Boundary-fitted solutions for 3D nonlinear water wave-structure interaction

Allan P. Engsig-Karup
Department of Informatics and Mathematical Modeling
Technical University of Denmark
E-mail: apek@imm.dtu.dk

Harry B. Bingham
Department of Mechanical Engineering
Technical University of Denmark
E-mail: hbb@mek.dtu.dk

Background

This abstract describes our recent progress in developing a robust, efficient, flexible-order finite difference model for nonlinear water waves and their interaction with fixed and floating structures. This builds on work which was presented at the 22nd and 23rd [1, 3] workshops. The goal of this work is a computational tool suitable for large-scale prediction of nonlinear wave-wave, wave-bottom and wave-structure interaction in the coastal and offshore environment.

The method is presented in detail in [2, 4], which includes a stability and accuracy analysis in both two- and three-dimensions, together with a range of validation tests that confirm both the accuracy and the efficiency of the model. As demonstrated in [3, 4] the use of multigrid preconditioning makes it possible to achieve an optimal scaling of the overall solution effort, i.e. the effort scales directly with n the total number of grid points.

The next step in the development of the model is to include support for boundary-fitted, curvilinear physical domains that can be mapped to a computational domain of logically structured blocks. This will allow for the treatment of arbitrarily complex boundary geometries. Procedures for doing this have been established for many years, *e.g.* [7, 6]. As a first-step, we consider one 2D transformation in the horizontal plane to allow treatment of general fixed, bottom-mounted structures extending vertically troughout the depth of the fluid. The bottom typography is also arbitrary. This opens a large class of coastal geometries to a fully nonlinear analysis in the context of a potential flow and up to the point of wave breaking.

Governing equations

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 $z = \zeta(\boldsymbol{x}, t)$ and the gravitational acceleration $g = 9.81m^2/s$ is assumed to be constant.

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

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

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

expressed in terms of the free surface quantities $\tilde{\phi} = \phi(\boldsymbol{x}, \zeta, t)$ and $\tilde{w} = \partial_z\phi|_{z=\zeta}$. 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 ζ, together with the kinematic bottom boundary condition:

$$\begin{aligned} \phi &= \tilde{\phi}, \quad z = \zeta, & \text{(2a)} \\ \nabla^2 \phi + \partial_{zz}\phi &= 0, \quad -h \le z < \zeta, & \text{(2b)} \\ \partial_z \phi + \nabla h \cdot \nabla \phi &= 0, \quad z = -h. & \text{(2c)} \end{aligned}$$

At the structural boundaries of the domain, the flow field must be everywhere parallel to the solid boundary surfaces, implying the no-normal flow condition

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

where $\boldsymbol{n} = (n_x, n_y, n_z)$ is an outward pointing normal vector to the boundary surface $\partial\Omega$.

The boundary-fitted coordinate transformation

We assume that the fluid is bounded by a set of lines drawn on the horizontal plane which define surfaces extending vertically throughout the depth of the fluid. On a section of the horizonatal plane we define a logically structured grid of points (x, y), for example as shown in Figure 1. The transformations between the physical grid and the computational

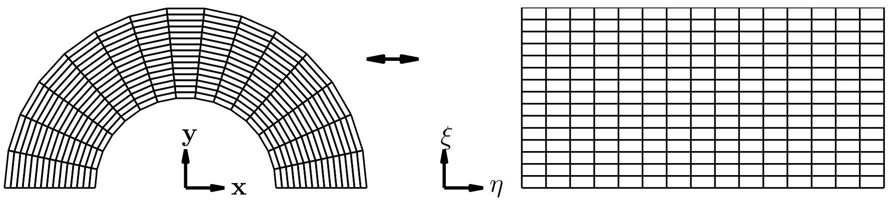

Figure 1: An example of a grid transformation.

unit-spaced square grid are given by

$$[\xi(x, y), \eta(x, y)], \quad \text{and} \quad [x(\xi, \eta), y(\xi, \eta)]. \tag{4}$$

