HydroChrono: a hydrodynamics package for modeling wave energy converters and floating offshore wind turbines

David Ogden, Zuriah Quinton, Tristan de Lataillade, Maxime Pallud and Sahand Sabet

Abstract—**This paper introduces HydroChrono, a novel numerical modeling package developed for simulating wave energy converters and designed to work seamlessly with the open-source physics engine, Project Chrono. We present the underlying theory, provide verification of the accuracy of the implementation, and offer a performance comparison against existing tools. Additionally, we demonstrate the application of HydroChrono to a floating offshore wind turbine model, utilizing finite-element modeling for the tower, blades, and moorings. This successful application opens up future opportunities to model floating bodies with finite-element modeling in the context of marine renewable energy systems. HydroChrono aims to facilitate the development, optimization, design, and analysis of wave energy converters and other floating renewable energy systems, enhancing their performance, survivability, and cost-effectiveness.**

Index Terms—**Numerical Modeling, Wave Energy, Offshore Wind, Multibody Dynamics, Hydrodynamics, Potential Flow, Project Chrono, BEM, Capytaine**

I. Introduction

A. Background & motivation

WAVE energy converters (WECs) are a promising marine energy technology with the potential to make a significant contribution to a sustainable energy future. As clean energy demand increases, efficient, reliable, and cost-effective WECs become imperative. Existing tools like WEC-Sim have been instrumental in WEC design, analysis, and optimization but face several limitations, including licensing restrictions, speed limitations, modeling complex hydrodynamic interactions, finite-element modeling for deformable bodies, and advanced control strategy integration.

Project Chrono [1] is a versatile, open-source physics engine boasting finite-element capabilities, collision detection, and autonomous vehicle simulation. Chrono has already been used to model WECs via its coupling with the high-fidelity, smoothed particle hydrodynamics (SPH) packages, such as GPUSPH [2] and DualSPHysics [3], [4]. Capitalizing on Project Chrono's potential and the importance of open-source software for wave energy research, we developed HydroChrono, a potential-flow-based hydrodynamics package that integrates seamlessly with Project Chrono. HydroChrono aims to address existing tool limitations. By building on Project Chrono and its existing coupling to high-fidelity fluid-solid interaction software, we also aim to eventually develop a multifidelity WEC numerical modeling framework—to simulate a device with either potential flow or SPH—for faster validation, drag coefficient estimation, and more focused studies (e.g., extreme waves or identifying slamming events). Its hoped that this can enable WEC developers to simulate more complex phenomena earlier in the numerical modeling campaign.

Additionally, the rising interest in floating offshore wind turbines (FOWTs) highlights an opportunity to foster collaboration in the development of numerical modeling tools that cater to the entire offshore renewable energy domain, given the substantial overlap in the underlying theoretical foundations. This interdisciplinary approach can lead to more comprehensive and versatile modeling tools, ultimately benefiting both WEC and FOWT research and development.

B. Project Chrono

Chrono is an open-source, cross-platform physics engine designed for simulating rigid and flexible multibody dynamics, fluid-solid interactions, and granular dynamics, as shown in Figure 1.

It was developed with the aim of providing a robust, flexible, and extensible framework for researchers and developers working in various fields such as robotics, vehicle dynamics, biomechanics, and aerospace engineering, and it has the following capabilities:

- Rigid and flexible multibody dynamics: Chrono can simulate complex mechanical systems consisting of interconnected rigid and flexible bodies (with the latter using the finite-element method). The engine supports many joint/constraint types and frictional contacts/collisions.
- Fluid-solid interactions: Chrono couples its multibody dynamics solver with the SPH method, a

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Water Power Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

David Ogden, Zuriah Quinton and Sahand Sabet are with the National Renewable Energy Laboratory, 5013 Denver West Parkway, Golden, CO 80401 U.S. (e-mail: david.ogden.nrel@outlook.com, zuriah.quinton@nrel.gov, sahand.sabet@nrel.gov).

Tristan de Lataillade and Maxime Pallud are with TotalEnergies OneTech, 91120 Palaiseau, France (e-mail: tristan.delataillade@totalenergies.com, maxime.pallud@totalenergies.com).

Digital Object Identifier:
https://doi.org/10.36688/ewtec-2023-473

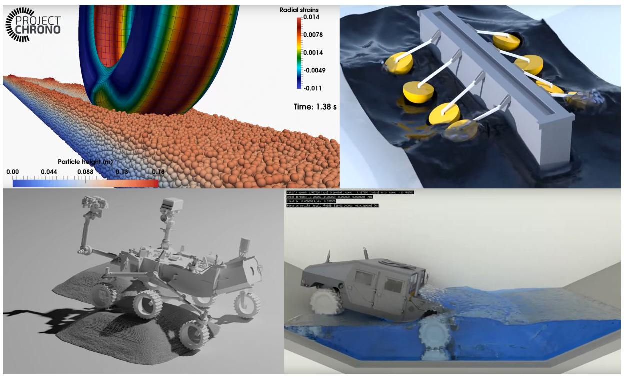

Fig. 1. Montage to showcase some of the features/capabilities/use-cases of Project Chrono; including fluid-solid interaction, finite-element analysis, discrete-element method, and autonomous vehicle control (images obtained from https://projectchrono.org/).

popular meshless Lagrangian approach for modeling fluid flow.

- Granular dynamics: Chrono includes a module for modeling large-scale granular dynamics, enabling the simulation of phenomena such as soil-tool interactions.
- Vehicle dynamics: Chrono simulates the behavior of wheeled and tracked vehicles, including suspension systems, tire models, and powertrains.
- Parallel computing: Chrono utilizes CUDA for GPUs, OpenMP for multi-core computing and MPI for multi-node computing.
- Interfacing and extensibility: The engine provides APIs for C++ and Python, enabling seamless integration with existing software and tools. Furthermore, Chrono has been coupled to MATLAB/Simulink. Since Chrono is fully open-source, users can modify, adapt, and extend to the code to meet their needs.

