Received: 23 February 2022 | Revised: 28 June 2022 | Accepted: 27 August 2022

DOI: 10.1002/nme.7115

RESEARCH ARTICLE

WILEY

A 3D fully-nonlinear potential-flow solver for efficient simulations of large-scale free-surface waves

Finn-Christian Wickmann Hanssen[1] | **Jens Bloch Helmers**[2] | **Marilena Greco**[1,3] | **Yanlin Shao**[4]

[1]Centre for Autonomous Marine Operations and Systems (NTNU AMOS), Department of Marine Technology, NTNU, Trondheim, Norway

[2]DNV, Ship Hydrodynamics, Oslo, Norway

[3]CNR-INM, Institute of Marine Engineering, Roma, Italy

[4]Technical University of Denmark, Department of Mechanical Engineering, Lyngby, Denmark

Correspondence
Finn-Christian Wickmann Hanssen, Centre for Autonomous Marine Operations and Systems (NTNU AMOS), Department of Marine Technology, NTNU, Trondheim, Norway.
Email: finn-christian.hanssen@ntnu.no

Funding information
Norges Forskningsråd; Research Council of Norway, Grant/Award Number: 223254-AMOS

Abstract

In the quest for a numerical method for surface waves and wave-induced effects applicable when linear or weakly nonlinear methods are insufficient, a three-dimensional numerical wave tank assuming fully-nonlinear potential-flow theory is proposed. When viscous-flow effects, breaking waves or other violent flow-phenomena are not of primary importance, potential-flow methods may have similar capability in capturing the involved physics as Navier–Stokes solvers while being potentially more accurate in handling wave-propagation mechanism and more computationally efficient. If made sufficiently accurate, efficient and numerically robust, fully-nonlinear potential flow models can therefore represent a powerful tool in the study of ocean waves and their interaction with marine structures, which is the main motivation behind the present work. The governing Laplace equation for the velocity potential is solved using the harmonic polynomial cell method, which is a field method giving high-order accuracy provided that the cells used to describe the water domain have no stretching or distortion. This can only be achieved in a grid with cubic cells, which leads to poor numerical efficiency unless measures are introduced to refine the grid locally. Here, to improve the efficiency using strictly cubic cells, an adaptive grid refinement technique is introduced. It is shown that this has the ability to improve the computational speed with a factor of up to 20 without sacrificing accuracy. Numerical results are shown to be in good agreement with highly accurate nonlinear reference solutions for regular and irregular waves of various steepness up to the limit of theoretical wave breaking. For long-crested irregular waves, significant discrepancies with a second-order theory for the crest-height distribution are identified, while the second-order theory appears to provide a better description of the crest height for the single short-crested irregular sea state simulated. Having demonstrated that the proposed numerical method accurately models nonlinear wave phenomena up to the limit of wave breaking, future work should seek to implement wave-body interaction capabilities. The adaptive grid refinement technique, which refines

the grid dynamically depending on the position of boundaries of interest, is developed with this application in mind. Except from providing a robust way of dealing with wave-body intersection points, extending the method to account for wave-body interactions should therefore involve limited difficulty.

KEYWORDS

adaptive grid refinement, harmonic polynomial cell, nonlinear waves, potential flow

1 | INTRODUCTION

Marine structures, whether fixed or floating, are subject to loading from stochastic waves. A thorough understanding of ocean-wave processes is needed to explain extreme observations such as rogue waves, where the most infamous example is the New Year wave recorded in the North Sea 1st January 1995.[1] Transfer of energy between different wave components due to nonlinear dispersion effects may play an important role in the formation of high wave crests in a stochastic sea state leading to rogue waves. Nonlinear wave effects can be important to predict accurate loads on a structure, for example, slamming loads related to the height of and kinematics under an approaching wave crest, or the slowly-varying excitation of a moored floater in a stochastic sea state. It is the analysis of such nonlinear wave- and wave-induced phenomena, where linear or weakly nonlinear methods may be insufficient, that fully nonlinear numerical methods become attractive. However, computational fluid dynamics (CFD) methods solving the Navier–Stokes (NS) equations are still widely considered impractical as an engineering tool for large-scale wave modeling due to their low computational efficiency, especially without the use of powerful computational clusters. Fully-nonlinear potential flow (FNPF) methods can on the other hand be significantly faster, and as demonstrated by Wang et al.,[2] may have similar accuracy as long as flow-separation, overturning waves or other violent or viscous phenomena are not of primary importance. Motivated by this, the objective of the present work is to develop a three-dimensional (3D) FNPF numerical wave tank (NWT) able to study large-scale free-surface wave scenarios efficiently. The governing Laplace equation is solved using the harmonic polynomial cell (HPC) field method proposed by Shao and Faltinsen[3,4] using high-order harmonic polynomials to represent the solution variables in the water domain. Partially because of this high-order representation, the HPC method can solve idealized problems such as a mixed Dirichlet–Neumann boundary-value problem (BVP) on a linearized domain to a specified accuracy more efficiently than other methods of lower order. Xu et al.[5,6] used the HPC method in a 2D NWT to provide kinematics in long-crested irregular waves, and applied these to estimate the wave loads on a moored floating wind turbine (FWT) through the Morison equation. Their studies, while not applicable for short-crested waves, showed a significant increase in surge motions and mooring-line forces when fully nonlinear instead of linear wave kinematics were used. This work does not only highlight the importance of nonlinear wave effects, but also demonstrates the utility of the HPC method in a practical application assuming 2D flow.

The HPC method's high-order accuracy is attractive in a NWT, among others because it means that the parts of the numerical domain where the water accelerations are small can be modeled using large computational cells, thereby limiting the total number of unknowns in the numerical problem. However, as shown in two dimensions (2D) by Ma et al.,[7] the accuracy of the standard HPC method requires that square non-skewed cells are used. To model wave- and wave-body interaction problems using uniform Cartesian grids with square cells, therefore, Hanssen et al.,[8] Hanssen[9] combined the HPC method with an immersed-boundary modeling technique for complex boundaries. They also used an overlapping-grid technique to refine the numerical solution locally in an efficient manner. In the present work, the immersed-boundary modeling is extended to 3D. Instead of using overlapping grids, however, an adaptive grid refinement (AGR) approach is implemented. It is noted that while the AGR shares many features with the 2D work by Tong et al.,[10] the two methods are developed independently. Only wave-propagation scenarios are considered here, but the NWT is developed with future extensions to include wave-body interactions in mind. Several additional authors have used the HPC method in 2D studies considering various problems in marine hydrodynamics, including Liang et al.,[11] Strand,[12] Robaux and Benoit[13] and Shen et al.[14] 3D applications are more limited, where the most relevant examples are the work by Shao and Faltinsen[4] for wave propagation in a narrow NWT and wave run-up around bottom-mounted cylinders using a vertically stretched free-surface fitted grid, and the work by Law et al.[15] simulating multi-directional waves generated

by wave paddles using an overset grid. In contrast, the present work documents the first time that the HPC method is combined with an immersed-boundary technique and AGR in 3D.

The rest of the article is organized as follows: Section 2 gives the theoretical background and formulation including how waves are generated and absorbed towards the NWT outer boundaries. In Section 3, the numerical modeling is described with particular focus on the immersed-boundary technique for the free-surface. The properties of the local numerical solution in a single cell subject to stretching and skewness, and the effect of introducing an immersed boundary, are investigated in Section 4. The implementation of the AGR is explained in detail in Section 5, before numerical wave-propagation cases including regular waves and long- and short-crested irregular waves are investigated in Section 6. The article is concluded in Section 7.

2 | THEORY AND FORMULATION

We are interested in the generation and simulated time evolution of gravity waves within the framework of potential-flow theory. This excludes strong plunging-breaking phenomena, while spilling breakers can be modeled by incorporating a local breaking model like in Hanssen et al.[8] Mathematically, the scenario can be described by a velocity potential φ satisfying a hydrodynamic BVP governed by the Laplace equation and with mixed Dirichlet and Neumann boundary conditions, that is,

$$\begin{aligned} \nabla^2\varphi &= 0 \quad \text{in} \quad \Omega, \\ \varphi &= \hat{\varphi} \quad \text{on} \quad S_{FS}, \\ \nabla\varphi \cdot \boldsymbol{n} &= U_n \quad \text{on} \quad S_{TS\cup SB}, \end{aligned} \tag{1}$$

Here, $\hat{\varphi}$ represents the velocity potential on the free surface, U_n the normal velocity on the Neumann boundaries and $\boldsymbol{n}$ a normal vector pointing from the Neumann boundaries into the water. On fixed, impermeable walls, $U_n = 0$, while on the surfaces where waves are generated, U_n is given by some reference solution. Hence, on the parts of S_{TS} where waves are generated, U_n represents a flux condition. On the flat seabed S_{SB}, $U_n = 0$ is always enforced. Figure 1 indicates the various surfaces involved in the BVP and the global coordinate system. Hereafter, $\boldsymbol{x} = (x, y, z)$ and $\tilde{\boldsymbol{x}} = (x, y)$ represent the space and horizontal coordinates, respectively, of a generic point defined in such coordinate system.

The pressure in the water relative to the atmospheric pressure is given by the Bernoulli equation,

$$p(\boldsymbol{x}, t) = -\rho\left(\frac{\partial\varphi}{\partial t} + \frac{1}{2}|\nabla\varphi|^2 + gz\right), \tag{2}$$

where ρ is the mass density of water and g is the acceleration of gravity. The time derivative $\partial\varphi/\partial t$ satisfies a BVP formally similar to (1) for φ, that is,

$$\begin{aligned} \nabla^2\varphi_t &= 0 \quad \text{in} \quad \Omega, \\ \varphi_t &= \hat{\varphi}_t \quad \text{on} \quad S_{FS}, \\ \nabla\varphi_t \cdot \boldsymbol{n} &= U_{n,t} \quad \text{on} \quad S_{TS\cup SB}, \end{aligned} \tag{3}$$

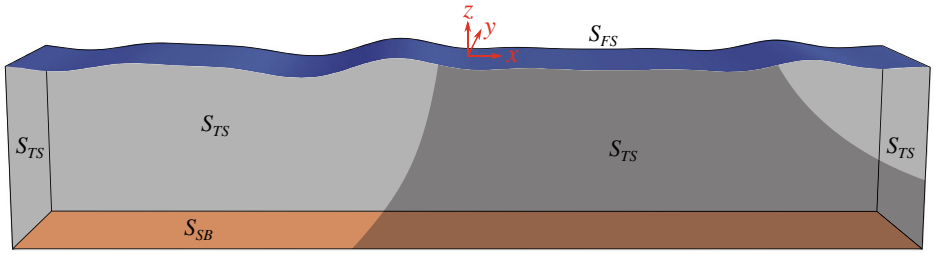

FIGURE 1 Definition of hydrodynamic BVP for a NWT with a global $Oxyz$ coordinate system with origin in the still-water level in the center of the tank. S_{SF} is the free surface, S_{TS} are tank sides and S_{SB} is the seabed.

where $U_{n,t} = \partial U_n/\partial t$. Evaluating $\partial\varphi/\partial t$ is mainly of interest when studying wave-structure interactions, where the pressure is integrated over the wet part of the structure to obtain wave loads. Although not included in the present work, later application to wave-structure interactions is relevant, and hence the numerical solution strategy for $\partial\varphi/\partial t$ is addressed.

With the free-surface elevation and potential in a point $\tilde{\boldsymbol{x}} = (x, y)$ denoted as $\zeta(\tilde{\boldsymbol{x}}, t)$ and $\hat{\varphi}(\tilde{\boldsymbol{x}}, t)$, respectively, the time evolution of the free surface is governed by the kinematic and dynamic free-surface conditions

$$\begin{aligned} \frac{\delta\zeta}{\delta t} &= \frac{\partial\varphi}{\partial z} - \tilde{\nabla}\varphi \cdot \tilde{\nabla}\zeta, \\ \frac{\delta\hat{\varphi}}{\delta t} &= -\frac{1}{2}|\nabla\varphi|^2 - g\zeta + \frac{\delta\zeta}{\delta t}\frac{\partial\varphi}{\partial z}, \end{aligned} \tag{4}$$

where $\tilde{\nabla} = (\partial/\partial x, \partial/\partial y)$. The equations are written in semi-Lagrangian form, where $\delta/\delta t$ is a Lagrangian time derivative following the vertical motion of a point on the free surface in the global coordinate system. This implies that the free surface is assumed to be a single-valued function of $\tilde{\boldsymbol{x}}$. The free surface is numerically tracked in time by following massless markers whose position and velocity potential are obtained using the free-surface conditions (4) in a standard explicit fourth-order Runge–Kutta scheme.

Instead of modeling physical wavemakers to generate waves such as done in 2D by Hanssen et al.[8] and in 3D by Liang et al.,[16] waves are here generated at fixed control surfaces. For this method to be accurate and numerically robust, it is found necessary to combine it with a relaxation-zone technique used in well-known codes such as OceanWave3D,[17] the Waves2Foam toolbox for OpenFOAM[18] and REEF3D::FNPF.[19] Consider a variable $\chi(\tilde{\boldsymbol{x}}, t)$ that may represent either $\zeta(\tilde{\boldsymbol{x}}, t)$ or $\hat{\varphi}(\tilde{\boldsymbol{x}}, t)$. Each time the free surface is modified in the time-integration scheme, the following correction is applied to markers inside a relaxation zone:

$$\chi_{\text{num}}(\tilde{\boldsymbol{x}}, t) = \alpha_r \chi^*_{\text{num}}(\tilde{\boldsymbol{x}}, t) + (1 - \alpha_r)\chi_{\text{ref}}(\tilde{\boldsymbol{x}}, t). \tag{5}$$

χ^*_{num} and χ_{num} are the values of χ in the numerical solution before and after applying the relaxation-zone correction, respectively, and the reference solution χ_{ref} is typically given by an analytical wave theory. To avoid diffraction and reflection effects in a fully nonlinear framework, such as the one considered here, the reference solution should provide an accurate representation of nonlinear wave properties. Relaxation zones may also be used to dissipate wave energy in the same way as a traditional numerical beach, which implies $\chi_{\text{ref}} = 0$. Using relaxation zones to eliminate waves is advantageous as it guarantees that the free surface is brought to rest at the end of the relaxation zone, but on the other hand, one should carefully choose the relaxation zone parameters so that wave reflections from the relaxation zone itself are minimized.

The relaxation-zone coefficient α_r is defined as

$$\alpha_r = \begin{cases} 1 - \frac{\exp(\bar{r}^\beta) - 1}{\exp(1) - 1} & \text{for} \quad \bar{r} \in [0, 1] \\ 1 & \text{for} \quad \bar{r} > 1. \end{cases} \tag{6}$$

β is taken as 3.5, similar to the value used by Jacobsen et al.[18] For a point $\boldsymbol{x}_p$, $\bar{r}$ is the normalized minimum horizontal distance to the points $\boldsymbol{x}_{S_{TS}}$ along the tank side associated with the relaxation zone, that is,

$$\bar{r}(\boldsymbol{x}_p) = \frac{1}{L_r} \min\left(\left\|\tilde{\boldsymbol{x}}_p - \tilde{\boldsymbol{x}}_{S_{TS}}\right\|\right), \tag{7}$$

where L_r is the length of the relaxation zone. As illustrated by Figure 2, this definition of $\bar{r}$ gives a smooth variation of α_r also with relaxation zones along multiple adjacent tank side walls. In the present work, L_r is generally taken between 1 and 1.5 times the characteristic wavelength. In wave-eliminating relaxation zones where $\chi_{\text{ref}} = 0$, studies with regular waves indicate that a value of two times the characteristic wavelength is more appropriate, giving similar performance as standard numerical beaches such as the one applied by Greco.[20]

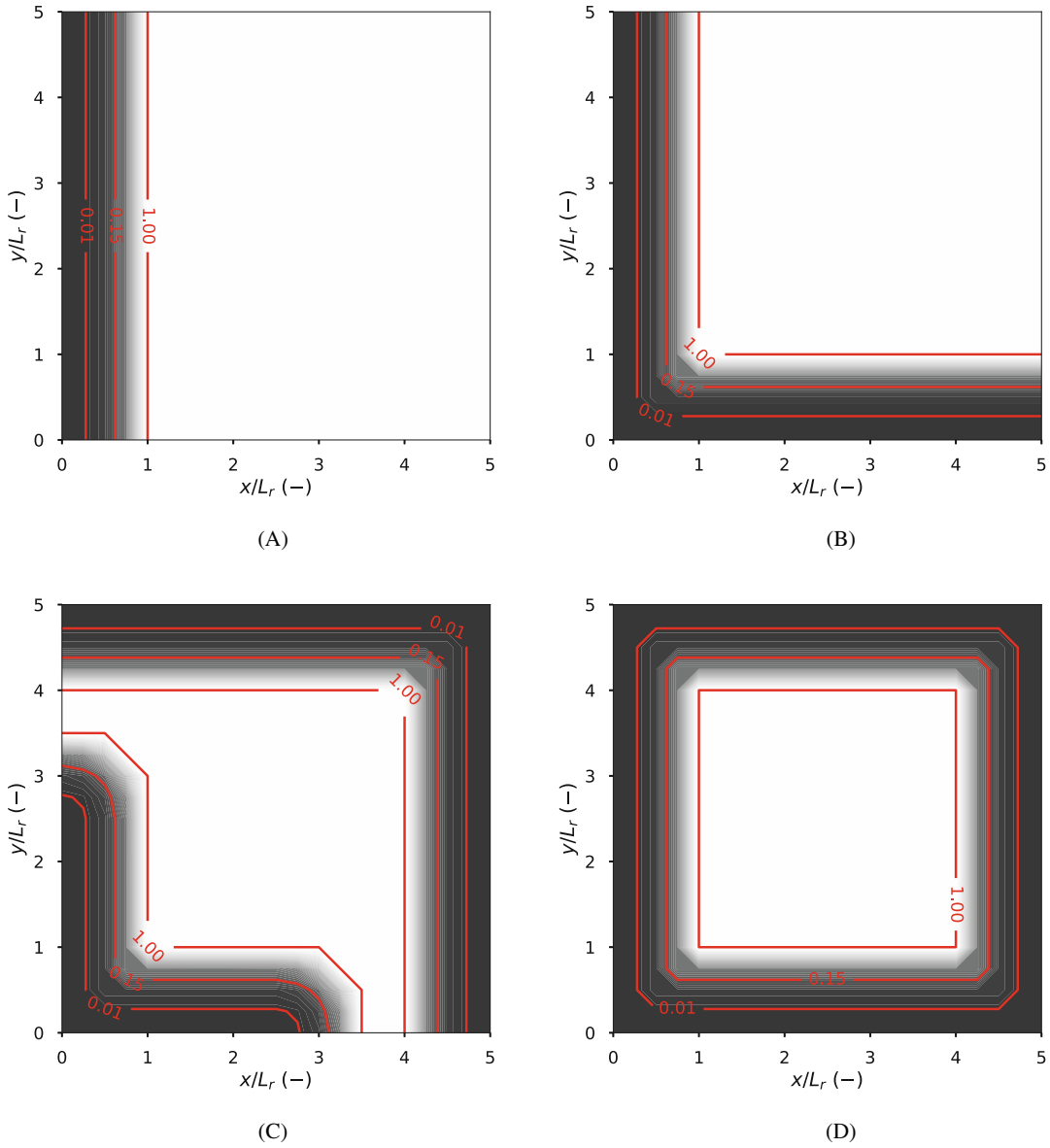

FIGURE 2 Examples showing α_r plotted for different relaxation-zone configurations on a square domain: (A) One whole surface, (B) two whole surfaces, (C) two whole and two partial surfaces, (D) four whole surfaces.

For points $\boldsymbol{x}_{S_{TS}}$ along the tank sides where $\bar{r} < 1$, the normal velocity for the Neumann boundary condition defined in Equation (1) is

$$U_n(\boldsymbol{x}_{S_{TS}}, t) = (1 - \alpha_r)\nabla\varphi_{\text{ref}}(\boldsymbol{x}_{S_{TS}}, t) \cdot \boldsymbol{n}. \tag{8}$$

For $\boldsymbol{x}_{S_{TS}}$ in regions with $0 < \alpha_r < 1$, such as found in Figure 2C along the sides with $x/L_r = 0$ and $y/L_r = 0$, equation (8) represents a weighted average between the normal velocity given by the reference solution and zero normal velocity. For wave-eliminating relaxation zones, Equation (8) gives $U_n = 0$ for any value of α_r.

3 | BASIC NUMERICAL MODELING

The hydrodynamic BVP defined in Equation (1) is solved numerically using the HPC method, representing the velocity potential in overlapping cells with high-order polynomials. Shao and Faltinsen[4] demonstrated that the method can solve

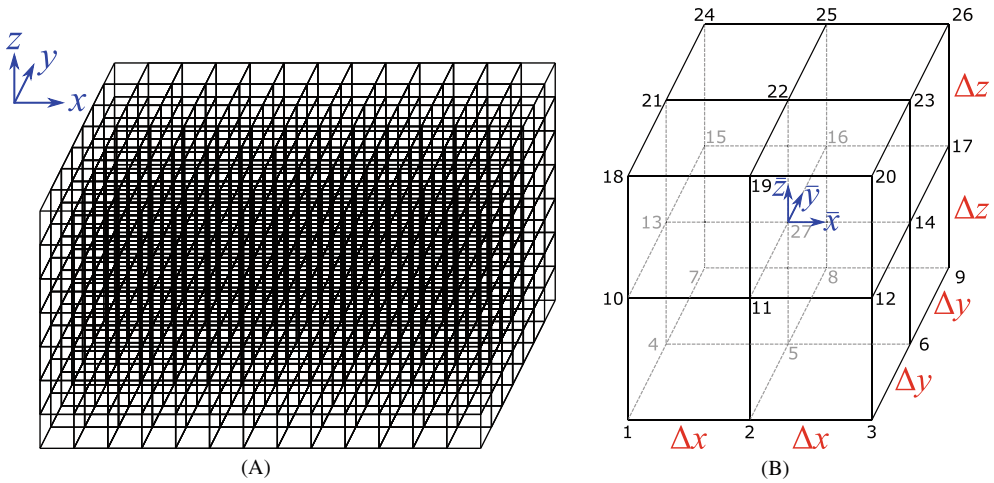

FIGURE 3 (A) A Cartesian grid with uniform grid spacing Δx, Δy, and Δz and overlapping cells defined in a global $Oxyz$ coordinate system. (B) Each cell has 26 boundary nodes with local indices $1-26$, one interior node with local index 27 and a local $o\overline{xyz}$ coordinate system with origin in the interior node.

a mixed Dirichlet–Neumann BVP on a cubic domain to a given accuracy significantly more efficient than a boundary element method (BEM). However, as it will be discussed later, maximizing the accuracy of the HPC method requires that certain modeling guidelines are adopted.

