

Article

# Numerical Simulations of Tank Sloshing Problems Based on Moving Pseudo-Boundary Method of Fundamental Solution

Chengyan Wang<sup>1,2</sup>, Yuanting Zou<sup>1</sup>, Ji Huang<sup>1,3,\*</sup>  and Chia-Ming Fan<sup>4</sup> 

<sup>1</sup> Naval Architecture and Shipping College, Guangdong Ocean University, Zhanjiang 524088, China; wangcy@mail.nwpu.edu.cn (C.W.); noname939@stu.gdou.edu.cn (Y.Z.)

<sup>2</sup> School of Power and Energy, Northwestern Polytechnical University, Xi'an 710072, China

<sup>3</sup> Doctoral Degree Program in Ocean Engineering Technology, National Taiwan Ocean University, Keelung 20224, Taiwan

<sup>4</sup> Department of Harbor and River Engineering and Computation and Simulation Center, National Taiwan Ocean University, Keelung 20224, Taiwan; cmfan@ntou.edu.tw

\* Correspondence: huangji@gdou.edu.cn

**Abstract:** The moving pseudo-boundary method of fundamental solutions (MFS) was employed to solve the Laplace equation, which describes the potential flow in a two-dimensional (2D) numerical wave tank. The MFS is known for its ease of programming and the advantage of its high precision. The solution of the boundary value can be expressed by a linear combination of the fundamental solutions. The major issue with such an implementation is the optimal distribution of source nodes in the pseudo-boundary. Traditionally, the positions of the source nodes are assumed to be fixed to keep the set of equations closed. However, in the moving boundary problem, the distribution of source nodes may influence the stability of numerical calculations. Moreover, MFS is unstable in time iterations. Hence, it is necessary to constantly revise the weighting coefficients of fundamental solutions. In this study, the source nodes were free, and their locations were determined by solving a nonlinear least-squares problem using the Levenberg–Marquardt algorithm. To solve the above least-squares problem, the MATLAB© routine lsqnonlin was adopted. Additionally, the weighting coefficients of fundamental solutions were solved as a nonlinear least-squares problem using the aforementioned method. The numerical results indicated that the numerical simulation method adopted in this paper is accurate and reliable in solving the problem of 2D tank sloshing. The main contribution of this study is to expand the application of the MFS in engineering by integrating it with the optimal configuration problem of pseudo-boundaries to solve practical engineering problems.

**Keywords:** sloshing; method of fundamental solutions; adaptive source nodes' placement; second-order Runge–Kutta



**Citation:** Wang, C.; Zou, Y.; Huang, J.; Fan, C.-M. Numerical Simulations of Tank Sloshing Problems Based on Moving Pseudo-Boundary Method of Fundamental Solution. *J. Mar. Sci. Eng.* **2023**, *11*, 1448. <https://doi.org/10.3390/jmse11071448>

Academic Editor: Constantine Michailides

Received: 29 June 2023

Revised: 12 July 2023

Accepted: 17 July 2023

Published: 19 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Sloshing refers to the nonlinear changes of the free surface of a partially filled liquid tank with time under an external force [1,2]. This phenomenon is widespread in engineering, such as the shaking of aircraft fuel tanks, the sloshing of ship tanks, and the shaking of oil storage tanks during earthquakes [3]. When the oscillation frequency or resonance is close to the natural frequency of the numerical wave tank (NWT), it can cause structural damage, resulting in serious consequences. Therefore, insights into the physical mechanisms of liquid sloshing enable the quick prediction of the free surface changes. Over the past few decades, many researchers have adopted a variety of research methods to study the problem of tank sloshing, including theoretical research, experiments, and numerical simulations. In the study of theoretical research and numerical simulations, the fluid in the tank is usually assumed to be an inviscid, irrotational, and incompressible potential flow to simplify the calculations. In this paper, the fluid in the NWT was also assumed to be a potential flow, and its governing equation was considered to be the Laplace equation.

In the past few decades, numerical simulation has become a mainstream research method due to its convenience and low cost. Therefore, many numerical schemes have been proposed to simulate nonlinear waves in the sloshing problem, including mesh and meshless methods. For the mesh method, several numerical schemes are widely implemented, including the finite element method (FEM) [4,5], the finite volume method (FVM) [6,7], the boundary element method (BEM) [8], and the finite difference method (FDM) [9]. Gómez-Goñi et al. [6] adopted the volume of fluid (VOF) method and compared the numerical simulation results to the multimodal codes STAR-CCM+ and Open-FOAM. Both open-source and commercial CFD software packages have high accuracy. Moreover, Tang et al. [7] compared the suppression effect of different height distribution schemes on tank sloshing by setting different baffle height gradients. Wang et al. [10] investigated liquid sloshing under irregular excitations and explained the mechanism of sloshing responses. However, since meshless methods have simple programs and do not require time-consuming numerical integration, they have progressed rapidly with various formulations, such as the radial basis function collocation method (RBF-CM) [11], the modified collocation Trefftz method (MCTM) [12], the generalized finite difference method (GFDM) [13], and the meshless local Petrov–Galerkin (MLPG) method [14]. Zhang et al. [13] proposed the GFDM, an easy-to-program and simple numerical scheme, to deal with the 2D Laplace equation at each time step and predict the nonlinear free surface in the NWT efficiently and accurately. Pal et al. [14] used the MLPG method to evaluate the pressure of fluid in an oscillated rectangular tank and developed a local symmetric weak form (LSWF) for linearized sloshing. It can be observed that the development of efficient and accurate numerical methods holds significant research value in addressing the sloshing problem. Ren et al. [15] simulated a series of sloshing phenomena, including sloshing without baffle, sloshing with a rigid baffle, and sloshing with an elastic baffle, based on SPHinXsys, which is an open-source SPH-based multi-physics library. Zhang et al. [16] simulated and studied the damping effect of a vertical slotted screen under rotation excitation based on the BM-MPS method and discussed the influence of baffle porosity and rotation amplitude on the resonance period and impact pressure. Li et al. [17] investigated the sloshing effects from baffled and non-baffled water tanks under different external excitation frequencies and amplitudes based on the weakly compressible moving particle explicit (WCMPE) method. Additionally, Gholamipour et al. [18] developed a meshless numerical method based on local radial basis functions for studying the sloshing of arbitrarily shaped tanks, such as trapezoidal, V-shaped, semi-elliptical, and semi-circular tanks under horizontal motion. Moreover, artificial intelligence (AI) techniques are being used to deal with the sloshing problem [19]. The above survey of the current research status shows that the development of an efficient and accurate computer simulation model for the liquid sloshing problem is still one of the key research directions in the international academic community. Although traditional numerical schemes have been proven and are stable, they often require time-consuming work, such as building the mesh and performing numerical integration. In this paper, we propose an easy-to-program and simple numerical scheme to accurately and efficiently analyze sloshing problems. In this study, the method of fundamental solutions (MFS) and the second-order Runge–Kutta method were adopted for spatial and temporal discretizations of this moving-boundary problem. The MFS has the great advantages of decreasing computational time and being easy to implement. Discretization by the second-order Runge–Kutta method solved the elevation of the free surface and boundary value problems (BVPs) at each time step. Likewise, the MFS, a boundary-type meshless method, was proposed to deal with BVPs at every time step.

The MFS was proposed in 1964 to solve BVPs governed by certain partial differential equations [20,21], such as the Laplace equation, Helmholtz equation, modified Helmholtz equation, and Biharmonic equation [22–24]. Since the MFS was used to analyze the elliptic partial differential equation in 1977, it has garnered attention for its simplicity in programming [25]. The MFS is composed of a linear combination of fundamental solutions of partial differential equations with weightings. These fundamental solutions

can be expressed by source and boundary nodes, which are allocated in the pseudo- and physical boundaries, respectively. Since the boundary nodes are known, it is critical to allocate the source nodes. Traditionally, it has been assumed that the source nodes are known and fixed [26]. There have been many investigations on the optimal allocation and shape selection of pseudo-boundaries [27–29]. Furthermore, some researchers have adopted artificial intelligence, such as the genetic algorithm and immune algorithm, to deal with the placement of source nodes, which is also known as the adaptive dynamic approach [30,31]. In 2019, Grabski et al. proposed the moving pseudo-boundary MFS to solve nonlinear potential problems. The dynamic approach was modified by assuming that the parameters controlling the position of each source were unknown [32]. The system of nonlinear equations resulting from the MFS dynamic approach was solved with the state-of-the-art MATLAB© routine `lsqnonlin`. Grabski et al. studied steady problems and further researched unsteady problems where the physical boundary was also moving.

