



Discontinuous Galerkin scheme for the spherical shallow water equations with applications to tsunami modeling and prediction

Boris Bonev^{a,*}, Jan S. Hesthaven^a, Francis X. Giraldo^b, Michal A. Kopera^c

^a École polytechnique fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

^b Naval Postgraduate School, Monterey, CA 93943, USA

^c University of California Santa Cruz, Santa Cruz, CA 95064, USA

ARTICLE INFO

Article history:

Received 14 November 2017

Received in revised form 5 February 2018

Accepted 5 February 2018

Keywords:

Shallow water equations

Well-balanced scheme

Wetting and drying

Discontinuous Galerkin method

Curved mesh

Non-conforming mesh

ABSTRACT

We present a novel high-order discontinuous Galerkin discretization for the spherical shallow water equations, able to handle wetting/drying and non-conforming, curved meshes in a well-balanced manner. This requires a well-balanced discretization, that cannot rely on exact quadrature, due to the curved mesh. Using the strong form of the discontinuous Galerkin discretization, we achieve a splitting of the well-balanced condition into individual problems for the flux and volume terms, which has significant advantages: It allows for the construction of non-conforming, well-balanced flux discretizations, i.e. we can perform non-conforming mesh refinement while preserving the well-balanced property of the scheme. More importantly, this approach enables the development of a new method for handling wet/dry transitions. In contrast to other wetting/drying methods, it is well-balanced and able to handle wetting/drying robustly at any polynomial order, without the introduction of physical model assumptions such as viscosity, artificial porosity or cancellation of gravity.

We perform a series of one-dimensional tests and analyze the properties of our scheme. In order to validate our method for the simulation of large-scale tsunami events on the rotating sphere, we perform numerical simulations of the 2011 Tohoku tsunami and compare our results to real-world buoy data. The method is able to predict arrival times and wave amplitudes accurately even over long distances. This indicates that our method accurately captures all physical phenomena relevant to the long-term evolution of tsunami waves.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Numerical methods for the simulation of the shallow water equations have seen a considerable amount of research interest during the last decades, as they can be used to model a wide variety of physical phenomena from storm surges to tsunami propagation. One of the long-standing goals is the development of a tsunami early warning system for hazard

* Corresponding author.

E-mail addresses: boris.bonev@epfl.ch (B. Bonev), jan.hesthaven@epfl.ch (J.S. Hesthaven), fxgiraldo@nps.edu (F.X. Giraldo), makopera@ucsc.edu (M.A. Kopera).

<https://doi.org/10.1016/j.jcp.2018.02.008>

0021-9991/© 2018 Elsevier Inc. All rights reserved.

forecast and risk assessment. This requires a physically accurate model and a robust and efficient numerical method for the simulation of the tsunami propagation.

The simulation of large-scale tsunami events pose some challenges for the physical model and the numerical method. On the model side, it is crucial to take interaction with land masses and effects of Earth's curvature into account. Tsunami events affect oceans globally, thus for realistic simulations at large scales, we have to incorporate curvature of the sphere, wetting/drying and the Coriolis force into the model.

The numerical method on the other hand has to be adaptive, robust and highly efficient. A particularly promising numerical method is the discontinuous Galerkin (dG) method, as it offers numerous advantages over competing methods for the simulation of wave propagation problems. Among them are high-order accuracy, innate parallelism and the flexibility of *hp*-adaptivity [1]. Additionally, for the simulation of tsunamis we require the well-balanced property and the ability to handle wetting and drying accurately. Since tsunamis can be regarded as perturbations of the water at rest steady state, it is important to use well-balanced schemes, which can conserve the steady state numerically [2–4]. Flooding and drying is important as tsunami waves interact with land masses and capturing the correct, physical behavior at the wet/dry transition is crucial for the accurate simulation of tsunamis.

In this work, we present a novel method for the simulation of the shallow water equations on the surface of a rotating sphere using the discontinuous Galerkin method. This requires the development of a high-order dG method, which can handle curved, non-conforming meshes with discontinuous bottom topography and wet/dry transitions in a well-balanced manner.

Although well-balanced schemes have received considerable attention, there appears to be only limited past work considering well-balanced discontinuous Galerkin schemes on curved meshes. The key difficulty is the lack of exact numerical integration on which many of the well-balanced methods rely on [5–8]. Notable exceptions are presented by Chandrashekar and Zenk [9] and Wintermeyer et al. [10]. In both papers, the authors exploit the strong form of the discontinuous Galerkin formulation to match the discretizations of flux and source terms and construct a well-balanced scheme in this way. In the latter, a well-balanced and entropy-stable dG scheme is presented for the shallow water equations on curvilinear elements. However, no discussion of non-conforming meshes or wetting/drying is included.

