A periodic fundamental solution method for efficient simulation of fully nonlinear surface gravity waves

Kaiyuan Shi[a,c], Renchuan Zhu[a], Yulong Li[b,*]

[a] *State Key Laboratory of Ocean Engineering, School of Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China*
[b] *Technology Centre for Offshore and Marine, Singapore (TCOMS), 118411, Singapore*
[c] *Yazhou Bay Institute of Deepsea Science and Technology, Hainan Research Institute, Shanghai Jiao Tong University, Hainan, 572024, China*

Abstract

Accurate simulation of fully nonlinear surface gravity waves remains challenging, as most numerical methods suffer from inadequate efficiency or reduced accuracy under strong nonlinearity. In this study, we propose a novel *periodic fundamental solution* (PFS) method for simulating strongly nonlinear waves, which advances the classical method of fundamental solutions by reducing the computational complexity from $O(N^3)$ to $O(N \log N)$. The PFS method transforms the free-surface Dirichlet problem into a linear system with a convolution-structured coefficient matrix by leveraging source periodicity and a tailored fitting procedure, allowing efficient solution via fast Fourier transforms combined with a preconditioned iterative solver. Numerical experiments show that the PFS approach typically converges within two to three iterations, preserves high accuracy even for wave steepness as large as $\varepsilon = 0.4$, and reconstructs subsurface velocity fields with greater stability than the Dirichlet–Neumann operator method. Overall, the PFS method provides a scalable and robust framework for long-term, large-domain simulations of fully nonlinear ocean waves in deep and intermediate water depths.

Keywords: Surface gravity waves, Fully nonlinear, Method of fundamental solution, Periodic fundamental solution, Dirichlet–Neumann operator

1. Introduction

Surface gravity waves constitute a fundamental aspect of oceanic environments, and their accurate simulation and prediction are essential to the field of ocean fluid mechanics. Understanding wave dynamics not only advances theoretical knowledge, but also informs practical applications such as coastal protection, design of marine structures, and navigation safety. Under extreme oceanic conditions, these waves may exhibit complex, highly nonlinear behaviors. Consequently, accurate numerical modeling of nonlinear surface gravity waves is critical for capturing their complex dynamics, deepening our understanding of wave evolution, and supporting reliable engineering safety assessments.

*Corresponding author. *E-mail address:* Li_Yulong@tcoms.sg

Based on the governing equations they employ, commonly used numerical approaches for simulating water waves include computational fluid dynamics (CFD) methods that resolve viscous flows [1–3], depth-integrated models [4–8], and potential flow methods[9–19]. Among these, viscous CFD methods solve the Navier–Stokes (N–S) equations by discretizing the entire flow field. They typically rely on interface-capturing techniques, such as the Volume of Fluid or Level Set methods, which enable precise tracking of the free surface and air–water interface. As a result, CFD approaches are particularly effective at capturing strongly nonlinear and complex hydrodynamic phenomena, including wave breaking, overturning, and bubble generation [1–3], and thus offer robust and physically accurate simulations. However, these methods face practical challenges, such as numerical dissipation and mass conservation errors at the air–water interface [20]. Moreover, their reliance on high-resolution grids and localized mesh refinement near the free surface results in substantial computational costs, which restricts their practicality for large-scale or long-duration simulations. Consequently, CFD methods are generally less suitable for large-scale or long-duration wave simulations and are instead primarily employed for detailed analyses of local wave flow fields where strongly nonlinear and viscous effects, such as wave breaking and overturning, are significant.

In contrast, depth-integrated models offer a more computationally efficient framework to simulate wave evolution. These models are formulated by vertical integration of the hydrodynamic equations, resulting in simplified equations that describe horizontal volumetric flux. Prominent examples include the shallow water equations [4], Boussinesq-type equations [5, 6], and more advanced forms such as the Green-Naghdi equations [7, 8]. Due to their efficiency, depth-integrated models are well suited for applications requiring broad spatial coverage and extended simulation periods, and are widely used in studies of nearshore wave evolution[6, 8]. However, standard shallow water equations are primarily applicable to long waves and shallow water environments [21]. Although fully nonlinear Boussinesq models [22, 23] and high-level Green-Naghdi equations [24, 25] have extended depth-integrated models to deep-water conditions, their reliance on high-order derivatives leads to wide discretization stencils and implicit linear systems that must be solved at each time step, substantially increasing computational cost.

In comparison to CFD methods, which directly solve the N–S equations, and depth-integrated models, which simplify the vertical flow structure, potential flow methods employ a different approach. By neglecting fluid viscosity, these methods describe the flow field using a velocity potential function and reduce the governing equations to the Laplace equation. The free surface is modeled explicitly, with wave evolution governed by kinematic and dynamic boundary conditions applied at the free surface. Potential flow methods offer an effective balance between computational efficiency and accuracy in simulating wave propagation, particularly in scenarios where viscous effects are minimal and wave breaking is absent. As a result, they are widely used in the study of surface gravity waves [16, 26–29].

A central consideration in the numerical implementation of potential flow theory is obtaining the velocity field (or equivalently, the Neumann condition for the velocity potential) at the free surface when the velocity potential is specified by a Dirichlet boundary condition. The classical boundary element method (BEM) [13, 26, 27] addresses this by constructing boundary integral equations based on Green's second identity, distributing Green's functions along the evolving free surface to directly solve for

the normal velocity at the boundary. However, the singularity of the Green's function makes the accuracy of BEM highly sensitive to the discretization scheme. To overcome this limitation, several researchers [30, 31] have developed the desingularized Rankine source method, later extended to nonlinear free-surface flows and wave–body interaction problems [32–34], now commonly regarded as a method of fundamental solutions (MFS). In this approach, free-space Green's functions are placed outside the boundary and the source strengths are determined to indirectly obtain the velocity potential, effectively avoiding boundary singularities and achieving higher accuracy for smooth free surfaces. Despite these improvements, both BEM and MFS require the assembly and solution of a dense $N \times N$ matrix at each time step as the free surface boundary evolves, which presents significant computational challenges. To improve efficiency, various acceleration techniques have been introduced. The fast multipole method (FMM) [16] accelerates the evaluation of Green's functions, theoretically reducing computational complexity to $O(N)$, though in practice, its intricate expansions and tree operations can introduce considerable overhead. Additionally, GPU-based approaches [17, 18] have been explored to further enhance computational performance.

Beyond the methods discussed above, the Dirichlet–Neumann operator (DNO) approach [11, 12], also known as the High-Order Spectral (HOS) method [9, 10, 28, 29, 35], efficiently maps the Dirichlet boundary condition of the velocity potential at the free surface to its Neumann counterpart. The DNO is a pseudo-spectral technique that perturbs the free surface and expands it in a Taylor series, explicitly calculating the velocity potential at each order. Leveraging fast Fourier transforms (FFT), the method achieves computational complexity of $O(N \log N)$, making it considerably faster than other potential flow approaches. For moderate wave steepness ($\varepsilon \leq 0.35$), the method exhibits exponential convergence $O(\varepsilon^n)$ and delivers exceptionally high simulation accuracy[9]. Consequently, it has become one of the most widely used numerical strategies for nonlinear wave simulation, with applications in numerical wave tanks [29], nonlinear evolution of irregular ocean waves [28, 35], and even capillary wave studies [36]. Similarly, Fructus et al. [14] integrated boundary integral and spectral expansion techniques to develop the Spectral Boundary Integral (SBI) method, which also solves the free surface problem with $O(N \log N)$ complexity and further improves the accuracy of fully nonlinear wave simulations. More recently, Wang [19] introduced the enhanced SBI (ESBI) method, extending these capabilities to wave simulations over complex sea bottom topography.

Despite the marked progress made with methods such as DNO, several limitations persist. The DNO technique, which expands the solution around the still water surface, becomes less effective when simulating strongly nonlinear wave evolution at high wave steepness [9]. Previous research [14, 37–39] has identified a tendency for ill-conditioning in the DNO method, particularly with higher-order expansions or under large wave steepness, often resulting in numerical instability or divergence. Nicholls and Reitich [37] systematically analyzed this issue and proposed refined expansion strategies to address it. In addition, although the DNO method enables efficient computation of the vertical velocity at the free surface, accurately reconstructing the flow field beneath remains challenging, which is also a limitation of the SBI methods. Bateman et al. [38] noted that the modal velocity field near the free surface, as computed by the DNO method, can be prone to significant high-frequency errors. To improve velocity field reconstruction, Bateman introduced the H_2 operator method. However, for cases involving strong nonlinearity and high wave steepness, the H_2 operator requires relatively

large number of additional auxiliary layers compared to the standard DNO approach, resulting in higher computational costs.

Distinct from BEM- and DNO-type methods, a third class of potential-flow approaches discretizes the entire fluid domain and solves the Laplace equation directly. The harmonic polynomial cell method [15] employs analytical solutions of the Laplace equation as local basis functions, attaining high-order accuracy on a sparse coefficient matrix. Yates and Benoit [40] combined high-order horizontal finite differences with either finite differences or Chebyshev spectral collocation in the vertical, with the spectral variant achieving exponential convergence. Klahn et al. [41] developed a three-dimensional pseudo-spectral method that truncates the vertical domain via an artificial boundary condition, substantially reducing the problem size while retaining high accuracy for strongly nonlinear waves. Eskilsson et al. [42] formulated a fully nonlinear model based on the spectral/hp element method. Compared with BEM/MFS, volumetric discretization yields a sparse system that avoids the $O(N^2)$ cost; unlike DNO-type methods, it makes no perturbation expansion about the still-water level and is therefore more robust for steep, strongly nonlinear waves. The cost is a substantially larger number of unknowns, so that even with $O(N)$ sparse solvers the per-step cost typically exceeds that of DNO methods.

Across these approaches, a persistent trade-off remains between computational efficiency and the ability to robustly capture strongly nonlinear wave dynamics: DNO-type methods are highly efficient but degrade at large steepness, whereas volumetric and boundary-integral formulations handle strong nonlinearity more reliably at a higher per-step cost. To address this gap, this work presents a numerical approach for fully nonlinear surface gravity waves in deep and intermediate water depths that matches the efficiency of the DNO method, while providing enhanced suitability for simulating strongly nonlinear waves and more robust flow field reconstruction. The proposed periodic fundamental solution (PFS) method builds upon the desingularized Rankine source method, a representative MFS [30, 31, 33], formulating the flow field as a superposition of fundamental solutions. By solving the Dirichlet boundary conditions on the free surface, PFS efficiently determines the source strengths and thus obtains the velocity distribution. Distinct from the conventional MFS, PFS transforms the key matrix-vector multiplication into a series of convolution operations, allowing FFT techniques to significantly accelerate computation. Combined with iterative algorithms and effective preconditioning, the method achieves rapid convergence, typically within 2–5 iterations. The overall computational complexity is $O(N \log N)$, which is comparable to, and in certain scenarios may surpass, that of the DNO method. Crucially, PFS combines this high computational efficiency with clear physical and mathematical interpretability. In addition to matching DNO in speed, PFS demonstrates two distinct advantages: reliable simulation of strongly nonlinear wave dynamics and efficient, precise reconstruction of velocity distributions below the wave surface.

The remainder of this paper is organized as follows. Section 2 outlines the governing equations and boundary conditions for surface gravity wave evolution in open water. Section 3 details the fundamental concepts and numerical implementation of the PFS method. Section 4 presents numerical simulations, emphasizing the effects of grid resolution and other parameters on the accuracy of the PFS method, and compares the computational results with those obtained using the DNO method. Finally, Section 5 summarizes the key results and discusses potential directions for future research.

2. Governing equations

In an idealized ocean of infinite extent, we establish a Cartesian coordinate system with the z-axis oriented vertically upward, setting the still water surface at $z = 0$. The fluid is assumed to be incompressible, homogeneous, and inviscid, with irrotational flow. Under these conditions, a velocity potential ϕ exists that satisfies the Laplace equation, and its gradient defines the fluid velocity field:

$$\nabla\phi = (u, v, w), \tag{1}$$

$$\nabla^2\phi = \nabla \cdot (u, v, w) = 0. \tag{2}$$

The domain is bounded above by a time-dependent free surface $z = \zeta$ and below by a flat seabed at $z = -d$. At the free surface, the boundary conditions can be expressed in Zakharov's form [43]:

$$\zeta_t = (1 + |\nabla_{\boldsymbol{x}}\zeta|^2)\phi_z - \nabla_{\boldsymbol{x}}\phi^S \cdot \nabla_{\boldsymbol{x}}\zeta \quad \text{on } z = \zeta(\boldsymbol{x}, t), \tag{3}$$

$$\phi_t^S = -g\zeta - \frac{1}{2}|\nabla_{\boldsymbol{x}}\phi^S|^2 + \frac{1}{2}(1 + |\nabla_{\boldsymbol{x}}\zeta|^2)\phi_z^2 \quad \text{on } z = \zeta(\boldsymbol{x}, t). \tag{4}$$

Here, g denotes gravitational acceleration, $\boldsymbol{x} \equiv (x, y)$ specifies horizontal coordinates, and $\nabla_{\boldsymbol{x}} \equiv (\partial/\partial x, \partial/\partial y)$ is the horizontal gradient operator. The wave elevation is given by $\zeta(\boldsymbol{x}, t)$, while the surface velocity potential $\phi^S(\boldsymbol{x}, t)$ is defined as $\phi^S(\boldsymbol{x}, t) = \phi(\boldsymbol{x}, \zeta(\boldsymbol{x}, t), t)$.

At the seabed, the fluid must satisfy the impermeable boundary condition:

$$\phi_z = 0 \quad \text{on } z = -d, \tag{5}$$

where d is the water depth.

Together, Eqs. (2)–(5) provide the governing equations and boundary conditions for nonlinear surface gravity waves in open water. Once the initial wave elevation ζ and surface velocity potential ϕ^S are specified, the time derivatives can be calculated using the free surface conditions (3) and (4), allowing simulation of the temporal evolution of the system. At each time step, given ζ and ϕ^S, the horizontal gradients $\nabla_{\boldsymbol{x}}\zeta$ and $\nabla_{\boldsymbol{x}}\phi^S$ are readily obtained. The crucial computational challenge is to map ϕ^S to $\phi_z\big|_{z=\zeta}$ under the appropriate boundary constraints.

