf{x} \cdot \mathbf{n}_{Plane} = c \tag{V.1}$$

where $\mathbf{n}_{Plane}$ and c are the panel normal and constant. For instance for the plane $z = 0$, $\mathbf{n}_{Plane} = {}^{e}\mathbf{z}_{e}$ and $c = 0$.

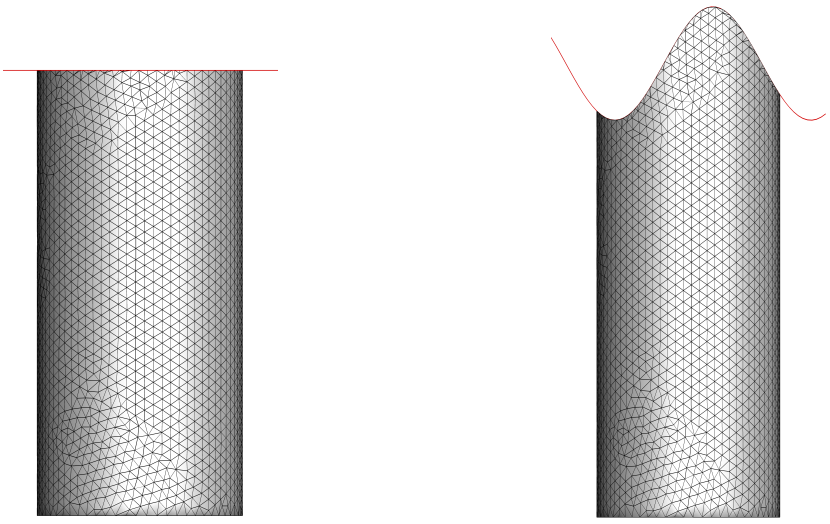
The intersection point $\mathbf{x}$ between the edge $(\mathbf{p}_0\mathbf{p}_1)$ and the plane equation (V.1) satisfies:

$$\mathbf{x} = t\mathbf{p}_0 + (1-t)\mathbf{p}_1 \tag{V.2}$$

with:

$$t = \frac{c - \mathbf{p}_1 \cdot \mathbf{n}_{Plane}}{\mathbf{p}_0 \cdot \mathbf{n}_{Plane} - \mathbf{p}_1 \cdot \mathbf{n}_{Plane}} \tag{V.3}$$

For an arbitrary single-valued incident wave field, a mathematical model is not necessarily available. For the sake of simplicity and robustness, a numerical approach is preferred. A bisection method has been implemented in *Meshmagick* to do so. Thus, if an edge has one node above the free surface and one below, the intersection node is searched iteratively. Examples of clipping are displayed in Figure V.4. The clipping of the present cylinder with a regular wave of amplitude 0.1 m and wave frequency 12 rad/s is used as example in the rest of this section.



(a) Plane $z = 0$

(b) Regular wave of amplitude 0.1 m and wave frequency 12 rad/s

Figure V.4 – Clippings of a vertical cylinder against a plane and a regular wave. The red line represents the analytical incident wave elevation.

The clipping process can lead to large deformations of the panels located close to the intersection curve (Figure V.5). Two problems arise:

- The density of nodes and panels is much higher at the interface than in the rest of the mesh;
- The panel shape is poor.

As a consequence, the mesh cannot be used in a hydrodynamic solver. An extra step of mesh enhancement is mandatory.

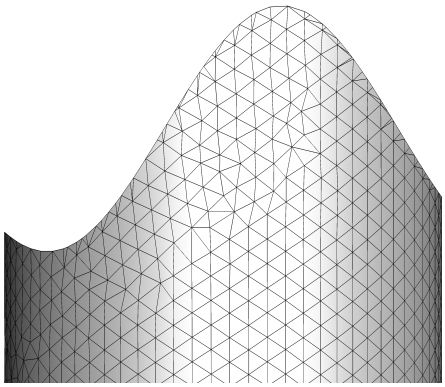


Figure V.5 – Deformed panels close to the intersection curve. Zoom from Figure V.4b.

V.4.2 Panel merging

One can distinguish two types of very deformed panels:

- The *vertical tiny triangles* where two vertices are close to each other on the intersection curve;
- The *horizontal tiny triangles* where two vertices are on the intersection curve and the third one is strictly in the mesh but close to the interface.

The following figure displays deformed panels from Figure V.5.

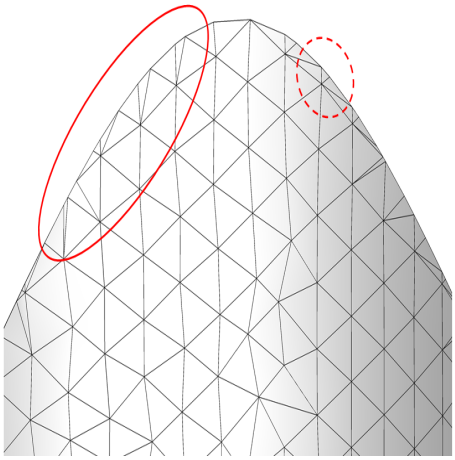
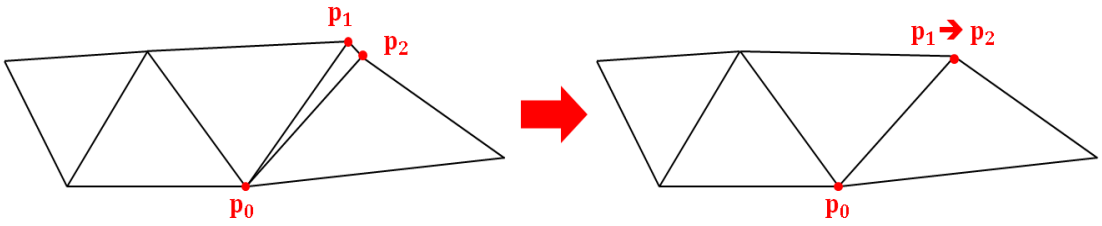


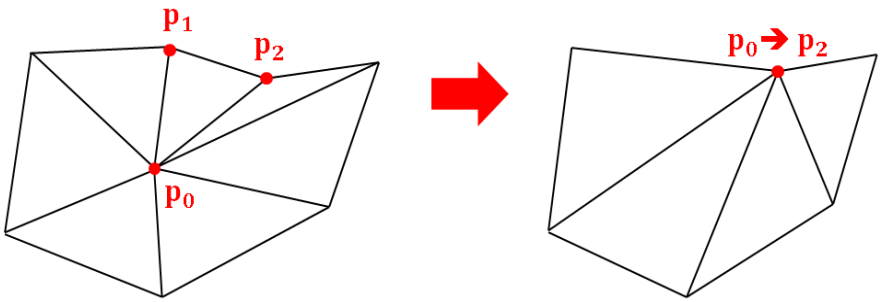
Figure V.6 – Examples of vertical (red solid line) and horizontal (red dashed line) tiny triangles with $\alpha_H = \beta_H = 0.3$, $\alpha_V = 0.4$ and $\beta_V = 0.3$

By deleting these tiny panels, the density of nodes and panels on the intersection curve decreases. This solves one of the two mesh problems presented in the subsection V.4.1. Regarding the vertical tiny triangles, the two nodes on the intersection curve are merged and located at the position of one of these nodes, as shown in Figure V.7a. This leads to the deletion of one node and one panel. For the horizontal tiny triangles, the single node below the waterline is merged with one node on the intersection curve, the second node on the intersection curve being also deleted, as displayed in

Figure V.7b. This involves the deletion of at least two nodes and one panel, depending on the connectivities between the nodes.



(a) Deletion of a vertical tiny triangle



(b) Deletion of a horizontal tiny triangle

Figure V.7 – Schemes of the panel merging algorithms

To preserve the geometry of the mesh, these two algorithms are applied conditionally. Two geometric criteria are defined per algorithm: the first one about the panel area, the second one about a characteristic distance. These criteria are detailed in Table V.1.

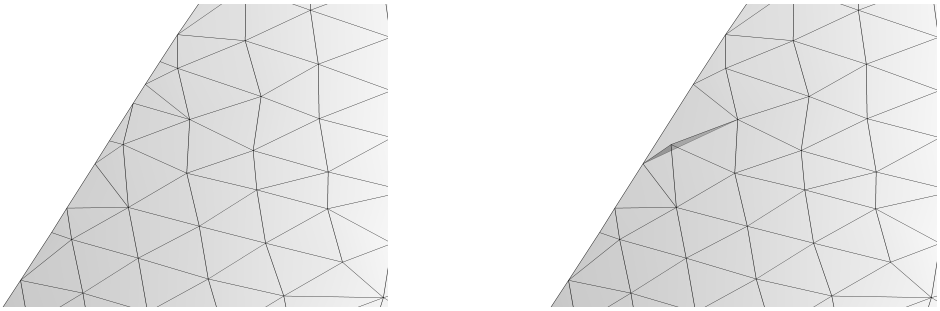
Criterion	Vertical	Horizontal
Area	$\mathcal{A} \leqslant \alpha_V \mathcal{A}_{mean}$	$\mathcal{A} \leqslant \alpha_H \mathcal{A}_{mean}$
Distance	$\|\mathbf{p}_1\mathbf{p}_2\| \leqslant \beta_V \ell_{mean}$	$\left\|z - \eta^I\right\| \leqslant \beta_H \ell_{mean}$

Table V.1 – Geometric criteria to apply the panel merging

where:

- $\mathcal{A}$ is the panel area;
- $\mathcal{A}_{mean}$ denotes the mean of the panel areas in the initial mesh;
- ℓ_{mean} represents the mean of the edge lengths in the initial mesh;
- $\mathbf{p}_1$ and $\mathbf{p}_2$ are defined in Figure V.7a
- z is the vertical coordinate of the node $\mathbf{p}_0$ in Figure V.7b;
- η^I means the incident wave elevation at the vertical of the node $\mathbf{p}_0$ in Figure V.7b;
- α_V, α_H, β_V and β_H are the panel merging coefficients. The subscript indicates if the coefficient is used for the vertical (V) or horizontal (H) tiny panels.

The higher these coefficients are, the more important the number of deleted panels is, **but**, the more important the risk of interpenetration of neighboring panels is. An example of a panel overlapping is shown in Figure V.8.



(a) $\alpha_H = \beta_H = 0.3$, $\alpha_V = 0.4$ and $\beta_V = 0.3$ **(b)** $\alpha_H = \beta_H = \alpha_V = \beta_V = 0.5$. The dark area indicates the panel overlapping.

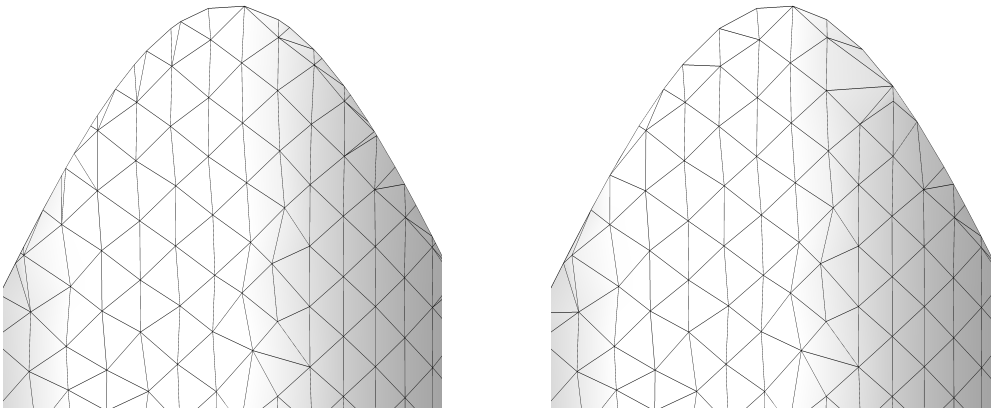
Figure V.8 – Example of a panel overlapping. The wave height is 0.15 m.

The values of α_V, α_H, β_V and β_H are chosen empirically. An example of a set of coefficients is given below:

$$\begin{cases} \alpha_V = 0.4 \\ \alpha_H = 0.1 \\ \beta_V = 0.3 \\ \beta_H = 0.1 \end{cases} \tag{V.4}$$

A special attention is paid to the nodes along the sharp edges of the mesh. They are not moved. The sharp edges are tracked using the discontinuity of the panel normals.

Figure V.9 shows the application of the panel merging algorithm. The density of nodes and panels is reduced on the intersection curve as wanted, but the panel shape is still not good enough. This problem is addressed in the next section.



(a) Mesh before the panel merging **(b)** Mesh after the panel merging

Figure V.9 – Example of the application of the panel merging with $\alpha_H = \beta_H = 0.3$, $\alpha_V = 0.4$ and $\beta_V = 0.3$

V.4.3 The spring analogy method

The panel merging method reduces the density of nodes and panels close to the intersection curve but the node positions and the panel shapes are still not good enough. To improve them, an unstructured mesh smoothing algorithm based on the spring analogy is used. As explained in subsection III.4.1, it is based on a vertex spring analogy.

The expression of the spring loads is:

$$\mathbf{F}_i = \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{x}_j - \mathbf{x}_i) \tag{V.5}$$

As we want the small panels to get larger and the large panels to get smaller, all the panels have to be able to be deformed. So, the stiffness is taken constant. Its value has no influence so it is chosen as unity:

$$k_{ij} = 1 \tag{V.6}$$

The same notations as in subsection III.4.1 are used. The position is decomposed into three components:

$$\mathbf{x}_i = x_u^i \mathbf{u}_i + x_v^i \mathbf{v}_i + x_w^i \mathbf{n}_i \tag{V.7}$$

To ensure the nodes stay on the body surfaces, the normal displacement is zeroed:

$$(\mathbf{x}_i - \mathbf{x}_i^{old}) \cdot \mathbf{n}_i = 0 \tag{V.8}$$

On the intersection curve or at the sharp edges in the body mesh, a node on this intersection has to remain on it. Thus, $\mathbf{n}_i$ is normal to one of the surfaces, $\mathbf{u}_i$ is along the intersection line and $\mathbf{v}_i$ is chosen such as the local basis is orthonormal and direct. In that case, the displacement along $\mathbf{v}_i$ is zeroed:

$$(\mathbf{x}_i - \mathbf{x}_i^{old}) \cdot \mathbf{v}_i = 0 \tag{V.9}$$

If three surfaces intersect each other at the same node i, then the displacement along $\mathbf{u}_i$ is also zeroed:

$$(\mathbf{x}_i - \mathbf{x}_i^{old}) \cdot \mathbf{u}_i = 0 \tag{V.10}$$

In case of a node on a smooth surface, the final system of equations to solve yields:

$$\begin{cases} \mathbf{F}_i \cdot \mathbf{u}_i = 0 \\ \mathbf{F}_i \cdot \mathbf{v}_i = 0 \\ (\mathbf{x}_i - \mathbf{x}_i^{old}) \cdot \mathbf{n}_i = 0 \end{cases} \tag{V.11}$$

And after some mathematical developments and by including the condition on the normal displacement, the system of three equations is reduced to a bi-dimensional problem:

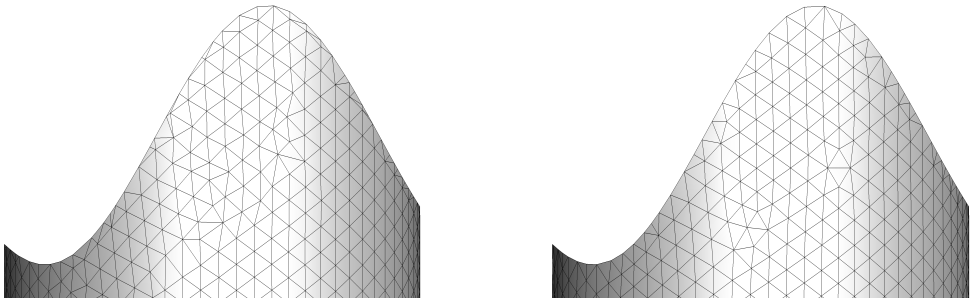
$$\begin{cases} \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_j \cdot \mathbf{u}_i)x_u^j - \left[\sum_{j=1}^{N_v^i} k_{ij}\right] x_u^i + \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{v}_j \cdot \mathbf{u}_i)x_v^j = -\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_i \cdot \mathbf{n}_j)(\mathbf{x}_j^{old} \cdot \mathbf{n}_j) \\ \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{u}_j \cdot \mathbf{v}_i)x_u^j + \sum_{j=1}^{N_v^i} k_{ij}(\mathbf{v}_j \cdot \mathbf{v}_i)x_v^j - \left[\sum_{j=1}^{N_v^i} k_{ij}\right] x_v^i = -\sum_{j=1}^{N_v^i} k_{ij}(\mathbf{v}_i \cdot \mathbf{n}_j)(\mathbf{x}_j^{old} \cdot \mathbf{n}_j) \end{cases} \tag{V.12}$$

Equation V.12 leads to a sparse linear system of size twice the number of nodes in the body mesh such as $\mathbf{AX} = \mathbf{B}$ with:

$$\mathbf{X} = \begin{pmatrix} \vdots \\ x_u^i \\ x_v^i \\ \vdots \end{pmatrix} \tag{V.13}$$

The solution is obtained using the same sparse linear system solver *PARDISO* [184] as in subsection III.4.1.

Figure V.10 presents the application of the spring analogy method once the mesh clipping and the panel merging are applied. The panel shape is improved on the intersection curve as expected. This mesh is now good enough to be used in the fluid solver.



(a) Mesh after the panel merging **(b)** Mesh after the spring analogy method

Figure V.10 – Example of the application of the vertex spring analogy method

The use of the spring analogy method faintly deforms the geometry of the body. To avoid this numerical error, the algorithm is not applied to the full body mesh but only to a part of the mesh close to the intersection curve. The nodes below this part are considered as fixed and the three displacement conditions (V.8), (V.9) and (V.10) are assumed satisfied. The geometry is preserved and the number of non-zero coefficients in the linear system is reduced, involving a CPU time reduction.

V.4.4 Multiple node tracking

In the previous section, it was assumed multiple nodes were known. Therefore, the boundary conditions (V.9) and (V.10) are directly applied at the proper nodes. But, when an initial mesh is used, it only includes a set of nodes and a table of connectivities. Consequently, multiple nodes need to be automatically identified.

The multiple node tracking algorithm unfolds in three steps:

– The detection of multiple nodes is achieved from the discontinuity of the outward panel normals in the body mesh (at this stage, the free surface mesh is not generated yet). A node is multiple if and only if it exists at least two neighboring panels of normal vectors $\mathbf{u}$ and $\mathbf{v}$ such as:

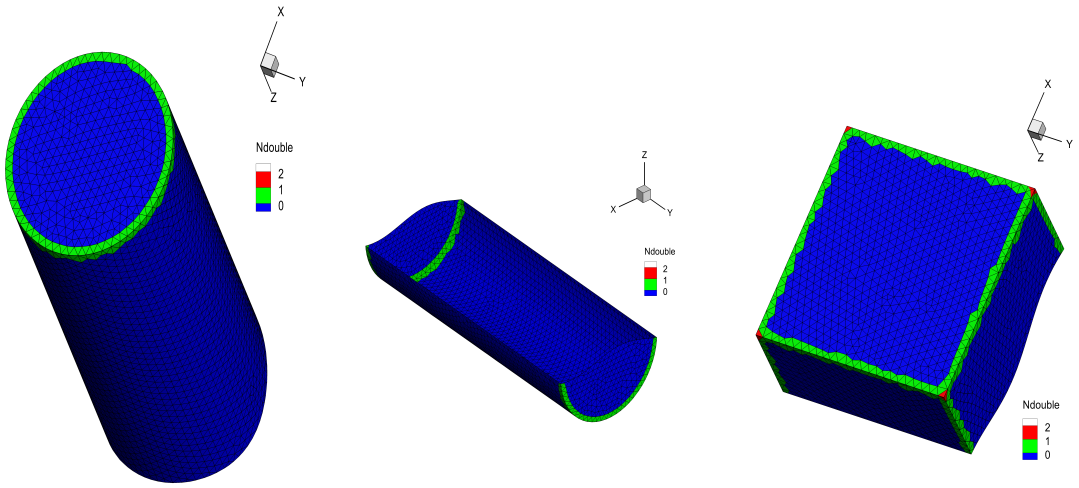
$$\mathbf{u} \cdot \mathbf{v} < \epsilon \tag{V.14}$$

ϵ is a constant, arbitrary defined as equal to $\cos(20°)$. Consequently, an angle of 20° between two neighboring normal vectors leads to a multiple node.

- Then, the order of multiplicity of the node needs to be figured out. It is necessary to know if the node is double, triple, quadruple, etc. The following rule is applied:
  - If it exists a neighboring normal vector $\mathbf{w}$ such as $\begin{cases} \mathbf{u} \cdot \mathbf{w} < \epsilon \\ \mathbf{v} \cdot \mathbf{w} < \epsilon \end{cases}$, then the node is *at least* triple, otherwise it is double;
  - If it exists a neighboring normal vector $\mathbf{x}$ such as $\begin{cases} \mathbf{u} \cdot \mathbf{x} < \epsilon \\ \mathbf{v} \cdot \mathbf{x} < \epsilon \\ \mathbf{w} \cdot \mathbf{x} < \epsilon \end{cases}$, then the node is *at least* quadruple, otherwise it is triple;
  - Etc.
- Once a multiple node is found with its order of multiplicity, new nodes are created at the same position. Therefore, each multiple node is made of several *elementary* nodes: two for a double node, three for a triple node, four for a quadruple node, etc. The table of connectivities needs to be updated to associate each elementary node to a surface. Each neighboring panel of normal vector $\mathbf{n}$ is connected to the correct elementary node from the following rule (here for a quadruple node) using $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ previously defined:
  - If $\begin{cases} \mathbf{u} \cdot \mathbf{n} > \epsilon \\ \mathbf{v} \cdot \mathbf{n} < \epsilon \\ \mathbf{w} \cdot \mathbf{n} < \epsilon \end{cases}$, then the neighboring panel belongs to the same surface as $\mathbf{u}$;
  - If $\begin{cases} \mathbf{u} \cdot \mathbf{n} < \epsilon \\ \mathbf{v} \cdot \mathbf{n} > \epsilon \\ \mathbf{w} \cdot \mathbf{n} < \epsilon \end{cases}$, then the neighboring panel belongs to the same surface as $\mathbf{v}$;
  - If $\begin{cases} \mathbf{u} \cdot \mathbf{n} < \epsilon \\ \mathbf{v} \cdot \mathbf{n} < \epsilon \\ \mathbf{w} \cdot \mathbf{n} > \epsilon \end{cases}$, then the neighboring panel belongs to the same surface as $\mathbf{w}$;
  - Otherwise, $\mathbf{n}$ belongs to the fourth surface.

Only double, triple and quadruple nodes are tracked.

Examples of multiple nodes tracking are shown in Figure V.11. Multiple nodes are properly found.



(a) Vertical cylinder **(b)** Horizontal cylinder **(c)** Cube

Figure V.11 – Examples of multiple node tracking. The panels including a simple, double or triple node are, respectively, in blue, green and red.

V.4.5 Mesh qualities

To evaluate the mesh qualities, algebraic metrics f are used. They have the following properties [185, 186]:

- f is dimensionless;
- f is referenced to an ideal element, here an equilateral triangle of edge length ℓ;
- $f \in [\![0\ ;\ 1]\!]$ with $f = 1$ if and only if the panel matches an equilateral triangle of edge length ℓ and $f = 0$ if and only if the triangle is degenerated;
- f is invariant by translation;
- f does not depend on the orientation of the panel (except for the orientation metrics which are not studied here);
- f is a function of the node positions.

Three metrics are used:

- The relative size metric f_{size} which detects triangles which are unusually large or small relative to an equilateral triangle of reference. Its definition is:

$$f_{size} = min\left(\tau, \frac{1}{\tau}\right) \tag{V.15}$$

with τ the ratio of the panel area $\mathcal{A}$ to the area of an equilateral triangle of edge length ℓ:

$$\tau = \frac{\mathcal{A}}{\frac{\sqrt{3}}{4}\ell^2} \tag{V.16}$$

$f_{size} = 1$ if and only if the triangle has the same area as an equilateral triangle of edge length ℓ and $f_{size} = 0$ if the triangle is degenerate.

– The shape metric f_{shape} which detects distortions in the shape of a triangle, independently of its size, compared to an equilateral triangle. Its definition is:

$$f_{shape} = \frac{2\sqrt{3}\mathcal{A}}{\mathbf{L}_1 \cdot \mathbf{L}_1 + \mathbf{L}_1 \cdot \mathbf{L}_2 + \mathbf{L}_2 \cdot \mathbf{L}_2} \tag{V.17}$$

with $\mathbf{L}_1 = \mathbf{p}_0 - \mathbf{p}_2$ and $\mathbf{L}_2 = \mathbf{p}_1 - \mathbf{p}_0$, by keeping the notations of Figure V.3. $f_{shape} = 1$ if and only if the triangle is equilateral and $f_{shape} = 0$ if the triangle is degenerate.

– The size-shape metric $f_{size-shape}$ that measures both size and shape simultaneously:

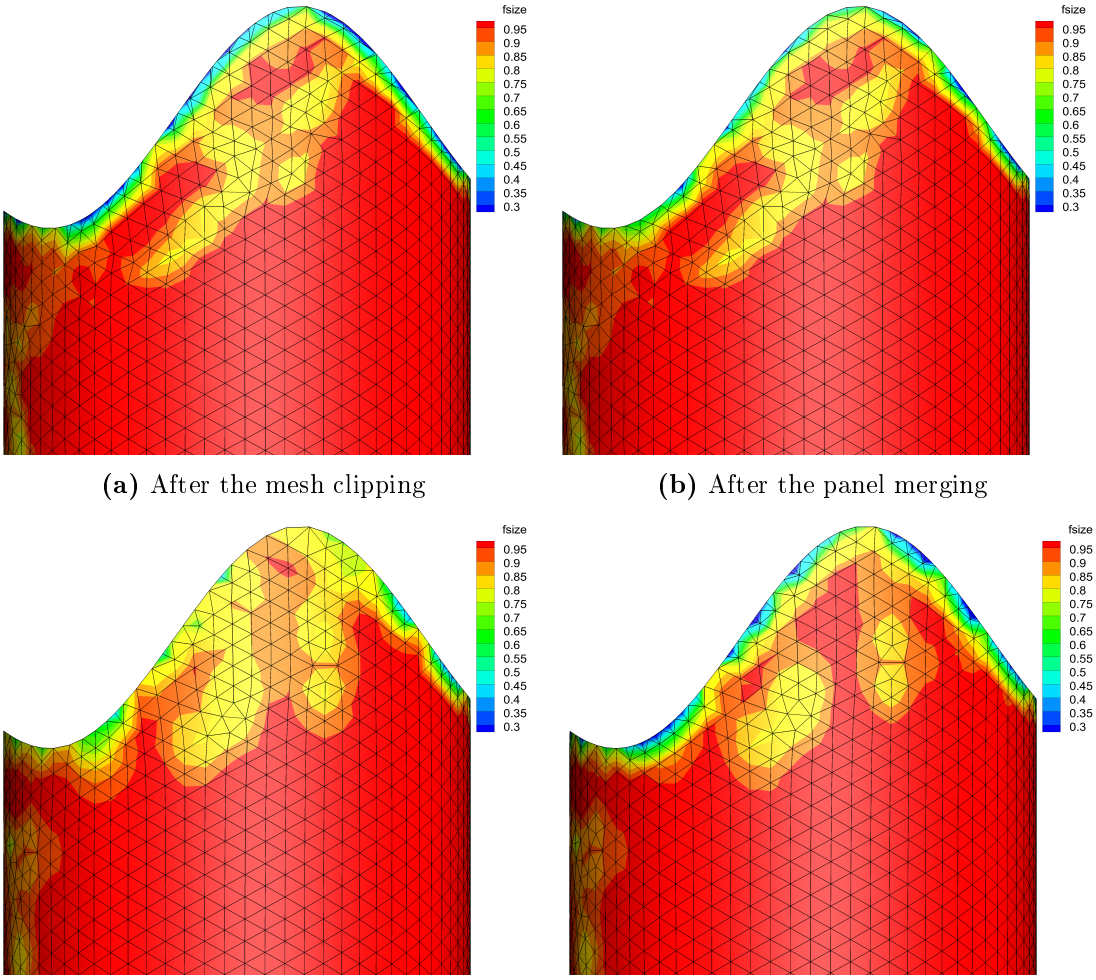
$$f_{size-shape} = f_{size} f_{shape} \tag{V.18}$$

$f_{shape} = 1$ if and only if the triangle is equilateral with an area equal to $\frac{\sqrt{3}}{4}\ell^2$ and $f_{shape} = 0$ if the triangle is degenerate.

The area of a panel is given by:

$$\mathcal{A} = \frac{1}{2}\|\mathbf{L}_1 \times \mathbf{L}_2\| \tag{V.19}$$

Figures V.12, V.13 and V.14 show the evolution of f_{size}, f_{shape} and $f_{size-shape}$ during the mesh generation. In this cases, $\ell = 0.02\,\text{m}$. The use of both the panel merging and the vertex spring analogy method improves significantly the quality of the mesh, especially on the intersection curve. If the mesh optimization algorithm is not applied, the mesh stays of poor quality.



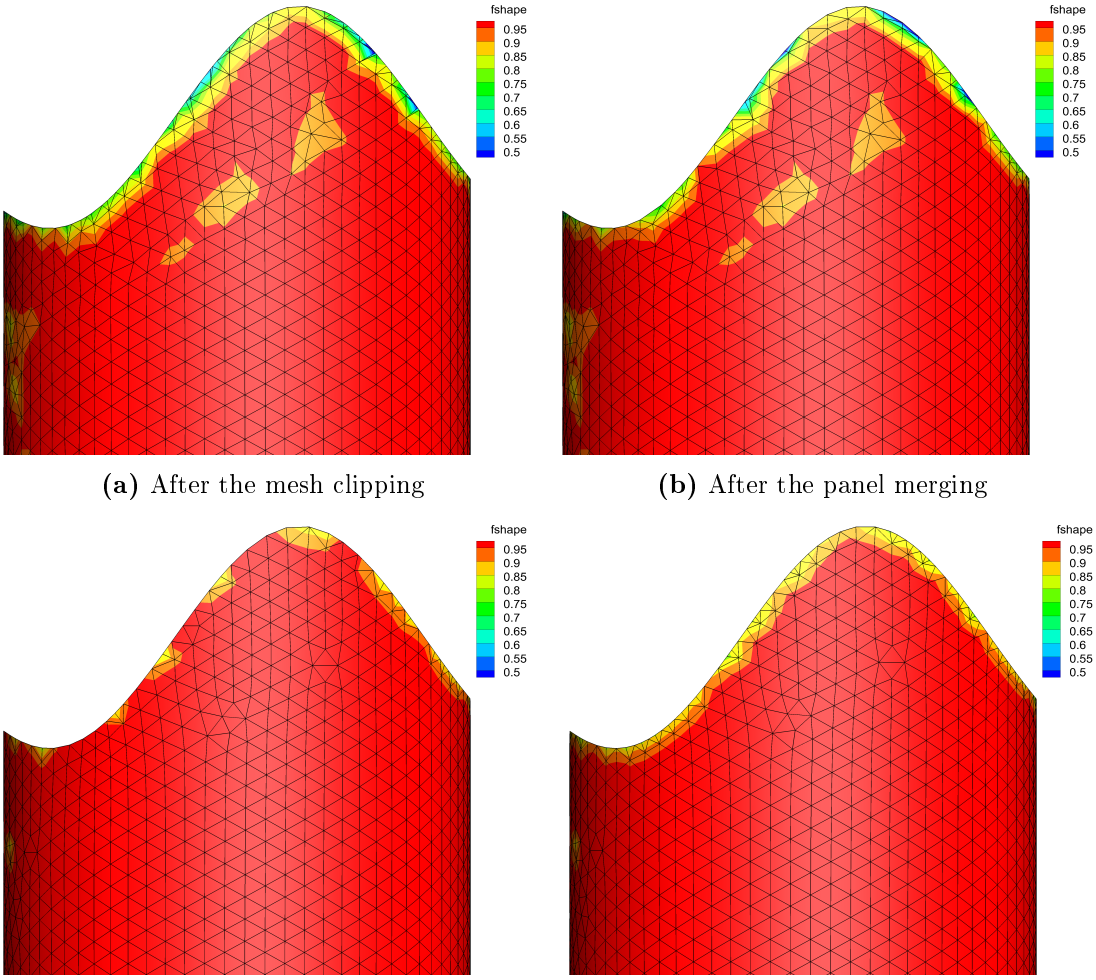
(a) After the mesh clipping

(b) After the panel merging

(c) After the spring analogy method

(d) After the spring analogy method without using the panel merging

Figure V.12 – Evolution of f_{size} during the mesh generation



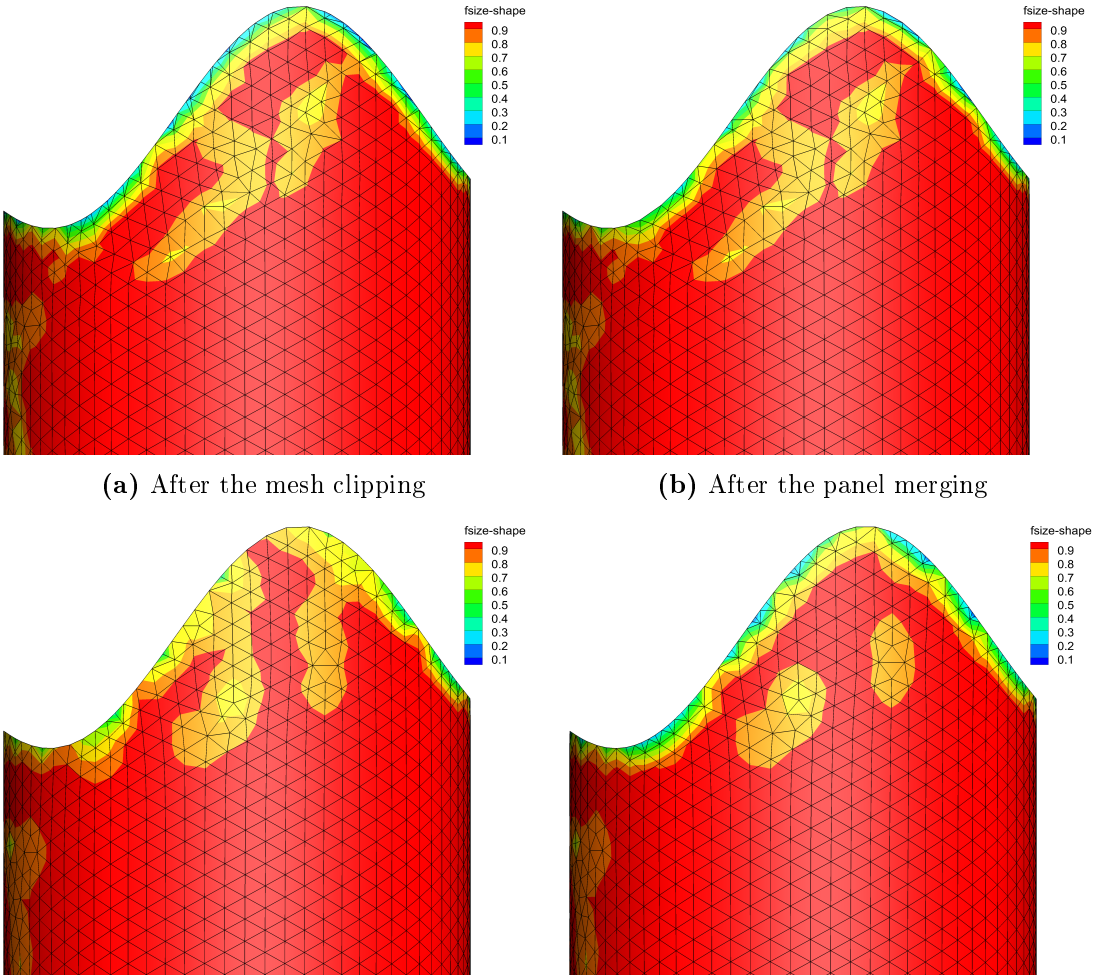
(a) After the mesh clipping

(b) After the panel merging

(c) After the spring analogy method

(d) After the spring analogy method without using the panel merging

Figure V.13 – Evolution of f_{shape} during the mesh generation



(a) After the mesh clipping

(b) After the panel merging

(c) After the spring analogy method

(d) After the spring analogy method without using the panel merging

Figure V.14 – Evolution of $f_{size-shape}$ during the mesh generation

V.4.6 Intersection curve tracking and free surface mesh generation

Once the final body mesh is obtained, the intersection curve is tracked from the nodes which satisfies $z = \eta^I$. But, doing so, the list of nodes does not form an oriented curve which is mandatory to generate the free surface mesh. Consequently, the nodes on the intersection curve have to be sorted. The process unfolds in three steps.

- An initial node $\mathbf{x}_{init}$ is randomly chosen on the intersection curve. This point has exactly two neighbors, $\mathbf{x}_a$ and $\mathbf{x}_b$. The curve orientation demands to define a predecessor, $\mathbf{x}_{pred}$, and a successor, $\mathbf{x}_{suc}$. For instance, let us assume $\mathbf{x}_{pred} = \mathbf{x}_a$. $\mathbf{x}_b$ becomes the successor of $\mathbf{x}_{init}$, so its ancestor is known.
- For each node, the ancestor is known, so there is only one possibility for the successor. Thus, the curve is built step by step along the succeeding nodes.
- When the successor matches the initial point, the curve is defined and oriented.

When the intersection curve is found, the free surface mesh is generated using the advance front method presented in subsection III.3.2. Then, the body mesh and the

free surface mesh are connected. An example of an intersection curve and a final mesh are displayed in Figures V.15 and V.16. The same vertical cylinder is used as in the previous section. The amplitude of the regular wave is 0.02 m and its wave frequency 12 rad/s. Other final meshes for two academic geometries are shown: a horizontal cylinder in Figure V.17 and a cube in Figure V.18. Three non-academic geometries are also displayed:

- The wave energy converter: *SEAREV*[11];
- A *FPSO*;
- A tension-leg platform (*TLP*).

These three geometries are presented in Figure V.19. Their incorporation in a free surface mesh is displayed in Figure V.20 for the SEAREV, in Figure V.21 for the FPSO and in Figure V.22 for the TLP.

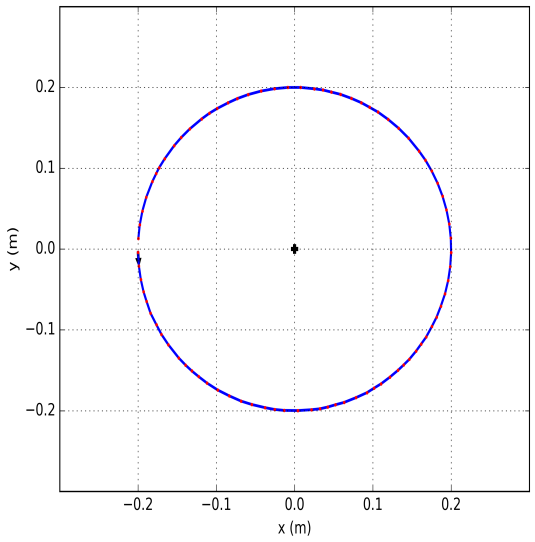
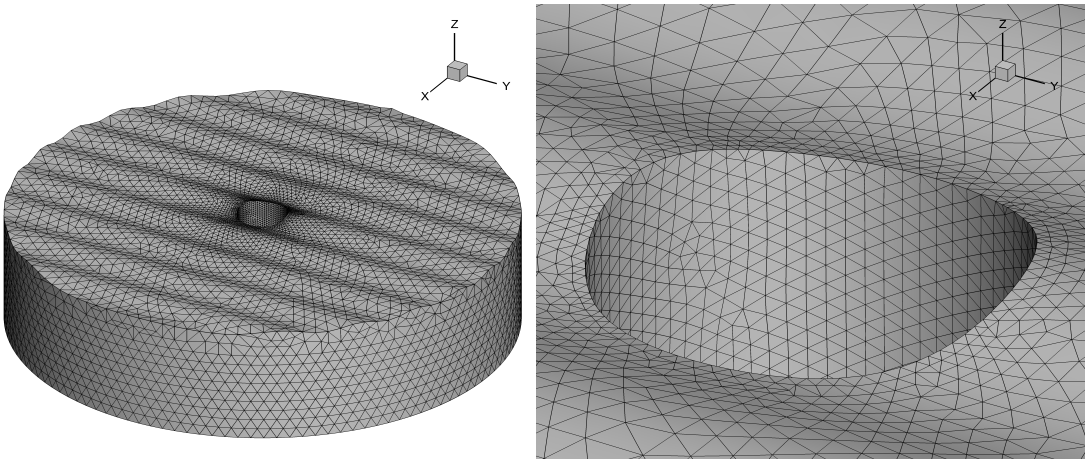


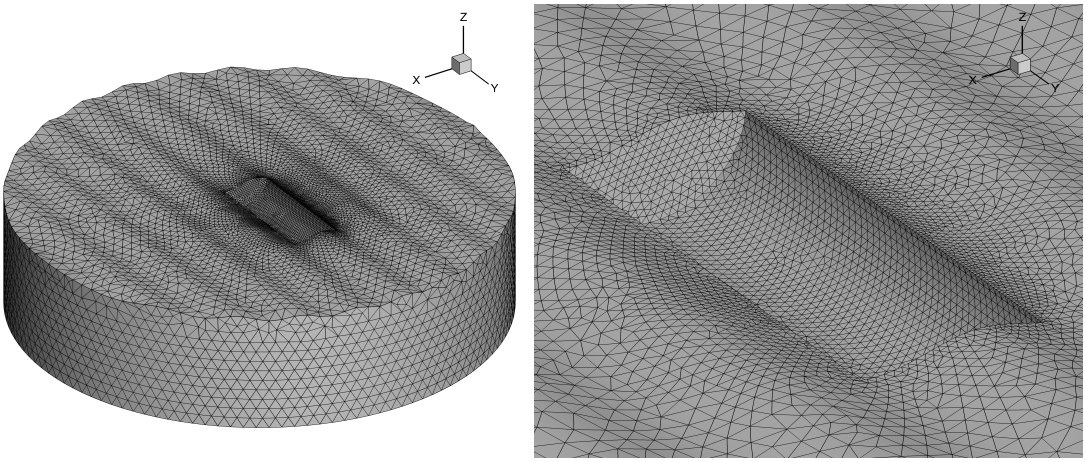
Figure V.15 – Example of an intersection curve for a vertical cylinder. The red points are the nodes, the blue line is the intersection curve, the black arrow represents the orientation of the curve, the black cross denotes the gravity center of the water-plane area.