In this study, the Levenberg–Marquardt algorithm was adopted to solve nonlinear least-squares problems for weighting the coefficients of the fundamental solution and the appropriate distribution of the source nodes. The parameter that controls the position of sources is the distance of the sources along the normal to the boundary. The parameters were updated with time iteration. Additionally, the effect of three shapes of pseudo-boundaries on the computational accuracy and efficiency was investigated. The shape of the pseudo-boundary was taken to be similar to that of the boundary, a circle enclosing the domain, or pseudo-boundaries assigned outward only at each edge of the domain. The numerical results revealed that the calculation procedure used in this study exhibited good stability and accuracy, even with a small number of collocation nodes. Although some scholars have successfully conducted sloshing studies using the MFS [33], some deployment methods have obtained less-satisfactory results, indicating the poor stability of the calculation. Therefore, achieving the optimal source node configuration of the pseudo-boundary in the method of fundamental solutions is crucial. Compared with traditional methods, the least-squares-based MFS proposed in this paper remains computationally stable with fewer source nodes. In summary, this paper provides a good example of solving the optimal configuration problem of pseudo-boundaries for engineering problems.

The paper is organized as follows. Section 2 presents the governing equation and boundary conditions. Section 3 describes the numerical methods, including the details of the MFS implementation and time marching. Section 4 presents a comparison between the numerical results presented in this paper and those reported in the literature. Specifically, the results include standing waves in a fixed numerical tank, a vertically excited numerical tank, and a horizontally excited wave tank. Finally, Section 5 provides some concluding remarks and ideas for future work.

## 2. Governing Equation and Boundary Conditions

In this paper, we considered the two-dimensional liquid sloshing problem in a rectangular NWT. The NWT was moved by excitation in the horizontal and vertical directions, respectively, in an inertial Cartesian coordinate system  $(X, Z)$ , where the horizontal and vertical axes are denoted by  $X$  and  $Z$ , respectively. To describe the fluid motion, it is convenient to refer to a moving Cartesian coordinate system  $(x, z)$ , as shown in Figure 1.

Since the liquid was assumed to be inviscid, incompressible, and irrotational, it is governed by the Laplace equation for velocity potential:

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial z^2} = 0, \quad (x, z) \in \Omega, \quad (1)$$

where  $\phi$  is the velocity potential and  $\Omega$  is the computational domain. The wall of the NWT is rigid and impenetrable, corresponding to the second type of boundary condition.

Therefore, the bottom boundary and the vertical walls on both sides of the NWT need to satisfy the impenetrable boundary condition:

$$\frac{\partial \phi}{\partial z} \Big|_{z=0} = 0, \tag{2}$$

$$\frac{\partial \phi}{\partial x} \Big|_{x=0,b} = 0. \tag{3}$$

where  $b$  is the breadth of the tank.

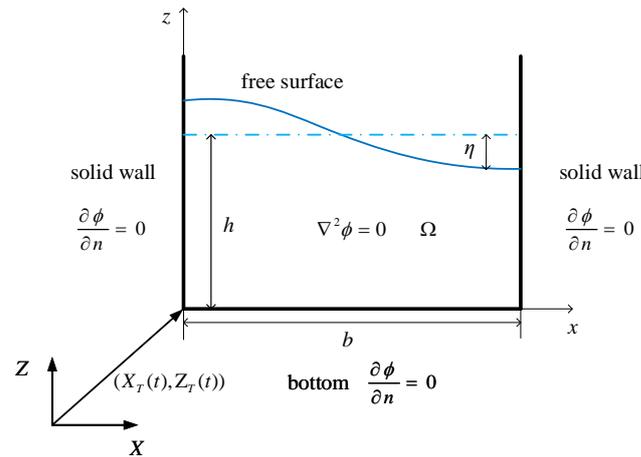


Figure 1. The schematic diagram of the NWT of the sloshing phenomenon.

As for the free liquid surface, also known as the first boundary condition, the following dynamic boundary conditions must be satisfied:

$$\phi \Big|_{z=\eta+h} = \phi, \tag{4}$$

where  $\eta = \eta(x, t)$  denotes the free surface elevation, which is calculated from the still water level, and  $h$  denotes the height of the still water.

The dynamic and kinematic boundary conditions, denoted by  $\phi$  and  $\eta$ , respectively, are updated in the time domain using the following equation: [9]:

$$\frac{\partial \phi}{\partial t} \Big|_{z=\eta+h} = \frac{1}{2} \left[ \left( \frac{\partial \phi}{\partial x} \right)^2 + \left( \frac{\partial \phi}{\partial z} \right)^2 \right] - (g + Z_T''(t))\eta - xX_T''(t), \tag{5}$$

$$\frac{\partial \eta}{\partial t} \Big|_{z=\eta+h} = \frac{\partial \phi}{\partial z} - \frac{\partial \phi}{\partial x} \frac{\partial \eta}{\partial x}, \tag{6}$$

where  $g$  denotes the acceleration due to gravity and  $X_T''$  and  $Z_T''$  denote the acceleration in the horizontal and vertical directions of the NWT, respectively.

Therefore, the dynamic and kinematic boundary conditions for the free surface can be expressed using the following semi-Lagrangian approach [13]:

$$\frac{\partial \phi}{\partial t} \Big|_{z=\eta+h} = \frac{1}{2} \left[ \left( \frac{\partial \phi}{\partial x} \right)^2 + \left( \frac{\partial \phi}{\partial z} \right)^2 \right] + \frac{\partial \phi}{\partial z} \left( \frac{\partial \phi}{\partial z} - \frac{\partial \phi}{\partial x} \frac{\partial \eta}{\partial x} \right) - (g + Z_T''(t))\eta - xX_T''(t), \tag{7}$$

$$\frac{\partial \eta}{\partial t} \Big|_{z=\eta+h} = \frac{\partial \phi}{\partial z} - \frac{\partial \phi}{\partial x} \frac{\partial \eta}{\partial x}. \tag{8}$$

### 3. Numerical Methods

For temporal discretization, the second-order Runge–Kutta method was adopted due to its second-order accuracy and ease of implementation. The MFS was proposed for spatial

discretization. The boundary nodes collocated on the free surface were only allowed to move vertically, and the task of distributing nodes was executed twice at each time step.

Since the potential flow was governed by the Laplace equation, the MFS was implemented to analyze the solutions. Additionally, the MATLAB® routine lsqnonlin was utilized to solve the weightings of the fundamental solutions and the locations of the fundamental solutions. Detailed descriptions of the time-marching method using the second-order Runge–Kutta and MFS are provided in the following subsections.

### 3.1. The Implementation of MFS

The values of the boundary nodes and inner nodes can be expressed as a linear combination of the fundamental solution and the weightings of the source nodes. Figure 2 illustrates the schematic diagram of the MFS. The boundary nodes are arranged on physical boundaries, and the source nodes are arranged on a pseudo-boundary outside the computational domain. The fundamental solution of the Laplace equation is defined as [20–22]:

$$u_i^* = \ln(r_{ij}), \tag{9}$$

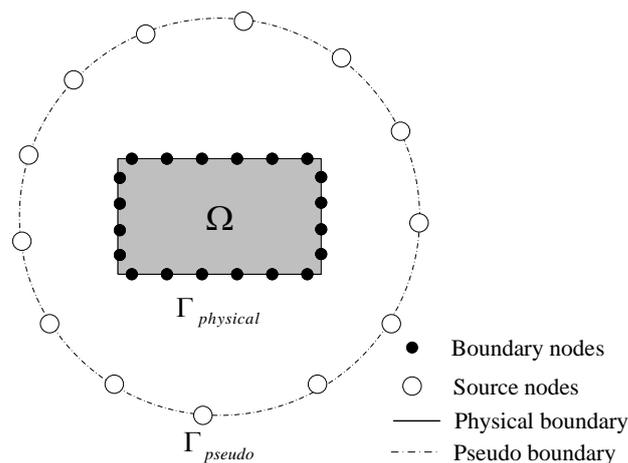
where  $r_{ij}$  is the distance from any boundary nodes  $\mathbf{x}_i$ , where  $\mathbf{x}_i = (x_i, z_i) \in \Gamma_{\text{physical}}$ , to the source nodes  $\mathbf{s}_j$ , where  $\mathbf{s}_j = (s_j^x, s_j^z) \in \Gamma_{\text{pseudo}}$ . It can be expressed as:

$$r_{ij} = |\mathbf{x}_i - \mathbf{s}_j|, \tag{10}$$

Since the boundary conditions already satisfied the governing equation, they could be expressed as a linear combination of the fundamental solutions with various weightings. Furthermore, the boundary condition is also known. Therefore, the weightings can be solved:

$$\phi(x_i, z_i) = \sum_{j=1}^N \alpha_j u_i^*, \tag{11}$$

where  $\alpha_j$  represents the weightings of  $u_i^*$  and  $N$  denotes the number of source nodes.