The time-dependent nature of the free surface introduces additional complexity. Conventional methods, such as the BEM or MFS, require reconstruction of the coefficient matrix at every time step to accommodate the shifting boundary, in order to solve the Dirichlet problem and obtain ϕ_z at the free surface. Since these matrices cannot be reused across time steps, a new $N \times N$ matrix must be formed and a linear system solved at each iteration, resulting in substantial computational expense. Spectral approaches, such as those using the DNO, can reduce computational complexity to $O(N \log N)$, but typically rely on the assumption of small wave steepness. This limits their applicability for strongly nonlinear waves, where convergence may be difficult to achieve.

To overcome these limitations, the present work derives and introduces the PFS method, which leverages the fundamental solution approach for improved efficiency and broader applicability.

3. Periodic fundamental solution (PFS) method

3.1. Definition of the PFS

In this work, we introduce the PFS, a specialized class of fundamental solutions characterized by three key properties: (i) it satisfies the Laplace equation; (ii) it exhibits periodicity in the horizontal direction; and (iii) it contains periodic singularities, which motivates the term *periodic fundamental solution*.

To illustrate the concept, consider the two-dimensional case of an infinitely extended periodic domain with period length L. At each location $(nL, 0)$, where n is an integer, identical sources of the Laplace fundamental solution, $\ln r^2$, are placed. The resulting velocity potential is described by:

$$P(x,z) = P(x \pm nL, z) = \sum_{n\in\mathbb{Z}} \ln\left((x+nL)^2 + z^2\right) + C. \tag{6}$$

Although the series in Eq. (6) is formally divergent, it can be reformulated into a well-defined functional expression. By invoking Euler's infinite product formula, we obtain:

$$\sin\left(\frac{x \pm \mathrm{i}z}{L}\pi\right) = \frac{x \pm \mathrm{i}z}{L}\pi \prod_{n\in\mathbb{Z},\, n\neq 0} \left(1 + \frac{x \pm \mathrm{i}z}{nL}\right). \tag{7}$$

Finally, we arrive at the following closed-form expression for the two-dimensional PFS:

$$\begin{aligned} P(x,z) &= \ln\left[\prod_{n\in\mathbb{Z}} \left((x+nL)^2 + z^2\right)\right] + C \\ &= \ln\left[(x^2+z^2) \prod_{n\in\mathbb{Z},\, n\neq 0} \left(1 + \frac{x+\mathrm{i}z}{nL}\right)\left(1 + \frac{x-\mathrm{i}z}{nL}\right)\right] + C \\ &= \ln\left[\sin\left(\frac{x+\mathrm{i}z}{L}\pi\right)\sin\left(\frac{x-\mathrm{i}z}{L}\pi\right)\right] + C \\ &= \ln\left[\cosh\left(\frac{2\pi}{L}z\right) - \cos\left(\frac{2\pi}{L}x\right)\right] + C, \end{aligned} \tag{8}$$

where the divergent constant terms independent of x and z arising in the derivation have been absorbed into C. We note that the closed-form expression in Eq. (8) is not new; similar forms such as $\ln\left|2\sin\left(\frac{\pi}{L}(x+\mathrm{i}z)\right)\right|$ have appeared in the literature on quasi-periodic Green's functions [44], and a related derivation via Fourier transforms can be found in [45].

The approach extends naturally to three dimensions, where the periodic fundamental solution is constructed as a superposition of $1/r$ singularities:

$$P(\boldsymbol{x}, z) = P(\boldsymbol{x} + \boldsymbol{nL}, z) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} \frac{1}{\sqrt{|\boldsymbol{x} + \boldsymbol{nL}|^2 + z^2}} + C. \tag{9}$$

Here, $\boldsymbol{L} \equiv (L_x, L_y)$ defines the periodic domain lengths in the horizontal directions. The three-dimensional case corresponds to the classical lattice sum of $1/r$ singularities, which has been extensively studied in electrostatics, molecular simulation, and periodic scattering problems [46–48]. The function described in Eq. (9) does not admit an elementary closed-form, but its spatial representation can be numerically evaluated using the Ewald 2D method [46, 47] or the Lekner method [48]. Appendix A provides details on the numerical computation based on the Lekner Summation technique.

3.2. Discretization and determination of source strengths

Section 2 established the governing equations for nonlinear surface gravity waves under the potential flow framework. To simulate the temporal evolution of the free surface using Eqs. (3) and (4), it is essential to compute $\phi_z\big|_{z=\zeta}$ at every time step, based on the surface velocity potential ϕ^S. This process is typically the most computationally intensive part of potential flow simulations. Distinct from the spectral DNO approach, the PFS method addresses this challenge using fundamental solutions.

As illustrated in Fig. 1, the computational domain (with water depth d) is discretized into uniformly spaced grid points along the horizontal axis. Collocation points are placed on the free surface, with the corresponding wave elevation and surface potential denoted by ζ_i and ϕ_i^S, and coordinates $(\boldsymbol{x}_i, \zeta_i)$.

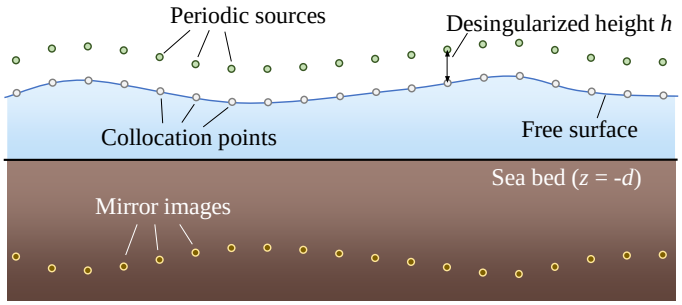


Figure 1: Layout of collocation points and periodic sources.

At a vertical offset h above each collocation point $(\boldsymbol{x}_i, \zeta_i)$, a PFS is placed. Additionally, a mirror image source of equal strength is positioned symmetrically with respect to the seabed ($z = -d$). This configuration inherently satisfies both the Laplace equation and the bottom boundary condition (5), though the Dirichlet condition at the free surface must still be imposed. To enforce this, we solve the following linear system to determine the source strengths:

$$\boldsymbol{G}\boldsymbol{\sigma} = \boldsymbol{\phi}^S. \tag{10}$$

Here, $\boldsymbol{G}$ is the coefficient matrix associated with the MFS whose entries G_{ij} quantify the velocity potential at field point i induced by source pair j:

$$\begin{aligned} G_{ij} = {} & P(\boldsymbol{x}_j - \boldsymbol{x}_i,\, \zeta_j - \zeta_i + h) \\ & + P(\boldsymbol{x}_j - \boldsymbol{x}_i,\, \zeta_j + \zeta_i + h + 2d), \end{aligned} \tag{11}$$

where the function P and its evaluation are described in Section 3.1.

After solving for the source strengths $\boldsymbol{\sigma}$, the velocity field at any position is reconstructed by superposing the gradients of all fundamental solutions and their mirror images. For a general point $(\boldsymbol{x}, z)$, the velocity components are given by:

$$(u, v, w) = \sum_j \sigma_j \left[\nabla P(\boldsymbol{x}_j - \boldsymbol{x},\, \zeta_j - z + h) + \nabla P(\boldsymbol{x}_j - \boldsymbol{x},\, \zeta_j + z + h + 2d)\right], \tag{12}$$

where σ_j is the strength of source j, and the sum runs over all sources.

3.3. Fast evaluation of source strengths

3.3.1. Convolution-based matrix representation

For clarity, this section focuses on the rapid computation of source strengths in the two-dimensional, infinite-depth case. The extension to three-dimensional and finite-depth domains follows a similar framework and is discussed in Appendix B.

Directly assembling the dense matrix $\boldsymbol{G}$ and solving $\boldsymbol{G\sigma} = \boldsymbol{\phi}^S$ scales as $O(N^3)$, which is prohibitive for large-scale applications. To overcome this bottleneck, we seek an efficient scheme for evaluating the matrix-vector product $\boldsymbol{G\sigma}$, which enables the use of iterative solvers to compute $\boldsymbol{\sigma}$ with controllable accuracy. A natural idea is to apply a Taylor expansion to $\boldsymbol{G}$ near the still water surface. Interestingly, this leads to a formulation that is identical to the DNO method, as detailed in Appendix C. In this work, we introduce a convolution-based functional fitting strategy for the fast and accurate evaluation of $\boldsymbol{G\sigma}$.

Consider the elements of $\boldsymbol{G}$, given by $G_{ij} = P_{ij}(x_j - x_i, h + z_j - z_i)$. Owing to the inherent periodicity of the PFS in the horizontal direction and the uniform grid arrangement, the first argument in G_{ij} (horizontal source-field separation) induces a circulant structure. In contrast, the second argument in G_{ij} (vertical separation) is anti-symmetric and non-periodic. This separation motivates a variable-splitting approach, allowing $\boldsymbol{G}$ to be efficiently represented and computed as a sum of convolutions.

For simplicity, in the two-dimensional, infinite-depth case, we introduce the following notation:

$$P_{ij}(z) = P(x_j - x_i, h - z). \tag{13}$$

Given fixed horizontal positions $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we approximate $P_{ij}(z)$ over the interval $[z_{\min}, z_{\max}]$. Let $\mathcal{V}_n = \mathrm{span}\{\psi_1(z), \psi_2(z), \ldots, \psi_n(z)\}$ denote a set of linearly independent real-valued functions. The fitted representation is

$$\hat{P}_{ij}(z) = P_{ij}(0) + \sum_{l=1}^{n} a_{ij}^{(l)} \psi_l(z), \tag{14}$$

where the coefficients $a_{ij}^{(l)}$ are obtained via least squares:

$$\boldsymbol{M}\boldsymbol{a}_{ij} = \boldsymbol{b}_{ij}, \tag{15}$$

with

$$M_{lm} = \langle \psi_l, \psi_m \rangle, \qquad b_{ij}^{(l)} = \langle \psi_l, P_{ij} - P_{ij}(0) \rangle, \tag{16}$$

and the inner product defined by

$$\langle f, g \rangle := \int_{z_{\min}}^{z_{\max}} f(z) g(z)\, \mathrm{d}z. \tag{17}$$

The basis $\mathcal{V}$ may be chosen as polynomials ($\mathcal{V} = \mathrm{span}\{z, z^2, z^3, \ldots\}$) or Fourier modes ($\mathcal{V} = \mathrm{span}\{\sin(kz), \cos(kz) - 1, \sin(2kz), \cos(2kz) - 1, \ldots\}$). Increasing the number of basis functions improves accuracy but increases computational cost. In this work, $n = 4$ is found sufficient, with

$$\begin{aligned} \psi_1(z) &= \sin(k_0 z), \\ \psi_2(z) &= \cos(k_0 z) - 1, \\ \psi_3(z) &= \sin(2k_0 z), \\ \psi_4(z) &= \cos(2k_0 z) - 1, \end{aligned} \tag{18}$$

where $k_0 = \pi/L$ is the fundamental wavenumber.

By expanding the basis and grouping terms, each matrix element G_{ij} can be approximated as

$$\begin{aligned} G_{ij} &\approx P_{ij}(0) + \sum_{l=1}^{4} a_{ij}^{(l)} \psi_l(\zeta_i - \zeta_j) \\ &= P_{ij}(0) + \sin(k_0\zeta_i) a_{ij}^{(1)} \cos(k_0\zeta_j) - \cos(k_0\zeta_i) a_{ij}^{(1)} \sin(k_0\zeta_j) \\ &+ \cos(k_0\zeta_i) a_{ij}^{(2)} \cos(k_0\zeta_j) + \sin(k_0\zeta_i) a_{ij}^{(2)} \sin(k_0\zeta_j) - a_{ij}^{(2)} \\ &+ \sin(2k_0\zeta_i) a_{ij}^{(3)} \cos(2k_0\zeta_j) - \cos(2k_0\zeta_i) a_{ij}^{(3)} \sin(2k_0\zeta_j) \\ &+ \cos(2k_0\zeta_i) a_{ij}^{(4)} \cos(2k_0\zeta_j) + \sin(2k_0\zeta_i) a_{ij}^{(4)} \sin(2k_0\zeta_j) - a_{ij}^{(4)}. \end{aligned} \tag{19}$$

This expansion separates the horizontal and vertical dependencies. Due to the periodicity of the PFS, the matrix $\boldsymbol{P}_0$, whose elements are $P_{ij}(0)$, and the matrices $\boldsymbol{A}^{(l)}$, with elements $a_{ij}^{(l)}$, both exhibit circulant structure. The fitted matrix, denoted $\hat{\boldsymbol{G}}$, can then be succinctly expressed as

$$\boldsymbol{G} \approx \hat{\boldsymbol{G}} = \mathcal{C}(\boldsymbol{a}_0) + \Re\left[e^{\mathrm{i}k_0\boldsymbol{\zeta}}\,\mathcal{C}(\boldsymbol{a}_1)\,e^{-\mathrm{i}k_0\boldsymbol{\zeta}}\right] + \Re\left[e^{\mathrm{i}2k_0\boldsymbol{\zeta}}\,\mathcal{C}(\boldsymbol{a}_2)\,e^{-\mathrm{i}2k_0\boldsymbol{\zeta}}\right], \tag{20}$$

where $e^{\mathrm{i}k\boldsymbol{\zeta}} = \mathrm{diag}(e^{\mathrm{i}k\zeta_i})$ is diagonal, and $\mathcal{C}(\boldsymbol{a}_{0,1,2})$ are circulant matrices generated from kernel vector $\boldsymbol{a}_{0,1,2}$:

$$\begin{aligned} \mathcal{C}(\boldsymbol{a}_0) &= \boldsymbol{P}_0 - \boldsymbol{A}^{(2)} - \boldsymbol{A}^{(4)}, \\ \mathcal{C}(\boldsymbol{a}_1) &= \boldsymbol{A}^{(2)} - \mathrm{i}\boldsymbol{A}^{(1)}, \\ \mathcal{C}(\boldsymbol{a}_2) &= \boldsymbol{A}^{(4)} - \mathrm{i}\boldsymbol{A}^{(3)}. \end{aligned} \tag{21}$$

Because circulant matrices and diagonal matrices interact efficiently under convolution, the resulting matrix-vector product $\hat{\boldsymbol{G}}\boldsymbol{\sigma}$ can be computed as

$$\begin{aligned} \hat{\boldsymbol{G}}\boldsymbol{\sigma} &= \boldsymbol{a}_0 \circledast \boldsymbol{\sigma} + \Re\left[e^{\mathrm{i}k_0\boldsymbol{\zeta}}(\boldsymbol{a}_1 \circledast (e^{-\mathrm{i}k_0\boldsymbol{\zeta}}\boldsymbol{\sigma})) + e^{\mathrm{i}2k_0\boldsymbol{\zeta}}(\boldsymbol{a}_2 \circledast (e^{-\mathrm{i}2k_0\boldsymbol{\zeta}}\boldsymbol{\sigma}))\right] \\ &= \mathcal{F}^{-1}\tilde{\boldsymbol{a}}_0\tilde{\boldsymbol{\sigma}} + \Re\left[e^{\mathrm{i}k_0\boldsymbol{\zeta}}\mathcal{F}^{-1}\tilde{\boldsymbol{a}}_1\mathcal{F}(e^{-\mathrm{i}k_0\boldsymbol{\zeta}}\boldsymbol{\sigma}) + e^{\mathrm{i}2k_0\boldsymbol{\zeta}}\mathcal{F}^{-1}\tilde{\boldsymbol{a}}_2\mathcal{F}(e^{-\mathrm{i}2k_0\boldsymbol{\zeta}}\boldsymbol{\sigma})\right], \end{aligned} \tag{22}$$

where $\circledast$ denotes circular convolution, $\Re$ denotes the real part, and $\mathcal{F}, \mathcal{F}^{-1}$ are the discrete Fourier transform and its inverse. The tilde notation $\tilde{\boldsymbol{a}} = \mathcal{F}\boldsymbol{a}$ indicates the wavenumber domain representation. Direct computation of $\boldsymbol{G}\boldsymbol{\sigma}$ scales as $O(N^2)$. By exploiting three FFT-based convolutions, the computational cost is reduced to $O(N \log N)$, significantly improving efficiency.

