



HAL
open science

Development and validation of a numerical wave tank based on the Harmonic Polynomial Cell and Immersed Boundary methods to model nonlinear wave-structure interaction

Fabien Robaux, Michel Benoit

► To cite this version:

Fabien Robaux, Michel Benoit. Development and validation of a numerical wave tank based on the Harmonic Polynomial Cell and Immersed Boundary methods to model nonlinear wave-structure interaction. *Journal of Computational Physics*, 2021, 446, pp.110560. 10.1016/j.jcp.2021.110560 . hal-03327679

HAL Id: hal-03327679

<https://hal.science/hal-03327679>

Submitted on 27 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Development and validation of a numerical wave tank based on the Harmonic Polynomial Cell and Immersed Boundary methods to model nonlinear wave-structure interaction

Fabien Robaux¹ and Michel Benoit^{1,2}

¹roboux@irphe.univ-mrs.fr, Aix Marseille Univ, CNRS, Centrale Marseille, Institut de Recherche sur les Phénomènes Hors-Equilibre (IRPHE), Marseille, France
²benoit@irphe.univ-mrs.fr, Centrale Marseille, Marseille, France

August 27, 2021

Accepted manuscript for publication, [Robaux and Benoit \(2021\)](#), *Journal of Computational Physics*, ISSN 0021-9991.

Abstract

A fully nonlinear potential Numerical Wave Tank (NWT) is developed in two dimensions, using a combination of the Harmonic Polynomial Cell (HPC) method for solving the Laplace problem on the wave potential and the Immersed Boundary Method (IBM) for capturing the free surface motion. This NWT can consider fixed, submerged or wall-sided surface piercing, bodies. To compute the flow around the body and associated pressure field, a novel multi overlapping grid method is implemented. Each grid having its own free surface, a two-way communication is ensured between the problem in the body vicinity and the larger scale wave propagation problem. Pressure field and nonlinear loads on the structure are computed by solving a boundary value problem on the time derivative of the potential. The stability and convergence properties of the solver are studied basing on extensive tests with standing waves of large to extreme wave steepness, up to $H/\lambda = 0.2$ (H is the crest-to-trough wave height and λ the wavelength). Ranges of optimal time and spatial discretizations are determined and high-order convergence properties are verified, first without using any filter. For cases with either high level of nonlinearity or long simulation duration, the use of mild Savitzky-Golay filters is shown to extend the range of applicability of the model. Then, the NWT is tested against two wave flume experiments, analyzing forces on bodies in various wave conditions. First, nonlinear components of the vertical force acting on a small horizontal circular cylinder with low submergence below the mean water level are shown to be accurately simulated up to the third order in wave steepness. The second case is a dedicated experiment with a floating barge of rectangular cross-section. This very challenging case (body with sharp corners in large waves) allows to examine the behavior of the model in situations at and beyond the limits of its formal application domain. Though effects associated with viscosity and flow separation manifest experimentally, the NWT proves able to capture the main features of the wave-structure interaction and associated loads.

keywords: Fully nonlinear waves, Numerical Wave Tank , Harmonic Polynomial Cell, Boundary Value Problem, Immersed Boundary Method, Immersed overlapping grids

1 Introduction and scope of the study

To model the propagation of oceanic and coastal water waves of high steepness and their interaction with offshore structures fast and accurate numerical models are needed by scientists and engineers. Questions related to an accurate estimation of wave loads and structure dynamics in moderate to severe sea state conditions are of utmost importance for optimizing the design of such structures at sea and, for instance, assessing their behavior in extreme storm conditions and/or for fatigue analysis in the life-cycle of the structure. A wide range of models have been developed to predict wave fields and hydrodynamic loads at any scale, from the simple linear potential boundary element method to complex Computational Fluid Dynamics (CFD) codes directly solving the Navier-Stokes equations.

In the last decade, the use of CFD code has become increasingly popular in the scientific community, see for instance the applications of industrial or research codes like OpenFOAM® (Jacobsen et al., 2011; Hu et al., 2016; Higuera et al., 2013; Windt et al., 2019), STAR-CCM+ (Oggiano et al., 2017; Jiao and Huang, 2020), ANSYS-FLUENT (Kim et al., 2016; Feng and Wu, 2019), and REEF3D (Bihs et al., 2016), just to mention few of them.

Although these CFD models are well adapted to solve the wave-structure interaction at small scale, in particular when complex physical processes such as wave breaking, formation of jets, air entrapment occur, their use is still mostly restricted to a limited number of cases by the computational cost when targeting applications on ocean domains where the accurate representation of wave propagation is of first importance. Limitations associated with the employed numerical methods (*e.g.* numerical diffusion and difficulty to resolve the dynamics of the free surface) are other reasons which still hinder the applicability of such models to large scale wave propagation problems. Their performance can be greatly increased by carefully selecting the different numerical schemes and parameters, to reduce the computational cost at the expense of some accuracy (Larsen et al., 2019).

On the other hand, models based on a potential flow approach (*i.e.* neglecting viscous effects and assuming irrotational flow) are widely used to describe the dynamics of wave-structure interaction flows (see *e.g.* the review by Tanizawa, 2000). Among potential models, simplified linear versions are often used in the engineering field due to their low computational cost, allowing to capture the main effects on a wide range of parameters at a contained CPU cost. WAMIT (Lee, 1995), ANSYS-AQWA (Ansys, 2013) or Nemoh (Fàbregas Flavià et al., 2016) are examples of such linear models. However, the linear assumption is often used outside its prescribed validity domain, for example in extreme cases where the allegedly small parameters, usually the wave steepness, becomes large. In those conditions many aspects of the dynamics of both the incident waves and the wave-body interaction are not properly modeled.

Efforts have been done to extend these potential linear models to weakly nonlinear conditions at second order, mostly by adding terms like the Froude-Krylov forces or the complete Quadratic Transfer Functions, see *e.g.* Pinkster (1980) or more recently Philippe et al. (2015). However in some cases, mostly extreme wave conditions, even third order effects can have a significant effect on the wave load (see *e.g.* Fedele et al., 2017). Both the nonlinear wave dynamics and the nonlinear wave-structure interaction need to be taken into account. In those cases, alternative approaches capturing higher-order or fully nonlinear effects are needed. Several developments have been made to compute such nonlinear effect exactly or at high orders in the case of periodic regular waves in uniform water depth. For example, models based on analytical theories such as the Stokes wave theory (see *e.g.* Sobey, 1989) or the so-called stream function method (Dean, 1965; Fenton, 1988) can only be applied with constant or simple geometry of the sea floor. A review of methods to describe wave propagation in a potential flow framework is given by Fenton (1999).

High order spectral (HOS) methods are also fast and accurate to compute the flow behavior even up to large wave steepness. Such methods, though very efficient in computing the wave elevation even for large domains, might be limited in terms of geometry (complex bodies or variable bathymetry).

They are mostly applied for wave maker modeling (Ducrozet et al., 2012), or to compute the incident wave field within a more complex method to resolve the wave-structure interactions. For example, in the SWENSE method (see *e.g.* Luquet et al., 2007), the diffracted field is computed separately with a Navier-Stokes based solver.

In order to develop a versatile Numerical Wave Tank (NWT), with the possibility to include bodies or sea bed with complex geometries, a time domain resolution involving a mesh in the spatial coordinates appears to be practical at the cost of an increase in the computational load. The commonly used Boundary Element Method (BEM), in which the Laplace equation is projected onto the spatially discretized boundaries using the Green's identities, has been proven to be effective in both 2D and 3D cases. For example, Grilli and Horrillo (1997) used a high-order BEM method to generate and absorb waves in 2D. This model was extended to 3D by Grilli et al. (2001). For an overview of work on the BEM methods up to the end of the 20th century, the reader is referred to Kim et al. (1999). More recently, Guerber et al. (2012) and Dombre et al. (2019) presented in great detail and implemented a complete NWT, in 2D and 3D respectively. Note that with the BEM schemes, a special attention must be given to the treatment of sharp corners (Hague and Swan, 2009).

Another type of time domain wave simulators is volume field solvers. With the cost of increasing the number of unknowns, the resulting matrix is mostly sparse, allowing the use of efficient solvers. Both potential and Navier-Stokes solvers can be found in this category. Notable works on NWT based on the finite difference method (FDM) are given by Tavassoli and Kim (2001), or Engsig-Karup et al. (2009) (OceanWave3D code). Finite element methods (FEM) were also successfully applied to both the potential problem (*e.g.* Ma and Yan, 2006; Yan and Ma, 2007; Engsig-Karup et al., 2016, 2019) and the Navier-Stokes equations (*e.g.* Wu et al., 2013).

A different potential model that solves for the volume field was recently proposed by Shao and Faltinsen (2012, 2014a,b) and tested against several methods including the BEM, FDM and FEM. This innovative technique, called the "harmonic polynomial cell" (HPC) method, was proven to be promising both in 2D and 3D (Shao and Faltinsen, 2014a; Hanssen et al., 2015, 2017a). Although relatively new, this method was used to study a relatively important range of flows and phenomena: from a closed flexible fish cage (Strand and Faltinsen, 2019) to hydrodynamics lifting problems (Liang et al., 2015). It was also extended to solve the Poisson equation by Bardazzi et al. (2015). The numerical aspects of the method were studied in details by Ma et al. (2017) and applied as a 2D NWT by Zhu et al. (2017).

The treatment of the free surface conditions and body boundary condition can be done in several ways. A classical and straightforward approach is to use a grid that conforms the boundary shape. With this method, the boundary node values are explicitly enforced in the linear system. The drawback of this method is that the grid needs to be deformed at each time step so as to match the free surface. A lot of solutions have been successfully applied to tackle this issue. For instance, Ma and Yan (2006) used a Quasi Arbitrary Lagrangian-Eulerian (ALE) method combined with the FEM spatial discretization to prevent the mesh to have to be regenerated at each time step. Yan and Ma (2007) extended the mesh conformation technique to include a freely floating body. With this same FEM discretization, Wu et al. (2013) used in addition an hybrid Cartesian immersed boundary method for the body boundary condition.

In this work, a NWT based on the HPC method is developed, its convergence assessed, and tested against both numerical and experimental data. Relaxation zones are used to generate and absorb waves in a similar manner as in the OpenFOAM® toolbox waveFoam (Jacobsen et al., 2011): the values and positions of the free surface nodes are imposed from a stream-function theory over a given distance. The free surface is tracked in a semi-Lagrangian way following Hanssen et al. (2017a) whereas for the solid bodies, an additional grid fitted to the boundary is defined using the advances presented in Ma et al. (2017). In order for the body to pierce the free surface, an additional free

surface marker list is defined in the body fitted grid. "External" and "internal" curves – *i.e.* free surface boundaries evolving in the background and body fitted grids respectively – also overlap each other and communicate through relaxation zones. In order to tackle the singular nodes, a null flow flux method is imposed on the sharp corners which lay on a Neumann body boundary condition in a similar manner as in [Zhu et al. \(2017\)](#), but applied at external body corners (described in § 5.4).

The remainder of this article is organized as follows. In § 2 the mathematical formulation of the potential flow problem is recalled. The numerical methods based on the HPC approach are presented in § 3, with particular attention devoted to the treatment of the free surface. In § 4, a convergence study is performed on a freely evolving standing wave case compared to a highly accurate numerical solution. Numerous tests are performed to determine the optimal ranges of discretization parameters, first without using any filter. The effects associated with mild filters are then analyzed. The case of a standing wave of near maximum amplitude is run to confirm the nonlinear capabilities of the model. Then, a novel fitted mesh overlapping grid method is described and implemented in § 5. This double mesh strategy is tested on two selected cases in § 6. The first one is an horizontal fixed circular cylinder, completely immersed although close to the free surface, from [Chaplin \(1984\)](#) and the second one involves a free surface piercing rectangular barge. That second experiment is an original campaign done during this study. Results of the present NWT on the two test cases are compared with experimental data and numerical results from the literature. In § 7, the main findings from this study and research outlook are summarized and discussed.

2 Fully nonlinear potential flow modeling approach

Three main assumptions are used in the potential model: i. we consider a fluid of constant and homogeneous density ρ (incompressible flow); ii. the flow is assumed to be irrotational, implying that $\nabla \times \mathbf{v} = 0$, where $\mathbf{v}(x, y, z, t)$ denotes the velocity field; and iii. viscous effects are neglected (ideal fluid). Here, (x, y) denote horizontal coordinates, z the vertical coordinate on a vertical axis pointing upwards, and t the time. The gradient operator is defined as $\nabla f \equiv (f_x, f_y, f_z)^T$, where subscripts denote partial derivatives (*e.g.* $f_x = \frac{\partial f}{\partial x}$).

The complete description of the velocity field \mathbf{v} can thus be reduced to the knowledge of the potential scalar field $\phi(x, y, z, t)$, such that: $\mathbf{v} = \nabla \phi$. Due to the incompressibility of the flow, the potential ϕ satisfies the Laplace equation inside the fluid domain:

$$\nabla^2 \phi = 0, \quad -h(x, y) \leq z \leq \eta(x, y, t) \quad (1)$$

where $\eta(x, y, t)$ is the free surface elevation and $h(x, y)$ the water depth relative to the still water level (SWL). In order to solve this equation, boundary conditions need to be considered. On the time varying free surface $z = \eta(x, y, t)$, the Kinematic Free Surface Boundary Condition (KFSBC) and the Dynamic Free Surface Boundary Condition (DFSBC) apply:

$$\eta_t + \nabla_H \eta \cdot \nabla_H \phi - \phi_z = 0 \quad \text{on } z = \eta(x, y, t), \quad (2)$$

$$\phi_t + \frac{1}{2} (\nabla_H \phi)^2 + g\eta = 0 \quad \text{on } z = \eta(x, y, t), \quad (3)$$

where $\nabla_H f \equiv (f_x, f_y)^T$ denotes the horizontal gradient operator and g the acceleration due to gravity. At the bottom (impermeable and fixed in time), the Bottom Boundary Condition (BBC) reads:

$$\nabla_H h \cdot \nabla_H \phi + \phi_z = 0 \quad \text{on } z = -h(x, y). \quad (4)$$

On the body surface, the slip boundary condition expresses that the velocity component of the flow normal to the body face equals the normal component of the body velocity. Here, we restrict our attention to fixed bodies, thus, denoting \mathbf{n} the unit vector normal to the body boundary, this condition reduces to:

$$\frac{\partial \phi}{\partial \mathbf{n}} = \nabla \phi \cdot \mathbf{n} = 0 \quad \text{on the body.} \quad (5)$$

Note that, using the free surface velocity potential and the vertical component of the velocity at the free surface, defined respectively as:

$$\tilde{\phi}(x, y, t) = \phi(x, y, \eta(x, y, t), t) \quad (6)$$

$$\tilde{w}(x, y, t) = \frac{\partial \phi}{\partial z}(x, y, \eta(x, y, t), t) \quad (7)$$

the KFSBC and the DFSBC can be reformulated following [Zakharov \(1968\)](#) as:

$$\eta_t = -\nabla_H \eta \cdot \nabla_H \tilde{\phi} + \tilde{w}(1 + (\nabla_H \eta)^2) \quad (8)$$

$$\tilde{\phi}_t = -g\eta - \frac{1}{2}(\nabla_H \tilde{\phi})^2 + \frac{1}{2}\tilde{w}^2(1 + (\nabla_H \eta)^2) \quad (9)$$

It can be noted that the Laplace eq. (1), the BBC (eq. (4)) and the body BC (eq. (5)) are all linear equations. Thus, the nonlinearity of the problem originates uniquely from the free surface boundary conditions (eqs. (2) and (3)) or (eqs. (8) and (9)). A classical approach at that point is to assume that surface waves are of small amplitude relative to their wavelength, so that these boundary conditions can be linearized and applied at the SWL (*i.e.* at $z = 0$). Here, we intend to retain full nonlinearity of wave motion by considering the complete conditions (eqs. (8) and (9)). These two equations are used to compute the time evolution of the free surface elevation η and the free surface potential $\tilde{\phi}$. This requires obtaining \tilde{w} from $(\eta, \tilde{\phi})$, a problem usually referred to as Dirichlet-to-Neumann (DtN) problem.

For given values of $(\eta, \tilde{\phi})$, the DtN problem is here solved by solving a BVP problem in the fluid domain on the wave potential $\phi(x, y, z, t)$, composed of the Laplace equation (eq. (1)), the BBC (eq. (4)), the body BC (eq. (5)), the imposed value $\phi(z = \eta) = \tilde{\phi}$ (Dirichlet condition) on the free surface $z = \eta$, supplemented with boundary conditions on lateral boundaries of the domain (of *e.g.* Dirichlet, Neumann, etc. type). The numerical methods to solve the BVP are presented in the next section.

3 The Harmonic Polynomial Cell method (HPC) with immersed free surface

3.1 General principle of the HPC method

In order to solve the above mentioned BVP at a given time, the HPC method introduced by [Shao and Faltinsen \(2012\)](#) is used. It is briefly described here, and more details can be found in [Shao and Faltinsen \(2014b\)](#), [Hanssen et al. \(2015, 2017a\)](#), [Hanssen \(2019\)](#) and [Ma et al. \(2017\)](#). In this work, the HPC approach is implemented and tested in 2 spatial dimensions, *i.e.* in the vertical plane (x, z) , for a wide range of parameters.

The fluid domain is discretized with overlapping macro-cells which are composed of 9 nodes in 2 dimensions. Those macro-cells are obtained by assembling four adjacent quadrilateral cells on an underlying quadrangular mesh. The four cells of a macro-cell share a same vertex node, called the "central node" or "center" of the macro-cell. A typical macro-cell is schematically shown in fig. 1, with the corresponding local index numbers of the 9 nodes. With this convention, any node with global index n has the local index "9" in the considered macro-cell and is considered as an interior fluid point, whereas for example node with local index "4" can either be a fluid point or a point lying on a boundary. In the case node "4" is also an interior fluid point, it defines a new macro-cell overlapping the original one: nodes denoted 6,7,8,9,2 and 1 in fig. 1 also belong to this macro-cell centered by "4".

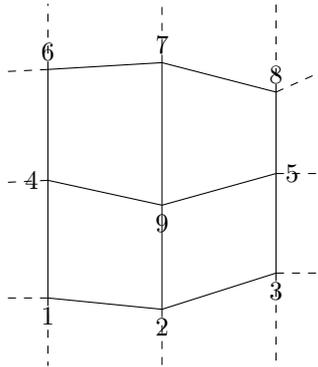


Figure 1: Definition sketch of 9-node macro-cell used for the HPC method, with local numbering of the nodes

In each macro-cell, the velocity potential is approximated as a weighted sum of the 8 first harmonic polynomials (HP), the latter being fundamental polynomial solutions of the Laplace eq. (1). A discussion about which of the HP are to be chosen is given in Ma et al. (2017). Here, we follow Shao and Faltinsen (2012), and select all polynomials of order 0 to 3 plus one fourth-order polynomial, namely: $f_1(\mathbf{x}) = 1$, $f_2(\mathbf{x}) = x$, $f_3(\mathbf{x}) = z$, $f_4(\mathbf{x}) = xz$, $f_5(\mathbf{x}) = x^2 - z^2$, $f_6(\mathbf{x}) = x^3 - 3xz^2$, $f_7(\mathbf{x}) = -z^3 + 3x^2z$ and $f_8(\mathbf{x}) = x^4 - 6x^2z^2 + z^4$. Here, $\mathbf{x} = (x, z)$ represents the spatial coordinates. Thereafter, we define $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}_9$ the same spatial coordinate in the local reference frame of the macro-cell, with \mathbf{x}_9 being the center node of the macro cell. From a given macro-cell, the potential can be approximated at a location \mathbf{x} as:

$$\phi(\mathbf{x}) = \sum_{j=1}^8 b_j f_j(\bar{\mathbf{x}}) \quad (10)$$

As every HP is a solution of the Laplace eq. (1) which is linear, any linear combination of them is also solution of this equation. Thus, the goal now becomes to match the local expressions (LE) given by eq. (10): if this equation is satisfied at each node ($j = 1, \dots, 8$) of each macro-cell, a solution of the BVP is available everywhere. For that reason, macro-cells overlap each other. Note that this study deals with 2D problems, but the method can be extended to 3D cases as shown by Shao and Faltinsen (2014a) considering cubic-like macro-cells with 27 nodes.