[11] *Système Électrique Autonome de Récupération de l'Energie des Vagues*



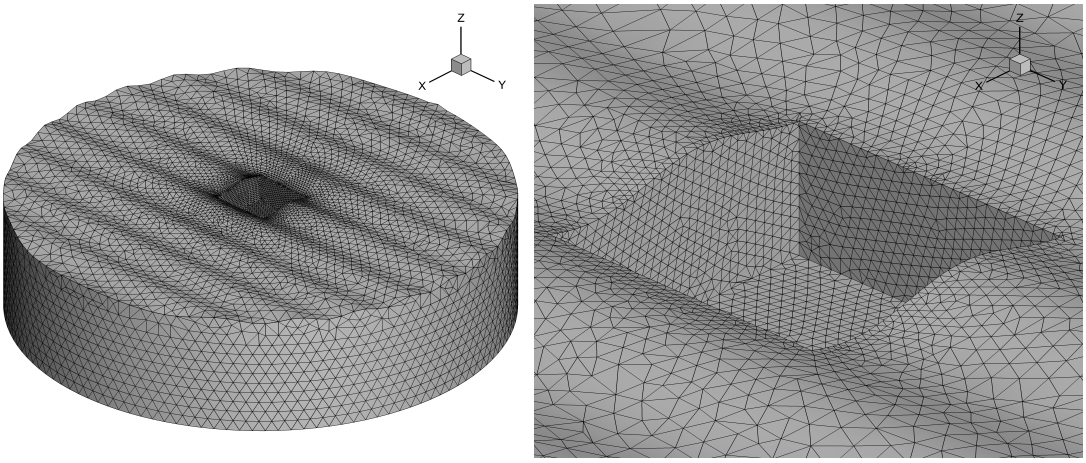
(a) Whole mesh **(b)** Zoom on the intersection curve

Figure V.16 – Vertical cylinder



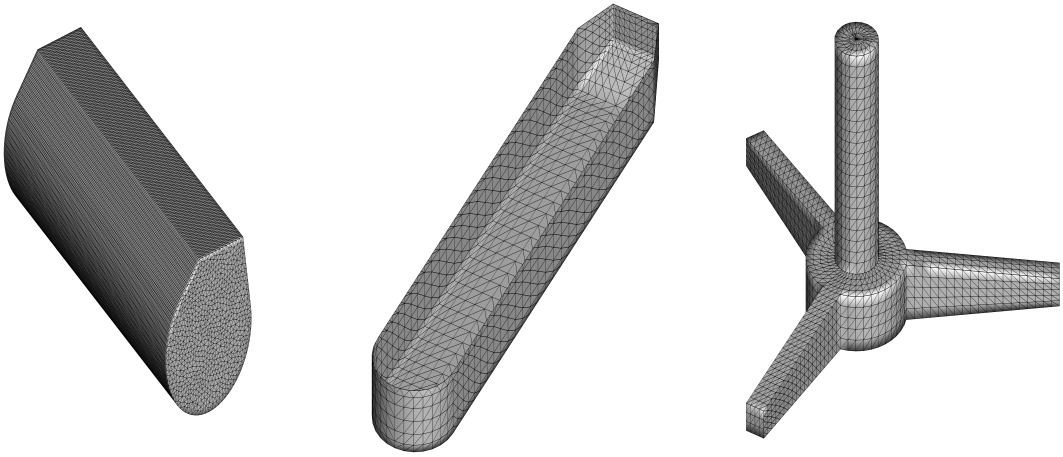
(a) Whole mesh **(b)** Zoom on the intersection curve

Figure V.17 – Horizontal cylinder



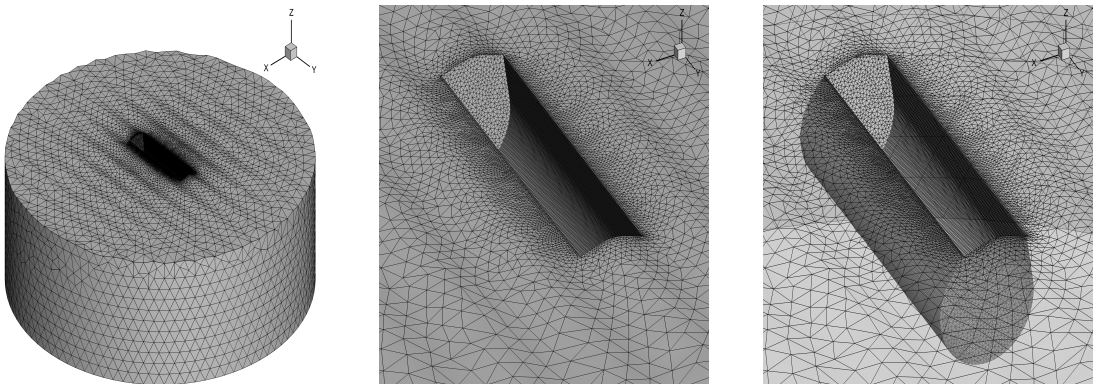
(a) Whole mesh **(b)** Zoom on the intersection curve

Figure V.18 – Cube



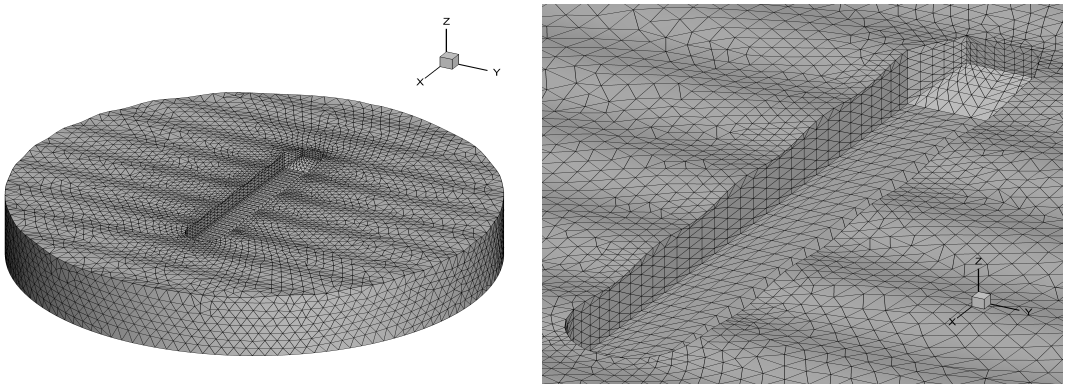
(a) SEAREV **(b)** FPSO **(c)** TLP

Figure V.19 – Non-academic initial meshes



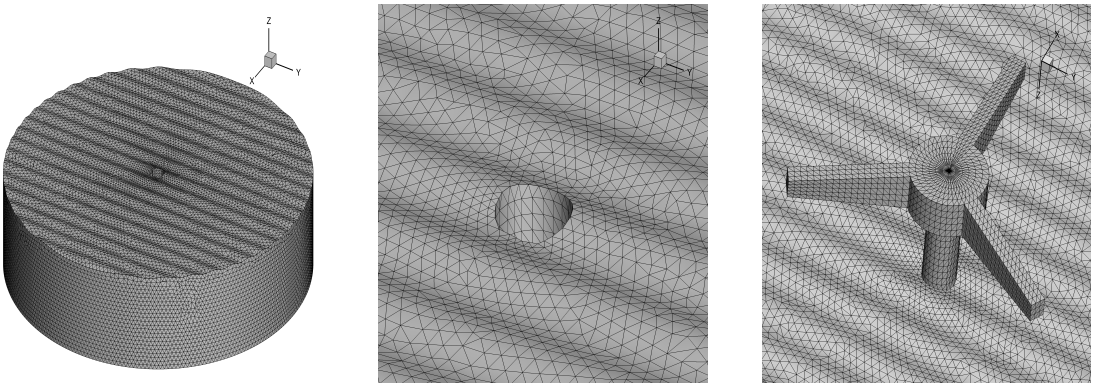
(a) Whole mesh **(b)** Zoom on the intersection curve **(c)** Zoom on the intersection curve with transparency

Figure V.20 – SEAREV



(a) Whole mesh **(b)** Zoom on the intersection curve

Figure V.21 – FPSO



(a) Whole mesh **(b)** Zoom on the intersection curve **(c)** View from below

Figure V.22 – TLP

V.5 Language binding

Meshmagick is implemented in Python, therefore this code becomes a new module of the existing coupling between *InWave*, *WS_CN* and *CableDyn*. Figure V.23 displays the updated scheme of the coupling. **This is the final state of the numerical tool developed during this PhD work**. In the rest of the PhD thesis, when *Meshmagick* is used, the mechanical solver is always *InWave*.

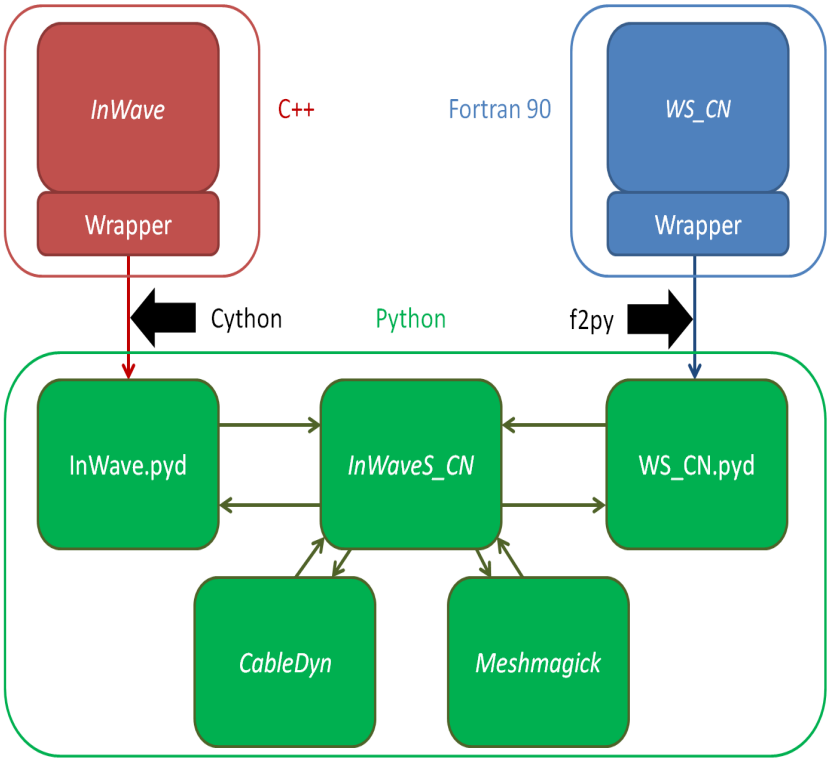


Figure V.23 – Language binding between *InWave*, *WS_CN*, *CableDyn* and *Meshmagick* using Python as glue code language. The red, green and blue colors represent the C++, Python and Fortran languages

Table V.2 shows the distribution of each step of the new mesh generator amongst the modules and the languages.

Module	*Meshmagick*	*InWaveS_CN*	*WS_CN*
Language	Python	Python	Fortran
Mesh clipping	X		
Panel merging	X		
Multiple node tracking			X
Spring analogy method			X
Intersection curve tracking		X	X
Free surface mesh generation			X
Final mesh			X

Table V.2 – Distribution of the steps of the new mesh generator amongst the modules and the languages

V.6 Profiling and CPU time comparison

The CPU time used by the initial mesh generators is presented in Table V.3, while the results for the new mesh generator are displayed in Table V.4. The same test case is run to obtain these results and leads to a mesh of roughly 9500 panels with a single surface-piercing body. As expected, the time necessary to create the free surface and numerical walls meshes is almost constant with the two methods. The time to track the intersection curve is always small. The use of the new mesh generator involves an increase of 62 % of the CPU time. Nevertheless, it stays in the same order of magnitude and small compared to the time of a whole *WS_CN* simulation.

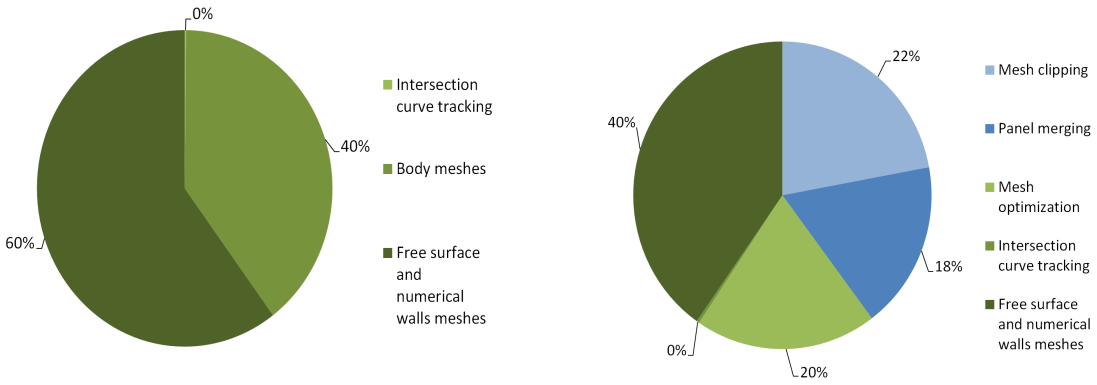
Task	CPU time (s)
Intersection curve tracking	0.003
Body meshes	0.771
Free surface and numerical walls meshes	1.170
Total	1.944

Table V.3 – CPU time for every main task of the initial mesh generator

The profiling of the two mesh generators is shown in Figure V.24. 40 % of CPU time to create a whole mesh based on the new mesh strategy comes from *Meshmagick*, that is to say the Python scripts (in blue on the pie chart). As Python is slower than Fortran, the time required for the mesh clipping and the panel merging could be significantly reduced by a translation of the Python scripts into Fortran.

Task	CPU time (s)
Mesh clipping	0.696
Panel merging	0.558
Mesh optimization	0.621
Intersection curve tracking	0.009
Free surface and numerical walls meshes	1.270
Total	3.154

Table V.4 – CPU time for every main task of the new mesh generator



(a) Initial mesh generator **(b)** New mesh generator

Figure V.24 – Profiling of the new mesh generator. The colors follow the rule defined in Figure V.23: green for Python, blue for Fortran.

Note

If the sparsity of the linear system obtained in the spring analogy method is not used (i.e. *PARDISO* is not used) and if each node is considered with three degrees of freedom instead of two (tri-dimensional problem instead of bi-dimensional), then it involves an increase of the mesh optimization part of 1270 %, or 8.513 s instead of 0.621 s. The method becomes too long to be applied efficiently. The profiling is updated in V.25.

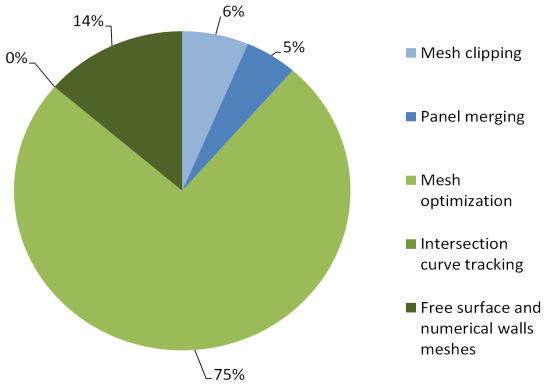


Figure V.25 – Profiling of the new mesh generator without using *PARDISO*. The colors follow the rule defined in Figure V.23: green for Python, blue for Fortran.

V.7 Validation

The initial and new mesh generators are compared using validation test cases that both of them can run. A vertical cylinder of radius 0.2 m, length 0.2 m and draft 0.1 m is considered. Two test cases are chosen:

- A free motion in heave from the equilibrium position;
- A free motion in pitch around the geometrical center of the cylinder with an initial angle of 4°.

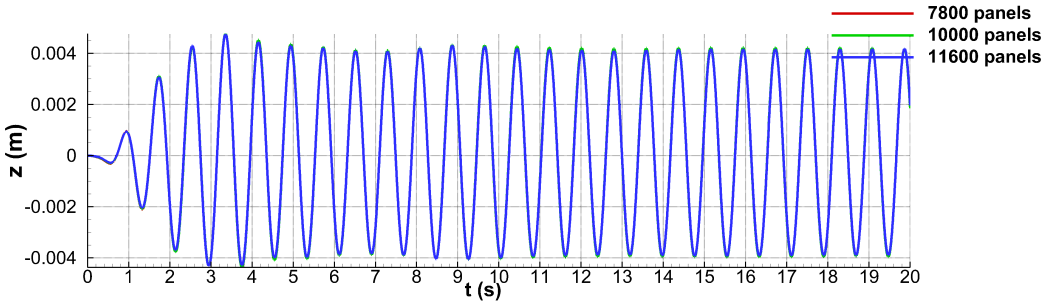
The cylinder mass is 12.88 kg and its inertia in pitch 0.5 kg m2. The center of gravity of the cylinder is located at its geometrical center. The same incoming regular wave of amplitude 0.005 m and wave frequency 8 rad/s is used in the two test cases. A wave probe is present at the position $(0.4, 0, 0)$.

V.7.1 Heave test case

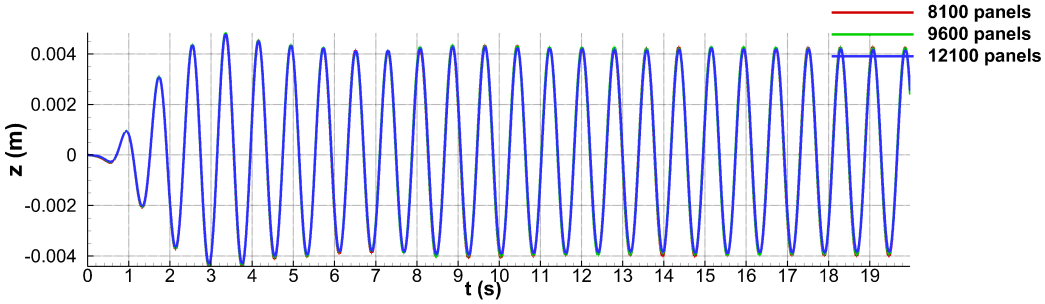
The mesh convergence for the two strategies is displayed in Figure V.26. It shows that a mesh of 10000 panels for the initial mesh generator and of 9600 panels for the new one are sufficient. The initial meshes for the heave test case are shown in Figure V.28. The time step convergence is presented in Figure V.27, showing that a time step of 0.01 s is enough. The comparison between the numerical results of the two mesh generators is shown in Figure V.29 for the heave motion and the wave elevation at the wave probe. The perturbed wave pattern at $t = 5.15$ s is displayed in Figure V.30.

A very good agreement is obtained for all these results. Some slight differences are observed, probably due to the permanent remeshing process in the new mesh generator whereas in the initial mesh strategy the mesh is only deformed if the regeneration is not necessary. The remeshing involves an interpolation between the old and the new mesh as explained in section III.8 and so numerical errors occur.

The sum of the incident wave amplitude and the amplitude of the heave body motion represents 9 % of the draft, which involves a significant deformation of the body mesh during the simulation.

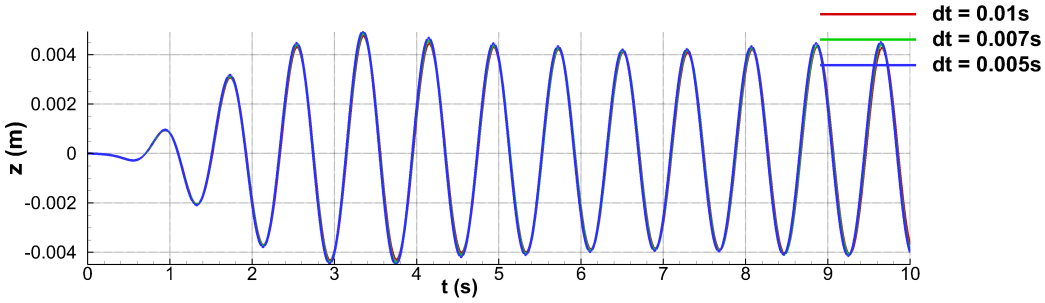


(a) Initial mesh generator

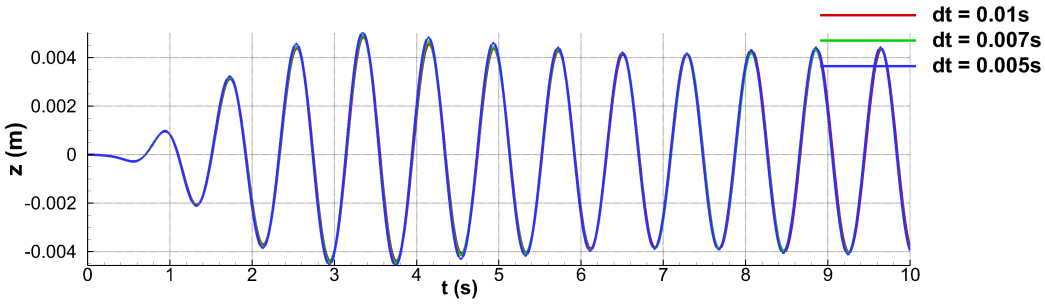


(b) New mesh generator

Figure V.26 – Mesh convergence for the heave test case

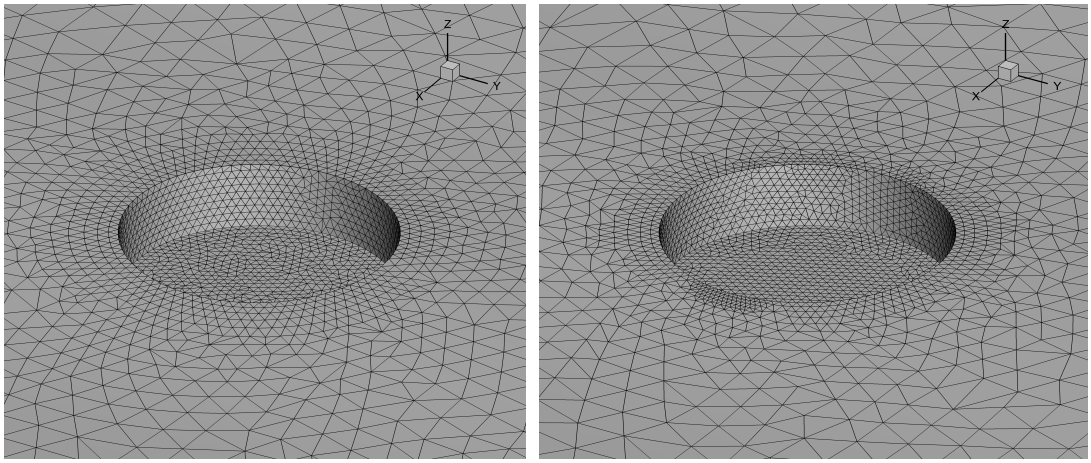


(a) Initial mesh generator



(b) New mesh generator

Figure V.27 – Time step convergence for the heave test case



(a) Initial mesh generator **(b)** New mesh generator

Figure V.28 – Initial meshes for the heave test case

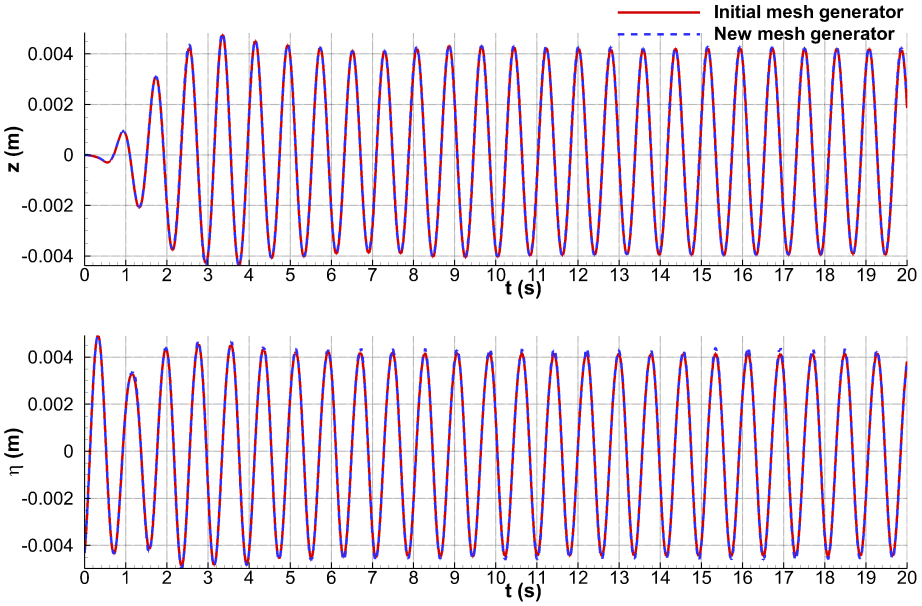
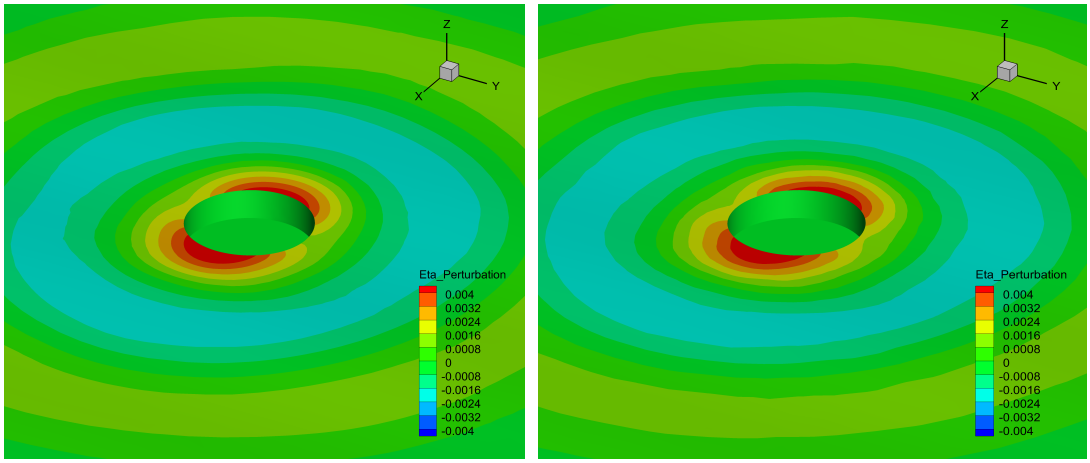


Figure V.29 – Comparison of time series of the heave motion and the wave elevation from numerical results using the two mesh generators



(a) Initial mesh generator **(b)** New mesh generator

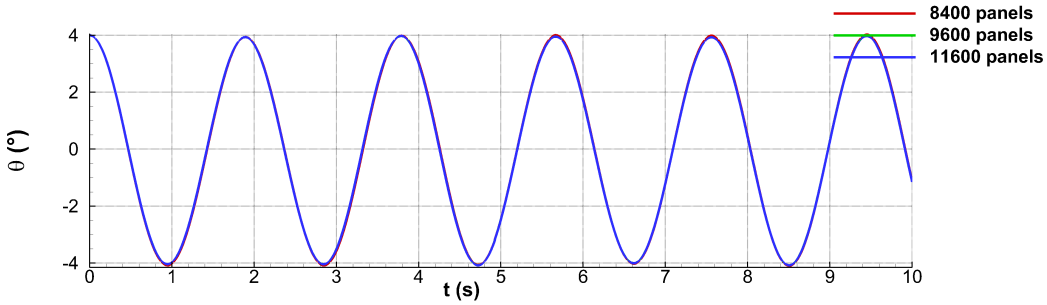
Figure V.30 – Perturbed component of the wave pattern (η^P) at $t = 5.15$ s

V.7.2 Pitch test case

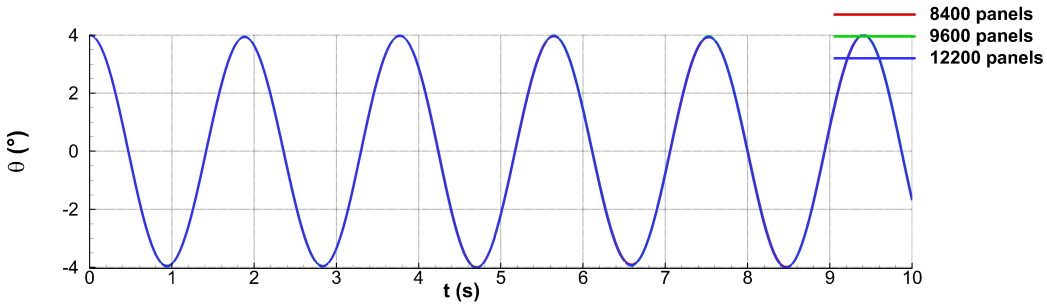
In this test case, the heave motion is blocked and only the pitch motion is free. The mesh convergence for the two strategies is displayed in Figure V.31. It results a mesh of 9600 panels for both the initial mesh generator and the new one is sufficient. The time step convergence is presented in Figure V.32, a time step of 0.01 s is enough. The comparison between the numerical results of the two mesh generators is shown in Figure V.33 for the pitch motion and the wave elevation at the wave probe. The perturbed wave pattern at $t = 5.95$ s is displayed in Figure V.34.

A good agreement is obtained for all these results, except the appearance of a slow phase shift in the pitch motion. As in subsection V.7.1, it is probably the consequence of the permanent remeshing with the new mesh strategy which leads to numerical errors. Moreover, the mesh quality of the new mesh generator is less good compared to the initial one.

To prove the effect of the permanent remeshing in the differences observed in Figure V.33, a simulation using the initial mesh generator with a forced remeshing at every time step has been performed. The results are displayed in Figure V.35. The numerical results obtained with the new mesh generator and the initial mesh generator with a forced remeshing match perfectly. This agreement enforces the conclusion that the permanent remeshing is the cause of the numerical differences between the two mesh strategies.

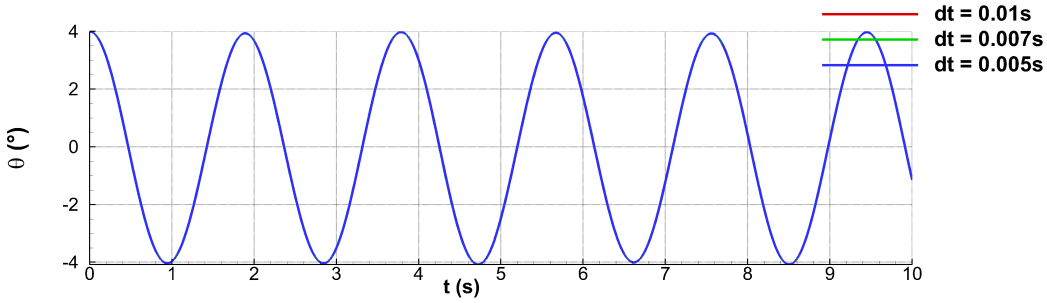


(a) Initial mesh generator

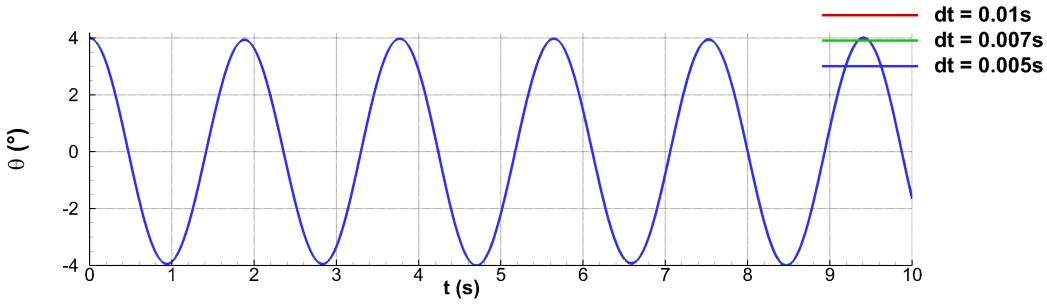


(b) New mesh generator

Figure V.31 – Mesh convergence for the pitch test case



(a) Initial mesh generator



(b) New mesh generator

Figure V.32 – Time step convergence for the pitch test case

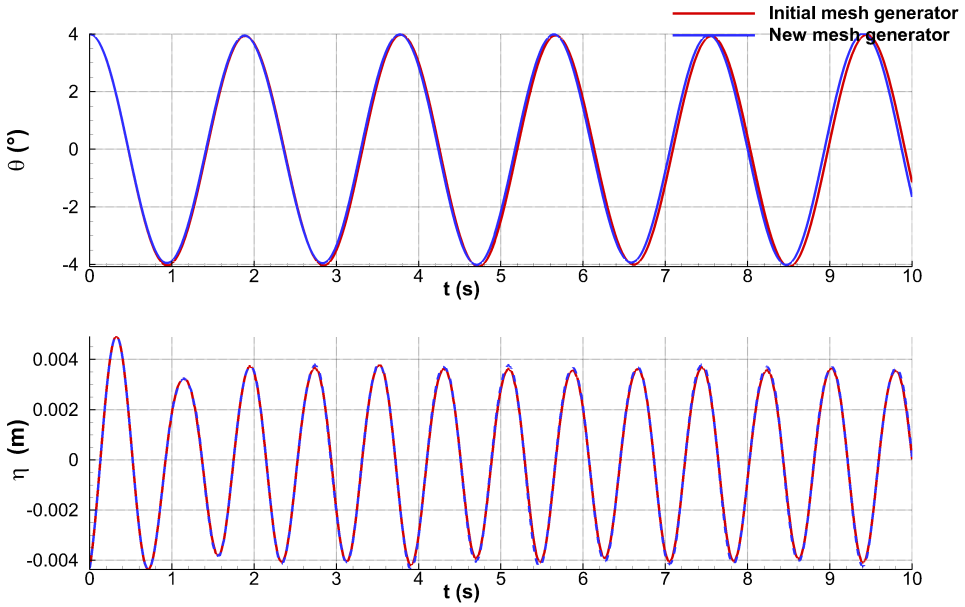
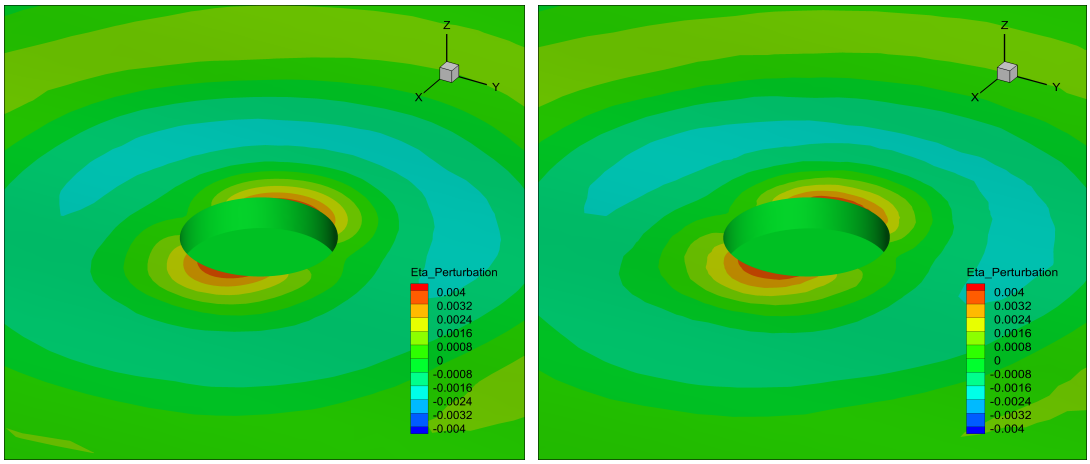


Figure V.33 – Comparison of time series of the pitch motion and the wave elevation from numerical results using the two mesh generators



(a) Initial mesh generator **(b)** New mesh generator

Figure V.34 – Perturbed component of the wave pattern (η^P) at $t = 5.95$ s

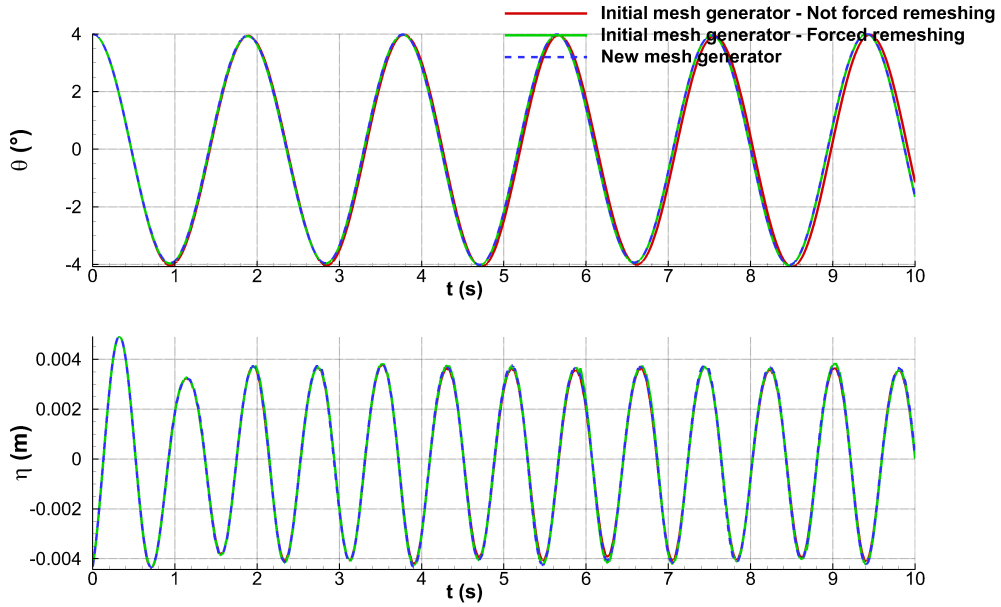


Figure V.35 – Comparison of time series of the pitch motion and the wave elevation from numerical results using the two mesh generators with forced free surface remeshing or not

V.8 Compliance with the functional specification and conclusions

The functional specification of the new mesh generator, listed in section V.2, is now checked:

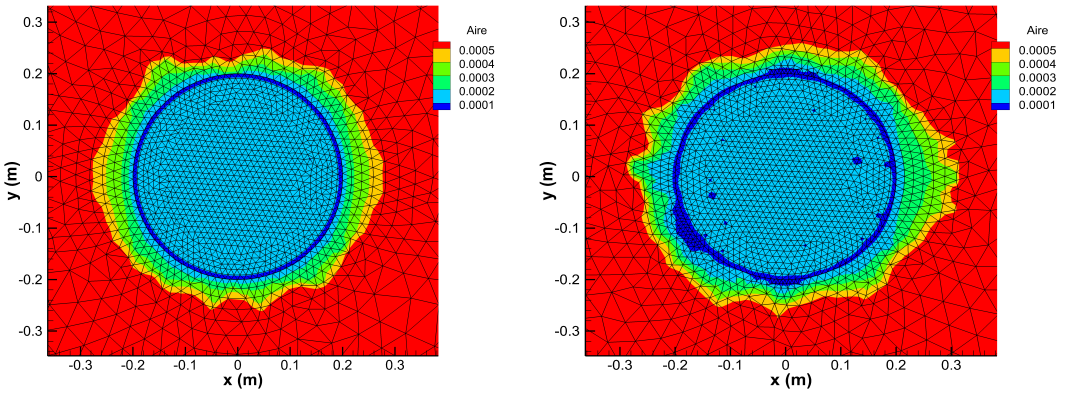
- The *quality* of the new meshes is good and sufficient for performing a time-domain simulation. Nevertheless, at the time of writing, the free surface mesh close to the intersection curve is not as good as in the initial mesh. For example, Figure V.36 shows a close-up of the mesh around the intersection curve in the case of a surface-piercing vertical cylinder. The distribution of the area of the panels should be axisymmetric with the two mesh strategies. It is the case with the initial mesh generator but not with the new mesh generator where some regions with smaller panels arise. This is due to the non-equidistance of the nodes on the intersection line in the new mesh generator, despite the use of the spring analogy method. This equidistance is guaranteed in the initial mesh generator because the intersection curve is meshed first, the body and free surface meshes being generated from it. Possible enhancements would be the improvement of the panel merging method and the projection of the nodes on the intersection curve before clipping the mesh to avoid the creation of tiny panels. The use of an another open-source library such as *GTS*[12] which deals with surface meshes is also a possibility.

- The *non-regression* of the new mesh generator is proved by the two test cases presented in section V.7. The same simulations could be performed, involving a good accuracy in the numerical results.

[12] http://gts.sourceforge.net

- The *robustness* is demonstrated by the creation of meshes which could not be taken into account by the initial mesh generator (horizontal cylinder, SEAREV, TLP, FPSO). The possibilities offered by the new mesh generator mainly depend on the quality of the initial mesh. So far, only one body in one piece and involving a single intersection curve (no moon pool) is considered. The new mesh strategy is feasible with any wave model by the use of a bisection method to find the intersection curve as presented in subsection V.4.1.
- The *modularity* is checked as displayed on the scheme of the coupling in Figure V.23;
- The *user-friendlyness* of the new mesh strategy is ensured by the use of an initial mesh. The rest of the process is totally automatic for any body. No extra-coding is required.
- The *memory allocation* is good, without memory leak in the scripts of the new mesh generator which were implemented for this purpose. Consequently, only the memory leaks coming from the use of the advance front method remain. But those created for the body mesh generation and the intersection curve tracking in the initial mesh generator disappear.
- The *process time* is slightly more important with the new mesh generator than with the initial one due to the spring analogy method and some parts in Python (cf. section V.6). The translation of the Python code into Fortran and the use of another smoothing method would reduce the CPU time.

Hence, a new mesh strategy has been developed in this chapter, based on the idea: *the body mesh for the user, the free surface mesh for the solver*. It enables to mesh non-academic surface-piercing bodies and gives very good results compared to the initial mesh generator. *Meshmagick* becomes the last module of *InWaveS_CN*. The next chapter presents an application of *InWaveS_CN* in the case of an upending operation.



