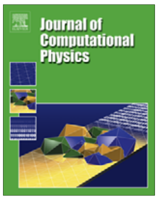
Journal of Computational Physics 228 (2009) 2100–2118

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



An efficient flexible-order model for 3D nonlinear water waves

A.P. Engsig-Karup [a,*], H.B. Bingham [b], O. Lindberg [b]

[a] Technical University of Denmark, Department of Informatics and Mathematical Modeling, Lyngby, Denmark
[b] Technical University of Denmark, Department of Mechanical Engineering, Lyngby, Denmark

ARTICLE INFO

Article history:
Received 7 July 2008
Received in revised form 18 November 2008
Accepted 20 November 2008
Available online 7 December 2008

Keywords:
Nonlinear waves
Ocean engineering
Finite differences
Multigrid
Potential flow
Time domain

ABSTRACT

The flexible-order, finite difference based fully nonlinear potential flow model described in [H.B. Bingham, H. Zhang, On the accuracy of finite difference solutions for nonlinear water waves, J. Eng. Math. 58 (2007) 211–228] is extended to three dimensions (3D). In order to obtain an optimal scaling of the solution effort multigrid is employed to precondition a GMRES iterative solution of the discretized Laplace problem. A robust multigrid method based on Gauss–Seidel smoothing is found to require special treatment of the boundary conditions along solid boundaries, and in particular on the sea bottom. A new discretization scheme using one layer of grid points outside the fluid domain is presented and shown to provide convergent solutions over the full physical and discrete parameter space of interest. Linear analysis of the fundamental properties of the scheme with respect to accuracy, robustness and energy conservation are presented together with demonstrations of grid independent iteration count and optimal scaling of the solution effort. Calculations are made for 3D nonlinear wave problems for steep nonlinear waves and a shoaling problem which show good agreement with experimental measurements and other calculations from the literature.

1. Introduction

The goal of this work is a computational tool suitable for large scale simulation of nonlinear wave–wave, wave–bottom and wave–structure interaction in the coastal and offshore environment. For the engineering problems that we have in mind (*e.g.* coastal and offshore wind turbines, wave-power devices, ships, or offshore platforms) viscous effects are typically only important very close to solid boundaries and in regions where strong wave breaking occurs. Reynolds numbers for these problems are typically on the order of 10^5 to 10^9 which makes a direct solution of the Navier–Stokes equations out of the question. With Large Eddy Simulation (LES) or Reynolds averaged (RANS) turbulence modelling, domain sizes are typically on the order of one wavelength at current levels of computational power. Our strategy is therefore to solve the problem under the assumptions of a potential flow. In regions where viscous effects may be important, we will attempt to model the physics as a surface forcing term on the otherwise potential flow and/or couple our potential flow solver with a more refined viscous flow solver. In this paper however, we will only consider the solution of the potential flow problem.

We discuss here a flexible-order, finite difference based solution of the fully nonlinear potential flow problem for waves on a fluid of variable depth. The time-varying physical domain is mapped to a time-invariant boundary-fitted computational domain to obtain time-constant discrete differential operators weighted by time-dependent coefficients. The application of this basic technique is widespread, and its use for the simulation of unsteady free surface flows goes back at least three

[*] Corresponding author. Tel.: +45 45251370.
E-mail address: apek@imm.dtu.dk (A.P. Engsig-Karup).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.11.028

decades [10]. Related applications were later described in [4,1,5,14,2]. In principle, this approach can be applied to a Lagrangian or semi-Lagrangian description of the flow which allows for overturning waves. In practice however, most implementations are based on an Eulerian description of the flow which limits the resultant model to free surface shapes which are a single-valued function of the horizontal coordinate. This is mainly motivated by computational efficiency issues and justified by our focus on the modelling of waves that are not overturning. In line with the strategy outlined above, we intend to model wave breaking by adding dissipative terms on the free surface along the lines proposed for example in [7] or [22]. In this context, it may be more appropriate to solve the Euler equations rather than a pure potential flow. This is a straightforward extension of the current method where the Laplace solver becomes a Poisson solver, and the full 3D velocity field is time-stepped instead of just the free surface potential. For a broader discussion of the general topic of numerical solution of free surface flows we refer to reviews in [21,19,12,6].

Having chosen to solve the full 3D potential flow problem directly as stated, the remaining issues are purely numerical, *i.e.* how accurately and efficiently can we solve the problem? Ten years ago, a multigrid solution was developed [14] to the second-order discretization of the problem on a uniform grid. More recently, this approach was modified in [2] to allow arbitrary order finite difference schemes as well as non-uniform grid spacing. A significant advantage was found for high-order schemes on a vertically clustered grid relative to second-order schemes on a uniform grid. This should actually not be surprising given the exponential decay of the wave motion in the vertical, and the fact that the advantage of using high-order schemes for solving this problem was pointed out 30 years ago in [10]. Instead of using multigrid, the resultant sparse linear system of equations was solved in [2] using the Generalized Minimum Residual (GMRES) method [16], preconditioned by a direct solution of the linear (time-invariant) second-order discretized system matrix. This led to optimal scaling of the solution effort in two dimensions (2D).

In this paper, we extend the method described in [2] to 3D. In 3D, a direct solution of the preconditioning problem leads to a super-linear scaling of the solution effort with increasing problem size. Using multigrid to solve the preconditioning problem, however, we show that the solution retains an optimal scaling for 3D problems. The motivation for using multigrid only for the preconditioning step is that the preconditioning matrix is time-constant and thus the multigrid operators need only be determined once during the initial set-up, which increases the efficiency of the subsequent solve steps. This strategy has proved to be more efficient and robust than applying multigrid to the full high-order, nonlinear system. The use of just one multigrid *F*-cycle in preconditioning is found to be most effective, providing a nearly equal distribution of effort between the matrix–vector product and the preconditioning. The performance of just one *V*- or *W*-cycle is comparable however. Average iteration counts are typically less than 10 to satisfy a relative residual tolerance of 10^{-6}. At this stage, the cross-over point where multigrid becomes faster than a direct solution of the preconditioning step is found to be approximately $n = 10^5$ total grid points. In general, the relative efficiency of the two strategies, measured by the resultant average iteration count, is problem dependent and will change this number.

It is well known that the multigrid solution of the Laplace problem using Gauss–Seidel smoothers is sensitive to the direct introduction of Neumann-type boundary equations, *e.g.* at the bottom, and particularly on anisotropic grids. This seems to be primarily due to the resultant reduction of diagonal dominance in the matrix. Vertical line smoothing instead of point smoothing can help the situation, but in our experience all Gauss–Seidel schemes fail to converge on highly anisotropic grids. Here we introduce an alternative method for satisfying the boundary conditions along solid boundaries by introducing a set of computational points outside of the fluid domain. Derivatives at the boundary are then expressed including these 'ghost' points, and used to satisfy the impermeability condition at the boundary. Solving the boundary condition equations explicitly allows the ghost point contributions to be eliminated from the Laplace equation at the boundary, thus satisfying both the Laplace equation and the boundary condition at the same time. This modified system is found to be convergent under all forms of Gauss–Seidel iteration for all practically relevant values of boundary slope, relative water depth, and grid anisotropy.

While it is important to have an optimal scaling of the computational effort to achieve an efficient numerical method, perhaps equally important is to ensure that the required Random Access Memory (RAM) scales linearly with problem size. In this respect the multigrid method is even more advantageous over a direct solution, as all matrix operations are carried out explicitly and can thus be done using only local operations. Such a matrix-free implementation leads to minimal memory requirements and optimal scaling of the memory use, thus allowing larger problems to be treated. The algorithms are also well suited to parallelization and this is the topic of ongoing work.

The remainder of the paper is organized as follows. In Section 2, the formulation of the exact potential flow wave problem is reviewed. The numerical solution is described in Section 3 including the new method of imposing boundary conditions along solid boundaries. Section 3.3 describes the solution of the preconditioning problem using multigrid. An analysis of the spectral radius of the system under Gauss–Seidel iteration, combined with numerical calculations, is used to demonstrate the improved robustness of the new boundary condition scheme. Here we also demonstrate the optimal scaling of the solution effort and memory use, to our knowledge for the first time. In Section 4, we establish the linear stability, accuracy and convergence properties of the method. Nonlinear stability and accuracy are also addressed and some guidelines given. In Section 5, some 3D validation test cases are computed including standing waves, highly nonlinear periodic waves, and the evolution of initially mildly nonlinear waves as they propagate over a focusing bar. These last calculations compare well with experimental measurements and to other high-accuracy numerical results from the literature. Conclusions are drawn in Section 6.

2. Formulation of the problem

A Cartesian coordinate system is adopted with the xy-plane located at the still water level and the z-axis pointing upwards. The still water depth is given by $h(\boldsymbol{x})$ with $\boldsymbol{x} = (x, y)$ the horizontal coordinate. The position of the free surface is defined by $z = \eta(\boldsymbol{x}, t)$ and the gravitational acceleration $g = 9.81$ m^2/s is assumed to be constant.

Assuming an inviscid fluid and an irrotational flow, the fluid velocity $(\boldsymbol{u}, w) = (u, v, w) = (\nabla\phi, \partial_z\phi)$ is defined by the gradient of a scalar velocity potential $\phi(\boldsymbol{x}, z, t)$, where $\nabla = (\partial_x, \partial_y)$ is the horizontal gradient operator. The evolution of the free surface is governed by the kinematic and dynamic boundary conditions

$$\partial_t\eta = -\nabla\eta\cdot\nabla\tilde{\phi} + \tilde{w}(1 + \nabla\eta\cdot\nabla\eta), \tag{1a}$$

$$\partial_t\tilde{\phi} = -g\eta - \frac{1}{2}(\nabla\tilde{\phi}\cdot\nabla\tilde{\phi} - \tilde{w}^2(1 + \nabla\eta\cdot\nabla\eta)), \tag{1b}$$

which are expressed in terms of the free surface quantities $\tilde{\phi} = \phi(\boldsymbol{x}, \eta, t)$ and $\tilde{w} = \partial_z\phi|_{z=\eta}$. To find $\tilde{w}$ and evolve these equations forward in time requires solving the Laplace equation in the fluid volume with a known $\tilde{\phi}$ and η, together with the kinematic bottom boundary condition

$$\phi = \tilde{\phi}, \quad z = \eta, \tag{2a}$$

$$\nabla^2\phi + \partial_{zz}\phi = 0, \quad -h \leqslant z < \eta, \tag{2b}$$

$$\partial_z\phi + \nabla h\cdot\nabla\phi = 0, \quad z = -h. \tag{2c}$$

At the structural boundaries of the domain, the flow field must be everywhere parallel to the boundary surfaces, implying that the velocity potential ϕ must satisfy the no-normal flow condition (expressed here in physical coordinates)

$$\boldsymbol{n}\cdot(\nabla, \partial_z)\phi = 0, \quad (\boldsymbol{x}, z) \in \partial\Omega, \tag{3}$$

where $\boldsymbol{n} = (n_x, n_y, n_z)$ is an outward pointing normal vector to the solid boundary surfaces $\partial\Omega$. At present, we assume that all structural boundaries except the fluid bottom are vertical and aligned with one of the horizontal coordinates, but the extension to general boundaries is implementation-wise conceptually identical to the treatment of the bottom boundary and is in progress.