The inclusion of land masses poses the question of how to treat the shoreline in the discontinuous Galerkin context. One approach is to handle shorelines as free boundaries [11], which is in line with the physical model as it loses its validity in dry areas. This has the advantage that numerical problems, associated with dry areas in the solution, are avoided. On the other hand, an accurate model for the dynamics of the shoreline is required in order to calculate the position of the moving boundary. Often, constant remeshing is required and topological changes need to be taken care of, amounting to prohibitive computational costs. An alternative adopted in numerous works is to handle wetting and drying within the computational domain. These approaches can be seen as a special type of immersed boundary method. Wet/dry transitions introduce three distinct numerical difficulties: maintaining positivity of the water height, introduction of unphysical pressure gradients, and numerical instabilities due to the discontinuities associated with the transition. Various methods exist to ensure positivity of the approximate solution [7,12,13]. A popular method introduced by Xing et al. [12] is to maintain the positivity of cell-averages using a timestep restriction. Positivity on the nodes is then ensured by using a positivity-preserving limiter, which rescales the polynomial around the positive average. While it is claimed that this method is well-balanced [12,14], it is not unconditionally well-balanced as partly dry cells are neglected. Semi-dry cells introduce artificial gradients and generate unphysical waves at the shores. This effect, also called “numerical storms” sometimes [10], has been observed by other authors, but there exists no satisfactory solution to this problem. Kesserwani and Liang [8] propose a reconstruction of nodal values such that the pressure gradients vanish for the “lake at rest” solution. The authors present this method for a piecewise linear method in one dimension and it is unclear how this approach performs for higher-order methods. Bunya et al. [15] as well as [6] propose to cancel gravity in these cells to eliminate the problem of artificial pressure gradients. This requires the introduction of dual-valued fluxes in order to make the scheme well-balanced for the case of the “lake at rest” solution. While both wet/dry treatments make the scheme well-balanced, they are not consistent with the physical model at the shores and appear to be restricted to piecewise linear polynomials. Other approaches use “artificial porosity” and introduce a fraction indicator to represent how much of the cell is wet and how much is dry [16,17]. This allows for implicit time integration with large steps but introduces other problems such as higher wave speeds in the wet/dry region and an altered shallow water model. Finally there is the issue of stability at the wet/dry interface. Most previously mentioned wetting/drying algorithms reduce the order of the solution to linear polynomials and apply a slope limiter to prevent unphysical discharges [12,15]. Meister and Ortleb [7] use an implicit scheme with a modal filter and a shock indicator to stabilize the scheme in the nearly dry regions.

The numerical solution of the shallow water equations on the sphere using the discontinuous Galerkin method has been investigated by various authors, although typically neglecting the influence of bottom topography [18,19]. Blaise and St-Cyr [20] construct a discontinuous Galerkin method on the sphere for tsunami simulations and use an adjoint method in order to optimize initial conditions based on buoy data comparison [21]. This method is promising as it demonstrates the feasibility of using discontinuous Galerkin simulations on the sphere in combination with an adjoint method to reconstruct initial conditions from buoy data to obtain accurate early warnings. Their method lacks wetting/drying however, and shores are treated as reflecting boundaries, which result in unphysical reflections at the shorelines.

In this paper, we propose a general method for the construction of well-balanced dG discretizations on curved, non-conforming meshes with wet/dry transitions. Our method is based on the observation that the condition for a well-balanced

scheme is split into two individual conditions for the flux and volume terms by utilizing the strong form of the dG method. In this way, we are able to independently construct a well-balanced flux discretization for non-conforming meshes and a well-balanced volume discretization for the treatment of partly dry cells. The wet/dry treatment has the advantage that it is well-balanced, works for any order and is consistent with the physical model without adding any computational cost. We demonstrate the well-balanced property of our method both for the one-dimensional shallow water equations and for the spherical shallow water equations presented in [18]. Adding to that, we showcase the accuracy of our wetting and drying algorithm using one-dimensional examples. On the sphere, we demonstrate that our method can be applied to the simulation of large-scale tsunami events. Based on either cubed-sphere or icosahedral grids, we can use local refinement to create meshes adapted to our problem. This is enabled by our well-balanced flux, which allows non-conforming meshes and therefore, static and dynamic local refinement. We perform a validation of the 2011 Tohoku tsunami event and compare our results to buoy data gathered by the DART network [22]. The comparison with the buoy data shows that the method is capable of accurately predicting the arrival times and wave amplitudes of such large-scale tsunami events. As our method is parallelizable by design and as the structured meshes allow easy optimization and load-balancing, we expect that this approach is able to give highly accurate forecasts within minutes if implemented on GPUs [23,24].

2. The numerical scheme

In the following, we give a brief introduction to the discontinuous Galerkin scheme for the spherical shallow water equations as presented in [18]. As it will be beneficial to revert to a simpler model, we will also recall the one-dimensional shallow water equations. Within a spatial domain $\mathbf{x} \in \Omega$ and the temporal domain $t \in (0, \infty)$, we express the models as an initial value problem

$$\begin{cases} \frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{S}(\mathbf{x}, \mathbf{q}) & \text{in } \Omega \times (0, \infty) \\ \mathbf{q} = \mathbf{q}_0 & \text{on } \Omega \times \{t = 0\}, \end{cases} \quad (1)$$

comprising m conservation laws, an initial condition $\mathbf{q}_0 : \Omega \rightarrow \mathbb{R}^m$ and suitable boundary conditions on $\partial\Omega$, which will be specified when necessary. Here $\mathbf{q} = \mathbf{q}(\mathbf{x}, t) : \Omega \times (0, \infty) \rightarrow \mathbb{R}^m$ is the unknown solution which maps the coordinates onto the conserved variables. $\mathbf{F}(\mathbf{q})$ and $\mathbf{S}(\mathbf{x}, \mathbf{q})$ are flux and source functions respectively, which we will specify independently for both problems.

2.1. One-dimensional shallow water equations

In the one-dimensional case, we have $\Omega = [x_0, x_1] \subset \mathbb{R}$ and a system of two conservation laws for the state variables

$$\mathbf{q} = [\varphi \quad \varphi u]^\top. \quad (2)$$

Here $\varphi = gh$ denotes the geopotential height, where h is the water column height and g the acceleration due to gravity. Physical solutions only permit strictly positive water heights $\varphi > 0$. The second unknown φu is the discharge and represents the momentum of the water column as it moves with velocity u along the first coordinate direction. The flux $\mathbf{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ and source $\mathbf{S} : \Omega \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ are specified as

$$\mathbf{F}(\mathbf{q}) = \begin{bmatrix} \varphi u \\ \varphi u^2 + \frac{1}{2} \varphi^2 \end{bmatrix} \quad (3)$$

and

$$\mathbf{S}(\mathbf{x}, \mathbf{q}) = \begin{bmatrix} 0 \\ -\varphi \nabla \tau \end{bmatrix}. \quad (4)$$

The source term accounts for the pressure gradient, which is introduced by the slope of the bottom topography $\tau : \Omega \rightarrow \mathbb{R}$. Again, we measure the bottom topography $\tau = gb$ in geopotential units, where b is the altitude of the topography relative to the sea level.

