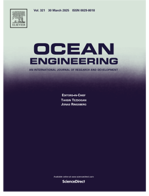
Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Research paper

A methodology for CFD simulations of floating structures in waves coupled with mooring dynamics

Tanausú Almeida-Medina [a,b,*], Jose Luis Cercos-Pita [b], Anne Gosset [a], Vicente Díaz-Casás [a]

[a] *Universidade da Coruna, Campus Industrial de Ferrol, CITENI, 15403, Spain*
[b] *Coremarine Solutions SL, Madrid, Spain*

ARTICLE INFO

Keywords:
Coupled simulations
FloatStepper
FoamMooring
Moordyn
Mooring dynamics
OpenFOAM
Wave-structure interaction

ABSTRACT

Prediction of the responses in wave fields of moored floating structures, such as floating offshore wind turbines (FOWT), wave energy converters (WEC), aquaculture farms (AF), or floating breakwaters (FB), requires high-fidelity yet efficient coupled analysis between Computational Fluid Dynamics (CFD) and mooring dynamic models. Among the coupled models available to users for the open-source CFD software OpenFOAM and mooring libraries like MoorDyn, we can find foamMooring and FloatStepper. These address issues such as added mass instability, and offer different computational methods for free-surface capturing and rigid body motion. If we also consider the various dynamic mesh techniques, users have multiple options for the general configuration of a simulation. Bearing this in mind, the present study proposes an efficient and accurate methodology for simulations with OpenFOAM coupled with the lumped-mass mooring model Moordyn for a simple floating structure, with the intention of applying it to more complex structures, or even performing survivability studies. Pursuing this goal, different dynamic mesh techniques, free-surface capturing methods and the available algorithms for rigid body motions including an added-mass correction algorithm, are investigated systematically and validated against experimental data. In the course of this work, new features have been added to the lastest version of Moordyn and also coupled with the non-iterative algorithm FloatStepper. The surface elevation, floating body motions and mooring line tensions were considered to compare the different settings. The results showed that all the tested options were accurate, but the overset technique was the most efficient in terms of computational time.

1. Introduction

Floating Offshore Wind Turbines (FOWT) are designed to operate in deep waters to exploit more favorable wind resources. However, this environment is characterized by challenging and complex metocean conditions, posing difficulties in designing moored floating foundations. Numerical approaches to the hydrodynamics of these structures often employ linear wave models based on potential flow theory and/or the semi-empirical Morison equation. The second-order hydrodynamic model, by incorporating quadratic transfer functions (QTF) or second-order terms, significantly improves predictions compared to the linear models. This enhancement is particularly evident in scenarios involving small wave amplitudes and small structure motions, where the assumptions of weak non-linearity remain valid. These approaches are advantageous for their low computational cost and suitability for predicting FOWT's hydrodynamic behavior in operational conditions. Nevertheless, in deeper waters, survivability analysis becomes crucial, where large amplitude waves give rise to highly non-linear phenomena such as wave run-up, breaking waves, and wave slamming. These can significantly impact the structure. For such scenarios, traditional tools based on potential flow theory struggle to capture non-linear phenomena, requiring the use of high-fidelity methods like Computational Fluid Dynamics (CFD) to provide more accurate estimations, see Zhang et al. (2024) or Huang et al. (2022). Moreover, considering the significant impact of mooring restraints on the dynamic behavior of floating structures, see Brown and Mavrakos (1999), the use of a coupled solver becomes essential to accurately capture non-linear phenomena.

Therefore the integration of mooring dynamics into CFD codes represents a significant progress, with implications for the understanding of non-linear hydrodynamic in floating structures. In recent years, there

[*] Corresponding author.
E-mail addresses: tanausu.almeida@udc.es (T. Almeida-Medina), jlc@core-marine.com (J.L. Cercos-Pita), anne.gosset@udc.es (A. Gosset), vicente.diaz.casas@udc.es (V. Díaz-Casás).

https://doi.org/10.1016/j.oceaneng.2025.121764
Received 8 January 2025; Received in revised form 21 May 2025; Accepted 3 June 2025
Available online 13 June 2025

has been a notable surge in collaborative efforts by both industry and academics to integrate mooring models in CFD codes. By considering the dynamic nature of the mooring system within the CFD simulations, a more comprehensive understanding of how these structures respond under realistic and challenging conditions was reached, ultimately contributing to the improvement of their design, safety, and performance. These efforts have resulted in the development and implementation of advanced coupling techniques in open-source and propietary CFD codes. Among the latter, one can cite Star-CCM+, ANSYS Fluent/CFX, or FLOW-3D, as discussed in Zeng et al. (2024). However, to the best of the authors knowledge, these codes exhibit a lack of an accurate mooring dynamics model. In particular, a proper modeling of catenaries, seabed interaction, inclusion of non-linear material for the lines are scarcerly proposed. In addition, a quasi-static approach is often used instead of the required dynamic modeling. On the other hand, we find the open-source libraries such as OpenFOAM, in which greater efforts have been devoted in the coupling with dynamic mooring models.

The lumped-mass (Hall et al., 2020(@), higher-order finite element (Palm et al., 2016) and recently introduced finite difference method (Chen et al., 2018) are the most commonly used formulations for the mooring coupled models with OpenFOAM. Chen and Basu (2018) developed a dynamic mooring model based on the finite difference method, which accounted for current loads. The study investigated the effects of nonlinear mooring dynamics on a spar-type FOWT in the presence of currents. They concluded that current loads can significantly affect the restoring effect of the mooring system, thereby influencing the response of the spar-type FOWT. Their open source code called *openmoor* can be accessed at https://github.com/chen-lin/openmoor. In Palm and Eskilsson (2018), recent developments of an hp-adaptative high-order discontinuous Galerkin mooring solver *Moody* is presented. An OpenFOAM-Moody analysis for a cylindrical buoy in regular waves can be found in Palm et al. (2016), and validated against experimental data (Paredes et al., 2016). Their results showed that in spite of a high computational cost for long-term estimations, the CFD properly captured non-linear effects. The open-source mooring dynamics model, MoorDyn, based on lumped mass approach, was successfully validated against experimental results in Hall et al. (2020(@). The model uses a lumped-mass formulation to simulate axial elasticity, hydrodynamic drag and added mass based on Morison's equation and reactions from the seabed. A second version of the model is currently under development, which incorporates cable bending stiffness for power cable simulation and wave kinematics into the model (Hall, 2020; Hall et al., 2021).

Recently, several authors have studied the motion of floating bodies and tensions in their mooring lines in a regular wave field by using the coupled solver OpenFOAM-Moordyn. In Lee et al. (2021), a mesh morphing technique was used to accomodate the motion of a floating body, that required customizing and recompiling the flow solver. Chen and Hall (2022), in addition to using the first version of the lumped-mass mooring model MoorDyn, coupled the quasi-static mooring model MAP++ (Masciola et al., 2013) and the finite element model Moody (Palm et al., 2016) for the body motion solver in OpenFOAM. The resulting coupled code, foamMooring, is open-source and can be accessed at https://gitlab.com/hfchen20/foamMooring. Most recently, this code has been extended to multiple moored floating structures in Chen et al. (2024). Jiang and Moctar (2022) validated the implementation of an implicit coupling approach between the dynamic model MoorDyn and OpenFOAM, namely the mooring-sixDoFRigidBodyMotion solver. In addition to validating the model for a single body, they also predicted successfully the motion of a moored and articulated multibody in waves. Aliyar et al. (2022), developed a coupling between the MoorDyn libraries and an in-house CFD code called foamStar, which is built upon the OpenFOAM open-source libraries. In Jeon et al. (2023), the motion responses of a cubic-shaped box and a semi-submersible platform are analyzed, both attached with tensioned mooring lines using an in-house solver that couples OpenFOAM and MoorDyn. More recently, a non-iterative algorithm called FloatStepper, developed by Roenby et al. (2024), has been coupled with the first version of Moordyn. The innovation of this 6-degree of freedom (DoF) algorithm is that it circumvents the well-known added mass instability problem when lightweight floating bodies interact with heavy fluids, see Devolder et al. (2017).

Thus, there is a wide variety of choices available to the user for configuring the simulations of FOWT, from methods for free-surface capture to solvers for correcting added mass instability. In addition, to the best of the author's knowledge, no simulations have been conducted with the second version of Moordyn, which presents significant improvements and corrections compared to the first version. The aim of the present paper, as a first step in investigating the coupled dynamic behaviour of more complex floating structures, is to develop and validate an efficient methodology for such simulations, using a moored cubic-shaped box as a benchmark case. Pursuing this objective, the present study compares in a systematic way various dynamic mesh techniques such as morphing and overset, different methods for capturing the free surface (VoF and isoAdvector), as well as the FloatStepper algorithm to study the influence of the added mass correction. New features have been implemented in the latest version of MoorDyn (version 2), and it has been coupled with the FloatStepper solver. The numerical results of surface elevation, floating body motion and mooring line tensions are compared with other available numerical simulations in scientific literature as (Chen and Hall, 2022; Jeon et al., 2023) and with experimental measurements in Wu et al. (2019).

This paper is outlined as follows. Section 2 summarizes the underlying numerical methods, including the flow governing equations, rigid body dynamic equations, FloatStepper algorithm, dynamic mesh methods, and the mooring dynamics model. In Section 3, the case setup is introduced along with a description of the numerical model, computational settings, wave dimensionless parameters, boundary conditions, and numerical schemes. The solver parameters used in MoorDyn and in the CFD simulations are also detailed. Section 4 presents the results of the different models validated against numerical and experimental data for a floating box with four catenary lines exposed to regular waves. In this section, the surface elevation, floating body motions (surge, heave, and pitch), and the tensions in the mooring lines at the anchor and fairlead are presented. To ensure a consistent and transparent comparison, a single variable control strategy was systematically adopted: surface elevation was used to assess free-surface capturing methods, floating body motions were used to evaluate dynamic mesh techniques, and mooring line tensions served to compare the performance of MoorDyn version 2. Complementary studies were also conducted to explore the influence of the Courant number and axial stiffness, further supporting the robustness of the proposed methodology. The sensitivity of the results to the models and parameters used is then analyzed, leading to the definition of an optimum simulation approach. The paper closes with conclusions in Section 5.

2. Numerical model

This section begins with a brief overview of the governing equations for incompressible two-phase flow, followed by the derivation of rigid body dynamic equations and mesh dynamics technique. It concludes with the presentation of the implemented dynamic mooring model.

2.1. Flow governing equations

An unsteady, isothermal, and incompressible two-phase flow is considered here. The governing equations are the mass continuity and momentum conservation in an Eulerian frame of reference:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1a}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p^* + \rho \mathbf{g} + \nabla \cdot (\mu \nabla \mathbf{u}) + \mathbf{f} \tag{1b}$$

where ρ is the fluid density, $\mathbf{u}$ the velocity field, $\mathbf{g}$ the gravity acceleration, $\mathbf{f}$ represents other forces such as surface tension and external forces and $p*$ stands for the pseudo-dynamic (or modified) pressure.

Despite the wide availability of turbulence models in OpenFOAM, no universally accepted model has yet been established for accurate wave propagation in two-phase flows. As discussed by Larsen and Fuhrman (2018), conventional turbulence closures often overestimate turbulent mixing beneath surface waves, making their applicability in such contexts questionable. Consequently, several previous studies have opted to model the flow as laminar, including (Chen and Hall, 2022; Pinguet et al., 2020; Roenby et al., 2024), while others, such as (Palm et al., 2016; Rentschler et al., 2022), have incorporated turbulence modeling using the RNG k–ε and k–ω SST models, respectively. In the present work, the turbulence model was intentionally deactivated. This decision is supported by findings from earlier mesh-based CFD and SPH simulations, where omitting turbulence models has shown negligible impact on key hydrodynamic responses, including wave elevation, body motion, and mooring line tensions (Dominguez et al., 2019; Chen et al., 2023; He et al., 2023). Although turning off turbulence modeling implies that near-wall effects and boundary layers are not fully resolved, the primary viscous forces, particularly those induced by shear in the wave field, remain reasonably captured within the laminar assumption, especially given the moderate Reynolds numbers characterizing the present setup. Moreover, for configurations involving gap resonance or tightly spaced floating structures, the use of laminar flow has proven effective in reproducing fluid–structure interactions with satisfactory accuracy. Notably, (Feng et al., 2017; Gao et al., 2021, 2022) reported that activating turbulence models yielded no significant improvements in simulation results. Based on these considerations, and to limit computational overhead, laminar flow modeling was adopted herein. Nonetheless, we acknowledge that turbulence may play a more substantial role in highly nonlinear or extreme sea states, and this remains an open topic for future investigation.

In the two immiscible phases, the local fluid density and dynamic viscosity vary between ρ_w, μ_w and ρ_a, μ_a, where the suscripts w and a denote water and air, respectively. Both ρ and μ are described in the Volume of Fluid (VoF) surface capturing method by the volume fraction parameter α, see Hirt and Nichols (1981). In this method, the parameter α is expressed as a Heaviside function, taking the value 1 in the reference fluid and 0 in the other, with the air-water interface cells usually approximated by an isosurface with $\alpha = 0.5$. The fluid properties in each cell is then weighted by:

$$\rho = \alpha\rho_w + (1-\alpha)\rho_a \tag{2a}$$

$$\mu = \alpha\mu_w + (1-\alpha)\mu_a \tag{2b}$$

and Eq. (1a) can also be written as the transport of the liquid volume fraction:

$$\frac{\partial\alpha}{\partial t} + \nabla\cdot(\alpha\mathbf{u}) = 0 \tag{3}$$

This system of equations is the basis of the VoF schemes used to track the fluid interface. They may be divided into two categories: geometric methods and algebraic methods. While geometric methods explicitly reconstruct the interface from the α fields, algebraic methods are much simpler and are not restricted to structured meshes. In this work, we investigate the influence of both methods by comparing the isoAdvector algorithm developed by Roenby et al. (2016) as a geometric model, and a semi-implicit variant of the interface compression method known as Multidimensional Universal Limiter for Explicit Solution (MULES) as an algebraic model, see e.g. Larsen et al. (2018).

2.2. Rigid body dynamics

Applying the conservation of linear and angular momentum laws of Newton-Euler dynamics, the equations of motion for the floating body in six degrees of freedom with respect to the floating body's centre of mass (Gatin et al., 2017):

$$\frac{\partial\mathbf{v}}{\partial t} = \frac{\mathbf{F}}{m} \tag{4a}$$

$$\frac{\partial\boldsymbol{\omega}}{\partial t} = \mathbf{I}^{-1}\cdot[\boldsymbol{\tau} - \boldsymbol{\omega}\times(\mathbf{I}\,\boldsymbol{\omega})] \tag{4b}$$

Here, $\mathbf{v}$ and $\boldsymbol{\omega}$ are the linear and angular velocities , respectively, with respect to the floating body's centre of mass, while m represents the body mass, and $\mathbf{I}$ is the time-invariant tensor of inertia with respect to the floating body's centre of mass. $\mathbf{F}$ and $\boldsymbol{\tau}$ denote the total external forces and moments acting on the body. At each time step, the resulting force and moment on the body are computed by integrating the normal pressure and tangential shear stress over the body's surface (S):

$$\mathbf{F} = \iint_S (p\mathbf{I} + \boldsymbol{\tau})\mathrm{d}\mathbf{S} + \mathbf{F}_{moor} + m_f\mathbf{g} \tag{5}$$

$$\boldsymbol{\tau} = \iint_S \mathbf{r}_{cs}\times(p\mathbf{I} + \boldsymbol{\tau})\mathrm{d}\mathbf{S} + \mathbf{r}_{cm}\times\mathbf{F}_{moor} + \mathbf{r}_{cg}\times m_f\mathbf{g} \tag{6}$$

where $\boldsymbol{\tau}$ represents the viscous stress, $\mathbf{F}_{moor}$ the mooring reaction force and $\mathbf{r}_{cs}$, $\mathbf{r}_{cm}$, and $\mathbf{r}_{cg}$ are the lever arms of the hydrodynamic force, mooring force, and gravity force, respectively. When the center of mass and the center of rotation are identical, $\mathbf{r}_{cg} = 0$. Based on the force and moment reactions derived from Eqs. (5) and (6), respectively, the velocity, position and orientation of the floating body are updated by applying a second-order implicit solver based on the Newmark integration scheme in the native sixDoFRigidBodyMotion solver in OpenFOAM, see Chen et al. (2019).