Since the free surface is a time-dependent moving boundary with an *a priori* unknown position, it is convenient to make a change of variable in the vertical which maps the solution to a time-invariant domain using the following (non-conformal) σ-coordinate transformation

$$\sigma \equiv \frac{z + h(\boldsymbol{x})}{\eta(\boldsymbol{x}, t) + h(\boldsymbol{x})} \equiv \frac{z + h(\boldsymbol{x})}{d(\boldsymbol{x}, t)}. \tag{4}$$

The Laplace problem in the transformed computational domain becomes

$$\Phi = \tilde{\phi}, \quad \sigma = 1, \tag{5a}$$

$$\nabla^2\Phi + \nabla^2\sigma(\partial_\sigma\Phi) + 2\nabla\sigma\cdot\nabla(\partial_\sigma\Phi) + (\nabla\sigma\cdot\nabla\sigma + (\partial_z\sigma)^2)\partial_{\sigma\sigma}\Phi = 0, \quad 0 \leqslant \sigma < 1, \tag{5b}$$

$$(\partial_z\sigma + \nabla h\cdot\nabla\sigma)(\partial_\sigma\Phi) + \nabla h\cdot\nabla\Phi = 0, \quad \sigma = 0, \tag{5c}$$

where $\Phi(\boldsymbol{x}, \sigma, t) = \phi(\boldsymbol{x}, z, t)$ and the derivatives of the coordinate σ can be written as

$$\nabla\sigma = \frac{1-\sigma}{d}\nabla h - \frac{\sigma}{d}\nabla\eta, \tag{6a}$$

$$\nabla^2\sigma = \frac{1-\sigma}{d}\left(\nabla^2 h - \frac{\nabla h\cdot\nabla h}{d}\right) - \frac{\sigma}{d}\left(\nabla^2\eta - \frac{\nabla\eta\cdot\nabla\eta}{d}\right) - \frac{1-2\sigma}{d^2}\nabla h\cdot\nabla\eta - \frac{\nabla\sigma}{d}\cdot(\nabla h + \nabla\eta), \tag{6b}$$

$$\partial_z\sigma = \frac{1}{d}. \tag{6c}$$

Note that all of these nonlinear coefficients can be determined from the known free surface and bottom positions.

In the σ-coordinates, the structural boundary conditions takes the form

$$\boldsymbol{n}\cdot(\nabla, \partial_z\sigma\partial_\sigma)\phi = 0, \quad (\boldsymbol{x}, \sigma) \in \partial\Omega. \tag{7}$$

Having obtained a solution for the function Φ in the σ-domain, the physical internal flow kinematics are obtained via the chain rule

$$\boldsymbol{u}(\boldsymbol{x}, z) = \nabla\phi(\boldsymbol{x}, z) = \nabla\Phi(\boldsymbol{x}, \sigma) + \nabla\sigma\partial_\sigma\Phi(\boldsymbol{x}, \sigma), \tag{8a}$$

$$w(\boldsymbol{x}, z) = \partial_z\phi(\boldsymbol{x}, z) = \partial_\sigma\Phi(\boldsymbol{x}, \sigma)\partial_z\sigma. \tag{8b}$$

3. Numerical solution

A method of lines approach is used for the discretization of the continuous problem stated above. For the time-integration of the free surface conditions (1) the classical explicit four-stage, fourth-order Runge–Kutta scheme (see *e.g.* [11]) is employed. For the spatial discretization, a grid of (N_x, N_y) points is defined along the horizontal *xy*-axes at which the free surface variables η and $\tilde{\phi}$ are to be evolved. Spatial derivatives are replaced by the discrete counterparts using the finite difference method and nonlinear terms are treated by direct product approximations at the collocation points. At the structural boundaries of the domain, *i.e.* at the bottom and wall sides, Neumann (specified normal component of velocity) conditions, (5c) and (7), are imposed as described below.

For the solution of the transformed Laplace problem (5), N_σ points are defined in the vertical below each horizontal free surface grid point, arbitrarily spaced in $0 \leqslant \sigma \leqslant 1$. The grid is thus structured, but with one arbitrarily spaced set of values along each of the coordinate directions. Choosing r nearby points, allows order $(r-1)$ finite difference schemes for the 1D first and second derivatives in (x, y, σ) to be developed in the standard way using Taylor series expansion (see *e.g.* [9]) at each of the $x,\ y$ and σ positions on the grid.

By keeping the order of the spatial discretization schemes flexible, two convergence strategies are available, namely, *h*- and *p*-adaptivity where either the spatial resolution or the order of the scheme is increased, respectively.

3.1. Finite difference discretizations

For the one-dimensional first- and second-derivatives in x, y and σ, $r = \alpha + \beta + 1$ points are used where α indicates the number of points in the positive coordinate direction, and β the number of points in the negative direction from the point of interest. For interior points all derivatives are centrally discretized with $\alpha = \beta$ in each coordinate direction. For points where a centered stencil would reach beyond the last computational point in the domain, the stencils become off-centered to use only the available grid points. In [14,2], computational points were distributed along the solid boundaries and inside the fluid domain, and each boundary point gave rise to one equation for imposing the boundary condition. Thus, the Laplace equation at the boundary point was exchanged for the boundary condition. The numerical evidence suggests that this exchange introduces instability (in particular, divergence of multigrid) which is accentuated by shallow water physics, large bottom gradients, and high anisotropy in the grid. A possible explanation for this behavior is suggested in Section 3.2. Here we develop a more robust solution by ensuring the satisfaction of both the Laplace equation and the boundary condition at all boundary points. This is done by introducing additional fictitious computational points outside the physical domain, and using the associated extra degrees of freedom to impose both equations at the solid boundaries. This is intuitively a nice feature, and it turns out to be critical for obtaining robust multigrid solutions.

In the special case of a vertical wall aligned with one of the horizontal axes, we simply take η and ϕ to be symmetric about the boundary and reflect the centered finite difference schemes for an even function to eliminate the ghost points from the system. For boundaries not aligned with the coordinates such as the bottom boundary, only one layer of ghost points is introduced and it is used to satisfy the boundary condition, while the Laplace equation is imposed using the equations corresponding to the boundary points themselves. This is discussed in more detail in Section 3.2. Mixed $x\sigma$- and $y\sigma$-derivatives are obtained by successive application of 1D schemes. In the case of a direct solution where the matrix must be formed, these operators are obtained by multiplication of the discrete 1D matrix operators, which gives 2D stencils of at most r^2 points. In this way, all derivatives are formally accurate to $\mathcal{O}(\Delta x_*^{r-1})$ where Δx_* is the maximum grid spacing in either x, y or σ. This approach also leads naturally to a stable treatment of internal corners and allows the introduction of fixed bottom-mounted structures, which is the topic of a follow-up paper.

This discretization procedure gives a rank $n = N_x N_y N_z$ linear system of equations

$$A\Phi = b, \tag{9}$$

where A is the coefficient matrix, Φ a vector of unknown Φ's at each grid point at the given instant of time and b a vector holding zeros, except at those points corresponding to inhomogeneous boundary conditions.

For the solution of (9) we employ the GMRES method preconditioned on the left by the linearized second-order accurate version of the coefficient matrix A which we refer to as A_2. This matrix is time-constant and has a relatively simple banded structure. One initial LU-factorization (after re-ordering to minimize fill-in) and subsequent back-substitution for each preconditioning step is effective up to a point, but for large problems the fill-in becomes excessive and this strategy leads to a super-linear scaling of both CPU time and RAM memory use. To retain an optimal scaling in 3D, we replace the direct solution of the preconditioning step with a multigrid solver as described in Section 3.3. First however, we discuss in more detail the new discretization scheme along solid boundaries.

3.2. Detailed discussion of the kinematic bottom boundary condition

The discretization schemes described above provide two strategies for imposing the kinematic boundary condition along a solid boundary: (1) replace the Laplace equation at the boundary points with the kinematic boundary condition (as was done in [14] and at the bottom in [2]). We will refer to this strategy as BC. (2) Impose both the Laplace equation and the boundary condition at all boundary points by adding fictitious ghost points outside of the fluid boundaries. We will refer to this strategy as BC + LAPLACE.

These two strategies are applied to the bottom boundary condition as pictured in Fig. 1. For illustration, consider the 2D case with a uniform grid spacing $(\Delta x, \Delta\sigma)$ and second-order finite difference operators. Let Φ_{jk} represent the discrete value of $\Phi(x, \sigma, t)$ at grid location (x_j, σ_k) and take the ordering of the grid points to follow the coordinate directions. For strategy BC shown in Fig. 1(a), $j,\ k$ corresponds to the bottom boundary point itself and the bottom boundary condition takes the form

$$\frac{a_j}{2\Delta\sigma}(-3\Phi_{j,k} + 4\Phi_{j,k+1} - \Phi_{j,k+2}) + \frac{b_j}{2\Delta x}(\Phi_{j+1,k} - \Phi_{j-1,k}) = 0, \tag{10}$$

where $a_j = \frac{1}{h(x_j)+\eta(x_j,t)}(1 + b_j^2)$ and $b_j = \partial_x h|_{x=x_j}$ are the factors appearing in (5c) and the classical uniform grid, second-order schemes have been applied. For strategy BC + LAPLACE shown in Fig. 1(b), $j,\ k$ corresponds to the ghost point below the bottom boundary which is obtained by reflecting the next grid point above the bottom about $\sigma = 0$. In this case, the bottom boundary condition takes the form

$$\frac{a_j}{2\Delta\sigma}(\Phi_{j,k+2} - \Phi_{j,k}) + \frac{b_j}{2\Delta x}(\Phi_{j+1,k+1} - \Phi_{j-1,k+1}) = 0 \tag{11}$$

with the coefficients evaluated at $\sigma = 0$. For strategy BC, the Laplace equation (5b) is imposed one grid point above the bottom boundary as shown in Fig. 2(a). As the expressions are rather lengthy, we avoid writing them down here, but note that the four corner points of the stencil have been introduced by the mixed $\sigma - x$ derivative. For strategy BC + LAPLACE, the equation corresponding to the grid point on the bottom boundary itself is the Laplace equation, and the stencil for this equation is shown in Fig. 2(b). Eq. (11) is now used to solve explicitly for the ghost point values in terms of their neighbors inside the fluid domain. Their contributions can then be eliminated from the expressions for the Laplace equation at the boundary points to arrive at the resulting stencil shown in Fig. 2(c). For the linear case on a constant depth the Laplace equation, after the elimination procedure, simplifies to

$$\frac{1}{\Delta x^2}\Phi_{j-1,k} - 2\left(\frac{1}{\Delta x^2} + \frac{1}{h^2\Delta\sigma^2}\right)\Phi_{j,k} + \frac{2}{h^2\Delta\sigma^2}\Phi_{j,k+1} + \frac{1}{\Delta x^2}\Phi_{j+1,k} = 0, \tag{12}$$

where the index $j,\ k$ corresponds to the filled dot in Fig. 2(c).

