J Eng Math (2006) 56:351–370
DOI 10.1007/s10665-006-9064-z

ORIGINAL PAPER

Nodal DG-FEM solution of high-order Boussinesq-type equations

Allan P. Engsig-Karup · Jan S. Hesthaven · Harry B. Bingham · Per A. Madsen

Received: 3 January 2006 / Accepted: 11 July 2006 /
Published online: 11 October 2006

Abstract A discontinuous Galerkin finite-element method (DG-FEM) solution to a set of high-order Boussinesq-type equations for modelling highly nonlinear and dispersive water waves in one horizontal dimension is presented. The continuous equations are discretized using nodal polynomial basis functions of arbitrary order in space on each element of an unstructured computational domain. A fourth-order explicit Runge-Kutta scheme is used to advance the solution in time. Methods for introducing artificial damping to control mild nonlinear instabilities are also discussed. The accuracy and convergence of the model with both h (grid size) and p (order) refinement are confirmed for the linearized equations, and calculations are provided for two nonlinear test cases in one horizontal dimension: harmonic generation over a submerged bar, and reflection of a steep solitary wave from a vertical wall. Test cases for two horizontal dimensions will be considered in future work.

Keywords Discontinuous Galerkin finite-element method · Gravity waves · High-order Boussinesq-type equations · Unstructured grids

1 Introduction

Boussinesq-type equations are used routinely by coastal engineers for the numerical simulation of nonlinear wave motion in near-shore regions. The main idea behind these equations is to reduce the three-dimensional potential-flow problem to a two-dimensional problem by assuming a polynomial variation in the vertical direction. Such equations represent an important engineering tool for predicting wave–wave, wave–bottom, and coastal wave–structure interaction. These phenomena are critical to the design of, e.g., offshore windmill foundations, harbours, and exposed loading facilities.

A. P. Engsig-Karup (✉) · H. B. Bingham · P. A. Madsen
Department of Mechanical Engineering, Technical University of Denmark, 2800 Lyngby, Denmark
e-mail: apek@tdcadsl.dk

J. S. Hesthaven
Division of Applied Mathematics, Brown University, Providence, USA

A large number of Boussinesq-type models are in use today; a comprehensive review of the history of Boussinesq theory can be found in [1]. Basically, there are two classes of Boussinesq models: *two-variable* formulations in terms of an elevation and a horizontal velocity or flux variable, and *three-variable* formulations that also retain the vertical component of velocity as an unknown. Two-variable formulations, for example [2–4], are generally more efficient, but have a limited radius of convergence and are thus restricted to a distinct value of kd (k the wave number and d the water depth) regardless of the order retained in the expansion in terms of dispersion. Three-variable formulations, on the other hand, can have an infinite radius of convergence, as shown by Madsen and Agnon [5]. We focus here on the most accurate of these yet developed, that of Madsen et al. [6], and refer to two different versions of the same Boussinesq method. These versions can be characterized by the number of terms (n) retained in the Padé (n,n) enhanced operators in the derivation of the equation system, which are included to improve the embedded linear and nonlinear properties.

The Padé (4,4) version of this method is capable of accurately modelling nonlinear waves up to the point of breaking and out to relative water depths of approximately $kd = 25$, while accurate kinematics (vertical variation of the flow) are obtained out to approximately $kd = 10$. The Padé (2,2) version obtained by using the optimized coefficients determined by Jamois et al. [7] is accurate in dispersion up to approximately $kd = 10$ and with accurate kinematics up to approximately $kd = 4$. These equations are thus appropriate for modelling a broad range of water wave phenomena in near-coastal environments. This additional accuracy and range of dispersion is, however, obtained at the cost of additional variables (degrees of freedom) and more complicated systems of linear equations relative to the two-equation models, and this provides new challenges for obtaining efficient numerical solutions.

Many of these challenges have been successfully overcome using a finite-difference solution on a structured (uniformly spaced, rectangular) grid as reported by Fuhrman and Bingham [8] and by Fuhrman et al. [9]. This finite-difference scheme is, however, at a fixed order of accuracy, and lacks geometric flexibility both in terms of local grid refinement and the inclusion of other than piecewise rectangular structures. The goal of the present work is to provide a solution with both geometric and order flexibility by means of an unstructured discontinuous Galerkin finite-element method (DG-FEM). We note that the unstructured grid is particularly attractive by being adaptable to the physical scales of the problem, and this can potentially lead to a significant reduction in the computational effort.

The unstructured finite-element method (FEM) has been applied to various two-variable Boussinesq-type problems. The classical Boussinesq equations of Peregrine [2] were solved by Antunes Do Carmo et al. [10], Ambrosi and Quartapelle [11], and again by Sherwin and Eskilsson [12]. A set of weakly dispersive Boussinesq-type equations was solved and studied by Langtangen and Pedersen [13]. Li et al. [14] solved the improved Boussinesq equations by Beji and Nadaoka [15]. The extended Boussinesq equations of Madsen and Sørensen [3] were solved using a FEM model by Sørensen et al. [16], while the equations of Nwogu [4] were solved by Walkley and Berzins [17, 18], and Eskilsson and Sherwin [19, 20]. Recently, Eskilsson and Sherwin [21, 22] solved the same set of extended equations using a DG-FEM method.

At first glance, the use of a discontinuous method to solve continuous problems where we expect smooth solutions (i.e. up to wave breaking) may seem strange, especially since a continuous form of the method is available. The reasons for this choice are: ease of implementation; and improved computational efficiency due to the local nature of the DG-FEM, which is especially important for implementing higher-order operators (in this case up to fifth-order derivatives).

The remainder of this paper is organized as follows. Section 2 reviews the governing equations of the high-order Boussinesq-type formulation to be solved. Section 3 presents a method-of-lines stability analysis of the continuous linearized equations in both one and two horizontal dimensions. Section 4 describes the numerical methods chosen for discretizing the governing equations in one horizontal dimension. A discrete stability analysis is given along with descriptions of the techniques used for wave generation and absorption, as well as mild filtering techniques that is necessary to stabilize the nonlinear calculations.

Section 5 provides some numerical results obtained using the proposed methodology. Section 6 provides a summary along with concluding remarks.

2 The high-order Boussinesq-type equations

If viscous effects are small enough to be neglected, then potential-flow theory provides us with a reasonably good mathematical description of the flow of water waves. The main idea of Boussinesq-type equations is to simplify these standard equations of potential flow by assuming a polynomial variation in the vertical direction in terms of a velocity potential function. The high-order Boussinesq-type equations are derived from an exact infinite-series solution to the Laplace problem in the interior, which can be related to an arbitrary expansion level below the free surface inside the fluid volume. The exact solution in the interior can then be combined with the exact representation of the free surface and bottom boundary conditions. By carefully truncating and enhancing the resulting finite operators using Padé approximants, it is possible to arrive at the final high-order Boussinesq-type velocity formulation given in the original work by Madsen et al. [6]. Derivation details for the system of equations can be found in the original paper.

The fluid motion is described by the horizontal velocity vector $\hat{\boldsymbol{u}}$, the vertical velocity $\hat{w}$, evaluated at the expansion level $z = \hat{z}$, and the surface elevation η. The variables defined at the expansion level can be related to the surface variables defined at $z = \eta(\boldsymbol{x}, t)$ denoted with a tilde. The still-water depth $d(\boldsymbol{x})$ is measured from the still-water level (SWL), which is positioned at the still water surface where $z = 0$, as illustrated in Fig. 1. The gravitational acceleration g is set to $9{\cdot}81\ \mathrm{m}^2/\mathrm{s}$.

The set of high-order Boussinesq-type equations are given in terms of variables defined directly on the free surface and given by the exact *kinematic* and *dynamic* free surface boundary conditions as

$$\partial_t \eta = \tilde{w} - \nabla\eta \cdot (\tilde{\boldsymbol{U}} - \tilde{w}\nabla\eta), \tag{1}$$

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

where $\tilde{\boldsymbol{U}} \equiv \tilde{\boldsymbol{u}} + \tilde{w}\nabla\eta = (\tilde{U}, \tilde{V})^T$ has been introduced, $\nabla = (\partial_x, \partial_y)^T$ is the two-dimensional gradient operator in a Cartesian coordinate system, and ∂_x is the partial derivative with respect to the $\boldsymbol{x}$-coordinate. From appropriate initial conditions at time t_0, the free-surface elevation $\eta(\boldsymbol{x}, t)$ can be evolved for $t > t_0$ using (1) and the tangential free-surface velocity $\tilde{\boldsymbol{U}}(\boldsymbol{x}, t)$ can be updated using (2). To evolve these variables the vertical free-surface velocity $\tilde{w}(\boldsymbol{x}, t)$ has to be determined. This is done by first solving an algebraic system that combines the solution of the (truncated) Laplace problem in the interior with a *kinematic bottom boundary condition*, and then relating this solution to the vertical free-surface velocity. Thus, in one horizontal dimension (1D), we need to solve a time-dependent system of the form

$$\begin{bmatrix} \tilde{U} \\ 0 \end{bmatrix} = \begin{bmatrix} A_2 - \partial_x\eta \cdot B_2 & B_2 + \partial_x\eta \cdot A_2 \\ A_3 + \partial_x d \cdot S_1 & B_3 - \partial_x d \cdot S_2 \end{bmatrix} \begin{bmatrix} \hat{u}^* \\ \hat{w}^* \end{bmatrix}, \tag{3}$$

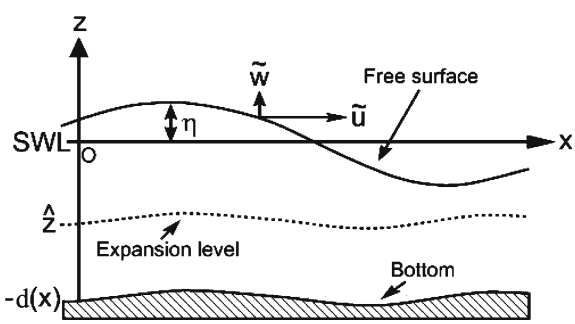

Fig. 1 Definition sketch in one horizontal dimension

and then the vertical free-surface velocities can be obtained from

$$\tilde{w} = -B_2\hat{u}^* + A_2\hat{w}^*. \tag{4}$$

Note that the auxiliary variables $\hat{\boldsymbol{u}}^*, \hat{w}^*$ (also see below) have been introduced as part of the Padé enhancement procedure in the derivations. These auxiliary variables are related to the physical velocities through a linear operator.