3.3.2. Fitting error analysis

For surface gravity waves observed in nature, the following inequality holds:

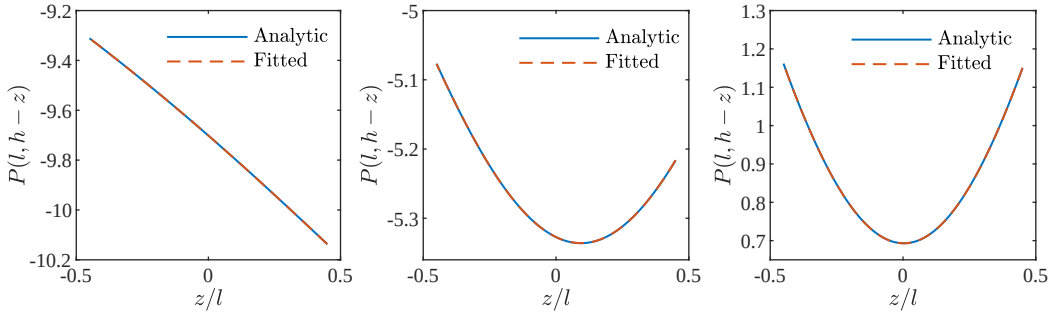
$$\frac{|\zeta_j - \zeta_i|}{|\boldsymbol{x}_j - \boldsymbol{x}_i|} < \max\left(|\nabla\zeta|\right) = \epsilon < \epsilon_{\max}. \tag{23}$$

Here, ϵ denotes the maximum wave steepness in the simulation, while $\epsilon_{\max}$ is the physical threshold for wave breaking, which is the largest slope the free surface ζ can achieve before instability occurs. Typically, $\epsilon_{\max}$ is less than 0.45 [49]. This physical constraint allows the bounds in Eq. (17) to be set according to the horizontal separation

between source and field points. Specifically, in two dimensions, for any $P(l, h-z)$, the permissible range for z is $[-\epsilon l,\ \epsilon l]$, where ϵ is the current maximum steepness.

To improve computational efficiency, convolution kernels corresponding to different steepness values can be precomputed. During simulation, the kernel that matches the instantaneous maximum steepness is selected at each time step, allowing for both fast and accurate evaluation of $\hat{\boldsymbol{G}}\boldsymbol{\sigma}$ while eliminating redundant kernel generation.

As an illustration, Fig. 2 shows both $P(l, h-z)$ and its fitted curves using the basis function space $\mathcal{V}$ defined in Eq. (18), for several values of horizontal distance l (with $L = 1024$, $h = 1.5$, and $\epsilon = 0.45$). Within the interval $z \in [-\epsilon l,\ \epsilon l]$, the four chosen basis functions yield highly accurate approximations. Figure 3 shows the L_∞ fitting error over $z \in [-\epsilon l,\ \epsilon l]$ for varying l, normalized by $||\boldsymbol{P}_0||_\infty$ for comparison. As shown, the proposed fitting strategy delivers high accuracy.



(a) $l = 1$ (b) $l = 16$ (c) $l = 512$

Figure 2: Comparison between $P(l, h-z)$ and fitted curves for various l values ($L = 1024$, $h = 1.5$, $\epsilon = 0.45$).

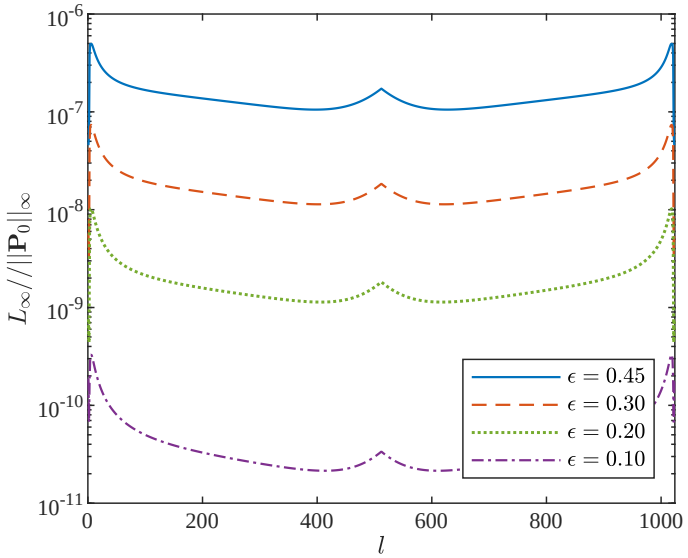


Figure 3: L_∞ error between $P(l, h-z)$ and the fitted function over $z \in [-\epsilon l,\ \epsilon l]$ for different values of l (with $L = 1024$, $h = 1.5$, and $\epsilon = 0.45$).

Given that four basis functions already achieve excellent accuracy with very modest computational overhead, further increasing the number of basis functions offers diminishing returns. For waves with low steepness, the fitting error is negligible; for

steep waves, the error from the fundamental solution method itself dominates. Thus, pursuing even higher fitting precision is generally unnecessary, as further discussed in Section 4.

3.3.3. Accelerated iterative solver

To efficiently solve the linear system $\hat{\boldsymbol{G}}\boldsymbol{\sigma} = \boldsymbol{\phi}^S$, we employ the generalized minimal residual (GMRES) method. Recognizing the similarity between $\hat{\boldsymbol{G}}$ and $\boldsymbol{P}_0$, we use $\boldsymbol{P}_0$ as a preconditioner to enhance the convergence rate.

The preconditioned GMRES system can be written as:

$$\boldsymbol{P}_0^{-1}\hat{\boldsymbol{G}}\boldsymbol{\sigma} = \boldsymbol{P}_0^{-1}\boldsymbol{\phi}^S. \tag{24}$$

During each GMRES iteration, the required matrix-vector products are computed using only 7 FFT or inverse FFT operations. The process can be expressed as:

$$\begin{aligned}(\boldsymbol{P}_0^{-1}\hat{\boldsymbol{G}})\boldsymbol{\sigma} = \mathcal{F}^{-1}\tilde{\boldsymbol{p}}_0^{-1}\mathcal{F}\Re\Big[&e^{\mathrm{i}k_0\boldsymbol{\zeta}}\mathcal{F}^{-1}\tilde{\boldsymbol{a}}_1\mathcal{F}(e^{-\mathrm{i}k_0\boldsymbol{\zeta}}\boldsymbol{\sigma}) \\ &+e^{\mathrm{i}2k_0\boldsymbol{\zeta}}\mathcal{F}^{-1}\tilde{\boldsymbol{a}}_2\mathcal{F}(e^{-\mathrm{i}2k_0\boldsymbol{\zeta}}\boldsymbol{\sigma})\Big] + \mathcal{F}^{-1}\tilde{\boldsymbol{p}}_0^{-1}\tilde{\boldsymbol{a}}_0\mathcal{F}\boldsymbol{\sigma}.\end{aligned} \tag{25}$$

where $\boldsymbol{p}_0$ denotes the convolution kernel corresponding to the circulant matrix $\boldsymbol{P}_0$.

The initial guess for the solution is chosen as:

$$\boldsymbol{\sigma} = \mathcal{F}^{-1}\tilde{\boldsymbol{p}}_0^{-1}\mathcal{F}\boldsymbol{\phi}^S. \tag{26}$$

In practice, this strategy yields rapid convergence. Numerical experiments indicate that restricting the number of GMRES iterations to two or three provides an effective balance between solution accuracy and computational efficiency (see Section 4 for details).

3.4. Evaluation of velocity fields

3.4.1. Vertical velocity at the free surface

To advance the free surface in time using the kinematic and dynamic boundary conditions (Eqs. (3) and (4)), it is necessary to evaluate the vertical velocity component at the free surface, $\boldsymbol{\phi}_z\big|_{z=\zeta}$. At the collocation points, this is given by:

$$\boldsymbol{\phi}_z\big|_{z=\zeta} = \boldsymbol{G}_z\boldsymbol{\sigma}, \tag{27}$$

where the matrix $\boldsymbol{G}_z$ is the derivative of $\boldsymbol{G}$ with respect to z. Following the strategy described in Section 3.3.1, $\boldsymbol{G}_z$ is efficiently approximated as a sum of four basis functions, requiring only three convolution operations:

$$\boldsymbol{\phi}_z\big|_{z=\zeta} \approx \boldsymbol{a}_0^{(z)} \circledast \boldsymbol{\sigma} + \Re\big[e^{\mathrm{i}k_0\boldsymbol{\zeta}}(\boldsymbol{a}_1^{(z)} \circledast (e^{-\mathrm{i}k_0\boldsymbol{\zeta}}\boldsymbol{\sigma})) + e^{\mathrm{i}2k_0\boldsymbol{\zeta}}(\boldsymbol{a}_2^{(z)} \circledast (e^{-\mathrm{i}2k_0\boldsymbol{\zeta}}\boldsymbol{\sigma}))\big]. \tag{28}$$

Here, $\boldsymbol{a}_{0,1,2}^{(z)}$ are the discrete convolution kernels obtained by sampling the continuous kernel functions $a_{0,1,2}^{(z)}$ on the computational grid.The fitting function,

$$a_0^{(z)}(l) + \Re\left\{a_1^{(z)}(l)e^{\mathrm{i}k_0 z} + a_2^{(z)}(l)e^{\mathrm{i}2k_0 z}\right\}, \tag{29}$$

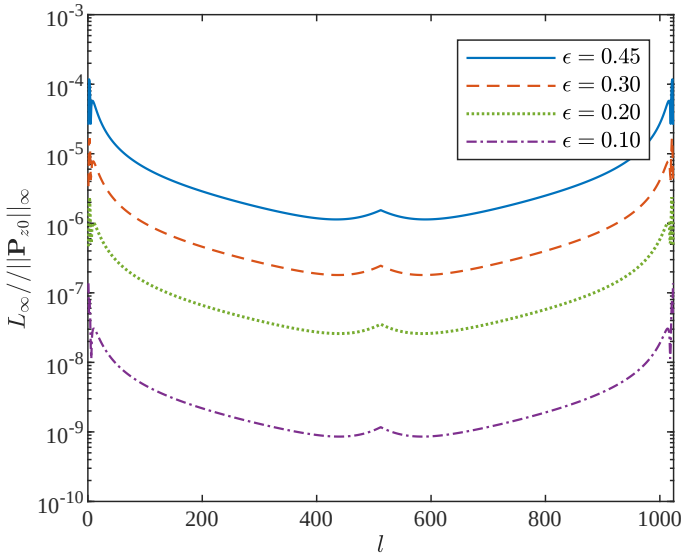


Figure 4: L_∞ error between $P_z(l, h-z)$ and the fitted function over $z \in [-\epsilon l,\, \epsilon l]$ for different values of l (with $L = 1024$, $h = 1.5$, and $\epsilon = 0.45$).

is constructed to approximate $P_z(l, h-z)$ in the least-squares sense over $z \in [-\epsilon l,\, \epsilon l]$.

Figure 4 illustrates the L_∞ fitting error for the vertical derivative P_z of the two-dimensional PFS across various horizontal separations. The proposed approach yields highly accurate approximations for $\boldsymbol{G}_z$. Notably, compared to the fundamental solution P itself, its vertical derivative P_z decays more rapidly with distance. This means that the induced vertical velocity is much stronger in the near field and diminishes quickly in the far field. As a consequence, the absolute error is larger for collocation points close to the source and much smaller for distant points. Throughout this work, both PFS and its derivatives are consistently fitted using four basis functions as described in Eq. (18). For applications demanding even greater precision, near-field errors can be further reduced by introducing sparse matrix corrections to $\boldsymbol{G}_z$.

3.4.2. Velocity field below the free surface

Beyond tracking the evolution of the free surface, many wave-related applications require accurate reconstruction of the velocity field from the surface velocity potential. This is particularly important for wave dynamics, wave load estimation, and coupling between viscous and potential flow models. The PFS framework introduced in this study enables both efficient and accurate subsurface flow reconstruction.

The PFS can be interpreted as a superposition of fundamental singularities, specifically $\ln r$ sources in two dimensions and $1/r$ sources in three dimensions. This framework allows the use of fast summation techniques, such as the FMM or adaptive cross approximation, for rapid evaluation of the velocity field at arbitrary locations.

In addition, the convolution-based algorithm proposed here provides a highly accurate and computationally efficient alternative for reconstructing the velocity field. By predefining a set of h_{offset} values for the target interpolation region, convolution operators can be computed in advance to fit $\nabla P(l, h-z+h_{\text{offset}})$. At each time step, Eq. (28) is employed to evaluate the velocity field at $z = \zeta(\boldsymbol{x}) - h_{\text{offset}}$, after which interpolation yields the velocity at any desired position.

Compared to the DNO approach, an important advantage of the PFS method is that the velocity field is constructed from locally decaying fundamental solutions, which mitigates the exponential amplification of short-wave errors that can affect spectral DNO methods. Section 4.5 presents specific examples and a detailed discussion of this benefit.