This can be immediately identified as the Laplace equation with the boundary condition built in. Thus, this strategy makes the boundary equations essentially the same as the interior point equations and the resultant matrix becomes mildly diagonally dominant in the limit of a linear problem on a constant depth. Gauss–Seidel iteration is thus guaranteed to be convergent in this limit [18]. Although a non-zero bottom slope will work to reduce the diagonal dominance in the system, we

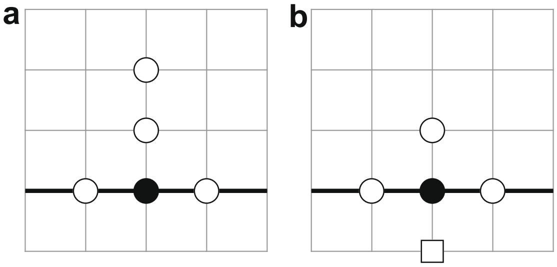


Fig. 1. Implementation of the bottom boundary condition along the thick black line. Strategy (a) BC and (b) BC + LAPLACE. The square indicates a ghost point outside the fluid domain and the equation is imposed at the filled circle.

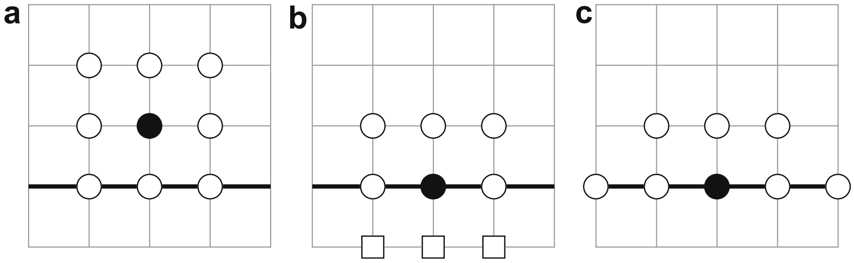


Fig. 2. Implementation of the Laplace equation near the boundary (thick black line). Square points are below the fluid bottom and the equation is imposed at the filled circle. (a) Strategy BC. (b) Strategy BC + LAPLACE. (c) Strategy BC + LAPLACE after elimination of the ghost points.

have so far not found any divergent examples regardless of the physical parameters and/or the grid anisotropy, as is discussed further in Section 3.3.2. On the other hand, the exchange of the Laplace equation for (10), apparently reduces the diagonal dominance of the matrix to such a degree that Gauss–Seidel iteration becomes divergent, even at relatively small values of bottom slope and/or grid anisotropy.

3.3. Multigrid solution of the preconditioning step

This section outlines our left preconditioned GMRES iterative solution of the Laplace problem which is written symbolically as

$$A_2^{-1}\{A\Phi = b\}, \tag{13}$$

where Φ is the solution vector, b the right hand side, A the nonlinear flexible-order system matrix and A_2 the linearized, second-order accurate version of the system matrix used for preconditioning. Starting with the initial guess of the solution Φ_0 (obtained using linear extrapolation from the previous two time steps on first and third Runge–Kutta stages and otherwise just the last computed solution) the initial residual is computed, $r_0 = A\Phi_0 - b$. This is preconditioned by solving the system $A_2u_0 = r_0$. The GMRES procedure then builds up the Krylov subspace to drive the residual down to the convergence tolerance and provide a correction to the initial guess. This process requires one preconditioning operation per iteration solving a system of the form

$$A_2u_m = r_m, \tag{14}$$

where m indicates the iteration number.

We apply geometric multigrid [3,18] to solve this preconditioning problem with an initial zero guess for u_0. Multigrid exploits the smoothing properties of the basic stationary iterative methods which efficiently remove the high-frequency errors but not the low-frequency ones. Transferring low-frequency error to a coarser grid makes it higher frequency and restores the effectiveness of the basic iterative method.

A set of K increasingly finer grids $\{G^k : G^k \in \Omega_\sigma,\ k = 1, 2, \ldots, K\}$ is thus defined, where k denotes the grid level, K the finest grid and $\Omega_\sigma = [0, L_x] \times [0, L_y] \times [0, 1]$ the computational domain. To move quantities back and forth between the grids, prolongation and restriction operators are defined as

$$P^k : u^{k-1} \to u^k, \quad R^k : u^k \to u^{k-1}, \quad k = 2, \ldots, K. \tag{15}$$

Each grid level must also have a matrix operator A_2^k. The two classical approaches for determining the coarse grid operator A_2^k are: the Direct Coarse grid Approximation (DCA) where the operator is determined on the coarse grid in exactly the same way as is done on the finest grid; and the Galerkin Coarse grid Approximation (GCA) where the coarse grid operator is determined as the product of the fine grid operators $A_2^{k-1} = R^kA_2^kP^k$ [18]. We have tried both methods and not found any significant difference, we thus use DCA since it is slightly more efficient and straightforward to implement matrix-free.

3.3.1. Restriction and prolongation operators

The restriction to each coarse grid point is determined in d dimensions by tensor products of the one-dimensional restriction operators for each coordinate direction. Discrete one-dimensional first-order full-weighting operators are derived from the discrete version of the volume-preserving condition

$$\int_{\Omega_f^k} \Phi^k(\boldsymbol{x}, \sigma)d\Omega^k = \int_{\Omega_f^k} R^k\Phi^k(\boldsymbol{x}, \sigma)d\Omega^{k-1} \tag{16}$$

for $\Omega_f^k = [x_i^k - \Delta x_1^k, x + \Delta x_2^k] \times [y_j^k - \Delta y_1^k, y_j^k + \Delta y_2^k] \times [\sigma_k^k - \Delta\sigma_1^k, \sigma_k^k + \Delta\sigma_2^k]$ on the fine grid, which accommodates the use of uneven grids. The trapezoidal rule is used to approximate the left hand side of the integral and the midpoint rule for the right hand side.

The discrete full-weighting operators are modified at boundary points by imposing the boundary conditions using the same ghost point method that was used for the spatial discretization of the governing equations in Section 3.2. For restrictions at the bottom boundary we approximate the kinematic bottom boundary condition with a homogeneous Neumann boundary condition $\partial_\sigma\Phi = 0$. This approximation is only strictly correct for a flat bottom, but it is convenient in terms of implementation and does not appear to effect the overall performance of the method.

Alternatively, we could employ cubic nine-point full-weighting restriction stencils independently of the choice of coarsening strategy. However, it turns out that by the approach outlined above we minimize the size of the restriction stencil and thereby the work load of a multigrid cycle, *e.g.* for semi-coarsening in one direction we then obtain a three-point full-weighting restriction stencil.

The prolongation operators are determined using tri-linear interpolation from the coarse to the fine grid where standard coarsening is employed and linear interpolation where semi-coarsening is employed. Thus, if the fine and coarse grid points coincide along one of the coordinate directions, then direct injection is employed in that coordinate.

3.3.2. Residual relaxation schemes

Gauss–Seidel has been applied as the basic iterative scheme (the smoother) on which the multigrid solution is built. Gauss–Seidel is a stationary iterative method which can be written in the form

$$x_{m+1} = M^{-1}(Nx_m + b), \tag{17}$$

where x_m is the approximate solution at iteration number m and the matrix has been split into $A = M - N$. M here is a lower *non-singular* triangular matrix for point methods ($M_{ij} = 0,\ \forall i > j$) and a lower block triangular matrix for block methods ($M_{ij} = 0,\ \forall i > j + s$, s a parameter dependent on the method), while N is an upper triangular matrix [18]. The iterative scheme will be (asymptotically) convergent if the spectral radius of the iteration matrix fulfills $\rho(M^{-1}N) < 1$, where $\rho(G)$ is the maximum absolute value of the eigenvalues of G.

A discrete parameter study has been carried out to investigate the dependence of the spectral radius of the iteration matrix $\rho(M^{-1}N)$ on discrete anisotropy and bottom gradient. We consider the point-wise Gauss–Seidel method based on forward vertical ordering of the unknowns (*i.e.* column wise from bottom left to top right). Since the spectral radius of the iteration matrix determines the convergence rate for the slowest error mode, it determines the asymptotic convergence rate of the method. The tests are carried out in two spatial dimensions by defining the bottom function as

$$h(x) = h_0 + 0.5\alpha \tanh\left(5.3\frac{x - x_0}{\beta}\right). \tag{18}$$

Using this function we can vary the maximum normalized gradient $L\max|h_x/h|$. By choosing $\beta = 0.5L$ (measure of shoal width), where L is the domain length, and with $|d'(x)|$ chosen, we can determine α (the magnitude of the gradient) from the expression $\max|d'(x)| = 2.65|\alpha/\beta|$.

In Fig. 3, we have visualized the results for parameter studies on uniform and vertically non-uniform grids, respectively for the BC in sub-figures (a) and (c), and for the BC + LAPLACE strategies in the sub-figures (b) and (d). For the BC formulation the discretizations may be divergent if the discrete anisotropy is very large or if large relative bottom gradients are experienced. In general, the convergence rate of the smoother is fastest for flat bottoms, small discrete anisotropy ($\Delta x \approx \Delta\sigma$) and relatively deep water. If the discrete BC + LAPLACE formulation is used we have not been able to find any divergent schemes for any choices of parameters, nor have we experienced divergence problems in practical simulations for much larger problems, *e.g.* see the simulations in Section 5.

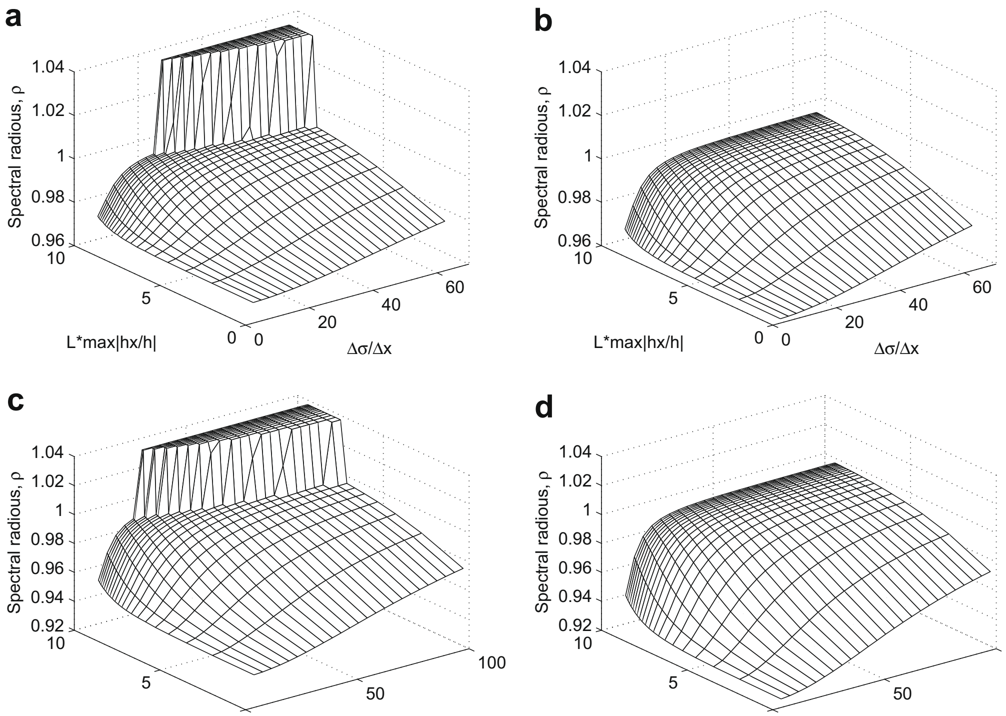


Fig. 3. Parameter studies of the spectral radius of the point-wise Gauss–Seidel iteration matrix $\rho(M^{-1}N)$ with $A_2 = M - N$ for the two different ways of imposing the kinematic bottom boundary conditions. Either uniform (top row) or vertically clustered grids (bottom row) are used. In sub-figures (a) and (c) for the BC and (b) and (d) for the BC + LAPLACE the variation in spectral radius is given as a function of discrete anisotropy $\Delta\sigma/\Delta x$ and dimensionless maximum bottom gradient $\max|h_x|/h_0$. Large spectral radius are visualized as $\rho = 1.04$.