**Figure 2.** The sketch of the boundary nodes and source nodes distributed on the physical boundary and pseudo-boundary.

Conventionally, the source nodes are fixed, and only the weightings  $\alpha_j$  are unknown. The number of boundary nodes is denoted by  $M$ , and it was set to be greater than or equal to the number of source nodes, i.e.,  $M \geq N$ .

However, in the current approach, the source nodes are free, which means that the parameter controlling the source position is unknown. Therefore, more boundary nodes are required to ensure that the number of equations is at least as large as the number of unknowns. Two schemes were proposed to allocate the source nodes. The boundary nodes are denoted by

$(\mathbf{x}_i)_{i=1}^M$ , where  $\mathbf{x}_i = (x_i, z_i) \in \Gamma_{\text{physical}}$ , and  $(\tilde{\mathbf{x}}_j)_{j=1}^M$ , where  $\tilde{\mathbf{x}}_j = (x_j, z_j) \in \Gamma_{\text{physical}}$ . Here,  $\Gamma_{\text{physical}}$  represents the physical boundary of the NWT.

In Scheme 1, the source node is defined as  $(\mathbf{s}_j)_{j=1}^N$ , where  $\mathbf{s}_j = (s_j^x, s_j^z)$ :

$$(s_j^x, s_j^z) = (x_c, z_c) + \beta[(x_j, z_j) - (x_c, z_c)], \tag{12}$$

where  $(x_c, z_c)$  denotes the coordinates of the center of the domain. The unknown parameter  $\beta$  controls the magnification of the distance of the node  $\tilde{\mathbf{x}}_j$  from the center of the domain. The rules for laying out the nodes should satisfy  $M \geq N + 1$ .

In Scheme 2, the source node is defined as  $(\mathbf{s}_j)_{j=1}^N$ , where  $\mathbf{s}_j = (s_j^x, s_j^z)$ :

$$(s_j^x, s_j^z) = (x_c, z_c) + \sigma_j[(x_j, z_j) - (x_c, z_c)], \tag{13}$$

where  $\sigma_j = (\sigma_1, \sigma_2, \dots, \sigma_N)$  is a set of column vectors that give  $\mathbf{s}_j$ , a unique parameter controlling the distance between the source  $\mathbf{s}_j$  and the node  $\tilde{\mathbf{x}}_j$ . Since a total of  $2N$  unknown quantities are introduced, the deployment method should satisfy  $M \geq 2N$ .

In addition, it was assumed that the weightings are known and the initial values are zero. The initial values of the parameters  $\beta$  and  $\sigma$ , which denote the normal distance between the physical and pseudo-boundaries, were set to 2.0. The numerical solution can be obtained from the linear combination of the fundamental solutions with various weightings. If these solutions are incorrect, a corrective action should be taken. The MATLAB® routine lsqnonlin was used to analyze the following discretized nonlinear equation system:

$$y_i = \begin{cases} \left( \phi(\mathbf{x}_i) - \sum_{j=1}^N \alpha_j u_i^* \right) & (x_i, y_i) \in \text{Free surface}, \\ \left( 0 - \frac{\partial \left( \sum_{j=1}^N \alpha_j u_i^* \right)}{\partial n} \right) & (x_i, y_i) \in \text{Other boundary}, \end{cases} \tag{14}$$

and the routine lsqnonlin was used to converge  $y_i$ , where  $i = (1, 2, \dots, M)$ , to 0. In this process, the computational program iterates until the convergence criterion is reached. The options of lsqnonlin that were set are displayed in Table 1. At the end of the lsqnonlin iteration, the approximate values of the weightings  $\alpha$  and the parameter  $\beta$  or  $\sigma$  are obtained. The parameters  $\beta$  and  $\sigma$  are updated adaptively with lsqnonlin, meaning that the pseudo-boundary moves with each iteration.

**Table 1.** Customization options of lsqnonlin.

Options of lsqnonlin	Default Value	Set Value
StepTolerance	$1 \times 10^{-6}$	$1 \times 10^{-7}$
FunctionTolerance	$1 \times 10^{-6}$	$1 \times 10^{-16}$
MaxFunctionEvaluations	100 × number of variables	$1 \times 10^7$
MaxIterations	400	$1 \times 10^5$
OptimalityTolerance	$1 \times 10^{-6}$	$1 \times 10^{-12}$

### 3.2. Time Marching

In order to solve Equations (1)~(6), the second-order Runge–Kutta method was used to perform the time marching of  $\phi$  and  $\eta$ . This method provides second-order accuracy in time. Since Equations (2) and (3) are time-independent, the second-order Runge–Kutta method were used to update the free surface boundary Equations (5) and (6) as follows:

$$\phi^n|_{z=\eta+h} = \frac{1}{2}(\phi^{n-1} + \phi^{n*}), \tag{15}$$

$$\eta^n|_{z=\eta+h} = \frac{1}{2}(\eta^{n-1} + \eta^{n*}), \tag{16}$$

where  $\phi^{n-1}$  and  $\eta^{n-1}$  represent the velocity potential and free surface elevation at the  $n - 1$ th time step, respectively, while  $\phi^n$  and  $\eta^n$  represent the velocity potential and free surface elevation at the  $n$ th time step, respectively. Additionally,  $\phi^{n*}$  and  $\eta^{n*}$  can be expressed as follows:

$$\phi^{n*}|_{z=\eta+h} = \phi^{n*-1} + \Delta t \left\{ \frac{1}{2} \left[ \left( \frac{\partial \phi^{n*-1}}{\partial x} \right)^2 + \left( \frac{\partial \phi^{n*-1}}{\partial z} \right)^2 \right] + \frac{\partial \phi^{n*-1}}{\partial z} \left( \frac{\partial \phi^{n*-1}}{\partial z} - \frac{\partial \phi^{n*-1}}{\partial x} \frac{\partial \eta^{n*-1}}{\partial x} \right) - \left( g + Z_T''(t^{n*-1}) \right) \eta^{n*-1} - x X_T''(t^{n*-1}) \right\}, \tag{17}$$

$$\eta^{n*}|_{z=\eta+h} = \eta^{n*-1} + \Delta t \left( \frac{\partial \phi^{n*-1}}{\partial z} - \frac{\partial \phi^{n*-1}}{\partial x} \frac{\partial \eta^{n*-1}}{\partial x} \right), \tag{18}$$

where  $\phi^{n*-1}$  and  $\eta^{n*-1}$  can be expressed in a similar form as  $\phi^{n-1}$  and  $\eta^{n-1}$ :

$$\phi^{n*-1}|_{z=\eta+h} = \phi^{n-1} + \Delta t \left\{ \frac{1}{2} \left[ \left( \frac{\partial \phi^{n-1}}{\partial x} \right)^2 + \left( \frac{\partial \phi^{n-1}}{\partial z} \right)^2 \right] + \frac{\partial \phi^{n-1}}{\partial z} \left( \frac{\partial \phi^{n-1}}{\partial z} - \frac{\partial \phi^{n-1}}{\partial x} \frac{\partial \eta^{n-1}}{\partial x} \right) - \left( g + Z_T''(t^{n-1}) \right) \eta^{n-1} - x X_T''(t^{n-1}) \right\}, \tag{19}$$

$$\eta^{n*-1}|_{z=\eta+h} = \eta^{n-1} + \Delta t \left( \frac{\partial \phi^{n-1}}{\partial z} - \frac{\partial \phi^{n-1}}{\partial x} \frac{\partial \eta^{n-1}}{\partial x} \right), \tag{20}$$

In Equations (17)~(20), variables  $\frac{\partial \eta}{\partial x}$ ,  $\frac{\partial \phi}{\partial x}$ , and  $\frac{\partial \phi}{\partial z}$  can be gained through the following expressions.

