Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 17–39
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Space–time discontinuous Galerkin method for nonlinear water waves

J.J.W. van der Vegt *, Yan Xu

Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands

Received 17 September 2006; received in revised form 27 November 2006; accepted 29 November 2006
Available online 19 January 2007

Abstract

A space–time discontinuous Galerkin (DG) finite element method for nonlinear water waves in an inviscid and incompressible fluid is presented. The space–time DG method results in a conservative numerical discretization on time dependent deforming meshes which follow the free surface evolution. The algorithm is higher order accurate, both in space and time, and closely related to an arbitrary Lagrangian Eulerian (ALE) approach. A detailed derivation of the numerical algorithm is given including an efficient procedure to solve the nonlinear algebraic equations resulting from the space–time discretization. Numerical examples are shown on a series of model problems to demonstrate the accuracy and capabilities of the method.

Keywords: Space–time discontinuous Galerkin method; Nonlinear waves; Free surfaces; Elliptic–hyperbolic partial differential equations; Arbitrary Lagrangian Eulerian method

1. Introduction

In many applications the evolution of large amplitude water waves is dominated by gravity and the fluid can be considered inviscid and incompressible, with the velocity field irrotational. This allows the introduction of a potential function which satisfies the Laplace equation together with nonlinear boundary conditions at the free surface. The position of the free surface is unknown and must be determined as part of the solution process. The solution of free surface problems is, however, nontrivial and finite element methods provide an excellent technique to solve these problems, in particular when one is interested in higher order accuracy. In addition, they allow the use of efficient (iterative) sparse matrix solvers and can also be extended to more complex physical models, e.g. flows with vorticity and/or viscosity.

An important aspect in the numerical discretization of water wave problems is that the motion of the free surface requires that the computational mesh is continuously updated in order to follow the free surface evolution. This requires that the finite element algorithm can efficiently and accurately deal with

* Corresponding author.
E-mail addresses: j.j.w.vandervegt@math.utwente.nl (J.J.W. van der Vegt), y.xu@math.utwente.nl (Y. Xu).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.11.031

deforming elements. The main challenge in computing large amplitude water waves is that one needs a highly accurate numerical scheme in order to simulate waves with minimal dispersion and dissipation errors for a long period of time, while the nonlinearity and mesh deformation can easily introduce numerical instabilities.

In this article we discuss a new space–time discontinuous Galerkin (DG) method for nonlinear water wave problems. In this method basis functions are used which are discontinuous both in space and time and the problem is directly considered in four dimensional space. This approach is well suited for unsteady free surface problems since the numerical discretization remains conservative on time dependent deforming meshes. The space–time DG discretization also results in an extremely local, element based discretization, which is beneficial for parallel computing and maintaining high order accuracy on unstructured meshes. In particular, space–time methods are well suited for *hp*-adaptation, which consists of local mesh refinement and/or the adjustment of the polynomial order in individual elements. The main motivation for the algorithm discussed in this article originates from the space–time DG techniques which we developed for compressible flows [13,25,26,28] and the DG technique for linear free surface water waves [27]. These two approaches are combined in this article and provide a new way to compute nonlinear water waves. More general information about DG methods for elliptic and hyperbolic partial differential equations can be found in [4,8–10].

The study of water waves has interested researchers over many years. Overviews of the theoretical analysis of free surface waves can be found in for instance [15,20,30]. Finite element methods for nonlinear water wave problems modeled by the Laplace equation have been developed in [7,16,17,29,31,32], and for the closely related spectral element methods in [11,21,22]. A particularly important technique is provided by the so called arbitrary Lagrangian Eulerian (ALE) method. The ALE technique provides more flexibility in choosing the velocity of the mesh points in the mesh deformation algorithm, which only needs to be tangential to the free surface and not necessarily equal to the fluid particle velocity. This makes it much easier to maintain a reasonable mesh quality during the mesh deformation process. The ALE method is extensively used for many applications requiring moving interfaces and boundaries and applied to free surface waves in for instance [5,12,18,23,24]. The space–time discontinuous Galerkin finite element method discussed in this article also belongs to the class of ALE schemes, but this is not immediately obvious. For an explanation of the link between space–time and ALE methods see [19,26]. The space–time DG method has so far only been applied to shallow water flows [1–3], including the wetting and drying of sand banks and beaches, but not to nonlinear water waves modeled by the Laplace equation.

The article is organized as follows. In Section 2, we present the equations governing nonlinear water waves. The geometry of the space–time domain, some necessary function spaces and trace operators are defined in Section 3. In Section 4, we derive the weak formulation for the space–time discontinuous Galerkin finite element discretization for nonlinear water waves. This formulation results in a large system of nonlinear algebraic equations which are solved with a Newton method. The main aspects of this algorithm are discussed in Section 5, whereas the details are given in the appendix. Section 6 contains numerical results for linear and nonlinear problems to demonstrate the accuracy and capabilities of the method. Finally, some concluding remarks are drawn in Section 7.

2. Equations governing nonlinear water waves

Assume that the fluid is incompressible and inviscid, with the velocity field irrotational. The flow domain $\Omega(t) \subset \mathbb{R}^3$ at time t is bounded by a free surface Γ_S, a solid boundary Γ_N, and in certain cases also a periodic boundary Γ_P, such that $\overline{\Gamma}_S \cup \overline{\Gamma}_N \cup \overline{\Gamma}_P = \partial\Omega$. We assume that Γ_S, Γ_N and Γ_P are nonoverlapping and Lipschitz continuous, with Γ_S a nonzero surface measure. The free surface is defined as

$$\Gamma_S := \{(t,x,y,z) \in \mathbb{R}^4 | f(t,x,y,z) = 0\} \tag{2.1}$$

with x, y, z the spatial coordinates in a standard Cartesian coordinate system. We assume that the flat water surface is given by $z = 0$, with the z-direction pointing upward from the flat water surface.

We introduce a velocity potential $\phi(t,x,y,z)$, with $u = \overline{\nabla}\phi$ the fluid velocity, where $\overline{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^{\mathrm{T}}$ denotes the nabla-operator. We make the various parameters dimensionless by redefining them as

$$\phi \to H\sqrt{Hg_c}\phi, \quad (x,y,z) \to H(x,y,z), \quad t \to \sqrt{\frac{H}{g_c}}t, \quad \text{and} \quad u \to u\sqrt{Hg_c},$$

where g_c is the gravitational constant and H the average water depth. The potential function ϕ for nonlinear water waves then must satisfy for $t \in (t_0, T)$ the Laplace equation:

$$-\overline{\nabla} \cdot \overline{\nabla}\phi = 0 \quad \text{in } \Omega(t). \tag{2.2}$$

In addition, we have the following boundary conditions. At solid surfaces we impose the boundary condition:

$$\bar{n} \cdot \overline{\nabla}\phi = g_N(t) \quad \text{at } \Gamma_N(t) \tag{2.3}$$

with $\bar{n} \in \mathbb{R}^3$ the unit outward normal vector to $\partial\Omega$ and $g_N : \Gamma_N \to \mathbb{R}$ the prescribed normal velocity at Γ_N.

The nonlinear free surface boundary conditions consist of the dynamic condition:

$$\frac{\partial\phi(t,x,y,z)}{\partial t} + \frac{1}{2}\overline{\nabla}\phi(t,x,y,z) \cdot \overline{\nabla}\phi(t,x,y,z) + z = 0, \quad \forall (x,y,z) \in \Gamma_S(t) \tag{2.4}$$

and the kinematic condition:

$$\frac{\partial f(t,x,y,z)}{\partial t} + \overline{\nabla}\phi(t,x,y,z) \cdot \overline{\nabla} f(t,x,y,z) = 0, \quad \forall (x,y,z) \in \Gamma_S(t). \tag{2.5}$$

For certain problems we also consider periodic boundary conditions at Γ_P:

$$\phi(t, x + L_x, y + L_y, z) = \phi(t,x,y,z). \tag{2.6}$$

Here L_x, L_y are the length of the periodic domain in the x- and y-direction, respectively. If the waves are not overturning then we can introduce a wave height ζ, such that

$$f(t,x,y,z) = \zeta(t,x,y) - z = 0. \tag{2.7}$$

As initial conditions, we either start without any waves with $\phi(t_0,x,y,z) = \zeta(t_0,x,y) = 0$ and the waves are generated by a wave maker; or, we start with an analytic wave field and ϕ, ζ are known at initial time.

Note, if we change $\phi \to \phi + c$, with $c \in \mathbb{R}$ an arbitrary constant, then the equations and boundary conditions (2.2)–(2.6) remain unchanged. Hence, the potential ϕ is determined up to an arbitrary constant.

3. Space–time domain, function spaces and traces

Since the domain boundaries $\partial\Omega$ are time dependent and need to be determined as part of the solution, it is beneficial to introduce a space–time formulation. In a space–time formulation no distinction is made between space and time and the discretization is obtained directly in $\mathbb{R}^4$. The key benefit of this approach is that the numerical scheme will be exactly conservative on a moving and deforming mesh. Consider a domain $\mathscr{E} \subset \mathbb{R}^4$. A point $x \in \mathbb{R}^4$ has coordinates $(t, \bar{x})$, with $\bar{x} = (x,y,z)$. The space–time domain boundary $\partial\mathscr{E}$ consists of the hypersurfaces $\Omega_0 := \{x \in \partial\mathscr{E} | t = t_0\}$, $\Omega_T := \{x \in \partial\mathscr{E} | t = T\}$ and $\mathcal{Q} := \{x \in \partial\mathscr{E} | t_0 < t < T\}$. The space–time normal vector at $\partial\mathscr{E}$ is defined as $n = (n_t, \bar{n})^{\mathrm{T}}$, with n_t the temporal component and $\bar{n}$ the spatial component. The space–time boundary $\mathcal{Q}$ is further subdivided as $\mathcal{Q} = \Gamma_{\mathscr{S}} \cup \Gamma_{\mathscr{N}} \cup (\Gamma_P \times (t_0, T))$, with $\Gamma_{\mathscr{S}}$ and $\Gamma_{\mathscr{N}}$ describing the evolution of the free surface and solid boundary in $\mathbb{R}^4$, respectively, such that $\Gamma_S(t) = \Gamma_{\mathscr{S}} \cap \{t\}$ and $\Gamma_N(t) = \Gamma_{\mathscr{N}} \cap \{t\}$.

The space–time domain is subdivided into a finite number of space–time slabs, which are constructed as follows. Consider the time interval $\mathscr{I} = [t_0, T]$, partitioned by an ordered series of time levels $t_0 < t_1 < \cdots < t_N = T$. We denote the nth time interval $I_n = (t_n, t_{n+1})$, with $\mathscr{I} = \cup_n \overline{I}_n$. A space–time slab is now defined as the domain $\mathscr{E}^n := \mathscr{E} \cap I_n$, with boundaries $\Omega(t_n)$, $\Omega(t_{n+1})$ and $\mathcal{Q}^n = \partial\mathscr{E}^n \setminus (\Omega(t_n) \cup \Omega(t_{n+1}))$.

In each space–time slab we introduce a finite element tessellation $\mathscr{T}_h^n$ with space–time elements $\mathscr{K}$. Let $\Omega_h(t_n)$ be subdivided into $N_{\mathscr{T}}$ nonoverlapping shape regular spatial elements K^n, such that $\Omega_h(t_n) \to \Omega(t_n)$ as $h \to 0$, with h the radius of the smallest sphere containing all elements K_j^n, $j = 1, \ldots, N_{\mathscr{T}}$. At t_{n+1} the spatial elements K^{n+1} are obtained by mapping the vertices of the elements K^n to their position at t_{n+1}. The space–time elements $\mathscr{K} \in \mathscr{T}_h^n$ are then obtained by interpolation in time between the elements K^n and K^{n+1}, such that $\mathscr{E}_h^n = \cup_{\mathscr{K} \in \mathscr{T}_h^n} \mathscr{K} \to \mathscr{E}^n$ as $h \to 0$. More details about the specific element mappings will be given in Section 5.

In the space–time slab we consider the following faces. Let Γ^n denote the union of the faces of all elements $\mathscr{K} \in \mathscr{T}_h^n$, i.e. $\Gamma^n = \bigcup_{\mathscr{K}\in\mathscr{T}_h^n} \partial\mathscr{K}$, $\Gamma_0^n = \Gamma^n \setminus \partial\mathscr{E}_h^n$ the union of all interior faces, and $\Gamma_{\mathscr{Q}}^n := \Gamma^n \setminus (\Omega(t_n) \cup \Omega(t_{n+1}))$ the union of all interior faces and faces at $\mathscr{Q}_h := \partial\mathscr{E}_h^n \setminus (\Omega(t_n) \cup \Omega(T_{n+1}))$. We denote the set of all faces in $\mathscr{T}_h^n$ by $\{\mathscr{F}^n\}$, all interior faces by $\{\mathscr{F}_I^n\}$, all faces on $\Gamma_{\mathscr{N}}^n := \Gamma_{\mathscr{N}} \cap I_n$ by $\{\mathscr{F}_{\mathscr{N}}^n\}$, all faces on $\Gamma_{\mathscr{S}}^n := \Gamma_{\mathscr{S}} \cap I_n$ by $\{\mathscr{F}_{\mathscr{S}}^n\}$ and all faces on $\partial\mathscr{E}_h^n$ by $\{\mathscr{F}_{\partial\mathscr{E}}^n\}$. Note, faces at $\Gamma_P \times (t_0, T)$ are considered interior faces, where the periodicity relation (2.6) is used to connect the external part of Γ_P to the interior of the domain Ω.

We define the finite element spaces V_p^n and Σ_p^n associated with the tessellation $\mathscr{T}_h^n$ as

$$V_p^n := \left\{ v \in L^2(\mathscr{E}_h^n) \middle| v|_{\mathscr{K}} \in \mathscr{P}_p(\mathscr{K}), \quad \forall \mathscr{K} \in \mathscr{T}_h^n \right\},$$
$$\Sigma_p^n := \left\{ \sigma \in [L^2(\mathscr{E}_h^n)]^3 \middle| \sigma|_{\mathscr{K}} \in [\mathscr{P}_p(\mathscr{K})]^3, \quad \forall \mathscr{K} \in \mathscr{T}_h^n \right\}$$

with $L^2(\mathscr{E}_h^n)$ the space of Lebesgue square integrable functions on $\mathscr{E}_h^n$. The finite element space W_p^n associated with the free surface is defined as

$$W_p^n := \left\{ v \in L^2(\Gamma_{\mathscr{S}}^n) \middle| v|_{\mathscr{F}_{\mathscr{S}}^n} \in \mathscr{P}_p(\mathscr{F}_{\mathscr{S}}^n), \quad \forall \mathscr{F}_{\mathscr{S}}^n \subset \Gamma_{\mathscr{S}}^n \right\},$$

where $L^2(\Gamma_{\mathscr{S}}^n)$ is the space of Lebesgue square integrable functions on $\Gamma_{\mathscr{S}}^n$ and $\mathscr{P}_p$ polynomials of order p.