(a) Initial mesh generator **(b)** New mesh generator

Figure V.36 – Area of the free surface panels close to the intersection curve with the two mesh strategies

Chapter VI

Experiments

Contents

In order to validate the numerical tool created during this PhD work, some experiments were conducted in a wave basin of Ecole Centrale de Nantes. The numerical simulations of these experiments involve the use of multibody dynamics (InWave), cable dynamics and a winch (CableDyn), hydrodynamics for a body with a large amplitude motion (WS_CN) and with a complex geometry (Meshmagick). All the modules constituting InWaveS_CN are required. First, the experiments are detailed. The numerical results and the comparison with the measurements close this final chapter.

VI.1 Presentation of the experiments

In order to validate the coupling between the Composite-Rigid-Body algorithm and the unsteady potential flow theory based on the weak-scatterer hypothesis, experiments were conducted in the shallow water tank of *Ecole Centrale de Nantes*. These experiments represent the upending of a spar in waves. A spar is moved from a horizontal position to its equilibrium vertical position using a cable on a winch. These experiments follow three steps:

- At the beginning of the experiments, the spar is kept horizontal with a cable (Figure VI.1a);
- The cable is unwound using a winch, the spar has a rotating motion around a horizontal fixed axis (Figure VI.1b);
- The spar reaches its vertical equilibrium position (Figure VI.1c).

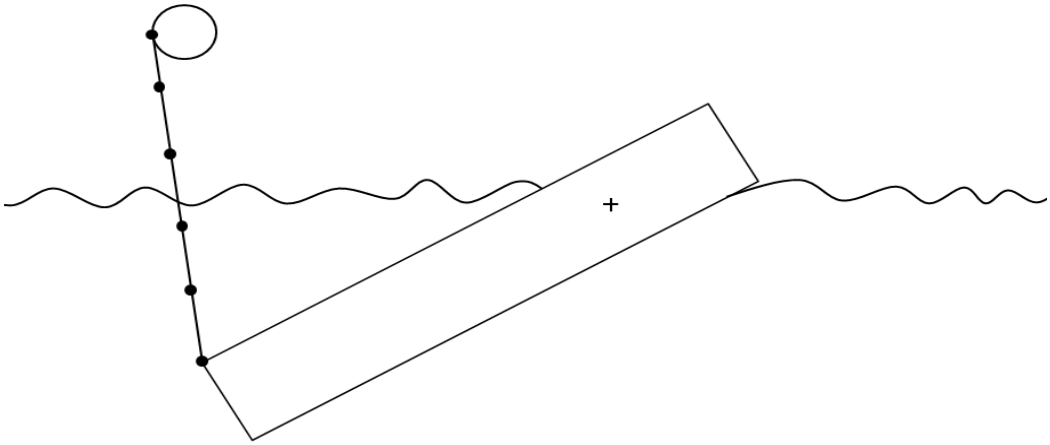
An upending operation was preferred to a lowering operation because the experimental set-up was more straightforward. Only one body, available in the laboratory, was required and the set-up was easier and quicker to install: one body linked with a cable and a winch to the fixed footbridge instead of two bodies in free motion linked to each other through a cable and winch.

The interests of these experiments are:

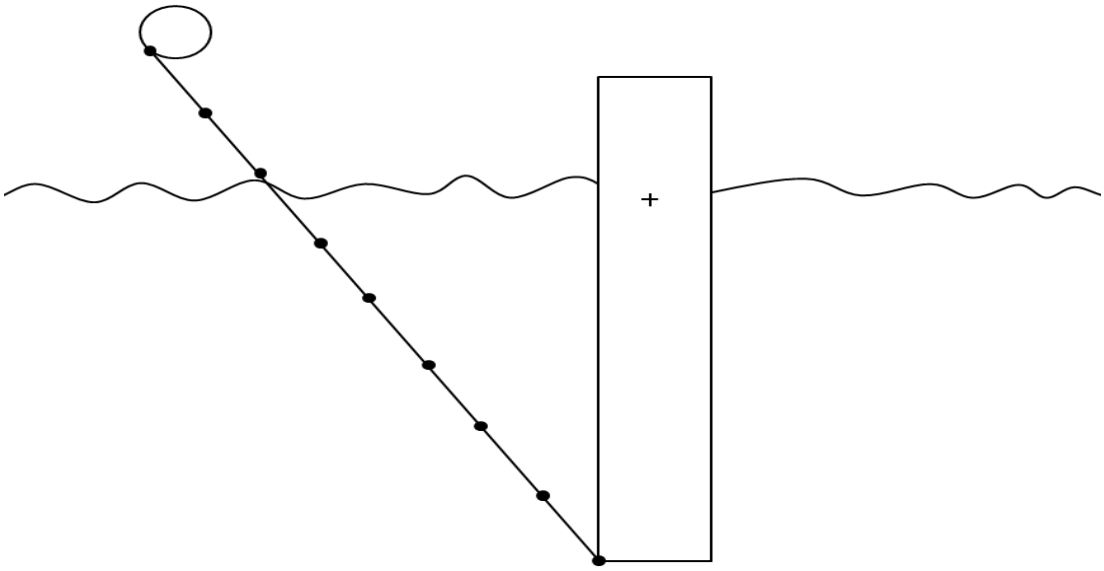
- The presence of an articulated multibody system (the spar and its revolute joint);
- A cable is unwound using a winch;
- The wetted surface of the spar is subject to a large deformation;
- The spar has a free motion so a fluid-structure coupling arises.

These characteristics are at the heart of the numerical tool developed during our work (*InWaveS_CN*) and match the work achieved in the previous five chapters of the present PhD thesis.

(a) Initial horizontal position



(b) Unwound cable



(c) Vertical equilibrium position

Figure VI.1 – Three steps of the spar upending. The fixed axis is indicated by the cross and the winch by the circle.

VI.1.1 Experimental set-up

The experiments were conducted in the shallow water basin of *Ecole Centrale de Nantes* (Figure VI.2). Its dimensions are 20 m × 9.5 m with a water depth of 1 m. The basin is equipped with a flap-type wave maker generating unidirectional waves.

Opposite the wave generator, a passive wave absorber is installed (absorbing beach). The size of the basin makes it suitable for short waves. The basin is filled of freshwater so the water density is $1000\,\mathrm{kg/m^3}$.

Figure VI.2 – Shallow water basin of *Ecole Centrale de Nantes*

The spar is a cylindrical buoy (illustrated in Figures VI.3 and VI.4) made of three parts:

- An external PVC waterproof pipe;
- A lead weight;
- A threaded rod.

Tables VI.1 and VI.2 present the geometrical and inertia characteristics of the buoy. The total mass is 28 kg and the center of gravity is located at 37 cm from the bottom of the buoy. The draft in the vertical equilibrium position is 0.89 m. The buoy and the footbridge of the basin are connected by a revolute joint. Its axis is positioned above the mean free surface in the vertical equilibrium position of the buoy. By this way, no joint loads exist at the equilibrium. The revolute joint system is displayed in Figure VI.5.

Figure VI.3 – Buoy[1]

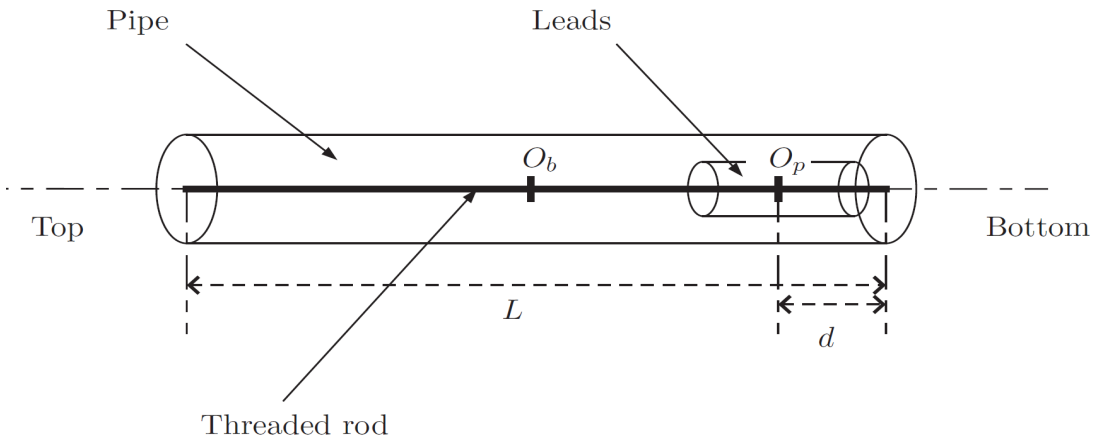


Figure VI.4 – Sketch of the buoy

[1] The heave plate was not present during the experiments.

Length	Symbol	Value (m)
Buoy length	L	1.32
Buoy diameter	D	0.2
Leads location from the bottom	d	0.13

Table VI.1 – Geometrical characteristics of the buoy

Element	Mass (kg)	Inertia (kg m2)	Reference point
Pipe	10	1.5	O_b
Leads	16	0.058	O_p
Threaded rod	2	0.27	O_b

Table VI.2 – Mass and inertia of each part of the buoy. The points are defined in Figure VI.4.

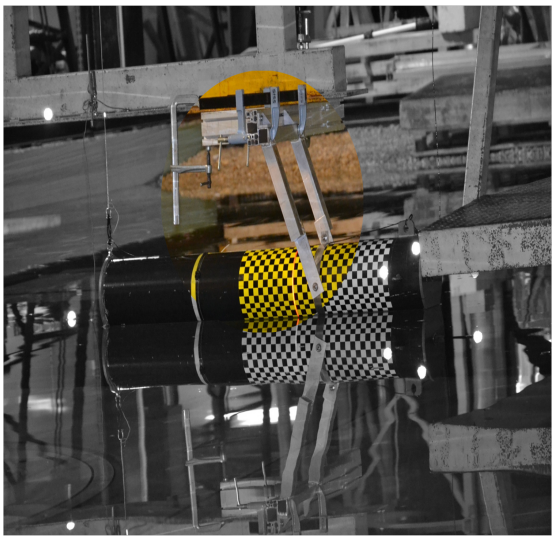


Figure VI.5 – Revolute joint system

The cable is made of Dyneema with a diameter of 4 mm. This leads to a very stiff cable. The cable is fixed to the spar at its bottom. The winch, shown in Figure VI.6, unwinds the cable at a constant velocity after a linear velocity ramp.

Finally, the experimental set-up is presented in Figure VI.7.

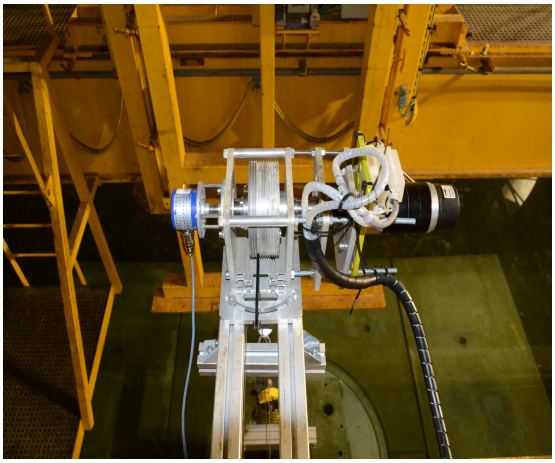


Figure VI.6 – Winch

Figure VI.7 – Experimental set-up

VI.1.2 Test matrix

The test matrix depends on four parameters:

- The wave amplitude;
- The wave period;
- The orientation of the moving buoy with respect to the fixed wave direction;
- The lowering velocity of the cable.

Only regular waves were used as *WS_ CN* cannot deal with irregular waves without adding extra developments. The test matrix is given by Table VI.3. Cases 1, 8 and 15 represent three tests without waves. Each test lasts less than one minute.

Case	Wave amplitude (m)	Wave period (s)	Heading angle (°)	Lowering velocity (m/s)
1	0	ø	90	0.067
2	0.01	0.7	90	0.067
3	0.01	1	90	0.067
4	0.01	1.5	90	0.067
5	0.02	0.7	90	0.067
6	0.02	1	90	0.067
7	0.02	1.5	90	0.067
8	0	ø	0	0.067
9	0.01	0.7	0	0.067
10	0.01	1	0	0.067
11	0.01	1.5	0	0.067
12	0.02	0.7	0	0.067
13	0.02	1	0	0.067
14	0.02	1.5	0	0.067
15	0	ø	0	0.033
16	0.01	0.7	0	0.033
17	0.01	1	0	0.033
18	0.01	1.5	0	0.033
19	0.02	0.7	0	0.033
20	0.02	1	0	0.033
21	0.02	1.5	0	0.033

Table VI.3 – Test matrix of the experimental tests. The heading angle is the angle between the buoy vertical axis and the wave direction.

VI.1.3 Output data

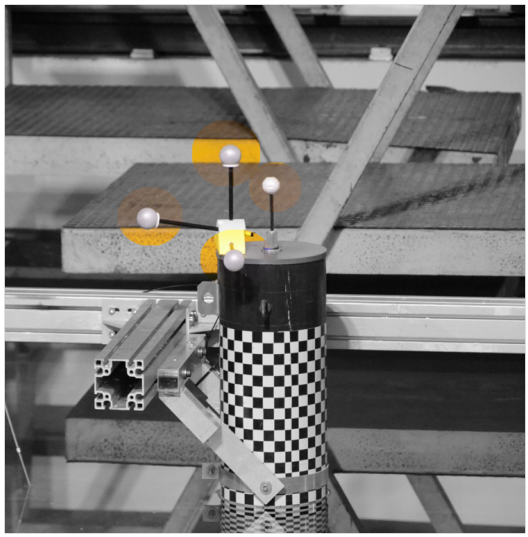
Five output data are measured:

- The position of the buoy;
- The position of the cable;
- The tension in the cable;

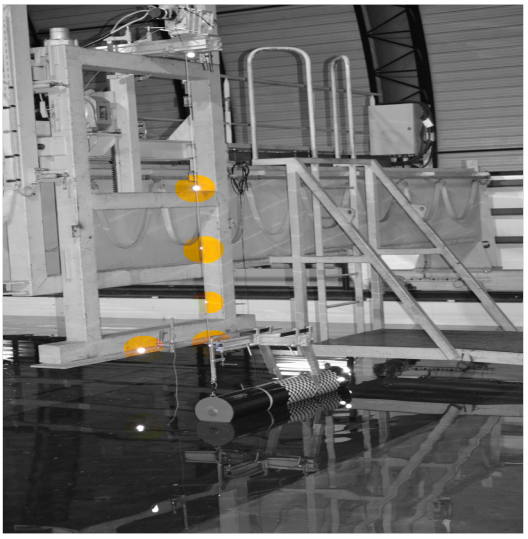
– The lowering velocity;

– The wave elevation.

The positions of the buoy, the wave elevation and the cable profile are tracked using spherical markers. The motion capture software *Qualisys* follows the markers with two infrared cameras. Three markers are necessary to track the three-dimensional motion of the buoy. Figure VI.8 shows the markers of the buoy, the cable and the winch. Figure VI.9 presents the results of *Qualisys* at the initial stage for Case 15.



(a) Buoy



(b) Cable, footbridge and free surface

(c) Winch

Figure VI.8 – Positions of the markers

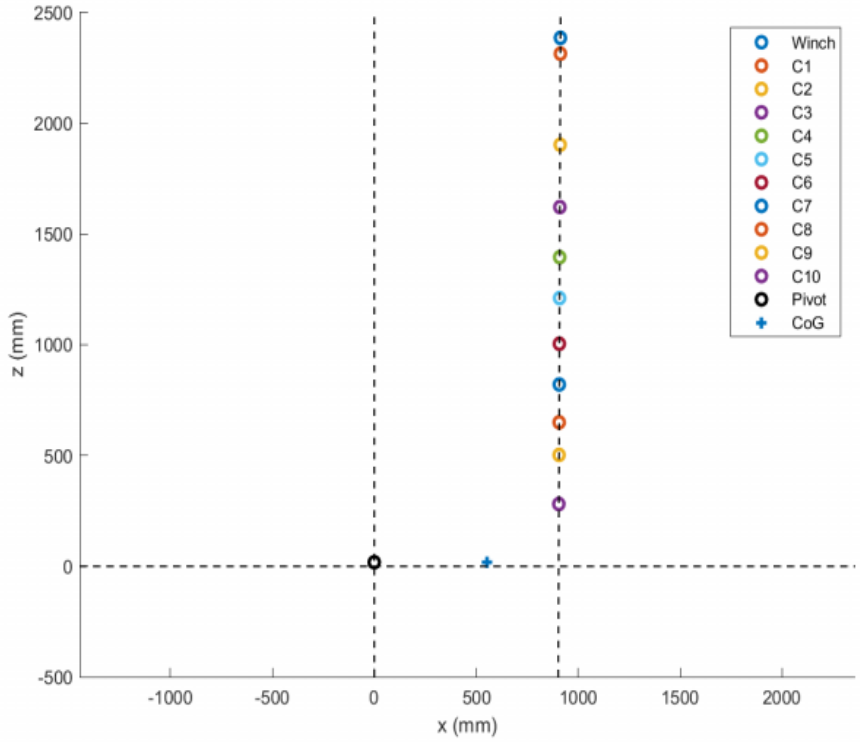


Figure VI.9 – Markers position using *Qualisys* for Case 15. *CoG* denotes the position of the center of gravity of the buoy. *C#* represents the cable markers. The horizontal dashed line indicates the mean sea level. The left vertical dashed line is the vertical axis at the revolute axis and the right vertical dashed line is the initial cable profile.

VI.1.4 Measurements

VI.1.4.1 Winch velocity

In Cases 15, 16, 17 and 18, the winch velocity is 0.033 m/s. A linear ramp is used for 3 s. The winch velocity command signal is displayed in Figure VI.10.

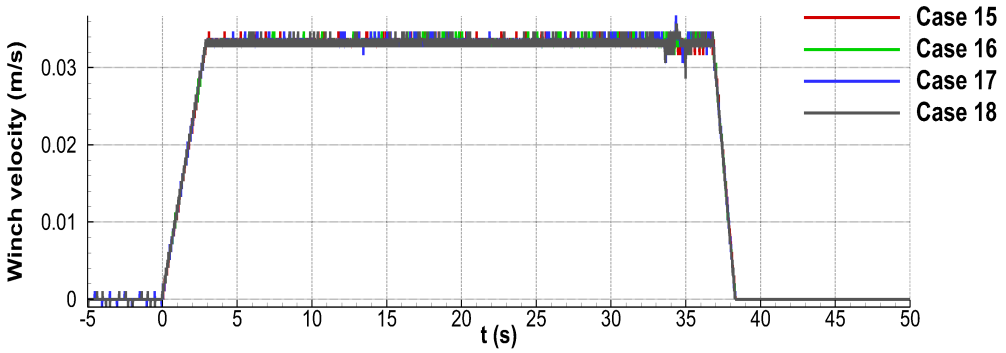


Figure VI.10 – Winch velocity for Cases 15, 16, 17 and 18. $t = 0$ s denotes the starting time of the winch.

VI.1.4.2 Buoy motion and tension

The cable tension and the motion of the buoy are presented in Figure VI.11. The initial position is horizontal and once the cable is lowered, a vertical position is reached. Oscillations are present around the vertical equilibrium position.

Regarding the cable tension, the initial tension when the buoy is horizontal is 122 N. When the floating body is lowered into the water, this value decreases due to the increase of the hydrostatic loads. Once the vertical equilibrium position of the cylinder is reached, the cable is slack and so the tension is zero.

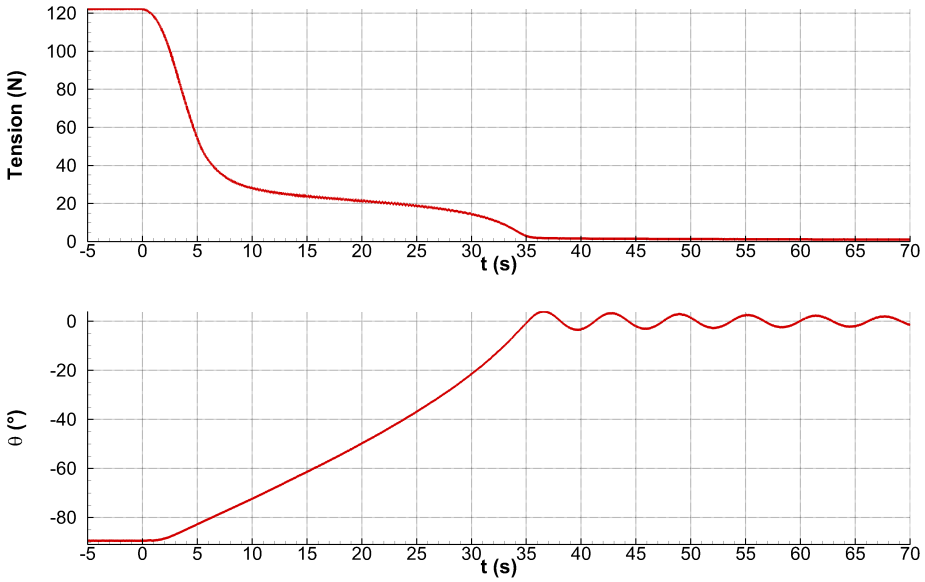


Figure VI.11 – Cable tension and rotational motion for Case 15. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

Analytically, the equilibrium of the buoy at the initial stage is given by the following motion equation written at the keel:

$$\mathbf{P}_{Buoy} + \mathbf{F}_{Buoy}^{Static} + \mathbf{T}_{Buoy} = \mathbf{0}_{3\times1} \tag{VI.1}$$

$$\mathbf{P}_{Buoy} \times \mathbf{KG} + \mathbf{F}_{Buoy}^{Static} \times \mathbf{KB} + \mathbf{T}_{Buoy} \times \mathbf{KF} = \mathbf{0}_{3\times1} \tag{VI.2}$$

with:

- $\mathbf{P}_{Buoy}$ the weight of the buoy;
- $\mathbf{F}_{Buoy}^{Static}$ the hydrostatic loads;
- $\mathbf{T}_{Buoy}$ the cable tension;
- $\mathbf{K}$ the keel position;
- $\mathbf{G}$ the position of the center of gravity;
- $\mathbf{B}$ the buoyancy center position;
- $\mathbf{F}$ the fairlead position.

The projection of (VI.1) along the vertical axis and (VI.2) along the axis perpendicular to the plane including the buoy and the cable give:

$$-mg + \rho V g + T = 0 \tag{VI.3}$$

$$-mgKG + \rho V g KB + TKT = 0 \tag{VI.4}$$

where m, g and V are the mass of the buoy, the gravity constant and the immersed volume.

Equation VI.3 leads to:

$$V = \frac{1}{\rho}\left(m - \frac{T}{g}\right) \tag{VI.5}$$

Equation VI.4 involves:

$$T = mg\left(\frac{KB - KG}{KB - KT}\right) \tag{VI.6}$$

The cable is fixed at the keel of the buoy so $KT = 0$ and at the horizontal initial position $KB = \frac{L}{2}$ with L the length of the buoy, consequently:

$$T = mg\left(1 - \frac{2}{L}KG\right) \tag{VI.7}$$

Thus, the tension in the initial configuration is 120.65 N. This does not match perfectly with the experimental value observed in Figure VI.13 because the buoy is not totally horizontal in its initial position.

Figure VI.12 presents the wave elevation for Case 17 for a regular wave of amplitude 0.01 m and a wave period of 1 s. The effect of the waves on the buoy rotation motion is insignificant (Figure VI.13). But it is clearly noticeable on the cable tension. The tension difference between the tension in still water and in waves is shown in Figure VI.14. The dynamic tension clearly depends on the wave period. But, when the buoy reaches its final position, the tension is independent of the waves, because the cable is slack.

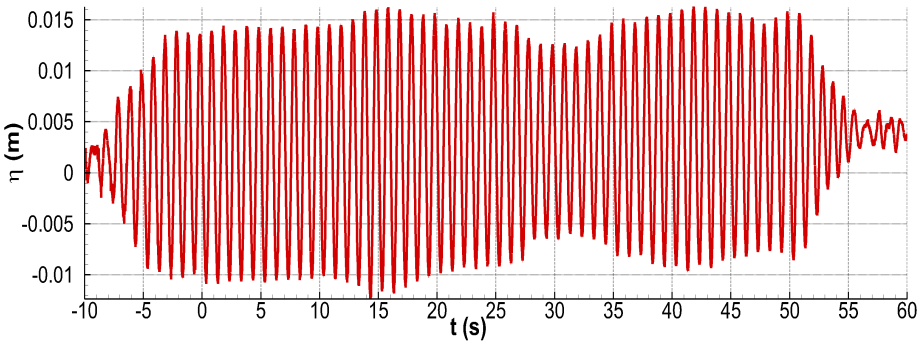


Figure VI.12 – Wave elevation for Case 17. $t = 0$ s denotes the starting time of the winch.

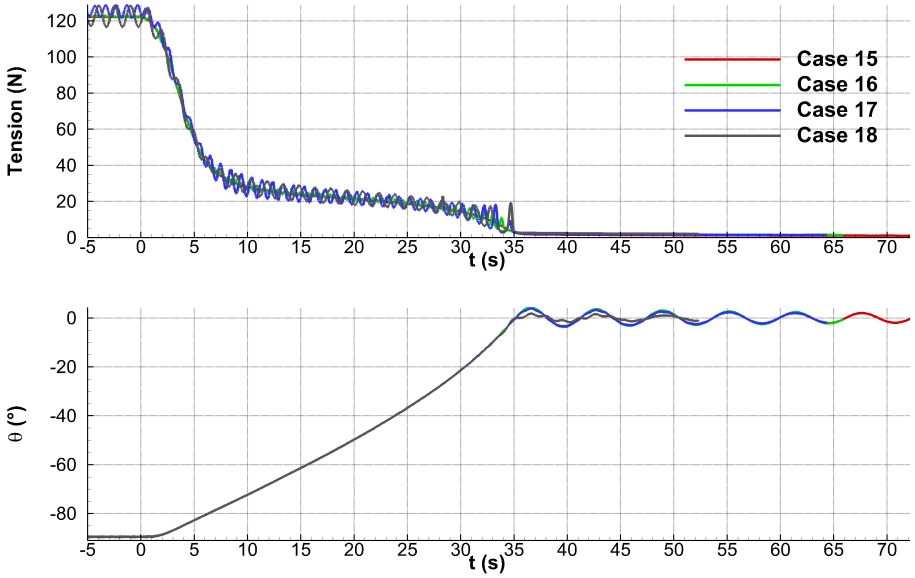


Figure VI.13 – Cable tension and rotational motion for Cases 15, 16, 17 and 18. $t = 0$ s denotes the starting time of the winch.

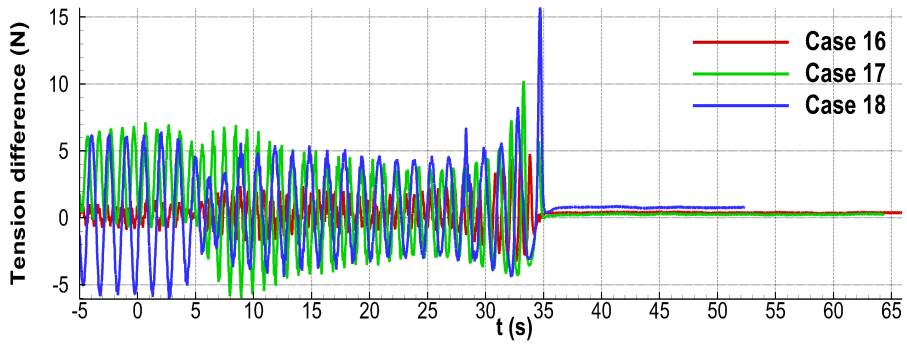


Figure VI.14 – Tension difference between Case 15 (still water) and Cases 16, 17 and 18 (in waves). $t = 0$ s denotes the starting time of the winch.

VI.1.4.3 Cable profile

The trajectory of the markers in Case 15 is presented in Figure VI.15. As expected, the winch is motionless. The position of the center of gravity follows a circular trajectory because of the constrained buoy motion due to the rotation axis. The center of the circular motion enables to find the position of the revolute axis. The other markers show the lowering of the cable and its horizontal displacement due to the buoy motion. When the cable is slack, the marker trajectories become arbitrary. Once the cable markers are in the water, they could not be tracked anymore by the infrared cameras. The markers on the buoy could be tracked because there were never immersed.

The in-plane horizontal and vertical motions of the markers trajectory are displayed in Figure VI.16 while the out-of-plane motion is shown in Figure VI.17. The cable is out-of-plane when it becomes slack.

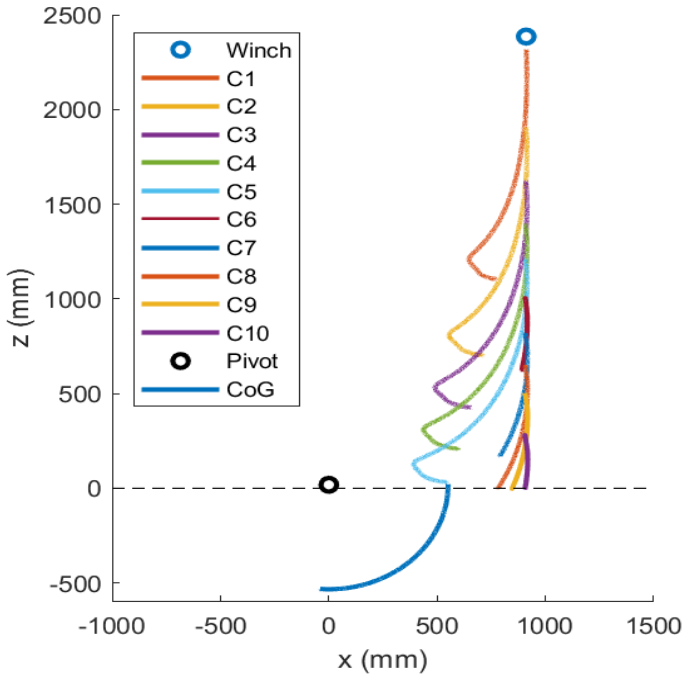


Figure VI.15 – Trajectory of the markers for Case 15. CoG denotes the position of the center of gravity of the buoy. $C\#$ represents the cable markers.

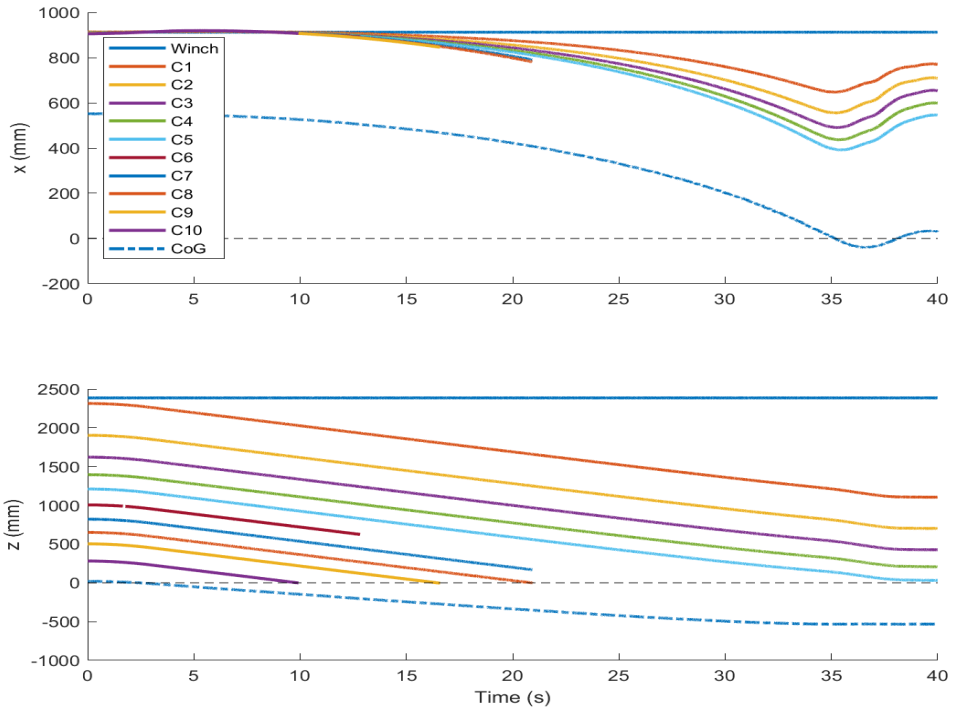


Figure VI.16 – Horizontal (x) and vertical (z) motion of the markers for Case 15. CoG denotes the position of the center of gravity of the buoy. $C\#$ represents the cable markers. $t = 0\,\mathrm{s}$ is the starting time of the winch.

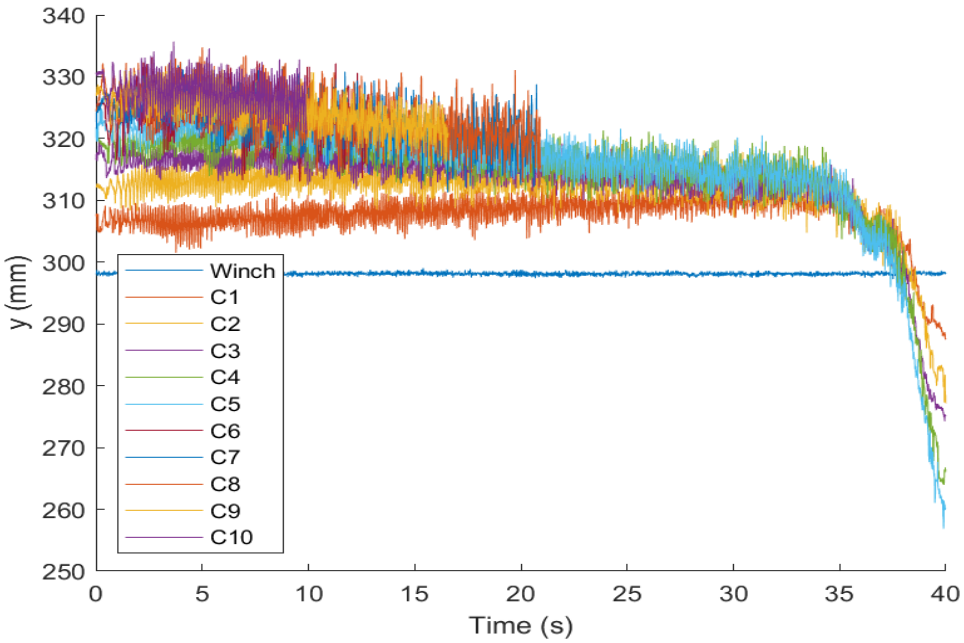


Figure VI.17 – Out-of-plane (y) motion of the markers for Case 15. CoG denotes the position of the center of gravity of the buoy. $C\#$ represents the cable markers. $t = 0\,\mathrm{s}$ is the starting time of the winch.

The cable profile is presented in Figures VI.18, VI.19 and VI.20 for Case 18. The effects of the waves is only noticeable when the cable is slack of the end of the experiments.

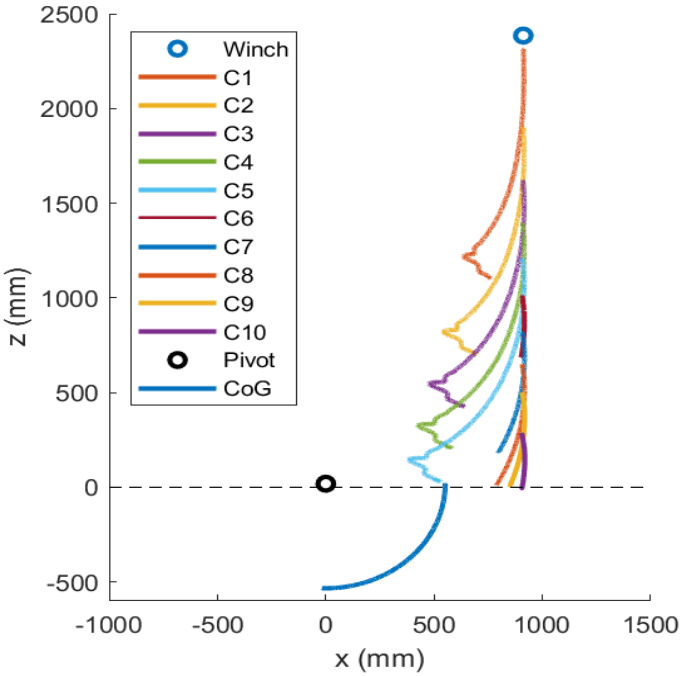


Figure VI.18 – Trajectory of the markers for Case 18. CoG denotes the position of the center of gravity of the buoy. $C\#$ represents the cable markers.

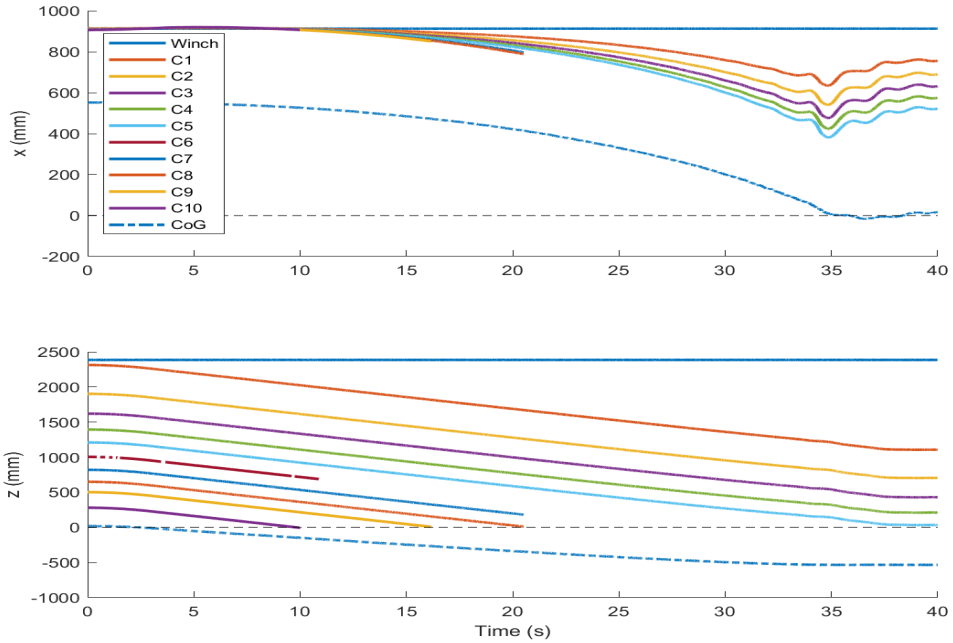


Figure VI.19 – Horizontal (x) and vertical (z) motion of the markers for Case 18. CoG denotes the position of the center of gravity of the buoy. $C\#$ represents the cable markers. $t = 0\,\mathrm{s}$ is the starting time of the winch.

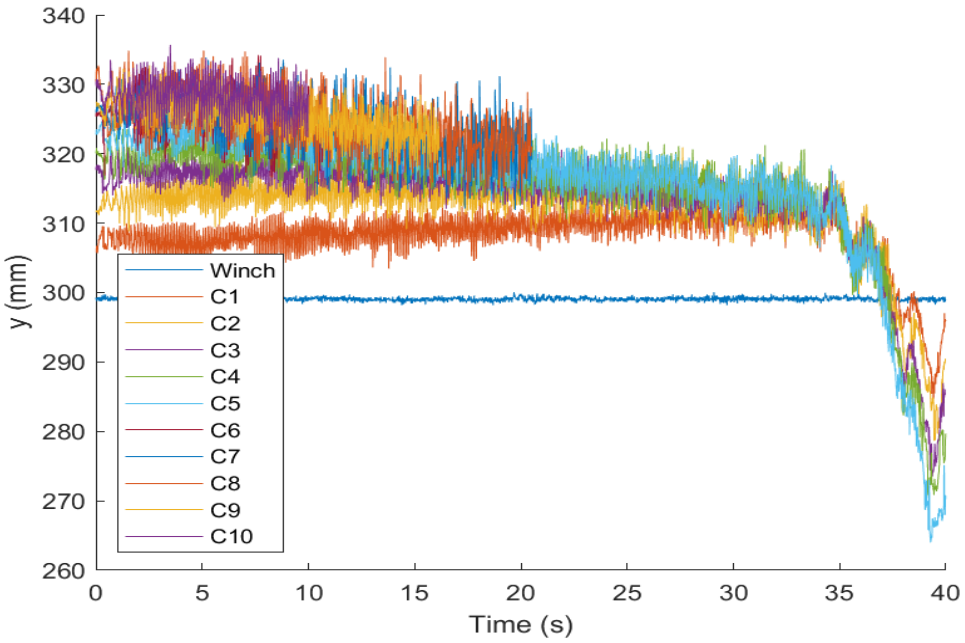


Figure VI.20 – Out-of-plane (y) motion of the markers for Case 18. CoG denotes the position of the center of gravity of the buoy. $C\#$ represents the cable markers. $t = 0\,\mathrm{s}$ is the starting time of the winch.

The experimental results for other Cases are presented in Appendix C.

VI.2 Comparisons between the experiments and the numerical simulations

The final coupling between *InWave*, *CableDyn*, *WS_CN* and *Meshmagick*, as presented in Figure V.23, is used to perform the numerical simulations of the experimental tests.

VI.2.1 Case 15

The initial mesh of the buoy given in input of *Meshmagick* is shown in Figure VI.21. It is made of 8862 panels.

The multibody system is composed of two bodies:

- The base (body 0) which represents the footbridge of the basin and is motionless:

$$\boldsymbol{\eta}_0 = \mathbf{0}_{6\times 1} \tag{VI.8}$$

- The buoy (body 1) linked to the base by a revolute joint to model the rotation axis. The initial modified Denavit-Hartenberg parameters are given in Table VI.4.

j	σ_j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	0	$\frac{\pi}{2}$	0.0183	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0

Table VI.4 – *mDH* parameters for Case 15

The cable is made of one element. The fairlead position in the body frame is:

$$^1\mathbf{O}_1\mathbf{A}_1 = (-0.1, -0.908, 0)^T \tag{VI.9}$$

The hydrodynamic loads on the cable are neglected.