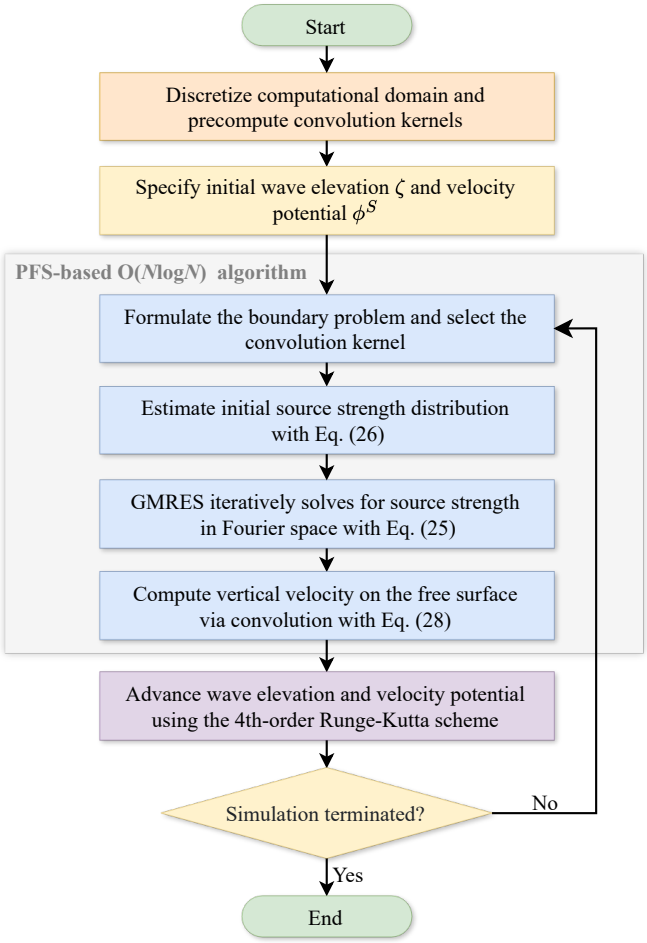


Figure 5: Schematic workflow of the periodic fundamental solution (PFS) method.

Figure 5 provides an overview of the workflow for the PFS method. Prior to the simulation, convolution kernels are precomputed for a range of wave steepness values ϵ, tailored to the discretized computational domain. At each time step, the maximum wave steepness is calculated from the current wave profile ζ, and the appropriate convolution kernels are selected. These kernels are then used within the GMRES algorithm (see Eqs. (25) and (26)) to solve for the source-strength distribution that enforces the Dirichlet boundary condition on the free surface.

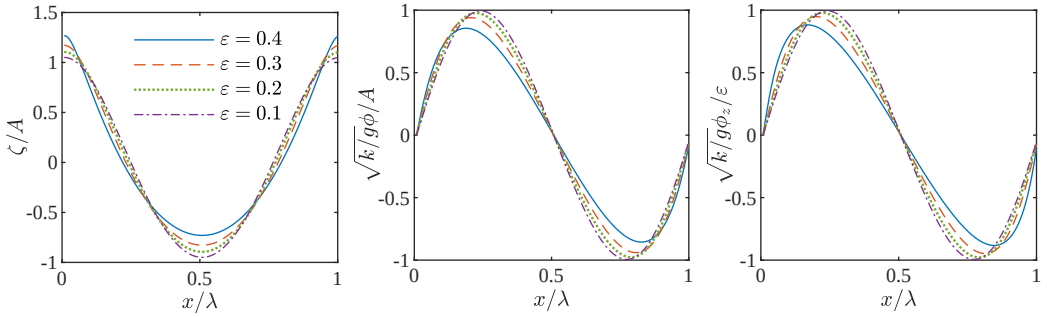
Once the source strengths are determined, Eq. (28) enables efficient calculation of the vertical velocity at the free surface. This velocity is subsequently substituted into the free-surface boundary conditions (Eqs. (3) and (4)), allowing the wave evolution to be advanced in time. For this purpose, a time-stepping scheme such as the fourth-order Runge–Kutta method is employed.

4. Numerical results and discussion

4.1. Grid convergence study

As detailed in Section 3, the PFS method is a specialized variant of the MFS. By introducing convolution operations and a preconditioned GMRES solver, PFS method reduces the computational cost from the $O(N^3)$ complexity of traditional MFS to $O(N \log N)$. The accuracy of PFS is therefore shaped by both inherited properties of MFS, such as grid resolution and the desingularization distance h, and by PFS-specific factors, including the number of GMRES iterations and the accuracy of convolution fitting. Notably, both grid resolution and h also have a direct impact on the convergence behavior of the PFS algorithm.

To systematically assess the accuracy and convergence of mapping ϕ^S to $\phi_z\big|_{z=\zeta}$, we adopt fully nonlinear steady waves of varying steepness as benchmark cases, computed from stream-function theory using the high-accuracy procedure of Clamond and Dutykh [50]. The solution is first evaluated on a fine non-uniform grid and then interpolated onto the uniform grid required by the present scheme; this initialization strategy is consistent with that of Klahn et al. [41]. Figure 6 displays the free-surface profile η, velocity potential $\phi|_{z=\eta}$, and vertical velocity $\phi_z|_{z=\eta}$. Here, λ denotes the wavelength, A is the amplitude, the wavenumber is defined as $k = 2\pi/\lambda$, and the wave steepness $\varepsilon = kA$ provides a measure of nonlinearity. Figures 6(a) and (b) show the wave elevation and velocity potential, respectively, defining a classical boundary value problem; the corresponding vertical velocity is presented in Fig. 6(c). For small ε, the free surface closely resembles a sinusoidal curve, and the flow field remains symmetric. As steepness increases, the wave profile develops sharp crests and flat troughs, while nonlinear effects in the velocity field become stronger and more difficult to model accurately.



(a) Wave elevation (b) Surface potential (c) Surface vertical velocity

Figure 6: Stream-function solutions of deep-water nonlinear waves with different wave steepnesses, computed following Clamond and Dutykh [50].

To assess the effect of grid density on convergence and accuracy, we consider a computational domain covering a single wavelength. The domain is discretized into N evenly spaced points, where the grid spacing is nondimensionalized to one. The desingularization distance is specified as $h = 1.5$, i.e., 1.5 times the grid spacing. Table 1 summarizes the L_2 relative errors for different numerical methods with respect to the reference solution of Clamond and Dutykh [50], with N denoting the number of grid points per wavelength. For reference, two additional methods are included:

1) The *MFS* column shows the L_2 error for the traditional MFS approach, where the system $\boldsymbol{G\sigma} = \boldsymbol{\phi}^S$ is solved directly and $\phi_z\big|_{z=\zeta} = \boldsymbol{G}_z\boldsymbol{\sigma}$ is computed, with a computational cost of $O(N^3)$.

2) The *DNO* column reports the L_2 error for the DNO method, implemented at expansion order 3 in the open-source HOS-Ocean code [35], which, like PFS, operates with $O(N \log N)$ complexity.

Table 1: The L_2 error of numerical results relative to solutions of Clamond and Dutykh [50] under different grid resolutions ($h = 1.5$).

ε	N	Number of GMRES iterations					MFS	DNO
		1	2	3	4	5		
0.1	32	1.17×10^{-3}	1.95×10^{-4}	1.83×10^{-4}	1.82×10^{-4}	1.82×10^{-4}	1.82×10^{-4}	3.63×10^{-4}
	64	3.75×10^{-4}	1.51×10^{-4}	1.51×10^{-4}	1.51×10^{-4}	1.51×10^{-4}	1.51×10^{-4}	3.63×10^{-4}
	128	1.74×10^{-4}	1.40×10^{-4}	1.40×10^{-4}	1.40×10^{-4}	1.40×10^{-4}	1.40×10^{-4}	3.63×10^{-4}
0.2	32	4.96×10^{-3}	6.08×10^{-4}	1.44×10^{-4}	1.06×10^{-4}	1.05×10^{-4}	1.01×10^{-4}	3.08×10^{-3}
	64	1.23×10^{-3}	8.25×10^{-5}	7.36×10^{-5}	7.38×10^{-5}	7.38×10^{-5}	7.04×10^{-5}	3.08×10^{-3}
	128	2.65×10^{-4}	6.03×10^{-5}	6.37×10^{-5}	6.37×10^{-5}	6.37×10^{-5}	6.05×10^{-5}	3.08×10^{-3}
0.3	32	1.70×10^{-2}	4.47×10^{-3}	1.09×10^{-3}	3.14×10^{-4}	2.70×10^{-4}	2.80×10^{-4}	1.16×10^{-2}
	64	3.17×10^{-3}	3.50×10^{-4}	2.41×10^{-4}	2.62×10^{-4}	2.65×10^{-4}	2.34×10^{-4}	1.16×10^{-2}
	128	6.35×10^{-4}	2.46×10^{-4}	2.48×10^{-4}	2.48×10^{-4}	2.48×10^{-4}	2.16×10^{-4}	1.16×10^{-2}
0.4	32	4.18×10^{-2}	1.99×10^{-2}	1.14×10^{-2}	5.62×10^{-3}	2.10×10^{-3}	8.31×10^{-4}	3.43×10^{-2}
	64	1.87×10^{-2}	5.32×10^{-3}	1.20×10^{-3}	9.15×10^{-4}	6.07×10^{-4}	6.95×10^{-4}	3.39×10^{-2}
	128	2.88×10^{-3}	1.05×10^{-3}	8.82×10^{-4}	8.77×10^{-4}	8.76×10^{-4}	6.37×10^{-4}	3.39×10^{-2}

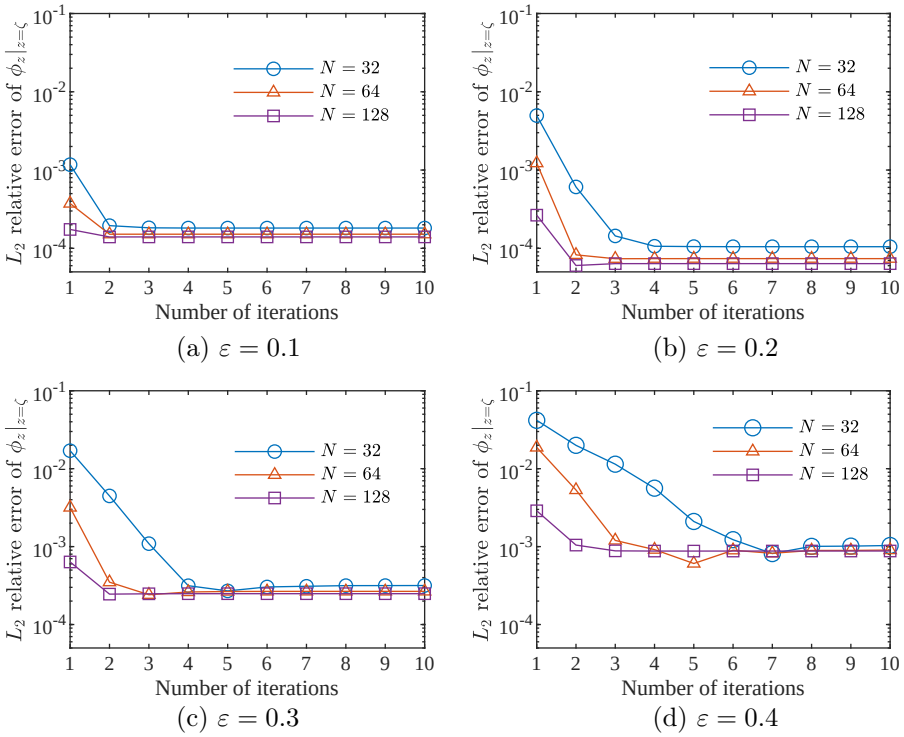


Figure 7: L_2 relative error of PFS as a function of GMRES iteration count for $\varepsilon \in \{0.1, 0.2, 0.3, 0.4\}$.

The results in Table 1 show that grid resolution strongly affects the convergence of the PFS method. To make these trends more directly readable, Fig. 7 visualizes the PFS error along the iteration axis. For waves of varying steepness, increasing N moderately improves the accuracy of $\phi_z|_{z=\zeta}$ and reduces the number of GMRES iterations needed for convergence. Solutions with small wave steepness converge more readily, whereas larger steepness requires additional iterations. This trend reflects the reduced similarity between the matrix $\hat{\boldsymbol{G}}$ and the preconditioner $\boldsymbol{P}_0$ under strongly

nonlinear conditions, which raises the condition number of the preconditioned system $\boldsymbol{P}_0^{-1}\hat{\boldsymbol{G}}$ and slows convergence.

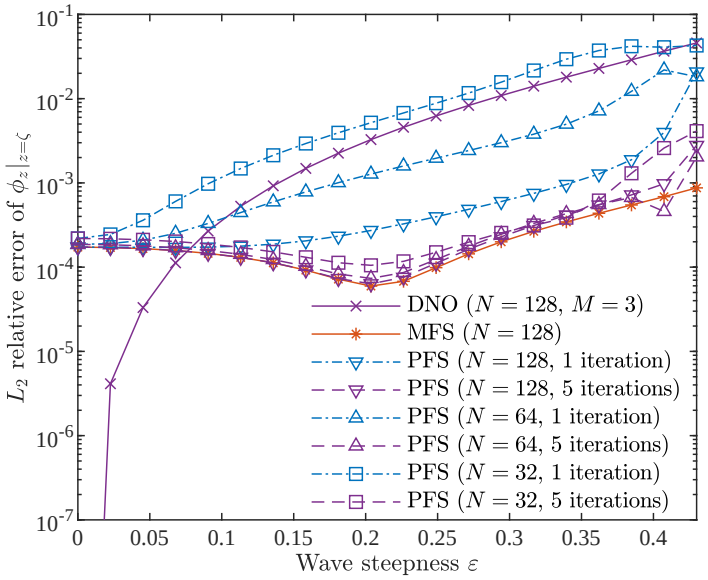


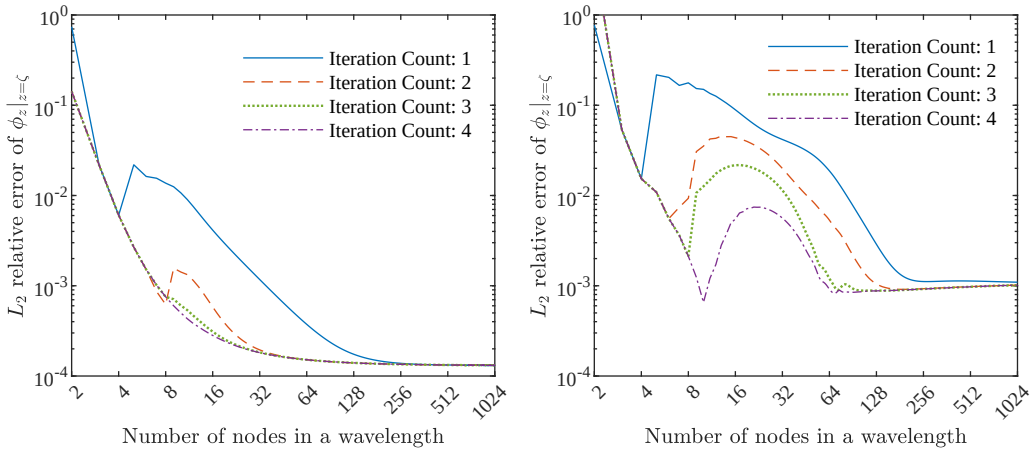
Figure 8: L_2 relative error of $\phi_z|_{z=\zeta}$ versus wave steepness ε for DNO, MFS, and PFS at different grid resolutions and GMRES iteration counts.