$$\begin{cases} \frac{\partial \eta}{\partial x} = \frac{\eta(0+\Delta x) - \eta(0)}{\Delta x}, & x = 0 \\ \frac{\partial \eta}{\partial x} = \frac{\eta(x+\Delta x) - \eta(x-\Delta x)}{2\Delta x}, & x \in [\Delta x, b - \Delta x], \\ \frac{\partial \eta}{\partial x} = \frac{\eta(b) - \eta(b-\Delta x)}{\Delta x}, & x = b \end{cases} \tag{21}$$

$$\begin{pmatrix} \left( \frac{\partial \phi}{\partial x_1}, \frac{\partial \phi}{\partial z_1} \right) \\ \left( \frac{\partial \phi}{\partial x_2}, \frac{\partial \phi}{\partial z_2} \right) \\ \vdots \\ \left( \frac{\partial \phi}{\partial x_n}, \frac{\partial \phi}{\partial z_n} \right) \end{pmatrix} = \begin{pmatrix} \frac{r_{11}^{-1} \vec{n}_1}{r_{11}^2} & \frac{r_{12}^{-1} \vec{n}_1}{r_{12}^2} & \dots & \frac{r_{1n}^{-1} \vec{n}_1}{r_{1n}^2} \\ \frac{r_{21}^{-1} \vec{n}_2}{r_{21}^2} & \frac{r_{22}^{-1} \vec{n}_2}{r_{22}^2} & \dots & \frac{r_{2n}^{-1} \vec{n}_2}{r_{2n}^2} \\ \vdots & \vdots & \dots & \vdots \\ \frac{r_{m1}^{-1} \vec{n}_m}{r_{m1}^2} & \frac{r_{m2}^{-1} \vec{n}_m}{r_{m2}^2} & \dots & \frac{r_{mn}^{-1} \vec{n}_m}{r_{mn}^2} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}, \text{ on } \Gamma_{FS}, \tag{22}$$

In Equation (21), the first term of  $\frac{\partial \eta}{\partial x}$  is obtained using the forward difference scheme, the last term is obtained using the backward difference scheme, and the remaining terms are obtained using the central difference scheme. In Equation (22),  $r_{mn}^2$  denotes the square of the distance between the boundary node and the source node, where  $m$  is the number of boundary nodes and  $n$  is the number of source nodes. Here,  $\vec{n}$  represents the normal vector of the free surface and  $\alpha_m$  denotes the previously obtained weighting coefficients.

To stabilize the solution, ninth-order polynomials were used to fit the free surface, dynamic, and kinematic boundary conditions. The coefficients of the ninth-order polynomials were determined using the least-squares method. The form of a ninth-order polynomial is as follows (The flowchart of the proposed meshless numerical scheme for solving sloshing phenomena is presented in Figure 3.):

$$p(x) = p_1 x^9 + p_2 x^8 + \dots + p_9 x + p_{10}, \tag{23}$$

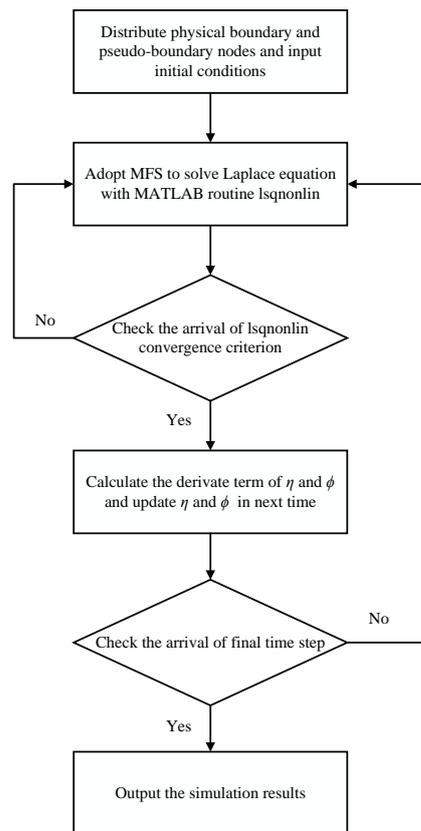


Figure 3. Flowchart of the proposed numerical method.

#### 4. Numerical Results and Comparisons

All numerical computations were performed on a MATLAB© R2018b platform running on Windows 10 (64 bit) with an Intel Core(TM) i5–10600KF CPU and 32 GB of memory. To verify the accuracy of the proposed numerical method, three examples are provided, including standing waves in a fixed NWT, a vertically excited NWT, and a horizontally excited NWT. These examples have been studied by other researchers in the past [9,34,35], allowing for comparisons to verify the accuracy and reliability of the numerical method presented in this paper. It was assumed that the width of the NWT is  $b = 1$  m, the still water elevation is  $h = 0.5$  m, and the number of waves is  $k_1 = \pi/b$ . The natural frequency of water can be expressed as:

$$\omega_1 = \sqrt{gk_1 \tanh(k_1 h)}, \tag{24}$$

##### 4.1. Standing Waves in Fixed NWT

For the fixed NWT with standing waves, the NWT was not subjected to any external forces in the horizontal and vertical directions, i.e.,  $X_T'' = 0$  and  $Z_T'' = 0$ . The following functions represent the initial dynamic and kinematic boundaries:

$$\phi(x, z)|_{t=0} = 0, \tag{25}$$

$$\eta(x)|_{t=0} = a \cos(k_1 x), \tag{26}$$

where the amplitude of the initial wave profile is  $a = g\varepsilon/\omega_1^2$ , where  $g$  is the acceleration due to gravity,  $\varepsilon = 0.0014$  represents the wave steepness, and  $\omega_1$  denotes the natural frequency of water. Based on the given boundaries and initial conditions, the free surface will vibrate freely in the form of standing waves due to gravity.

#### 4.1.1. Convergence Study of the Number of Nodes and Time Step Size

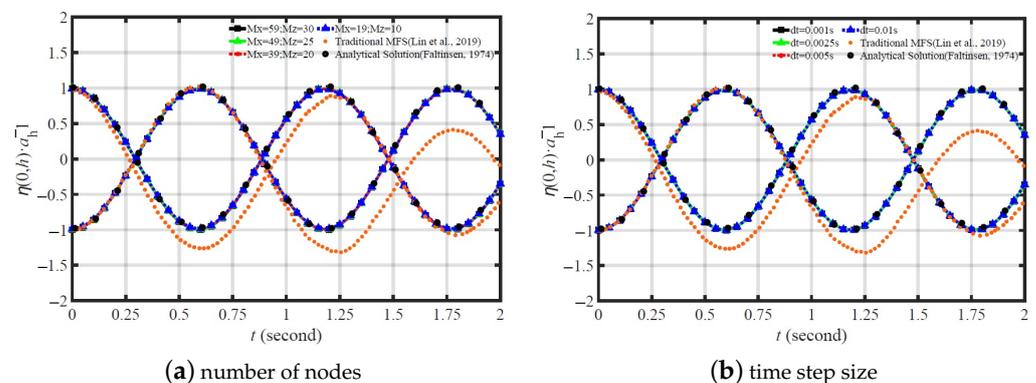
To investigate the dependency between the number of boundary nodes and the accuracy of the numerical results, four cases were simulated to check for the convergence of the numerical results. The specific settings for each case are displayed in Table 2. The number of nodes along the  $x$  and  $z$  directions for the physical boundaries is denoted by  $M_x$  and  $M_z$ , respectively, while for the pseudo-boundaries, the number of nodes along the  $x$  and  $z$  directions is denoted by  $N_x$  and  $N_z$ , respectively. Two allocation methods were adopted to investigate the computational efficiency. In Scheme 2, the introduction of more variables complicated the nonlinear system of equations, thereby disrupting the computational convergence process and increasing the computation time. Thus, Scheme 1 was adopted to allocate the source nodes. The convergence of the solution was checked at each time step, and the allocation of fewer nodes on the boundaries did not result in shorter computation times. While fewer nodes were allocated at the boundaries in Case 4 of Table 2, the computation time was still high.

**Table 2.** Details of nodes' convergence case settings.

Case Number	Number of Physical Boundary Nodes	Number of Pseudo-Boundary Nodes	Time Cost (Scheme 1)	Time Cost (Scheme 2)
Case 1	$M_x = 59, M_z = 30$	$N_x = 29, N_z = 15$	3232.9 s	6998.1 s
Case 2	$M_x = 49, M_z = 25$	$N_x = 19, N_z = 10$	1094.1 s	2658.6 s
Case 3	$M_x = 39, M_z = 20$	$N_x = 19, N_z = 10$	1021.7 s	2186.1 s
Case 4	$M_x = 19, M_z = 10$	$N_x = 9, N_z = 5$	1245.2 s	2879.5 s

The numerical results for the four cases are presented in Figure 4a, which displays the free surface elevation at both walls. As shown in the figure, the numerical results for all cases were in good agreement with Faltinsen's analytical solutions [35], confirming the accuracy and reliability of the proposed numerical method. Considering the time cost and the fact that the numerical solutions were very close, the node-allocation method used in Case 3 was chosen. The results were then compared with those obtained by Lin et al. [33], under the condition that the source node configuration method was consistent. The calculated results exhibited good accuracy and stability.

In order to investigate time step independence, a time step analysis was conducted, as shown in Figure 4b. Time step sizes of  $\Delta t = 0.001$  s, 0.0025 s, 0.005 s, and 0.01 s were selected. Even with a relatively large time step, the calculated results were still in good agreement with the literature [35]. The figure demonstrates that the proposed numerical scheme has good reliability and accuracy, while also possessing good computational efficiency. In future research, a larger time step may be selected to save the computation time.