The main principles of the HPC method are explained with reference to Figure 3, where subfigure (A) illustrates a computational domain defined in a global $Oxyz$ coordinate system represented with a Cartesian grid. The number of nodes in x, y, and z-direction, respectively, is denoted N_x, N_y, and N_z. Adjacent nodes define cells with 26 boundary nodes and one interior node as shown in Figure 3B. The boundary nodes have local indices $1-26$, while the interior node, which is a boundary node in overlapping cells, has index 27. In each cell, a local $o\overline{xyz}$ coordinate system is defined with origin in the interior node. A point with global coordinates $\boldsymbol{x} = (x, y, z)$ has local cell coordinates $\overline{\boldsymbol{x}} = (\overline{x}, \overline{y}, \overline{z}) = \boldsymbol{x} - \boldsymbol{x}_{27}$ where $\boldsymbol{x}_{27} = (x_{27}, y_{27}, z_{27})$ are the global coordinates of the cell's interior node. The velocity potential in any point inside the cell is represented as

$$\varphi\left(\overline{\boldsymbol{x}}\right) = \sum_{i=1}^{26}\left[\sum_{j=1}^{26} c_{j,i} f_j\left(\overline{\boldsymbol{x}}\right)\right]\varphi_i; \quad \overline{\boldsymbol{x}} \in \{|\overline{x}| \le \Delta x \wedge |\overline{y}| \le \Delta y \wedge |\overline{z}| \le \Delta z\}, \tag{9}$$

where $\varphi_i, i = 1, \ldots, 26$ are the values of the velocity potential in the boundary nodes, $f_j, j = 1, \ldots, 26$ are the same harmonic polynomials as used by Shao and Faltinsen[4] and $c_{j,i}$ is an element of the inverse of the matrix $\boldsymbol{D}$ with elements $d_{i,j} = f_j(\overline{\boldsymbol{x}}_i)$. Taking the gradient of Equation (9) gives the velocity:

$$\nabla\varphi\left(\overline{\boldsymbol{x}}\right) = \sum_{i=1}^{26}\left[\sum_{j=1}^{26} c_{j,i} \nabla f_j\left(\overline{\boldsymbol{x}}\right)\right]\varphi_i; \quad \overline{\boldsymbol{x}} \in \{|\overline{x}| \le \Delta x \wedge |\overline{y}| \le \Delta y \wedge |\overline{z}| \le \Delta z\}. \tag{10}$$

In the interior node of a cell, where $\overline{\boldsymbol{x}} = \boldsymbol{0}$, we have that $f_1 = 1$ and $f_j = 0, j > 1$. Hence, Equation (9) gives

$$-\varphi_{27} + \sum_{i=1}^{26} c_{1,i}\varphi_i = 0. \tag{11}$$

Since the grid consists of overlapping cells, Equation (11) ensures continuity of φ through the grid. Replacing φ with φ_t, Equations (9)–(11) apply directly also for the time derivative of the velocity potential.

For the grid in Figure 3A, enforcing the continuity Equation (11) for interior grid nodes together with Equation (9) on Dirichlet boundaries and Equation (10) on Neumann boundaries, gives an algebraic global matrix equation for the

velocity potential that can be written as

$$\boldsymbol{A}_{N\times N} \cdot \boldsymbol{\varphi}_{N\times 1} = \boldsymbol{b}_{N\times 1}. \tag{12}$$

Here, $\boldsymbol{A}$ is a coefficient matrix, $\boldsymbol{\varphi}$ is a vector with the velocity potential in all N grid nodes and $\boldsymbol{b}$ is the right-hand side vector. $\boldsymbol{A}$ is sparse with nonzero entries confined to a narrow region around the matrix's diagonal, and is built and stored as a compressed sparse column matrix where only the elements that differ from zero are specified. In the grid refinement strategy described later, some off-diagonal terms are introduced, but the matrix can still be considered diagonally dominated and sparse. Equation (12) can therefore be solved numerical for the unknown vector $\boldsymbol{\varphi}$ in an efficient manner using a sparse, iterative solver with preconditioning. In the present work, we use the *BiCGStab* solver and *spilu* preconditioner included in the *SciPy* sparse linear algebra library for Python. Setting the tolerance, which defines the convergence criterion for the iterative solver, to $5 \cdot 10^{-9}$ and the preconditioner's drop tolerance, which controls the LU decomposition of $\boldsymbol{A}$, to $1 \cdot 10^{-5}$ is generally found to give robust solutions of Equation (12). Using a drop tolerance larger than $1 \cdot 10^{-5}$ is not advised, which is in contrast to the 2D implementation used by Hanssen et al.[8,21] where a value of $1 \cdot 10^{-4}$ was found sufficient.

A similar global matrix equation as (12), sharing the same coefficient matrix $\boldsymbol{A}$, applies also for the time derivative of the velocity potential, so that the preconditioner matrix can be reused. As a consequence, the global solution vector $\boldsymbol{\varphi}_t$ for the time derivative of the velocity potential in all grid nodes is found without significantly increasing the computational time.

3.1 | Free-surface modeling

Generating boundary-fitted grids with adequate cell shapes for complex, non-stationary boundaries can be a difficult and time-consuming task in itself. Here, it also means that the coefficients $c_{j,i}$ in Equation (9) must be recomputed each time a cell changes, which for grids with many cells influences the efficiency of the numerical method. Moreover, from 2D it is known that the accuracy of the HPC solution is strongly reduced in grids with stretched or distorted cells,[7] and it will later be demonstrated that this also applies in 3D. To avoid the challenges related to boundary-fitted grids, the immersed-boundary method (IBM) used by Hanssen et al.,[8] Hanssen[9] is here extended from 2D to 3D. The free surface is described by a set of massless markers with horizontal coordinates $\tilde{\boldsymbol{x}}$ equal to the (x, y) coordinates of the Cartesian grid nodes, and the markers' vertical position and velocity potential are tracked in time using Equations (4) in a fourth-order Runge–Kutta scheme. Since the semi-Lagrangian free-surface conditions only allow the markers to move in z-direction, they are uniformly distributed in x and y-direction at all times as long as the grid is uniform.

The markers define discrete points along the free-surface Dirichlet boundary in the hydrodynamic BVP in Equation (1), where Equation (9) is used to enforce Dirichlet conditions for the velocity potential in cells that contain the markers. These cells, denoted as free-surface cells, contain nodes above the free surface, denoted as ghost nodes. The water nodes and ghost nodes are referred to as *active* nodes, whereas the nodes that are above the free surface and not part of free-surface cells are referred to as *inactive*. As an example, Figure 4 shows a 2D projection in the xz-plane of the active nodes in an immersed-boundary representation of the NWT illustrated in Figure 1. The active nodes correspond to the unknowns in the algebraic global matrix equation (12), where the inactive nodes are disregarded.

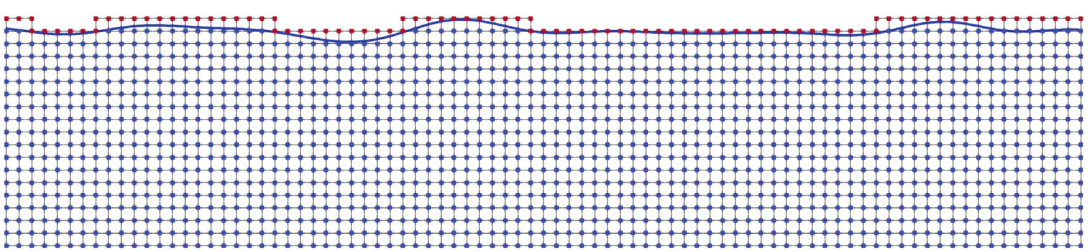

FIGURE 4 xz-projection of the free surface of a long crested wave profile immersed in a uniform grid. The blue nodes are water nodes and the red nodes are ghost nodes above the free surface. Inactive nodes above the ghost nodes are not shown.

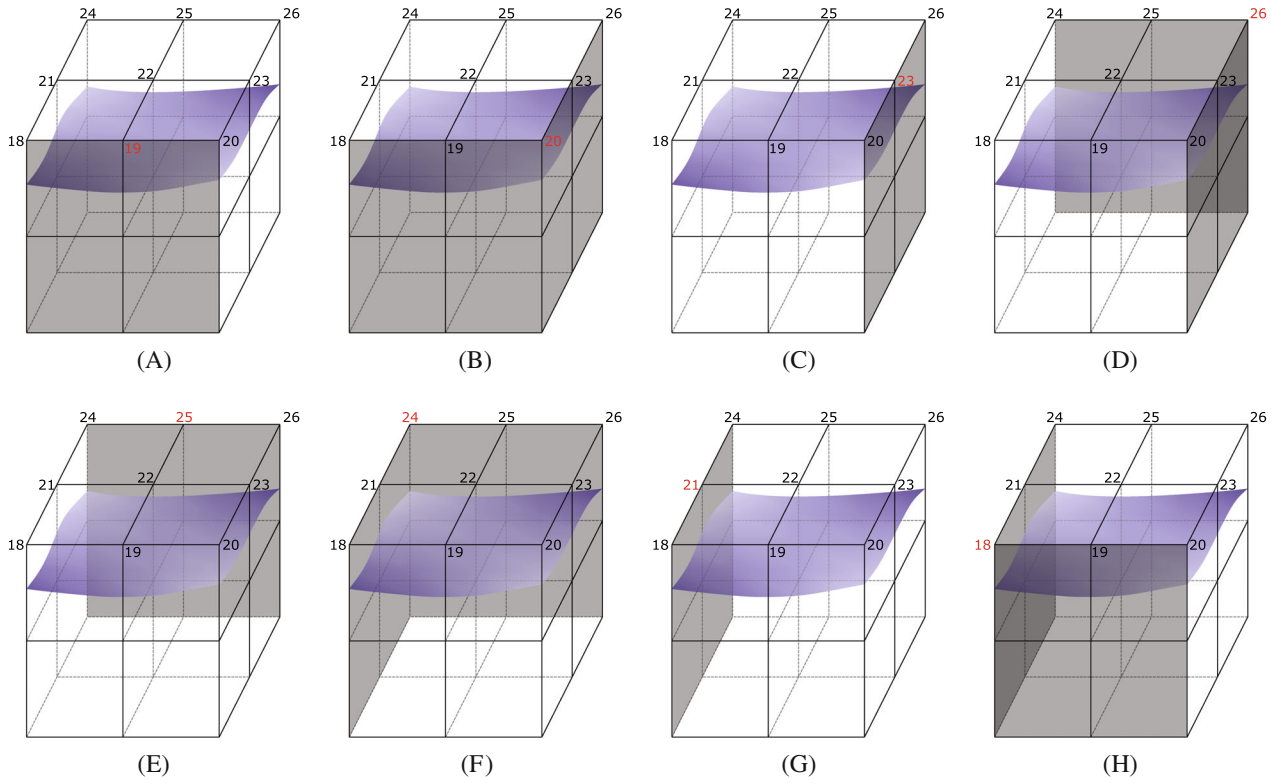

FIGURE 5 Free-surface cell to be associated with the red ghost node intersecting one or two physical boundaries indicated in grey (Step 1)

The process of identifying active and inactive nodes and free-surface cells consists of four steps. The two first steps are strictly necessary to obtain a global matrix system with equal number of unique equations and unknowns, whereas the two last steps are introduced to improve the condition of the global coefficient matrix and thereby the solvability of the matrix equation.

Step 1: Identifying initial ghost nodes and free-surface cells

The global coordinate system is defined with origin in the middle of the domain with dimensions L_x, L_y, and L_z, so that the flat seabed has vertical coordinate $z = -0.5L_z$. For each marker with coordinates $\boldsymbol{x}_m = (x_m, y_m, z_m)$, the first node with coordinates $x = x_m, y = y_m$, and $z \geq z_m$ is identified as a ghost node. The nodes with the same (x, y) coordinates below the ghost node are per definition below the free surface and therefore defined as water nodes, while the nodes above the ghost node are defined as inactive. If the ghost node is not along any of the vertical domain boundaries, that is, $|x| \neq 0.5L_x$, $|y| \neq 0.5L_y$, the cell where the ghost node has local index 22 is selected as the associated free-surface cell. The relationship between a ghost node and the associated free-surface cell can be described as follows:

- The free-surface cell is the cell used to enforce the free-surface Dirichlet condition for the marker with $\boldsymbol{x} = \boldsymbol{x}_m$ using Equation (9), and
- the equation is written in the row of the ghost node in the algebraic global matrix equation (12) for φ (and φ_t).

For ghost nodes along any of the physical domain boundaries, the choice of associated free-surface cells is explained in Figure 5.

Step 2: Identifying added ghost nodes

Some of the free-surface cells identified may contain inactive nodes above the free surface after completing Step 1. This scenario is illustrated in Figure 6A, where the three white nodes indicated with arrows are inactive, leading to an algebraic

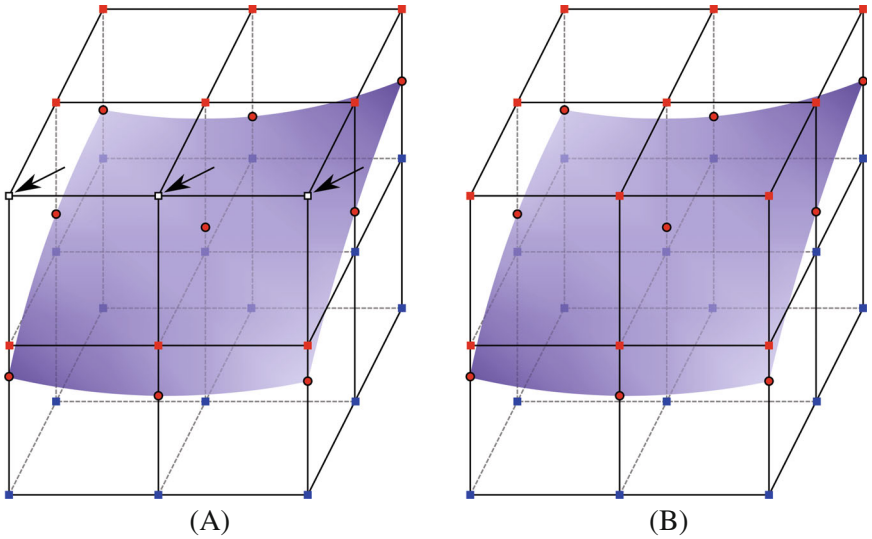

FIGURE 6 The inactive white nodes indicated with arrows in the free-surface cell in (A) are turned into ghost nodes (red) in (B). The blue nodes are water nodes below the free surface and the red circles indicate semi-Lagrangian free-surface markers (Step 2).

global matrix equation (12) with fewer equations than unknowns. To avoid this scenario, all free-surface cells identified in Step 1 are screened for inactive nodes. When inactive nodes are discovered, these are turned into (active) ghost nodes such as illustrated in Figure 6B. Clearly, this may lead to two ghost nodes having the same (x, y) coordinates, and since for each ghost node the free-surface Dirichlet condition is enforced for the free-surface marker with the same horizontal coordinates, this means that the Dirichlet condition for the same marker may be enforced twice. However, the two ghost nodes with the same horizontal coordinates will always be associated with different free-surface cells, so that the same marker has different local coordinates in the two cells. This is because the ghost node is always among one of the upper nodes with local index between 18 and 24 in a cell. The local expression (9) is therefore different in the two cells, and consequently the two ghost nodes having the same horizontal coordinates do not lead to a singular coefficient matrix $\boldsymbol{A}$ in Equation (12). Step 2 can therefore be considered a strategy to increase the number of equations in the algebraic global matrix equation without introducing additional constraints.

Step 3: Modifying choice of free-surface cells for additional ghost nodes

After completing Step 2, the free-surface cells associated with the inactive-turned-into-ghost nodes may not be optimal for the numerical solution. Consider for example the additional ghost node indicated with an arrow in Figure 7A with local index 24 in the green-shaded free-surface cell associated after Step 2. Because all horizontally adjacent nodes are active, this ghost node may instead be associated with the free-surface cell where it has local index 22 in (B). By using a free-surface cell where the ghost node is not a corner node, the continuity of the solution with overlapping cells is improved, which through experience is found to enhance the robustness of the numerical solution and reduces the number of iterations required by the iterative solver.

If it is not possible to select the cell where the node has local index 22, it may still be possible to select a free-surface cell where the node has local cell index 19, 21, 23, or 25, which also improves continuity. The algorithm to possibly modify free-surface cells associated with the additional ghost nodes identified in Step 2 is schematically illustrated in Figure 8. Here, n_{agn} is the node index of the ghost node that initially is associated with free-surface cell c_{fs0}, and the function loc_ind(n, c) returns the local index of a node with index n in cell c. If the node is associated with a free-surface cell where it has local index 22, the choice of free-surface cell is already optimal. Otherwise, if the local index in c_{fs0} differs from 22, *Loop 1* is performed to determine if the choice of free-surface cell can be modified to a cell where the node has local index 22. If *Loop 1* fails, *Loop 2* is performed to identify possible free-surface cells where the node has local index 19, 21, 23, or 25. These cells are included in a set ($\boldsymbol{C}$) where

- if there are no cells in ($\boldsymbol{C}$), the associated free-surface cell from Step 2 is kept

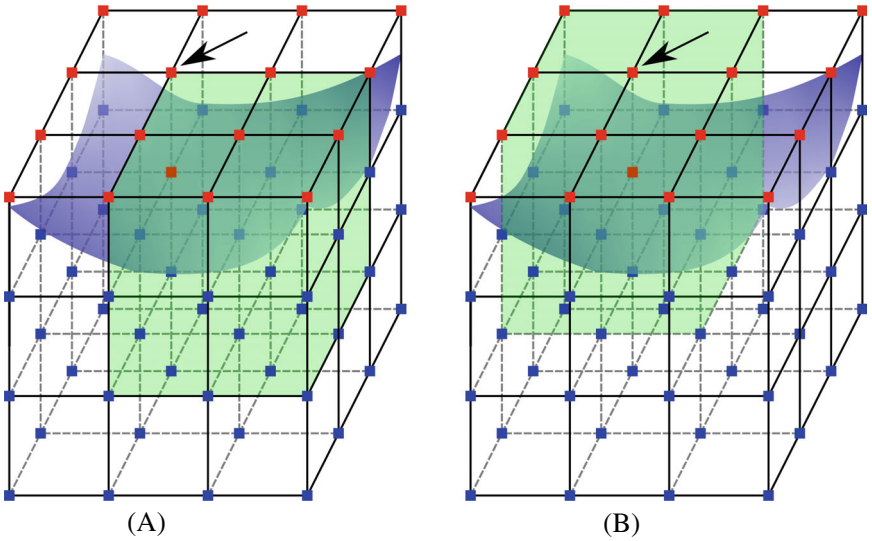

FIGURE 7 The node indicated with an arrow is an additional ghost node associated with the green-shaded free-surface cell in (A) after completion of Step 2. To improve continuity, the node is instead associated with the free-surface cell in (B) (Step 3). (A) Original free-surface cell, (B) modified free-surface cell.

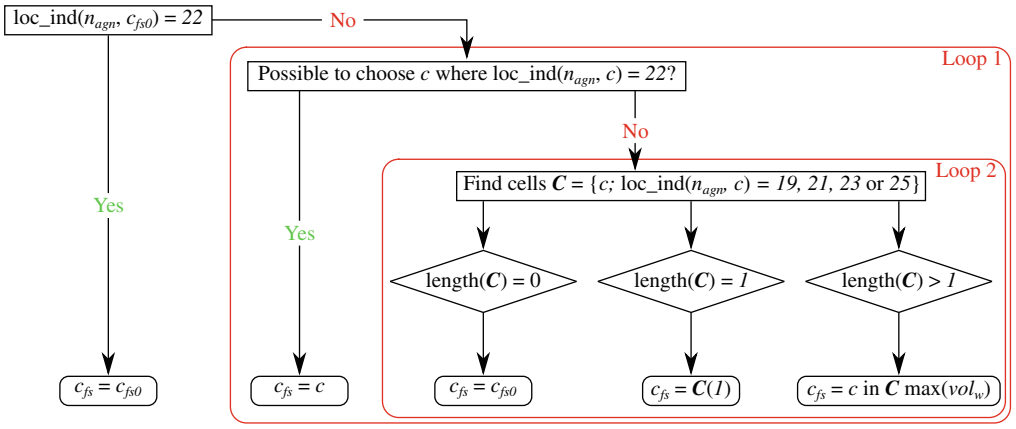

FIGURE 8 Algorithm to modify free-surface cell for additional ghost node n_{agn}. c_{fs0} is the free-surface cell initially associated with n_{agn} after Step 2, and c_{fs} is the modified free-surface cell.

- if there is one cell in ($\boldsymbol{C}$), the associated free-surface cell from Step 2 is modified to this one
- if there are more than one cell in ($\boldsymbol{C}$), the associated free-surface cell from Step 2 is modified to the cell in ($\boldsymbol{C}$) which has the largest volume of water vol_w.

The reason for choosing the vol_w criterion, which appears numerically robust, is strictly pragmatic, and other strategies have not been investigated.

The algorithm in Figure 8 serves to optimize the continuity in the global coefficient matrix $\boldsymbol{A}$.

Step 4: Eliminating interior ghost nodes

Having identified additional ghost nodes in Step 2, some of the ghost nodes identified in Step 1 may become superfluous. Such is the situation for ghost node 1 in Figure 9, which is just beneath the additional ghost node 2 and is surrounded by no inactive nodes. Since node 1 and 2 are used to enforce Dirichlet conditions for the same free-surface marker, the Dirichlet condition for node 1 can be replaced with the same connectivity equation (11) as enforced for interior nodes

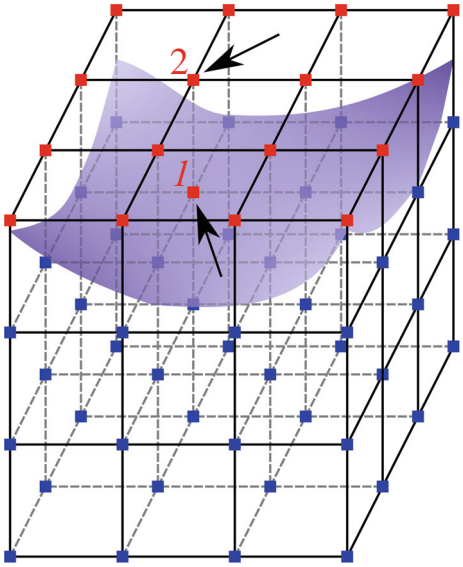

FIGURE 9 The node indicated as 2 is identified as an additional ghost node in Step 2, therefore the node indicated as 1, identified as a ghost node in Step 1, is changed into an air-continuity node (Step 4).

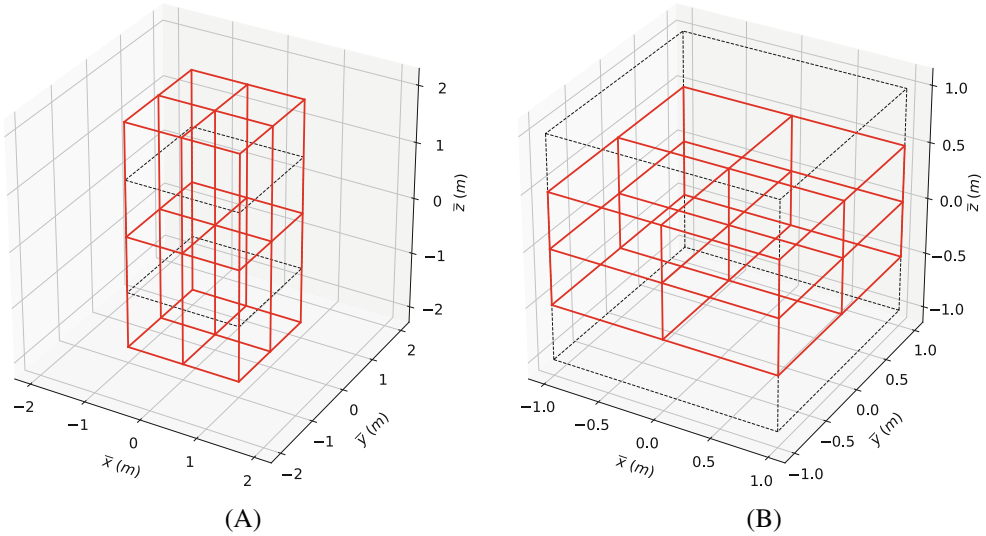

FIGURE 10 Cells with different aspect ratios. The black dashed lines indicate the outline of an undistorted, cubic cell with $\Delta x = \Delta y = \Delta z = 1.0$ m. (A) $\Delta x/\Delta z = 0.5$, (B) $\Delta x/\Delta z = 2.0$.

below the free surface. This does not lead to any loss of physical information, but is found to relax the numerical solution and makes it more robust. Node 1 is conveniently referred to as an air-continuity node.