In two horizontal dimensions (2D) we need to solve

$$\begin{bmatrix} \tilde{U} \\ \tilde{V} \\ 0 \end{bmatrix} = \begin{bmatrix} A_{11} - \partial_x\eta \cdot B_{11} & A_2 - \partial_x\eta \cdot B_{12} & B_{11} + \partial_x\eta \cdot A_1 \\ A_2 - \partial_y\eta \cdot B_{11} & A_{22} - \partial_y\eta \cdot B_{12} & B_{12} + \partial_y\eta \cdot A_1 \\ A_{01} + S_{01} & A_{02} + S_{02} & B_0 + S_{03} \end{bmatrix} \begin{bmatrix} \hat{u}^* \\ \hat{v}^* \\ \hat{w}^* \end{bmatrix}. \tag{5}$$

As in 1D, the vertical free-surface velocities can be computed from the solution of this algebraic system as

$$\tilde{w} = -B_{11}\hat{u}^* - B_{12}\hat{v}^* + A_1\hat{w}^*. \tag{6}$$

The terms allowing for a mildly varying bottom are given as

$$S_{01} = \partial_x d \cdot C_{11} + \partial_y d \cdot C_2, \tag{7}$$

$$S_{02} = \partial_x d \cdot C_2 + \partial_y d \cdot C_{22}, \tag{8}$$

$$S_{03} = -\partial_x d \cdot C_{13} - \partial_y d \cdot C_{23}. \tag{9}$$

The different continuous operators used to define the complete equation systems (3–9) are given in Sect. 7 The structure of the global operators on the right-hand sides in (3) or (5) is *unsymmetric*.

A solution procedure similar to that suggested by Madsen et al. [6] is used for solving the set of governing equations. Madsen et al. also concluded that a constant expansion level of approximately $\hat{z} = -0{\cdot}5d$ leads to optimal accuracy with respect to the vertical velocity distribution; this expansion level is thus chosen throughout this paper. This choice simplifies the linear systems in (3) and (5) and results in a block-symmetric system, that reduces the number of different operators.

At the physical boundaries, we generally specify impermeable wall conditions. If needed, we generate waves internally using relaxation zones and approximate radiation boundaries using sponge layers as discussed in Sect. 4.9.

3 Stability considerations

Fuhrman et al. [23] considered the numerical stability of their finite-difference model and found it to be linearly stable and, under sufficient artificial damping, nonlinearly stable.

In this section, we consider the linear stability of the analytic problem in a more general way by a Fourier analysis. It will be shown that the eigenspectra of the continuous operators are purely imaginary, and upper bounds for the maximum eigenvalues will be established. In Sect. 4.6, it will be shown that these results are also valid for the discrete problem for our choice of discretization.

3.1 Fourier analysis

The starting point for the Fourier analysis is the linearized equation system in a periodic domain containing a flat bottom, given in the semi-discrete form as

$$\partial_t \boldsymbol{q} = L(\boldsymbol{q}) = J\boldsymbol{q}, \tag{10}$$

where $\boldsymbol{q} = (\eta, \boldsymbol{u})^T$ is a vector of primitive variables. In 1D, this gives

$$\partial_t \begin{bmatrix} \eta \\ u \end{bmatrix} = \begin{bmatrix} 0 & J_{12} \\ -g\partial_x & 0 \end{bmatrix} \begin{bmatrix} \eta \\ u \end{bmatrix}. \tag{11}$$

Solving Eq. 3 and inserting the solution into (4), we obtain the continuous operator

$$J_{12} = \frac{A_2A_3 + B_2B_3}{A_3B_2 - A_2B_3}. \tag{12}$$

We perform a Fourier linear stability analysis of the Padè (2,2) version by Madsen et al. [6] assuming a harmonic variation in space,

$$\eta(x,t) = \hat{\eta}(t)\mathrm{e}^{\mathrm{i}kx}, \quad u(x,t) = \hat{u}(t)\mathrm{e}^{\mathrm{i}kx}, \tag{13}$$

in (11). In Fourier space, the operator is given as

$$J_{12} = -\mathrm{i}\frac{24kd\left(10 + (kd)^2\right)\left(60 + (kd)^2\right)}{14400 + 6480(kd)^2 + 264(kd)^4 + (kd)^6}, \tag{14}$$

and the eigenvalues of the Jacobian J in Eq. (11) are complex and imaginary

$$\lambda(J) = \pm\mathrm{i}\sqrt{\frac{(g/d)(kd)^2 24\left(600 + 70(kd)^2 + (kd)^4\right)}{14400 + 6480(kd)^2 + 264(kd)^4 + (kd)^6}}. \tag{15}$$

In the shallow-water limit, $kd < 0{\cdot}31$, the eigenvalues reduce to those of the linearized shallow water equations, $\lambda(J_{\mathrm{SWE}}) = \pm k\sqrt{gd}$.

We conclude that (i) the eigenvalues of the linearized Boussinesq formulation reduce to the shallow-water eigenvalues for relatively low depths, (ii) the exact eigenspectra of the linearized formulation are imaginary, and (iii) the maximum absolute eigenvalue of the continuous problem is bounded for a constant still-water depth, d, since from (15) we find that

$$|\lambda_{\max}^{1D}| = \lim_{kd\to\infty} |\lambda(J)| = 2\sqrt{6}\sqrt{\frac{g}{d}}. \tag{16}$$

In other words, for a given still-water depth, the maximum absolute eigenvalue of the Jacobian increases with kd. Increasing kd on a fixed depth corresponds to allowing the wave length $L \to 0$, i.e. the smallest (and slowest) wave harmonic determines the size of the absolute maximum eigenvalue on a fixed depth.

Similar results are obtained for the Padé (2,2) version, and for both versions of the formulation in 2D, as shown in Table 1. The upper bound for the eigenvalues remains the same for the Padé (2,2) formulation in both 1D and 2D. For the Padé (4,4) formulation, the upper bound is lower than that found for the one-dimensional case. Thus, the above analysis suggests that the less accurate Padé (2,2) formulation retaining third-order rather than fifth-order differential operators, as in the Padé (4,4) formulation, has better stability properties if the discrete properties of the Jacobian mimic this behaviour.

4 Numerical methods

In this section we outline the numerical methods for solving the governing Eqs. 3–9. A method-of-lines approach has been used, where the spatial and temporal discretizations are considered separately. The DG-FEM is used for the spatial discretization of the governing equations, and the resulting semi-discrete equation system is solved using an explicit Runge–Kutta method to evolve the equations in time from appropriate initial conditions.

Table 1 Analytical bounds for $|\lambda_{\max}|\sqrt{\frac{d}{g}}$ for the linearized formulations in different horizontal dimensions for $\hat{z} = -0{\cdot}5d$

	1D	2D
Padé (2,2)	$2\sqrt{6}$	$2\sqrt{6}$
Padé (4,4)	$2\sqrt{15}$	$3\sqrt{5}$

4.1 The discontinuous Galerkin finite-element method

The DG-FEM is used for the spatial discretization. Originally, the method was developed for problems governed by conservation laws, e.g., see review on DG-FEM in [24]. In that paper, the DG-FEM is used to discretize and solve the high-order Boussinesq-type formulation, which is not given in conservation form.

The DG-FEM is a high-order *spectral* finite-element method. On each element, it is based on the Galerkin form of the more general Mean Weighted Residual method, which determines the way we satisfy (or minimize the residual of) our governing PDE in seeking an approximate solution. Before constructing any DG-FEM scheme, the domain is subdivided into elements. The main idea of the DG-FEM is to allow the solution to be discontinuous over the local-element boundaries and only couple adjacent elements using suitable numerical fluxes. The choice of numerical fluxes can be borrowed from the finite-volume framework, which is theoretically well-established. The DG-FEM combined with nodes defined directly at the boundaries offers the opportunity to determine the numerical fluxes directly. Further, the method can by construction easily handle complex geometries, irregular meshes, and polynomials of different degrees in different elements. Thus, the method supports *hp*-adaptive strategies for obtaining convergent solutions. A major advantage of the DG-FEM is that it conceptually generalizes to higher dimensions; therefore, once familiar with the one-dimensional implementation, it is straightforward to apply the concepts to higher dimensions.