Next, we define some trace operators to manipulate the numerical fluxes in the discontinuous Galerkin formulation. For $v \in V_p^n$ we define the average $\langle v \rangle$ and jump $[[v]]$ operators of v at an internal face $\mathscr{F} \in \mathscr{F}_I^n$ as follows:

$$\langle v \rangle := \frac{1}{2}(v_L + v_R), \quad [[v]] := v_L \bar{n}_L + v_R \bar{n}_R \tag{3.1}$$

with $v_L := v|_{\partial\mathscr{K}_L}$ and $v_R := v|_{\partial\mathscr{K}_R}$ and $\mathscr{K}_L$, $\mathscr{K}_R$ the elements connected to the face $\mathscr{F} \in \mathscr{F}_I^n$ with outward space normal vectors $\bar{n}_L$ and $\bar{n}_R$, respectively. For $q \in \Sigma_p^n$ we similarly define q_L and q_R and set:

$$\langle q \rangle := \frac{1}{2}(q_L + q_R), \quad [[q]] := q_L \cdot \bar{n}_L + q_R \cdot \bar{n}_R, \quad \text{at } \mathscr{F} \in \mathscr{F}_I^n. \tag{3.2}$$

For $\mathscr{F} \in \mathscr{F}_{\partial\mathscr{E}}^n$, the set of exterior boundary faces, each $v \in V_p^n$ and $q \in \Sigma_p^n$ has a uniquely defined restriction on $\mathscr{F}$ and we define:

$$[[v]] := v\bar{n}, \quad \langle q \rangle := q \quad \text{at } \mathscr{F} \in \mathscr{F}_{\partial\mathscr{E}}^n. \tag{3.3}$$

Since we do not require either of the quantities $\langle v \rangle$ or $[[q]]$ on boundary faces, we leave them undefined.

For the definition of the primal DG formulation we need to define the following lifting operator $\mathscr{R} : [L^2(\Gamma_{\mathscr{Q}}^n)]^3 \to \Sigma_p^n$:

$$\int_{\mathscr{E}_h^n} \mathscr{R}(q) \cdot \sigma \, \mathrm{d}x = \int_{\Gamma_{\mathscr{Q}}^n} q \cdot \langle \sigma \rangle \, \mathrm{d}s, \quad \forall \sigma \in \Sigma_p^n \tag{3.4}$$

and for a face $\mathscr{F} \in \mathscr{F}^n$ also the local lifting operator $\mathscr{R}_{\mathscr{F}} : [L^2(\mathscr{F})]^3 \to \Sigma_p^n$:

$$\int_{\mathscr{E}_h^n} \mathscr{R}_{\mathscr{F}}(q) \cdot \sigma \, \mathrm{d}x = \int_{\mathscr{F}} q \cdot \langle \sigma \rangle \, \mathrm{d}s, \quad \forall \sigma \in \Sigma_p^n. \tag{3.5}$$

Note, the local lifting operator $\mathscr{R}_{\mathscr{F}}$ is only nonzero in the two elements connected to the face $\mathscr{F}$.

4. Space–time discontinuous Galerkin formulation

In this section we summarize the derivation of the space–time discontinuous Galerkin finite element discretization for nonlinear water waves given by (2.2)–(2.6). We will follow the approach from Brezzi et al. [6], which is analyzed in detail for the Laplace equation in [4] and for linear water waves in [27]. The key benefit of this technique is that the DG formulation does not contain mesh dependent stabilization coefficients, as for instance occur in the interior penalty method.

The novel ingredient in the finite element formulation discussed in this section is to incorporate the kinematic condition at the free surface (2.5) as a natural boundary condition in the finite element formulation. The

space–time finite element discretization then will automatically account for the mesh movement necessary to follow the free surface waves. In order to accomplish this we need to establish a relation between the function f describing the free surface (2.1), the wave height ζ and the space–time normal vector n. A straightforward calculation using (2.7) shows that

$$n = (n_t, \bar{n}) = \frac{\nabla f}{|\nabla f|} = \frac{\left(\frac{\partial f}{\partial t}, \overline{\nabla} f\right)^{\mathrm{T}}}{|\nabla f|} = \frac{1}{|\nabla(\zeta - z)|}\left(\frac{\partial \zeta}{\partial t}, \frac{\partial \zeta}{\partial x}, \frac{\partial \zeta}{\partial y}, -1\right)^{\mathrm{T}} \tag{4.1}$$

with ∇ the space–time nabla operator defined as $\nabla = \left(\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^{\mathrm{T}}$. We can rewrite the kinematic condition (2.5) as

$$\frac{1}{|\nabla f|}\frac{\partial f}{\partial t} + \overline{\nabla}\phi \cdot \frac{\overline{\nabla} f}{|\nabla f|} = 0, \tag{4.2}$$

which implies using (4.1) that the space component of the normal velocity at the free surface $\Gamma_{\mathscr{S}}$ is equal to

$$\bar{n} \cdot \overline{\nabla}\phi = \frac{-1}{|\nabla(\zeta - z)|}\frac{\partial \zeta}{\partial t}. \tag{4.3}$$

This relation can be used directly in the DG formulation of the Laplace equation. For the DG discretization of the Laplace equation (2.2) we rewrite this equation into a first order system:

$$u = \overline{\nabla}\phi, \quad -\overline{\nabla} \cdot u = 0, \qquad \text{in } \Omega, \tag{4.4}$$

where u represents the fluid velocity and ϕ the potential. Multiplying (4.4) with arbitrary test functions $\sigma \in \Sigma_p^n$ and $v \in V_p^n$, integrating by parts with respect to $\bar{x}$ over each element $\mathscr{K} \in \mathscr{T}_h^n$ (twice for (4.5)), and summation over all elements in $\mathscr{T}_h^n$, we obtain for $u_h \in \Sigma_p^n$ and $\phi_h \in V_p^n$ the relation:

$$\int_{\mathscr{E}_h^n} u_h \cdot \sigma \,\mathrm{d}x = \sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\mathscr{K}} \overline{\nabla}_h \phi_h \cdot \sigma \,\mathrm{d}x + \sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\partial \mathscr{K}} (\hat{\phi}_K - \phi_h)\sigma \cdot \bar{n}_K \,\mathrm{d}s, \tag{4.5}$$

$$-\sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\mathscr{K}} (\overline{\nabla}_h \cdot u_h) v \,\mathrm{d}x = \sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\mathscr{K}} u_h \cdot \overline{\nabla}_h v \,\mathrm{d}x - \sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\partial \mathscr{K}} \bar{n}_K \cdot \hat{u}_K v \,\mathrm{d}s = 0 \tag{4.6}$$

for $\forall \sigma \in \Sigma_p^n$ and $\forall v \in V_p^n$, where $\overline{\nabla}_h = \overline{\nabla}|_{\mathscr{K}}$ for all $\mathscr{K} \in \mathscr{T}_h$. Here, the numerical fluxes $\hat{u}(u_h)$ and $\widehat{\sigma}(\overline{\nabla}_h u_h, \sigma_h)$ are introduced to account for the multivalued traces at Γ. The different discontinuous Galerkin formulations proposed so far in the literature can all be derived by specifying these numerical fluxes. A full account is given in [4]. In order to simplify the sums over the element boundaries $\partial\mathscr{K}$, we use a slightly modified identity (3.3) in [4], which states for $q_{\mathscr{K}} \in \mathbb{R}^3$:

$$\sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\partial \mathscr{K}} \varphi_{\mathscr{K}} (0, q_{\mathscr{K}})^{\mathrm{T}} \cdot n_{\mathscr{K}} \,\mathrm{d}s = \sum_{\mathscr{K} \in \mathscr{T}_h^n} \int_{\partial \mathscr{K}} \varphi_{\mathscr{K}} q_{\mathscr{K}} \cdot \bar{n}_{\mathscr{K}} \,\mathrm{d}s, \tag{4.7}$$

$$= \int_{\Gamma_{\mathscr{Q}}^n} [[\varphi]] \cdot \langle q \rangle \,\mathrm{d}s + \int_{\Gamma_0^n} \langle \varphi \rangle [[q]] \,\mathrm{d}s \tag{4.8}$$

and can be proved by a straightforward calculation using the fact that at $\Omega(t_n)$ and $\Omega(t_{n+1})$ the space–time normal vector n is equal to $n = (\pm 1, 0, 0, 0)^{\mathrm{T}}$. Note, this relation also implies that there is no direct coupling between the different space–time slabs. If we introduce (4.8) into (4.5) and (4.6) we obtain:

$$\int_{\mathscr{E}_h^n} u_h \cdot \sigma \,\mathrm{d}x = \int_{\mathscr{E}_h^n} \overline{\nabla}_h \phi_h \cdot \sigma \,\mathrm{d}x + \int_{\Gamma_{\mathscr{Q}}^n} [[\hat{\phi} - \phi_h]] \cdot \langle \sigma \rangle \,\mathrm{d}s + \int_{\Gamma_0^n} \langle \hat{\phi} - \phi_h \rangle [[\sigma]] \,\mathrm{d}s, \tag{4.9}$$

$$\int_{\mathscr{E}_h^n} u_h \cdot \overline{\nabla}_h v \,\mathrm{d}x - \int_{\Gamma_{\mathscr{Q}}^n} [[v]] \cdot \langle \hat{u} \rangle \,\mathrm{d}s - \int_{\Gamma_0^n} \langle v \rangle [[\hat{u}]] \,\mathrm{d}s = 0. \tag{4.10}$$

The numerical fluxes in the DG formulation from Brezzi et al. [6] are defined as

$$\hat{\phi} = \begin{cases} \langle \phi_h \rangle & \text{on } \Gamma_0^n, \\ \phi_h & \text{on } \Gamma_{\mathscr{N}}^n \cup \Gamma_{\mathscr{S}}^n, \end{cases} \tag{4.11}$$

$$\hat{u} = \langle u_h \rangle - \alpha_r([[\phi_h]]) \quad \text{on } \Gamma_0^n, \tag{4.12}$$

$$\hat{u} = u \quad \text{on } \Gamma_{\mathscr{N}}^n \cup \Gamma_{\mathscr{S}}^n, \tag{4.13}$$

where $\alpha_r(q) = \eta_{\mathscr{F}} \langle \mathscr{R}_{\mathscr{F}}(q) \rangle$ for $\mathscr{F} \in \mathscr{F}_I^n$, $\eta_{\mathscr{F}} \in \mathbb{R}^+$ a positive number. If we introduce the lifting operator (3.4) into (4.9) and use (4.11) in the last contribution, then we obtain the following equation for u_h, almost everywhere:

$$u_h = \overline{\nabla}_h \phi_h + \mathscr{R}([[\hat{\phi} - \phi_h]]). \tag{4.14}$$

The weak formulation for the potential ϕ_h is now obtained by introducing the relation (4.14) for u_h into (4.10):

Find a $\phi_h \in V_p^n$, such that for all $v \in V_p^n$, the following relation is satisfied:

$$\int_{\mathscr{E}_h^n} \overline{\nabla}_h \phi_h \cdot \overline{\nabla}_h v \, \mathrm{d}x + \int_{\Gamma_0^n} [[\hat{\phi} - \phi_h]] \cdot \langle \overline{\nabla}_h v \rangle \, \mathrm{d}s - \int_{\Gamma_0^n} [[v]] \cdot \langle \hat{u} \rangle \, \mathrm{d}s - \int_{\Gamma_0^n} \langle v \rangle \cdot [[\hat{u}]] \, \mathrm{d}s - \int_{\Gamma_{\mathscr{N}}^n} v g_N \, \mathrm{d}s + \int_{\Gamma_{\mathscr{S}}^n} \frac{v}{|\nabla_h(\zeta_h - z)|} \frac{\partial \zeta_h}{\partial t} \, \mathrm{d}s = 0. \tag{4.15}$$

Here we used (3.4) to transform the integral with the lifting operator over $\mathscr{E}_h^n$ into an integral over Γ_0^n, since $[[\hat{\phi} - \phi_h]] = 0$ at $\Gamma_{\mathscr{N}}^n \cup \Gamma_{\mathscr{S}}^n$. We also introduced the boundary condition (2.3) at the solid surfaces $\Gamma_{\mathscr{N}}^n$ and the kinematic free surface boundary condition (4.3) at $\Gamma_{\mathscr{S}}^n$. Using the definitions of the jump and average operators (3.1)–(3.3), the numerical fluxes (4.11)–(4.13), and the local lifting operator (3.5), we can transform the following integrals over Γ_0^n in (4.15) into

$$\begin{aligned} \int_{\Gamma_0^n} [[\widehat{\phi} - \phi_h]] \cdot \langle \overline{\nabla}_h v \rangle \, \mathrm{d}s &= - \int_{\Gamma_0^n} [[\phi_h]] \cdot \langle \overline{\nabla}_h v \rangle \, \mathrm{d}s \\ \int_{\Gamma_0^n} [[v]] \cdot \langle \hat{u} \rangle \, \mathrm{d}s &= \int_{\Gamma_0^n} [[v]] \cdot \langle \overline{\nabla}_h \phi_h \rangle \, \mathrm{d}s + \sum_{\mathscr{F} \in \mathscr{F}_I^n} \int_{\mathscr{E}_h^n} \mathscr{R}_{\mathscr{F}}([[v]]) \cdot \mathscr{R}([[\hat{\phi} - \phi_h]]) \, \mathrm{d}x \\ &\quad - \sum_{\mathscr{F} \in \mathscr{F}_I^n} \eta_{\mathscr{F}} \int_{\mathscr{E}_h^n} \mathscr{R}_{\mathscr{F}}([[v]]) \cdot \mathscr{R}_{\mathscr{F}}([[\phi_h]]) \, \mathrm{d}x, \end{aligned} \tag{4.16}$$

while the fourth integral in (4.15) is zero. The integrand in the second integral at the righthand of (4.16) increases the stencil of the DG discretization for each element $\mathscr{K}$ beyond its nearest neighbors. Using (3.5), the fact that $[[\hat{\phi} - \phi_h]] = 0$ at $\Gamma_{\mathscr{N}} \cup \Gamma_{\mathscr{S}}^n$, and the relation $[[\hat{\phi} - \phi_h]] = -[[\phi_h]]$ at Γ_0^n, we can approximate this contribution as

$$\sum_{\mathscr{F} \in \mathscr{F}_I^n} \int_{\mathscr{E}_h^n} \mathscr{R}_{\mathscr{F}}([[v]]) \cdot \mathscr{R}([[\hat{\phi} - \phi_h]]) \, \mathrm{d}x \cong -n_f \sum_{\mathscr{F} \in \mathscr{F}_I^n} \int_{\mathscr{E}_h^n} \mathscr{R}_{\mathscr{F}}([[v]]) \cdot \mathscr{R}_{\mathscr{F}}([[\phi_h]]) \, \mathrm{d}x$$

with n_f the number of faces of an element in the tessellation. This approximation has no effect on the accuracy and stability of the space–time DG discretization, see [25], and the stencil now only contains contributions from the element itself and its nearest neighbors.

The discretization of the dynamic boundary condition (2.4) can be straightforwardly obtained by multiplying this equation with test functions $v_2 \in W_p^n$ and integration over the free surface.