II. Theory

A. Multibody dynamics theory in Project Chrono

In this section, the fundamental multibody dynamics theory used by Chrono is described. For a multibody system comprising n_b bodies, the generalized position of each body i is represented by the vector $r_i = [x_i, y_i, z_i]^T$, and its orientation is given by the Euler angles, $e_i = [\psi_i, \theta_i, \phi_i]^T$. Hence, the generalized coordinates for the entire system can be expressed as:

$$q = [r_1^T e_1^T \dots r_{n_b}^T e_{n_b}^T]^T \in \mathbb{R}^p,\ p = 6n_b \tag{1}$$

Where R is the set of real numbers and p is a variable representing the total number of generalized coordinates in the multibody system (i.e., each body has three variables for position and three for orientation).

In practice, quaternions (which have four variables) are typically used to describe orientation to avoid issues such as gimbal lock. If flexible bodies are present, deformation modes can be added to the position generalized coordinates.

Constrained mechanical systems have joints connecting bodies, which impose restrictions on the relative motion and result in constraints on the generalized coordinates. The kinematic constraints are formulated as algebraic expressions involving generalized coordinates:

$$\Phi(q,t) = [\Phi_1(q,t) \dots \Phi_m(q,t)]^T = 0 \tag{2}$$

where m represents the total number of independent constraint equations that the generalized coordinates must satisfy throughout the simulation. For simplicity, only holonomic constraints (i.e., position-dependent constraints) are considered in this section.

Taking the time derivative of eq. (2) yields the velocity kinematic constraint equation:

$$\Phi_q(q,t)\dot{q} + \Phi_t(q,t) = 0 \tag{3}$$

Here, $\dot{q}$ denotes generalized velocity, and the subscripts denote the partial derivatives: $\Phi_q = \left[\frac{\partial \Phi_i}{\partial q_j}\right]$ and $\Phi_t = \left[\frac{\partial \Phi}{\partial t}\right]$. Differentiating eq. (3) with respect to time results in the acceleration kinematic constraint equation:

$$\Phi_q(q,t)\ddot{q} + (\Phi_q(q,t)\dot{q})q\dot{q} + 2\Phi qt(q,t)\dot{q} + \Phi_{tt}(q,t) = 0 \tag{4}$$

The time evolution of the mechanical system, governed by the Lagrange multiplier form of the constrained equations of motion, must satisfy eqs. (2) to (4) at all times. The constrained equations of motion are given by:

$$M(q)\ddot{q} + \Phi_q^T(q)\lambda = Q(\dot{q}, q, t) \tag{5}$$

where $M(q) \in \mathbb{R}^{p\times p}$ represents the generalized mass matrix, and $Q(\dot{q}, q, t) \in \mathbb{R}^p$ denotes the action force acting on the generalized coordinates $q \in \mathbb{R}^p$. The reaction force $\Phi_q^T(q)\lambda$, where $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier associated with the kinematic constraints, results from the constraint equations in eq. (2). The combination of differential equations describing the system dynamics and algebraic equations describing the system's constraints leads to eq. (5) being referred to as a *differential-algebraic equation* (DAE). Because the constraint equations are described at the position level, these DAEs are referred to as "index-3."

The index-3 DAEs are neither linear nor ordinary differential. Hence, the HHT algorithm addresses these challenges by discretizing the equations of motion and enforcing position-level kinematic constraints. The core equation for the HHT integrator is:

$$\frac{1}{1+\alpha}(M\ddot{q})_{n+1} + (\Phi_q^T\lambda - Q)_{n+1} - \frac{\alpha}{1+\alpha}(\Phi_q^T\lambda - Q)_n) = 0 \tag{6}$$

Here, α improves numerical stability. The HHT algorithm iteratively solves the nonlinear equations for the unknowns using a Newton-like method, which eliminates ill-conditioning issues typically associated with integrating index-3 DAEs.

B. Hydrodynamics theory in HydroChrono

In order to combine potential-flow-based hydrodynamics with Project Chrono, eq. (5) must be modified to include added mass on the left-hand side and hydrodynamic force functions (e.g., radiation damping force, wave excitation force) on the right-hand side. The constraint equations, which could include internal forces arising from joints, power take-offs, ropes, or other subsystems, do not need to be modified in order to incorporate hydrodynamics. Furthermore, the time-integration of the system's equations of motion is handled by Project Chrono, and does not need to be modified.

To include the added mass and hydrodynamic forces, we can modify eq. (5) as follows:

$$(M(q)+A)\ddot{q}+\Phi_q^T(q)\lambda = Q(\dot{q},q,t)+F_h(\dot{q},q,t) \tag{7}$$

Here, $A \in \mathbb{R}^{p\times p}$ represents the added mass matrix, and $F_h(\dot{q},q,t) \in \mathbb{R}^p$ denotes the hydrodynamic force vector acting on the generalized coordinates $q \in \mathbb{R}^p$.

To further elaborate on the hydrodynamic force vector $F_h(\dot{q},q,t)$, we can decompose it into three main components: hydrostatic force $F_{hs}(q)$, radiation damping force $F_{rad}(\dot{q},q)$, and wave excitation force $F_{exc}(t)$. This leads to the following equation:

$$F_h(\dot{q},q,t) = F_{hs}(q,t)+F_{rad}(\dot{q},t)+F_{exc}(t) \tag{8}$$

The following sections explain these components in more detail.

1) Hydrostatic force, $F_{hs}(q,t)$: This force represents the restoring force due to buoyancy and changes in the body's waterplane area. It is the product of the hydrostatic stiffness matrix K_h and the displacement vector Δq, which is the difference between the system's position q and its equilibrium position q_0:

$$F_{hs}(q,t) = K_h\Delta q = K_h(q-q_0) \tag{9}$$

The hydrostatic stiffness matrix K_h can be obtained by numerically integrating over the floating body's waterplane area. This can be performed with open-source boundary element method (BEM) codes such as Capytaine [5] or HAMS [6], or via the open-source mesh and hydrostatics package MeshMagick[1].

2) Radiation damping force, $F_{rad}(\dot{q},t)$: This force represents the energy dissipated as a floating body oscillates in water. It can be obtained through a convolution integral between the radiation impulse response function $K_{rad}(t)$ and the system's velocity history $\dot{q}(\tau)$:

$$F_{rad}(\dot{q},t) = \int_{-\infty}^{t} K_{rad}(t-\tau)\dot{q}(\tau)d\tau \tag{10}$$

$K_{rad}(t)$ is obtained by taking the inverse continuous cosine transform (a Fourier-related transform) of the frequency-domain radiation damping coefficients, $B(\omega)$:

$$K_{rad}(t) = \frac{2}{\pi}\int_0^{\infty} B(\omega)\cos(\omega t)d\omega \tag{11}$$

Hence, the radiation damping coefficients $B(\omega)$ are transformed from the frequency domain to the time domain to obtain the radiation impulse response function, $K_{rad}(t)$. $B(\omega)$ can be obtained with open-source BEM software.