2.2.1. FloatStepper algorithm

FloatStepper is a new non-iterative algorithm developed by Roenby et al. (2024) for removing the numerical added mass instability problem in the coupling of the motion of a rigid body and an incompressible fluid in CFD simulations. The added mass instability problem arises when a low structural mass interacts with a surrounding heavy fluid, see e.g. Devolder et al. (2017). The core idea behind FloatStepper lies in calculating the added mass matrix, A, for the rigid body and exploiting it to update the body's motion. More details about the FloatStepper algorithm can be found in Roenby et al. (2024) and the source code on its github repository. This approach is based on decomposing the net force (and torque) acting on the body into two components, the added mass term and all other contributions. If we consider a rigid body exposed to a net force $\mathbf{F}$, this can be written as:

$$\mathbf{F} = \mathbf{F}_{other} - A\dot{\mathbf{v}}_{\mathbf{b}} \tag{7}$$

with A as the 6×6 added mass matrix and $\dot{\mathbf{v}}_{\mathbf{b}} = [\dot{v} \quad \dot{\omega}]^T$ the instantaneous acceleration of the body. $\mathbf{F}_{other}$ denotes all other forces on the body (gravity, buoyancy, mooring, etc). Expressing Eqs. (4a) and (4b) in a more compact form:

$$\begin{bmatrix} mI_3 & 0_3 \\ 0_3 & \mathbf{I} \end{bmatrix}\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} - \boldsymbol{\omega}\times(\mathbf{I}\,\boldsymbol{\omega}) \end{bmatrix} \tag{8}$$

where the force and torque contain terms that are proportional to the acceleration:

$$\begin{bmatrix} \mathbf{F} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{other} \\ \boldsymbol{\tau}_{other} \end{bmatrix} - A\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} \tag{9}$$

By substituting the decomposed force from Eq. (7) into the body's equations of motion (8) and isolating the acceleration, we obtain:

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = (M + A)^{-1}\begin{bmatrix} \mathbf{F}_{other} \\ \boldsymbol{\tau}_{other} + \boldsymbol{\omega}\times(\mathbf{I}\,\boldsymbol{\omega}) \end{bmatrix} \tag{10}$$

Then FloatStepper determines F_{other} by performing a test time step with zero body acceleration before the actual time step. The component of the added mass matrix, A, are then numerically computed as the resulting force (and torque) per unit of the body's linear and angular acceleration. Subsequently, the position of the body, mesh and fluid state are reversed to their initial value and $\dot{\mathbf{v}}_{\mathbf{b}}$ is calculated using Eq. (10). A comprehensive description of the FloatStepper algorithm is available in Roenby et al. (2024) and its GitHub repository.[1]

2.3. Dynamic mesh methods

Two dynamic mesh motion techniques are used in this study to capture the motion of the floating body. The first is the mesh morphing method, which is the most common method for floating body simulations without topological changes (Jasak and Tuković, 2010). It involves dynamically adjusting the position of mesh nodes to accommodate the motion of the floating structure. It is also known as mesh deformation, and ensures that the mesh conforms to the changing geometry while preserving the integrity and shape of the floating body itself. The initial mesh is generated around the floating body, ensuring an appropiate distribution of nodes to capture the geometry accurately. As the floating body moves, the positions of the near nodes are updated. This adjustment is based on the motion of the floating body and is calculated to maintain the mesh quality and prevent excessive skewness. By only updating the nodal positions rather than remeshing the entire domain, the computational cost is significantly reduced compared to methods like overset meshes. This makes the morphing mesh technique partcularly efficient for simulations where the floating body undergoes small displacements. For large deformations, there is a risk of mesh skewness, which can lead to solution divergence. To mitigate this, the technique includes checks and adjustments to maintain mesh quality (in scenarios where the floating body is subject to wave-induced motion, the mesh morphing technique can adapt the mesh to follow the body's movement, ensuring accurate simulation of the fluid-structure interaction). Additionally, Jacobsen et al. (2012) found that it is crucial to maintain cell aspect ratios around 1.0 in the vicinity of the free surface to accurately simulate breaking waves. In the context of the morphing mesh technique applied to moving bodies, the computational domain is typically divided into three regions: inner, middle, and outer. The inner region, closest to the floating body, has a high mesh density to capture the detailed motion of the body. The middle region serves as a transitional zone, smoothing the deformation of the mesh. Here, the cells are moved based on Spherical Linear Interpolation (SLERP) to ensure smooth transitions and prevent abrupt changes in mesh quality. The outer region remains largely unaffected by the floating body's motion and provides a stable boundary with a coarser mesh to reduce computational cost.

The second is the overset method which is a powerful method for simulating floating bodies and other complex moving geometries, making it specially suitable for scenarios involving large amplitude motions (Chen et al., 2019). In this method, a background mesh (fixed mesh) is first generated and then a separate mesh is generated around the body. This second mesh is attached to the body and moves according to Newton's Second Law. The two grids are not deformable, so that why maintain their initial quality over time. The cells are classified into *hole*, *interpolated* and *calculated* types. The hole cells are geometrically inside the moving floating body patches and are obscured cells (the solution is not computed). In the *calculated* cells the solution is computed by solving the flow governing equations. In the *interpolated* cells the information travels from the background mesh to the floating body fitted mesh through the overset patches and vice versa. The cell size close to overset patches (background and overset meshes) must be similar to minimize interpolation errors. In addition, and based on experience, there is a minimum number of cells between the boundary of the floating body and the outer boundary of the fitted mesh to ensure the quality of the interpolation.

2.4. Mooring dynamics model

The lumped-mass model discretizes the cable into N equal length segments connected by N + 1 nodes with masses and massless springs and dampers, see Fig. 1. The node index starts with 0 at the bottom anchor node and ends at the top fairlead node with index N + 1. Each node i is located by a position vector $\mathbf{r_i} = [x_i\, y_i\, z_i]^T$ in an inertial reference frame which defines the z-axis, pointing positive upwards, towards the still water level. Half of the total mass of the two neighboring segments is allocated to each node. The cable segments are assumed to be rigid and massless, and subject to hydrodynamic forces.

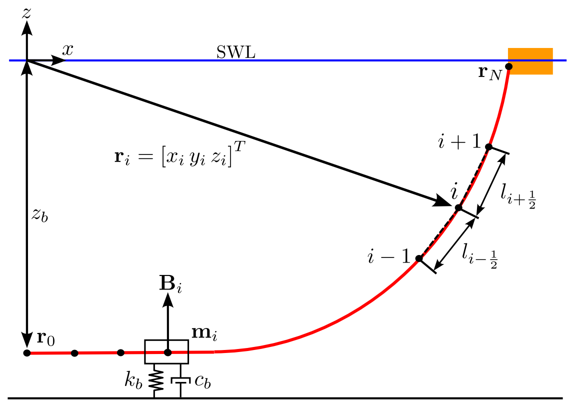


Fig. 1. Lumped-mass model discretization.

Mooring lines are considered as slender structures, and as a result, bending stresses are typically orders of magnitude smaller than axial stresses, which is why they are usually omitted. The weight and buoyancy forces, hydrodynamic loads and forces due to interaction with seabed are lumped at nodes together with masses in the dynamic mooring model MoorDyn (Hall et al., 2020(@). Hydrodynamic forces are calculated based on Morison's equation. The tension in the segments due to axial stiffness is modeled by specifying a linear stiffness. An internal damping force is also specified for each segment to dampen non-physical resonances caused by the lumped-mass discretization. The seabed reaction is modeled by a linear spring-damper approach. For further details, see Hall et al. (2020(@); Hall (2020); Hall et al. (2021).

2.4.1. Internal forces

In the dynamic mooring model MoorDyn, the internal forces considered include axial stiffness, axial damping, and weight , following detailed in Hall et al. (2020(@), see Fig. 2 for a free-body diagram of a node i. The net buoyancy, considered with weight for simplicity, of each segment $i + \frac{1}{2}$ is:

$$W_{i+\frac{1}{2}} = \frac{\pi}{4} d^2 l \left(\rho_w - \rho\right) g \tag{11}$$

where d is the volume-equivalent diameter of the mooring line, l is the unstretched length of each cable segment, ρ_w is the water density, ρ is the mooring line density and g is the acceleration due to gravity. The net buoyancy at segment $i + \frac{1}{2}$ is divided among the two connecting nodes i and $i + 1$:

$$\mathbf{W}_i = \frac{1}{2}\left(\mathbf{W}_{i+\frac{1}{2}} + \mathbf{W}_{i-\frac{1}{2}}\right)\hat{\mathbf{e}}_z \tag{12}$$

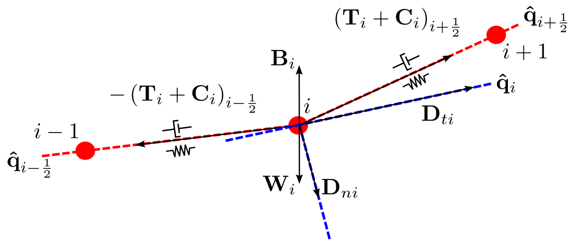

$(\mathbf{T}_i + \mathbf{C}_i)_{i+\frac{1}{2}}$
$\hat{\mathbf{q}}_{i+\frac{1}{2}}$
$i+1$
$\mathbf{B}_i$
$\hat{\mathbf{q}}_i$
$-(\mathbf{T}_i + \mathbf{C}_i)_{i-\frac{1}{2}}$
i
$\mathbf{D}_{ti}$
$i-1$
$\hat{\mathbf{q}}_{i-\frac{1}{2}}$
$\mathbf{W}_i$
$\mathbf{D}_{ni}$

Fig. 2. Free-body diagram of the node i.

[1] https://https://github.com/FloatStepper/FloatStepper.git

where $\hat{\mathbf{e}}_z$ is a unit vector in the positive direction. The tension in a cable segment $i + \frac{1}{2}$ due to axial stiffness is:

$$\mathbf{T}_{i+\frac{1}{2}} = \begin{cases} E\frac{\pi}{4}d^2\varepsilon_{i+\frac{1}{2}}\hat{\mathbf{q}}_{i+\frac{1}{2}} & \text{if} \quad \left|\mathbf{r}_{i+1} - \mathbf{r}_i\right| > l \\ 0 & \text{if} \quad \left|\mathbf{r}_{i+1} - \mathbf{r}_i\right| \leqslant l \end{cases} \tag{13}$$

where $\varepsilon_{i+\frac{1}{2}}$ is the strain, $\hat{\mathbf{q}}_{i+\frac{1}{2}}$ is a tangent unit vector directed from node $\mathbf{r}_i$ to $\mathbf{r}_{i+1}$:

$$\hat{\mathbf{q}}_{i+\frac{1}{2}} = \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\left|\mathbf{r}_{i+1} - \mathbf{r}_i\right|} \tag{14}$$

Eq. (13) assumes that a tension force is exerted only when there is a positive tension in the line, meaning $\left|\mathbf{r}_{i+1} - \mathbf{r}_i\right| > l$. Otherwise, the tension force is set to zero, as no compression force is modeled. The resultant tension force at node i considering both the adjacent segments $i + \frac{1}{2}$ and $i - \frac{1}{2}$ is:

$$\mathbf{T}_i = \mathbf{T}_{i+\frac{1}{2}} - \mathbf{T}_{i-\frac{1}{2}} \tag{15}$$

The internal damping force in the segment, which plays a significant role in ensuring numerical stability is:

$$\mathbf{C}_{i+\frac{1}{2}} = C_{int}\frac{\pi}{4}d^2\dot{\varepsilon}_{i+\frac{1}{2}}\hat{\mathbf{q}}_{i+\frac{1}{2}} \tag{16}$$

where C_{int} represents the numerical internal damping coefficient, and $\dot{\varepsilon}_{i+\frac{1}{2}}$ denotes the strain rate:

$$\dot{\varepsilon}_{i+\frac{1}{2}} = \frac{\partial\varepsilon_{i+\frac{1}{2}}}{\partial t} \tag{17}$$

Similar to the tension T_i, the internal damping force at node i is also expressed as:

$$\mathbf{C}_i = \mathbf{C}_{i+\frac{1}{2}} - \mathbf{C}_{i-\frac{1}{2}} \tag{18}$$

2.4.2. *External forces*

The cable hydrodynamic forces, including both drag and added mass, are computed by using the Morison equation applied at node i, treating the cable as a slender structure. The drag force applied to node i is composed of a transverse component D_{ni} and a tangential component D_{ti}:

$$\mathbf{D}_i = \mathbf{D}_{ni} + \mathbf{D}_{ti} \tag{19}$$

where each component is calculated as:

$$\mathbf{D}_{ni} = \frac{1}{2}\rho_w C_{dn} dl \left|\left(\dot{\mathbf{r}}_i \cdot \hat{\mathbf{q}}_i\right)\hat{\mathbf{q}}_i - \dot{\mathbf{r}}_i\right| \left[\left(\dot{\mathbf{r}}_i \cdot \hat{\mathbf{q}}_i\right)\hat{\mathbf{q}}_i - \dot{\mathbf{r}}_i\right] \tag{20}$$

$$\mathbf{D}_{ti} = \frac{1}{2}\rho_w C_{dt} dl \left|\left(-\dot{\mathbf{r}}_i \cdot \hat{\mathbf{q}}_i\right)\hat{\mathbf{q}}_i\right| \left(\dot{\mathbf{r}}_i \cdot \hat{\mathbf{q}}_i\right)\hat{\mathbf{q}}_i \tag{21}$$

in which $\dot{\mathbf{r}}_i$ represents the flow velocity at the cable node i and $\hat{\mathbf{q}}_i$ is the unit tangent vector at each node i and is approximated by the line connecting the two adjacent nodes:

$$\hat{\mathbf{q}}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{\left|\mathbf{r}_{i+1} - \mathbf{r}_{i-1}\right|} \tag{22}$$

The added mass force at node i, $\mathbf{A}_i$, is also composed of a transverse $\mathbf{A}_{ni}$ and tangential component $\mathbf{A}_{ti}$:

$$\mathbf{A}_{ni} = \mathbf{a}_{ni}\,\ddot{\mathbf{r}}_i = \rho_w\frac{\pi}{4}d^2 l C_{an}\left[\left(\ddot{\mathbf{r}}_i \cdot \hat{\mathbf{q}}_i\right)\hat{\mathbf{q}}_i - \ddot{\mathbf{r}}_i\right] \tag{23}$$

$$\mathbf{A}_{ti} = \mathbf{a}_{ti}\,\ddot{\mathbf{r}}_i = \rho_w\frac{\pi}{4}d^2 l C_{at}\left(-\ddot{\mathbf{r}}_i \cdot \hat{\mathbf{q}}_i\right)\hat{\mathbf{q}}_i \tag{24}$$

where C_{an} and C_{at} are the transverse and tangential added mass coefficients and $\ddot{\mathbf{r}}_i$ the flow acceleration at node i. $\mathbf{a}_{ni}$ and $\mathbf{a}_{ti}$ are the corresponding transverse and tangential added mass respectively. Factoring out $\ddot{\mathbf{r}}_i$ in Eqs. (23) and (24) gives the combined 3×3 added mass matrix $\mathbf{a}_i = \mathbf{a}_{ni} + \mathbf{a}_{ti}$:

$$\mathbf{a}_i = \rho_w\frac{\pi}{4}d^2 l\left[C_{an}\left(\mathbf{I} - \hat{\mathbf{q}}_i\hat{\mathbf{q}}_i^T\right) + C_{at}\left(\hat{\mathbf{q}}_i\hat{\mathbf{q}}_i^T\right)\right] \tag{25}$$

where $\mathbf{I}$ is the identity matrix.