If we introduce now the bilinear form $B_h : V_p^n \times V_p^n \to \mathbb{R}$ and the linear form $L_h : V_p^n \to \mathbb{R}$, which are defined as

$$B_h(u, v) = \int_{\mathscr{E}_h^n} \overline{\nabla}_h u \cdot \overline{\nabla}_h v \, \mathrm{d}x - \int_{\Gamma_0^n} ([[u]] \cdot \langle \overline{\nabla}_h v \rangle + [[v]] \cdot \langle \overline{\nabla}_h u \rangle) \, \mathrm{d}s + \sum_{\mathscr{F} \in \mathscr{F}_I^n} (\eta_{\mathscr{F}} + n_f) \int_{\mathscr{E}_h^n} \mathscr{R}_{\mathscr{F}}([[u]]) \cdot \mathscr{R}_{\mathscr{F}}([[v]]) \, \mathrm{d}x, \tag{4.17}$$

$$L_h(v) = \int_{\Gamma^n_{\mathscr{N}}} v g_N \, \mathrm{d}s,$$

then the weak formulation describing the evolution of nonlinear water waves can be formulated as

Find a $(\phi_h, \zeta_h) \in V_p^n \times W_p^n$, such that for all $(v_1, v_2) \in V_p^n \times W_p^n$,

$$B_h(\phi_h, v_1) + \left(\frac{1}{|\nabla_h(\zeta_h - z)|} \frac{\partial \zeta_h}{\partial t}, v_1 \right)_{\Gamma^n_{\mathscr{G}}} + \left(\frac{\zeta_h^+ - \zeta_h^-}{|\nabla_h(\zeta_h^+ - z)|}, v_1^+ \right)_{\Gamma_{\mathscr{G}}(t_n^+)} = L_h(v_1), \tag{4.18}$$

$$\left(\frac{\partial \phi_h}{\partial t}, v_2 \right)_{\Gamma^n_{\mathscr{G}}} + \left(\frac{1}{2} \overline{\nabla}_h \phi_h \cdot \overline{\nabla}_h \phi_h, v_2 \right)_{\Gamma^n_{\mathscr{G}}} + (\phi_h^+ - \phi_h^-, v_2^+)_{\Gamma_{\mathscr{G}}(t_n^+)} + (\zeta_h, v_2)_{\Gamma^n_{\mathscr{G}}} = 0, \tag{4.19}$$

where $(u,v)_{\Gamma^n_{\mathscr{G}}} = \int_{\Gamma^n_{\mathscr{G}}} uv \, \mathrm{d}s$ denotes the L^2-inner product at the free surface $\Gamma^n_{\mathscr{G}}$, which is approximated as $z = \zeta_h(t,x,y)$. In addition, the following notation is used $\Gamma_{\mathscr{G}}(t_n^+) = \lim_{\epsilon \downarrow 0} \Gamma_{\mathscr{G}}(t_n + \epsilon)$, $\phi_h^{\pm} = \lim_{\epsilon \downarrow 0} \phi_h(t_n \pm \epsilon)$, $\zeta_h^{\pm} = \lim_{\epsilon \downarrow 0} \zeta_h(t_n \pm \epsilon)$, with ϕ_h^- and ζ_h^- the known potential and wave height from the space–time slab $\mathscr{E}_h^{n-1}$.

Remark 4.1. In (4.18) and (4.19) also weak coupling terms are added for the wave height and potential function, which are inner products at $\Gamma_{\mathscr{G}}(t_n^+)$. These terms are necessary because the polynomial basis functions in the space–time slab $\mathscr{E}_h^n$ are discontinuous in time and the solution would otherwise not be connected to the previous space–time slab $\mathscr{E}_h^{n-1}$. We can, however, not use standard weak coupling terms in (4.18), as we did in (4.19), but we need to introduce the scaling factor $\frac{1}{|\nabla_h(\zeta_h^+ - z)|}$ in order to balance the weak coupling term with the contribution resulting from the time derivative of the wave height. Otherwise, for small time steps or large wave heights, the solution would effectively be uncoupled from the solution in the previous time slab.

Remark 4.2. The second contribution in (4.19) is related to the fluid velocity u_h (see (4.4)) and $\overline{\nabla}_h \phi_h$ should actually be replaced by

$$u_h = \overline{\nabla}_h \phi_h - \sum_{\mathscr{F} \in \mathscr{F}_I^n} \mathscr{R}_{\mathscr{F}}([[\phi_h]]) \quad \text{a.e. on } \Gamma^n_{\mathscr{G}} \tag{4.20}$$

using (4.14) and the numerical flux for $\hat{\phi}$, given by (4.11). The use of this relation in (4.19) results in a coupling of the dynamic boundary condition with the expansion coefficients in the neighboring elements and significantly increases the complexity of the space–time discretization. When the computational mesh is, however, fine enough the contribution $[[\phi_h]]$ rapidly becomes negligible. Since our test cases did not show any beneficial effect of adding the contribution of the lifting operator $\mathscr{R}_{\mathscr{F}}([[\phi_h]])$ to (4.19) we decided to omit this contribution.

5. Algebraic system for the space–time discontinuous Galerkin discretization

The weak formulation for nonlinear water waves (4.18) and (4.19) is transformed into a system of nonlinear algebraic equations by representing the potential function ϕ_h, the wave height ζ_h and the test functions v_1, v_2 in each space–time element by a polynomial expansion. For instance, the potential function $\phi_h \in V_p^n$ on hexahedral elements using a tensor product basis can be expressed as

$$\phi_h(t,x,y,z) = \sum_{j=1}^{N_p} \phi_{\mathscr{K},j} \mathscr{N}_{\mathscr{K},j}(t,x,y,z), \quad \forall \mathscr{K} \in \mathscr{T}_h^n \tag{5.1}$$

with $N_p = (p+1)^4$ the number of polynomial coefficients in the DG discretization and p the polynomial order. The wave height $\zeta_h(x)$ is represented as

$$\zeta_h(t,x,y) = \sum_{j=1}^{N_s} \zeta_{\mathscr{S},j} \mathscr{N}_{\mathscr{S},j}(t,x,y), \quad \forall \mathscr{S} \in \mathscr{F}_{\mathscr{G}}^n \tag{5.2}$$

with $N_s = (p+1)^3$ the number of expansion coefficients for the wave height. The basis functions $\mathscr{N}_{\mathscr{S},j}$ and $\mathscr{N}_{\mathscr{K},j}$ are related as

$$\mathscr{N}_{\mathscr{S},j}(t,x,y) = \mathscr{N}_{\mathscr{K},\mathscr{I}(j)}(t,x,y,\zeta(t,x,y)), \quad \text{for } j = 1,\ldots,N_s$$

with $\mathscr{I}(j)$ the index set with the vertices of an element $\mathscr{K}$ connected to the face $\mathscr{S} \subset \Gamma_{\mathscr{S}}$. Similar expressions are used for the test functions v_1 and v_2.

The space–time elements are hypercubes, but the procedure to derive the algebraic equations is essentially independent of the type of element. The basis functions $\mathscr{N}_{\mathscr{K},j}$ in each space–time element $\mathscr{K}$ are defined as

$$\mathscr{N}_{\mathscr{K},j}(t,x,y,z) = \widehat{\mathscr{N}}_j \circ F_{\mathscr{K}}^{-1}(t,x,y,z)$$

using the isoparametric mapping:

$$F_{\mathscr{K}} : [-1,1]^4 \to \mathscr{K}; (t,x,y,z)^{\mathrm{T}} = \sum_{j=1}^{N_p} x_{\mathscr{K},j} \widehat{\mathscr{N}}_j(\xi_1,\xi_2,\xi_3,\xi_4) \tag{5.3}$$

with $x_{\mathscr{K},j} = (t_{\mathscr{K},j}, x_{\mathscr{K},j}, y_{\mathscr{K},j}, z_{\mathscr{K},j})^{\mathrm{T}} \in \mathbb{R}^4$ the coordinates of the vertices of the space–time element and $\xi = (\xi_1,\xi_2,\xi_3,\xi_4)^{\mathrm{T}} \in \widehat{\mathscr{K}}$ the local coordinates in the reference element $\widehat{\mathscr{K}} := [-1,1]^4$. The basis functions $\mathscr{N}_{\mathscr{S},j}$ in each face at the free surface $\Gamma_{\mathscr{S}}^n$ are defined as

$$\mathscr{N}_{\mathscr{S},j}(t,x,y) = \widetilde{\mathscr{N}}_j \circ F_{\mathscr{S}}^{-1}(\xi_1,\xi_2,\xi_3)$$

using the isoparametric mapping:

$$\begin{aligned} &F_{\mathscr{S}} := F_{\mathscr{K}}|_{\mathscr{S}} : [-1,1]^3 \to \mathscr{S}; \\ &(t,x,y,z)^{\mathrm{T}} = \sum_{j=1}^{N_p} x_{\mathscr{K},\mathscr{I}(j)} \widehat{\mathscr{N}}_{\mathscr{I}(j)}(\xi_1,\xi_2,\xi_3,1) \\ &\qquad = \sum_{j=1}^{N_s} (t_{\mathscr{K},\mathscr{I}(j)}, x_{\mathscr{K},\mathscr{I}(j)}, y_{\mathscr{K},\mathscr{I}(j)}, \zeta_{\mathscr{S},j})^{\mathrm{T}} \widetilde{\mathscr{N}}_j(\xi_1,\xi_2,\xi_3), \end{aligned} \tag{5.4}$$

where we assume that the local coordinate $\xi_4 = 1$ is at the free surface. The basis functions $\widehat{\mathscr{N}}_j$ and $\widetilde{\mathscr{N}}_j$ are defined as the tensorproducts of pth order polynomials in the local coordinates ξ. More details are provided in Appendix A.

In order to define the vertices $x_{\mathscr{K},j}$ of the space–time elements in the isoparametric mapping (5.3) we need $p+1$ spatial meshes at different time levels in the space–time slab $\mathscr{E}_h^n$, each having the same connectivity. The first $(p+1)^3$ vertices are the vertices $\bar{x}_{\mathscr{K},j}^n$ of $K(t_n)$, the space–time element at $t = t_n$, hence $x_{\mathscr{K},j} = (t_n, \bar{x}_{\mathscr{K},j})^{\mathrm{T}}$ for $j = 1,\ldots,(p+1)^3$. The last $(p+1)^3$ vertices are the vertices $\bar{x}_{\mathscr{K},j}^{n+1}$ of $K(t_{n+1})$, hence $x_{\mathscr{K},j}^{n+1} = (t_{n+1}, \bar{x}_{\mathscr{K},j}^{n+1})^{\mathrm{T}}$ for $j = N_p - (p+1)^3 + 1,\ldots,N_p$. When $p \geqslant 2$ then also vertices of the elements $K(t_n + (m-1)\triangle t/(p-1))$, with $2 \leqslant m \leqslant p-1$, are necessary for elements at the boundary. In the interior these vertices can be obtained using linear interpolation, whereas at the domain boundary they must lie at the actual boundary surface.

For a concise description of the finite element discretization we introduce the following vectors $\mathscr{I}^{\mathscr{K};\mathscr{F}} \in \mathbb{R}^{N_p}$, matrices: $\mathscr{A}^{\mathscr{K}}, \mathscr{B}^{\mathscr{K}}, \mathscr{C}^{LR;\mathscr{F}} \in \mathbb{R}^{N_p \times N_p}$, $\mathscr{H}^{\mathscr{K};\mathscr{F}}, \mathscr{U}^{+;\mathscr{F}}, \mathscr{V}^{-;\mathscr{F}} \in \mathbb{R}^{\mathscr{N}_s \times \mathscr{N}_p}$, $\mathscr{S}^{\mathscr{K};\mathscr{F}} \in \mathbb{R}^{N_s \times N_s}$ and tensors $\mathscr{D}^{LR;\mathscr{F}} \in \mathbb{R}^{N_p \times N_p \times 3}$, $\mathscr{Q}^{\mathscr{K};\mathscr{F}} \in \mathbb{R}^{N_s \times N_p \times N_p}$ which are defined as

$$\begin{aligned} &\mathscr{A}_{ij}^{\mathscr{K}} := \int_{\mathscr{K}} \mathscr{N}_{\mathscr{K},i} \mathscr{N}_{\mathscr{K},j} \,\mathrm{d}x, \quad \mathscr{B}_{ij}^{\mathscr{K}} := \int_{\mathscr{K}} \overline{\nabla}_h \mathscr{N}_{\mathscr{K},i} \cdot \overline{\nabla}_h \mathscr{N}_{\mathscr{K},j} \,\mathrm{d}x, \\ &\mathscr{C}_{ij}^{LR;\mathscr{F}} := \int_{\mathscr{F}} \mathscr{N}_{L,j} \bar{n}_L \cdot \overline{\nabla}_h \mathscr{N}_{R,i} \,\mathrm{d}s, \quad \mathscr{D}_{ijk}^{LR;\mathscr{F}} := \int_{\mathscr{F}} \bar{n}_{L,k} \mathscr{N}_{L,i} \mathscr{N}_{R,j} \,\mathrm{d}s, \\ &\mathscr{H}_{ij}^{\mathscr{K};\mathscr{F}} := \int_{\mathscr{F}} \mathscr{N}_{\mathscr{S},i} \frac{\partial \mathscr{N}_{\mathscr{K},j}}{\partial t} \,\mathrm{d}s, \quad \mathscr{I}_i^{\mathscr{K};\mathscr{F}} := \int_{\mathscr{F}} g_N \mathscr{N}_{\mathscr{K},i} \,\mathrm{d}s, \\ &\mathscr{U}_{ij}^{+;\mathscr{F}} := \int_{\mathscr{F}(t_n^+)} \mathscr{N}_{\mathscr{S},i}(t_n^+) \mathscr{N}_{\mathscr{K},j}(t_n^+) \,\mathrm{d}s, \quad \mathscr{V}_{ij}^{-;\mathscr{F}} := \int_{\mathscr{F}(t_n^+)} \mathscr{N}_{\mathscr{S},i}(t_n^+) \mathscr{N}_{\mathscr{K},j}(t_n^-) \,\mathrm{d}s, \end{aligned}$$

$$\mathscr{S}_{ij}^{\mathscr{K};\mathscr{F}} := \int_{\mathscr{F}} \mathscr{N}_{\mathscr{S},i}\mathscr{N}_{\mathscr{S},j}\,\mathrm{d}s, \quad \mathscr{Q}_{ijl}^{\mathscr{K};\mathscr{F}} := \frac{1}{2}\int_{\mathscr{F}} \mathscr{N}_{\mathscr{S},i}\overline{\nabla}\mathscr{N}_{\mathscr{K},j}\cdot\overline{\nabla}\mathscr{N}_{\mathscr{K},l}\,\mathrm{d}s. \tag{5.5}$$