2.2. Spherical shallow water equations

In the spherical case, the spatial domain Ω is the two-dimensional sphere $S^2(R) \subset \mathbb{R}^3$ with radius R . We adopt the approach in [18], which avoids the use of spherical coordinates by using Cartesian coordinates. This comes at the cost of another state variable and we have the four-dimensional state vector

$$\mathbf{q} = [\varphi \quad \varphi u \quad \varphi v \quad \varphi w]^\top, \quad (5)$$

which contains the three-dimensional velocity vector $\mathbf{u} = [u \ v \ w]^\top$. The flux is given as

$$\begin{aligned} \mathbf{F}(\mathbf{q}) &= \mathbf{f}_x(\mathbf{q}) \hat{\mathbf{i}} + \mathbf{f}_y(\mathbf{q}) \hat{\mathbf{j}} + \mathbf{f}_z(\mathbf{q}) \hat{\mathbf{k}} \\ &= \begin{bmatrix} \varphi u \\ \varphi u^2 + \frac{1}{2}\varphi^2 \\ \varphi uv \\ \varphi uw \end{bmatrix} \hat{\mathbf{i}} + \begin{bmatrix} \varphi v \\ \varphi uv \\ \varphi v^2 + \frac{1}{2}\varphi^2 \\ \varphi vw \end{bmatrix} \hat{\mathbf{j}} + \begin{bmatrix} \varphi w \\ \varphi uw \\ \varphi vw \\ \varphi w^2 + \frac{1}{2}\varphi^2 \end{bmatrix} \hat{\mathbf{k}}, \end{aligned} \quad (6)$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are the unit vectors along the coordinate axes in \mathbb{R}^3 . The above notation has to be understood in the sense that scalar-products with three-dimensional, Cartesian vectors act on the unit vectors. The source term $\mathbf{S} = [0, \tilde{\mathbf{S}}]^\top$ only acts on the conservation of momentum and consists of three terms:

$$\tilde{\mathbf{S}}(\mathbf{x}, \mathbf{q}) = \mathbf{C}(\mathbf{x}, \mathbf{u}) - \varphi \nabla \tau(\mathbf{x}) + \mu \mathbf{x}. \quad (7)$$

The first term is the Coriolis force

$$\mathbf{C}(\mathbf{x}, \mathbf{u}) = -\frac{2\omega z \varphi}{R^2} \mathbf{x} \times \mathbf{u}, \quad (8)$$

where ω is the angular velocity of the sphere. Note that we have chosen the z-axis to be the rotation axis of the sphere. The second term accounts for the pressure gradient, caused by the slope of the bottom topography $\tau = gb(\mathbf{x})$. Finally, we must ensure that the velocity vector \mathbf{u} remains tangential to the surface of the sphere. This is enforced through the Lagrange multiplier

$$\mu(\mathbf{x}, \mathbf{q}) = \frac{1}{R^2} \mathbf{x} \cdot (\varphi \nabla \tau + \nabla \cdot \tilde{\mathbf{F}}), \quad (9)$$

which projects the change in \mathbf{u} to the surface of the sphere. Here, $\tilde{\mathbf{F}}$ denotes the parts of the flux which act on the discharges.

2.3. Discontinuous Galerkin discretization

Let us briefly introduce the Runge–Kutta discontinuous Galerkin scheme [18] for the solution of the spherical SWEs. We will focus on the spherical case as most aspects of the scheme translate directly to the one-dimensional case. We divide the physical domain Ω into a set \mathcal{G} of non-overlapping quadrilateral, curvilinear elements $D^k \in \mathcal{G}$ such that $\Omega = \bigcup_{k=1}^K D^k$. The numerical solution $\mathbf{q}_N(\mathbf{x}, t)$ in the space of piecewise polynomials

$$\mathbb{V}(\Omega, \mathcal{G}) := \{v \in L^2(\Omega) \mid \forall D^k \in \mathcal{G}, v|_{D^k} \in \mathbb{P}(D^k)\} \quad (10)$$

is expressed as the direct sum

$$\mathbf{q}(\mathbf{x}, t) \approx \mathbf{q}_N(\mathbf{x}, t) = \bigoplus_{k=1}^K \mathbf{q}_N^k(\mathbf{x}, t), \quad (11)$$

where $\mathbf{q}_N^k(\mathbf{x}, t) \in \mathbb{P}(D^k)$ is the local polynomial approximation to the solution in D^k . For the sake of brevity, we drop the element-index k unless necessary. To construct a basis on each element, we introduce the reference element $I = [-1, 1]$ in the one-dimensional case and $I = [-1, 1]^2$ in the spherical case. Then, there exists a bijective mapping $\Psi: D \rightarrow I$, so that we can translate coordinates $\mathbf{x} \in D$ in the physical domain to coordinates $\xi = \Psi(\mathbf{x})$ within the reference element (Fig. 1). In the reference element, we construct a polynomial basis $\{L_i(\xi)\}$ of N -th order using Lagrange basis functions on a set of points chosen to allow a high order accurate interpolation. In one dimension, we choose Legendre–Gauss–Lobatto (LGL) points $\{\xi_i\}_{i=1}^{N+1} \subset [-1, 1]$ and construct Lagrange interpolating polynomials

