A high-order and mesh-free computational model for non-linear water waves

Nielsen, Morten Eggert; Fornberg, Bengt; Damkilde, Lars

Published in:
VIII International Conference on Computational Methods in Marine Engineering

Publication date:
2019

Document Version
Other version

Link to publication from Aalborg University

Citation for published version (APA):
Nielsen, M. E., Fornberg, B., & Damkilde, L. (2019). A high-order and mesh-free computational model for non-linear water waves. In R. Bensow, & J. Ringsberg (Eds.), *VIII International Conference on Computational Methods in Marine Engineering: MARINE 2019* International Center for Numerical Methods in Engineering.

A HIGH-ORDER AND MESH-FREE COMPUTATIONAL MODEL FOR NON-LINEAR WATER WAVES

Morten E. Nielsen[*], Bengt Fornberg[†] and Lars Damkilde[‡]

[*] Department of Civil Engineering
Aalborg University
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
e-mail: men@civil.aau.dk

[†] Department of Applied Mathematics
University of Colorado
526 UCB, Boulder, CO 80309, USA
e-mail: bengt.fornberg@colorado.edu

[‡] Department of Civil Engineering
Aalborg University
Thomas Manns Vej 23, 9220 Aalborg Ø, Denmark
e-mail: lda@civil.aau.dk

Key words: mesh-free methods, radial basis function-generated finite differences, high-order, node generation, water waves.

Abstract. In this paper, we present the ongoing developments of a novel computational model for non-linear water waves that aims to provide a suitable framework for wave-structure interaction. The proposed model is based on radial basis function-generated finite differences, which allow for arbitrary and moving boundaries without the use of ghost nodes. In order to take advantage of the mesh-free setting, we propose a node generation strategy, suitable for moving boundaries. Numerical properties of the proposed model are investigated and finally the model is benchmarked. The proposed model is expected to provide a suitable computational framework for wave-structure interaction problems, due to its geometric flexibility and high-order nature.

1 INTRODUCTION

During the latest decades, the interest in computational models for wave-structure interaction has increased, e.g. due to an increased demand for renewable energy. On the basis of this interest, several research projects [1, 2, 3] have been initiated, which seek to improve existing computational models for wave-structure interaction. Overall the improvements concern non-linear interactions, high fidelity fluid models and computational efficiency. Ultimately, the goal is to enhance the computational models that are used for analysis and optimization of offshore structures, such as e.g. floating offshore wind turbines and wave energy converters.

Several models have been proposed for the non-linear water wave problem, see e.g. [4, 5, 6, 7, 8, 9, 10, 11], which are based on different discretizations, ranging from traditional finite differences to spectral elements. The main concerns regarding computational modelling of non-linear water waves, in relation to wave-structure interaction, are (1) temporal stability, (2) accuracy, (3) geometric flexibility and (4) computation speed. This paper deals with the development of a novel computational model that aims to deal with the difficulties stated above, while providing a suitable framework for future coupling with structural models, e.g. the one used in [12, 13]. Specifically, we address (1) and (2) by investigating the use of radial basis function-generated finite differences (RBF-FD). The advantages of RBF-FD are geometric flexibility and high-order approximations [14, 15, 16].

Initially, the non-linear water wave problem is introduced in section 2. Next, the methodology is outlined in section 3, including a node generation strategy for the time-varying computational domain. Numerical properties of the resulting computational model are investigated in section 4, while non-linear stream functions waves are used for benchmarking of the non-linear stability and accuracy in section 5. Finally, some conclusions are stated in section 6.

2 MATHEMATICAL FORMULATION

In this paper, the formulation of the fully non-linear water wave problem is based on potential flow theory and the assumption of no wave breaking to occur. In the following subsections, the governing equation and boundary conditions are presented for the two-dimensional case, which will be used in this paper.

2.1 Interior fluid domain

Non-linear water waves may be modelled by means of potential flow theory, which makes it possible to describe the fluid flow by the scalar velocity potential function $\Phi(\boldsymbol{x}, t)$ found by solving the Laplace equation expressed as

$$\nabla^2 \Phi = 0, \quad \boldsymbol{x} \in \Omega(t), \tag{1}$$

where $\nabla = [\frac{\partial}{\partial x} \ \frac{\partial}{\partial z}]^T$ is the gradient operator. Thus, equation (1) must be fulfilled within the time-varying fluid domain $\Omega(t)$, as illustrated in figure 1.