Based on the results in Fig. 3(a), we conclude that imposing the kinematic bottom boundary condition by exchanging the discrete Laplace-type equation at the boundary can result in a numerical scheme which is not robust. Instead, to achieve a robust Gauss–Seidel iterative scheme, the spatial boundary conditions should be imposed together with the Laplace equations at all spatial boundaries. Although these results are only for the point Gauss–Seidel method, we have also tested vertical line smoothing and other orderings of the points/lines, but not found significant differences in the overall performance or robustness [15]. Thus, we use point Gauss–Seidel in the following.

We note that this study seems to contradict the conclusions in [14] where it was concluded that the point Gauss–Seidel method was divergent for some of their shallow water test cases, but that the problem was fixed by switching to vertical line smoothing. Although they do not mention the implementation details in the paper, we speculate that the explanation for this lies in the imposition of the horizontal boundary conditions which leads to stringent stability conditions in shallow water as discussed in Section 4.

3.3.3. Grid coarsening strategy

In the present work, we have only employed nested grids as this is sufficient for demonstrating the efficiency of the method. However, the extension to allow for fine grids allowing for arbitrary combinations of N_x and N_y is straightforward. Hence, the number of points in each coordinate direction must be odd on all grids except the coarsest grid, $N_i^k = 2(N_i^{k-1} - 1) + 1,\ k = 2, \ldots, K,\ i = x, y, \sigma$, where $k = 1$ corresponds to the coarsest grid. The advantage of this approach is that we can employ standard coarsening or semi-coarsening. Since sufficient accuracy is generally obtained by using $O(10)$ grid points in the vertical, for practical applications of the model there will typically be many more points in the horizontal directions than in the vertical. The spatial resolution will thus be dominated by the plane directions and therefore strongly coupled along these, and we can expect the discrete equation system to become anisotropic. This can lead to poor convergence rates for the smoothers, and therefore semi-coarsening is employed to improve the convergence rate of the smoother on the coarser grids.

The current grid coarsening strategy is based on a combined semi- and standard-coarsening strategy. Semi-coarsening is done independently along each horizontal coordinate until the spatial resolution (measured by the number of points) matches the spatial resolution in the vertical. Thereafter, standard-coarsening is employed until the coarsest grid is reached. The coarsest grid ($k = 1$) should be a grid where the solution effort for the direct solution of the linear system is negligible compared to the overall effort.

3.3.4. Multigrid solution procedure

Following the work described in [18] we outline the complete multigrid solution procedure. The pseudo-code for one multigrid solve step returning $u_i^K = (A_2^K)^{-1} r_i^K$ in *maxiter* iterations with A_2^K the fine grid preconditioner and r_i^K the fine grid residual for the *i*th GMRES iteration is as follows

```
$u_{i,m}^K = 0;\ m = 0;$
WHILE $m < maxiter$ DO
  $u_{i,m+1}^K = \mathrm{MGCYC}(K, \gamma, u_{i,m}^K, A_2^K, r_i^K, \nu_1, \nu_2)$
  $m = m + 1$
END
```

Here (ν_1, ν_2) are the numbers of pre- and post-smoothing operations per cycle, γ is the cycle index indicating how many times the recursive multigrid solver MGCYC is called at the *k*th grid, $k = K, \ldots, 1$. For example, $\gamma = 1$ corresponds to so-called *V*-cycles and $\gamma = 2$ to *W*-cycles. Note that the convergence criterion is given solely in terms of the maximum number of iterations. Since multigrid is only used for the preconditioning step, we are not concerned with a strict convergence of this solution, but only in maximizing the efficiency of the GMRES iterative solution to the full nonlinear problem. We can thus avoid computation of the fine grid defect $d_{i,m+1}^K = r_{i,m+1}^K - A_2^K u_{i,m+1}^K$ and save one relatively expensive matrix–vector product per cycle.

The pseudo-code for the recursive multigrid cycle is given by

```
**recursive function** $u_{m+1}^k = \mathrm{MGCYC}(k, \gamma, u_m^k, A^k, r^k, \nu_1, \nu_2)$
IF $k = 1$ THEN
  $u_{m+1}^k = (A^{-1})^k r_m^k$
ELSE
  $\hat{u}_m^k = \mathrm{SMOOTH}^{\nu_1}(u_m^k, A^k, r^k)$
  $d_m^k = r^k - A^k \hat{u}_m^k$
  $d_m^{k-1} = R^k d_m^k$
  $\delta u_m^k = \mathrm{MGCYC}(k - 1, \gamma, 0, A^{k-1}, d_m^{k-1}, \nu_1, \nu_2)$
  $\hat{u}_m^k = \hat{u}_m^k + P^k \delta u_m^k$
  $u_{m+1}^k = \mathrm{SMOOTH}^{\nu_2}(\hat{u}_m^k, A^k, r^k)$
END
```

where SMOOTH$^{\nu}$ is a function that smooths the solution ν times, *e.g.* using a point-wise Gauss–Seidel method. Only the *V*-cycle has been given here for brevity.

3.3.5. Model performance

The performance of the algorithm is governed by the discrete properties of the model and the physical properties of the simulated problem. There are five dimensionless quantities which can be used to describe the dynamical behavior of the algorithm and influence the amount of computational work. The choice of discretization can be characterized by the Courant number and the discrete anisotropy. The physics can be characterized by the dimensionless depth, the nonlinearity and relative bed slope. In addition the numerical precision and specified error tolerances for the iterative GMRES solution procedure are important parameters.

The performance of the algorithm has been tested using a sequential implementation in Fortran 90 for a three-dimensional test case where the initial condition on a square domain $(x, y) \in [0, 1]^2$ is a two-dimensional sinusoidal free surface elevation and no kinetic energy, above a bottom that varies from shallow to deep water. Thus, all features of the model are tested. The BC + LAPLACE strategy has been used to impose the kinematic bottom constraint with a vertical clustering of the nodes toward the free surface using the symmetric half of the Chebyshev–Gauss–Lobatto grid distribution $\sigma_i = \sin(\pi i/(2(N_z - 1)))$ for $i = 0, \dots, N_z - 1$. The number of nodes in the vertical has been kept fixed at $N_z = 10$ with one being a ghost point $\sigma_{-1} = \sigma_1$. Sixth-order finite difference operators have been applied.

We seek to determine the optimal multigrid preconditioning strategy in terms of the type and number of multigrid cycles and pre- and post-smoothing operations and possibly the number of grids. We have chosen a fine grid of size 129 × 129 × 10 and varied the other parameters. The relative and absolute error tolerances in the SPARSKIT GMRES routine are in general set to be `atol` = 100 · `rtol` with `rtol` = 10^{-6} unless otherwise specified. These tolerances are used with the following stopping criteria $\|R\| \leqslant \text{rtol}\|b\| + \text{atol}$ where R is the residual vector computed for the linear system $Ax = b$.

Result from tests with different numbers of multigrid cycles and numbers of pre- and post-smoothing operations for the multigrid preconditioner are shown in Table 1. The number of GMRES iterations varies with the number of smoothing operations, and so does the efficiency. Increasing the number of smoothing operations improves the convergence rate of the multigrid cycles and reduces the number of GMRES iterations toward the iteration count for the direct solution of the preconditioner. However, since every smoothing effectively corresponds to a matrix–vector product, doing one extra smoothing has to be compensated for by a drop in the GMRES iteration count in order to pay off. From the results in Table 1, we conclude that one multigrid cycle with one pre- and post-smoothing is near-optimal. For the present test case a single *F*-cycle with one pre- and one post-smoothing is found to be optimal and therefore this strategy is adopted in the following.

The efficiency of the multigrid preconditioning step can be measured by the average number of iterations which is linearly proportional to the computational work. The average number of iterations for a fixed Courant number $C_r = 0.5$ for different spatial resolution is investigated for strongly nonlinear streamfunction waves with nonlinearity $H/L = 90\%(H/L)_{\max}$ in both shallow $(kh = 0.5)$ and deep $(kh = 2\pi)$ water. The computational domain is chosen such that the waves propagate in parallel with the longitudinal direction of a wave tank with longitudinal length $L_{tank} = 6L_{wave}$ and width $W_{tank} = L_{wave} = 1$ m. The width is resolved using seven grid points in tests. Fig. 4 shows the average number of iterations determined using 100 time increments to solve the same test case described above. This is done for increasing numbers of grid points in the longitudinal direction and for different formal orders of the finite difference operators. On average we can expect iteration counts between 2 and 5. The highest iterations counts are experienced in shallow water for steep nonlinear waves,

Table 1
Average iteration count per RK4 stage for different multigrid-preconditioning strategies on a fixed grid. Vertical clustering has been employed with a ghost point layer for kinematic bottom boundary conditions. One hundred time steps with $\Delta t = 0.0039$. Notation: MG-1V(2,1) indicates one *V*-cycle with two pre- and one post-smoothing, etc.

Preconditioning method	Grid	Total grids	Average iteration	CPU (s)
LU	129 × 129 × 10	1	1.90	239
MG-1V(1,1)	129 × 129 × 10	5	2.13	237
MG-2V(1,1)	129 × 129 × 10	5	1.91	404
MG-1V(2,1)	129 × 129 × 10	5	2.09	342
MG-1V(1,2)	129 × 129 × 10	5	2.09	264
MG-1V(2,2)	129 × 129 × 10	5	2.08	419
MG-1W(1,1)	129 × 129 × 10	5	1.93	233
MG-2W(1,1)	129 × 129 × 10	5	1.90	453
MG-1W(2,1)	129 × 129 × 10	5	1.91	323
MG-1W(1,2)	129 × 129 × 10	5	1.90	303
MG-1W(2,2)	129 × 129 × 10	5	1.90	402
MG-1F(1,1)	129 × 129 × 10	5	1.93	231
MG-2F(1,1)	129 × 129 × 10	5	1.90	381
MG-1F(2,1)	129 × 129 × 10	5	1.92	348
MG-1F(1,2)	129 × 129 × 10	5	1.91	324
MG-1F(2,2)	129 × 129 × 10	5	1.92	316

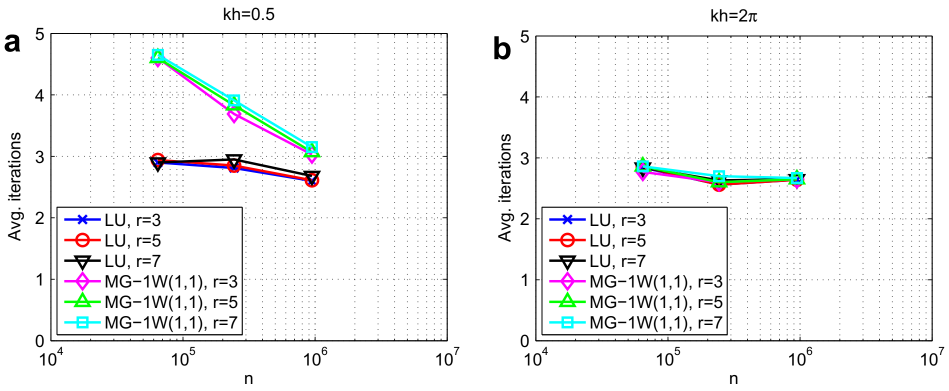


Fig. 4. Average iteration counts for steep streamfunction waves $(H/L = 90\%(H/L)_{\max})$ for different orders of the scheme and problem sizes. (a) Shallow water, $kh = 0.5$, and (b) deep waver, $kh = 2\pi$.

where high spatial resolution accuracy is required to resolve the wave profile accurately. In shallow water, the average iteration count has a tendency to decrease with increasing n since the discrete anisotropy $d\sigma/dx$ decreases under refinement in both the plane and vertical directions when a uniform grid distribution is employed in the horizontal and a clustered grid in the vertical. Thus, we have demonstrated that the iteration count is nearly independent of both n, the physics and the order of the finite difference operators. The latter property makes it possible to adjust the order of the scheme to the accuracy that is needed in order to balance the total amount of computational work.