$$L_i(\xi) = l_i(\xi) = \prod_{j=1, j \neq i}^{N+1} \frac{\xi - \xi_j}{\xi_i - \xi_j}, \quad (12)$$

with the property $l_i(\xi_j) = \delta_{ij}$. In the spherical case, we utilize a tensor-product of the one-dimensional Lagrange functions (12) and create the basis functions

$$L_m(\xi) = l_i(\xi) l_j(\eta), \quad (13)$$

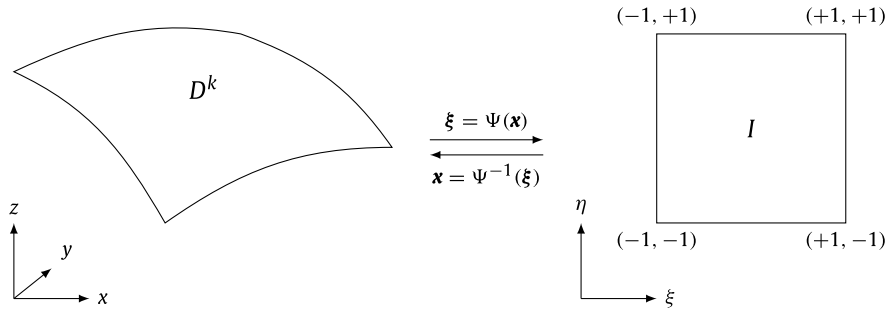


Fig. 1. Transformation into the reference element.

where $m = 1, 2, \dots, (N+1)^2$ is a unique index for each node $\xi_m = (\xi_i, \eta_j)$ with individual indices $i, j = 1, 2, \dots, N+1$. With the nodal basis in place, we can now represent the solution as a linear combination

$$\mathbf{q}_N^k(\mathbf{x}, t) = \sum_{m=1}^{(N+1)^2} \hat{\mathbf{q}}_m^k(t) L_m(\mathbf{x}) = \sum_{i,j=1}^{N+1} \mathbf{q}_N^k(\mathbf{x}(\xi_i, \eta_j), t) l_i(\xi(\mathbf{x})) l_j(\eta(\mathbf{x})) \quad (14)$$

using the nodal values $\hat{\mathbf{q}}_j^k(t) = \mathbf{q}_N^k(\mathbf{x}_j^k, t)$. With a slight abuse of notation, we will use $L_i(\mathbf{x})$ to denote $L_i(\xi(\mathbf{x}))$ and similarly $\mathbf{x}_i = \mathbf{x}(\xi_i)$. For our purposes, we fix the polynomial order to be the same in all elements.

Using the Legendre–Gauss–Lobatto nodes on the reference element, we can construct quadrature rules \mathcal{Q}_D , $\mathcal{Q}_{\partial D}$ for the approximate evaluation of volume and surface integrals on D and ∂D . In the spherical case, we approximate the volume integral of an integrable function $g \in L^1(D)$ by

$$\begin{aligned} \int_D g(\mathbf{x}) d\mathbf{x} &= \int_I g(\xi) J_D(\xi) d\xi \\ &\approx \sum_{i,j=1}^N g(\xi_i, \eta_j) J_D(\xi_i, \eta_j) \omega_i \omega_j =: \mathcal{Q}_D[g(\mathbf{x})], \end{aligned} \quad (15)$$

where $J_D(\xi)$ is the Jacobian determinant of the reference transformation $\Psi^{-1}(\xi)$ and ω_i, ω_j are the quadrature weights associated with the Legendre–Gauss–Lobatto nodes ξ_i, η_j . LGL nodes provide quadrature points at the edges of the reference element, which makes the construction of a quadrature rule for boundary integrals natural:

$$\begin{aligned} \oint_{\partial D} g(\mathbf{x}) d\mathbf{x} &= \oint_{\partial I} g(\xi) J_{\partial D}(\xi) d\xi \\ &\approx \sum_{i=1}^N g(\xi_i, -1) J_{\partial D}(\xi_i, -1) \omega_i + \sum_{j=1}^N g(1, \eta_j) J_{\partial D}(1, \eta_j) \omega_j \\ &\quad + \sum_{i=1}^N g(\xi_i, 1) J_{\partial D}(\xi_i, 1) \omega_i + \sum_{j=1}^N g(-1, \eta_j) J_{\partial D}(-1, \eta_j) \omega_j \\ &=: \mathcal{Q}_{\partial D}[g(\mathbf{x})]. \end{aligned} \quad (16)$$

Here, we have introduced a second Jacobian determinant $J_{\partial D}$, which arises from the transformation onto the edges of the reference element. The construction of quadrature rules in one-dimension can be done in the same way: volume integrals are approximated as the sum over the quadrature nodes and the surface integrals are reduced to the evaluation at the element boundary. The quadrature rules (15) and (16) provide exact integration for integrands with polynomial degrees up to $2N-1$ [25]. Due to the transformation onto the curved elements, we cannot obtain exact integration as the integrands contain Jacobian determinants, which are not polynomials.