The first objective is to determine the vector of coefficients $b_j, j = 1, \dots, 8$ for the selected macro-cell. Recalling that eq. (10) should be verified at the location of each point of the macro-cell (with local index running from 1 to 9), this equation applied at the 8 neighboring nodes (1–8) of the center yields:

$$\phi_i = \phi(\mathbf{x}_i) = \sum_{j=1}^8 b_j f_j(\bar{\mathbf{x}}_i) \quad \text{for } i = 1, \dots, 8 \quad (11)$$

which represents, in vector notation, a relation between the vector of size 8 of the values of the potential ϕ_i at the outer nodes with the vector of size 8 of the b_j coefficients. The 8x8 local matrix linking these two vectors is denoted \mathbf{C} , and defined by $C_{ij} = f_j(\mathbf{x}_i)$. Note that \mathbf{C} is defined geometrically, thus it only depends on the position of the outer nodes i relatively to the position of the central node. \mathbf{C} can be inverted and its inverse is denoted \mathbf{C}^{-1} . The b_j coefficients are then obtained for the given macro-cell as a function of the potentials at the 8 neighboring nodes of the central node of that macro-cell:

$$b_j = \sum_{i=1}^8 C_{ji}^{-1} \phi_i \quad \text{for } j = 1, \dots, 8. \quad (12)$$

Injecting this result into the interpolation eq. (10), a relation is found providing an approximation for the potential at any point located inside the macro-cell using the values of the potential of the eight surrounding nodes of the central node:

$$\phi(\mathbf{x}) = \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} f_j(\bar{\mathbf{x}}) \right) \phi_i \quad (13)$$

This equation will be referred to as local expression (LE) of the potential. It will be used to derive the boundary conditions equations and the fluid node equations that need to be solved in the BVP. Also note that this LE provides a really good interpolation function that can be used for every additional computation once the nodal values of the potential are known (*i.e.* potential derivatives at the free surface or close to the body to compute the pressure field).

Note that the accuracy of LE depends only on the geometry: coordinates at which this equation is applied, shape of the macro-cell, etc. Those dependencies are investigated in details by [Ma et al. \(2017\)](#).

3.2 Treatment of nodes inside the fluid domain

We first consider the general case of macro-cells whose central node is an interior node of the fluid domain. Applying the LE (eq. (13)) at the central node yields a linear relation between the values of the potential at the nine nodes of this macro-cell:

$$\phi_9 = \phi(\mathbf{x}_9) = \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} f_j(\bar{\mathbf{x}}_9) \right) \phi_i \quad (14)$$

We may further simplify this equation by noting that, as $\bar{\mathbf{x}}_9 = (0, 0)$ in local coordinates, all $f_j(\bar{\mathbf{x}}_9)$ vanish, except $f_1(\bar{\mathbf{x}}_9)$ which is constant and equal to 1. Equation (14) then simplifies to:

$$\phi_9 = \sum_{i=1}^8 C_{1i}^{-1} \phi_i \quad (15)$$

meaning that only the first row of the matrix \mathbf{C}^{-1} is needed here.

In order to solve the global potential problem, *i.e.* to find the nodal values of the potential at all grid points (whose total number is denoted N), a global linear system of equations is formed, with general form $\mathbf{A} \cdot \phi = \mathbf{B}$, or:

$$\sum_{l=1}^N A_{kl} \phi_l = B_k \quad \text{for } k = 1, \dots, N. \quad (16)$$

where k and l are global indexes of the nodes. For each interior node in the fluid domain, with global index k and associated macro-cell, an equation of the form of eq. (15) allows to fill a row of the global matrix \mathbf{A} . This row k of the matrix involves only the considered node and its 8 neighboring nodes, making the matrix \mathbf{A} very sparse (at most 9 non-zero elements out of N terms). Moreover, the corresponding right-hand-side (RHS) term B_k is null. Note that all the 8 neighboring nodes of the macro-cell associated with center k should also have a dedicated equation in the global matrix in order to close the system.

3.3 Nodes where a Dirichlet or Neumann boundary condition is imposed

If a Dirichlet boundary condition with value ϕ_D of the potential has to be imposed at the node of global index k , the corresponding equation is simply $\phi_k = \phi_D$, so that only the diagonal element of the global matrix is non-null and equal to 1 for the corresponding row k : $A_{kl} = \delta_{kl} \quad \forall l \in [1, N]$. The corresponding term on the RHS is set to $B_k = \phi_D$.

If a Neumann condition has to imposed at a given node of global index k , the relation set in the global matrix is found through the spatial derivation along the imposed normal \mathbf{n} of the LE (eq. (13)) of any macro-cell on which k appears. In practice, the macro-cell whose center is the closest to the node k is chosen, and we then use:

$$\nabla \phi(\mathbf{x}_k) \cdot \mathbf{n} = \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} \nabla f_j(\bar{\mathbf{x}}_k) \cdot \mathbf{n} \right) \phi_i \quad (17)$$

Thus, a relation is set in the row k of the global matrix to enforce the value of $B_k = \nabla \phi(\mathbf{x}_k) \cdot \mathbf{n}$ at position \mathbf{x}_k . In that case, a maximum of 8 non-zero values appear in this row on the global matrix as the potential of the central node of the macro-cell does not intervene here.

3.4 Treatment of the free surface

As already mentioned, in order to solve the BVP at a given time-step, the system of equations needs to be closed, meaning that each neighbor of a node in the fluid domain should have a dedicated equation. We consider now the case of nodes lying on or in the vicinity of the (time varying) free surface. The free surface potential should be involved here, either directly at a node fitted to the free surface through a Dirichlet condition described in the previous sub-section, or through alternative techniques.

For instance, an Immersed Boundary Method (IBM) was first suggested in the HPC framework by [Hanssen et al. \(2015\)](#) to tackle body boundary conditions. More recently, [Ma et al. \(2017\)](#) compared a modified version of the IBM with two different multi-grid (MG) approaches (fitted or combined with an IBM) for both body and free surface boundary conditions. [Hanssen et al. \(2017a\)](#) and [Hanssen et al. \(2017b\)](#) also made in-depth comparisons of the MG and IB approaches, focusing on the free surface tracking. Both methods showed promising results. [Zhu et al. \(2017\)](#) introduced a similar yet slightly different IB approach with one or two ghost node layers, then realized a comparison between

this IB approach and the original fitted mesh approach. In the present work, the IBM was chosen for the treatment of the free surface, though the fitting mesh method is shortly described thereafter.

3.4.1 Fitted mesh approach for the free surface

The first possibility is to fit the mesh to the actual free surface position at any time when the BVP has to be solved. The mesh is deformed so that the upper node at any abscissa always lies on the free surface. That way, the computational domain is completely closed and the free surface potential is simply enforced as a Dirichlet boundary condition at the correct position $z = \eta$ as explained in § 3.3. With this approach, the algorithm, given the boundary values at the considered time, can be summarized as:

- Deform the mesh to fit the current free surface elevation,
- Build and then invert the local geometric matrices \mathbf{C} ,
- Fill the global matrix \mathbf{A} and RHS \mathbf{B} , using the corresponding Dirichlet conditions at nodes lying on the free surface,
- Invert the global problem to obtain the potential everywhere

Recently, Ma et al. (2017) pointed out that the HPC method efficiency (in terms of accuracy and convergence rate) is greatly improved when a fixed mesh of perfectly-squared cells is used. In this work, the negative effects of a deforming mesh outlined in the previous subsection were also encountered. Especially, for some particular cell shapes, a high increase of the local condition number was observed, leading to difficulty of matrix inversion and important errors on the approximated potential. As a consequence, results were highly dependent on the mesh deformation method employed, especially in the vicinity of a fixed fully-immersed body.

3.4.2 Immersed free surface approach

In order to work with regular fixed grids, an IBM technique was developed and implemented to describe the free surface dynamics. Hanssen et al. (2015) introduced a first version of this method applied on the boundaries of a moving body. This method was recently extended to the free surface and compared to a fitted MG method by Ma et al. (2017) and Hanssen et al. (2017a). In the current work, a semi-Lagrangian IB method introduced by Hanssen et al. (2017a) is chosen.

In this method, the free surface is discretized with markers, evenly spaced and positioned at each vertical intersection with the background fixed grid, as shown in fig. 2. Those markers are semi-Lagrangian in such a way that they are only allowed to move vertically, following eqs. (8) and (9).

At a given time, every node located below the free surface (*i.e.* below a marker) is considered as a node in the fluid domain ("fluid" node), and defines a macro-cell with its 8 neighbors. The global matrix is classically filled with the local expression (eq. (13)) at those nodes. As a consequence, in order to close the system, each neighbor of a node just above the free surface must also have a dedicated equation in the global matrix. These neighbors, represented with grey circles on fig. 2, are denoted as "ghost" nodes. The chosen equation to close the system at a node of this type is the local expression (eq. (13)) applied at the marker position in a given macro-cell:

$$\phi_m = \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} f_j(\bar{\mathbf{x}}_m) \right) \phi_i \quad (18)$$

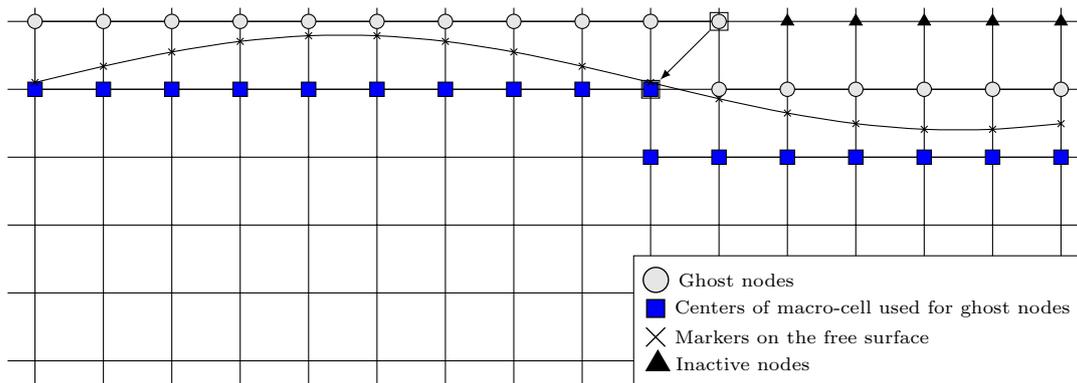


Figure 2: Schematic representation of the immersed free surface in a fixed grid

where $\bar{\mathbf{x}}_m = (x_m, \eta(x_m)) - \mathbf{x}_c$ is the position of the marker in the macro-cell's reference frame (\mathbf{x}_c is the global position of the center node of the chosen macro-cell) and ϕ_m its potential (known at this stage). This ensures that the potential at the free surface point is equal to the potential at the position of the marker from the interpolation equation. In other words, if one wants to interpolate the computed field ϕ at the particular location of the marker \mathbf{x}_m , the results should be consistent and yield the potential ϕ_m .

Note that this eq. (18) is cell dependent (through C_{ji}^{-1} , the involved ϕ_i and the position of the center node \mathbf{x}_c), but also depends on the chosen marker (through \mathbf{x}_m and ϕ_m). The only mathematical restriction on the choice the macro-cell to consider is that the ghost point potential should intervene as one of the ϕ_i in order to impose the needed constraint at this point.

An important note is that the latter eq. (18) is not dependent on the ghost point in any fashion. This implies that if the same couple (marker, macro-cell) is chosen to close the system at two different ghost points, the global matrix will have two strictly identical rows. Its inversion would thus not be possible. Particularly, two vertically aligned ghost points cannot use the same macro-cell equation at the same marker position. Here stands the difference between the IB method of [Hanssen et al. \(2017b\)](#), [Ma et al. \(2017\)](#) and the one chosen by [Zhu et al. \(2017\)](#). [Zhu et al. \(2017\)](#) decided to only impose the marker potential once in the first layer (or two first layers) and to constraint the upper potentials to an arbitrary value (in practice if the point is not used directly, the potential is set to the first point below which potential is used). The method used during this work is closer to the one by [Hanssen et al. \(2017b\)](#) and [Ma et al. \(2017\)](#): if a node needs a constraint but does not have a marker directly underneath (case of two ghost points vertically aligned), the ghost point on the top should invoke the local expression of the macro-cell centered on the closest fluid point instead of the cell centered on the vertically aligned fluid point (case indicated by an arrow in fig. 2). With that method, in such a situation, the potential of the selected marker – *i.e.* vertically aligned with the two ghost points – is imposed twice in two different adjacent macro-cells. A comparison between those two methods had not been conducted and would be of great interest.

Regardless of the kind of IB method used, the main goal is achieved: it is not needed to deform the mesh in time. As a consequence, the computation and inversion of the local (geometric) matrices is only done once, at the beginning of the computation. However, a step of identification of the type of each node, which was proven to be time consuming, is needed instead. Note that this identification algorithm could be greatly improved and is relatively slow in its current implementation. The general algorithm at one time step becomes:

- Identify nodes inside the fluid domain,
- Identify ghost nodes needed to close the system, associated markers and macro-cells,
- Fill global matrix \mathbf{A} and RHS \mathbf{B} ,
- Invert global problem to obtain the potential everywhere,

3.5 Linear solver and advance in time

To solve the global linear sparse system of equations, an iterative GMRES solver, based on Arnoldi inversion, was used for all computations. The base solver was developed by Saad (2003) for sparse matrix (SPARSEKIT library), and includes an incomplete LU factorization preconditionner. In the current implementation, a modified version of the latter is used with the improvement proposed by Baker et al. (2009). Except for stalling during the study of a standing wave at very long time, this solver was proven to be robust. Improvements of the construction step of the global matrix could further be made in order to increase the efficiency of its inversion. Also, the initial guess in the GMRES solver could also be improved taking advantages of the already computed potential values. The number of inner iterations of the GMRES algorithm was chosen as $m \in [30, 60]$ and the iterative solution is considered converged when the residual is lower than $5 \cdot 10^{-9}$.

Marching in time thanks to eqs. (8) and (9) yields the free surface elevation and the free surface potential at the next time step. Note that the steps of computing the RHS terms of these equations are straightforward for most terms directly from the local expression (eq. (13)) of the closest macro-cell. In addition, the spatial derivative of η is computed with a finite difference method. A centered scheme of order 4 is chosen for this work with the objective to maintain the theoretical order 4 of spatial convergence provided by the HPC method.

In order to integrate eqs. (8) and (9), the classical four-step explicit Runge-Kutta method of order 4 (RK4) was selected as time-marching algorithm. During a given simulation, the time step (δt) was chosen to remain constant. Its value is made nondimensional by considering the Courant-Friedrichs-Lewy (CFL) number C_o based on the phase velocity $C = \lambda/T$, where λ is the wavelength and T the wave period:

$$C_o = \frac{C \delta t}{\delta x} = \frac{\lambda / \delta x}{T / \delta t} \quad (19)$$

The CFL number thus corresponds to the ratio of the number of spatial grid-steps per wavelength ($N_x = \lambda / \delta x$) divided by the number of time-steps per wave period ($N_t = T / \delta t$), *i.e.* $C_o = N_x / N_t$.

3.6 Computation of the time derivative of the potential

The pressure inside the fluid domain is obtained from the Bernoulli equation:

$$p(x, z, t) = -\rho \left(\frac{\partial \phi}{\partial t} + \frac{1}{2} (\nabla \phi)^2 + gz \right) \quad (20)$$

This equation is used in every potential based NWT in order to obtain the wave loads. For this reason, the knowledge of the time derivative of the potential is required. In a first attempt, the time derivative of the potential was estimated using a backward finite difference scheme. However, this method is not well suited when important variations of the potential are at play. Moreover, in the case of the IB method, it is not possible to obtain the value of the pressure at a point that was previously above the free surface, and thus for which a time derivative of the potential cannot be computed by the finite difference scheme.

A fairly accurate method is to introduce the (Eulerian) time derivative of the potential as a new variable $\phi_t = \frac{\partial\phi}{\partial t}$ and to solve a similar BVP as described previously on this newly defined variable, noting that ϕ_t has to satisfy the same Laplace equation as ϕ in the fluid domain. This method, first used by [Cointe et al. \(1991\)](#); [Tanizawa \(1996\)](#), has been recently applied by *e.g.* [Guerber \(2011\)](#) in the BEM framework or by [Ma et al. \(2017\)](#) in the HPC method.

Note that the local macro-cell matrices and coefficients, which are only geometrically dependent, do not change. In the different expressions presented above that are used to fill the global matrix, the coefficients linking the different potentials are not time dependent. Thus, the matrix to invert is exactly the same for the ϕ_t field and the ϕ field. However, the boundary conditions on ϕ and ϕ_t might differ, leading to different RHS. This is the case only when the RHS on ϕ is different from zero, as for example, for free surface related closure points.

Remember that the equations at a (non-moving) Neumann condition and at a point inside the fluid domain yield a zero value in the RHS, and thus the equations at those points are exactly the same for the potential variable and for its derivative. At a (non-moving) Dirichlet boundary condition, one would simply impose $\phi_t = 0$ instead of $\phi = \phi_D$. At the IB ghost points (*i.e.* to enforce the free surface condition), the ϕ_t is imposed to match the derivative of the potential with respect to time, known at the marker positions from eq. (9). Even though the global matrices are exactly the same, the RHS being different and the chosen resolution method being iterative (GMRES solver), the easiest way is just to solve twice the almost same problem. A more clever way could maybe be investigated by taking advantage of the previous inversion, but this is left for future work.

4 Validation and convergence study on a nonlinear standing wave

4.1 Presentation of the test-case

The first case consists in simulating a nonlinear standing wave in a domain of uniform water depth h whose extent is equal to one wavelength λ . This case is quite demanding as the wave height H (difference between the maximum and minimum values of free surface elevation at antinode locations) is fixed by choosing a large wave steepness $H/\lambda = 10\%$ (or $kH/2 = \pi/10 \approx 0.314$). We also choose to work in deep water conditions by selecting $h = \lambda = 64$ m (or $kh = 2\pi \approx 6.28$). The water domain at rest is thus of square shape in the (x, z) plane, as illustrated in [fig. 3](#).

Initial elevations of free surface $\eta(x, t = 0)$ are computed from the numerical method proposed by [Tsai and Jeng \(1994\)](#). The initial phase is chosen such that the imposed potential field is null at $t = 0$ at any point in the water domain. This initial state corresponds to a maximum wave elevation at the beginning and the end of the domain ($x/\lambda = 0$ and 1), and a minimum wave elevation at the center point of the domain ($x/\lambda = 0.5$), these three locations being antinodes of the standing wave.

Wall boundary conditions are enforced as null Neumann conditions, *i.e.* $\nabla\phi \cdot \mathbf{n} = 0$ on the three wet walls, where \mathbf{n} is the considered wall normal vector.