The MFS results in Table 1 represent the theoretical minimum error achievable by the PFS approach, and Fig. 8 compares the errors of the different methods along the steepness axis. As nonlinearity increases, the gap between the fitted matrix $\hat{\boldsymbol{G}}$ and the exact matrix $\boldsymbol{G}$ widens, preventing GMRES from attaining this lower bound. Nevertheless, even for strongly nonlinear cases ($\varepsilon = 0.4$), the fast algorithm remains highly accurate, closely matching the exact solution. This demonstrates that the strategy of fitting the PFS with four basis functions provides sufficient accuracy.

The comparison between the DNO and PFS results in Table 1 underscores important distinctions. DNO accuracy is largely insensitive to grid resolution; increasing the grid density primarily increases the number of available spectral modes, with minimal impact on accuracy. In contrast, finer grids in PFS improve both convergence and solution accuracy to some extent. For example, at $\varepsilon = 0.4$, doubling N significantly reduces the error after the first GMRES iteration. However, the effect of N on the converged solution is much less pronounced, regardless of grid resolution, the error after five iterations remains at a comparable level. This indicates that the asymptotic accuracy of PFS is primarily governed by the MFS framework itself rather than by grid resolution, and does not improve indefinitely with increasing N. The PFS method can match, and in some cases exceed, the accuracy of DNO, with solutions after just two GMRES iterations matching the DNO result with third-order expansion. This advantage becomes even more pronounced at higher wave steepness.

Figure 9 further illustrates how L_2 error varies with grid resolution and the number of GMRES iterations in PFS method. The proposed method demonstrates robust convergence: with more than 200 grid points per wavelength, a single GMRES iteration is typically sufficient for converged results; at lower resolutions, two or three iterations generally achieve satisfactory accuracy. Beyond a certain grid density, further refine-

ment yields diminishing returns, confirming that the solution accuracy is bounded by the fundamental solution representation. In strongly nonlinear scenarios ($\varepsilon = 0.4$), numerical errors rise sharply when fewer than 32 grid points per wavelength are used. In these cases, the fifth-order harmonic amplitude reaches 2.34% of the first-order amplitude, making its contribution significant. To accurately resolve high-order harmonics, we recommend using at least 6 grid points per harmonic.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.4$

Figure 9: L_2 error under different grid resolutions and iteration counts ($h = 1.5$).

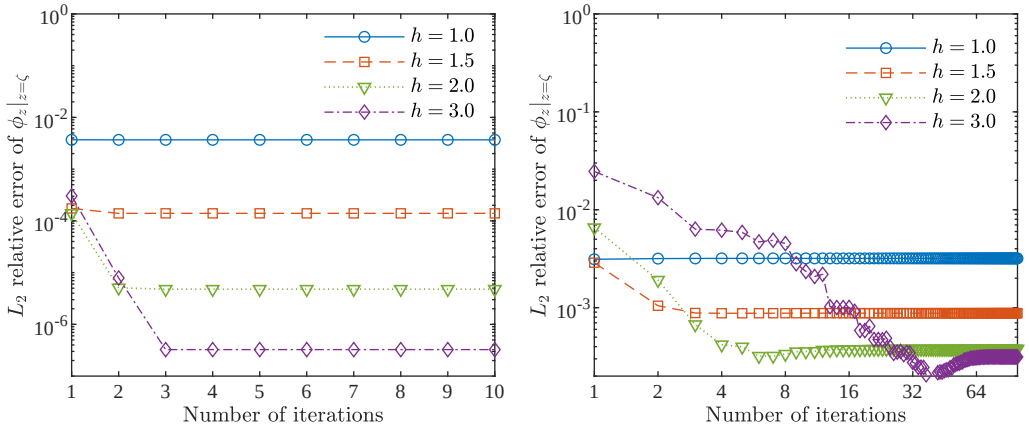
4.2. Sensitivity to the desingularization parameter h

In the PFS method, the desingularization distance h, which is the vertical offset between each point source and its collocation point, is a critical parameter affecting both accuracy and convergence. Figure 10 shows how the error and iterative convergence vary with different values of h. When h is small, GMRES converges quickly with fewer iterations, but the final solution accuracy is limited. Increasing h generally enhances accuracy, though it also requires more iterations for convergence. For larger h values, such as $h = 2.0$ and above, convergence demands progressively more iterations, while accuracy gains are minimal, making further increases in h of limited benefit.

Overall, selecting h requires balancing the trade-off between avoiding singularity-induced errors and maintaining favorable matrix conditioning. If h is too small, the proximity of the point source to the collocation point introduces unavoidable singularity effects, reducing accuracy. Conversely, setting h too large makes the coefficient matrix $\boldsymbol{G}$ more ill-conditioned, slowing convergence. For most practical applications, choosing h between 1.5 and 2 strikes an effective balance between accuracy and computational efficiency.

4.3. Efficiency comparison with the Dirichlet–Neumann operator (DNO) method

To evaluate the relative efficiency of the proposed PFS method and the conventional DNO approach, we solve for the $\phi_z\big|_{z=\zeta}$ waves using both PFS and DNO methods, varying the number of iterations (for PFS) and expansion orders (for DNO). DNO calculations are performed with the open-source HOS-Ocean code [35]. For all tests, the periodic computational domain is discretized with 128 collocation points; the PFS method uses a desingularization distance of $h = 1.5$, while the DNO method employs additional zero-padding in the frequency domain to ensure aliasing-free results.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.4$

Figure 10: Error and convergence characteristics for different desingularization distances h ($L = 128$).

Figure 11 presents the error characteristics of both approaches. In the DNO method, errors decrease exponentially (as $O(\varepsilon^n)$) with increasing expansion order. By contrast, the PFS approach achieves rapid error reduction within the first few iterations, after which the improvement plateaus. For waves with large steepness and strong nonlinearity, PFS demonstrates a clear accuracy advantage, converging faster than DNO. For example, in the case of $\varepsilon = 0.4$, the DNO method requires an 8th-order expansion (72 FFTs) to achieve the accuracy that the PFS method reaches with just two iterations (21 FFTs). However, for small wave steepness, the pseudo-spectral nature of DNO enables it to reach high precision with just a few iterations, while the accuracy of PFS is inherently limited by the fundamental solution framework, typically stabilizing around 10^{-4}.

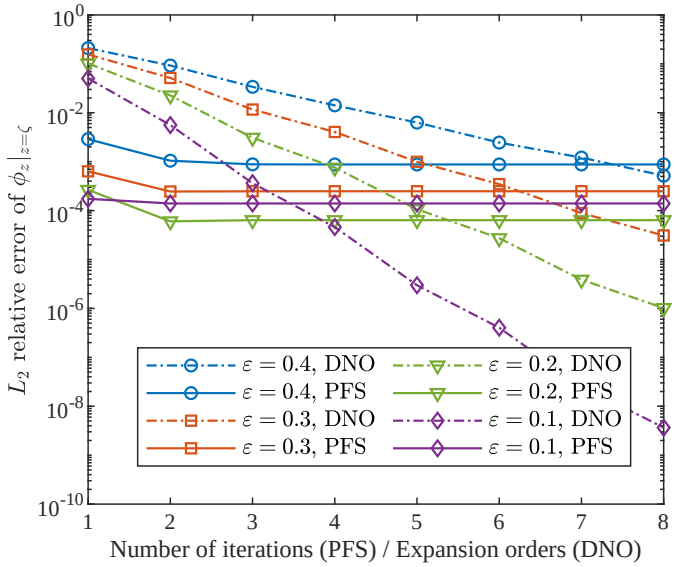


Figure 11: Comparison of L_2 errors from different methods.

In terms of computational cost, each additional PFS iteration requires 7 extra FFTs. Mapping ϕ^S to $\phi_z\big|_{z=\zeta}$ involves $7M+7$ FFT or IFFT operations, where M is the number of iterations. The DNO method, implemented via perturbation expansion [9], requires

$M(M+1)$ FFT or IFFT operations. In practice, third- and fifth-order DNO algorithms ($M = 3$ and $M = 5$) are commonly used. Neglecting additional FFT padding for de-aliasing, their computational cost is roughly equivalent to one and three PFS iterations, respectively. From this perspective, the PFS method may achieve higher accuracy with lower computational effort.

4.4. Stability and conditioning issues

It is important to note that the DNO method does not always converge as the expansion order increases. Previous studies [14, 37–39] have reported that, especially at high grid resolutions or with many iterations, DNO may suffer from severe ill-conditioning. To illustrate this, and following the example from Nicholls and Reitich [37], we consider the following forms for the free surface and velocity potential:

$$\begin{aligned} k\zeta(x) &= \varepsilon\,|4x/L - 2| - \varepsilon, \quad x \in [0, L], \\ (k^3/g)^{1/2}\phi(x,z) &= \varepsilon \exp(kz)\sin(kx). \end{aligned} \tag{30}$$

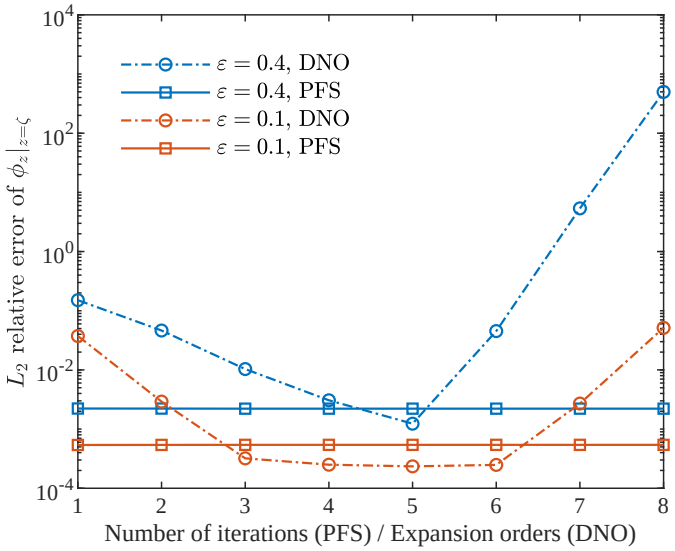


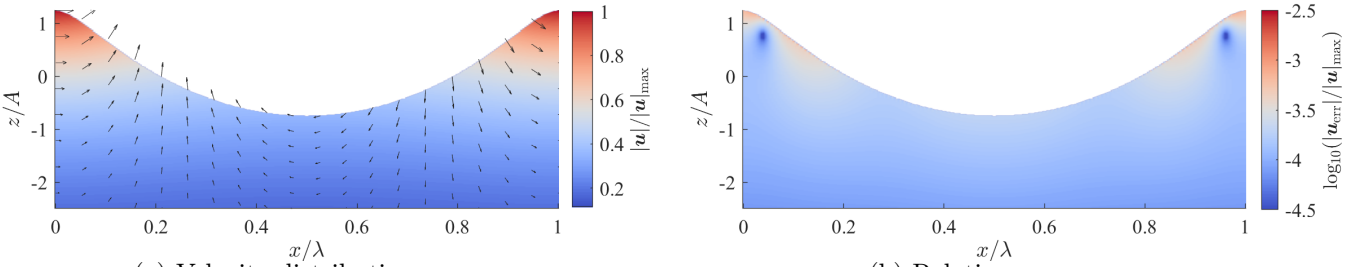
Figure 12: L_2 relative error in solving the boundary value problem of the rough profile using different methods.

We solve the corresponding boundary value problem in a computational domain of $L = 1024$, comparing the L_2 relative errors of $\phi_z\big|_{z=\zeta}$ obtained by both PFS and DNO methods, as shown in Fig. 12. The results show that when the boundary is non-smooth, increasing the perturbation expansion order in DNO does not always improve accuracy; instead, the results diverge after more than five iterations. This issue is observed under both weak ($\varepsilon = 0.1$) and strong ($\varepsilon = 0.4$) nonlinearity.

In contrast, the proposed PFS method benefits from the GMRES algorithm and does not exhibit divergence with additional iterations. However, as a specialized form of the MFS, PFS still faces accuracy challenges on non-smooth surfaces like those in this example, indicating potential areas for future improvement.

4.5. Velocity Field Reconstruction

Figure 13(a) presents the velocity field of a nonlinear steady wave, reconstructed throughout the domain using the PFS method under prescribed wave elevation ζ and surface potential ϕ^S ($h = 1.5$, $L = 128$, with two GMRES iterations). Figure 13(b) displays the relative error between the reconstructed velocity field and the reference solution. As shown, even for strongly nonlinear waves with $\varepsilon = 0.4$, the PFS method achieves highly accurate reconstruction of the velocity field.



(a) Velocity distribution (b) Relative error

Figure 13: Velocity field of a nonlinear steady wave reconstructed by the PFS method and its error relative to the reference solution ($\varepsilon = 0.4$).

