A high-order finite difference method for nonlinear wave-structure interaction

Harry B. Bingham*, Allan P. Engsig-Karup and Ole Lindberg
Technical University of Denmark,
2800 Lyngby, Denmark
(hbb,apek)@mek.dtu.dk, s022036@student.dtu.dk

This abstract summarises our recent progress in developing a finite difference based numerical solution of the exact three-dimensional (3D) potential flow problem for nonlinear waves and their interaction with fixed or floating structures. A more thorough description of the work including stability and accuracy analysis and with results in two-dimensions (2D) can be found in [2], while here we present some preliminary 3D applications. The initial goal is to solve the problem with high efficiency on a structured domain with limited geometric flexibility. Ultimately we intend to treat more complicated situations by domain decomposition perhaps with an unstructured method applied to blocks containing complex structures (*e.g.* ships).

We express the exact potential flow problem in a Cartesian coordinate system with origin on the still water plane and the z-axis pointing vertically upwards; $\mathbf{x} = [x, y]$ is a horizontal vector and t is time. The fluid domain is bounded by the sea bottom at $z = -h(\mathbf{x})$ and the free-surface at $z = \eta(\mathbf{x}, t)$. The kinematic and dynamic free surface boundary conditions are expressed in terms of the velocity potential and the vertical component of velocity evaluated directly on the free-surface: $\tilde{\phi} = \phi(\mathbf{x}, \eta, t)$, and $\tilde{w} = \frac{\partial \phi}{\partial z}\big|_{z=\eta}$

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

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

Here $\nabla = [\partial/\partial x, \partial/\partial y]$ is the horizontal gradient operator, g the gravitational acceleration and partial differentiation is indicated when the independent variables appear as subscripts. These provide evolution equations for η and $\tilde{\phi}$ to be integrated forward in time from initial conditions, which is done using the classical explicit fourth-order Runge-Kutta method. To obtain the vertical component of velocity $\tilde{w}$ from the known η and $\tilde{\phi}$, requires satisfying the Laplace equation throughout the depth of the fluid along with the kinematic bottom boundary condition:

$$\nabla^2\phi + \phi_{zz} = 0, \qquad -h < z < \eta \tag{3}$$

$$\phi_z + \nabla h \cdot \nabla\phi = 0, \qquad z = -h. \tag{4}$$

Our strategy for efficient solution of this Laplace problem on one structured block begins by assuming that both η and h are single valued functions of $\mathbf{x}$. A sigma transform is then applied in the vertical coordinate to obtain a rectangular, time independent computational geometry and thus avoid the need to re-grid and re-compute the discrete spatial derivative operators at every stage of the time integration. The sigma transform is given by

$$\sigma(\mathbf{x}, z, t) = \frac{z + h(\mathbf{x})}{\eta(\mathbf{x}, t) + h(\mathbf{x})} = \frac{z + h(\mathbf{x})}{d(\mathbf{x}, t)}, \tag{5}$$

*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.

where the total thickness of the fluid layer $d = \eta + h$ has been introduced. This converts the Laplace problem to

$$\nabla^2\phi + \nabla^2\sigma\,\phi_\sigma + 2\nabla\sigma\cdot\nabla\phi_\sigma + \left(\nabla\sigma\cdot\nabla\sigma + \sigma_z^2\right)\phi_{\sigma\sigma} = 0, \quad 0 < \sigma < 1 \tag{6}$$

$$(\sigma_z + \nabla h\cdot\nabla\sigma)\,\phi_\sigma + \nabla h\cdot\nabla\phi = 0 \quad \sigma = 0, \tag{7}$$

where the derivatives of σ involve first and second derivatives of both h and η (see [2]). After solving (6) & (7) for the potential $\phi(\mathbf{x},\sigma)$, the vertical component of fluid velocity on the free surface is given by $\tilde{w} = \frac{1}{d}\,\phi_\sigma|_{\sigma=1}$ which allows (1) & (2) to be stepped forward in time, closing the problem. If the internal kinematics of the flow are desired they can also be computed from $\phi(\mathbf{x},\sigma)$.

To solve the transformed Laplace problem (6) & (7), the rectangular computational domain is discretised using a variable grid spacing in all three directions, but only one stretching is allowed in each direction to keep the grid structured and orthogonal. On this (fixed) grid, r-point, $(r-1)$-order finite difference schemes are developed to approximate the first- and second-derivatives in each direction. This step requires the inversion of an $r \times r$ matrix at every grid point and along one line in each coordinate direction, but this step need only be done once for a given geometry and choice of r. Applying these finite difference operators to the Laplace problem results in a sparse linear system of equations to solve at every stage of the time integration,