The wave is freely evolving under the effect of gravity: in theory one should observe a fully periodic motion without any damping as the viscosity is neglected. At each time step, a spatial $L_2(\eta)$ error on η is computed relative to the theoretical solution of [Tsai and Jeng \(1994\)](#) (denoted η^{th} hereafter), and normalized with the wave height:

$$L_2(\eta, t) = \frac{1}{H} \sqrt{\frac{1}{n_p} \sum_{i=1}^{n_p} (\eta(x_i, t) - \eta^{th}(x_i, t))^2} \quad (21)$$

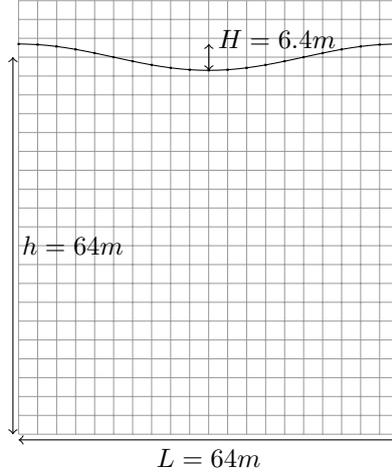


Figure 3: Schematic representation of the nonlinear standing wave with steepness $H/\lambda = 10\%$ at $t = 0$. Mesh grid represented with $N_x = 20$.

where i represents the index of a point on the free surface and n_p the total number of points on the free surface.

4.2 Evolution of $L_2(\eta)$ error with space and time discretizations

The result of this $L_2(\eta)$ -error is represented as a color map at four different times $t/T = 1, 10, 50$ and 100 in fig. 4 as a function of the number of nodes per wavelength ($N_x = \lambda/\delta x$, where δx is the spatial step-size) and the CFL number C_o (defined in § 3.5). Wide ranges of the two discretization parameters are explored, namely $N_x \in [10, 90]$ and $C_o \in [0.05, 4.0]$. Simulations that ran till the end of the requested duration of $100T$ are represented with coloured squares. A circle is chosen as a marker when the computation breaks down before the end of that duration. Nonetheless the markers are colored if the computation did not yet diverge at the time instant shown on the corresponding panel.

Results of fig. 4 show that a large number of simulations were completed over this rather long physical time of $100T$. In this §, no filter were used, so some of the numerical simulations tend to be unstable for extreme values of the discretization parameters. For instance, when $C_o \leq 1$, the computation is mostly unstable and breaks down: before $50T$ when N_x is small (*i.e.* below 40) and between $50T$ and $100T$ when N_x is larger. Note that $C_o = 1$ corresponds to a time step ranging from $N_t = T/\delta t = 10$ to 90 , for $N_x = 10$ and 90 respectively. This value of $C_o = 1$, and associated time step, is the lower stable limit exhibited by these simulations.

On the other hand, when the C_o is too high (*i.e.* larger than 3.5) instabilities also occur almost at the beginning of the simulation ($t/T < 10$), particularly when N_x is small. For very small N_x (in the range 10-15) and whatever the C_o , the computation tends to be unstable. This is probably due to the discretization of the immersed free surface being the same as the discretization of the background mesh. A coarse discretization of the free surface leads to an inaccurate computation of the spatial derivative of η : instabilities may then occur.

A suitable range of parameters is thus determined to avoid instabilities: $1.5 \leq C_o \leq 3.5$ and $40 \leq N_x \leq 90$. This zone is represented in fig. 4 as a rectangular box with a dashed contour. In that zone, all the computations ran with the requested time step over a duration of $100T$. Note that the CPU cost scales with $N_x^2 N_t \sim N_x^3 / C_o$ and thus the most expensive computations in this stable zone

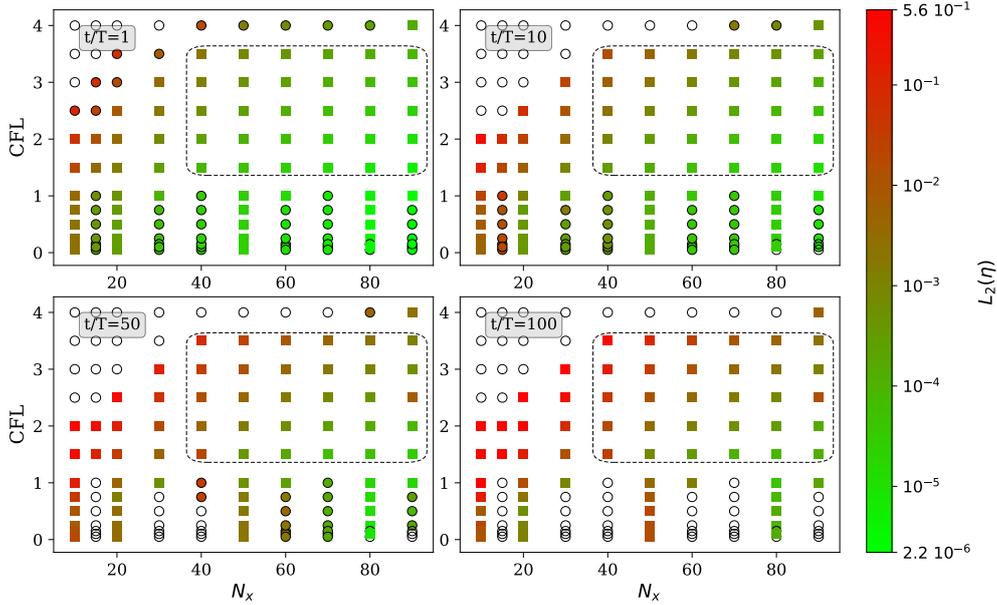


Figure 4: L_2 error on η on the nonlinear standing wave case at four time instants ($t/T = 1, 10, 50$ and 100) as a function of the spatial and temporal discretizations. The color scale indicates the $L_2(\eta)$ error respective to the theoretical solution by Tsai and Jeng (1994). See text for explanations on the significance of the markers shapes.

are approximately 25 times slower than the least expensive ones in the same zone. Also note that the lowest error is almost systematically reached in this zone. The $L_2(\eta)$ error is as small as 2.10^{-6} after $1T$. After $100T$, the lowest error is approximately 10^{-4} . Moreover, the evolution of the value of the error is qualitatively consistent with the mesh refinement and time refinement.

The stability was not assessed for finer mesh than $N_x = 90$ points per wave length, due to increasing computation cost on one hand, and the fact that finer resolutions would lie out of the range of discretizations targeted for real-case applications. In addition, at long time, a discretization of $N_x = 90$ already exhibits behavior that does not match the expected convergence rate, as will be discussed hereafter in greater detail.

4.3 Convergence with time discretization

In order to study the convergence of the method in a more quantitative manner, the $L_2(\eta)$ error is shown as a function of the C_o number for different spatial discretizations N_x at $t/T = 1$ in fig. 5a and at $t/T = 100$ in fig. 5b. A C_o^α regression line is computed and fitted on the linear convergence range of the log-log plot of the error. That will be called "linear range" for simplicity, though it corresponds to an algebraic rate of convergence of the error. Note that this linear range corresponds exactly to the zone in which the computations remain stable (with the exception of one particular point at $t/T = 100$ and $N_x = 90$ excluded from the determination of the convergence rate). At $t/T = 1$, the minimum error is, as expected, obtained for small C_o numbers and large N_x : $L_2(\eta) \sim 10^{-5}$ in the linear range and the minimal error reached is 2.10^{-6} for the finer discretization $N_x = 90$. The algebraic order of convergence is close to 4. This was expected as the temporal scheme is the RK4 method at order 4. Moreover, at this early stage of the simulation the error decreases with a power 4 law only when C_o

$\gtrsim 1.0$. The lowest errors are achieved at $C_o \approx 0.75$. Below that C_o number, a threshold is met: the error remains constant when the time step (and C_o number) is further decreased; it is then controlled by the spatial discretization. It is also possible to note that the CFL number C_o seems to be a relevant metric when testing the convergence of the method: the range of C_o in which the results converge is the same across the 4 considered spatial discretizations.

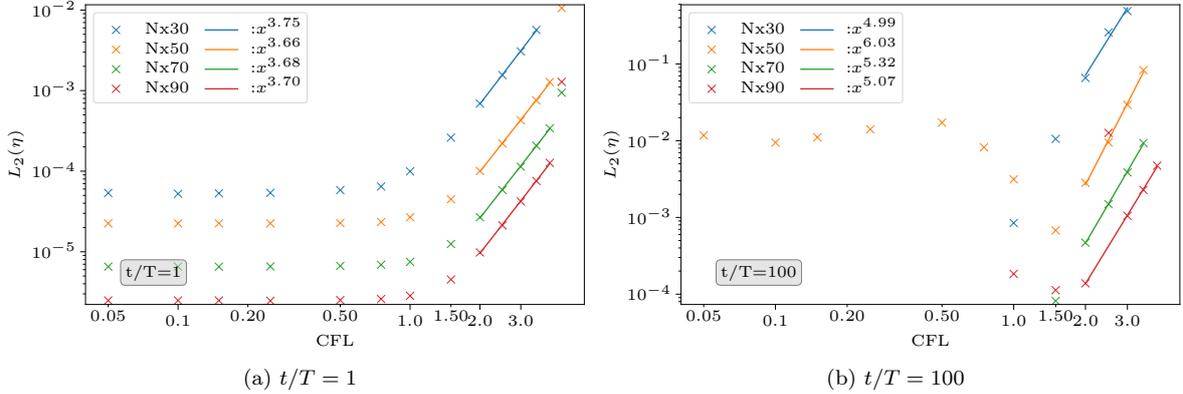


Figure 5: Convergence of the L_2 error on η (crosses) with respect to the temporal discretization at two different physical times: $t/T = 1$ (left panel) and $t/T = 100$ (right panel). The spatial discretization is fixed for a given line. Solid lines represent power regression of the error in the "linear range", the computed power is reported in the legend of the fitted straight lines.

Remark 1 Significant differences in terms of C_o with the work of [Hanssen et al. \(2017a\)](#) have to be stressed. In their simulations the chosen numbers of points per wavelength were similar to the ones used here ($N_x \in [15, 90]$), but the time step was constant and fixed at a small value of $\delta t/T = 1/N_t = 1/250$. This value yields a C_o between 0.06 and 0.36. This range of C_o was shown to be out of the domain of convergence in time in our case. For the same C_o (and apparently the same RK4 time scheme), the computation is indeed converged with respect to the time discretization and yields low error during the first periods, but instabilities then occur when the wave are freely evolving on a longer time scale. [Hanssen et al. \(2017a\)](#) also encountered instabilities with this IB method. To counteract these instabilities, they used a 12th order Savitzky-Golay filter in order to suppress, or at least, attenuate them. No filtering nor smoothing was used in our simulations. This may explain the differences of behavior with [Hanssen et al. \(2017a\)](#) in terms of C_o number. Similarly, in [Zhu et al. \(2017\)](#), also with the RK4 time scheme, the time-step is chosen as $\delta t/T = 1/200$ for a spatial discretization of $\delta x = h/10$. Converted to our numerical case, this would correspond to $N_x = 100$, and so a C_o fixed at 0.5. During the investigations of [Ma et al. \(2017\)](#) on periodic wave propagation, their spatial discretizations ranged from $N_x = 16$ to $N_x = 128$. The equivalent C_o number is thus comprised between 0.4 and 3.2.

At long time $t/T = 100$ (see fig. 5b), the error behaves differently. First, the error is approximately one order of magnitude higher compared to the time $t/T = 1$, but the convergence rate is also slightly different, actually higher. As a matter of fact, at $t/T = 100$, an order 4 of convergence is still found on the wave period and on the amplitude of the computed wave: fig. 6 shows the dependence in CFL of the error in wave period and amplitude at the center of the domain $x/\lambda = 0.5$ at $t/T = 100$ for a fixed $N_x = 30$ as a function of the CFL number C_o . The reference case used here is the one with

$N_x = 100$ at $C_o = 0.05$ computed on one period. On a given case, the period is computed through the mean time separating two successive maximums, then a sliding Fast Fourier Transformation (FFT) is performed to obtain an accurate estimation of the amplitude of the free surface elevation.

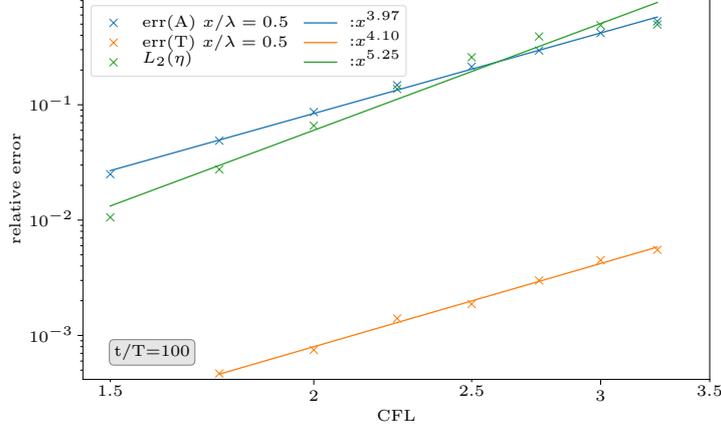


Figure 6: Convergence of the error on wave period and amplitude of the wave elevation at $x/\lambda = 0.5$ for $N_x = 30$. The $L_2(\eta)$ error -combination of both- is also added.

However, the behavior of the $L_2(\eta)$ error results from a combined effect of both the error on the wave period and the error on the amplitude. The relative effect of those errors on the total error is analyzed in detail in appendix A, and a brief summary is given here. Let e be the relative error between two cosine functions. The first is the target function and the second one tends to the first one in amplitude as $\epsilon_a = f_A d^4$ and in period as $\epsilon_t = f_T d^4$. Here d is a discretization variable -either C_o or $1/N_x$ in our case-, which drives the convergence. f_A and f_T are constants with respect to d . At a whole number of periods - *i.e.* $t/T \in \mathbb{N}$ - a Taylor expansion of e when $d \rightarrow 0$ can be performed:

$$e = f_A d^4 + 2\pi^2 \frac{t^2}{T^2} f_T^2 d^8 + (2\pi \frac{t}{T} f_T^3 - 2f_A \pi^2 \frac{t^2}{T^2} f_T^2) d^{12} + O(d^{16}) \quad (22)$$

Note that f_A depends on t/T because the error on amplitude increases with time (in practice a linear dependence was observed at long time, *i.e.* $f_A = \bar{f}_A t/T$). However f_A and f_T should not depend on the convergence parameter d . The order 8 of convergence should disappear for small enough d whatever t/T . In that case, the error on period is negligible compared to the error on amplitude: this results from the presence of the cosine, which elevates the error to the power 2. However, if the error in amplitude f_A increases in time slower than $t^2 f_T^2$, there exists a time after which the error on the wave period will play an important role (order 8 will be predominant). This effect is thought to explain the seemingly high order of convergence of the $L_2(\eta)$ error in fig. 5b. Appendix A shows detailed comparisons at $t/T = 100$ with values of f_A and f_T extracted from our results.

4.4 Convergence with spatial discretization

The convergence with spatial refinement (*i.e.* as a function of $N_x = \lambda/\delta x$) is analyzed in the same way and shown in fig. 7. The order of convergence in space is again 4. Due to the choice of the set of HP including polynomials up to order 4, and the fact that the finite difference scheme of order 4 is used to compute the derivatives of free surface variables, this order 4 was the expected order of convergence.

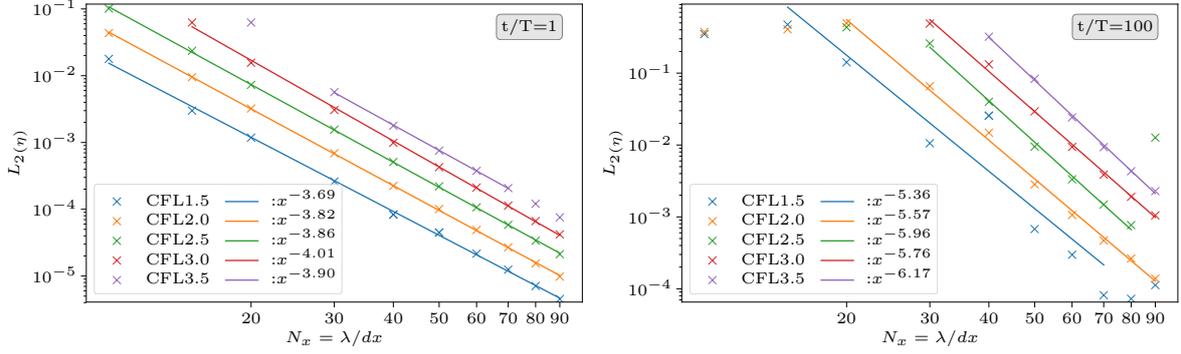


Figure 7: Convergence of the L_2 error on η in mesh refinement, with temporal discretization fixed. Solid lines correspond to the power regression of the error.

At long time the convergence rate exhibits the same behavior as shown in the convergence with time resolution. The latter comments concerning the long time evolution of the error still holds (with, here, $d \equiv \delta x$), and is again thought to explain the increasing order of convergence of the total error $L_2(\eta)$ with time. Of course the values of the corresponding constants f_A and f_T are different. Another effect also occurs: when the C_o number increases, so does the order of convergence. This small yet clear effect at time $t/T = 100$ is not completely understood. It would mean that the value of f_T increases faster with the C_o number than the value of f_A does.

4.5 Effect of filtering the free surface variables

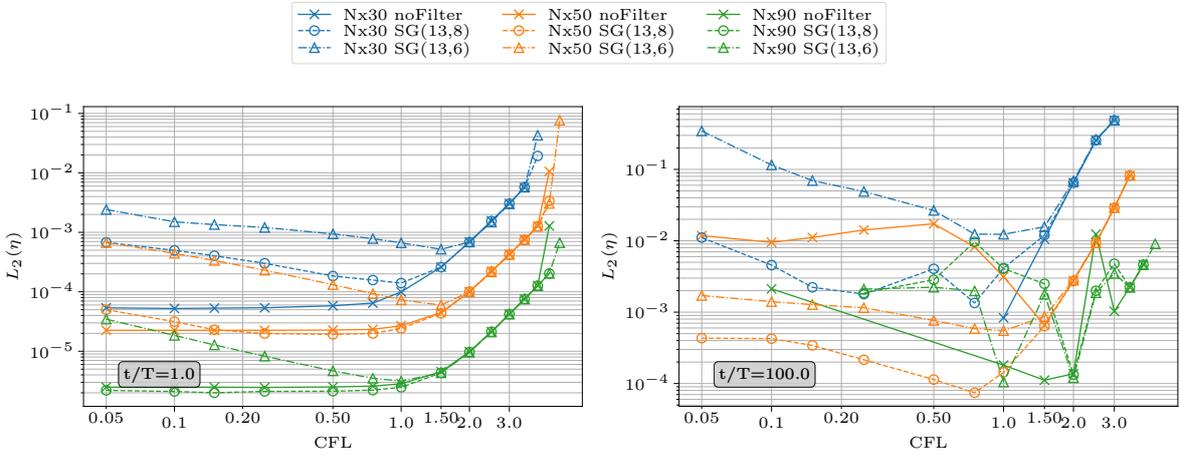


Figure 8: Convergence of the L_2 error on η with respect to the CFL number at two different physical times: $t/T = 1$ (left panel) and $t/T = 100$ (right panel), showing the effect of Savitzky-Golay filters SG(13,6) (triangles) and SG(13,8) (circles) in comparison to simulations run without any filter (crosses).

As shown in the previous sub-section, stable computations are obtained for a range of spatial and

temporal discretizations without any filtering. The phenomenon limiting this range is the so-called "sawtooth instability" developing on the free surface immersed boundary. In this sub-section, we evaluate the effect of applying a centered Savitzky-Golay filter (Savitzky and Golay, 1964) on the achievable range of discretization parameters. This filter is denoted $SG(N,O)$ where N is the total number of points of the filter (*i.e.* $(N - 1)/2$ points are involved on each side of a considered node) and O its order. An in-depth investigation on the effect of such filter is given in Ducroz et al. (2014).