4.2 Subdivision of the computational domain

Before applying the discontinuous Galerkin procedure to discretize these equations in space, we partition the computational domain into K non-overlapping elements, such that $\Omega = \bigcup_{k=1}^{K} \Omega_k$. The characteristic size of the k'th element is denoted by h_k; in one horizontal dimension, this is the size of the element.

On the k'th element, the solution is approximated locally by a finite sum as

$$u^k(\boldsymbol{x}) = \sum_{n=0}^{N} \hat{u}_n^k \phi_n(\boldsymbol{x}) = \sum_{n=0}^{N} u_n^k l_n(\boldsymbol{x}), \tag{17}$$

where we have $N+1$ degrees of freedom inside each element in terms of the *unknown* modal coefficients $\hat{u}_n^k$ or nodal coefficients u_n^k for $n = 0, \ldots, N$. $l_n(\boldsymbol{x})$ are the Lagrange polynomials defined on the set of nodes used in combination with the chosen orthogonal basis. The basis functions $\phi_n, n = 0, \ldots, N$, are chosen such that $u^k \in \mathbb{V}_k^N$, where the approximation space $\mathbb{V}_k^N$ is defined as

$$\mathbb{V}_k^N = \{v : v_k \in \mathbb{P}^N(\Omega_k), \forall \Omega_k \in \Omega\}, \tag{18}$$

i.e., polynomials of at most order N in each element. The classical Legendre polynomials $P_n(x)$ are used as our orthogonal basis in 1D, normalized such that

$$\phi_n(x) = \frac{P_n(x)}{\sqrt{2n+1}}. \tag{19}$$

As shown by Hesthaven and Warburton [25], we can choose to represent our polynomial basis functions in a Lagrangian (i.e., a nodal) basis; we use the Legendre–Gauss–Lobatto nodes in one dimension. By uniqueness of the polynomial interpolations, the following (constant) transformation matrix can be defined for transforming the variables between the modal and nodal spaces

$$\mathcal{V}_{ij} = \phi_j(\boldsymbol{x}_i), \quad i, j = 0, \ldots, N. \tag{20}$$

Here $\boldsymbol{x}_i$ is the i'th nodal point inside the k'th element. This leaves us with the following relationship between the modal and nodal coefficients and basis functions

$$u^k = \mathcal{V}\hat{u}^k, \quad \phi_i(\boldsymbol{x}) = \mathcal{V}^T l_i(\boldsymbol{x}). \tag{21}$$

These relationships allow us to determine both the mass matrix and the stiffness matrix exactly for a standard element without using high-order Gaussian quadrature rules for the element integrals. By a unique linear map the physical coordinates $\boldsymbol{x}$ can be transformed to the coordinates of the standard element; thus only one set of operators needs to be stored.

4.3 Spatial discretization

Because the high-order Boussinesq formulation constitutes a complex system of equations, we are faced with a number of challenges. In particular, the time-dependent equations (1–2) are nonlinear and not given in a conservative form, and the involved algebraic system (3) or (5) contains nonlinear coefficient terms, is time-dependent, and contains high-order differential operators.

To apply the DG-FEM, we first rewrite (1–2) slightly by introducing two auxiliary functions $Q(\boldsymbol{x})$ and $F(\boldsymbol{x})$, as

$$\partial_t \eta = Q(\boldsymbol{x}), \tag{22}$$

$$Q(\boldsymbol{x}) = \tilde{w} - \nabla\eta \cdot (\tilde{\boldsymbol{U}} - \tilde{w}\nabla\eta), \tag{23}$$

where $Q(\boldsymbol{x})$ is a source function, and

$$\partial_t \tilde{\boldsymbol{U}} = -\nabla F(\boldsymbol{x}), \tag{24}$$

$$F(\boldsymbol{x}) = g\eta + \frac{\tilde{\boldsymbol{U}} \cdot \tilde{\boldsymbol{U}}}{2} - \frac{\tilde{w}^2}{2}(1 + \nabla\eta \cdot \nabla\eta). \tag{25}$$

Following the DG-FEM procedure, for each element in the domain we multiply (22) and (24) with a smooth weight function $v(x) = \phi(x)$. Two integration by parts are carried out for the right-hand side term in (24). In the intermediate step of these partial integrations, the analytic flux function $\boldsymbol{n} \cdot F(\boldsymbol{x})$ is interchanged with a continuous numerical flux function $\boldsymbol{n} \cdot F^*(\boldsymbol{x}^{k,-}, \boldsymbol{x}^{k,+})$ to be chosen, which allow us to connect adjacent elements. By this approach, the starting point for the *strong* DG-FEM formulation of (22) and (24) for the k'th element, $k = 1, \ldots, K$, becomes

$$\int_{\Omega_k} \phi \partial_t \eta \mathrm{d}\boldsymbol{x}^k = \int_{\Omega_k} \phi Q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}^k, \tag{26}$$

$$\int_{\Omega_k} \phi \partial_t \tilde{\boldsymbol{U}} \mathrm{d}\boldsymbol{x}^k = -\int_{\Omega_k} \phi \nabla F(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}^k + \oint_{\partial\Omega_k} \phi \boldsymbol{n} \cdot (F(\boldsymbol{x}) - F^*(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}^k, \tag{27}$$

where $\boldsymbol{n} = (n_x, n_y)^T$ is the outward pointing normal vector on the boundary of the k'th element.

Before arriving at the final numerical scheme, we replace the continuous variables with finite polynomial approximations of the form given in (17), and choose suitable consistent numerical fluxes. We choose the simplest numerical fluxes available, namely central fluxes or Lax–Friedrichs fluxes given by the trace operators

$$F^*_{CF}(u^-, u^+) = 0{\cdot}5(F(u^-) + F(u^+)), \tag{28}$$

$$F^*_{\mathrm{LF}}(u^-, u^+) = 0{\cdot}5(F(u^-) + F(u^+) + s(u^- - u^+)), \tag{29}$$

$$s = \max_{x\in\Omega}\{|\partial_u F(u(x))|\},$$

where s is the so-called maximum (characteristic) wave speed.

The central fluxes are used for computing spatial derivatives and the (global) Lax–Friedrichs fluxes are used for the time-dependent momentum Eq. 27 to introduce a small amount of artificial viscosity.

In 1D, the final scheme for the time-dependent equations can be written in a compact form (discrete operator notation borrowed from Sherwin [26]) using the following discrete elemental operators defined

and computed for a standard element as

$$\mathcal{M}_{ij} = \int_{-1}^{1} \phi_i(\xi)\phi_j(\xi)\mathrm{d}\xi, \quad \mathcal{M} = (\mathcal{V}\mathcal{V}^T)^{-1}, \tag{30}$$

$$\mathcal{S}_{ij} = \int_{-1}^{1} \phi_i(\xi)\phi_j'(\xi)\mathrm{d}\xi, \quad \mathcal{S} = \mathcal{M}\mathcal{D}, \tag{31}$$

$$\mathcal{E}_{ij} = \phi_i(-1)\phi_j(-1), \quad \mathcal{F}_{ij} = \phi_i(-1)\phi_j(1), \tag{32}$$

$$\mathcal{G}_{ij} = \phi_i(1)\phi_j(-1), \quad \mathcal{H}_{ij} = \phi_i(1)\phi_j(1), \tag{33}$$

where $\mathcal{M}$ is the *Mass matrix*, $\mathcal{S}$ is the *Stiffness matrix*, and $\mathcal{D}$ is a differentiation matrix, which can be determined from the mass and the stiffness matrices for a given set of nodes.

Using the given discrete standard operators with a nodal representation, we can write the numerical scheme in 1D for the k'th element in compact notation as

$$\frac{h_k}{2}\mathcal{M}\frac{\mathrm{d}}{\mathrm{d}t}\eta^k = \frac{h_k}{2}\mathcal{M}Q^k, \tag{34}$$

$$Q^k = \tilde{w}^k - f^k(\tilde{U}^k - \tilde{w}^k f^k), \tag{35}$$

$$\frac{h_k}{2}\mathcal{M}f^k = \mathcal{S}\eta^k + 0{\cdot}5(\mathcal{G}\eta^{k+1} - \mathcal{H}\eta^k) - 0{\cdot}5(\mathcal{F}\eta^{k-1} - \mathcal{E}\eta^k), \tag{36}$$

and using Lax–Friedrichs numerical fluxes in

$$\begin{aligned}\frac{h_k}{2}\mathcal{M}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{U}^k = {} & \mathcal{S}F^k + 0{\cdot}5(\mathcal{G}F^{k+1} - \mathcal{H}F^k) - 0{\cdot}5(\mathcal{F}F^{k-1} - \mathcal{E}F^k) \\ & -s(0{\cdot}5(\mathcal{G}\tilde{U}^k - \mathcal{H}\tilde{U}^{k-1}) - 0{\cdot}5(\mathcal{F}\tilde{U}^k - \mathcal{E}\tilde{U}^{k-1})),\end{aligned} \tag{37}$$

$$F^k = g\eta^k + \frac{\left(\tilde{U}^k\right)^2}{2} - \frac{\left(\tilde{w}^k\right)^2}{2}(1 + (f^k)^2). \tag{38}$$

To arrive at this numerical scheme, it should be clear that in (36) we obtained a DG-FEM approximation to $f = \partial_x\eta$ by using a strong form DG-FEM and central fluxes. Also, the nonlinear terms in the functions given in (23) and (25) are determined using direct products as shown in (35) and (38) at the grid nodes in a collocation manner instead of using aliasing-free orthogonal projections. Although, this approach is used for efficiency, while being convenient and consistent, it may lead to aliasing errors in the computations requiring the use of de-aliasing.