With the polynomial representation and the approximation of integrals in place, we move on to the problem of satisfying the conservation law (1). We write the weak form of the discontinuous Galerkin formulation

$$\int_D \left(\frac{\partial \mathbf{q}_N}{\partial t} - \mathbf{F}_N \cdot \nabla - \mathbf{S}_N \right) L_i d\mathbf{x} = - \oint_{\partial D} \hat{\mathbf{n}} \cdot \mathbf{F}_N^* L_i d\mathbf{x}, \quad (17)$$

which should be satisfied by the numerical solution \mathbf{q}_N in each element and for every basis function L_i . Alternatively, we can use the strong form of the Galerkin formulation

$$\int_D \left(\frac{\partial \mathbf{q}_N}{\partial t} + \nabla \cdot \mathbf{F}_N - \mathbf{S}_N \right) L_i \, d\mathbf{x} = \oint_{\partial D} \hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*) L_i \, d\mathbf{x}, \quad (18)$$

which is acquired by integrating (17) by parts. Here we have introduced polynomial approximations for the flux

$$\mathbf{F}_N(\mathbf{q}_N) = \sum_{j=1}^{(N+1)^2} \mathbf{F}(\mathbf{q}(\mathbf{x}_j)) L_j(\mathbf{x}), \quad (19)$$

and source term

$$\mathbf{S}_N(\mathbf{x}, \mathbf{q}_N) = \sum_{j=1}^{(N+1)^2} \mathbf{S}(\mathbf{x}_j, \mathbf{q}(\mathbf{x}_j)) L_j(\mathbf{x}). \quad (20)$$

Due to the discontinuous nature of the ansatz functions, both formulations require the introduction of a suitable numerical flux \mathbf{F}^* , which connects the solutions in the individual elements through a single-valued flux. We use the local Lax–Friedrichs flux given as

$$\mathbf{F}_N^*(\mathbf{q}_N^-, \mathbf{q}_N^+) = \frac{1}{2} (\mathbf{F}_N(\mathbf{q}_N^-) + \mathbf{F}_N(\mathbf{q}_N^+)) - \frac{c}{2} (\mathbf{q}_N^+ - \mathbf{q}_N^-), \quad (21)$$

where \mathbf{q}_N^- stands for the local solution within the element and \mathbf{q}_N^+ for the solution in the neighboring element. The factor c represents the maximum local wave speed across the element boundary and it is calculated as

$$c = \max \left\{ |\hat{\mathbf{n}} \cdot \mathbf{u}_N^-| + \sqrt{\varphi_N^-}, |\hat{\mathbf{n}} \cdot \mathbf{u}_N^+| + \sqrt{\varphi_N^+} \right\}. \quad (22)$$

We project the numerical flux onto the normals of the element edges $\hat{\mathbf{n}}$, thus evaluating the flux through the edge $\hat{\mathbf{n}} \cdot \mathbf{F}_N^*$.

Finally, let us discuss the discretization of the flux and source terms (19), (20) in more detail. In the weak form, we write out the flux term as

$$\mathbf{F}_N \cdot \nabla L_i = \sum_{j=1}^{(N+1)^2} \left(\mathbf{f}_x(\hat{\mathbf{q}}_j) L_j \frac{\partial L_i}{\partial x} + \mathbf{f}_y(\hat{\mathbf{q}}_j) L_j \frac{\partial L_i}{\partial y} + \mathbf{f}_z(\hat{\mathbf{q}}_j) L_j \frac{\partial L_i}{\partial z} \right) \quad (23)$$

and in the strong form as

$$(\nabla \cdot \mathbf{F}_N) L_i = \sum_{j=1}^{(N+1)^2} \left(\mathbf{f}_x(\hat{\mathbf{q}}_j) \frac{\partial L_j}{\partial x} L_i + \mathbf{f}_y(\hat{\mathbf{q}}_j) \frac{\partial L_j}{\partial y} L_i + \mathbf{f}_z(\hat{\mathbf{q}}_j) \frac{\partial L_j}{\partial z} L_i \right). \quad (24)$$

The difference between the two lies in the derivative acting on either the test function or on the Lagrange function over which the sum is formed. This will become important later, when we discuss how to achieve the well-balanced property in both formulations.

The source term discretization is then constructed in the following way: A numerical approximation of the bottom topography is constructed as

$$\tau_N(\mathbf{x}) = \sum_{i=1}^{(N+1)^2} \tau(\mathbf{x}_i) L_i(\mathbf{x}) \quad (25)$$

by interpolating the topography within each element D . Using this piecewise polynomial approximation, we construct the source term

$$\tilde{\mathbf{S}}_N = \sum_{j=1}^{(N+1)^2} \left(\mathbf{c}(\mathbf{x}_j, \mathbf{u}(\mathbf{x}_j)) L_j - \varphi(\mathbf{x}_j) L_j \sum_{k=1}^{(N+1)^2} \tau(\mathbf{x}_k) \nabla L_k(\mathbf{x}_j) \right). \quad (26)$$

Here we have omitted the Lagrange multiplier and enforce it implicitly by projecting the change in velocity onto the sphere [18].

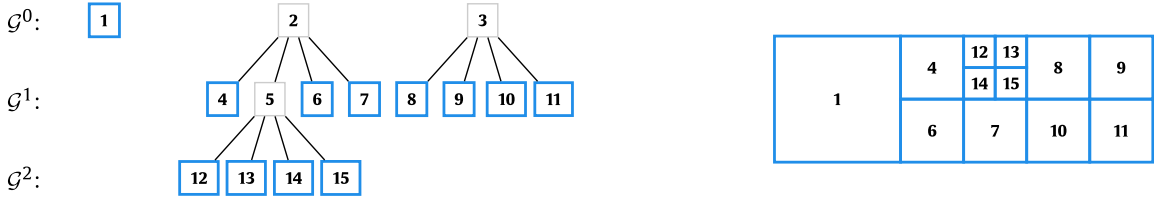


Fig. 2. The grid refinement hierarchy is shown on the left, where highlighted elements represent cells that are active in the mesh. The corresponding mesh is depicted on the right. The grid is balanced in the sense that the difference in refinement levels of any two neighboring cells is at most one.

