

Technical University of Denmark



## Efficient Solution of the 3D Laplace Problem for Nonlinear Wave-Structure Interaction

**Bingham, Harry B.; Engsig-Karup, Allan Peter**

*Published in:*

ASME 27. International Conference on Offshore Mechanics and Arctic Engineering

*Publication date:*

2008

*Document Version*

Early version, also known as pre-print

[Link back to DTU Orbit](#)

*Citation (APA):*

Bingham, H. B., & Engsig-Karup, A. P. (2008). Efficient Solution of the 3D Laplace Problem for Nonlinear Wave-Structure Interaction. In ASME 27. International Conference on Offshore Mechanics and Arctic Engineering American Society of Mechanical Engineers.

### DTU Library

Technical Information Center of Denmark

---

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

OMAE2008-57384

## EFFICIENT SOLUTION OF THE 3D LAPLACE PROBLEM FOR NONLINEAR WAVE-STRUCTURE INTERACTION

**Harry B. Bingham**

Department of Mechanical Engineering  
Technical University of Denmark  
2800 Lyngby, Denmark  
Email: hbb@mek.dtu.dk

**Allan P. Engsig-Karup**

Department of Mechanical Engineering  
Technical University of Denmark  
2800 Lyngby, Denmark  
Email: apek@mek.dtu.dk

### ABSTRACT

*This contribution presents our recent progress on developing an efficient solution for fully nonlinear wave-structure interaction. The approach is to solve directly the three-dimensional (3D) potential flow problem. The time evolution of the wave field is captured by integrating the free-surface boundary conditions using a fourth-order Runge-Kutta scheme. A coordinate-transformation is employed to obtain a time-constant spatial computational domain which is discretized using arbitrary-order finite difference schemes on a grid with one stretching in each coordinate direction. The resultant linear system of equations is solved by the GMRES iterative method, preconditioned using a multigrid solution to the linearized, lowest-order version of the matrix. The computational effort and required memory use are shown to scale linearly with increasing problem size (total number of grid points). Preliminary examples of nonlinear wave interaction with variable bottom bathymetry and simple bottom mounted structures are given.*

### INTRODUCTION

This paper describes a flexible-order, finite-difference based solution of the exact potential flow problem for nonlinear 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. The application of this basic technique is widespread, and its use for the simulation of unsteady free-surface flows goes back at least to [1]. Related applications were later done by [2–7] among

others. For comprehensive reviews of the nonlinear water wave problem see *e.g.* [8–10].

Ten years ago, [11] developed a multigrid solution to the second-order finite-difference based discretization of this problem on a uniform grid. More recently, [12] extended this approach to allow arbitrary order finite-difference schemes as well as non-uniform grid spacing. They found a significant advantage for high-order schemes on a vertically clustered grid relative to second-order schemes on a uniform grid. In fact, this conclusion should not be surprising since the advantage of high-order schemes for solving this problem was pointed out thirty years ago by [1]. Instead of using multigrid, [12] solved the resultant sparse linear system of equations using the Generalized Minimum Residual (GMRES) method of [13], 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 3D, a direct solution of the preconditioning problem leads to a super-linear scaling of the solution effort with increasing problem size. To retain optimal scaling for 3D problems, we employ multigrid to solve the preconditioning problem instead. 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 built once during the initial set-up, which increases the efficiency of the subsequent solve steps. We also expect this strategy to be more robust than applying multigrid directly to the full high-order, nonlinear system. We conclude that the use of just one multigrid V-cycle, with one pre- and one post-smoothing operation, is most effective as this

provides a nearly equal distribution of effort between the matrix-vector product and the preconditioning. Total iteration counts are typically from five to ten to satisfy a relative residual tolerance of  $10^{-8}$ . At this stage, the cross-over point where multigrid becomes faster than a direct solution of the preconditioning step is found to lie between  $n = 10^4$  and  $n = 10^5$  total grid points, although further optimization of the code may well reduce this number.

In the following sections we outline our implementation of the method, highlighting several discretization issues which are important for stable and robust solution of the problem. We also present some preliminary results for 3D wave-bottom and wave-structure interaction problems. A more complete description of this work can be found in [14].

## FORMULATION

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  $h(\mathbf{x})$  with  $\mathbf{x} = (x, y)$  the horizontal coordinate. The position of the free surface is defined by  $z = \eta(\mathbf{x}, t)$  and the acceleration due to gravity  $g = 9.82m^2/s$  is assumed to be constant.