$$\mathbf{A}\mathbf{p} = \mathbf{b}. \tag{8}$$

The matrix $\mathbf{A}$ is of order N the total number of grid points used. In 3D, due to the mixed derivative terms, there can be as many as $2r^2$ non-zero entries per row and the sparsity pattern is not especially compact; thus a direct solution can only be contemplated for very small problems. Even in 2D a direct solution becomes quickly prohibitive with an effort which scales super-linearly with N[2]. The sigma transform, the variable bottom, and the non-uniform grid spacing conspire to ensure that the matrix is non-symmetric in general, with possibly large anisotropy. We thus employ the GMRES (Generalised Minimal RESidual) Krylov subspace iterative method to solve the system, with the previous stage solution as the initial guess. The performance of GMRES is very sensitive to the preconditioning, and for this purpose we use the linearised version of the matrix discretised to second-order accuracy, thus $\mathbf{M} = \mathbf{A}_2^0$ where the subscript 2 indicates second-order accurate operators and the superscript 0 indicates that η has been set to zero in all derivatives of σ. A preconditioning operation thus requires solving a linear system of the form

$$\mathbf{M}\mathbf{q} = \mathbf{s}. \tag{9}$$

In 2D, $\mathbf{M}$ has a relatively simple structure. For a grid of $N_x \times N_z$ points and following the z-points first there are three bands along the diagonal and two sets of three centered at a distance of N_z from the main diagonal. A direct factorisation of this matrix with a subsequent back-substitution at each preconditioning step is thus an effective strategy and leads to an optimum scaling of the solution effort (*i.e.* directly with $N = N_x N_z$) as was shown in [2]. Iteration counts to solve the system to a relative tolerance of 10^{-7} are found to be generally less than twenty and typically closer to ten, independently of the problem size, the order of the finite difference schemes used, and the relative water depth.

In 3D, $\mathbf{M}$ has five sets of three bands with the outer bands centered at distances of N_z and $N_y N_z$ from the main diagonal where N_y is the number of grid points in the y-direction. Even with re-ordering, the fill-in required to factor this matrix is substantial

and grows too rapidly with N to allow optimal scaling for a direct solution. This provides motivation to consider a multigrid solution to the preconditioning problem. At this point we have obtained some preliminary results in both 2D and 3D, more complete results and description will appear in a future publication now in preparation [1].

Figure 1 plots (on a log scale) the average CPU time per time-step versus N required to solve a nonlinear standing wave test case. 2D calculations appear on the left and 3D on the right. For this test case we set either a one- or a two-dimensional linear standing wave initial

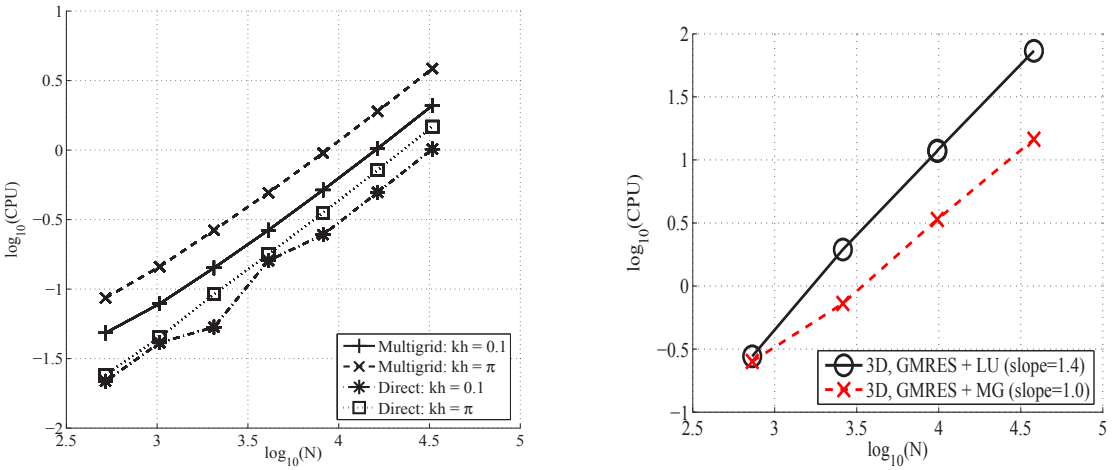

Figure 1: Computational effort per time-step using a direct and a multigrid solution of the preconditioning operation. Left: 2D problem, right: 3D problem, $kh = \pi$.

condition, but with an amplitude such that the wave is mildly nonlinear. The nonlinear problem is then solved for a number of time steps and the CPU time per time step evaluated. These plots are for $kh = \pi$ and 0.1 (k the wave number). Fourth-order operators are used in $\mathbf{A}$ as these were suggested to be optimal by [2]. These results are for an "engineering"-type test where N_z is held fixed at nine points (which was found to be sufficiently accurate by [2]) while only N_x and N_y are increased to increase the problem size. This reflects the way the code will be used in practise. Also relevant is a "convergence"-type test where the number of grid points is increased in all directions as was done in [2]. The engineering test is advantageous for the direct method since the band width of the preconditioning matrix does not increase with N in 2D, and increases relatively slowly in 3D. At the same time this test is particularly difficult for the multigrid method since the problem becomes more and more anisotropic as N increases. To obtain robust multigrid performance in shallow water, and in general for highly anisotropic cases, we use the multiple semi-coarsened grid method. As can be seen in Fig. 1, the direct method is indeed faster in 2D but in 3D the scaling of the direct method becomes super-linear and multigrid is faster, retaining an optimal scaling which is directly proportional to N.

