INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:709–733
Published online 7 September 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.778

An implicit three-dimensional fully non-hydrostatic model for free-surface flows

Hengliang Yuan and Chin H. Wu*,†

Department of Civil and Environmental Engineering, University of Wisconsin, Madison, U.S.A.

SUMMARY

An implicit method is developed for solving the complete three-dimensional (3D) Navier–Stokes equations. The algorithm is based upon a staggered finite difference Crank-Nicholson scheme on a Cartesian grid. A new top-layer pressure treatment and a partial cell bottom treatment are introduced so that the 3D model is fully non-hydrostatic and is free of any hydrostatic assumption. A domain decomposition method is used to segregate the resulting 3D matrix system into a series of two-dimensional vertical plane problems, for each of which a block tri-diagonal system can be directly solved for the unknown horizontal velocity. Numerical tests including linear standing waves, nonlinear sloshing motions, and progressive wave interactions with uneven bottoms are performed. It is found that the model is capable to simulate accurately a range of free-surface flow problems using a very small number of vertical layers (e.g. two–four layers). The developed model is second-order accuracy in time and space and is unconditionally stable; and it can be effectively used to model 3D surface wave motions.

KEY WORDS: implicit; three-dimensional; fully non-hydrostatic; top-layer pressure treatment; free-surface flows

1. INTRODUCTION

Numerical modeling of three-dimensional (3D) free-surface flows has become computationally affordable nowadays. Hydrostatic models based upon the 3D Navier–Stokes equations (NSE) are widely applied in riverine, estuarine, and coastal flow simulations [1–4]. However, it is well known that the hydrostatic pressure assumption is not valid in the cases of short surface waves, stratifications induced by strong density gradients, and flows over abruptly changed

*Correspondence to: Chin H. Wu, Department of Civil and Environmental Engineering, 1269D Engineering Hall, University of Wisconsin, Madison, U.S.A.
†E-mail: chinwu@engr.wisc.edu

Contract/grant sponsor: National Science Foundation
Contract/grant sponsor: Wisconsin Alumni Research Foundation; contract/grant number: 040061

Received 23 February 2004
Revised 13 July 2004

bottom topographies where the effects of vertical acceleration become important. In addition, it is recognized that 3D hydrostatic models with open boundary conditions are ill posed [5, 6].

In recent years efforts have been focused on developing non-hydrostatic models. For example, Casulli [7] proposed a semi-implicit, fractional step method by which intermediate free-surface elevations and velocities are calculated during the hydrostatic step, and the non-hydrostatic pressure correction is solved from the pressure Poisson equation (PPE). Following the semi-implicit procedure, Mahadevan *et al.* [6] used a control volume method to develop a meso-scale non-hydrostatic ocean model in a sigma co-ordinate. Chen [8] further introduced a double predictor-corrector semi-implicit procedure to include the non-hydrostatic effects in the final free-surface elevation. In a different approach to the semi-implicit framework, Chorin [9] suggested an explicit projection method that advances the projected intermediate velocities by solving advection and diffusion terms explicitly at each time step, and corrects the projected velocities after solving the PPE. Li and Fleming [10] further used the explicit McCormack scheme to discretize the momentum equations in a sigma co-ordinate and solved the PPE using a multigrid method. In their model, the total pressure is decomposed into hydrostatic and non-hydrostatic components. Lin and Li [11] used the combination of a backward characteristic method and the Lax–Wendroff scheme but adopted a non-staggered arrangement of variables in the sigma co-ordinate without decomposing the total pressure. Recently a fully implicit algorithm for solving the two-dimensional vertical plane (2DV) NSE with free-surface boundary condition was proposed by Namin *et al.* [12]. Instead of solving the PPE, the implicit algorithm forms a block tri-diagonal system with the unknown horizontal velocity so that a direct matrix solver can be applied without any iteration. Yuan and Wu [13] further extended the implicit NSE model to a sigma co-ordinate.

For free-surface flows simulations, one of the main issues is to numerically express the moving boundary (air–water interface). Several methods have been successfully incorporated in the NSE, e.g. the marker and cell (MAC) method [14], the volume of fluid (VOF) method [15], and the arbitrary Lagrangian–Eulerian (ALE) method [16]. Generally speaking, these methods are capable of dealing with complicated free surfaces (e.g. breaking waves), but their applications are limited by high computational expense and strict stability requirement. In geophysical and environmental flow applications, the free-surface elevation can be reasonably treated as a single-valued function of horizontal position. Therefore, the free-surface elevation can be calculated using either the free-surface equation or kinematic free-surface boundary condition. While non-hydrostatic models of this kind cannot deal with steep-fronted or overturning free surfaces, they effectively track the free surface motions at relatively small computational cost, and have been widely applied [6–8, 10–12]. In addition, these models are developed under a staggered grid framework [6–8, 10, 12]. As a result, applying the pressure boundary condition at the free surface raises the difficulty in treating the cell-centered pressure at the top layer. Some models [6, 10, 12, 17] assume a hydrostatic pressure distribution at the top layer along with the whole solution procedure. Other models [7, 8, 18] introduce a corrected hydrostatic relation at the top layer. Consequently a non-zero intermediate non-hydrostatic pressure is first obtained at the top layer, and then used to correct the final free-surface elevation. Nevertheless, the role of the top-layer non-hydrostatic pressure is still not well represented while solving the PPE at each time step. Therefore, as indicated by Stelling and Zijlema [19], a sufficiently large number of vertical grid points, about 10–20, is usually required in models with either a hydrostatic pressure assumption or hydrostatic relation at the top layer for simulating

non-hydrostatic free-surface flows, especially for describing wave dispersion characteristics up to an acceptable level of accuracy.

To address the above issue, Stelling and Zijlema [19] proposed the Keller-box method to replace the staggered grid in the vertical direction, which enables the pressure to be located at the cell faces rather than the cell centers. The pressure boundary condition at the free surface can be exactly assigned to zero without any approximation. Using a very small number of vertical layers (e.g. in the order of 1–3), their model can accurately predicts free-surface flows, suggesting that accurate treatment of the top-layer pressure offers a means of decreasing the vertical grid number for the model. Yuan and Wu [13] proposed an integral method, different to the Keller-box scheme, to remove the top-layer hydrostatic assumption under a staggered grid framework. In their model, the vertical momentum equation is integrated from the center of the top layer to the moving free surface. The top-layer pressure is implicitly expressed as a function of the free-surface elevation (hydrostatic pressure component) and the vertical acceleration (non-hydrostatic pressure component). The model results show that the inclusion of top-layer non-hydrostatic pressure significantly reduces phase errors for simulating dispersive waves.

To simulate 3D non-hydrostatic flows, the computational cost is a critical issue due to a large size of the resulting matrix system. For models involving a PPE, iterative algorithms (e.g. conjugate and/or biconjugate gradient methods) are commonly used to solve the multi-diagonal matrix system (e.g. seven-diagonal system in the Cartesian model [7] and nineteen-diagonal system in the sigma model [11]). Many efforts have been paid to save computational cost by using multi-grid methods to reduce iterations [10] or applying the Keller-box scheme to decrease the number of vertical layers in the computational domain [19]. On the other hand, the extension of the 2DV implicit method proposed by Namin *et al.* [12] and Yuan and Wu [13] to 3D applications becomes challenging since the resulting matrix system for horizontal velocities is neither diagonal nor block-banded. Directly solving the matrix is computationally expensive. A new algorithm with an affordable computational cost is highly desired.

In this paper, an implicit algorithm for solving the complete 3D NSE in the free-surface flows is presented. To address the computational cost, a new top-layer pressure treatment on a Cartesian co-ordinate that accounts for the non-hydrostatic pressure is developed. The model is discretized following the Crank–Nicolson scheme (CNS) with a second order accuracy in both time and space. A partial cell method [20] is adopted to represent the bottom topography in a Cartesian framework. To solve the resulting 3D matrix system, we decompose the 3D system into a series of 2DV problems, for each of which a block tri-diagonal system is arranged with the unknown horizontal velocity. The algorithm simultaneously solves flow-field components (i.e. the pressure, velocity, and free-surface elevation) at each time step. Since the domain decomposition is identical in two horizontal directions and each 2DV problem is independent, the algorithm is potentially suitable for parallel computations. Section 2 presents the mathematical formulation with boundary conditions. The numerical method is described in Section 3. In Section 4, the model is validated against four examples, including linear 3D standing waves in a closed basin, finite amplitude sloshing motions, wave propagation over a submerged bar, and 3D wave transformation over an elliptic shoal on a sloped bottom. Numerical results are compared to either analytical solutions or experimental data to demonstrate the capability, accuracy, and efficiency of the model.

2. MATHEMATICAL FOMULATION

The governing equations for simulating free-surface flows are the unsteady, incompressible, 3D NSE in a Cartesian co-ordinate (x, y, z) with time t. Figure 1 shows the physical domain bounded by the moving free surface, $z = \eta(x, y, t)$, and the bottom, $z = -h(x, y)$. The corresponding continuity and momentum equations are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{\partial P}{\partial x} + \nu_x\frac{\partial^2 u}{\partial x^2} + \nu_y\frac{\partial^2 u}{\partial y^2} + \nu_z\frac{\partial^2 u}{\partial z^2} \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{\partial P}{\partial y} + \nu_x\frac{\partial^2 v}{\partial x^2} + \nu_y\frac{\partial^2 v}{\partial y^2} + \nu_z\frac{\partial^2 v}{\partial z^2} \tag{3}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{\partial P}{\partial z} + \nu_x\frac{\partial^2 w}{\partial x^2} + \nu_y\frac{\partial^2 w}{\partial y^2} + \nu_z\frac{\partial^2 w}{\partial z^2} - g \tag{4}$$

where $u(x, y, z, t)$, $v(x, y, z, t)$, and $w(x, y, z, t)$ are the velocity components in the horizontal x, y, and vertical z directions, respectively; the normalized pressure P is defined as the pressure divided by the constant water density; g is the gravitational acceleration; and ν_x, ν_y, and ν_z are constant eddy viscosities in the x, y, and z directions, respectively.

Since the bottom surface may not be regular, the kinematic boundary condition at the impermeable bottom is

$$u\frac{\partial h}{\partial x} + v\frac{\partial h}{\partial y} + w = 0 \tag{5}$$

Similarly, the kinematic boundary condition at the moving free surface is

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} = w \tag{6}$$

Integrating the continuity equation (1) over the water depth and applying the Leibniz's rule with the kinematic boundary conditions (5) and (6) gives the free-surface equation

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\int_{-h}^{\eta} u\,\mathrm{d}z + \frac{\partial}{\partial y}\int_{-h}^{\eta} v\,\mathrm{d}z = 0 \tag{7}$$

At the free surface, the continuity of tangential and normal stresses is enforced. In the case of no wind effect, the tangential stresses are zero. A Dirichlet boundary condition for the total pressure is specified at the free surface, i.e. $P = P_a$, in which P_a is the atmospheric pressure (set as zero here). For solid walls, the impermeability condition is specified, i.e. velocity normal to the wall is zero. A no-slip boundary condition is used for viscous flows. If viscosity is set to zero, a null gradient condition for the tangential velocity components is applied.