Figure VI.22 presents the comparison of the initial position of the winch, the cable, the buoy center of gravity and the rotation axis between the experiments and the numerical simulation.

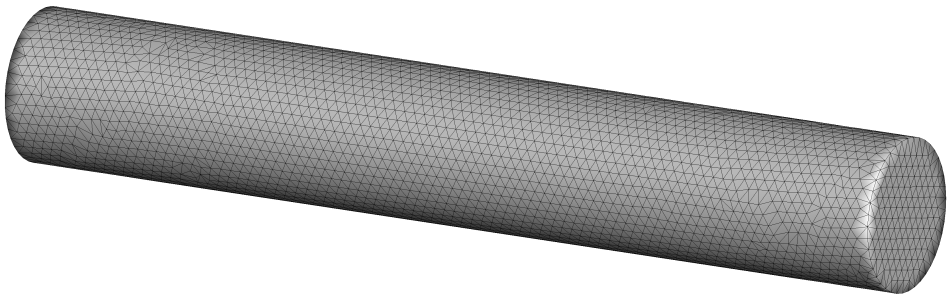


Figure VI.21 – Initial mesh of the buoy with 8862 panels

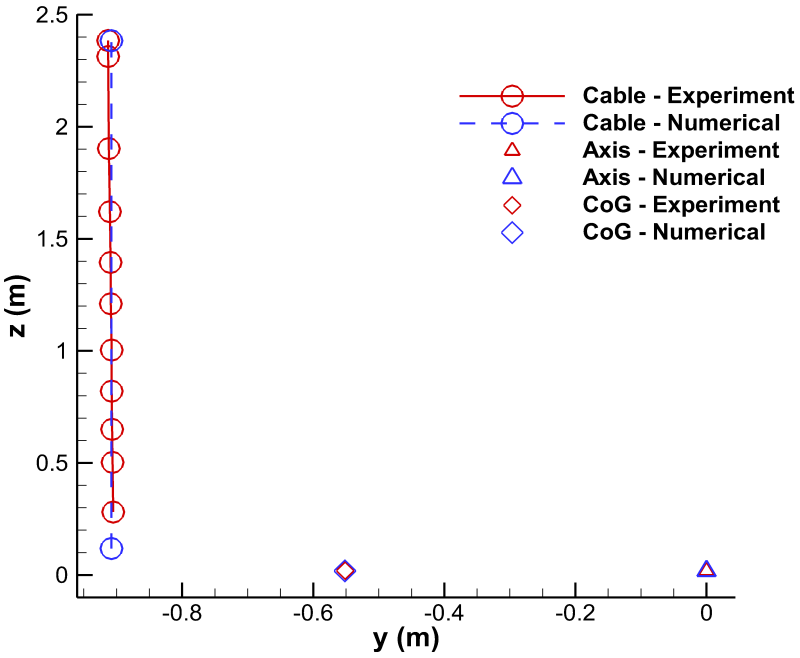


Figure VI.22 – Position of the cable, the buoy center of gravity (CoG) and the rotation axis ($Axis$) in the experiments (red) and the numerical simulation (blue) for Case 15

The command of the lowering velocity at the winch is shown in Figure VI.23. A constant velocity of 0.033 m/s is reached after a linear ramp of 3 s. The time series are identical in the experimental and numerical results.

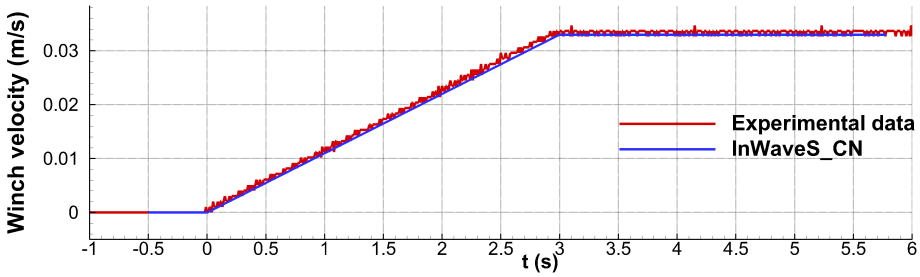


Figure VI.23 – Comparison of time series of the lowering velocity at the winch from numerical and experimental results for Case 15. $t = 0$ s denotes the starting time of the winch.

The mesh convergence is presented in Figure VI.24 and indicates a total mesh of 8000 panels is enough for Case 15. This mesh is displayed in Figures VI.25 and VI.26. The domain is cylindrical with a radius of 2 m and an absorbing beach of 1 m. The

Meshmagick input coefficients are:

$$\begin{cases} \alpha_V = 0.5 \\ \alpha_H = 0.2 \\ \beta_V = 0.4 \\ \beta_H = 0.2 \end{cases} \tag{VI.10}$$

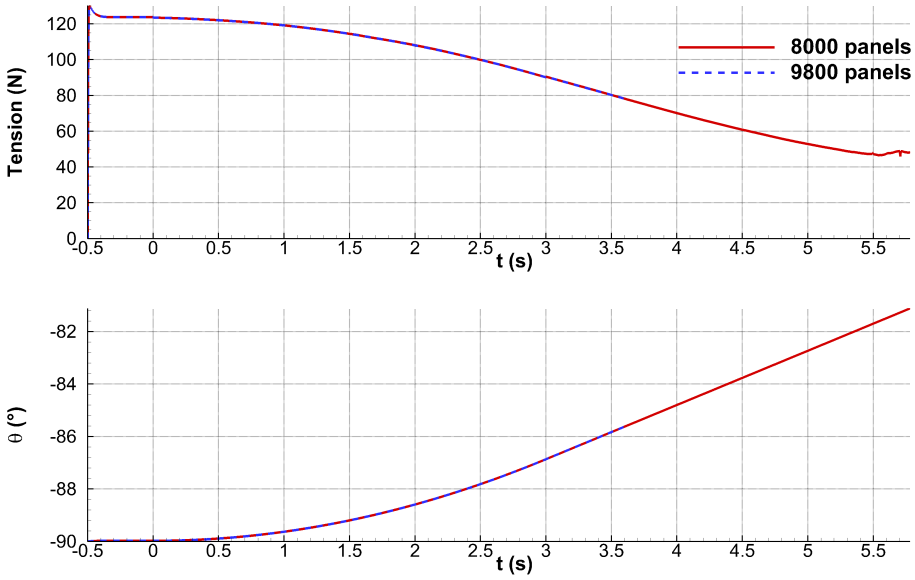


Figure VI.24 – Comparison of time series of the cable tension and the angular position from numerical results with different meshes for Case 15. $t = 0$ s denotes the starting time of the winch.

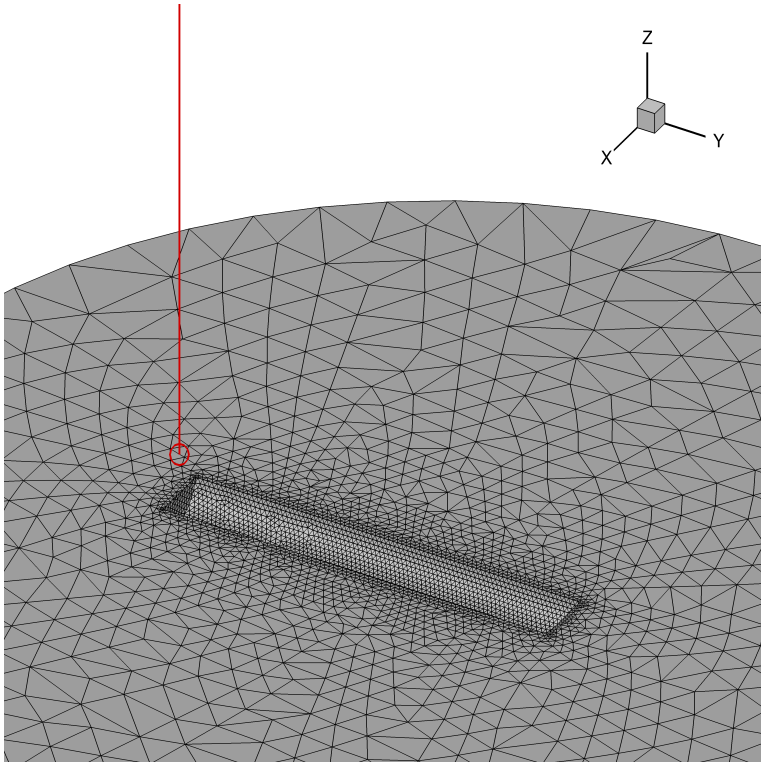


Figure VI.25 – Three-dimensional view of the initial mesh for Case 15. The cable is in red.

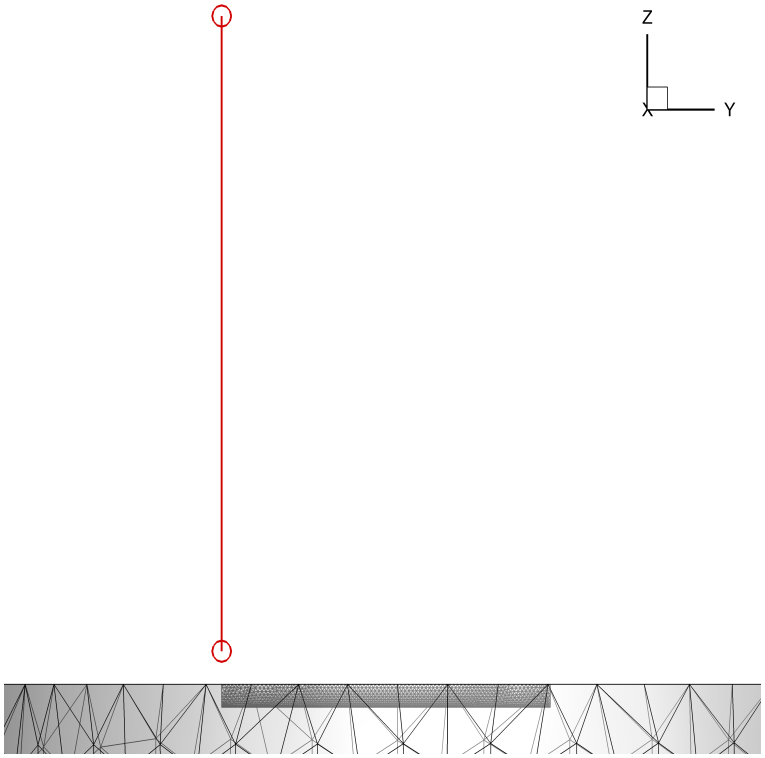


Figure VI.26 – Bi-dimensional view of the initial mesh for Case 15. The cable is in red.

The cable used in the experiments is stiff so the value of the numerical cable stiffness has to be high enough not to modify the numerical results. If the stiffness increases,

the necessary time step to be converged decreases. Consequently, the influence of the cable stiffness is studied at the same time as the time step convergence. Three cases are considered:

- An axial stiffness of 6.7×10^5 N and an axial damping coefficient of 3×10^7 N s/m^2, which form the reference;
- A stiffness which is twice as high as the reference with the same damping coefficient;
- A stiffness which is five times as high as the reference with the same damping coefficient.

The time step convergence for these three cases are presented in Figures VI.27, VI.28 and VI.29. As expected, a very small time step is required when the cable stiffness is higher. A comparison of the numerical results after convergence for the three values of the cable stiffness is displayed in Figure VI.30. The tension and the rotational motion are independent of the cable stiffness. The simulations do not last the same duration because of memory leak problems. Finally, an axial stiffness of 6.7×10^5 N, an axial damping coefficient of 3×10^7 N s/m^2 and a time step of 0.001 s are chosen.

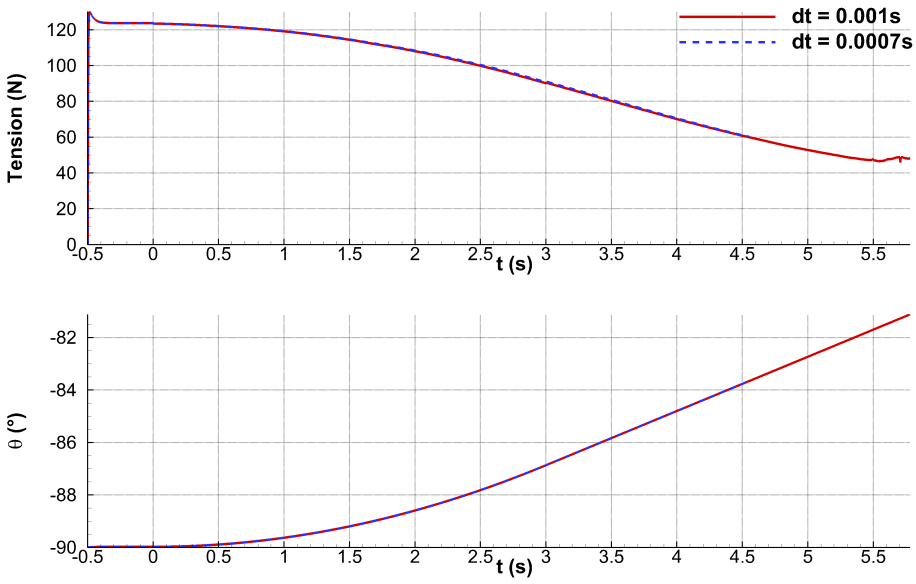


Figure VI.27 – Comparison of time series of the cable tension and the angular position from numerical results for Case 15 for a cable stiffness of 6.7×10^5 N. $t = 0$ s denotes the starting time of the winch.

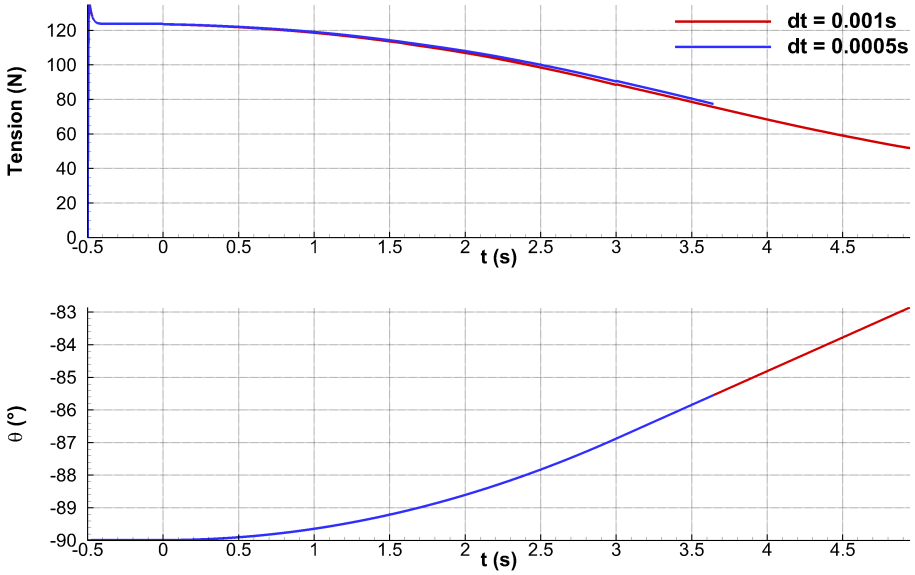


Figure VI.28 – Comparison of time series of the cable tension and the angular position from numerical results for Case 15 for a cable stiffness of 1.34×10^6 N. $t = 0$ s denotes the starting time of the winch.

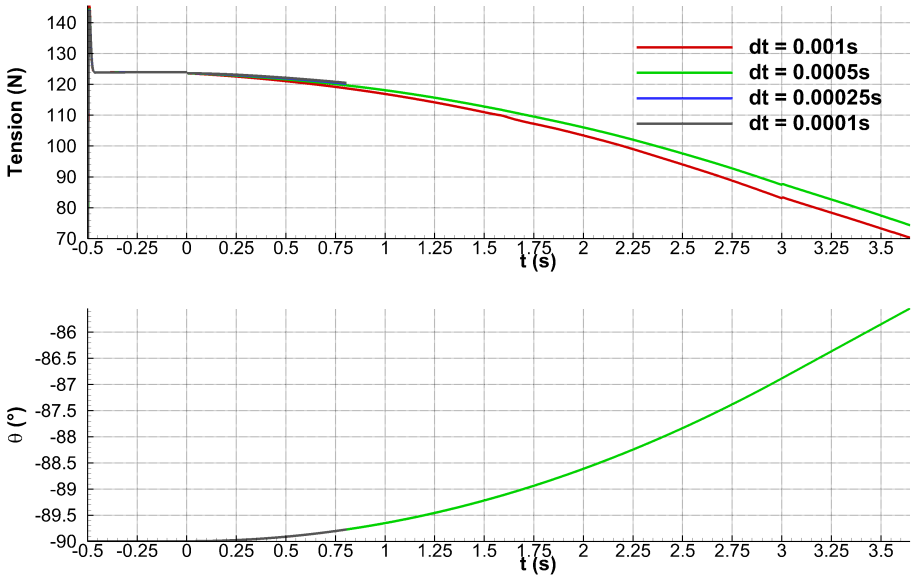


Figure VI.29 – Comparison of time series of the cable tension and the angular position from numerical results for Case 15 for a cable stiffness of 3.35×10^6 N. $t = 0$ s denotes the starting time of the winch.

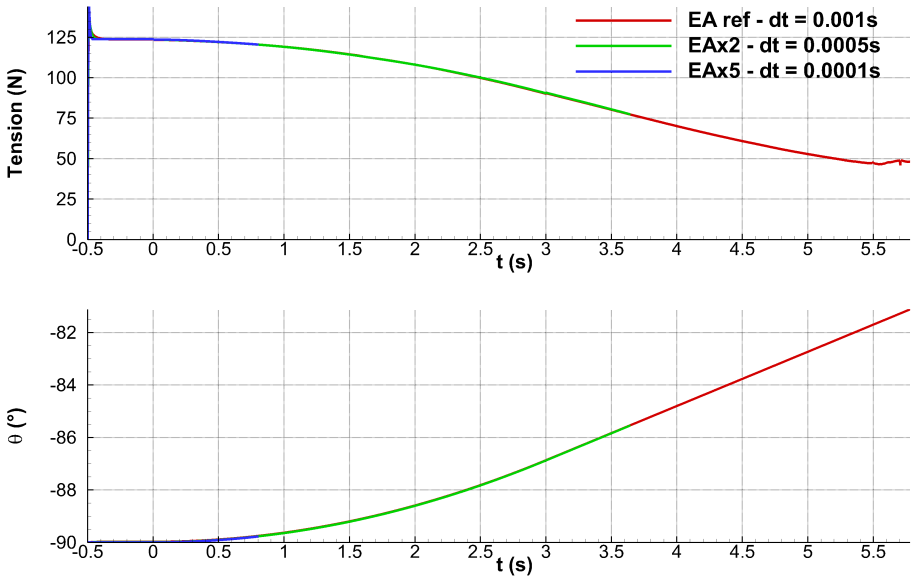


Figure VI.30 – Comparison of time series of the cable tension and the angular position from numerical results for Case 15 for a cable stiffness of 3.35×10^6 N. $t = 0$ s denotes the starting time of the winch.

Figure VI.31 presents the comparison of the angular position of the buoy and the cable tension between the experimental and numerical results. It can be seen the horizontality of the buoy at the initial time was not perfect during the experiments. Despite that, for both the motion and the tension, the agreement is very good. The tension decrease during the increase of the lowering velocity due to the ramp is well captured. The perturbed wave pattern at $t = 4.5$ s is shown in Figure VI.32.

The simulation only lasts 5.8 s because at that moment, one extremity of the buoy (bottom) is fully immersed whereas it was piercing at the initial time. Consequently the body mesh is tangent to the free surface mesh which leads to important numerical errors. The mesh at the final time step is displayed in Figure VI.33.

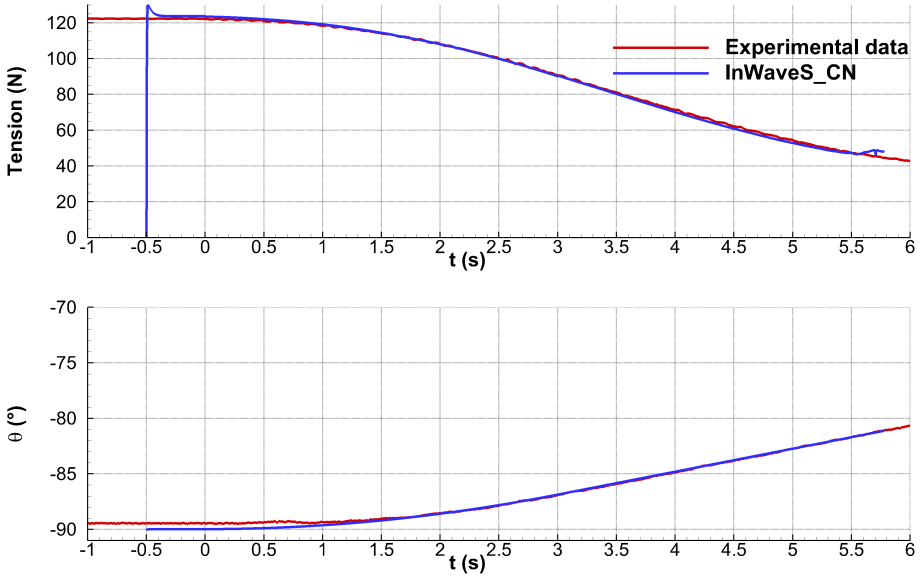


Figure VI.31 – Comparison of time series of the cable tension and the angular position from numerical and experimental results for Case 15. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

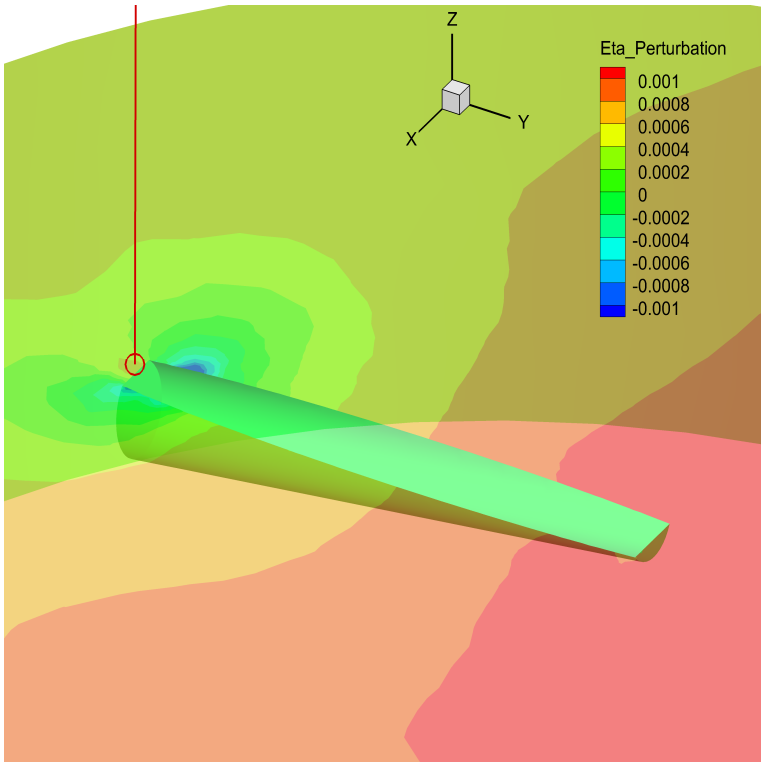


Figure VI.32 – Perturbed component of the wave pattern (η^P) at $t = 4.5$ s. The cable is in red.

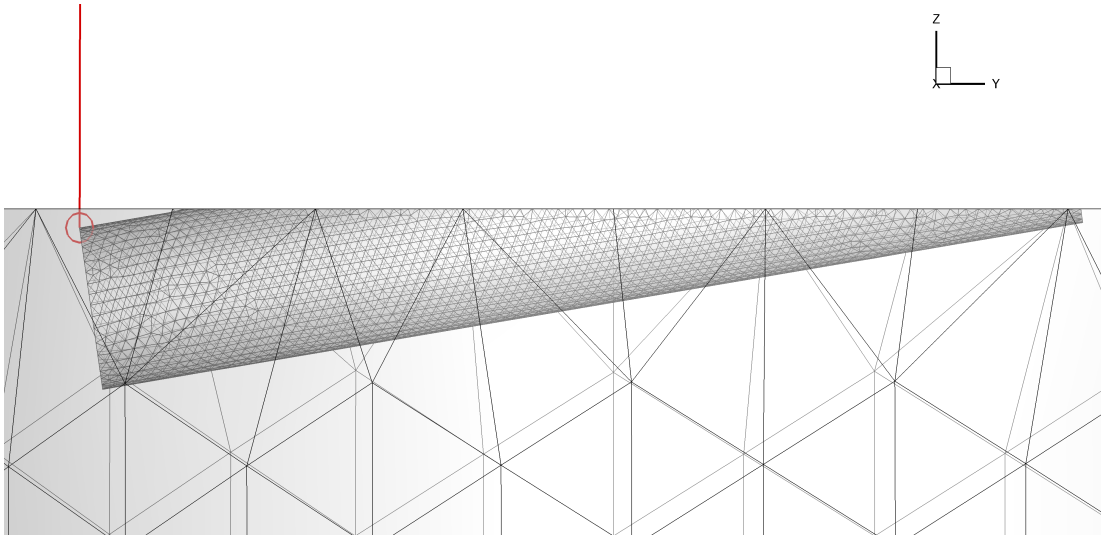


Figure VI.33 – Bi-dimensional view of the final mesh for Case 15. The cable is in red.

VI.2.2 Case 8

Case 8 is now studied. Compared to Case 15, the lowering velocity is double (0.067 m/s instead of 0.033 m/s) with a half linear ramp (1.5 s instead of 3 s). The position of the rotational axis is given in Table VI.5. The lowering velocity at the winch is presented in Figure VI.34. The cable tension and the angular position of the buoy are displayed in Figure VI.35. The initial tension is different from the experimental value due to the different initial position. The tension loss, after the activation of the winch, is well captured. As for Case 15, the simulation stops when one extremity of the buoy is close to be immersed.

j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	$\frac{\pi}{2}$	0.0116	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0

Table VI.5 – *mDH* parameters for Case 8

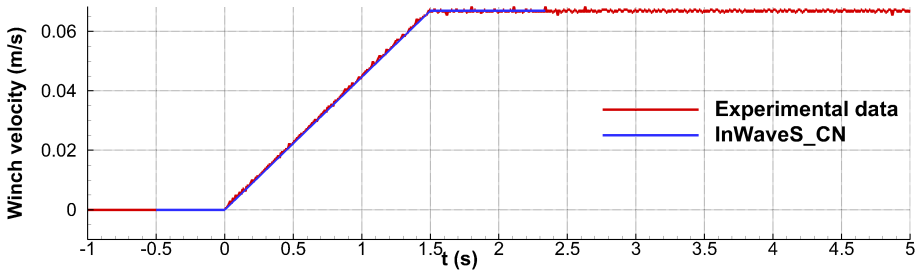


Figure VI.34 – Comparison of time series of the lowering velocity at the winch from numerical and experimental results for Case 8. $t = 0$ s denotes the starting time of the winch.

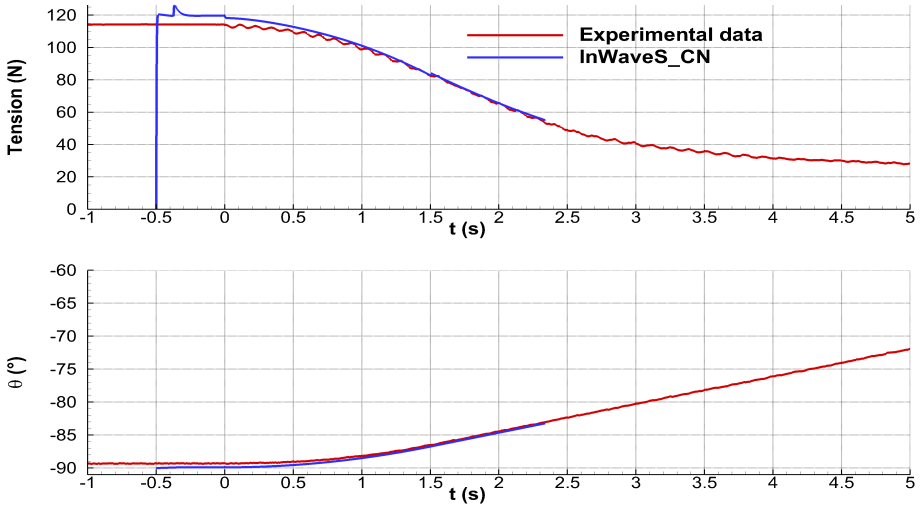


Figure VI.35 – Comparison of time series of the cable tension and the angular position from numerical and experimental results for Case 8. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

VI.2.3 Case 17

Case 17 is now considered. An incident regular wave of amplitude 0.01 m and wave period 1 s is present. The lowering velocity is 0.033 m/s with a linear ramp of 3 s. Because of the presence of the waves, the radius of the domain is fixed to two wave lengths (3 m) and the numerical absorbing to one wave length (1.5 m). The position of the rotational axis is given in Table VI.6. The lowering velocity at the winch is presented in Figure VI.36. The cable tension and the angular position of the buoy are displayed in Figure VI.37. A good agreement is observed between the experimental data and the numerical results. As with Cases 8 and 15, a difference is noticed in the initial tension. The effects of the waves is well captured for both the tension and the rotational motion. The tension decreases due to the winch and oscillates at the wave frequency. The angular motion is not affected by the waves. Once more, the simulation stops when one extremity of the buoy is close to be immersed.

j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	$\frac{\pi}{2}$	0.024	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0

Table VI.6 – *mDH* parameters for Case 17

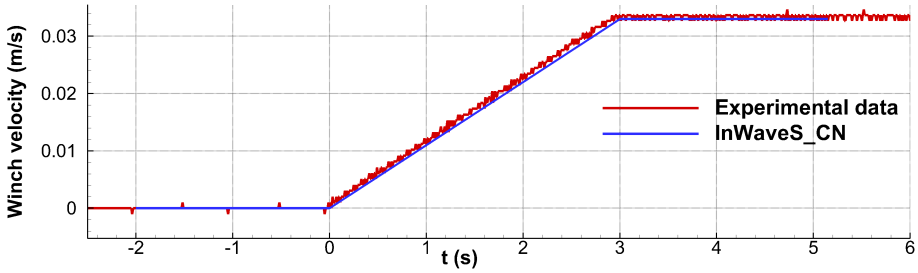


Figure VI.36 – Comparison of time series of the lowering velocity at the winch from numerical and experimental results for Case 17. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

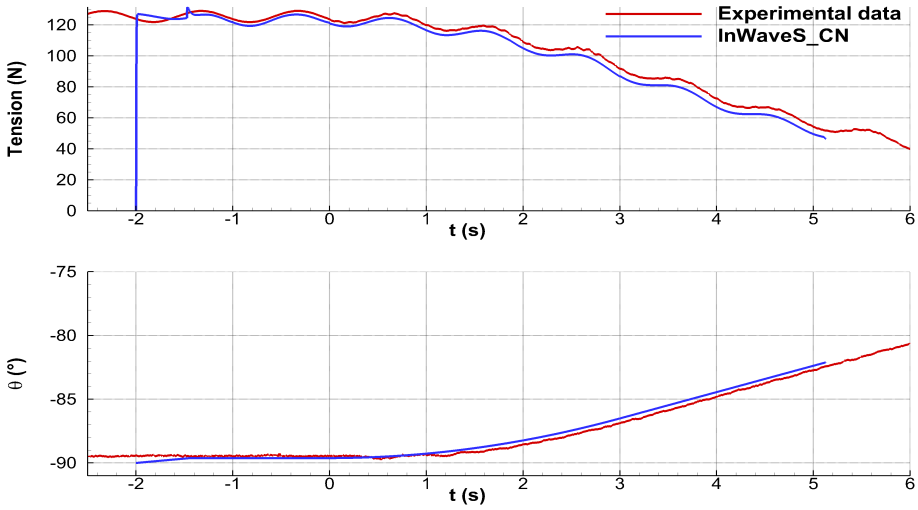


Figure VI.37 – Comparison of time series of the cable tension and the angular position from numerical and experimental results for Case 17. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

VI.2.4 Case 18

Case 18 allows to change the wave period compared to Case 17. It is fixed to 1.5 s . The other parameters are unchanged. The position of the rotational axis is given in Table VI.7. The lowering velocity at the winch is presented in Figure VI.38. The cable tension and the angular position of the buoy are displayed in Figure VI.39. A good agreement is observed between the experimental data and the numerical results with the modification of the wave period. The effect of the waves on the angular motion is still negligible. As for the other cases, the simulations stops when one extremity of the buoy is close to the free surface.

j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	$\frac{\pi}{2}$	0.019	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0

Table VI.7 – *mDH* parameters for Case 18

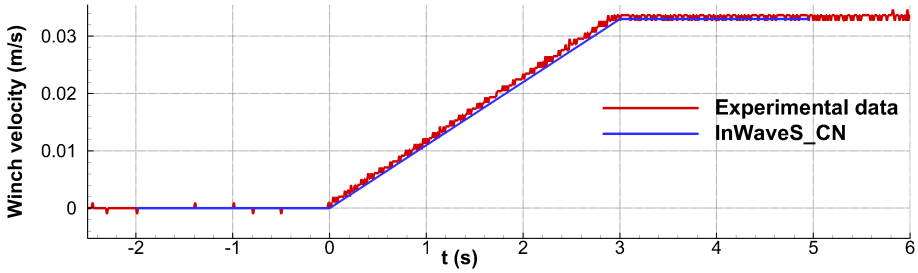


Figure VI.38 – Comparison of time series of the lowering velocity at the winch from numerical and experimental results for Case 18. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

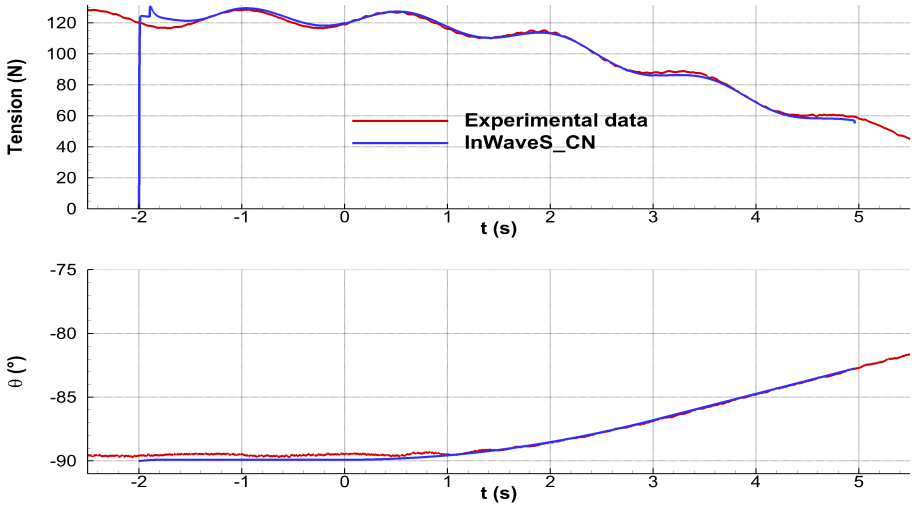


Figure VI.39 – Comparison of time series of the cable tension and the angular position from numerical and experimental results for Case 18. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

VI.2.5 Case 20

The effect of the wave amplitude may be studied from Case 20. Compared to Case 17, the wave amplitude is double. Other parameters (wave period, lowering velocity and ramp) stay identical. The position of the rotational axis is given in Table VI.8. The lowering velocity at the winch is presented in Figure VI.40. The cable tension and the angular position of the buoy are displayed in Figure VI.41. A good agreement is observed and the modification of the wave amplitude compared to Case 17 is well captured in the cable tension. The angular motion is not affected by the waves.

j	γ_j (rad)	b_j (m)	α_j (rad)	d_j (m)	θ_j (rad)	r_j (m)
1	$\frac{\pi}{2}$	0.020	$\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0

Table VI.8 – *mDH* parameters for Case 20

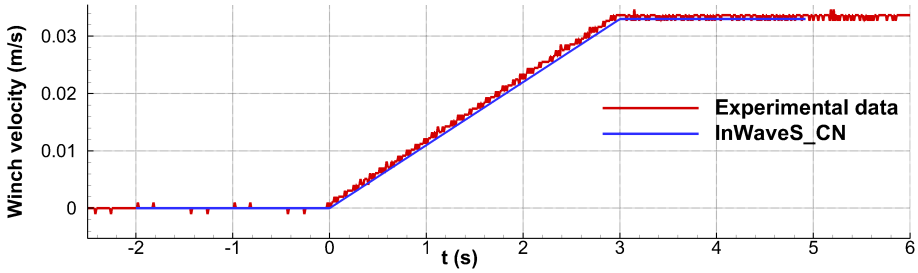


Figure VI.40 – Comparison of time series of the lowering velocity at the winch from numerical and experimental results for Case 20. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

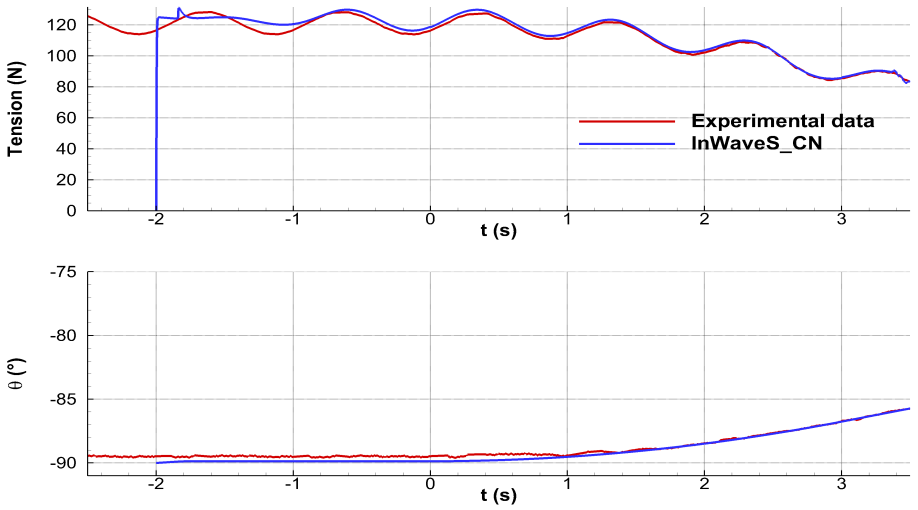


Figure VI.41 – Comparison of time series of the cable tension and the angular position from numerical and experimental results for Case 20. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

VI.3 Conclusion

This chapter presented the experiments of an upending operation conducted in the shallow water basin of *Ecole Centrale de Nantes* and the comparisons with the numerical results of *InWaveS_ CN*. All the modules constituting this numerical tool were required. Five cases were investigated, two cases without waves and with different lowering velocities and ramps, three cases with an incident regular wave with different wave parameters. Each comparison gave a good agreement between the experimental data and the numerical results, in term of cable tension and angular position of the buoy. The effects of the winch (velocity and linear ramp) and the waves were well captured. The difference in the initial horizontal position of the cylinder at the starting time could be responsible of the differences observed. In every case, the numerical simulation stopped when one extremity of the buoy was close to be immersed, leading to a body mesh tangent to the free surface mesh and therefore numerical errors. Nevertheless, these comparisons to experimental data has given a first and promising validation of *InWaveS_ CN*.

Conclusion and perspectives

The goal of our PhD work was the numerical simulation of marine operations, in particular of lowering and lifting operations. Such simulations require models of multibody dynamics, cable dynamics, hydrodynamics and fluid-structure interaction. This analysis has driven our project.

A multibody theory has been handled to simulate mechanical interactions in multibody systems. No kinematic loops were considered. Bodies were rigid, with six degrees of freedom, separated one from each other by a single-degree-of-freedom joint and located using relative coordinates (modified Denavit-Hartenberg parameters), except the base of the multibody system which was tracked with Cartesian coordinates. The direct dynamic algorithm was the Composite-Rigid-Body Algorithm (*CRBA*). Its implementation, in the numerical tool *InWave*, was done prior to the beginning of this PhD work.

The *CRBA* has been used to simulate cable dynamics. Cable elements were modelled using three joints (two revolute joints and one prismatic joint). The internal loads in the prismatic joint was modeled using a spring-damper system for simulating the stretching of the cable. No bending nor torsion were considered. This multibody model has been validated by comparison with the classical low-order lumped mass theory. This latter theory is based on the discretization of a cable into lumped masses linked to each other by a spring-damper system. The lumped mass theory has been implemented in a numerical tool, *CableDyn* and validated with the numerical and experimental data given by Buckham [64]. In order to speed-up the multibody solver when cables were simulated, a cable joint has been elaborated and the *CRBA* updated in that way. A reduction of 49.2 % of the CPU time was observed.

A winch model has been developed to simulate the lowering or the lifting of a payload. The approach was based on the adding or deletion of cable elements during the simulation. The cable element linked to the winch became variable-unstretched-length. Both the multibody and the lumped mass approaches provided the same results after comparisons.

In the literature, marine operations are mainly modelled using the linear potential flow theory for the computation of hydrodynamic loads. This theory assumes small amplitude motions of the bodies around their mean position and small steepness waves. In case of large (relative) amplitude motions and/or steep waves, this theory is not valid anymore. In contrast, the originality of this PhD thesis has been the use of an unsteady potential flow theory based on the weak-scatterer hypothesis to simulate marine operations. The velocity potential and the wave elevation were split into a known incident component and an unknown perturbed component. The perturbed quantities were assumed small in comparison to the incident ones. The free surface

boundary conditions were linearized around the incident free surface elevation. This approach is consistent in case of large (relative) amplitude motions and steep waves. The velocity potential was computed using a boundary element method. The fluid-structure interaction was formulated with a monolithic coupling using a second boundary element method to evaluate both the time-derivative of the velocity potential and the body acceleration. The implementation of this hydrodynamic theory, in the numerical tool *WS_ CN*, was already done before the start of our work in case of a single immersed or surface-piercing body.