Assuming an inviscid fluid and an irrotational flow, the fluid velocity  $(\mathbf{u}, w) = (u, v, w) = (\nabla\phi, \partial_z\phi)$  is defined by the gradient of a scalar velocity potential  $\phi(\mathbf{x}, z, t)$ , 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

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

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

which are expressed in terms of the free surface quantities  $\tilde{\phi} = \phi(\mathbf{x}, \eta, t)$  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  $\tilde{\phi}$ , together with the kinematic bottom boundary condition

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

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

$$\partial_z\phi + \nabla h \cdot \nabla\phi = 0, \quad z = -h. \quad (2c)$$

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 direction which maps the solution to a time-invariant domain using the following (non-orthogonal)  $\sigma$ -coordinate transformation

$$\sigma \equiv \frac{z + h(\mathbf{x})}{\eta(\mathbf{x}, t) + h(\mathbf{x})} = \frac{z + h(\mathbf{x})}{d(\mathbf{x}, t)}. \quad (3)$$

The Laplace problem in the transformed computational domain becomes

$$\Phi = \tilde{\phi}, \quad \sigma = 1, \quad (4a)$$

$$\nabla^2\Phi + \nabla^2\sigma(\partial_\sigma\Phi) + 2\nabla\sigma \cdot \nabla(\partial_\sigma\Phi) + (\nabla\sigma \cdot \nabla\sigma + (\partial_z\sigma)^2)\partial_{\sigma\sigma}\Phi = 0, \quad 0 \leq \sigma < 1, \quad (4b)$$

$$(\partial_z\sigma + \nabla h \cdot \nabla\sigma)(\partial_\sigma\Phi) + \nabla h \cdot \nabla\Phi = 0, \quad \sigma = 0, \quad (4c)$$

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

$$\nabla\sigma = \frac{1-\sigma}{d}\nabla h - \frac{\sigma}{d}\nabla\eta, \quad (5a)$$

$$\nabla^2\sigma = \frac{1-\sigma}{d}\left(\nabla^2 h - \frac{\nabla h \cdot \nabla h}{d}\right) - \frac{\sigma}{d}\left(\nabla^2\eta - \frac{\nabla\eta \cdot \nabla\eta}{d}\right) - \frac{1-2\sigma}{d^2}\nabla h \cdot \nabla\eta - \frac{\nabla\sigma}{d} \cdot (\nabla h + \nabla\eta) \quad (5b)$$

$$\partial_z\sigma = \frac{1}{d}. \quad (5c)$$

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

At the structural boundaries of the domain, the flow field must be everywhere parallel to the boundary surfaces, implying that the velocity potential  $\phi$  must satisfy the slip condition

$$(\mathbf{n}, n_z) \cdot (\nabla, \partial_z)\phi = 0, \quad (\mathbf{x}, z) \in \partial\Omega, \quad (6)$$

where  $(\mathbf{n}, n_z) = (n_x, n_y, n_z)$  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, bottom-mounted and aligned with one of the horizontal coordinates. The extension to general boundaries is conceptually straightforward, although it introduces many practical complications related to 3D grid generation. We are now in the process of designing such an extension in order to allow general structures and domain boundaries to be treated.

Having obtained a solution for the function  $\Phi$  in the  $\sigma$ -domain, the physical internal flow kinematics are obtained via the chain rule

$$\mathbf{u}(\mathbf{x}, z, t) = \nabla\phi(\mathbf{x}, z, t) = \nabla\Phi(\mathbf{x}, \sigma, t) + \nabla\sigma\partial_\sigma\Phi(\mathbf{x}, \sigma, t) \quad (7a)$$

$$w(\mathbf{x}, z, t) = \partial_z\phi(\mathbf{x}, z, t) = \partial_\sigma\Phi(\mathbf{x}, \sigma, t)\partial_z\sigma. \quad (7b)$$

## 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) we employ the classical explicit four-stage, fourth-order Runge-Kutta scheme (see *e.g.* [15]). 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  $\eta$  and  $\tilde{\phi}$  are to be evolved. At the structural boundaries of the domain, *i.e.* at the bottom and wall sides, Neumann (specified normal component of velocity) conditions, (4c) and (6), are imposed. The imposition of Neumann-type boundary conditions turns out to be very important to the stability of the resulting method and critical for robust iterative solution of the system using multigrid, as is discussed further below.

For the solution of the transformed Laplace problem (4),  $N_z$  points are defined in the vertical below each horizontal free surface grid point, arbitrarily spaced in  $0 \leq \sigma \leq 1$ . 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  $(x, y, \sigma)$  to be developed in the standard way using Taylor series expansion (see *e.g.* [16]). Thus there are a total of  $2N_x$   $x$ -derivative,  $2N_y$   $y$ -derivative and  $2N_z$   $\sigma$ -derivative schemes to be computed and stored during the initial set-up stage.

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.

### Spatial Discretization Issues

For the spatial one-dimensional first- and second-derivatives in each coordinate,  $r = \alpha + \beta + 1$  points are used where  $\alpha$  indicates the number of points in the positive coordinate direction, and  $\beta$  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 [11] and [12], 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.