3) Wave excitation force, $F_{exc}(t)$: There are two different approaches to computing the wave excitation force. The first is via convolution between the excitation impulse response function $K_{exc}(t)$ and the wave elevation time series $\eta(t)$:

$$F_{exc}(t) = \int_{-\infty}^{+\infty} K_{exc}(\tau)\eta(x,y,t-\tau)d\tau \tag{12}$$

The excitation impulse response function $K_{exc}(t)$ is computed from the frequency-domain excitation coefficients $F_{exc}(\omega)$ via inverse continuous cosine transform:

$$K_{exc}(t) = \frac{2}{\pi}\int_0^{\infty} F_{exc}(\omega)\cos(\omega t)d\omega \tag{13}$$

This equation transforms the excitation coefficients $F_{exc}(\omega)$ from the frequency domain to the time domain, resulting in the excitation impulse response function, $K_{exc}(t)$. $F_{exc}(\omega)$ can be obtained with open-source BEM software.

The term $\eta(x,y,t)$ in the equation represents the wave elevation at a given point (x,y) on the free surface at a given time (t). Wave elevation can be computed using different wave theories, depending on the complexity of the wave environment and the level of accuracy required.

For a simple linear wave theory, the irregular wave elevation $\eta(x,y,t)$ is calculated as the superposition of multiple wave components:

$$\eta(x,y,t) = \sum_i \frac{H_i}{2}\cos\left(\omega_i t - k_i(x\cos\theta_i + y\sin\theta_i)+\phi_i\right) \tag{14}$$

where H_i is the wave height for the i-th wave component, ω_i is the angular frequency, k_i is the wave number, θ_i is the angle of incidence, and ϕ_i is the phase angle. The summation runs over all the wave components, which together form the irregular wave field.

The irregular wave spectrum is defined to describe the distribution of wave energy across a range of frequencies. Two commonly used wave spectra are the Pierson-Moskowitz (PM) and the Joint North Sea Wave Project (JONSWAP) spectra.

The Pierson-Moskowitz spectrum is given by:

$$S_{PM}(\omega) = \frac{5}{16}\frac{\omega_p^4}{\omega^5}\exp\left(-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4\right) \tag{15}$$

where ω is the angular frequency and ω_p is the peak angular frequency, which is related to the significant wave height H_s and the peak period T_p.

The JONSWAP spectrum is a modification of the Pierson-Moskowitz spectrum, including a peak-shape

[1] https://github.com/LHEEA/meshmagick

parameter γ that accounts for the more peaked shape of the spectrum commonly observed in fetch-limited conditions or shallow water environments:

$$S_{JONSWAP}(\omega) = S_{PM}(\omega)\gamma^{\exp\left(-\frac{1}{2}\left(\frac{\omega-\omega_p}{\sigma\omega_p}\right)^2\right)} \tag{16}$$

where σ is a dimensionless parameter that depends on the frequency range:

$$\sigma = \begin{cases} 0.07 & \text{if } \omega \leq \omega_p \\ 0.09 & \text{if } \omega > \omega_p \end{cases} \tag{17}$$

Alternatively, the irregular excitation force can be computed as the real part of a sum over all wave frequencies in spectrum:

$$F_{exc}(t) = \Re\left[R_f(t)\sum_{j=1}^{N} F_{exc}(\omega_j,\theta)e^{i(\omega_j t+\phi_j)}\sqrt{2S(\omega_j)d\omega_j}\right] \tag{18}$$

Where N denotes the number of frequency bands chosen to discretize the wave spectrum.

III. Methods

A. HydroChrono Workflow

The overarching workflow of HydroChrono (Figure 2) shows how several different tools work together to model offshore systems.

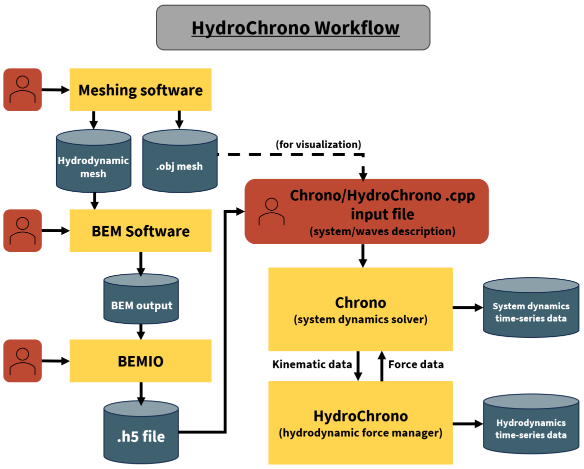

Fig. 2. Overview diagram of the HydroChrono workflow and the different tools used in the process. Red denotes user inputs, blue denotes data, and yellow denotes processes.

The first step is to create a mesh, which is passed to a BEM code (e.g. Capytaine, WAMIT) to compute hydrodynamic coefficients in the frequency domain. Because each BEM code has its own unique output format, BEMIO is used to convert the coefficients into a standard .h5 file format.

The system model is developed in the standard way using Chrono via the C++ API, which allows access to all of the capabilities of Chrono. HydroChrono is also invoked from this .cpp file; hence, when the model is built and the binary executed, Chrono and HydroChrono work together to pass kinematic and force data back and forth to simulate the system.

Since Chrono handles the system dynamics, this is where the system's output data (time-series data) is saved from. The hydrodynamics data (e.g., radiation damping force time series) is saved from HydroChrono.

Figure 2 does not show postprocessing because we have not yet developed a standard output format and the postprocessing scripts to accompany it. However, this is intended in our future work.

The organization of HydroChrono is shown in more detail in Figure 3.