The main contribution to *WS_ CN* has been the extension to multibody simulations with large motions. Both the mesh generator and the hydrodynamic solver have been modified accordingly. A first validation has been achieved in forced motion by comparison with the experimental data of Watai et al. [145]. Two cylinders were considered. One body was still while the second body was subject to a prescribed large amplitude motion. Comparisons were done based on the hydrodynamic loads of the fixed cylinder and the wave elevations at three wave probes. The weak-scatterer approach showed a very good agreement with the experiments, being able to simulate the hydrodynamic interactions between the floating bodies and the modulation of the hydrodynamic loads. A large number of panels were required to reach the converge of the numerical results, which led to important CPU times and computer memory demands for every simulation. Super-computing facilities were used. *WS_ CN* has also been switched to a body-exact formulation (linearization of the free surface boundary equations around the mean position) and a fully linear formulation (linearization of both the free surface and the body boundary equations around the mean position). The body-exact approach provided a good agreement with the experimental data too with a lower CPU time because of coarser meshes. The agreement was good due to the small steepness of the waves. When the wave steepness was increased, more differences were observed between the weak-scatterer and the body-exact models. The fully linear approach did not give accurate results, the body nonlinearities being important. Theses results have proved the interest of an unsteady weakly nonlinear potential flow theory, compared to a fully linear approach, in case of large relative motions between bodies.

A second validation of the multibody extension was proposed, based on a comparison between the weak-scatterer approach and a classical linear frequency-domain approach in free motion. This case presented a good agreement between the numerical results of the two theories. Other developments have been achieved in *WS_ CN*: the implementation of a free surface remeshing process to ensure more robust simulations, the parallelization of the computation of the influence coefficients to speed up the code and the reduction of the memory footprint with the use of the sparse linear system solver *PARDISO*.

Thus, our contribution to *WS_ CN* has enabled to reach multibody simulations which are more robust, stable and faster. From the comparisons between the different approximations included in the code (weak-scatterer, body-exact and linear), it seems relevant to us that *WS_ CN* is seen as a multi-level potential flow based numerical tool able to fit the hydrodynamic theory for every case. The weak-scatterer method should be used in case of large amplitude motion and steep waves. If the wave steepness is low then a body-exact approximation would be sufficient and a linear approximation would only be required when both the motion amplitude and the wave steepness are small.

The potential flow theory based on the weak-scatterer hypothesis and the Composite-Rigid-Body Algorithm have been coupled to solve the fluid-structure interaction and

enable the simulation of multibody systems in waves with large relative motions, as expected in marine operations. For the sake of robustness and CPU time, a tight coupling has been selected. The coupling equation, which gathered the multibody motion equation provided by the *CRBA* and the second boundary value problem of the potential flow theory, was derived and demonstrated. These developments have led to the creation of a new numerical tool, named *InWaveS_CN*. A significant emphasis was placed on its modularity through the language binding using Python as glue code language. A validation of this coupling has been done by the comparison between *InWaveS_CN* and *WS_CN*. A perfect agreement was noticed, giving the proof of the good derivation of the coupling equation. The tight coupling was tested by comparison with two loose couplings (explicit and implicit). As expected, the tight coupling matched the implicit loose coupling results and the explicit loose coupling presented some differences because of the time lag between the solvers.

The numerical tool *CableDyn* has been added as a module of *InWaveS_CN*, using an explicit loose coupling, to simulate cables when kinematic loops appeared, as they could not be handled by the *CRBA*.

The mesh generator of *WS_CN*, implemented prior to this PhD, was based on a semi-analytical approach to track the intersection curve between a surface-piercing body and the incident free surface. This approach was a marching method which required the knowledge of a parametric equation of the body surface. Then, an advance front method was used to generate the mesh of the bodies, the free surface and the numerical tank walls. This mesh strategy suffered of a lack of robustness and was only used with academic bodies. Furthermore, it would involve the development of a computer-aided design tool inside *WS_CN*, which is dedicated to the simulation and the analysis of wave-structure interaction, without using the capacity of a professional mesh generator. Consequently, it was decided to develop a new mesh generator based on a simple idea: *the body mesh for the user, the free surface mesh for the solver*. The body mesh was assumed to be obtained using a third-party mesh generator, then *WS_CN* had only to generate the free surface mesh. The inclusion of the body mesh in the free surface mesh was done with a panel cutting method, using the Python numerical tool *Meshmagick*, coupled with the advance front method of *WS_CN* through the language binding developed previously. A special care was taken to generate good quality meshes by the use of a panel merging algorithm and a spring analogy method to avoid tiny panels and optimize the position of the nodes. *Meshmagick* has become a new module of *InWaveS_CN*. Two validation test cases were presented based on comparisons between the initial mesh generator using *WS_CN* and the new mesh generator of *InWaveS_CN*. A very good agreement was noticed. Some differences were observed, probably due to the remeshing process used at every time step.

Finally, the experiments of an upending operation in waves were conducted in the shallow water basin of *Ecole Centrale de Nantes* and were presented in details. A horizontal cylinder in waves was kept horizontal using a cable, then the cable was unwound and the cylinder reached its vertical equilibrium position. *InWaveS_CN* was used to simulate the experiments. All the modules constituting this numerical tool were required: *InWave* for the multibody dynamics, *CableDyn* for the cable dynamics and the winch, *WS_CN* for the hydrodynamics with a body subject to a large amplitude motion and *Meshmagick* for the complex geometry. Five cases were presented: two cases without waves and with different lowering velocities and ramps, three cases with an incident

regular wave with different wave parameters. A good agreement was observed in term of buoy position and cable tension. The effects of the winch velocity, its linear ramp and the waves were well captured. But, when one extremity of the cylinder crossed the free surface, large numerical errors occurred. Nevertheless, these numerical simulations constituted the first and promising validation of *InWaveS_CN* with experiments of a marine operation.

This PhD work made effective progress in the field of the simulation of marine operations:

- The potential flow theory based on the weak-scatterer hypothesis has been extended to multibody simulations;.
- The interest of the weak-scatterer theory has been compared to other approaches: body-exact and linear, leading to a better understanding of its use;
- A multibody solver, including a cable model and a winch model, has been coupled to an unsteady potential flow based solver through a tight coupling;
- A new strategy of mesh generation based on a panel cutting method coupled with an advance front method has been developed and applied with an unsteady potential flow based solver;
- A consistent frame has been developed to simulate operations at sea with large relative motions and compared to experimental data.

From this PhD work and its conclusions, future works may be considered. The Composite-Rigid-Body Algorithm used in *InWave* cannot handle kinematic loops. As seen in the simulation of the experiments, it may be problematic. A more general mechanical solver could be useful for considering kinematic loops, contact-force models, flexible bodies, etc. This is the goal of the PhD of David Ogden which has coupled the frequency-domain potential flow-based solver *Nemoh* with the multibody solver *HotInt* [58]. A coupling of this multibody solver with *WS_CN* could lead to the simulation of more complex articulated systems with large relative motions.

The simulation of cables requires small time steps because of the presence of spring-damper systems. In *InWaveS_CN*, the time integrator is the fourth-order explicit Runge-Kutta scheme (*RK4*) with a constant time step and is shared by all modules. Consequently, the time step is fixed by the cable solver *CableDyn* or the multibody solver *InWave* whereas most of the CPU time is due to the hydrodynamic solver *WS_CN*. Simulations are too time-consuming uselessly when cables are present. This time step could be reduced by fitting the time integrator: for instance by using a Newmark's scheme to add numerical damping (cf. subsection II.1.2) in the mechanical and cable solvers only and by keeping the *RK4* scheme in *WS_CN* by simplicity or by changing the time integrator globally. Another possibility would be to add subcyles in the temporal loop to use a lower time step with the cable and mechanical solvers.

Bending and torsion loads could be added to the cable model in case of low-tension tethers, for example with a marine operation involving a Remotely Operated Vehicles (*ROV*) [64] or with a cable laying operation.

Motion compensation systems have not been considered in this PhD whereas they are present for some particular operations in the lowering and lifting operations. Therefore, their inclusion in the model is necessary for more realistic simulations.

Vincent Leroy has coupled *InWave* with two aerodynamic solvers to compute the aerodynamic loads affecting floating wind turbines: a free-vortex wake theory-based unsteady aerodynamic solver and a steady double multiple streamtube theory-based solver [56]. It would be interesting to gather the couplings presented in our PhD work with those developed by Vincent Leroy in order to simulate floating wind turbines subject to large amplitude motions in waves (because of the drift and/or the inclination of the turbine due to the wind for instance).

Regarding *WS_ CN*, several works could be planned. The code suffers of memory leaks in the marching method to track intersection curves and in the advance front method. They reduce the performance and the duration of the simulations. If these memory leaks cannot be fixed in the present implementation, a new implementation of the algorithms would be necessary. The CPU times required by *WS_ CN* are still large, even after the parallelization of the computation of the influence coefficients. This is why, the PhD work of Yohan Poirier has started in 2017 accordingly. He has achieved a *GPU* parallelization of the code and he now is developing a Parareal algorithm in *WS_ CN*. The Parareal algorithm is a parallel-in-time integration method [187]. The goal is to reach real-time computations with a single body. If the body-exact approximation is used, the implementation of the free surface Green's function could reduce the size of the mesh and so the CPU time. The expression of the free surface Green's function in the case of a weak-scatterer approximation does not exist yet. So far, *WS_ CN* has only handled incident regular waves, consequently, it would be interesting to extend the incident wave models to irregular waves, by using a High-Order Spectral (*HOS*) model [97] for instance. A fully nonlinear potential flow model could be implemented in *WS_ CN* in the cases where the weak-scatterer hypothesis is not satisfied. *WS_ CN* could also be extended to a second order of magnitude of the weak-scatterer hypothesis (on the same principle as the Stokes' series expansion used in the linear models) to add more free surface nonlinearities. Lots of application cases of the code have not been tested yet: ships with forward speed (including hydrofoils for a larger motion), side-by-side operations, slamming, etc.

The enhancement of the new mesh strategy, developed in this PhD, would require the improvement of the panel merging method to reach better quality meshes, especially at the intersection curve. The projection of the nodes on the intersection curve before clipping the mesh could also be a solution to study. A body mesh morphing algorithm should be developed to reduce the numerical errors due to the permanent remeshing and update the intersection curve. The CPU time would be smaller too. An extension of the new mesh generator to multibody simulations could be done easily.

InWaveS_ CN could be more intensively used in case of lowering and lifting operations in order to quantify with accuracy the interest of the weak-scatterer method. The effects of the distance between the payload and the vessel, the free surface or the seabed and of the lowering and lifting velocities could be studied for example. The extension to simulations with a payload in the air which is then immersed should be considered through the modeling of the slamming loads in *WS_ CN*. Finally, aerodynamic loads, neglected in our work whereas winds may be strong far offshore [6], could be included in the model to improve accuracy of the analyses performed for marine operations.

To the happy few

Bibliography

[1] C. Crabtree, D. Zappalá, and S. Hogg. Wind energy: UK experiences and offshore operational challenges. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 229(7):727–746, 2015. doi: 10.1177/0957650915597560.

[2] LEANWIND consortium. Driving cost reductions in offshore wind: The LEANWIND project final publication. Technical report, 2017.

[3] Z. Gao, A. Verma, Y. Zhao, Z. Jiang, and Z. Ren. A summary of the recent work at ntnu on marine operations related to installation of offshore wind turbines. In *Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering, OMAE2018*, 2018. doi: 10.1115/OMAE2018-78334.

[4] T. Jacobsen and B. J. Leira. Numerical and experimental studies of submerged towing of a subsea template. *Ocean Engineering*, 42:147–154, 2012. doi: 10.1016/j.oceaneng.2012.01.003.

[5] B. W. Nam, N. W. Kim, and S. Y. Hong. Experimental and numerical study on coupled motion responses of a floating crane vessel and a lifted subsea manifold in deep water. *International Journal of Naval Architecture and Ocean Engineering*, 9(5):552–567, 2017. doi: 10.1016/j.ijnaoe.2017.01.002.

[6] Z. Ren, Z. Jiang, R. Skjetne, and Z. Gao. Development and application of a simulator for offshore wind turbine blades installation. *Ocean Engineering*, 166:380–395, 2018. doi: 10.1016/j.oceaneng.2018.05.011.

[7] N. Ku and M.-I. Roh. Dynamic response simulation of an offshore wind turbine suspended by a floating crane. *Ships and Offshore Structures*, 10(6):621–634, 2015. doi: 10.1080/17445302.2014.942504.

[8] Det Norske Veritas. DNV-RP-H103: Modelling and analysis of marine operations. Technical report, 2014.

[9] J. Wang, L. Sun, Z. Deng, G. Ma, X. Zhu, and Z. Hou. Research on influence of horizontal offsets of underwater equipment based on pendulous installation method. In *ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering*, 2018. doi: 10.1115/OMAE2018-77734.

[10] H. Mork and J. Lunde. A cost-effective and safe method and for transportation and installation of subsea and structures - the pencil and buoy method. In *Offshore Europe*, 2007. doi: 10.2118/108608-MS.

[11] M. A. Hannan. *Numerical simulation of submerged payload coupled with crane barge in waves.* PhD thesis, National university of Singapore, 2014.

[12] R. E. Elling and A. McClinton. Dynamic loading of shipboard cranes. In *IEEE International Conference on Engineering in the Ocean Environment*, pages 174–177, 1973. doi: 10.1109/OCEANS.1973.1161241.

[13] C. Chin, A. H. Nayfeh, and E. Abdel-Rahman. Nonlinear dynamics of a boom crane. *Journal of Vibration and Control*, 7(2):199–220, 2001. doi: 10.1177/107754630100700204.

[14] N. Nojiri and T. Sasaki. Motion characteristics of crane vessels in lifting operation. In *Offshore technology Conference*, 1983. doi: 10.4043/4603-MS.

[15] T. E. Schellin, T. Jiang, and S. D. Sharma. Crane ship response to wave groups. *Journal of Offshore Mechanics and Arctic Engineering*, 113(3):211–218, 1991. doi: 10.1115/1.2919922.

[16] T. E. Schellin, T. Jiang, and C. Ostergaard. Response analysis and operating limits of crane ships. *Journal of Ship Research*, 37(3):225–238, 1993.

[17] K. Ellermann, E. Kreuzer, and M. Markiewicz. Nonlinear dynamics of floating cranes. *Nonlinear Dynamics*, 27(2):107–183, 2002. doi: 10.1023/A:1014256405213.

[18] K. Ellermann and E. Kreuzer. Nonlinear dynamics in the motion of floating cranes. *Multibody System Dynamics*, 9(4):377–387, 2003. doi: 10.1023/A:1023361314261.

[19] H. J. J. Van Den Boom, J. N. Dekker, and R. P. Dallinga. Computer analysis of heavy lift operations. In *Offshore Technology Conference*, 1988. doi: 10.4043/5819-MS.

[20] J. A. Witz. Parametric excitation of crane loads in moderate sea states. *Ocean Engineering*, 22(4):411–420, 1995. doi: 10.1016/0029-8018(94)00015-Y.

[21] S. Malenica, M. Orozco, and X.-B. Chen. Some aspects of seakeeping of the floating body with attached pendulum. In *Proceedings of the 11th International Congress of the International Maritime Association of the Mediterranean*, 2005.

[22] W. Bai, M. A. Hannan, and K. K. Ang. Numerical simulation of fully nonlinear wave interaction with submerged structures: Fixed or subjected to constrained motion. *Journal of Fluids and Structures*, 49:534–553, 2014. doi: 10.1016/j.jfluidstructs.2014.05.011.

[23] M.A. Hannan and W. Bai. Analysis of nonlinear dynamics of fully submerged payload hanging from offshore crane vessel. *Ocean Engineering*, 128:132–146, 2016. doi: 10.1016/j.oceaneng.2016.10.030.

[24] M. Bashir, M. Evans, S. Benson, and A. Murphy. Simulations of dynamic interaction between a bluff body and installation vessel during launch and recovery in rough seas. In *Proceedings of the ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering, OMAE2017*, 2017. doi: 10.1115/OMAE2017-61319.

[25] J.-H. Cha, M.-I. Roh, and K.-Y. Lee. Dynamic response simulation of heavy cargo suspended by a floating crane based on multibody system dynamics. *Ocean Engineering*, 37(14):1273–1291, 2010. doi: 10.1016/j.oceaneng.2010.06.008.

[26] N. Ku and S. Ha. Dynamic response analysis of heavy load lifting operation in shipyard using multi-cranes. *Ocean Engineering*, 83:63–75, 2014. doi: 10.1016/j.oceaneng.2014.03.026.

[27] S.-H. Ham, M.-I. Roh, H. Lee, and S. Ha. Multibody dynamic analysis of a heavy load suspended by a floating crane with constraint-based wire rope. *Ocean Engineering*, 109:145–160, 2015. doi: 10.1016/j.oceaneng.2015.08.050.

[28] H.-L. Ren, X.-L. Wang, Y.-J. Hu, and C.-G. Li. Dynamic response analysis of a moored crane-ship with a flexible boom. *Journal of Zhejiang University - Science A: Applied Physics & Engineering*, 9(1):26–31, 2008. doi: 10.1631/jzus.A071308.

[29] K.-P. Park, J.-H. Cha, and K.-Y. Lee. Dynamic factor analysis considering elastic boom effects in heavy lifting operations. *Ocean Engineering*, 38(10):1100–1113, 2011. doi: 10.1016/j.oceaneng.2011.04.007.

[30] J.-H. Cha, K.-P. Park, and K.-Y. Lee. Development of a simulation framework and applications to new production processes in shipyards. *Computer-Aided Design*, 44(3):241–252, mar 2012. doi: 10.1016/j.cad.2011.06.010.

[31] Seung-Ho Ham, Myung-Il Roh, Hyewon Lee, Jin-Wuk Hong, and Hong-Rae Lee. Development and validation of a simulation-based safety evaluation program for a mega floating crane. *Advances in Engineering Software*, 112:101–116, oct 2017. doi: 10.1016/j.advengsoft.2017.04.009.

[32] Kyu-Yeul Lee, Ju-Hwan Cha, and Kwang-Phil Park. Dynamic response of a floating crane in waves by considering the nonlinear effect of hydrostatic force. *Ship Technology Research*, 57(1):64–73, jan 2010. doi: 10.1179/str.2010.57.1.006.

[33] J. K. Woodacre, R. J. Bauer, and R. A. Irani. A review of vertical motion heave compensation systems. *Ocean Engineering*, 104:140–154, 2015. doi: 10.1016/j.oceaneng.2015.05.004.

[34] Zhengru Ren, Zhiyu Jiang, Zhen Gao, and Roger Skjetne. Active tugger line force control for single blade installation. *Wind Energy*, 21:1344–1358, 2018. doi: 10.1002/we.2258.

[35] M. Kimiaei, X. Jiajing, and H. Yu. Comparing the results of a simplified numerical model with DNV guidelines for installation of subsea platforms. In *Proceedings of the ASME 2009 28th International Conference on Ocean, Offshore and Arctic Engineering, OMAE2009*, volume 1, pages 319–326, 2009. doi: 10.1115/OMAE2009-79356.

[36] *SIMO - Theory Manual Version 3.6, rev: 2*, 2009.

[37] *aNySIM Time Domain Analysis of Multi Body Dynamics for Offshore Operations.*

[38] *OrcaFlex Manual v10.0a.* Orcina Ltd., 2015.

[39] *DeepLines 4.5 Theory Manual.* Principia and IFP Energies Nouvelles, 2011.

[40] *ProteusDS Manual v2.45.* Dynamic Systems Analysis Ltd.

[41] *Ariane8 theoretical manual.* Bureau Veritas, 2015.

[42] *Reference Manual for MOSES Version 7.07*, 2013.

[43] F. Rongère and A. H. Clément. Systematic dynamic modeling and simulation of multibody offshore structures: application to wave energy converters. In *Proceedings of the ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering, OMAE2013*, 2013.

[44] L. Letournel. *Développement d'un outil de simulation numérique basé sur l'approche weak-scatterer pour l'étude des systèmes houlomoteurs en grands mouvements.* PhD thesis, Ecole Centrale de Nantes, 2015.

[45] C. Chauvigné. *Tenue à la mer d'un flotteur animé de grands mouvements pour les Energies Marines Renouvelables.* PhD thesis, Ecole Centrale de Nantes, 2016.

[46] A. Combourieu, M. Philippe, F. Rongère, and A. Babarit. InWave: a new flexible design tool dedicated to wave energy converters. In *Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Artic Engineering, OMAE2014*, 2014. doi: 10.1115/OMAE2014-24564.

[47] A. Babarit and G. Delhommeau. Theoretical and numerical aspects of the open source BEM solver NEMOH. In *Proceedings of the 11th European Wave and Tidal Energy Conference, EWTEC2015*, 2015.

[48] R. Featherstone. *Rigid body dynamics algorithms.* Springer, 2008. doi: 10.1007/978-1-4899-7560-7.

[49] R. Featherstone and D. Orin. Robot dynamics: Equations and algorithms. In *Proceedings of the 2000 ICRA IEEE International Conference on Robotics and Automation*, volume 1, 2000. doi: 10.1109/ROBOT.2000.844153.

[50] A. A. Shabana. *Computational dynamics Second Edition.* John Wiley & Sons, Ltd, 2001. doi: 10.1002/9780470686850.

[51] P. Masarati. *Comprehensive multibody aeroservoelastic analysis of integrated rotorcraft active controls.* PhD thesis, Politecnico di Milano, 1999.

[52] B. Siciliano and O. Khatib. *Handbook of robotics.* Springer-Verlag, 2008. doi: 10.1007/978-3-540-30301-5.

[53] D. Negrut. *On the implicit integration of differential-algebraic equations of multibody dynamics.* PhD thesis, University of Iowa, 1998.

[54] T. K. Uchida. *Real-time dynamic simulation of constrained multibody systems using symbolic computation.* PhD thesis, University of Waterloo (Canada), 2011.

[55] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1(1):1–16, 1972. doi: 10.1016/0045-7825(72)90018-7.

[56] V. Leroy, J.-C. Gilloteaux, M. Philippe, A. Babarit, and P. Ferrant. Development of a simulation tool coupling hydrodynamics and unsteady aerodynamics to study floating wind turbines. In *Proceedings of the ASME 2017 36th International Conference on Ocean, Offshore and Artic Engineering, OMAE2017*, volume 10, 2017. doi: 10.1115/OMAE2017-61203.

[57] V. Leroy, J.-C. Gilloteaux, A. Combourieu, A. Babarit, and F. Ferrant. Impact of the aerodynamic model on the modelling of the behaviour of a floating vertical axis wind turbine. In *EERA DeepWind*, 2018.

[58] D. Ogden, R. Pascal, A. Combourieu, D. Forehand, L. Johanning, and Z.-M. Yuan. New mechanical and features for time-domain and WEC modelling and in InWave. In *Proceedings of the 7th International Conference on Ocean Energy, ICOE2018*, 2018.

[59] W. Khalil and J.-F. Kleinfinger. A new geometric notation for open and closed-loop robots. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, volume 3, 1986. doi: 10.1109/ROBOT.1986.1087552.

[60] F. Rongère. *Simulation dynamique des systèmes Bateau-Avirons-Rameur(s).* PhD thesis, Ecole Centrale de Nantes, 2012.

[61] V. Leroy, A. Combourieu, M. Philippe, A. Babarit, and F. Rongère. Benchmarking of the new design tool InWave on a selection of wave energy converters from NumWEC project. In *Proceedings of the Asian Wave and Tidal Energy Conference, AWTEC2014*, 2014.

[62] F. Rongère. Modèle dynamique des système multicorps flottants avec prise en compte des interactions hydrodynamiques entre les corps (in french). Technical report, 2014.

[63] C. Spraul, P. Arnal, V. andCartraud, and C. Berhault. Parameter calibration in dynamic simulations of power cables in shallow water to improve fatigure damage estimation. In *Proceedings of the ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering, OMAE2017*, 2017. doi: 10.1115/OMAE2017-61821.

[64] B. J. Buckham. *Dynamics Modelling of Low-Tension Tethers for Submerged Remotely Operated Vehicles.* PhD thesis, University of Victoria (Canada), 2003.

[65] J. Davidson and J. V. Ringwood. Mathematical modelling of mooring systems for wave energy converters - A review. *Energies*, 10(5), may 2017. doi: 10.3390/en10050666.

[66] W.-T. Hsu, K. P. Thiagarajan, M. Hall, M. MacNicoll, and R. Akers. Snap loads on mooring lines of a floating offshore wind turbine structure. In *Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering, OMAE2014*, 2014. doi: 10.1115/OMAE2014-23587.

[67] M. Masciola, J. Jonkman, and A. Robertson. Implementation of a multisegmented, quasi-static cable model. In *Proceedings of the Twenty-third International Offshore and Polar Engineering Conference, ISOPE2013*, 2013.

[68] M. Borg, M. Collu, and A. Kolios. Offshore floating vertical axis wind turbines, dynamics modelling state of the art. part II: Mooring line and structural dynamics. *Renewable and Sustainable Energy Reviews*, 39:1226–1234, 2014. doi: 10.1016/j.rser.2014.07.122.

[69] S. Gueydon and J. Jonkman. Update on the comparison of second-order loads on a tension leg platform for wind turbines. In *Proceedings of the 2016 Twenty-sixth International Ocean and Polar Engineering Conference, ISOPE2016*, 2016.

[70] M. Hall and A. Goupee. Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data. *Ocean Engineering*, 104:590–603, 2015. doi: 10.1016/j.oceaneng.2015.05.035.

[71] M. Masciola, J. Jonkman, and A. Robertson. Extending the capabilities of the mooring analysis program: A survey of dynamic mooring line theories for integration into FAST. In *Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering, OMAE2014*, 2014. doi: 10.1115/OMAE2014-23508.

[72] F. R. Driscoll, R. G. Lueck, and M. Nahon. Development and validation of a lumped-mass dynamics model of a deep-sea ROV system. *Applied Ocean Research*, 22(3):169–182, 2000. doi: 10.1016/S0141-1187(00)00002-X.

[73] G. Vissio, B. Passione, M. Hall, and M. Raffero. Expanding ISWEC modelling with a lumped-mass mooring line model. In *Proceedings of the 11th European Wave and Tidal Energy Conference, EWTEC2015*, 2015.

[74] D. L. Garrett. Dynamic analysis of slender rods. *Journal of Energy Resources Technology*, 104(4):302–306, 1982. doi: 10.1115/1.3230419.

[75] S. Huang. Dynamic analysis of three dimensional marine cables. *Ocean Engineering*, 21(6):587–605, 1994. doi: 10.1016/0029-8018(94)90008-6.

[76] J. I. Gobat. *The Dynamics of Geometrically Compliant Mooring Systems*. PhD thesis, Massachusetts Institute of Technology, 2000.

[77] M. Servin and C. Lacoursière. Rigid body cable for virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):783–796, 2008. doi: 10.1109/TVCG.2007.70629.

[78] M. Servin and C. Lacoursière. Massless cable and for real-time and simulation. *Computer Graphics Forum*, 26(2):172–184, 2007. doi: 10.1111/j.1467-8659.2007.01014.x.

[79] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959.

[80] S. Kontoe. *Development of time integration schemes and advanced boundary conditions for dynamic geotechnical analysis*. PhD thesis, University of London, 2006.

[81] T. C. Fung. Numerical dissipation in time-step integration algorithms for structural dynamic analysis. *Progress in Structural Engineering and Materials*, 5(3):167–180, 2003. doi: 10.1002/pse.149.

[82] H. M. Hilber, T. J. R. Hugues, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5(3):283–292, 1977. doi: 10.1002/eqe.4290050306.

[83] W. L. Wood, M. Bossak, and O. C. Zienkiewicz. An alpha modification of Newmark's method. *International Journal for Numerical Methods in Engineering*, 15 (10):1562–1566, 1980. doi: 10.1002/nme.1620151011.

[84] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-alpha method. *Journal of Applied Mechanics*, 60(2):371–375, 1993. doi: 10.1115/1.2900803.

[85] N. Mahjoubi. *Méthode générale de couplage de schéma d'intégration multi-échelles en temps en dynamique des structures (In french)*. PhD thesis, INSA de Lyon, 2010.

[86] M. Hall. *MoorDyn User's Guide*, 2017.

[87] M. Hall. Efficient modelling of seabed friction and multi-floater mooring systems in MoorDyn. In *Proceedings of the 12th European Wave and Tidal Energy Conference, EWTEC2017*, 2017.

[88] J. R. Morison, M. P. O'Brien, J. W. Johnson, and S. A. Schaaf. The force exerted by surface waves on piles. *Petroleum Transactions*, 189:149–154, 1950. doi: 10.2118/950149-G.

[89] M. A. Vaz and M. H. Patel. Transient behaviour of towed marine cables in two dimensions. *Applied Ocean Research*, 17(3):143–153, 1995. doi: 10.1016/0141-1187(95)00012-7.

[90] P.-Y. Wuillaume, F. Rongère, A. Babarit, M. Philippe, and P. Ferrant. Development and adaptation of the composite rigid body algorithm and the weak-scatterer approach in view of the modeling of marine operations. In *Proceedings of the 23ème Congrès Français de Mécanique, CFM2017*, 2017.

[91] J. W. Kamman and R. L. Huston. Modeling of variable length towed and tethered cable systems. *Journal of Guidance Control and Dynamics*, 22(4):602–608, 1999. doi: 10.2514/2.4423.

[92] S. Prabhakar and B. Buckham. Dynamics modeling and control of a variable length remotely operated vehicle tether. In *Proceedings of OCEANS 2005 MTS/IEEE*, 2005. doi: 10.1109/OCEANS.2005.1639927.

[93] A. K. Banerjee and V. N. Do. Deployment control of a cable connecting a ship to an underwater vehicle. *Journal of Guidance, Control, and Dynamics*, 17(6):1327–1332, 1994. doi: 10.2514/3.21351.

[94] P.-H. Wang, R.-F. Fung, and M.-J. Lee. Finite element analysis of a three-dimensional underwater cable with time-dependent length. *Journal of Sound and Vibration*, 209(2):223–249, 1998. doi: 10.1006/jsvi.1997.1227.

[95] J. D. M. Zand, B. J. Buckham, D. Steinke, and D. Constantienescu. Ship and winch regulation for remotely operated vehicle waypoint navigation. 19(3), 2009.

[96] M. Borg, A. Shires, and M. Collu. Offshore floating vertical axis wind turbines, dynamics modelling state of the art. part I: Aerodynamics. *Renewable and Sustainable Energy Reviews*, 39:1214–1225, 2014. doi: 10.1016/j.rser.2014.07.096.

[97] G. Ducrozet, F. Bonnefoy, D. Le Touzé, and P. Ferrant. 3-D HOS simulations of extreme waves in open seas. *Natural hazards and earth system sciences*, 7:109–122, 2007. doi: 10.5194/nhess-7-109-2007.

[98] P. Ferrant, D. Le Touzé, and K. Pelletier. Non-linear time-domain models for irregular wave diffraction about offshore structures. *International Journal for Numerical Methods in Fluids*, 43(10):1257–1277, 2003.

[99] W. Bai and R. Eatock Taylor. Fully nonlinear simulation of wave interaction with fixed and floating flared structures. *Ocean Engineering*, 36(3):223–236, 2009. doi: 10.1016/j.oceaneng.2008.11.003.

[100] R. A. Watai. *A time-domain boundary elements method for the seakeeping analysis of offshore systems.* PhD thesis, University of Sao Paulo, 2015.

[101] J.-C. Gilloteaux. *Mouvements de grande amplitude d'un corps flottant en fluide parfait. Application à la récupération de l'énergie des vagues.* PhD thesis, Ecole Centrale de Nantes, 2007.

[102] W. E. Cummins. The impulse response function and ship motions. Technical report, Department of the Navy, 1962.

[103] J. A. Pinkster. *Low Frequency Second Order Wave Exciting Forces on Floating Structures.* PhD thesis, Technical University Delft, 1980.

[104] T. Bunnik. A simulation approach for large relative motions of multi-body offshore operations in waves. In *Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering, OMAE2014*, 2014.

[105] M.A. Hannan and W. Bai. Nonlinear hydrodynamic responses of submerged moving payload in vicinity of a crane barge in waves. *Marine Structures*, 41:154–179, 2015. doi: 10.1016/j.marstruc.2015.01.002.

[106] J. S. Pawlowski and D. W. Bass. A theoretical and numerical model of ship motions in heavy seas. *SNAME Transactions*, 99, 1991.

[107] J. S. Pawlowski. On the application of the weak-scatterer hypothesis to the prediction of ship motions in heavy seas. 1992.

[108] J. S. Pawlowski. A nonlinear theory of ship motion in waves. In *Proceedings of the 19th Symposium on Naval Hydrodynamics*, 1992.

[109] W. M. Lin, M. Meinhold, N. Salvensen, and D. K. Yue. Large amplitude motions and wave loads for ship design. In *Proceedings of the 20th Symposium Naval Hydrodynamics*, 1994.

[110] W. M. Lin, M. Collette, D. Lavis, S. Jessup, and J. Kuhn. Recent hydrodynamic and tool development and validation for and motions and slam and loads on ocean-going and high-speed vessels. In *Proceedings of the 10th International Symposium on Practical Design of Ships and Other Floating Structures*, 2007.

[111] D. C. Kring, Y. Huang, P. D. Sclavounos, T. Vada, and A. Braathen. Nonlinear ship motions and wave-induced loads by a Rankine method. In *Proceedings of the 21th Symposium on Naval hydrodynamics*, 1996.

[112] Y. Huang. *Nonlinear ship motions by a Rankine panel method.* PhD thesis, Massachusetts Institute of Technology, 1997.

[113] G. J. Grigoropoulos, C. Katsikis, and D. S. Chalkias. Experimental verification of the linear and non-linear versions of a panel code. *International Journal of Naval Architecture and Ocean Engineering*, 3(1):27–36, mar 2011. doi: 10.3744/JNAOE.2011.3.1.027.

[114] Y. Kim, K.-H. Kim, J.-H. Kim, T. Kim, M.-G. Seo, and Y. Kim. Time-domain analysis of nonlinear motion responses and structural loads on ships and offshore structures: development of WISH programs. *International Journal of Naval Architecture and Ocean Engineering*, 3(1):37–52, 2011. doi: 10.2478/ijnaoe-2013-0044.

[115] K.-H. Kim and Y. Kim. Time-domain analysis of nonlinear ship motion responses based on weak-scatterer hypothesis. In *Proceedings of the Nineteenth International Offshore and Polar Engineering Conference, ISOPE2009*, 2009.

[116] J. G. Bretl. *A time domain model for wave induced motions coupled to energy extraction.* PhD thesis, University of Michigan, 2009.

[117] L. Letournel, C. Chauvigné, B. Gelly, A. Babarit, G. Ducrozet, and P. Ferrant. Weakly nonlinear modeling of submerged wave energy converters. *Applied Ocean Research*, 75:201–222, 2018. doi: 10.1016/j.apor.2018.03.014.

[118] M.-J. Song, K.-H. Kim, and Y. Kim. Numerical analysis and validation of weakly nonlinear ship motions and structural loads on a moden containership. *Ocean Engineering*, 38(1):77–87, 2011. doi: 10.1016/j.oceaneng.2010.09.017.

[119] M. Greco and C. Lugni. 3-D seakeeping analysis with water on deck and slamming. Part 1: Numerical solver. *Journal of Fluids and Structures*, 33:127–147, 2012. doi: 10.1016/j.jfluidstructs.2012.04.005.

[120] F. Ruggeri, R. A. Watai, and A. N. Simos. A higher order time domain rankine panel method for linear and weakly non-linear computation. In *Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering, OMAE2015*, 2015. doi: 10.1115/OMAE2015-42234.

[121] G. Fourey, C. Hermange, D. Le Touzé, and G. Oger. An efficient FSI coupling strategy between smoothed particle hydrodynamics and finite element methods. *Computer Physics Communications*, 217:66–81, 2016. doi: 10.1016/j.cpc.2017.04.005.

[122] W. Koo and M.-H. Kim. Freely floating-body simulation by a 2D fully nonlinear numerical wave tank. *Ocean Engineering*, 31(16):2011–2046, 2004. doi: 10.1016/j.oceaneng.2004.05.003.

[123] P. J. Bandyk and R. F. Beck. The acceleration potential in fluid-body interaction problems. *Journal of Engineering Mathematics*, 70(1-3):147–163, 2011. doi: 10.1007/s10665-010-9446-0.

[124] E. Guerber. *Modélisation numérique des interactions non-linéaires entre vagues et structures immergées, appliquée à la simulatio de systèmes houlomoteurs.* PhD thesis, Université Paris-Est, 2011.

[125] Y. Cao, R. F. Beck, and W. W. Schultz. Nonlinear computation of wave loads and motions of floating bodies in incident waves. In *Proceedings of the 9th International Workshop on Water waves and Floating Bodies*, 1994.

[126] G. X. Wu and R. Eatock Taylor. Transient motion of a floating body in steep water waves. In *Proceedings of the 11th International Workshop Water Waves Floating Bodies*, 1996.

[127] G. X. Wu and R. Eatock Taylor. The coupled finite element and boundary and element analysis of nonlinear interactions between waves and bodies. *Ocean Engineering*, 30(3):387–400, 2003. doi: 10.1016/S0029-8018(02)00037-9.

[128] Y. Li. *Fully nonlinear numerical simulations of wave interactions with multiple structures at resonance.* PhD thesis, University College London, 2017.

[129] R. Cointe, P. Geyer, B. King, B. Molin, and M. Tramoni. Nonlinear and linear motions of a rectangular barge in a perfect fluid. In *Proceedings of the 18th Symposium on naval hydrodynamics*, pages 85–99, 1990.

[130] E. F. G. Van Daalen. *Numerical and theoretical studies of water waves and floating bodies.* PhD thesis, University of Twente, 1993.

[131] K. Tanizawa. A nonlinear simulation method of 3-D body motions in waves (1st report). 178(178):179–191, 1995. doi: 10.2534/jjasnaoe1968.1995.178_179.

[132] P. J. F. Berkvens. *Floating bodies interacting with water waves: devlopment of a time-domain panel method.* PhD thesis, University of Twente, 1998.

[133] L. Letournel, G. Ducrozet, A. Babarit, and P. Ferrant. Proof of the equivalence of Tanizawa-Berkvens' and Cointe-van Daalen's formulations for the time derivative of the velocity potential for non-linear potential flow solvers. *Applied Ocean Research*, 63:184–199, 2017. doi: 10.1016/j.apor.2017.01.010.

[134] J. T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, 28(8):1381–1388, 1990. doi: 10.2514/3.25229.

[135] A. Leroyer. *Etude du couplage écoulement / mouvement pour des corps solides ou à déformations imposée par résolution des équations de Navier-Stokes. Contribution à la modélisation numérique de la cavitation.* PhD thesis, Ecole Centrale de Nantes, 2004.

[136] E. Jacquin. *Navire autopropulsé en manoeuvres : simulation numérique et optimisation des performances hydrodynamiques.* PhD thesis, Ecole Centrale de Nantes, 2007.

[137] F. J. Blom. Considerations on the spring analogy. *International Journal for Numerical Methods in Fluids*, 32(6):647–668, 2000. doi: 10.1002/(SICI)1097-0363(20000330)32:6<647::AID-FLD979>3.0.CO;2-K.

[138] J. Schmidt and B. Stoevesandt. Dynamic mesh optimization based on the spring analogy. In *ITM Web of Conferences*, volume 2, 2014. doi: 10.1051/itmconf/20140203001.

[139] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163:231–245, 1998. doi: 10.1016/S0045-7825(98)00016-4.

[140] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(3):305–316, 2002. doi: 10.1016/S0045-7949(02)00002-0.

[141] M. M. Selim and R. P. Koomullil. Mesh deformation approaches – a survey. *Journal of Physical Mathematics*, 7(2), 2016. doi: 10.4172/2090-0902.1000181.

[142] A. de Boer, M. S. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers and Structures*, 85(11):784–795, 2007. doi: 10.1016/j.compstruc.2007.01.013.

[143] M. M. Rienecker and J. D. Fenton. A Fourier approximation method for steady water waves. *Journal of Fluid Mechanics*, 104:119–137, 1981. doi: 10.1017/S0022112081002851.

[144] L. Letournel. Quantification de l'effet du lissage Gaussien ajouté dans le code WS. Technical report, Ecole Centrale de Nantes, 2017.

[145] R. A. Watai, F. Ruggeri, and A. N. Simos. A new time domain Rankine panel method for simulations involving multiple bodies with large relative displacements. *Applied Ocean Research*, 59:93–114, 2016. doi: 10.1016/j.apor.2016.05.002.

[146] C. A. Felippa and K. C. Park. Staggered transient analysis procedures for coupled mechanical systems: Formulation. *Computer Methods in Applied Mechanics and Engineering*, 24(1):61–111, 1980. doi: 10.1016/0045-7825(80)90040-7.

[147] J-F Sigrist. Méthodes numériques de calculs couplés fluide/structure - cas du fluide stagnant : introduction. Technical report, Techniques de l'ingénieur, 2010.

[148] K. C. Park, C. A. Felippa, and J. A. Deruntz. Stabilization of staggered solution procedures for fluide-structure interaction analysis. *Computational Methods for Fluid-Structure Interaction Problems*, 26:94–124, 1977.

[149] C. A. Felippa, K. C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001. doi: 10.1016/S0045-7825(00)00391-1.

[150] Z. Li. *Développement d'une méthode de simulation de couplage fluide-structure à l'aide de la méthode SPH.* PhD thesis, Ecole Centrale de Lyon, 2013.

[151] F. Blom. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Computer Methods in Applied Mechanics and Engineering*, 167(3):369–391, 1998. doi: 10.1016/S0045-7825(98)00151-0.

[152] K. C. Park. Partitioned transient analysis procedures for coupled-field problems: Stability analysis. *Journal of Applied Mechanics*, 47(2):370–376, 1980. doi: 10.1115/1.3153671.

[153] K. C. Park and C. A. Felippa. Partitioned transient analysis procedures for coupled-field problems: Accuracy analysis. *Journal of Applied Mechanics*, 47 (4):919–926, 1980. doi: 10.1115/1.3153814.

[154] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part 1: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124:79–112, 1994. doi: 10.1016/0045-7825(95)92707-9.

[155] C. Farhat and M. Lesmoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182:499–515, 2000. doi: 10.1016/S0045-7825(99)00206-6.