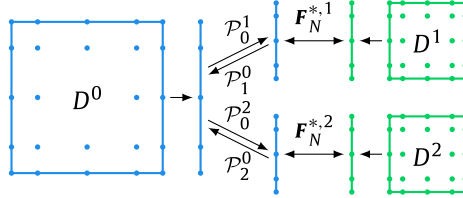


Fig. 3. Treatment of the hanging node in non-conforming discretizations. The solution on the edge of the parent element D^0 is projected onto the edges of the children elements for the computation of the numerical fluxes. This solution is then projected back to the parent edge.

By replacing the integrals in the dG formulation with the quadrature rules (15) and (16), we obtain a semi-discrete scheme, in the form of a nonlinear system of ordinary differential equations

$$\frac{\partial \hat{\mathbf{q}}_N(t)}{\partial t} = \mathbf{R}_N(\hat{\mathbf{q}}_N(t)), \quad (27)$$

for the vector of unknowns $\hat{\mathbf{q}}_N = [\hat{\mathbf{q}}_1^1, \hat{\mathbf{q}}_2^1, \dots, \hat{\mathbf{q}}_{(N+1)^2}^K]^T$ with the right-hand side \mathbf{R}_N . We use the popular three-step, third-order strong stability preserving Runge–Kutta method SSPRK(3, 3) presented in [26], which can be written as a convex combination of four Euler timesteps

$$\hat{\mathbf{q}}_N(t + \Delta t) = \hat{\mathbf{q}}_N(t) + \Delta t \mathbf{R}_N(\hat{\mathbf{q}}_N(t)). \quad (28)$$

2.4. Non-conforming meshes

We use structured grids, such as the icosahedral grids presented in [18], in combination with either static or dynamic mesh refinement to generate locally refined grids. We briefly outline our method of handling such non-conforming meshes and restrict ourselves to the analysis of balanced non-conforming grids, in which only one “hanging node” is permitted at each interface. We refer the reader to [27–29] for the details of the presented mesh refinement procedures.

Let \mathcal{G}^0 denote a conforming, non-overlapping partition of Ω into quadrilaterals. Subdivision of each quadrilateral into four children elements generates a series of conforming partitions $\mathcal{G}^0, \mathcal{G}^1, \mathcal{G}^2, \dots$. For each of these grids, we call the superscript the refinement level of said grid. We arrange these into a quadtree forest \mathcal{F} , such that each element is connected to its parent element. At any time t , we will use a subforest $\mathcal{G}(t) \subset \mathcal{F}$, such that $\mathcal{G}(t)$ is a balanced, non-overlapping partition of Ω . A balanced grid is characterized by the property that the difference in refinement levels of two neighboring elements cannot be greater than 1. In other words, there are no more than three elements interfacing at one edge. An example of a balanced grid with corresponding forest structure is displayed in Fig. 2.

Any non-conforming refinement of the mesh leads to edges with hanging nodes. This makes the computation of fluxes more challenging as high order points do not align on non-conforming edges. To resolve this, an intermediate solution with matching high order points is computed, which allows the calculation of the numerical flux. This method is otherwise known as the mortar element method [30]. Let $D^0 \in \mathcal{G}^0$ and $D^1, D^2 \in \mathcal{G}^1$ be elements interfacing at the common edge $E^0 = E^1 \cup E^2$, where $E^1 = \partial D^1 \cap \partial D^0$ and $E^2 = \partial D^2 \cap \partial D^0$ are the edges of the children elements. Such a situation is depicted in Fig. 3. For this edge, we have to specify the computation of the three fluxes $\mathbf{F}_N^{*,0}(\mathbf{q}_N^0, \mathbf{q}_N^1 \oplus \mathbf{q}_N^2)$, $\mathbf{F}_N^{*,1}(\mathbf{q}_N^1, \mathbf{q}_N^0)$ and $\mathbf{F}_N^{*,2}(\mathbf{q}_N^2, \mathbf{q}_N^0)$. Here we have introduced superscripts in order to indicate the nodes that are used to represent the numerical fluxes. We first evaluate the fluxes on the high order points of the children elements:

$$\mathbf{F}_N^{*,1}(\mathbf{q}_N^1, \mathbf{q}_N^0) = \mathbf{F}_N^*(\mathbf{q}_N^1, \mathcal{P}_0^1 \mathbf{q}_N^0), \quad (29)$$

$$\mathbf{F}_N^{*,2}(\mathbf{q}_N^2, \mathbf{q}_N^0) = \mathbf{F}_N^*(\mathbf{q}_N^2, \mathcal{P}_0^2 \mathbf{q}_N^0), \quad (30)$$

where we have introduced the “scatter” projection operators $\mathcal{P}_0^1, \mathcal{P}_0^2$, which project the polynomial on E^0 onto the nodal bases on E^1 and E^2 , respectively. Because the polynomial bases are of the same degree, these projection operators do not change the polynomial itself. We write the projection operators, however, to indicate the change of the nodal basis. In this way, the definition of the numerical fluxes (21) is unambiguous, as both polynomials in the argument are defined on the same set of nodes. In a similar way, we define the “gather” projection operators $\mathcal{P}_1^0, \mathcal{P}_2^0$, which project the polynomials from the nodes of the children edges E^1 and E^2 onto the nodes of the parent edge E^0 . Using these projections, we define the numerical flux on the parent node as

$$\mathbf{F}_N^{*,0}(\mathbf{q}_N^0, \mathbf{q}_N^1 \oplus \mathbf{q}_N^2) = \frac{1}{2}(\mathcal{P}_1^0 \mathbf{F}_N^*(\mathcal{P}_0^1 \mathbf{q}_N^0, \mathbf{q}_N^1) + \mathcal{P}_2^0 \mathbf{F}_N^*(\mathcal{P}_0^2 \mathbf{q}_N^0, \mathbf{q}_N^2)). \quad (31)$$

Note that the flux function is first evaluated on the nodal points of the children edges and then projected to the high-order points of the parent element. As such it has $2N$ degrees of freedom and it is represented as a piecewise polynomial. Thus we can break down the L^2 projection into the two individual L^2 projections $\mathcal{P}_1^0, \mathcal{P}_2^0$ of their respective domains. This includes loss of information as the flux is projected onto a basis of dimension N . In the context of the mortar element method, this approach of performing the flux computation on the edge with the higher approximation order is also known as the *maximum rule* [31].