Screening free-surface cells for inactive nodes in Step 2 does not significantly increase the computational time, and since the cells where additional ghost nodes are introduced are much fewer than the total amount of free-surface cells, neither do Step 3 or 4.

Filtering of free surface

Hanssen et al.[8] showed that using semi-Lagrangian free-surface markers generally leads to robust time evolution of the free-surface properties, but nevertheless, sawtooth-type instabilities will develop over time. Additionally, instabilities may develop due to what one may describe as accumulation errors, that is, small numerical errors that accumulate over time,

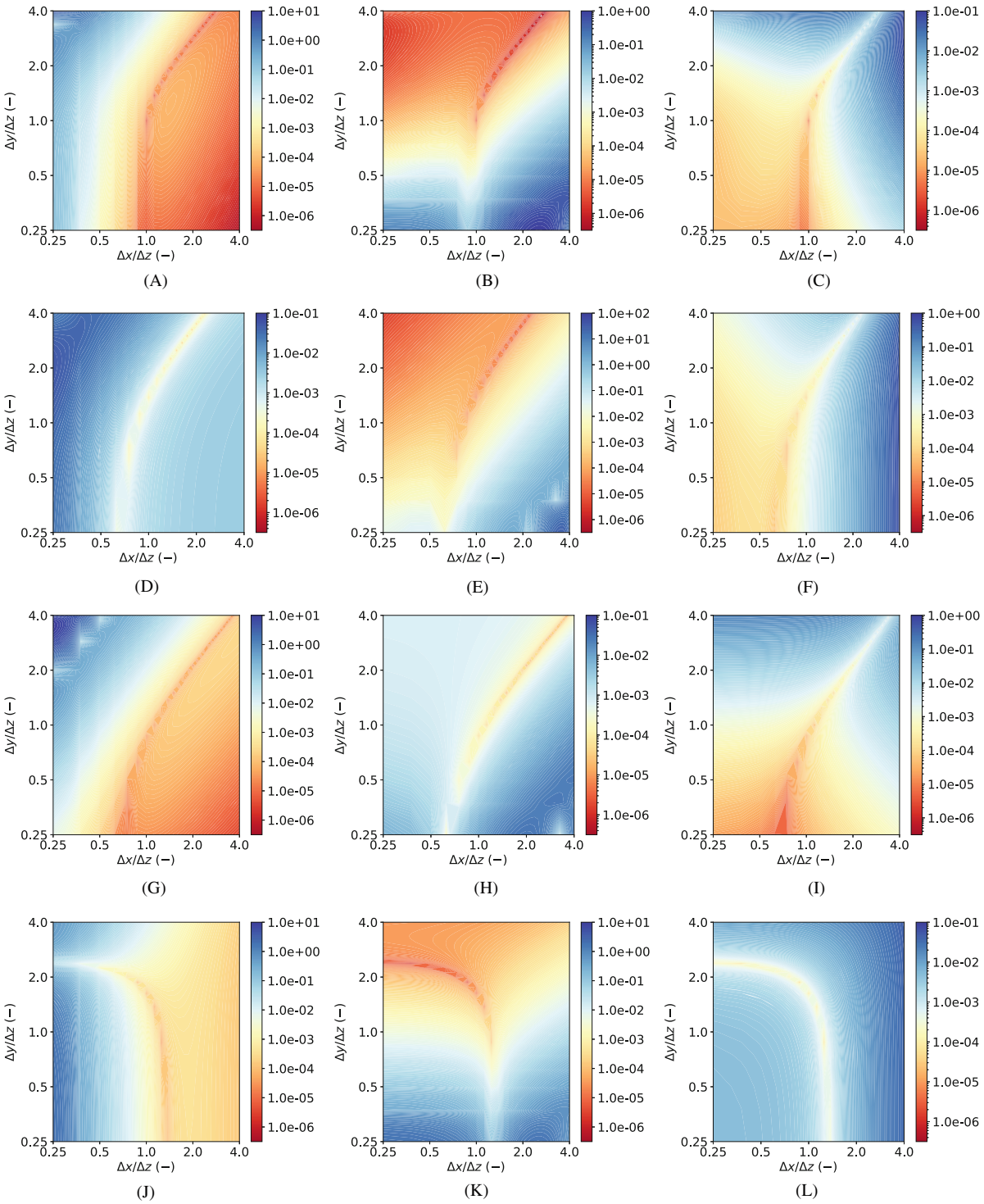

FIGURE 11 Absolute error of numerical solution in interior cell node computed with Equation (13) for stretched cells. (A) $\epsilon_{\varphi}(\boldsymbol{x}_{27})(\Delta x = 1\text{ m})$, (B) $\epsilon_{\varphi}(\boldsymbol{x}_{27})(\Delta y = 1\text{ m})$, (C) $\epsilon_{\varphi}(\boldsymbol{x}_{27})(\Delta z = 1\text{ m})$, (D) $\epsilon_{\varphi_x}(\boldsymbol{x}_{27})(\Delta x = 1\text{ m})$, (E) $\epsilon_{\varphi_x}(\boldsymbol{x}_{27})(\Delta y = 1\text{ m})$, (F) $\epsilon_{\varphi_x}(\boldsymbol{x}_{27})(\Delta z = 1\text{ m})$, (G) $\epsilon_{\varphi_y}(\boldsymbol{x}_{27})(\Delta x = 1\text{ m})$, (H) $\epsilon_{\varphi_y}(\boldsymbol{x}_{27})(\Delta y = 1\text{ m})$, (I) $\epsilon_{\varphi_y}(\boldsymbol{x}_{27})(\Delta z = 1\text{ m})$, (J) $\epsilon_{\varphi_z}(\boldsymbol{x}_{27})(\Delta x = 1\text{ m})$, (K) $\epsilon_{\varphi_z}(\boldsymbol{x}_{27})(\Delta y = 1\text{ m})$, (L) $\epsilon_{\varphi_z}(\boldsymbol{x}_{27})(\Delta z = 1\text{ m})$.

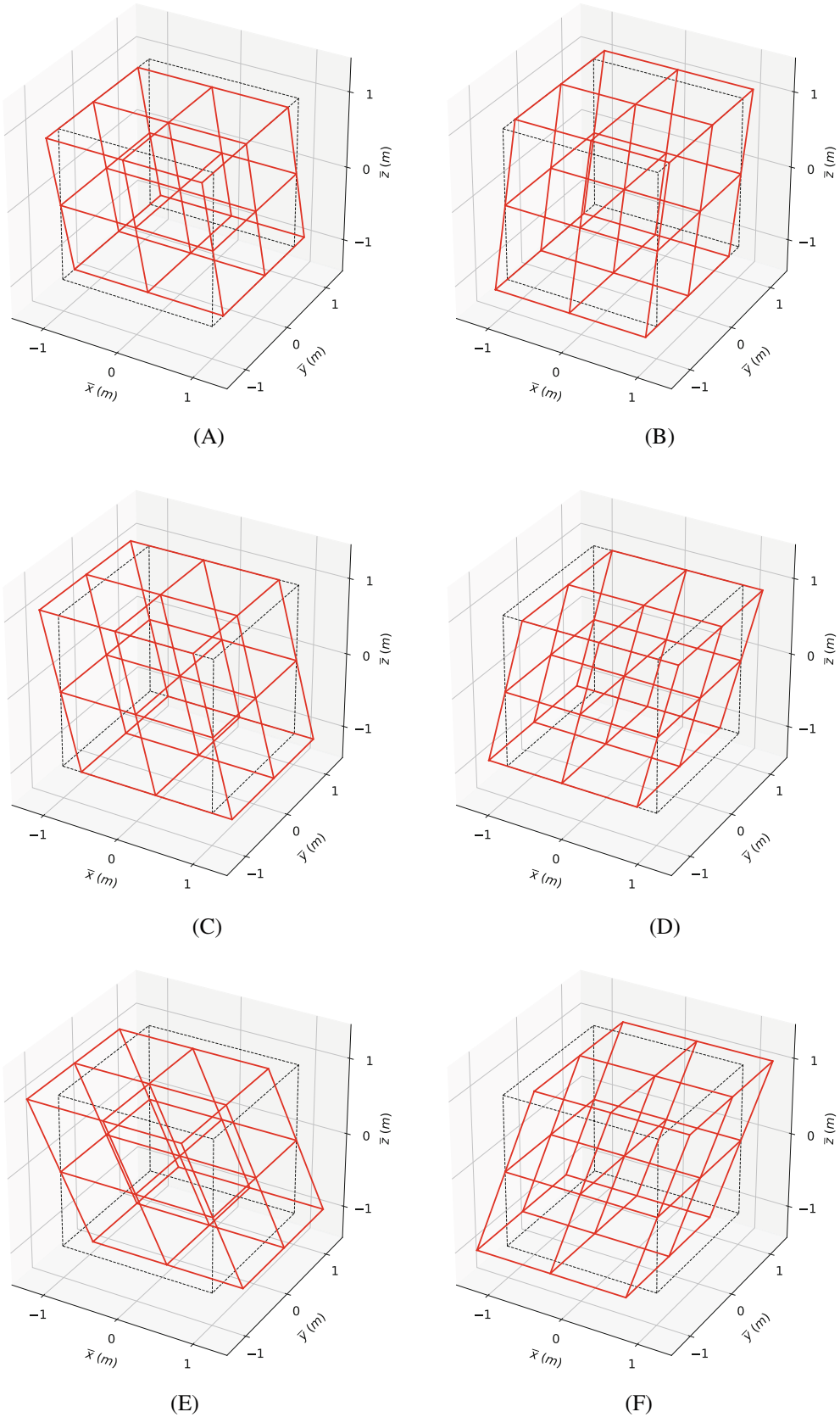

FIGURE 12 Cells with different skewness. The black dashed lines indicate the outline of an undistorted, cubic cell with $\Delta x = \Delta y = \Delta z = 1.0$ m. (A) $\gamma_x = -14.32°$, (B) $\gamma_x = 14.32°$, (C) $\gamma_y = -14.32°$, (D) $\gamma_y = 14.32°$, (E) $\gamma_x = \gamma_y = -14.32°$, (F) $\gamma_x = \gamma_y = 14.32°$.

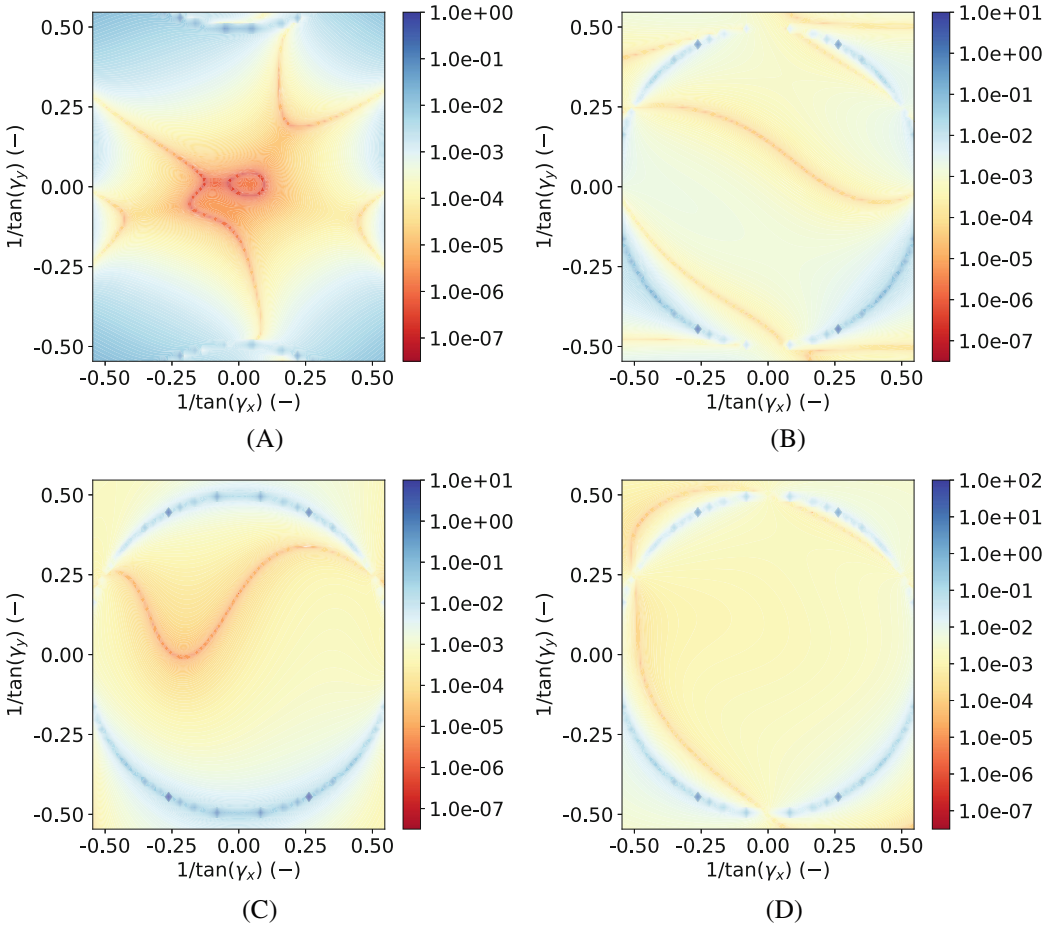

FIGURE 13 Absolute error of numerical solution in interior cell node computed with Equation (13) for distorted cells. (A) ϵ_{φ}, (B) ϵ_{φ_x}, (C) ϵ_{φ_y}, (D) ϵ_{φ_z}.

for instance caused by round off due to the computer's finite precision. Irrespective of their cause, to prevent instabilities from growing, the free-surface elevation and potential are here smoothed after every physical time step using spatial filters. To avoid the slow convergence of 2D filtering, this is replaced by sequential one-dimensional (1D) filtering in x and y-direction, respectively. Kuo et al.[22] showed that such sequential 1D filtering is equivalent to 2D filtering. With markers distributed evenly in each direction, a weak 11th-order Savitzky–Golay (SG) filter is used away from domain boundaries. It was shown by Hanssen et al.[8] in 2D that this filter conserves energy and mass for waves with steepness close to the breaking limit. Towards boundaries, where it was shown by Ducrozet et al.[23] that SG filters may ultimately amplify rather than reduce noise and/or errors, the SG filter is replaced by the selective filters developed by Berland et al.[24] It is noted that filtering towards the domain boundaries is of little practical importance here due to the use of relaxation zones, but it may be of importance in later applications where wave-structure interactions are considered. As a future development, the filters proposed by Shao et al.,[25] improving the SG filters' ability to avoid sawtooth instabilities, may be considered.

Estimation of free-surface properties

The free-surface conditions (4), used to evolve the free-surface elevation and potential in time, require that their gradients $\tilde{\nabla}\zeta$ and $\nabla\varphi$ are estimated in the position of each free-surface marker. Here, we evaluate $\tilde{\nabla}\zeta$ from third-order B-splines that are fitted through ζ in x and y-direction. Alternatively, a finite-difference scheme could have been used since the

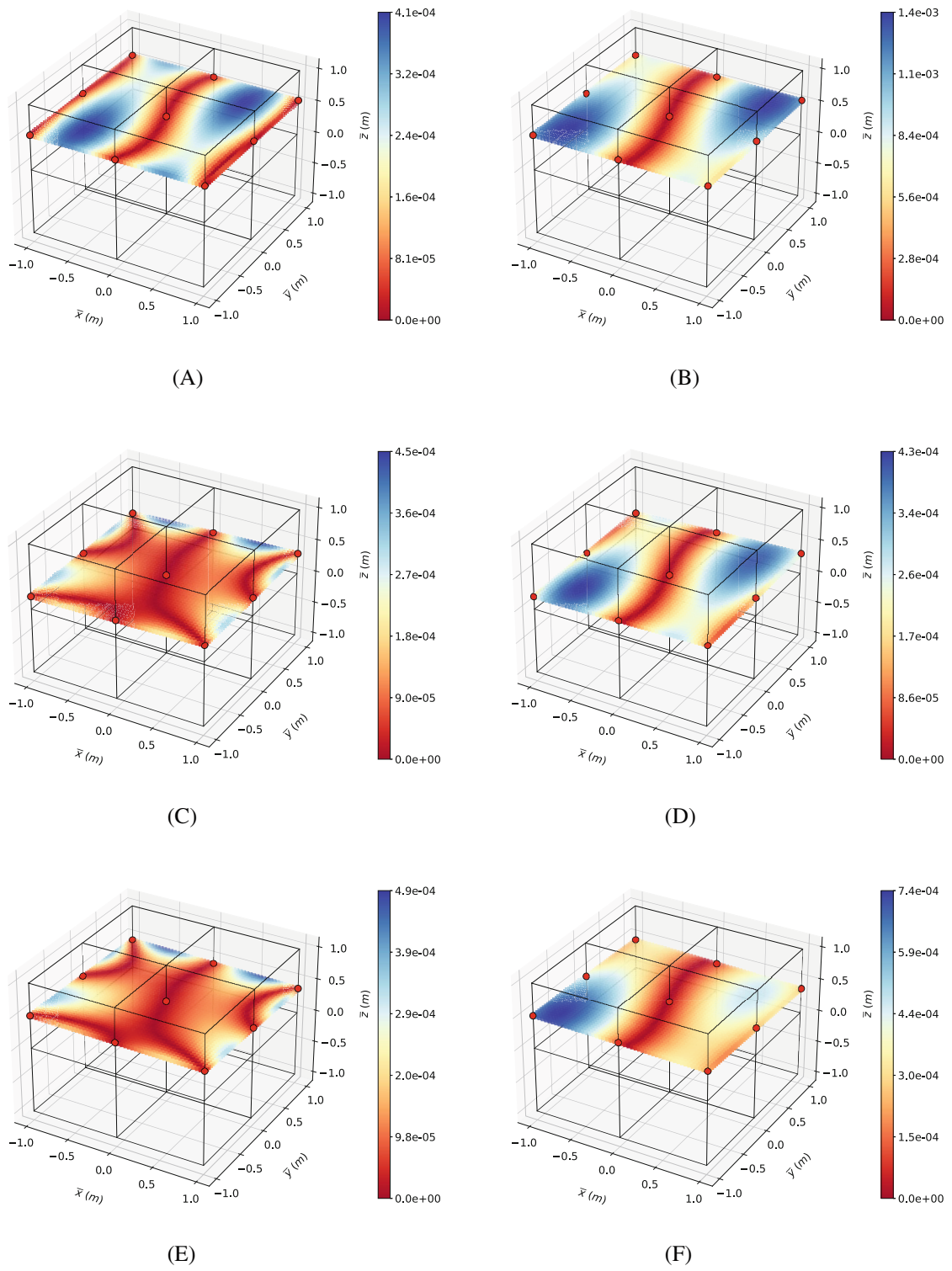

FIGURE 14 Normalized absolute errors of φ and $\partial\varphi/\partial z$ for $z_{27} = -0.35\Delta z$. The red circles indicate the position of free-surface markers. (A) $\overline{\epsilon}_{\varphi}(t/T = 0.0)$, (B) $\overline{\epsilon}_{\varphi_z}(t/T = 0.0)$, (C) $\overline{\epsilon}_{\varphi}(t/T = 0.33)$, (D) $\overline{\epsilon}_{\varphi_z}(t/T = 0.33)$, (E) $\overline{\epsilon}_{\varphi}(t/T = 0.83)$, (F) $\overline{\epsilon}_{\varphi_z}(t/T = 0.83)$.

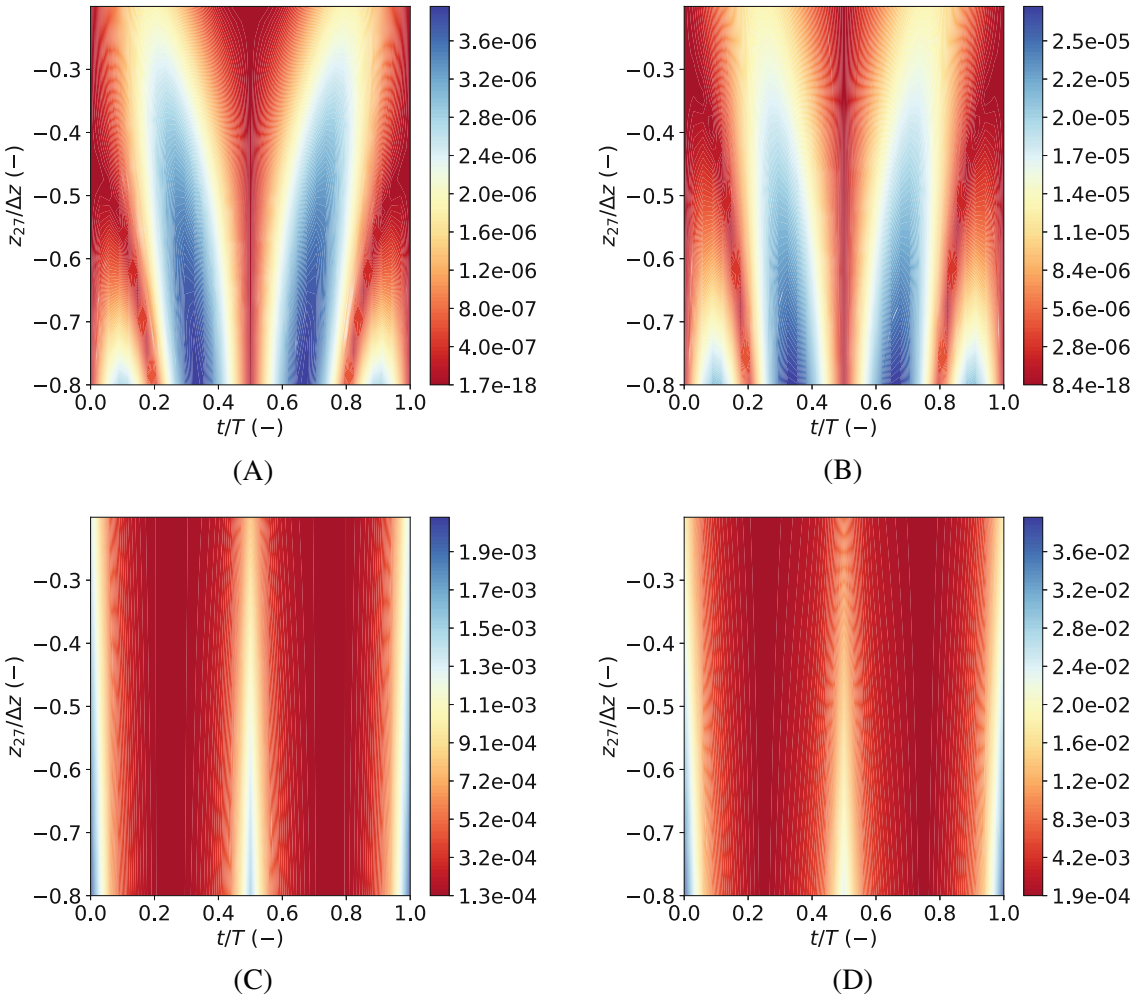

FIGURE 15 Numerical errors of φ and $\partial\varphi/\partial z$ plotted as a function of the cell's vertical position z_{27} relative to the mean free-surface level and the phase of the analytical solution represented by t/T. (A) $\overline{\epsilon}_{\varphi}$ in cell center, (B) $\overline{\epsilon}_{\varphi_z}$ in interior free-surface marker, (C) $\epsilon_{L_2}(\varphi)$ on free surface, (D) $\epsilon_{L_2}(\varphi_z)$ on free surface.

markers are evenly spaced. $\nabla\varphi$ is estimated from Equation (10) using the free-surface cell that contains the marker and where the marker is closest to the cell center.