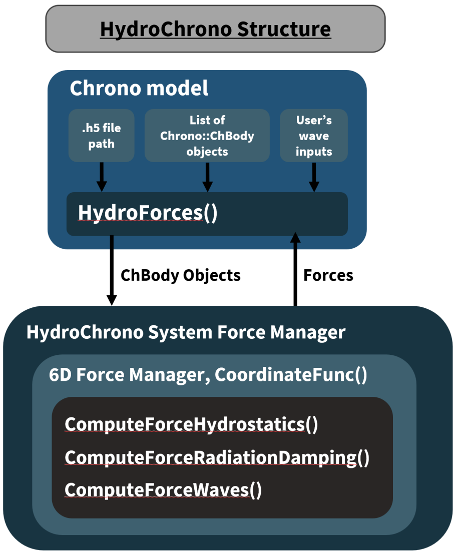
HydroChrono Structure
Chrono model
.h5 file path
List of Chrono::ChBody objects
User's wave inputs
HydroForces()
ChBody Objects
Forces
HydroChrono System Force Manager
6D Force Manager, CoordinateFunc()
ComputeForceHydrostatics()
ComputeForceRadiationDamping()
ComputeForceWaves()

Fig. 3. Diagram to show the high-level structure of HydroChrono.

The first block in Figure 3 shows the main entry point to HydroChrono, the `HydroForces` object, which is instantiated directly in the Chrono model .cpp input file using the standard Chrono C++ API.

Figure 3 also shows the main inputs that `HydroForces` requires: the .h5 file path, the list of `chrono::ChBody` objects in the system (rigid bodies for which the .h5 file contains hydrodynamic coefficients), and the user's wave input data, which could range from simply specifying still water to a JONSWAP wave spectrum.

Because `HydroForces` is invoked directly in the Chrono input file and only receives specific rigid bodies in the system, one can use the full range of Chrono's capabilities to model complex subsystems.

Within the HydroChrono force manager, data from each rigid body object is used to compute 6D force vectors, which are returned to Chrono to be included in the main solver. As shown in Figure 3, current force functions available include hydrostatics, radiation damping, and wave excitation.

Not included in the diagram is the added mass implementation in HydroChrono, which has been implemented via direct addition using a dense added mass matrix. This approach uses a predefined interface for adding a dense matrix to a Chrono system mass matrix called `chrono::ChLoadCustomMultiple`. Project Chrono handles the rest of the computations every time step internally.

This section provided a high-level overview of how the theory described in Section II is implemented in our software. The next section presents verification results for HydroChrono to provide confidence that the theory has been implemented correctly.

IV. HydroChrono Verification: Results and Analysis

This section presents the results of several different verification tests performed with different WEC models to provide confidence that the following functionality is working correctly:

- Input of hydrodynamic coefficients.
- Direct addition of a dense $6n_{body} \times 6n_{body}$ added mass matrix.
- Implementation of radiation damping force via convolution integral.
- Implementation of wave excitation forces for both regular and irregular waves.

A. *Verification Case 1: IEA OES Task 10 Sphere*

The International Energy Agency (IEA) Ocean Energy Systems (OES) Technology Collaboration Programme Task 10 sphere model has been used to compare many different numerical models [7] and serves as a simple but highly useful verification case to ensure that the fundamental linear hydrodynamics functions are working correctly.

1) Model overview: The sphere's hydrodynamic mesh (in .nemoh format), shown on the left-hand side of Figure 4, was used by the BEM code Capytaine to compute the hydrodynamic coefficients. The right-hand side of Figure 4 shows the complete sphere mesh (in .obj format) loaded in the Chrono real-time visualization window.

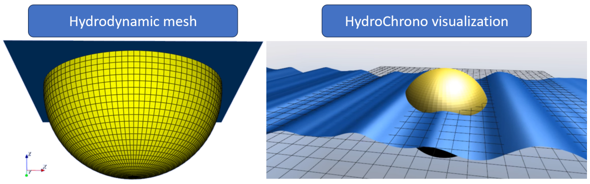

Fig. 4. Visualization of the sphere's hydrodynamic mesh and screenshot of the sphere in the Chrono GUI with irregular waves acting on the body.

The main properties of the sphere that have been used in this model are shown in Table I. Because the center of the sphere is on the waterline, the wetted surface (and therefore the mesh used by Capytaine) is a hemisphere, as shown in Figure 4. In the Chrono visualization, a mesh of the complete sphere has been used.

Figures 5 and 6 show the verification of the sphere decay test and response amplitude operator (RAO) results obtained with HydroChrono versus some of the other participants of the IEA OES Task 10 project.

From Figures 5 and 6, we can observe very good agreement with the other numerical models, building confidence in our implementation of added mass, radiation damping force (by convolution integral), regular wave excitation force, and hydrostatic force.

TABLE I
General Properties of the Heaving Sphere

Parameters	Assigned Values	Unit
Sphere radius	5	m
Sphere center	$\begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$	m
Sphere center of gravity	$\begin{bmatrix} 0.0 \\ 0.0 \\ -2.0 \end{bmatrix}$	m
Sphere mass	261.8×10^3	kg
Water depth	∞	m
Water density	1000	$\mathrm{kg/m^3}$

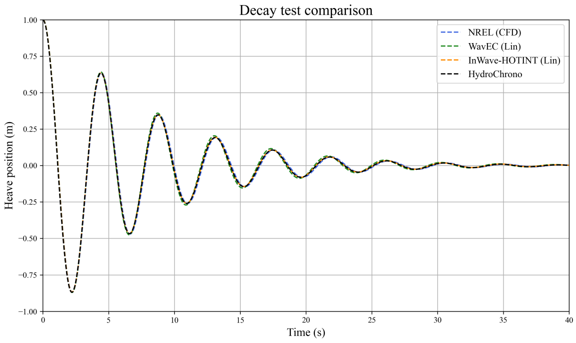

Fig. 5. Verification results for hydrodynamic simulations of a heaving sphere 1 m decay test.

Figure 7 shows the time-series comparison for the sphere heaving in irregular waves—HydroChrono versus WEC-Sim. In order to do this comparison, the same wave elevation time series was loaded into both programs, and the excitation forces were computed by convolution of the wave elevation and excitation impulse response functions. The visualization of the simulation in the Chrono GUI is shown in Figure 4.

Again, excellent agreement is observed with WEC-Sim, providing confidence in our implementation of the wave excitation convolution integral function. However, the IEA OES Task 10 model only considers the sphere moving in the heave degree of freedom. To verify our implementation for other degrees of freedom, we need to consider a range of different systems.