Here, $\mathscr{K}$ is the index of the elements in the tessellation $\mathscr{T}_h^n$, which contains $N_{\mathscr{T}}$ elements, and L, R are the element indices of the two elements connected to each side of the face with index $\mathscr{F}$. The number of free surface faces is denoted as $N_{\mathscr{S}}$. In addition, we need to define $\mathscr{P}^{\mathscr{K};\mathscr{F}}$, $\mathscr{Y}^{-;\mathscr{F}} \in \mathbb{R}^{\mathscr{N}_p\times\mathscr{N}_s}$ and $\mathscr{X}^{-;\mathscr{F}}$, $\mathscr{R}^{\mathscr{K};\mathscr{F}} \in \mathbb{R}^{N_p\times N_s\times N_s\times N_s}$, which are related to the free surface contribution:

$$\begin{aligned}
\mathscr{P}_{ij}^{\mathscr{K};\mathscr{F}}(\zeta_h) &:= \int_{\mathscr{F}} \frac{1}{|\nabla_h(\zeta_h - z)|}\mathscr{N}_{\mathscr{K},i}\frac{\partial\mathscr{N}_{\mathscr{S},j}}{\partial t}\,\mathrm{d}s + \int_{\mathscr{F}(t_n^+)} \frac{1}{|\nabla_h(\zeta_h^+ - z)|}\mathscr{N}_{\mathscr{K},i}(t_n^+)\mathscr{N}_{\mathscr{S},j}(t_n^+)\,\mathrm{d}s,\\
\mathscr{Y}_{ij}^{-;\mathscr{F}}(\zeta_h) &:= \int_{\mathscr{F}(t_n^+)} \frac{1}{|\nabla_h(\zeta_h^+ - z)|}\mathscr{N}_{\mathscr{K},i}(t_n^+)\mathscr{N}_{\mathscr{S},j}(t_n^-)\,\mathrm{d}s,\\
\mathscr{R}_{ijkl}^{\mathscr{K};\mathscr{F}}(\zeta_h) &:= \int_{\mathscr{F}} \frac{1}{|\nabla_h(\zeta_h - z)|^3}\mathscr{N}_{\mathscr{K},i}\frac{\partial\mathscr{N}_{\mathscr{S},j}}{\partial t}(\nabla_S\mathscr{N}_{\mathscr{S},k}\cdot\nabla_S\mathscr{N}_{\mathscr{S},l})\,\mathrm{d}s\\
&\quad + \int_{\mathscr{F}(t_n^+)} \frac{1}{|\nabla_h(\zeta_h^+ - z)|^3}\mathscr{N}_{\mathscr{K},i}(t_n^+)\mathscr{N}_{\mathscr{S},j}(t_n^+)(\nabla_S\mathscr{N}_{\mathscr{S},k}(t_n^+)\cdot\nabla_S\mathscr{N}_{\mathscr{S},l}(t_n^+))\,\mathrm{d}s,\\
\mathscr{X}_{ijkl}^{-;\mathscr{F}}(\zeta_h) &:= \int_{\mathscr{F}(t_n^+)} \frac{1}{|\nabla_h(\zeta_h^+ - z)|^3}\mathscr{N}_{\mathscr{K},i}(t_n^+)\mathscr{N}_{\mathscr{S},j}(t_n^-)(\nabla_S\mathscr{N}_{\mathscr{S},k}(t_n^+)\cdot\nabla_S\mathscr{N}_{\mathscr{S},l}(t_n^+))\,\mathrm{d}s
\end{aligned} \tag{5.6}$$

with $\nabla_S = (\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y})^{\mathrm{T}}$. The integrals are computed by transforming them to the reference element $\widehat{\mathscr{K}}$ or the reference face $\widehat{\mathscr{F}} := [-1,1]^3$, respectively, using the mappings (5.3) and (5.4). More details are provided in Appendix B.

The algebraic equations for the expansion coefficients $\phi_{\mathscr{K},j}$ and $\zeta_{\mathscr{S},j}$ of the potential and wave height in the space–time slab $\mathscr{E}_h^n$, respectively, are obtained by introducing the polynomial representations for ϕ_h and ζ_h and the test functions v_1, v_2, defined in (5.1) and (5.2), into (4.18) and (4.19) and using the fact that these equations must be satisfied for arbitrary test functions.

The algebraic equations for the discontinuous Galerkin discretization of (4.18) in the space–time slab $\mathscr{E}_h^n$ can be expressed symbolically as

$$L_{11}\Phi^n + Q(\mathscr{Z}^n) - X(\mathscr{Z}^n;\mathscr{Z}^{n-1}) = L_h \tag{5.7}$$

with $\Phi^n \in \mathbb{R}^{N_pN_{\mathscr{T}}}$ and $\mathscr{Z}^n \in \mathbb{R}^{N_sN_{\mathscr{S}}}$ the expansion coefficients for the potential and wave height, respectively. The construction of the matrix $L_{11} \in \mathbb{R}^{N_pN_{\mathscr{T}}\times N_pN_{\mathscr{T}}}$ is discussed in Appendix C. The nonlinear free surface contribution Q, $X \in \mathbb{R}^{N_pN_{\mathscr{T}}}$ are defined as

$$\begin{aligned}
Q_i(\mathscr{Z}) = \sum_{j=1}^{N_s}\zeta_{\mathscr{S},j}\Bigg(&\int_{\mathscr{F}} \frac{\mathscr{N}_{\mathscr{K},i}\frac{\partial\mathscr{N}_{\mathscr{S},j}}{\partial t}\,\mathrm{d}s}{\sqrt{\sum_{p=1}^{N_s}\sum_{q=1}^{N_s}\zeta_{\mathscr{S},p}\zeta_{\mathscr{S},q}\nabla_S\mathscr{N}_{\mathscr{S},p}\cdot\nabla_S\mathscr{N}_{\mathscr{S},q}+1}}\\
&+\int_{\mathscr{F}(t_n^+)} \frac{\mathscr{N}_{\mathscr{K},i}(t_n^+)\mathscr{N}_{\mathscr{S},j}(t_n^+)\,\mathrm{d}s}{\sqrt{\sum_{p=1}^{N_s}\sum_{q=1}^{N_s}\zeta_{\mathscr{S},p}\zeta_{\mathscr{S},q}\nabla_S\mathscr{N}_{\mathscr{S},p}(t_n^+)\cdot\nabla_S\mathscr{N}_{\mathscr{S},q}(t_n^+)+1}}\Bigg)
\end{aligned}$$

and

$$X_i(\mathscr{Z};\mathscr{Z}^{n-1}) = \sum_{j=1}^{N_s}\zeta_{\mathscr{S},j}^{n-1}\int_{\mathscr{F}(t_n^+)} \frac{\mathscr{N}_{\mathscr{K},i}(t_n^+)\mathscr{N}_{\mathscr{S},j}(t_n^-)\,\mathrm{d}s}{\sqrt{\sum_{p=1}^{N_s}\sum_{q=1}^{N_s}\zeta_{\mathscr{S},p}\zeta_{\mathscr{S},q}\nabla_S\mathscr{N}_{\mathscr{S},p}(t_n^+)\cdot\nabla_S\mathscr{N}_{\mathscr{S},q}(t_n^+)+1}}.$$

The nonlinear algebraic system (5.7) is solved with a Newton method. For this purpose we linearize Q and X around the mth Newton iterate $\mathscr{Z}^{(m)}$:

$$Q_i(\mathscr{Z}) \cong Q_i(\mathscr{Z}^{(m)}) + \sum_{l=1}^{N_s} \frac{\partial Q_i}{\partial \mathscr{Z}_l}(\mathscr{Z}_l - \mathscr{Z}_l^{(m)}),$$

$$X_i(\mathscr{Z};\mathscr{Z}^{n-1}) \cong X_i(\mathscr{Z}^{(m)};\mathscr{Z}^{n-1}) + \sum_{l=1}^{N_s} \frac{\partial X_i}{\partial \mathscr{Z}_l}(\mathscr{Z}_l - \mathscr{Z}_l^{(m)})$$

with

$$\frac{\partial Q_i}{\partial \mathscr{Z}_l} = \mathscr{P}_{il}^{\mathscr{K};\mathscr{F}}(\mathscr{Z}^{(m)}) - \sum_{j=1}^{N_s}\sum_{k=1}^{N_s} \mathscr{Z}_j^{(m)} \mathscr{Z}_k^{(m)} \mathscr{R}_{ijkl}^{\mathscr{K};\mathscr{F}}(\mathscr{Z}^{(m)}),$$

$$\frac{\partial X_i}{\partial \mathscr{Z}_l} = -\sum_{j=1}^{N_s}\sum_{k=1}^{N_s} \mathscr{Z}_j^{n-1} \mathscr{Z}_k^{(m)} \mathscr{X}_{ijkl}^{-;\mathscr{F}}(\mathscr{Z}^{(m)}).$$

If we introduce the matrix

$$\Theta(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1}) := \frac{\partial Q(\mathscr{Z}^{n;(m)})}{\partial \mathscr{Z}} - \frac{\partial X(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1})}{\partial \mathscr{Z}} \in \mathbb{R}^{N_p N_{\mathscr{T}} \times N_s N_{\mathscr{S}}},$$

then we can write the linearized form of (5.7) as

$$L_{11}\Phi^n + \Theta(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1})\mathscr{Z}^{n;(m+1)} = \Theta(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1})\mathscr{Z}^{n;(m)} - Q(\mathscr{Z}^{n;(m)}) + X(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1}) + L_h. \tag{5.8}$$

Using the element integrals defined in (5.5), the algebraic equations for the dynamic condition at the free surface (4.19) can be expressed in each element $\mathscr{K}$ connected to the free surface $\Gamma_{\mathscr{S}}$ as

$$\sum_{j=1}^{N_p} \phi_{\mathscr{K},j}^n \mathscr{H}_{ij}^{\mathscr{K};\mathscr{F}} + \sum_{j=1}^{N_p} \phi_{\mathscr{K},j}^n \mathscr{U}_{ij}^{+;\mathscr{F}} + \sum_{j=1}^{N_p}\sum_{l=1}^{N_p} \phi_{\mathscr{K},j}^n \phi_{\mathscr{K},l}^n \mathscr{Q}_{ijl}^{\mathscr{K};\mathscr{F}} + \sum_{j=1}^{N_s} \zeta_{\mathscr{S},j}^n \mathscr{S}_{ij}^{\mathscr{K};\mathscr{F}} = \sum_{j=1}^{N_p} \phi_{\mathscr{K},j}^{n-1} \mathscr{V}_{ij}^{-;\mathscr{F}}.$$

These nonlinear equations are also solved with a Newton method. Assume $\phi_{\mathscr{K},j}^{(m)}$ is the mth iterate for the expansion coefficients $\phi_{\mathscr{K},j}$ in element $\mathscr{K}$ in the Newton method, then the next approximation is found by solving the linear system:

$$\sum_{j=1}^{N_p}\left(\mathscr{H}_{ij}^{\mathscr{K};\mathscr{F}} + \mathscr{U}_{ij}^{+;\mathscr{F}} + 2\sum_{l=1}^{N_p} \phi_{\mathscr{K},l}^{n;(m)} \mathscr{Q}_{ijl}^{\mathscr{K};\mathscr{F}}\right)\phi_{\mathscr{K},j}^{n;(m+1)} + \sum_{j=1}^{N_s} \zeta_{\mathscr{S},j}^n \mathscr{S}_{ij}^{\mathscr{K};\mathscr{F}}$$
$$= \sum_{j=1}^{N_p}\sum_{l=1}^{N_p} \phi_{\mathscr{K},j}^{n;(m)} \phi_{\mathscr{K},l}^{n;(m)} \mathscr{Q}_{ijl}^{\mathscr{K};\mathscr{F}} + \sum_{j=1}^{N_p} \phi_{\mathscr{K},j}^{n-1} \mathscr{V}_{ij}^{-;\mathscr{F}} \quad \forall \mathscr{F} \in \mathscr{F}_{\mathscr{S}}. \tag{5.9}$$

The coupled algebraic systems in the Newton procedure (5.8) and (5.9) can be written in matrix form as

$$\mathscr{M}\Psi^{n;(m+1)} = F(\Psi^{n;(m)};\Psi^{n-1}) \tag{5.10}$$

with $\mathscr{M} \in \mathbb{R}^{m_{\mathscr{T}} \times m_{\mathscr{T}}}$, $\Psi^{n;(m+1)}$, $\Psi^{n-1} \in \mathbb{R}^{m_{\mathscr{T}}}$, $F \in \mathbb{R}^{m_{\mathscr{T}}}$ and $m_{\mathscr{T}} = N_p N_{\mathscr{T}} + N_s N_{\mathscr{S}}$. The linear system (5.10) can be represented as

$$\begin{pmatrix} L_{11} & L_{12}(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1}) \\ L_{21}(\Phi^{n;(m)}) & L_{22} \end{pmatrix}\begin{pmatrix} \Phi^{n;(m+1)} \\ \mathscr{Z}^{n;(m+1)} \end{pmatrix} = \begin{pmatrix} F_1(\mathscr{Z}^{n;(m)};\mathscr{Z}^{n-1}) \\ F_2(\Phi^{n;(m)};\Phi^{n-1}) \end{pmatrix}$$

with $\Phi \in \mathbb{R}^{N_p N_{\mathscr{T}}}$ and $\mathscr{Z} \in \mathbb{R}^{N_s N_{\mathscr{S}}}$ the expansion coefficients for the potential and wave height, respectively. The matrices L_{ij} have dimension $L_{11} \in \mathbb{R}^{N_p N_{\mathscr{T}} \times N_p N_{\mathscr{T}}}, L_{12} \in \mathbb{R}^{N_p N_{\mathscr{T}} \times N_s N_{\mathscr{S}}}, L_{21} \in \mathbb{R}^{N_s N_{\mathscr{S}} \times N_p N_{\mathscr{T}}}, L_{22} \in \mathbb{R}^{N_s N_{\mathscr{S}} \times N_s N_{\mathscr{S}}}$. Note, L_{12} only depends on the expansion coefficients for the wave height, while L_{21} only depends on the expansion coefficients for the potential. As long as the space–time mesh is not deformed the other contributions in the matrix $\mathscr{M}$ therefore do not have to be updated. The details of the matrix construction for a general unstructured mesh, which requires some care in the DG algorithm, are given in Appendix C.

The computation of the wave field in a space–time slab now proceeds as follows. Given the mesh at $t = t_n$ we construct a space–time mesh by translating the mesh at $t = t_n$ to the new time level t_{n+1}. On this space–time

mesh we perform a number of Newton iterations, which consist of solving (5.10), updating L_{12} and F_1 with the new coefficients $\mathscr{Z}^{n;(m+1)}$ and L_{21} and F_2 with the new coefficients $\Phi^{n;(m+1)}$, until a specified tolerance in $|\Phi^{n;(m+1)} - \Phi^{n;(m)}|$ and $|\mathscr{Z}^{n;(m+1)} - \mathscr{Z}^{n;(m)}|$ is obtained. Next, the free surface is updated using the computed wave height coefficients $\mathscr{Z}^{n;(m+1)}$ and the interior mesh are moved to maintain a consistent mesh without grid folding and with a proper distribution of the mesh resolution. Next, a new potential and wave height is computed on the updated mesh. This process is continued until the difference in wave height and potential on two consecutive meshes is within a prescribed tolerance.

The mesh is deformed by the area-orthogonality (AO) grid generator in [14], which we summarize for two-dimensional problems. The AO method is based on the solution of the following pair of partial differential equations for the mapping functions x and y:

$$g_{22}x_{\xi\xi} + 4x_\xi x_\eta x_{\xi\eta} + g_{11}x_{\eta\eta} + 2(x_\xi y_\eta + x_\eta y_\xi)y_{\xi\eta} = 0,$$
$$g_{22}y_{\xi\xi} + 4y_\xi y_\eta y_{\xi\eta} + g_{11}y_{\eta\eta} + 2(x_\xi y_\eta + x_\eta y_\xi)x_{\xi\eta} = 0$$

with

$$g_{11} = x_\xi^2 + y_\xi^2,$$
$$g_{22} = x_\eta^2 + y_\eta^2.$$

These equations are generally solved as a Dirichlet problem with boundary values obtained from the physical boundaries of the free surface and, if present, the prescribed motion of a wave maker. The AO method permits direct control over geometric grid qualities, such as smoothness, area and "orthogonality". The mesh generation equations are discretized by a finite difference method and solved with the successive overrelaxation method (SOR).