For wave propagation simulations, proper inflow and outflow boundary conditions are essential. At inflow, the velocity components are specified from either analytical solutions

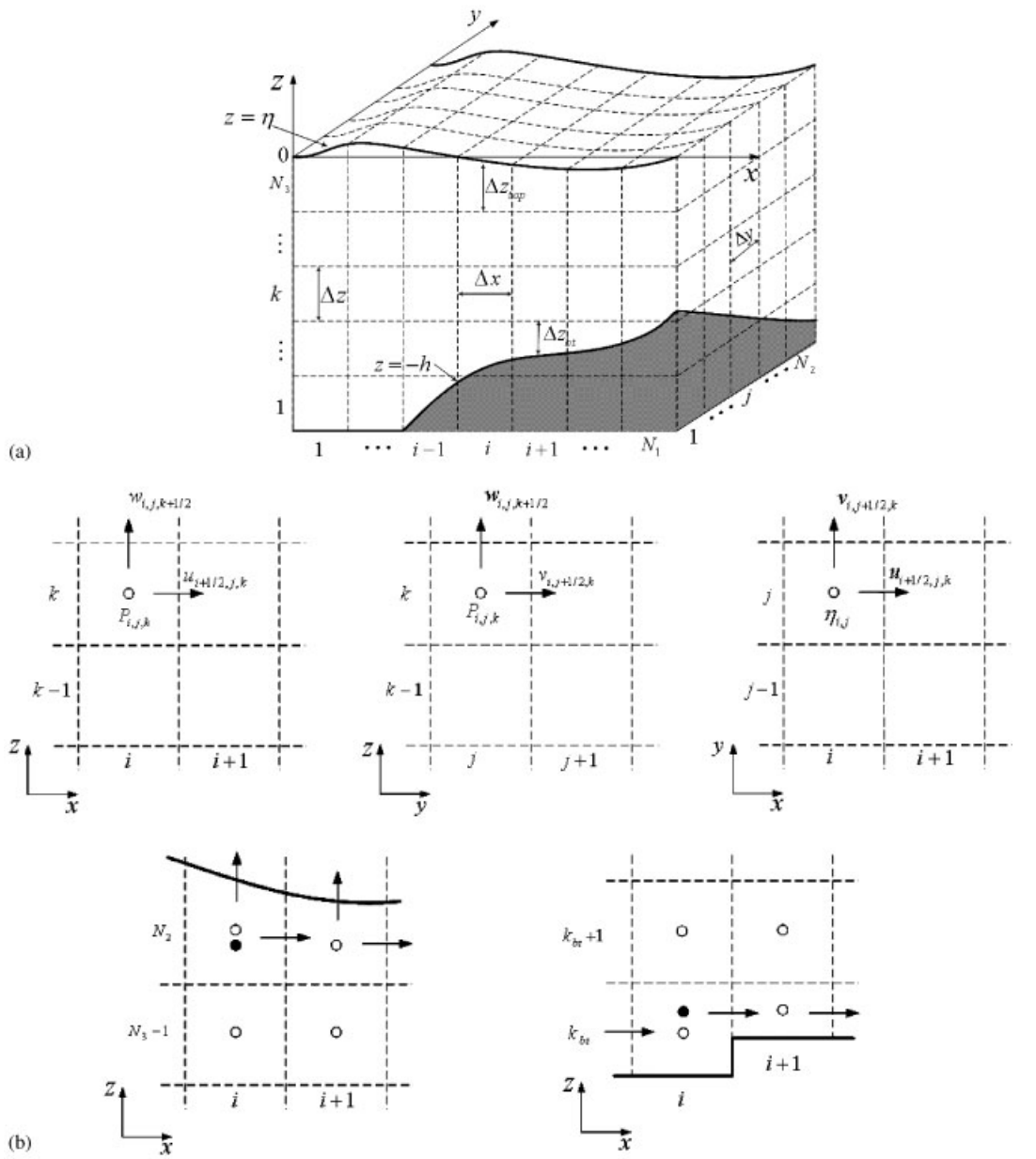

Figure 1. A Cartesian grid system: (a) a sketch of the computational domain; and (b) staggered arrangements of variables in stencils.

or laboratory conditions. At outflow, a combination of a sponge layer technique [14] and a Sommerfeld-type radiation boundary condition is applied to minimize wave reflection. For the sponge layer, the artificial damping terms are added to the right-hand side of horizontal

momentum equations (2) and (3), respectively, i.e.

$$\left[\alpha\left(\frac{x-x_0}{l_x}\right)^2\left(\frac{z_b-z}{z_b-z_f}\right)\right]u, \quad x \geqslant x_0 \tag{8a}$$

$$\left[\alpha\left(\frac{y-y_0}{l_y}\right)^2\left(\frac{z_b-z}{z_b-z_{fs}}\right)\right]v, \quad y \geqslant y_0 \tag{8b}$$

where x_0 and y_0 denote the starting point of the damping zone; l_x and l_y are the length of the sponge layer at the x and y directions, respectively; z_b and z_f represent the bottom and the free surface, respectively; and the damping strength parameter, α, determines the damping rate, being adjustable for particular problems. Outside the sponge layer, i.e. $x < x_0$ and $y < y_0$, the damping term is set to be zero. At the end of the sponge layer, the Sommerfeld-type radiation condition

$$\frac{\partial\phi}{\partial t} + c\cos(\beta)\frac{\partial\phi}{\partial x} + c\sin(\beta)\frac{\partial\phi}{\partial y} = 0 \tag{9}$$

is used further to enhance wave absorption. In Equation (9), β is the angle between the normal direction of the boundary and the direction of the propagating wave; c is the wave celerity; and ϕ can be η, u, v, or w.

3. NUMERICAL METHOD

In this paper, a staggered finite difference grid is chosen to solve the governing equations under a Cartesian framework. The computational domain is discretized by N_1, N_2, and N_3 cells in the x, y, and z directions, and the grid indexes i, j, and k are also introduced, respectively. The cell center is denoted by (i,j,k) and the cell faces are marked by $(i+1/2,j,k)$, $(i,j+1/2,k)$, and $(i,j,k+1/2)$. The pressure and free-surface elevation are defined in the center of the cell, while the velocity components are located at the faces as shown in Figure 1(a) and (b). In the horizontal direction a uniform grid interval, i.e. $\Delta x_i = \Delta x$ and $\Delta y_i = \Delta y$, is used. In the vertical direction, the computational domain is first divided by N_3 layers. For the interior layers, denoted by $k = k_{\mathrm{bt}}+1,\ldots,N_3-1$ (k_{bt} is the index of the bottom layer), the actual vertical cell size is uniform, i.e. $\Delta z_k = \Delta z$. The bottom-layer cell size, $\Delta z_{k_{\mathrm{bt}}}$, can be varied and is defined as the distance between the real bottom and the top of the bottom layer, i.e. $\Delta z_{k_{\mathrm{bt}}} = \Delta z_{k_{\mathrm{bt}}}(x,y)$. The top-layer cell size,

$$\Delta z_{N_3}(x,y,t) = \Delta z + \eta(x,y,t) \tag{10}$$

is allowed to vary with respect to both the horizontal locations and time.

From the above discretization the computational domain is vertically bounded by the bottom layer, $k = k_{\mathrm{bt}}$, and the top layer, $k = N_3$. For an irregular bottom, k_{bt} can vary at different horizontal locations and is not necessarily equal to 1. A 'flag' strategy is used, allowing the computational domain to always start from $k = 1$. For the non-fluid cells, i.e. the cells below the bottom cell ($1 \leqslant k < k_{\mathrm{bt}}$), an insulating condition is applied. The k index of the top layer

is always equal to N_3, a constant number with respect to both the horizontal position and time. The rest of the cells are the fluid cells ($k_{\mathrm{bt}} \leqslant k \leqslant N_3$). The 'flag' strategy not only marks the irregular bottom positions, but also effectively keeps track with the size of the resulting linear system. If two layers are chosen in the vertical direction for model simulations, only the bottom layer (with the cell size of $\Delta z_{k_{\mathrm{bt}}}$) and the top layer (with the cell size of Δz_{N_3}) remain.

3.1. General discretization

Using the CNS, a second order trapezoid rule is applied for the temporal discretization of the governing equations. The governing equations (1)–(4) take the following CNS discretized forms:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j,k}^{n+1} + \left(\frac{\partial v}{\partial y}\right)_{i,j,k}^{n+1} + \left(\frac{\partial w}{\partial z}\right)_{i,j,k}^{n+1} = 0 \tag{11}$$

where $i = 1, \ldots, N_1$, $j = 1, \ldots, N_2$, and $k = 1, \ldots, N_3$;

$$\begin{aligned}
&\frac{u_{i+1/2,j,k}^{n+1} - u_{i+1/2,j,k}^{n}}{\Delta t} + \theta\left(u\frac{\partial u}{\partial x}\right)_{i+1/2,j,k}^{n+1} + (1-\theta)\left(u\frac{\partial u}{\partial x}\right)_{i+1/2,j,k}^{n} + \theta\left(v\frac{\partial u}{\partial y}\right)_{i+1/2,j,k}^{n+1} \\
&\quad + (1-\theta)\left(v\frac{\partial u}{\partial y}\right)_{i+1/2,j,k}^{n} + \theta\left(w\frac{\partial u}{\partial z}\right)_{i+1/2,j,k}^{n+1} + (1-\theta)\left(w\frac{\partial u}{\partial z}\right)_{i+1/2,j,k}^{n} \\
&\quad = -\theta\left(\frac{\partial P}{\partial x}\right)_{i+1/2,j,k}^{n+1} - (1-\theta)\left(\frac{\partial P}{\partial x}\right)_{i+1/2,j,k}^{n} + \theta\left(v_x\frac{\partial^2 u}{\partial x^2} + v_y\frac{\partial^2 u}{\partial y^2} + v_z\frac{\partial^2 u}{\partial z^2}\right)_{i+1/2,j,k}^{n+1} \\
&\qquad + (1-\theta)\left(v_x\frac{\partial^2 u}{\partial x^2} + v_y\frac{\partial^2 u}{\partial y^2} + v_z\frac{\partial^2 u}{\partial z^2}\right)_{i+1/2,j,k}^{n}
\end{aligned} \tag{12}$$

where $i = 1, \ldots, N_1 - 1$, $j = 1, \ldots, N_2$, and $k = 1, \ldots, N_3$;

$$\begin{aligned}
&\frac{v_{i,j+1/2,k}^{n+1} - v_{i,j+1/2,k}^{n}}{\Delta t} + \theta\left(u\frac{\partial v}{\partial x}\right)_{i,j+1/2,k}^{n+1} + (1-\theta)\left(u\frac{\partial v}{\partial x}\right)_{i,j+1/2,k}^{n} + \theta\left(v\frac{\partial v}{\partial y}\right)_{i,j+1/2,k}^{n+1} \\
&\quad + (1-\theta)\left(v\frac{\partial v}{\partial y}\right)_{i,j+1/2,k}^{n} + \theta\left(w\frac{\partial v}{\partial z}\right)_{i,j+1/2,k}^{n+1} + (1-\theta)\left(w\frac{\partial v}{\partial z}\right)_{i,j+1/2,k}^{n} \\
&\quad = -\theta\left(\frac{\partial P}{\partial y}\right)_{i,j+1/2,k}^{n+1} - (1-\theta)\left(\frac{\partial P}{\partial y}\right)_{i,j+1/2,k}^{n} + \theta\left(v_x\frac{\partial^2 v}{\partial x^2} + v_y\frac{\partial^2 v}{\partial y^2} + v_z\frac{\partial^2 v}{\partial z^2}\right)_{i,j+1/2,k}^{n+1} \\
&\qquad + (1-\theta)\left(v_x\frac{\partial^2 v}{\partial x^2} + v_y\frac{\partial^2 v}{\partial y^2} + v_z\frac{\partial^2 v}{\partial z^2}\right)_{i,j+1/2,k}^{n}
\end{aligned} \tag{13}$$