B. *Verification Case 2: The oscillating surge wave energy converter*

Many variations of the oscillating surge wave energy converter (OSWEC) have been modeled over the years. In this verification case, we focus on the OSWEC example model that it is included in the WEC-Sim package and perform our verification directly against WEC-Sim results.

1) Model overview: The OSWEC flap hydrodynamic mesh (in .nemoh format), shown on the left-hand side of Figure 8, was used by the BEM code Capytaine to compute the hydrodynamic coefficients. The right-hand side of Figure 8 shows the complete OSWEC

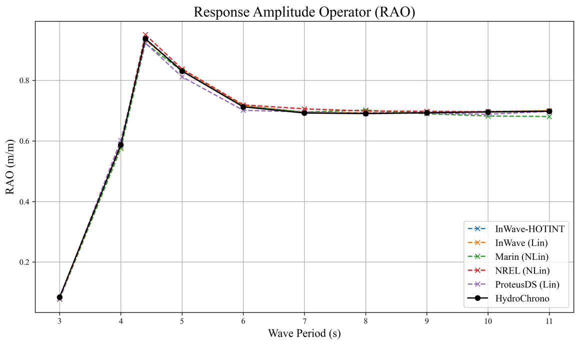

Fig. 6. Verification results for hydrodynamic simulations of a heaving sphere in regular waves.

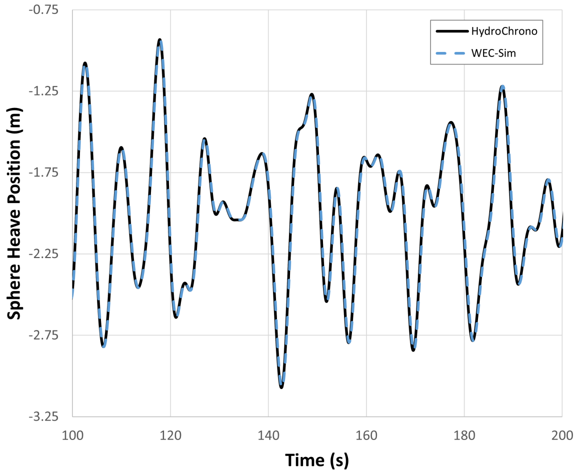

Fig. 7. Verification results for hydrodynamic simulations of a heaving sphere in irregular waves.

system (flap and base meshes in .obj format) loaded in the Chrono real-time visualization GUI window.

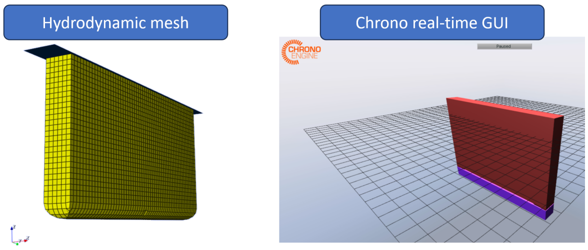

Fig. 8. Visualization of the OSWEC's hydrodynamic mesh and screenshot of the OSWEC in the Chrono GUI.

The main properties of the OSWEC system are summarized in Table II.

Figure 9 shows excellent agreement between HydroChrono and WEC-Sim for the OSWEC 10 degree pitch decay test. This builds confidence in the implementation of added mass, radiation damping (by convolution integral), and hydrostatic stiffness forces.

Figure 10 generally shows good agreement between the WEC-Sim and HydroChrono RAOs, but there is a noticeable difference at the peak of the RAO, with HydroChrono's response being slightly smaller. This could be due to small errors in the interpolation of hydrodynamic coefficients, which could be more noticeable near the peak. It could also be due to small errors in the numerical solvers used in each model (Chrono and Simscape Multibody), which become more noticeable at the resonance frequency.

TABLE II
General Properties of the OSWEC model.

Parameters	Assigned Values	Unit
Hinge location	$\begin{bmatrix} 0.0 \\ 0.0 \\ -8.9 \end{bmatrix}$	m
Flap center of gravity	$\begin{bmatrix} 0.0 \\ 0.0 \\ -3.9 \end{bmatrix}$	m
Flap mass	127×10^3	kg
Flap pitch inertia (I_{yy}	1.85×10^6	$\text{kg} \cdot \text{m}^2$
Water depth	10.9	m
Water density	1000	kg/m^3

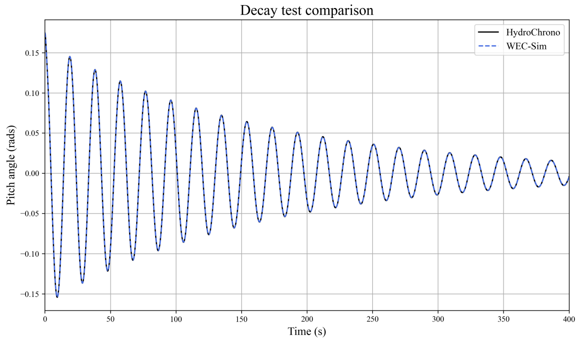

Fig. 9. Verification results for hydrodynamic simulations of a 10° OSWEC pitch decay test.

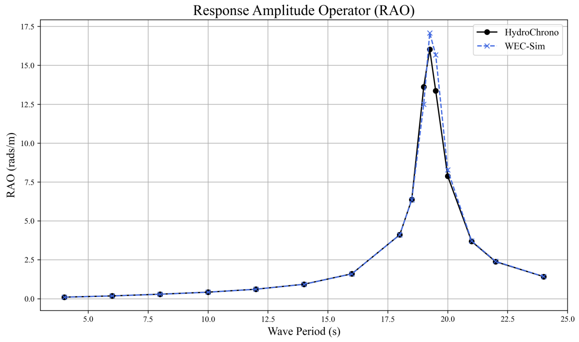

Fig. 10. Verification results for hydrodynamic simulations of the OSWEC model in regular waves.

The OSWEC results build confidence in our implementation of hydrodynamic forces for rotational degrees of freedom and demonstrate that our implementation can handle multibody systems because the base is treated as a hydrodynamic body in this simulation. However, because the base is stationary, we cannot really evaluate how well the code is handling hydrodynamic interactions from the OSWEC model alone.

C. *Verification Case 3: F3OF*

The F3OF model was first developed by Babarit et al. for the NumWEC project [8] and has since been used in other verification projects such as the WEC^3 code-to-code comparison project [9].