The filter is applied to the free surface elevation first. Afterwards, a correction step of the free surface potential was shown to be necessary to avoid the growth of instabilities: the free surface potential is enforced from an interpolation using the LE eq. (13) at the updated point elevation. The same filter is then applied to the newly obtained free surface velocity potential. This procedure is applied once per time step, after the resolution of the main BVP.

Figure 8 shows the dependence on CFL of the free surface elevation error after 1 and 100 wave periods without filter (similar curves as in fig. 5) and with two SG filters with $N = 13$ and orders $O = 6$ and 8 respectively. A great improvement on the width of the accessible parameter range is obtained, especially for the long simulation period ($t/T = 100$), for which almost all the simulations ran to their target duration.

It can be noted that no significant gain in terms of accuracy can be obtained at short time ($t/T = 1$). Rather, the coarsest mesh resolutions exhibit a significant increase in total error. When increasing the spatial grid step δx at fixed CFL value, two opposite effects are present: 1) the time step δt increases, leading to a reduced number of filter applications for a given simulation duration, 2) however, each application of the SG filter increases in strength as shown by Ducroz et al. (2014). Thus, as the total error increases as the mesh gets coarser, it is deduced that the second effect is dominant over the first one. This effect is more marked with $SG(13,6)$ in comparison with $SG(13,8)$. It is also possible to observe the former effect at fixed N_x with decreasing CFL, as the total error increases.

After a longer time evolution, benefits of the filter can be pointed out: lower CFL can be used without breaking the simulation. For some spatial discretizations, this leads to a lower minimal achievable error. For example, at $N_x = 50$ the simulation at $CFL = 0.75$ is stabilized, leading to the best achieved accuracy, slightly better than with $N_x = 90$, $CFL = 2$.

Overall, this type of filter, in particular used with $SG(13,8)$ setting, appears to be able to contain the growth of instabilities and allows to run simulations with large intervals of discretization parameters. Though an increase in error is noted in some cases characterized by low CFL values at short times, the long term results are improved regarding both the number of stable simulations and the final error levels.

Because good enough stability was denoted in a relatively large range of N_x, CFL when the simulated time is contained ($\leq 100T$), the use of a filter was not found to be necessary for the rest of the computations presented in this paper, unless stated otherwise.

4.6 Highly nonlinear standing wave

Though the standing wave case considered above has a rather high steepness ($H/\lambda = 10\%$), in order to approach the maximal achievable standing wave height, a test case inspired from Ducroz et al. (2014) was simulated. A nonlinear regular wave train with incident wave steepness $H_I/\lambda_I = 7.5\%$ is generated at the left boundary of the domain using a stream-function solution of order 20. These waves are fully reflected by a vertical wall at the right boundary. The incident wavelength is set to $\lambda_I = 1$ m for a constant water depth of $h/\lambda_I = 1$. A $SG(13,8)$ filter was found to be necessary to simulate such level of nonlinearity.

Figure 9 depicts the obtained wave profiles over one wavelength close to the reflecting wall at the peak and trough time instants compared the experiments of Taylor (1953) and OceanWave3D results

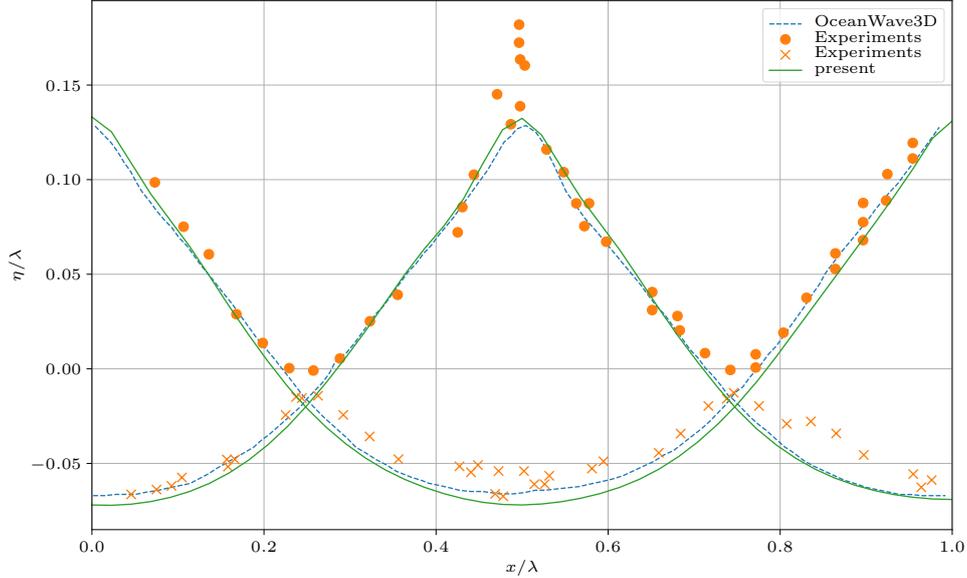


Figure 9: Free surface elevation profiles at time instants corresponding to minimum and maximum amplitude at an antinode position (*i.e.* at $x/\lambda = 0.5$) of the extreme standing wave generated by a full reflection on a vertical wall of an incident wave with $H_I/\lambda_I = 7.5\%$. Comparisons are made with the experiments of Taylor (1953) (with filled circles for data from the downward pointing probe and crosses for data from the upward pointing probe) and the numerical results of OceanWave3D taken from Ducroz et al. (2014).

presented by Ducroz et al. (2014). A good agreement with both sources of data is obtained, with very limited differences with OceanWave3D. The obtained steepness, relative to the standing wave wavelength, is larger than twice the incident one, and is very close to the theoretical maximum of $H_S/\lambda_S \approx 0.20$.

4.7 Note on the computational efficiency

Figure 10 shows the computational burdens of the current implementation of the NWT on the standing wave case of § 4.1. Figure 10b represents some of the CPU costs encountered during the convergence study ($CFL = 2.0$ on fig. 7) with a one-wavelength long domain and different values of N_x , while fig. 10a is representative of more classical applications, simulating a domain of variable length (from 1 to 64 wavelengths) and a fixed value $N_x = 60$.

Note that many routines could still be optimized, especially the mesh related ones. For example, the current implementation of the routine that determines whether a node is outside or inside the water domain (“LocateNodes” on the figure), is currently of complexity $O(N_{bb}N_s)$ where N_{bb} is the number of nodes within the smallest box that contains all the free surface nodes and N_s is the number of segments on the free surface. When only the domain width is varying, this reduces to a complexity of $O(N^2)$ where N is the total number of nodes, which is confirmed on fig. 10a.

While not needed in the standing wave case because no loads are to be computed, the problem on ϕ_t is resolved anyway to give a better grasp of what the computational costs would be in a real application case. The problems on ϕ and ϕ_t yield the exact same matrices, and their resolutions are

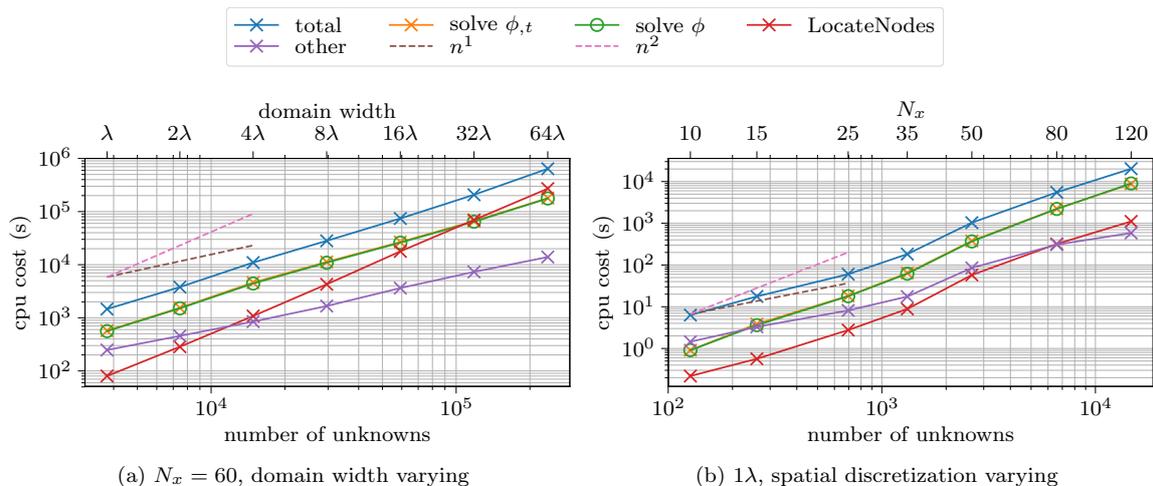


Figure 10: Computational cost of 100 wave periods with different mesh widths (10a) with $N_x = 60$, and different spatial discretizations N_x for a domain width of one wavelength (10b). The CFL number is here fixed at 2. All computations are run in sequential mode on an Intel® Xeon® CPU E5-2650 v3 at 2.30 GHz

thus of same cost. However, one could take advantage of the first resolution. This constitutes a second example of possible optimization of the current code.

When the spatial discretization is reduced, the CFL being fixed implies that the time step decreases. Thus, the cost of computing a given number of periods is increased by a factor N : the time spent on each routine is increased by both the number of points but also the number of times it is repeated.

From fig. 10a, a rough estimate of the CPU time required for the simulation of t/T periods on a real application case (of domain width 8λ) and water depth h , assuming that the problem is of complexity $O(N)$, can be given as:

$$c_{CPU} = \frac{t}{100T} \frac{h}{\lambda} c_{s,t/T=100,8\lambda} \quad (23)$$

where $c_{s,t/T=100,8\lambda} = 2 \cdot 10^4$ s = 5 h 33 min, *i.e.* the CPU cost for the simulation of 100 periods of the standing wave case on a domain of width 8λ with a water depth of $h = \lambda$. Of course, this expression constitutes a crude approximation and different routines, corresponding to the methods presented later in § 5, might increase the CPU cost of the overall simulation, even more so if they are not optimized.

4.8 Summary of numerical convergence study

After a comprehensive numerical study of the HPC method on challenging nonlinear standing wave cases, the efficiency and accuracy of the IBM modeling of the free surface applied on a fixed underlying spatial mesh was demonstrated. Optimal ranges of spatial and temporal discretization parameters were determined:

- the spatial discretization δx should be chosen to have $N_x = \lambda/\delta x$ between 40 and 90 nodes per wavelength, which is a reasonable range of values for practical applications.

- the temporal discretization δt should be best selected to have a CFL number $1.5 \leq C_o \leq 3.5$, meaning that the number of time steps per period $N_t = T/\delta t$, also given by N_x/C_o , is then comprised between $N_x/3.5$ and $N_x/1.5$. This is highly beneficial as it authorizes rather large time-steps for practical applications.
- Moreover, the application of a Savitzky-Golay filter broadens this stable range for long simulation duration, especially for low CFL numbers, allowing to reach smaller total error at the cost of an increased CPU time.
- the present implementation of the HPC shows an algebraic convergence rate with the spatial resolution of order greater than 4.
- it also shows an algebraic convergence rate with the temporal resolution of order comprised between 4 and 5 (for long time simulations).
- extrapolating the obtained computational burdens predicts a contained cost of the numerical model.

Finally, the computational burden is shown to remain limited, even for relatively long simulation duration. This is true even when a domain representative of an engineering application case is selected.

5 Introduction of a fixed body in the NWT

5.1 A double mesh strategy to adapt the resolution in the vicinity of a body

After having validated the method with nonlinear waves in the previous section, and particularly the immersed free-surface strategy, the next objective is to include a body in the fluid domain, either fully submerged or floating. Obviously, a desirable solution would retain cells of square shape and constant geometry as much as possible, even in the case of a moving body (not treated here however).

Once again, different strategies are possible. [Hanssen et al. \(2015\)](#) first introduced an IB method for bodies in waves in the HPC framework. [Ma et al. \(2017\)](#) compared this method with an immersed overlapping grid fitted to the boundaries (corresponding to the body in our case). Recently, this technique was successfully applied to a 3D problem by [Liang et al. \(2020\)](#): the liquid sloshing problem in an upright circular tank. Accurate results were obtained when compared to weakly nonlinear and linear model theories. This newly introduced grid will often be referred to as the "fitted mesh" for simplicity. In the current study, the latter strategy is chosen. Two main reasons led to this choice: first, an oscillatory behavior was exhibited by [Ma et al. \(2017, Figs. 24,25\)](#) when studying the spatial convergence with an IBM. This oscillatory behavior is also present when the body is moving, with a large magnitude. This is mainly due to an incremental change in the chosen ghost nodes which can turn to be favorable at some time steps (*i.e.* for certain grid configurations) and unfavorable at some other ones. Moreover, this oscillatory behavior does not come with a reduction of the error, both for the fixed and oscillating body.

The second reason is that adding a new fitted mesh allows to decouple the discretization of the wave propagation part (usually defined with respect to the wavelength, *e.g.* approximately $N_x \in [40 - 90]$ as shown in the previous section) from the discretization appropriate for the resolution of the potential close to the body (usually defined with respect to the body characteristic dimension, denoted D). Thus, by using two different grids, a suitable discretization for both the wavelength and the computation of the loads would be possible. For instance, including a small body relative to the incoming wavelength would be challenging with the IBM: too many nodes would be required so as

to correctly solve the BVP in the vicinity of the body whereas less nodes would be needed further away. From a quantitative point of view, the case inspired from Chaplin (1984) and treated in § 6.1 hereafter involves an important ratio $\lambda/D \approx 15$. Thus, if the far-field discretization is set as $N_x = 90$ to correctly capture the propagation of the waves, then D is discretized with only 6 nodes. Such high values of the λ/D ratio, often encountered in practical engineering applications, are more easily taken into account with a second grid fitting the body than with a choice of higher order cell or with a local refinement of the grid. Note that the solution combining both a secondary grid and a solid immersed boundary has been tested by Hanssen (2019); Hanssen and Greco (2021) with promising results. On this subject Ma et al. (2017, § 4.2.2) applied this combination to model the free surface and obtained a important reduction of the resulting error.

5.2 The two way-communication inside the fluid domain

Thus, a boundary fitted grid (BFG) is added locally around the body, overlapping the background grid (BGG). These grids and the points associated to the method are represented on fig. 11. The Laplace problem is solved on both these grids simultaneously, *i.e.* both domains are solved in the same global matrix problem. The global matrix size is increased by the number of nodes of the BFG and decreased by the number of nodes inactivated in the BGG. One can note that the global matrix is thus almost defined by block, each corresponding to a grid.

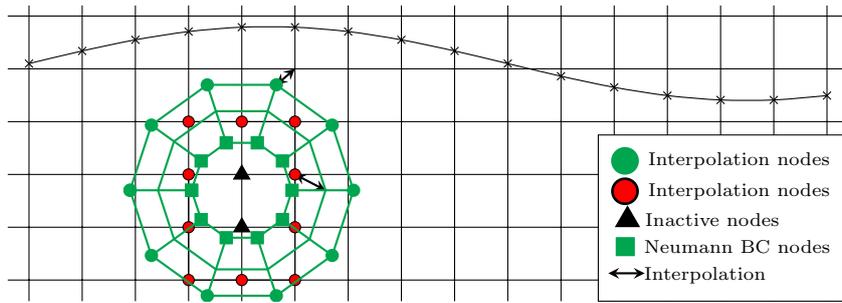


Figure 11: Schematic representation of the immersed free surface and body below the free surface. A circle means an "interpolation" node (described in the text) while the colors are used to identify particular nodes on the two grids, as indicated in the legend.

Thus, the boundary nodes of the new BFG also need a dedicated equation in order for the system to be closed. For a node laying on the body boundary, a simple Neumann boundary condition is set and enforced in the global matrix, as described in § 3.3. The nodes indicated by circle markers on fig. 11 serve as communication nodes between the two grids and are denoted "interpolation nodes". For example, for a node P_f located on the outer contour of the BFG (green circle markers on fig. 11), the imposed equation in the global matrix is the interpolation equation from the closest macro-cell in the BGG (eq. (13)). On fig. 11, a double arrow gives a representative example of the link between P_f and the center of the closest macro-cell in the BGG. This ensures – in an implicit manner – that the potential at the location \mathbf{x}_{P_f} is the same in both meshes:

$$\phi_{P_f}^{(f)} = \phi^{(bg)}(\mathbf{x}_{P_f}) \quad (24)$$

where $\phi_{P_f}^{(f)}$ is the potential of the particular node P_f (directly an unknown of our system of equations), and $\phi^{(bg)}(\mathbf{x}_{P_f})$ represents the value of the interpolation equation from the background potential field at

the given coordinate \mathbf{x}_{P_f} . Further developing eq. (24) and using the closest macro-cell local expression (eq. (13)) yields an implicit interpolation equation:

$$\phi_{P_f}^{(f)} = \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} f_j(\bar{\mathbf{x}}_{P_f}) \right) \phi_{P_i} \Big|_{(bgc)} \quad (25)$$

Where the notation $|_{(bgc)}$ emphasizes that the local expression is applied on the closest background macro-cell. ϕ_{P_i} are the potential of the bounding nodes of this macro-cell. Note that the two problems associated with the different nodes are almost independent. The global matrix is thus almost defined by block, except at those interpolation nodes: the potential of a node P_f in the fitted grid is implicitly linked with the potentials of the neighboring background nodes.

Points belonging to the BGG situated inside the body are inactivated (black triangles on fig. 11). Thus, equations are needed for the points of the BGG surrounding those inactive points (red circle markers on fig. 11). The same method is used here: the interpolation equation (eq. (13)) is enforced such that the interpolation of the fitted grid potential matches the node potential. In other words, the interpolation is effective from the BFG to the BGG, using the LE of the closest cell of the fitted mesh. Thus, denoting this point P_{bg} and its coordinates $\mathbf{x}_{P_{bg}}$:

$$\phi_{P_{bg}}^{(bg)} = \phi^{(f)}(\mathbf{x}_{P_{bg}}) \quad (26)$$

Here again, this relation is represented for one particular node on fig. 11 by a double arrow.

So, by considering the various type of nodes discussed above, the proposed method ensures a consistent implicit two-way communication between the two meshes of interest, as the BVP problems (on ϕ and ϕ_t) are solved on both grids simultaneously, *i.e.* through the inversion of a unique global linear problem.

5.3 Free surface piercing body

At this stage, a two way communication is ensured between the fitted grid (BFG) and the background grid (BGG). The problem is closed in the sense that every node involved in the global matrix has its own dedicated equation. Still, a difficulty arises when the fitted mesh pierces the free surface. Indeed, it is not possible to interpolate outer points of the fitted mesh where no solution is computed (above the free surface): this issue is solved by introducing a new free surface, evolving in the BFG, which allows to solve and advance the free surface locally at the scale of the body. In this work, having a dedicated discretization in the vicinity of the body was proven to be necessary, when for example the impact of the incident wave on the front of the body resulted in the generation of waves of short wavelength and large steepness. Note that using a single free surface with different set of markers is also possible (see *e.g.* Hanssen, 2019; Hanssen and Greco, 2021). The free surface evolving in the background grid is truncated such that no marker is defined inside the body (*i.e.* markers are only present in the fluid domain) as can be seen in fig. 12.

For simplification purposes, the new free surface evolving in the BFG will be called "fitted free surface" even though this free surface also uses the IBM described in § 3.4.2. Thus, we obtain two free surfaces, with different resolutions in space, following their respective grid discretizations, that overlap each other in the vicinity of the body. To ensure the communication between both free surface curves, the outer nodes positions and values of variables ϕ and ϕ_t of one free surface are interpolated and enforced through a 1D B-spline interpolation from the other free surface. Referring to the schematic representation in fig. 12, the position and values of the outer right node of the background free surface is enforced so as to match the fitted free surface. Reciprocally, the position and values of the outer left node of the fitted free surface is enforced so as to match the background free surface.