To test the efficiency of the GMRES-MG solution strategy a test is carried out where the Courant number $C_r = 0.5$ is fixed and the temporal integration is carried out over 100 time steps (time span different for each run). In Fig. 5, results are presented for two different solution strategies for the linear system, namely, a GMRES solver preconditioned with either MG-1F(1,1) or with a full direct LU-factorization. The multigrid preconditioning strategy is the most efficient and achieves ideal $\mathcal{O}(n)$-scaling. This result is in contrast to the direct LU preconditioner which can be seen to scale super-linearly with n. Recall, that in [2] ideal scaling was demonstrated for the direct LU preconditioner in two dimensions where the bandwidth of the sparse coefficient matrix can be kept small and therefore solved efficiently. The break-even point where the MG preconditioning becomes more efficient than the direct LU, was found to be approximately $n_{BE} = 9 \cdot 10^4$. However, in actual calculations the break even point is dependent on the problem and the relative efficiency of the two different preconditioning strategies, which can be estimated using the ratio between the average iteration counts for each strategy for a given problem.

4. Stability, accuracy and convergence

The stability, accuracy and convergence properties of the model were analyzed in [2, Section 4]. This analysis was done using the 2D linearized traveling wave solution on a constant depth with periodic lateral boundary conditions and strategy BC applied to the bottom boundary condition. Here we extend this analysis to confirm that the new discretization strategy at the lateral and bottom boundaries does not adversely affect the expected convergence and stability properties of the method. The addition of a second horizontal dimension to the problem should not change anything in this respect but for completeness we also present a 3D convergence test in Section 5.1 to verify that this indeed the case.

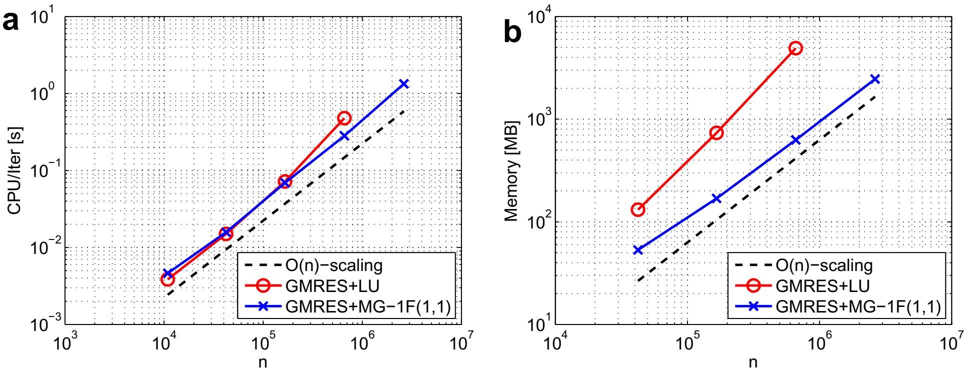


Fig. 5. Scaling of (a) computational effort, and (b) RAM memory use.

4.1. Stability

Following the discussion in [2], Section 4, we express the time-stepping problem in the form

$$\frac{\partial}{\partial t}\begin{bmatrix}\eta\\ \tilde{\phi}\end{bmatrix} = \begin{bmatrix}0 & J_{12}\\ -g & 0\end{bmatrix}\begin{bmatrix}\eta\\ \tilde{\phi}\end{bmatrix}, \tag{19}$$

where for the linear problem $\tilde{\phi} = \phi(x,0,t)$. For a given discretization of the problem on a rectangle of length L and depth h we can solve explicitly for the vertical free surface velocity to get

$$\tilde{w} = J_{12}\tilde{\phi}, \quad J_{12} = D_z^0 A^{-1} P^0. \tag{20}$$

Here D_z^0 is the discrete matrix operator which takes the vertical derivative of ϕ on $z = 0$ and P^0 is the matrix operator which builds the right hand side vector b from the values of $\tilde{\phi}(x)$. For small values of N_x and N_z, we can build the discrete matrix operator J_{12} and hence the complete Jacobian matrix in (19). The largest magnitude of the eigenvalues of the Jacobian, in combination with the stability region of a particular time-integration scheme, will then determine the stability of the method.

The exact linear solution on a periodic domain is given by

$$\eta(x,t) = \frac{H}{2}\cos(kx - \omega t), \quad \omega = \sqrt{gk\tanh kh}, \tag{21a}$$

$$\phi(x,z,t) = \frac{H}{2}\frac{g\omega}{k}\frac{\cosh[k(z+h)]}{\cosh(kh)}\sin(kx - \omega t). \tag{21b}$$

Thus for the continuous linear problem, we have

$$\tilde{w} = \left.\frac{\partial\phi}{\partial z}\right|_{z=0} = k\tanh(kh)\tilde{\phi} \tag{22}$$

or in other words, $J_{12} = k\tanh(kh)$. The continuous eigenvalues of the Jacobian are thus simply the wave frequencies $\pm i\sqrt{gk\tanh(kh)}$.

Fig. 6 plots the maximum magnitude of the eigenvalues of the Jacobian for these two boundary condition strategies as functions of the relative water depth $k_N h$ where L is the domain length, $\Delta x = L/(N_x - 1)$, and the Nyquist wave number is $k_N = \pi/\Delta x$. The eigenvalues have been normalized by the Nyquist frequency $\omega_N = \sqrt{gk_N \tanh k_n h}$. This plot has been made using $N_x = 17$ and $N_z = 9$, but the picture is qualitatively unchanged for different choices of the number of grid points. From the plot it is clear that the two strategies lead to the same stability requirement in deep water but they are dramatically different in shallow water. The eigenvalues of the discrete system are purely imaginary, so stability under fourth-order Runge–Kutta time-stepping is governed by $\Delta t|\lambda|_{\max} \leqslant 2\sqrt{2}$ (*i.e.* the extent of the imaginary axis included by the stability region). For example when $h = L/100$, strategy BC will require a time step which is approximately 10 times smaller than the one required for $h = L$. On the other hand, for strategy BC + LAPLACE the stability requirements are roughly independent of relative water depth and identical to those given by the periodic domain analysis in [2].

These results were obtained using strategy BC at the bottom boundary, as was also done in [2] to obtain their Figs. 3 and 4, but the corresponding results using strategy BC + LAPLACE at the bottom are very nearly identical.

4.2. Convergence

Having established that using the ghost point method does not adversely affect stability, we now confirm that it does not change the convergence properties of the method. To this end, we repeat the analysis of Bingham and Zhang [2, Section 4] leading up to their Fig. 2, with strategy BC replaced by strategy BC + LAPLACE. Given the linear solution for $\tilde{\phi} = \phi(x,0)$ at a

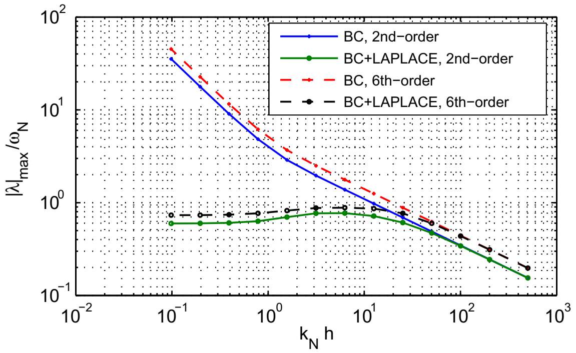


Fig. 6. Maximum eigenvalues of the Jacobian matrix for the two discretization schemes along lateral boundaries as a function of relative water depth.

fixed value of $kh = 4$, we compute semi-analytically the value of $\tilde{w}$ obtained by the model using increasingly fine resolutions in x and z. Semi-analytically in the sense that we assume a uniform grid spacing in the x-direction with periodic boundary conditions which allows the discrete x-derivatives to be expressed in closed form, leaving only the vertical derivatives to be solved for numerically. The resulting relative errors appear in Fig. 7. This plot shows results obtained using the symmetric half of a Chebyshev–Gauss–Lobatto grid distribution in the vertical (same as in [2]). Fourth- and sixth-order finite difference operators ($r = 5, 7$, respectively) have been used. Comparing these results with the corresponding plots from Bingham and Zhang [2] it is clear that there are only very minor differences and the essential character of convergence is unchanged, approaching asymptotically the expected rates of fourth- and sixth-order. This leaves open the (unlikely) possibility that applying strategy BC + LAPLACE at the lateral boundaries could affect the convergence of the method. To verify that this is not the case, we provide one example of convergence for the full 3D numerical solution in Section 5.1.

We thus conclude that the model converges at the expected rate using both strategy BC and BC + LAPLACE at solid boundaries, but that stability is negatively affected by applying strategy BC at the lateral boundaries.

4.3. Linear dispersion and kinematics

We provide here a short discussion of the accuracy of the model with respect to linear dispersion and internal flow kinematics. This analysis complements that presented in [2] (their Figs. 5–8) by presenting the results for fixed resolution as a function of relative water depth kh. The analysis here is done using the BC + LAPLACE strategy at the bottom boundary, although practically identical results are obtained using strategy BC.

For the linear, periodic solution on a constant depth (21) $\tilde{w}$ and $\tilde{\phi}$ are related by (22) which can be written

$$\frac{\tilde{w}}{k\tilde{\phi}} = \tanh(kh), \tag{23}$$

which is the non-dimensional linear dispersion operator. Thus, by specifying the exact linear $\tilde{\phi}$ as a boundary condition, we use the model to compute the corresponding discrete $\tilde{w}$ and compare the non-dimensional result (the left hand side of (23)) with the exact value (the right hand side of (23)) to obtain the error.

Fig. 8 collects the result of this calculation for several choices of spatial discretization. The top two plots show the dispersion errors using second-order schemes on a uniform grid (*i.e.* the model of [14]) with 50 and 150 points per wavelength and varying numbers of grid points in the vertical. From the first plot it is clear that 50 points per wavelength is not enough to give an accuracy of better than 10^{-3} for any but the very longest waves ($kh < 0.5$). With 150 points per wavelength however, a good limiting accuracy can be achieved. The applicable range of kh (at a fixed N_x) can be extended by increasing the number of points in the vertical.