[156] H. G. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*, 195(17): 2028–2049, 2006. doi: 10.1016/j.cma.2004.11.032.

[157] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part 2: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190(24): 3147–3170, 2001. doi: 10.1016/S0045-7825(00)00386-8.

[158] N. Mahjoubi, A. Gravouil, A. Combescure, and N. Greffet. A monolithic energy conserving method to couple heterogeneous time integrators with incompatible time steps in structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1069–1086, 2011. doi: 10.1016/j.cma.2010.12.004.

[159] A. Gasmi, M. Sprague, J. Jonkman, and W. Jones. Numerical stability and accuracy of temporally coupled multi-physics modules in wind-turbine CAE tools. In *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013. doi: 10.2514/6.2013-203.

[160] C. Yvin. *Interaction fluide-structure pour des configurations multi-corps. Applications aux liaisons complexes, lois de commande d'actionneur et systèmes souples dans le domaine maritime.* PhD thesis, Ecole Centrale de Nantes, 2014.

[161] F. Belanger, Paidoussis, and E. de Langre. Time-marching analysis of fluid-coupled systems with large added mass. *AIAA Journal*, 33(4):752–757, 1995. doi: 10.2514/3.12641.

[162] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4506–4527, 2005. doi: 10.1016/j.cma.2004.12.005.

[163] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007. doi: 10.1016/j.cma.2006.09.002.

[164] E. H. van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *Journal of Applied Mechanics*, 76(2), 2009. doi: 10.1115/1.3059565.

[165] J. Jonkman. The new modularization framework for the FAST wind turbine CAE tool. In *Proceedings of the AIAA Aerospace Sciences Meeting, including the New Horizons Forum and Aerospace Exposition*, 2013. doi: 10.2514/6.2013-202.

[166] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science and Engineering*, 13(2): 31–39, 2011. doi: 10.1109/MCSE.2010.118.

[167] P. Peterson. F2PY: a tool for connecting fortran and python programs. *International Journal of Computational Science and Engineering*, 4(4):296–305, 2009. doi: 10.1504/ijcse.2009.029165.

[168] A. Hourtash. The kinematic hessian and higher derivatives. In *Proceedings of the 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2005. doi: 10.1109/CIRA.2005.1554272.

[169] W. Khalil and C. Chevallereau. An efficient algorithm for the dynamic control of robots in the cartesian space. In *Proceedings of the 26th IEEE Conference on Decision and Control*, 1987. doi: 10.1109/CDC.1987.272906.

[170] B. P. Jacob, R. de Almeida Bahiense, F. Nogueira Correa, and B. Martins Jacovazzo. Parallel implementations of coupled formulations for the analysis of floating production systems, part I: Coupling formulations. *Ocean Engineering*, 55:206–218, 2012. doi: 10.1016/j.oceaneng.2012.06.019.

[171] K. H. Ko, T. Park, K.-H. Kim, Y. Kim, and D. H. Yoon. Development of panel generation system for seakeeping analysis. *Computer-Aided Design*, 43(8):848–862, 2011. doi: 10.1016/j.cad.2011.04.013.

[172] K. H. Ko. A survey: application of geometric modeling techniques to ship modeling and design. *International Journal of Naval Architecture and Ocean Engineering*, 2(4):177–184, 2010. doi: 10.2478/ijnaoe-2013-0034.

[173] D.-W. Park and H.-J. Choi. Hydrodynamic hull form design using an optimization technique. *International Journal of Ocean System Engineering*, 3(1):1–9, 2013. doi: 10.5574/IJOSE.2013.3.1.001.

[174] S. Foufou, J. M. Brun, and A. Bouras. Surface / surface intersections: a three states classification. 1996.

[175] J. M. Rodrigues and C. Guedes Soares. Exact pressure integration on submerged bodies in waves using a quadtree adaptative mesh algorithm. *International Journal for Numerical Methods in Fluids*, 2014.

[176] J. M. Rodrigues and C. Guedes Soares. Froude-Krylov forces from exact pressure integrations on adaptive panel meshes in a time domain partially nonlinear model for ship motions. *Ocean Engineering*, 139:169–183, 2017. doi: 10.1016/j.oceaneng.2017.04.041.

[177] B. Horel, P.-E. Guillerm, J.-M. Rousset, and B. Alessandrini. A method of immersed surface capture for broaching application. In *Proceedings of the ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering, OMAE2013*, 2013. doi: 10.1115/OMAE2013-11527.

[178] D. Sengupta, R. Datta, and D. Sen. A simplified approach for computation of nonlinear ship loads and motions using a 3D time-domain panel method. *Ocean Engineering*, 117:99–113, 2016. doi: 10.1016/j.oceaneng.2016.03.039.

[179] K.-H. Lee and P.-S. Lee. Nonlinear hydrostatic analysis of flexible floating structures. *Applied Ocean Research*, 59:165–182, 2016. doi: 10.1016/j.apor.2016.05.016.

[180] H.-J. Choi, H.-H. Chun, I.-R. Park, and J. Kim. Panel cutting method: new approach to generate panels on a hull in Rankine source potential approximation. *International Journal of Naval Architecture and Ocean Engineering*, 3(4):225–232, 2011. doi: 10.2478/IJNAOE-2013-0066.

[181] H. C. Raven. *A solution method for the nonlinear ship wave resistance problem.* PhD thesis, Technical university of Delft, 1996.

[182] P. Bigay. *Développement d'un solveur faiblement compressible sur maillage cartésien pour les écoulements hydrodynamiques autour de corps.* PhD thesis, Ecole Centrale de Nantes, 2015.

[183] D. M. Ingram, D. M. Causon, and C. G. Mingham. Developments in cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3):561–572, 2003. doi: 10.1016/S0378-4754(02)00107-6.

[184] O. Schenk and K. Gartne. *PARDISO - User Guide Version 5.0.0*, 2014.

[185] P. M. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3):217–241, 2003. doi: 10.1016/S0168-874X(02)00070-7.

[186] C. J. Stimpson, C. D. Ernst, P. Knupp, P. P. Pébay, and D. Thompson. The verdict geometric quality library. Technical report, Sandia National Laboratories, 2007.

[187] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps "pararéel". *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 332(7):661–668, 2001.

[188] J. A. Armesto, R. Guanche, F. del Jesus, A. Iturrioz, and I. J. Losada. Comparative analysis of the methods to compute the radiation term in Cummins' equation. *Journal of Ocean Engineering and Marine Energy*, 1(4):377–393, 2015. doi: 10.1007/s40722-015-0027-1.

Appendix A

Multibody equations

This appendix presents the demonstration of two equations used in chapter I.

A.1 Kinematic recursive equation for accelerations

The goal of this section is to prove (I.48).

A.1.1 Linear acceleration

The time-differentiation of (I.24) gives:

$$\left(\frac{d^e\mathbf{v}(\mathbf{O}_j,\Sigma_j/\Sigma_e)}{dt}\right)_{/\Sigma_e} = {}^e\dot{\mathbf{v}}(\mathbf{O}_j,\Sigma_j/\Sigma_e) \tag{A.1}$$

$$= \left(\frac{d^e\mathbf{v}(\mathbf{O}_j,\Sigma_j/\Sigma_i)}{dt}\right)_{/\Sigma_e} + \left(\frac{d^e\mathbf{v}(\mathbf{O}_i,\Sigma_i/\Sigma_e)}{dt}\right)_{/\Sigma_e} + \left(\frac{d({}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e)\times{}^e\mathbf{O}_i\mathbf{O}_j)}{dt}\right)_{/\Sigma_e} \tag{A.2}$$

By using (I.25):

$$\left(\frac{d^e\mathbf{v}(\mathbf{O}_j,\Sigma_j/\Sigma_i)}{dt}\right)_{/\Sigma_e} = \left(\frac{d^e\mathbf{v}(\mathbf{O}_j,\Sigma_j/\Sigma_i)}{dt}\right)_{/\Sigma_i} + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e)\times{}^e\mathbf{v}(\mathbf{O}_j,\Sigma_j/\Sigma_i) \tag{A.3}$$

$$= \sigma_j\ddot{q}_j\,{}^e\mathbf{z}_j + \sigma_j\dot{q}_j\,{}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e)\times{}^e\mathbf{z}_j \tag{A.4}$$

And also:

$$\left(\frac{d({}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e)\times{}^e\mathbf{O}_i\mathbf{O}_j)}{dt}\right)_{/\Sigma_e}$$

$$= \left(\frac{d^e\mathbf{\Omega}(\Sigma_i/\Sigma_e)}{dt}\right)_{/\Sigma_e} \times {}^e\mathbf{O}_i\mathbf{O}_j + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times \left(\frac{d^e\mathbf{O}_i\mathbf{O}_j}{dt}\right)_{/\Sigma_e} \tag{A.5}$$

$$= {}^e\dot{\mathbf{\Omega}}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times \left[\left(\frac{d^e\mathbf{O}_i\mathbf{O}_j}{dt}\right)_{/\Sigma_j} + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j\right] \tag{A.6}$$

$$= {}^e\dot{\mathbf{\Omega}}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times [{}^e\mathbf{v}(\mathbf{O}_j, \Sigma_j/\Sigma_i) + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j] \tag{A.7}$$

$$= {}^e\dot{\mathbf{\Omega}}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times [\sigma_j\dot{q}_j\,{}^e\mathbf{z}_j + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j] \tag{A.8}$$

Finally:

$${}^e\dot{\mathbf{v}}(\mathbf{O}_j, \Sigma_j/\Sigma_e) = {}^e\dot{\mathbf{v}}(\mathbf{O}_i, \Sigma_i/\Sigma_e) + {}^e\dot{\mathbf{\Omega}}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times [{}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{O}_i\mathbf{O}_j] + 2\sigma_j\dot{q}_j\,{}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{z}_j + \sigma_j\ddot{q}_j\,{}^e\mathbf{z}_j \tag{A.9}$$

And by projecting in the local frame of body j:

$${}^j\dot{\mathbf{v}}_j = {}^j\mathbf{R}_i\,{}^i\dot{\mathbf{v}}_i - {}^j\mathbf{R}_i\mathbf{S}({}^i\mathbf{P}_j)\,{}^i\dot{\boldsymbol{\omega}}_i + {}^j\mathbf{R}_i\mathbf{S}({}^i\boldsymbol{\omega}_i)\mathbf{S}({}^i\boldsymbol{\omega}_i)\,{}^i\mathbf{P}_j + 2\sigma_j\dot{q}_j\mathbf{S}({}^j\mathbf{R}_i\,{}^i\boldsymbol{\omega}_i)\,{}^j\mathbf{z}_j + \sigma_j\ddot{q}_j\,{}^j\mathbf{z}_j \tag{A.10}$$

A.1.2 Angular acceleration

The time-differentiation of (I.27) gives:

$$\left(\frac{d^e\mathbf{\Omega}(\Sigma_j/\Sigma_e)}{dt}\right)_{/\Sigma_e} = {}^e\dot{\mathbf{\Omega}}(\Sigma_j/\Sigma_e) \tag{A.11}$$

$$= \left(\frac{d^e\mathbf{\Omega}(\Sigma_j/\Sigma_i)}{dt}\right)_{/\Sigma_e} + \left(\frac{d^e\mathbf{\Omega}(\Sigma_i/\Sigma_e)}{dt}\right)_{/\Sigma_e} \tag{A.12}$$

$$= \left(\frac{d^e\mathbf{\Omega}(\Sigma_j/\Sigma_i)}{dt}\right)_{/\Sigma_i} + {}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{\Omega}(\Sigma_j/\Sigma_i) + {}^e\dot{\mathbf{\Omega}}(\Sigma_i/\Sigma_e) \tag{A.13}$$

After using (I.28):

$${}^e\dot{\mathbf{\Omega}}(\Sigma_j/\Sigma_e) = {}^e\dot{\mathbf{\Omega}}(\Sigma_i/\Sigma_e) + \bar{\sigma}_j\dot{q}_j\,{}^e\mathbf{\Omega}(\Sigma_i/\Sigma_e) \times {}^e\mathbf{z}_j + \bar{\sigma}_j\ddot{q}_j\,{}^e\mathbf{z}_j \tag{A.14}$$

By projecting in the local frame of body j:

$${}^j\dot{\boldsymbol{\omega}}_j = {}^j\mathbf{R}_i\,{}^i\dot{\boldsymbol{\omega}}_i + \bar{\sigma}_j\dot{q}_j\mathbf{S}({}^j\mathbf{R}_i\,{}^i\boldsymbol{\omega}_i)\,{}^j\mathbf{z}_j + \bar{\sigma}_j\ddot{q}_j\,{}^j\mathbf{z}_j \tag{A.15}$$

A.1.3 Final results

Equations (A.10) and (A.15) lead to:

$${}^j\dot{\mathbf{V}}_j = {}^j\mathbf{T}_i\,{}^i\dot{\mathbf{V}}_i + {}^j\boldsymbol{\gamma}_j + \ddot{q}_j\,{}^j\mathbf{a}_j \tag{A.16}$$

with:

$$ {}^{j}\mathbf{T}_{i} = \begin{pmatrix} {}^{j}\mathbf{R}_{i} & -{}^{j}\mathbf{R}_{i}\mathbf{S}({}^{i}\mathbf{P}_{j}) \\ \mathbf{0}_{3\times 3} & {}^{j}\mathbf{R}_{i} \end{pmatrix} \tag{A.17} $$

$$ {}^{j}\boldsymbol{\gamma}_{j} = \begin{pmatrix} {}^{j}\mathbf{R}_{i}\mathbf{S}({}^{i}\boldsymbol{\omega}_{i})\mathbf{S}({}^{i}\boldsymbol{\omega}_{i})\,{}^{i}\mathbf{P}_{j} + 2\sigma_{j}\dot{q}_{j}\mathbf{S}({}^{j}\mathbf{R}_{i}\,{}^{i}\boldsymbol{\omega}_{i})\,{}^{j}\mathbf{z}_{j} \\ \bar{\sigma}_{j}\dot{q}_{j}\mathbf{S}({}^{j}\mathbf{R}_{i}\,{}^{i}\boldsymbol{\omega}_{i})\,{}^{j}\mathbf{z}_{j} \end{pmatrix} \tag{A.18} $$

$$ {}^{j}\mathbf{a}_{j} = \begin{pmatrix} \sigma_{j}\,{}^{j}\mathbf{z}_{j}^{T} & \bar{\sigma}_{j}\,{}^{j}\mathbf{z}_{j}^{T} \end{pmatrix}^{T} \tag{A.19} $$

A.2 Time-differentiation of the linear and angular momenta

The goal of this section is to prove (I.65). By definition, the time-differentiation of the linear and angular momenta at $\mathbf{O}_j$ and expressed in Σ_e is:

$$ {}^{e}\boldsymbol{\Phi}_{j} = \begin{pmatrix} m_{j}\left(\dfrac{d^{e}\mathbf{v}(\mathbf{G}_{j},\Sigma_{j}/\Sigma_{e})}{dt}\right)_{/\Sigma_{e}} \\ {}^{e}\boldsymbol{\delta}(\mathbf{O}_{j},\Sigma_{j}/\Sigma_{e}) \end{pmatrix} \tag{A.20} $$

where ${}^{e}\boldsymbol{\delta}(\mathbf{O}_{j},\Sigma_{j}/\Sigma_{e})$ is the time-derivative of the angular momentum at $\mathbf{O}_j$.

A.2.1 Linear momentum

The transport of the linear momentum at $\mathbf{O}_j$ gives:

$$ m_{j}\left(\frac{d^{e}\mathbf{v}(\mathbf{G}_{j},\Sigma_{j}/\Sigma_{e})}{dt}\right)_{/\Sigma_{e}} $$

$$ = m_{j}\left(\frac{d({}^{e}\mathbf{v}(\mathbf{G}_{j},\Sigma_{j}/\Sigma_{e}) + {}^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e}) \times {}^{e}\mathbf{O}_{j}\mathbf{G}_{j})}{dt}\right)_{/\Sigma_{e}} \tag{A.21} $$

$$ = m_{j}\left[{}^{e}\dot{\mathbf{v}}(\mathbf{O}_{j},\Sigma_{j}/\Sigma_{e}) + \left(\frac{d^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e})}{dt}\right)_{/\Sigma_{e}} \times {}^{e}\mathbf{O}_{j}\mathbf{G}_{j} + {}^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e}) \times \left(\frac{d^{e}\mathbf{O}_{j}\mathbf{G}_{j}}{dt}\right)_{/\Sigma_{e}}\right] \tag{A.22} $$

$$ = m_{j}\left[{}^{e}\dot{\mathbf{v}}(\mathbf{O}_{j},\Sigma_{j}/\Sigma_{e}) + {}^{e}\dot{\boldsymbol{\Omega}}(\Sigma_{j}/\Sigma_{e}) \times {}^{e}\mathbf{O}_{j}\mathbf{G}_{j}\right] $$

$$ + m_{j}\left[{}^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e}) \times \left(\left(\frac{d^{e}\mathbf{O}_{j}\mathbf{G}_{j}}{dt}\right)_{/\Sigma_{j}} + {}^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e}) \times {}^{e}\mathbf{O}_{j}\mathbf{G}_{j}\right)\right] \tag{A.23} $$

But each body is rigid, consequently:

$$ \left(\frac{d^{e}\mathbf{O}_{j}\mathbf{G}_{j}}{dt}\right)_{/\Sigma_{j}} = \mathbf{0}_{3\times 1} \tag{A.24} $$

So:

$$ m_{j}\left(\frac{d^{e}\mathbf{v}(\mathbf{G}_{j},\Sigma_{j}/\Sigma_{e})}{dt}\right)_{/\Sigma_{e}} = m_{j}\left[{}^{e}\dot{\mathbf{v}}(\mathbf{O}_{j},\Sigma_{j}/\Sigma_{e}) - \mathbf{S}({}^{e}\mathbf{O}_{j}\mathbf{G}_{j})\,{}^{e}\dot{\boldsymbol{\Omega}}(\Sigma_{j}/\Sigma_{e})\right] $$
$$ + m_{j}[\mathbf{S}({}^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e}))\mathbf{S}({}^{e}\boldsymbol{\Omega}(\Sigma_{j}/\Sigma_{e}))\,{}^{e}\mathbf{O}_{j}\mathbf{G}_{j}] \tag{A.25} $$

A.2.2 Angular momentum

The transport of the angular momentum at $\mathbf{G}_j$ gives:

$$ {}^e\boldsymbol{\delta}(\mathbf{O}_j, \Sigma_j/\Sigma_e) = {}^e\boldsymbol{\delta}(\mathbf{G}_j, \Sigma_j/\Sigma_e) + m_j\, {}^e\dot{\mathbf{v}}(\mathbf{G}_j, \Sigma_j/\Sigma_e) \times {}^e\mathbf{G}_j\mathbf{O}_j \tag{A.26} $$

Using (A.25):

$$ \begin{aligned} {}^e\boldsymbol{\delta}(\mathbf{O}_j, \Sigma_j/\Sigma_e) = {}^e\boldsymbol{\delta}(\mathbf{G}_j, \Sigma_j/\Sigma_e) + m_j\Big[{}^e\dot{\mathbf{v}}(\mathbf{O}_j, \Sigma_j/\Sigma_e) - \mathbf{S}({}^e\mathbf{O}_j\mathbf{G}_j)\,{}^e\dot{\boldsymbol{\Omega}}(\Sigma_j/\Sigma_e)\Big] \times {}^e\mathbf{G}_j\mathbf{O}_j \\ + m_j[\mathbf{S}({}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e))\mathbf{S}({}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e))\,{}^e\mathbf{O}_j\mathbf{G}_j] \times {}^e\mathbf{G}_j\mathbf{O}_j \end{aligned} \tag{A.27} $$

with:

$$ {}^e\boldsymbol{\delta}(\mathbf{G}_j, \Sigma_j/\Sigma_e) = \left(\frac{d\,{}^e\mathbf{I}_{\mathbf{G}j}\,{}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e)}{dt}\right)_{/\Sigma_e} \tag{A.28} $$

$$ = {}^e\mathbf{I}_{\mathbf{G}j}\,{}^e\dot{\boldsymbol{\Omega}}(\Sigma_j/\Sigma_e) + \left(\frac{d\Big({}^e\mathbf{R}_j\,{}^j\mathbf{I}_{\mathbf{G}j}\,{}^e\mathbf{R}_j^T\Big)}{dt}\right)_{/\Sigma_e} {}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e) \tag{A.29} $$

But each body is rigid, consequently:

$$ \left(\frac{d\,{}^j\mathbf{I}_{\mathbf{G}j}}{dt}\right)_{/\Sigma_j} = \mathbf{0}_{3\times3} \tag{A.30} $$

And using (I.51):

$$ {}^e\boldsymbol{\delta}(\mathbf{G}_j, \Sigma_j/\Sigma_e) = {}^e\mathbf{I}_{\mathbf{G}j}\,{}^e\dot{\boldsymbol{\Omega}}(\Sigma_j/\Sigma_e) + {}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e) \times \big({}^e\mathbf{I}_{\mathbf{G}j}\,{}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e)\big) \tag{A.31} $$

The transport of the inertia matrix from $\mathbf{G}_j$ to $\mathbf{O}_j$, using the Huygens theorem gives:

$$ {}^e\mathbf{I}_{\mathbf{O}j} = {}^e\mathbf{I}_{\mathbf{G}j} - m_j\mathbf{S}({}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e))\mathbf{S}({}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e)) \tag{A.32} $$

So:

$$ {}^e\boldsymbol{\delta}(\mathbf{O}_j, \Sigma_j/\Sigma_e) = {}^e\mathbf{I}_{\mathbf{O}j}\,{}^e\dot{\boldsymbol{\Omega}}(\Sigma_j/\Sigma_e) + m_j\mathbf{S}({}^e\mathbf{O}_j\mathbf{G}_j)\,{}^e\dot{\mathbf{v}}(\mathbf{O}_j, \Sigma_j/\Sigma_e) + \mathbf{S}({}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e))\big({}^e\mathbf{I}_{\mathbf{O}j}\,{}^e\boldsymbol{\Omega}(\Sigma_j/\Sigma_e)\big) \tag{A.33} $$

A.2.3 Final results

After projecting in the local frame of body j, Equations (A.25) and (A.33) lead to:

$$ {}^j\boldsymbol{\Phi}_j = {}^j\mathbf{M}_j\,{}^j\dot{\mathbf{V}}_j + \begin{pmatrix} m_j\mathbf{S}({}^j\boldsymbol{\omega}_j)\mathbf{S}({}^j\boldsymbol{\omega}_j)\,{}^j\mathbf{S}_j \\ \mathbf{S}({}^j\boldsymbol{\omega}_j)\,{}^j\mathbf{I}_{\mathbf{O}j}\,{}^j\boldsymbol{\omega}_j \end{pmatrix} \tag{A.34} $$

with:

$$ {}^j\mathbf{M}_j = \begin{pmatrix} m_j\mathbf{I}_3 & -m_j\mathbf{S}({}^j\mathbf{S}_j) \\ m_j\mathbf{S}({}^j\mathbf{S}_j) & {}^j\mathbf{I}_{\mathbf{O}j} \end{pmatrix} \tag{A.35} $$

Appendix B

Cummins' equation

This appendix presents Cummins' equation and how to use hydrodynamic databases to perform time-domain simulations.

When a fully linear potential flow-based solver is used in frequency-domain, as *Nemoh* or *WAMIT*, three numerical results are obtained, constituting a hydrodynamic database:

- Added mass matrices $\mathbf{A}(\omega)$;
- Hydrodynamic damping matrices $\mathbf{B}(\omega)$;
- Excitation force vectors $\mathbf{f}^{Exc}(\omega)$.

$\mathbf{A}$ and $\mathbf{B}$ are square matrices of size $6 \times N_b$ while $\mathbf{f}^{Exc}$ is a complex vector of size $6 \times N_b$, with N_b the number of bodies.

From the hydrodynamic database, the time-domain hydrodynamic loads are computed:

- The excitation force which represents the sum of the Froude-Krylov loads and the diffraction loads:

$$\mathbf{f}^{Exc}(t) = \int_{-\infty}^{\infty} \mathbf{K}^{Exc}(t-\tau)\eta^I(\tau)\, d\tau \tag{B.1}$$

  where η^I denotes the incident wave elevation and $\mathbf{K}^{Exc}$ is the impulse response function vector of the complex excitation force vectors:

$$\mathbf{K}^{Exc}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{Re}\big(\mathbf{f}^{Exc}(\omega)e^{i\omega t}\big)\, d\omega \tag{B.2}$$

- The radiation force:

$$\mathbf{f}^{R}(t) = -\mathbf{A}(\infty)\ddot{\mathbf{X}}(t) - \int_0^t \mathbf{K}^{R}(t-\tau)\dot{\mathbf{X}}(\tau)\, d\tau \tag{B.3}$$

  where $\mathbf{X}$ is the Cartesian position of the bodies, $\mathbf{A}(\infty)$ represents the added mass matrix at infinity frequency and $\mathbf{K}^R$ is the impulse response function matrix of the radiation force:

$$\mathbf{K}^{R}(t) = \frac{2}{\pi}\int_0^{\infty} \mathbf{B}(\omega)\cos(\omega t)\, d\omega \tag{B.4}$$

Finally the motion equation of a body or Cummin's equation arises [102]:

$$[\mathbf{M} + \mathbf{A}(\infty)]\ddot{\mathbf{X}}(t) + \int_0^t \mathbf{K}^R(t-\tau)\dot{\mathbf{X}}(\tau)\, d\tau + \mathbf{C}\mathbf{X}(t) = \mathbf{f}^{Exc}(t) \tag{B.5}$$

with $\mathbf{M}$ is the mass matrix and $\mathbf{C}$ denotes the hydrostatic stiffness matrix.

Other methods exist to compute the convolution integral of Cummins' equation, such as Prony's method [188].

Appendix C

Measurements

This appendix presents the experimental results of the Cases not displayed in chapter VI.

C.1 Case 1

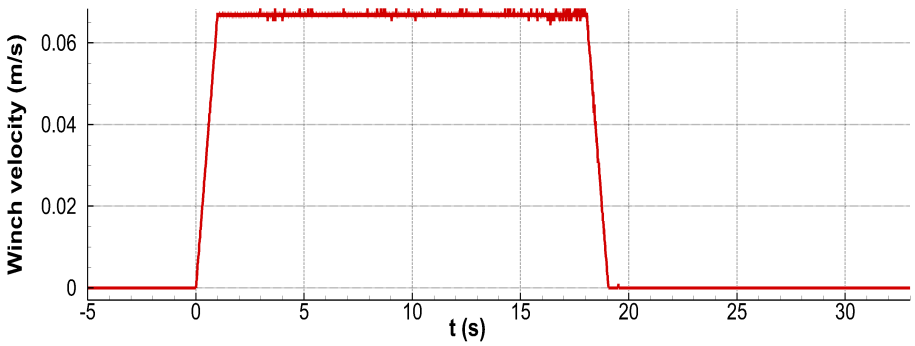


Figure C.1 – Winch velocity for Case 1. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

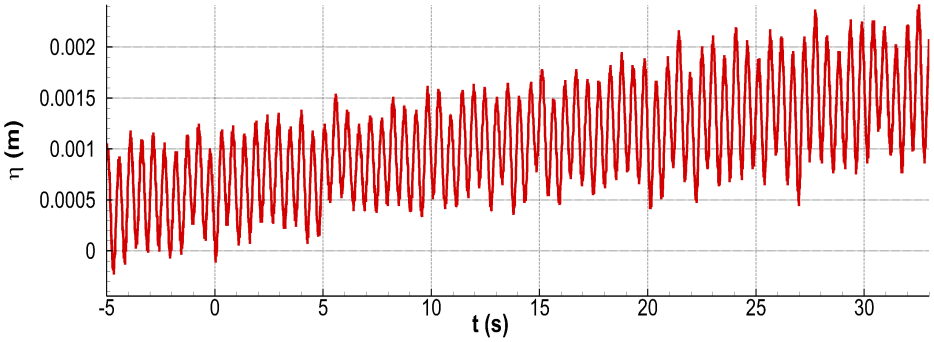


Figure C.2 – Wave elevation for Case 1. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

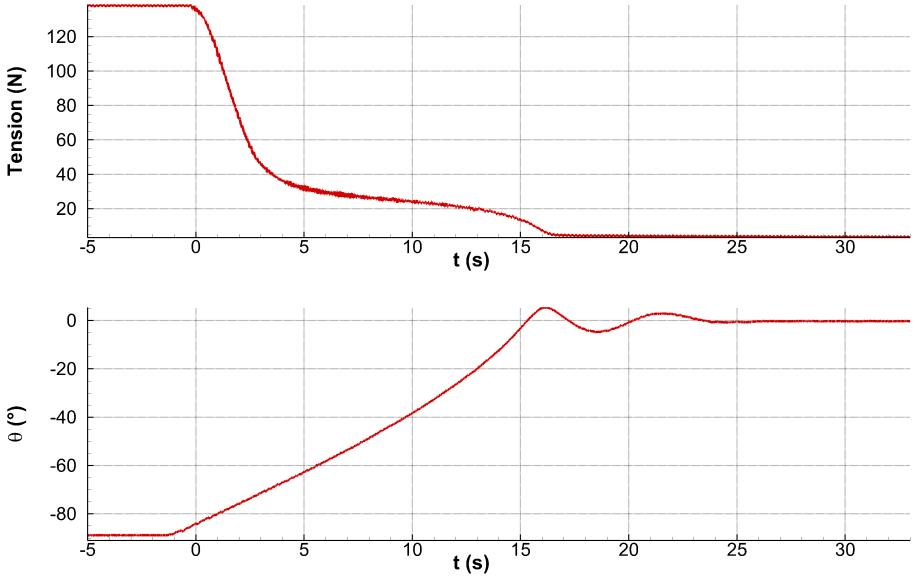


Figure C.3 – Cable tension and rotational motion for Case 1. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.2 Case 2

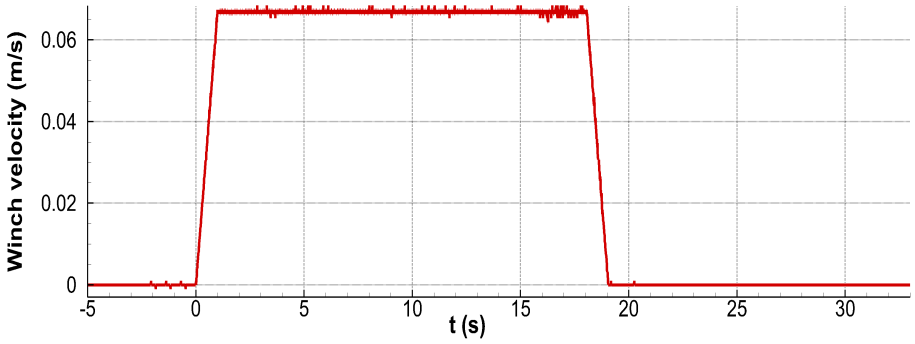


Figure C.4 – Winch velocity for Case 2. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

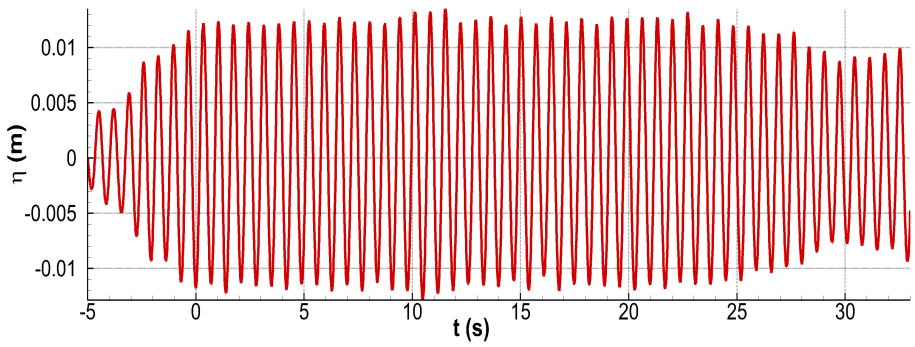


Figure C.5 – Wave elevation for Case 2. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

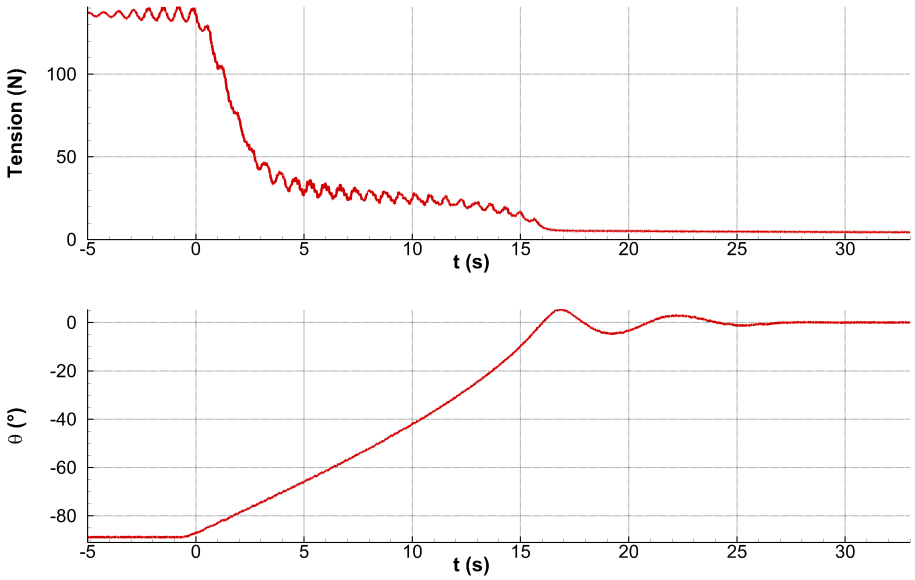


Figure C.6 – Cable tension and rotational motion for Case 2. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.3 Case 3

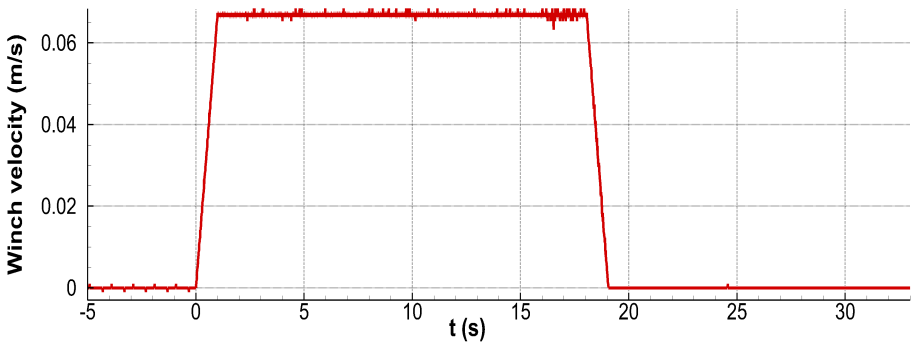


Figure C.7 – Winch velocity for Case 3. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

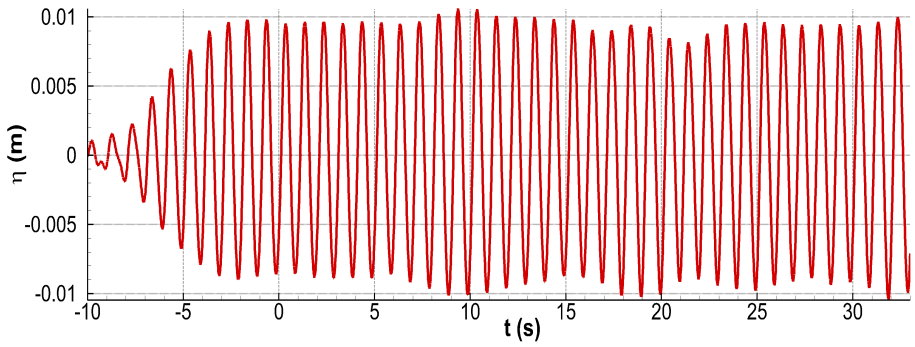


Figure C.8 – Wave elevation for Case 3. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

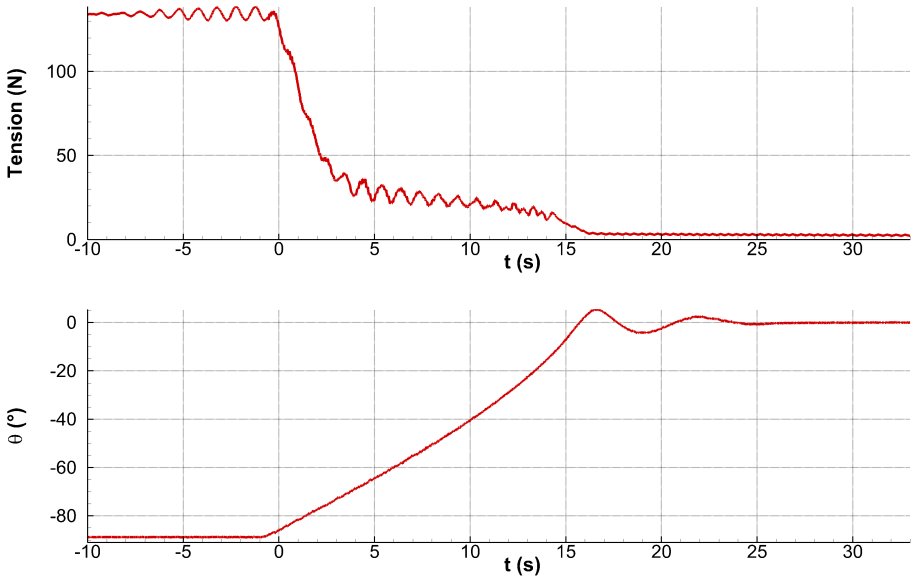


Figure C.9 – Cable tension and rotational motion for Case 3. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.4 Case 4

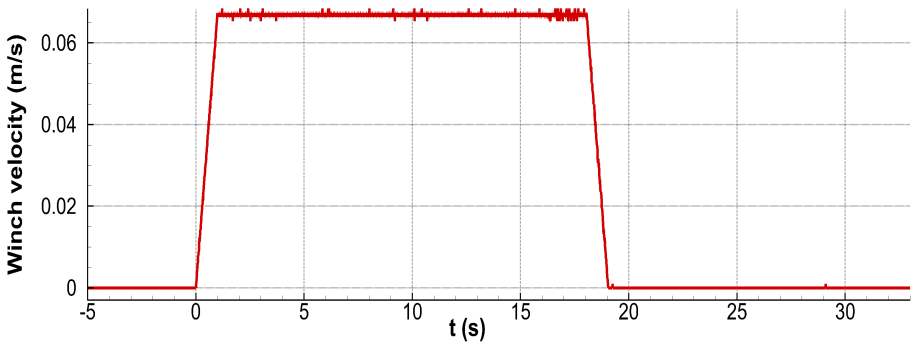


Figure C.10 – Winch velocity for Case 4. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

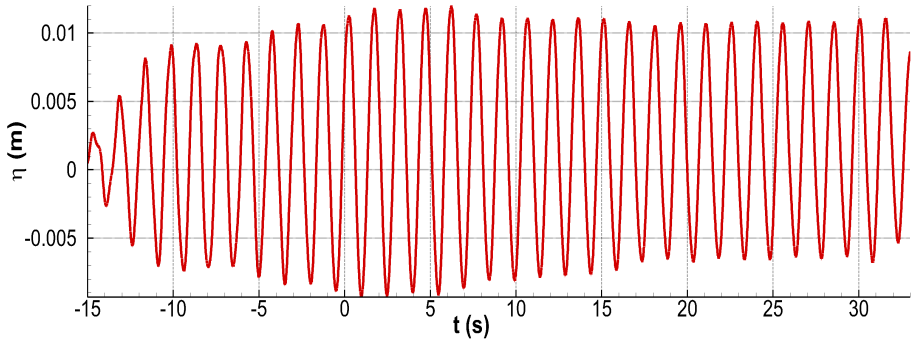


Figure C.11 – Wave elevation for Case 4. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

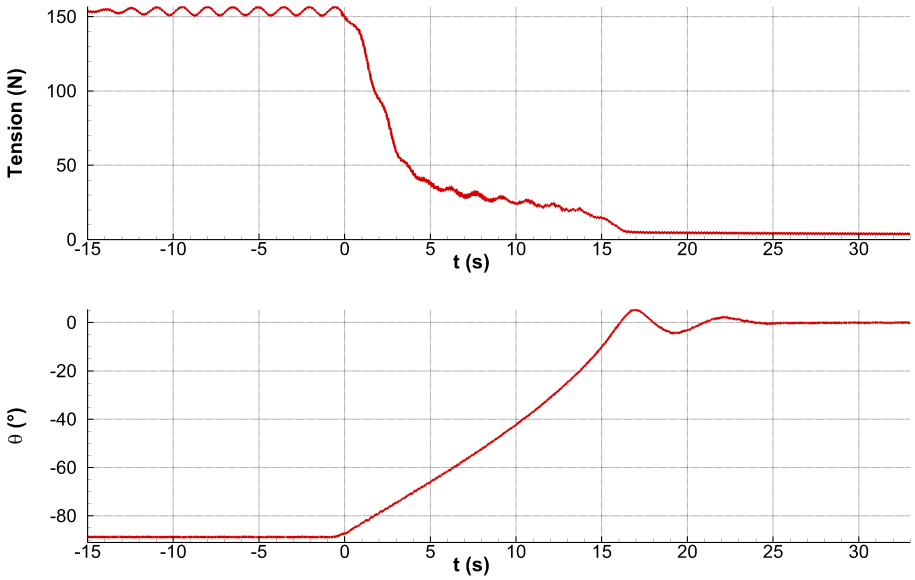


Figure C.12 – Cable tension and rotational motion for Case 4. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.5 Case 5

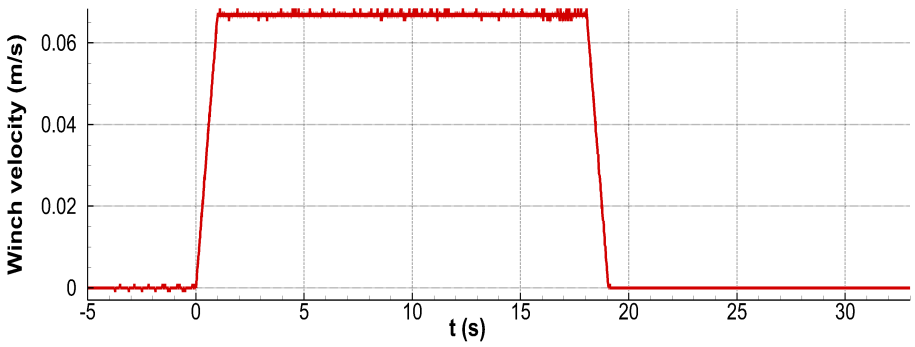


Figure C.13 – Winch velocity for Case 5. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