To implement a numerical scheme for the algebraic system (3) or (5) require a global assembly procedure for assembling the local discrete operators into global operators. For the finite-domain case, this procedure involves the construction of the first-order differential operators having suitable boundary conditions imposed. However, for convenience, we present the assembly procedure for a periodic domain, since the modifications for the finite-domain case are straightforward.

The discrete global operators are constructed using a Local Discontinuous Galerkin (LDG) method. Central fluxes are used to connect the elements as in Bassi and Rebay [27]. Each of the high-order differential operators in the system are decomposed into a set of first order differential operators, e.g., the third derivative operator in x as

$$z = \partial_{xxx}u \rightarrow z = \partial_x q, \quad q = \partial_x r, \quad r = \partial_x u. \tag{39}$$

The high-order differential operators are determined by matrix–matrix products of the discrete global first-order differential operators with suitable boundary conditions imposed.

The global assembly of the discrete first-order differential operators defined on a periodic domain needs to be determined from the local representation on each element. Consider the PDE

$$\partial_x u = f, \ \ x \in [0, L], \tag{40}$$

which by the strong form of the DG-FEM procedure leads to the local scheme

$$\int_{\Omega^k} \phi \partial_x u \, \mathrm{d}x^k + \oint_{\partial\Omega^k} \phi \boldsymbol{n} \cdot (u^* - u) \, \mathrm{d}x^k = \int_{\Omega_k} \phi f \, \mathrm{d}x^k. \tag{41}$$

The computational domain is subdivided in the usual way into K elements. Then for each element, we can write the numerical nodal scheme as

$$\mathcal{S}u^k + 0{\cdot}5(\mathcal{G}u^{k+1} - \mathcal{H}u^k) - 0{\cdot}5(\mathcal{F}u^{k-1} - \mathcal{E}u^k) = \frac{h_k}{2}\mathcal{M}f^k. \tag{42}$$

Now, if we only consider two elements (set $K = 2$) for simplicity, and use central fluxes to connect our elements, we can write the globally assembled numerical scheme as

$$\begin{bmatrix} \mathcal{S} + 0{\cdot}5(\mathcal{E} - \mathcal{H}) & 0{\cdot}5(\mathcal{G} - \mathcal{F}) \\ 0{\cdot}5(\mathcal{G} - \mathcal{F}) & \mathcal{S} + 0{\cdot}5(\mathcal{E} - \mathcal{H}) \end{bmatrix} \begin{bmatrix} u^1 \\ u^2 \end{bmatrix} = \begin{bmatrix} \frac{h_1}{2}\mathcal{M} & 0 \\ 0 & \frac{h_2}{2}\mathcal{M} \end{bmatrix} \begin{bmatrix} f^1 \\ f^2 \end{bmatrix}, \tag{43}$$

or in a more compact form as

$$\mathcal{D}_x^G u = f, \tag{44}$$

where both u and f are DoF $\times$ 1 sized vectors, DoF $= K \cdot (N + 1)$, and the global differentiation operator with respect to x with boundary terms included is defined as

$$\mathcal{D}_x^G = \begin{bmatrix} \frac{h_1}{2}\mathcal{M} & 0 \\ 0 & \frac{h_2}{2}\mathcal{M} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{S} + 0{\cdot}5(\mathcal{E} - \mathcal{H}) & 0{\cdot}5(\mathcal{G} - \mathcal{F}) \\ 0{\cdot}5(\mathcal{G} - \mathcal{F}) & \mathcal{S} + 0{\cdot}5(\mathcal{E} - \mathcal{H}) \end{bmatrix}. \tag{45}$$

The global third-order differential operator given in (39) can easily be constructed from this global operator as

$$\partial_{xxx} \to \mathcal{D}_x^G \cdot \mathcal{D}_x^G \cdot \mathcal{D}_x^G. \tag{46}$$

By this approach, the high-order differential operators can be built. For example, the 1D operator A_2 in Eq. 3, can be discretized as

$$A_2^G = \mathcal{I} - \alpha_2 \cdot (\mathcal{D}_x^G)^2 + \alpha_4 \cdot (\mathcal{D}_x^G)^4, \tag{47}$$

where $\mathcal{I}$ is the unitary matrix of size DoF $\times$ DoF and $\alpha_i, i = 2, 4$ are diagonal matrices. When each of the discrete block-differential operators are constructed in this way, we can assemble the discrete global operator in (3).

The assembly of the nonlinear operators has been done in a collocation manner. For example,

$$\partial_x \eta \cdot B_2 \to \eta_x \cdot B_2^G, \tag{48}$$

where η_x is a diagonal matrix holding the discrete nodal values of $\partial_x \eta$. Although, this approach may lead to aliasing errors, it allows for a fast and efficient way of reconstructing the operators, which is considered to be necessary to keep the computational costs low in the fully nonlinear simulations, and for the implementation of the method to be relatively simple and practical.

Due to the locality of the DG-FEM, the global assembly results in a relatively sparse global matrix. This matrix can be reordered in a preprocessing step such that the bandwidth of the matrix in the 1D finite-domain case is given as $2 \times (N + 1) \times (2q_{\max} + 1)$, where $q_{\max}$ is the order of the largest derivative in the formulation. Thus, the bandwidth is dependent on both the local polynomial order and the highest derivative in the block operators. The global matrix system is usually sparse and hence it can be solved efficiently by an appropriate sparse solution method, e.g., using the sparse matrix solver UMFPACK; see [28]. Since this global matrix system has to be both built (for the fully nonlinear problem) and solved at every time step, this becomes the computational bottleneck in the numerical scheme. For the present computations, we have used a direct sparse solution technique.

4.4 Boundary conditions

In the current models, we have imposed impermeable wall boundary conditions using a *symmetry technique* (or mirror principle). Whenever needed, radiation boundaries are set up using relaxation zones.

The impermeable wall boundary conditions are imposed weakly at the outer boundaries of the computational domain through the numerical flux function, which is denoted by q^*. Hence, by the symmetry technique, Dirichlet boundary-type conditions are imposed as

$$q^* = -q^-, \tag{49}$$

such that the averages of the two distinct boundary values at each boundary face correspond to the values we seek to impose. Neumann boundary-type conditions are imposed as

$$q^* = q^-, \tag{50}$$

such that the averages are left unmodified.

By this technique, the boundary conditions are imposed approximately to the accuracy of the scheme.

4.5 Temporal discretization

A fourth-order 2N-low storage explicit Runge–Kutta scheme with five stages (labelled LSERK45 here) by Carpenter and Kennedy [29] is used for solving the semi-discrete governing equations in time. The absolute stability region encloses a large part $\lambda\Delta t \leq 3{\cdot}345$ of the imaginary (inviscid stability) axis in the complex plane and is easily implemented.

For hyperbolic-type problems solved using explicit time-integration schemes, where $\lambda\Delta t$ should reside inside the absolute stability region of the chosen time integration scheme for the numerical scheme to be *absolutely stable*, the maximum time step $\Delta t_{\max}$ is restricted by

$$\Delta t_{\max} \leq \frac{|s_A|}{|\lambda_{\max}|}, \tag{51}$$

where s_A is the size of the absolute stability region measured on the imaginary axis at the point where the absolute stability region crosses the imaginary axis in the complex plane.

4.6 Linear stability

A numerical study of linear stability in 1D was carried out to investigate the properties of the DG-FEM scheme and determine if the analytical bounds given in Table 1 holds for the discrete cases. The study was performed using structured domains at various constant depths, where both the number of elements and the fixed local polynomial orders were varied. From the resulting discrete eigenspectra, we numerically determined the magnitude of the largest eigenvalue for the given discretization and depth.

It was found that the discrete eigenspectra of the numerical schemes based on the linearized formulation in 1D result in an upper bound closely matching the analytical results presented in Table 1. This is illustrated in Fig. 2 by two tests; the first test uses few elements and a high local polynomial order, and the second test uses many elements and low local polynomial order. Remarkably, linear stability in 1D was found to be *independent* of the size of the elements, the number of elements, and also the local polynomial order used within the elements (as long as the chosen order lead to well-conditioned local operators, which is the case for all moderate polynomial orders used in applications, usually $N < 10$).