3. Well-balanced property

We are interested in solutions which can be regarded as perturbations of the so-called “lake at rest” solution $\bar{\mathbf{q}}$ given as

$$\varphi = \max(\varphi_0 - \tau, 0), \quad (32)$$

$$\varphi \mathbf{u} = \mathbf{0}, \quad (33)$$

where φ_0 is a constant water surface height. For both spherical and one-dimensional shallow water equations, this solution satisfies

$$\nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{S}(\mathbf{x}, \mathbf{q}). \quad (34)$$

The notion of a ‘well-balanced’ scheme originates in the idea of achieving this balance numerically:

Definition 1. Let $\bar{\mathbf{q}}_N = [\bar{\varphi}_N, \bar{\varphi} \mathbf{u}_N]^T$ denote the numerical representation of the ‘lake at rest’ solution, such that $\forall \mathbf{x}_i$:

$$\bar{\varphi}_N(\mathbf{x}_i) = \max\{\varphi_0 - \tau_N(\mathbf{x}_i), 0\} \quad (35)$$

$$\bar{\varphi} \mathbf{u}_N(\mathbf{x}_i) = \mathbf{0}. \quad (36)$$

We call a scheme with right-hand side $\mathbf{R}_N(\mathbf{q}_N)$ *well-balanced*, iff it exactly preserves the ‘lake at rest’ solution, i.e.

$$\mathbf{R}_N(\bar{\mathbf{q}}_N) = \mathbf{0}. \quad (37)$$

In fact, with the word exact we mean $\mathcal{O}(\epsilon\varphi)$, where ϵ is the relative rounding error of floating point arithmetic performed on the machine. This is due to cancellation effects, which cannot be avoided. For the sake of readability we assume that floating point arithmetic is performed exactly.

3.1. Construction of a well-balanced scheme

The construction of a well-balanced scheme requires careful attention to the discretization. To demonstrate this, consider the weak form of the scheme. By replacing the integrals with the quadrature rules, the right-hand side for the weak form becomes

$$\begin{aligned} \mathbf{R}(\mathbf{q}_N) &= \int_D \mathbf{F}_N \cdot \nabla L_i - \mathbf{S}_N L_i \, d\mathbf{x} + \oint_{\partial D} \hat{\mathbf{n}} \cdot \mathbf{F}_N^* \, d\mathbf{x} \\ &\approx \mathcal{Q}_D[\mathbf{F}_N \cdot \nabla L_i - \mathbf{S}_N L_i] + \mathcal{Q}_{\partial D}[\hat{\mathbf{n}} \cdot \mathbf{F}_N^* L_i] =: \mathbf{R}_N(\mathbf{q}_N). \end{aligned} \quad (38)$$

By inserting the numerical representation of the steady-state solution $\bar{\mathbf{q}}_N$, we find that generally, this formulation will not be well-balanced. Unless both integrands are 0 individually, we have to perform integration by parts in order to use the balanced property (34) of $\bar{\mathbf{q}}_N$. Integration by parts relies on exact numerical integration, which is a property that the

curvilinear discretization does not possess. The strong form

$$\begin{aligned} \mathbf{R}(\mathbf{q}_N) &= \int_D (\nabla \cdot \mathbf{F}_N - \mathbf{S}_N) L_i \, d\mathbf{x} - \oint_{\partial D} \hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*) L_i \, d\mathbf{x} \\ &\approx \mathcal{Q}_D[(\nabla \cdot \mathbf{F}_N - \mathbf{S}_N)L_i] - \mathcal{Q}_{\partial D}[\hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*) L_i] =: \mathbf{R}_N(\mathbf{q}_N) \end{aligned} \quad (39)$$

on the other hand, is obtained through integration by parts and we observe that we can directly plug in the well-balanced property (34) without relying on exact quadrature. The property that divergence-free fields remain constant in this form is connected to the discrete version of the metric identities, which are satisfied by the strong form of the discontinuous Galerkin method. This has been originally observed by Kopriva [32], who provides a detailed analysis on the metric identities in both formulations.

Proposition 1. Let $\bar{\mathbf{q}}_N$ satisfy the discrete, nodal balance condition $\nabla \cdot \mathbf{F}_N(\bar{\mathbf{q}}_N) - \mathbf{S}_N(\bar{\mathbf{q}}_N) = 0$. Moreover, let the numerical flux be consistent for this solution, such that $\mathbf{F}_N^*(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+) = \mathbf{F}_N(\bar{\mathbf{q}}_N)$. Then, the strong form of the discontinuous Galerkin scheme $\mathbf{R}_N(\mathbf{q}_N)$ is well-balanced, regardless of the exactness of the quadrature rule.

Proof. Plugging in yields $\mathbf{R}_N(\bar{\mathbf{q}}_N) = \mathcal{Q}_D[0] - \mathcal{Q}_{\partial D}[0] = 0$.