The vertical reaction forces due to the interaction of the mooring line with the seabed at each node is calculated using a linear spring-damper model. This interaction is activated when a mooring line node contacts the seabed (i.e. $z_i \leqslant z_b$):

$$\mathbf{B}_i = \begin{cases} dl\left[\left(z_b - z_i\right)k_b - \dot{\mathbf{z}}_i c_b\right]\hat{\mathbf{e}}_z & \text{if} \quad z_i \leqslant z_b \\ 0 & \text{if} \quad z_i > z_b \end{cases} \tag{26}$$

where k_b is the stiffness coefficient per unit area of the seabed, $\mathbf{c_b}$ is the viscous damping coefficient per unit area, z_b is the seabed vertical coordinate, z_i and $\dot{z}_i$ are each node vertical coordinate and vertical velocity, respectively, and $\hat{\mathbf{z}}_z$ is the unit vector in the positive z direction.

2.4.3. *Integration of governing equations*

The equation of motion at each node i is obtained as:

$$\left(\mathbf{m}_i + \mathbf{a}_i\right) = \mathbf{T}_i + \mathbf{C}_i + \mathbf{W}_i + \mathbf{B}_i + \mathbf{D}_i \tag{27}$$

where $\mathbf{m}_i$ is the 3×3 lumped-mass matrix at each node i. It is defined by assigning half of the total mass of the two neighboring segments and expressed as:

$$\mathbf{m}_i = \frac{\pi}{4}d^2 l\rho\mathbf{I} \tag{28}$$

Eq. (27) represents a second-order system of ordinary differential equations (ODEs) that is reduced to a first-order system ones by introducing auxiliary variables, thus transforming the problem into a system of first-order ODEs. MoorDyn offers several time integration schemes to solve the equations of motion, including Euler, Runge–Kutta or Adams–Bashforth methods. In this study we used the Runge–Kutta 4th order (RK4) scheme with the same time step in mooring integration.

3. Case set-up

3.1. *Numerical model and computational settings*

The test case used in this study is a solid floating cuboid (referred as floating box here after) with geometric features listed in Table 1. The results are validated against experimental data generated during the experimental campaigns for the European MaRINET2, EsfloWC project by the Coastal Engineering Research Group of Ghent University (Wu et al., 2019). The floating box is 0.2 m in length, 0.2 m in width and 0.132 m in height, with an initial draft of 0.0786 m. The center of gravity is located at the geometrical center of the box, 0.066 m above the bottom of the box along the z-axis. The moments of inertia with respect to the centre of gravity are $\mathrm{I}_{xx}\mathrm{I}_{yy} = 0.015\ \mathrm{kg\,m}^2$ and $\mathrm{I}_{zz} = 0.021\ \mathrm{kg\,m}^2$. The floating box is connected to the bottom through a four-symmetric catenary mooring system, as shown in Fig. 3. Mooring lines 1 and 2 are located at the front, facing the incident waves, while lines 3 and 4 are positioned at the rear of the floating box. Table 2 lists the parameters of the mooring lines, and Table 3 provides the coordinates for the anchor and fairlead connections to the bottom and the box, respectively (see Fig. 4). In the case of the fairleads, they are located at a distance of 0.061 m from the center of gravity of the box, that is, 0.005 m above the bottom of the box.

Two meshing strategies were employed in this study: an overset mesh technique and a morphing mesh technique. Both approaches were configured to ensure accurate wave resolution and stable fluid-structure interaction, while keeping computational cost manageable. In both cases, the mesh was refined near the free surface and the floating structure

Table 1
Parameters of the floating box used in the simulations.

Parameter	Value
Box length	0.2 m
Box width	0.2 m
Box height	0.132 m
Box center of gravity	(0, 0, −0.0126) m
Box initial draft	0.0786 m
Box mass	3.148 kg
Box moment of inertia	(0.015, 0.015, 0.021) kg m^2

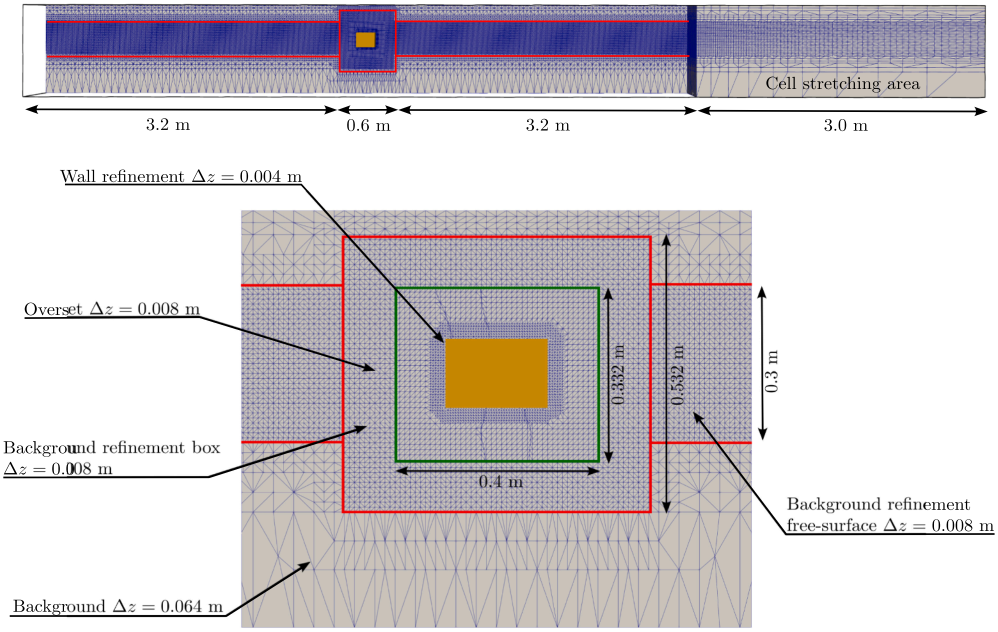

Cell stretching area
3.2 m
0.6 m
3.2 m
3.0 m
Wall refinement $\Delta z = 0.004$ m
Overset $\Delta z = 0.008$ m
Background refinement box
$\Delta z = 0.008$ m
Background $\Delta z = 0.064$ m
0.332 m
0.532 m
0.3 m
0.4 m
Background refinement
free-surface $\Delta z = 0.008$ m

Fig. 3. Description of the mesh and refinement areas.

Table 2
Parameters of the mooring lines used in the simulations.

Parameter	Value
Diameter (m)	3.656×10^{-3}
Mass in air per unit length (kg/m)	0.0607
Unstretched length (m)	1.455
Axial stiffness (N)	29
Internal damping (−)	−1.0
Transverse drag coefficient (−)	1.6
Tangential drag coefficient (−)	0.05
Transverse added mass coefficient (−)	1.0
Tangential added mass coefficient (−)	0.0
Discretized segments line (−)	20, 40, 80

Table 3
Coordinates of the mooring line anchor and fairlead connections used in the simulations.

Point	Coordinate
Fairlead 1 (F1)	(−0.1, 0.1, −0.0736) m
Fairlead 2 (F2)	(−0.1, −0.1, −0.0736) m
Fairlead 3 (F3)	(0.1, 0.1, −0.0736) m
Fairlead 4 (F4)	(0.1, −0.1, −0.0736) m
Anchor 1 (A1)	(−1.385, 0.423, −0.5) m
Anchor 2 (A2)	(−1.385, −0.423, −0.5) m
Anchor 3 (A3)	(1.385, 0.423, −0.5) m
Anchor 4 (A4)	(1.385, −0.423, −0.5) m

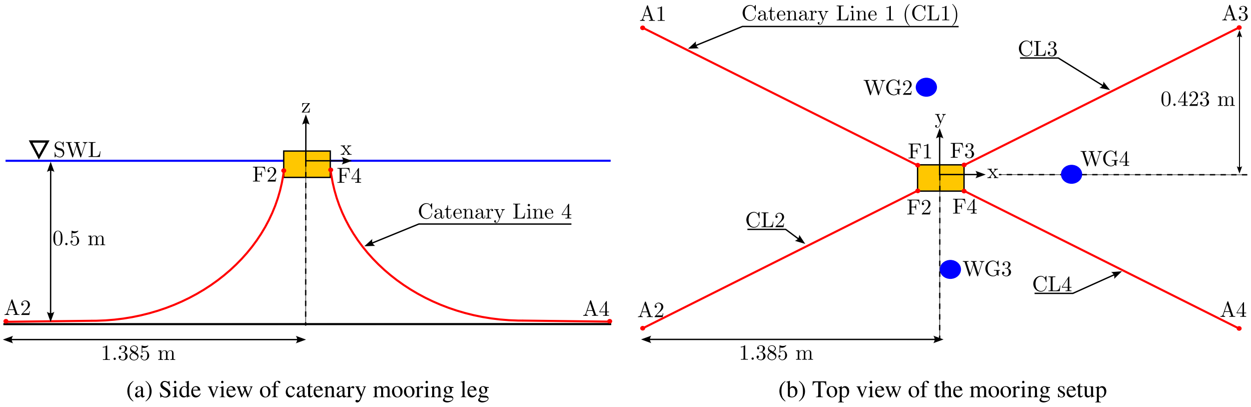
Table 4
Numerical schemes.

Numerical Scheme	OpenFOAM
Time $\left[\partial(\phi)/\partial t\right]$	Euler
Gradient $[\nabla \phi]$	Gauss linear
Divergence $[\nabla \cdot (\rho\phi\mathbf{U})]$	Gauss limitedLinearV 1;
Divergence $[\nabla \cdot (\phi\alpha)]$	Gauss vanLeer;
Divergence $\left[\nabla \cdot \left(\phi_{rb}\alpha\right)\right]$	Gauss linear;
Laplacian $\left[\nabla^2\right]$	Gauss linear corrected
Overset interpolation	inverseDistance

with a uniform cell size of 0.008 m. This refinement guarantees a sufficient resolution to capture the wave and body dynamics across all test cases. Specifically, for the shorter wave condition (H12T18), this corresponds to 15 cells per wave height (CPH) and 357 cells per wavelength (CPL), which aligns with the ITTC recommendations ITTC (2014). In both the overset and morphing mesh configurations, stretching regions were applied in the background mesh to control cell growth. Vertical stretching was defined with an aspect ratio of 1:4, while longitudinal stretching used a 1:20 ratio near the domain boundaries.

This graded resolution helps dissipate wave energy away from the region of interest, reduces the overall cell count, and minimizes artificial reflections, all while preserving mesh quality around the floating box, see Fig. 4. In the overset strategy, this resolution was applied to the body-fitted overset mesh and the free-surface refinement zones of the background mesh. For the morphing mesh, the same resolution was extended throughout a uniform band around the floating body, which increased the total number of cells due to the lack of topological separation between body and background. As a result, the final mesh contains approximately 3.4 million cells in the overset configuration and 3.7 million cells in the morphing mesh case. This mesh configuration was validated through a grid convergence analysis (see Appendix A), which demonstrated that the intermediate mesh resolution provided a good balance between accuracy and efficiency, with converged results for both surface elevation and body motion. The overset mesh ensures high resolution around the body while maintaining non-deforming, high-quality mesh blocks. In contrast, the morphing mesh solution uses a single continuous mesh and is more sensitive to large deformations. The morphing domain was therefore extended in height to reduce skewness during motion. Finally, a grid convergence analysis was performed using three mesh resolutions (Appendix A). The intermediate mesh (mesh B) was selected for all simulations, as it demonstrated converged results in surface elevation and floating body motions with significantly lower computational cost compared to the finest mesh.

The real flow computational time is 18 s, which corresponds to 10 wave periods. For time marching, the Euler scheme was used while the other discretization schemes are detailed in Table 4. The time step was dynamically adjusted during the simulation, with maximum Courant



(a) Side view of catenary mooring leg (b) Top view of the mooring setup

Fig. 4. Geometric description of the experimental setup and the mooring cable configuration.

Table 5
Description of the boundary conditions for volume fraction, pressure and velocity.

Boundary	Alpha (α)	Pressure (p)	Velocity (**U**)
Inlet	waveAlpha	fixedFluxPressure	waveVelocity
Oulet	zeroGradient	fixedFluxPressure	waveVelocity
Atmosphere	inletOutlet	totalPressure	pressureInletOutletVelocity
stationaryWalls	zeroGradient	fixedFluxPressure	noSlip
Floating Body	zeroGradient	fixedFluxPressure	movingWallVelocity
Overset patch	overset	overset	overset

number limit of 0.9. It is recommended to optimize for a lower Courant number when employing overset simulation, even though this choice may result in a trade-off with increased computational time. In practical applications, favorable outcomes have been demonstrated with a Courant number of 1, as evidence by Pinguet (2021); Jeon et al. (2023). The influence of the maximum CFL is analyzed in Section 4.5. For each time step, we used 2 outer correctors (PIMPLE iterations) and two pressure correctors per PISO loop. To prevent numerical instabilities, we applied an 0.8 under-relaxation factor to the floating body acceleration without using any acceleration damping. In our presented simulations, no turbulence models is activated.

The computational domain comprises six boundaries, as outlined in Table 5. The stationaryWalls within the domain include the bottom, front and back boundaries. The key distinction from a non-overset boundaries case lies in the use of a specific boundary condition, designated as *overset*. For the inlet and outlet boundaries of the wave flume, the active wavemaker IHFoam was applied (Higuera et al., 2013) to generate regular waves and prevent reflection at the outlet. For the top boundary, an open atmosphere condition is used, which switches between a Neumann condition when the fluid is flowing out of the domain, and Dirichlet boundary condition when the fluid is flowing into the domain. For the stationaryWalls, no-slip boundary condition is applied, see Table 5. The moving wall boundary condition is applied to the surface of the floating body.

To optimize computational efficiency, a convergence analysis of mooring line discretization was conducted to determine the optimal number of segments and the appropriate timestep for MoorDyn, see Appendix B. The mooring line discretization introduces an artificial resonance that needs to be damped out by adjusting the damping ratio towards its critical value. In this study, the mooring line discretization was performed using 20, 40 and 80 segments, with a damping ratio of −1.0. A timestep of 10^{-4} s for the three discretizations was used. The latest version of Moordyn (version 2) was employed for all discretization studies. Additionally, a comparative study between both versions of Moordyn was conducted using the same segment discretization (N = 40). The analysis of the numerical results for both studies is presented in Section 4.

The computational domain is decomposed into 64 sub-domains for both overset and morphing mesh simulations. The simulations were run on the HPC Centro de Supercomputación de Galicia (CESGA) with the Scientific Linux operating system and Intel Xeon Ice Lake processors (8352Y@2.2 GHz).

3.2. Wave conditions and resolution parameters

To validate the moored motions and tensile loads of the floating box in response to waves, two incident wave 2nd and 5th order profiles obtained from experimental data in Wu et al. (2019) were used. The specific wave parameters considered in this study are summarized in Table 6. Both waves have a wavelength and period of 3.57 m and 1.8 s, respectively, with wave heights of 0.12 m (H12T18) and 0.15 m (H15T18). To compare the surface elevation history with experimental results, three wave probes located at the positions listed in Table 7 were used for both cases H12T18 and H15T18. The mesh resolution, based on the specified discretization criteria, is 357 cells per wavelength (CPL) and 15 cells per wave height (CPH) in both cases. The suggested resolution in the ITTC guidelines ITTC (2014) is CPH = 20 and CPL = 40, and thus, the CPL values align with the recommendations. To avoid large cell aspect ratios that could affect simulation results, they were maintained around 1.0 in the free-surface refinement area. This is why the CPL was not reduced to 40 as recommended by the ITTC guidelines. Regarding the CPH, a dedicated study was conducted to verify its impact and optimize the computational efficiency. The present investigation includes an analysis with different values of CPH (10, 15 and 20). The results and comparisons are presented in Fig. 13 of Appendix A.