Using the chain rule for partial differentiation, the following covariant linear relationships between the partial derivatives in the physical domain and those in the computational domain can be developed

$$\begin{aligned} \partial_x &= \xi_x \partial_\xi + \eta_x \partial_\eta, \quad \partial_y = \xi_y \partial_\xi + \eta_y \partial_\eta, \\ \partial_{xy} &= \xi_{xy}\partial_\xi + \eta_{xy}\partial_\eta + (\xi_x \eta_y + \xi_y \eta_x)\partial_{\xi\eta} + \xi_x \xi_y \partial_{\xi\xi} + \eta_x \eta_y \partial_{\eta\eta}, \\ \partial_{xx} &= \xi_{xx}\partial_\xi + \xi_x^2 \partial_{\xi\xi} + 2\xi_x \eta_x \partial_{\xi\eta} + \eta_{xx}\partial_\eta + \eta_x^2 \partial_{\eta\eta}, \\ \partial_{yy} &= \xi_{yy}\partial_\xi + \xi_y^2 \partial_{\xi\xi} + 2\xi_y \eta_y \partial_{\xi\eta} + \eta_{yy}\partial_\eta + \eta_y^2 \partial_{\eta\eta}, \end{aligned} \tag{5}$$

where the sub-scripts indicate partial differentiation. This set of operations defines how to determine the usual Cartesian derivatives in the physical space via weighted operations on a Cartesian grid in computational space. Since we are interested in computing all derivatives in the computational space, we now express all derivatives of (ξ, η) by equivalent combinations of derivatives of (x, y). These relations follow from the identity

$$\begin{bmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{bmatrix} \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Rightarrow \quad \begin{bmatrix} \xi_x & \eta_x \\ \xi_y & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix}^{-1} \tag{6}$$

which gives, for example: $\xi_x = y_\eta / J$, $J = (x_\xi y_\eta - x_\eta y_\xi)$. Thus, the transformation weights can all be determined by applying finite difference operators in the computational space to the grid point positions (x, y). Since the weights are solely a function of the geometry of the problem, they need only be determined once on a given fixed physical grid. For the procedure to be robust and accurate however, it is important to avoid singular transformations. Although orthogonality is not required, the closer the grid is to orthogonal the more accurate the approximations will be. A detailed discussion of grid generation can be found in [6].

The transformation discussed above allows all horizontal derivatives to be determined by finite difference approximations on the computational (ξ, η) grid. For the vertical coordinate, we apply the σ-transformation discussed in detail in [2, 4, 1, 3].

$$\sigma \equiv \frac{z + h(\boldsymbol{x})}{\zeta(\boldsymbol{x}, t) + h(\boldsymbol{x})} \equiv \frac{z + h(\boldsymbol{x})}{d(\boldsymbol{x}, t)}. \tag{7}$$

This maps the time-dependent moving free surface boundary to a fixed plane of the computational space, and produces a set of time-dependent weights for the vertical derivatives which are in terms of horizontal derivatives of h and ζ.

With this set of transformation, we avoid any issues associated with re-gridding, and the discrete derivative operators need only be determined once in a pre-processing step.

Numerical solution

A method of lines approach is used for the discretization of the governing equations. For the time-integration of the free-surface conditions (1) the classical explicit fourth-order Runge-Kutta scheme is employed. For the spatial discretization, a grid of $N_x \times N_y$ points is defined on the horizontal xy-plane at which the free surface variables ζ and $\tilde{\phi}$ are to be evolved. For the solution of the transformed Laplace problem, 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 in the transformed domain is thus orthogonal and structured. Choosing r nearby points, allows order $(r - 1)$ finite difference schemes for the 1D first- and second-derivatives in (ξ, η, σ) to be developed in the standard way using Taylor series expansion [5]. The resultant discrete operators are then used to define the coordinate mapping of (5) and thus discretize (2) to solve for ϕ. Nonlinear terms in the free-surface conditions are treated by direct product approximations at the collocation points. At the solid boundaries of the domain, Neumann conditions, (2c) and (3), are imposed using a ghost point technique which is important for the overall robustness of the model as described in detail in [4].