Along Neumann-type boundaries this exchange tends to weaken stability, and the effect is accentuated by shallow water physics, large bottom gradients, and high anisotropy in the grid. A more robust strategy is to ensure the satisfaction of both the Laplace equation and the boundary condition at all boundary points. We do this by introducing extra 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  $\eta$  and  $\phi$  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 below.

Having developed all one-dimensional derivative operators, including boundary conditions, second-derivatives are obtained by successive application of these operators. In the case of a direct solution where a matrix must be formed, second-order operators are obtained by matrix multiplication of the discrete 1D operators, which gives 2D stencils of at most  $r^2$  points. In this way, all derivatives are formally accurate to  $O(\Delta x_*^{r-1})$  where  $\Delta x_*$  is the maximum grid spacing in either  $x$ ,  $y$  or  $\sigma$ . This approach also leads naturally to a stable treatment of structural corners placed inside the fluid domain.

The result of this discretization procedure can be expressed as the rank  $n = N_x N_y N_z$  linear system of equations

$$A\Phi = b, \quad (8)$$

where  $A$  is the coefficient matrix,  $\Phi$  a vector of the unknown potential values at each grid point, and  $b$  a vector holding zeros, except at those points corresponding to inhomogeneous boundary conditions.

For the solution of (8) 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.

### Linear Stability Analysis on a Constant Depth

The above described discretization procedure can be analyzed for stability in the standard way using the 2D, linearized problem on a constant depth. Linearizing based on small amplitude waves we can express the time-stepping problem in the form

$$\frac{\partial}{\partial t} \begin{bmatrix} \tilde{\phi} \\ \tilde{\eta} \end{bmatrix} = \begin{bmatrix} 0 & -g \\ J_1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\phi} \\ \tilde{\eta} \end{bmatrix} \quad (9)$$

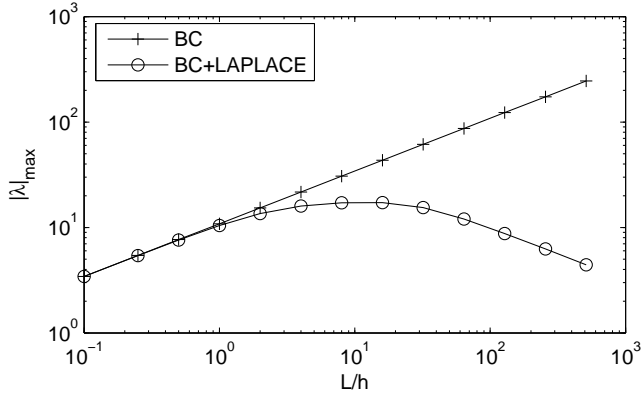


Figure 1. MAXIMUM EIGENVALUES OF THE JACOBIAN MATRIX FOR THE TWO DISCRETIZATION SCHEMES ALONG SOLID BOUNDARIES.

where for the linear problem  $\tilde{\phi} = \phi(x, 0, t)$ , and we have solved explicitly for the vertical free-surface velocity to get

$$\tilde{w} = J_1 \tilde{\phi}, \quad J_1 = D_z^0 A^{-1} P^0. \quad (10)$$

Here  $D_z^0$  is the matrix operator which takes the vertical derivative of  $\phi(x, z)$  on  $z = 0$  and  $P^0$  is the 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  $J_1$  and hence the complete Jacobian matrix in (9). 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.

As discussed above, there are two strategies for imposing the kinematic boundary conditions along solid boundaries: 1) Replace the Laplace equation at the boundary points with the kinematic boundary condition (as was done by [11] and [12]), which we will refer to as strategy “BC”. 2) Impose both the Laplace equation and the boundary condition at all boundary points by adding ghost points outside of the fluid boundaries, which we will refer to as strategy “BC+LAPLACE”.

Figure 1 plots the maximum magnitude of the eigenvalues of the Jacobian for these two boundary condition strategies as functions of the relative water depth  $h/L$  where  $L$  is the domain length. This plot has been made using  $N_x = 17$  and  $N_z = 9$ , but the picture is unchanged for different numbers of grid points. This shows that the two methods have the same stability requirement in deep water but they are dramatically different in shallow water. The eigenvalues of the discrete system are very nearly pure imaginary, so stability under fourth-order Runge-Kutta time-stepping is governed by  $\Delta t |\lambda|_{max} < 2\sqrt{2}$ , (*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 ten 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.

Only the treatment of the enclosing vertical wall boundary conditions influences this stability analysis, and the picture is the same whether the bottom boundary condition is implemented using strategy BC or BC+LAPLACE. Gauss-Seidel iteration (and hence multigrid) on the other hand, turns out to be sensitive to the treatment of the bottom boundary condition as is discussed further below.