2.2 Free surface conditions

For the fluid flow to evolve over time, the free surface boundary conditions must be updated over time. In the Lagrangian frame of reference, the kinematic and dynamic free surface conditions can be expressed as

$$\frac{D\boldsymbol{r}}{Dt} = \nabla \Phi, \quad \boldsymbol{x} \in \Gamma_\eta(t), \tag{2}$$

$$\frac{D\Phi}{Dt} = -g\eta + \frac{1}{2}\left(\nabla \Phi \cdot \nabla \Phi\right), \quad \boldsymbol{x} \in \Gamma_\eta(t), \tag{3}$$

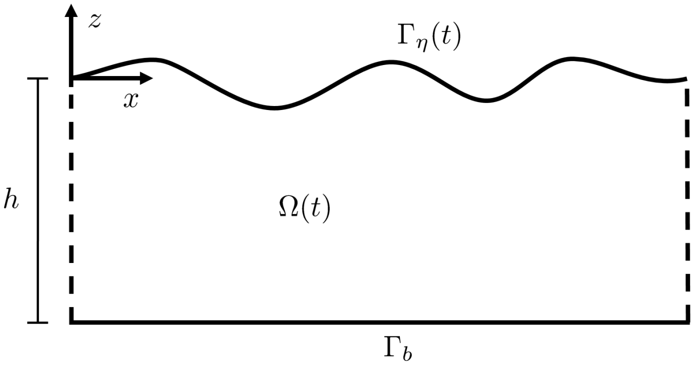


Figure 1: Fluid domain $\Omega(t)$ with specification of the free surface $\Gamma_\eta(t)$ and bottom boundary Γ_b.

where g is the gravitational acceleration, $\boldsymbol{r} = [x\ \eta]^T$ is the material node position vector, $\frac{D}{Dt}$ is the material derivative and $\Gamma_\eta(t)$ is the subset illustrated in figure 1.

2.3 Bottom boundary condition

The free-slip condition is enforced at the seabed,

$$\frac{\partial \Phi}{\partial \boldsymbol{n}} = 0, \quad \boldsymbol{x} \in \Gamma_b, \tag{4}$$

where $\boldsymbol{n} = [n_x\ n_z]^T$ is the outward-pointing unit normal vector.

2.4 Summary of the periodic non-linear water wave problem

For the sake of simplicity, we assume horizontal periodicity for all computations in this work, i.e. no lateral boundaries are present, and thus the problem is finally stated as

$$\begin{cases} \nabla^2 \Phi = 0, & \boldsymbol{x} \in \Omega(t), \\ \frac{\partial \Phi}{\partial z} = 0, & \boldsymbol{x} \in \Gamma_b, \\ \frac{Dx}{Dt} = \frac{\partial \Phi}{\partial x}, & \boldsymbol{x} \in \Gamma_\eta(t), \\ \frac{D\eta}{Dt} = \frac{\partial \Phi}{\partial z}, & \boldsymbol{x} \in \Gamma_\eta(t), \\ \frac{D\Phi}{Dt} = -g\eta + \frac{1}{2}\left(\nabla \Phi \cdot \nabla \Phi\right), & \boldsymbol{x} \in \Gamma_\eta(t), \end{cases} \tag{5}$$

where also constant water depth is assumed and $t = [0, T]$ specifies the time interval of the problem.

3 METHODOLOGY

3.1 Radial basis function-generated finite differences

In radial basis function generated-finite differences (RBF-FD), the methodology is similar to the traditional finite difference method. The finite difference weights are approximated as the linear combination

$$\sum_{i=1}^{n} w_i u_i \approx \mathcal{L}u(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_e}, \tag{6}$$

where $\mathcal{L}$ is a linear operator, w_i are the corresponding weights that approximate $\mathcal{L}u(\boldsymbol{x})$ at evaluation point $\boldsymbol{x} = \boldsymbol{x}_e$ and n is the number of neighboring nodes used in the stencil.