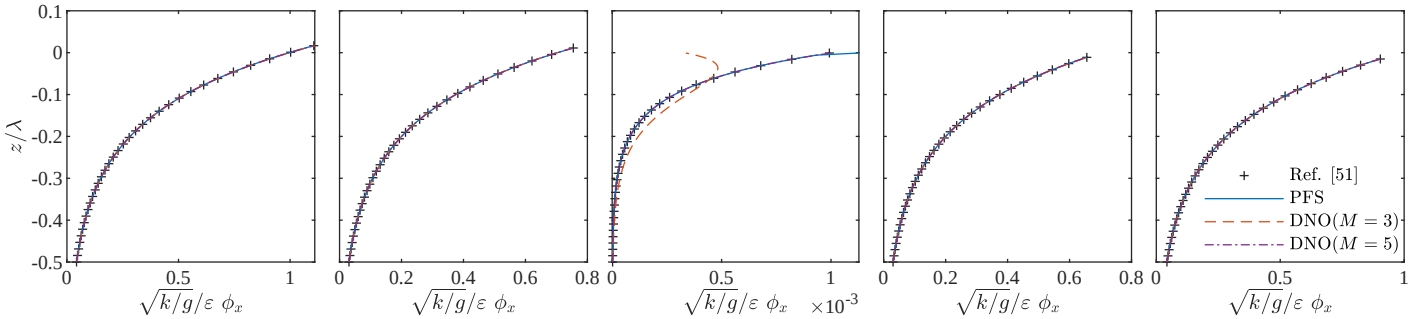
To further highlight the differences in velocity field reconstruction between the PFS and DNO methods, Figures 14–17 compare velocity fields computed by various numerical approaches with the reference solution, where $x = 0$ denotes the wave crest. The DNO results shown are obtained using the H-operator approach proposed by Bateman et al. [38], which is equivalent to the conventional DNO method [51].

For waves with small steepness, both PFS and DNO methods accurately reconstruct the horizontal velocity field. With DNO at $M = 3$, minor deviations from the reference solution are observed near $x = 1/4\lambda$, while at $M = 5$, the DNO results are nearly indistinguishable from the reference solution. However, as wave steepness ε exceeds 0.3, DNO begins to show significant discrepancies, particularly near the free surface, whereas PFS results continues to match the reference solution closely. This error stems from an inherent limitation of pseudo-spectral methods: the Fourier basis functions, which grow exponentially in the vertical direction as $\exp(kz)$, can amplify high-frequency errors during velocity reconstruction. As a result, the accuracy of the reconstructed velocity field can deteriorate rapidly, posing challenges for reliable numerical simulation.

In contrast, the PFS method leverages the superposition of locally decaying fundamental solutions, which naturally suppresses the amplification of high-frequency errors. Once the source strength distribution $\boldsymbol{\sigma}$ is obtained, the velocity field at any horizontal plane can be efficiently computed with just three convolution operations. As shown in Figs. 14–17, PFS maintains high reconstruction accuracy even for strongly nonlinear waves. This capability is especially valuable in engineering practice, for example, when calculating wave loads or providing potential flow velocity fields for viscous–potential flow coupling simulations.

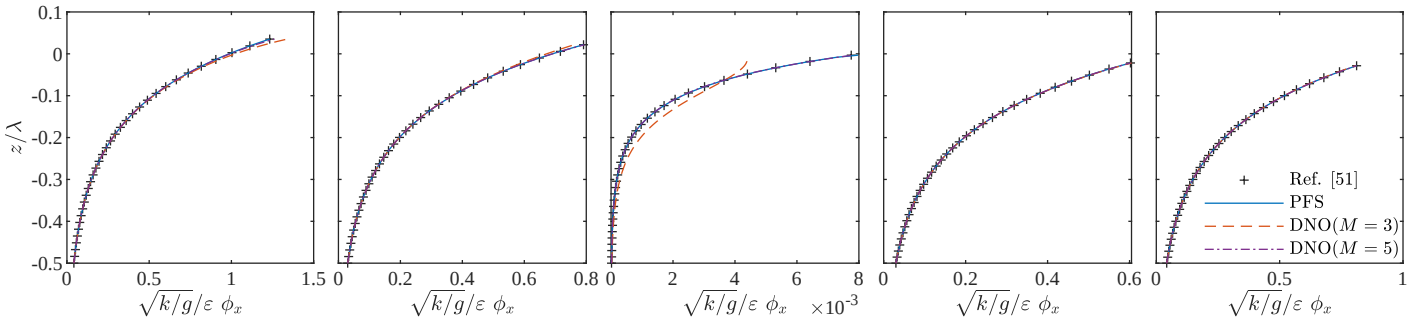
4.6. Time-domain simulation

To assess the long-term accuracy and stability of our proposed method, we perform forward and backward simulations of deep-water steady waves. In these experiments, the waves are first propagated forward for 100 periods; subsequently, the time variable is reversed, and the simulation is integrated backward to return the system to its initial



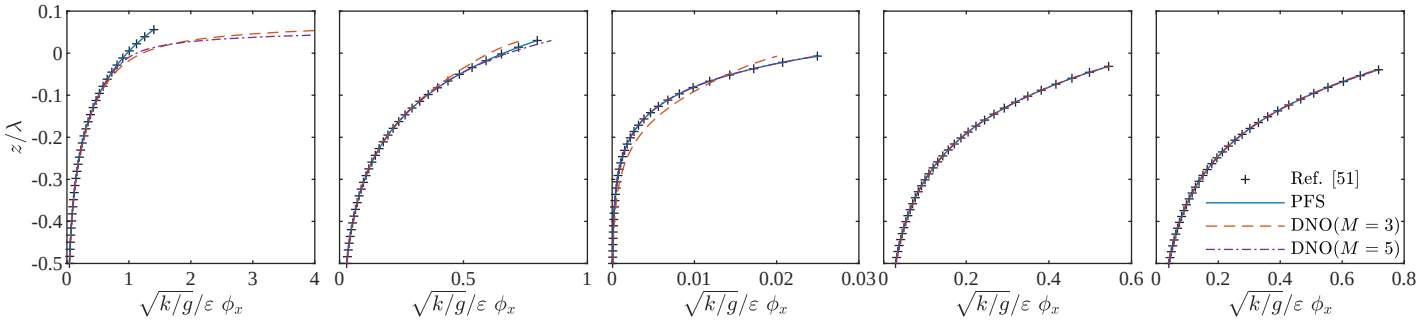
(a) $x = 0$ (b) $x = 1/8\lambda$ (c) $x = 1/4\lambda$ (d) $x = 3/8\lambda$ (e) $x = 1/2\lambda$

Figure 14: Horizontal velocity of nonlinear steady waves at different positions: comparison between numerical results and the reference solution of Clamond and Dutykh [50] ($\varepsilon = 0.1$).



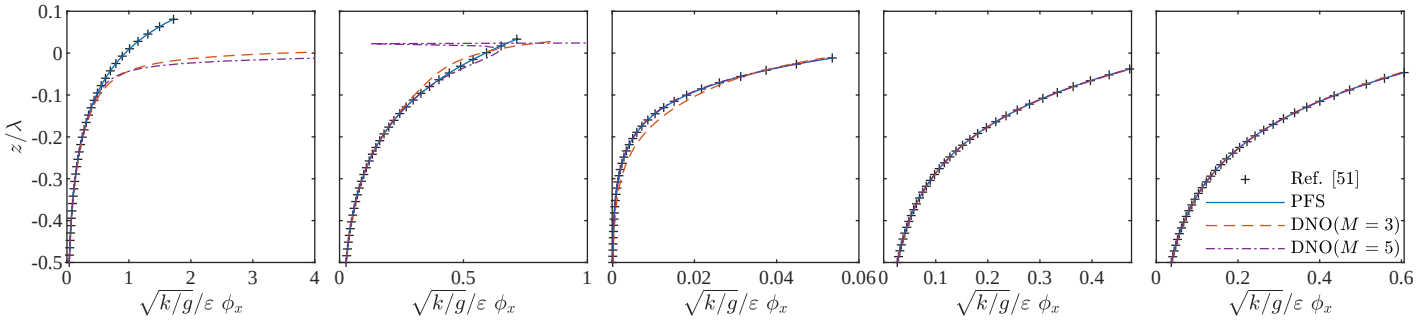
(a) $x = 0$ (b) $x = 1/8\lambda$ (c) $x = 1/4\lambda$ (d) $x = 3/8\lambda$ (e) $x = 1/2\lambda$

Figure 15: Horizontal velocity of nonlinear steady waves at different positions: comparison between numerical results and the reference solution of Clamond and Dutykh [50] ($\varepsilon = 0.2$).



(a) $x = 0$ (b) $x = 1/8\lambda$ (c) $x = 1/4\lambda$ (d) $x = 3/8\lambda$ (e) $x = 1/2\lambda$

Figure 16: Horizontal velocity of nonlinear steady waves at different positions: comparison between numerical results and the reference solution of Clamond and Dutykh [50] ($\varepsilon = 0.3$).



(a) $x = 0$ (b) $x = 1/8\lambda$ (c) $x = 1/4\lambda$ (d) $x = 3/8\lambda$ (e) $x = 1/2\lambda$

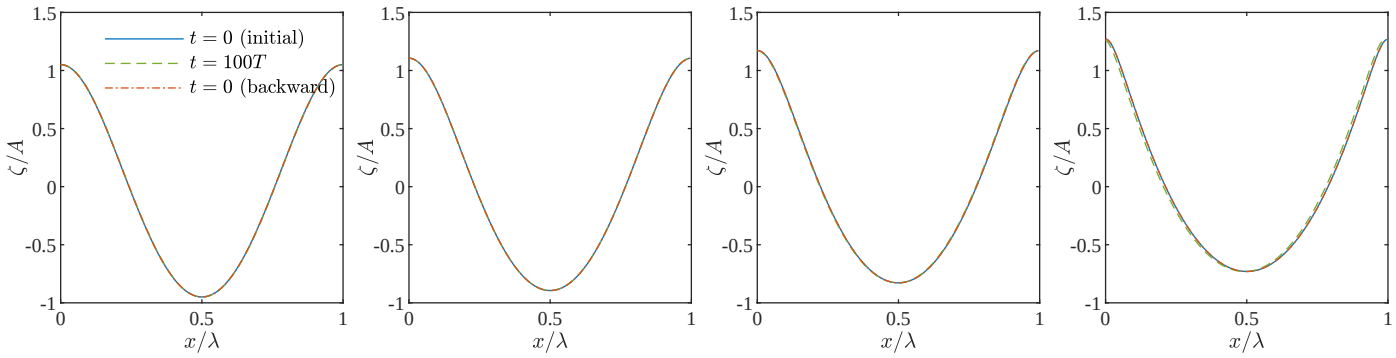
Figure 17: Horizontal velocity of nonlinear steady waves at different positions: comparison between numerical results and the reference solution of Clamond and Dutykh [50] ($\varepsilon = 0.4$).

state at $t = 0$. For precise resolution, each wavelength is discretized with 256 collocation points, and three GMRES iterations are performed per time step.

To suppress potential *saw-tooth* instabilities, a spectral filter is applied at each time step to both the wave elevation ζ and the free-surface velocity potential ϕ^S in the wavenumber domain:

$$\Lambda(k) = \exp\left[-\left(\frac{k}{2}\right)^{12}\right]. \tag{31}$$

Figure 18 displays the wave profiles throughout the forward–backward simulation for various wave steepness values. The results highlight the high fidelity of the proposed approach: after 100 periods of forward propagation, the wave elevations remain stable and closely follow the theoretical solution, exhibiting only minor phase deviations. Backward integration over the same duration further confirms the method's accuracy, as the simulated wave profiles recover the initial conditions almost perfectly. For strongly nonlinear waves ($\varepsilon = 0.4$), traditional fast algorithms such as DNO fail to maintain stable simulations[9], while the proposed PFS method continues to capture the wave evolution accurately. Compared to the reference solution, the relative errors in wave period are 8.7×10^{-6}, 2.8×10^{-6}, 1.3×10^{-5}, and 7.5×10^{-5} for $\varepsilon = 0.1, 0.2, 0.3, 0.4$, respectively.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.2$ (c) $\varepsilon = 0.3$ (d) $\varepsilon = 0.4$

Figure 18: Wave elevation of regular waves at different times in the forward–backward simulation.

Figure 19 further illustrates the evolution of wave energy over time for different wave steepness values. Throughout simulations spanning up to 200 periods, our method maintains excellent energy conservation. Only for the case of $\varepsilon = 0.4$ is a slight long-term decrease in energy observed, which can be attributed to the filtering step removing high-frequency energy from the system at each time step.

5. Conclusion

We have introduced an efficient method for simulating fully nonlinear surface gravity waves, termed the PFS method. Building on the classical MFS, PFS method constructs the velocity potential field by placing periodic fundamental solutions above the free surface and determining their source strengths under Dirichlet boundary conditions. Unlike traditional MFS approaches, PFS efficiently approximates the originally dense coefficient matrix using just three convolution operations. For numerical implementation, PFS employs the coefficient matrix from the linearized problem as a preconditioner and utilizes the GMRES iterative solver, reducing computational complexity from $O(N^3)$

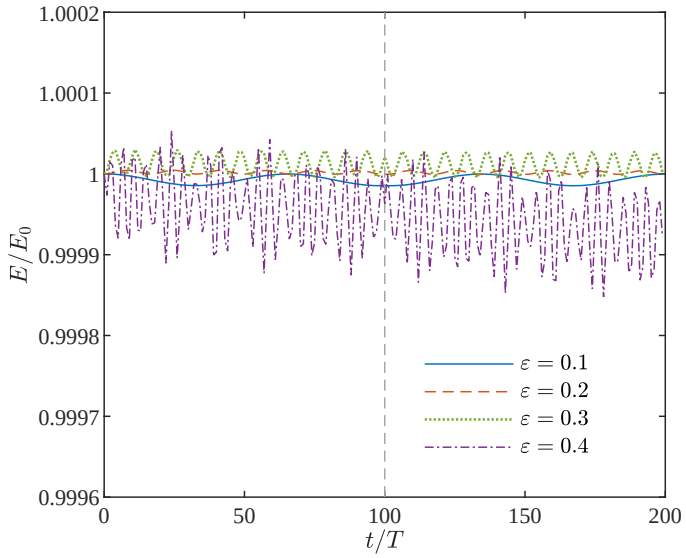


Figure 19: The variation of wave energy with numerical simulation time. Simulations run forward for 100 wave periods and then reverse along the time axis to simulate an additional 100 periods.

(in conventional MFS) to $O(N \log N)$. This improvement significantly enhances scalability and practicality for large-scale, high-resolution, and long-duration simulations of surface gravity waves, especially in open ocean environments.