### Imposing the Kinematic Bottom Boundary Condition

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

$$\frac{a_j}{2\Delta\sigma} (-3\Phi_{j,k} + 4\Phi_{j,k+1} - \Phi_{j,k+2}) + \frac{b_j}{2\Delta x} (\Phi_{j+1,k} - \Phi_{j-1,k}) = 0 \quad (11)$$

where  $a_j = (1 + b_j^2)/[h(x_j) + \eta(x_j, t)]$  and  $b_j = \partial_x h|_{x=x_j}$  are the factors appearing in (4c) and the classical even-grid, second-order schemes have been applied. For strategy BC+LAPLACE shown in Figure 2 (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

$$\frac{a_j}{2\Delta\sigma} (\Phi_{j,k+2} - \Phi_{j,k}) + \frac{b_j}{2\Delta x} (\Phi_{j+1,k+1} - \Phi_{j-1,k+1}) = 0. \quad (12)$$

For strategy BC, the Laplace equation (4b) is imposed one grid point above the bottom boundary as shown in Figure 3 (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 Figure 3 (b). Equation (12) 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 stencil shown in Figure 3 (c). The full expression is again rather lengthy, but in the limit of

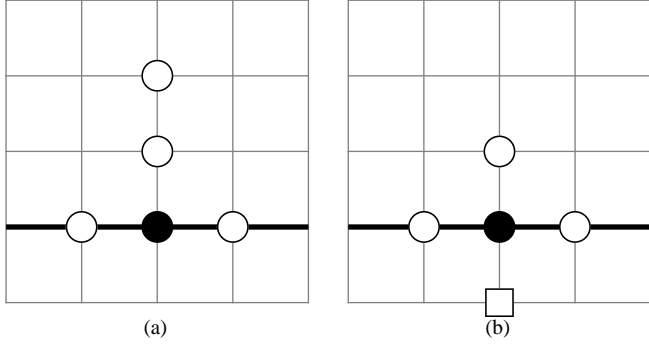


Figure 2. DISCRETIZATION OF THE BOTTOM BOUNDARY CONDITION. (a) STRATEGY BC, (b) STRATEGY BC+LAPLACE. SQUARES INDICATE GHOST POINTS, THE EQUATION IS IMPOSED AT THE FILLED CIRCLE.

a linear problem on a constant depth the bottom equation, after the elimination procedure, simplifies to

$$\frac{1}{\Delta x^2} \Phi_{j-1,k} - 2 \left( \frac{1}{\Delta x^2} + \frac{1}{h^2 \Delta \sigma^2} \right) \Phi_{j,k} + \frac{1}{\Delta x^2} \Phi_{j+1,k} + \frac{2}{h^2 \Delta \sigma^2} \Phi_{j,k+1} = 0 \quad (13)$$

where the index  $jk$  corresponds to the filled dot in 3 (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 linear waves on a constant depth. Gauss-Seidel iteration is thus guaranteed to be convergent in this limit [17]. Although a non-zero bottom slope will generally work to reduce the diagonal dominance in the system, we have so far not found any divergent examples regardless of the physical parameters and/or the grid anisotropy [14]. On the other hand, the exchange of the Laplace equation for the bottom boundary condition inherent in strategy BC, 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.

### MULTIGRID PRECONDITIONING

With left preconditioning, our GMRES iterative solution of the Laplace problem is written symbolically as

$$A_2^{-1} \{A\Phi = b\} \quad (14)$$

where  $\Phi$  is the solution vector,  $b$  the right hand side,  $A$  the nonlinear flexible-order system matrix and  $A_2$  the linearized, second-

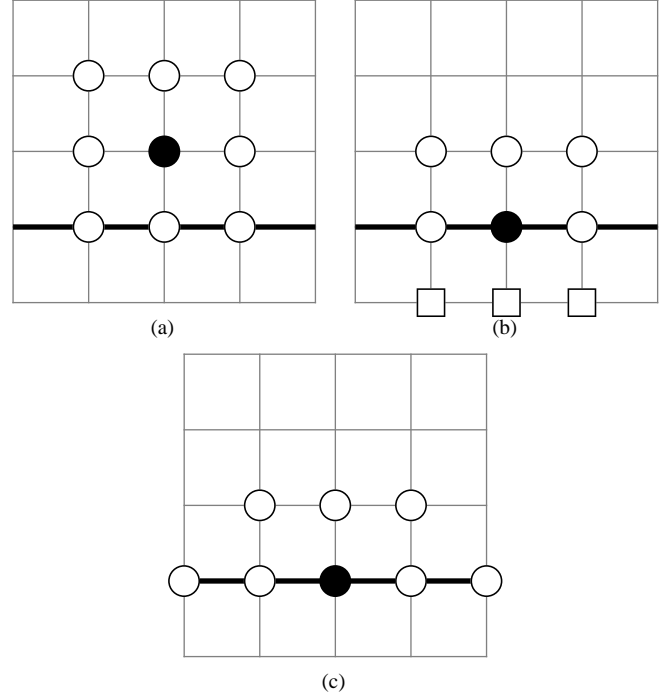


Figure 3. DISCRETIZATION OF THE LAPLACE EQUATION NEAR THE BOUNDARY. (a) STRATEGY BC. (b) STRATEGY BC+LAPLACE. (c) STRATEGY BC+LAPLACE AFTER ELIMINATION OF THE GHOST POINTS.

order accurate version of the system matrix used for preconditioning. Starting with the initial guess of the solution  $\Phi_0$  (obtained using linear extrapolation from the previous two time-steps) the initial residual is computed,  $r_0 = A\Phi_0 - b$ . 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

$$A_2 u_m = r_m, \quad (15)$$

where  $m$  indicates the iteration number.