The linear system (5.10) is nonsingular due to the stabilization terms in the DG discretization, but it has a large condition number because one of the eigenvalues is close to zero. This does not result in a serious loss of accuracy when the zero mean condition $\int_\Omega \phi_h \,\mathrm{d}x = 0$ is imposed after the linear system has been solved. This ensures that a unique solution is obtained for the equation of the potential (2.2)–(2.6). The zero mean condition is imposed by correcting the numerical solution after each Newton iteration step as $\phi_h \leftarrow \phi_h - \bar{\phi}_h$, with $\bar{\phi}_h = \frac{1}{|\Omega|}\int_\Omega \phi_h \,\mathrm{d}x$, after which the new expansion coefficients of ϕ_h are obtained in each element $\mathscr{K}$ using an L_2-projection onto the basis functions $\mathscr{N}_{\mathscr{K}j}$.

6. Numerical examples

In this section we provide numerical examples to illustrate the accuracy and capabilities of the space–time DG method for linear and nonlinear water waves. In all examples, the figures present the solution obtained with a particular choice of the mesh. We have verified with the aid of successive mesh refinements, that in all cases, the results shown are numerically convergent.

Example 6.1. Accuracy test for linear free surface waves.

The nonlinear space–time DG (STDG) method also works for linear free surface waves. We compare the accuracy of the STDG algorithm therefore also with the DG algorithm for linear waves discussed in [27]. Linear free surface waves satisfy the equations

$$\begin{aligned} &-\Delta\phi = 0, \quad \text{in } \Omega = [0,2]\times[-1,0],\\ &n\cdot\overline{\nabla}\phi = 0, \quad \text{at } z = -1 \end{aligned} \tag{6.1}$$

with linear free-surface boundary conditions at Γ_S

$$\frac{\partial\phi}{\partial t} + \zeta = 0, \tag{6.2}$$
$$\frac{\partial\zeta}{\partial t} - \frac{\partial\phi}{\partial z} = 0 \tag{6.3}$$

and periodic boundary conditions at $x = 0$ and $x = 2$. The analytic solution of this problem is given by

$$\phi = \phi_0 \cosh(k(z+1)) \cos(\omega t - kx),$$

where ϕ_0 denotes the amplitude of the velocity potential, k the wave number and ω the frequency of the oscillations, which satisfies the dispersion relation

$$\omega^2 = k \tanh(k).$$

The initial conditions for the free surface potential and wave height at $z = 0$ are

$$\phi = \phi_0 \cosh(k) \cos(kx),$$
$$\zeta = \phi_0 \cosh(k) \sin(kx).$$

In Tables 6.1 and 6.2 we present the error in the wave height and potential at time $t = 2.5$ with $\phi_0 = 0.1$ in the L^2 and L^∞-norm for the STDG method and the DG method discussed in [27]. For the STDG method, the basis functions have polynomial degrees $p = 1$ and $p = 2$, both in space and time. Also, the order of accuracy for the different polynomial degrees is given, which indicates that the STDG method has an order of accuracy $O(h^{p+1} + \triangle t^{p+1})$ and shows superconvergence in the L^2-norm. For the DG method in [27], the basis functions also have polynomial degrees $p = 1$ and $p = 2$ in space, but the time discretization is only second order accurate. That is the reason why we obtain second order accuracy for $p = 2$ polynomials for this DG method. We can also compare the error levels, which are significantly smaller for the STDG method than the DG method.

Example 6.2. Linear waves generated by a wave maker.

Next, we consider linear waves generated by a wave maker and compare them with the results obtained in [27]. The domain is $\Omega = [0, 10] \times [-1, 0]$. The time harmonic normal velocity profile at the wave maker, which is positioned at $x = 0$, is linear in the vertical direction, starting from zero at the bottom and with an amplitude of 0.05 at the free surface. The frequency of the time-harmonic motion is $\omega_w = 2$. Homogeneous Neumann boundary conditions representing no flow through the boundary surface are assumed at the bottom $z = -1$ and at the end of the domain at $x = 10$. The initial free-surface height and velocity potential are zero. All simulations are done on a mesh of 80×20 elements, where the mesh points have a random displacement with a maximum of 30% of the edge length.

The wave profiles in the domain at $T = 20$ and $T = 50$ are presented in Figs. 6.1 and 6.2. The results are compared with the linear DG algorithm discussed in [27] and show an excellent agreement at $T = 20$ both for polynomial order $p = 1$ and 2.

At $T = 50$, the STDG solution is more accurate for $p = 1$ than the linear DG method. This is also visible in the results in Tables 6.1 and 6.2. The results for $p = 2$ for both methods are still very close, but the STDG method will be more accurate for longer simulation times since this scheme is third order accurate in time, whereas the linear DG method is only second order accurate in time.

Table 6.1
Accuracy and order of convergence of the wave height computed with the STDG and the linear DG method [27] for linear free surface waves

	$nx \times ny$	STDG				DG			
		L^2 error	Order	L^∞ error	Order	L^2 error	Order	L^∞ error	Order
p^1	6×6	3.62e−03	–	3.09e−02	–	6.56e−02	–	7.74e−02	–
	12×12	6.43e−04	2.49	7.74e−03	2.00	1.74e−02	1.92	1.76e−02	2.14
	24×24	1.14e−04	2.49	1.99e−03	1.96	4.42e−03	1.97	4.70e−03	1.90
	48×48	2.01e−05	2.51	4.86e−04	2.03	1.09e−03	2.02	1.16e−03	2.02
	6×6	3.38e−04	–	4.26e−03	–	8.03e−03	–	8.83e−03	–
p^2	12×12	4.28e−05	2.98	8.05e−04	2.40	1.44e−03	2.48	2.36e−03	1.90
	24×24	7.43e−06	2.53	1.99e−04	2.01	4.11e−04	1.80	6.74e−04	1.81
	48×48	6.78e−07	3.45	2.53e−05	2.98	1.07e−04	1.94	1.47e−04	2.20

Table 6.2
Accuracy and order of convergence of the potential computed with the STDG and the linear DG method for linear free surface waves

	$nx \times ny$	STDG				DG			
		L^2 error	Order	L^∞ error	Order	L^2 error	Order	L^∞ error	Order
p^1	6×6	1.48e−03	–	1.15e−02	–	2.35e−02	–	2.37e−02	–
	12×12	2.64e−04	2.49	3.07e−03	1.90	6.76e−03	1.80	6.13e−03	1.95
	24×24	4.68e−05	2.50	7.77e−04	1.98	1.72e−03	1.97	1.77e−03	1.79
	48×48	8.29e−06	2.50	1.95e−04	2.00	4.32e−04	2.00	4.52e−04	1.97
	6×6	5.55e−05	–	6.17e−04	–	3.27e−03	–	3.42e−03	–
p^2	12×12	5.29e−06	3.39	7.71e−05	3.00	5.47e−04	2.58	7.05e−04	2.28
	24×24	4.20e−07	3.65	7.91e−06	3.29	1.37e−04	1.99	1.68e−04	2.07
	48×48	3.42e−08	3.62	8.43e−07	3.23	3.40e−05	2.01	3.71e−05	2.17

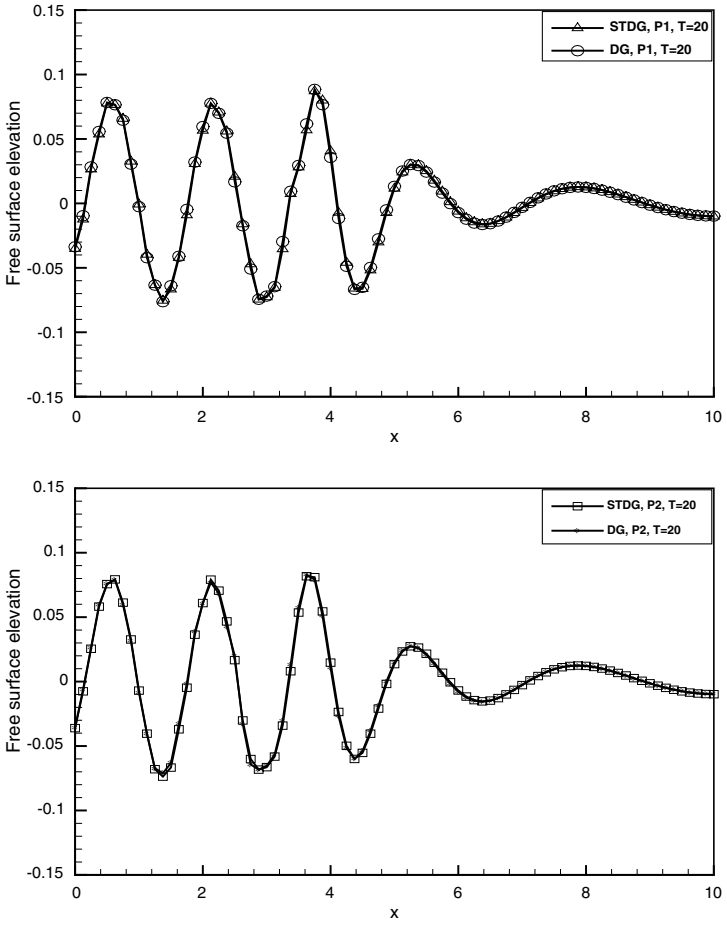

Fig. 6.1. Wave profile at $T = 20$ generated by a wave maker at $x = 0$ (randomly disturbed mesh 80×20 points, wave maker amplitude 0.05) for polynomial basis functions of degree $p = 1$ and 2 using the STDG method and the linear DG method in [27].

Example 6.3. Evolution of a slowly modulated wave packet.

In this example we consider nonlinear water waves governed by (2.2)–(2.7) with periodic boundary conditions in the x-direction. We consider the same case as in [11] to compute the evolution of a slowly modulated wave packet

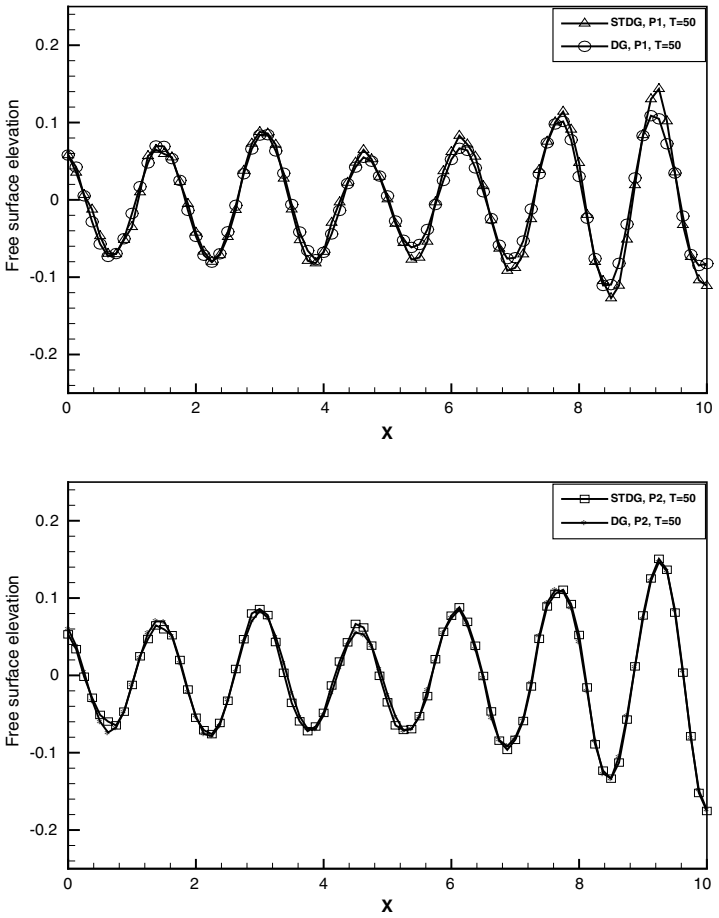

Fig. 6.2. Wave profile at $T = 50$ generated by a wave maker at $x = 0$ (randomly disturbed mesh 80×20 points, wave maker amplitude 0.05) for polynomial basis functions of degree $p = 1$ and 2 using the STDG method and the linear DG method in [27].

$$\zeta_0(x) = 0.01 \exp\left(-\frac{3}{4}(x-\pi)^2\right)\cos(10x) \tag{6.4}$$

with zero initial velocity potential in the domain $\Omega = [0, 2\pi] \times [-1, 0]$.

In Fig. 6.3, the wave profiles at $T = 5$ and $T = 10$, computed with the STDG method on different meshes for polynomial orders $p = 1$ and $p = 2$, respectively, and initial condition (6.4) are presented. The results for $p = 2$ on the coarser mesh compare well with the $p = 1$ results on the fine mesh. The numerical results converge for different meshes and polynomial orders. The long time evolution of the wave profiles can be compared with Fig. 1a in [11] and show very similar wave interaction phenomena.

Example 6.4. Comparison with exact nonlinear wave solution.

In this example we consider the time domain problem discussed in [31] in a rectangular container with depth $d = 1$ and width $b = 2$. The nonlinear free surface waves satisfy (2.2)–(2.7) with reflecting boundary conditions. The initial condition is

$$\zeta_0(x) = \alpha\cos(2\pi x/b), \tag{6.5}$$

where α is the amplitude. In [31], numerical results and a second order analytic solution for this problem are presented.

The space–time computations are conducted for the polynomial order $p = 1$ on a mesh with 40×10 elements, and for $p = 2$ on a mesh with 20×10 elements. In Fig. 6.4 the wave elevation ζ/α at the midpoint in

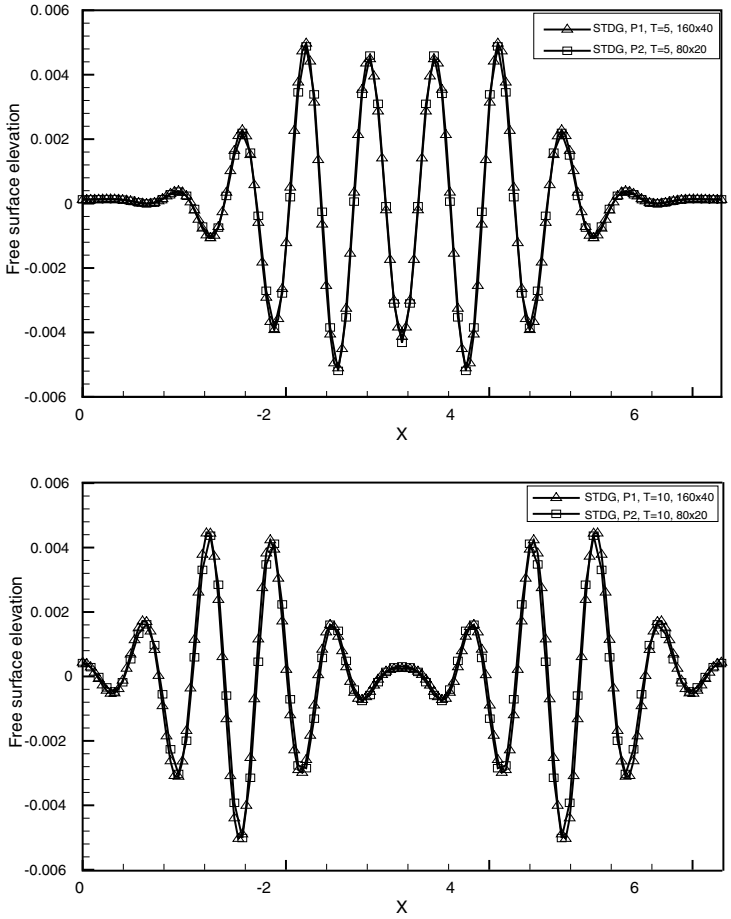

Fig. 6.3. Wave profile at $T = 5$ and $T = 10$, respectively, for nonlinear waves with initial condition (6.4).