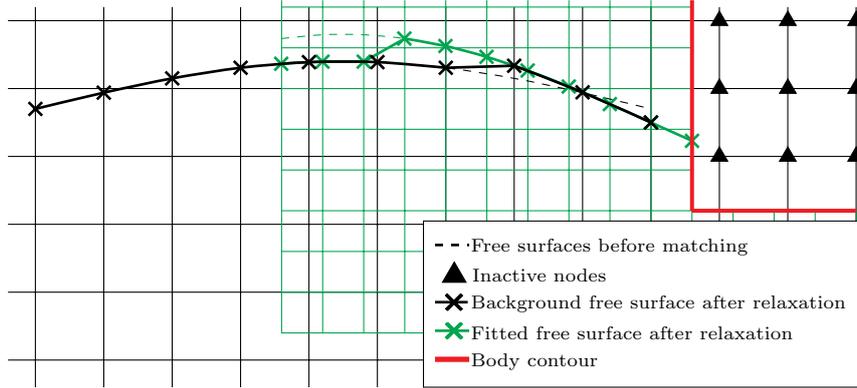


Figure 12: Schematic representation of the background free surface and the fitted free surface. A matching is enforced between the two free surfaces. The difference between the two free surfaces is here exaggerated for clarity.

However, if this enforcement affects only one marker at the extremity, instabilities may occur. For example a stencil of two points on each side is the minimal length to maintain a 4th order of spatial convergence with a 1D centered finite difference scheme. To prevent this from having an important impact, relaxation zones are set to incrementally match the free surfaces at their extremities. Relaxation formulas and weights are thus needed for every marker:

$$\gamma_e = (1 - \alpha)\gamma_i + \alpha\gamma_t \quad (27)$$

where γ_e represents the value to enforce, γ_i the initial marker value, γ_t the target value (interpolated value from the other free surface) and α an arbitrary weight function of the marker position that evolves between 0 and 1. Note that for this application, γ stands for either ϕ , ϕ_t or η . Many different functions for α were tested and implemented without significant impact. In practice free surfaces are completely matched over a given length (*i.e.* $\alpha = 1$ if the marker distance to the free surface extremity is lower than a certain threshold, $\alpha = 0$ otherwise). The fig. 12 emphasizes the effect of such relaxation functions: it shows the free surfaces before (dotted lines) and after the matching (solid lines) using this method. Other methods exist for this type of matching. For example, in the context of a domain decomposition approach, a buffer zone on which a common free surface is computed and enforced in both domains is presented in Kim et al. (2010) and Lu et al. (2017). This latter approach, involving only one buffer zone, may be regarded as a particular case of the method presented here. Another hybrid coupling between the OceanWave3D and OpenFOAM codes that uses relaxation zones to transfer information from one model to the other was recently investigated by Kemper et al. (2019). For a more detailed review of coupling methods, the reader is referred to Di Paolo et al. (2021), and in particular their Table 1.

Note that if the body is not at the extremity of the computational domain, a second fitted free surface is needed on the other side of the body. This will be used in § 6.2.

From a numerical point of view, the fitted mesh is considered as an unstructured grid. At a price of additional coding efforts and an increase of CPU time when identifying points (as well as an increase in the memory usage), a gain is made on the simplicity of inclusion of complex bodies of arbitrary shape. However, as already stated, the implemented HPC method requires square cells to be most effective (Ma et al., 2017). Taking advantage of the fact that the BGG will remain a structured mono block, it would thus be possible to modify the methods on this grid to reduce both the RAM requirement

and the necessary CPU time associated with identification of node types and interpolations between the resolutions of the BVP themselves.

5.4 Singular node treatment (sharp corners)

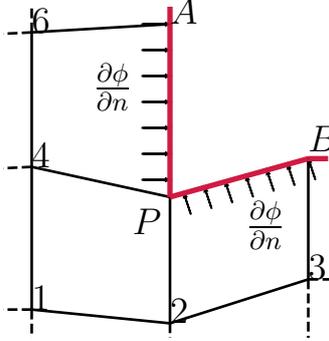


Figure 13: Illustrative sketch of a sharp body or domain corner.

Figure 13 depicts a sharp corner situation. In this situation the choice of the Neumann direction to impose a wall condition at the sharp corner node (denoted P on the figure) is not obvious. However, in the classical HPC approach, only one equation can be set for that node.

To overcome this issue, a first approach was suggested by Liang et al. (2015). They applied a domain decomposition method in the vicinity of the corner apex: a local corner-flow solution was then coupled with the HPC problem.

Another method, suggested by Zhu et al. (2017), consists in imposing a null total flux through the line APB . In the current study, an extension of this method is implemented to tackle external corners such as body corners. In this framework, the obtained and enforced equation links the unknowns appearing in two different macro-cells. This equation is $F_{APB} = 0$ where, considering straight segments, the total flux can be expressed as:

$$\begin{aligned}
 F_{APB} &= l_{PA} \int_{PA} (\nabla \phi(\mathbf{x}(s)) \cdot \mathbf{n}) ds + l_{PB} \int_{PB} (\nabla \phi(\mathbf{x}(s)) \cdot \mathbf{n}) ds & (28) \\
 &= l_{AP} \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} \mathbf{I}_j(AP) \cdot \mathbf{n}(AP) \right) \phi_i \Big|_{c_{AP}} + l_{PB} \sum_{i=1}^8 \left(\sum_{j=1}^8 C_{ji}^{-1} \mathbf{I}_j(PB) \cdot \mathbf{n}(PB) \right) \phi_i \Big|_{c_{PB}} & (29)
 \end{aligned}$$

where $\mathbf{I}_j(AP) = \int_{AP} \nabla f_j(\bar{\mathbf{x}}(s)) ds$, s is the scaled abscissa varying in $[0, 1]$ within the given segment, $\mathbf{n}(AP)$, $\mathbf{n}(PB)$ are the normal vectors to the segments AP and PB respectively. The length of a segment is denoted with l . A vertical bar is added to underline that each segment considers a different macro-cell for ϕ_i , \mathbf{I}_j , C_{ij} , \mathbf{n} , and for the computation of the local coordinate $\bar{\mathbf{x}}(s)$. Note that the vector integrals \mathbf{I}_j can be computed analytically, knowing the analytical expressions of the harmonic polynomials.

The implementation of this method is assessed on a simple case shown in fig. 14 for which the boundary conditions are set according to an arbitrary analytical function, given in the legend of the figure. This figure shows the normalized error on the potential over the domain after the solution of the

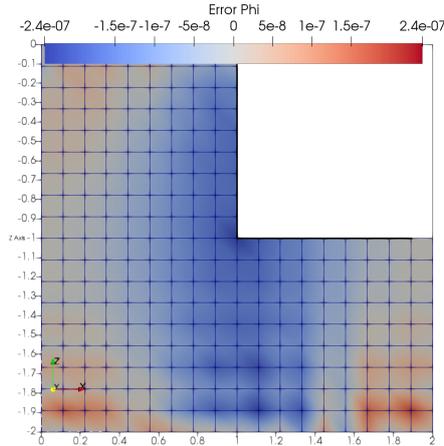


Figure 14: Sharp external body corner. Local error on the potential $(\phi - \phi_{th})/\max(\phi_{th})$. $\phi_{th} = \cosh(k(z + h/2))\cos(kx)$ where $k = 2\pi/L$. $L = 2$ m and $h = 2$ m are respectively the total width and height of the domain. This analytical function is imposed as Neumann condition on the black lines and as Dirichlet condition at the other boundary nodes. The sharp corner method is used on the center point of coordinates $(L/2, -h/2) = (1, -1)$.

Laplace BVP. It can be observed that the error in the vicinity of the corner is of similar magnitude as other nodes in the domain, which validates the implementation. In practice this method was selected for mostly three reasons: i) it removes the need of an arbitrary choice of normal vector at the corner node, ii) it results in a better total mass conservation, and iii) it increases the numerical stability on real wave-body interaction cases (not shown here).

6 Validation of wave-body interaction against two flume experiments

In order to validate the methods presented above, two experimental test cases were selected. The first one is chosen to verify the boundary-fitted overlapping grid method with a fully submerged horizontal cylinder of circular cross-section. The second one is a free surface piercing barge of a rectangular cross-section.

6.1 Fixed horizontal submerged cylinder

Chaplin (1984) studied a fixed horizontal cylinder, with a low submergence below the SWL, in regular waves of period $T = 1$ s. Accurate experimental results about the nonlinearities of wave loads on the cylinder were obtained and are often used to validate NWTs (e.g. Guerber, 2011). The total water depth is $d = 0.85$ m which, together with the period, imposes a wavelength of approximately $\lambda = 1.56$ m (slightly varying with the incident wave height). The cylinder of diameter $D = 0.102$ m is immersed with its center located at $z_c = -D$ below the SWL.

This problem is numerically difficult to solve for volume field methods as the cylinder is close to the free surface, such that the fluid domain right above the cylinder is reduced to a small water gap of height $D/2 \approx \lambda/30$ (when the water is at rest). This water gap needs to be meshed and resolved

with the HPC method. Thus, a spatial discretization of $\lambda/\delta x \in [40, 90]$ would yield a discretization of this gap with only $\sim 2 - 3$ nodes.

The nonlinear regular incident waves are generated using the so-called stream function theory (Fenton, 1988). To avoid reflection on both the wave maker side and the outlet Neumann wall and to impose the target incident wave field, relaxation zones are introduced to enforce the requested values over a distance chosen as $L_{relax} = \lambda$ through a commonly used exponential weighting function (e.g. Jacobsen et al., 2011). Note that other techniques of waves generation and absorption are possible. For example Clamond et al. (2005) introduced a damping term in the Bernoulli equation in order to modify the DFSBC: the wave elevation is smoothly driven to the SWL. Figure 15 shows at scale the computational domain including the two relaxation zones.

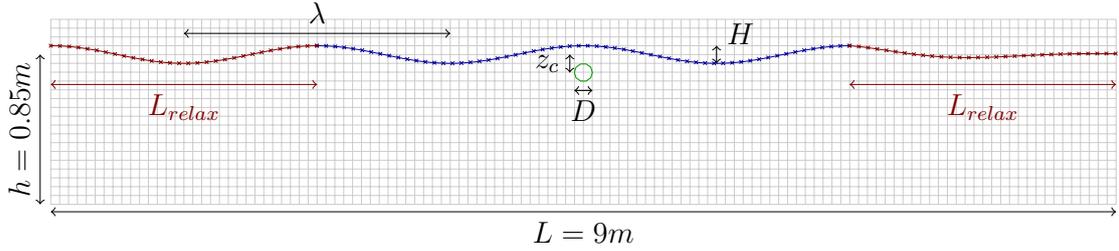


Figure 15: Schematic representation of the numerical set-up inspired from Chaplin (1984). The local refined mesh fitted to the cylinder (not represented in this figure) is shown in fig. 16.

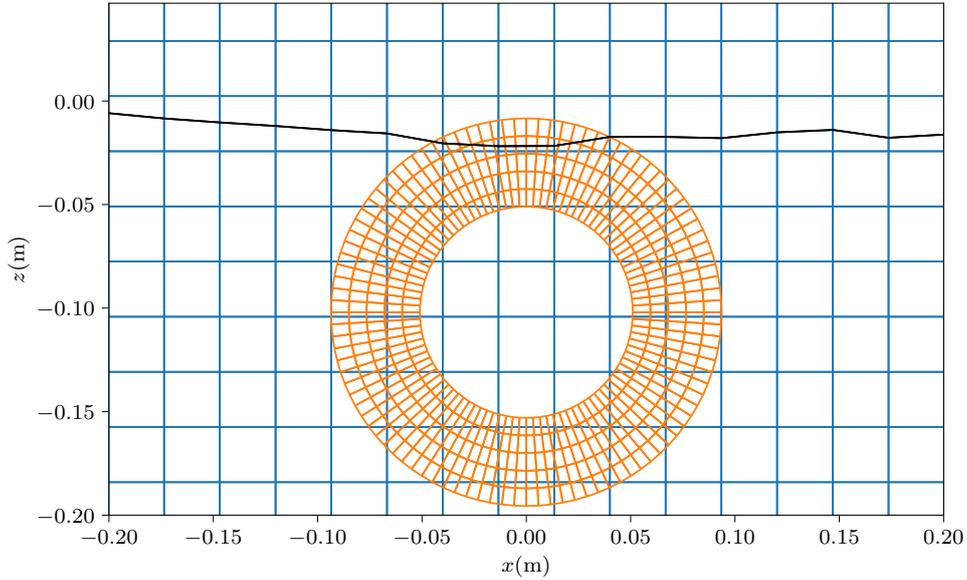


Figure 16: Example of mesh grids and free surface at the passage of a wave trough with $H/\lambda = 2.3\%$. The SWL is at $z = 0$ m.

We focus our attention to the vertical force exerted on the cylinder once the periodic wave motion is established in the NWT. A Fourier analysis is applied to the computed time series of vertical force.

The normalized amplitudes of the harmonics of the vertical force and the mean vertical force (drift force) are plotted in fig. 17 as a function of the Keulegan-Carpenter number (KC), defined as:

$$KC = \pi \frac{H}{D} \exp(kz_c) \quad (30)$$

On this figure, the results from the present NWT are compared with the experimental values from Chaplin (1984) and the numerical results from Guerber (2011). The linear theory results from Ogilvie (1963) are also added for comparison of the amplitude of the first harmonic.

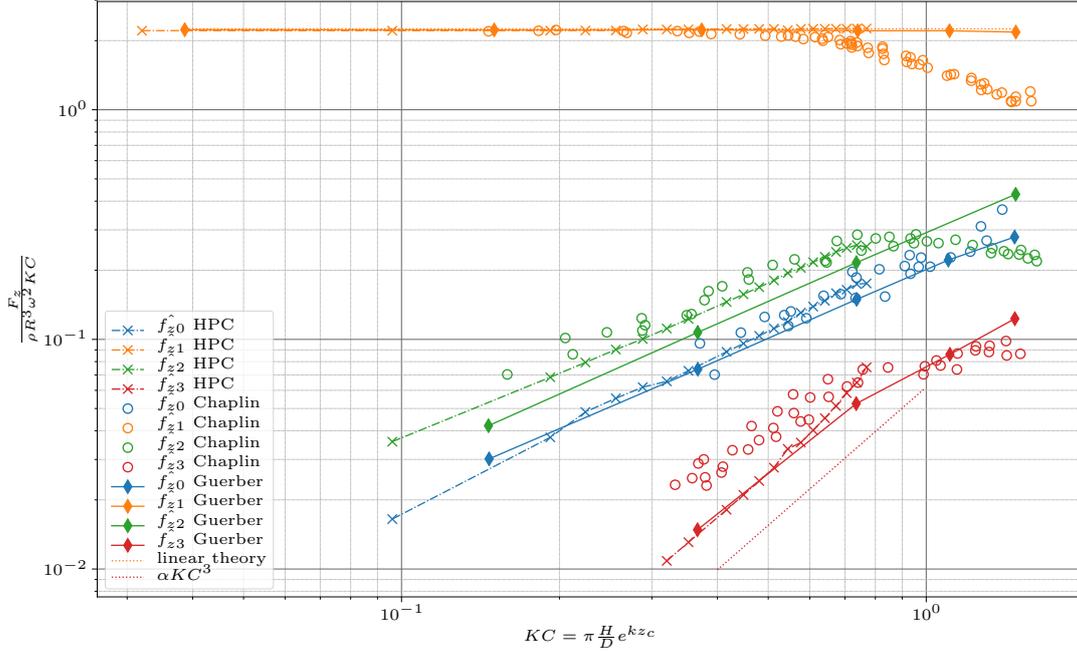


Figure 17: Amplitudes of the various harmonic components of the vertical load on the horizontal circular cylinder. Current results (crosses) compared with numerical simulations from Guerber (2011) (lines with diamonds) and experiments from Chaplin (1984) (empty circles). The amplitude of the first order harmonic based on linear prediction is added as well as a third order model line, to compare with the evolution of the amplitude of the third order harmonic.

All harmonics amplitudes up to third order are in relative good agreement with the BEM simulations from Guerber (2011), although some discrepancies can be denoted. The mean value and first order harmonic are difficult to distinguish between the two sets of numerical results, whereas the amplitude of the second order harmonic is closer from the experimental data with the current HPC method. For the third order harmonic, which is of very small relative amplitude, it is difficult to identify the most accurate method.

Moreover, the behavior of the different harmonics amplitudes seems to agree with the expected theoretical results: the amplitude of the first harmonic increases as KC^1 (i.e. f_{z1}/KC is constant), the drift force and second order harmonic amplitude both increase as KC^2 , and the third order harmonic amplitude increases as KC^3 . Regarding the latter, the HPC method reproduces more closely that order 3 in KC in comparison with both Guerber (2011) and the experimental results. Of course, limitations in the comparison with the experiments can be observed, in particular for larger wave

heights. This is mainly due to the viscous effects (not considered here, nor in the simulations of Guerber (2011)). When viscous simulations are done, the experimental variation can be retrieved (*e.g.* Tavassoli and Kim, 2001).

However, a difficulty arises with the HPC method when KC increases (*i.e.* for larger wave heights). As the Laplace equation is solved in the different cells inside the fluid volume, there should always be at least a fluid point in the volume above the cylinder top. In waves of large amplitude, the cylinder top is very close to the free surface, in particular when a wave trough passes over the cylinder. In this situation, it is not possible to keep the number of points per wave length N_x in the previously selected range $[40, 90]$ and keep square cells. Saw-tooth instabilities appear as KC exceeds 0.80 approximately. No filter was used in this case, as the main objective is to emphasize the limits of the method itself. Above a wave steepness of $H/\lambda = 2.6\%$, (*i.e.* $KC = 0.86$), no computation could remain stable after $\sim 2 - 3$ wave periods.

As a conclusion, even if a limitation in wave height is met, those results, correct up to third order, give us confidence on a case which is particularly challenging for volume field methods.

6.2 Rectangular barge, experiments and numerical comparison

In order to evaluate the NWT and its limitations with a (fixed) free surface piercing body, dedicated experiments were conducted in a wave flume at Centrale Marseille with a barge of rectangular cross-section.

6.2.1 Experimental set-up

The flume is 17 m long and 0.65 m wide. The water depth was set to $d = 0.509$ m. The body is a rectangular barge of draft 0.10 m for a length of 0.30 m (in the longitudinal direction of the flume) mounted on a 6-axis load cell measuring device. The width of the barge spans the width of the flume minus a small water gap of about 2 mm on both sides between the barge and the flume side-walls (fig. 18). A perforated metallic beach is placed at the end of the flume to dissipate the energy of the transmitted waves, and so to avoid reflection. The waves are generated with a flap type wave maker. The body is placed such that its front face is located at $x_b = 11.52$ m from the wave maker. 13 wave gauges are installed all along the wave flume. Unfortunately, the wave gauge dedicated to measure the run-up on the front face of the barge was found *a posteriori* to be defective. For some cases, a video of the experiment was recorded in the vicinity of the body, allowing to extract free surface profile and run-up at the front and rear faces of the barge.

Approximately 40 cases were tested in regular wave conditions, with varying wave period and height, as illustrated in fig. 19. A focus is made here on two periods: $T = 1.1$ s and $T = 1.5$ s. For the latter one, however, high wave steepness yielded high wave heights, resulting in dewetting and breaking, with a lot of turbulent effects, recirculations, and air entrainment. Thus, for this period, few relevant comparisons can be made with the potential model (which neglects all those effects).