In this paper, the RBF used is the polyharmonic spline $\phi(r) = r^{2m-1}$ where $m \in \mathbb{N}_+$. By augmenting the RBF by polynomial terms, up to an arbitrary order, several advantages may be achieved [14, 15, 16]. In order to compute the weights, we introduce an interpolant as

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) + \sum_{j=1}^{l} \mu_j p_j(\boldsymbol{x}), \tag{7}$$

with matching conditions, that seeks to minimize far-field growth, as

$$\sum_{i=1}^{n} \lambda_i p_j(\boldsymbol{x}_i) = 0, \qquad j = 1, 2, .., l, \tag{8}$$

where $l = \binom{k+d}{k}$ is the number of terms in a kth order, d-dimensional polynomial. Now, the linear system for computing the weights w in $\mathbb{R}^d$, when the RBF is augmented by polynomial terms up to degree k, is expressed as

$$\left[\begin{array}{ccc|ccc} & & & p_1(\boldsymbol{x}_1) & \dots & p_l(\boldsymbol{x}_1) \\ & \boldsymbol{A} & & \vdots & \ddots & \vdots \\ & & & p_1(\boldsymbol{x}_n) & \dots & p_l(\boldsymbol{x}_n) \\ p_1(\boldsymbol{x}_1) & \dots & p_1(\boldsymbol{x}_n) & & & \\ \vdots & \ddots & \vdots & & \boldsymbol{0} & \\ p_l(\boldsymbol{x}_1) & \dots & p_l(\boldsymbol{x}_n) & & & \end{array}\right] \left[\begin{array}{c} w_1 \\ \vdots \\ w_n \\ w_{n+1} \\ \vdots \\ w_{n+l} \end{array}\right] = \left[\begin{array}{c} \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{x}_1\|)|_{\boldsymbol{x}=\boldsymbol{x}_e} \\ \vdots \\ \mathcal{L}\phi(\|\boldsymbol{x} - \boldsymbol{x}_n\|)|_{\boldsymbol{x}=\boldsymbol{x}_e} \\ \mathcal{L}p_1(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_e} \\ \vdots \\ \mathcal{L}p_l(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_e} \end{array}\right], \tag{9}$$

where $\boldsymbol{A} = A_{ij} = \phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$ is the $n \times n$ RBF collocation matrix and $\boldsymbol{0}$ is a $l \times l$ zero matrix. Hence, only the differentiation weights $\boldsymbol{w} = [w_1 \ w_2 \ \dots \ w_n]^T$ are used for approximation of the differential operators described in section 2. In all computations conducted in this paper, the stencil sizes have been chosen to $n \approx 2.5\ell$, and the order of the polyharmonic splines have been chosen as $m = k$ for k odd and $m = k + 1$ for k even.

3.2 Explicit fourth-order Runge-Kutta time stepping

In order to evolve the free surface variables in time, the method of lines approach is used. Specifically, the explicit fourth-order Runge-Kutta method is used, but with a slight modification. At the end of each time step, the free surface variables are reconfigured spatially by means of interpolation, such that the free surface nodes do not become too clustered along the free surface. Other strategies could have been implemented, however, only this strategy will be considered here.

3.3 Node generation strategy

In this paper, the implemented node generation strategy is similar to the ones given in [14, 17]. However, additional steps are introduced due to the time-varying computational domain. The node generation strategy is outlined as follows:

1. Generate nodes on the boundary of the computational domain, $\Omega(t)$.
2. Generate background node set that covers $\Omega(t)$ sufficiently.
3. Remove nodes from the background node set if $r_s < \rho/2$, where ρ is a local node density measure and r_s is the smallest distance to a static boundary node.
4. Perform node repelling on a subset of each of the static boundary nodes' stencil nodes.
5. Save the repelled version of the background node set.
6. Remove nodes from the background node set if $r_d < \rho/2$, where ρ is a local node density measure and r_d is the smallest distance to a dynamic boundary node.
7. Perform node repelling on a subset of each of the dynamic boundary nodes' stencil nodes.
8. Remove nodes from the background node set if they are located outside $\Omega(t)$.
9. Repeat step 6-8 at each substep, where the saved background node set from step 5 is re-used each time.