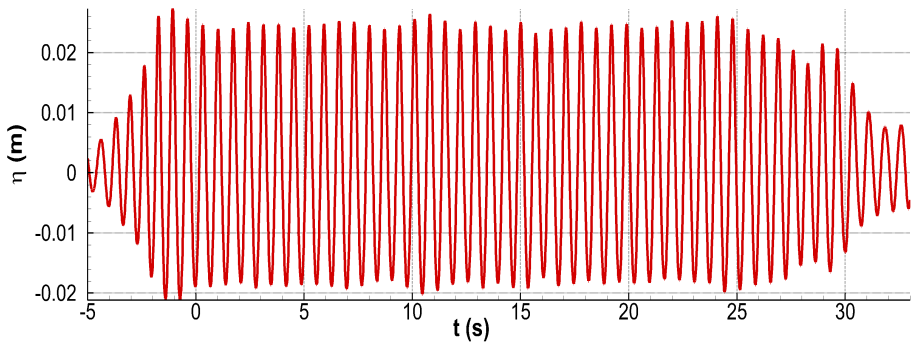


Figure C.14 – Wave elevation for Case 5. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

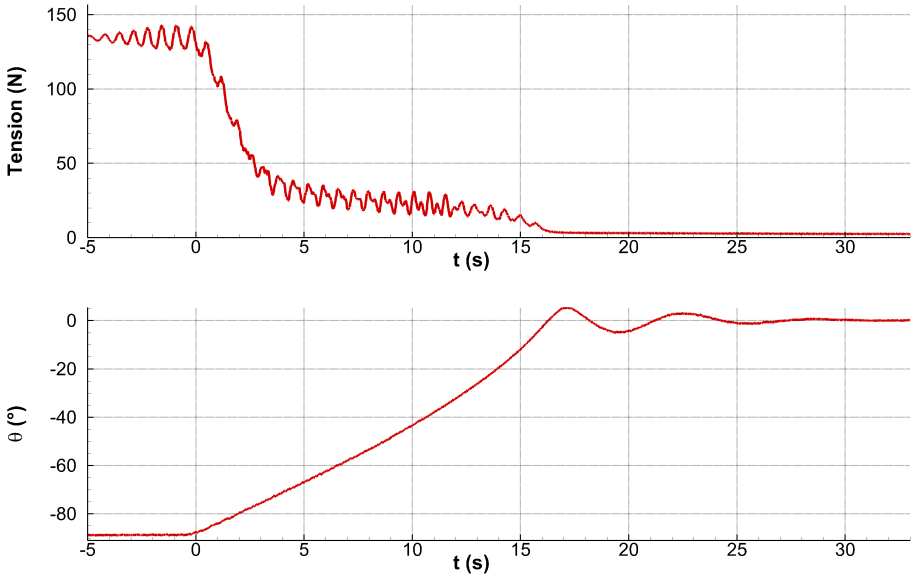


Figure C.15 – Cable tension and rotational motion for Case 5. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.6 Case 6

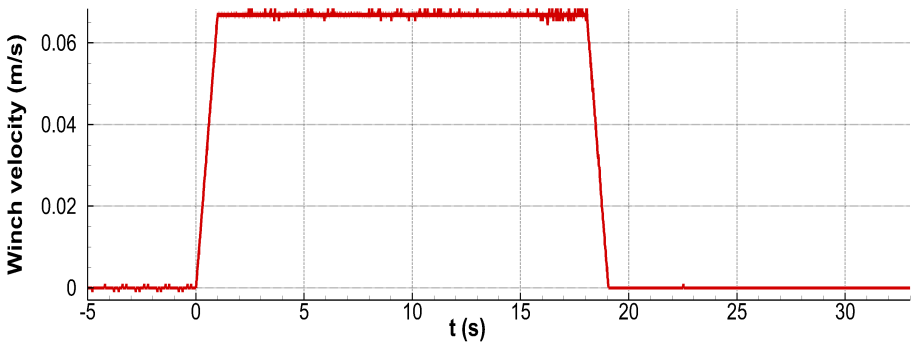


Figure C.16 – Winch velocity for Case 6. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

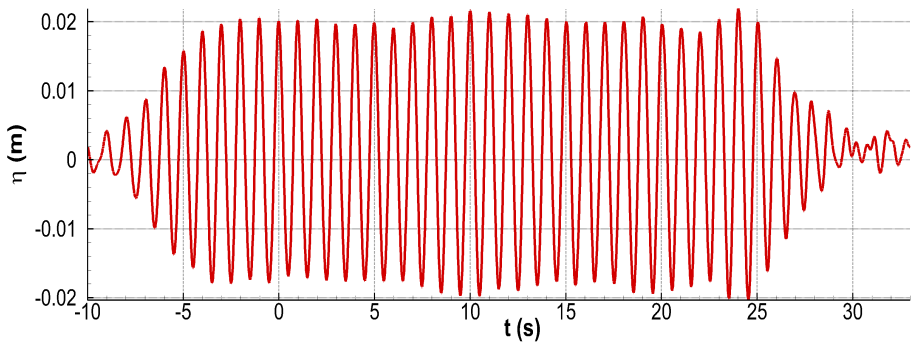


Figure C.17 – Wave elevation for Case 6. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

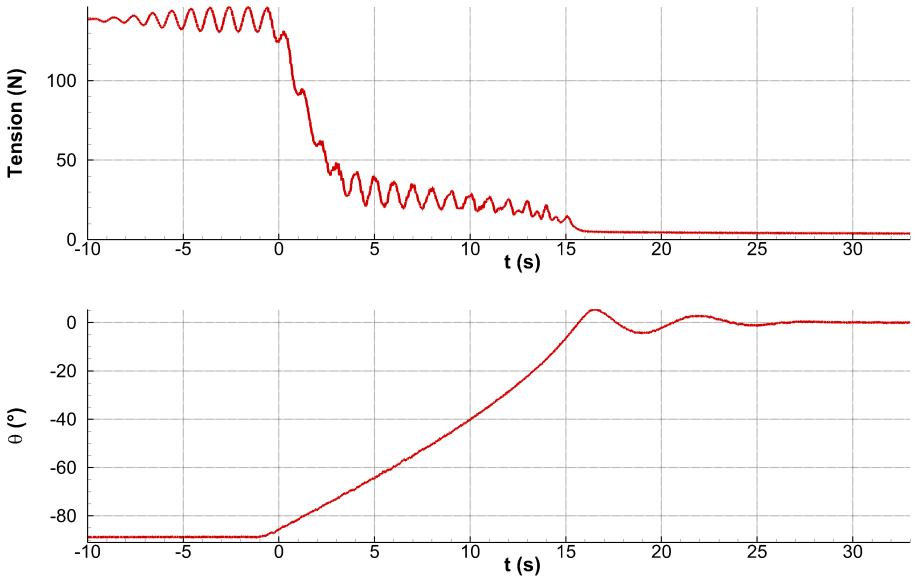


Figure C.18 – Cable tension and rotational motion for Case 6. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.7 Case 7

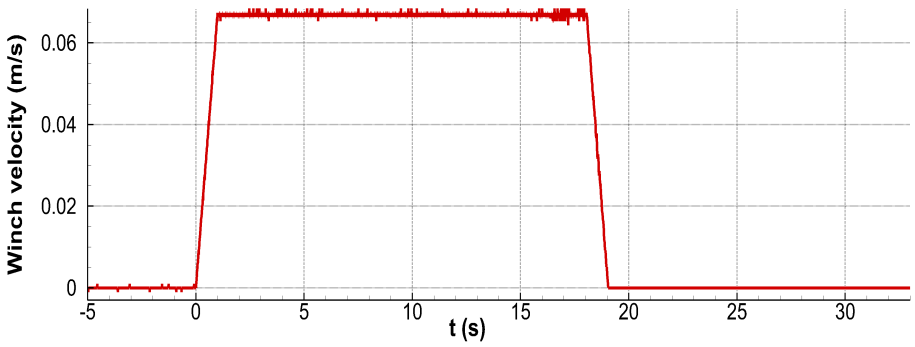


Figure C.19 – Winch velocity for Case 7. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

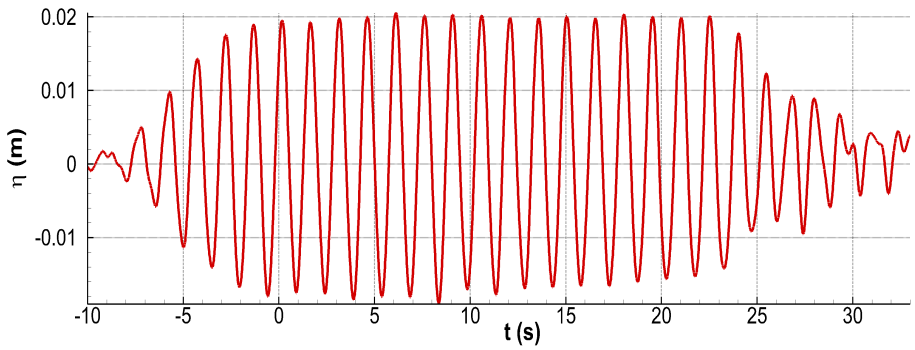


Figure C.20 – Wave elevation for Case 7. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

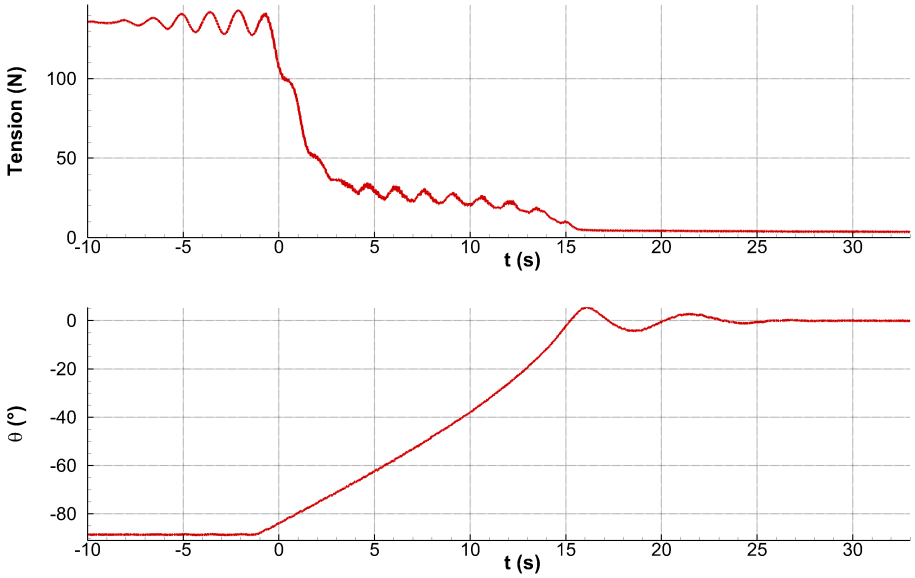


Figure C.21 – Cable tension and rotational motion for Case 7. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.8 Case 8

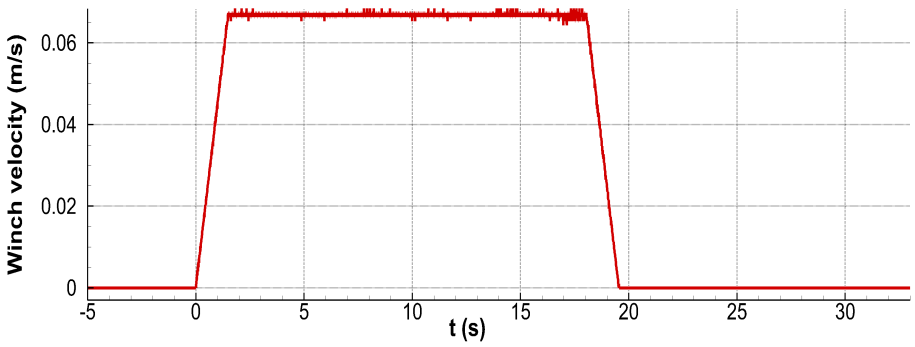


Figure C.22 – Winch velocity for Case 8. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

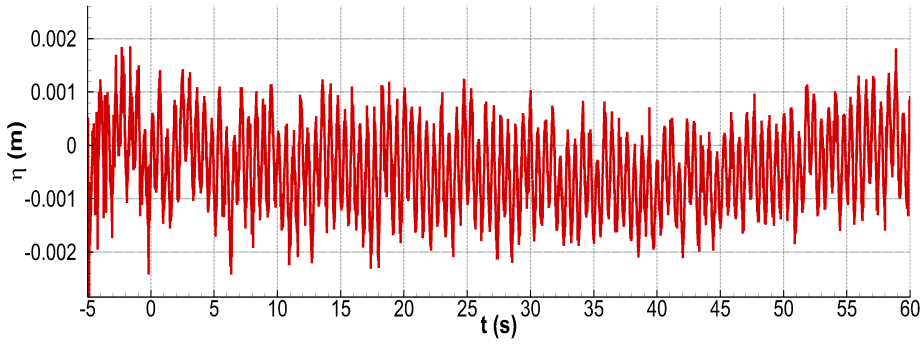


Figure C.23 – Wave elevation for Case 8. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

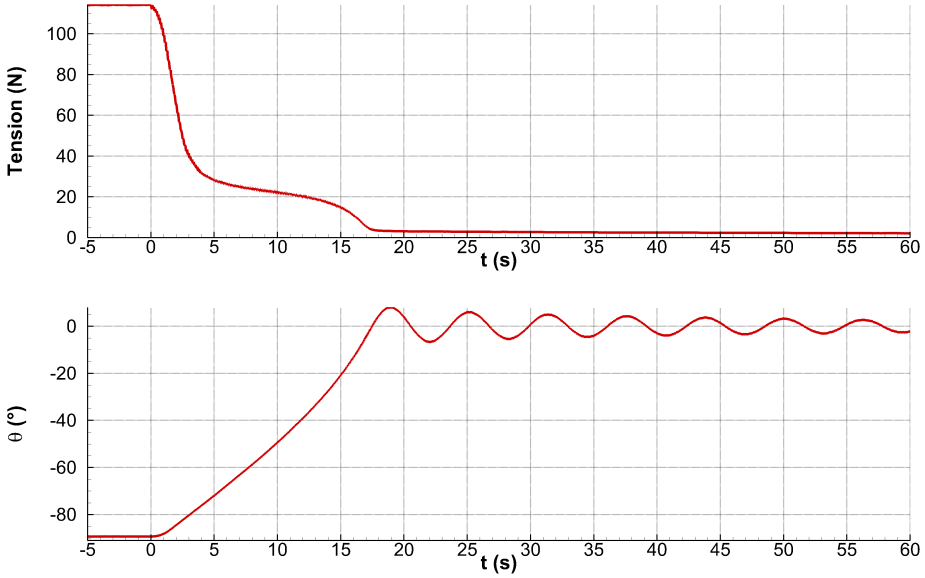


Figure C.24 – Cable tension and rotational motion for Case 8. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.9 Case 9

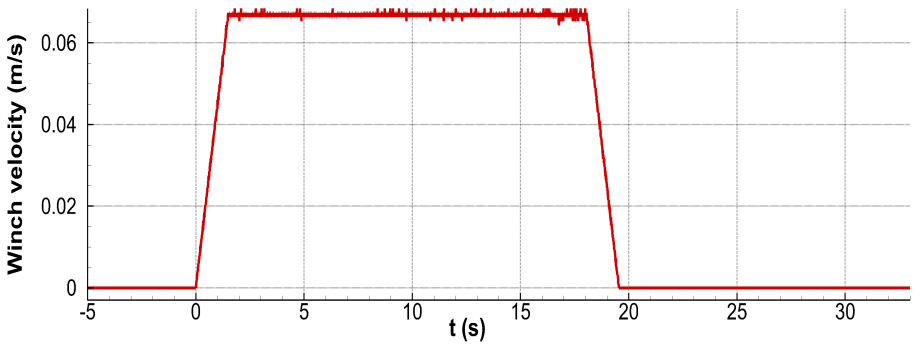


Figure C.25 – Winch velocity for Case 9. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

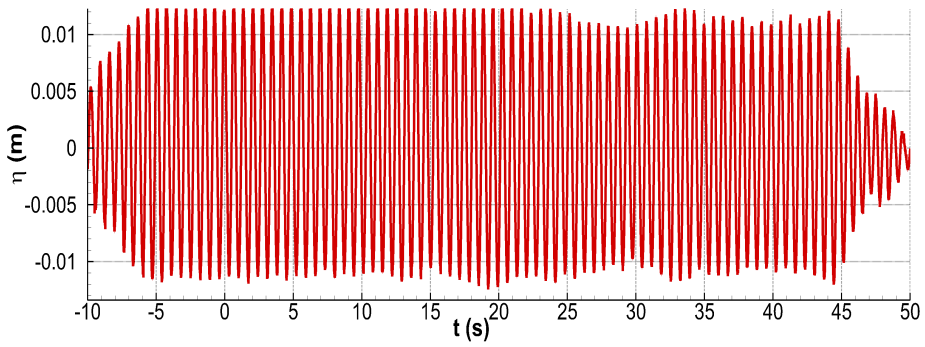


Figure C.26 – Wave elevation for Case 9. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

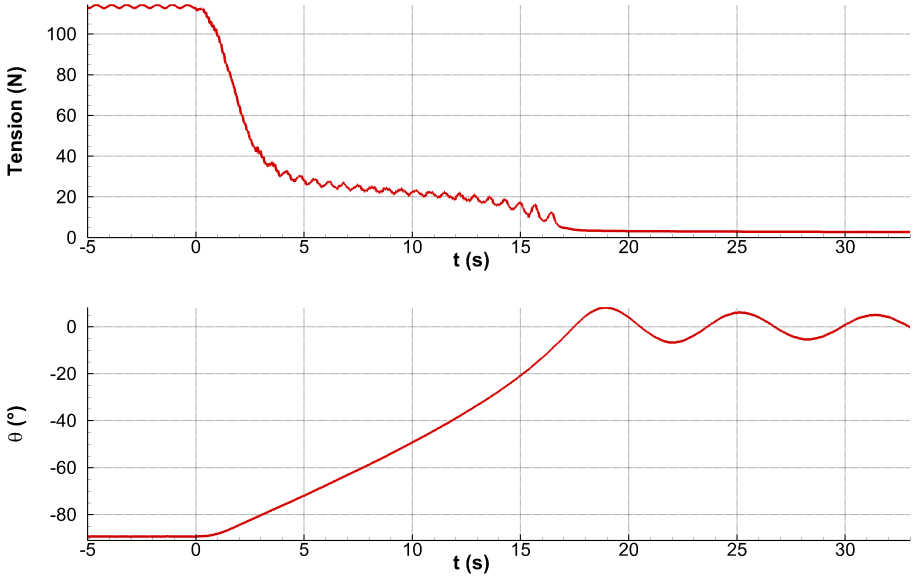


Figure C.27 – Cable tension and rotational motion for Case 9. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.10 Case 10

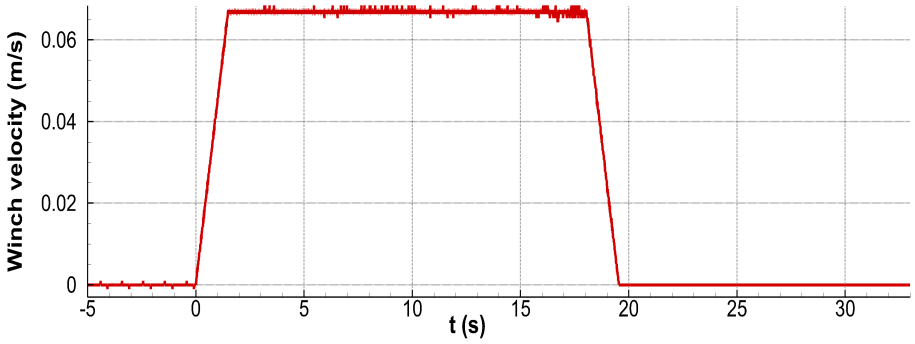


Figure C.28 – Winch velocity for Case 10. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

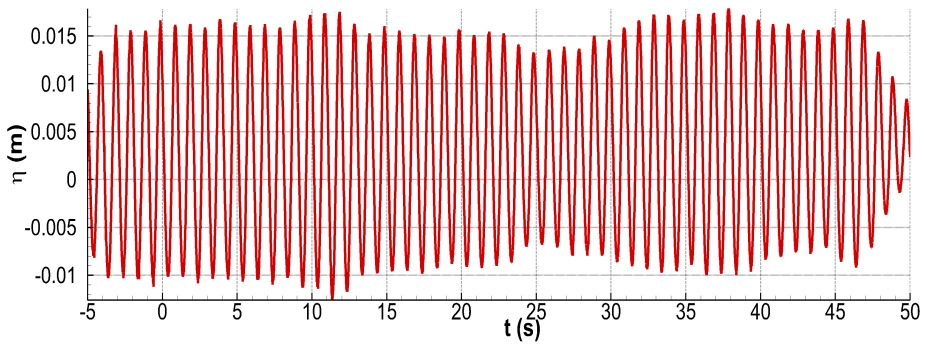


Figure C.29 – Wave elevation for Case 10. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

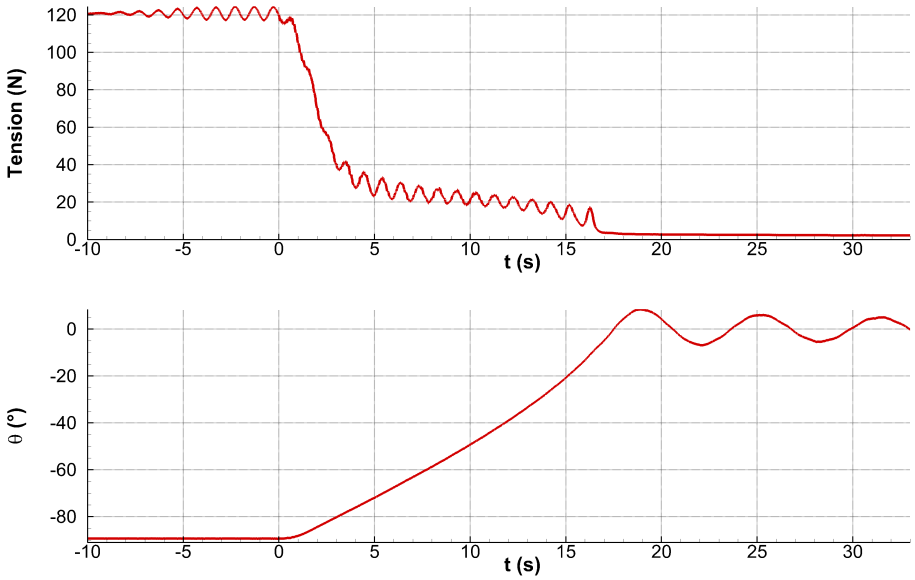


Figure C.30 – Cable tension and rotational motion for Case 10. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.11 Case 11

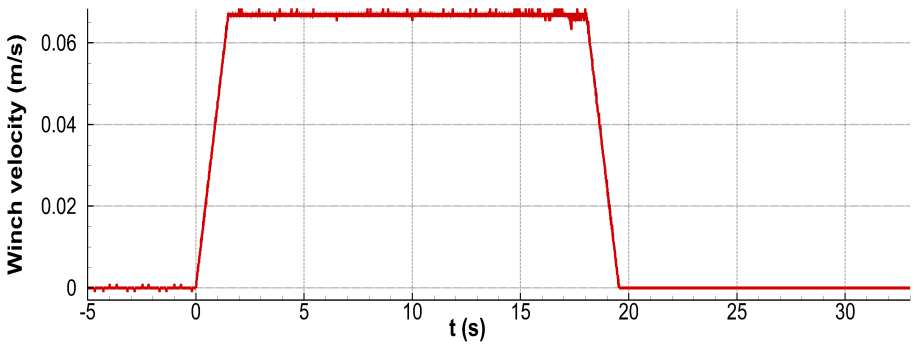


Figure C.31 – Winch velocity for Case 11. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

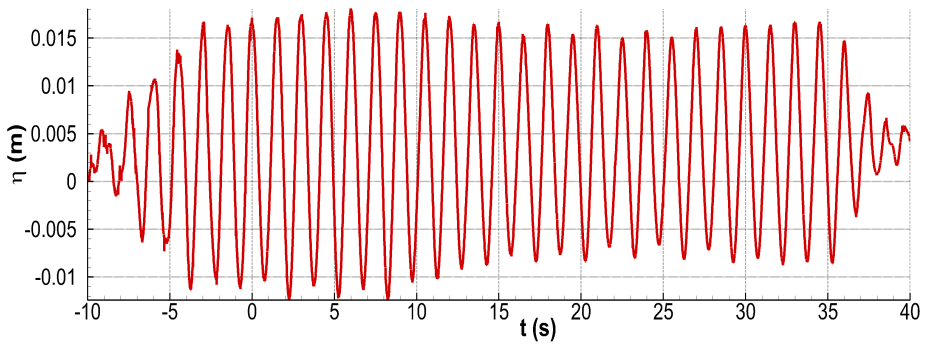


Figure C.32 – Wave elevation for Case 11. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

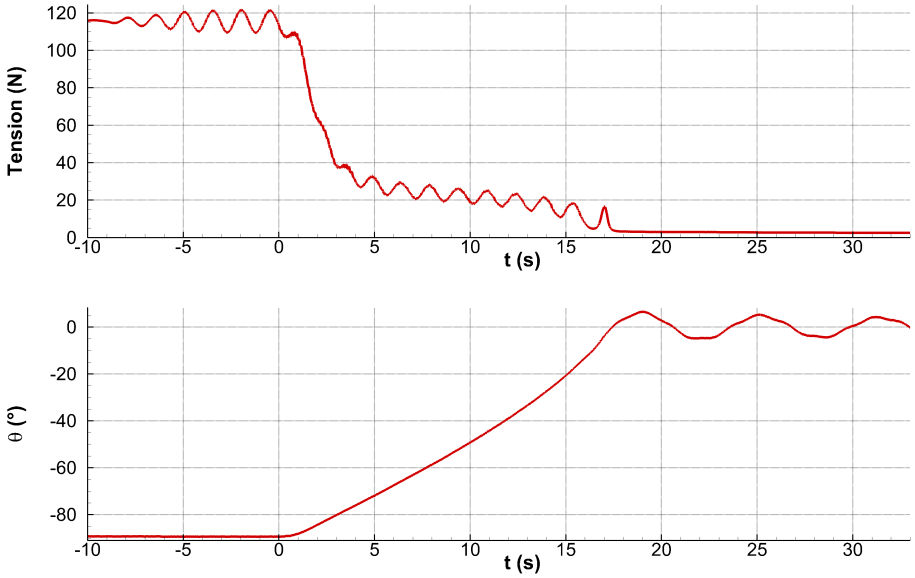


Figure C.33 – Cable tension and rotational motion for Case 11. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.12 Case 12

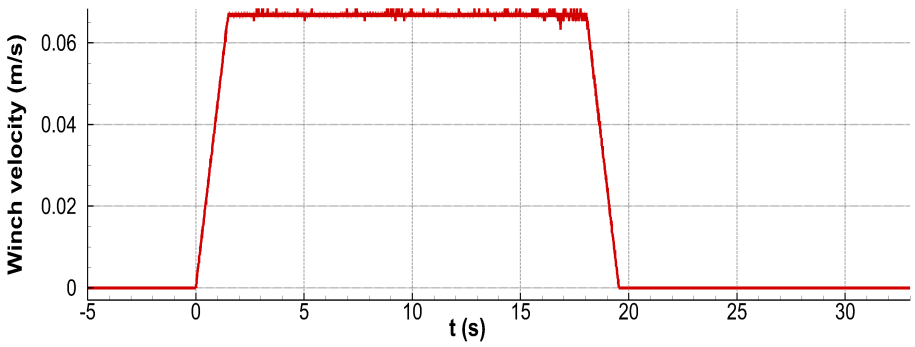


Figure C.34 – Winch velocity for Case 12. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

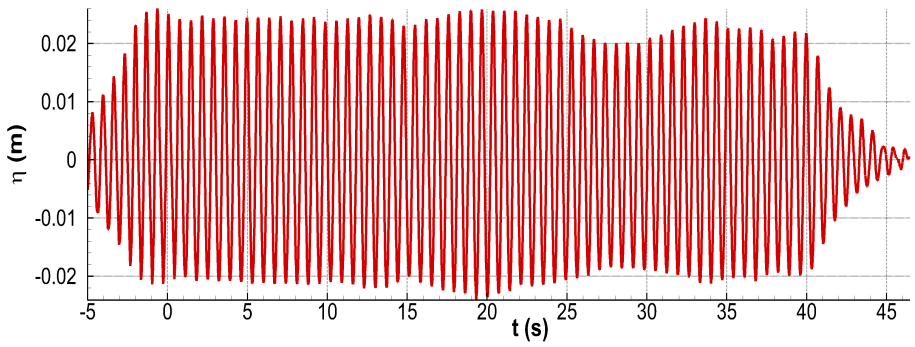


Figure C.35 – Wave elevation for Case 12. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

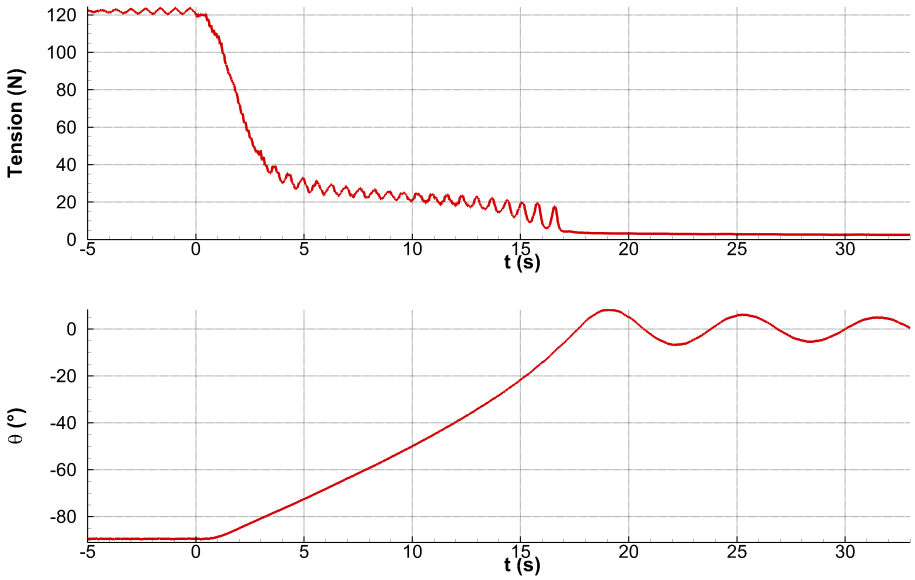


Figure C.36 – Cable tension and rotational motion for Case 12. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.13 Case 13

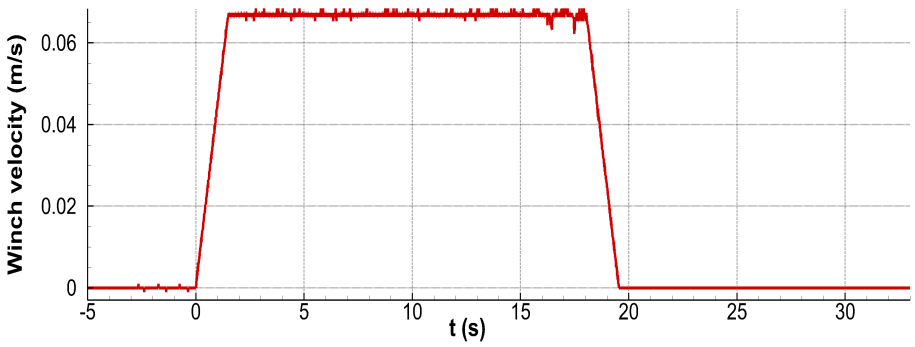


Figure C.37 – Winch velocity for Case 13. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

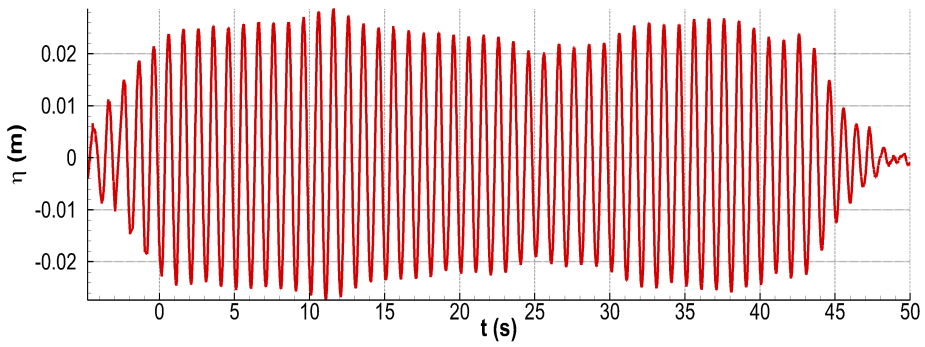


Figure C.38 – Wave elevation for Case 13. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

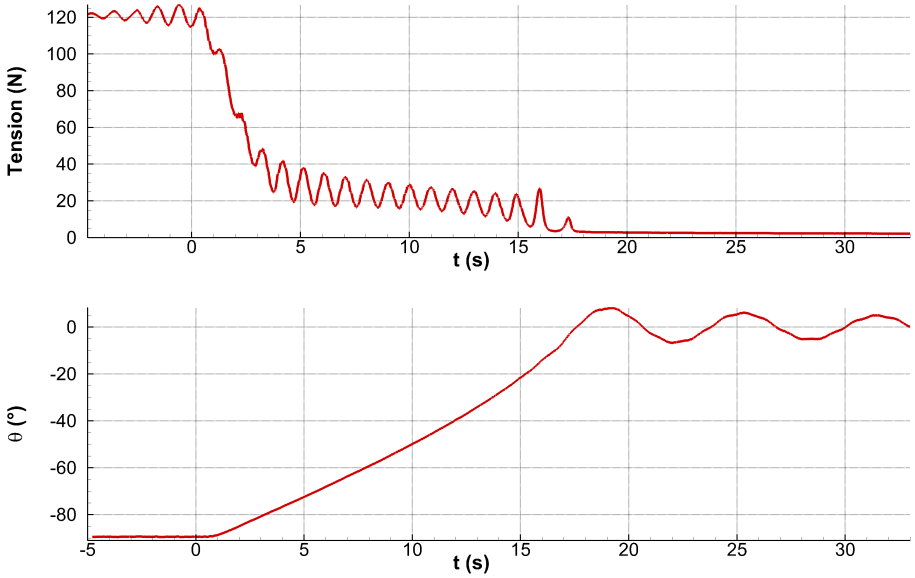


Figure C.39 – Cable tension and rotational motion for Case 13. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.14 Case 14

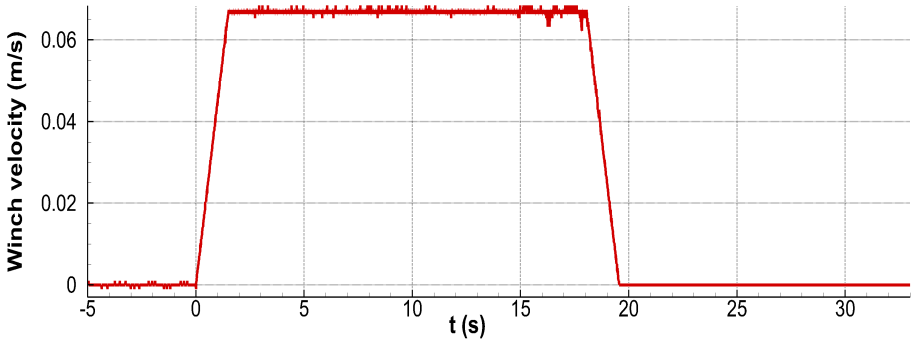


Figure C.40 – Winch velocity for Case 14. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

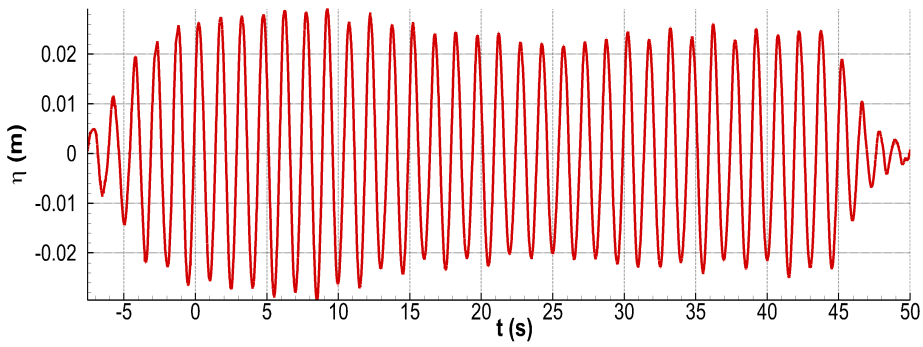


Figure C.41 – Wave elevation for Case 14. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

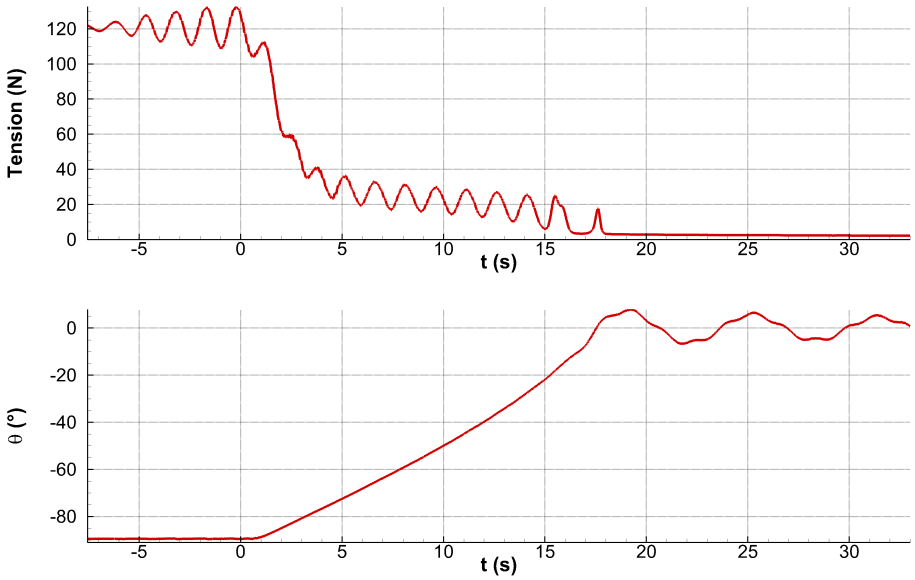


Figure C.42 – Cable tension and rotational motion for Case 14. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.15 Case 15

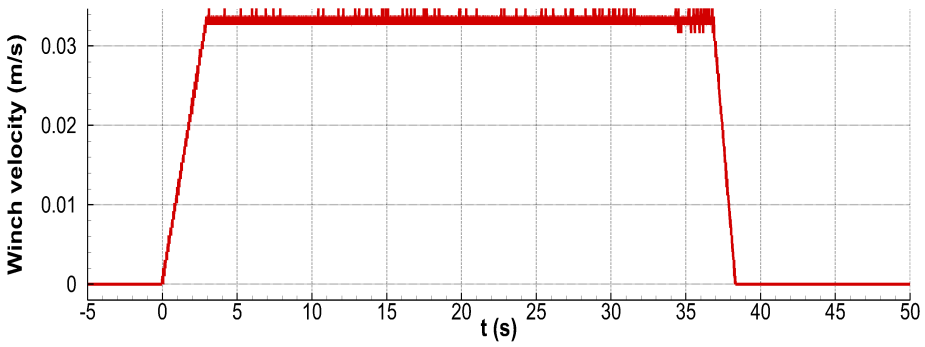


Figure C.43 – Winch velocity for Case 15. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

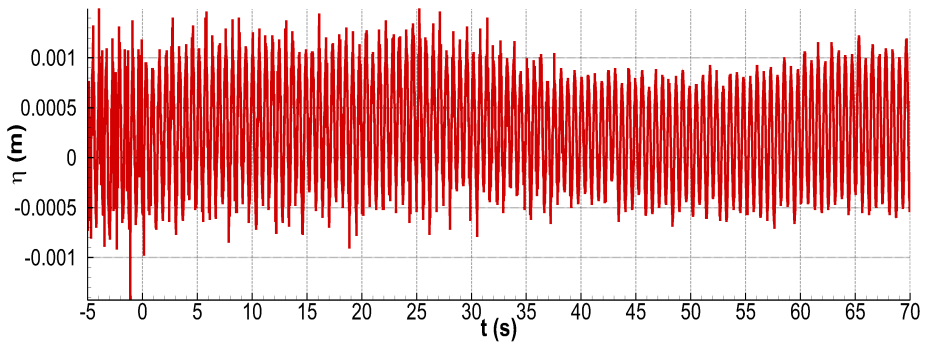


Figure C.44 – Wave elevation for Case 15. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.16 Case 16

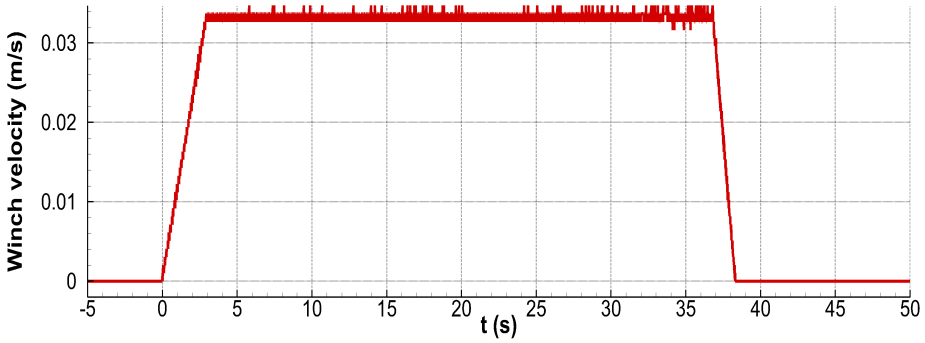


Figure C.45 – Winch velocity for Case 16. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