**Figure 4.** The convergence study of the free surface elevation at both walls for the standing wave problem.

#### 4.1.2. The Numerical Solution of Standing Waves

After conducting the convergence study of the number of nodes and the time step sizes, a complete numerical simulation was performed, and the calculated results were compared with Frandsen’s numerical solution [9]. In addition, to further verify the accuracy of the method, the complete numerical results were also compared with those obtained by the finite difference method [9] and radial basis function (RBF) method [18]. As shown in Figure 5, the peaks and troughs of the relative amplitudes were the same, indicating that the sloshing amplitude of the free surface would remain constant without any energy loss, as long as gravity was considered in the potential flow field. The profiles of the free surface at  $t \times \omega_1 = 0, t \times \omega_1 = 19.94, t \times \omega_1 = 20.73, t \times \omega_1 = 21.80$ , and a still water level are evaluated in Figure 6. As the free surface profile varied with time, the computational domain required node reallocation at each time step. However, node reallocation was easily managed by the MFS, highlighting its advantages as a meshless method.

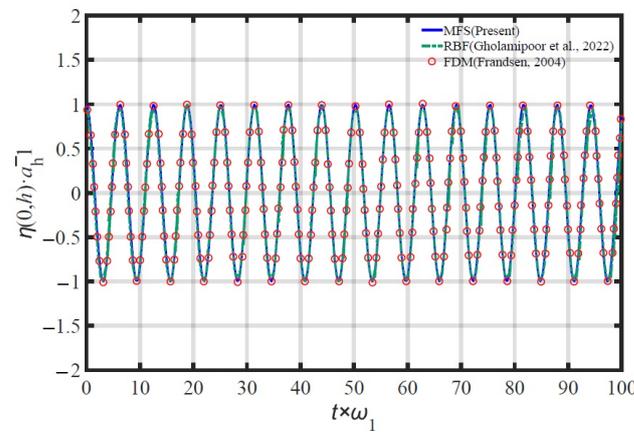


Figure 5. Comparison of the relative amplitude of left-end node of the free surface under free vibrations.

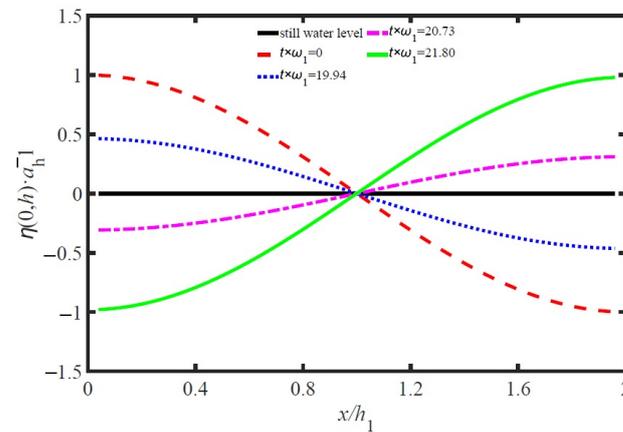


Figure 6. Wave profiles along the free surface at some specific time.

#### 4.2. Vertically Excited NWT

For the simulation of sloshing in a vertically excited NWT, the acceleration in the vertical direction is expressed as follows:

$$Z_T'' = -a_v \omega_v^2 \cos(\omega_v t), \tag{27}$$

where  $a_v = gk_v/\omega_v^2$  is the vertical forcing amplitude,  $k_v = 0.5$  is the non-dimensional forcing amplitude, and  $\omega_v = \omega_1/1.253$  is the angular frequency. In this case, the horizontal acceleration was assumed to be zero. The initial velocity potential condition is  $\phi(x, z)|_{t=0} = 0$  and  $X_T'' = 0$ . The initial kinematic condition is  $\eta(x)|_{t=0} = a \cos(k_1 x)$ ,

where  $k_1 = \pi/b$  is the wave number,  $a = (g\varepsilon)/\omega_1^2$ , and  $\varepsilon = 0.0014$  is the wave steepness. The parameter  $\omega_1$  is the inherent frequency of the liquid, which can be obtained from Equation (24). Likewise, the allocation method of Case 3 in Table 2 was chosen to improve the calculation efficiency. The time step size was set to  $\Delta t = 0.0025$  s.

In this section, we investigated the effect of different pseudo-boundaries on the calculation accuracy and efficiency. Three types of pseudo-boundaries were considered. The first type was the domain-dependent pseudo-boundary given by Equation (12). The initial value of  $\beta$  was set to 2.0.

For the second type of pseudo-boundary geometry, a circular shape was selected, with the pseudo-boundary points evenly distributed around the circle.

$$(s_j^x, s_j^z) = (x_c + R \cos \theta_j, z_c + R \sin \theta_j), \tag{28}$$

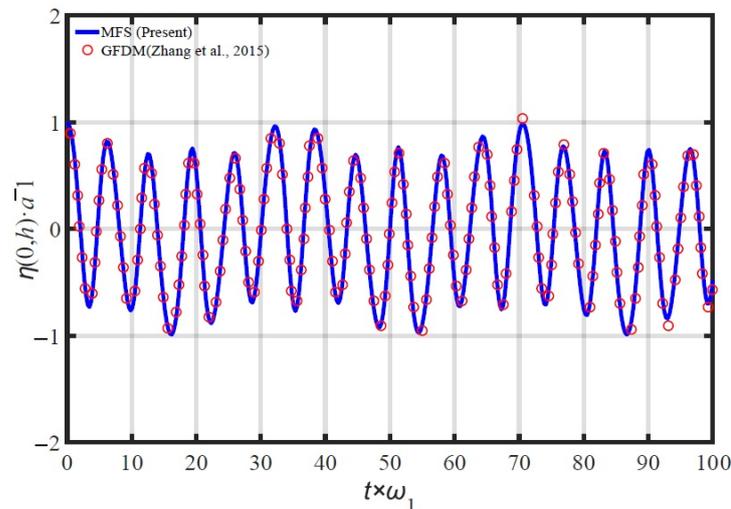
where  $R$  is the radius of the circle and  $\theta_j$  is the corresponding radian.

The third pseudo-boundary is defined by the following equation:

$$(s_j^x, s_j^z) = (x_j, z_j) + \beta \vec{n}, \tag{29}$$

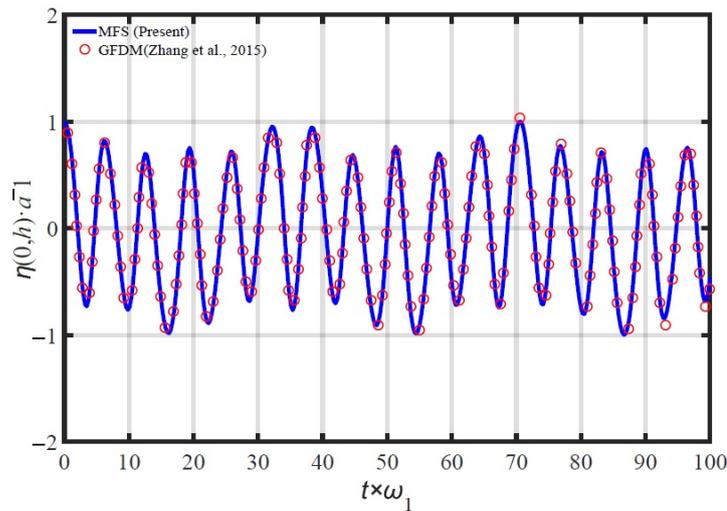
where  $\vec{n}$  is the normal vector of the boundary nodes.

Figure 7 displays a comparison of the results for all three pseudo-boundaries. The evolutionary profile at the left-end node of the free surface exhibited high nonlinearity and irregularity. The time cost was evaluated to further investigate the computational efficiency under different pseudo-boundary types (Table 3).

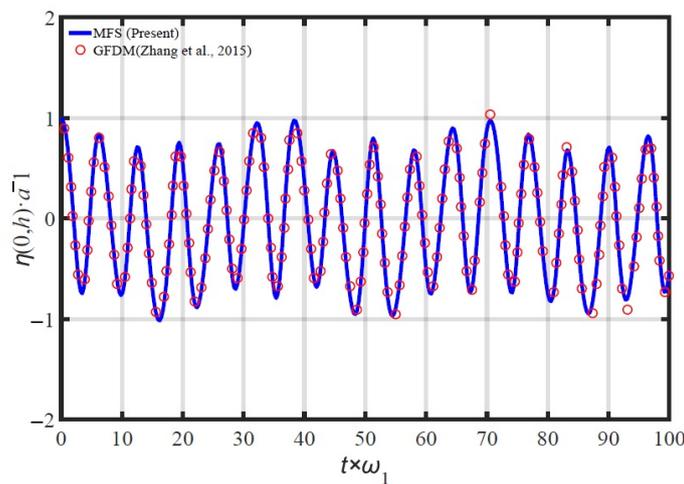


(a) Calculation results under the first pseudo-boundary geometry

Figure 7. Cont.