6.2.2 Numerical set-up

The computational domain is automatically generated and meshed by the NWT depending on the case: its length is chosen as $\sim 12\lambda$ and relaxation zones are set in the inlet and outlet, over a length of 2λ . A second mesh is fitted to the body in order to ensure a precise computation of the flow dynamics in its vicinity. The mesh fitting the body is of breadth 0.30 m on each side (*i.e.* a total extent of 0.90 m). Both meshes communicate through previously described interpolation boundaries. Their free surface curves communicate through relaxation zones of length 0.14 m and constant function of



Figure 18: Photograph of the experimental set-up of the rectangular barge in the wave flume (the wave maker is on the left side).

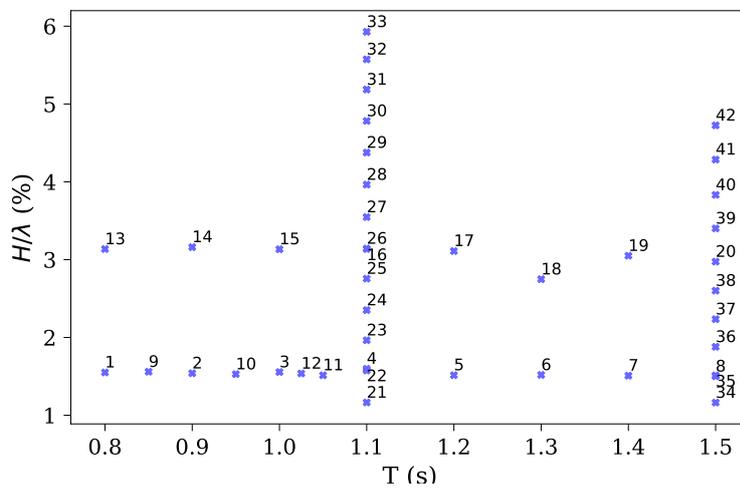


Figure 19: Overview of the conducted experiments in the $(T, H/\lambda)$ plane.

unitary weight. In order to interpolate a free surface from the other one, a 1D B-spline interpolation is used. Different discretizations of the fitted mesh were tested, without significant difference.

6.2.3 Numerical results and comparisons with experiments

For the wave period $T = 1.1$ s, the steepest cases (31-33) were found to numerically break down. This is due to an large run-down which leads to a dewetting at the bottom left corner of the barge. During the experiments, recirculations and turbulent effects were clearly visible for those steepest cases.

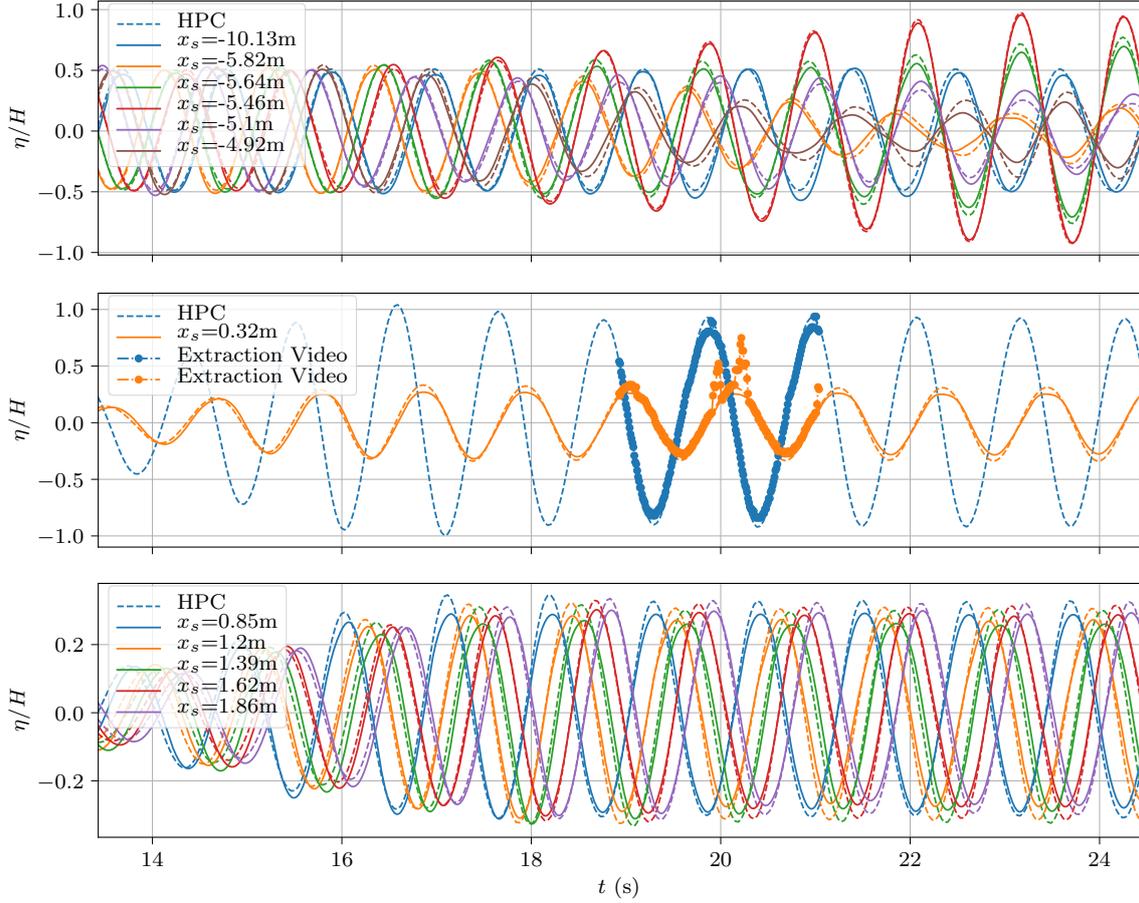


Figure 20: Comparisons of the free surface elevations recorded at different positions: HPC (dashed lines), experimental wave gauges (solid lines) and extracted elevation from the experimental video (circle markers). Wave gauges are divided in three groups (corresponding to subfigures): incident waves upstream (subfig 1), front and rear run-up (subfig 2), transmitted waves (downstream waves, subfig 3). Case 21: $T = 1.1$ s, $H/\lambda = 1.2\%$.

On figs. 20 and 21, wave gauge measurements are compared to the free surface elevation time series from the HPC method for cases 21 and 30 respectively. Note the time synchronisation between measurements and simulations was done on the first gauge only, and the determined phase shift was then applied to all the remaining gauges. This means that relative phases of the wave elevations

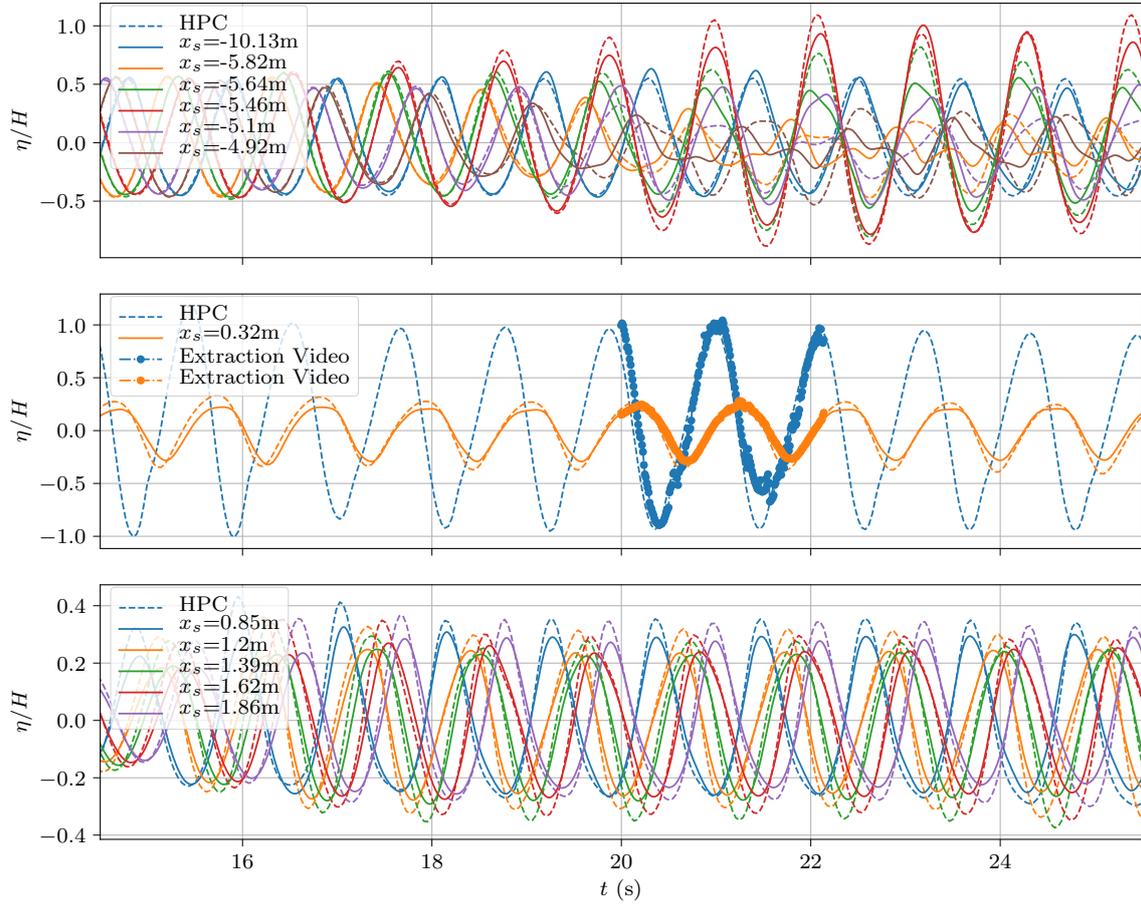


Figure 21: Comparisons of the free surface elevations recorded at different positions: HPC (dashed lines), experimental wave gauges (solid lines) and extracted elevation from the experimental video (circle markers). Wave gauges are divided in three groups (corresponding to subfigures): Incident waves upstream (subfig 1), Front and rear run-up (subfig 2), transmitted waves (downstream waves, subfig 3). Case 30: $T = 1.1$ s, $H/\lambda = 4.8\%$.

in the numerical simulations are represented adequately. More generally, a very good agreement is found on both wave amplitudes and phases, at the various locations along the wave flume. However, some discrepancies of the run-up at the front and rear faces of the barge as well as concerning the downstream wave elevations can be observed. Those discrepancies become more marked when the wave steepness increases. It can be observed that the HPC model tends to overestimate larger run-up events, and transmitted wave heights. The underlying causes of those discrepancies are thought to be related to the mathematical model itself. It is well known that non-dissipative models cannot correctly describe the flow in the vicinity of sharp angles even for linear incident waves. Thus, the present potential model is not perfectly appropriate to model the behavior of this type of flow. In practice, this effect is often counteracted by introducing a numerical lid in the vicinity of the body to numerically dissipate some energy (so trying to mimic dissipation due to viscosity). With such a method, the length and strength of the lid need to be tuned to match the expected result. This option was not tested here.

Reflection (C_r) and transmission (C_t) coefficients (in terms of wave height ratios) have been calculated on a 2-period time window ($t \in [23 \text{ s}, 25.2 \text{ s}]$) for the case 21 (fig. 20) using the nonlinear method of Andersen et al. (2017) with 5 wave probes on each side of the body. We obtain $C_r = 81\%$ from the experiments and $C_r = 79\%$ from the numerical predictions. In the same manner the transmission coefficients are in good agreement: $C_t = 60\%$ and 62% in the experiment and the numerical simulation respectively.

Obviously, the observed discrepancies in terms of run-up are expected to impact the loads exerted on the barge, mainly through the difference of run-up on the front and rear faces, impacting for example the elevation at which the pressure is set to the atmospheric pressure (dynamic boundary condition at the free surface).

Figure 22 shows at different time instants the free surface elevation in front of the body obtained from the HPC computation superimposed on snapshots from the experiments. On these pictures, we clearly denote aspects which cannot be taken into account in the potential model: during the rise of the water level, air entrainment and wave breaking take place, leading to important complex turbulent effects. At this stage, it is expected that the viscous effects play an important role. Although those dissipative effects are neglected in the HPC model, a relative fair agreement can be seen on the figure. The run-up is approximately correctly captured, as well as the reflected wave emerging during the elevation of the run-up. Moreover, the numerically computed reflected wave seems to be slightly faster than the experimental one. The difference can again be attributed to viscous effects, delaying the apparition of the reflected wave.

Time series of horizontal (F_x) and vertical (F_z) force components on the body are depicted on fig. 23 for the cases 21 and 30, respectively the most linear and nonlinear cases with $T = 1.1 \text{ s}$. The horizontal orange lines (symmetric with respect to the zero-force line) represent the amplitude of linear predictions of these loads. When the incoming waves are close to be linear (case 21), a good agreement is found between the linear and HPC models: the amplitudes are almost equal though the mean value computed with the HPC model slightly moves the vertical load extrema from the symmetric horizontal lines of the linear model result. Note that the effective (local) steepness, which determines the degree of nonlinearity, close to the body is approximately twice the incident steepness H/λ : only a small part of the energy is transmitted and thus the reflected wave adds to the incident wave locally. For a quantitative analysis, the amplitudes of incident and transmitted wave elevations can be compared in fig. 20.

When compared with experimental data, the computed loads are of approximately 15% higher magnitude. Note that, from an engineering point of view, the model is conservative in this case, and as expected, the differences in terms of run-up on the front and rear sides of the body lead to a slight overprediction of both the horizontal and vertical loads. This overprediction is seen for all cases, and

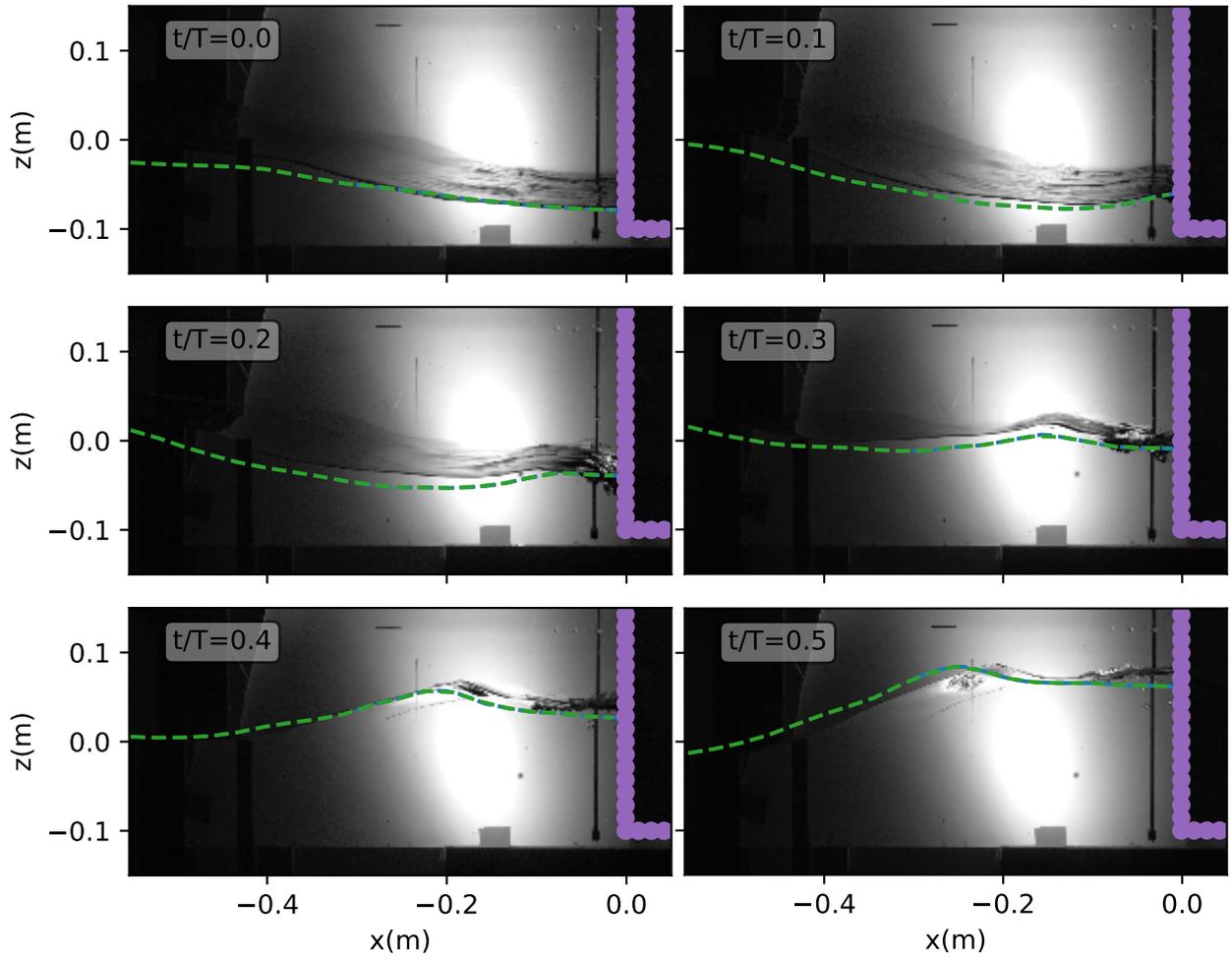


Figure 22: Case 30: $T = 1.1$ s, $H/\lambda = 4.8\%$. Comparison of the computed free surface elevation (green dashed lines) in front of the rectangular barge with the experiment at six different time instants. Incident waves come from the left.

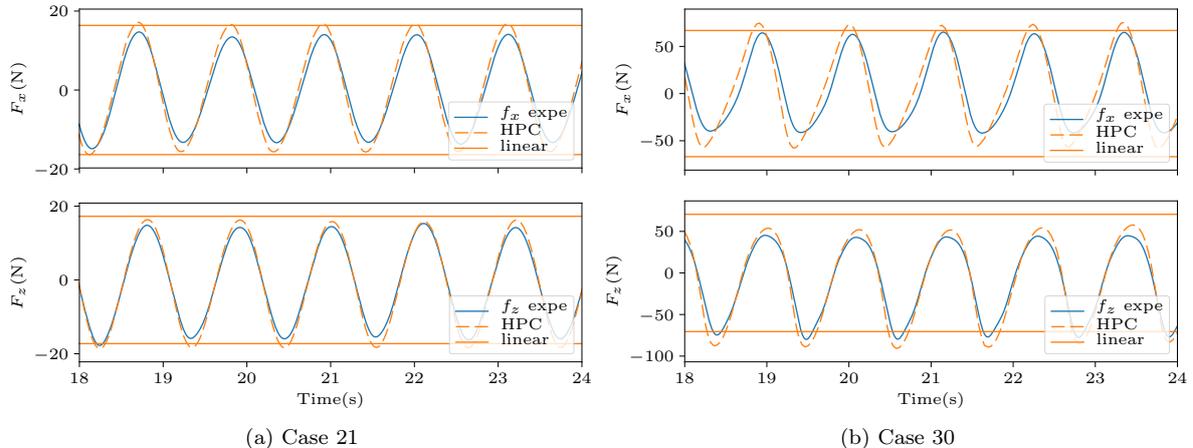


Figure 23: Cases 21 (left) and 30 (right): $T = 1.1$ s, $H/\lambda = 1.2\%$ and $H/\lambda = 4.8\%$. Comparison of time series of the hydrodynamic loads on the barge: experiments, linear results and HPC simulations.

consistently with approximately the same relative value. For steep incoming waves, the discrepancies between the linear and nonlinear models increase, as one could expect. In particular, the mean value (drift force) is no longer null. For case 30, one can note that, although the loads are overpredicted in magnitude, the shape of the time series of the loads (and thus the magnitude of the higher order harmonics of the loads) seems to be in good agreement with the experiment.