4 | LOCAL PROPERTIES OF THE HPC METHOD IN 3D

To achieve an accurate global numerical solution, it is strictly required that the local representation of the solution variable inside each cell is accurate. Furthermore, it is of interest to understand how the errors vary in particular regions of a cell. Ma et al.[7] carefully examined the local solution in 2D cells not intersected by the free surface, and made two key observations: (1) square cells without any stretching or distortion gave superior accuracy, and (2) the numerical solution generally was most accurate in a region close to the cell center. We here investigate local properties in 3D, however, instead of considering the error distribution in the entire cell we focus on the error distribution over the free surface in cells where this is modeled as an immersed boundary.

We consider a single cell such as the one in Figure 3B with the cell's geometry characterized by the node spacing Δx, Δy and Δz in addition to the angles γ_x and γ_y, where γ_x and γ_y are the angles that the vertical line through local nodes 5-27-22 is rotated with about the x and y-axes, respectively. A cell where either γ_x or γ_y differ from zero is referred to as *distorted*. As implied by this definition, we only consider scenarios where the vertical cell sides

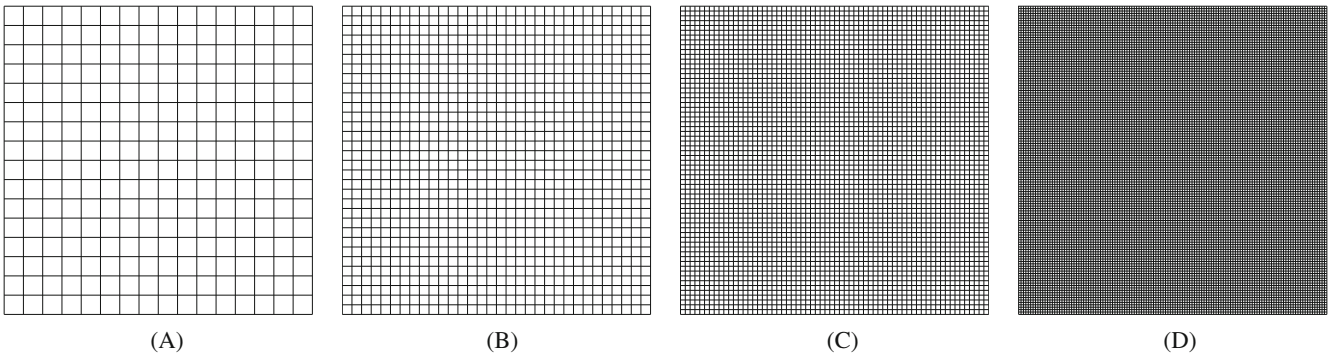

FIGURE 16 2D projection of pre-generated grids with $N_G = 4$ refinement levels and $N_{x,1} = N_{y,1} = N_{z,1} = 17$. (A) $g = 1$, (B) $g = 2$, (C) $g = 3$, (D) $g = 4$.

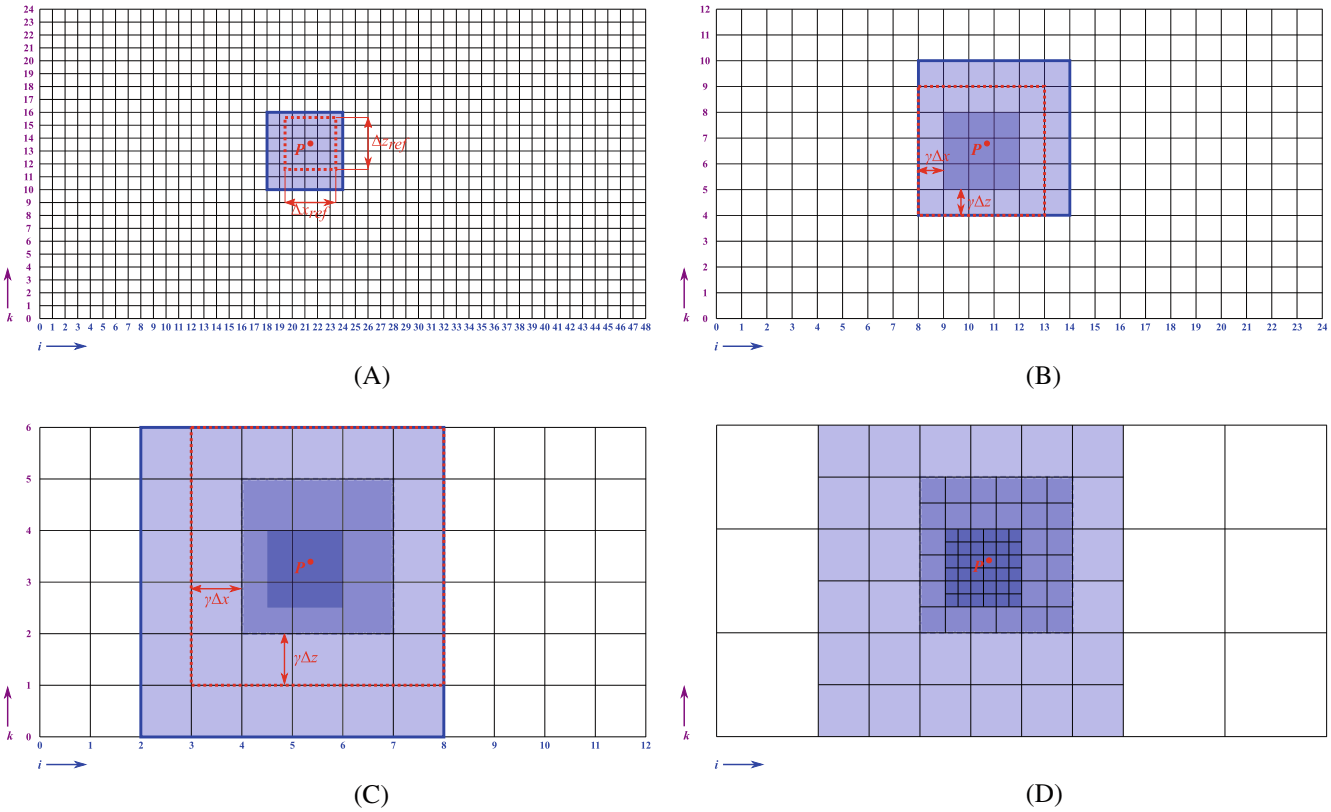

FIGURE 17 2D projection of adaptive refinement around a point $\boldsymbol{P}$ for a grid with $N_G = 4$. (A) $g = 4$, (B) $g = 3$, (C) $g = 2$, (D) refined grid.

have the same rotation. Completely random cell geometries are not investigated, however, it is reasonable to assume that the conclusions of the present study can be generalized for such cases. To be consistent with the notation used by Ma et al.,[7] cell stretching is indicated by the aspect ratio of the cell defined through the dimensionless parameters $\Delta x/\Delta z$ and $\Delta y/\Delta z$, and the distortion of the cell is indicated by skewness parameters $\cot(\gamma_x) = 1/\tan(\gamma_x)$ and $\cot(\gamma_y) = 1/\tan(\gamma_y)$.

Dirichlet conditions enforced in boundary nodes, or in free-surface markers when this is present, are defined by an analytical solution corresponding to Airy wave theory:

$$\varphi_{\text{ana}}(x, y, z, t) = \frac{\omega\zeta_A}{k} \frac{\cosh(k(z+h))}{\sinh(kh)} \sin(k_x x + k_y y - \omega t). \tag{13}$$

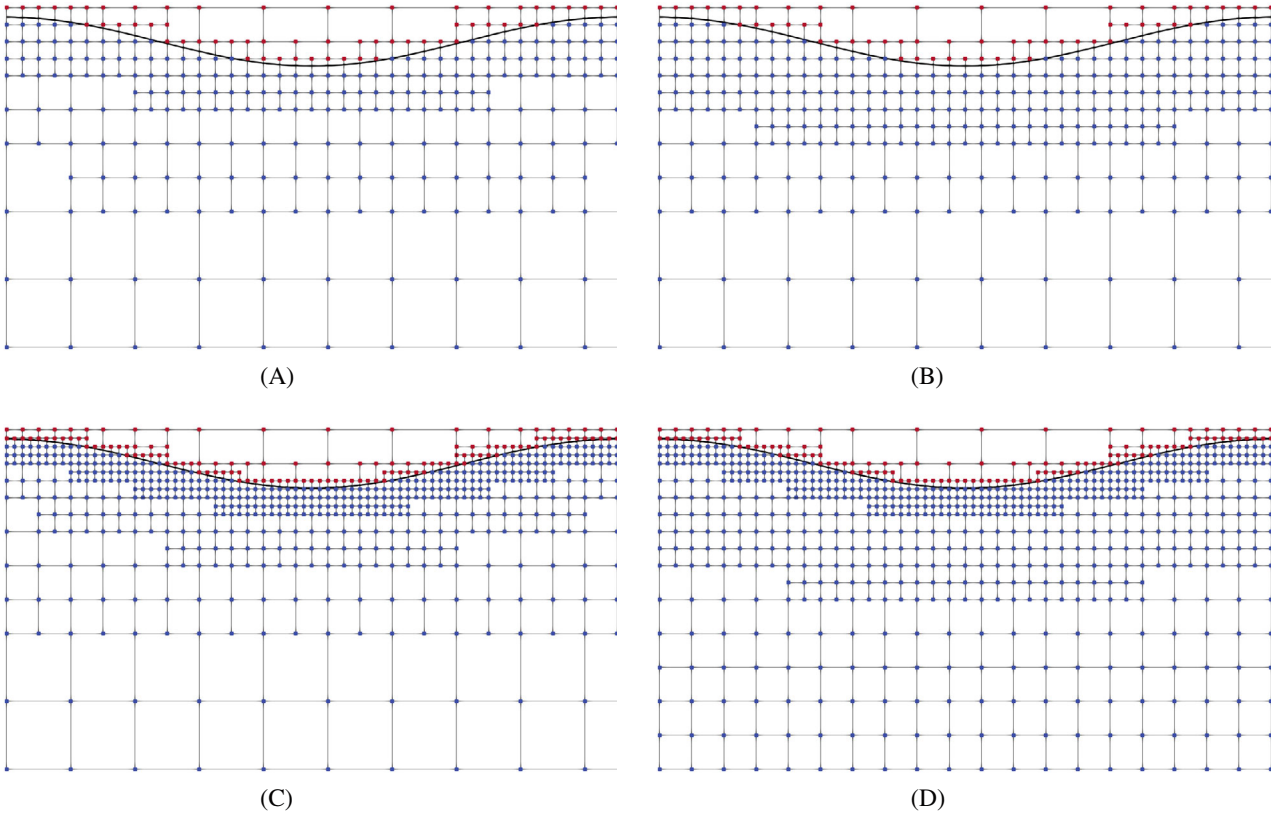

FIGURE 18 Example 2D projections showing grid refinement around a sinusoidal free surface with different AGR settings. The blue nodes are below the free surface and the red nodes are free-surface ghost nodes used to enforce free-surface Dirichlet conditions in the hydrodynamic BVPs. (A) $N_G = 3, n_{\Delta x_{ref}} = n_{\Delta y_{ref}} = n_{\Delta z_{ref}} = 4, \gamma = 1$, (B) $N_G = 3, n_{\Delta x_{ref}} = n_{\Delta y_{ref}} = n_{\Delta z_{ref}} = 8, \gamma = 1$, (C) $N_G = 4, n_{\Delta x_{ref}} = n_{\Delta y_{ref}} = n_{\Delta z_{ref}} = 4, \gamma = 1$, (D) $N_G = 4, n_{\Delta x_{ref}} = n_{\Delta y_{ref}} = n_{\Delta z_{ref}} = 4, \gamma = 3$.

Here, ζ_A is the wave amplitude, t is the time, ω is the wave frequency corresponding to a wave period $T = 2\pi/\omega$, $k_x = k\cos\beta$ and $k_y = k\sin\beta$ are wave-number components in x and y-direction, respectively, with k the wave number and β the wave direction relative to the x-axis. The water depth h is assumed sufficiently large so that the deep-water linear dispersion relationship applies, that is, $\omega = \sqrt{gk}$, and the acceleration of gravity is taken as $g = 9.81\ \mathrm{m/s^2}$.

In cells intersected by the free surface, the position of this is specified by the analytical wave elevation

$$\zeta_{\mathrm{ana}}(x, y, t) = \zeta_A \cos(k_x x + k_y y - \omega t). \tag{14}$$

The cells are defined with coinciding local and global origins so that $\bar{\boldsymbol{x}} = \boldsymbol{x}$. For cells intersected by the free surface, however, a sensitivity analysis is performed varying the vertical position of the cell center relative to $z = 0$.

The accuracy of the numerical solution is examined using the absolute error in a specific point defined as

$$\epsilon_\chi(\boldsymbol{x}) = |\chi_{\mathrm{num}}(\boldsymbol{x}) - \chi_{\mathrm{ana}}(\boldsymbol{x})| \tag{15}$$

and the $L2$ error taken over a region defined by $n = 1, \ldots, N$ discrete points:

$$\epsilon_{L2}(\chi) = \frac{\sum_{n=1}^{N} |\chi_{\mathrm{num}}(\boldsymbol{x}_n) - \chi_{\mathrm{ana}}(\boldsymbol{x}_n)|^2}{\sum_{n=1}^{N} |\chi_{\mathrm{ana}}(\boldsymbol{x}_n)|^2}. \tag{16}$$

The subscripts *num* and *ana* refer to the numerical and analytical solution, respectively, and χ may represent either φ or any of the components of $\nabla\varphi$. A normalized absolute error is defined as $\bar{\epsilon}_\chi(\boldsymbol{x}) = \epsilon_\chi(\boldsymbol{x})/(\omega\zeta_A/k)$. Note that this normalized error is only nondimensional if $\chi = \varphi$.

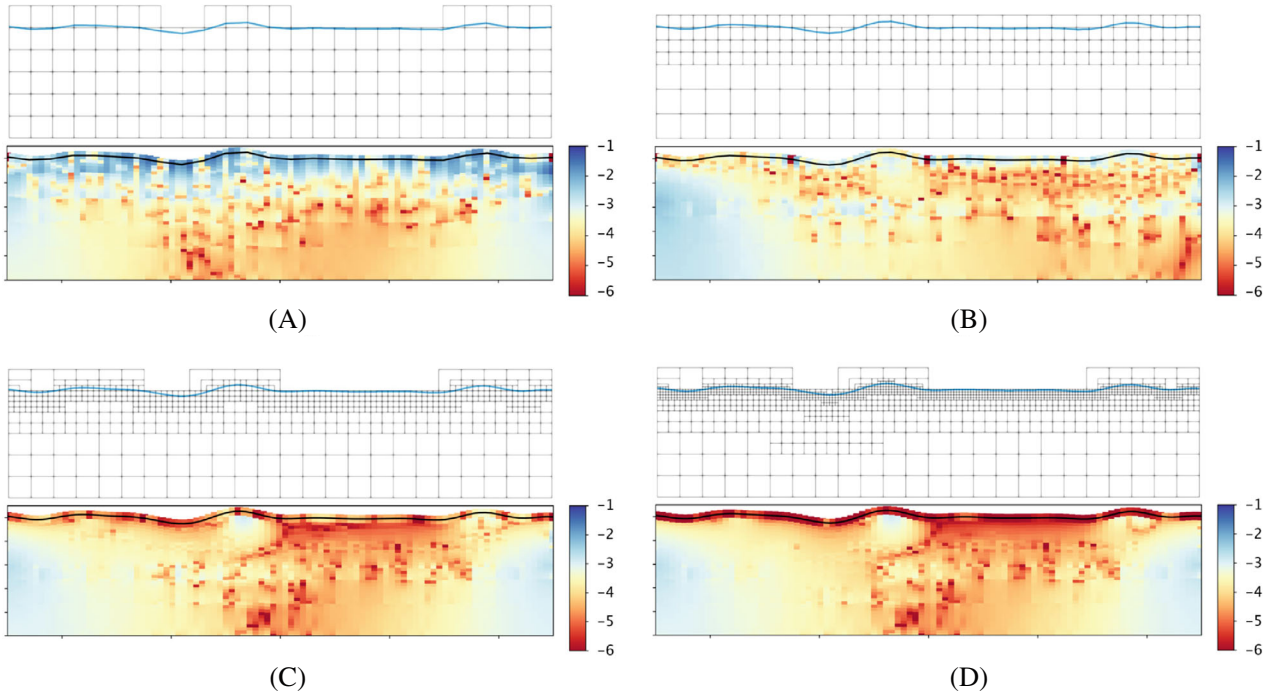

FIGURE 19 2D projection of $\log_{10}(\epsilon_\varphi)$ for grids with $\lambda_p/\Delta x_1 \approx 7.5$. The remaining AGR settings are $n_{\Delta x_{\text{ref}}} = n_{\Delta y_{\text{ref}}} = n_{\Delta z_{\text{ref}}} = 4$ and $\gamma = 1$. All cells are cubic. (A) $N_G = 1$, (B) $N_G = 2$, (C) $N_G = 3$, (D) $N_G = 4$.

TABLE 1 Regular stream-function wave cases with wave height $H = 2\zeta_A$, wavelength λ, k is the wave number and wave period T

Simulation id	h (m)	H (m)	λ (m)	h/λ (−)	$k\zeta_A$ (−)	T (s)	N_G (−)	$T/\Delta t$ (−)	$\lambda/\Delta x_1$ (−)	$\lambda/\Delta x_G$ (−)
Reg_H2_deep_N4	50.0	2.0	50.0	1.0	0.13	5.61	4	40.0	4.0	32.0
Reg_H3_deep_N4	50.0	3.0	50.0	1.0	0.19	5.56	4	40.0	4.0	32.0
Reg_H4_deep_N4	50.0	4.0	50.0	1.0	0.25	5.48	4	40.0	4.0	32.0
Reg_H5_deep_N4	50.0	5.0	50.0	1.0	0.31	5.39	4	40.0	4.0	32.0
Reg_H2_interm_N3	15.0	2.0	50.0	0.3	0.13	5.74	3	40.0	10.0	40.0
Reg_H3_interm_N3	15.0	3.0	50.0	0.3	0.19	5.68	3	40.0	10.0	40.0
Reg_H4_interm_N3	15.0	4.0	50.0	0.3	0.25	5.59	3	40.0	10.0	40.0
Reg_H5_interm_N3	15.0	5.0	50.0	0.3	0.31	5.48	3	40.0	10.0	40.0

4.1 | Accuracy in fully submerged stretched and distorted cells

Figure 10 shows examples of cells with two different aspect ratios obtained by stretching (or compressing) in $\bar{z}$ direction. In the following, we perform a systematic parameter investigation of the numerical error in the interior cell node by varying the aspect ratios $\Delta x/\Delta z$ and $\Delta y/\Delta z$ between 0.25 and 4 for the three scenarios with either Δx, Δy or Δz fixed. The analytical solution in Equation (13) is enforced in all boundary nodes with the following parameters: $k = 0.5236$ m^{-1} corresponding to a wavelength $\lambda = 12$ m, $\beta = 45°$, $\zeta_A = 1.0$ m and $h = 0.5\lambda$. From the deep-water dispersion relation we have $\omega = 2.2664$ s^{-1}, corresponding to a wave period $T = 2.77$ s. The time instant is taken as $t = 0.2$ T.

The absolute errors from equation (15) in the cell node with local index 27 for φ and the components of $\nabla\varphi$ are plotted in Figure 11 as a function of the cell's aspect ratios. Similar to the observations by Ma et al.[7] in 2D, $\epsilon_\varphi(\boldsymbol{x}_{27})$ is always small for cubic cells with $\Delta x/\Delta z = \Delta y/\Delta z = 1$. Although there are certain combinations of aspect ratios away from unity that give minimum errors for $\nabla\varphi$, cubic cells always give limited velocity errors irrespective of which cell dimension is fixed. Since the interior cell node is used to enforce continuity in a global solution, it is crucial to optimize the accuracy of φ in

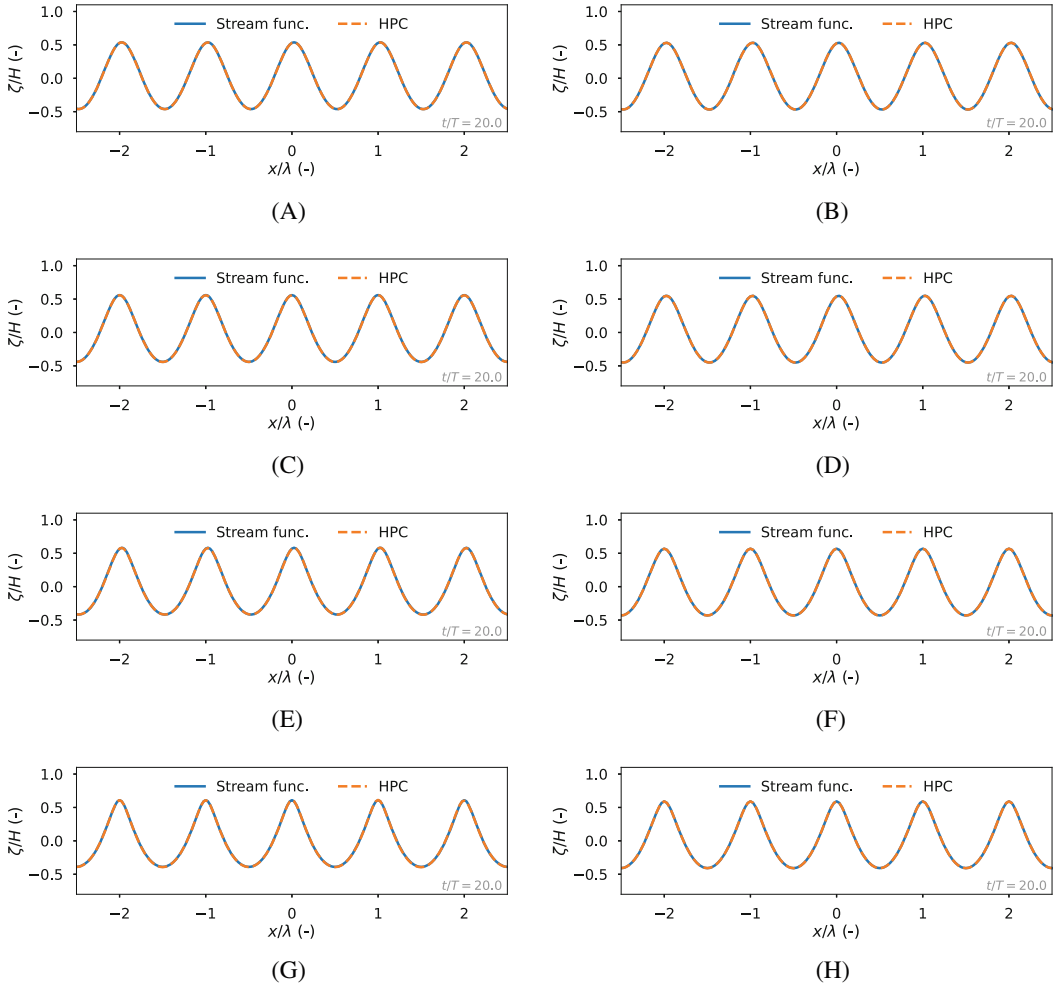

FIGURE 20 Snapshot of wave elevation for regular waves in Table 1 through a cut $y = 0$ at $t = 20T$. Numerical (HPC) results compared with reference (stream function) solution. (A) $H = 2.0$ m, $h = 15.0$ m, (B) $H = 2.0$ m, $h = 50.0$ m, (C) $H = 3.0$ m, $h = 15.0$ m, (D) $H = 3.0$ m, $h = 50.0$ m, (E) $H = 4.0$ m, $h = 15.0$ m, (F) $H = 4.0$ m, $h = 50.0$ m, (G) $H = 5.0$ m, $h = 15.0$ m, (H) $H = 5.0$ m, $h = 50.0$ m.

this point, and the present study shows that this generally is best achieved by using cubic cells in 3D in the same way as it is best to use square cells in 2D.