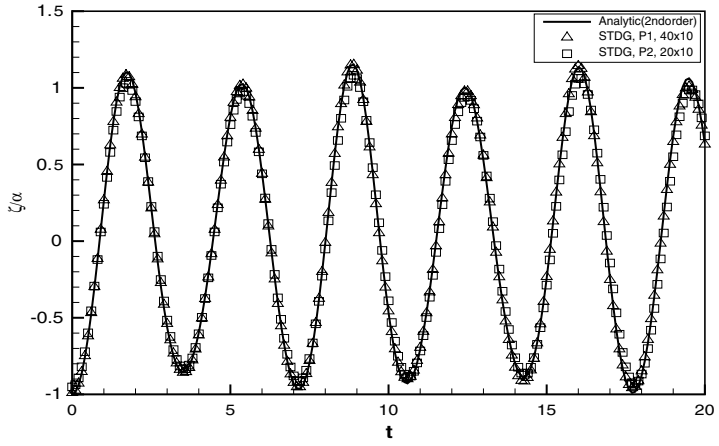

Fig. 6.4. Time history of particle in nonlinear wave with initial condition (6.5) with wave amplitude $\alpha = 0.05$.

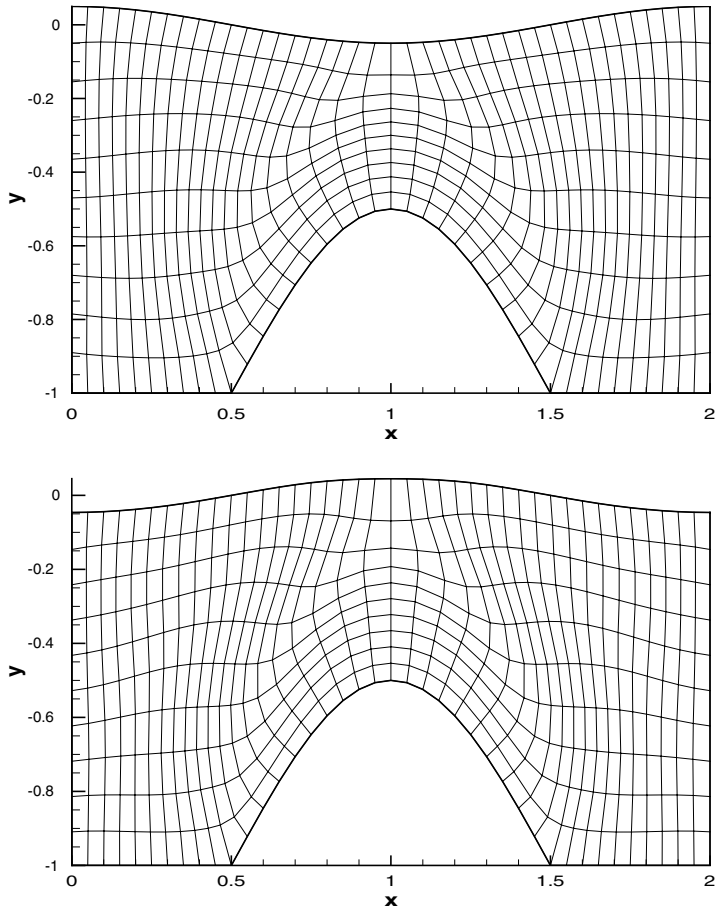

Fig. 6.5. Grid at $T = 0$ (top) and $T = 20$ (below) for nonlinear wave in a basin with a bump for initial condition (6.5) with wave amplitude $\alpha = 0.05$.

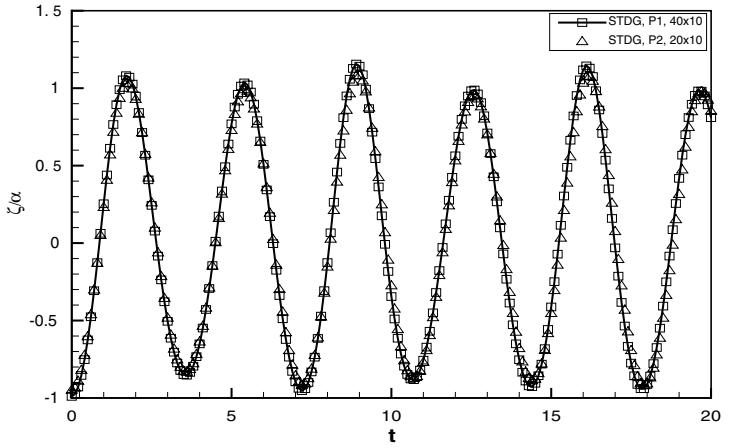

Fig. 6.6. Time history of particle in nonlinear wave in a basin with a bump for initial condition (6.5) with wave amplitude $\alpha = 0.05$.

the x-direction ($x = b/2$) is given. The STDG results and the second order analytic solution compare well for the different polynomial orders.

Example 6.5. Nonlinear free-surface waves in a basin with a bump.

In this example we consider the same problem which is discussed in Example 6.4, but now in a basin with a bump. The aim of these computations is to investigate the effect of large mesh deformation on the accuracy. In Fig. 6.6 the wave elevation ζ/α at the midpoint in the x-direction ($x = b/2$) is given. We also show the mesh deformation at time $T = 0$ and $T = 20$ in Fig. 6.5. Even with the relative coarse meshes, the particle history compare pretty well with the flat bottom case in Example 6.4

Example 6.6. Nonlinear free-surface waves generated by a wave maker.

Next, we consider nonlinear free-surface waves generated by a wave maker. The domain is $\Omega = [0,4] \times [-1,0]$. The frequency of the time-harmonic motion is $\omega_w = 2$. Homogeneous Neumann boundary conditions are assumed at the bottom $z = -1$ and at the end of the domain at $x = 4$. The initial free-surface height and velocity potential are zero. At each time step the mesh is deformed to account for the free surface motion.

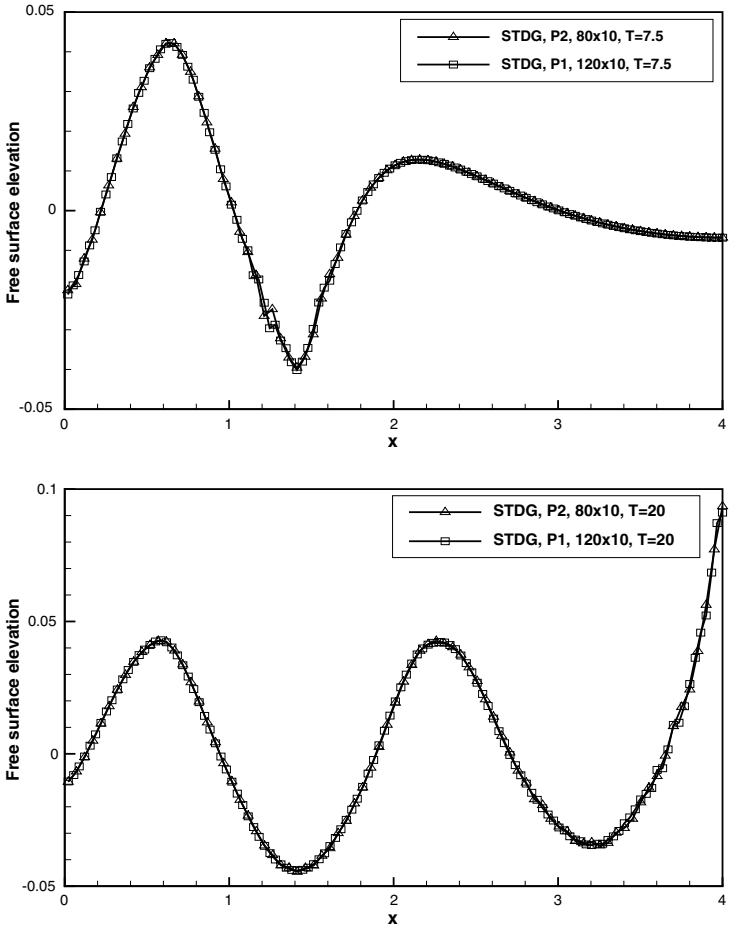

Fig. 6.7. Wave profile at $T = 7.5$ (top) and $T = 20$ (below) for nonlinear waves generated by a wave maker at $x = 0$ for polynomial basis functions of degree $p = 1$ and 2 using the STDG method.

The wave profiles in the domain at $T = 7.5$ and $T = 20$ are presented in Fig. 6.7 and show the typical increase in wave height when the wave is reflected from the wall opposite to the wave maker. The wave profiles compare well for different meshes and different polynomial order.

7. Concluding remarks

In this article we have developed a space–time discontinuous Galerkin method for nonlinear water waves. The key features of the method are that the DG discretization is done in four dimensional space, with time as the fourth dimension, and the use of test and trial functions which are discontinuous both in space and time. This technique results in a higher order accurate conservative numerical scheme on time dependent deforming meshes which are necessary to follow the free surface evolution. A novel ingredient in the finite element formulation is the incorporation of the kinematic boundary condition directly into the weak formulation. The space–time discretization then will automatically account for the mesh movement induced by the wave motion. An efficient solution technique for the algebraic equations resulting from the space–time discretization is obtained using a Newton method. This results in a rapidly converging iterative process which can be combined with efficient (iterative) sparse matrix techniques to solve the resulting linear system. The mesh deformation is controlled by solving elliptic partial differential equations for the position of the mesh points. This technique results in a high quality mesh which follows the evolution of the free surface without locally resulting in meshes which are too coarse or fine. Numerical examples include linear and nonlinear free-surface wave simulations and illustrate the accuracy and capabilities of the method by comparing the results to known solutions and verifying dependence of the error on the mesh size. The application to three dimensional problems is in progress and will aim at the construction of a numerical wave tank which can simulate large amplitude waves in a model basin. The main purpose of these activities is to provide accurate information about choosing the proper set up of model experiments and the optimal control of wave makers.

Acknowledgments

The first author would like to thank Dr. S.K. Tomar from RICAM, Austrian Academy of Sciences, Linz, Austria for the collaboration on earlier versions of the DG algorithm for nonlinear water waves. Also, the discussions with Dr. O. Bokhove on various aspects of wave problems are greatly appreciated.

Appendix A. Basis functions

The following functions are used to construct the tensor product basis functions on hexahedra and hypercubes:

$$
\begin{aligned}
p = 1: \qquad & \widehat{\Xi}_1(\xi) = \frac{1}{2}(1-\xi) \\
& \widehat{\Xi}_2(\xi) = \frac{1}{2}(1+\xi) \\
p = 2: \qquad & \widehat{\Xi}_1(\xi) = -\frac{1}{2}\xi(1-\xi) \\
& \widehat{\Xi}_2(\xi) = (1-\xi)(1+\xi) \\
& \widehat{\Xi}_3(\xi) = \frac{1}{2}\xi(1+\xi).
\end{aligned}
$$

The tensor product basis functions $\widetilde{\mathcal{N}}_j$ on the reference element $[-1,1]^3$ are then defined as

$$\widetilde{\mathcal{N}}_j(\xi_1,\xi_2,\xi_3) = \widehat{\Xi}_{i_1}(\xi_1)\widehat{\Xi}_{i_2}(\xi_2)\widehat{\Xi}_{i_3}(\xi_3)$$

with $j = i_1 + (i_2-1)(p+1) + (i_3-1)(p+1)^2$ for $i_1, i_2, i_3 \in \{1, 2, \ldots, p+1\}$.

The tensor product basis functions $\widehat{\mathscr{N}}_j$ on the reference element $[-1,1]^4$ are defined as

$$\widehat{\mathscr{N}}_j(\xi_1,\xi_2,\xi_3,\xi_4) = \widehat{\Xi}_{i_1}(\xi_1)\widehat{\Xi}_{i_2}(\xi_2)\widehat{\Xi}_{i_3}(\xi_3)\widehat{\Xi}_{i_4}(\xi_4)$$

with $j = i_1 + (i_2-1)(p+1) + (i_3-1)(p+1)^2 + (i_4-1)(p+1)^3$ for $i_1,i_2,i_3,i_4 \in \{1,2,\ldots,p+1\}$.

Appendix B. Element and face integrals

Given the mapping $F_{\mathscr{K}}$, defined in (5.3) we can transform the integrals in the DG formulation into integrals over the reference element $\widehat{\mathscr{K}} = [-1,1]^4$ or the reference face $\widehat{\mathscr{F}} = [-1,1]^3$. These integrals can then be straightforwardly evaluated with a Gaussian quadrature rule. For this purpose we define the Jacobian matrix $J \in \mathbb{R}^{4\times 4}$ as

$$J = \frac{\partial F_{\mathscr{K}}}{\partial(\xi_1,\xi_2,\xi_3,\xi_4)} = \begin{pmatrix} \frac{\partial t}{\partial \xi_1} & \frac{\partial t}{\partial \xi_2} & \frac{\partial t}{\partial \xi_3} & \frac{\partial t}{\partial \xi_4} \\ \frac{\partial x}{\partial \xi_1} & \frac{\partial x}{\partial \xi_2} & \frac{\partial x}{\partial \xi_3} & \frac{\partial x}{\partial \xi_4} \\ \frac{\partial y}{\partial \xi_1} & \frac{\partial y}{\partial \xi_2} & \frac{\partial y}{\partial \xi_3} & \frac{\partial y}{\partial \xi_4} \\ \frac{\partial z}{\partial \xi_1} & \frac{\partial z}{\partial \xi_2} & \frac{\partial z}{\partial \xi_3} & \frac{\partial z}{\partial \xi_4} \end{pmatrix} \tag{B.1}$$

with $(\xi_1,\xi_2,\xi_3,\xi_4)^{\mathrm{T}} \in [-1,1]^4$. The inverse of J is equal to

$$J^{-1} = \begin{pmatrix} \frac{\partial \xi_1}{\partial t} & \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} & \frac{\partial \xi_1}{\partial z} \\ \frac{\partial \xi_2}{\partial t} & \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} & \frac{\partial \xi_2}{\partial z} \\ \frac{\partial \xi_3}{\partial t} & \frac{\partial \xi_3}{\partial x} & \frac{\partial \xi_3}{\partial y} & \frac{\partial \xi_3}{\partial z} \\ \frac{\partial \xi_4}{\partial t} & \frac{\partial \xi_4}{\partial x} & \frac{\partial \xi_4}{\partial y} & \frac{\partial \xi_4}{\partial z} \end{pmatrix}. \tag{B.2}$$

We denote the first column of J^{-1} as $J_t^{-1} \in \mathbb{R}^{4\times 1}$ and the last three columns of J^{-1} as $J_{\bar{x}}^{-1} \in \mathbb{R}^{4\times 3}$ and use $J^{-\mathrm{T}}$ for the transposed of the inverse of J^{-1} with similar relations for the other Jacobian matrices. The nabla operator on the reference element $\widehat{\mathscr{K}}$ is defined as $\widehat{\nabla} = \left(\frac{\partial}{\partial\xi_1},\frac{\partial}{\partial\xi_2},\frac{\partial}{\partial\xi_3},\frac{\partial}{\partial\xi_4}\right)^{\mathrm{T}}$ and on the reference face $\widehat{\mathscr{F}}$ as $\widetilde{\nabla} = \left(\frac{\partial}{\partial\xi_1},\frac{\partial}{\partial\xi_2},\frac{\partial}{\partial\xi_3}\right)^{\mathrm{T}}$.