In order to compare the computed loads with the experimental ones more precisely, a Fourier decomposition of the time series is performed. The Fourier coefficients of this decomposition are shown for all the cases performed at $T = 1.1$ s on fig. 24. As expected the amplitudes of the first harmonic exhibit an over-estimation compared to the experimental data, whereas the mean values (\hat{f}_{xz}^0) are in correct agreement. However, the behavior of all the harmonics with the wave height is found to be exactly the same as measured during the experiments (experimental and computed lines are parallel). For the vertical loads, all harmonics are roughly consistent with the measurements. An important difference can be noticed for the third order harmonic of the vertical load for small steepness. First, it can be denoted that this harmonic is about 2 orders of magnitude lower than the first order harmonic, and even 3 orders magnitude lower than the hydrostatic force. Thus, an accurate computation as well as an accurate measurement of this small contribution is difficult to achieve. Moreover, the variations of the potential NWT values seem to be more closely fitted with a model of order 3 in wave steepness. This observation, together with the fact that numerical and experimental third order harmonics match for higher wave steepness, can be regarded as an indication of model's capability to resolve high-order nonlinear effects. Two reasons might explain these discrepancies: 1) either the measurements, the numerical method, or the post-processing method are not accurate enough to resolve or compute such small contributions, or 2) a physical effect bringing energy to the third harmonic plays a role not properly taken into account by the model.

For the horizontal loads, more discrepancies are visible. First order and drift force show an overestimation of about 15%. The amplitude of the third order harmonic seems to be well captured, but the amplitude of the second order one exhibits a larger over-estimation by about 70%. This effect is still under investigation, but the impact on the free surface of the viscous effects seems to be significant in this case. Note that the hydrostatic contribution originating from instantaneous wet free surface is one of the main contributions to the second order component. In the present set-up, this

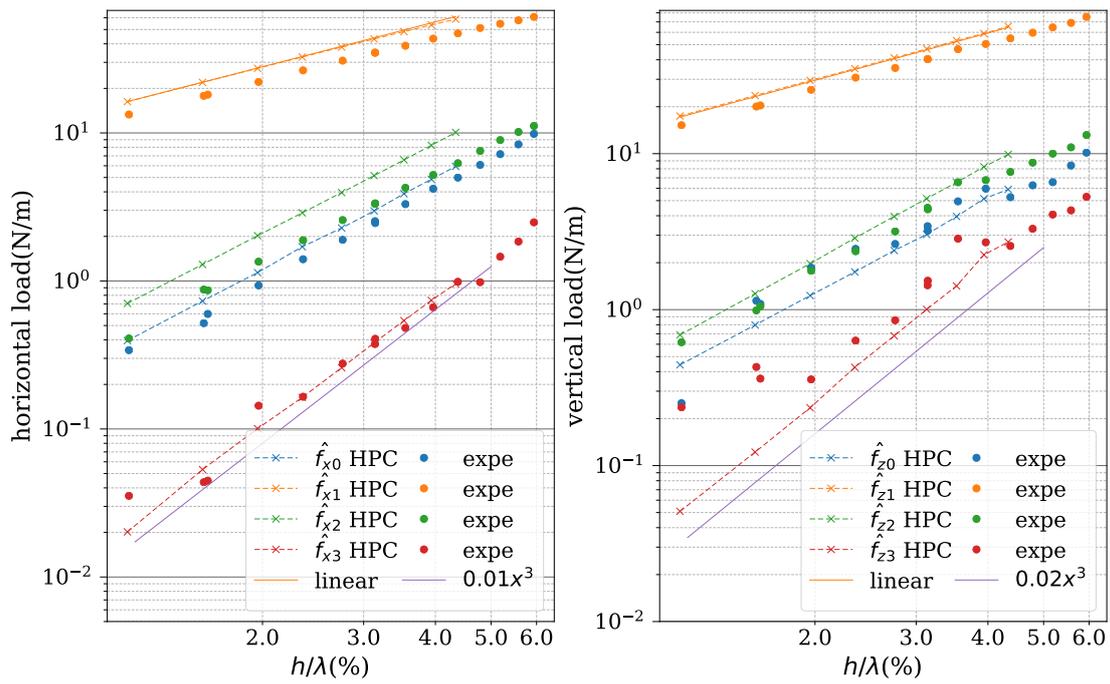


Figure 24: $T = 1.1$ s, Amplitudes of harmonics of the vertical and horizontal loads on the barge. HPC models (dashed lines), experimental results (dots) and linear transfer functions (solid orange line). A power function of order 3 is added to compare the increase rate (solid purple line).

term will play a role only on the horizontal load as the wet free surface changes only on the vertical walls of the barge. Thus, an important difference between the experiments and the numerical model is expected on this particular second order harmonic of the horizontal load: it was shown previously that our computations do not capture exactly the effective run-up elevations, probably due to the viscous effects not taken into account in the HPC model.

7 Summary of conclusions and outlook

In this work, an implementation of the Harmonic Polynomial Cell (HPC) method for solving the fully nonlinear potential flow problem in combination with an Immersed Boundary Method (IBM) to track the free surface was achieved and validated. The newly developed Numerical Wave Tank (NWT) allows to simulate a wide range of situations in ocean and coastal engineering applications, involving nonlinear waves and wave-body interaction. In this version, the bodies can be immersed or surface piercing, but have to be fixed. The numerical techniques have been described, in particular the implications of choosing to work with non-deforming grids which has led to the development of an accurate and robust IBM variant.

After investigating the convergence properties of the numerical methods with respect to a refinement in space and time on a standing wave of steepness $H/\lambda = 10\%$, freely evolving over up to 100 wave periods, it was shown that the NWT exhibits a high level of accuracy and other interesting features. First, a large range of time steps are shown to lead to stable computations over the target duration, although for small CFL C_o numbers the simulations tend to become unstable when no filter is applied. Most importantly, the recommended range of CFL was shown to be $C_o \in [1.5, 3.5]$, which is of great benefit as these large values permit to use large time steps, and so to reduce the computational burden. Regarding space discretization, the number of nodes per wavelength should lie in the range $[40, 90]$. On this case, results are found to be accurate and converging with an order of convergence comprised between 4 and 5, when refining either the spatial or temporal discretization. An original theoretical investigation of the evolution of the numerical error as a function of the simulation duration was also proposed to explain the observed trends.

In addition, the application of a mild Savitzky-Golay filter (typically SG(13,8)) was shown to extend significantly the broadness of the stability domain. More specifically, smaller CFL numbers could be used, leading to a reduction of the minimal achievable error.

Then, a variation of the standing wave case was set up to approach the maximum achievable wave height. Here, we simulated the reflection on a vertical wall of a nonlinear incident wave of steepness $H_I/\lambda_I = 7.5\%$, leading to a standing wave of steepness reaching $H/\lambda \approx 20\%$, *i.e.* close to the theoretical limit. The agreement of the present results with both experimental measurements and numerical results from another nonlinear NWT demonstrates the capabilities of the method.

A second Boundary Value Problem is introduced and solved on the time derivative of the potential in order to obtain an accurate estimation of the pressure in the fluid domain through the Bernoulli equation. This method ensures a precise computation of loads on bodies, which will further be needed to compute the movements of freely moving bodies.

An multi overlapping-mesh method was also developed as a way to compute the flow accurately in the vicinity of an object subjected to incoming waves, using a finer local body-fitted grid. In case the body pierces the free surface, a second free surface is added close to the body and evolves in the body fitted mesh. Specific strategies to couple the two (here fixed) grids and associated (time varying) free surface curves were proposed. The proposed double mesh technique ensures an accurate computation of complex flow patterns arising due to the presence of the body. In addition, the method introduced by [Zhu et al. \(2017\)](#) to treat the singular node problem was extended to internal body corners and was shown to yield small volume errors on a simple dedicated BVP.

Coupled with relaxation zones to both generate and absorb waves at extremities of the NWT, the method is shown to be accurate for several 2D cases. In particular, the double mesh approach give accurate results on the case of a fully submerged horizontal cylinder located very close to the free surface (flume experiments by [Chaplin, 1984](#)). We have shown that the mean (drift) vertical force on the cylinder and the amplitudes of the first three harmonics of the vertical force are properly estimated for a range of Keulegan-Carpenter number. The nonlinear capabilities of the model are thus confirmed.

As a final test, the NWT is used to simulate dedicated experiments performed in the wave flume at Centrale Marseille with a surface piercing rectangular barge. As the present model does not consider viscous effects, this case proved to be really challenging, in particular in high wave conditions and due to the presence of sharp corners at the bottom of the barge. In these conditions, indeed, dissipative terms cannot be neglected if one wants an accurate prediction of the physical values. Nevertheless, the NWT showed very good results for cases with low steepness incident waves (loads on the barge, run-up on the barge's vertical sides, transmitted and reflected waves). As the wave steepness increases, the simulations still reproduce the time evolution of free surface elevation, loads and run-up signals, but the differences in amplitude become more marked. In these conditions, errors versus the experiments can be up to 15% on the amplitude of the first harmonic of the loads. Higher harmonics tend also to be overestimated compared to the experimental results. We however would like to insist on the fact that some of the cases simulated here were extreme in the sense that waves started to break on the barge and we almost reached the dewetting of the front face of the barge (as shown in the pictures of the experiment in [fig. 22](#)). This being considered, the ability of the NWT to run on such cases and to deliver a correct (though slightly overestimated) order of magnitude of loads and run-up elevations is regarded as a valuable outcome of this study.

Future developments of this work will encompass four main aspects, namely: i) extension to 3D in order to simulate more realistic cases, ii) simulation of moving bodies in waves, iii) implementation a fully Lagrangian tracking method for the free surface nodes, and iv) coupling this potential NWT with local viscous models in the vicinity of bodies. This last development is meant to improve the simulation of cases where viscous effects are significant, as shown here for the rectangular barge.

Acknowledgments

The participation and help provided by Dr. Olivier Kimmoun during the experiments conducted in the wave flume at Centrale Marseille and presented in [§ 6.2](#) are gratefully acknowledged. The authors also express their gratitude to Pr. Bernard Molin for fruitful discussions during this work, and to the three anonymous reviewers whose comments helped improving earlier versions of this article.

Funding This work was supported by the Ecole Normale Supérieure (ENS) de Cachan (France) in the form of a *Ph.D.* grant attributed to Fabien Robaux. This research did not receive any other specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

A Convergence of a perturbed cosine in both period and amplitude at long time

In order to analyze the rate of convergence of our implementation of the HPC method on the case of the nonlinear standing wave presented in § 4, we consider that the theoretical (reference) solution for the free surface elevation at the center of the numerical wave tank, denoted $f_{th}(t)$, is a cosine function of period T_{th} and amplitude a_{th} :

$$f_{th}(t) = a_{th} \cos\left(2\pi \frac{t}{T_{th}}\right) \quad (31)$$

Assume then that the HPC solution for the free surface elevation at the same location can also be expressed as a cosine function:

$$f(t) = a \cos\left(2\pi \frac{t}{T}\right) \quad (32)$$

with amplitude a and period T , being close to a_{th} and T_{th} respectively.

Our numerical convergence tests have shown (see fig. 6) that the errors on amplitude and period of the surface elevation at the center of the tank decrease as C_o^4 . Lets generalize this with a more convenient parameter d and thus assume that the error both in amplitude and period decreases as d^4 . In this case $d \equiv C_o$ but the exact same reasoning can be applied with $d \equiv \delta x$. With this notation we have:

$$a = a_{th}(1 - e_A) = a_{th}(1 - f_A d^4) \quad (33)$$

$$T = T_{th}(1 - e_T) = T_{th}(1 - f_T d^4) \quad (34)$$

Note that for every representation in this article, we set $f_A = -5.5 \cdot 10^{-3}$ and $f_T = -5.4 \cdot 10^{-5}$. Those values are extracted from the convergence depicted fig. 6 with the aim of fitting the convergence of our HPC model on the standing wave for $d \equiv C_o$ and $t/T = 100$.

Lets define e the total relative error at a given time t :

$$e(t) = \frac{f_{th}(t) - f(t)}{a_{th}} \quad (35)$$

This error is the combined effect of both the error on amplitude and the error on period. A representation of the impact of e_A and e_T and their combined impact - with the assumption that they are constant in time - are shown in sketch fig. 25.

Developing this expression with our cosine models (eqs. (31) and (32)), and injecting the modeled expression of the error (eqs. (33) and (34)) yields an expression of the total error as a function of t and d .

$$e(t) = \cos\left(2\pi \frac{t}{T_{th}}\right) - \frac{a}{a_{th}} \cos\left(2\pi \frac{t}{T}\right) \quad (36)$$

$$e(t) = \left(\cos\left(2\pi \frac{t}{T_{th}}\right) - \cos\left(2\pi \frac{t}{T}\right)\right) + f_A d^4 \cos\left(2\pi \frac{t}{T}\right) \quad (37)$$

This error is computed for different times $\frac{t}{T_{th}}$ and shown on fig. 26 as a function of d . In the range $d \in [1.5, 2.5]$ a power interpolation is computed and added.

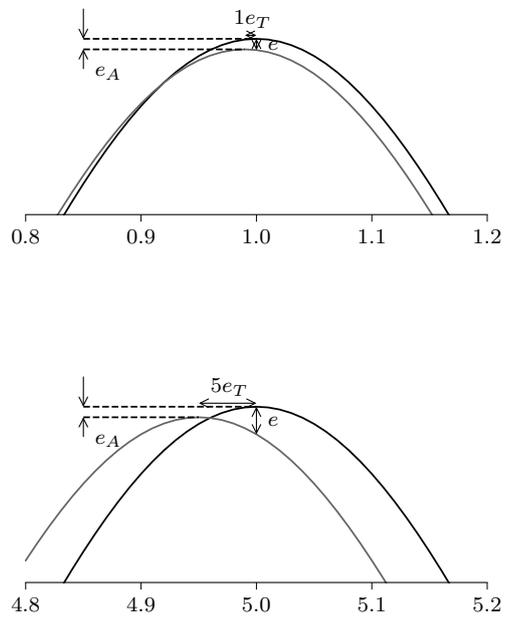


Figure 25: Sketch of the period, amplitude and total error of a perturbed cosine (grey) at two different whole periods. The assumption is made that e_A and e_T do not depend on time.

A.1 Taylor expansion

In this part, a Taylor expansion of the latter expression of the total error is done in the vicinity of a whole number of period t/T_{th} :

$$e(t/T_{th} \in \mathbb{N}) = (1 - \cos(2\pi \frac{t}{T})) + f_A d^4 \cos(2\pi \frac{t}{T}) \quad (38)$$

Let's first expand t/T using the modeled behavior of T when $f_T d^4 \rightarrow 0$ the given in the expression (eq. (34)):

$$2\pi \frac{t}{T} = 2\pi \frac{t}{T_{th}} \frac{1}{1 - f_T d^4} \quad (39)$$

$$2\pi \frac{t}{T} = 2\pi \frac{t}{T_{th}} (1 + f_T d^4 + f_T^2 d^8 + O(d^{16})) \quad (40)$$

$$2\pi \frac{t}{T} = 2\pi \frac{t}{T_{th}} + X \quad (41)$$

where $X = 2\pi \frac{t}{T_{th}} (f_T d^4 + f_T^2 d^8 + O(d^{16}))$. X also tends to 0, so it is possible to expand the cosine in the vicinity of its maximum (whole number of periods):

$$e(t/T_{th} \in \mathbb{N}) = (1 - [1 - X^2/2 + O(X^4)]) + f_A d^4 [1 - X^2/2 + O(X^4)] \quad (42)$$

$$e(t/T_{th} \in \mathbb{N}) = (2\pi^2 \frac{t^2}{T_{th}^2} f_T^2 d^8 + 2\pi \frac{t}{T_{th}} f_T^3 d^{12} + O(d^{16})) + f_A d^4 [1 - 2\pi^2 \frac{t^2}{T_{th}^2} f_T^2 d^8 + O(d^{12})] \quad (43)$$

At the end, we obtain the Taylor expansion of e at order 12:

$$e(t/T_{th} \in \mathbb{N}) = f_A d^4 + 2\pi^2 \frac{t^2}{T_{th}^2} f_T^2 d^8 + (2\pi \frac{t}{T_{th}} f_T^3 - 2f_A \pi^2 \frac{t^2}{T_{th}^2} f_T^2) d^{12} + O(d^{16}) \quad (44)$$

A.2 Representation and interpretation

Equation (22) is computed at $o(8)$ and $o(12)$ and represented as a function of d on fig. 26 (respectively labelled $e^{(8)}$ and $e^{(12)}$) at $\frac{t}{T_{th}} = 100$. The parameters are kept fixed to $f_A = 5.5 \cdot 10^{-3}$ and $f_T = 5.4 \cdot 10^{-5}$ so as to fit the obtained results depicted on fig. 6. Thus, we model the case where $N_x/\lambda = 30$ and the error is computed as the relative difference of the free surface elevation at the center point ($x/\lambda = 0.5$) with $d \equiv C_o$. Moreover, expression (eq. (35)) of e is also shown at different times and for different convergence parameter in the range $[1, 3.5]$.

For most lines a power regression is computed and added in the figure (dashed lines). For the sake of simplicity, the values of f_A and f_T are supposed to be independent of t/T_{th} and set from the observed convergence at t/T_{th} . In reality, this assumption is not verified.

Note that the order of convergence is 4 when $\frac{t}{T_{th}} = 0$ (*i.e.* $t = 0$). For $t \neq 0$ the asymptotic error when d converges to 0 is equal to the amplitude error. Mathematically, this is due to the square elevation when expanding the cosine. Thus, on fig. 26, the d^4 order model is not shown as it corresponds to the amplitude error.

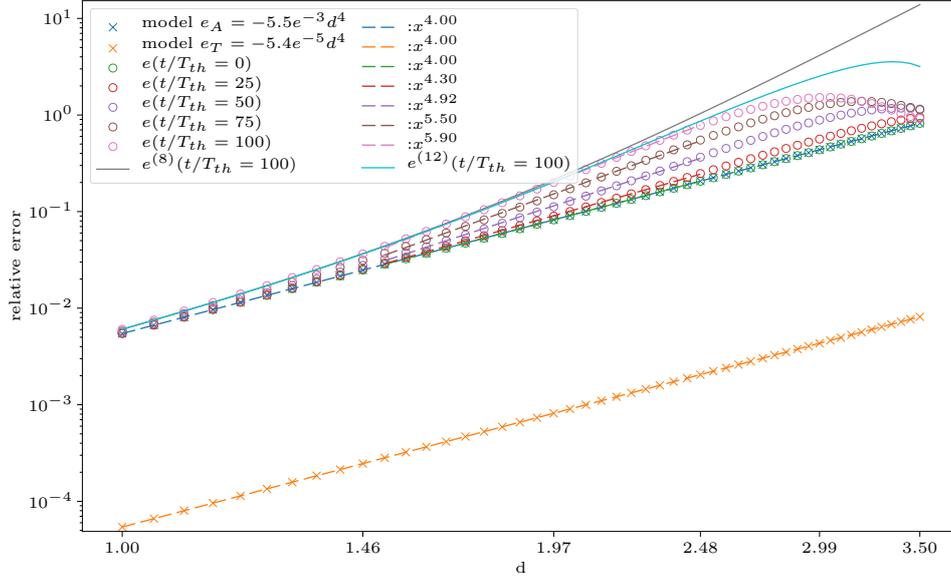


Figure 26: Relative error of models in period and amplitude (e_T and e_A respectively). Total relative error e after $\frac{t}{T_{th}}$ periods between two cosine (eq. (35)) as a function of d which parameterize the order 4 convergence between their amplitudes and period. $e^{(n)}$ is the Taylor expansion of e at order n

In order to compare the different regimes of convergence, the ratio between the 8th order and 4th order is computed

$$r_{8,4} = \frac{2\pi^2 \frac{t^2}{T_{th}^2} f_T^2}{f_A} d^4 \quad (45)$$

If f_A increases with time slower than $t^2 f_T^2$, then the order 8 will have a growing importance over time ($r_{8,4}$ increase). As f_A , and f_T were set to model the magnitude of the error after $t/T_{th} = 100$ and $d \equiv C_o$, the ratio presented above is computed at this time step for different parameter d : 8th order and 4th (*i.e.* $r_{8,4} = 1$) are of the same importance when $\frac{t}{T_{th}} = 100$ as soon as $d = 1.75$. For d as small as 1.00, the 8th order already represents $r_{8,4} = 10\%$ of the error. Thus, it is both mathematically and graphically predictable that the convergence in time in our standing wave case would be of order higher than 4 after $t/T_{th} = 100$ in our range of the C_o parameter given the convergence rate of the amplitude and period at this time step.