1) Model overview: The visualization of the F3OF system in the Chrono GUI is shown in Figure 11, which shows the fore flaps slightly rotated at the beginning of a flap decay test.

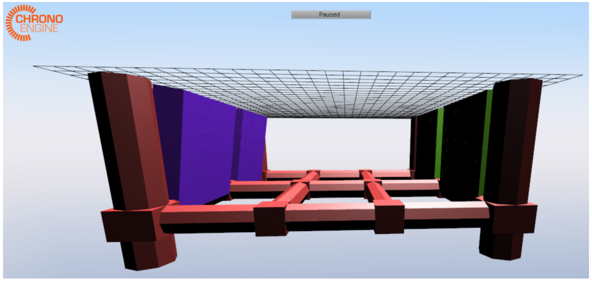

Fig. 11. Screenshot of the F3OF in the Chrono GUI.

The main properties of the F3OF system are summarized in Table III.

TABLE III
GENERAL PROPERTIES OF THE F3OF MODEL.

Parameters	Assigned Values	Unit
Base center of gravity	$\begin{bmatrix}0.0\\0.0\\-9.0\end{bmatrix}$	m
Base mass	1089825.0	kg
Base pitch inertia (I_{yy})	76.3×10^6	kg · m^2
Flap center of gravity	$\begin{bmatrix}\pm 12.5\\0.0\\-5.5\end{bmatrix}$	m
Flap mass	179250.0	kg
Flap pitch inertia (I_{yy})	1.3×10^6	kg · m^2
Water depth	200	m
Water density	1025	kg/m^3

The flap decay test helps to check if the hydrodynamic interactions are being computed properly—as the fore flap starts oscillating, it radiates waves that excite the aft flap. The verification results are shown in Figure 12.

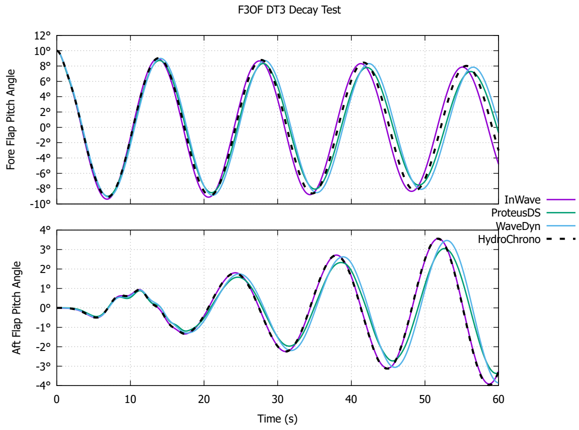

Fig. 12. F3OF flap decay test verification results.

Figure 12 shows reasonable agreement with the other numerical models, particularly in the first 20 seconds where all of the codes predict a higher frequency response in the aft flap motion, within the envelope frequency (in other words, a "wobble"). However, the results start to drift apart over time due to slight differences in natural frequency and decay rate. It is not clear which results we should have the most confidence in; ideally, the F3OF model should be simulated with computational fluid dynamics in order to provide some guidance. High-fidelity modeling of the F3OF system is potential future work.

V. HydroChrono Application Case: Floating Offshore Wind Turbine

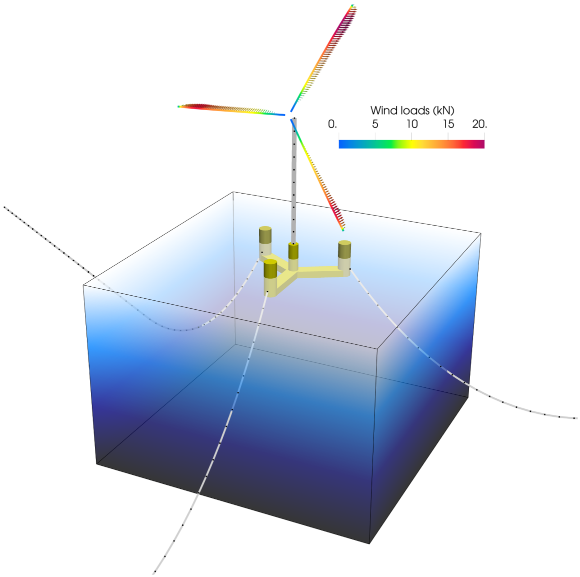

Fig. 13. Visualization of FOWT simulation in SEAHOWL with HydroChrono floater.

A. Model Overview

HydroChrono has been coupled to SEAHOWL (Servo-Elasto-Aero-Hydro Offshore Wind Lab), a multiphysics multifidelity framework developed at TotalEnergies for the numerical simulation of wind energy converters. This enables the simulation of FOWTs with the turbine dynamics coming from SEAHOWL and the time-domain potential flow coming from HydroChrono. Since SEAHOWL also uses Project Chrono, all structural dynamics are monolithically coupled, including HydroChrono bodies. Such coupling ensures superior numerical stability and accuracy when compared to explicit coupling and does not require computationally costly iterations found in implicit coupling.

For introducing the SEAHOWL-HydroChrono coupling through the simulation of a FOWT under regular

TABLE IV
GENERAL PROPERTIES OF THE FOWT

Parameters	Assigned Values	Unit
FOWT mass (floater + turbine)	20.252×10^6	kg
Floater hull displacement	$20,206$	m^3
Initial draft	20	m
Center of gravity	$\begin{bmatrix}0.0\\0.0\\-14.94\end{bmatrix}$	m
Inertia tensor diagonal	$\begin{bmatrix}1.251 \times 10^{10}\\1.251 \times 10^{10}\\2.367 \times 10^{10}\end{bmatrix}$	kg · m^2
Water depth	200	m
Water density	1025	kg/m^3

TABLE V
MOTIONS AND FAIRLEAD TENSIONS OF THE FOWT PLATFORM UNDER REGULAR WAVES AND CONSTANT WIND SPEED.