As mentioned above, our long-term strategy is to develop highly efficient solutions on structured blocks on which the bottom variation is arbitrary but horizontal boundaries are uniform in the vertical and single valued functions of the horizontal coordinate. An obvious extension to the method as presented above which allows curved (and moving) horizontal boundaries is to apply sigma-transforms in all three directions. This increases the complexity of the Laplace problem on paper but does not significantly affect the coding or solution effort. For more complicated structures or boundaries, it may be possible to develop special blocks on which an unstructured method is applied and this is the topic of ongoing research. At this point we are still in the initial phases of the 3D implementation,

but we have made some simple tests with piecewise rectangular bottom mounted structures two of which are shown in Figures 2 and 3. These figures show linear diffraction from a channel and through a gap in an infinitely thin breakwater. Also shown are elevation RAO contours and the exact result for the gap diffraction case. The comparison can be seen to be quite good. More results will be presented at the workshop.

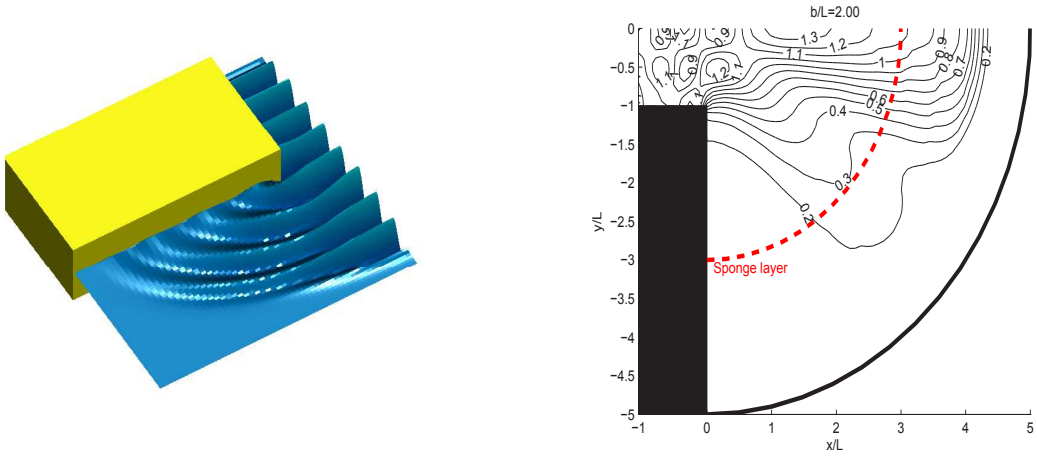

Figure 2: Linear diffraction of a wave of length L from a channel of width b. Left: snapshot. Right: wave amplitude RAO contours. An absorption zone is applied between the dashed line and the black circular line which is the end of the computational domain.

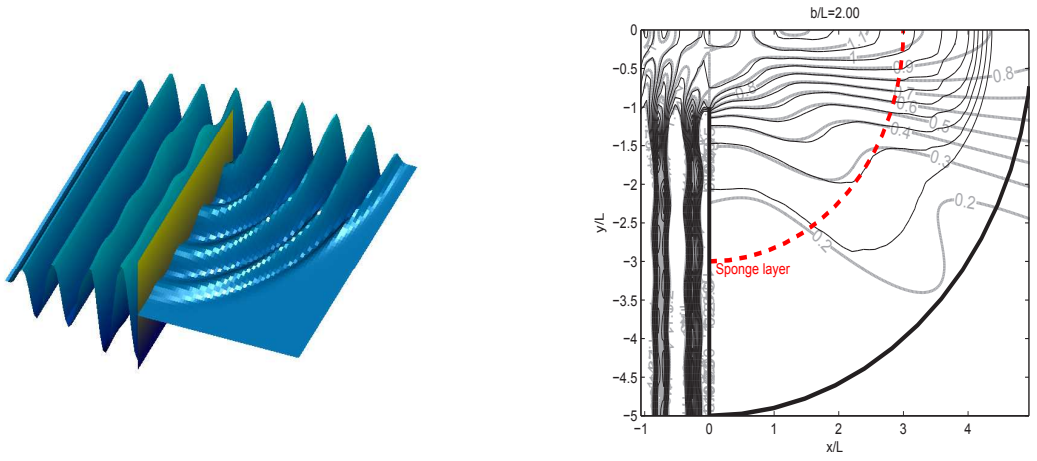

Figure 3: Linear gap diffraction through an infinitely thin breakwater. As in Fig. 2.

References

[1] H. B. Bingham, O. Lindberg, and A. P. Engsig-Karup. Multigrid preconditioning for 3D nonlinear water waves. (In preparation), 2007.

[2] H. B. Bingham and H. Zhang. On the accuracy of finite difference solutions for nonlinear water waves. *J. Engineering Math.*, (in press), 2007.