We now consider the effect of cell distortion by fixing the cells dimension $\Delta x = \Delta y = \Delta z = 1$ m and varying $1/\tan(\gamma_x)$ and $1/\tan(\gamma_y)$ between -0.5 and 0.5. Examples of distorted-cell geometries are plotted in Figure 12. Figure 13 shows that the absolute errors of φ in the interior cell node is minimized close to zero skewness about both the x and y-axes, which is equivalent to the 2D observations by Ma et al.[7] The reason for seeing local minima that differ slightly from $\gamma_x = \gamma_y = 0$ is related to the parameters chosen for the analytical solution used to enforce outer boundary conditions. This is also why the components of $\nabla\varphi$ have their minima away from $\gamma_x = \gamma_y = 0$. However, although the dependency on γ_x and γ_y vary with the analytical solution, the error of φ is always minimized, and the errors of $\nabla\varphi$ are always limited, for a cell with zero skewness.

4.2 | Accuracy in cells intersected by the free surface

With the immersed boundary method, cells may be intersected by the free surface in arbitrary positions in the upper half-plane with $\bar{z} > 0$. The influence of the immersed-boundary modeling on the local accuracy is here studied on a cubic, undistorted cell with $\Delta x = \Delta y = \Delta z = 1$m. In wave and wave-body interaction problems, the local accuracy of the numerical solution in free-surface cells strongly influences the accuracy of the global solution. The free-surface elevation is given by Equation (14) with $\zeta_A = 0.2$ m, $k = 0.2513$ m^{-1} corresponding to $\lambda = 25$ m, $\beta = 45°$, $\omega = 1.5702$ s^{-1}

TABLE 2 Regular stream-function wave cases to study the influence of the number of refinement levels N_G for a fixed grid spacing Δx at the free surface

Simulation id	h (m)	H (m)	λ (m)	T (s)	N_G (−)	$T/\Delta t$ (−)	$\lambda/\Delta x_1$ (−)	$\lambda/\Delta x_G$ (−)
Reg_H5_deep_N1	50.0	5.0	50.0	5.39	1	40.0	32.0	32.0
Reg_H5_deep_N2	50.0	5.0	50.0	5.39	2	40.0	16.0	32.0
Reg_H5_deep_N3	50.0	5.0	50.0	5.39	3	40.0	8.0	32.0
Reg_H5_deep_N4	50.0	5.0	50.0	5.39	4	40.0	4.0	32.0
Reg_H5_interm_N1	15.0	5.0	50.0	5.48	1	40.0	40.0	40.0
Reg_H5_interm_N2	15.0	5.0	50.0	5.48	2	40.0	20.0	40.0
Reg_H5_interm_N3	15.0	5.0	50.0	5.48	3	40.0	10.0	40.0

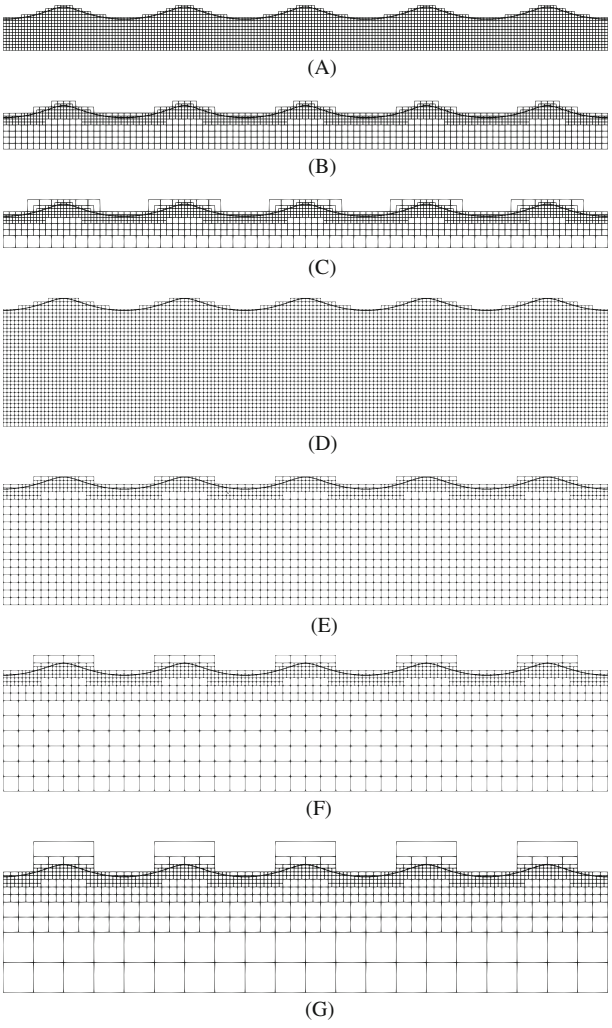

FIGURE 21 Plots of grids from Table 2. (A) $N_G = 1,\ h = 15.0$ m, (B) $N_G = 2,\ h = 15.0$ m, (C) $N_G = 3,\ h = 15.0$ m, (D) $N_G = 1,\ h = 50.0$ m, (E) $N_G = 2,\ h = 50.0$ m, (F) $N_G = 3,\ h = 50.0$ m, (G) $N_G = 4,\ h = 50.0$ m.

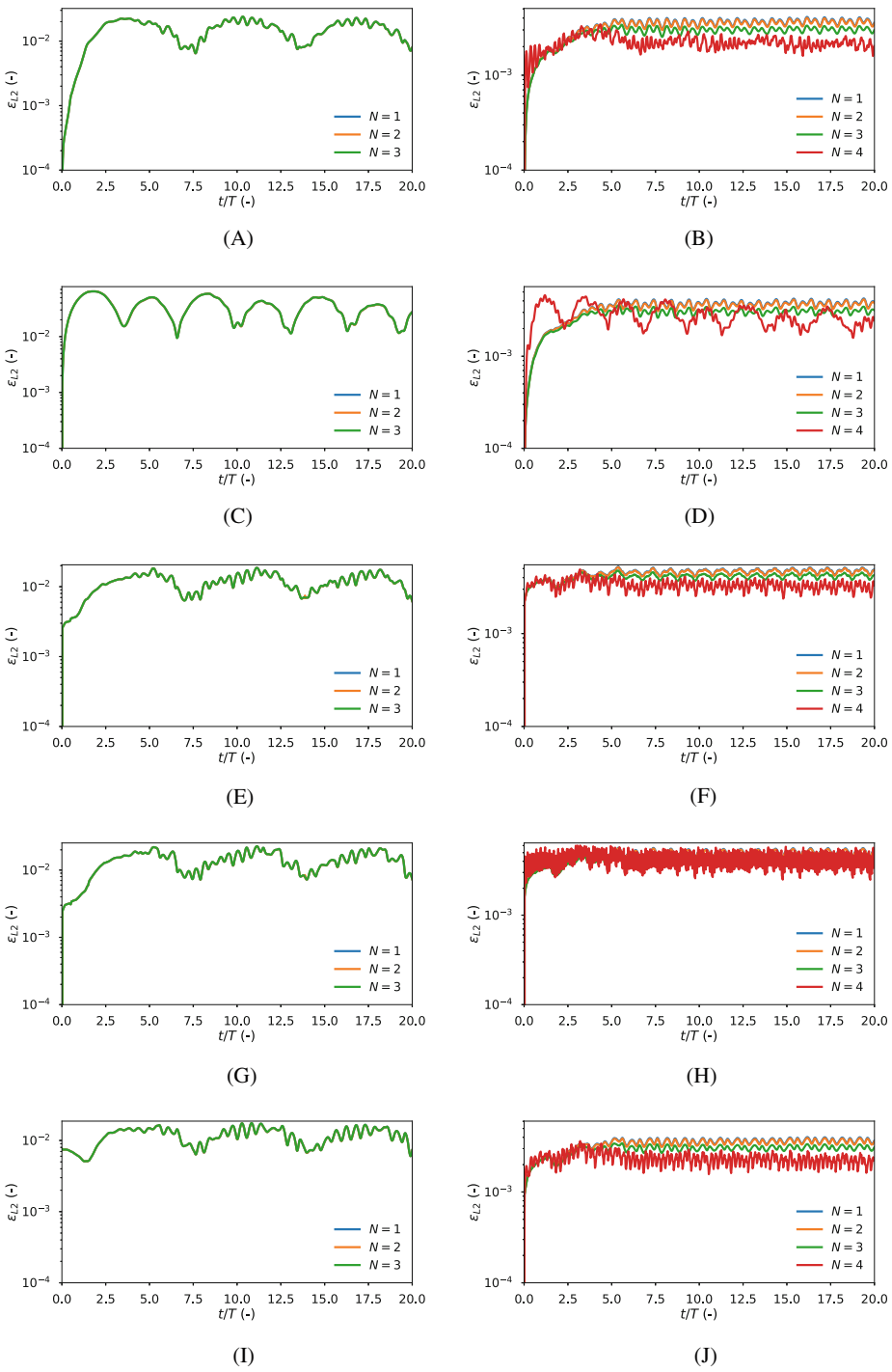

FIGURE 22 Time series of $L2$ errors of free surface elevation and the following variables on the free surface: φ, φ_x, φ_z, and φ_t. (A) ζ, $h = 15.0$ m, (B) ζ, $h = 50.0$ m, (C) $\hat{\varphi}$, $h = 15.0$ m, (D) $\hat{\varphi}$, $h = 50.0$ m, (E) $\hat{\varphi}_x$, $h = 15.0$ m, (F) $\hat{\varphi}_x$, $h = 50.0$ m, (G) $\hat{\varphi}_z$, $h = 15.0$ m, (H) $\hat{\varphi}_z$, $h = 50.0$ m, (I) $\hat{\varphi}_t$, $h = 15.0$ m, (J) $\hat{\varphi}_t$, $h = 50.0$ m.

corresponding to $T = 4$ s and $h = 12.5$ m. Equation (13) is used to enforce Dirichlet conditions in boundary nodes below the free surface. Conditions for the nine boundary nodes above the free surface are replaced with Dirichlet conditions enforced in the position of the nine markers on the free surface. Note that the analytical expression for φ here is applied also above the mean free-surface level without any linear approximation. The position of the free surface inside the cell is governed by the vertical global coordinate of the interior cell node z_{27}, which describes the submergence of the cell. With $z_{27} = 0$, the position of the interior cell node corresponds to the mean free-surface level.

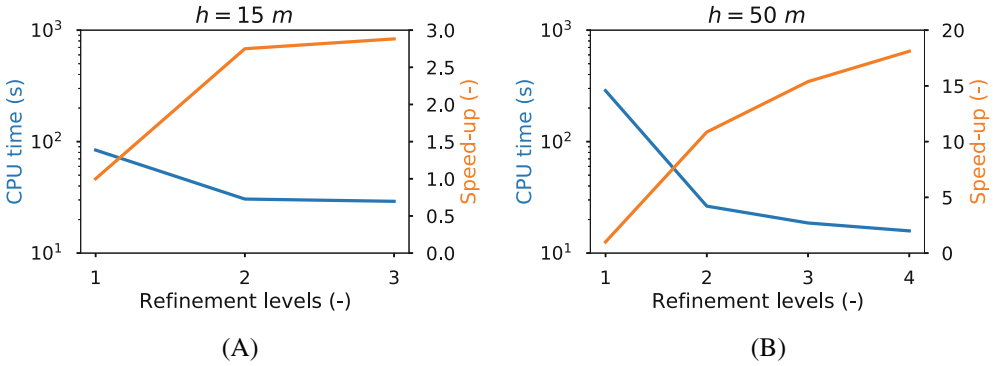

FIGURE 23 Computational time per time step (blue) and speed-up factor (orange) as a function of the number of grid refinement levels N_G for regular wave simulations on a laptop with an Intel Core i7 2.11 GHz processor, 32 GB RAM and four cores. (A) Intermediate depth, (B) deep water.

TABLE 3 Regular stream-function wave cases to study the influence of the grid spacing $\Delta x = \Delta y = \Delta z$ at the free surface for a fixed number of refinement levels N_G

Simulation id	h (m)	H (m)	λ (m)	T (s)	N_G (–)	$T/\Delta t$ (–)	$\lambda/\Delta x_1$ (–)	$\lambda/\Delta x_G$ (–)
Reg_H5_deep_N4_dx1	50.0	5.0	50.0	5.39	4	40.0	4.0	32.0
Reg_H5_deep_N4_dx2	50.0	5.0	50.0	5.39	4	40.0	6.0	48.0
Reg_H5_deep_N4_dx3	50.0	5.0	50.0	5.39	4	40.0	8.0	64.0

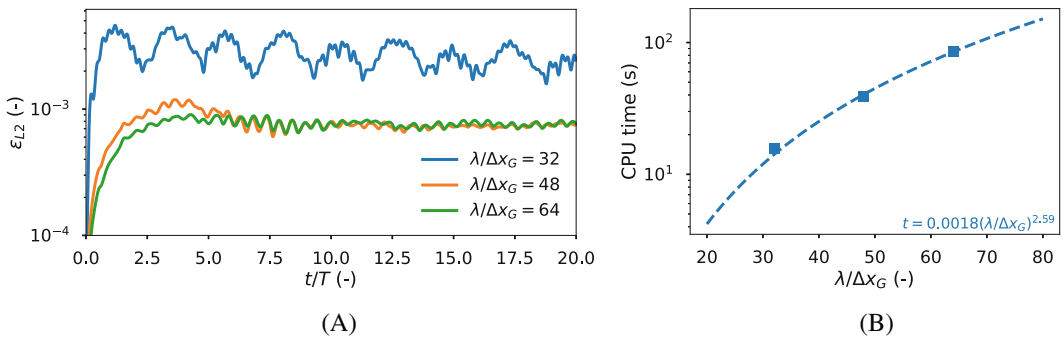

FIGURE 24 Time series of $L2$ error of free surface-velocity potential and computational time per time step for different grid spacings. All simulations are performed on a laptop with an Intel Core i7 2.11 GHz processor, 32 GB RAM and four cores. (A) L_2 error for $\hat{\varphi}$, (B) computational time.

Figure 14 shows the distribution of the normalized absolute errors $\bar{\epsilon}_\varphi$ and $\bar{\epsilon}_{\varphi_z}$ at different time instants for a cell with $z_{27} = -0.35\Delta z$. The numerical errors of φ are zero in the locations of the free-surface markers, having enforced analytical Dirichlet conditions there. Away from the markers, the distribution of $\bar{\epsilon}_\varphi$ varies with time. This should be kept in mind when applying fully-Lagrangian free-surface markers, since it may contribute to numerical instabilities as discussed for a 2D NWT by Hanssen.[9] Notably, the error of $\partial\varphi/\partial z$, which is the dominating velocity component in the kinematic free-surface condition (4) and therefore affects wave dispersion, is small for all t in the interior marker.

Figure 15 shows errors of variables that are considered especially important for the robustness and stability of the numerical solution as a function of the cell's submergence and the phase of the analytical solution. These are the normalized absolute error of φ in the interior cell node, the normalized absolute error of $\partial\varphi/\partial z$ in the interior free-surface marker, and the $L2$ errors of φ and $\partial\varphi/\partial z$ on the free surface computed with Equation (16). The cell's submergence is varied so that the entire free surface is always in the upper half of the cell, that is, $-\Delta z + \zeta_A < z_{27} < -\zeta_A$. The errors generally vary little with z_{27}, but the absolute errors of φ in the interior cell node and $\partial\varphi/\partial z$ in the interior free-surface marker

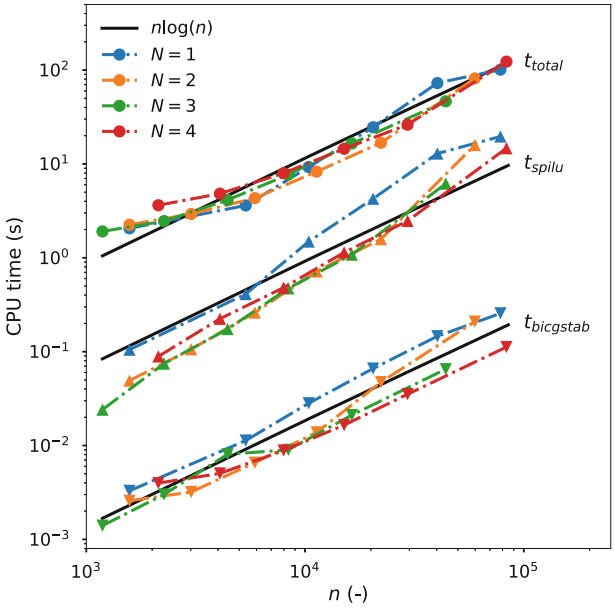

FIGURE 25 Computational time plotted as a function of the total number of active grid nodes n for different number of refinement levels N. t_{total} is the total time for one physical time step, and t_{spilu} and t_{bicgstab} denote the time spent by the preconditioner and solver, respectively, for solving the global matrix equation for φ one time. All simulations are performed on a laptop with an Intel Core i7 2.11 GHz processor, 32 GB RAM and four cores.

depend strongly on t/T. The latter variation may be explained with the fact that the gradients of the analytical solution in the cell vary with the solution's phase, which is inherent in studying dynamic fluid problems in time domain.

4.3 | Discussion and modeling recommendations

It was shown by Shao and Faltinsen[4] that the HPC method in 3D gives between third and fourth-order spatial accuracy when solving a mixed Dirichlet–Neumann BVP. Having here examined the local properties of the HPC method in 3D further, the following recommendations are offered to provide an accurate and robust global model:

- Grids with cubic or close-to-cubic cells should be used. Significantly stretched or distorted cells should be avoided at all cost.
- In free-surface cells, $\nabla\varphi$ should be estimated in the position of the interior free-surface marker.

The latter assumes that the free surface is modeled with semi-Lagrangian markers that have horizontal coordinates coinciding with those of the cell nodes.

When the free surface is modeled as an immersed boundary, the accuracy of the numerical solution is not sensitive to the vertical position of the free surface relative to the cell center. This offers flexibility in the numerical modeling and choice of grid parameters.

5 | ADAPTIVE GRID REFINEMENT

Using uniform grids with cubic cells in 3D for typical water-wave problems leads to a large number of nodes and thus inefficient numerical solutions. Motivated by this, we here introduce an adaptive grid refinement (AGR) method that allows for local grid refinement using solely cubic cells following the idea of octree grids. The AGR benefits from two distinct properties of the HPC method: (1) the high-order spatial accuracy ensures that regions with small velocity gradients can be suitably modeled using large cells, and (2) the polynomial formulation enables straightforward coupling of the

TABLE 4 Definition of long-crested irregular sea states with significant wave height H_s, spectral peak period T_p and peak-enhancement factor γ_s. $k_p = 2\pi/\lambda_p$ is the wave number estimated from the linear dispersion relation $\omega_p^2 = gk_p \tanh k_p h$ with $\omega_p = 2\pi/T_p$. $S_p = 2\pi H_s/gT_p^2$ is a steepness parameter

Simulation id	Spectrum	h (m)	H_s (m)	T_p (s)	γ_s (−)	λ_p (m)	S_p (−)	$0.5k_pH_s$ (−)	h/λ_p (−)
Hs3m_Tp10s_deep	JONSWAP	150.00	3.00	10.00	1.00	156.13	0.019	0.060	0.961
Hs5m_Tp10s_deep	JONSWAP	150.00	5.00	10.00	1.83	156.13	0.032	0.101	0.961
Hs7m_Tp10s_deep	JONSWAP	150.00	7.00	10.00	4.07	156.13	0.045	0.141	0.961
Hs9m_Tp10s_deep	JONSWAP	150.00	9.00	10.00	5.00	156.13	0.058	0.181	0.961
Hs3m_Tp10s_interm	JONSWAP	50.00	3.00	10.00	1.00	151.30	0.019	0.062	0.331
Hs5m_Tp10s_interm	JONSWAP	50.00	5.00	10.00	1.83	151.30	0.032	0.104	0.331
Hs7m_Tp10s_interm	JONSWAP	50.00	7.00	10.00	4.07	151.30	0.045	0.145	0.331
Hs9m_Tp10s_interm	JONSWAP	50.00	9.00	10.00	5.00	151.30	0.058	0.187	0.331
Hs3m_Tp10s_shallow	TMA	25.00	3.00	10.00	1.00	130.38	0.019	0.072	0.192
Hs5m_Tp10s_shallow	TMA	25.00	5.00	10.00	1.83	130.38	0.032	0.121	0.192
Hs7m_Tp10s_shallow	TMA	25.00	7.00	10.00	4.07	130.38	0.045	0.169	0.192
Hs9m_Tp10s_shallow	TMA	25.00	9.00	10.00	5.00	130.38	0.058	0.217	0.192

solution on different refinement levels without requiring additional interpolation schemes that may influence the solution's accuracy. Motivation is drawn from the work by Wang et al.[26] and Tong et al.[10] who demonstrated efficient AGR schemes for the HPC method in 2D.