	Surge (m)		Sway (m)		Heave (m)		Roll (°)		Pitch (°)		Yaw (°)		T1 (MN)		T2 (MN)		T3 (MN)	
	M	A	M	A	M	A	M	A	M	A	M	A	M	A	M	A	M	A
SEAHOWL	17.806	0.231	0.174	0.000	-0.443	0.098	0.018	0.002	3.073	0.142	0.574	0.000	3.852	0.068	2.050	0.032	2.040	0.031
OpenFAST	17.946	0.233	0.022	0.000	-0.465	0.098	0.392	0.001	2.981	0.147	0.222	0.000	3.775	0.058	2.061	0.030	2.051	0.031

M: mean; A: amplitude; T1, T2, T3: tensions at fairlead 1, 2, and 3, respectively.

wave and constant wind loads, the following physics are selected:

- Servodynamics: blade pitch and electrical torque control using the ROSCO controller as described in [10] with floating feedback enabled based on pitch motion of the nacelle.
- Elastodynamics: structural part for rigid bodies, joints, and finite elements for flexible components modeled with Project Chrono.
- Aerodynamics: blade element momentum theory for calculating loads on the blades, and Morison equation for loads on the tower.
- Hydrodynamics: time-domain potential flow with HydroChrono for calculating loads on the floater, and Morison equation for loads on the moorings.

The numerical FOWT model studied here combines the IEA 15 MW offshore reference wind turbine [11] with the University of Maine VolturnUS-S reference platform [12]. Basic properties are compiled in Table IV, and the mooring system is kept the same as in [12].

The elasto model of the turbine is entirely built using Project Chrono: the blades and tower are described with beam elements (50 elements per blade), and the rotor-nacelle assembly as rigid bodies (nacelle, shaft, hub, yaw bearing mass) and joints (revolute joint for hub shaft). The floater is represented as a rigid body (point mass and associated inertia matrix) with an additional damping matrix to account for viscous effects, and hydrodynamic loads in the time domain are computed using HydroChrono. The mooring system is composed of three catenary chains and is discretized using absolute nodal coordinate formulation (ANCF) cable elements, similar to work presented in [13], [14] with 50 elements per mooring line. All values used for building the numerical model can be found in the reference reports cited above, with a notable change here of the tower structural properties. This is done to avoid a resonance issue with the 3P frequency since the first natural frequencies of the original floating tower design were found to be closer to 0.4 Hz rather than the intended 0.5 Hz. This structural change leads to a total tower mass of 1,466 tons (against 1,263 tons originally) for a natural frequency of 0.49 Hz. A visualization of the complete FOWT system is presented in Figure 13.

B. Simulation results

For the fully coupled FOWT simulation presented here, environmental conditions are regular waves with height of 1.916 m and period of 7.5 s in a water depth of 200 m and a constant wind speed of 12 m/s (above rated wind speed for the IEA 15 MW turbine) at a reference height of 150 m with a shear coefficient of 0.12.

The results of the SEAHOWL-HydroChrono coupling are compared to results of the same FOWT implemented in OpenFAST (multiphysics wind turbine simulator developed by the National Renewable Energy Laboratory) that uses its own hydrodynamics solver (potential flow with HydroDyn), aerodynamics solver (blade element momentum theory with AeroDyn), elastodynamics solver (modal dynamics for tower and blades with ElastoDyn), moorings solver (lumped mass model with MoorDyn), and ROSCO for the servodynamics. All results are computed using time steps of 0.025 s, and analyzed over a duration equivalent to the last 300 wave periods (well after the initial transient part of the simulation). Results of all degrees of freedom as well as the fairlead tensions of the floating platform are presented in table V, where the amplitude value is taken at the frequency with the highest amplitude (in this case, corresponding to the regular wave frequency of 7.5 s for all variables recorded). There is close agreement between the two codes on most values, especially on most excited degrees of freedom such as surge, heave, and pitch. One of the most notable differences is the amplitude of the tension signal of the fairlead 1 (attached to the most solicited line with its anchor directly upwind) with an overestimation of 10 kN for SEAHOWL when compared to OpenFAST (17% difference). Differences can be expected when comparing results of a case involving many different types of physics coupled together and using different solvers. It has been observed several times in [15] where three different codes (HAWC2, DeepLines Wind, and DIEGO) were compared for the response of a similar FOWT. For example, with an above-rated wind speed of 15.53 m/s, [15] found that the mean surge could have a difference of up to 8.5% from code to code, and standard variation of tower base forces and moments 20% and 30%, respectively. This type of discrepancy can be expected when comparing multiphysics codes that use their own

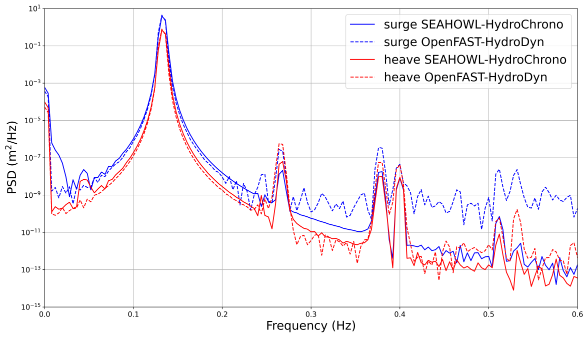

Fig. 14. Surge and heave response of the FOWT platform under regular waves and constant wind speed.

solvers and different numerical approaches. For the heave, both SEAHOWL and OpenFAST give a drop of -0.45 m when compared to the expected initial position of the floater as given by [12], which is expected because the new tower design is heavier than the original tower design. In Figure 14, the power spectral density of the heave motion shows good agreement between the two codes, both in terms of location and amplitude of the lower-frequency peaks. Overall, the response of the FOWT platform between SEAHOWL-HydroChrono and OpenFAST-HydroDyn is very similar.

VI. Comparison of Computation Times

Many factors, including the chosen time-step size, numerical method, compiler options, among others, can affect computational time. Hence, its very important to conduct profiling and optimization of software to ensure optimal computational performance.

In this project, our primary focus has been on expanding HydroChrono's functionality and verifying its accuracy. Speed optimization is yet to be addressed. Despite this, it is useful to benchmark it against existing tools. Therefore, in this section we present some computational times to provide preliminary insights into how HydroChrono fares in comparison to WEC-Sim, as demonstrated in Table VI. To minimize the number of variables impacting speed, we used the simple sphere decay model. The computation time in this case is largely determined by the convolution integral employed in calculating the radiation damping forces.