We apply geometric multigrid [17, 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  $\{G^k : G^k \in \Omega_\sigma, k = 1, 2, \dots, K\}$  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

$$P^k : u^{k-1} \rightarrow u^k, \quad R^k : u^k \rightarrow u^{k-1}, \quad k = 2, \dots, K. \quad (16)$$

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 built 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 built as the product of the fine grid operators  $A_2^{k-1} = R^k A_2^k P^k$  [17]. 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.

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

$$\int_{\Omega_f^k} \Phi^k(\mathbf{x}, \sigma) d\Omega^k = \int_{\Omega_f^k} R^k \Phi^k(\mathbf{x}, \sigma) d\Omega^k, \quad (17)$$

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 as discussed above. 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.

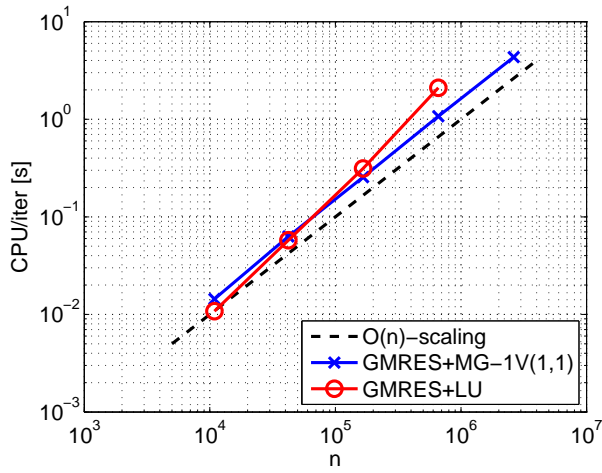
The prolongation operators are determined using bi-linear interpolation from the coarse to the fine grid for standard coarsening, and linear interpolation for semi-coarsening. Thus, if the fine and coarse grid points coincide along one of the coordinate directions, then direct injection is used in that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

The Gauss-Seidel method is used as the basic iterative scheme (the smoother) on which the multigrid solution is built. A careful study of the convergence properties of different flavors of Gauss-Seidel (point, line, red-black, etc.) for this problem using the two bottom boundary discretization strategies has been carried out and is reported in [14, 19]. The conclusion of this study is that no form of Gauss-Seidel iteration is robust for

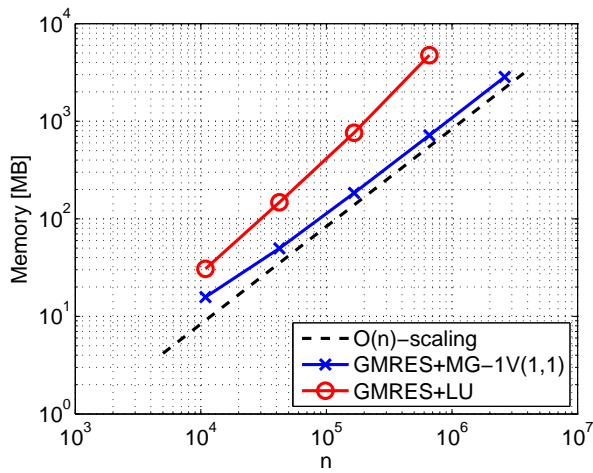
strategy BC, while all forms are robust and convergent for strategy BC+LAPLACE. No significant difference in convergence rates was found between the different flavors, so we use point Gauss-Seidel as it is more efficient. Different multigrid strategies, *e.g.* V-, W-, and F-cycles, with different numbers of pre- and post-smoothing operations (*i.e.* on the way down or up the tree of grid levels respectively) were also evaluated in those reports. The conclusion was that one V-cycle with one post- and one pre-smoothing was most efficient and we have thus adopted this strategy. (A V-cycle involves one pass down and then up the tree of grid levels for each multigrid iteration.)

For the moment, we restrict  $N_x$  and  $N_y$  to be powers of two, but the extension to allow arbitrary combinations of  $N_x$  and  $N_y$  is straightforward, consisting of one special coarsening step to bring these values down to the nearest power of two. Since sufficient accuracy is generally obtained by using  $O(10)$  grid points in the vertical, typical applications of the model will have 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 lines, 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. This 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.