5.1 | Initialization

The fundamental idea of the AGR proposed here is that the number of refinement levels N_G is predefined, where $N_G = 1$ corresponds to a uniform grid with no local refinement. When $N_G > 1$, separate grids are initially generated over the entire computational domain with dimensions L_x, L_y, and L_z. The number of grid nodes in each direction for refinement level g is given by the relationship

$$\begin{aligned} N_{x,g} &= 2^{g-1}(N_{x,1} - 1) + 1, \\ N_{y,g} &= 2^{g-1}(N_{y,1} - 1) + 1, \\ N_{z,g} &= 2^{g-1}(N_{z,1} - 1) + 1, \end{aligned} \tag{17}$$

where $N_{x,1}, N_{y,1}$, and $N_{z,1}$ are the number of grid nodes in each direction for the least refined (base) grid. The corresponding grid spacing on each refinement level are

$$\begin{aligned} \Delta x_g &= L_x/(N_{x,g} - 1), \\ \Delta y_g &= L_y/(N_{y,g} - 1), \\ \Delta z_g &= L_z/(N_{z,g} - 1). \end{aligned} \tag{18}$$

The number of grid nodes in each direction in the coarsest level ($N_{x,1}, N_{y,1}, N_{z,1}$), and the number of refinement levels N_G, must therefore be selected so that the grid spacing ($\Delta x_G, \Delta y_G, \Delta z_G$) in the most refined level are appropriate to the problem considered. Here, $\Delta x_G = \Delta x_{g=G}$, $\Delta y_G = \Delta y_{g=G}$ and $\Delta z_G = \Delta z_{g=G}$. An example showing 2D projections of the pre-generated grids on a cubic domain with $N_G = 4$ is shown in Figure 16.

The nodes in each of the grids are assigned with local node numbers $n_g = i_g + j_g \cdot N_{x,g} + k_g \cdot N_{x,g} \cdot N_{y,g}$, where i_g, j_g and k_g are node indices in x, y, and z direction, respectively. Based on these, a local cell-node connectivity matrix is established for each refinement level. The nodes are further assigned with global node numbers in ascending order starting with the

TABLE 5 Simulation parameters used for the sea states in Table 4

Simulation id	L_x (m)	L_y (m)	N_G (−)	Δt (s)	$\lambda_p/\Delta x_1$ (−)	$\lambda_p/\Delta x_G$ (−)	T_{sim} (s)
Hs3m_Tp10s_deep	1250.0	100.00	4	0.33	3.1	25.0	1300.0
Hs5m_Tp10s_deep	1250.0	100.00	4	0.33	3.1	25.0	1300.0
Hs7m_Tp10s_deep	1250.0	100.00	4	0.33	3.1	25.0	1300.0
Hs9m_Tp10s_deep	1250.0	100.00	4	0.33	3.1	25.0	1300.0
Hs3m_Tp10s_interm	1250.0	100.00	3	0.33	6.1	24.2	1300.0
Hs5m_Tp10s_interm	1250.0	100.00	3	0.33	6.1	24.2	1300.0
Hs7m_Tp10s_interm	1250.0	100.00	3	0.33	6.1	24.2	1300.0
Hs8m_Tp10s_interm	1250.0	100.00	3	0.33	6.1	24.2	1300.0
Hs3m_Tp10s_shallow	1000.0	30.00	2	0.33	13.0	26.1	1300.0
Hs5m_Tp10s_shallow	1000.0	30.00	2	0.33	13.0	26.1	1300.0
Hs7m_Tp10s_shallow	1000.0	30.00	2	0.33	13.0	26.1	1300.0
Hs9m_Tp10s_shallow	1000.0	30.00	2	0.33	13.0	26.1	1300.0

coarsest refinement level. Parent-child connectivity matrices are established between all adjacent grid levels, where for grid level g, $g-1$ is the parent level and $g+1$ is the child level. The parent-child connectivity matrices are constructed so that for a g-level node the coinciding parent and child nodes may be looked up (if they exist). Furthermore, all child cells inside a g-level cell may be looked up if $g < N_G$. Compared to redefining the entire connectivity relations each time the grid refinement changes, these look-up operations can be made fast, and pregenerating all required connectivity relationships therefore contributes favorably to the efficiency of the AGR.

5.2 | Dynamic refinement

The AGR is implemented so that each time the numerical solution is updated, either in a Runge–Kutta sub step or in a physical time step, the grid is refined from the highest refinement level ($g = N_G$) towards the lowest one ($g = 1$). The regions in the grid to be refined are contained within cuboids centred around some point. In the present work we only consider wave-propagation cases, with each free-surface marker in the center of a refinement cuboid that follows the motion of the marker. If the refinement cuboids overlap with each other, they constitute a larger continuous region in the grid to be refined. The details of the refinement algorithm are explained with reference to Figure 17, where a refinement cuboid with dimensions $\Delta x_{\text{ref}} \cdot \Delta y_{\text{ref}} \cdot \Delta z_{\text{ref}}$ is centred around a point $\boldsymbol{P}$ in a grid with $N_G = 4$. We associate the dimensions of the refinement cuboid with the smallest grid size, that is, $\Delta x_{\text{ref}} = n_{\Delta x_{\text{ref}}} \Delta x_G$, $\Delta y_{\text{ref}} = n_{\Delta y_{\text{ref}}} \Delta y_G$, and $\Delta z_{\text{ref}} = n_{\Delta z_{\text{ref}}} \Delta z_G$. Unless otherwise specified, the default values are $n_{\Delta x_{\text{ref}}} = n_{\Delta y_{\text{ref}}} = n_{\Delta z_{\text{ref}}} = 4$ which have been found to work well for the type of problems studied here.

As shown by the red dashed square in Figure 17A, this generally leads to a refinement cuboid with corners that do not coincide with any of the grid nodes. Therefore, the region to be refined is modified into the blue shaded region bounded by a blue square. Furthermore, to ensure that the corners of the refined region coincide with nodes in the parent-level grid, which is important for communication between the grids, if necessary we extend the region so that its corner nodes have even node indices i, j, and k in x, y, and z-direction, respectively. Defining the corner coordinates of the original refinement cuboid as $(x, y, z)_{\min}$ and $(x, y, z)_{\max}$, the procedure for doing this is as follows: $(x, y, z)_{\min}$ is modified to $(i_0 \Delta x_G, j_0 \Delta y_G, k_0 \Delta z_G)$ where i_0 is the largest even node index satisfying $i \Delta x_G \leq x_{\min}$, j_0 is the largest even node index satisfying $j \Delta y_G \leq y_{\min}$ and k_0 is the largest even node index satisfying $k \Delta z_G \leq z_{\min}$. $(x, y, z)_{\max}$ is modified to $(i_1 \Delta x_G, j_1 \Delta y_G, k_1 \Delta z_G)$ where i_1 is the smallest even node index satisfying $i \Delta x_G \geq x_{\max}$, j_1 is the smallest even node index satisfying $j \Delta y_G \geq y_{\max}$ and k_1 is the smallest even node index satisfying $k \Delta z_G \geq z_{\max}$.

Having defined the region to be refined in the most refined grid level, we move to the parent-level grid $g = N_G - 1$. The region to be refined here is defined by expanding the refinement region in the child-level grid $g+1$ with a distance $\gamma \Delta x_g$, $\gamma \Delta y_g$ and $\gamma \Delta z_g$ in each direction, where $\gamma = 1$ is found to be an adequate default value. Choosing γ smaller than unity

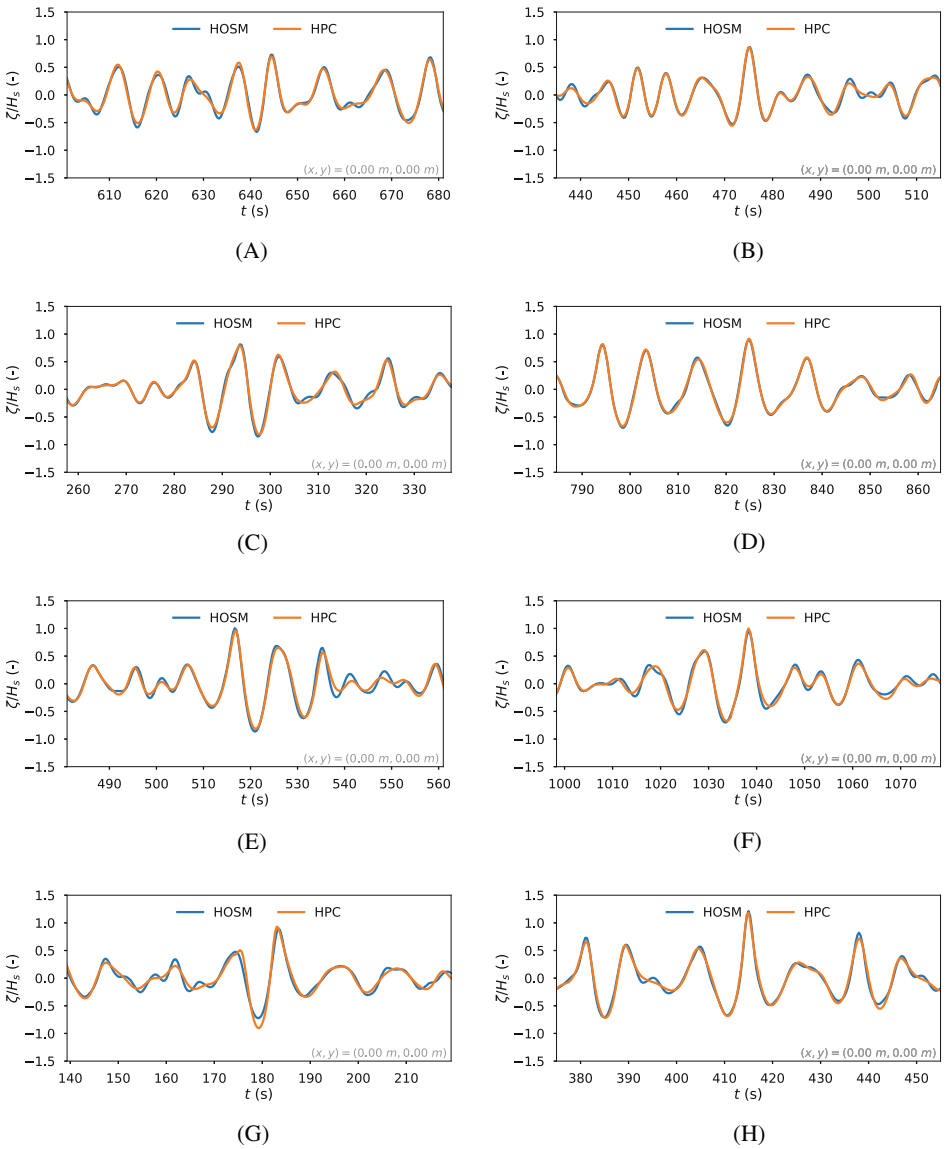

FIGURE 26 Time-series excerpts showing deepest troughs (left) and highest crests (right) in the middle of the tank for the deep-water cases in Table 4. (A) $H_s = 3.0$ m (deepest trough), (B) $H_s = 3.0$ m (highest crest), (C) $H_s = 5.0$ m (deepest trough), (D) $H_s = 5.0$ m (highest crest), (E) $H_s = 7.0$ m (deepest trough), (F) $H_s = 7.0$ m (highest crest), (G) $H_s = 9.0$ m (deepest trough), (H) $H_s = 9.0$ m (highest crest).

should be avoided, since it may lead to discontinuous refinement jumps. Using Figure 17B as example, this expansion leads to the refinement region bounded by the red dashed square in the $g = 3$ grid. However, the corner nodes of this region do not solely coincide with nodes with even indices (i, j, k), so that it has to be modified into the region bounded by the blue square following the procedure explained above. Then, the refinement region for the next parent grid ($g = 2$) is established in the same way in Figure 17C, finally resulting in the refined grid in Figure 17D.

Summarized, the AGR is controlled by five parameters:

1. N_G defines the number of refinement levels
2. $n_{\Delta x_{\text{ref}}}$, $n_{\Delta y_{\text{ref}}}$, and $n_{\Delta z_{\text{ref}}}$ define the size of the region with maximum refinement
3. γ defines the expansion rate going from child to parent grid levels.

The effect of the different parameters is illustrated in Figure 18. The AGR is efficient because it relates the refinement regions to indices i, j, and k instead of coordinates x, y, and z, and combines this with looking up in predefined connectivity matrices.

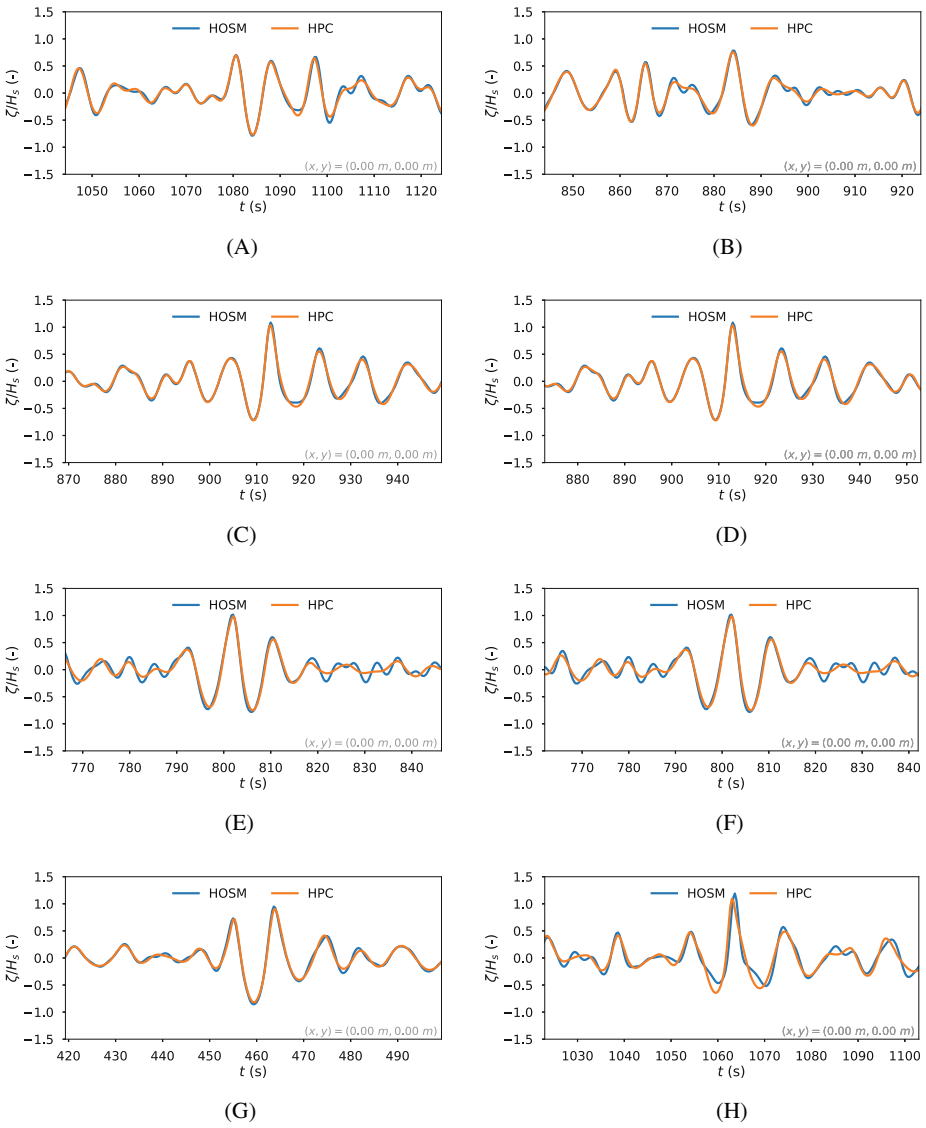

FIGURE 27 Time-series excerpts showing deepest troughs (left) and highest crests (right) in the middle of the tank for the intermediate-depth cases in Table 4. (A) $H_s = 3.0$ m (deepest trough), (B) $H_s = 3.0$ m (highest crest), (C) $H_s = 5.0$ m (deepest trough), (D) $H_s = 5.0$ m (highest crest), (E) $H_s = 7.0$ m (deepest trough), (F) $H_s = 7.0$ m (highest crest), (G) $H_s = 9.0$ m (deepest trough), (H) $H_s = 9.0$ m (highest crest).

In constructing the global matrix equation (12) for φ, the coupling relationship enforced for parent-child nodes with coinciding positions is

$$\varphi_{\text{child}} = \varphi_{\text{parent}}, \tag{19}$$

where φ_{child} and φ_{parent} are the values of φ in the child and parent nodes, respectively. For child nodes that do not coincide with nodes in the parent grid, equation (9) is used:

$$\varphi_{\text{child}} = \sum_{i=1}^{26} \left[\sum_{j=1}^{26} c_{j,i} f_j\left(\bar{\boldsymbol{x}}\right) \right] \varphi_i \Big|_{\text{parent}}, \tag{20}$$

where the right-hand side refers to a parent-level cell that contains the child node and no inactive nodes. The global matrix equation for the time derivative of φ is obtained by replacing φ with φ_t in the coupling relationships (19) and (20), leading to an unchanged global coefficient matrix $\boldsymbol{A}$.

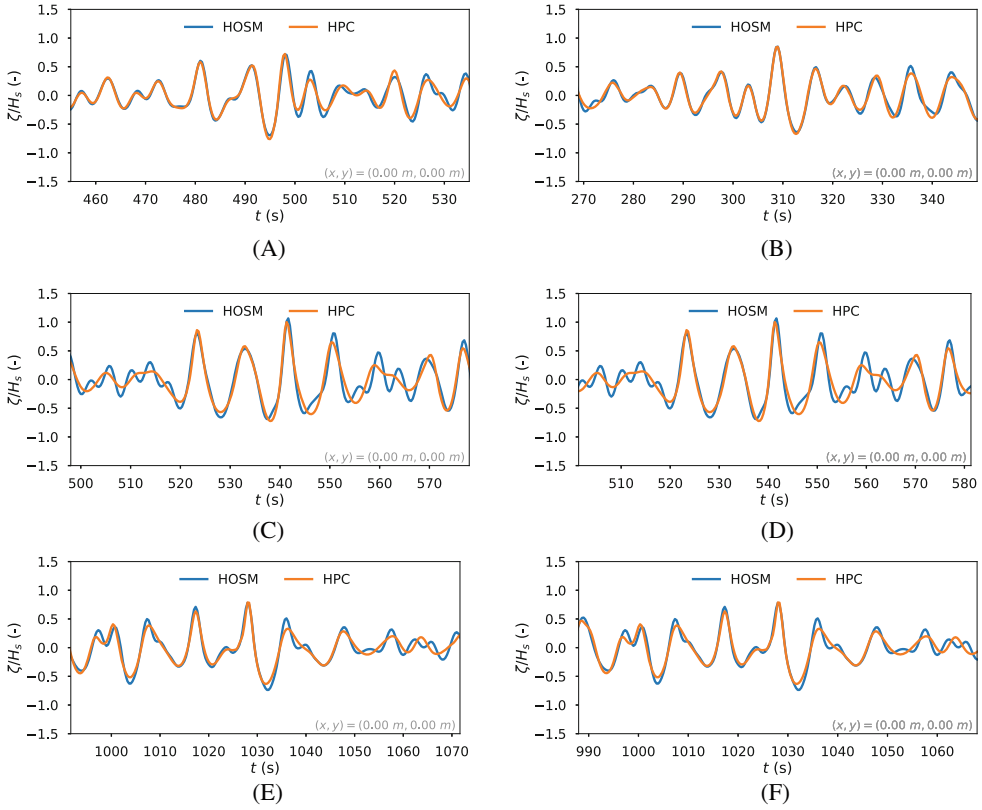

FIGURE 28 Time-series excerpts showing deepest troughs (left) and highest crests (right) in the middle of the tank for the shallow-water cases in Table 4. (A) $H_s = 3.0$ m (deepest trough), (B) $H_s = 3.0$ m (highest crest), (C) $H_s = 5.0$ m (deepest trough), (D) $H_s = 5.0$ m (highest crest), (E) $H_s = 7.0$ m (deepest trough), (F) $H_s = 7.0$ m (highest crest).

5.3 | Free-surface modeling with AGR

Semi-Lagrangian free-surface markers are distributed with coordinates corresponding to the horizontal coordinates of nodes in the most refined grid level ($g = N_G$). Since a refinement cuboid is attached to each marker, the grid is always refined around the free surface. The selection of free-surface cells and ghost nodes explained for a uniform grid in connection with Figure 4 is then separately applied in all grid-refinement levels. This means that there are ghost nodes above the free surface in each of the $g = 1, \ldots, N_G$ grids, using cells in their respective grids to form local expressions (9).

5.4 | Accuracy of global solution

Examples showing the distribution of absolute numerical errors of φ at, and underneath, the free surface with different values of N_G are given in Figure 19, where the hydrodynamic BVP is solved at a fixed time instant. The dimensions of the numerical domain are $L_x = 500$ m, $L_y = 100$ m and $L_z = 200$ m with the flat seabed located at $z = -h = -0.5L_z$. As analytical solution, a higher-order spectral method (HOSM) realization of a long-crested irregular sea state with significant wave height $H_s = 8$ m, spectral peak period $T_p = 10$ s and still-water level $z = 0$ is used. Denoting the reference velocity potential as φ_{ana} and the reference wave elevation as $\zeta_{\text{ana}}(x, y)$, the following conditions are enforced in the BVP for φ: $\varphi = \varphi_{\text{ana}}$ on the free surface with coordinates $(x, y, z) = (x, y, \zeta_{\text{ana}}(x, y))$, $\nabla\varphi \cdot \boldsymbol{n} = \nabla\varphi_{\text{ana}} \cdot \boldsymbol{n}$ on the rigid vertical walls $(x = \pm 0.5L_x \vee y = \pm 0.5L_y) \wedge z < \zeta_{\text{ana}}$ and $\nabla\varphi \cdot \boldsymbol{n} = 0$ on the flat seabed. All grids have a grid size of approximately $\lambda_p/\Delta x_1 \approx 7.5$ in the coarsest level, where λ_p is the wavelength corresponding to the spectral peak period. As a result, the

grid size at the free surface is approximately eight times smaller with $N_G = 4$ than with $N_G = 1$. All cells in all grids are cubic, and the remaining AGR settings are the same for all grids.

The errors in Figure 19 are computed in evenly distributed points that generally do not coincide with grid nodes or free-surface markers. Clearly, the AGR enhances the accuracy of the solution in and near the region that is refined (the free surface) as N_G increases, while the solution in the regions that are not refined remains largely unaffected. The AGR is therefore successful in its mission, which is to refine the solution locally and robustly couple the solution between regions with different grid sizes. With robustness we here refer to the fact that the numerical solution varies smoothly across regions in the domain with different grid sizes, and that notable local errors related to the exchange of information between different refinement levels in the grid are not observed. It is noted in Figure 19 that the errors are generally larger underneath wave crests due to large water particle velocities in these regions.

6 | NUMERICAL RESULTS

In the following, the performance of the NWT is examined for relevant wave-propagation cases. Results from systematic studies of regular waves and long-crested irregular waves are first presented, before results for an irregular, short-crested sea state are shown to approach towards realistic stormy ocean conditions.