In the strategy outlined above, static and dynamic boundary nodes refer to their variation over time. Hence, only nodes near moving boundaries are updated at each substep. In this paper, the node generation algorithm from [17] is used to generate the background node set.

3.4 Solution procedure

The overall solution procedure is outlined in algorithm 1. Using an explicit time stepping method, e.g. the RK4 as presented previously, the time derivatives need to be computed, as outlined in section 2 and 3. First the initial- and boundary conditions $\left\{\Phi^0_{(x,\eta)^0}, \eta^0_{x^0}, x^0\right\}$ are provided as input. These are used to solve the Laplace equation in (1) for the velocity potential $\Phi(x,z,t) = \Phi^t_{(x,z)}$ within $\Omega(t)$. Next, the particle velocities are computed at the free surface, which are then used to compute the temporal derivatives given in equations (2,3). In this manner, the free surface variables are stepped forward in time. Before proceeding, the new boundary conditions, $\left\{\Phi^t_{(x^t,\eta^t_{x^t})}, \eta^t_{x^t}, x^t\right\}$, are interpolated back to their initial horizontal position, x^0, and the next time step is initialized from this spatial configuration, i.e. $\left\{\Phi^t_{(x^0,\eta^t_{x^0})}, \eta^t_{x^0}, x^0\right\}$.

Algorithm 1: Overall solution procedure

```
Input: $\left\{\Phi^0_{(x,\eta_{x^0})^0}, \eta^0_{x^0}, x^0\right\}$
$t := 0$
while $t < T$ do
    $\left\{\Phi^{t+\Delta t}_{(x,\eta)^{t+\Delta t}}, \eta^{t+\Delta t}_{x^{t+\Delta t}}, x^{t+\Delta t}\right\} \leftarrow \mathrm{RK4}(\mathcal{L}, \Phi^t_{(x^0,\eta^t_{x^0})}, \eta^t_{x^0}, x^0)$;    % Time stepping
    $\left\{\eta^{t+\Delta t}_{x^0}\right\} \leftarrow \mathrm{interp}(\eta^{t+\Delta t}_{x^{t+\Delta t}}, x^{t+\Delta t}, x^0)$;    % Interpolate free surface elevation
    $\left\{\Phi^{t+\Delta t}_{(x^0,\eta^{t+\Delta t}_{x^0})}\right\} \leftarrow \mathrm{interp}(\Phi^{t+\Delta t}_{(x,\eta)^{t+\Delta t}}, x^{t+\Delta t}, x^0)$;    % Interpolate velocity potential
    $t := t + \Delta t$
end
Output: $\left\{\Phi^0_{(x,\eta)^0}, \cdots, \Phi^T_{(x^0,\eta^T_{x^0})}\right\}, \left\{\eta^0_{x^0}, \cdots, \eta^T_{x^0}\right\}, x^0$
```

4 NUMERICAL ACCURACY AND STABILITY

First, the accuracy of computing the vertical particle velocity is examined for different ranges of non-linearities. Next, the temporal stability is investigated and it is demonstrated how to improve the stability by means of a previously proposed method [18].

4.1 Accuracy

The spatial accuracy is illustrated by the relative error between vertical particle velocities, at the free surface, computed by stream function theory and the proposed model, respectively. The wave steepness (H/L), node density (ρ) and order of augmented polynomials (P) are varied, while the wave length $L = 2\pi$ is kept constant. The results are illustrated in figure 2 along with an example of the node set for the corresponding wave steepness. From figure 2, it is noticed that the convergence rate decreases as the wave steepness approaches the breaking limit.