The lower two plots in Fig. 8 shows similar results using fourth- and sixth-order schemes and a clustered vertical grid based on the symmetric half of the Chebyshev–Gauss–Lobatto (CBL) nodes. This shows that a similar level of accuracy can be achieved by using an order of magnitude fewer points in each direction. Since the extra computational effort per grid point involved in moving from second-order to sixth-order operators is generally a factor of two or less for our solution, 3D problems can be solved roughly a thousand times more efficiently using high-order operators and a clustered vertical grid relative to second-order operators on a uniform grid as was also pointed out in [2].

The accuracy of both linear dispersion and the description of the internal flow kinematics is influenced by the choice of grid distribution in the vertical. To measure the accuracy of the kinematics for a given spatial discretization, we compare the computed potential $\phi_c(z)$ with the exact solution (21) ϕ_e at the vertical computational points using the discrete norm

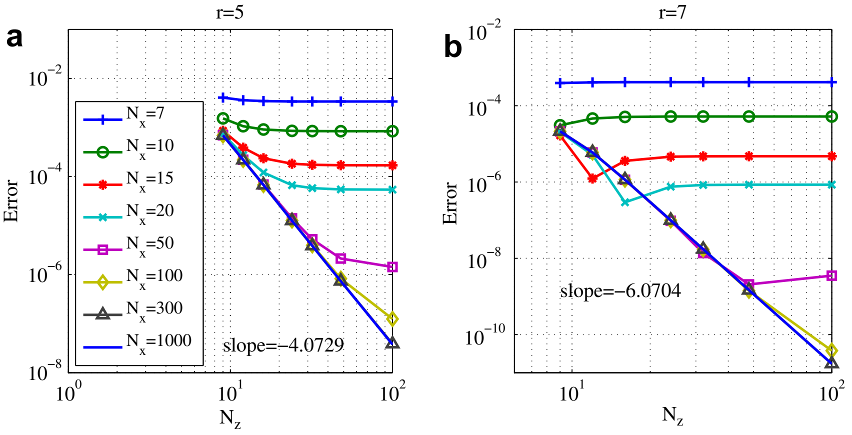


Fig. 7. Convergence of the linear $\tilde{w}$ calculated from $\tilde{\phi}$ at $kh = 4$. Periodic horizontal conditions with strategy BC + LAPLACE at the bottom. The "slope" is the asymptotic slope of the line with $N_x = 1000$.

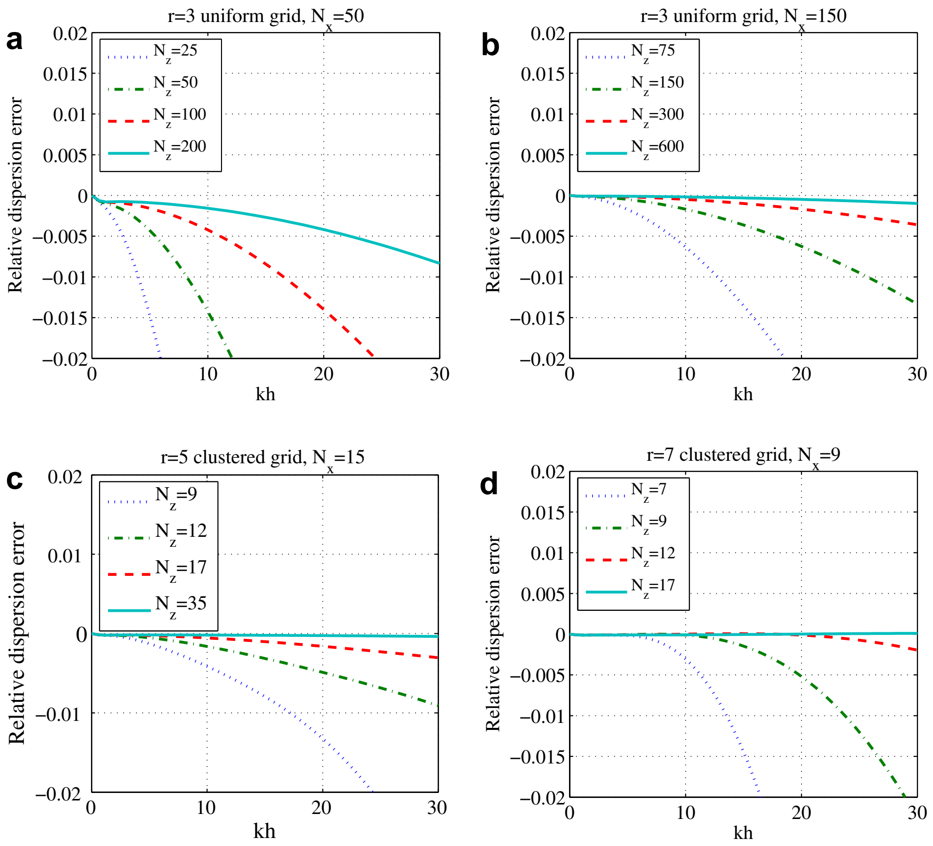


Fig. 8. Dispersion errors as a function of *kh* for different resolutions.

$$\epsilon = -\frac{\|\phi_e - \phi_c\|_2}{\|\phi_e\|_2}, \quad \|\phi\|_2 \equiv \sqrt{\sum_{i,j} |\phi_{ij}|^2}. \tag{24}$$

This is illustrated in Fig. 9 for uniform, Chebyshev–Gauss–Lobatto and Legendre–Gauss–Lobatto (LGL) nodes, respectively, where the symmetric half of the grid points are employed in the two latter cases. The best balance between accuracy in dispersion and internal kinematics is obtained using the CBL nodes, so we adopt this vertical grid as our default.

4.4. Energy conservation and nonlinear stability

We conclude this section with a brief discussion of the energy conservation and nonlinear stability properties of the model. For the linear problem, energy conservation can be inferred from the analysis of Bingham and Zhang [2, Section 4, Figs. 5 and 6]. Since the energy in a wave is proportional to the amplitude squared, the amplitude errors shown there represent the square root of the error in energy conservation per wave period as a function of resolution.

For the nonlinear problem, we here consider the behavior of the model under the ideal circumstances of a 2D periodic problem on a constant depth, while in Section 5 a more practical 3D example is given. Under the assumptions of a potential flow, the total energy in a wave can be expressed as

$$E = \frac{\rho}{2} \mathcal{S}_0 (\tilde{\phi}\eta_t + g\eta^2) dx\, dy, \tag{25}$$

where $\mathcal{S}_0$ is the still water free surface. The first term represents the kinetic energy and the second the potential energy measured with respect to the fluid at rest. This form of the kinetic energy is obtained by an application of the divergence theorem to the original volume integral expression, subsequently invoking the Laplace equation and the boundary conditions.

Highly accurate nonlinear, periodic and constant depth waves computed using the method of Fenton [8] will be used to check energy conservation. We consider a deep water wave with $kh = 2\pi$ at a height which is 90% of the stable limit, $H/L = 0.1273$. This solution provides the initial conditions for η and $\tilde{\phi}$ in the 2D numerical model with periodic lateral boundary conditions. The solution is then allowed to propagate for ten wave periods. Sixth-order finite difference operators are used and the Courant number is fixed at $C_r = 0.5$. Different resolutions N_x and N_z are considered and the influence of the

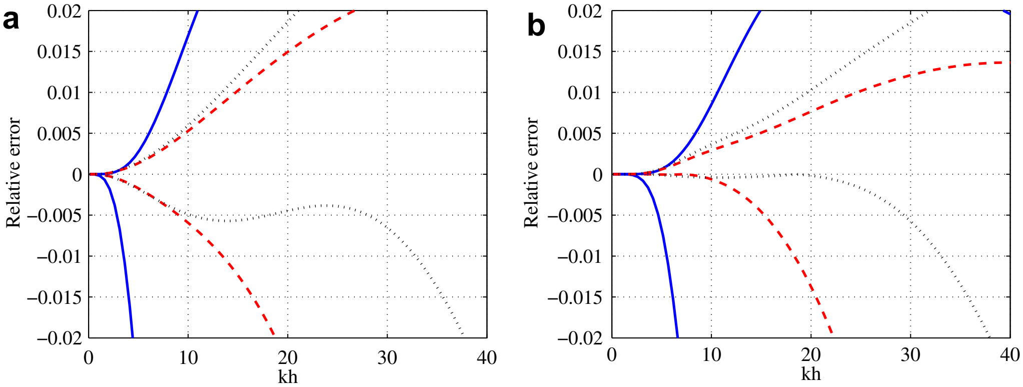


Fig. 9. Relative error in linear dispersion (negative axis) and internal kinematics (positive axis) as a function of kh for different vertical grid distributions and order of numerical scheme for well-resolved free surface waves. (a) Fourth-order scheme and (b) sixth-order scheme. Grid employed with $N_z = 9$ nodes in the vertical and are either uniform (full line), CBL nodes (dashed) and LGL nodes (dotted).

GMRES convergence tolerance is also investigated. The energy is monitored by evaluating (25). For this periodic problem, a spectrally accurate evaluation of the integral is obtained by forming the argument at each grid point in the domain and taking the zero frequency component of it's Fourier transform.

The results are collected in Fig. 10. Fig. 10(a) shows a snapshot of the initial free surface elevation at $t = 0$ along with the final result at $t = 10T$ using a resolution of $N_x = 32$ and $N_z = 9$. For these highly nonlinear periodic problems, the calculations generally break down at some point if the simulation is run long enough. We attribute this to aliasing errors which tend to accumulate at high frequencies eventually leading to instability. The growth of the instability is slow however and can be prevented by the occasional application of a low-pass filter to η and $\tilde{\phi}$. The cleanest way to do this is to apply a top hat filter

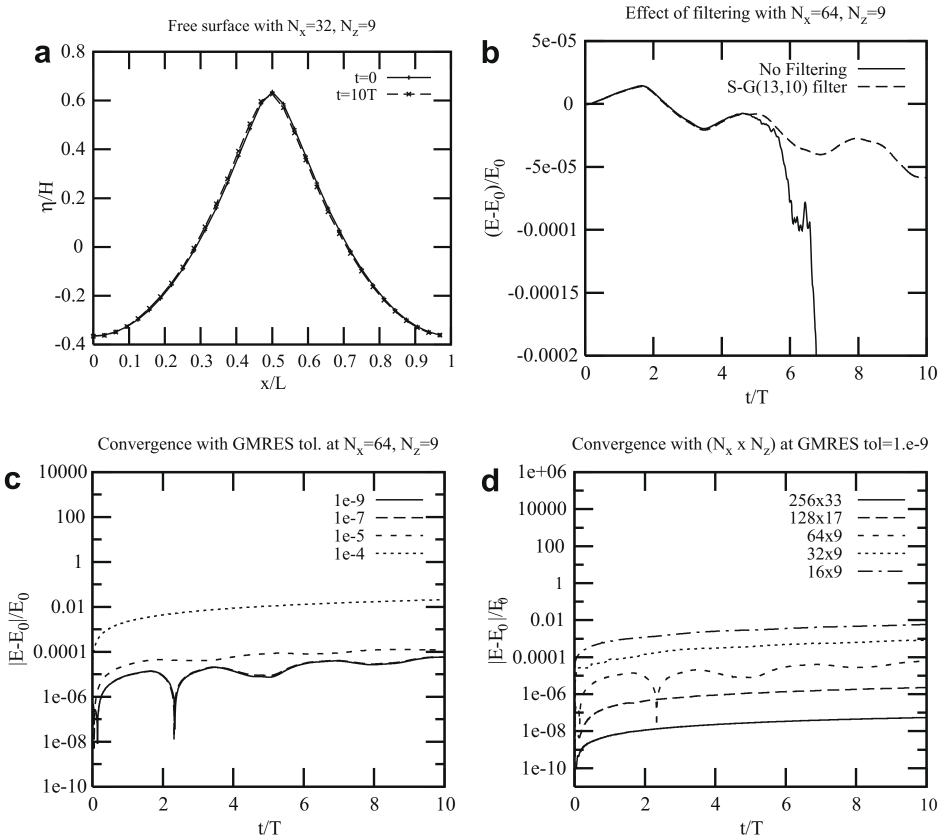


Fig. 10. Energy conservation for a nonlinear periodic wave at $kh = 2\pi$ with $H/L = 0.1273 = 90\%$ of breaking. Showing the effect of filtering as well as convergence with resolution and relative GMRES tolerance.

in the frequency domain which sets to zero the last 1/3 of the frequency space and leaves the first 2/3 of the space untouched. This is essentially a standard de-aliasing technique for quadratic nonlinear products and it works very well for this problem. For the general applications that we have in mind however, *e.g.* when there are structures in the domain and/or the enclosing boundaries are not rectangular, the required Fourier transform is complicated if not impossible to obtain robustly. Instead, we have chosen to apply high-order Savitzky–Golay type filters [17] which have a similar behavior but are straightforward to apply to logically structured grids.