3.3. OpenFOAM solvers and computational strategy

Table 8 presents an overview of the solvers evaluated in this study, detailing their interface capturing schemes, dynamic mesh strategies,

Table 6
Description of regular wave for simulations Wu et al. (2019).

Label	Wave height (m)	Wave period (s)	Wave length (m)	Steepness (−)	KC (−)	*Re* (−)	Wave model
H12T18	0.12	1.8	3.57	0.034	1.885	0.418×10^5	Stokes 2nd
H15T18	0.15	1.8	3.57	0.042	2.356	0.524×10^5	Stokes 5th

Table 7
Positions of the wave gauges in the numerical wave tank and according to experimental setup Wu et al. (2019).

Label	Coordinates
WG2	(−0.05, 0.26, 0) m
WG3	(0.07, −0.36, 0) m
WG4	(0.55, 0, 0) m

Table 8
Solver configurations for all tested cases, including interface capturing method, mesh motion strategy, and grid resolution.

Solver	Interface method	Dynamic mesh	Total cells (million)
`overInterDyMFoam`	VoF	Overset	3.4
`interFoam`	VoF	Morphing	3.7
`interIsoFoam`	isoAdvector	Morphing	3.7
`FloatStepper (1)`	isoAdvector	Morphing	3.7
`FloatStepper (2)`	isoAdvector	Morphing	3.7
`FloatStepper (100)`	isoAdvector	Morphing	3.7

and total mesh resolutions. The numerical experiments were conducted using three solver families: `overInterDyMFoam`, `interIsoFoam`, and `FloatStepper`, each of which implements a distinct approach to handling multiphase interfaces and mesh motion. The `overInterDyMFoam` solver is equipped with overset (or chimera) grid capabilities that allow independent mesh zones to move without deformation, making it particularly well suited for simulations involving complex geometries and large-amplitude body motions. The solver employs a classical Volume of Fluid (VoF) method for interface capturing. While the rigid motion of the grid permits larger time steps and avoids mesh quality degradation, the overset methodology introduces additional computational overhead due to the need to manage hole cells, interpolation stencils, and cell connectivity across mesh interfaces.

The `interIsoFoam` solver utilizes a morphing mesh strategy and integrates the geometric `isoAdvector` algorithm for interface reconstruction, which enhances accuracy and boundedness in VoF-based multiphase simulations. As of OpenFOAM v2306, no publicly available solver natively combines overset grid motion with the `isoAdvector` scheme. Therefore, simulations requiring geometric interface capturing were performed exclusively with morphing meshes.

The `FloatStepper` solver also uses the `isoAdvector` scheme with a morphing mesh configuration. A distinguishing feature of `FloatStepper` is its ability to explicitly separate added mass effects from the total hydrodynamic force to avoid added-mass instability. Each time step consists of three key stages: (i) a probe step where the body acceleration is temporarily set to zero, allowing for the computation of the non-inertial hydrodynamic force and torque, followed by a rewind to restore the initial state; (ii) numerical evaluation of the added mass matrix by sequentially applying unit acceleration perturbations in each active degree of freedom and solving the corresponding linearized pressure equations; and (iii) a physical update step in which the body acceleration is calculated from Newton's law using the known hydrodynamic force and added mass, followed by mesh deformation and flow field advancement. This procedure eliminates the need for outer coupling iterations and improves numerical robustness.

To reduce computational cost, the added mass matrix can be updated less frequently using the `MaddUpdateFreq` parameter. Three values were examined in this study: 1, 2, and 100. Setting `MaddUpdateFreq` = 1 implies that added mass is recomputed at every time step, leading to approximately eight pressure solves per time increment. For `MaddUpdateFreq` = 2, the added mass is updated every second time step, resulting in an average of five pressure solves per step. When `MaddUpdateFreq` = 100, only one in every 100 steps triggers a full added mass update, yielding an average of two pressure solves per time step.

Table 9
Comparison of solver configurations and computational performance for case H15T18.

Solver	Physical time (s)	Runtime (h)	Runtime per physical time (min)
`overInterDyMFoam`	16	11.2	42.0
`interFoam`	16	17.2	64.5
`interIsoFoam`	16	20.1	75.4
`FloatStepper (100)`	16	65.6	246.0
`FloatStepper (2)`	16	73.6	276.0
`FloatStepper (1)`	16	83.2	312.0

Table 9 reports the total computational cost associated with these different settings.

Although decreasing the update frequency reduces runtime-by approximately 20 % in this test case-the gain is moderate. This is attributable to the fact that the submerged volume of the floating body remains nearly constant throughout the simulation. Consequently, the added mass matrix evolves slowly and less frequent updates do not compromise accuracy. Nonetheless, this finding is case-specific. In highly dynamic scenarios, such as water-entry problems or configurations with significant immersion variation, frequent updates (i.e., `MaddUpdateFreq` = 1) would be essential to maintain numerical fidelity. In summary, while the `MaddUpdateFreq` parameter offers a means to optimize computational efficiency, its selection must be based on a sound understanding of the hydrodynamic behavior of the system. Reducing the update frequency is justified only in cases where the added mass does not vary significantly. Otherwise, aggressive underuse of this update mechanism may yield computationally efficient but physically inaccurate results.

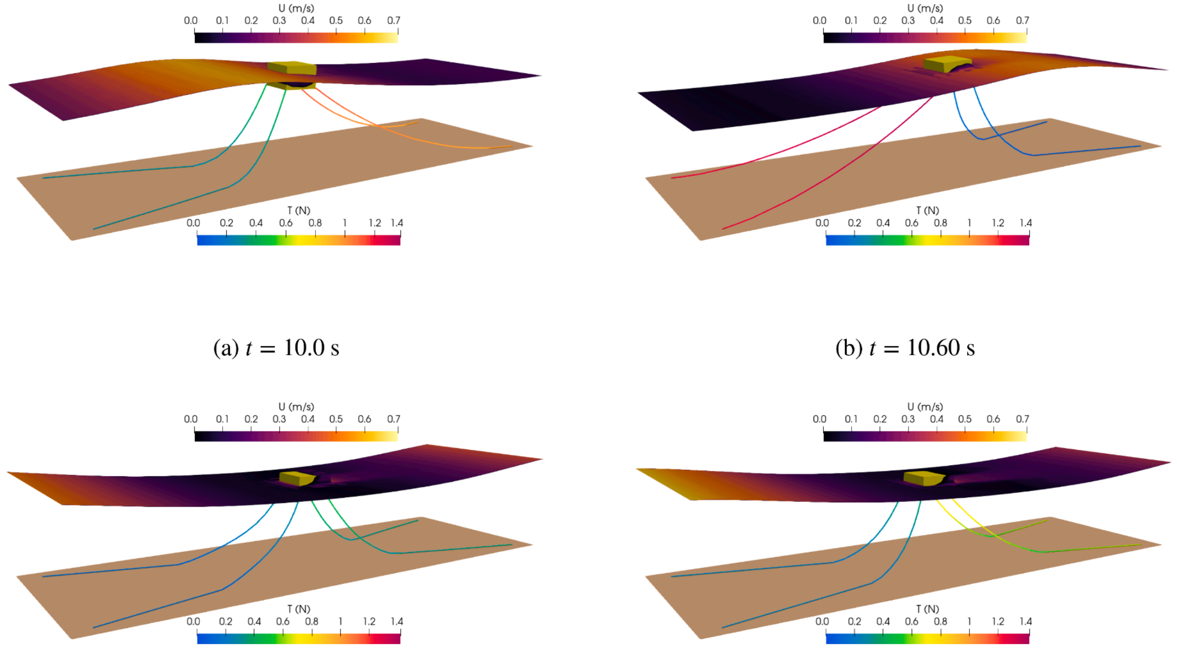
3.3.1. MoorDyn version 2 new features and coupling with FloatStepper

The version 2 of MoorDyn version 2 incorporates significant additions and changes with respect to its first version, as documented by Hall (2020). It includes all features of version 1 and introduces new capabilities such as the simulation of six-degree-of-freedom objects, non-linear tension, wave kinematics, bending stiffness, bathymetry, and seabed friction. As part of this work, new features have been integrated into the latest version of MoorDyn and coupled with the solver FloatStepper. The newly implemented features include:

- **New Time Integration Schemes:** Added options with several time schemes to enhance simulation efficiency. These schemes are divided into two main categories: explicit and implicit. The available explicit schemes include Euler, Local Time-Step Euler, Heun, Runge–Kutta, Adams–Bashforth, and Local Time-Step Adams–Bashforth. Among the implicit schemes, some examples are Backward Euler, Midpoint, Backward Euler with Anderson's acceleration, Average Constant Acceleration, and the Wilson-Theta scheme.
- **CFL (Courant–Friedrichs–Lewy) Option:** Users can now define the time step (dtM) or the Courant-Friedrich-Levy (CFL) criterion for adaptative time step, or both, with the more restrictive option taking precedence. The CFL factor is defined as the ratio between the time step and the mooring line natural period.
- **Seabed consideration in the catenary solver:** The seabed interaction in the catenary solver has been refined to activate only when one of the line ends is sufficiently close to the bottom, avoiding initialization issues caused by incorrectly considering midline points in contact with the seabed. This adjustment ensures more robust handling of scenarios, where both line ends are at similar heights.

For detailed information on these new features, please refer to the MoorDyn repository.[2] The coupling with FloatStepper solver is freely

[2] https://github.com/FloatingArrayDesign/MoorDyn.git



(a) t = 10.0 s

(b) t = 10.60 s

(c) t = 11.20 s

(d) t = 11.35 s

Fig. 5. View of instantaneous velocity field on the free surface around floating box and the four catenary mooring tension for a cycle corresponding to a wave period for case H12T18 (an animation has been provided as part of supplementary material).

accessible on its github repository.[3] This coupling employs the methodology outline by Roenby et al. (2024); Aliyar et al. (2022), wherein the motion solver provides the floater's position and velocity to MoorDyn. Then MoorDyn computes and returns the net restraining forces and moments from all fairlead tensions. Notably, in the current coupling, the added mass matrix is calculated directly within the PIMPLE loop, as was done in the coupling approach for MoorDyn version 1 detailed by Roenby et al. (2024).

The coupling between OpenFOAM and MoorDyn requires careful handling of time-stepping because the two solvers typically operate with different time-step sizes due to the distinct physical processes they simulate. OpenFOAM uses a very small time step, dictated by the Courant number (Co), to resolve fluid dynamics accurately, especially for wave propagation. In contrast, MoorDyn simulates the mooring line dynamics, where the time step (dtM) or CFL number is primarily governed by the natural frequencies of the mooring system. This allows MoorDyn to tolerate larger time steps compared to OpenFOAM. Both coupling methodologies (foamMooring and FloatStepper) address this disparity using a subcycling approach. In this approach, MoorDyn operates independently with its internal time step, often performing multiple substeps during a single OpenFOAM time step to accurately resolve the mooring dynamics. At each coupling interval (the end of OpenFOAM's time step), the following data exchange occurs:

- OpenFOAM → MoorDyn: OpenFOAM provides updated floater's motion data (position and velocity) for the floating structure.
- MoorDyn → OpenFOAM: MoorDyn calculates the mooring forces based on the motion data and sends these forces back to OpenFOAM for its use in the next time step.

Between these coupling intervals, MoorDyn independently solves the mooring system's dynamics, ensuring accurate force predictions without interfering with OpenFOAM's time-stepping. Notably, MoorDyn and OpenFOAM exhibit a two order of magnitude difference in CPU time, with MoorDyn being significantly more computationally efficient.

4. Results and discussion

As an example to illustrate the flow phenomenology, Fig. 5 shows the instantaneous velocity fields on the free surface around the floating body and the four mooring line tensions for a cycle corresponding to a wave period for case H12T18 and using the overset and VoF methods. When the crest of the wave pushes the floating box towards the onshore (positive x-direction) it produces high tension on the forward mooring lines (line 1 and 2), while the trough of the wave, on the other hand, pulls the floating body towards the offshore (negative x-direction), and produces low tension on forward lines. The tension in the mooring lines periodically varies with changes in the free surface elevation, impacting the tension in offshore lines (lines 1 and 2) and onshore lines (lines 3 and 4) during opposite wave phases. Additionally, wave overtopping (green water phenomena) is briefly observed, as a small amount of water runs up from the lateral sides of the floating structure.

As a reference, Table 9 provides for the three solvers, overInterDyMFoam, interIsoFoam, and FloatStepper the computational cost. For the latter, the computational times are also given for two different values of the parameter for added mass correction (MaddUpdateFreq), specifically 1, 2 and 100, respectively. In order to evaluate the level of agreement between the experimental results of Wu et al. (2019) and the numerical results of Chen and Hall (2022) with the present study, the Normalized Mean Squared Error (NMSE) has been computed to compare the numerical and experimental data, as summarized in Table 10.

4.1. *Surface elevation*

Fig. 6 illustrates the time histories of wave elevation registered during numerical simulations by wave gauges 2, 3, and 4 (WG2, WG3, and WG4 respectively, see Table 7), and compared with numerical results

[3] https://https://github.com/FloatStepper/FloatStepper.git

Table 10
Comparison of values between the experimental data (Wu et al., 2019), previous numerical results (Chen and Hall, 2022), and the present study.

NMSE					
Case	Parameter	Chen and Hall (2022)	overInterDyMFoam	interIsoFoam	FloatStepper
H12T18	WG2	0.0209	0.0201	0.0219	–
H12T18	WG3	0.0672	0.0550	0.0548	–
H12T18	WG4	0.0838	0.0809	0.0821	–
H12T18	Surge	0.1887	0.1110	0.1592	0.1437
H12T18	Heave	0.0991	0.0918	0.0927	0.0925
H12T18	Pitch	0.5845	0.5251	0.6857	0.7619
H15T18	WG2	0.0927	0.0455	0.0572	–
H15T18	WG3	0.0679	0.0517	0.0436	–
H15T18	WG4	0.1001	0.0849	0.1033	–
H15T18	Surge	0.1821	0.0254	0.0240	0.0268
H15T18	Heave	0.1425	0.0792	0.0890	0.0936
H15T18	Pitch	0.7950	0.5893	0.6750	0.7028