It is then also consistent that the orders of convergence seem to increase with $\frac{t}{T_{th}}$ for all other computations.

References

Andersen, T.L., Eldrup, M.R., Frigaard, P., 2017. Estimation of incident and reflected components in highly nonlinear regular waves. Coastal Eng 119, 51–64.

- Anslys, A., 2013. AQWA theory manual. ed. Canonsburg, PA 15317, USA .
- Baker, A.H., Jessup, E.R., Kolev, T.V., 2009. A simple strategy for varying the restart parameter in GMRES(m). *J. Comput. Appl. Math.* 230, 751–761. doi:[10.1016/j.cam.2009.01.009](https://doi.org/10.1016/j.cam.2009.01.009).
- Bardazzi, A., Lugni, C., Antuono, M., Graziani, G., Faltinsen, O.M., 2015. Generalized HPC method for the Poisson equation. *J. Comput. Phys.* 299, 630–648. doi:[10.1016/j.jcp.2015.07.026](https://doi.org/10.1016/j.jcp.2015.07.026).
- Bihs, H., Kamath, A., Alagan Chella, M., Aggarwal, A., Arntsen, Ø.A., 2016. A new level set numerical wave tank with improved density interpolation for complex wave hydrodynamics. *Comput. Fluids* 140, 191–208. doi:[10.1016/j.compfluid.2016.09.012](https://doi.org/10.1016/j.compfluid.2016.09.012).
- Chaplin, J.R., 1984. Nonlinear forces on a horizontal cylinder beneath waves. *J. Fluid Mech.* 147, 449. doi:[10.1017/s0022112084002160](https://doi.org/10.1017/s0022112084002160).
- Clamond, D., Fructus, D., Grue, J., Kristiansen, Ø., 2005. An efficient model for three-dimensional surface wave simulations. Part II: Generation and absorption. *J. Comput. Phys.* 205, 686–705. doi:[10.1016/j.jcp.2004.11.038](https://doi.org/10.1016/j.jcp.2004.11.038).
- Cointe, R., Geyer, P., King, B., Molin, B., Tramoni, B., 1991. Nonlinear and linear motions of a rectangular barge in a perfect fluid, in: *Naval Hydrodynamics, 18th Symposium*, p. 85.
- Dean, R., 1965. Stream function representation of nonlinear ocean waves. *J. Geophys. Res.* 70, 4561–4572. doi:[10.1029/jz070i018p04561](https://doi.org/10.1029/jz070i018p04561).
- Di Paolo, B., Lara, J.L., Barajas, G., Losada, Í.J., 2021. Wave and structure interaction using multi-domain couplings for Navier-Stokes solvers in OpenFOAM®. Part I: Implementation and validation. *Coastal Eng* 164, 103799. doi:[10.1016/j.coastaleng.2020.103799](https://doi.org/10.1016/j.coastaleng.2020.103799).
- Dombre, E., Harris, J.C., Benoit, M., Violeau, D., Peyrard, C., 2019. A 3D parallel boundary element method on unstructured triangular grids for fully nonlinear wave-body interactions. *Ocean Eng* 171, 505–518.
- Ducrozet, G., Bonnefoy, F., Le Touzé, D., Ferrant, P., 2012. A modified High-Order Spectral method for wavemaker modeling in a numerical wave tank. *Eur. J. Mech. B. Fluids* 34, 19–34. doi:[10.1016/j.euromechflu.2012.01.017](https://doi.org/10.1016/j.euromechflu.2012.01.017).
- Ducrozet, G., Engsig-Karup, A.P., Bingham, H.B., Ferrant, P., 2014. A non-linear wave decomposition model for efficient wave–structure interaction. Part A: Formulation, validations and analysis. *J. Comput. Phys.* 257, 863–883.
- Engsig-Karup, A.P., Bingham, H.B., Lindberg, O., 2009. An efficient flexible-order model for 3D nonlinear water waves. *J. Comput. Phys.* 228, 2100–2118. doi:[10.1016/j.jcp.2008.11.028](https://doi.org/10.1016/j.jcp.2008.11.028).
- Engsig-Karup, A.P., Eskilsson, C., Bigoni, D., 2016. A stabilised nodal spectral element method for fully nonlinear water waves. *J. Comput. Phys.* 318, 1–21.
- Engsig-Karup, A.P., Monteserin, C., Eskilsson, C., 2019. A Mixed Eulerian–Lagrangian Spectral Element Method for Nonlinear Wave Interaction with Fixed Structures. *Water Waves* 1, 315–342. doi:[10.1007/s42286-019-00018-5](https://doi.org/10.1007/s42286-019-00018-5).
- Fàbregas Flavià, F., McNatt, C., Rongère, F., Babarit, A., Clément, A.H., 2016. Computation of the diffraction transfer matrix and the radiation characteristics in the open-source BEM code NEMOH, in: *35th International Conference on Offshore Mechanics and Arctic Engineering, American Society of Mechanical Engineers, South Korea*. doi:[10.1115/omae2016-54130](https://doi.org/10.1115/omae2016-54130).

- Fedele, F., Lugni, C., Chawla, A., 2017. The sinking of the El Faro: Predicting real world rogue waves during Hurricane Joaquin. *Sci Rep* 7, 11188. doi:[10.1038/s41598-017-11505-5](https://doi.org/10.1038/s41598-017-11505-5).
- Feng, X., Wu, W., 2019. Generation of water waves using momentum source wave-maker applied to a RANS solver. *Mathematical Problems in Engineering* 2019, 1–11. doi:[10.1155/2019/1308960](https://doi.org/10.1155/2019/1308960).
- Fenton, J., 1988. The numerical solution of steady water wave problems. *Comput. Geosci.* 14, 357–368. doi:[10.1016/0098-3004\(88\)90066-0](https://doi.org/10.1016/0098-3004(88)90066-0).
- Fenton, J.D., 1999. Numerical methods for nonlinear waves, in: *Advances in coastal and ocean engineering*. WORLD SCIENTIFIC, pp. 241–324. doi:[10.1142/9789812797544_0005](https://doi.org/10.1142/9789812797544_0005).
- Grilli, S.T., Guyenne, P., Dias, F., 2001. A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom. *Int. J. Numer. Meth. Fluids* 35, 829–867. doi:[10.1002/1097-0363\(20010415\)35:7<829::AID-FLD115>3.0.CO;2-2](https://doi.org/10.1002/1097-0363(20010415)35:7<829::AID-FLD115>3.0.CO;2-2).
- Grilli, S.T., Horrillo, J., 1997. Numerical generation and absorption of fully nonlinear periodic waves. *J. Eng. Mech.* 123, 1060–1069. doi:[10.1061/\(asce\)0733-9399\(1997\)123:10\(1060\)](https://doi.org/10.1061/(asce)0733-9399(1997)123:10(1060)).
- Guerber, E., 2011. Modélisation numérique des interactions non-linéaires entre vagues et structures immergées, appliquée à la simulation de systèmes houlomoteurs. Ph.D. thesis. Université Paris-Est. Paris, France.
- Guerber, E., Benoit, M., Grilli, S.T., Buvat, C., 2012. A fully nonlinear implicit model for wave interactions with submerged structures in forced or free motion. *Eng. Anal. Boundary Elem.* 36, 1151–1163. doi:[10.1016/j.enganabound.2012.02.005](https://doi.org/10.1016/j.enganabound.2012.02.005).
- Hague, C.H., Swan, C., 2009. A multiple flux boundary element method applied to the description of surface water waves. *J. Comput. Phys.* 228, 5111–5128. doi:[10.1016/j.jcp.2009.04.012](https://doi.org/10.1016/j.jcp.2009.04.012).
- Hanssen, F.C.W., 2019. Non-linear wave-body interaction in severe waves. Ph.D. thesis. Norwegian University of Science and Technology. Trondheim, Norway.
- Hanssen, F.C.W., Bardazzi, A., Lugni, C., Greco, M., 2017a. Free-surface tracking in 2D with the harmonic polynomial cell method: Two alternative strategies. *Int J Numer Methods Eng* 113, 311–351. doi:[10.1002/nme.5615](https://doi.org/10.1002/nme.5615).
- Hanssen, F.C.W., Greco, M., 2021. A potential flow method combining immersed boundaries and overlapping grids: Formulation, validation and verification. *Ocean Eng* 227, 108841. doi:[10.1016/j.oceaneng.2021.108841](https://doi.org/10.1016/j.oceaneng.2021.108841).
- Hanssen, F.C.W., Greco, M., Faltinsen, O.M., 2017b. Wave-body interaction with overlapping structured grids in the HPC method, in: 32nd IWWWFB, dalian, china.
- Hanssen, F.C.W., Greco, M., Shao, Y., 2015. The harmonic polynomial cell method for moving bodies immersed in a Cartesian background grid, in: 34th International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers. doi:[10.1115/omae2015-41282](https://doi.org/10.1115/omae2015-41282).
- Higuera, P., Lara, J.L., Losada, I.J., 2013. Realistic wave generation and active wave absorption for Navier–Stokes models. *Coastal Eng* 71, 102–118. doi:[10.1016/j.coastaleng.2012.07.002](https://doi.org/10.1016/j.coastaleng.2012.07.002).
- Hu, Z.Z., Greaves, D., Raby, A., 2016. Numerical wave tank study of extreme waves and wave-structure interaction using OpenFoam®. *Ocean Eng* 126, 329–342. doi:[10.1016/j.oceaneng.2016.09.017](https://doi.org/10.1016/j.oceaneng.2016.09.017).

- Jacobsen, N.G., Fuhrman, D.R., Fredsøe, J., 2011. A wave generation toolbox for the open-source CFD library: Openfoam®. *Int. J. Numer. Meth. Fluids* 70, 1073–1088. doi:[10.1002/flid.2726](https://doi.org/10.1002/flid.2726).
- Jiao, J., Huang, S., 2020. CFD simulation of ship seakeeping performance and slamming loads in bi-directional cross wave. *JMSE* 8, 312. doi:[10.3390/jmse8050312](https://doi.org/10.3390/jmse8050312).
- Kemper, J., Windt, C., Graf, K., Ringwood, J.V., 2019. Development towards a nested hydrodynamic model for the numerical analysis of ocean wave energy systems, in: 13th European Wave and Tidal Energy Conference, Napoli, Italy. pp. 1–6.
- Kim, C., Clement, A., Tanizawa, K., 1999. Recent research and development of numerical wave tanks—a review. *Int. J. Offshore Polar Eng.* 9.
- Kim, S.H., Yamashiro, M., Yoshida, A., 2010. A simple two-way coupling method of BEM and VOF model for random wave calculations. *Coastal Eng* 57, 1018–1028. doi:[10.1016/j.coastaleng.2010.06.006](https://doi.org/10.1016/j.coastaleng.2010.06.006).
- Kim, S.Y., Kim, K.M., Park, J.C., Jeon, G.M., Chun, H.H., 2016. Numerical simulation of wave and current interaction with a fixed offshore substructure. *Int. J. Nav. Archit. Ocean Eng.* 8, 188–197. doi:[10.1016/j.ijnaoe.2016.02.002](https://doi.org/10.1016/j.ijnaoe.2016.02.002).
- Larsen, B.E., Fuhrman, D.R., Roenby, J., 2019. Performance of interFoam on the simulation of progressive waves. *Coastal Eng J* 61, 380–400.
- Lee, C.H., 1995. WAMIT theory manual. Massachusetts Institute of Technology, Department of Ocean Engineering.
- Liang, H., Faltinsen, O.M., Shao, Y.L., 2015. Application of a 2D harmonic polynomial cell (HPC) method to singular flows and lifting problems. *Appl. Ocean Res.* 53, 75–90. doi:[10.1016/j.apor.2015.07.011](https://doi.org/10.1016/j.apor.2015.07.011).
- Liang, H., Santo, H., Shao, Y., Law, Y.Z., Chan, E.S., 2020. Liquid sloshing in an upright circular tank under periodic and transient excitations. *Physical Review Fluids* 5, 084801.
- Lu, X., Chandar, D.D.J., Chen, Y., Lou, J., 2017. An overlapping domain decomposition based near-far field coupling method for wave structure interaction simulations. *Coastal Eng* 126, 37–50. doi:[10.1016/j.coastaleng.2017.04.009](https://doi.org/10.1016/j.coastaleng.2017.04.009).
- Luquet, R., Ferrant, P., Alessandrini, B., Ducrozet, G., Gentaz, L., 2007. Simulation of a TLP in waves using the SWENSE scheme, in: The Seventeenth International Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers, Lisbon, Portugal.
- Ma, Q.W., Yan, S., 2006. Quasi ALE finite element method for nonlinear water waves. *J. Comput. Phys.* 212, 52–72. doi:[10.1016/j.jcp.2005.06.014](https://doi.org/10.1016/j.jcp.2005.06.014).
- Ma, S., Hanssen, F.C.W., Siddiqui, M., Greco, M., Faltinsen, O.M., 2017. Local and global properties of the harmonic polynomial cell method: In-depth analysis in two dimensions. *Int J Numer Methods Eng* 113, 681–718. doi:[10.1002/nme.5631](https://doi.org/10.1002/nme.5631).
- Oggiano, L., Pierella, F., Nygaard, T.A., De Vaal, J., Arens, E., 2017. Reproduction of steep long crested irregular waves with CFD using the VOF method. *Energy Procedia* 137, 273–281. doi:[10.1016/j.egypro.2017.10.351](https://doi.org/10.1016/j.egypro.2017.10.351).

- Ogilvie, T.F., 1963. First- and second-order forces on a cylinder submerged under a free surface. *J. Fluid Mech.* 16, 451. doi:[10.1017/s0022112063000896](https://doi.org/10.1017/s0022112063000896).
- Philippe, M., Combourieu, A., Peyrard, C., Robaux, F., Delhommeau, G., Babarit, A., 2015. Introducing second order low frequency loads in the open-source boundary element method code nemoh, in: 11th European Wave and Tidal Energy Conference, Nantes, France.
- Pinkster, J.A., 1980. Low frequency second order wave exciting forces on floating structures. Ph.D. thesis. TU Delft. Delft, The Netherlands.
- Robaux, F., Benoit, M., 2021. Development and validation of a numerical wave tank based on the Harmonic Polynomial Cell and Immersed Boundary methods to model nonlinear wave-structure interaction. *Journal of Computational Physics* 446, 110560. doi:[10.1016/j.jcp.2021.110560](https://doi.org/10.1016/j.jcp.2021.110560).
- Saad, Y., 2003. Iterative methods for sparse linear systems. volume 82. Society for Industrial and Applied Mathematics. doi:[10.1137/1.9780898718003](https://doi.org/10.1137/1.9780898718003).
- Savitzky, A., Golay, M.J., 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 1627–1639.
- Shao, Y.L., Faltinsen, O.M., 2012. Towards efficient fully-nonlinear potential-flow solvers in marine hydrodynamics, in: 31st International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, Rio de Janeiro, Brazil. pp. 369–380. doi:[10.1115/omae2012-83319](https://doi.org/10.1115/omae2012-83319).
- Shao, Y.L., Faltinsen, O.M., 2014a. Fully-nonlinear wave-current-body interaction analysis by a harmonic polynomial cell method. *J. Offshore Mech. Arct. Eng.* 136, 031301. doi:[10.1115/1.4026960](https://doi.org/10.1115/1.4026960).
- Shao, Y.L., Faltinsen, O.M., 2014b. A harmonic polynomial cell (HPC) method for 3D Laplace equation with application in marine hydrodynamics. *J. Comput. Phys.* 274, 312–332. doi:[10.1016/j.jcp.2014.06.021](https://doi.org/10.1016/j.jcp.2014.06.021).
- Sobey, R.J., 1989. Variations on Fourier wave theory. *Int. J. Numer. Meth. Fluids* 9, 1453–1467. doi:[10.1002/flid.1650091203](https://doi.org/10.1002/flid.1650091203).
- Strand, I.M., Faltinsen, O.M., 2019. Linear wave response of a 2D closed flexible fish cage. *J. Fluids Struct.* 87, 58–83. doi:[10.1016/j.jfluidstructs.2019.03.005](https://doi.org/10.1016/j.jfluidstructs.2019.03.005).
- Tanizawa, K., 1996. Long time fully nonlinear simulation of floating body motions with artificial damping zone. *Journal of the Society of Naval Architects of Japan* 1996, 311–319.
- Tanizawa, K., 2000. The state of the art on numerical wave tank, in: Proceedings of 4th osaka colloquium on seakeeping performance of ships 2000, pp. 95–114.
- Tavassoli, A., Kim, M., 2001. Interactions of fully nonlinear waves with submerged bodies by a 2D viscous NWT, in: The Eleventh International Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers, Stavanger, Norway.
- Taylor, G.I., 1953. An experimental study of standing waves. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 218, 44–59.
- Tsai, C.P., Jeng, D.S., 1994. Numerical Fourier solutions of standing waves in finite water depth. *Appl. Ocean Res.* 16, 185–193. doi:[10.1016/0141-1187\(94\)90028-0](https://doi.org/10.1016/0141-1187(94)90028-0).

- Windt, C., Davidson, J., Schmitt, P., Ringwood, J., 2019. On the assessment of numerical wave makers in CFD simulations. *JMSE* 7, 47. doi:[10.3390/jmse7020047](https://doi.org/10.3390/jmse7020047).
- Wu, C.S., Young, D.L., Chiu, C.L., 2013. Simulation of wave–structure interaction by hybrid Cartesian/immersed boundary and arbitrary Lagrangian–Eulerian finite-element method. *J. Comput. Phys.* 254, 155–183. doi:[10.1016/j.jcp.2013.07.014](https://doi.org/10.1016/j.jcp.2013.07.014).
- Yan, S., Ma, Q.W., 2007. Numerical simulation of fully nonlinear interaction between steep waves and 2D floating bodies using the QALE-FEM method. *J. Comput. Phys.* 221, 666–692. doi:[10.1016/j.jcp.2006.06.046](https://doi.org/10.1016/j.jcp.2006.06.046).
- Zakharov, V., 1968. Stability of periodic waves of finite amplitude on the surface of a deep fluid. *J Appl Mech Tech Phys* 9, 190–194. doi:[10.1007/bf00913182](https://doi.org/10.1007/bf00913182).
- Zhu, W., Greco, M., Shao, Y., 2017. Improved HPC method for nonlinear wave tank. *Int. J. Nav. Archit. Ocean Eng.* 9, 598–612. doi:[10.1016/j.ijnaoe.2017.03.009](https://doi.org/10.1016/j.ijnaoe.2017.03.009).