To provide a reference solution in all the simulations discussed, the NWT is implicitly coupled with the open-source spectral wave data (SWD) application programming interface (API) developed by DNV available through Helmers and Gramstad.[27] The SWD API is also used to evaluate the reference solution anywhere in the domain where this is compared directly with the NWT results. For regular waves, the high-order stream function theory by Rienecker and Fenton[28] is taken as reference solution using the raschii Python package developed by Landet[29] to provide input files to the SWD API. Fifth-order HOSM realizations of wave spectra are used as reference solutions for irregular-wave cases, where the input files for the SWD API are generated using the DNV WAMOD software. The HOSM realizations use linear sea states based on discretization of analytical wave spectra with random phases and amplitudes as initial conditions. As time evolves, the energy distribution will shift to other frequencies due to nonlinear dispersion, modulation, and possibly wave breaking (a wave-breaking suppression model is included). This is in contrast to linear theory where the spectral density for each frequency is independent on time. To establish initial conditions providing the target non-linear spectral distribution is a research topic beyond the scope of this article and thus, deviations from the initial energy distribution is expected and will be more pronounced for steeper sea states. Obtaining a stationary energy distribution equal to a target distribution in the HOSM sea-state realizations will require additional effort, which is not within the scope of this article. This must be kept in mind when HPC and HOSM wave spectra for irregular sea states are directly compared. With the limited simulation times involved, redistribution of energy may also influence the HPC results. Hence, some discrepancies between HPC and HOSM wave spectra for certain wave-frequency ranges may be expected. An in-depth investigation of the limitations of the HOSM related to wave parameters and water depth can be found in the work by Ducrozet et al.[30]

6.1 | Regular waves

Nonlinear, regular waves propagating in positive x-direction are simulated in a wave tank with dimensions $L_x = 5\lambda$ and $L_y = 0.5\lambda$, where λ is the wavelength. Both deep water with $h = \lambda$ and intermediate water depth with $h = 0.3\lambda$ are considered, each with four different wave heights H between 2 and 5 m, equivalent to wave steepness $k\zeta_A$ between 0.13 and 0.31. k is here the wave number based on the linear dispersion relation and $\zeta_A = 0.5H$, thus underestimating the crest steepness for the highest waves. The simulations are carried out for 20 wave periods with a time step $\Delta t = T/40$, where T is the wave period. Four AGR levels are used for the deep-water simulations, whereas only three levels are used in intermediate depth. The latter is because the grid is refined all the way to the seabed with $N_G = 3$, so that additional levels become superfluous. This results in a grid refinement $\Delta x_3 = \lambda/32$ in deep water and $\Delta x_4 = \lambda/40$ in intermediate water. An overview of the main simulation parameters is given in Table 1. All grids are uniform with $\Delta x = \Delta y = \Delta z$, and the default settings $n_{\Delta x_{\text{ref}}} = n_{\Delta y_{\text{ref}}} = n_{\Delta z_{\text{ref}}} = 4$ and $\gamma = 1$ are used in all simulations. The free surface is initialized from the reference solution at $t = 0$, and $L_r = \lambda$ long relaxation zones are thereafter used towards the tank ends $x = \pm 0.5L_x$. Note that the relaxation zones are included in the tank length L_x, meaning that free wave propagation occurs in $x = \pm 0.5|L_x - L_r|$.

Figure 20 presents snapshots of the wave elevation in a cut through $y = 0$ at $t = 20T$, showing that, with adequately chosen simulation parameters the numerical (HPC) method is capable of simulating steep waves in both deep and intermediate water without introducing any visible errors in amplitude or phase.

Sensitivity study on number of AGR levels

We consider the steepest wave cases from Table 1, and perform a sensitivity analysis varying the number of refinements level N_G between 1 and 4 in deep water ($h = 50$ m) and between 1 and 3 in intermediate ($h = 15$ m) water depth. The refinement in the $g = 1$ grid is adjusted so that for all N_G the grid refinement at the free surface is equal to $\Delta x = \lambda/32$ in deep water and $\Delta x = \lambda/40$ in intermediate depth with $n_{\Delta x_{\text{ref}}} = n_{\Delta y_{\text{ref}}} = n_{\Delta z_{\text{ref}}} = 4$ and $\gamma = 1$. The simulation cases are summarized in Table 2 with the resulting grids shown in Figure 21, illustrating how different N_G results in a similar grid refinement near the free surface, but a significantly different total number of grid nodes in the domain.

In Figure 22, $L2$ errors of the free-surface elevation and the following variables on the free surface are plotted as a function of time: the velocity potential φ, the velocity components φ_x and φ_z, and the time derivative of the velocity potential φ_t. At each time step, the $L2$ errors are computed over all free-surface markers that are not inside any of the relaxation zones using Equation (16). In intermediate depth, the results with different N_G cannot be visually distinguished from each other, and in deep water, the $L2$ errors even tend to be reduced with increasing N_G. The reason for the latter observation is not completely understood, but the differences between $N_G = 1$ and 4 are generally small. While the AGR does not negatively influence the accuracy of the solution on the free surface, it leads to significantly improved computational efficiency. From Figure 23, it is seen that the simulations become approximately three times faster with $N_G = 3$ than with $N_G = 1$ in intermediate depth, and almost 20 times faster with $N_G = 4$ than with $N_G = 1$ in deep water. Thus, when using the AGR to refine the grid at the free surface, the possible CPU speed-up increases with water depth.

Sensitivity study on free-surface grid refinement

To investigate the influence of the grid spacing Δx_G at the free surface, we consider three different grid refinements for the steepest deep-water case from Table 1 with $N_G = 4$, and with all other parameters left unchanged (Table 3).

The time series of the $L2$ error of $\hat{\varphi}$ in Figure 24, computed in the same way as in Figure 22, indicate that a grid spacing $\Delta x_G = \lambda/48$ is necessary to give approximately converged results. However, by increasing $\lambda/\Delta x_G$ from 32 to 48, the computational time increases with almost a factor three. Although giving somewhat larger errors, their magnitude is considered acceptable, and $\Delta x_G = \lambda/32$ is thus considered a good compromise between accuracy and efficiency.

Scaling of numerical solution

The steepest regular-wave case with $H = 5$ m, $\lambda = 50$ m and $h = 50$ m is used to investigate how the numerical solution scales with the number of unknowns n in the global algebraic matrix system. n is here equal to the number of active nodes in the grid, i.e. all grid nodes that are either in water or free-surface ghost nodes. Refinement levels N_G from 1 (uniform grid) to 4 are considered, and the parameters are adjusted so that $\lambda/\Delta x_G = 32$ in all cases. n is varied by varying the dimensions L_x and L_y of the numerical domain.

In Figure 25, the following computational times are plotted as a function of n: t_{spilu} and t_{bicgstab} are the time used by the preconditioner and iterative solver, respectively, to solve the global algebraic matrix equation (12) one time, while t_{total} is the total computational time to complete one physical time step. Since a fourth-order Runge–Kutta time integration scheme is used, thus, the preconditioner is invoked four times and the iterative solver eight times (because also $\partial\varphi/\partial t$ is solved for) in each physical time step. t_{total} also includes all overhead time used to refine the grid in the AGR algorithm, to build the global coefficient matrix and to write data to file. The results show that the scaling is little influenced by the number of refinement levels, clearly indicating that the AGR algorithm is efficiently implemented.

6.2 | Long-crested irregular waves

After a thorough assessment for regular waves, the irregular long-crested sea states in Table 4 with waves propagating in the positive x-direction are considered. The spectral peak period T_p is fixed at 10 s, and the sea-state steepness parameter

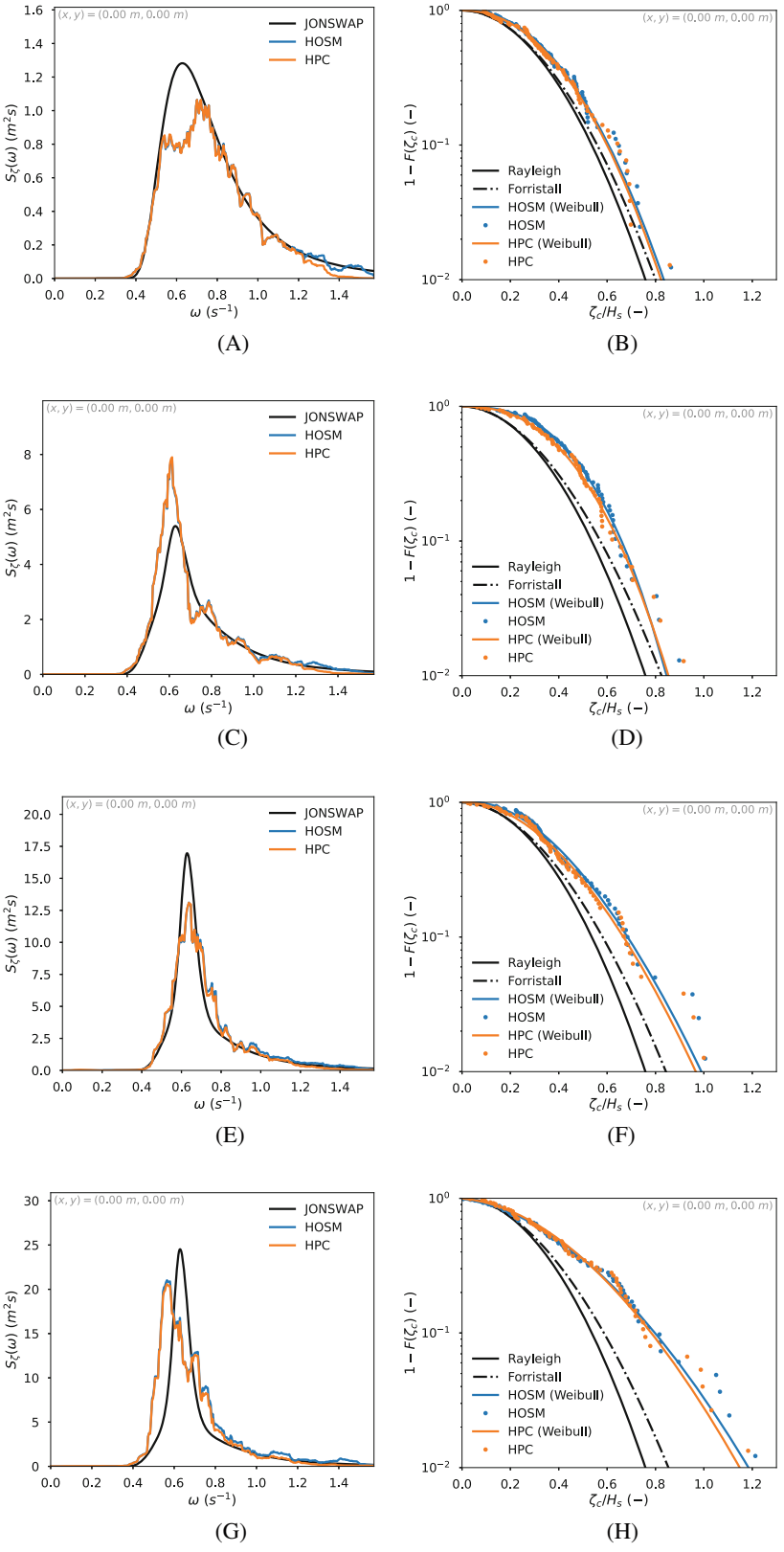

FIGURE 29 Wave spectra (left) and crest distributions (right) in the middle of the tank for the deep-water cases in Table 4. The empirical HPC and HOSM crest distributions are computed with Equation (21). (A) $H_s = 3.0$ m wave spectrum, (B) $H_s = 3.0$ m crest distribution, (C) $H_s = 5.0$ m wave spectrum, (D) $H_s = 5.0$ m crest distribution, (E) $H_s = 7.0$ m wave spectrum, (F) $H_s = 7.0$ m crest distribution, (G) $H_s = 9.0$ m wave spectrum, (H) $H_s = 9.0$ m crest distribution.

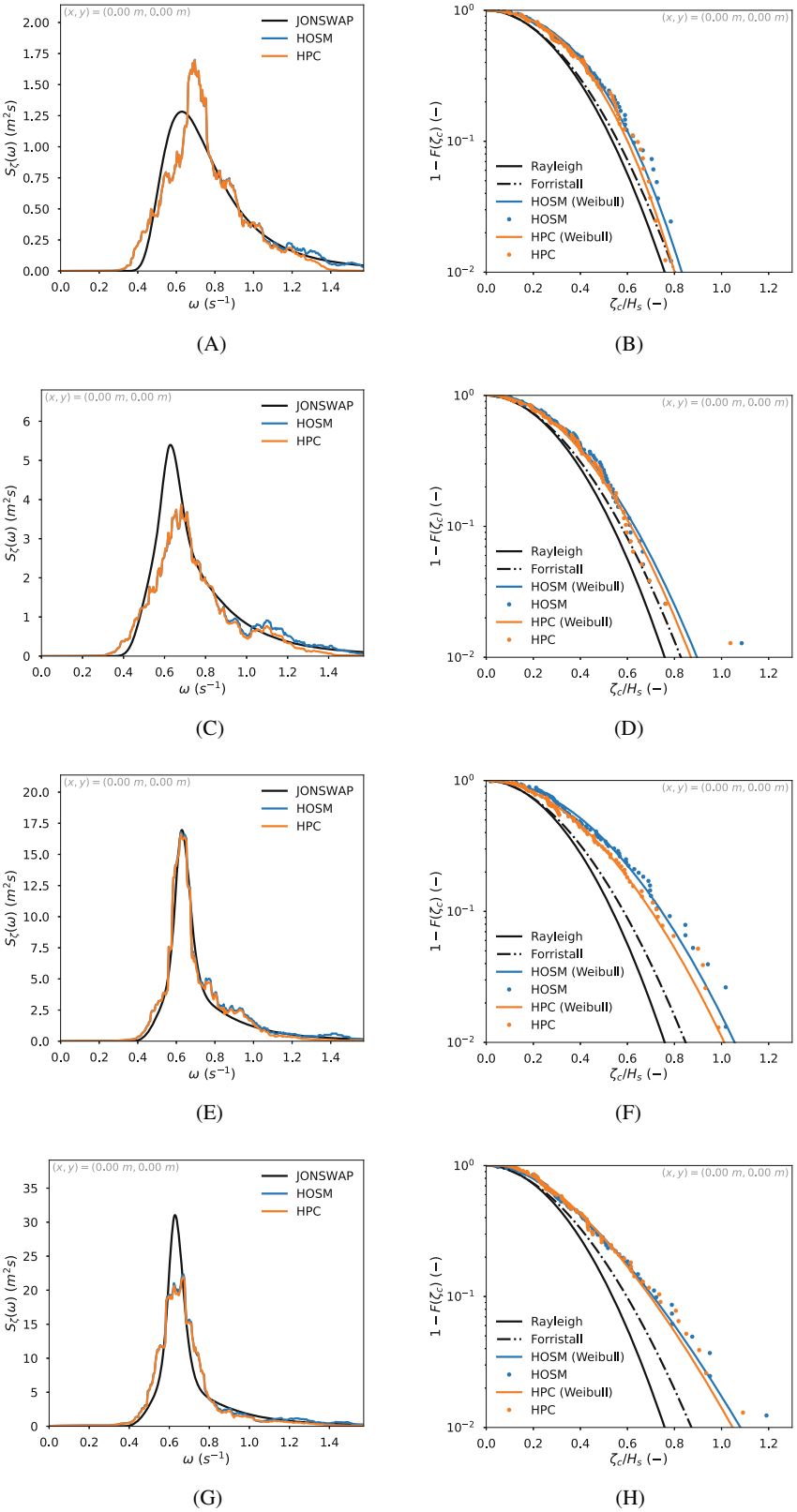

FIGURE 30 Wave spectra (left) and crest distributions (right) in the middle of the tank for the intermediate-depth cases in Table 4. The empirical HPC and HOSM crest distributions are computed with Equation (21). (A) $H_s = 3.0$ m wave spectrum, (B) $H_s = 3.0$ m crest distribution, (C) $H_s = 5.0$ m wave spectrum, (D) $H_s = 5.0$ m crest distribution, (E) $H_s = 7.0$ m wave spectrum, (F) $H_s = 7.0$ m crest distribution, (G) $H_s = 9.0$ m wave spectrum, (H) $H_s = 9.0$ m crest distribution.

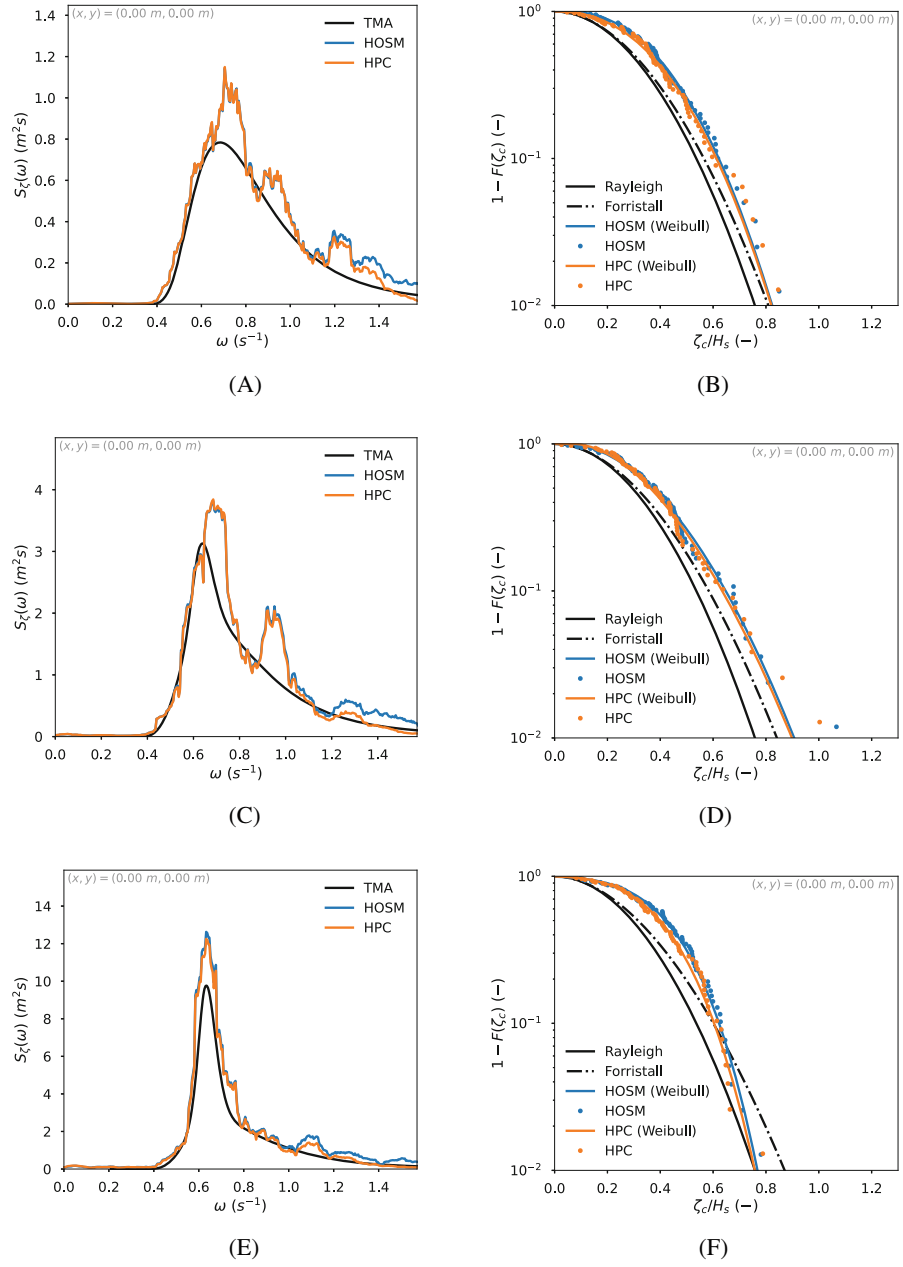

FIGURE 31 Wave spectra (left) and crest distributions (right) in the middle of the tank for the shallow-water cases in Table 4. The empirical HPC and HOSM crest distributions are computed with Equation (21). (A) $H_s = 3.0$ m wave spectrum, (B) $H_s = 3.0$ m crest distribution, (C) $H_s = 5.0$ m wave spectrum, (D) $H_s = 5.0$ m crest distribution, (E) $H_s = 7.0$ m wave spectrum, (F) $H_s = 7.0$ m crest distribution.

S_p is varied by increasing the significant wave heights H_s in steps. As an alternative steepness estimate, which is more recognizable compared to the regular-wave steepness $k\zeta_A$, Table 4 also includes $0.5k_pH_s$, where k_p is estimated using T_p in the linear dispersion relation. Three water depths are considered: deep water, with $h/\lambda_p \approx 0.93$, intermediate depth with $h/\lambda_p = 0.33$ and shallow water with $h/\lambda_p \approx 0.19$. The deep-water and intermediate-depth sea states are modeled with a JONSWAP wave spectrum, whereas the shallow-depth sea states are modeled with a TMA wave spectrum. According to DNV GL,[31] the maximum steepness with $T_p = 10$ s is $S_p \approx 0.059$ which corresponds to $H_s \approx 9.2$ m. This value is based on measurement data from the Norwegian Continental Shelf, but it is stated that the criterion is expected to have a wider general validity. The Norwegian Continental Shelf has water depths of 100 m or more, so that the criterion is expected to be especially relevant for the deep-water cases simulated here. For the intermediate and shallow depths, however, the

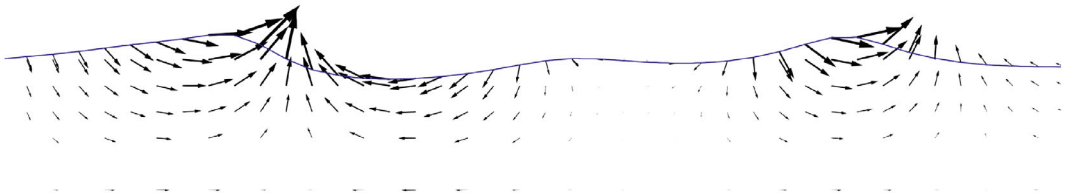

FIGURE 32 Snapshot of HPC velocity in the vicinity of large crests in simulation *Hs7m_Tp10s_shallow*.

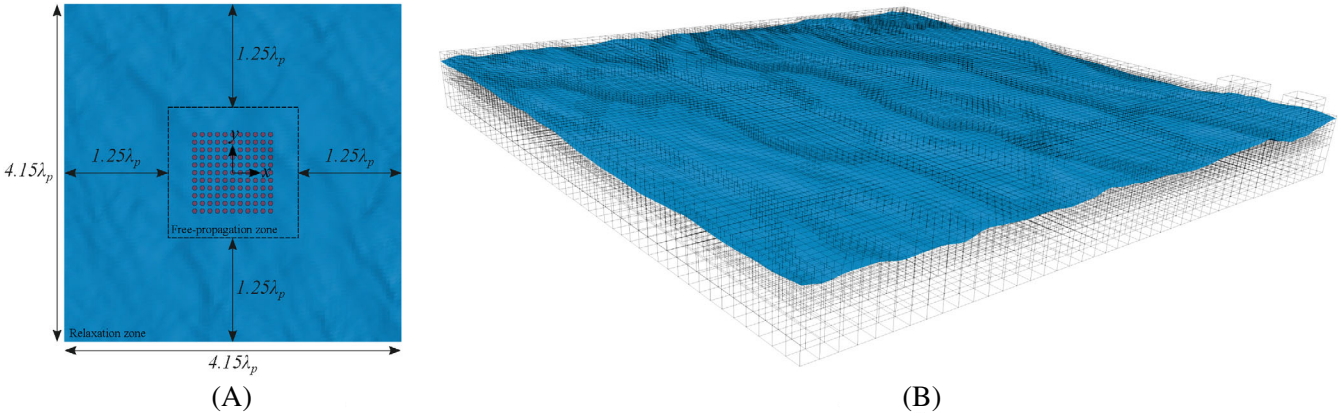

FIGURE 33 Definition of numerical domain used for short-crested irregular sea state with 121 wave probes in free-propagation zone (left) and snapshot of grid refinement around instantaneous free surface (right). (A) 2D domain definition, (B) 3D snapshot.

criterion may be less relevant since it is well known that waves become more nonlinear and break earlier under such conditions.