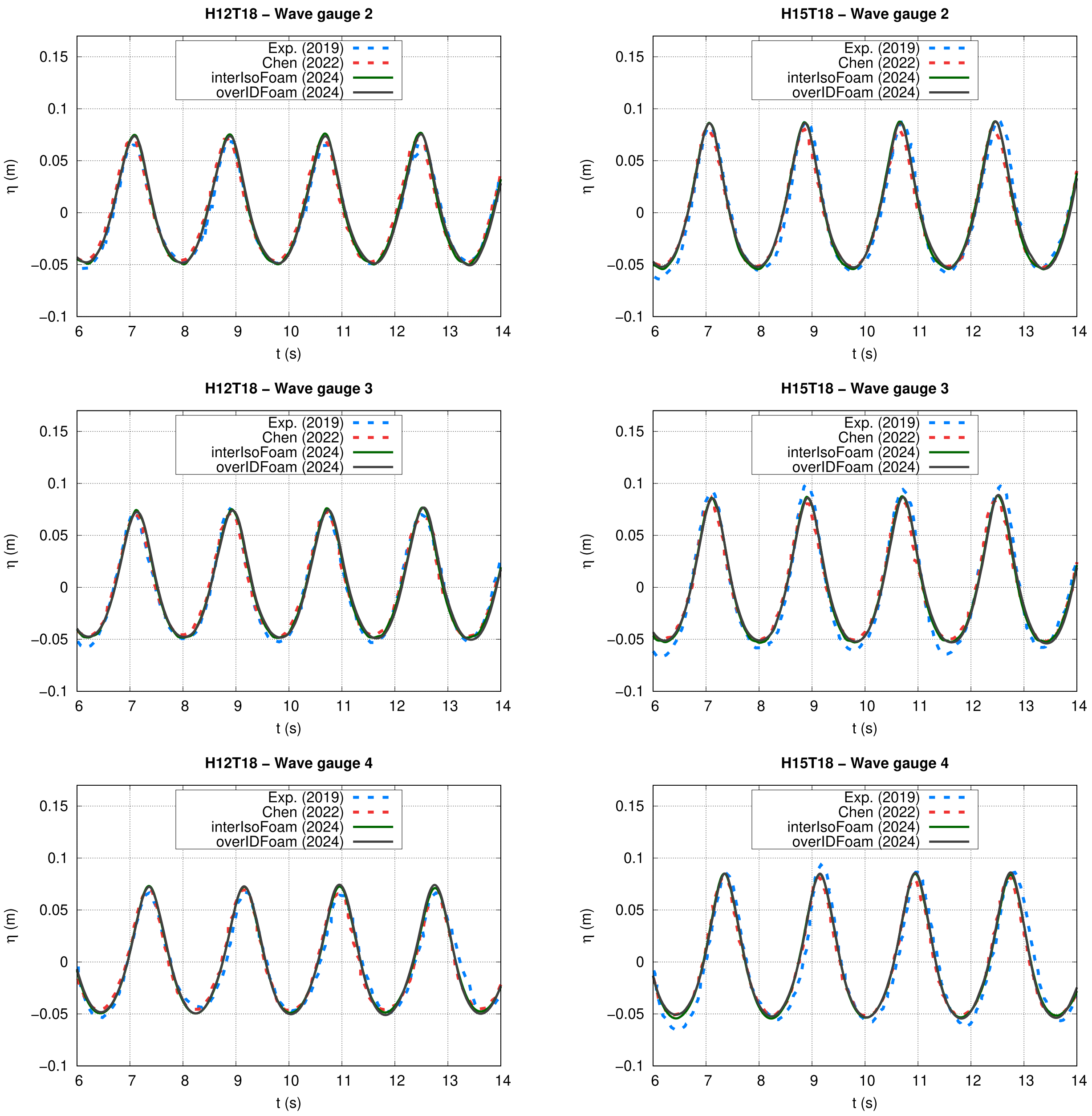


Fig. 6. Comparison of surface elevation with experimental Wu et al. (2019) and numerical Chen and Hall (2022) for cases H12T18 and H15T18.

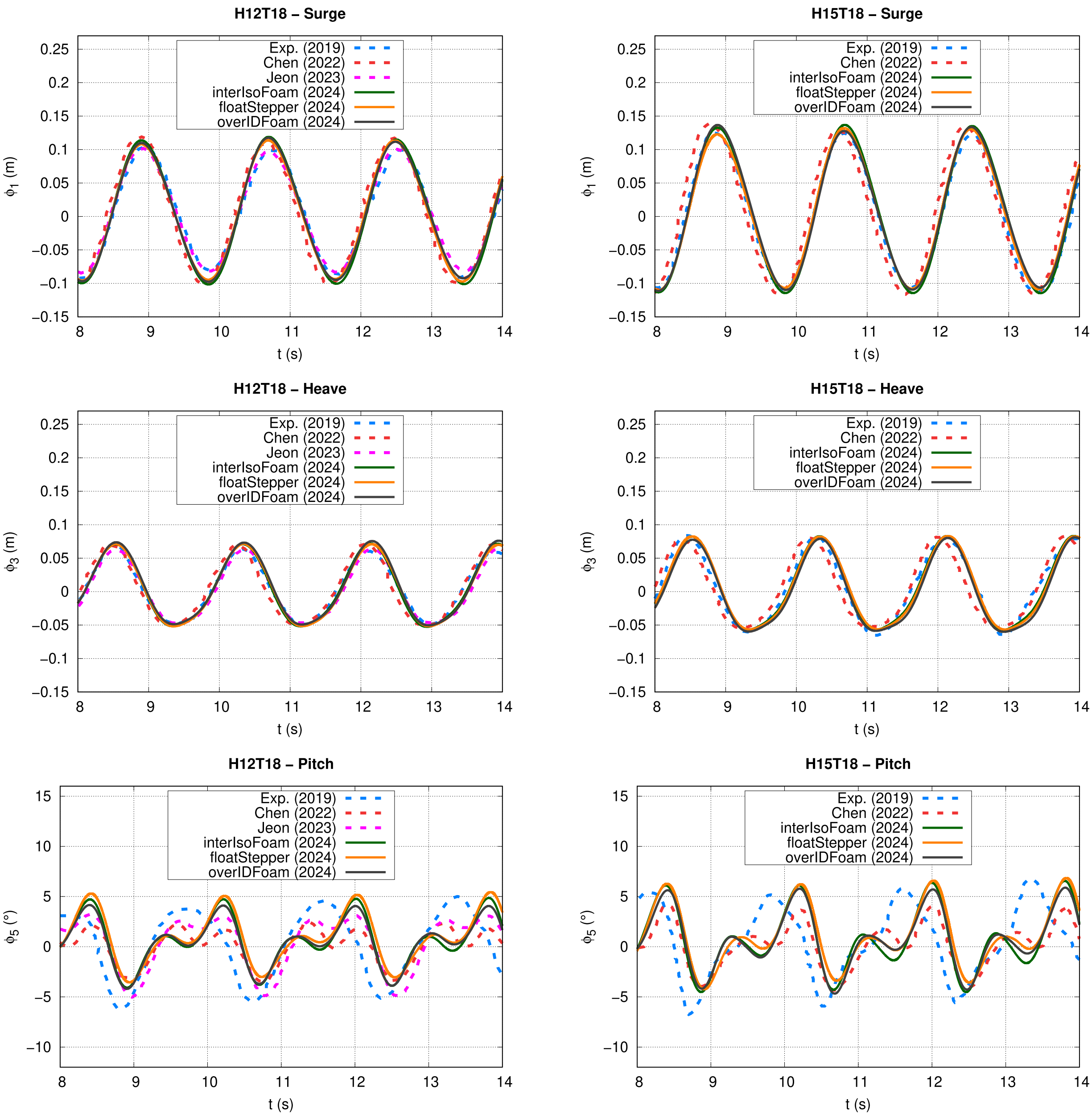


Fig. 7. Comparison of floating box motions with experimental Wu et al. (2019) and numerical (Chen and Hall, 2022; Jeon et al., 2023) for cases H12T18 and H15T18.

obtained by Chen and Hall (2022) and experimental data from Wu et al. (2019), for both cases H12T18 and H15T18. As the solver FloatStepper uses isoAdvector as the interface reconstruction method, Fig. 6 shows the results for interIsoFoam (green line) and overInterDyMFoam (black line). As observed, the results of Chen and Hall (2022) (depicted as dotted red lines) closely align with those from the current simulations using overInterDyMFoam (black line) and interIsoFoam (green line). Overall, the good agreement between the present simulations and the experimental measurements (represented as blue dotted lines) confirms the accuracy of the wave generation and propagation in the numerical wave tank, ensuring that the same wave condition is simulated. For clarity, results from FloatStepper were not included in Fig. 6, since its surface elevation output using the isoAdvector method is identical to that of interIsoFoam. Similarly, results for any additional solver using the VoF method are equivalent to those of overInterDyMFoam. Therefore, only one representative curve is shown per free surface capturing method. The errors associated with both interface capturing approaches (VoF and isoAdvector) are of the same order of magnitude, as shown in Table 10. All simulations were conducted using OpenFOAM version 2306 and version 2 of MoorDyn.

4.2. Floating box motion

Fig. 7 shows the numerical simulations of the floating box motion in regular waves for cases H12T18 and H15T18, obtained using three solvers: interIsoFoam (green line), FloatStepper (yellow line), and overInterDyMFoam (black line). The comparison is presented alongside experimental results (blue dotted line) and previous numerical results from Chen and Hall (2022) (red dotted line) and Jeon et al. (2023) (pink dotted line), in terms of surge (ϕ_1), heave (ϕ_3), and pitch (ϕ_5) response. The corresponding NMSE values for each case are listed in Table 10.

For clarity, results from interFoam were not included in Fig. 7, as they are indistinguishable from those obtained with overInterDyMFoam.

Both solvers use the same VoF-based free surface capturing method, resulting in identical motion predictions. In contrast, `FloatStepper` and `interIsoFoam`, although both based on the `isoAdvector` method, produce distinct results due to differences in their motion coupling algorithms.

For the surge motion, the simulations with the three solvers for the shortest wave case (H12T18) agree well with experimental data, reproducing a similar period and a slightly overestimated amplitude. The results from Chen and Hall (2022) also tend to overpredict amplitude with correct phase, while Jeon et al. (2023) achieve both amplitude and phase alignment with experiments. For the highest wave case (H15T18), the numerical simulations reveal improved surge accuracy relative to H12T18, with `overInterDyMFoam` showing particularly good agreement. Minor discrepancies are still visible in the results of Chen and Hall (2022). Although the surge period remains consistent across all solvers, the `interIsoFoam` solver slightly overestimates the amplitude, whereas `overInterDyMFoam` and `FloatStepper` yield comparable and more accurate responses. According to Table 10, the results obtained using `overInterDyMFoam` show the closest alignment with the experimental data.

There is similarly good agreement in the heave response for both wave cases. For H12T18, the solvers reproduce the period accurately, with minor phase shifts and moderate amplitude overestimation. Both (Chen and Hall, 2022; Jeon et al., 2023) produce consistent results in heave as well. For both wave conditions, the three solvers deliver similar predictions for translational heave motion, matching the experiments in both period and amplitude. In case H15T18, a flattening of the trough in the heave time series causes a shift in the zero-up crossing, slightly affecting the computed period compared to both experimental results and previous numerical studies Chen and Hall (2022).

Discrepancies in the pitch response were observed for both H12T18 and H15T18, with underpredictions in amplitude and the appearance of additional peaks as the rotation angle increased. This phenomenon, where rotational motion significantly affects the results, has also been noted in the studies by (Palm et al., 2016; Chen and Hall, 2022; Jeon et al., 2023; Roenby et al., 2024). The deviations in pitch motion have been attributed by previous authors to inconsistencies related to the modeling of the chain attachment (or fairlead position), or alterations of the inertial properties, such as the distance from the centre of gravity to the waterline. Palm et al. (2016) demonstrated that moving the center of gravity and moment of inertia resulted in better agreement with experimental data. Chen and Hall (2022); Jeon et al. (2023) hypothesized that these discrepancies are due to the definition of the floating box geometry. The absence of a plate for the camera marker receivers attached to the floating box in the simulation (see Wu et al., 2019) was found to cause a significant difference in the moment of inertia, thereby affecting the rotational motion. Roenby et al. (2024) relocated the fairlead attachment point 1 cm higher on the box and repeated the simulation, which led to significant differences in pitch motion. In the mooring setup, certain experimental configurations, such as the chain attachment, cannot be accurately represented by MoorDyn, as discussed in Dominguez et al. (2019), thereby increasing the discrepancies with the numerical model. These limitations, which are associated with geometric simplifications and mooring implementation constraints, are consistent with the pitch motion discrepancies observed in the present results.

4.3. Mooring line tensions

In this section, the predicted mooring line tension response for the two regular wave cases outlined in Table 6 is initially showcased. Following that, an analysis of the impact on tensions and motion responses resulting of the axial stiffness (EA) is conducted. Additionally, a convergence analysis using three different mooring line discretizations of 20, 40, and 80 was performed, as detailed in Appendix B.

Fig. 8 show the anchor and fairlead tensions for one offshore line (line 1) and one onshore (line 3) for case H12T18 predicted by the three solvers interIsoFoam (green line), overInterDyMFoam (black line) and FloatStepper (yellow line). As the four mooring lines are symmetric, the results in lines 2 and 4 are similar to the 1 and 3, respectively. The anchor tensions in line 1 ($\tau_{a,1}$) and line 3 ($\tau_{a,3}$) are compared against experimental data from Wu et al. (2019) (dotted blue line) and other numerical results obtained by Chen and Hall (2022) (dotted red line) and Jeon et al. (2023) (pink line). The obtained anchor tension force in line 1 is in average higher than in the experiments for case H12T18. The larger tension predicted by the model and for the seeward line (line 1) in particular, is caused by the over-prediction of the floating box's surge response, see Fig. 7. The tensions in the offshore lines (lines 1 and 2) exhibit greater dynamics, characterized by snap load conditions (see Fig. 9), whereas the tension in the onshore line (lines 3 and 4) is more consistent and smoother.

The predicted anchor tension force in the rear lines reasonably aligns with experimental data (depicted as points in blue), albeit with a small shift in period. These results are comparable to those obtained by Chen and Hall (2022). The numerical results from Jeon et al. (2023) show a similar period and amplitude when compared to the experimental data. The fairlead tensions in line 3, when compared to line 1, shows slightly higher values, similar to anchor tensions. Nevertheless, Jeon et al. (2023) indicate similar tensions in both line 1 and line 3, see Fig. 8.

Additionally, numerical oscillations can be noted in the low tension region when the cable is completely slack, especially in line 3, see Fig. 9. The tension in the subsequent lower region illustrates how the snap load propagates back and forth along the cable. It can be observed how the slack amplitude is continuously decreasing as the shock propagates. This is mainly attributed to the dynamic friction force from the ground, rather than the shock of hydrodynamic damping. The peak force aligns closely; however, it's noteworthy that the slack contribution during the peak is more prominent in the numerical results compared to the experimental findings. Overall, however, Fig. 9 shows that the complex dynamic behaviour of the cable during the upstrake motion is very well captured by the numerical model.

For the steepest and most non-linear case, H15T18, the anchor and fairlead tension in the offshore lines are higher than those in the onshore lines, see Fig. 10. This is consistent with the under-prediction of the box's surge motion, see Fig. 7, causing smaller tension than in experiments in both line 1 and 3. The anchor tension in lines 1 and 3 gave consistent predictions compared with Chen and Hall (2022), although with a small period delay. It is also observed that the largest tension in numerical results is in the offshore cables for both cases, while it is only occurring for the larger wave case H15T18 in experiments.

The dynamics of the lines are better captured in seeward lines than the leeward lines with transient snap loads, particularly for case H15T18, see Figs. 8 and 9. In addition the tension in anchor line 3 presents almost null tension values in the lower tension region, indicating that as the wave passes, there remains a length of the mooring lines resting on the seabed, as can be observed in Fig. 5b. As expected, the tensions obtained with the interIsoFoam solver are slightly higher than those from overInterDyMFoam and FloatStepper. This is because the displacements in surge for this solver were also higher compared to the other two.

Both anchor and fairlead tensions display again a snap load phenomenon propagating from lower tension region to almost peak amplitude, see Figs. 8 and 9. This vibration in the low tension region on leeward line is caused by the influence of the seabed model and particularly the seabed damping coefficient value. In Chen and Hall (2022), a finite element based model, Moody (Palm et al., 2016) and Moordyn version 1 are compared for the H15T18 case. The Moody model was developed with the intention to capture snap loads in mooring cables with higher accuracy, see Palm et al. (2017). However Moody made no better prediction of tension than the lumped-mass model Moordyn.