(b) Calculation results under the second pseudo-boundary geometry



(c) Calculation results under the third pseudo-boundary geometry

Figure 7. Evolutionary profile of the left-end node of the free surface.

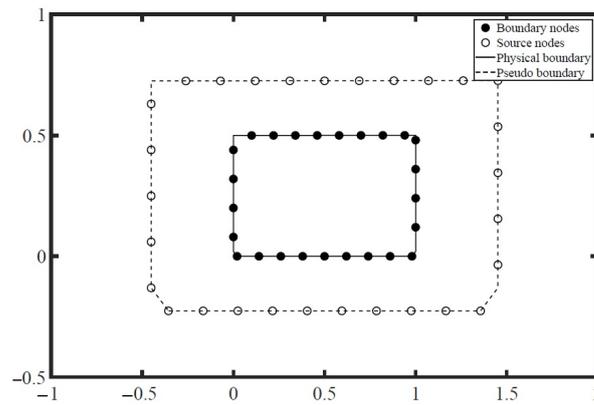
Table 3. Details of time consumption.

Case	Number of Physical Boundary Nodes	Number of Pseudo-Boundary Nodes	Time Consumption
Domain-dependent pseudo-boundary type	148	58	3.36 h
Circular pseudo-boundary type	148	58	2.10 h
Boundary-dependent pseudo-boundary type	148	58	4.19 h

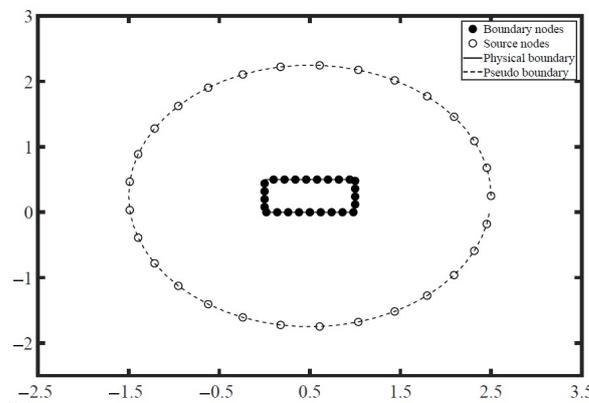
As observed in Table 3, the circular pseudo-boundary type was the most-efficient type, with approximately twice the efficiency of the boundary-dependent type. The shape of the pseudo-boundary affects the computational efficiency. On the contrary, the type of pseudo-boundary does not affect its efficiency, but has an impact on accuracy. In the current approach, the shape of the pseudo-boundary affected the convergence of the lsqnonlin routine, indicating the importance of an optimized pseudo-boundary geometry.

Likewise, this reflected the stability of the correction function of lsqnonlin in solving the Laplace equation.

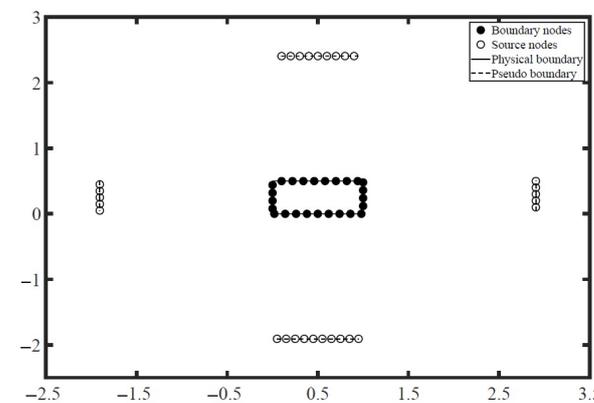
Boundaries and source nodes are located in their respective physical and pseudo-boundaries (Figure 8). Figure 8a–c show domain-dependent-type, circular-type, and boundary dependent-type pseudo-boundary, respectively. In the current approach, the shape of the pseudo-boundary was fixed, but its size was adaptively changing in each iteration. The wave profile along the free surface at  $t \times \omega_1 = 19.13$  and  $t \times \omega_1 = 22.33$  is depicted in Figure 9. As shown in Figure 9, the profile of the free surface will change with time. Meanwhile, the irregular computational domain will vary at each time step as well. Therefore, the MFS, one kind of meshless method, has advantages in dealing with such problems.



(a) Calculation results under the first pseudo-boundary geometry



(b) Calculation results under the second pseudo-boundary geometry



(c) Calculation results under the third pseudo-boundary geometry

Figure 8. Distribution of source and boundary nodes using lsqnonlin at  $t = 14.0$  s.

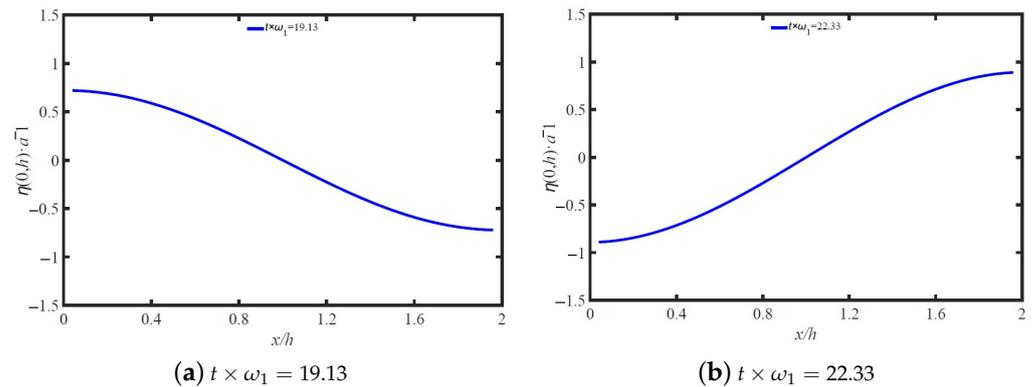


Figure 9. Profiles of the free surfaces at some specific time.

4.3. Horizontally Excited NWT

4.3.1. Validation of Numerical Scheme

In the simulation of horizontally excited NWT, the initial velocity potential condition was  $\phi(x, z)|_{t=0} = 0$ , and  $ZT'' = 0$ . The tank oscillated as follows:

$$X_T''(t) = -a_h \omega_h^2 \sin(\omega_h t), \tag{30}$$

where  $a_h = gk_h / \omega_h^2$  is the amplitude and  $\omega_h$  is the horizontal radial acceleration. For the simulation, four sets of cases were set up based on the different relationships between  $\omega_h$  and  $\omega_1$ , which were  $\omega_h = 0.7\omega_1$ ,  $\omega_h = 1.0\omega_1$ ,  $\omega_h = 1.1\omega_1$ , and  $\omega_h = 1.3\omega_1$ . The results were compared with those of Frandsen [9] and Faltinsen [35]. The time step size was selected as 0.0025 s, and the allocation method refers to Case 3 in Table 2.

Figure 10 displays the evolutionary profiles of the left-end node of the free surface under different oscillation frequencies. When  $\omega_h$  was closer to  $\omega_1$ , the amplitude of the free surface was larger. When the horizontal forced oscillation frequency  $\omega_h$  was equal to the natural frequency  $\omega_1$ , the sloshing amplitude of the free surface would be higher, implying regularity (Figure 10b). Although there was no obvious continuous resonance, a “first-order beat” was observed (Figure 10c). Under other oscillation frequencies, irregular evolutionary profiles were observed (Figure 10a,b).

When resonance occurred, the sloshing amplitude of the free surface increased sharply. When using the traditional MFS collocation method, the accuracy and reliability of the calculation were affected. This study utilized the adaptive allocation method and yielded comparable results based on lsqnonlin.