The simulations for the deep and intermediate depth cases are performed in a NWT with dimensions $L_x \approx 8\lambda_p$ and $L_y \approx 0.6\lambda_p$, and with slightly smaller horizontal dimensions in shallow water. The simulation length is a little more than 20 minutes, and the time step is $\Delta t \approx T_p/30$. The grid refinement around the free surface is $\Delta x_G \approx \lambda_p/25$. As listed in Table 5 this is achieved with $N_G = 4$ AGR levels in deep water, $N_G = 3$ in intermediate depth and $N_G = 2$ in shallow water and otherwise using default AGR settings $n_{\Delta x_{\text{ref}}} = n_{\Delta y_{\text{ref}}} = n_{\Delta z_{\text{ref}}} = 4$ and $\gamma = 1$. The reason for decreasing N_G with decreasing water depth is that the grid is refined all the way to the seabed, so that using a larger N_G will not improve the efficiency. $L_r = 1.5\lambda_p$ long relaxation zones are used towards $x = \pm 0.5L_x$.

Time-series excerpts of the simulated (HPC) wave elevation showing the deepest troughs and highest crests, respectively, are compared with the fifth-order (HOSM) reference solution for the deep-, intermediate- and shallow-water depth cases of Table 4 in Figures 26–28. Generally, the extreme troughs and crests in the HPC and HOSM time series are in good agreement throughout. However, with increasing H_s and decreasing water depth, the characteristic behavior of the time series differ. Notably, the HOSM time series seemingly contain more short-period oscillations just prior to and after deep troughs and high crests. This is reflected in the left columns of Figures 29–31, showing wave spectra estimated from the HPC and HOSM measured wave elevation time series. The processed wave spectra generally differ significantly from the target spectra, which is related to the relatively short time series. While the HPC and HOSM spectra agree well with each other for $\omega < 1$ s^{-1}, the HOSM spectra have more energetic high-frequency tails. For a wave frequency of $\omega = 1$ s^{-1}, there are approximately ten grid nodes per wavelength in the deep-water and intermediate-depth simulations, and approximately twelve in the shallow-water simulations. In deep water and intermediate water depth, the high-frequency tails of the HOSM spectra compare well with the target (JONSWAP) spectra, whereas in shallow water the HPC spectra's high-frequency tails compare better with the target (TMA) spectra. Sensitivity studies with increased spatial resolution in the HPC simulations indicate that the difference in high-frequency spectral energy between HPC and HOSM cannot be explained by the numerical discretization in the HPC simulations alone, and further investigation considering also the HOSM parameters is required for a complete understanding. This is however considered outside the scope of the present work, and is planned to be addressed in a subsequent dedicated study.

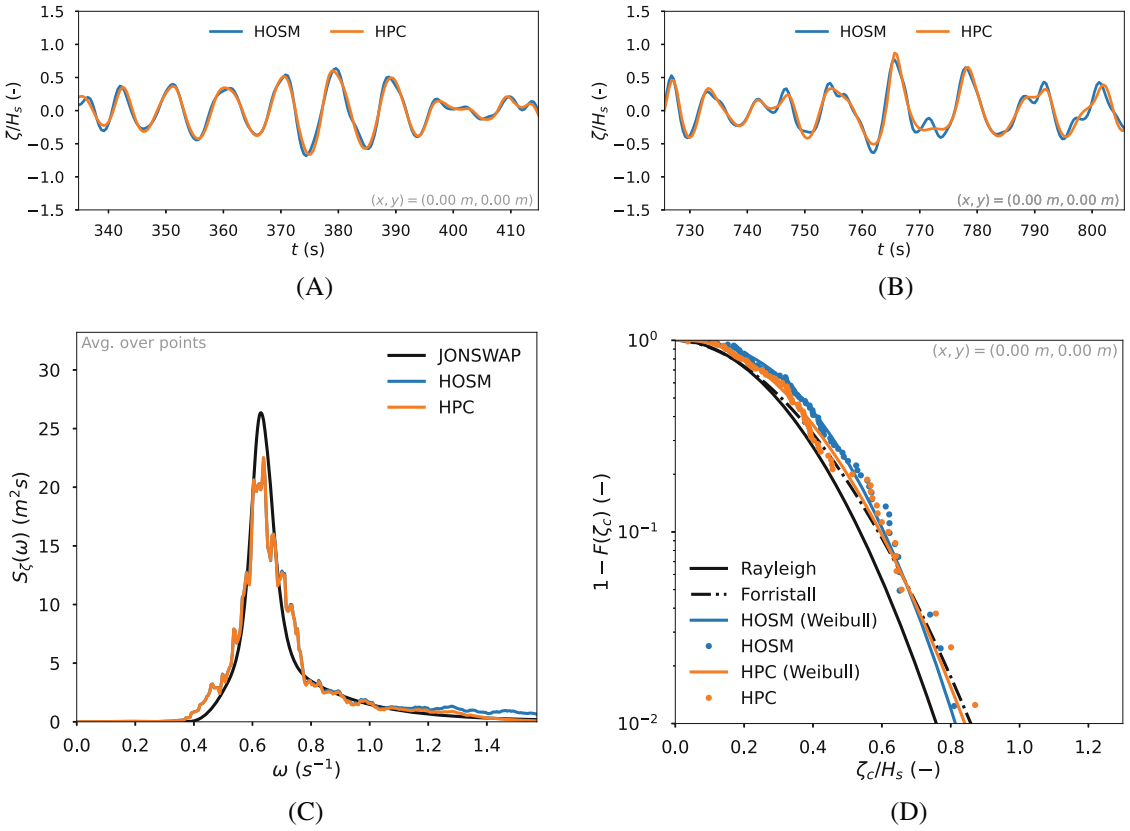

FIGURE 34 Selected results for short-crested irregular sea state. (A) Deepest trough, (B) highest crest, (C) wave spectrum (avg. of all probes), (D) crest distribution.

The right columns of Figures 29 through 31 present cumulative distribution functions (CDF) for wave-crest amplitudes $F(\zeta_c)$. The CDFs are plotted as $1 - F(\zeta_c)$, that is, the probability that an arbitrary wave-crest amplitude is larger than ζ_c. The discrete HPC and HOSM scatter points are estimated from the empirical CDF

$$F(\hat{\zeta}_{c,i}) = \frac{i}{N_c + 1}, \tag{21}$$

where N_c is the number of crests identified in the time series that are separated by at least T_p. The samples of selected crests are ordered in ascending order so that $\hat{\zeta}_{c,1} \leq \ldots \leq \hat{\zeta}_{c,i} \leq \hat{\zeta}_{c,N}$. Three-parameter Weibull distributions are fitted to the empirical CDFs with parameters estimated using the method of moments. HPC and HOSM CDFs are compared to the Rayleigh distribution and Forristall crest distributions, respectively. The Rayleigh CDF is given as

$$F(\zeta_c) = 1 - \exp\left[-\frac{1}{2}\left(\frac{\zeta_c}{\sigma_\zeta}\right)^2\right], \tag{22}$$

where σ_ζ is the standard deviation of the wave elevation. The Rayleigh CDF assumes that the instantaneous wave elevation is described by a Gaussian process corresponding to linear wave theory with $\sigma_\zeta = H_s/4$. To improve the statistical description of crest amplitudes, Forristall[32] proposed a model based on second-order theory to take into account wave non-linearity. The resulting CDF is

$$F(\zeta_c) = 1 - \exp\left[-\left(\frac{\zeta_c}{\alpha H_s}\right)^\beta\right], \tag{23}$$

with the parameters α and β expressed as functions of the water depth and the wave spectrum; different sets of parameters exist for short and long-crested waves. In deep water and with intermediate depth, the Forristall distribution compares well with the HPC and HOSM distributions when the sea-state steepness is moderate. With higher values of H_s, the Forristall distribution underestimates the empirically-fitted CDFs. In shallow water, the HPC and HOSM data compare well with the Forristall distribution for $H_s = 3$ m, whereas the Forristall distribution over-predicts the crest amplitudes for $H_s = 7$ m. This can presumably be explained by shallow-water wave mechanics that deviate significantly from the second-order assumption of Forristall.[32] The HOSM and HPC crest distributions are generally consistent with each other in all water depths, and particularly in shallow water the fitted HPC and HOSM Weibull distributions are in excellent agreement.

Figure 32 shows a snapshot of the HPC velocity under two large wave crests encountered in the shallow-water case with $H_s = 7$ m, which is the largest wave height successfully simulated with $h = 25$ m. The simulation for the shallow-water case with $H_s = 9$ m broke down due to numerical instability, and results for this case are therefore omitted. The encountered instability is likely related to wave-breaking, which is analogue with the 2D simulations by Hanssen et al.[21] Also there, numerical instability was encountered for a lower H_s in shallow water than in intermediate water depth.

The results for long-crested, irregular waves presented here indicate that while the Forristall crest distribution based on second-order theory compares well with the nonlinear HPC and HOSM crest distributions for sea states with limited steepness, it is insufficient for the cases with $H_s > 5$ m. For $h = 25$ m, the crest distributions estimated from the fully nonlinear numerical results differ in shape from the linear (Rayleigh) and Forristall distributions, likely due to nonlinear shallow-water wave mechanics. Although there is considerable quantitative uncertainty due to the limited duration of the simulations used to estimate the crest distributions, the results consistently indicate that fully nonlinear simulations are required to estimate extreme wave elevations in steep irregular sea states. Only small differences are observed between the HPC and HOSM results, expect for the occurrence of more energetic high-frequency wave components in the latter. However, while the present study only compares properties derived from the wave elevation, in their 2D analysis Hanssen et al.[21] also compared HPC and HOSM velocities. They observed that even when HPC and HOSM predict similar crest amplitudes, the two methods give significantly different velocities underneath the crest in steep sea states. Later research should therefore seek to compare HPC and HOSM wave kinematics and not only wave elevations.

6.3 | Short-crested irregular waves

While a systematic investigation of short-crested waves is left as future research, a single realization of a short-crested sea state described by a JONSWAP wave spectrum with $H_s = 9$ m, $T_p = 10$ s and $\gamma_s = 5$ is considered here. The main direction of wave propagation $\theta = 0°$ is along the positive x-axis with a $\cos^4(\theta)$ spreading function as defined in the DNV GL[33] guidelines, and the water depth is $h = 50$ m giving $\lambda_p = 151.3$ m. The sea state is simulated on a $4.15\lambda_p \times 4.15\lambda_p$ domain with $1.25\lambda_p$ long relaxation zones along each of the four vertical domain boundaries. As shown in Figure 33, this means that there is a $1.65\lambda_p \times 1.65\lambda_p$ square in the middle of the domain where the waves propagate freely. $N_G = 2$ grid refinement levels are used with $\Delta x_2 = \Delta y_2 = \Delta z_2 \approx \lambda_p/21.2$, equivalent to 89×89 free-surface markers. The simulation is carried out for 1300 s with a time step of $\Delta t = T_p/25$, and the free-surface elevation is recorded in the 121 wave probes indicated in Figure 33A.

Figure 34 provides key results for the short-crested simulation: (A) and (B) show, respectively, the time series of the wave elevation truncated around the deepest through and largest crest recorded in the numerical simulation in the middle of the basin, (C) shows the average of the wave spectra recorder in all 121 wave probes and (D) shows the crest distribution in the middle of the basin. In the latter case, the short-crested parameters of the Forristall distribution (23) are used. Similar to the long-crested simulation with $H_s = 9$ m in Figure 27, the time series of the simulated HPC wave elevation around the time of the deepest trough and highest crest are in fair agreement with the fifth-order HOSM results. By comparison, the absolute values of the extreme trough and crest are lower in short-crested than in long-crested waves. While the single deepest trough and highest crests may differ due to short-term variability related to the particular realization of the sea state, the empirical crest-distribution is in closer agreement with the Forristall distribution in short-crested than in long-crested waves. Although a more thorough investigation with increased simulation time and domain dimensions is needed, this indicates that wave spreading tends to reduce some of the nonlinear amplification that was observed in long-crested sea states. Similar to in long-crested waves, the HPC wave spectrum differs somewhat from the HOSM wave spectrum for the highest wave frequencies. However, the visible differences occur at slightly higher frequencies in short-crested waves despite the time step being slightly larger. Since nonlinear wave effects appear to be more pronounced

in long-crested waves, this may indeed suggest that the difference in high-frequency spectral energy is related to wave nonlinearity.

Computational time

For the irregular-wave simulation presented above, each physical time step takes approximately 80 s to complete on a standard laptop with an Intel Core i7 2.11 GHz processor, 32 GB RAM and four cores. The time spent on the most time-consuming operations, relative to the total computational time, is distributed approximately as follows:

(A) Updating AGR, identifying ghost cells and nodes and estimating free-surface velocities <10%
(B) Construction of global coefficient matrix 20%
(C) Computing the preconditioner matrix 35%
(D) Post-processing 20%.

The preconditioner used in (C) is included in the scipy.sparse.linalg package, which uses parallel processing. Nevertheless, other methods to solve the global algebraic matrix equations involved in the simulations may prove more efficient. Tasks (A), (B), and (D) are not parallelized, hence there is a potential to reduce the computational time by distributing these over multiple cores. Task (A), although not parallelized, takes less than 10% of the total computational time, where the majority of this time is spent identifying appropriate ghost cells and nodes. The limited fraction of the computational time used for (A), even in the case of an irregular, short-crested sea state on a large domain, is achieved through a dedicated effort to reduce the use of *for* and *while* loops as much as possible. Finally, it is noted that the computational time likely will be significantly reduced using a supercomputer, which may be required for a just comparison against NS solvers.

7 | CONCLUSIONS

The present work is devoted to developing a fully nonlinear 3D numerical wave tank (NWT) applicable for large-scale free-surface wave simulations when wave breaking, viscous-flow, or other violent flow phenomena are not of primary importance. In such scenarios, the assumptions of potential-flow theory are valid. The high-order harmonic polynomial cell (HPC) method is used to solve the governing Laplace equation for the velocity potential. This is combined with using an immersed-boundary modeling of the free surface and an adaptive grid-refinement (AGR) technique to increase the computational efficiency using Cartesian grids with cubic cells. These modeling techniques are used, amongst others, because it was shown that using cubic cells is strictly required to utilize the numerical accuracy available with the HPC method, qualitatively confirming previous findings from 2D. It was demonstrated that the AGR can reduce the computational time with as much as a factor of 20 without decreasing the accuracy of the numerical solution. With such improvements in numerical efficiency, the proposed NWT represents a framework that may be attractive in practical and not only academic applications.

For regular waves, the NWT results were in good agreement with a highly accurate stream-function theory solution. The regular-wave simulations included sensitivity studies to determine appropriate choices for the number of grid-refinement levels and grid spacing close to the free surface. Thereafter, simulations were performed for long-crested irregular waves with different values of the sea-state steepness and for three different water depths. With the higher-order spectral method (HOSM) as reference solutions, wave elevations in the form of time series, processed wave spectra and crest distributions were generally consistent. The wave-crest amplitudes differed significantly from linear and second-order theory for large significant wave heights. Finally, a single case for a steep short-crested irregular sea state was presented. Although more simulations are required to conclude, the results indicated that second-order theory provides a better estimate for the crest distribution in short-crested than in long-crested waves. The comparison with HOSM was qualitatively similar as in long-crested waves. An assessment against HOSM considering wave kinematics in addition to wave elevations is proposed as future work. Furthermore, such study should also seek to answer the slight differences in high-frequency energy observed in the irregular-wave spectra.

To further develop the proposed NWT, introducing capabilities to study wave-body interactions with fixed and floating structures is proposed. Enhancing the numerical efficiency, for example, making as much as possible of the numerical code parallel, and investigating more efficient numerical matrix solvers, is proposed to make the method more attractive

for practical use. Taking a different approach to improve the efficiency by dramatically reducing the dimensions of the numerical domain when wave-body interactions are considered, the solver can be used to represent the body-disturbed solution in a weak-scatterer approach. This is separated from the incident-wave solution, which may be represented by a nonlinear wave theory such as the HOSM used in the present work. This approach appears particularly attractive in 3D, knowing that the scattered waves diminish more rapidly away from a structure than in 2D.

ACKNOWLEDGMENTS

This work has been carried out at the Centre for Autonomous Marine Operations and Systems (AMOS). The Research Council of Norway (Norges Forskningsråd) is acknowledged as the main sponsor of AMOS. This work was supported by the Research Council of Norway through the Centre of Excellence funding scheme, Project number 223254-AMOS.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Finn-Christian Wickmann Hanssen https://orcid.org/0000-0002-6001-4645
Yanlin Shao https://orcid.org/0000-0002-9080-8438

REFERENCES

1. Haver S. A possible freak wave event measured at the Draupner Jacket January 1 1995. Proceedings of the Rogue waves 2004, Brest France, 2004:1-8.
2. Wang W, Kamath A, Martin T, Pákozdi C, Bihs H. A comparison of different wave modelling techniques in an open-source hydrodynamic framework. *J Marine Sci Eng*. 2020;8(7):526.
3. Shao Y-L, Faltinsen OM. Towards efficient fully-nonlinear potential-flow solvers in marine hydrodynamics. Proceedings of the 31st International Conference on Ocean, Offshore and Arctic Engineering (OMAE), Rio de Janeiro, Brazil, 2012:369–380.
4. Shao Y-L, Faltinsen OM. A harmonic polynomial cell (HPC) method for 3d Laplace equation with application in marine hydrodynamics. *J Comput Phys*. 2014;274:312-332.
5. Xu K, Shao Y-L, Gao Z, Moan T. A study on fully nonlinear wave load effects on floating wind turbine. *J Fluids Struct*. 2019a;88:216-240.
6. Xu K, Zhang M, Shao Y-L, Gao Z, Moan T. Effect of wave nonlinearity on fatigue damage and extreme responses of a semi-submersible floating wind turbine. *Appl Ocean Res*. 2019b;91:101879.
7. Ma S, Hanssen F-CW, Siddiqui MA, Greco M, Faltinsen OM. Local and global properties of the harmonic polynomial cell method: In-depth analysis in two dimensions. *Int J Numer Methods Eng*. 2018;113(4):681-718.
8. Hanssen F-CW, Bardazzi A, Lugni C, Greco M. Free-surface tracking in 2D with the harmonic polynomial cell method: two alternative strategies. *Int J Numer Methods Eng*. 2018;113(2):311-351.
9. Hanssen F-CW. *Non-linear wave-body interaction in severe waves*. PhD thesis, Norwegian University of Science and Technology, 2019.
10. Tong C, Shao Y-L, Bingham HB, Hanssen F-CW. An adaptive harmonic polynomial cell method with immersed boundaries: Accuracy, stability and applications. *Int J Numer Methods Eng*. 2021;122(12):2945-2980.
11. Liang H, Faltinsen OM, Shao Y-L. Application of a 2d harmonic polynomial cell (hpc) method to singular flows and lifting problems. *Appl Ocean Res*. 2015;53:75-90.
12. Strand IM. *Sea loads on closed flexible fish cages*. PhD thesis, Norwegian University of Science and Technology, 2018.
13. Robaux F, Benoit M. Modeling nonlinear wave-body interaction with the harmonic polynomial cell method combined with the immersed boundary method on a fixed grid. Proceedings of the 33rd International Workshop on Water Waves and Floating Bodies, 2018.
14. Shen Y, Greco M, Faltinsen Odd M, Ma S. Numerical study towards closed fish farms in waves using two harmonic polynomial cell methods. Proceedings of the 35th International Workshop on Water Waves and Floating Bodies, 2020.
15. Law YZ, Liang H, Santo H, Lim KY, Chan ES, eds. Numerical investigation of the physics of higher order effects generated by wave paddles. *International Conference on Offshore Mechanics and Arctic Engineering*. Vol 84386. American Society of Mechanical Engineers; 2020:V06BT06A069.
16. Liang H, Law YZ, Santo H, Chan ES. Effect of wave paddle motions on water waves. Proceeding of the 34th International Workshop on Water Waves and Floating Bodies, Newcastle, Australia, 2019.
17. Engsig-Karup AP, Bingham HB, Lindberg O. An efficient flexible-order model for 3d nonlinear water waves. *J Comput Phys*. 2009;228(6):2100-2118.
18. Jacobsen NG, Fuhrman DR, Fredsøe J. A wave generation toolbox for the open-source cfd library: Openfoam®. *Int J Numer Methods Fluids*. 2012;70(9):1073-1088.
19. Bihs H, Wang W, Pakozdi C, Kamath A. REEF3D::FNPF—a flexible fully nonlinear potential flow solver. *J Offshore Mech Arct Eng*. 2020;142(4):4.
20. Greco M. *A two-dimensional study of green-water loading*. PhD thesis, Norwegian University of Science and Technology, 2001.

21. Hanssen F-CW, Helmers JB, Greco M. A coupled harmonic polynomial cell and higher-order spectral method for nonlinear wave propagation. *International Conference on Offshore Mechanics and Arctic Engineering*. Vol 84409. American Society of Mechanical Engineers; 2020:V008T08A016.
22. Kuo JE, Wang H, Pickup S. Multidimensional least-squares smoothing using orthogonal polynomials. *Anal Chem*. 1991;63(6):630-635.
23. Ducrozet G, Engsig-Karup AP, Bingham HB, Ferrant P. A non-linear wave decomposition model for efficient wave–structure interaction. part a: Formulation, validations and analysis. *J Comput Phys*. 2014;257:863-883.
24. Berland J, Bogey C, Marsden O, Bailly C. High-order, low dispersive and low dissipative explicit schemes for multiple-scale and boundary problems. *J Comput Phys*. 2007;224(2):637-662.
25. Shao Y-L, Zheng Z, Liang H, Chen J. A consistent second-order hydrodynamic model in the time domain for floating structures with large horizontal motions. *Comput Aided Civ Inf Eng*. 2021;37(7):894-914.
26. Wang J, Faltinsen OM, Duan WY. A high-order harmonic polynomial method for solving the Laplace equation with complex boundaries and its application to free-surface flows. Part I: two-dimensional cases. *Int J Numer Methods Eng*. 2020;121(17):3893-3925.
27. Helmers JB, Gramstad O. spectral wave data (github repository), 2020. https://github.com/SpectralWaveData/spectral_wave_data.
28. Rienecker MM, Fenton JD. A fourier approximation method for steady water waves. *J Fluid Mech*. 1981;104:119-137.
29. Landet T. raschii 1.0.4. a python implementation of nonlinear regular wave theories (stokes 2nd to 5th order and fenton stream function waves), 2020. https://pypi.org/project/raschii/
30. Ducrozet G, Bonnefoy F, Perignon Y. Applicability and limitations of highly non-linear potential flow solvers in the context of water waves. *Ocean Eng*. 2017;142:233-244.
31. DNV GL. *Recommended Practice DNVGL-RP-C205. Environmental Conditions and Environmental Loads*. DNV GL; 2017.
32. Forristall GZ. Wave crest distributions: Observations and second-order theory. *J Phys Oceanogr*. 2000;30(8):1931-1943.
33. DNV GL. *Class Guideline DNVGL-CG-0130. Wave loads*. DNV GL; 2018.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Hanssen F-CW, Helmers JB, Greco M, Shao Y. A 3D fully-nonlinear potential-flow solver for efficient simulations of large-scale free-surface waves. *Int J Numer Methods Eng*. 2023;124(1):119-158. doi: 10.1002/nme.7115