This indicates that apart from snap loads modeling, there are other sources of error in estimating line tensions. One stems from the estimation of parameters such as the axial stiffness (EA) and empirical

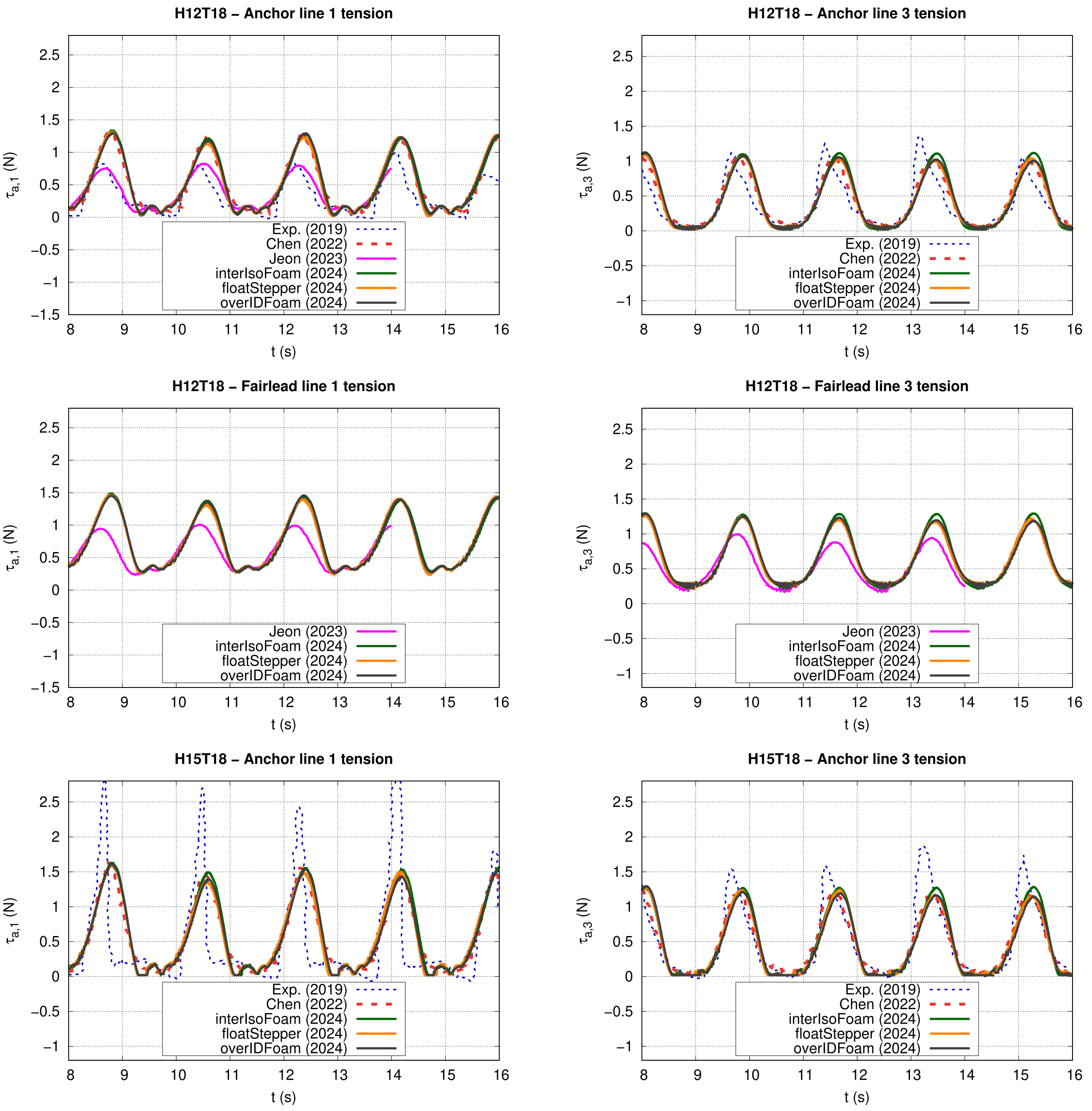


Fig. 8. Anchor ($\tau_{a,i}$) and farilead tensions ($\tau_{f,i}$) for cases H15T18 and H12T18 and comparison between Moordyn version (v1 and v2) and study of mooring line segment discretization.

coefficients, including added mass, drag coefficient, seabed stiffness, and damping coefficients. Inaccuracies in, the estimation of those parameters can contribute to discrepancies in the results. Furthermore, the hydrodynamic forces (inertia and drag) calculated by MoorDyn for each segment are based on quiescent water conditions. As a result, in steep wave scenarios, the mooring line tension may be underestimated because wave-induced velocities and accelerations are not fully incorporated into the force calculations. In such conditions, the significant water particle velocities and accelerations can contribute to additional hydrodynamic loads on the mooring lines. If these effects are simplified or ignored, the forces acting on the mooring lines may be inaccurately evaluated, potentially leading to an underprediction of the total mooring line tension. This limitation can affect the overall accuracy of simulations, particularly under dynamic or extreme sea states.

According to Chen and Hall (2022), another source of error comes from the experiment setup. The chain end connections to the load cells are carried out with two tight cable ties on the flume bottom and to the floating box through an eye hook. As pointed out by Chen and Hall (2022), Moordyn is not able to represent a mooring line being attached to or running through an iron hook, nor the elasticity and loose behaviour of cable ties used to make end connections. Thus, the approximations made to model numerically the mooring lines and the incorrect representation of the mooring line attachment compared to experiments can cause a model's failure to correctly reproduce the results. Extracting the flow motion from the CFD model and feeding them into the mooring model is an ongoing work.

4.4. *Influence of axial stiffness*

The mooring line elasticity of 2.78 MPa was manually tuned by Dominguez et al. (2019) in order to match the experimental tension response. In addition, and following Chen and Hall (2022) criteria, the axial stiffness (EA) was increased from 29 N to 50 N. Fig. 10 displays the numerical results of anchor tensions for line 1 and 3 obtained using

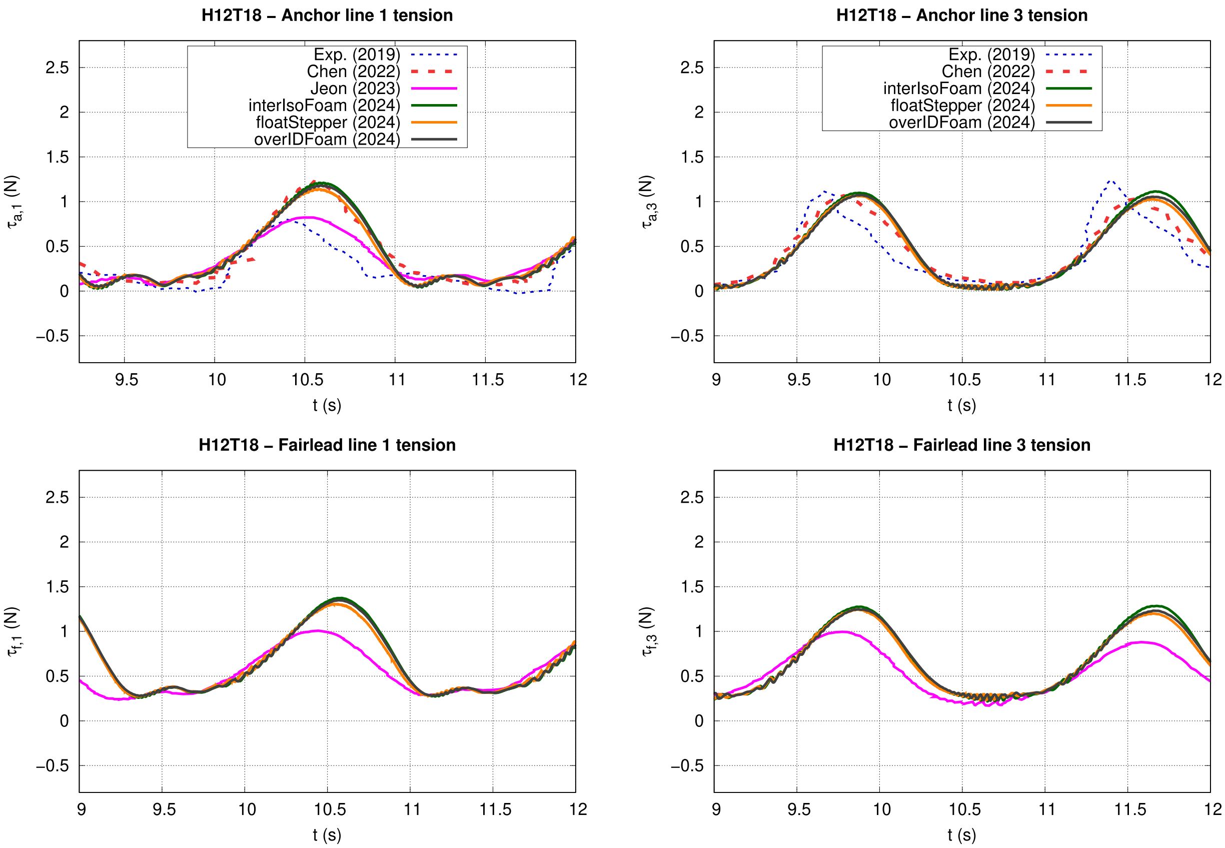


Fig. 9. Temporal zoom into the time histories of anchor and fairlead tension for case H12T18.

the three solvers, along with a temporal zoom into their time histories and the floating box motion response in surge, heave and pitch as a consequence of increasing the axial stiffness.

Increasing axial stiffness adequately captures the magnitude of anchor tension in both the offshore and onshore lines. The snap load condition is again observed and extending from the lower tension region to the peak amplitudes on both anchor and fairlead tension response. However, an overprediction of the surge response is obtained, being the heave practically the same and pitch completely mismatched in period and amplitude than those with experimental and numerical with EA = 29 N.

4.5. *Influence of co factor*

To investigate the influence of the Courant number on solution quality and computational cost, simulations were performed by setting the maximum allowable Courant number to 0.5 and 0.9, respectively, using OpenFOAM's adjustable time stepping algorithm. In this framework, the local Courant number varies throughout the domain and over time, while the maximum value is enforced via the `maxCo` parameter. Fig. 11 illustrates the resulting surface elevation at wave gauge 3 (WG3), as well as the surge and heave motions of the floating box for case H15T18 using the `overInterDyMFoam` solver. Results for `maxCo` equal to 0.5 (dotted yellow line) and 0.9 (black line) show that increasing the Courant number leads to slightly reduced amplitudes in both free surface elevation and body motion. However, these differences remain minimal over the entire time series. As expected, a higher Courant number enables larger time steps, which reduces computational cost. The total wall clock time was reduced from 21.75 h for `maxCo` equal to 0.5 to 11.2 h for `maxCo` equal to 0.9. Although stable simulations with Courant numbers greater than 1 are sometimes possible in OpenFOAM (Pinguet, 2021), such configurations may compromise numerical accuracy. These results indicate that a maximum Courant number equal to 0.9 provides a suitable balance between accuracy and computational efficiency for this case.

4.6. *Influence of MaddUpdateFreq*

Finally, Fig. 12 displays the results of the floating box motions (surge, heave and pitch) obtained with the FloatStepper algorithm and for two MaddUpdate values. A low MaddUpdateFreq value of 100 means the added mass is updated more frequently, leading to more accurate simulations but at the cost of a higher computational cost, see Table 9. In this case the solver recalculates the added mass every 100 time steps. For a higher MaddUpdateFreq (1000), the added mass is updated less frequently, each 1000 time steps, reducing the computational time but potentially introducing some inaccuracy. For the surge and heave motions, the choice of a low or high MaddUpdateFreq has a small impact in their amplitudes. In the case of pitch, the differences between both simulations have a greater impact, although it is not significant. This implies that the added mass matrix changes slowly throughout the simulation, making the choice of $MaddUpdateFreq = 1000$ the best option for saving computational resources. In conclusion: this parameter has a negligible influence on accuracy, and has to be turned with care to minimize the CPU time.

Finally, Fig. 12 presents the predicted motions of the floating box (surge, heave, and pitch) using the `FloatStepper` algorithm for different values of the `MaddUpdateFreq` parameter, which controls the frequency at which the added mass matrix is updated during the simulation. A lower value of `MaddUpdateFreq` (e.g., 100) results in more frequent updates, every 100 time steps, leading to improved accuracy at the expense of increased computational cost. Conversely, a higher value (e.g., 1000) updates the added mass matrix only once every 1000 time steps, reducing computational effort but potentially introducing some inaccuracies. Runtime results are summarized in Table 9.

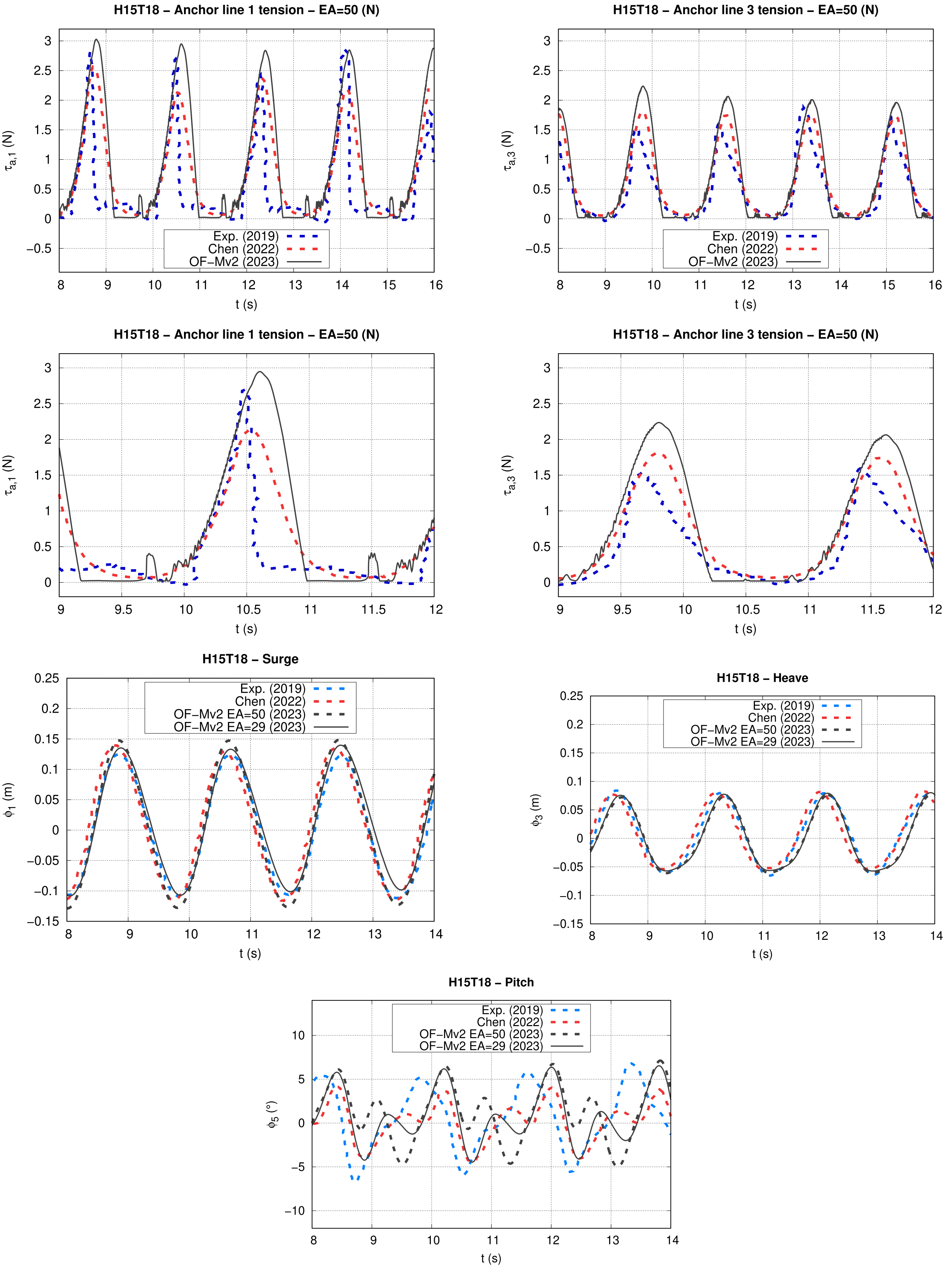


Fig. 10. Comparison of floating box motions and anchor ($\tau_{a,i}$) and farilead tensions ($\tau_{f,i}$) with experimental Wu et al. (2019) and numerical Chen and Hall (2022) for case H15T18 with increased axial stiffness (EA = 50 N).