Therefore, the time step size in the linearized model in 1D can, in fact, be chosen based on accuracy considerations alone as long as the analytical criterion is satisfied. However, we conclude from Fig. 2 that,

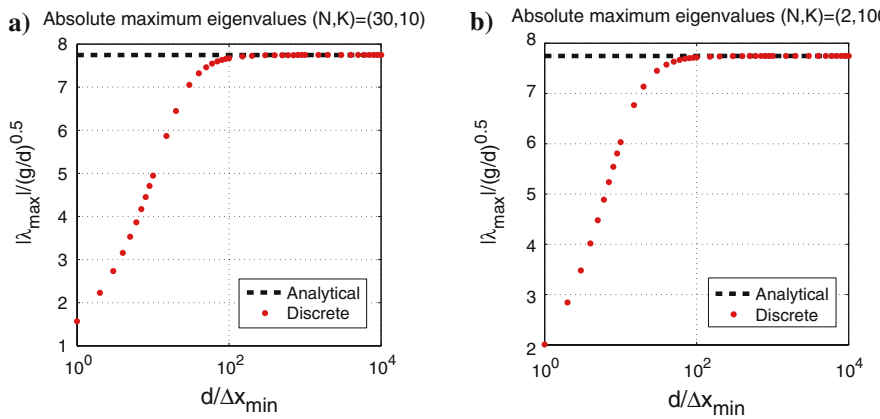

Fig. 2 Computed normalized absolute maximum eigenvalues for the linearized Padé (4,4) formulation for constant depths in 1D using a structured domain

for shallow depths, the upper bound for linear stability is too restrictive if the smallest scale waves cannot be resolved on the grid.

Finally, we note that the time step is usually chosen based on accuracy considerations alone, and therefore the above results are mainly important for testing an application.

4.7 General stability

For the more general cases, a CFL-like criterion is used. The general cases includes both varying depth and fully nonlinear problems, whereupon the linear stability criteria (51) or the analytical bounds in Table 1 do not hold. As is usually the case with numerical methods, the upper limit for the time step is restricted by the numerical growth of the eigenvalues of the discrete operators. Instead, we use the following CFL-like condition

$$\Delta t \leq \mathrm{CFL}\frac{\Delta x_{\min}}{c_{\max}}, \tag{52}$$

where the maximum wave propagation speed is estimated using the general linear dispersion relation, $c_{\max} \approx \sqrt{g/k \cdot \mathrm{Tanh}(kd)}$. Here $\Delta x_{\min}$ is set to be the smallest distance between any two points in an element inside the domain. The magnitude of the CFL constant is $O(1)$.

4.8 De-aliasing techniques

For nonlinear problems lacking (any) numerical diffusivity, we may need to maintain numerical stability by applying a general filtering or de-aliasing technique. In the presented models, we have used the DG-FEM in a collocation manner which may give rise to aliasing errors, since we have chosen to evaluate the nonlinear terms by direct products for the implementation to be practical and for computational efficiency. To reduce such aliasing errors we have used the filtering techniques mentioned here.

For mildly nonlinear cases in 1D, it is sufficient to use a mild nodal filter constructed in a similar way as the one presented by Fischer and Mullen [30]. Instead of constructing the filter on the Gauss–Legendre (GL) nodes, we instead base the filter on the Lobatto–Gauss–Legendre (LGL) nodes. Hence, following their procedure, the filter is constructed as follows. Let I_N^M be an interpolation operator for a polynomial

of degree N defined on $N+1$ GLL nodes onto the $M+1$ GLL nodes. Define $\prod_N := I_N^{N+1} I_{N+1}^N$ to be a projection from P_{N+1}^1 to P_N^1 on $[-1,1]$. The filter is constructed as

$$F_\alpha := \alpha \prod_N + (1-\alpha) I_{N+1}^{N+1}. \tag{53}$$

For mildly nonlinear problems in 1D, we have found that this filter works well with approximately $\alpha = 0{\cdot}1 - 0{\cdot}2$.

For more extreme nonlinear wave problems, the model has been successfully stabilized with a Savitzky–Golay-type (SG) polynomial averaging filter; see [31], which has also been used by Fuhrman and coworkers (e.g. [7]). The SG filter was then applied locally by using an iso-parametric approximation procedure after having averaged the interface values to an equi-distant/structured grid where the polynomial filtering could be carried out before returning (by interpolation back) to the original grid. It is noted that this approach is semi-local in the sense that the averaging is done *across* element interfaces locally. This approach was found to be practical in order to avoid unnecessary large jump discontinuities and reduce any aliasing errors in the solution. The computational cost of this approach is negligible in comparison with the cost of solving the algebraic system every time step when it is set up in the pre-processing stage before the computations.

However, in 2D models where complex geometries may be used, the SG filtering procedure is impractical, and hence alternative de-aliasing means are to be found.

4.9 Relaxation zones

We need to be able to both absorb and generate waves that are generally uni-directional inside the computational domain. A convenient method by Larsen and Dancy [32] for both generating and absorbing waves while preserving the outer domain boundary conditions is to modify the discrete function values in a smooth way in space using a relaxation function $\Gamma(\boldsymbol{x})$ such that the relaxed solution $u^*(\boldsymbol{x})$ is given by

$$u^*(\boldsymbol{x_i}) = \Gamma(\boldsymbol{x_i}) u(\boldsymbol{x_i}) + (1 - \Gamma(\boldsymbol{x_i})) u_e(\boldsymbol{x_i}), \tag{54}$$

where $\Gamma(\boldsymbol{x}) \in [0,1]$ has to be a single-valued function for $\boldsymbol{x} \in R^d$, where d is the number of horizontal dimensions, and u_e is an analytical solution. The first term acts as a "sponge layer" which will effectively damp any wave motion in the zone. The last terms containing u_e act as "source terms" within the relaxation zone; including them in the relaxation, allows us to generate waves.

This relaxation method is somehow *ad hoc* and the following parameters should be balanced before arriving at satisfactory results: (a) shape of relaxation functions, (b) position of relaxation zones, and (c) length of the relaxation zones.

The following relaxation functions have been used in the simulations for respectively sponge layers (positioned next to wall boundaries) and wave generation

$$\Gamma(x;p) = 1 - x^p, \tag{55}$$

$$\Gamma(x) = -2x^3 + 3x^2. \tag{56}$$

For absorption in sponge layers a good choice is that $\Gamma(x;5{\cdot}0)$ are positioned appropriately where waves are to be both/either generated and/or absorbed and the rule of thumb is that the relaxation should be invoked in a region covering approx. 1–2 wave lengths of the primary wave in the region. A disadvantage of the relaxation method is that it takes up a part of the computational domain, but the compensation is a straightforward procedure, that is both easy and flexible to use. For this reason, it is our method of choice.

5 Computational examples

We conclude with numerical examples confirming that the proposed methodology provide a basis for solving the high-order Boussinesq-type equations accurately.

5.1 Spatial accuracy

Consistency tests of the numerical model based on the Padé (4,4) formulation have been carried out to numerically determine convergence rates for *h*- and *p*-type refinement strategies. The linearized system of equations in 1D is used as the basis for modelling a linear standing wave in a finite domain $x \in [-1, 1]$. The tests are carried out in deep waters with a relative depth corresponding to $kd = 10$, which should be solved accurately by the high-order system. At this depth, all spatial derivatives are important for obtaining accurate solutions, and exceeds the practical deep water limit ($kd = 2\pi$) for most other Boussinesq-type formulations.

It is shown numerically that in the approximate limits, reached by either *h*- or *p*-type refinement or a combination thereof, where sufficient spatial resolution is used to capture the solution, the numerical model has the expected spatial order of accuracy, which is obtained by using central numerical fluxes.

Consistency tests have been carried out for a linear standing wave given by the initial surface elevation $\eta(x,t) = H\cos(\omega t)\cos(kx)$ and zero kinetic energy, where H is the wave height, and the angular frequency ω has been determined using the linear Boussinesq dispersion relation derived by Madsen et al. [6]. The numerical model was run for 100 time steps with a time step small enough for the spatial errors to dominate in each test. The normalized max-norm errors are determined using the analytical solution for a linearly standing wave.

As demonstrated in Fig. 3, convergence is obtained for both *h*-type and *p*-type strategies until the levels of truncation errors from the solution of the algebraic system dominates. Clearly, the *p*-type convergence strategy is superior giving rise to exponentially fast convergence as opposed to the alternative *h*-type convergence strategy leading to algebraic convergence. The spatial order of accuracy is found from the data presented in Fig. 3a to be optimal $\mathcal{O}(h^{N+1})$ for even polynomial orders and sub-optimal $\mathcal{O}(h^N)$ for odd polynomial orders. It is well-known that this even-odd behaviour can occur for problems discretized using central fluxes (see [24]).

5.2 Harmonic generation over a submerged bar

Although potential benefits of DG-FEM for solving the Boussinesq-type equations are expected to be more significant in solving 2D problems, we demonstrate the geometric flexibility and the potential benefits in using the proposed methodology by simulating harmonic generation over a submerged bar.