4.2 Linear stability analysis

The purpose of the linear stability analysis is to investigate the temporal stability when different polynomials are used, and to test if instabilities can be removed. First, we express the linearized and semi-discrete counterpart of the system in (5) as

$$\frac{D}{Dt}\begin{bmatrix} x \\ \eta \\ \Phi \end{bmatrix} = \mathcal{J}\begin{bmatrix} x \\ \eta \\ \Phi \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathcal{L}_x \\ \mathbf{0} & \mathbf{0} & \mathcal{L}_z \\ \mathbf{0} & -g\mathcal{I} & \mathbf{0} \end{bmatrix}\begin{bmatrix} x \\ \eta \\ \Phi \end{bmatrix}, \tag{10}$$

where $\mathcal{I}$ is the identity matrix and $\mathcal{L}_z$ is derived from the relation

$$\boldsymbol{D}_z\Phi|_{\boldsymbol{x}=\boldsymbol{r}} = \left[(\boldsymbol{D}_z)_{11}\Phi_1 + (\boldsymbol{D}_z)_{12}\Phi_2\right] = \left[(\boldsymbol{D}_z)_{11} - (\boldsymbol{D}_z)_{12}\boldsymbol{L}_{22}^{-1}\boldsymbol{L}_{21}\right]\Phi_1 = \mathcal{L}_z\Phi_1, \tag{11}$$

where $\boldsymbol{L}$ is the differential operator for the Laplace problem, introduced in section 2, $\boldsymbol{D}_z$ is the differentiation matrix with respect to the vertical direction and the matrices are decomposed as

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{12} \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} \end{bmatrix}, \tag{12}$$