Figure 4 plots the CPU time per iteration and the total RAM memory use vs.  $n$  on a log-log scale for a test problem which has been designed to exercise all the features of the model. For this test case, we fix the length of the domain in both horizontal directions to be  $L$  and let the bottom vary smoothly in both directions such that local relative water depths are in the range  $\pi \leq kh \leq .5$ . The initial condition is a 2D linear standing wave with  $k_x = k_y = 2\pi/L$  and a height such that it would be linear at the deep end but significantly nonlinear at the shallow end. The Courant number  $C_r = \Delta t / \Delta x = 0.5$  is held fixed and the temporal integration is carried out over 100 time steps. Figure 4 shows results using both multigrid and a direct factorization to solve the preconditioning problem. Multigrid can be seen to give an ideal  $O(n)$ -scaling of both CPU time and memory use, while the direct solution scales super-linearly. The break-even point where multigrid preconditioning becomes more efficient than the direct method, can be seen to be approximately  $n = 6 \cdot 10^4$ , although further optimization/improvement of the multigrid preconditioning could well reduce this number. These calculations were made on one AMD Opteron node with two dual-core processors running at 2.6 GHz



(a)

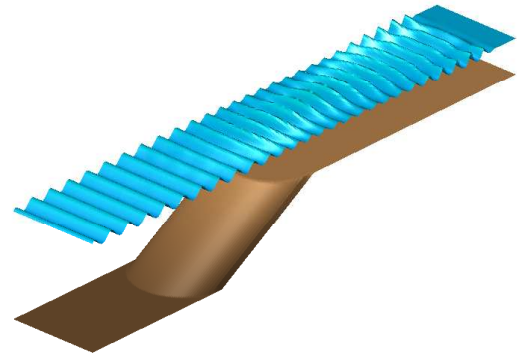


(b)

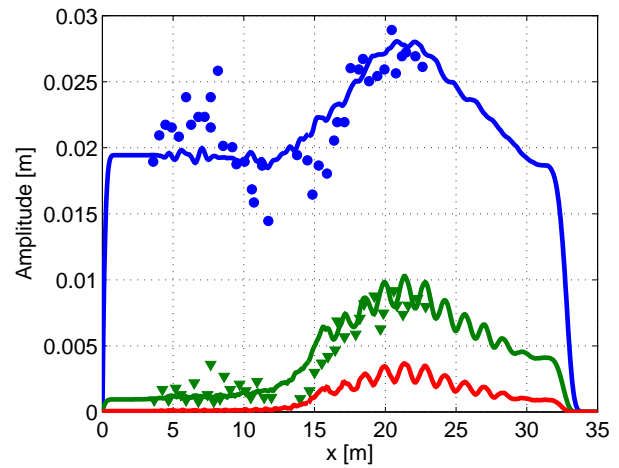
Figure 4. SCALING OF (a) COMPUTATIONAL EFFORT, AND (b) RAM MEMORY USE.

## APPLICATIONS

Some sample applications of the method are provided here. First we consider the experiments of [20] for waves shoaling on a semi-circular bar. A snapshot of the calculations after a steady-state has been reached is shown in Figure 5(a). The bar region causes the waves (incoming from the left) to refract and focus, accumulating bound harmonics which are released on the shelf. Three cases are considered with wave periods of  $T = 1, 2, 3$ s and incident wave heights of  $H = 0.039, 0.015, \& 0.0136$  meters. In the numerical model the wave generation and absorption of these fully nonlinear waves are handled using a relaxation method due to [21] together with the highly-accurate, periodic solutions of [22]. The waves are generated in the region  $0 \leq x \leq 5m$  and absorbed in the region  $30 \leq x \leq 35m$  for cases



(a)



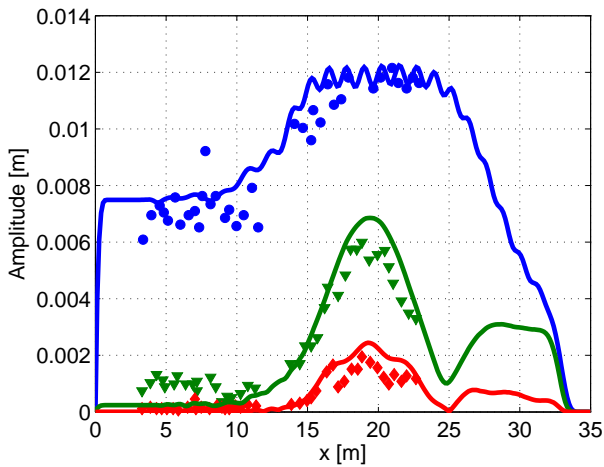
(b)

Figure 5. WAVES SHOALING ON A CIRCULAR BAR COMPARED WITH THE EXPERIMENTS OF [20],  $T = 1$ s CASE. (a) SNAPSHOT AT  $t = 50$ s (EXAGGERATED 15 TIMES), (b) 1st, 2nd & 3rd HARMONICS ALONG THE CENTER LINE.