where $i=1,\ldots,N_1$, $j=1,\ldots,N_2-1$, andr $k=1,\ldots,N_3$; and

$$\frac{w^{n+1}_{i,j,k+1/2}-w^{n}_{i,j,k+1/2}}{\Delta t}+\theta\left(u\frac{\partial w}{\partial x}\right)^{n+1}_{i,j,k+1/2}+(1-\theta)\left(u\frac{\partial w}{\partial x}\right)^{n}_{i,j,k+1/2}+\theta\left(v\frac{\partial w}{\partial y}\right)^{n+1}_{i,j,k+1/2}$$
$$+(1-\theta)\left(v\frac{\partial w}{\partial y}\right)^{n}_{i,j,k+1/2}+\theta\left(w\frac{\partial w}{\partial z}\right)^{n+1}_{i,j,k+1/2}+(1-\theta)\left(w\frac{\partial w}{\partial z}\right)^{n}_{i,j,k+1/2}$$
$$=-\theta\left(\frac{\partial P}{\partial z}\right)^{n+1}_{i,j,k+1/2}-(1-\theta)\left(\frac{\partial P}{\partial z}\right)^{n}_{i,j,k+1/2}+\theta\left(v_x\frac{\partial^2 w}{\partial x^2}+v_y\frac{\partial^2 w}{\partial y^2}+v_z\frac{\partial^2 w}{\partial z^2}\right)^{n+1}_{i,j,k+1/2}$$
$$+(1-\theta)\left(v_x\frac{\partial^2 w}{\partial x^2}+v_y\frac{\partial^2 w}{\partial y^2}+v_z\frac{\partial^2 w}{\partial z^2}\right)^{n}_{i,j,k+1/2}-g \tag{14}$$

where $i=1,\ldots,N_1$, $j=1,\ldots,N_2$, and $k=1,\ldots,N_3-1$. Similarly, the free-surface equation (7) is discretized as

$$\frac{\eta^{n+1}_{i,j}-\eta^{n}_{i,j}}{\Delta t}+\theta\left[\frac{\partial}{\partial x}\left(\int_{-h}^{\eta}u\,\mathrm{d}z\right)\right]^{n+1}_{i,j,k}+(1-\theta)\left[\frac{\partial}{\partial x}\left(\int_{-h}^{\eta}u\,\mathrm{d}z\right)\right]^{n}_{i,j,k}$$
$$+\theta\left[\frac{\partial}{\partial x}\left(\int_{-h}^{\eta}v\,\mathrm{d}z\right)\right]^{n+1}_{i,j,k}+(1-\theta)\left[\frac{\partial}{\partial x}\left(\int_{-h}^{\eta}v\,\mathrm{d}z\right)\right]^{n}_{i,j,k}=0 \tag{15}$$

where $i=1,\ldots,N_1$, $j=1,\ldots,N_2$. In Equations (11)–(15), Δt is the time step; the superscript n represents the nth time step; and the implicit weighting factor θ is set to be 0.5 for the CNS. The non-linear advection terms in Equations (12)–(14) are linearized following the procedure of Reference [21] to achieve a second-order temporal accuracy, e.g.

$$\left(v\frac{\partial u}{\partial y}\right)^{n+1}_{i+1/2,j,k}=\left[v^{n+1}\left(\frac{\partial u}{\partial y}\right)^{n}+v^{n}\left(\frac{\partial u}{\partial y}\right)^{n+1}-v^{n}\left(\frac{\partial u}{\partial y}\right)^{n}\right]_{i+1/2,j,k} \tag{16}$$

Other non-linear advection terms share the same linearization procedure as Equation (16). A central difference scheme is used for all spatial discretizations, e.g.

$$\left[v^{n+1}\left(\frac{\partial u}{\partial y}\right)^{n}\right]_{i+1/2,j,k}=v^{n+1}_{i+1/2,j,k}\,\frac{u^{n}_{i+1/2,j+1/2,k}-u^{n}_{i+1/2,j-1/2,k}}{\Delta y} \tag{17}$$

To obtain a unique solution of the discretized governing equations, different boundary conditions are needed. Special numerical treatments are also required to implement the boundary conditions at both the irregular bottom layer, $k=k_{\mathrm{bt}}$, and the top-layer, $k=N_3$. Details of the treatments are discussed in the following sections.

3.2. Bottom treatment

To simulate flows over irregular bottoms, both a z-level co-ordinate and a sigma co-ordinate are commonly used. The sigma-co-ordinate models [1, 6] are capable to follow complicated topographies but issues of large pressure gradient errors and increased numerical diffusions in steep slope regions need to be carefully addressed [22, 23]. The treatment of wetting and drying in shallow water areas using the sigma transformation is also challenging. In contrast, the z-level models alleviate these issues, and allow a varying number of grid points in the vertical direction depending on the water depth [20, 24]. As a result, a partial cell staircase shape and a piece-wise linear treatment [24, 25] have been successfully applied to hydrostatic models. Results show that both methods can simulate accurately the flow over irregular bathymetries. In addition, the partial cell method can be readily implemented in most existing models.

In this paper, we use a partial cell approach [20] to delineate bathymetries. This method represents topographies without the need of unduly high vertical resolutions. Figure 1(b) shows an example of a partial cell in the x–z plane. The kinematic boundary condition of Equation (5) therefore results in a zero vertical velocity at the impermeable bottom. Note that the thickness of each bottom cell can be different. The grid points within the bottom layer are not necessarily at the same z level. A linear interpolation is used to calculate the horizontal gradients (horizontal gradients of pressure and/or velocities) at the position marked as a closed circle. The pressure gradient in the horizontal x direction involving bottom cell is discretized as

$$\left(\frac{\partial P}{\partial x}\right)_{i+1/2,j,k} = \begin{cases} \dfrac{P_{i+1,j,k} - P'_{i,j,k}}{\Delta x} & (\Delta z_{i,j,k} \geqslant \Delta z_{i+1,j,k}) \\ \dfrac{P'_{i+1,j,k} - P_{i,j,k}}{\Delta x} & (\Delta z_{i,j,k} < \Delta z_{i+1,j,k}) \end{cases} \tag{18}$$

where k is the k index of the bottom cells, i.e. $k = k_{\mathrm{bt}}(i,j)$ or $k = k_{\mathrm{bt}}(i+1,j)$; and P'_k is obtained from a linear interpolation of P_k and P_{k+1}. For the $y-z$ plane with the pressure gradient along the y direction, a similar procedure is applied and not shown here.

3.3. Top-layer pressure treatment

The issue of using of the hydrostatic assumption at the top layer under a staggered grid framework has been recognized [7, 18, 19]. In this paper, we further extend the non-hydrostatic top-layer treatment method proposed by Yuan and Wu [13] to the Cartesian co-ordinate. The method is to explicitly express the top-layer pressure by the hydrostatic pressure component and vertical acceleration. To achieve this purpose, the vertical momentum equation (4) is integrated from the centre of top layer, $z = z^*$, to the free surface, $z = \eta$. Using the Leibniz's rule, specifying the free-surface pressure condition ($P = P_a = 0$), and applying the kinematic free-surface boundary condition (6) give

$$\begin{aligned} P_{z=z^*} = g[\eta - z^*] + \frac{\partial}{\partial t}\int_{z^*}^{\eta} w\,\mathrm{d}z + \frac{\partial}{\partial x}\int_{z^*}^{\eta} uw\,\mathrm{d}z + \frac{\partial}{\partial y}\int_{z^*}^{\eta} vw\,\mathrm{d}z - w^2|_{z^*} \\ + w|_{z^*}\frac{\partial z^*}{\partial t} + uw|_{z^*}\frac{\partial z^*}{\partial x} + vw|_{z^*}\frac{\partial z^*}{\partial y} \end{aligned} \tag{19}$$

Since z^* is a function of the moving free-surface elevation, i.e. $z^* = (\eta - \Delta z)/2$,

$$\frac{\partial z^*}{\partial t} = \frac{1}{2}\frac{\partial \eta}{\partial t}, \quad \frac{\partial z^*}{\partial x} = \frac{1}{2}\frac{\partial \eta}{\partial x}, \quad \text{and} \quad \frac{\partial z^*}{\partial y} = \frac{1}{2}\frac{\partial \eta}{\partial y}$$

Equation (19) algebraically represents the top-layer pressure in terms of the hydrostatic component, resulting in the free-surface elevation (the first term at the right-hand side), and the non-hydrostatic component, contributed by the vertical acceleration (all remaining terms at the right-hand side). This treatment accounts for the non-hydrostatic pressure at the top layer, consistent with the non-hydrostatic calculations at other layers.

Following the CNS, Equation (19) is approximated by

$$\begin{aligned}\theta P^{n+1}_{i,j,N_3} + (1-\theta)P^n_{i,j,N_3} = {} & \theta g \frac{\Delta z + \eta^{n+1}_{i,j}}{2} + (1-\theta)g\frac{\Delta z + \eta^n_{i,j}}{2} + \frac{\Delta z + \eta^n_{i,j}}{2}\frac{w^{n+1}_{i,j,N_3} - w^n_{i,j,N_3}}{\Delta t} \\ & + w^n_{i,j,N_3}\frac{\eta^{n+1}_{i,j} - \eta^n_{i,j}}{\Delta t} + \frac{\eta^n_{i,j} + \Delta z}{2}\left[\frac{\partial (uw)}{\partial x}\right]^n_{i,j,N_3} + (uw)^n_{i,j,N_3}\left(\frac{\partial \eta}{\partial x}\right)^n_{i,j} \\ & + \frac{\eta^n_{i,j} + \Delta z}{2}\left[\frac{\partial (vw)}{\partial y}\right]^n_{i,j,N_3} + (vw)^n_{i,j,N_3}\left(\frac{\partial \eta}{\partial y}\right)^n_{i,j} - (w^2)^n_{i,j,N_3} \end{aligned} \tag{20}$$

where the implicit weighting factor, θ, is taken as 0.5. The cell size, Δz_{N_3}, is a function of both the horizontal positions and time, so is the centre of the top layer, $z = z^*$. Therefore, special treatments are needed to calculate the horizontal pressure gradients in Equations (12) and (13). Here we employ a similar method to that for the bottom layer cells. For example, the pressure gradient in the horizontal x direction is evaluated as

$$\left(\frac{\partial P}{\partial x}\right)_{i+1/2,j,N_3} = \begin{cases} \dfrac{P_{i+1,j,N_3} - P'_{i,j,N_3}}{\Delta x} & (\Delta z^n_{i,j,N_3} \geqslant \Delta z^n_{i+1,j,N_3}) \\ \dfrac{P'_{i+1,j,N_3} - P_{i,j,N_3}}{\Delta x} & (\Delta z^n_{i,j,N_3} < \Delta z^n_{i+1,j,N_3}) \end{cases} \tag{21}$$

where $\Delta z^n_{N_3}$ represents the cell size of the top layer cells at nth time; and P'_{N_3} is interpolated from P_{N_3} and P_{N_3-1}.