TABLE VI
40s Sphere Decay Computation Time Comparison

Parameter	HydroChrono (s)	WEC-Sim (s)	Speed-up
Run 1	0.633	39.042	61.7x
Run 2	0.615	5.898	9.6x
Run 3	0.631	5.787	9.2x
Average	0.626	16.909	27.0x

As evidenced by Table VI, HydroChrono markedly outperforms WEC-Sim, with speedups of up to 61.7x. We attribute this primarily to the computational efficiency of compiled C++ code in the calculation of the convolution integral, as opposed to just-in-time compiled MATLAB code. WEC-Sim's performance is notably slower during the first run, followed by enhanced performance in the subsequent runs. We suspect that this is likely due to MATLAB caching various libraries and the model itself, which helps to boost performance in the second and third runs.

The higher computational expense of WEC-Sim's first run could be an important factor to consider in optimization studies (as it may not be possible to capitalize on caching as much if the model is changing between each run, or a new instance of MATLAB is launched). However, its also important to note that this comparison does not include HydroChrono's compilation time.

To better understand the performance of HydroChrono, the computation time for various models and settings needs to be explored and profiled - in order to provide guidance for speed optimization in the future.

VII. Discussion, Conclusions and Future Work

In summary, work has begun on the development of HydroChrono, a C++ hydrodynamics package designed to work with the open-source physics engine Project Chrono. In addition to overcoming current limitations with WEC-Sim (e.g., licensing restrictions, speed limitations, and interoperability), it is hoped that this platform can offer more advanced capabilities to WEC numerical modelers in the future, including structural analysis via the Chrono::FEA module, soil-anchor interactions using the discrete element method, collisions, advanced control algorithms, machine learning, and multifidelity modeling.

In the paper, we described the fundamental theory of HydroChrono and provided verification results for a range of different WEC systems to build confidence in our implementation. The results of HydroChrono agree very well with established reference cases.

Furthermore, HydroChrono has been designed to integrate into other code bases. As such, it has been successfully coupled to the multiphysics SEAHOWL framework for modeling the response of a FOWT to wind and wave loads.

Future work to extend HydroChrono capabilities includes coupling to MoorDyn, nonlinear hydrodynamics, quadratic transfer functions for second-order wave loads in potential flow, and strip theory formulation for submerged bodies with loads applied via Morison's equation, a hybrid approach combining potential flow and strip theory.

References

[1] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *High Performance Computing in Science and Engineering*, T. Kozubek, R. Blaheta, J. Šístek, M. Rozložník, and M. Čermák, Eds. Cham: Springer International Publishing, 2016, pp. 19–49.

[2] Z. Wei, B. L. Edge, R. A. Dalrymple, and A. Hérault, "Modeling of wave energy converters by gpusph and project chrono," *Ocean Engineering*, vol. 183, pp. 332–349, 2019.

[3] B. Tagliafierro *et al.*, "Numerical modelling of a point-absorbing wec model using dualsphysics coupled with a multiphysics library." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2019.

[4] I. Martínez-Estévez, J. El Rahi, J. Domínguez, B. Tagliafierro, V. Stratigaki, A. Crespo, P. Troch, T. Suzuki, and M. Gómez-Gesteira, "Coupling between dualsphysics and the finite element module of project chrono: multiphysics modelling of waves-wec interaction," in *General Assembly*, 2020, pp. 11–11.

[5] M. Ancellin and F. Dias, "Capytaine: a python-based linear potential flow solver," *Journal of Open Source Software*, vol. 4, no. 36, p. 1341, 2019. [Online]. Available: https://doi.org/10.21105/joss.01341

[6] Y. Liu, "Hams: A frequency-domain preprocessor for wave-structure interactions—theory, development, and application," *Journal of Marine Science and Engineering*, vol. 7, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2077-1312/7/3/81

[7] F. F. Wendt *et al.*, "International energy agency ocean energy systems task 10 wave energy converter modeling verification and validation: Preprint," vol. 2, no. 1, 10 2017. [Online]. Available: https://www.osti.gov/biblio/1401957

[8] A. Babarit, J. Hals, M. J. Muliawan, A. Kurniawan, T. Moan, and J. Krokstad, “Numerical benchmarking study of a selection of wave energy converters,” *Renewable Energy*, vol. 41, pp. 44–63, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148111005672

[9] A. Combourieu, M. Lawson, A. Babarit, K. Ruehl, A. Roy, R. Costello, P. Laporte Weywada, and H. Bailey, “Wec3: Wave energy converter code comparison project: Preprint,” vol. 1, no. 2, 1 2017. [Online]. Available: https://www.osti.gov/biblio/1339352

[10] N. J. Abbas, D. S. Zalkind, L. Pao, and A. Wright, “A reference open-source controller for fixed and floating offshore wind turbines,” *Wind Energy Science*, vol. 7, no. 1, pp. 53–73, 2022. [Online]. Available: https://wes.copernicus.org/articles/7/53/2022/

[11] E. Gaertner *et al.*, “Definition of the IEA wind 15-megawatt offshore reference wind turbine.”

[12] C. Allen, A. Viscelli, H. Dagher, A. Goupee, E. Gaertner, N. Abbas, M. Hall, and G. Barter, “Definition of the UMaine VolturnUS-S reference platform developed for the IEA wind 15-megawatt offshore reference wind turbine,” pp. NREL/TP–5000–76 773, 1 660 012, MainId:9434, 2020. [Online]. Available: https://www.osti.gov/servlets/purl/1660012/

[13] T. de Lataillade, A. Dimakopoulos, C. Kees, L. Johanning, D. Ingram, and T. Tezdogan, “CFD modelling coupled with floating structures and mooring dynamics for offshore renewable energy devices using the proteus simulation toolkit,” *European Wave and Tidal Energy Conference*, 2017.

[14] T. de Lataillade, A. Dimakopoulos, C. Kees, and L. Johanning, “A coupling strategy for modelling dynamics of moored floating structures,” in *Advanced Numerical Modelling of Wave Structure Interactions*. CRC Press, 2021, pp. 203–247.

[15] T. Kim, A. Natarajan, A. Lovera, E. Julan, C. Peyrard, M. Capaldo, G. Huwart, P. Bozonnet, and M. Guiton, “A comprehensive code-to-code comparison study with the modified IEA15MW-UMaine floating wind turbine for H2020 HIPERWIND project,” *Journal of Physics: Conference Series*, vol. 2265, no. 4, p. 042006, 2022. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/2265/4/042006