An experiment where an input wave is propagating over a submerged bar during which the wave undergo transformation was originally proposed by Beji and Battjes [33]. Later, an equivalent experiment,

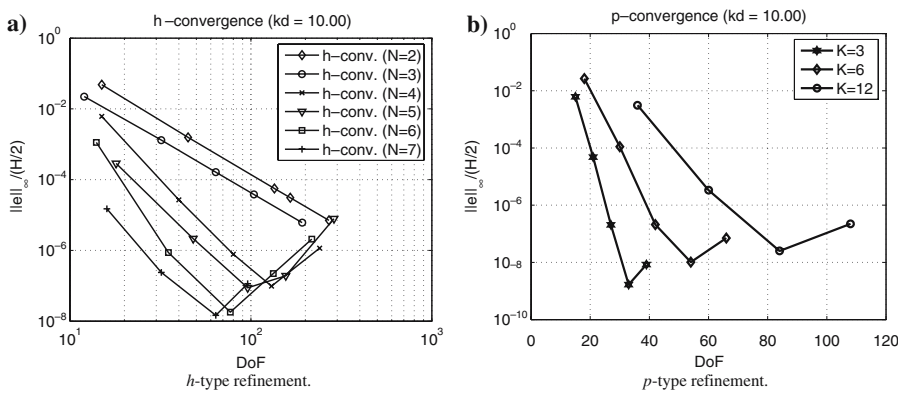

Fig. 3 Consistency test for different convergence strategies

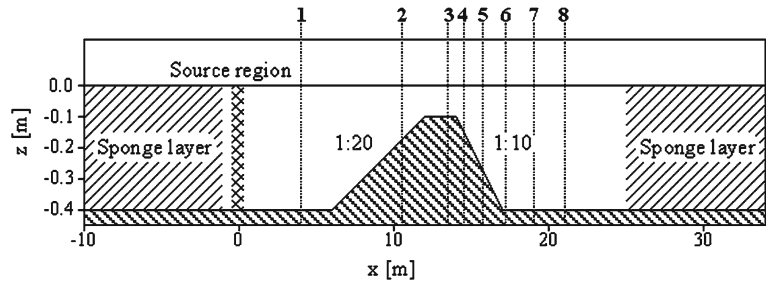

Fig. 4 Submerged bar test setup in original experiment

scaled by a factor of two, was carried out by Luth et al. [34]. The bathymetry was defined as shown in Fig. 4 for the original unscaled experiment.

In this experiment, the input wave propagates toward the bar on a constant bottom with no change in form until it reaches the bar. On the up-slope, the wave will then steepen due to nonlinear shoaling. At the top of the bar, bound harmonics will be released as free harmonics decomposing the wave into shorter waves, which results in a rapidly varying wave profile behind the bar. The shorter wave components or free higher harmonics require a highly accurate dispersion relation to be propagated correctly. Therefore, this experiment has been used as a benchmark test for dispersive wave models. See for example [1], [35] and [6].

We have considered a setup corresponding to Case A in the original experiment by Beji and Battjes. For this case, the input wave on a undisturbed depth of $d = 0{\cdot}40$ m is given as; Case A: $T = 2{\cdot}02$ s and $H = 2$ cm. The input wave was generated using second-order Stokes theory. Computed results are compared with the experimental data obtained by Luth et al. in Fig. 5. Locally on each element, a polynomial order of $N = 8$ is used and $K = 110$ equi-sized elements. At every stage of the explicit Runge–Kutta scheme, the spatial solutions for $\eta(x,t)$ and $\tilde{U}(x,t)$ are filtered using a mild nodal filter with $\alpha = 0{\cdot}1$. As shown in Fig. 5, we achieve excellent agreement with the data with minimal differences in surface elevations (due to the accurate dispersion relation) at all gauge locations.

A numerical simulation done using an unstructured grid adapted to the problem, as depicted in Fig. 6a. Inside the domain, the grid varies linearly from a element size of $h_0 = 0{\cdot}67$ m to an element size of $h_k = 0{\cdot}35$ m positioned at the top of the bar. From the top of the bar and forward, the element size remains constant to be slightly increased in the relaxation region where it is takes the value $h_k = 0{\cdot}4$ m. By use of a local polynomial order of $N = 8$ and a total of $K = 80$ elements, the degrees of freedom are reduced by 26% compared to the former experiment. This is possible because the spatial resolution is coarsened in the region to the left of the bar and fine in the region where the spurious (or faster) wave components are released after the bar. As depicted in Fig. 6 the results are indistinguishable from the ones presented in Fig. 5. This test serves to illustrate the potential gain that can be obtained by using an adapted grid compared to a structured grid.

5.3 High-amplitude solitary waves

In a demanding test we model the propagation of high-amplitude solitary waves reflected on a vertical solid wall. This experiment demands a model capable of capturing both nonlinear and dispersive characteristics. A highly accurate initial condition for the solitary waves can be computed using the method by Tanaka [36]. From the initial starting point, which is assumed to be sufficiently far from the walls for the boundary conditions to be satisfied, the wave will propagate inside the computational domain toward the left wall. Within a distance of approximately $2d$ from the wall measured from the wave crest, the wave crest will start to accelerate leading to a tall and thin jet shooting up in close proximity to the wall. This effect is a

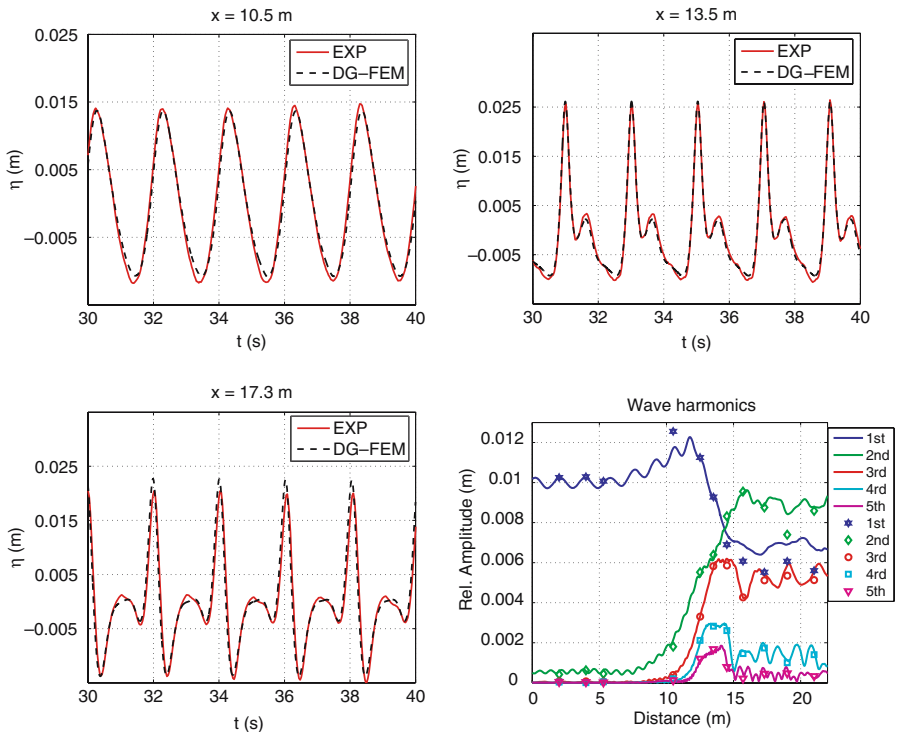

Fig. 5 Computed and measured time series of free-surface elevations for case A at three gauge positions; *before*, *at*, and *after* the top of the submerged bar. A *structured* grid is used. The last figure shows a comparison between computed and experimental data by a time-harmonics analysis

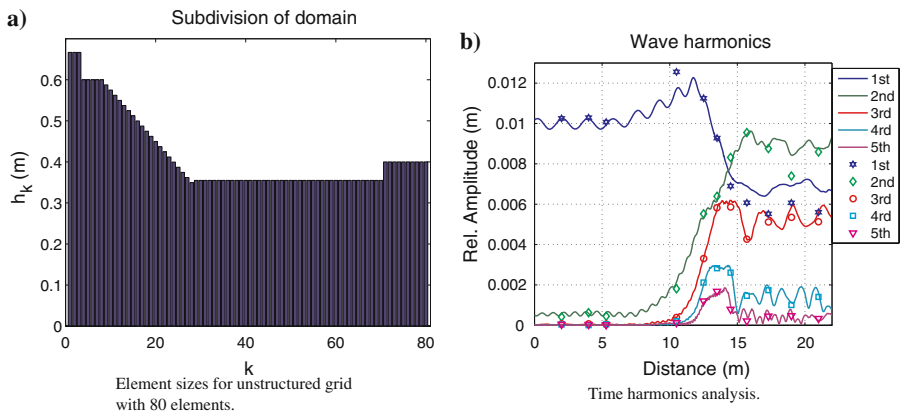

Fig. 6 Submerged bar test using an *unstructured* grid adapted to the physical scales

purely nonlinear. As concluded by Cooker et al. [37] on the basis of experimental data and in agreement with their numerical results, the phase-change during the reflection is spatially dependent and hence these waves are not solitons, as a (usually minor) loss of energy occurs, during the reflection of the solitary waves. Therefore, the wave cannot maintain its original shape after the reflection. However, the wave will again take the form of a solitary wave after the reflection; although with slightly smaller amplitude, and this adjustment will result in a dispersive trail.