To be able to optimize efficiency of the model for a given problem, the order of the spatial discretization schemes is kept flexible. Thus, two convergence strategies are available, namely, h- and p-adaptivity where either the spatial resolution or the order of the scheme is increased respectively.

A preliminary calculations is shown in Figure 2. This case considers linear waves propagating through a semi-circular channel. Figure 2(a) shows a snapshot of the solution with waves entering the channel from the straight section at the top right and being absorbed in the straight section towards the bottom left. Figure 2 (b) shows the computed wave elevation amplification factor in the curved section. Finally, Figures 2 (c) and (d) compare the computed wave elevation with a semi-analytical solution along the innner and outer channel perimeters respectively. At this point we have validated the method on simple geometries.

The next step is to test the algorithm on more complicated geometries and extend the idea to multiple overlapping blocks. More results will be presented at the workshop.

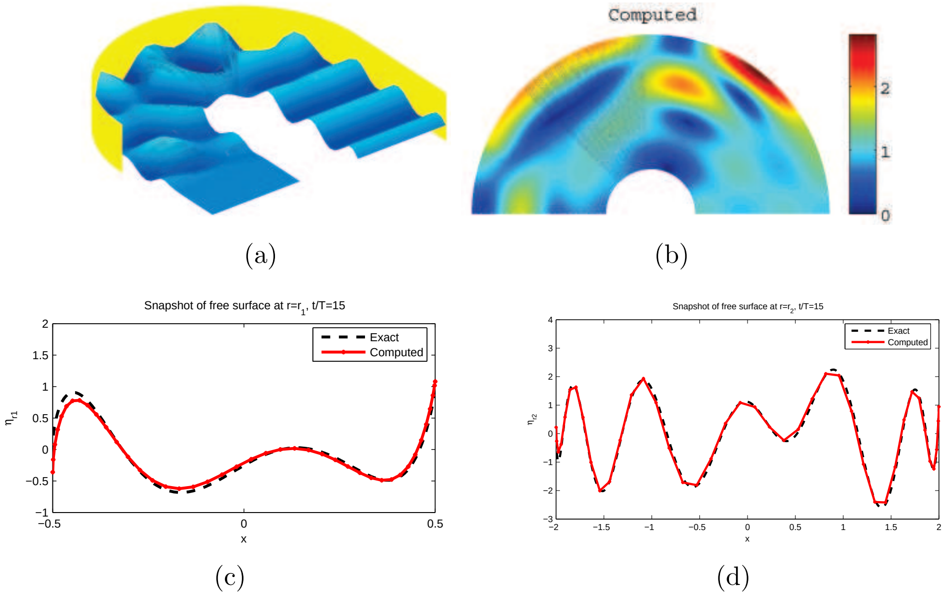

Figure 2: Propagation of waves through a semi-circular channel.

References

[1] H. B. Bingham, A. P. Engsig-Karup, and O. Lindberg. A high-order finite difference method for nonlinear wave-structure interaction. In *Proceedings of the 22nd International Workshop on Water Waves and Floating Bodies*, 2007.

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

[3] A. P. Engsig-Karup, H. B. Bingham, and O. Lindberg. Multigrid preconditioning for efficient solution of the 3d laplace problem for wave-body interaction. In *Proceedings of the 23rd International Workshop on Water Waves and Floating Bodies*, 2008.

[4] A.P. Engsig-Karup, H.B. Bingham, and O. Lindberg. An efficient flexible-order model for 3d nonlinear water waves. *Accepted for publication in Journal of Computational Physics*, 2008.

[5] Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, oct 1988.

[6] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of Grid generation*. CRC Press, 1999.

[7] Joe F. Thompson, Zahir U. A. Warsi, and C. Wayne Mastin. Boundary-fitted coordinate systems for numerical solution of partial differential equations—a review. *J. Comput. Phys.*, 47(1):1–108, 1982.