The Jacobian matrix $\widetilde{J} \in \mathbb{R}^{3\times 3}$ is defined as

$$\widetilde{J} = \frac{\partial F_{\mathscr{S}}}{\partial(\xi_1,\xi_2,\xi_3)} = \begin{pmatrix} \frac{\partial t}{\partial \xi_1} & \frac{\partial t}{\partial \xi_2} & \frac{\partial t}{\partial \xi_3} \\ \frac{\partial x}{\partial \xi_1} & \frac{\partial x}{\partial \xi_2} & \frac{\partial x}{\partial \xi_3} \\ \frac{\partial y}{\partial \xi_1} & \frac{\partial y}{\partial \xi_2} & \frac{\partial y}{\partial \xi_3} \end{pmatrix} \tag{B.3}$$

with $(\xi_1,\xi_2,\xi_3)^{\mathrm{T}} \in [-1,1]^3$. The inverse of $\widetilde{J}$ is equal to

$$\widetilde{J}^{-1} = \begin{pmatrix} \frac{\partial \xi_1}{\partial t} & \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial t} & \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \\ \frac{\partial \xi_3}{\partial t} & \frac{\partial \xi_3}{\partial x} & \frac{\partial \xi_3}{\partial y} \end{pmatrix}. \tag{B.4}$$

We denote the first column of $\widetilde{J}^{-1}$ as $\widetilde{J}_t^{-1} \in \mathbb{R}^{3\times 1}$.

The integrals defined in (5.5) and (5.6) can now be expressed as

$$\mathscr{A}_{ij}^{\mathscr{K}} = \int_{\widehat{\mathscr{K}}} \widehat{\mathscr{N}}_{\mathscr{K},i}\widehat{\mathscr{N}}_{\mathscr{K},j} |\det(J)|\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2\,\mathrm{d}\xi_3\,\mathrm{d}\xi_4, \tag{B.5}$$

$$\mathscr{B}_{ij}^{\mathscr{K}} = \int_{\widehat{\mathscr{K}}} (J_{\bar{x}}^{-\mathrm{T}}\widehat{\nabla}\widehat{\mathscr{N}}_{\mathscr{K},i}) \cdot (J_{\bar{x}}^{-\mathrm{T}}\widehat{\nabla}\widehat{\mathscr{N}}_{\mathscr{K},j}) |\det(J)|\,\mathrm{d}\xi_1\,\mathrm{d}\xi_2\,\mathrm{d}\xi_3\,\mathrm{d}\xi_4, \tag{B.6}$$

$$\mathscr{C}_{ij}^{LR;\mathscr{F}} = \int_{\widehat{\mathscr{F}}} \widehat{\mathscr{N}}_{L,j}\bar{n}_L \cdot (J_{\bar{x}}^{-\mathrm{T}}\widehat{\nabla}\widehat{\mathscr{N}}_{R,i}) J_{\mathscr{F}}\,\mathrm{d}\xi_{i_1}\,\mathrm{d}\xi_{i_2}\,\mathrm{d}\xi_{i_3}, \tag{B.7}$$

$$\mathscr{D}_{ijk}^{LR;\mathscr{F}} = \int_{\widehat{\mathscr{F}}} \bar{n}_{L,k}\widehat{\mathscr{N}}_{L,i}\widehat{\mathscr{N}}_{R,j} J_{\mathscr{F}}\,\mathrm{d}\xi_{i_1}\,\mathrm{d}\xi_{i_2}\,\mathrm{d}\xi_{i_3}, \tag{B.8}$$

$$\mathscr{H}_{ij}^{\mathscr{K};\mathscr{F}} = \int_{\widehat{\mathscr{F}}} \widetilde{\mathscr{N}}_i J_t^{-\mathrm{T}} \widehat{\nabla} \widehat{\mathscr{N}}_j J_{\mathscr{F}} \,\mathrm{d}\xi_{i_1} \,\mathrm{d}\xi_{i_2} \,\mathrm{d}\xi_{i_3}, \tag{B.9}$$

$$\mathscr{U}_{ij}^{+;\mathscr{F}} = \int_{\widehat{\mathscr{F}}^+} \widehat{\mathscr{N}}^+_{i;\xi_1=-1} \widehat{\mathscr{N}}^+_{j;\xi_1=-1} J_{\mathscr{F}^+} \,\mathrm{d}\xi_{j_2} \,\mathrm{d}\xi_{j_3}, \tag{B.10}$$

$$\mathscr{V}_{ij}^{-;\mathscr{F}} = \int_{\widehat{\mathscr{F}}^+} \widehat{\mathscr{N}}^+_{i;\xi_1=-1} \widehat{\mathscr{N}}^-_{j;\xi_1=1} J_{\mathscr{F}^+} \,\mathrm{d}\xi_{j_2} \,\mathrm{d}\xi_{j_3}, \tag{B.11}$$

$$\mathscr{Q}_{ijl}^{\mathscr{K}} = \frac{1}{2} \int_{\widehat{\mathscr{F}}} \widetilde{\mathscr{N}}_i (J_{\bar{x}}^{-\mathrm{T}} \widehat{\nabla} \widehat{\mathscr{N}}_j) \cdot (J_{\bar{x}}^{-\mathrm{T}} \widehat{\nabla} \widehat{\mathscr{N}}_l) J_{\mathscr{F}} \,\mathrm{d}\xi_{i_1} \,\mathrm{d}\xi_{i_2} \,\mathrm{d}\xi_{i_3}, \tag{B.12}$$

$$\mathscr{I}_i^{\mathscr{K};\mathscr{F}} = \int_{\widehat{\mathscr{F}}} g_N \widehat{\mathscr{N}}_i J_{\mathscr{F}} \,\mathrm{d}\xi_{i_1} \,\mathrm{d}\xi_{i_2} \,\mathrm{d}\xi_{i_3}, \tag{B.13}$$

$$\mathscr{S}_{ij}^{\mathscr{K};\mathscr{F}} = \int_{\widehat{\mathscr{F}}} \widetilde{\mathscr{N}}_i \widetilde{\mathscr{N}}_j J_{\mathscr{F}} \,\mathrm{d}\xi_{i_1} \,\mathrm{d}\xi_{i_2} \,\mathrm{d}\xi_{i_3}, \tag{B.14}$$

$$\begin{aligned}\mathscr{P}_{ij}^{\mathscr{K};\mathscr{F}}(\zeta_{\mathscr{S}}) = {} & \int_{\widehat{\mathscr{F}}} \frac{\widehat{\mathscr{N}}_i (\widetilde{J}_t^{-\mathrm{T}} \widetilde{\nabla} \widetilde{\mathscr{N}}_j) J_{\mathscr{F}} \,\mathrm{d}\xi_{i_1} \,\mathrm{d}\xi_{i_2} \,\mathrm{d}\xi_{i_3}}{\left(\sum_{p=1}^{N_s} \sum_{q=1}^{N_s} \zeta_{\mathscr{S},p} \zeta_{\mathscr{S},q} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}_p) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}_q) + 1\right)^{\frac{1}{2}}} \\ & + \int_{\widehat{\mathscr{F}}^+} \frac{\widehat{\mathscr{N}}^+_{i;\xi_1=-1} \widetilde{\mathscr{N}}^+_{j;\xi_1=-1} J_{\mathscr{F}^+} \,\mathrm{d}\xi_{j_2} \,\mathrm{d}\xi_{j_3}}{\left(\sum_{p=1}^{N_s} \sum_{q=1}^{N_s} \zeta_{\mathscr{S},p} \zeta_{\mathscr{S},q} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{p;\xi_1=-1}) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{q;\xi_1=-1}) + 1\right)^{\frac{1}{2}}},\end{aligned} \tag{B.15}$$

$$\begin{aligned}\mathscr{R}_{ijkl}^{\mathscr{K};\mathscr{F}}(\zeta_{\mathscr{S}}) = {} & \int_{\widehat{\mathscr{F}}} \frac{\widehat{\mathscr{N}}_i (\widetilde{J}_t^{-\mathrm{T}} \widetilde{\nabla} \widetilde{\mathscr{N}}_j) (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}_k) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}_l) J_{\mathscr{F}} \,\mathrm{d}\xi_{i_1} \,\mathrm{d}\xi_{i_2} \,\mathrm{d}\xi_{i_3}}{\left(\sum_{p=1}^{N_s} \sum_{q=1}^{N_s} \zeta_{\mathscr{S},p} \zeta_{\mathscr{S},q} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}_p) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}_q) + 1\right)^{\frac{3}{2}}} \\ & + \int_{\widehat{\mathscr{F}}^+} \frac{\widehat{\mathscr{N}}^+_{i;\xi_1=-1} \widetilde{\mathscr{N}}^+_{j;\xi_1=-1} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{k;\xi_1=-1}) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{l;\xi_1=-1}) J_{\mathscr{F}^+} \,\mathrm{d}\xi_{j_2} \,\mathrm{d}\xi_{j_3}}{\left(\sum_{p=1}^{N_s} \sum_{q=1}^{N_s} \zeta_{\mathscr{S},p} \zeta_{\mathscr{S},q} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{p;\xi_1=-1}) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{q;\xi_1=-1}) + 1\right)^{\frac{3}{2}}},\end{aligned} \tag{B.16}$$

$$\mathscr{X}_{ijkl}^{-;\mathscr{F}}(\zeta_{\mathscr{S}}) = \int_{\widehat{\mathscr{F}}^+} \frac{\widehat{\mathscr{N}}^+_{i;\xi_1=-1} \widetilde{\mathscr{N}}^-_{j;\xi_1=1} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{k;\xi_1=-1}) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{l;\xi_1=-1}) J_{\mathscr{F}^+} \,\mathrm{d}\xi_{j_2} \,\mathrm{d}\xi_{j_3}}{\left(\sum_{p=1}^{N_s} \sum_{q=1}^{N_s} \zeta_{\mathscr{S},p} \zeta_{\mathscr{S},q} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{p;\xi_1=-1}) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{q;\xi_1=-1}) + 1\right)^{\frac{3}{2}}}, \tag{B.17}$$

$$\mathscr{Y}_{ij}^{-;\mathscr{F}}(\zeta_{\mathscr{S}}) = \int_{\widehat{\mathscr{F}}^+} \frac{\widehat{\mathscr{N}}^+_{i;\xi_1=-1} \widetilde{\mathscr{N}}^-_{j;\xi_1=1} J_{\mathscr{F}^+} \,\mathrm{d}\xi_{j_2} \,\mathrm{d}\xi_{j_3}}{\left(\sum_{p=1}^{N_s} \sum_{q=1}^{N_s} \zeta_{\mathscr{S},p} \zeta_{\mathscr{S},q} (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{p;\xi_1=-1}) \cdot (\widetilde{J}^{-\mathrm{T}} \widetilde{\mathscr{N}}^+_{q;\xi_1=-1}) + 1\right)^{\frac{1}{2}}} \tag{B.18}$$

with $\xi_{i_1}, \xi_{i_2}, \xi_{i_3} \in \{\xi_1, \xi_2, \xi_3, \xi_4\}$ the three local coordinates defining the surface $\mathscr{F}$ and $\xi_{j_2}, \xi_{j_3} \in \{\xi_2, \xi_3, \xi_4\}$ the two local coordinates defining the surfaces $\mathscr{F}^+$ or $\mathscr{F}^-$. The surface Jacobian $J_{\mathscr{F}}$ is defined as

$$J_{\mathscr{F}} = \left| \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_1}} \wedge \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_2}} \wedge \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_3}} \right|$$

and the surface Jacobians $J_{\mathscr{F}^\pm}$ as

$$J_{\mathscr{F}^\pm} = \left| \frac{\partial F^\pm_{\mathscr{K}}}{\partial \xi_{j_2}} \wedge \frac{\partial F^\pm_{\mathscr{K}}}{\partial \xi_{j_3}} \right|.$$

Here, the jth component of the wedge product $w = a \wedge b \wedge c \in \mathbb{R}^4$, with $a, b, c \in \mathbb{R}^4$ is computed by the rule $w^j = \det(a, b, c, e_j)$, with e_j the jth basis vector in $\mathbb{R}^4$. Similarly, the space–time normal vector is computed as

$$n = s \frac{\frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_1}} \wedge \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_2}} \wedge \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_3}}}{\left| \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_1}} \wedge \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_2}} \wedge \frac{\partial F_{\mathscr{K}}}{\partial \xi_{i_3}} \right|},$$

where $s = \pm 1$ is such that the normal vector is pointing outward of element $\mathscr{K}$. The space normal vector $\bar{n} \in \mathbb{R}^3$ consists of the last three components of the space–time normal vector n.

Appendix C. Construction of linear system for the expansion coefficients in the Newton method

In this appendix we provide the details for the construction of the linear system for the expansion coefficients in the Newton method on an unstructured mesh.