Numerical tests demonstrate that the PFS method delivers high accuracy and robust numerical stability, typically requiring only 2–3 iterations for convergence in practical scenarios. Compared with the DNO method, PFS method offers three distinct advantages: (i) superior computational accuracy and stability for strongly nonlinear, steep waves, enabling reliable simulations of waves that are difficult for DNO; (ii) natural avoidance of the ill-conditioning issues encountered in high-order DNO expansions, owing to the robustness of GMRES iteration; (iii) efficient velocity reconstruction throughout the flow field, with effective suppression of high-frequency errors that often affect traditional spectral methods.

Despite these benefits, PFS also inherits certain limitations of MFS-type approaches. Its accuracy and convergence efficiency are dependent on sufficiently high spatial resolution at the free surface, and the theoretical lower bound on error is generally higher than that of spectral methods.

Given space constraints, this work has focused on the modeling and performance of PFS for two-dimensional deep-water wave problems. The formulations presented in this paper (including those in the appendix) employ a single set of four basis functions to represent the induced velocity potential throughout the field; this approximation degrades markedly in shallow water under strong nonlinearity, placing such regimes outside the method's current scope. Future work will extend PFS to three-dimensional directional wave simulations, with particular attention to realistic ocean conditions such as shallow-water regions, complex seabed topography, and non-periodic boundaries.

Acknowledgements

The authors gratefully acknowledge the support of the Hainan Natural Science Foundation Youth Fund (Grant No. 525QN399) and the Natural Science Foundation of

Shanghai (Grant No. 25ZR1402240).

Appendix A. Numerical computation of the PFS via Lekner summation

Consider a periodically distributed point source defined as:

$$P(x,y,z)=\sum_{n=-\infty}^{+\infty}\sum_{m=-\infty}^{+\infty}\frac{1}{\sqrt{(x+nL_x)^2+(y+mL_y)^2+z^2}}+C. \tag{A.1}$$

Disregarding the constant term, the Lekner summation representation [48] is:

$$\begin{aligned}P(x,y,z)=&\frac{4}{L_x}\sum_{n=1}^{+\infty}\cos\left(nk_{0x}x\right)\sum_{m=-\infty}^{+\infty}K_0\left[nk_{0x}\sqrt{(y+mL_y)^2+z^2}\right]\\&-\frac{1}{L_x}\ln\left[\cosh\left(k_{0y}z\right)-\cos\left(k_{0y}y\right)\right],\end{aligned} \tag{A.2}$$

or alternatively,

$$\begin{aligned}P(x,y,z)=&\frac{4}{L_y}\sum_{n=1}^{+\infty}\cos\left(nk_{0y}y\right)\sum_{m=-\infty}^{+\infty}K_0\left[nk_{0y}\sqrt{(x+mL_x)^2+z^2}\right]\\&-\frac{1}{L_y}\ln\left[\cosh\left(k_{0x}z\right)-\cos\left(k_{0x}x\right)\right],\end{aligned} \tag{A.3}$$

where $(k_{0x},k_{0y})=(2\pi/L_x,2\pi/L_y)$ and K_0 is the zeroth-order modified Bessel function.

For $L_x<L_y$, Eq. (A.2) shows that the argument of K_0 is always greater than $2\pi nm$. Since $K_0(6\pi)\approx 2\times10^{-9}$, truncating the series at $|nm|\le 3$ provides sufficient accuracy. In practical applications, the primary numerical difficulty arises when $m=0$ and both y and z are small, leading to slow convergence of the series. Notably, for $m=0$, the series can be reformulated as an integral:

$$\sum_{n=1}^{+\infty}\cos\left(nk_{0x}x\right)K_0\left(nk_{0x}\sqrt{y^2+z^2}\right)=\int_0^{+\infty}\Re\left(\frac{1}{e^{k_{0x}\sqrt{y^2+z^2}\cosh t-\mathrm{i}k_{0x}x}-1}\right)dt, \tag{A.4}$$

where $\Re$ denotes the real part. The rapid decay of the integrand for large t makes this integral form especially suitable for numerical evaluation.

For three-dimensional PFS, we adopt the following practical formulation:

$$\begin{aligned}P(x,y,z)=&\frac{4}{L_x}\sum_{n=1}^{3}\cos\left(nk_{0x}x\right)\sum_{m=1}^{\lfloor 3/n\rfloor}K_0\left[nk_{0x}\sqrt{(y+mL_y)^2+z^2}\right]\\&+\frac{4}{L_x}\sum_{n=1}^{3}\cos\left(nk_{0x}x\right)\sum_{m=1}^{\lfloor 3/n\rfloor}K_0\left[nk_{0x}\sqrt{(y-mL_y)^2+z^2}\right]\\&+\frac{4}{L_x}\int_0^{+\infty}\Re\left(\frac{1}{e^{k_{0x}\sqrt{y^2+z^2}\cosh t-\mathrm{i}k_{0x}x}-1}\right)dt\\&-\frac{1}{L_x}\ln\left[\cosh\left(k_{0y}z\right)-\cos\left(k_{0y}y\right)\right].\end{aligned} \tag{A.5}$$

Taking the derivative of each term in Eq. (A.5) with respect to z yields the z-derivative of the three-dimensional PFS.

Appendix B. Three-dimensional and finite-depth cases

For a horizontally periodic computational domain of size $L_x \times L_y$ and water depth d, the domain is discretized into an $N \times M$ uniform grid. The variables $\boldsymbol{\zeta}$, $\boldsymbol{\phi}^S$, and $\boldsymbol{\sigma}$ are represented as $N \times M$ matrices. The velocity potential induced by the source strength $\boldsymbol{\sigma}$, denoted as $\boldsymbol{\phi}^S_{\text{induced}}$, can be expressed as:

$$\begin{aligned}\boldsymbol{\phi}^S_{\text{induced}} \approx\ & \Re\left[\exp(\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_1 \circledast (\exp(-\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] \\ & + \Re\left[\exp(2\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_2 \circledast (\exp(-2\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] + \boldsymbol{a}_0 \circledast \boldsymbol{\sigma} \\ & + \Re\left[\exp(\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_4 \circledast (\exp(\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] \\ & + \Re\left[\exp(2\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_5 \circledast (\exp(2\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] + \boldsymbol{a}_3 \circledast \boldsymbol{\sigma},\end{aligned} \tag{B.1}$$

where $k_0 = \pi/\sqrt{L_x^2 + L_y^2}$ is the fundamental wavenumber for interpolation. The function $\exp(\cdot)$ is applied element-wise, $\circ$ denotes the Hadamard product (element-wise multiplication), and $\circledast$ represents two-dimensional convolution. The matrices $\boldsymbol{a}_{0,1,2}$ and $\boldsymbol{a}_{3,4,5}$ are discrete convolution kernels, sampled from continuous kernel functions $a_{0,1,2}(\boldsymbol{x})$ and $a_{3,4,5}(\boldsymbol{x})$ on the computational grid. These kernels are least-squares fits to $P(\boldsymbol{x}, h - z)$ and $P(\boldsymbol{x}, h + 2d + z)$:

$$\begin{aligned}P(\boldsymbol{x}, h - z) &\approx a_0(\boldsymbol{x}) + \Re\left[a_1(\boldsymbol{x})e^{\mathrm{i}k_0 z} + a_2(\boldsymbol{x})e^{\mathrm{i}2k_0 z}\right], \\ P(\boldsymbol{x}, h + 2d + z) &\approx a_3(\boldsymbol{x}) + \Re\left[a_4(\boldsymbol{x})e^{\mathrm{i}k_0 z} + a_5(\boldsymbol{x})e^{\mathrm{i}2k_0 z}\right].\end{aligned} \tag{B.2}$$

By flattening the two-dimensional matrix $\boldsymbol{\sigma}$, the source strengths can be efficiently solved using GMRES iterations with Eqs. (25) and (26). Once $\boldsymbol{\sigma}$ is obtained, the vertical velocity at the free surface is computed as:

$$\begin{aligned}\boldsymbol{\phi}_z\big|_{z=\boldsymbol{\zeta}} \approx\ & \Re\left[\exp(\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_1^{(z)} \circledast (\exp(-\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] \\ & + \Re\left[\exp(2\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_2^{(z)} \circledast (\exp(-2\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] + \boldsymbol{a}_0^{(z)} \circledast \boldsymbol{\sigma} \\ & + \Re\left[\exp(\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_4^{(z)} \circledast (\exp(\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] \\ & + \Re\left[\exp(2\mathrm{i}k_0\boldsymbol{\zeta}) \circ (\boldsymbol{a}_5^{(z)} \circledast (\exp(2\mathrm{i}k_0\boldsymbol{\zeta}) \circ \boldsymbol{\sigma}))\right] + \boldsymbol{a}_3^{(z)} \circledast \boldsymbol{\sigma},\end{aligned} \tag{B.3}$$

where $\boldsymbol{a}^{(z)}_{0,1,2}$ and $\boldsymbol{a}^{(z)}_{3,4,5}$ are discrete convolution kernels obtained by sampling the continuous kernel functions $a^{(z)}_{0,1,2}(\boldsymbol{x})$ and $a^{(z)}_{3,4,5}(\boldsymbol{x})$ on the computational grid and fitted in the least-squares sense:

$$\begin{aligned}P_z(\boldsymbol{x}, h - z) &\approx a_0^{(z)}(\boldsymbol{x}) + \Re\left[a_1^{(z)}(\boldsymbol{x})e^{\mathrm{i}k_0 z} + a_2^{(z)}(\boldsymbol{x})e^{\mathrm{i}2k_0 z}\right], \\ P_z(\boldsymbol{x}, h + 2d + z) &\approx a_3^{(z)}(\boldsymbol{x}) + \Re\left[a_4^{(z)}(\boldsymbol{x})e^{\mathrm{i}k_0 z} + a_5^{(z)}(\boldsymbol{x})e^{\mathrm{i}2k_0 z}\right].\end{aligned} \tag{B.4}$$

Appendix C. Fast algorithm based on Taylor expansion and perturbation

In the DNO method [11, 12] (also known as the HOS method [9, 10]), the wave elevation ζ is considered a small parameter. Spectral basis functions are expanded as power series at $z = 0$, and successive orders of the velocity potential are obtained via regular perturbation analysis, ultimately yielding the vertical velocity at the free surface.

Building upon this framework, we solve the equation $\boldsymbol{G}\boldsymbol{\sigma} = \boldsymbol{\phi}^S$ using a combination of Taylor expansion and regular perturbation.

Assume that both the operator $\boldsymbol{G}$ and the source strength vector $\boldsymbol{\sigma}$ can be expanded in powers of ζ:

$$\boldsymbol{G} = \sum_{i=0}^{\infty} \boldsymbol{G}_i, \qquad \boldsymbol{\sigma} = \sum_{i=0}^{\infty} \boldsymbol{\sigma}_i. \tag{C.1}$$

The source strength at each order is determined recursively by

$$\begin{aligned} \boldsymbol{G}_0\boldsymbol{\sigma}_0 &= \boldsymbol{\phi}^S, \\ \boldsymbol{G}_0\boldsymbol{\sigma}_m &= -\sum_{i=1}^{m} \boldsymbol{G}_i\boldsymbol{\sigma}_{m-i}, \quad m > 0. \end{aligned} \tag{C.2}$$

Here, each $\boldsymbol{G}_m$ is derived from the multivariate Taylor expansion of $\boldsymbol{G}$ at $z_i = 0, z_j = 0$:

$$\begin{aligned} \boldsymbol{G}_0 &= \boldsymbol{G}^{(0,0)}, \\ \boldsymbol{G}_m &= \sum_{k=0}^{m} \frac{1}{k!(m-k)!}\, \boldsymbol{\zeta}^{\,m-k}\, \boldsymbol{G}^{(m-k,k)}\, \boldsymbol{\zeta}^{\,k}, \end{aligned} \tag{C.3}$$

where the elements of $\boldsymbol{G}^{(l,n)}$ are defined as

$$G_{ij}^{(l,n)} = \frac{\partial^{l+n}}{\partial z_i^{\,l} \partial z_j^{\,n}} G(\boldsymbol{x}_i, \boldsymbol{x}_j, z_i, z_j)\bigg|_{z_i=0, z_j=0}. \tag{C.4}$$

The source strengths up to third order are then given by:

$$\begin{aligned} &\boldsymbol{G}^{(0,0)}\boldsymbol{\sigma}_0 = \boldsymbol{\phi}^S, \\ &\boldsymbol{G}^{(0,0)}\boldsymbol{\sigma}_1 = -\boldsymbol{G}^{(0,1)}\boldsymbol{\zeta}\boldsymbol{\sigma}_0 - \boldsymbol{\zeta}\boldsymbol{G}^{(1,0)}\boldsymbol{\sigma}_0, \\ &\boldsymbol{G}^{(0,0)}\boldsymbol{\sigma}_2 = -\boldsymbol{G}^{(0,1)}\boldsymbol{\zeta}\boldsymbol{\sigma}_1 - \boldsymbol{\zeta}\boldsymbol{G}^{(1,0)}\boldsymbol{\sigma}_1 - \tfrac{1}{2}\boldsymbol{G}^{(0,2)}\boldsymbol{\zeta}^2\boldsymbol{\sigma}_0 - \boldsymbol{\zeta}\boldsymbol{G}^{(1,1)}\boldsymbol{\zeta}\boldsymbol{\sigma}_0 - \tfrac{1}{2}\boldsymbol{\zeta}^2\boldsymbol{G}^{(2,0)}\boldsymbol{\sigma}_0. \end{aligned} \tag{C.5}$$

Similarly, by expanding the matrix $\boldsymbol{G}_z$ at the free surface and collecting terms of the same order, the expressions for each order of $\phi_z\big|_{z=\zeta}$ are obtained as:

$$\phi_z\big|_{z=\zeta} = \sum_{i=0}^{\infty} \phi_{zi}, \tag{C.6}$$

$$\phi_{zm} = \sum_{l=0}^{m} \sum_{k=0}^{l} \frac{1}{k!(l-k)!}\, \boldsymbol{\zeta}^{\,l-k}\, \boldsymbol{G}^{(l-k+1,k)}\, \boldsymbol{\zeta}^{\,k} \boldsymbol{\sigma}_{m-l} \tag{C.7}$$