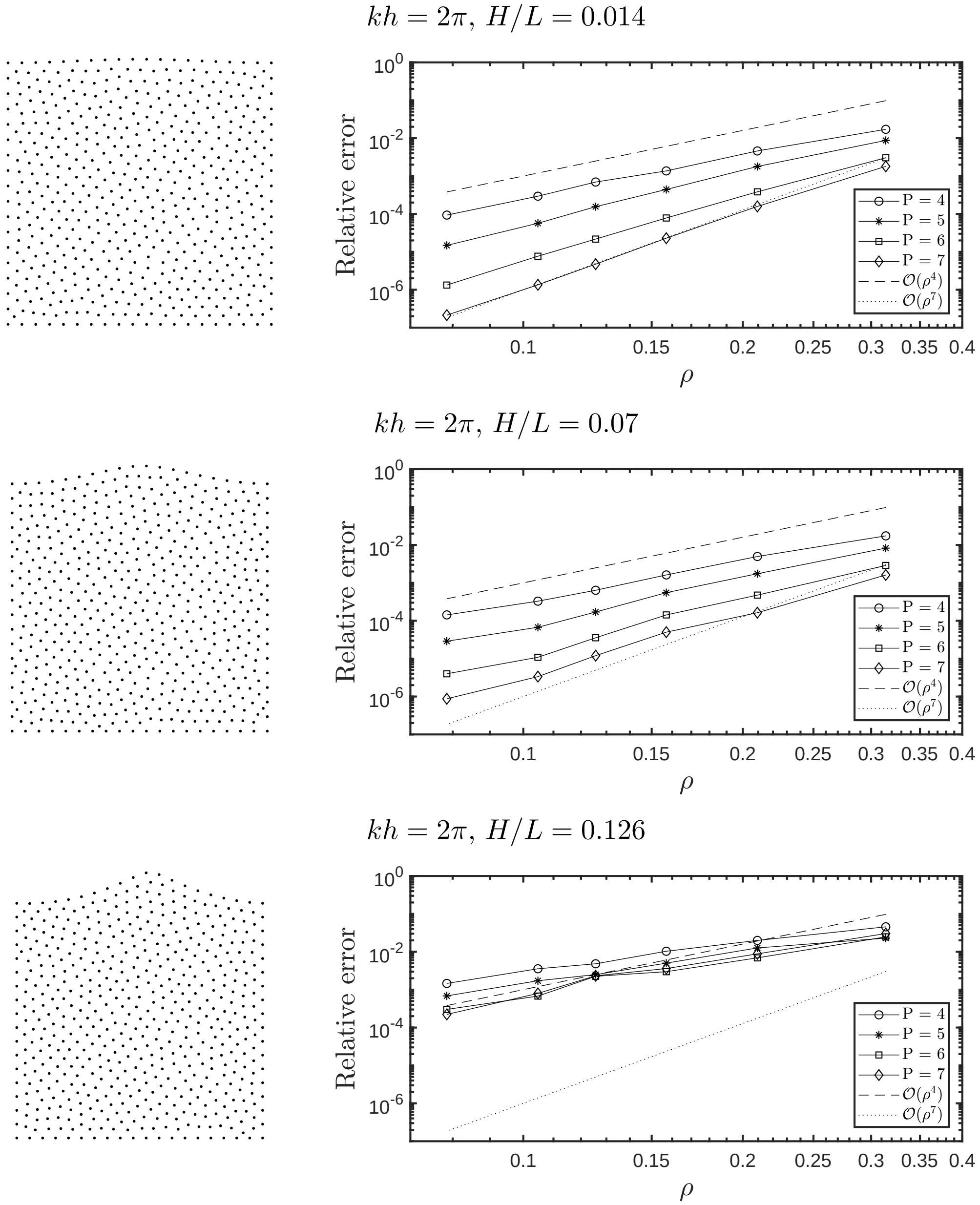
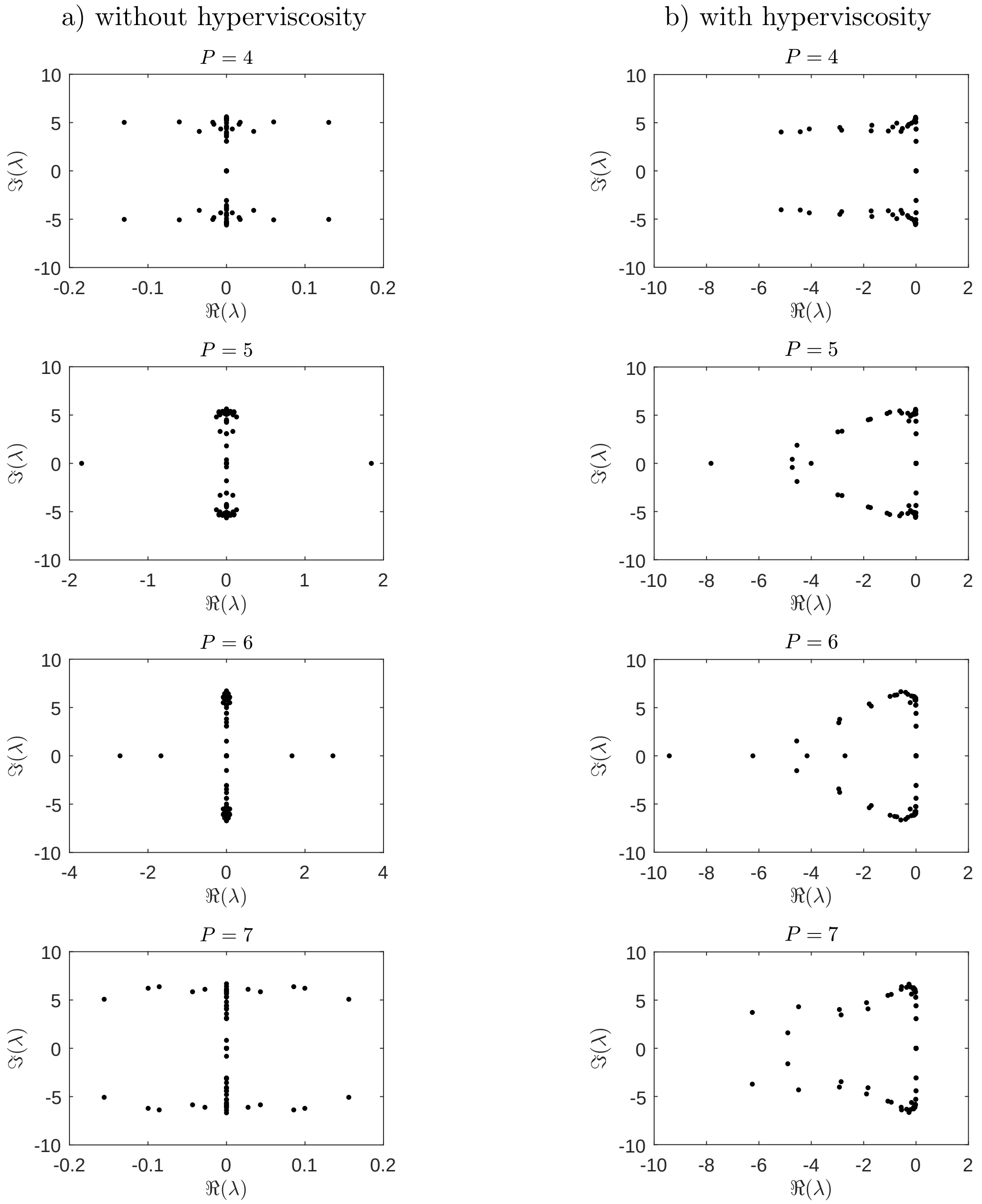