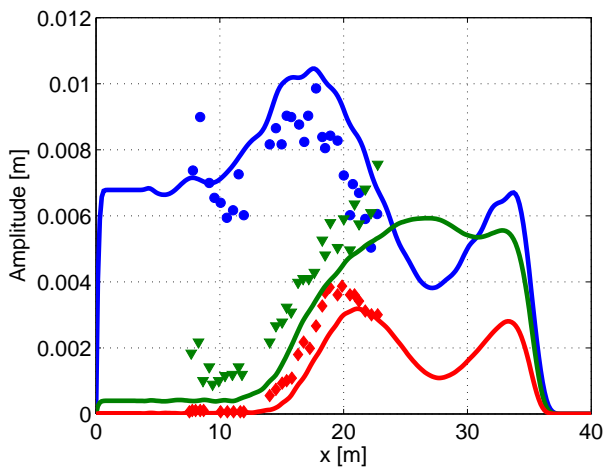
with  $T = 1$  and  $T = 2$ s and  $30 \leq x \leq 40m$  for  $T = 3$ s. The generation zone also absorbs any reflected waves returning towards the wave maker. The time step was set to  $\Delta t = 0.03$ s, and the spatial domain discretized with  $257 \times 33 \times 10$  points for the  $T = 2, 3$ s cases and  $513 \times 33 \times 10$  points for the  $T = 1$ s case. These resolutions are sufficient to resolve the first three free harmonics everywhere. The vertical grid was stretched to cluster points towards the free-surface, and fourth-order discrete operators were used.

Figures 5(b) & 6 show a harmonic analysis of the computed and measured time series along the tank center line. This has been performed by making a least squares fit to a sum of sinusoids at the first 6 harmonic frequencies. The first three harmon-





(a)



(b)

Figure 6. EXPERIMENTS OF [20] FIRST 3 HARMONICS FOR WAVE PERIODS (a)  $T=2s$ , (b)  $T=3s$ .

ics can be seen to be in good agreement with the experimental data. The computed results are also in good agreement with other high-accuracy numerical results from the literature.

A preliminary example of a problem including a structure is given in Figure 7 which shows linear diffraction of waves with length  $L$  through a gap of width  $b$  in an infinitely thin and long breakwater. The calculations are compared to the theoretical result from [23] which are based on the theory of [24]. A snapshot of the solution after steady-state is shown in the top plot, while the bottom plot shows the contours of the amplification factor in the shadow region behind the breakwater. The red dashed curve on this plot indicates the start of the wave absorption zone and

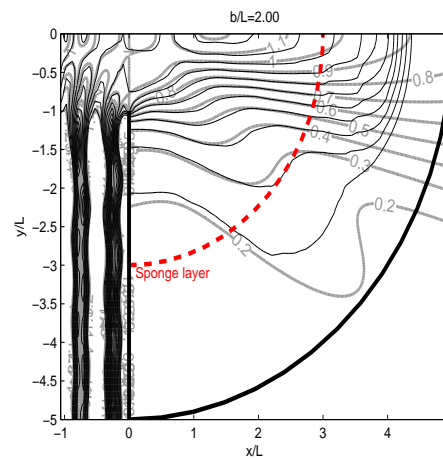
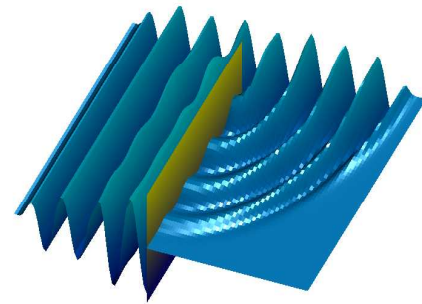


Figure 7. LINEAR GAP DIFFRACTION THROUGH AN INFINITELY THIN BREAKWATER. TOP: SNAPSHOT, BOTTOM: AMPLIFICATION FACTOR CONTOURS. GREY LINES WITH LEVELS INDICATE THE THEORY.

the dark black line the end of the computational domain. The grey lines with contour levels show the theory, while the black contours show the calculations.

## Conclusions

A flexible-order finite difference solution to the exact potential flow problem for waves interacting with structures on a variable depth fluid has been described and some sample applications given. The model is robust, with no need for filtering or smoothing, and the solution effort scales directly with  $n$  the total number of grid points used. When fourth- or sixth-order accurate discrete operators are applied, only about 10 grid points per free-harmonic wavelength, with 5-10 points in the vertical are required to obtain sufficiently accurate solutions for engineering purposes. Variable grid spacing is also supported to allow the

resolution to be tuned to the local wavelength and/or the presence of structures. Together, these features lead to an efficient analysis tool for many coastal and ocean engineering problems.