The first three orders of the vertical velocity are therefore:

$$\begin{aligned} &\boldsymbol{\phi}_{z0} = \boldsymbol{G}^{(1,0)}\boldsymbol{\sigma}_0, \\ &\boldsymbol{\phi}_{z1} = \boldsymbol{G}^{(1,0)}\boldsymbol{\sigma}_1 + \boldsymbol{\zeta}\boldsymbol{G}^{(2,0)}\boldsymbol{\sigma}_0 + \boldsymbol{G}^{(1,1)}\boldsymbol{\zeta}\boldsymbol{\sigma}_0, \\ &\boldsymbol{\phi}_{z2} = \boldsymbol{G}^{(1,0)}\boldsymbol{\sigma}_2 + \boldsymbol{G}^{(1,1)}\boldsymbol{\zeta}\boldsymbol{\sigma}_1 + \boldsymbol{\zeta}\boldsymbol{G}^{(2,0)}\boldsymbol{\sigma}_1 + \tfrac{1}{2}\boldsymbol{\zeta}^2\boldsymbol{G}^{(3,0)}\boldsymbol{\sigma}_0 + \boldsymbol{\zeta}\boldsymbol{G}^{(2,1)}\boldsymbol{\zeta}\boldsymbol{\sigma}_0 + \tfrac{1}{2}\boldsymbol{G}^{(1,2)}\boldsymbol{\zeta}^2\boldsymbol{\sigma}_0. \end{aligned} \tag{C.8}$$

Notably, this derivation ultimately yields expressions identical to those of the DNO. Expanding the three-dimensional PFS in a Fourier series and omitting the constant term gives the wavenumber-domain representation:

$$P(\boldsymbol{x}, z) = \sum_{\boldsymbol{n}\in\mathbb{Z}^2} \tilde{p}(\boldsymbol{k}, z) e^{\mathrm{i}\boldsymbol{k}\boldsymbol{x}}, \tag{C.9}$$

where $\boldsymbol{n} \equiv (n_x, n_y)$ spans all integers and $\boldsymbol{k} \equiv (2\pi n_x/L_x, 2\pi n_y/L_y)$ denotes the wavenumber vector. The Fourier coefficient $\tilde{p}(\boldsymbol{k}, z)$ is

$$\begin{aligned}
\tilde{p}(\boldsymbol{k}, z) &= \frac{1}{L_x L_y} \int_0^{L_y} \int_0^{L_x} P(\boldsymbol{x}, z) e^{\mathrm{i}\boldsymbol{k}\boldsymbol{x}} dxdy \\
&= \frac{1}{L_x L_y} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{x^2+y^2+z^2}} e^{\mathrm{i}\boldsymbol{k}\boldsymbol{x}} dxdy \\
&= \frac{2\pi}{L_x L_y} \frac{e^{-|\boldsymbol{k}||z|}}{|\boldsymbol{k}|}.
\end{aligned} \tag{C.10}$$

As a convolution operator, it follows that $\boldsymbol{G}^{(l,n)}\boldsymbol{\sigma} = \mathcal{F}^{-1}\tilde{\boldsymbol{g}}^{(l,n)}\mathcal{F}\boldsymbol{\sigma}$, where $\tilde{\boldsymbol{g}}^{(l,n)}$ is the kernel in wavenumber space. For a desingularization distance h and water depth d, the coefficient matrix represents the superposition of the source above the free surface and its image source, giving the kernel

$$\begin{aligned}
\tilde{g}^{(l,n)} &= \frac{\partial^{l+n}}{\partial z_i^l \partial z_j^n} (\tilde{p}(\boldsymbol{k}, h + z_j - z_i) + \tilde{p}(\boldsymbol{k}, h + 2d + z_j + z_i)) \Big|_{z_i=0, z_j=0} \\
&= \frac{2\pi e^{-|\boldsymbol{k}|h}}{L_x L_y |\boldsymbol{k}|} (-|\boldsymbol{k}|)^{l+n} (e^{-2|\boldsymbol{k}|d} + (-1)^l).
\end{aligned} \tag{C.11}$$

Let the convolution operator $D_0 = \mathcal{F}^{-1}|\boldsymbol{k}|\mathcal{F}$ and the operator $D_1 = \mathcal{F}^{-1}|\boldsymbol{k}|\tanh(|\boldsymbol{k}|d)\mathcal{F}$. Then, the final expressions for the velocity potential up to third order are:

$$\begin{aligned}
\boldsymbol{\phi}_{z0} &= D_1 \boldsymbol{\phi}^S, \\
\boldsymbol{\phi}_{z1} &= (\boldsymbol{\zeta} D_0^2 - D_1 \boldsymbol{\zeta} D_1) \boldsymbol{\phi}^S, \\
\boldsymbol{\phi}_{z2} &= (D_1 \boldsymbol{\zeta} D_1 \boldsymbol{\zeta} D_1 - \frac{1}{2} D_1 \boldsymbol{\zeta}^2 D_0^2 - \boldsymbol{\zeta} D_0^2 \boldsymbol{\zeta} D_1 + \frac{1}{2} \boldsymbol{\zeta}^2 D_1^2 D_0) \boldsymbol{\phi}^S.
\end{aligned} \tag{C.12}$$

These final expressions are exactly equivalent to the DNO formulation given by Bateman et al. [12].

References

[1] Y. Li, B. E. Larsen, D. R. Fuhrman, Reynolds stress turbulence modelling of surf zone breaking waves, J. Fluid Mech. 937 (2022) A7.

[2] W. Mostert, S. Popinet, L. Deike, High-resolution direct simulation of deep water breaking waves: transition to turbulence, bubbles and droplets production, J. Fluid Mech. 942 (2022) A27.

[3] D. G. Boettger, S. R. Keating, M. L. Banner, R. P. Morison, X. Barthélémy, An energetic signature for breaking inception in surface gravity waves, J. Fluid Mech. 959 (2023) A33.

[4] O. Delestre, S. Cordier, F. Darboux, F. James, A limitation of the hydrostatic reconstruction technique for shallow water equations, Cr. Math. 350 (2012) 677–681.

[5] M. Kazolea, A. I. Delis, C. E. Synolakis, Numerical treatment of wave breaking on unstructured finite volume approximations for extended Boussinesq-type equations, J. Comput. Phys. 271 (2014) 281–305.

[6] M. Ricchiuto, A. G. Filippini, Upwind residual discretization of enhanced Boussinesq equations for wave propagation over complex bathymetries, J. Comput. Phys. 271 (2014) 306–341.

[7] F. Marche, Combined hybridizable discontinuous Galerkin (HDG) and Runge-Kutta discontinuous Galerkin (RK-DG) formulations for Green-Naghdi equations on unstructured meshes, J. Comput. Phys. 418 (2020) 109637.

[8] S. Tkachenko, S. Gavrilyuk, J. Massoni, Extended Lagrangian approach for the numerical study of multidimensional dispersive waves: Applications to the Serre-Green-Naghdi equations, J. Comput. Phys. 477 (2023) 111901.

[9] D. G. Dommermuth, D. K. P. Yue, A high-order spectral method for the study of nonlinear gravity waves, J. Fluid Mech. 184 (1987) 267–288.

[10] B. J. West, K. A. Brueckner, R. S. Janda, D. M. Milder, R. L. Milton, A new numerical method for surface hydrodynamics, J. Geophys. Res. 92 (1987) 11803.

[11] W. Craig, C. Sulem, Numerical simulation of gravity waves, J. Comput. Phys. 108 (1993) 73–83.

[12] W. J. D. Bateman, C. Swan, P. H. Taylor, On the efficient numerical simulation of directionally spread surface water waves, J. Comput. Phys. 174 (2001) 277–305.

[13] S. Ryu, M. H. Kim, P. J. Lynett, Fully nonlinear wave-current interactions and kinematics by a BEM-based numerical wave tank, Comput. Mech. 32 (2003) 336–346.

[14] D. Fructus, D. Clamond, J. Grue, Ø. Kristiansen, An efficient model for three-dimensional surface wave simulations: Part I: Free space problems, J. Comput. Phys. 205 (2005) 665–685.

[15] Y.-L. Shao, O. M. Faltinsen, A harmonic polynomial cell (HPC) method for 3D Laplace equation with application in marine hydrodynamics, J. Comput. Phys. 274 (2014) 312–332.

[16] S. Mohanlal, J. C. Harris, M. L. Yates, S. T. Grilli, Simulation of depth-limited breaking waves in a 3D fully nonlinear potential flow model, J. Waterw. Port Coast. Ocean Eng. 150 (2024) 04024007.

[17] M. G. Madsen, S. L. Glimberg, A massively parallel GPU-accelerated model for analysis of fully nonlinear free surface waves, Int. J. Numer. Meth. Fl. 70 (2012) 20–36.

[18] K. Shi, R. Zhu, Efficient spectral coupled boundary element method for fully nonlinear wave–structure interaction simulation, Phys. Fluids 35 (2023).

[19] J. Wang, An enhanced spectral boundary integral method for modeling highly nonlinear water waves in variable depth, J. Comput. Phys. 521 (2025) 113525.

[20] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.

[21] H. Wang, G. Wang, J. Zheng, Q. Liang, A. Tao, Advancements in nearshore wave modeling: A unified one-layer nonhydrostatic approach, Phys. Fluids 35 (2023).

[22] P. A. Madsen, H. B. Bingham, H. Liu, A new Boussinesq method for fully nonlinear waves from shallow to deep water, J. Fluid Mech. 462 (2002) 1–30.

[23] P. A. Madsen, H. B. Bingham, H. Schäffer, Boussinesq-type formulations for fully nonlinear and extremely dispersive water waves: derivation and analysis, Proc. R. Soc. Lond. A Math. Phys. Sci. 459 (2003) 1075–1104.

[24] W. C. Webster, B. Zhao, The development of a high-accuracy, broadband, Green–Naghdi model for steep, deep-water ocean waves, J. Ocean Eng. Mar. Energy. 4 (2018) 273–291.

[25] L. He, B. Zhao, M. Hayatdavoodi, R. C. Ertekin, Three-dimensional deep-water focusing waves by irrotational green–naghdi equations, Appl. Ocean Res. 162 (2025) 104698.

[26] J. G. Herterich, F. Dias, Extreme long waves over a varying bathymetry, J. Fluid Mech. 878 (2019) 481–501.

[27] A. Khait, L. Shemer, On the kinematic criterion for the inception of breaking in surface gravity waves: Fully nonlinear numerical simulations and experimental verification, Phys. Fluids 30 (2018).

[28] S. Liu, T. Waseda, J. Yao, X. Zhang, Statistical properties of surface gravity waves and freak wave occurrence in crossing sea states, Phys. Rev. Fluids 7 (2022) 074805.

[29] S. Kim, B. Bouscasse, G. Ducrozet, M. Canard, G. De Hauteclocque, C. O. Housseine, P. Ferrant, Numerical and experimental study of a FORM-based design wave applying the HOS-NWT nonlinear wave solver, Ocean Eng. 263 (2022) 112287.

[30] Y. Cao, W. W. Schultz, R. F. Beck, Three-dimensional desingularized boundary integral methods for potential problems, Int. J. Numer. Meth. Fl. 12 (1991) 785–803.

[31] R. F. Beck, Fully nonlinear water wave computations using the desingularized method, in: 6th Int. Conf. on Numerical Ship Hydrodynamics, 1993, pp. 3–20.

[32] X. Song, X. Zhang, R. F. Beck, Numerical study on added resistance of ships based on time-domain desingularized-rankine panel method, Ocean Eng. 248 (2022) 110713.

[33] Y. M. Scolan, On some properties of the pressure in nonlinear two-dimensional potential flow with a free surface, Appl. Ocean Res. 150 (2024) 104123.

[34] K. Shi, R. Zhu, Y. Li, Spectral-fundamental solution approach for fully nonlinear ship wave simulations, Phys. Rev. Fluids 11 (2026) 034801.

[35] G. Ducrozet, F. Bonnefoy, D. Le Touzé, P. Ferrant, HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method, Comput. Phys. Commun. 203 (2016) 245–254.

[36] Y. Pan, High-order spectral method for the simulation of capillary waves with complete order consistency, J. Comput. Phys. 408 (2020) 109299.

[37] D. P. Nicholls, F. Reitich, Stability of high-order perturbative methods for the computation of Dirichlet–Neumann operators, J. Comput. Phys. 170 (2001) 276–298.

[38] W. J. D. Bateman, C. Swan, P. H. Taylor, On the calculation of the water particle kinematics arising in a directionally spread wavefield, J. Comput. Phys. 186 (2003) 70–92.

[39] L. Xu, P. Guyenne, Numerical simulation of three-dimensional nonlinear water waves, J. Comput. Phys. 228 (2009) 8446–8466.

[40] M. L. Yates, M. Benoit, Accuracy and efficiency of two numerical methods of solving the potential flow problem for highly nonlinear and dispersive water waves, Int. J. Numer. Meth. Fl. 77 (2015) 616–640.

[41] M. Klahn, P. A. Madsen, D. R. Fuhrman, Simulation of three-dimensional nonlinear water waves using a pseudospectral volumetric method with an artificial boundary condition, Int. J. Numer. Meth. Fl. 93 (2021) 1843–1870.

[42] C. Eskilsson, H. Abedi, A. P. Engsig-Karup, A co-simulation approach to modelling fully nonlinear water waves using the spectral/hp element method, Ocean Eng. 340 (2025) 122385.

[43] V. E. Zakharov, Stability of periodic waves of finite amplitude on the surface of a deep fluid, J. Appl. Mech. Tech. Phy. 9 (1968) 190–194.

[44] A. Moroz, Quasi-periodic Green's functions of the Helmholtz and Laplace equations, J. Phys. A Math. Gen. 39 (2006) 11247–11282.

[45] V. Milisic, U. Razafison, Weighted Sobolev spaces for the Laplace equation in periodic infinite strips, ArXiv preprint arXiv:1302.4253 (2013).

[46] P. P. Ewald, Die Berechnung optischer und elektrostatischer Gitterpotentiale, Ann. Phys. 369 (1921) 253–287.

[47] D. M. Heyes, M. Barber, J. H. R. Clarke, Molecular dynamics computer simulation of surface properties of crystalline potassium chloride, J. Chem. Soc., Faraday Trans. 2 73 (1977) 1485–1496.

[48] J. Lekner, Summation of Coulomb fields in computer-simulated disordered systems, Physica A 176 (1991) 485–498.

[49] S. P. Neill, M. R. Hashemi, Fundamentals of ocean renewable energy: generating electricity from the sea, Academic Press, 2018.

[50] D. Clamond, D. Dutykh, Accurate fast computation of steady two-dimensional surface gravity waves in arbitrary depth, J. Fluid Mech. 844 (2018) 491–518.

[51] H. A. Schäffer, Comparison of Dirichlet–Neumann operator expansions for nonlinear surface gravity waves, Coast. Eng. 55 (2008) 288–294.