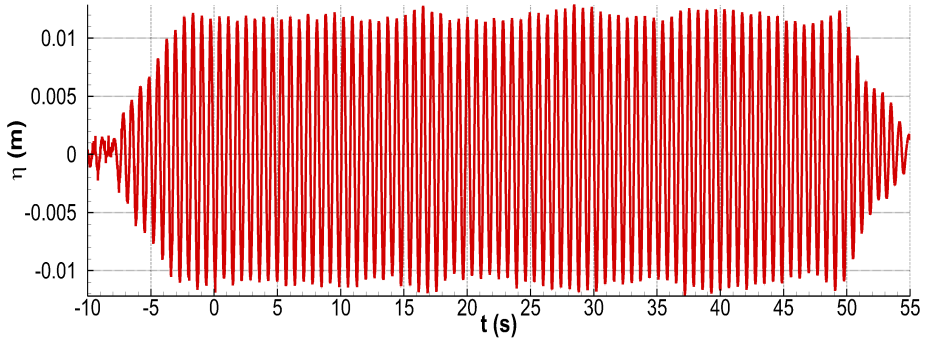


Figure C.46 – Wave elevation for Case 16. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

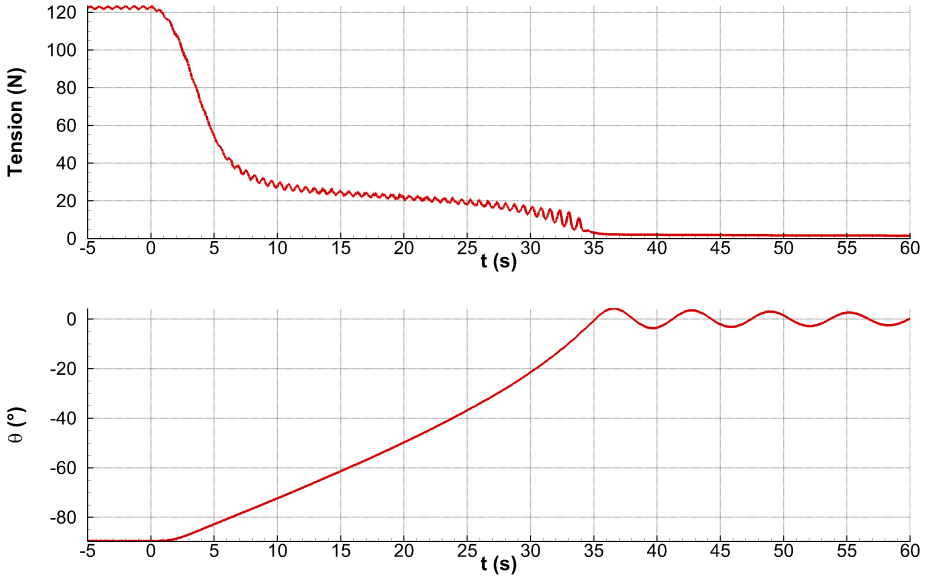


Figure C.47 – Cable tension and rotational motion for Case 16. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.17 Case 17

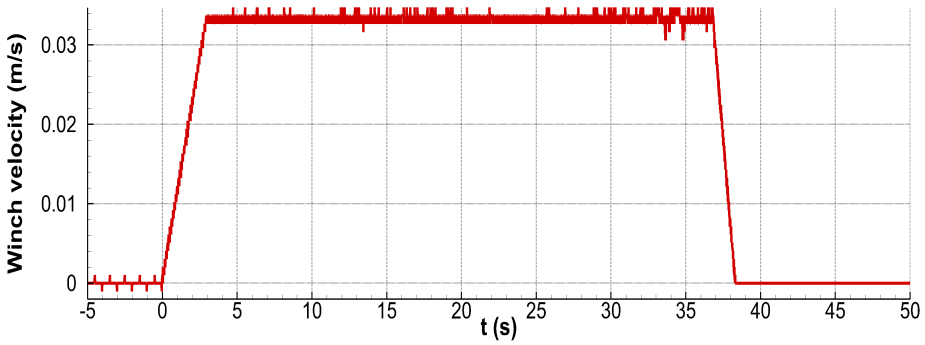


Figure C.48 – Winch velocity for Case 17. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

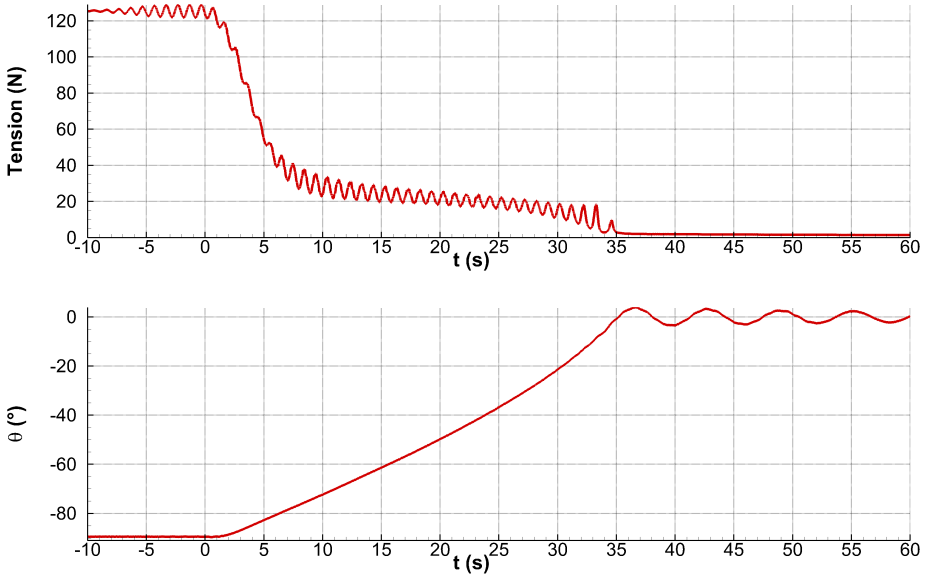


Figure C.49 – Cable tension and rotational motion for Case 17. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.18 Case 18

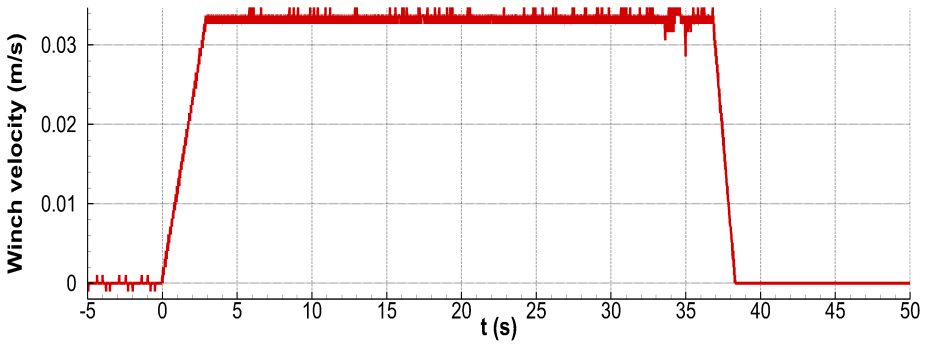


Figure C.50 – Winch velocity for Case 18. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

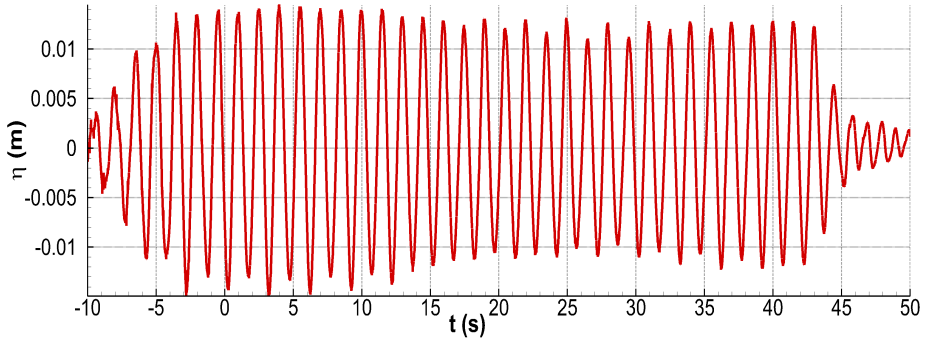


Figure C.51 – Wave elevation for Case 18. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

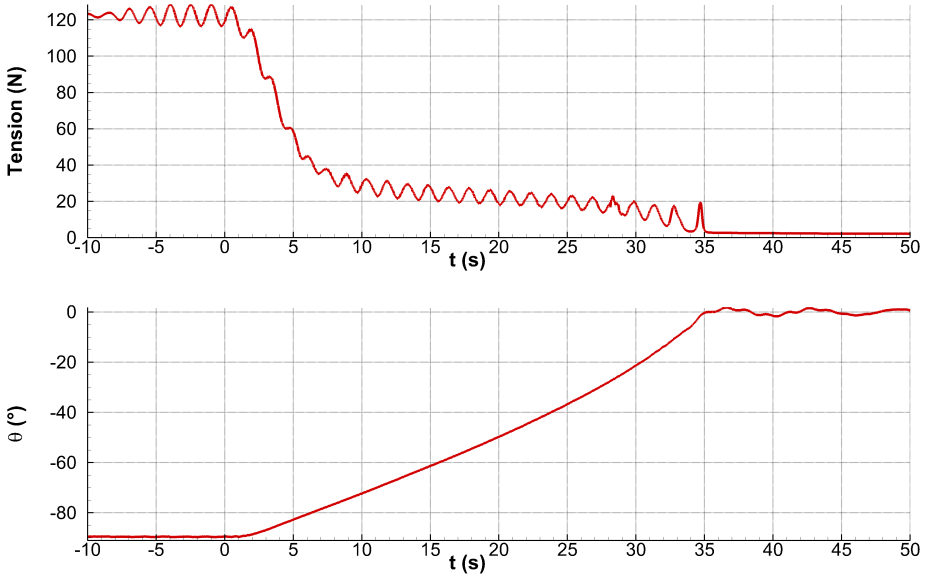


Figure C.52 – Cable tension and rotational motion for Case 18. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.19 Case 19

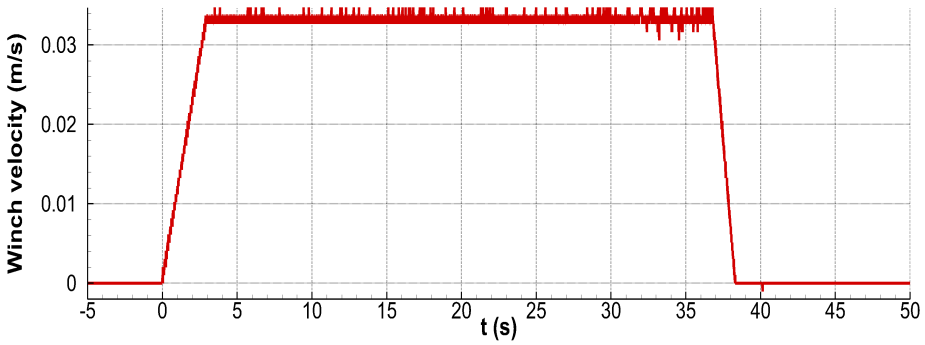


Figure C.53 – Winch velocity for Case 19. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

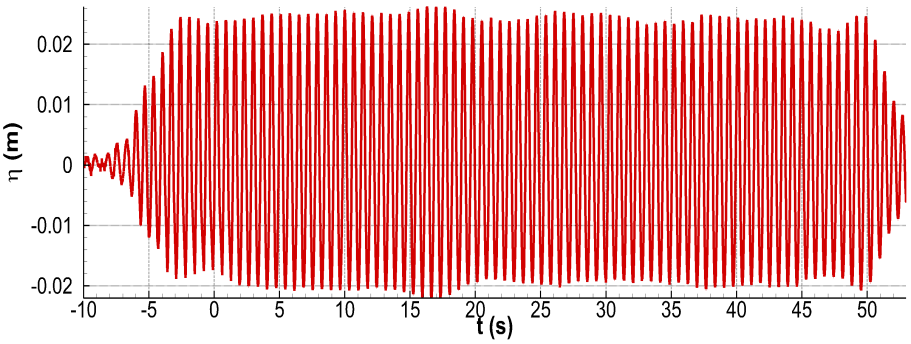


Figure C.54 – Wave elevation for Case 19. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

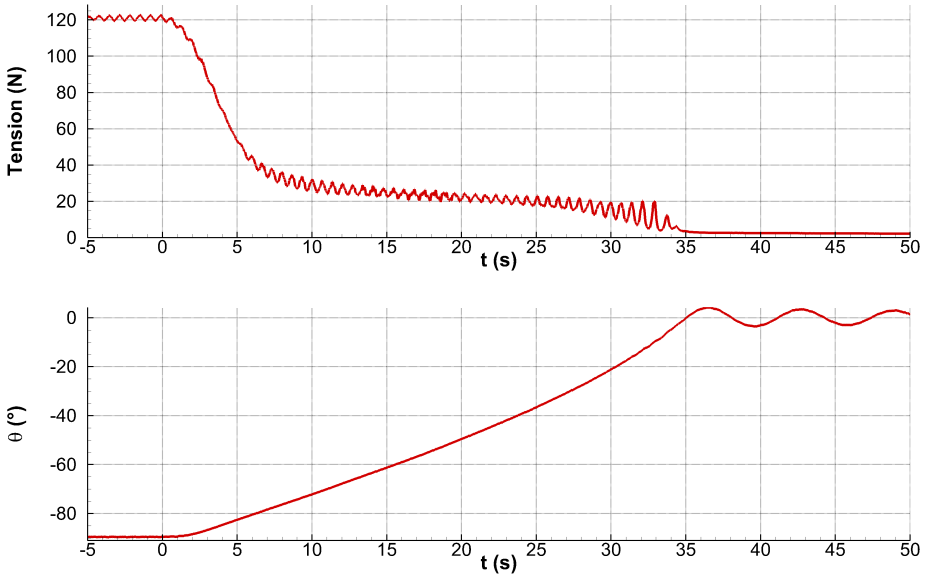


Figure C.55 – Cable tension and rotational motion for Case 19. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.20 Case 20

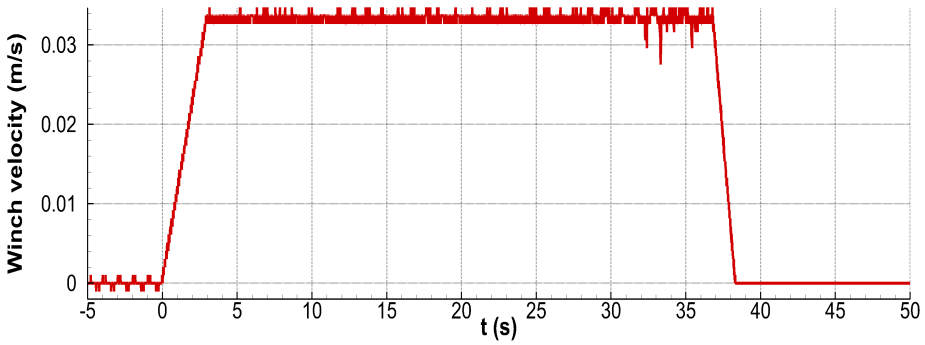


Figure C.56 – Winch velocity for Case 20. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

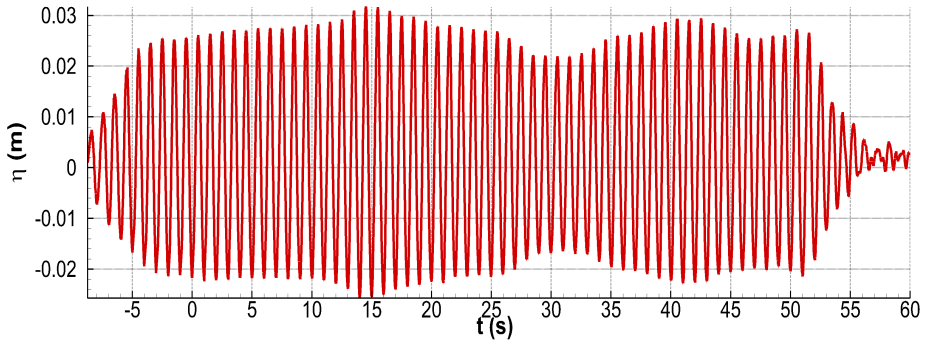


Figure C.57 – Wave elevation for Case 20. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

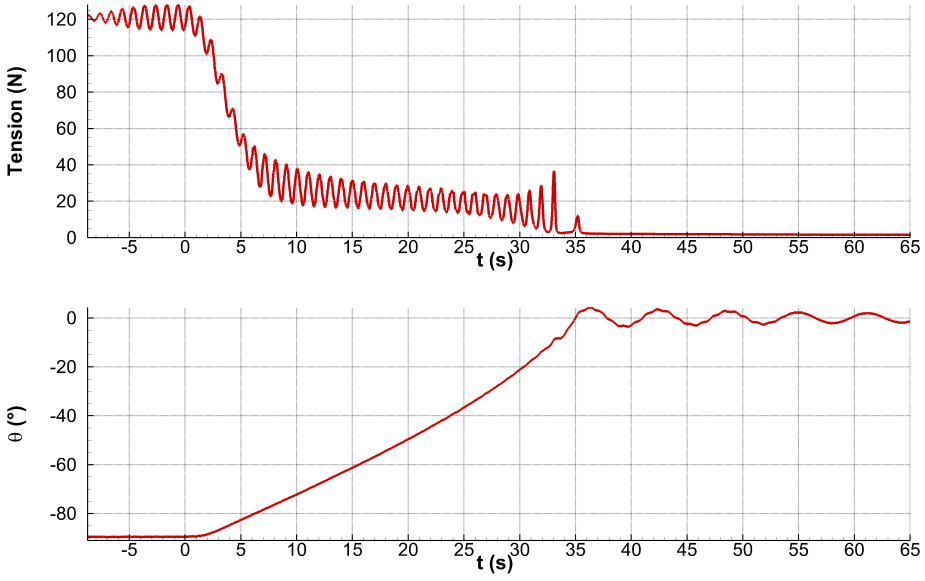


Figure C.58 – Cable tension and rotational motion for Case 20. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

C.21 Case 21

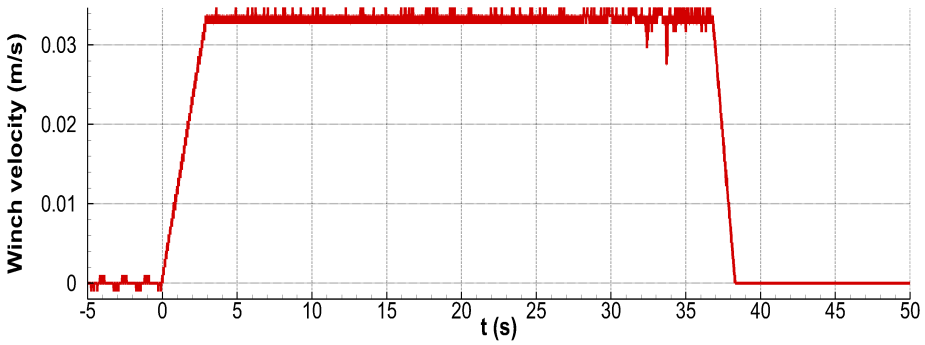


Figure C.59 – Winch velocity for Case 21. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

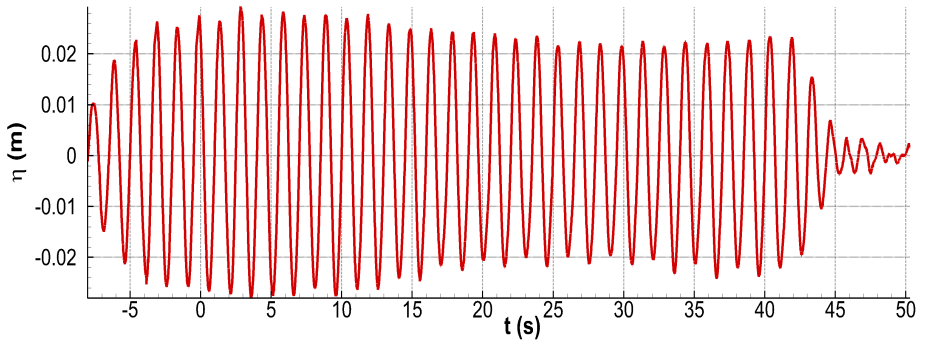


Figure C.60 – Wave elevation for Case 21. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

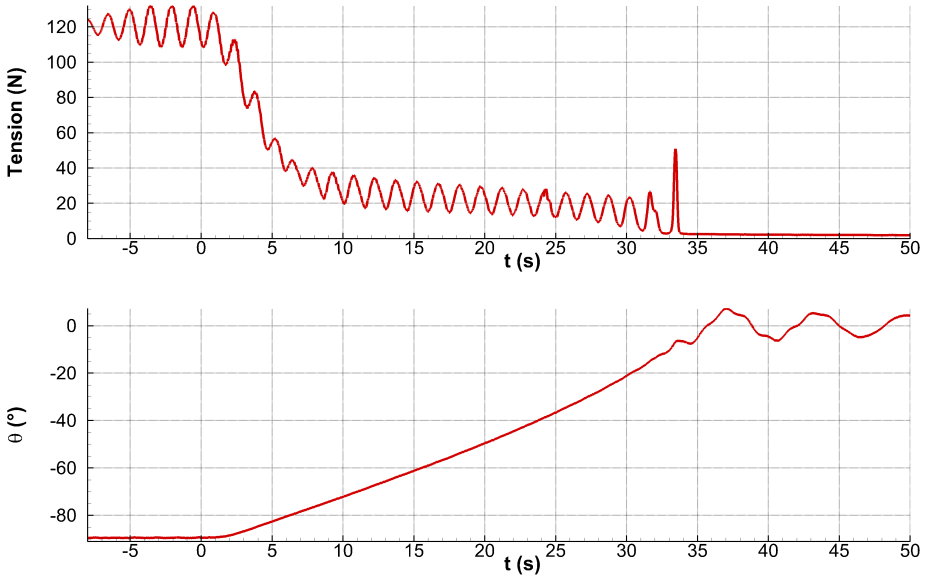


Figure C.61 – Cable tension and rotational motion for Case 21. $t = 0\,\mathrm{s}$ denotes the starting time of the winch.

Appendix D

Résumé substantiel

D.1 Introduction

Avec l'augmentation des dimensions des éoliennes ainsi que leur implantation dans des zones où les mers sont plus agitées et les vents plus forts, les opérations marines deviennent de plus en plus complexes. Le coût de ces opérations représente 30 % du prix final de l'électricité et influe donc significativement sur la compétitivité des énergies marines par rapport aux énergies conventionnelles (charbon, pétrole, gaz, etc.) [1]. D'où la nécessité d'étudier les opérations marines, aussi bien théoriquement qu'expérimentalement ou numériquement, pour en diminuer les risques et les coûts et ainsi favoriser le développement des énergies marines renouvelables.

Il existe différents types d'opérations marines, tels que les opérations de remorquage, de redressement, de déchargement, de dépose de câbles, d'enfoncement de monopieux, etc. Cette thèse porte principalement sur un autre type d'opération : les opérations de descente ou de remontée de colis lourds en mer. Ces dernières sont constituées d'un navire, d'une grue installée sur le navire, d'un câble de levage, d'un treuil, d'élingues ou de gréements et d'un colis à déposer ou remonter. Au cours de ces opérations, différents risques sont présents : l'imprécision de la position du colis, la collision entre le colis et le navire, des efforts brusques dans les câbles, etc. C'est pourquoi il est nécessaire de modéliser et simuler ce type d'opération pour en prévenir les risques.

La simulation numérique des opérations de descente et de remontée de colis a été intensément étudiée dans la littérature. Ces travaux peuvent être classés en trois catégories : les analyses mécaniques, hydrodynamiques, et de contrôle des mouvements.

Une opération de descente nécessite de simuler un système multicorps articulé. Le navire, les câbles et le colis sont en interaction mécanique. Les premiers modèles mécaniques étaient basés sur des approches simplifiées : découplage de la dynamique du colis de celle du navire [12], linéarisation des équations du mouvement [15] ou problème bidimensionnel [17]. Ensuite, des modèles tridimensionnels, non-linéaires et résolvant l'interaction mécanique entre les corps apparurent [16, 19, 20]. Ces études avaient pour point commun de maintenir le colis en l'air, de sorte que seul le navire subissait des efforts hydrodynamiques. Afin d'écrire et de résoudre automatiquement les équations de mouvement, des algorithmes de mécanique multicorps furent utilisés, basées sur les équations de Newton-Euler [26] ou de Euler-Lagrange [27].

Concernant le modèle hydrodynamique d'une opération de descente de colis, la plupart des travaux utilisèrent la théorie des écoulements potentiels linéarisés dans le domaine fréquentiel [7, 16, 19, 25]. Cette théorie suppose que les corps ont un mouvement

de faible amplitude et que la houle a une faible cambrure. Ces hypothèses sont valides dans le cas où le colis est en l'air et en mer calme mais quand le colis est dans l'eau en étant descendu ou remonté alors ces hypothèses sont violées (grand mouvement relatif entre les corps). C'est pourquoi Hannan [11] utilisa une théorie des écoulements potentiels non-linéaires dans sa thèse de doctorat car les hypothèses de la théorie linéaire n'ont pas à être satisfaites. Néanmoins le modèle mécanique était constitué d'une barge fixe et d'un câble rigide.

Les logiciels commerciaux de simulation numérique offshore (*OrcaFlex* [38], *DeepLines* [39], *SIMO* [36], etc.) sont capables de modéliser les interactions mécaniques mais sont tous basés sur une approche d'écoulement potentiel linéarisé. C'est pourquoi il est nécessaire de développer un outil de simulation numérique prenant en compte les interactions mécaniques et étant cohérent d'un point de vue hydrodynamique pour modéliser le couplage fluide-structure lors d'une opération de descente ou de remontée de colis.

Les objectifs de la thèse sont :

- L'implémentation d'un modèle de câble et de treuil dans le logiciel de simulation mécanique utilisé (*InWave*);
- L'extension du logiciel de simulation hydrodynamique (*WS_CN*), basé la théorie des écoulements potentiels instationnaires satisfaisant l'hypothèse de faible perturbation, dite «*weak-scatterer* », aux simulations multicorps;
- Le couplage entre *InWave* et *WS_CN* dans le but de simuler numériquement des opérations de descente et de remontée de colis;
- La comparaison entre une modélisation classique utilisant la théorie des écoulements potentiels linéarisés et l'outil développé pour quantifier l'intérêt de la théorie instationnaire «*weak-scatterer* ».

D.2 Dynamique des systèmes multicorps

La théorie de mécanique multicorps utilisée dans cette thèse permet de simuler des systèmes multicorps articulés de type arbre cinématique, c'est-à-dire qu'ils ne disposent pas de boucles cinématiques. Concrétement, chaque corps du système multicorps ne posséde qu'un unique prédécesseur mais potentiellement plusieurs successeurs. Dans le cas où plusieurs successeurs seraient présents, des boucles cinématiques apparaîtraient. Les corps sont considérés comme rigides et séparés les uns des autres par une articulation autorisant un seul degré de liberté soit en translation (liaison glissière) soit en rotation (liaison pivot). Chaque corps est repéré via des coordonnées relatives par rapport à son prédécesseur en utilisant les paramètres modifiés de Denavit-Hartenberg. Le corps racine (ou la base) du système multicorps est repéré avec des coordonnées Cartésiennes. Cette modélisation permet d'écrire les équations cinématiques récursives du système multicorps.

L'algorithme utilisé se nomme l'Algorithme aux Corps Rigides Composites (*ACRC*) et permet la résolution d'un problème de dynamique multicorps directe, c'est-à-dire où les efforts extérieurs et intérieurs sont connus et l'accélération des corps inconnue. L'équation du mouvement du système multicorps est :

$$\mathbf{H}(\mathbf{q})\begin{pmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{\Gamma} \end{pmatrix} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \tag{D.1}$$

avec :

- $\mathbf{H}$ la matrice masse généralisée du système multicorps;
- $\mathbf{C}$ le vecteur des efforts extérieurs et des accélérations d'entrainement et de Coriolis;
- $\mathbf{\Gamma}$ le vecteur des efforts internes projectés selon l'axe des liaisons;
- ${}^0\dot{\mathbf{V}}_0$ l'accélération de la base par rapport au repère inertiel et exprimée dans le repère de la base;
- $\ddot{\mathbf{q}}$ le vecteur des accélérations articulaires.

L'*ACRC* permet le calcul de la matrice $\mathbf{H}$ et du vecteur $\mathbf{C}$ à partir des équations cinématiques récursives. Connaissant $\mathbf{\Gamma}$, les accélérations peuvent être calculées. L'*ACRC* est implémenté dans le logiciel *InWave*. L'intégration en temps se fait grâce à un schéma de type Runge-Kutta d'ordre 4.

D.3 Dynamique de câbles

Les câbles sont simulés en utilisant un modèle dynamique, basé sur la discrétisation du câble en éléments et en noeuds, où la masse est concentrée. Chaque élément comprend un ressort et un amortisseur, afin de modéliser l'élasticité du câble et les frottements internes. Lorsque cette approche utilise des coordonnées Cartésiennes pour repérer les noeuds, elle est appelée la méthode «*lumped mass* ». Un programme, nommé *CableDyn* a été développé en ce sens. Il est validé par comparaisons avec les résultats numériques et expérimentaux fournis par Buckham [64]. Un bon accord est observé.

L'Algorithme aux Corps Rigides Composites est également utilisé pour simuler des câbles par cette même approche, via *InWave*. Chaque élément câble est modélisé par trois articulations : deux liaisons pivots et une liaison glissière. Les efforts internes à la liaison glissière permettent de modéliser l'élasticité et les frottements. Ils s'expriment comme suit :

$$\Gamma_j = \begin{cases} -\dfrac{EA}{L_u}(q_j + L_u) - \dfrac{CA}{L_u}\dot{q}_j & \text{si } |q_j| \geqslant L_u; \\ -\dfrac{CA}{L_u}\dot{q}_j & \text{sinon.} \end{cases} \tag{D.2}$$

où E, C, A et L_u représentent le module d'Young, le coefficient d'amortissement, l'aire de la section du câble et la longueur à vide de l'élément câble. L'indice j indique le numéro du corps dans le système multicorps. Il n'y a pas d'efforts internes dans les liaisons pivots, la flexion et la torsion du câble ne sont donc pas modélisées.

La méthode «*lumped mass*» et l'*ACRC* sont comparés et prouvent, par leur accord, que les approches sont identiques.

Afin d'accélérer les calculs de l'*ACRC* lorsque des câbles sont simulés, une réécriture de l'algorithme dynamique est effectuée. Une articulation câble est créee pour séparer deux corps par trois liaisons mécaniques, soit un élément câble. Une accélération de 49.2 % du temps de calcul est notée.

Enfin, un modèle de treuil est développé pour simuler le déroulement et l'enroulement d'un câble. Il se base sur l'ajout ou la suppression d'éléments câble pendant la simulation temporelle. L'élément câble connecté au treuil voit sa longueur à vide être modifiée. Les méthodes «*lumped mass*» et multicorps (*ACRC*) donnent les mêmes résultats pour ce modèle de treuil après comparaison.

D.4 Hydrodynamique

La théorie hydrodynamique utilisée dans cette thèse est la théorie des écoulements potentiels satisfaisant l'hypothèse de faible perturbation, dite «*weak-scatterer*». Le potentiel de vitesse (ϕ) et l'élévation de surface libre (η) sont décomposés en une partie incidente connue et une partie perturbée inconnue.

$$\begin{cases} \phi &= \phi^I + \phi^P \\ \eta &= \eta^I + \eta^P \end{cases} \tag{D.3}$$

avec :

- $(*)^I$ la composante incidente;
- $(*)^P$ la composante perturbée.

L'hypothèse de faible perturbation implique que la composante perturbée est faible devant la composante incidente.

$$\begin{cases} \phi^P = o(\phi^I) \\ \eta^P = o(\eta^I) \end{cases} \tag{D.4}$$

Les équations limites de surface libre sont ensuite linéarisées au niveau de la surface libre incidente. Cette méthode est cohérente en cas de mouvements de grandes amplitudes et de houles cambrées, contrairement à la théorie des écoulements potentiels linéarisés.

La composante perturbée du potentiel de vitesse est calculée à partir d'une méthode aux éléments frontières en utilisant une fonction de Green de type Rankine. Le couplage fluide-structure est basé sur une approche monolithique et nécessite l'utilisation d'une deuxième méthode aux éléments frontières pour calculer la dérivée temporelle du potentiel de vitesse. L'implémentation de cette théorie hydrodynamique a été effectuée dans le code *WS_ CN* avant le début de cette thèse, dans le cas d'un seul corps immergé ou flottant [44, 45].

La contribution principale apportée à *WS_ CN* est l'extension du solveur à des simulations multicorps en cas de mouvements de grandes amplitudes. Une première validation est entreprise par comparaison avec les résultats expérimentaux de Watai et al. [145]. Le dispositif expérimental comprend deux cylindres dont l'un est fixe et l'autre a un mouvement harmonique de grande amplitude. Une houle régulière est présente de direction l'axe centre à centre des cylindres en configuration initiale. L'approche hydrodynamique utilisée dans notre travail permet d'obtenir de très bons résultats numériques par rapport aux données expérimentales, que ça soit en terme d'efforts hydrodynamiques ou d'élévations de surface libre. Les interactions hydrodynamiques entre les différents corps sont bien captées. Une validation de l'extension aux simulations multicorps est également faite en mouvement libre, par comparaison avec *InWave* couplé au logiciel *Nemoh*. Ce dernier est basé sur la théorie des écoulements potentiels linéarisés. Les résultats numériques des deux théories sont en accord. D'autres développements dans *WS_ CN* sont effectués : l'implémentation d'un algorithme de remaillage de surface libre, la parallélisation du calcul des coefficients d'influence et la réduction de la quantité de mémoire utilisée via l'emploi d'un solveur de systèmes linéaires creux.

D.5 Interaction fluide-structure

La théorie des écoulements potentiels basés sur l'hypothèse de faible perturbation et l'Algorithme aux Corps Rigides Composites sont couplés pour calculer l'interaction fluide-structure d'un système multicorps ayant un mouvement relatif de grande amplitude dans une houle cambrée . Un couplage fort est choisi pour sa robustesse. L'équation de couplage, qui rassemble l'équation du mouvement du système multicorps et le deuxième problème aux éléments frontières, est établie et démontrée. Ces développements engendrent la création d'un nouveau logiciel nommé *InWaveS_ CN*. Comme *InWave* est implémenté en C++ et *WS_ CN* en Frotran, il est nécessaire d'assurer les communications entre les deux solveurs. Afin de garantir la modularité du logiciel, le code d'intégration est en Python. Une validation est effectuée par comparaison entre *InWaveS_ CN* et *WS_ CN* dans un cas multicorps sans interaction mécanique. Ceci permet de prouver la justesse de la démonstration et de l'implémentation de l'équation de couplage. Le couplage fort est comparé à d'autres formes de couplages (faibles, explicite et implicite) pour en démontrer la pertinence.

Le code de simulation de câbles, *CableDyn* est aussi couplé à *InWaveS_ CN* en utilisant un couplage faible explicite.

D.6 Développement d'un nouveau mailleur

Le mailleur de *WS_ CN*, implémenté avant le début de ce travail de doctorat [45], est basé sur une approche semi-analytique pour calculer la position de la courbe d'intersection entre le corps flottant et la surface libre incidente. Les maillages des corps et de la surface libre sont générés via une méthode d'avance de front. Un manque de robustesse de cette méthode est constaté, notamment pour mailler des géometries complexes. Une nouvelle stratégie de maillage est développée en supposant que le maillage du corps entier a été généré au préalable en utilisant un logiciel de maillage externe. Le maillage du corps entier est alors coupé au niveau de la surface libre incidente par un algorithme de découpe de maillages. Cette méthode déforme significativement le maillage au niveau de la courbe d'intersection avec la surface libre. Pour améliorer la qualité du maillage du corps au niveau de la découpe, des algorithmes de fusion de facettes et de déformation de maillages sont utilisés. Puis, le maillage du corps est connecté au maillage de surface libre, généré par la méthode d'avance de front comme utilisée initialement. L'outil de découpe de maillage se nomme *Meshmagick* et est incorporé à *InWaveS_ CN* en tant que nouveau module. Deux cas tests sont effectués afin de valider la méthode par des comparaisons entre le mailleur initial et la nouvelle stratégie de maillage. Un très bon accord est obtenu, validant la nouvelle approche de génération de maillages pour *WS_ CN*.

D.7 Expériences

Afin de valider le couplage entre *InWave* et *WS_ CN*, implémenté dans *InWaveS_ CN*, des essais en bassin ont été menés à l'*École Centrale de Nantes*. Ces expériences représentent une opération de redressement dans la houle d'un flotteur d'éolienne de type spar (cylindre). Initialement, le cylindre est maintenu horizontal au moyen d'un câble, puis ce câble est déroulé grâce à un treuil entrainant la rotation autour d'un axe fixe du cylindre, qui, finalement, atteint sa position d'équilibre verticale. Ces expériences sont pertinentes au regard du modèle développé dans ce travail de thèse car il y a

un système mécanique articulé (le cylindre et l'axe fixe), un câble, un treuil et un corps soumis à des efforts hydrodynamiques avec un mouvement libre de grande amplitude.

InWaveS_CN est utilisé pour simuler numériquement ces expériences. Tous les modules le consituant sont sollicités : *InWave* pour la dynamique multicorps, *CableDyn* pour la dynamique du cable et du treuil, *WS_CN* pour l'hydrodynamique, *Meshmagick* pour la génération de maillages complexes. Cinq cas sont présentés : deux sans houle et avec différentes lois de vitesses de déroulement du câble et trois en présence d'une houle incidente régulière avec différents paramètres de houle. Les effets de la vitesse de déroulement, de la rampe sur cette dernière et des vagues sont bien capturés. Mais, quand une extrémité du cylindre traverse la surface de l'eau, des erreurs numériques importantes apparaîssent. Néanmoins, la comparaison entre les simulations numériques et les données expérimentales offre une première et prometteuse validation d'*InWaveS_CN*.

Titre : Simulation numérique des opérations d'installation pour les fermes d'éoliennes offshore

Mots-clés : Opérations marines, Câbles, Ecoulement potentiel, Weak-scatterer, Interaction fluide-structure

Résumé : L'éolien offshore est l'énergie marine la plus avancée et utilisée dans le monde. Afin d'accroître l'énergie extraite du vent, les dimensions des éoliennes deviennent plus importantes et les parcs éoliens sont installées de plus en plus loin des côtes, où les mers sont plus agitées et les vents plus forts. De fait, les opérations marines sont plus complexes et plus chères et les fenêtres météo sont écourtées et se raréfient.

Dans le cadre de cette thèse, un logiciel de simulation numérique des opérations marines est développé, en particulier pour des applications de descentes et de remontées de colis lourds. L'Algorithme aux Corps Rigides Composites, implémenté dans le logiciel *InWave*, est utilisé pour modéliser le système multicorps. Un modèle de câble et de treuil est développé, suivant la théorie multicorps utilisée, et comparé à la théorie câble classique dite « lumped mass ». Les efforts hydrodynamiques ainsi que les interactions hydrodynamiques sont modélisés par une théorie potentiel instationnaire satisfaisant l'hypothèse de faible perturbation, dite « weak-scatterer ». L'approche « weak-scatterer » du logiciel *WS_CN* est étendue aux simulations multi-flotteurs et validée par comparaison avec des données expérimentales.

InWave et *WS_CN* sont couplés afin de résoudre l'interaction houle-structure pour des systèmes multicorps articulés en mer. Un couplage fort est adopté pour sa robustesse. L'équation de couplage est établie et validée via des comparaisons avec *WS_CN*. Le logiciel ainsi crée se nomme *InWaveS_CN* et utilise un code d'intégration en Python.

Une nouvelle stratégie de maillage, basée sur un algorithme de découpe de maillages et une méthode par avance de front, est développée dans *WS_CN*.

Enfin, des essais en bassin d'une opération de redressement ont été menés à l'*ECN*. La comparaison entre les simulations numériques et les données expérimentales offre une première et prometteuse validation d'*InWaveS_CN*.

Title : Numerical simulation of installation operations for offshore wind farms

Keywords: Marine operations, Multibody dynamics, Cable dynamics, Potential flow, Weak-scatterer, FSI

Abstract: Offshore wind represents the most advanced and used marine energy in the world. To increase the wind power extraction, turbines grow in size and wind farms are installed further offshore in presence of rough seas and strong winds. Marine operations become more challenging and expensive, weather windows are shorter and less frequent.

This PhD work focuses on the development of a numerical tool to simulate marine operations with consistency, in particular lowering and lifting operations. The Composite-Rigid-Body Algorithm, implemented in the numerical tool *InWave*, is used to model multibody systems. A cable model and a winch model are developed following this multibody approach and compared to the classical low-order lumped mass theory. Hydrodynamic loads and hydrodynamic interactions are simulated using an unsteady potential flow theory based on the weak-scatterer hypothesis, implemented in the numerical tool *WS_CN*. This approach is extended to multibody simulations and validated with comparisons to experimental data.

InWave and *WS_CN* are coupled to solve wave-structure interaction for articulated multibody systems with large relative motions in waves. A tight coupling is selected for its robustness. The coupling equation is derived and validated from comparisons with *WS_CN*. This leads to the creation of a new numerical tool, *InWaveS_CN*, using Python as glue code language.

A new mesh strategy, based on the coupling between a panel cutting method and an advance front method, is developed in *WS_CN*.

Experiments of an upending operation were conducted at *Ecole Centrale de Nantes*. The comparison between the numerical simulations and the experimental data leads to a first and promising validation of *InWaveS_CN*.