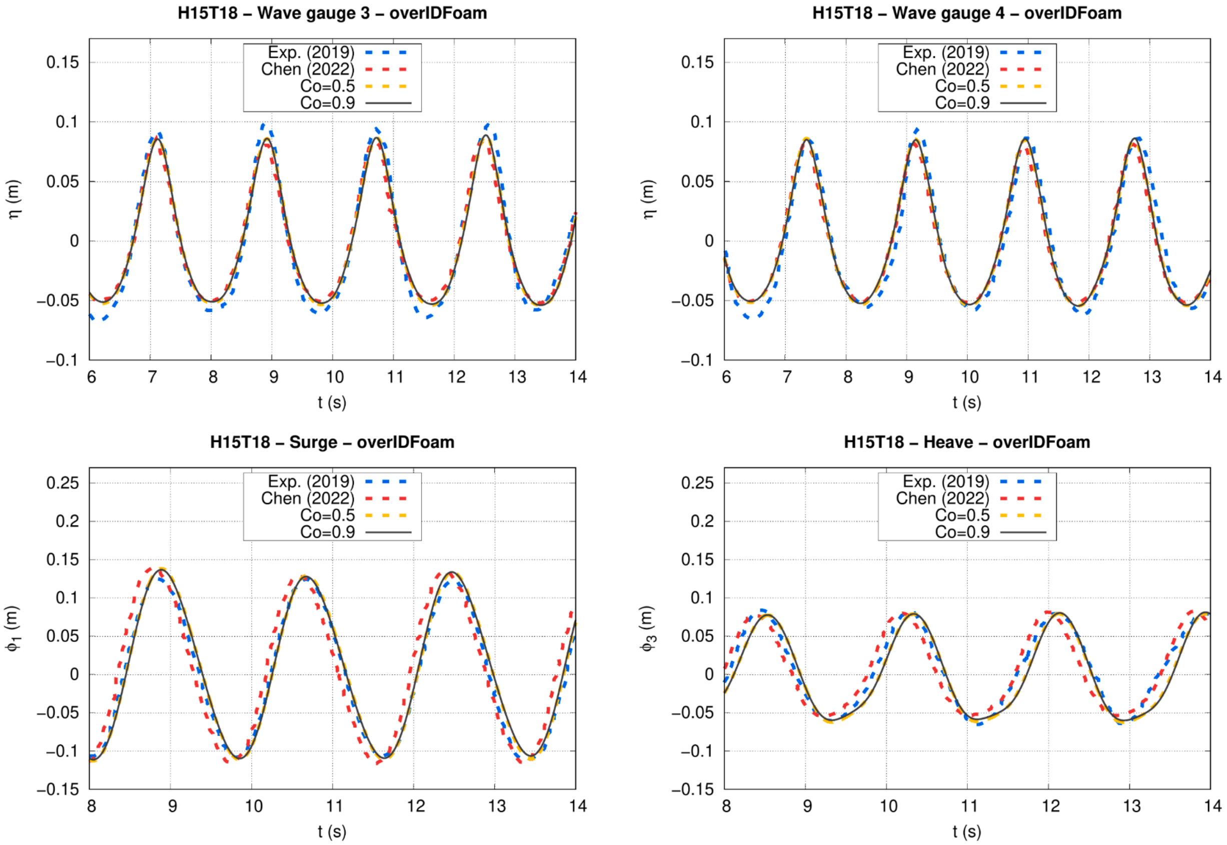


Fig. 11. Comparison of floating box motions (surge and heave) and surface elevation in WG3 for two Co values (0.5 and 0.9) with experimental Wu et al. (2019) and numerical Chen and Hall (2022).

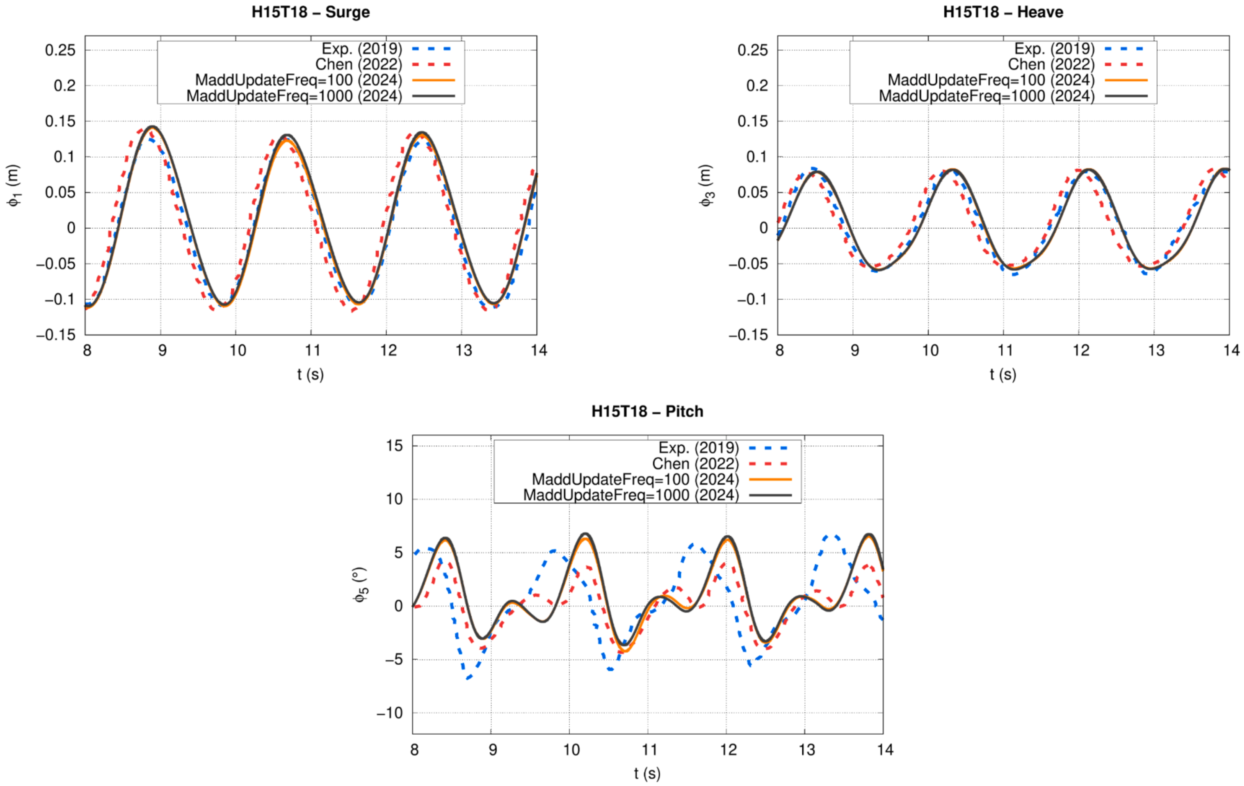


Fig. 12. Comparison of floating box motions with experimental Wu et al. (2019) and numerical Chen and Hall (2022) for case H15T18 and two MaddUpdateFreq values (100 and 1000) for the solver FloatStepper.

In this particular test case, where the submerged volume of the floating body remains relatively stable throughout the simulation, the effect of MaddUpdateFreq on the resulting motions is modest. For surge and heave, the differences in amplitude between the two configurations are negligible. For pitch motion, a more noticeable deviation is observed; however, it remains small in magnitude and does not significantly affect the overall dynamics. These findings indicate that the added mass matrix evolves slowly in time, making MaddUpdateFreq $= 1000$ a suitable compromise for reducing computational cost without sacrificing physical accuracy.

Nevertheless, caution is warranted when generalizing these results. In scenarios involving rapidly varying hydrodynamic conditions-such as a body entering the water from air or undergoing large changes in immersion-the added mass may change considerably over short timescales. In such cases, setting MaddUpdateFreq to a low value (e.g., 1) may be necessary to accurately capture the fluid-structure interaction dynamics. Therefore, while reducing MaddUpdateFreq can offer noticeable gains in computational efficiency (up to approximately 20 % in this case), it should only be increased when one has prior knowledge that the added mass remains nearly constant during the simulation. Otherwise, one risks obtaining fast but physically inaccurate results.

In conclusion, the influence of MaddUpdateFreq on accuracy is case-dependent. For configurations with limited variation in the submerged geometry, the parameter has limited influence on the predicted response and can be tuned to reduce CPU time. However, in more dynamic scenarios, careful adjustment is essential to preserve accuracy.

5. Conclusions

This study focuses on identifying an efficient methodology for the simulation of floating bodies with simple geometries, coupled with mooring dynamics models and subjected to regular wave fields, with the aim of applying it to platforms such as FOWTs, WECs, etc., in future works. The free surface elevation, motion response, and mooring line tensions are examined for a simple geometry subjected to two types of regular waves. Version 2 of MoorDyn is used as the mooring dynamics model, which offers several advantages over the first version used in simulations by other researchers. Several new features have been implemented in this second version, such as new time integration schemes, the option to use a CFL number for adaptive time steps, seabed effect and local time stepping, all of which are available in its official repository. Additionally, in order to test different available solvers, coupling was carried out with FloatStepper, an innovative solver that addresses the added mass instability problem. This coupling is available in the GitHub repository https://github.com/FloatStepper/FloatStepper.

The coupled CFD mooring approach demonstrated good agreement with experimental data for surface elevation, surge, and heave motions. The accuracy was particularly notable for the highest wave case (H15T18), where all solvers consistently captured the dynamic behavior of the floater. Although the predicted tensions in anchor line 1 showed significant deviations, especially in the shortest wave case (H12T18), these discrepancies were primarily attributed to the overestimation of surge motion and are consistent with trends reported in previous numerical studies. As such, they do not compromise the validity of the overall methodology.

Mesh resolution and solver efficiency were key aspects of this work. A discretization of 15 cells per wave height and 357 cells per wavelength proved sufficient for accurate wave and motion predictions, as confirmed through a dedicated grid convergence study. The overset mesh strategy, in particular, enabled high-quality resolution near the body with a manageable cell count of approximately 3.4 million cells. In contrast, the morphing mesh strategy, while accurate, resulted in a higher computational cost due to full-domain refinement and increased sensitivity to mesh deformation. Another important aspect addressed was computational performance. The use of a higher Courant number ($Co = 0.9$) significantly reduced simulation time without noticeably affecting solution quality. Among the solvers tested, overInterDyMFoam emerged as the most computationally efficient, while FloatStepper provided improved numerical stability in simulations influenced by added mass effects, albeit with increased runtime. In addition, this study explored the effect of varying the MaddUpdateFreq parameter in the FloatStepper solver, which controls how often the added mass matrix is recomputed. It was found that reducing the update frequency can substantially improve computational efficiency in steady-state or weakly time-dependent problems, with minimal accuracy degradation. However, more dynamic cases demand a finer temporal resolution of the added mass calculation to maintain fidelity. As such, the parameter can be an effective optimization tool, provided the hydrodynamic behavior of the system is well understood.

Moreover, this work offers a comparative evaluation of three multiphase solvers with different interface capturing strategies and mesh motion techniques: overInterDyMFoam, interIsoFoam, and FloatStepper. The overset-based overInterDyMFoam solver proved to be the most time-efficient and robust under large-amplitude motion, thanks to its grid independence and ability to handle rigid body translations without distortion. Meanwhile, interIsoFoam and FloatStepper, both based on geometric interface reconstruction with morphing meshes, achieved higher interface fidelity at the cost of increased mesh sensitivity and computational effort. The FloatStepper solver remains especially valuable in contexts where added mass instabilities could compromise conventional solvers, and its modular design allows tailored trade-offs between stability and efficiency. The methodology developed in this study combines second-generation mooring modeling capabilities with advanced CFD techniques in a reproducible, fully open-source framework. It provides a solid foundation for simulating more complex floating platforms, including those used in floating wind, wave energy, and aquaculture applications. Future work will focus on extending the overset approach to the FloatStepper solver and applying the framework to irregular sea states, survival scenarios, and full-scale configurations. Moreover, the insights gained here will inform the design of dedicated experimental campaigns for further validation.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Tanausú Almeida-Medina: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization; **Jose Luis Cercos-Pita:** Visualization, Validation, Software, Formal analysis; **Anne Gosset:** Supervision, Formal analysis, Conceptualization; **Vicente Díaz-Casás:** Supervision, Formal analysis, Conceptualization.

Acknowledgments

The first author has been funded by the financial support provided by Universidade da Coruña, Campus Industrial de Ferrol (Ferrol, 15403, A Coruña, Spain) and Coremarine Solutions SL (Madrid, Spain). The authors wish to thank the ‘Red Española de Supercomputación’ (RES) and ’Centro de Supercomputación de Galicia’ (CESGA) for the attribution of special computational resources at FinisTerrae III used in this work (IM-2024-2-0021).

Appendix A. Grid convergence analysis

An analysis of grid convergence was performed using three grid densities designated as A, B and C (see Table 11) and the solver overInterDyMFoam. The medium grid size (mesh B) was 0.01 m × 0.01 m × 0.008 m for the overset mesh. The background mesh, which represents the Numerical Wave Tank (NWT), is composed of cells with a size 2^3 times (level 3) larger than the overset cells. All the simulations were conducted for up to 18 s, with the adjustable maximum time step size (δt) controlled by a maximum Courant number of 0.9. The three mesh discretization criteria was based on the minimum number of cells per wave hegiht (CPH) and cells per wave lenght (CPL), according to ITTC guidelines ITTC (2014).

Fig. 12 shows the comparison of the free surface elevation at wave gauges WG2, WG3 and WG4, and the surge, heave and pitch motions of the floating box for case H12T18. As for the free surface elevation, the three meshes A, B and C, show virtually no differences, with convergence being observed in the motion response. In the case of surge (ϕ_1), the differences are barely noticeable among the three grids. Some slight discrepancies can be observed in the heave motion where the negative amplitude is lower and positive amplitude is larger for mesh A. It can be considered that the heave motions converged between the mesh B and C. For the pitch motions, there is more sensitive to grid resolutions than for surge or heave motions. In general, the results with the medium mesh size (mesh B) are consistent, capturing well the surface elevation and floating box motions, and it was selected for simulating all the cases.

Table 11
Mesh parameters for different grid density of background mesh (δx_b) and overset mesh (δx_o).