Similar to the results obtained by Madsen et al. [6], we present results for the attachment, detachment and max wave run up amplitudes. Numerical simulations using *structured* grids have been carried out for solitary waves with varying wave steepness. For solitary waves with dimensionless amplitudes $a/d \leq 0{\cdot}5$, a mild nodal filter with $\alpha = 0{\cdot}1 - 0{\cdot}2$ have been applied every time step. Here a is the height from the undisturbed water surface to the crest of the solitary wave. For the more extreme and steep solitary waves, the SG-polynomial averaging (nodal) filter has been applied every time step with stencil half size $\alpha = 6$ and filter order $p = 8$. In all simulations, a local polynomial order of $N = 8$ has been used. The spatial resolution needed increases with the magnitude of the solitary wave amplitude. The number of elements ranged from $K = 40-100$ for amplitudes $a/d = 0{\cdot}1-0{\cdot}6$ for the waves to be considered well-resolved in a computational domain $x \in [0, 45]$m. For the high-amplitude solitary waves to be accurately resolved, a model capable of adapting the grid "on the fly" to allow for changes in the spatial resolution where it is needed would be more appropriate in terms of computational efficiency. However, this is not possible in the current application.

As observed by both Cooker et al. [37] and Madsen et al. [6], it is difficult to model the thin jet shooting up numerically due to the rapid and subsequent collapse of the jet afterwards, which demands sufficient spatial resolution in this region. However, we introduced a small amount of artificial viscosity in the vicinity of the wall in order to suppress spurious oscillations, and thus we are able to continue the detachment curve where previous results have not been obtained in Fig. 7a even for the high-amplitude cases. Further, as can be seen in Fig. 7b, the numerical data for the non-dimensional wall forces at the left wall are in excellent agreement up to nearly $a/d = 0{\cdot}6$. Around this limit, it it found necessary to introduce the beforehand mentioned artificial viscosity.

6 Concluding remarks

We have presented a methodology based on the DG-FEM for solving high-order Boussinesq-type equations, which supports the use of unstructured meshes. This has potential benefits in terms of possible reductions of the degrees of freedom and local adaptivity in complex geometries.

It has been shown that the methodology can be used to produce accurate results for demanding test cases describing relevant dispersive and nonlinear wave phenomena.

Reducing the computational effort remains a future task. The computational bottleneck in solving the system of equations is to both reconstruct and solve the involved algebraic system efficiently. Hence, it is crucial for large-scale applications that an efficient sparse solver is used to enable the model to be cost-

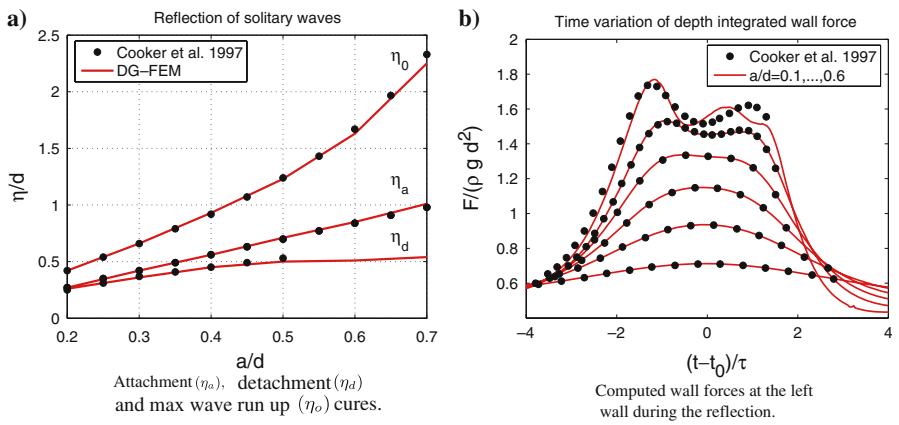

Fig. 7 Reflection of high-amplitude solitary waves

efficient. Currently, the methodology is based on simple approximations in a collocation manner possibly leading to aliasing errors. These aliasing errors can be removed by introducing frequency-dependent artificial viscosity by, e.g., nodal filtering or Savitzky–Golay polynomial averaging filtering.

The current application is only capable of adapting the grid to the initial geometry of the domain. For some problems, e.g. high-amplitude solitary waves problems, it is more convenient to be able to adapt the grid during the simulation based on for example steep gradients in the solution.

We seek to test the methodology for fully nonlinear problems in two horizontal dimensions and a thorough description along with more results will appear in a future paper.

7 Appendix A: Differential operators

A number of differential operators arise in the derivation procedure for the high-order Boussinesq-type formulation presented in Sect. 2. The definitions of these continuous differential operators for use with the velocity formulation in one and two horizontal dimensions are given here fore the Padé (4,4) version. The Padé (2,2) version is straightforwardly obtained by changing the coefficients to those given in the next section.

7.1 Coefficients for two different versions

The coefficients correspond to those for the truncated and Padé enhanced linear operators of the exact infinite-series solution for the velocity field satisfying the Laplace equation in the interior.

The Padé (4,4) formulation retaining up to fifth-order derivatives given in Madsen et al. [6] result in the following coefficients

$$\alpha_2 \equiv \frac{(z-\hat{z})^2}{2} - \frac{\hat{z}^2}{18}, \tag{57}$$

$$\alpha_4 \equiv \frac{(z-\hat{z})^4}{24} - \frac{\hat{z}^2(z-\hat{z})^2}{36} + \frac{\hat{z}^4}{504}, \tag{58}$$

$$\beta_1 \equiv z-\hat{z}, \tag{59}$$

$$\beta_3 \equiv \frac{(z-\hat{z})^3}{6} - \frac{\hat{z}^2(z-\hat{z})}{18}, \tag{60}$$

$$\beta_5 \equiv \frac{(z-\hat{z})^5}{120} - \frac{\hat{z}^2(z-\hat{z})^3}{108} + \frac{\hat{z}^4(z-\hat{z})}{504}, \tag{61}$$

$$\gamma_2 \equiv \frac{4}{9}, \quad \gamma_3 \equiv \frac{1}{9}, \quad \gamma_4 \equiv \frac{1}{63}, \quad \gamma_5 \equiv \frac{1}{945}. \tag{62}$$

The optimized parameters leading to a minimization of the linear shoaling errors over $kd \in [0, 30]$ ($\hat{z} = -0{\cdot}5d$) are given as

$$c_2 = 0{\cdot}357739, \quad c_4 = 0{\cdot}00663819,$$
$$s_3 = 0{\cdot}0753019, \quad s_5 = -6{\cdot}31532 \times 10^{-5}. \tag{63}$$

The Padé (2,2) formulation retaining up to third-order derivatives is given as

$$\alpha_2 \equiv \frac{(z-\hat{z})^2}{2} - \frac{\hat{z}^2}{10}, \tag{64}$$

$$\alpha_4 \equiv 0, \tag{65}$$

$$\beta_1 \equiv z-\hat{z}, \tag{66}$$

$$\beta_3 \equiv \frac{(z-\hat{z})^3}{6} - \frac{\hat{z}^2(z-\hat{z})}{10}, \tag{67}$$

$$\beta_5 \equiv 0, \tag{68}$$

$$\gamma_2 \equiv \frac{2}{5}, \quad \gamma_3 \equiv \frac{1}{15}, \quad \gamma_4 \equiv 0, \quad \gamma_5 \equiv 0. \tag{69}$$

The optimized parameters leading to a minimization of the linear shoaling errors over $kd \in [0,6]$ ($\hat{z} = -0{\cdot}5d$) presented in Jamois et al. [7] are given as

$$c_2 = -0{\cdot}0593982, \quad s_3 = -0{\cdot}00113222. \tag{70}$$

Further, the height of the water-column from the bottom to the chosen expansion level is defined as

$$\lambda \equiv d + \hat{z}. \tag{71}$$

7.2 Continuous differential operators

The continuous differential operators are derived from a set of power rules for the ∇-operator, and is the result of truncation of the exact and subsequently enhanced exact infinite series solution to the velocity field in the interior.