3.4. Description of the algorithm

The above numerical discretization of the governing equations together with the bottom and top-layer treatments has established implicit relations among the unknown variables. The remaining tasks are to eliminate the vertical velocity, w^{n+1}, and pressure, P^{n+1}, so that a system with the unknown horizontal velocities, i.e. u^{n+1} and v^{n+1}, can be obtained. Detailed description of the procedure is given below.

First, using the bottom kinematic boundary condition (5), the continuity equation (11) can be rearranged in the following manner:

$$w^{n+1}_{i,j,k+1/2} = [F_w(u^{n+1}, v^{n+1})]_{i,j,k+1/2} \tag{22}$$

where F_w represents a linear operator. Likewise, the free-surface equation (15) and top-layer pressure equation (20) are expressed as

$$\eta_{i,j}^{n+1} = [F_\eta(u^{n+1}, v^{n+1})]_{i,j} \tag{23}$$

$$P_{i,j,N_3}^{n+1} = [F_{\mathrm{P_{top}}}(w^{n+1}, \eta^{n+1})]_{i,j,N_3} \tag{24}$$

where F_η and $F_{\mathrm{P_{top}}}$ are both linear operators. Substituting Equations (22) and (23) into Equation (24) to eliminate w^{n+1} and η^{n+1} gives the top-layer pressure as a linear function of u^{n+1} and v^{n+1}.

Secondly, for layers below the top one, the vertical momentum equation (14) is written as

$$P_{i,j,k}^{n+1} = P_{i,j,k+1}^{n+1} + [F_{\mathrm{P_{below}}}(u^{n+1}, v^{n+1}, w^{n+1})]_{i,j,k+1/2} \tag{25}$$

where $k = 1, \ldots, N_3 - 1$, and $F_{\mathrm{P_{below}}}$ is also a linear operator. By substituting Equation (22) into Equation (25), the pressure in the kth layer, $P_{i,j,k}^{n+1}$, can be evaluated in terms of the pressure in the layer above the kth layer, $P_{i,j,k+1}^{n+1}$, and a linear function of u^{n+1} and v^{n+1}. Furthermore, using Equation (24) and starting from $k = N_3 - 1$ in Equation (25), $P_{i,j,k}^{n+1}$ is sequentially expressed as a function of the horizontal velocities, i.e.

$$P_{i,j,k}^{n+1} = [F_{\mathrm{P}}(u^{n+1}, v^{n+1})]_{i,j,k} \tag{26}$$

where $k = 1, \ldots, N_3$, and F_{P} represents a linear operator.

Finally, the horizontal momentum equations (12) and (13) are arranged as

$$[F_u(u^{n+1}, v^{n+1}, w^{n+1}, P^{n+1})]_{i+1/2,j,k} = 0 \tag{27a}$$

$$[F_v(u^{n+1}, v^{n+1}, w^{n+1}, P^{n+1})]_{i,j+1/2,k} = 0 \tag{27b}$$

where F_u and F_v both represent linear operators for the momentum equations in the x and y equation, respectively. Substituting Equations (22) and (26) into Equation (27a) and (27b) to eliminate w^{n+1} and P^{n+1} yields the following matrix system:

$$\begin{bmatrix} \mathbf{A_{uu}} & \mathbf{A_{uv}} \\ \mathbf{A_{vu}} & \mathbf{A_{vv}} \end{bmatrix}^n \times \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}^{n+1} = \begin{bmatrix} \mathbf{b_u} \\ \mathbf{b_v} \end{bmatrix}^n \tag{28}$$

where $\mathbf{A}_{\mathbf{uu}}^n$, $\mathbf{A}_{\mathbf{uv}}^n$, $\mathbf{A}_{\mathbf{vu}}^n$, and $\mathbf{A}_{\mathbf{vv}}^n$ are the block coefficient matrices with the dimension of $(N_3N_1N_2) \times (N_3N_1N_2)$; $[\mathbf{u}^{n+1}, \mathbf{v}^{n+1}]^{\mathbf{T}}$ represents the unknown vector of horizontal velocities; and $[\mathbf{b}_{\mathbf{u}}^n, \mathbf{b}_{\mathbf{v}}^n]^{\mathbf{T}}$ is a known vector. Because the matrix is neither diagonal nor block-banded, directly solving the matrix system (28) is expensive [26].

3.5. *Matrix arrangements and the solving method*

A domain decomposition method is used to solve the matrix system (28). The original 3D system is decomposed into a series of 2DV sub-systems by treating certain nonlinear advection terms and diffusion terms in Equations (11)–(15) (e.g. terms involving the y direction in the x–z plane) as explicit intermediate values. As a result, the explicitly treated terms (denoted

by the superscript $n+1$ in Equation (28)) are replaced by intermediate values (denoted by superscript $n+*$), and moved to the right-hand side. After this rearrangement, $\mathbf{A_{uu}}$ and $\mathbf{A_{vv}}$ in Equation (28) become block diagonal matrices, and $\mathbf{A_{uv}}$ and $\mathbf{A_{vu}}$ become null. The rearranged system is rewritten as

$$\left[\begin{array}{ccc|ccc} \mathbf{A_{u_1}} & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & \mathbf{A_{u_{N2}}} & & & \\ \hline & & & \mathbf{A_{v_1}} & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & \mathbf{A_{v_{N1}}} \end{array}\right]^{n+*} \times \left[\begin{array}{c} \mathbf{u_1} \\ \vdots \\ \mathbf{u_{N_2}} \\ \hline \mathbf{v_1} \\ \vdots \\ \mathbf{v_{N_1}} \end{array}\right]^{n+1} = \left[\begin{array}{c} \mathbf{b_{u_1}} \\ \vdots \\ \mathbf{b_{u_{N2}}} \\ \hline \mathbf{b_{v_1}} \\ \vdots \\ \mathbf{b_{v_{N1}}} \end{array}\right]^{n+*} \tag{29}$$

where each $\mathbf{A}_{\mathbf{u_j}}^{n+*}$ is a 2D sub-block matrix, with the dimension of $(N_3N_1)\times(N_3N_1)$, corresponding to the unknown vector, $\mathbf{u}_\mathbf{j}^{n+1}$, with the dimension of $(N_3N_1)\times 1$, in the x–z plane. Similarly, each $\mathbf{A}_{\mathbf{v_i}}^{n+*}$ is a 2D sub-block matrix, with the dimension of $(N_3N_2)\times(N_3N_2)$, corresponding to the unknown vector, $\mathbf{v}_\mathbf{i}^{n+1}$, with the dimension of $(N_3N_2)\times 1$, in the y–z plane. The vectors, $\mathbf{b}_{\mathbf{u_i}}^{n+*}$ and $\mathbf{b}_{\mathbf{v_j}}^{n+*}$, are the known values including the previous time (denoted by the superscript n) and intermediate time, $n+*$ (denoted by the superscript $n+*$).

The matrix system (29) can be expressed as each sub-block matrix

$$\mathbf{A}_{\mathbf{u_j}}^{n+*} \times \mathbf{u}_\mathbf{j}^{n+1} = \mathbf{b}_{\mathbf{u_j}}^{n+*} \tag{30a}$$

for the horizontal velocity u in each x–z plane ($\mathbf{j}=\mathbf{1},\ldots,\mathbf{N}_2$), and

$$\mathbf{A}_{\mathbf{v_i}}^{n+*} \times \mathbf{v}_\mathbf{i}^{n+1} = \mathbf{b}_{\mathbf{v_i}}^{n+*} \tag{30b}$$

for the horizontal velocity v in each y–z plane ($\mathbf{i}=\mathbf{1},\ldots,\mathbf{N}_1$). Equations (30a) can be further rewritten as

$$\mathbf{L}_{\mathbf{i,j}}^{n+*} \times U_{i-1/2,\mathbf{j}}^{n+1} + \mathbf{M}_{\mathbf{i,j}}^{n+*} \times U_{i+1/2,\mathbf{j}}^{n+1} + \mathbf{R}_{\mathbf{i,j}}^{n+*} \times U_{i+3/2,\mathbf{j}}^{n+1} = \mathbf{d}_{\mathbf{i,j}}^{n+*} \tag{31}$$

where $\mathbf{L}_{\mathbf{i,j}}^{n+*}$, $\mathbf{M}_{\mathbf{i,j}}^{n+*}$, and $\mathbf{R}_{\mathbf{i,j}}^{n+*}$, with the dimension of $N_3\times N_3$, are components of the 2DV sub-block tri-diagonal matrix, $\mathbf{A}_{\mathbf{u_j}}^{n+*}$; $U_{i-1/2,\mathbf{j}}^{n+1}$, $U_{i+1/2,\mathbf{j}}^{n+1}$, and $U_{i+3/3,\mathbf{j}}^{n+1}$, with the dimension of $N_3\times 1$, are unknown column vectors corresponding to the $\mathbf{u}_\mathbf{j}^{n+1}$. The matrix system (31) has the only unknown of u^{n+1}, which can be solved by a direct solver, e.g. the double-sweep method [27]. In addition, it is worthwhile to mention that the block coefficient matrices in Equation (31) have the dimension of $N_3\times N_3$, which is only determined by the number of vertical layers. Decreasing vertical layers can reduce the size of block matrices, which is a potential way to improve computational efficiency. A similar procedure can also be applied to solve the horizontal velocity component, $\mathbf{v}_\mathbf{i}^{n+1}$, for each fixed index $\mathbf{i}=\mathbf{1},\mathbf{2}$, or $\mathbf{N_1}$ in the y–z plane, which is not given here. Finally, an iteration procedure is applied to have the known intermediate values, denoted by $n+*$, converge to the unknown advanced ones, denoted by $n+1$, converge to the known intermediate ones, denoted by $n+*$.

The overall algorithm can be summarized below.

1. Initialize the computation domain and update the boundary conditions.
2. Set $u^{n+*} = u^n$, and $v^{n+*} = v^n$.
3. Arrange the block tri-diagonal matrix system in each vertical plane, i.e. solve Equations 30(a) and 30(b) for u^{n+1} and v^{n+1}.
4. If $|u^{n+1} - u^{n+*}| < \varepsilon$ and $|v^{n+1} - v^{n+*}| < \varepsilon$, where ε is an acceptable convergence criterion, go to step 5. Otherwise, update $u^{n+*} = u^{n+1}$ and $v^{n+*} = v^{n+1}$, and go to step 3.
5. Calculate the vertical velocity w^{n+1}, free-surface elevation η^{n+1}, and pressure P^{n+1} from Equations (22), (23), and (26), respectively.
6. Advance to the next time step by replacing the nth value by the $(n+1)$th value, and go back to step 1.

4. NUMERICAL EXPERIMENTS

The model is tested using the following four examples: (1) linear 3D standing waves in a closed basin, (2) finite amplitude sloshing motions, (3) wave propagation over a bar, and (4) wave transformation over an elliptic shoal on a sloped bottom. The viscosity is set to zero here. In the first example, model results using different numbers of vertical layers are compared with analytical solutions. The computational cost with different vertical resolutions is also addressed. The second example tests the capability of the model to simulate finite-amplitude waves, in which four vertical layers are shown to achieve a good agreement between numerical results and second-order analytical solutions. For the last two examples, two vertical layers are employed to demonstrate the model's ability to resolve short wave propagation over uneven bottoms for which experimental data are available.