Figure 2: Convergence plots of the relative error computing $\frac{\partial \Phi}{\partial z}$ at the free surface for different H/L ratios and augmented polynomials, while using a node density approximately equal to ρ.

where the subscript indices define whether the nodes are on the free surface (1) or not (2). $\mathcal{L}_x$ is computed in a similar manner as in equation (11). Eigenspectra of the system in equation (10) are illustrated in figure 3.a, where a quasi-uniform node set has been used and also different orders of augmented polynomials. As noticed, the eigenvalues in figure 3.a has positive real parts, which lead to temporal instability. However, as described in e.g. [18], convective PDEs may be stabilized by adding a slight amount of artificial dissipation, known as hyperviscosity. In this work, a third order Laplacian is used along with a tuning parameter γ, which depends on the discretization used. The effect of adding hyperviscosity to the free surface elevation and velocity potential, in equation (10), is illustrated in figure 3.b. Here $\gamma = 6 \cdot 10^{-6}$ has been used for all cases, although a more suitable choice exist for each individual discretization.


a) without hyperviscosity

b) with hyperviscosity

Figure 3: Eigenspectra of $\mathcal{J}$ in equation (10) using a quasi-uniform node set and augmented by up to the Pth order polynomial.

5 RESULTS

Stream function waves of varying steepness are used as benchmark for the non-linear stability and accuracy. Each case is simulated for five periods with the time step $\Delta t = 0.025s \approx T/80$, where T is the wave period. The maximum relative errors between the stream function solution and proposed model are shown in figure 4. For each discretization, a suitable γ-value is determined by linear stability analyses.

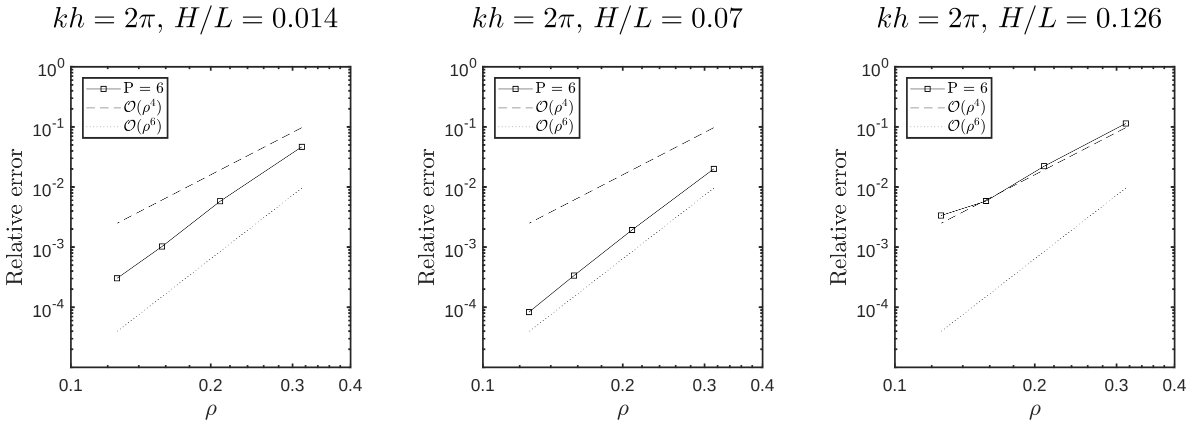


Figure 4: Convergence plots showing the relative error of the free surface elevation after five periods when using a 7th order polyharmonic spline augmented by up to 6th order polynomials.