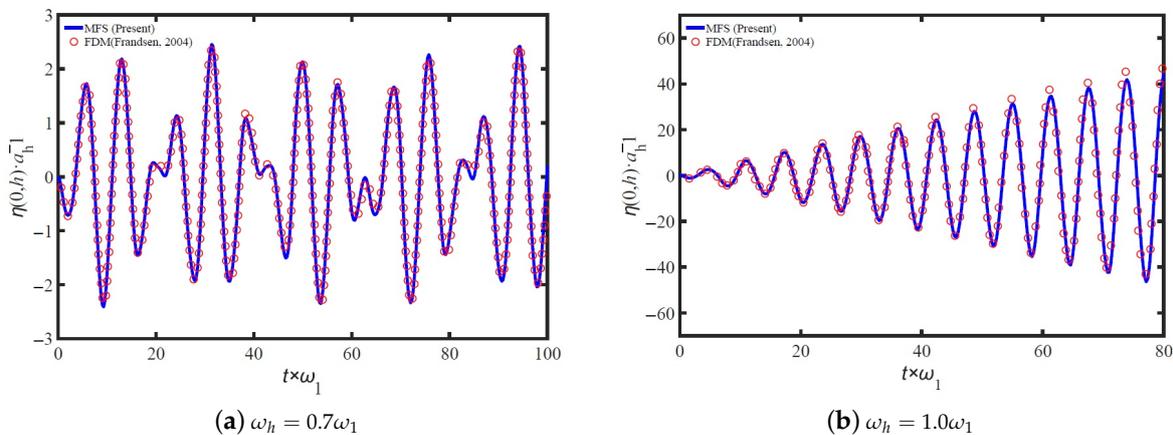


Figure 10. Cont.

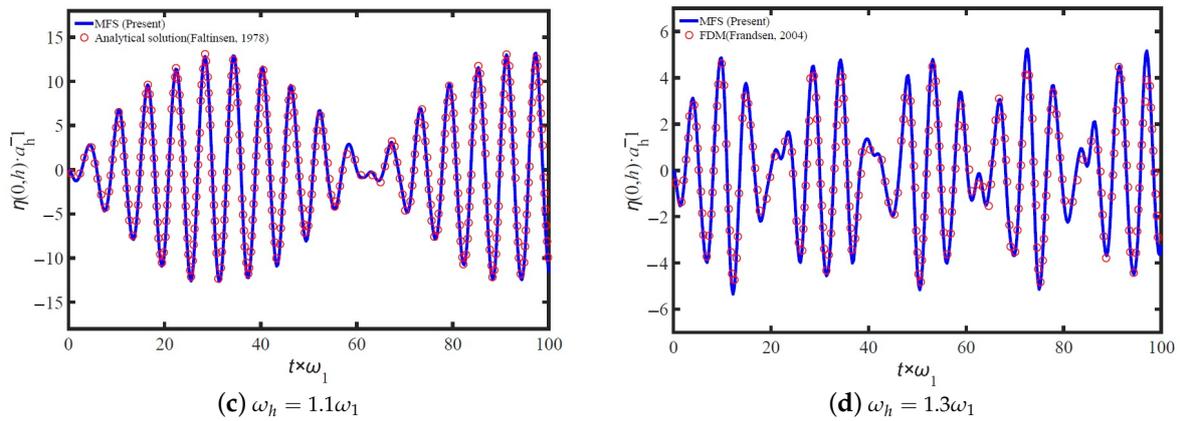


Figure 10. Evolutionary profiles of the left-end node of the free surface.

4.3.2. Distribution of Source Nodes after Iterations

The distribution of the source nodes after the iterations when  $\omega_h = 1.0\omega_1$  is displayed in Figure 10. In addition, the distribution of the boundary nodes and source nodes at  $t = 8.00\text{ s}$ ,  $t = 10.00\text{ s}$ ,  $t = 12.00\text{ s}$ , and  $t = 14.00\text{ s}$  is presented in Figure 11, respectively. These figures illustrate the changes in the boundary nodes and source nodes over time and allow for a clear visualization of the adaptive reconfiguration process of source nodes at the pseudo-boundary. This demonstrates the superiority of the MFS proposed in this study in terms of computational stability.

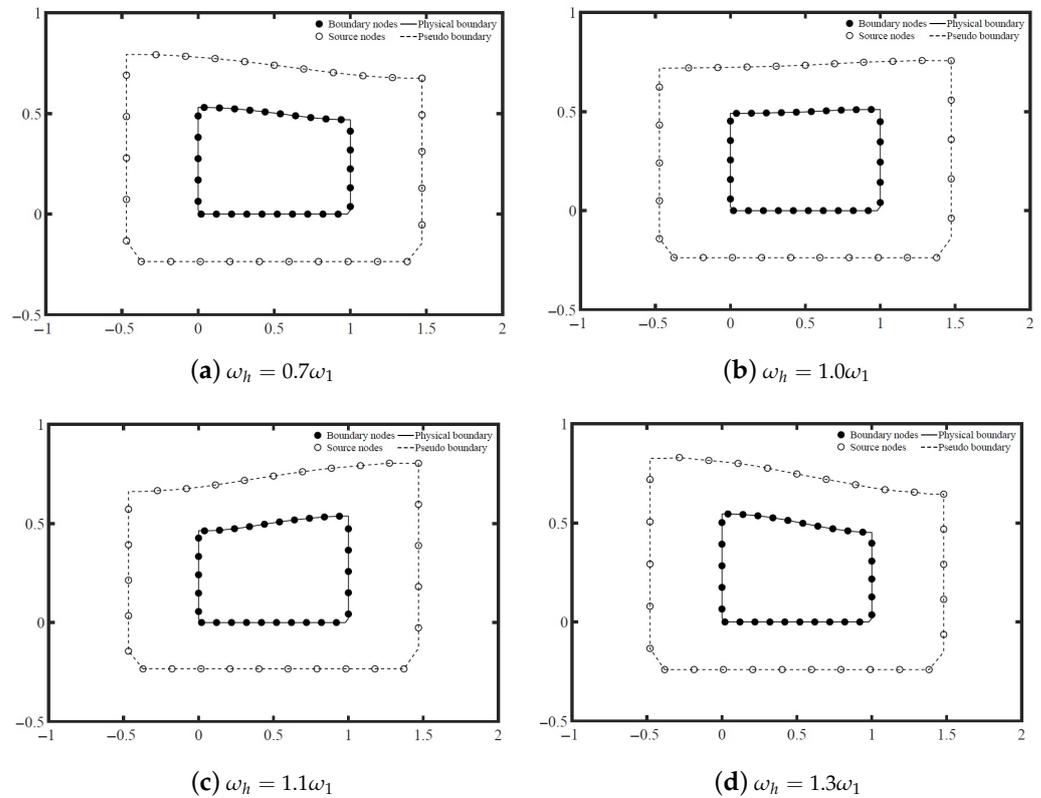


Figure 11. Distribution of source nodes after iterations using lsqnonlin.

The geometry of the pseudo-boundary evolved over time, which improved the stability and accuracy of the calculation. The parameter  $\sigma$ , which represents the normal distance between the physical and pseudo-boundaries, was updated adaptively. As a result, the

combination of the MFS and  $\text{lsqnonlin}$  was able to effectively analyze dynamic-boundary problems related to sloshing with good efficiency and accuracy.

## 5. Conclusions

In this study, a meshless method was proposed to simulate sloshing in a 2D rectangular NWT. The second-order Runge–Kutta method and the MFS, based on moving pseudo-boundaries, were implemented for temporal and spatial discretization, respectively. The second-order Runge–Kutta method guarantees the accuracy and stability of dynamic and kinematic free surface boundary updates. The MFS, a meshless method with easy programming and high computational accuracy traits, was responsible for dealing with the Laplace equation at each time step. This study treated the position of the source nodes as an unknown variable to be solved simultaneously with the weighting coefficients, which avoided the need for fixing the position of the source nodes. This enabled adaptive calculations and improved the stability of the calculations and the numerical methods. The solution of the Laplace equation can be obtained easily by combining weighting coefficients and the fundamental solution. Meanwhile, the parameters  $\beta$  and  $\sigma$  were adaptively updated using  $\text{lsqnonlin}$ .

Three numerical simulations were analyzed to verify the accuracy and stability of the proposed numerical method. The numerical results were compared to other works. Through the convergence study of the number of nodes and time steps, the MFS based on the least-squares algorithm reported good computational stability and accuracy with fewer collocation nodes and a larger time step. Besides that, the geometry of the pseudo-boundary could affect the computational efficiency of the calculation. Likewise, the computational efficiency of the circular pseudo-boundary was approximately twice that of the boundary-dependent type. The main contribution of this study was expanding the application of the MFS in engineering by integrating it with the optimal configuration problem of pseudo-boundaries to solve practical engineering problems. The research in this paper showed that the MFS we proposed has great potential to be applied to other related marine engineering problems.

In this study, the time increment was assumed to be a constant number. Moreover, the arrangement of the source nodes was still relatively fixed in this study, and only the normal distance was adjusted. In principle, the source nodes can be arranged arbitrarily outside the computational domain, prompting future research to examine the effect of source node positioning on computational efficiency. The findings from this study indicated that the MFS can accurately calculate the nonlinear free surface in 2D sloshing and can be applied in other related studies.