4.1. Linear 3D standing waves in a closed basin

This example examines the model's capability for simulating 3D linear waves. A closed basin has length $L = 10\,\text{m}$ and width $W = 10\,\text{m}$. The undisturbed water level is $h = 10\,\text{m}$. The wave amplitude, A, is set to 0.1 m, 1% of the water depth. Figure 2 shows the specified initial free-surface elevation

$$\eta(x, y, t = 0) = A\cos(k_x x)\cos(k_y y)\cos\left(\frac{2\pi}{T}t\right) \tag{32}$$

where T is the wave period; k_x and k_y are wave numbers of the first sloshing mode in the x and y directions, respectively, i.e. $k_x = \pi/L$ and $k_y = \pi/W$. Based upon the dispersion relation [28], the total wave number $k = \sqrt{k_x^2 + k_y^2} = 0.44\,\text{rad/m}$ and the corresponding wave period $T = 3.01\,\text{s}$. Since the wave steepness is sufficiently small, i.e. $Ak = 0.1 \times \sqrt{2}\pi/10 = 0.044$, linear wave theory [28] gives a satisfactory approximation of the orbital velocities

$$u = \frac{Agk_x}{\omega}\frac{\cosh[k(h+z)]}{\cosh(kh)}\sin(k_x x)\cos(k_y y)\sin(\omega t) \tag{33a}$$

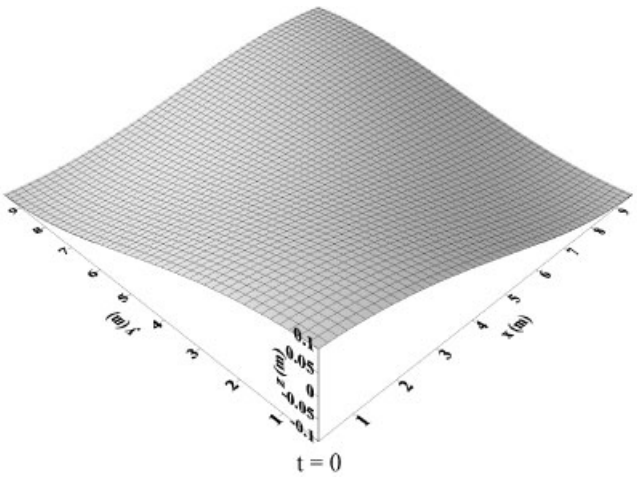

Figure 2. The initial free-surface profile for a linear 3D standing wave oscillation in a closed basin.

$$v = \frac{Agk_y}{\omega} \frac{\cosh[k(h+z)]}{\cosh(kh)} \cos(k_x x) \sin(k_y y) \sin(\omega t) \tag{33b}$$

$$w = -\frac{Agk}{\omega} \frac{\sinh[k(h+z)]}{\cosh(kh)} \cos(k_x x) \cos(k_y y) \sin(\omega t) \tag{33c}$$

The computational domain is discretized by a set of horizontally uniform 20×20 grids. In the vertical z direction different grid sizes are used. For all calculations, the time step is taken as 0.05 s. Figure 3(a)–3(c) compares the free-surface elevation of analytical solutions with those of the three different models using 20 vertical layers. The hydrostatic model (i.e. only the pressure and gravity terms in Equation (4) and the first term in Equation (19) are used) fails to calculate the wave period; the calculated amplitude continues to slightly grow since the hydrostatic model does not consider the dispersion effect but include the non-linear steepening characteristic. Similar results have been reported in other study [8]. In the case of the non-hydrostatic model with the hydrostatic pressure assumption at the top layer (i.e. only the first term in Equation (19) is considered), the wave amplitude is well simulated but the accumulated phase error becomes significant after several wave periods. In contrast, the fully non-hydrostatic model accurately predicts both the amplitude and phase of the wave. To further demonstrate the capability of the fully non-hydrostatic model, we use only two vertical layers (i.e. $N_3 = 2$) to run the case. Figure 3(d) shows that the calculated free-surface elevation is in excellent agreement with the analytical solutions. The results are consistent with those from the two-layer Stelling and Zijlema's model [19] using the Keller-box scheme. Figure 4(a)–4(c) shows the calculated u, v, and w velocity profiles between at different times. One can see that excellent agreement with analytical solutions has been achieved using only two vertical layers in the model.

As mentioned in Section 3.5, solving 3D problems involves an iteration procedure. A convergence criterion, ε, is chosen as 0.001 of expected maximum velocities, $u_{\max}$ and $v_{\max}$.

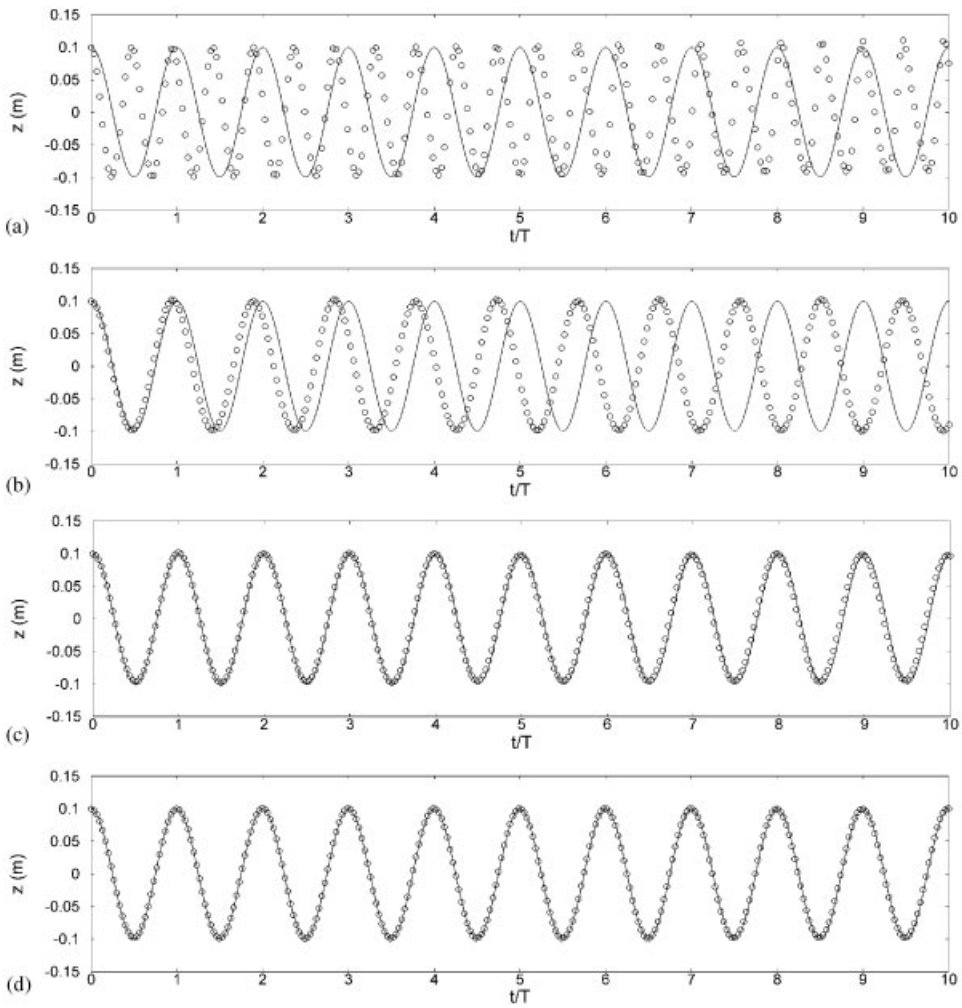

Figure 3. Comparisons of the free-surface elevation at $(x, y) = (0.25,\ 0.25\ \text{m})$ between analytical solutions (solid lines) and numerical results (circles) from different models: (a) hydrostatic model; (b) non-hydrostatic model with hydrostatic pressure representation at top-layer cells; (c) fully non-hydrostatic model; and (d) fully non-hydrostatic model with two vertical layers.

Through numerical tests, an average of 3.2 iterations is needed to achieve to a converged solution. The iterations and CPU times needed for using different layers are summarized in Table I. It is found that using two vertical layers drastically saves the computational cost (as discussed in Section 3.5).

To further check grid convergence and accuracy, a grid convergence index (GCI) proposed by Roache [29] is applied. First a coarser resolution, i.e. 20×20 horizontal grids, 2 vertical layers, and a time step of 0.05 sec, is chosen. The computed free-surface elevation using the coarse grid is shown in Figure 3(d). Then a finer resolution, i.e. 40×40 horizontal grids,

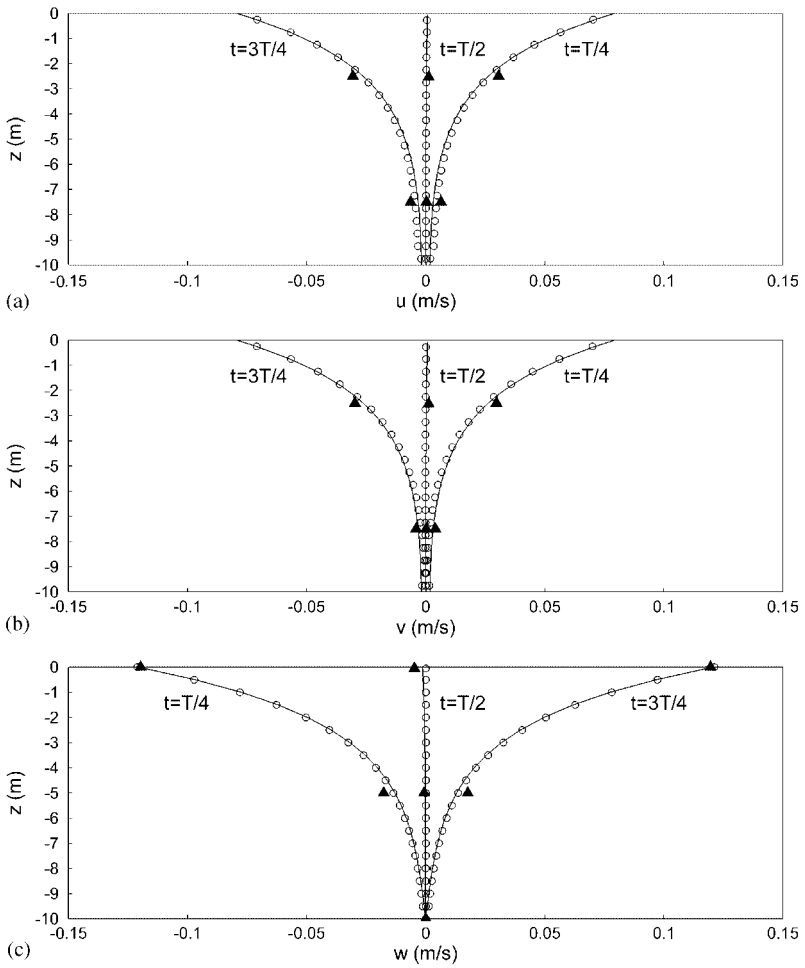

Figure 4. Comparisons of the (u, v, w) velocity profiles between analytical solutions (solid lines) and numerical results of 20-layer (circles) and 2-layer (triangles) at the position $(x, y) = (2.25,\ 2.25\,\text{m})$ at different times.

Table I. Comparison of CPU time for 3D standing wave oscillation in a closed basin.