Once again, it is noticed that the convergence rate tend to decrease as the wave steepness increase. As expected, it drops to the same convergence rate that was noticed in figure 2. However, as quasi-uniform nodes were used, it can be expected that node refinement near the free surface will improve the accuracy without increasing the total number of nodes used.

6 CONCLUSIONS

A novel computational model for non-linear water waves is introduced and the methodology outlined. The main novelty is the application of RBF-FD, which provide high-order approximations along with geometric flexibility for the moving free surface. Additionally, a modification of a previously developed node generation strategy is introduced, such that the computational domain is updated as the boundaries evolve over time. Finally, it is shown that stability can be achieved by adding hyperviscosity to the problem without influencing the convergence rate.

REFERENCES

[1] DeRisk, `http://www.derisk.dk`, [Online; accessed 11-February-2019] (2015).

[2] MIDWEST, `https://project.inria.fr/midwest/`, [Online; accessed 11-February-2019] (2016).

[3] CCP-WSI, `https://www.ccp-wsi.ac.uk/`, [Online; accessed 11-February-2019] (2016).

[4] A. P. Engsig-Karup, H. B. Bingham, O. Lindberg, An efficient flexible-order model for 3D nonlinear water waves, Journal of Computational Physics 228 (6) (2009) 2100–2118.

[5] G. Wu, R. E. Taylor, Finite element analysis of two-dimensional non-linear transient water waves, Applied Ocean Research 16 (6) (1994) 363–372.

[6] D. G. Dommermuth, D. K. Yue, A high-order spectral method for the study of nonlinear gravity waves, Journal of Fluid Mechanics 184 (1987) 267–288.

[7] I. Robertson, S. Sherwin, Free-surface flow simulation using hp/spectral elements, Journal of Computational Physics 155 (1) (1999) 26–53.

[8] S. Yan, Q. Ma, Qale-fem for modelling 3D overturning waves, International Journal for Numerical Methods in Fluids 63 (6) (2010) 743–768.

[9] S. T. Grilli, P. Guyenne, F. Dias, A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom, International Journal for Numerical Methods in Fluids 35 (7) (2001) 829–867.

[10] A. P. Engsig-Karup, C. Eskilsson, D. Bigoni, A stabilised nodal spectral element method for fully nonlinear water waves, Journal of Computational Physics 318 (2016) 1–21.

[11] C. M. Sanchez, A. P. Engsig-Karup, C. Eskilsson, Nonlinear wave-body interaction using a mixed-eulerian-lagrangian spectral element model, in: 37th International Conference on Ocean, Offshore and Artic Engineering (OMAE2018), 2018.

[12] M. E. Nielsen, M. D. Ulriksen, L. Damkilde, SOFIA - A simulation tool for bottom founded and floating offshore structures, Procedia engineering 199 (2017) 1308–1313.

[13] M. D. Ulriksen, D. Bernal, M. E. Nielsen, L. Damkilde, Damage localization in offshore structures using shaped inputs, Procedia engineering 199 (2017) 2282–2287.

[14] V. Bayona, N. Flyer, B. Fornberg, G. A. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, Journal of Computational Physics 332 (2017) 257–273.

[15] N. Flyer, G. A. Barnett, L. J. Wicker, Enhancing finite differences with Radial Basis Functions: Experiments on the Navier–Stokes equations, Journal of Computational Physics 316 (2016) 39–62.

[16] B. Fornberg, N. Flyer, Solving PDEs with Radial Basis Functions, Acta Numerica 24 (2015) 215–258.

[17] B. Fornberg, N. Flyer, Fast generation of 2-D node distributions for mesh-free PDE discretizations, Computers and Mathematics with Applications 7 (69) (2015) 531–544.

[18] B. Fornberg, E. Lehto, Stabilization of RBF-generated finite difference methods for convective PDEs, Journal of Computational Physics 230 (6) (2011) 2270–2285.