**Author Contributions:** Data curation, methodology, resources, software, validation, and writing—original draft preparation: C.W.; formal analysis, investigation, and visualization: Y.Z.; conceptualization, supervision, and writing—review and editing: J.H.; methodology, resources, and software: C.-M.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Guangdong provincial special fund for promoting high quality economic development (Grant No. GDNRC[2021]56); the National Natural Science Foundation of China (Grant No. 52171346); and The Young Innovative Talents Grants Programme of Guangdong Province (Grant No. 2022KQNCX024).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. This submitted manuscript has been approved by all authors for publication. We would like to declare that the work described is original research that has not been published previously and is not under consideration for publication elsewhere, in whole or in part. The authors listed have approved the manuscript that is enclosed.

## Nomenclature

$\eta$	free surface elevation (m)
$h$	height of still water (m)
$b$	width of the tank (m)
$a$	amplitude of the initial wave profile
$g$	acceleration due to gravity ( $\text{m/s}^2$ )
$k_1$	number of waves
$Z_T''$	acceleration in the vertical direction ( $\text{m/s}^2$ )
$X_T''$	acceleration in the horizontal direction ( $\text{m/s}^2$ )

## References

1. Gotoh, H.; Okayasu, A.; Watanabe, Y. *Computational Wave Dynamics*; World Scientific: Singapore, 2013.
2. Ma, Q. *Advances in Numerical Simulation of Nonlinear Water Waves*; World Scientific: Singapore, 2010.
3. Faltinsen, O.M.; Timokha, A.N. On sloshing modes in a circular tank. *J. Fluid Mech.* **2012**, *695*, 467–477. [[CrossRef](#)]
4. Wang, C.Z.; Khoo, B.C. Finite element analysis of two-dimensional nonlinear sloshing problems in random excitations. *Ocean Eng.* **2005**, *32*, 107–133. [[CrossRef](#)]
5. Mitra, S.; Upadhyay, P.P.; Sinhamahapatra, K.P. SLOSH dynamics of inviscid fluids in two-dimensional tanks of various geometry using finite element method. *Int. J. Numer. Methods Fluids* **2010**, *56*, 1625–1651. [[CrossRef](#)]
6. Gómez-GoI, J.; Garrido-Mendoza, C.A.; Cercós, J.; González, L. Two phase analysis of sloshing in a rectangular container with Volume of Fluid (VOF) methods. *Ocean Eng.* **2013**, *73*, 208–212. [[CrossRef](#)]
7. Tang, Y.Y.; Liu, Y.D.; Chen, C.; Chen, Z.; Zheng, M.M. Numerical study of liquid sloshing in 3D LNG tanks with unequal baffle height allocation schemes. *Ocean Eng.* **2021**, *234*, 109181. [[CrossRef](#)]
8. Hamano, K.; Murashige, S.; Hayami, K. Boundary element simulation of large amplitude standing waves in vessels. *Eng. Anal. Bound. Elem.* **2003**, *27*, 565–574. [[CrossRef](#)]
9. Frandsen, J.B. Sloshing motions in excited tanks. *J. Comput. Phys.* **2004**, *196*, 53–87. [[CrossRef](#)]
10. Wang, Z.H.; Jiang, S.C.; Bai, W.; Li, J.X. Liquid sloshing in a baffled rectangular tank under irregular excitations. *Ocean Eng.* **2023**, *278*, 114472.
11. Wu, N.J.; Chang, K.A. Simulation of free-surface waves in liquid sloshing using a domain-type meshless method. *Int. J. Numer. Methods Fluids* **2011**, *67*, 269–288. [[CrossRef](#)]
12. Chen, Y.W.; Liu, C.S.; Chang, C.M.; Chang, J.R. Applications of the modified Trefftz method to the simulation of sloshing behaviours. *Eng. Anal. Bound. Elem.* **2010**, *34*, 581–598. [[CrossRef](#)]
13. Zhang, T.; Ren, Y.F.; Fan, C.M.; Li, P.W. Simulation of two-dimensional sloshing phenomenon by generalized finite difference method. *Eng. Anal. Bound. Elem.* **2016**, *63*, 82–91. [[CrossRef](#)]
14. Pal, P. Slosh dynamics of liquid-filled rigid containers: Two-dimensional meshless local Petrov-Galerkin approach. *J. Eng. Mech.* **2012**, *138*, 567–581. [[CrossRef](#)]
15. Ren, Y.; Khayyer, A.; Lin, P.; Hu, X. Numerical modeling of sloshing flow interaction with an elastic baffle using SPHInXsys. *Ocean Eng.* **2023**, *267*, 113110.
16. Zhang, C.; Wang, L.; Xu, M. Study on the Damping Effect and Mechanism of Vertical Slotted Screens Based on the BM-MPS Method. *J. Mar. Sci. Eng.* **2023**, *11*, 1270.
17. Li, D.; Xiao, H.; Jin, Y.C. Design optimization of sloshing tank using weakly compressible mesh free model. *Ocean Eng.* **2023**, *284*, 115218.
18. Gholamipour, M.; Ghiasi, M. Numerical analysis of fully non-linear sloshing waves in an arbitrary shape tank by meshless method. *Eng. Anal. Bound. Elem.* **2022**, *144*, 366–379.
19. Luo, X.; Kareem, A.; Yu, L.; Yoo, S. A machine learning-based characterization framework for parametric representation of liquid sloshing. *Results Eng.* **2023**, *18*, 101148.
20. Kupradze, V.D.; Aleksidze, M.A. The method of functional equations for the approximate solution of certain boundary value problems. *USSR Comput. Math. Math. Phys.* **1964**, *4*, 82–126. [[CrossRef](#)]
21. Fairweather, G.; Karageorghis, A. The method of fundamental solutions for elliptic boundary value problems. *Adv. Comput. Math.* **1998**, *9*, 69–95. [[CrossRef](#)]
22. Kythe, P. *Fundamental Solutions for Differential Operators and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1996.
23. Young, D.; Hu, S.; Chen, C.; Fan, C.; Murugesan, K. Analysis of elliptical waveguides by the method of fundamental solutions. *Microw. Opt. Technol. Lett.* **2005**, *44*, 552–558. [[CrossRef](#)]
24. Young, D.; Chiu, C.; Fan, C.; Tsai, C.; Lin, Y. Method of fundamental solutions for multidimensional Stokes equations by the dual-potential formulation. *Eur. J. Mech.-B/Fluids* **2006**, *25*, 877–893. [[CrossRef](#)]
25. Mathon, R.; Johnston, R.L. The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM J. Numer. Anal.* **1977**, *14*, 638–650. [[CrossRef](#)]
26. Bogomolny, A. Fundamental solutions method for elliptic boundary value problems. *SIAM J. Numer. Anal.* **1985**, *22*, 644–669. [[CrossRef](#)]

27. Wang, F.; Liu, C.S.; Qu, W. Optimal sources in the MFS by minimizing a new merit function: Energy gap functional. *Appl. Math. Lett.* **2018**, *86*, 229–235. [[CrossRef](#)]
28. Hematiyan, M.; Haghighi, A.; Khosravifard, A. A two-constraint method for appropriate determination of the configuration of source and collocation points in the method of fundamental solutions for 2D Laplace equation. *Adv. Appl. Math. Mech.* **2017**. [[CrossRef](#)]
29. Gorzelańczyk, P.; Kołodziej, J.A. Some remarks concerning the shape of the source contour with application of the method of fundamental solutions to elastic torsion of prismatic rods. *Eng. Anal. Bound. Elem.* **2008**, *32*, 64–75. [[CrossRef](#)]
30. Nishimura, R.; Nishimori, K. Arrangement of fictitious charges and contour points in charge simulation method for electrodes with 3-D asymmetrical structure by immune algorithm. *J. Electrostat.* **2005**, *63*, 743–748. [[CrossRef](#)]
31. Jopek, H.; Kołodziej, J. Application of genetic algorithms for optimal positions of source points in method of fundamental solutions. *Comput. Assist. Mech. Eng. Sci.* **2008**, *15*, 215–224.
32. Grabski, J.K.; Karageorghis, A. Moving pseudo-boundary method of fundamental solutions for nonlinear potential problems. *Eng. Anal. Bound. Elem.* **2019**, *105*, 78–86. [[CrossRef](#)]
33. Lin, B.H.; Chen, B.F.; Tsai, C.C. Method of fundamental solutions on simulating sloshing liquids in a 2D tank. *Comput. Math. Appl.* **2021**, *88*, 52–69. [[CrossRef](#)]
34. Chen, B.F.; Chiang, H.W. Complete 2D and fully nonlinear analysis of ideal fluid in tanks. *J. Eng. Mech.* **1999**, *125*, 70–78. [[CrossRef](#)]
35. Faltinsen, O.M. A nonlinear theory of sloshing in rectangular tanks. *J. Ship Res.* **1974**, *18*, 224–241. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.