No. of vertical layers	Total grid No. ($x \times y \times z$)	Time step (Δt)	Averaged iterations of one time step calculation	Averaged CPU time* of one time step calculation
40	$20 \times 20 \times 40$	0.05 s	3.1	29.77 s
20	$20 \times 20 \times 20$		3.2	8.97 s
10	$20 \times 20 \times 10$		3.2	1.83 s
2	$20 \times 20 \times 2$		3.5	0.17 s

* CPU time is recorded using a PC with Intel Pentium 4 CPU 2.00 GHz.

4 vertical layers, and a 0.025 time step, is used to obtain the fine grid solution. The GCI for the fine grid solution ($\mathrm{GCI}_{\mathrm{fine}}$) is defined as

$$\mathrm{GCI}_{\mathrm{fine}} = \frac{3|e|}{r^p - 1} \tag{34}$$

where $e = (\max[(\eta_{\mathrm{coarse}} - \eta_{\mathrm{fine}})_{i,j}])/\max[(\eta_{\mathrm{fine}})_{i,j}]$, in which η_{coarse} and η_{fine} represent the calculated free-surface elevations using coarse and fine resolutions, respectively; (i,j) denotes the horizontal location at $(x, y) = (0.25\,\mathrm{m},\ 0.25\,\mathrm{m})$ and η_{fine} is interpolated to the same horizontal location as that of η_{coarse} for comparison; $r = 2$ is the grid refinement ratio for a grid doubling; and p is the order of convergence of a numerical scheme. Following the suggestion by Roache [29], the more conservative value of $p = 1$ should be used for a reporting purpose. For unsteady problem, a representative time $t = 5T$ is selected here. Equation (34) gives a $\mathrm{GCI}_{\mathrm{fine}} = 0.90\%$, indicating the computation is within the asymptotic range and the solution is well converged. We further examine the grid convergence in the z-direction ($\mathrm{GCI}_{\mathrm{fine}\_z}$) by using a finer resolution (20-layer) and a coarser resolution (2-layer), as shown in Figures 3(c) and 3(d). In this case, the grid refinement ratio $r = 10$ and the calculated $\mathrm{GCI}_{\mathrm{fine}\_z} = 0.96\%$, indicating the model using two vertical layers is sufficiently resolve free-surface elevations.

4.2. Finite amplitude sloshing motions

This example considers finite-amplitude deep-water sloshing motions in a 2DV rectangular basin. The basin has a length of $l = 20\,\mathrm{m}$ and a still water depth of $h = 10\,\mathrm{m}$. For the second sloshing mode, the first-order free-surface elevation can be obtained from linear wave theory [28], i.e.

$$\eta(x,t) = A\cos(k_2 x)\cos(\omega_2 t) \tag{35}$$

where A is wave amplitude, x is the horizontal distance measured from the left side of the basin, $k_2 = 2\pi/l$, and $\omega_2 = \sqrt{gk_2\tanh(k_2 h)}$. Two finite wave amplitudes, i.e. $A_1 = 0.5$ and $A_2 = 1.0\,\mathrm{m}$, are chosen in this example. The corresponding wave steepness, i.e. $A_1 k_2 = 0.5 \times 2\pi/20 = 0.157$ and $A_2 k_2 = 1 \times 2\pi/20 = 0.314$ are both too large for the first-order solution to hold. Using the perturbation technique described in Reference [30], the second-order analytical solution is

$$\begin{aligned}\eta(x,t) = A\Bigg(&\cos(k_2 x)\cos(\omega_2 t) + \frac{A\omega_2^2}{g}\Bigg(\frac{1}{8}\frac{\omega_2^4 + g^2k_2^2}{\omega_2^4} \\ &+ \left(\frac{1}{8}\frac{\omega_2^4 - g^2k_2^2}{\omega_2^4} - \frac{3}{2}\frac{\omega_2^4 - g^2k_2^2}{\omega_2^2(4\omega_2^2 - \omega_4^2)}\right)\cos(2\omega_2 t) \\ &+ \frac{1}{2}\frac{\omega_2^2\omega_4^4 - \omega_2^4 - 3g^2k_2^2}{\omega_2^2(4\omega_2^2 - \omega_4^2)}\cos(\omega_4 t)\Bigg)\cos(2k_2 x)\Bigg)\end{aligned} \tag{36}$$

where $k_4 = 4\pi/l$ and $\omega_4 = \sqrt{gk_4\tanh(k_4 h)}$.

In model calculation, the initial free-surface profile is prescribed by setting $t = 0$ in Equation (35) and a null velocity field is specified. The domain is discretized by 40 horizontal uniform grids and four vertical layers. The time step is 0.05 s. A comparison of the

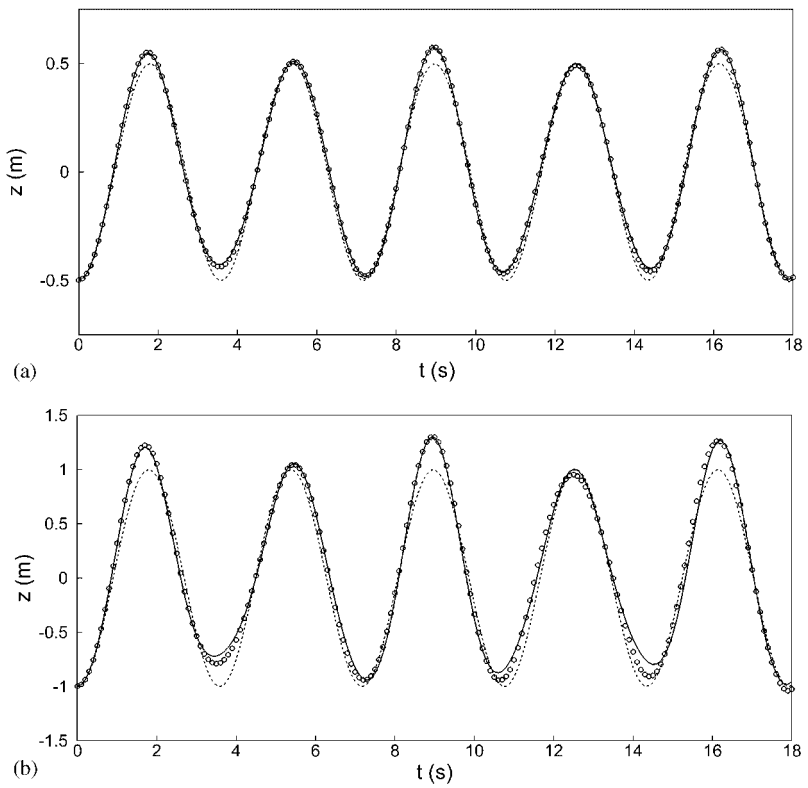

Figure 5. Comparisons of the time series finite amplitude sloshing motions at $x = l/2$ for the first-order analytical solutions (dashed lines) and the second-order solution (solid lines), and numerical results (circles): (a) $A = 0.5$ m; and (b) $A = 1.0$ m.

free-surface elevations at the middle of the basin, i.e. $x = l/2$, among the model result, first-order analytical solution, and second-order analytical solution is shown in Figure 5. In the case of moderate wave steepness, Figure 5(a) shows that the model predicts a higher wave crest and a lower wave trough than those in the first order analytical solution but is in excellent agreement with the second-order solution, indicating the model's capability to simulate nonlinear waves. For the larger wave steepness, Figure 5(b) shows good agreements between model results and the second-order solution except for wave crests and troughs, in which the noticeable discrepancies are also reported from other studies [31–34]. Figure 6 further shows that the predicted moderate and steep spatial wave profiles across the tank at different times are very similar to the second-order solutions. As the wave becomes steeper, dispersive effects are more pronounced at the nodes, i.e. $x = 5$ and 15 m, consistent with those reported by Chern *et al.* [32] and Turnbull *et al.* [33].

Finally, it is recognized that a sufficiently large number of vertical layers is needed to model accurately large-amplitude dispersive waves. For instance, at least 40–120 vertical layers are needed in several models [8, 31–34]. In contrast, our model uses only four vertical layers, one order of magnitude smaller than those models, to achieve comparable accuracy. In other

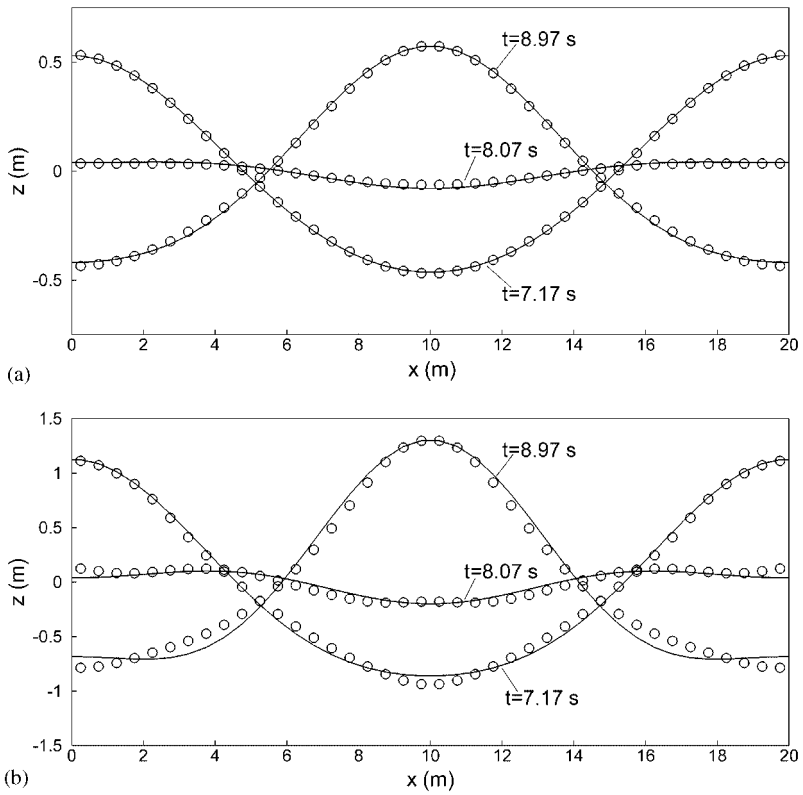

Figure 6. Comparisons of the spatial profile of the finite amplitude sloshing motions between the second-order analytical solutions (solid lines) and numerical results (circles) at three different times: (a) $A = 0.5$ m; and (b) $A = 1.0$ m.

words, using only a small number of vertical layers, the non-hydrostatic model with the new top-layer pressure treatment in this paper is capable of simulating nonlinear free surface waves.

4.3. Wave propagation over a submerged bar

In order to examine the capability of the model to simulate waves interacting with uneven bottoms, we investigate a regular wave train traveling over a submerged bar, which has been experimentally and numerically studied by Beji and Battjes [35], Casulli [7], Li and Fleming [10], Lin and Li [11], Chen [8], and Stelling and Zijlema [19]. It has been found that the shoaling would occur when the wave train passes over the upward slope. The nonlinearity would generate significant higher harmonics that travel phase-locked to the primary wave. Previous numerical tests also showed that the hydrostatic model gives a totally unrealistic prediction of the free-surface elevation in comparison with experimental data [8].