ID	δx_b (m)	δx_o (m)	δt (s)	N	CPH	CPL
A	0.096	0.012	0.042	2.1×10^6	10	238
B	0.064	0.008	0.028	2.9×10^6	15	357
C	0.048	0.006	0.021	5.9×10^6	20	476

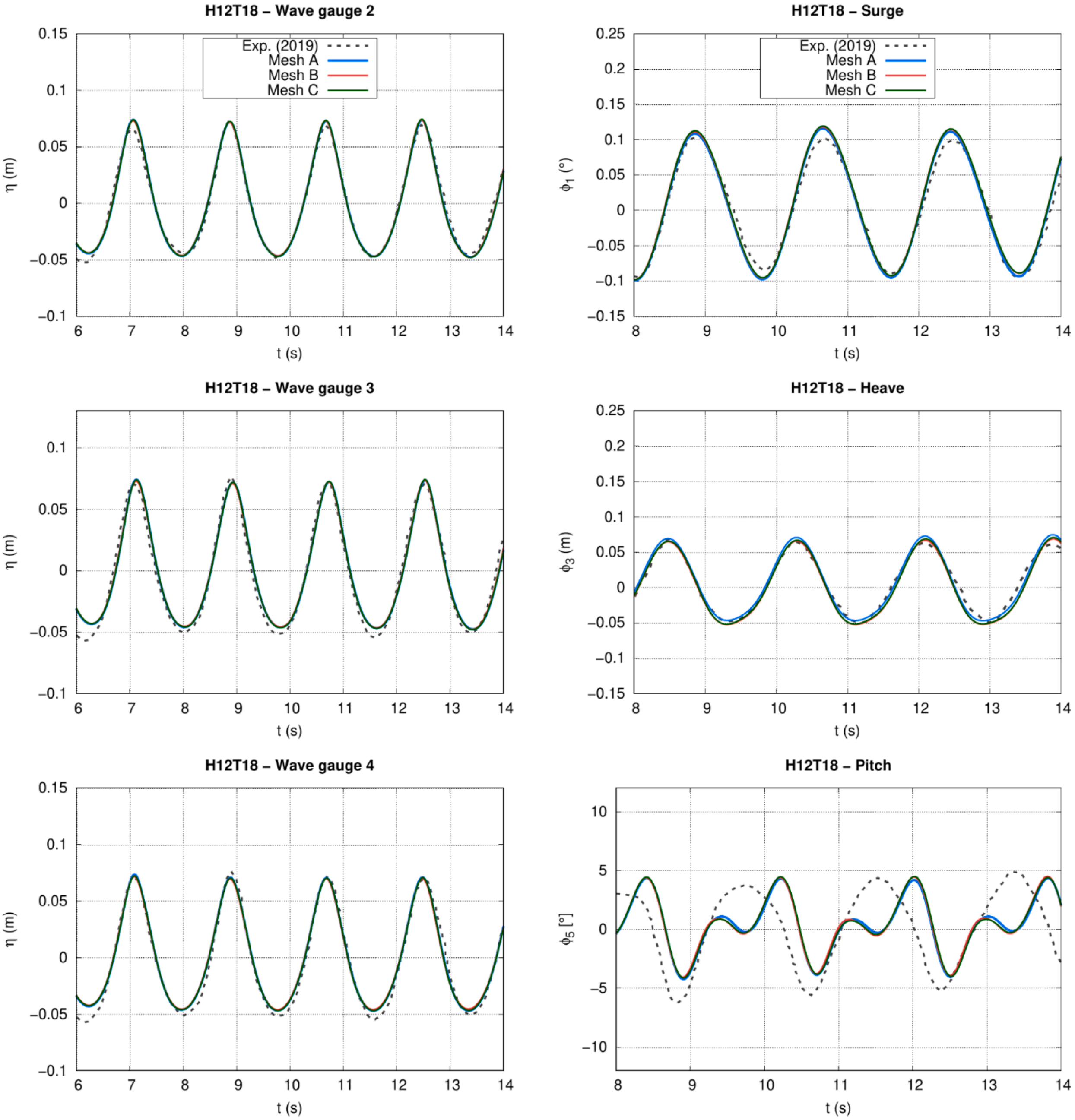


Fig. 13. Comparison of surface elevation and floating box motions by the three mesh discretizations for case H12T18.

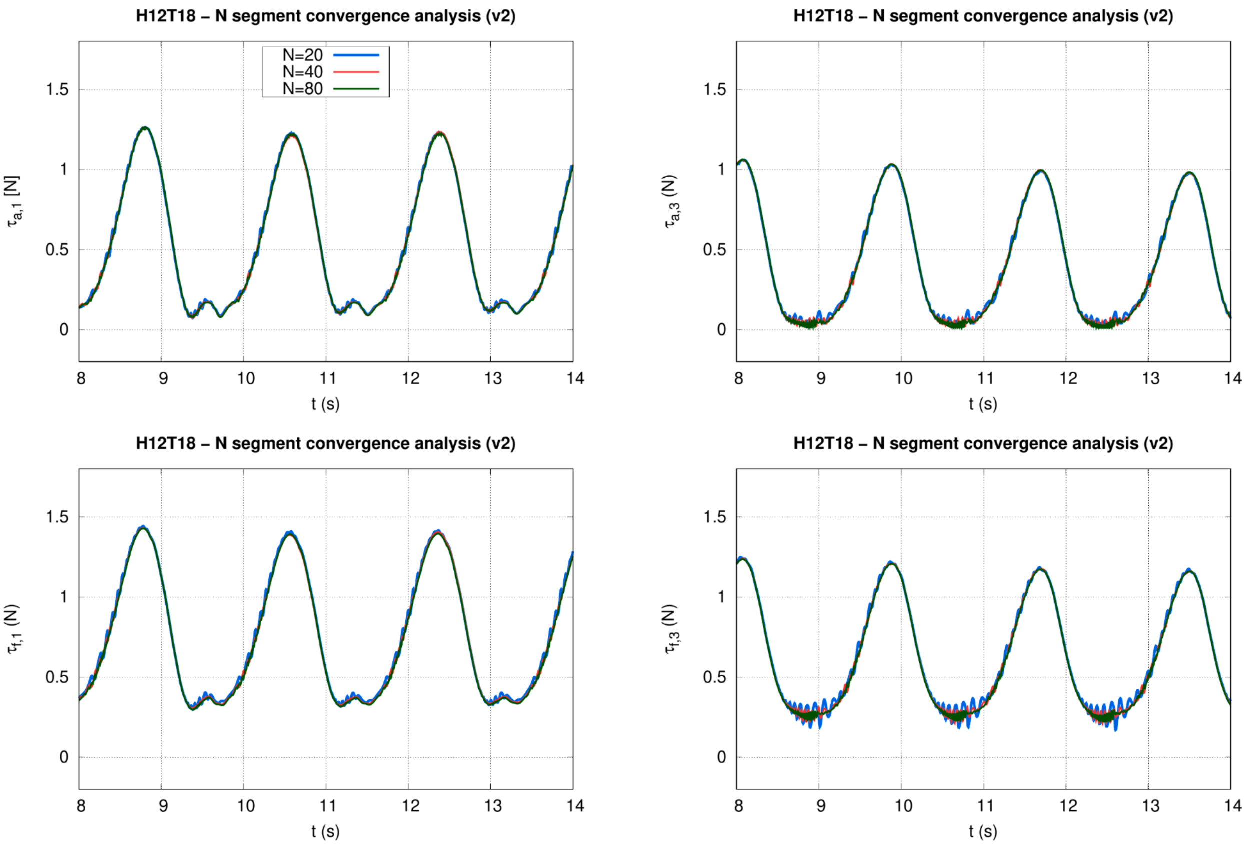


Fig. B.14. Anchor ($\tau_{a,i}$) and fairlead tensions ($\tau_{f,i}$) obtained from an analysis of mooring line segment discretization using Moordyn version 2 and overInterDyMFoam solver for case H12T18.

Appendix B. Influence of line discretization

In the lumped-mass model, parameters, such as internal damping for each segment, simulation time step, and the dynamic frequencies of the mooring line, are dependent on line discretization. To address this, a dedicated convergence analysis of the mooring line discretization was conducted by dividing the cables into 20, 40, and 80 segments (N) using the Moordyn version 2 and overInterDyMFoam solver for case H12T18. The simulations are conducted for up to 18 s, with the adjustable time step size controlled by a maximum Courant number of 0.9. The same numerical scheme Runge–Kutta 4th order (RK4), time step to use in mooring integration (dtM), time interval for analyzing convergence (dtIC) and bottom parameters (stiffness and damping) were used for all the simulations.

Fig. B.14 shows the comparison of anchor and fairlead tension in lines 1 and 3 using the three discretization criteria proposed. The tension responses for mooring lines divided into 40 and 80 segments are consistent, whereas for lower discretization $N = 20$ shows numerical resonance responses obtained with shock peaks starting at lower tension region and until the maximum peak. This phenomenon is more evident in the fairlead tension of line 3.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.oceaneng.2025.121764.

References

Aliyar, S., Ducrozet, G., Bouscasse, B., Bonnefoy, F., Sriram, V., Ferrant, P., 2022. Numerical coupling strategy using HOS-OpenFOAM-MoorDyn for OC3 Hywind SPAR type platform. Ocean Eng. 263, 112206.

Brown, D.T., Mavrakos, S., 1999. Comparative study on mooring line dynamic loading. Mar. struct. 12, 131–151.

Chen, H., Almeida-Medina, T., Cercos-Pita, J.L., 2024. CFD simulation of multiple moored floating structures using openfoam: an open-access mooring restraints library. Ocean Eng. 303, 117697.

Chen, H., Hall, M., 2022. CFD simulation of floating body motion with mooring dynamics: coupling moordyn with openfoam. Appl. Ocean Res. 124, 103210.

Chen, L., Basu, B., 2018. Wave-current interaction effects on structural responses of floating offshore wind turbines. Wind Energy 22, 327–339.

Chen, L., Basu, B., Nielsen, S. R.K., 2018. A coupled finite difference mooring dynamics model for floating offshore wind turbine analysis. Ocean Eng. 162, 304–315.

Chen, L.H., Qian, Z., Ma, W., Bai, Y., Li, D., Causon, C., Mingham, 2019. Application of an overset mesh based numerical wave tank for modelling realistic free-surface hydrodynamic problems. Ocean Eng. 176, 97–117.

Chen, Y. Y.K., Liu, D.D., Meringolo, J.M., Hu, 2023. Study on the hydrodynamics of a twin floating breakwater by using sph method. Coastal Eng. 179, 104230.

Devolder, B., Troch, P., Rauwoens, P., 2017. Accelerated numerical simulations of a heaving floating body by coupling a motion solver with a two-phase fluid solver. Int. Conf. Adv. Comput. Methods Eng. 77, 1605–1625.

Dominguez, J.M., Crespo, A.J., Hall, M., Altomare, C., Wu, M., Stratigaki, V., Gómez Gesteira, M.G., 2019. In: SPH simulation of floating structures with moorings, p. 153.

Feng, X., Bai, W., Chen, X.B., Qian, L., Ma, Z.H., 2017. Numerical investigation of viscous effects on the gap resonance between side-by-side barges. Ocean Eng. 145, 44–58.

Gao, J., He, Z., Huang, X., Liu, Q., Zang, J., Wang, G., 2021. Effects of free heave motion on wave resonance inside a narrow gap between two boxes under wave actions. Ocean Eng. 224, 108753.

Gao, J.L., Lyu, J., Wang, J. J.H., Zhang, Q., Liu, J., Zang, T., Zou, 2022. Study on transient gap resonance with consideration of the motion of floating body. China Ocean Eng. 36, 994–1006.

Gatin, I., Vukčević, V., Jasak, H., Rusche, H., 2017. Enhanced coupling of solid body motion and fluid flow in finite volume framework. Ocean Eng. 143, 295–304.

Hall, M., 2020. Moordyn v2: new capabilities in mooring system components and load cases. In: ASME 2020 39th International Conference on Ocean, Offshore and Arctic Engineering. American Society of Mechanical Engineers Digital Collection.

Hall, M., Goupee, A., 2015. Validation of a lumped-mass mooring line model with deepcwind semisubmersible model test data. Ocean Eng. 104, 590–603.

Hall, M., Sirnivas, S., Yu, Y.H., 2021. Implementation and verification of cable bending stiffness in moordyn. Vol. 84768. American Society of Mechanical Engineers. ASME 2021 3rd International Offshore Wind Technical Conference. page V001T01A011.

He, M., Liang, D., Ren, B., Li, J., Shao, S., 2023. Wave interactions with multi-float structures: SPH model, experimental validation, and parametric study. Coastal Eng. 184, 104333.

Higuera, P., Lara, J.L., Losada, I.J., 2013. Realistic wave generation and active wave absorption for Navier–Stokes models: application to openfoam. Coastal Eng. 71, 102–118.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. J. Comput. Phys. 39, 201–225.

Huang, L., Li, Y., Benites-Munoz, D., Windt, C., Feichtner, A., Tavakoli, S., Paredes, R., Davidson, J., Quintuna, T., Ransley, E., Colombo, M., Li, M., Cardiff, P., Tabor, G., 2022. A review on the modelling of wave-structure interaction based on openfoam. OpenFOAM J. 2, 116–142.

Ittc, 2014. Recommended Procedures and Guidelines: Practical Guidelines for Ship Resistance CFD.

Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2012. A wave generation toolbox for the open-source CFD library: openfoam®. Int. J. Numer. Methods Fluids 70 (9), 1073–1088.

Jasak, H., Tuković, Ž., 2010. Dynamic mesh handling in openfoam applied to fluid-structure interaction simulations.

Jeon, W., Park, S., Cho, S., 2023. Moored motion prediction of a semi-submersible offshore platform in waves using an openfoam and moordyn coupled solver. Int. J. Nav. Archit. Ocean Eng. 15, 100544.

Jiang, C., Moctar, O.E., 2022. Extension of a coupled mooring-viscous flow solver to account for mooring-joint-multibody interaction in waves. Vol. 9.

Larsen, B.E., Fuhrman, D.R., 2018. On the over-production of turbulence beneath surface waves in reynolds-averaged Navier–Stokes models. J. Fluid Mech. 853, 419–460.

Larsen, D. R. B.E., Fuhrman, J., Roenby, 2018. Performance of interfoam on the simulation of progressive waves. Coastal Eng. J. 61, 380–400.

Lee, S.C., Song, S., Park, S., 2021. Platform motions and mooring system coupled solver for a moored floating platform in a wave. Processes 9 (8), 1393.

Masciola, M., Jonkman, J., Robertson, A., 2013. Implementation of a multisegmented, quasi-static cable model. In: International Ocean and Polar Engineering Conference.

Palm, J., Eskilsson, C., 2018. MOODY User's manual version 2.0.

Palm, J., Eskilsson, C., Bergdahl, L., 2017. An hp-adaptative discontinuous Galerkin method for modelling snap loads in mooring cables. Ocean Eng. 144, 266–276.

Palm, J., Eskilsson, C., Paredes, G.M., Bergdahl, L., 2016. Coupled mooring analysis for floating wave energy converters using CFD: formulation and validation. Int. J. Mar. Energy 16, 83–99.

Paredes, G.M., Palm, J., Eskilsson, C., Bergdahl, L., Taveira-Pinto, F., 2016. Experimental investigation of mooring configurations for wave energy converters. Int. J. Mar. Energy 15, 56–67.

Pinguet, R., 2021. Hydrodynamics of semi-submersible floater for offsore wind turbines in highly nonlinear waves using Coputational Fluid Dynamics (CFD), and validation of overset meshing technique in a numerical wave tank. Technical Report. Ph.D. thesis. Ecole Centrale de Marseille.

Pinguet, R., Kanner, S., Benoit, M., Molin, B., 2020. Validation of open-source overset mesh method using free-decay tests of floating offshore wind turbine. In: International Society of Offshore and Polar Engineers (ISOPE).

Rentschler, M., Chandramouli, P., Vaz, G., Viré, A., Gonçalves, R.T., 2022. CFD code comparison, verification and validation for decay tests of a FOWT semi-submersible floater.

Roenby, J., Aliyar, S., Bredmose, H., 2024. A robust algorithm for computational floating body dynamics. R. Soc. Open Sci. 11 (4), 231453.

Roenby, J., Bredmose, H., Jasak, H., 2016. A computational method for sharp interface advection. R. Soc. Open Sci. 3 (11). The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License. which permits unrestricted use, provided the original author and source are credited. http://creativecommons.org/licenses/by/4.0/.

Wu, M., Stratigaki, V., Troch, P., Altomare, C., Verbrugghe, T., Crespo, A., Cappietti, L., Hall, M., Gómez-Gesteira, M., 2019. Experimental study of a moored floating oscillating water column wave-energy converter and of a moored cubic box. Energies 12 (10), Article 1834.

Zeng, X., Shao, Y., Feng, X., Xu, K., Jin, R., Li, H., 2024. Nonlinear hydrodynamics of floating offshore wind turbines: a review. Renew. Sustain. Energy Rev. 191.

Zhang, W., Calderon-Sanchez, J., Duque, D., Souto-Iglesias, A., 2024. Computational fluid dynamics (CFD) applications in floating offshore wind turbine (FOWT) dynamics: a review. Appl. Ocean Res. 150, 104075.