At this point the basic solver is in place and we are focused on designing and implementing the extension to a multi-block description of the geometry with boundary-fitted coordinates also in the horizontal directions to allow the introduction of more general structures and/or enclosing boundaries to the fluid domain.

## REFERENCES

- [1] Haussling, H. J., and Van Eseltine, R. T., 1975. "Finite-difference methods for transient potential flows with free surfaces". In Proc. 1st Intl. conf. Num. Ship Hydro.
- [2] Chan, R. K.-C., 1975. "Finite-difference simulation of the planar motion of a ship". In Proc. 1st Intl. conf. Num. Ship Hydro.
- [3] Chan, R. K.-C., and Chan, F. W.-K., 1980. "Numerical solution of transient and steady free-surface flows about a ship of general hull shape". In Proc. 13th Symp. on Naval Hydrodynamics.
- [4] Asaithambi, N. S., 1987. "Computation of free-surface flows". *J. Comput. Phys.*, **73**, pp. 380–394.
- [5] DeSilva, S. J., Geunther, R. B., and Hudspeth, R. T., 1996. "Irregular points in 2-D free surface flows with surface tension for the wavemaker boundary value problem". *Applied Ocean Res.*, **18**, pp. 293–302.
- [6] Yeung, R. W., and Ananthkrishnan, P., 1997. "Viscosity and surface tension effects on wave generation by a translating body". *J. Eng. Math.*, **32**, pp. 257–280.
- [7] Ananthkrishnan, P., 1999. "Radiation hydrodynamics of floating vertical cylinder in viscous fluid". *J. Eng. Mech.*, **7**, pp. 836–847.
- [8] Yeung, R. W., 1982. "Numerical methods in free-surface flows". *Ann. Rev. Fluid Mech.*, **14**, pp. 395–442.
- [9] Tsai, W.-T., and Yue, D. K.-P., 1996. "Computation of nonlinear free-surface flows". *Ann. Rev. Fluid Mech.*, **28**, pp. 249–278.
- [10] Dias, F., and Bridges, T. J., 2006. "The numerical computation of freely propagating time-dependent irrotational water waves". *Fluid Dyn. Res.*, **38**, pp. 803–830.
- [11] Li, B., and Fleming, C. A., 1997. "A three-dimensional multigrid model for fully nonlinear water waves". *Coastal Engineering*, **30**, pp. 235–258.
- [12] Bingham, H. B., and Zhang, H., 2007. "On the accuracy of finite difference solutions for nonlinear water waves". *J. Engineering Math.*, **58**, pp. 211–228.
- [13] Saad, Y., and Schultz, M. H., 1986. "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM J. Sci. Stat. Comput.*, **7**, pp. 856–869.
- [14] Engsig-Karup, A. P., Bingham, H. B., and Lindberg, O., 2008. "An efficient flexible-order model for 3D nonlinear water waves". *Submitted*.
- [15] Iserles, A. A., 1996. *A first course in the numerical analysis of differential equations*. Cambridge University Press.
- [16] Fornberg, B. A., 1998. *A practical guide to pseudospectral methods*. Cambridge University Press.
- [17] Trottenberg, U., Oosterlee, C. W., and Schüller, A., 2001. *Multigrid*. Elsevier, Ltd., Amsterdam.
- [18] Brandt, A., 1977. "Multilevel adaptive solutions to boundary value problems". *Math. Comp.*, **31**, pp. 333–390.
- [19] Lindberg, O., 2007. "A multigrid preconditioner to the Laplace problem for nonlinear ocean waves". Master's thesis, Technical University of Denmark, Lyngby, Denmark.
- [20] Whalin, R. W., 1971. The limit of applicability of linear wave refraction theory in a convergence zone. Tech. Rep. Res. Rep. H-71, U.S. Army Corps of Engineers, Waterways Expt. Station, Vicksburg, MS, U.S.A.
- [21] Larsen, J., and Dancy, H., 1983. "Open boundaries in short wave simulations - a new approach". *Coastal Engineering*, **7**, pp. 285–297.
- [22] Fenton, J. D., 1988. "The numerical solution of steady water wave problems". *Comput. Geosci.*, **14**(3), pp. 357–68.
- [23] Penny, W. G., and Price, A. T., 1952. "Part I. the diffraction theory of sea waves and the shelter afforded by breakwaters". *Proc. R. Soc. Lond. A*, **244**, pp. 236–253.
- [24] Sommerfeld, A., 1896. "Mathematische theorie der diffraktion". *Math. Ann.*, **47**, pp. 317–374.

## ACKNOWLEDGMENT

This work is made possible by the Danish Research Council for Technology and Production grant no. 274-06-0030, with supercomputing resources made available by the Danish Center for Super Computing.