In this test, the numerical results are compared to the experimental data from Beji and Battijes [35]. The geometry used in numerical calculation is depicted in Figure 7. The

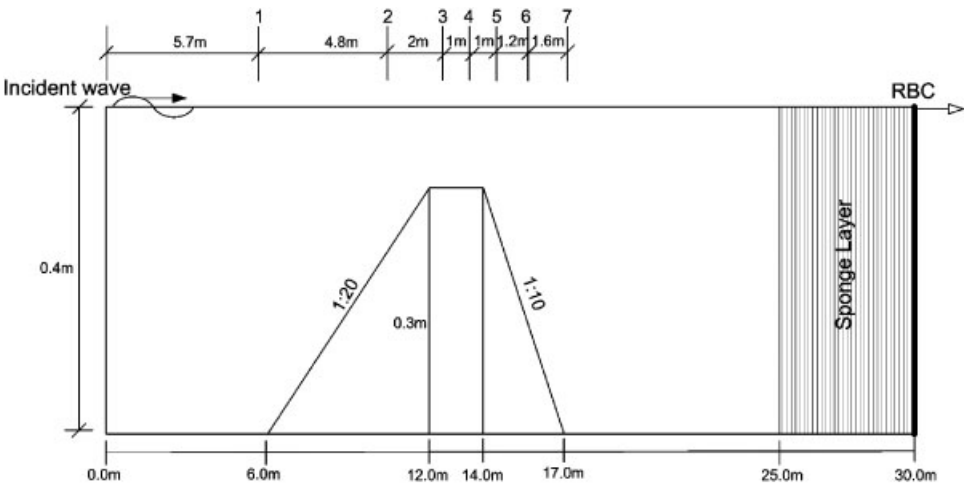

Figure 7. Sketch of the geometry for periodic wave propagation over a submerged bar.

computation domain has a total length of 30 m and the undisturbed mean water level of 0.4 m. An incident wave train with a wave height, $H_0 = 2$ cm, and a wave period, $T_0 = 2.0$ s, is prescribed at the inflow boundary. As indicated by Beji and Battjes [35], the beach at the downstream in their physical experimental setup only served for absorbing waves. Therefore applying numerical treatments to replace the physical beach downstream can minimize wave reflection. In this study we replace the physical beach by a 5m sponge layer with a Sommerfeld-type radiation boundary for the outflow boundary. Similar treatments can be found in other numerical models [10, 11, 19, 36]. To discretize the computational domain, 1200 constant horizontal grids and only two vertical layers are chosen. The grid size of the top-layer cells, Δz_{N_3}, is set to 0.08 m, resulting in the bottom cell size, $\Delta z_{\mathrm{bt}} = 0.02 \sim 0.32$ m. The time step is taken as 0.01 s.

Comparisons of the free-surface elevation at the six measured locations between numerical results and experimental data are plotted in Figure 8. The model correctly simulates the shoaling phenomenon at location 2 and the wave riding over the bar at location 3. The secondary wave mode at locations 4 and 5 is also well predicted. In addition, the model is capable to resolve the dispersion for the higher-frequency components occurring behind the bar at locations 6 and 7. The overall excellent agreements indicate that model's capability to simulate complex interactions between short waves and uneven bottoms.

Note that previous studies indicate that models applying either a hydrostatic pressure assumption or a hydrostatic relation at the top layer require approximately at least 10–20 vertical layers to simulate properly free-surface elevations over uneven bottoms [7, 8, 10, 11, 17]. Contrary to this, Stelling and Zijlema [19] found that the Keller-box scheme based non-hydrostatic model with two layers correctly resolves short waves. Similarly, our non-hydrostatic model with the top-layer non-hydrostatic pressure treatment can achieve accurate results using only two vertical layers. Specifically, the top-layer treatment is within a staggered grid framework and can be readily implemented to most existing models.

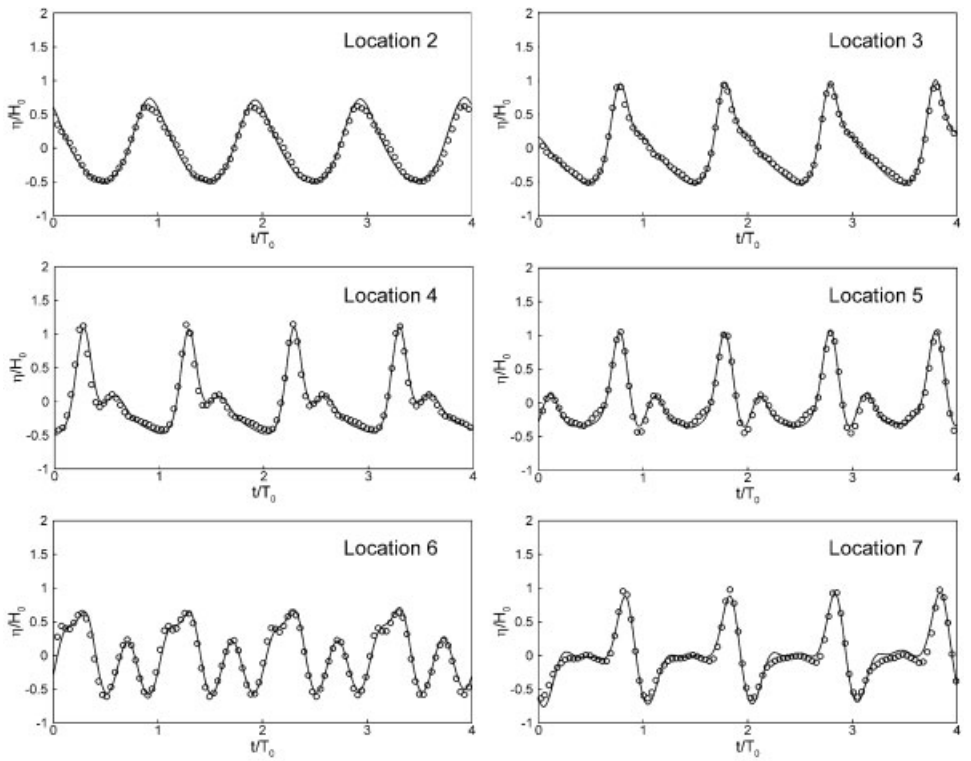

Figure 8. Comparisons of the free-surface elevation between numerical results (solid lines) and experimental data (circles) at six wave gauge locations for a periodic wave propagation over a submerged bar.

4.4. Wave transformation over an elliptic shoal on a sloped bottom

The last example is to test the capability of our model for simulating refraction and diffraction caused by wave propagation over a 3D uneven bottom. We compare the numerical results with experimental data from Berkhoff *et al.* [37], in which monochromatic wave propagation over an elliptic shoal on a sloped plane were considered.

Figure 9 shows the bathymetry that corresponds to the experimental setup. The incoming wave with a wave height $H_0 = 4.64\,\text{cm}$ and a wave period, $T_0 = 1.0\,\text{s}$ is specified at the inflow boundary based on linear wave theory. At the outflow, a sponge layer coupled with a radiation boundary is employed to minimize wave reflection. A reflecting wall condition is assigned at both lateral boundaries. In the horizontal plane, the grid spacing is set to $\Delta x = 0.1$ and $\Delta y = 0.05\,\text{m}$. The vertical grid size of the top-layer cells, Δz_{N_j}, is set to 0.08 m, which leads to the bottom cell size, Δz_{bt}, ranging from 0.02 to 0.37 m. The total grids in computation are therefore $200 \times 600 \times 2$. The time step is taken as 0.01 s and the total simulation time is up to 34 s.

Figure 10 shows the comparison of relative wave height at eight sections between the numerical results and experimental data. The predicted wave amplitude is obtained by

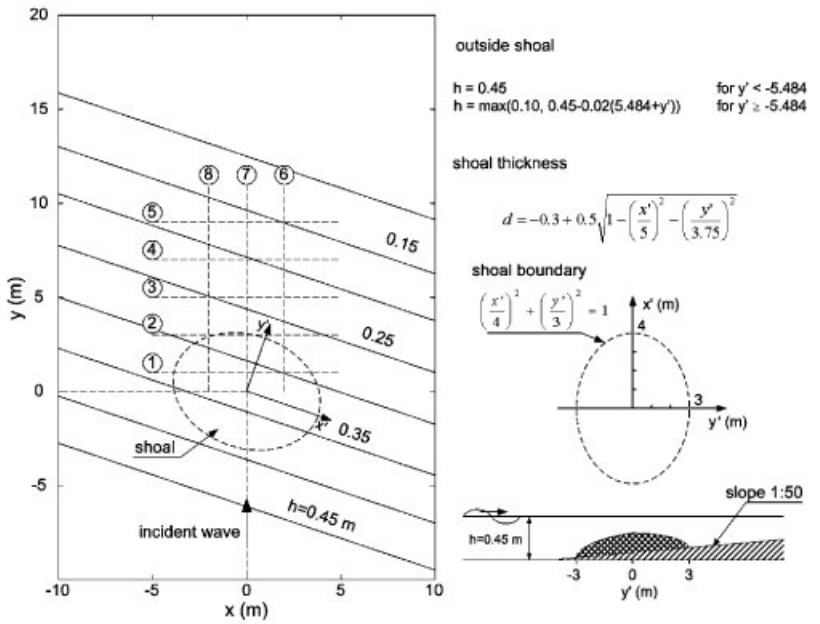

Figure 9. Bottom configuration for periodic wave propagation over an elliptic shoal, corresponding to the experimental setup of Berkhoff *et al.* [37].

averaging over four wave periods (i.e. from $t = 30$ to 34 s) once a steady solution is achieved. In section 1 where waves have propagated over half of the elliptic shoal, the model satisfactorily predicts shoaling effects. In section 2, minor discrepancy of the wave height along the x direction is predicted, consistent with the results from other studies [10, 38]. The focusing effect of the shoal is well predicted in sections 3, 4, and 5 where the maximum wave amplitude amplification factor is 2.2, 2.0 and 1.7, respectively. In sections 6, 7, and 8 along the y direction, the model results are generally close to the experimental data. Similar results have been obtained from Li and Fleming [10] using 11 vertical layers, and Stelling and Zijlema [19] using two vertical layers. Overall, the comparisons between the present model using only two vertical layers and experimental data are satisfactory, indicating the model's capability to resolve the nonlinear effects of wave refraction and diffraction over a 3D irregular bottom.