Fig. 10(b) shows the evolution of the total error in energy during the run using $N_x = 64$ and $N_z = 9$. Curves are shown without filtering and with a 13-point, 10th-order filter applied once per wave period. Clearly this mild level of filtering does not significantly influence the total energy in the wave, until about five to six periods of propagation, after which the unfiltered simulation rapidly breaks down but the filtered solution continues without problems. We thus adopt this filter for all the test cases discussed here and apply it every 1/4 to one wave period, depending on the resolution. Higher resolution leads to less total energy error which seems to require more frequent filtering to maintain.

Fig. 10(c) shows the influence of different choices for the GMRES convergence tolerance using $N_x = 64, N_z = 9$ grid points. For this case convergence is achieved at a tolerance of 10^{-10} and the asymptotic level of accuracy in energy conservation is around 10^{-4} after 10 periods of propagation. Fig. 10(d) shows convergence of the energy conservation with spacial resolution at a fixed GMRES convergence tolerance of 10^{-10}. From these results we conclude that the model can satisfy energy conservation to any desired level of accuracy by refinement of the resolution and the GMRES convergence tolerance. Of more practical interest, we conclude from these results that a resolution of 32×9 grid points using sixth-order operators and a $C_r = 0.5$ leads to highly accurate solutions and energy losses of less than 10^{-4} per wave period of propagation for highly nonlinear waves in deep water.

The picture is qualitatively the same in shallow water, although due to the character of these waves more horizontal and less vertical resolution is required to achieve the same level of accuracy. An example is shown in Fig. 11 which shows a snapshot of a wave at 90% of the breaking limit at $kh = 0.5$, along with the convergence of the energy conservation with grid resolution.

5. 3D applications

To validate the 3D performance and implementation of the model described and analyzed in 2D above, we consider the following three test cases: (1) periodic, linear standing waves in a rectangular tank of constant depth; (2) highly nonlinear monochromatic waves propagating at an angle through a rectangular domain; and (3) nonlinear waves shoaling over a semicircular shoal. The first case verifies the expected convergence rate of the model for the linear problem. The second case verifies that the 3D model is consistent with the 2D analysis of Section 4. The final case tests all the features of the fully nonlinear solution on a variable depth fluid.

5.1. Linear standing waves in a rectangular tank

The exact solution for two-dimensional linear standing waves in a rectangular tank can be expressed as

$$\begin{aligned} \eta(x,y,t) &= \frac{H}{2}\cos(\omega t)\cos(k_x x)\cos(k_y y), \\ \phi(x,y,z,t) &= -\frac{H}{2}c\frac{\cosh(k(z+h))}{\sinh(kh)}\sin(\omega t)\cos(k_x x)\cos(k_y y), \end{aligned} \tag{26}$$

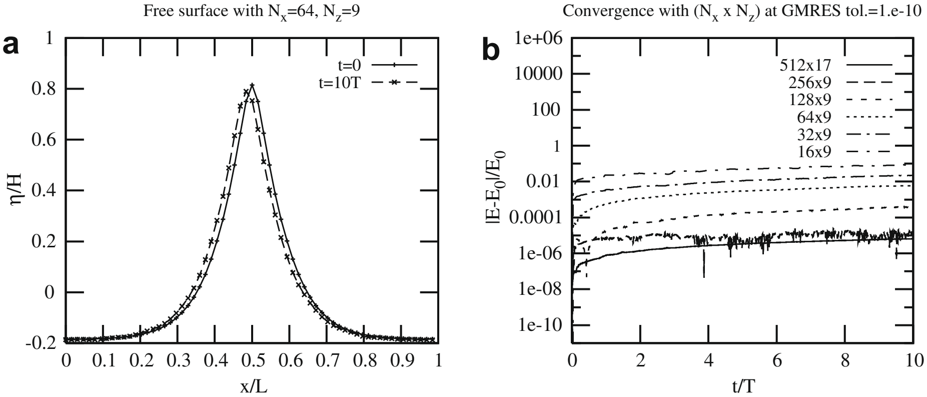


Fig. 11. Energy conservation for a nonlinear periodic shallow water wave at $kh = 0.5$ with $H/L = 0.0529 = 90\%$ of breaking. Snapshot of the surface elevation and convergence of energy conservation with resolution.

where the magnitude of the wave number is expressed in terms of the directional wave number components as $k = \sqrt{k_x^2 + k_y^2}$ which must satisfy the linear dispersion relation $c^2 = g/k\tanh(kh)$, c is the phase velocity, H is the wave height, ω is the angular frequency and h the still-water depth.

A semi-analytical convergence test for the three-dimensional model implementation is carried out to confirm the accuracy of the implementation of linear terms using the analytical solution (26) for a fixed value $kh = 4$. The relative errors are presented in Fig. 12. The test is similar to the two-dimensional convergence test presented in Section 4.2 which can be used in a comparison. The relative dispersion errors are comparable for both the two- and three-dimensional cases for a fixed spatial resolution and fixed rank (*i.e.* order) of finite difference stencils on the clustered grids. This validates the implementation of the linear terms and confirms that the accuracy of the algorithm is not degraded by the presence of a second horizontal dimension.

5.2. Propagation of steep regular nonlinear waves in deep water

To demonstrate that the model is able to solve problems with steep nonlinear waves near the breaking limit in 3D, we generate regular stream function waves [8] with no net mass flux below the wave crest (closed flume assumption) of wave

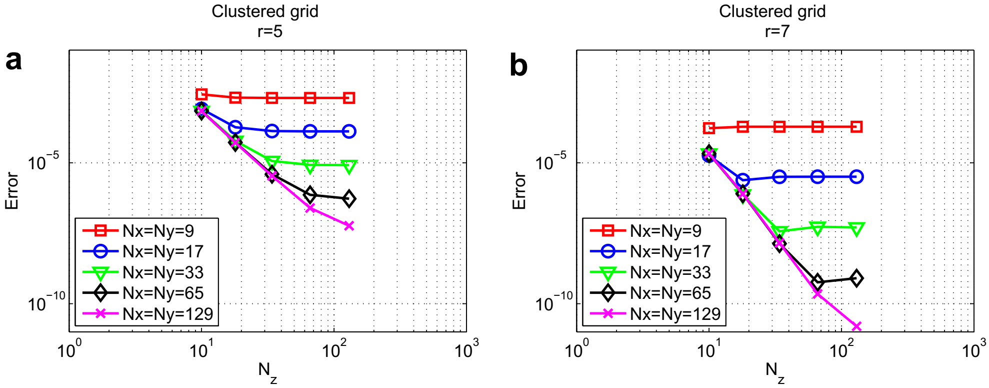


Fig. 12. Convergence of the linear $\tilde{w}$ calculated from $\tilde{\phi}$ at $kh = 4$ for linear standing waves in a square wave tank using the full 3D model. Impermeable wall boundary conditions in the horizontal with strategy BC + LAPLACE at the bottom.

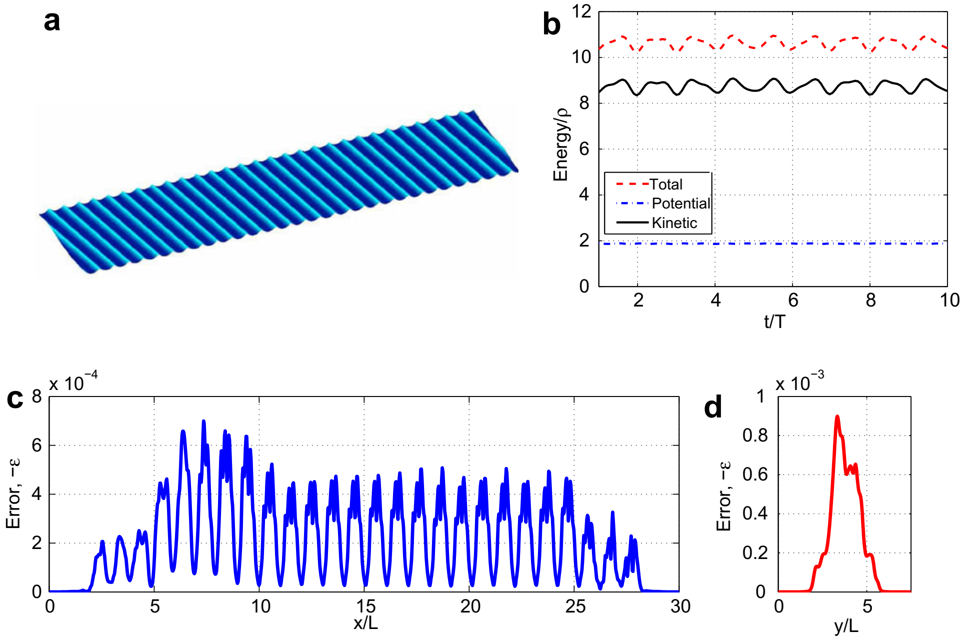


Fig. 13. Steep regular nonlinear waves in deep water having an incident angle of 10° to the x-axis. (a) Snapshot at time $t/T = 10$ of free surface elevation; (b) total energy; (c) and (d) relative two-norm errors along coordinate lines. $kh = 2\pi$. $H/L = 0.1273 = 90\%$ of breaking. Domain is non-periodic and of size $(30 \times 7.5 \times 1)$ m.

length $L = 1$m, nonlinearity $H/L = 90\%(H/L)_{\max}$ and in deep water with $kh = 2\pi$. The domain is of size $32 \times 7.5 \times 1$ m discretized by $1025 \times 257 \times 10 \approx 2.6$ million grid points. The wave makers are positioned along the boundaries and have a size of two wave lengths. The waves are generated at an angle incident to the x-axis corresponding to 10°, thus ensuring that the set-up is three-dimensional. The model is run until a final time of 10 wave periods is reached using sixth-order finite difference operators in space and a Courant number of $C_r = 0.5$.