7.2.1 One horizontal dimension

$$A_2 = 1 - \alpha_2(\partial_{xx}) + \alpha_4(\partial_{xxxx}), \tag{72}$$

$$A_3 = \lambda(\partial_x) - \gamma_3\lambda^3(\partial_{xxx}) + \gamma_5\lambda^5(\partial_{xxxxx}), \tag{73}$$

$$B_2 = \beta_1(\partial_x) - \beta_3(\partial_{xxx}) + \beta_5(\partial_{xxxxx}), \tag{74}$$

$$B_3 = 1 - \gamma_2\lambda^2(\partial_{xx}) + \gamma_2\lambda^4(\partial_{xxxx}), \tag{75}$$

$$S_1 = 1 - c_2\lambda^2(\partial_{xx}) + c_4\lambda^4(\partial_{xxxx}), \tag{76}$$

$$S_2 = \lambda(\partial_x) - s_3\lambda^3(\partial_{xxx}) + s_5\lambda^5(\partial_{xxxxx}). \tag{77}$$

7.2.2 Two horizontal dimensions

$$A_{01} = \lambda\partial_x - \gamma_3\lambda^3(\partial_{xxx} + \partial_{xyy}) + \gamma_5\lambda^5(\partial_{xxxxx} + 2\partial_{xxxyy} + \partial_{xyyyy}), \tag{78}$$

$$A_{02} = \lambda\partial_y - \gamma_3\lambda^3(\partial_{xxy} + \partial_{yyy}) + \gamma_5\lambda^5(\partial_{xxxxy} + 2\partial_{xxyyy} + \partial_{yyyyy}), \tag{79}$$

$$A_1 = 1 - \alpha_2(\partial_{xx} + \partial_{yy}) + \alpha_4(\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}), \tag{80}$$

$$A_{11} = 1 - \alpha_2(\partial_{xx}) + \alpha_4(\partial_{xxxx} + \partial_{xxyy}), \tag{81}$$

$$A_2 = -\alpha_2(\partial_{xy}) + \alpha_4(\partial_{xxxy} + \partial_{xyyy}), \tag{82}$$

$$A_{22} = 1 - \alpha_2(\partial_{yy}) + \alpha_4(\partial_{xxyy} + \partial_{yyyy}), \tag{83}$$

$$B_0 = 1 - \gamma_2\lambda^2(\partial_{xx} + \partial_{yy}) + \gamma_4\lambda^4(\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}), \tag{84}$$

$$B_{11} = \beta_1(\partial_x) - \beta_3(\partial_{xxx} + \partial_{xyy}) + \beta_5(\partial_{xxxxx} + 2\partial_{xxxyy} + \partial_{xyyyy}), \tag{85}$$

$$B_{12} = \beta_1(\partial_y) - \beta_3(\partial_{xxy} + \partial_{yyy}) + \beta_5(\partial_{xxxxy} + 2\partial_{xxyyy} + \partial_{yyyyy}), \tag{86}$$

$$C_{11} = 1 - c_2\lambda^2(\partial_{xx}) + c_4\lambda^4(\partial_{xxxx} + \partial_{xxyy}), \tag{87}$$

$$C_2 = -c_2\lambda^2(\partial_{xy}) + c_4\lambda^4(\partial_{xxxy} + \partial_{xyyy}), \tag{88}$$

$$C_{13} = \lambda(\partial_x) - s_3\lambda^3(\partial_{xxx} + \partial_{xyy}) + s_5\lambda^5(\partial_{xxxxx} + 2\partial_{xxxyy} + \partial_{xyyyy}), \tag{89}$$

$$C_{22} = 1 - c_2\lambda^2(\partial_{yy}) + c_4\lambda^4(\partial_{xxyy} + \partial_{yyyy}), \tag{90}$$

$$C_{23} = \lambda(\partial_y) - s_3\lambda^3(\partial_{xxy} + \partial_{yyy}) + s_5\lambda^5(\partial_{xxxxy} + 2\partial_{xxyyy} + \partial_{yyyyy}). \tag{91}$$

Acknowledgements The first author would like to thank the Division of Applied Mathematics at Brown University (Providence, USA) for an inspiring visit in the period from Sep 2004 to Mar 2005. Further, the helpful discussions with Professor Chi-Wang Shu of Brown University, USA, Claes Eskilsson of Chalmars University of Technology, Sweden, and David Fuhrman of Technical University of Denmark have been highly appreciated. The work of JSH was partially supported by NSF under contract DMS-01-32967.

References

1. Madsen PA, Schäffer HA (1999) A review of Boussinesq-type equations for gravity waves. Adv Coastal Ocean Engng. 5:1–95
2. Peregrine DH (1967) Long waves on a beach. J Fluid Mech 27:815–827
3. Madsen PA, Sørensen OR (1992) A new form of the Boussinesq equations with improved linear dispersion characteristics. Part 2. A slowly varying bathymetry. Coastal Engng 18:183–204
4. Nwogu O (1993) Alternative form of Boussinesq equations for nearshore wave propagation. J Waterway, Port, Coastal and Ocean Engng, 119(6):618–638
5. Madsen PA, Agnon Y (2003) Accuracy and convergence of velocity formulations for water waves in the framework of Boussinesq theory. J Fluid Mech 477:285–319
6. Madsen PA, Bingham HB, Liu H (2002) A new Boussinesq method for fully nonlinear waves from shallow to deep water. J Fluid Mech 462:1–30
7. Jamois E, Fuhrman DR, Bingham HB, Molin B (2006) A numerical study of nonlinear wave run-up on a vertical plate. Coastal Engng (to appear)
8. Fuhrman DR, Bingham HB (2004) Numerical solutions of fully non-linear and highly dispersive Boussinesq equations in two horizontal dimensions. Int J Num Meth Fluids 44:655–672
9. Fuhrman DR, Bingham HB, Madsen PA (2005) Nonlinear wave-structure interactions with a high-order Boussinesq model. Coastal Engng 52:655–672
10. Antunes Do Carmo JS, Seabra Santos FJ, Barthèlemy E (1993) Surface waves propagation in shallow water: a finite element model. Int J Num Meth Fluids 16:447–459
11. Ambrosi D, Quartapelle L (1998) A Taylor–Galerkin method for simulating nonlinear dispersive water waves. J Comp Phys 146:546–569
12. Sherwin SJ, Eskilsson C (2006) Spectral/*hp* discontinuous Galerkin methods for modelling 2D Boussinesq equations. J Comp Phys 212:566–589
13. Langtangen HP, Pedersen G (1998) Computational models for weakly dispersive nonlinear water waves. Comput Methods Appl Engng 160:337–358
14. Li YS, Liu S-X, Yu Y-X, Lai G-Z (1999) Numerical modelling of Boussinesq equations by finite element method. Coastal Engng 37:97–122
15. Beji S, Nadaoka K (1996) A formal derivation and numerical modelling of the improved Boussinesq equations for varying depth. Ocean Engng 23:691–704
16. Sørensen OR, Schäffer HA, Sørensen LS (2004) Boussinesq-type modelling using an unstructured finite element technique. Coastal Engng 50:181–198
17. Walkley MA, Berzins M (1999) A finite element method for the one-dimensional extended Boussinesq equations. Int J Num Meth Fluids 29:143–157
18. Walkley MA, Berzins M (2002) A finite element method for the two-dimensional extended Boussinesq equations. Int J Num Meth Fluids 39:865–885
19. Eskilsson C, Sherwin SJ (2003) An *hp*/spectral element model for efficient long-time integration of Boussinesq-type equations. Coastal Engng 45:295–320
20. Eskilsson C, Sherwin SJ (2005) An unstructured spectral/*hp* element model for enhanced Boussinesq-type equations. Submitted to Coastal Engng
21. Eskilsson C, Sherwin SJ (2002) A discontinuous spectral element model for Boussinesq-type equations. J Sci Comp 17:143–152
22. Eskilsson C, Sherwin SJ (2005) Discontinuous Galerkin spectral/*hp* element modelling of dispersive shallow water systems. J Sci Comp 22:269–288
23. Fuhrman DR, Bingham HB, Madsen PA, Thomsen PG (2004) Linear and non-linear stability analysis for finite difference discretizations of high-order Boussinesq equations. Int J Num Meth Fluids 45:751–773
24. Cockburn B, Shu C-W (2001) Runge-Kutta Discontinuous Galerkin methods for convection-dominated Problems. J Sci Comp 16(3):173–261
25. Hesthaven JS, Warburton T (2002) High-order nodal methods on unstructured grids. I. Time-domain solution of Maxwell's equations. J Comp Phys 181(1):186–221
26. Sherwin S (1999) Dispersion analysis of the continuous and discontinuous Galerkin formulations. In: Cockburn B, Karniadakis GE, Shu C-W (eds) Lecture notes in computational science and engineering: discontinuous Galerkin methods-theory, computation and applications. Springer, pp 425–432

27. Bassi F, Rebay S (1997) A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J Comp Phys 131:267–279
28. Davis TA (2005) UMFPACK version 4.6 user guide. University of Florida pp 1–133
29. Carpenter MH, Kennedy CA (1994) Fourth order 2*N*-storage Runge-Kutta scheme. Technical report NASA-TM-109112, NASA Langley Research Center, VA
30. Fischer P, Mullen JS (2001) Filter-based stabilization of spectral element methods. C R Acad Sci Paris 332:265–270
31. Savitzky A, Golay MJE (1964) Smoothening and differentiation of data by simplified least squares procedures. Anal Chem 36:1627–1639
32. Larsen J, Dancy H (1983) Open boundaries in short wave simulations - a new approach. Coastal Engng 7:285–297
33. Beji S, Battjes JA (1994) Numerical simulation of nonlinear-wave propagation over a bar. Coastal Engng 23:1–16
34. Luth HR, Klopman B, Kitou N (1994) Projects 13G: kinematics of waves breaking partially on an offshore bar: LDV measurements for waves with and without a net onshore current. Technical report H1573, Delft Hydraulics
35. Gobbi MF, Kirby JT (1999) Wave evolution over submerged sills: tests of a high-order Boussinesq model. Coastal Engng 37:57–96
36. Tanaka M (1986) The stability of solitary waves. J Phys Fluids 29:650–655
37. Cooker MJ, Weidman PD, Bale DS (1997) Reflection of a high-amplitude solitary wave at a vertical wall. J Fluid Mech 342:141–158