5. CONCLUSIONS

In this paper, we present an implicit, 3D, fully non-hydrostatic model. Specifically the new treatment of the non-hydrostatic pressure at the top layer makes the model free of any hydrostatic pressure assumption. The 3D NSE is discretized by the staggered Crank–Nicholson scheme, which gives the model a second-order accuracy. The resulting system with the unknown of horizontal velocities is solved by a decomposition method that involves an

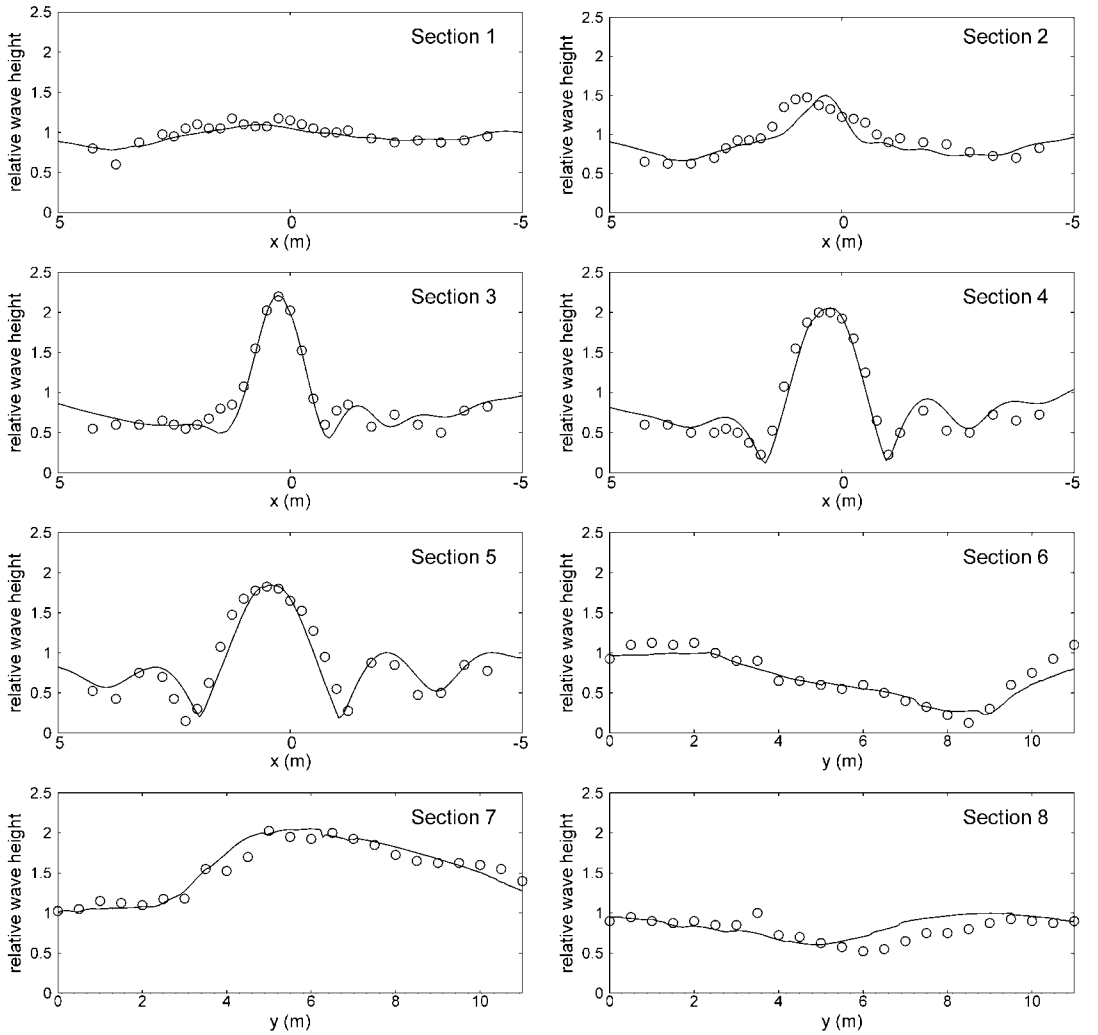

Figure 10. Comparisons of relative wave heights for periodic wave propagation over an elliptic shoal between numerical results (solid lines) and experimental data (circles) at eight sections.

iteration procedure. The implicit discretization enables the model to simultaneously solve all flow-field components (u, v, w, P, and η) at each time step.

The model can simulate complicated free-surface flow problems with a very small number of vertical layers. Numerical tests have shown that using only two layers the model is capable to simulate accurately the linear dispersive wave in Example 1, nonlinear waves interacting with uneven bottoms in Example 3, and wave refraction and diffraction in Example 4. For waves with higher steepness and larger nonlinearity (e.g. $Ak > 0.15$), the model gives reasonable results by using a limited number of vertical layers (i.e. four layers) in Example 2. It is

found that reducing the number of vertical layers drastically decreases the computational cost (see Table I). The models without the fully non-hydrostatic feature (e.g. non-hydrostatic models with hydrostatic top-layer pressure assumption), however, need more vertical layers (approximately 10–20 layers) to simulate correctly the free-surface flow problems. In addition, the horizontal resolutions for the four examples chosen in this study are approximately the same as those of others e.g. [7, 8, 10, 11, 18], suggesting that no strong interplay between horizontal and vertical resolutions occurs in model simulations.

Finally, the model is capable of simulating rotational flow over the entire range of water depths, and has the potential to be coupled with a turbulence closure scheme. In addition, the algorithm could be parallelized because the 2DV problems decomposed from the 3D problem are independent of each other. In this context, it is believed that the computational efficiency can be further improved. The model is currently extended to couple with a sophisticated turbulent model with an aim to simulate large-scale circulations in lakes.

ACKNOWLEDGEMENTS

This research grants was supported by both the NSF-funded North Temperate Lakes LTER program and the Wisconsin Alumni Research Foundation Project Number 040061.

REFERENCES

1. Blumberg AF, Mellor GL. A description of a three-dimensional coastal ocean circulation model. In *Three-Dimensional Coastal Ocean Circulation Models*, *Coastal and Estuarine Sciences*, vol. 4, Heaps NS (ed.). AGU: Washington, DC, 1987; 1–16.
2. Cheng RT, Casulli V, Gartner JW. Tidal, residual, intertidal mudflat (TRIM) model and its applications to San Francisco Bay, California. *Estuarine Coastal Shelf Science* 1993; **36**:235–280.
3. Sheng YP. On modeling three-dimensional estuarine and marine hydrodynamics. In *Three-Dimensional Models of Marine and Estuarine Dynamics*, Nihoul J, Jamart B (eds). Elsevier: Amsterdam, Netherlands, 1987; 35–54.
4. Hamrick JM. A three-dimensional environmental fluid dynamic computer code: theoretical and computational aspects. *Special Report in Applied Marine Science and Ocean Engineering*, No. 317. Virginia Institute of Marine Science, College of William & Mary, Gloucester Point, Virginia, 1992.
5. Oliger J, Sundstrom A. Theoretical and practical aspects of some initial boundary value problems in fluid dynamics. *SIAM Journal on Applied Mathematics* 1978; **35**:419–446.
6. Mahadevan A, Oliger J, Street R. A non-hydrostatic mesoscale ocean model. Part 1: well posedness and acaling. *Journal of Physics and Oceanography* 1996; **26**:1868–1880.
7. Casulli V. A semi-implicit finite difference method for non-hydrostatic, free-surface flows. *International Journal for Numerical Methods in Fluids* 1999; **30**:425–440.
8. Chen X. A fully hydrodynamic model for three-dimensional, free-surface flows. *International Journal for Numerical Methods in Fluids* 2003; **42**(9):929–952.
9. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
10. Li B, Fleming C. Three-dimensional model of Navier–Stokes equations for water waves. *ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering* 2001; **January/February**:16–25.
11. Lin P, Li C. A σ-co-ordinate three-dimensional numerical model for surface wave propagation. *International Journal for Numerical Methods in Fluids* 2002; **38**:1045–1068.
12. Namin M, Lin B, Falconer R. An implicit numerical algorithm for solving non-hydrostatic free-surface flow problems. *International Journal for Numerical Methods in Fluids* 2001; **35**:341–356.
13. Yuan HL, Wu CH. A two-dimensional vertical non-hydrostatic σ model with an implicit method for free-surface flows. *International Journal for Numerical Methods in Fluids* 2004; **44**:811–835.
14. Park JC, Kim MH, Miyata H. Fully non-linear free-surface simulations by a 3D viscous numerical wave tank. *International Journal for Numerical Methods in Fluids* 1999; **29**(6):685–703.
15. Hur DS, Mizutani N. Numerical estimation of the wave forces acting on a three-dimensional body on submerged breakwater. *Coastal Engineering* 2003; **47**:329–345.
16. Hodges BR, Street RL. On simulation of turbulent nonlinear free-surface flows. *Journal of Computational Physics* 1999; **151**(2):425–457.

17. Casulli V, Stelling GS. Numerical simulation of 3D quasi-hydrostatic free-surface flows. *Journal of Hydraulic Engineering* (ASCE) 1998; **124**(7):678–686.
18. Casulli V, Zanolli P. Semi-implicit numerical modeling of nonhydrostatic free-surface flows for environmental problems. *Mathematical and Computer Modeling* 2002; **36**(9–10):1131–1149.
19. Stelling G, Zijlema M. An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. *International Journal for Numerical Methods in Fluids* 2003; **43**:1–23.
20. Pacanowski RC, Gnanadesikan A. Transient response in a *Z*-level ocean model that resolves topography with partial cells. *Monthly Weather Review* 1998; **126**:3248–3270.
21. Beam RM, Warming RF. An implicit factored scheme for the compressible Navier–Stokes equations. *AIAA Journal* 1978; **16**:393–402.
22. Haney RL. On the pressure gradient force over steep topography in σ co-ordinate ocean models. *Journal of Physics and Oceanography* 1991; **21**:610–619.
23. Stelling G, Van Kester JATM. On the approximation of horizontal gradients in sigma co-ordinates for bathymetry with steep bottom slopes. *International Journal for Numerical Methods in Fluids* 1994; **18**:915–935.
24. Adcroft A, Hill C, Marshall J. Representation of topography by shaved cells in a height co-ordinate ocean model. *Monthly Weather Review* 1996; **125**:2293–2315.
25. Chen X. Responses of a hybrid *z*-level model to various topography treatments for a boundary-value problem and an initial-value problem. In *Estuarine and Coastal Modeling*, *Proceedings of the 7th International Conference*, Spaulding ML (ed.). ASCE: St. Petersburg, FL, 2001; 614–627.
26. Golub GH, Van Loan CF. *Matrix Computations*. The Johns Hopkins University Press: U.S.A., 1983.
27. Tannehill JC, Anderson DA, Pletcher RH. *Computational Fluid Mechanics and Heat Transfer* (2nd edn). In *Series in Computational and Physical Processes in Mechanics and Thermal Sciences*, Minkowycz WJ, Sparrow EM (eds). Taylor & Francis, Hemisphere Publishing Corporation: Bristol, PA, 1997; 717–724.
28. Mei CC. *The Applied Dynamics of Ocean Surface Waves*. Wiley Inter Science; 1983.
29. Roache PJ. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering* 1994; **116**:405–413.
30. Wu GX, Eatock Taylor R. Finite element analysis of two-dimensional non-linear transient water waves. *Applied Ocean Research* 1994; **16**:363–372.
31. Greaves DM, Borthwick AGL, Wu GX, Eatock Taylor R. A moving boundary finite element method for fully non-linear wave simulations. *Journal of Ship Research* 1997; **41**(3):181–194.
32. Chern MJ, Borthwick AGL, Eatock Taylor R. A pseudospectral σ-transformation model of 2-D nonlinear waves. *Journal of Fluids and Structures* 1999; **13**:607–630.
33. Turnbull MS, Borthwick AGL, Eatock Taylor R. Numerical wave tank based on a σ-transformed finite element inviscid flow solver. *International Journal for Numerical Methods in Fluids* 2003; **42**:641–663.
34. Frandsen JB, Borthwick AGL. Simulation of sloshing motions in fixed and vertically excited containers using a 2-D inviscid σ-transformed finite difference solver. *Journal of Fluids and Structure* 2003; **18**:197–214.
35. Beji S, Battjes JA. Experimental investigation of wave propagation over a bar. *Coastal Engineering* 1993; **19**:151–162.
36. Beji S, Battjes JA. Numerical simulation of non-linear waves propagation over a bar. *Coastal Engineering* 1994; **23**:1–16.
37. Berkhoff JCW, Booy N, Radder AC. Verification of numerical wave propagation models for simple harmonic linear water waves. *Coastal Engineering* 1982; **6**:255–279.
38. Wei G, Kirby JT, Sinha A. Generation of waves in Boussinesq models using a source function method. *Coastal Engineering* 1999; **36**:271–299.