In Fig. 13(a) a snapshot of the final free surface solution is shown. In Fig. 13(b) a graph for the total energy of the fluid during simulation time is shown. The total energy varies due to the wave incident angle and the visible fluctuating pattern has a timespan matching the wave periods with the fluctuating mean remaining constant implying that energy is conserved. In Fig. 13(c) and (d) the computed relative two-norm errors along the coordinate axes are determined using the analytical solution and are found to be small everywhere.

5.3. Nonlinear waves on a semi-circular shoal

We consider next the experiments of Whalin [20], for which the still water depth is given by (in m)

$$h(\boldsymbol{x}) = \begin{cases} 0.4572, & 0 \leqslant x < 10.67 - \Sigma(y) \\ 0.4572 + \frac{1}{25}(10.67 - \Sigma(y) - x), & 10.67 - \Sigma(y) \leqslant x < 18.29 - \Sigma(y) \\ 0.1524, & 18.29 \leqslant x \end{cases} \tag{27}$$

where $\Sigma(y) = \sqrt{y(6.096 - y)}$. The semi-circular shoal region causes the incoming waves (from the deep part) to refract and focus above and after the shoal. In the original experiment three different wave periods $T = 1, 2, 3$ s and wave heights $H = 0.0390, 0.0150, 0.0136$ m were used for the incident waves. In the numerical model the wave generation and absorption of these fully nonlinear waves are handled using a line source relaxation method due to Larsen and Dancy [13] and highly accurate, periodic solutions for the input signal using the method of Fenton [8]. The waves are generated and absorbed in the region $0 \leqslant x \leqslant 5$ m and absorbed in the region $30 \leqslant x \leqslant 35$ m for each case of different wave period $T = 1, 2, 3$ s. For each test a time step $\Delta t = 0.039$ s was used, and the spatial discretizations of the domain were respectively $257 \times 21 \times (6+1)$ points for the $T = 2, 3$ s cases and $513 \times 21 \times (6+1)$ points for the $T = 1$ s case, which are sufficient to resolve the first three harmonics everywhere. In each test sixth-order finite difference operators were used.

In Fig. 14, a Fast Fourier Transform (FFT) analysis has been carried out for both the computed and measured results for the three different experiments considered using both the direct LU (corresponds to a MG scheme with only one grid level) and

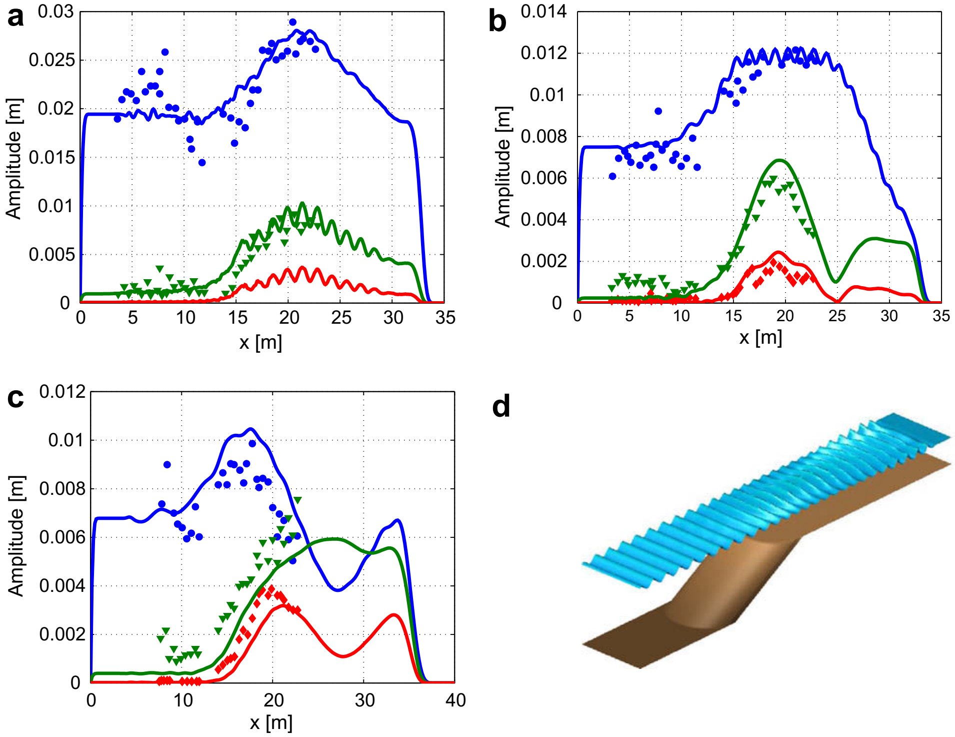


Fig. 14. Harmonic analysis of measured and computed results at shoal center line for the experiments of Whalin [20] for wave periods (a) $T = 1$ s, (b) $T = 2$ s and (c) $T = 3$ s. First, second and third harmonics shown. Snapshot of free surface and bottom for (d) $T = 1$ s at final time $t_f = 50$ s (exaggerated 15 times).

Table 2
Simulation parameters and timing results for experiments of Whalin [20] to reach a final time of $t_f = 50$ s in total 1283 time steps using a formally $\mathcal{O}(\Delta x_*^p)$ accurate scheme (p is order) in space and fourth-order in time for chosen multigrid preconditioning strategy.

T (s)	CPU (hh:mm:ss)	Average iteration	p	Total grids	Preconditioning
1	00:21:33	3.92	6th	1	LU
1	01:01:41	5.80	6th	5	MG-1F(1,1)
2	00:07:57	2.51	6th	1	LU
2	00:21:58	4.35	6th	5	MG-1F(1,1)
3	00:06:02	2.44	6th	1	LU
3	00:19:47	4.02	6th	4	MG-1F(1,1)

the MG-1F(1,1) preconditioning strategies for a numerical scheme which is formally $\mathcal{O}(\Delta x_*^6, \Delta t^4)$. For the multigrid solution five grid levels were used in every case to validate the algorithm. For problems of the given sizes this is choice is sub-optimal and therefore optimal timings for the direct LU have been included in Table 2 where the simulation times and average iteration counts for each test are presented. Time series of the solution at various points in x along the shoal center line at $y = 3.048$ m have been analyzed during the last three wave periods up to the final time of $t_f = 50$ s. Each signal has been decomposed into its harmonic amplitudes via a least-squares fit to a sum of sines and cosines. The first three harmonics are in good agreement with the experimental data. The computed results are also in good agreement with other high-accuracy numerical results from the literature. These computations were run at the Niflheim cluster at Technical University of Denmark (DTU) using an AMD Opteron node with two dual-core processors running at 2.6 GHz.

6. Conclusions

An efficient solution scheme for the 3D Laplace problem for nonlinear waves on a variable depth fluid has been developed. Boundary-fitted coordinates are used to obtain time-constant differential operators which are discretized using arbitrary-order finite difference schemes on a structured, but unevenly spaced grid. The choice of grid distribution in the vertical has an impact on the accuracy of dispersion and internal flow kinematics and the Chebyshev–Gauss–Lobatto points are found to give the best balance between the two. A GMRES iterative solution of the resultant linear system of equations leads to optimal scaling of both the solution effort and the memory use with increasing problem size. The key factors for achieving this result are: a discretization strategy which ensures the satisfaction of both the Laplace equation and the boundary condition at all solid boundaries; and preconditioning of the iterative solution via one multigrid cycle using the linearized, second-order accurate version of the system matrix. Average iteration counts are typically less than 10, and the solution is found to be robust for general nonlinear wave problems, with no need for additional smoothing or filtering over that imposed naturally by the finite difference schemes.

Future efforts are directed toward parallelization of the algorithm to allow larger problems to be treated, and extension of the method to allow more flexibility in representing the geometry of the solid boundaries of the fluid domain. This involves extending the current algorithm to support domain decomposition with conforming blocks communicating through fictitious ghost point layers, and the extension of the scheme to general curvilinear boundary-fitted coordinates in the horizontal directions.

Acknowledgments

We wish to thank the Danish Technical Research Council (STVF grant no. 274-06-0030) for financial support, and the Danish Center for Scientific Computing for supercomputing resources.

References

[1] N.S. Asaithambi, Computation of free-surface flows, J. Comput. Phys. 73 (1987) 380–394.

[2] H.B. Bingham, H. Zhang, On the accuracy of finite-difference solutions for nonlinear water waves, J. Eng. Math. 58 (2007) 211–228.

[3] A. Brandt, Multilevel adaptive solutions to boundary value problems, Math. Comput. 31 (1977) 333–390.

[4] R.K.C. Chan, Finite difference simulation of the planar motion of a ship, in: Proceedings of the International Conference on Numerical Ship Hydrodynamics, Univ. Extension Publ., Berkely, 1977, pp. 39–51.

[5] S.J. DeSilva, R.B. Guenther, R.T. Hudspeth, Irregular points in 2D free surface flows with surface tension for the wavemaker boundary value problem, Appl. Ocean Res. 18 (1996) 293–302.

[6] Frédéric Dias, Thomas J. Bridges, The numerical computation of freely propagating time-dependent irrotational water waves, Fluid Dynam. Res. 38 (12) (2006) 803–830.

[7] D. Dutykh, F. Dias, Dissipative Boussinesq equations, C.R. Mecanique 335 (2007) 559–585.

[8] J.D. Fenton, The numerical solution of steady water wave problems, Comput. Geosci. 14 (1988) 357–368.

[9] Bengt Fornberg, Generation of finite difference formulas on arbitrarily spaced grids, Math. Comput. 51 (184) (1988) 699–706.

[10] H.J. Haussling, R.T. Van Eseltine, Finite-difference methods for transient potential flows with free surfaces, in: Proceedings of the International Conference on Numerical Ship Hydrodynamics, Univ. Extension Publ., Berkely, 1975, pp. 295–313.

[11] A. Iserles, A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, 1996.

[12] C.H. Kim, A.H. ClTment, K. Tanizawa, Recent research and development of numerical wave tanks – a review, Int. J. Offshore Polar Eng. 9 (4) (1999) 241–256.

[13] J. Larsen, H. Dancy, Open boundaries in short wave simulations – a new approach, Coast. Eng. 7 (1983) 285–297.

[14] B. Li, C.A. Fleming, A three-dimensional multigrid model for fully nonlinear water waves, Coast. Eng. 30 (1997) 235–258.

[15] O. Lindberg, A multigrid preconditioner to the laplace problem for nonlinear ocean waves, Master's Thesis, Department of Mechanical Engineering, Technical University of Denmark, 2007.

[16] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.

[17] A. Savitsky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (1964) 1627–1639.

[18] U. Trottenberg, C.W. Oosterlee, A. Schuller, contributions by A. Brandt, P. Oswald, K. Stuben, Multigrid, Academic Press, San Diego, CA, 2001.

[19] Wu-Ting Tsai, Dick K.P. Yue, Computation of nonlinear free-surface flows, Annual Review of Fluid Mechanics, vol. 28, Annual Reviews, Palo Alto, CA, 1996, pp. 249–278.

[20] R.W. Whalin, The limit of applicability of linear wave refraction theory in a convergence zone, Research Report H-71-3, U.S. Army Corps of Engineers, WES, Vicksburg, MI, 1971.

[21] Ronald W. Yeung, Numerical methods in free-surface flows, Annual Review of Fluid Mechanics, vol. 14, Annual Reviews, Palo Alto, CA, 1982, pp. 395–442.

[22] J.A. Zelt, The run-up of nonbreaking and breaking solitary waves, Coast. Eng. 15 (1991) 205–246.