The solution of the weak formulation (4.18) requires the approximation of the local lifting operator $R_{\mathscr{F}} \in \mathbb{R}^3$, defined in (3.5). Instead of computing the local lifting operator separately for each face it is more efficient to introduce the local lifting operator directly into the algebraic system for the expansion coefficients. We briefly summarize this procedure. Since $R_{\mathscr{F}}$ is only nonzero in the two elements $\mathscr{K}_L$ and $\mathscr{K}_R$ which are connected to the face $\mathscr{F}$, we have:

$$\int_{\mathscr{K}_L} R_{\mathscr{F},L}([[\phi_h]]) \cdot v_L \,\mathrm{d}x + \int_{\mathscr{K}_R} R_{\mathscr{F},R}([[\phi_h]]) \cdot v_R \,\mathrm{d}x = \frac{1}{2}\int_{\mathscr{F}} (\phi_{L,h}\bar{n}_L + \phi_{R,h}\bar{n}_R)\cdot(v_L + v_R)\,\mathrm{d}s, \quad \forall v_L, v_R \in \Sigma_h^p. \tag{C.1}$$

We approximate the kth component of the lifting operator and test functions v_L, v_R, with $k = 1, 2, 3$, as

$$(R_{\mathscr{F},\mathscr{K}}([[\phi_h]](x))_k = \sum_{j=1}^{N_p} R_{jk}^{\mathscr{K};\mathscr{F}} \mathscr{N}_{\mathscr{K},j}(x), \quad \forall x \in \mathscr{K}, \tag{C.2}$$

$$(v_{\mathscr{K}}(x))_k = \sum_{j=1}^{N_p} v_{\mathscr{K},jk} \mathscr{N}_{\mathscr{K},j}(x), \quad \forall x \in \mathscr{K}. \tag{C.3}$$

If we introduce (C.2) and (C.3) into (C.1), and use the fact that this equation must be satisfied for arbitrary test functions v_L, v_R, then we obtain the following relations for the coefficients of the lifting operator in (C.2):

$$R_{nk}^{L;\mathscr{F}} = \frac{1}{2}\sum_{j=1}^{N_p}(E_{njk}^{L;\mathscr{F}}\phi_{L,j} + F_{njk}^{RL;\mathscr{F}}\phi_{R,j}),$$

$$R_{nk}^{R;\mathscr{F}} = \frac{1}{2}\sum_{j=1}^{N_p}(F_{njk}^{LR;\mathscr{F}}\phi_{L,j} + E_{njk}^{R;\mathscr{F}}\phi_{R,j})$$

with $E^{L;\mathscr{F}}, F^{LR;\mathscr{F}} \in \mathbb{R}^{N_p\times N_p\times 3}$ defined as

$$E_{njk}^{\mathscr{K};\mathscr{F}} := \sum_{i=1}^{N_p}(\mathscr{A}^{-1})_{ni}^{\mathscr{K}}\mathscr{D}_{ijk}^{\mathscr{K}\mathscr{K};\mathscr{F}}, \quad F_{njk}^{LR;\mathscr{F}} := \sum_{i=1}^{N_p}(\mathscr{A}^{-1})_{ni}^{R}\mathscr{D}_{jik}^{LR;\mathscr{F}}. \tag{C.4}$$

The matrix $\mathscr{M}$ is now constructed as follows:

(a) Initialize $\mathscr{M} = 0$,

(b) $[M_{ij}]^{\mathscr{K}\mathscr{K}} = \mathscr{B}_{ij}^{\mathscr{K}}, \quad \forall i,j \in \{1,\ldots,N_p\}, \forall \mathscr{K} \in \{1,\ldots,N_{\mathscr{T}}\}$,

(c) $$[M_{ij}]^{LL} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}_I^n}\left\{[M_{ij}]^{LL} - \frac{1}{2}(\mathscr{C}_{ij}^{LL;\mathscr{F}} + \mathscr{C}_{ji}^{LL;\mathscr{F}}) + \frac{1}{4}\tilde{\eta}_{\mathscr{F}}\sum_{k=1}^{d-1}\sum_{n=1}^{N_p}(E_{njk}^{L;\mathscr{F}}D_{ink}^{LL;\mathscr{F}} + F_{njk}^{LR;\mathscr{F}}D_{ink}^{LR;\mathscr{F}})\right\},$$

(d) $$[M_{ij}]^{LR} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}_I^n}\left\{[M_{ij}]^{LR} - \frac{1}{2}(\mathscr{C}_{ij}^{RL;\mathscr{F}} + \mathscr{C}_{ji}^{LR;\mathscr{F}}) + \frac{1}{4}\tilde{\eta}_{\mathscr{F}}\sum_{k=1}^{d-1}\sum_{n=1}^{N_p}(F_{njk}^{RL;\mathscr{F}}D_{ink}^{LL;\mathscr{F}} + E_{njk}^{R;\mathscr{F}}D_{ink}^{LR;\mathscr{F}})\right\},$$

(e) $$[M_{ij}]^{RL} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}_I^n}\left\{[M_{ij}]^{RL} - \frac{1}{2}(\mathscr{C}_{ij}^{LR;\mathscr{F}} + \mathscr{C}_{ji}^{RL;\mathscr{F}}) + \frac{1}{4}\tilde{\eta}_{\mathscr{F}}\sum_{k=1}^{d-1}\sum_{n=1}^{N_p}(E_{njk}^{L;\mathscr{F}}D_{ink}^{RL;\mathscr{F}} + F_{njk}^{LR;\mathscr{F}}D_{ink}^{RR;\mathscr{F}})\right\},$$

(f) $$[M_{ij}]^{RR} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}_I^n}\left\{[M_{ij}]^{RR} - \frac{1}{2}(\mathscr{C}_{ij}^{RR;\mathscr{F}} + \mathscr{C}_{ji}^{RR;\mathscr{F}}) + \frac{1}{4}\tilde{\eta}_{\mathscr{F}}\sum_{k=1}^{d-1}\sum_{n=1}^{N_p}(F_{njk}^{RL;\mathscr{F}}D_{ink}^{RL;\mathscr{F}} + E_{njk}^{R;\mathscr{F}}D_{ink}^{RR;\mathscr{F}})\right\},$$

$$\text{(g)}\quad [M_{ij}]^{\mathscr{K},\mathscr{K}+N_{\mathscr{T}}} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}^n_{\mathscr{S}}} \left\{ [M_{ij}]^{\mathscr{K},\mathscr{K}+N_{\mathscr{T}}} + \mathscr{P}^{\mathscr{K};\mathscr{F}}_{ij}(\zeta^{n,(m)}_{\mathscr{S}}) - \sum_{l=1}^{N_s}\sum_{k=1}^{N_s} \zeta^{n,(m)}_{\mathscr{S},l}\zeta^{n,(m)}_{\mathscr{S},k}\left(\mathscr{R}^{\mathscr{K};\mathscr{F}}_{ilkj}(\zeta^{n,(m)}_{\mathscr{S}}) - \mathscr{X}^{-;\mathscr{F}}_{ilkj}(\zeta^{n,(m)}_{\mathscr{S}})\right)\right\},$$
$$\forall i\in\{1,\ldots,N_p\},\forall j\in\{1,\ldots,N_s\},$$

$$\text{(h)}\quad [M_{ij}]^{\mathscr{K}+N_{\mathscr{T}},\mathscr{K}} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}^n_{\mathscr{S}}} \left\{ [M_{ij}]^{\mathscr{K}+N_{\mathscr{T}},\mathscr{K}} + \mathscr{H}^{\mathscr{K};\mathscr{F}}_{ij} + \mathscr{U}^{+;\mathscr{F}}_{ij} + 2\sum_{l=1}^{N_p}\mathscr{Q}^{\mathscr{K};\mathscr{F}}_{ijl}\Phi^{n;(m)}_{\mathscr{K},l}\right\},$$
$$\forall i\in\{1,\ldots,N_s\},\forall j\in\{1,\ldots,N_p\},$$

$$\text{(i)}\quad [M_{ij}]^{\mathscr{K}+N_{\mathscr{T}},\mathscr{K}+N_{\mathscr{T}}} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}^n_{\mathscr{S}}} \left\{ [M_{ij}]^{\mathscr{K}+N_{\mathscr{T}},\mathscr{K}+N_{\mathscr{T}}} + \mathscr{S}^{\mathscr{K};\mathscr{F}}_{ij}\right\}$$
$$\forall i,j\in\{1,\ldots,N_s\}$$

with $\tilde{\eta}_{\mathscr{F}} = \eta_{\mathscr{F}} + n_f$ and $d = \dim(\mathscr{E})$. The righthand side vector F is constructed as follows:

$$\text{(j)}\quad [F_i]^{\mathscr{K}} = 0,\quad \forall i\in\{1,\ldots,N_p\},\ \forall\mathscr{K}\in\{1,\ldots,N_{\mathscr{T}}+N_{\mathscr{S}}\},$$

$$\text{(k)}\quad [F_i]^{\mathscr{K}} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}^n_{\mathscr{N}}} \{[F_i]^{\mathscr{K}} + \mathscr{J}^{\mathscr{K};\mathscr{F}}_i\},\quad \forall i\in\{1,\ldots,N_p\},$$

$$\text{(l)}\quad [F_i]^{\mathscr{K}} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}^n_{\mathscr{S}}} \left\{ [F_i]^{\mathscr{K}} - \sum_{l=1}^{N_s}\sum_{j=1}^{N_s}\sum_{k=1}^{N_s} \zeta^{n,(m)}_{\mathscr{S},l}\zeta^{n,(m)}_{\mathscr{S},j}\zeta^{n,(m)}_{\mathscr{S},k}\mathscr{R}^{\mathscr{K};\mathscr{F}}_{ijkl}(\zeta^{n,(m)}_{\mathscr{S}}) \right.$$
$$\left. + \sum_{j=1}^{N_s}\zeta^{n-1}_{\mathscr{S},j}(\mathscr{Y}^{-;\mathscr{F}}_{ij}(\zeta^{n,(m)}_{\mathscr{S}}) + \sum_{l=1}^{N_s}\sum_{k=1}^{N_s}\zeta^{n,(m)}_{\mathscr{S},l}\zeta^{n,(m)}_{\mathscr{S},k}\mathscr{X}^{-;\mathscr{F}}_{ijkl}(\zeta^{n,(m)}_{\mathscr{S}}))\right\},\quad \forall i\in\{1,\ldots,N_p\},$$

$$\text{(m)}\quad [F_i]^{\mathscr{K}+N_{\mathscr{T}}} \leftarrow \sum_{\mathscr{F}\in\mathscr{F}^n_{\mathscr{S}}} \left\{ [F_i]^{\mathscr{K}+N_{\mathscr{T}}} + \sum_{j=1}^{N_p}\sum_{l=1}^{N_p}\phi^{n;(m)}_{\mathscr{K},j}\phi^{n;(m)}_{\mathscr{K},l}\mathscr{Q}^{\mathscr{K};\mathscr{F}}_{ijl} + \sum_{j=1}^{N_p}\phi^{n-1}_{\mathscr{K},j}\mathscr{V}^{-;\mathscr{F}}_{ij}\right\},$$
$$\forall i\in\{1,\ldots,N_s\}.$$

The construction of the matrix on a general unstructured mesh is now straightforward. First, we initialize all entries to zero, followed by a loop over all elements. Next under (c) till (i) we loop over all faces, compute the block matrix entries for the two elements with indices L and R which are connected to the face $\mathscr{F}$, and store these entries in the blocks $[M]^{LR}$. The same procedure then is followed for the righthand side. This whole procedure does not depend on the chosen type of elements and is suitable for any unstructured mesh.

References

[1] V.R. Ambati, Flooding and drying in discontinuous Galerkin discretizations of shallow water equations, in: Proceedings ECCOMAS CFD Conference, 5–8 September 2006, Egmond aan Zee, The Netherlands, http://proceedings.fyper.com/eccomascfd2006/, 2006.

[2] V.R. Ambati, O. Bokhove, Space–time discontinuous Galerkin finite element method for shallow water flows, J. Comput. Phys. and Appl. Math., in press, doi:10.1016/j.cam.2006.01.047.

[3] V.R. Ambati, O. Bokhove, Space–time discontinuous Galerkin finite element method for rotating shallow water flows on moving grids, TW Memorandum 1811, http://eprints.eemcs.utwente.nl/6972/01/1811.pdf, 2006.

[4] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2002) 1749–1779.

[5] H. Braess, P. Wriggers, Arbitrary Lagrangian Eulerian finite element analysis of free surface flow, Comput. Meth. Appl. Mech. Eng. 190 (2000) 95–109.

[6] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo, Discontinuous finite elements for diffusion problems, in: Atti Convegno in Onore di F. Brioschi, Instituto Lombardo, Accademia di Scienze e Lettere, Milan, Italy, 1999, pp. 197–217.

[7] X. Cai, H.P. Langtangen, B.F. Nielsen, A. Tvieto, A finite element method for fully nonlinear water waves, J. Comput. Phys. 143 (1998) 544–568.

[8] B. Cockburn, Discontinuous Galerkin methods for convection-dominated problems, in: High-Order Methods for Computational Physics, in: T.J. Barth, H. Deconinck (Eds.), Lecture Notes in Computational Science and Engineering, vol. 9, Springer, 1999, pp. 69–224.

[9] B. Cockburn, G. Karniadakis, C.-W. Shu, The development of discontinuous Galerkin methods, in: Discontinuous Galerkin Methods: Theory, Computation and Applications, in: B. Cockburn, G. Karniadakis, C.-W. Shu (Eds.), Lecture Notes in Computational Science and Engineering, vol. 11, Springer, 2000, pp. 3–50, Part I: Overview.
[10] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comp. 16 (2001) 173–261.
[11] W. Craig, C. Sulem, Numerical simulation of gravity waves, J. Comput. Phys. 108 (1993) 73–83.
[12] F. Duarte, R. Gormaz, S. Natesan, Arbitrary Lagrangian–Eulerian method for Navier–Stokes equations with moving boundaries, Comput. Meth. Appl. Mech. Eng. 193 (2004) 4819–4836.
[13] C.M. Klaij, J.J.W. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations, J. Comput. Phys. 217 (2006) 589–611.
[14] P.M. Knupp, A robust elliptic grid generator, J. Comput. Phys. 100 (1992) 409–418.
[15] M.J. Lighthill, Waves in Fluids, Cambridge University Press, UK, 1978.
[16] Q.W. Ma, G.X. Wu, R. Eatock Taylor, Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 1: Numerical results and validation, Int. J. Numer. Meth. Fluids 36 (2001) 287–308.
[17] Q.W. Ma, G.X. Wu, R. Eatock Taylor, Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 2: Methodology and numerical procedure, Int. J. Numer. Meth. Fluids 36 (2001) 265–285.
[18] Q.W. Ma, S. Yan, Quasi ALE finite element method for nonlinear water waves, J. Comput. Phys. 212 (2006) 52–72.
[19] A. Masud, T.J.R. Hughes, A space–time Galerkin/least-squares finite element formulation of the Navier–Stokes equations for moving domain problems, Comput. Meth. Appl. Mech. Eng. 146 (1997) 91–126.
[20] J.N. Newman, Marine Hydrodynamics, MIT Press, Cambridge, MA, 1977.
[21] D.P. Nicholls, Traveling water waves: spectral continuation methods with parallel implementation, J. Comput. Phys. 143 (1998) 224–240.
[22] I. Robertson, S.J. Sherwin, Free-surface flow simulation using *hp*/Spectral elements, J. Comput. Phys. 155 (1999) 26–53.
[23] A. Soulaimani, Y. Saad, An arbitrary Lagrangian–Eulerian finite element method for solving three-dimensional free surface flows, Comput. Meth. Appl. Mech. Eng. 162 (1998) 79–106.
[24] M. Souli, J.P. Zolesio, Arbitrary Lagrangian Eulerian and free surface methods in fluid mechanics, Comput. Meth. Appl. Mech. Eng. 191 (2001) 451–466.
[25] J.J. Sudirham, J.J.W. van der Vegt, R.M.J. van Damme, Space–time discontinuous Galerkin method for advection–diffusion problems on time-dependent domains, Appl. Numer. Math. 56 (2006) 1491–1518.
[26] J.J.W. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. Part I: General formulation, J. Comput. Phys. 182 (2002) 546–585.
[27] J.J.W. van der Vegt, S.K. Tomar, Discontinuous Galerkin method for linear free-surface gravity waves, J. Sci. Comput. 22–23 (2005) 531–567.
[28] H. van der Ven, J.J.W. van der Vegt, Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. Part II: Efficient flux quadrature, Comput. Meth. Appl. Mech. Eng. 191 (2002) 4747–4780.
[29] J.H. Westhuis, The numerical simulation of nonlinear waves in a hydrodynamic model test basin, Ph.D. thesis, University of Twente, Enschede, The Netherlands, 2001.
[30] G.B. Whitham, Linear and Nonlinear Waves, Wiley, New York, 1974.
[31] G.X. Wu, R.E. Taylor, Finite element analysis of two-dimensional non-linear transient water waves, Appl. Ocean Res. 16 (1994) 363–372.
[32] G.X. Wu, Z.Z. Hu, Simulation of nonlinear interactions between waves and floating bodies through a finite-element based numerical tank, Proc. R. Soc. Lond. Ser. A 460 (2004) 2797–2817.