The advantages of the strong form lie in the conditions for Proposition 1. In other words, to obtain a well-balanced scheme, it is sufficient to guarantee $\nabla \cdot \mathbf{F}_N(\bar{\mathbf{q}}_N) = \mathbf{S}_N(\bar{\mathbf{q}}_N)$ and $\mathbf{F}_N(\bar{\mathbf{q}}_N) = \mathbf{F}_N^*(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+)$ individually. Thus we do not need to pay attention to the volume discretization when constructing the face discretization. This will prove useful in the construction of the well-balanced wetting/drying method and the non-conforming flux-discretization.

Proposition 2. Let \mathcal{G} be a conforming mesh of Ω and let the approximation of the bathymetry $\tau_N \in \mathbb{V}(\Omega, \mathcal{G})$ be continuous on Ω . Moreover let $\varphi_0 > \tau_N$, i.e. there are no dry areas in Ω . Under these circumstances, the strong form of the scheme, as presented in Section 2 is well-balanced.

Proof. We start by showing $\mathbf{F}_N^*(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+) = \mathbf{F}_N(\bar{\mathbf{q}}_N)$. We have $\mathbf{u}_N = \mathbf{0}$ and $\varphi_N = \varphi_0 - \tau_N$, which is continuous due to the continuous bathymetry τ_N . This in turn implies

$$\mathbf{F}_N^*(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+) = \mathbf{F}_N^*(\bar{\mathbf{q}}_N, \bar{\mathbf{q}}_N) = \mathbf{F}_N(\bar{\mathbf{q}}_N)$$

due to the consistency of the Lax–Friedrichs flux.

For the second part, we have to establish $\nabla \cdot \mathbf{F}_N(\bar{\mathbf{q}}_N) = \mathbf{S}_N(\bar{\mathbf{q}}_N)$. We can ignore the Lagrange multiplier as it only projects the change in velocity to the surface of the sphere. Inserting $\mathbf{u}_N = \mathbf{0}$ then reduces the equations to

$$\varphi_N \nabla \varphi_N = -\varphi_N \nabla \tau_N,$$

which is trivially satisfied by $\bar{\mathbf{q}}_N$.

3.2. Well-balanced non-conforming meshes

With the well-balanced property established for conforming meshes, we aim for this property to carry over to non-conforming discretizations. This requires a few changes to the method presented in Section 2.4. The introduction of non-conforming edges leaves the volume terms unchanged and, therefore, we only have to ensure that $\mathbf{F}_N(\bar{\mathbf{q}}_N) = \mathbf{F}_N^*(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+)$. This flux calculation involves the projections onto the children elements, the flux calculation itself, and the projection back onto the parent edge.

The projection of the polynomial on the parent edge E^0 to the edges of the children elements E^1 and E^2 is exact as we are projecting a polynomial from one N -th order basis to another. If there are no dry areas, the projection will preserve the “lake at rest” condition $\varphi_N + \tau_N = \varphi_0$. Once dry nodes are introduced, we cannot allow negative nodal values and the water surface represented by the polynomial $\varphi_N + \tau_N$ cannot remain constant. The situation is depicted in Fig. 4. This implies that the nodal “lake at rest” property (35) is not sustained by the projected solutions $\mathcal{P}_0^1 \bar{\mathbf{q}}_N^0$ and $\mathcal{P}_0^2 \bar{\mathbf{q}}_N^0$, which is a consequence of the non-constant water surface and the choice of new nodes to represent φ_N and τ_N . To circumvent this issue, we choose to reconstruct the water surface using

$$\sigma_N(\mathbf{x}_i) = \begin{cases} \varphi_N(\mathbf{x}_i) + \tau_N(\mathbf{x}_i) & \varphi_N(\mathbf{x}_i) > \varphi_{\text{tol}} \\ \varphi_0 & \text{otherwise,} \end{cases} \quad (40)$$

where φ_{tol} is a minimum water height, below which we consider the solution dry. Projection is then performed on the reconstructed variable σ_N :

$$\mathcal{P}_0^{1,*} \varphi_N^0(\mathbf{x}_i^1) := \max \{ \mathcal{P}_0^1 \sigma_N^0(\mathbf{x}_i^1) - \mathcal{P}_0^1 \tau_N^0(\mathbf{x}_i^1), 0 \}, \quad (41)$$

$$\mathcal{P}_0^{1,*}(\varphi \mathbf{u})_N^0(\mathbf{x}_i^1) := \mathcal{P}_0^1(\varphi \mathbf{u})_N^0(\mathbf{x}_i^1). \quad (42)$$

In this way, we make sure that the nodal “lake at rest” property is carried over to the children nodes \mathbf{x}_i^1 by the projection operation.

Next, we need to ensure that the analytical flux and the numerical flux are the same on the children nodes. The fundamental challenge here is that the bathymetry is generally discontinuous in general due to the non-conforming mesh. This implies that the “lake at rest” solution \mathbf{q}_N itself is also discontinuous due to $\varphi_N = \varphi_0 - \tau_N$. Since the numerical flux is single-valued it cannot match the analytic fluxes on both sides simultaneously. This situation has previously been encountered in the construction of well-balanced finite volumes schemes. Audusse et al. [3] introduce hydrostatic reconstruction, a technique which utilizes a reconstruction of the source terms at the element interfaces to alleviate this problem. This technique has also been applied to higher-order Galerkin methods with discontinuous bottom topography so to achieve a well-balanced scheme [12,33]. We form the hydrostatic reconstruction of \mathbf{q}_N^\pm at the cell interfaces:

$$\mathbf{q}_N^{\bullet,\pm} := \begin{bmatrix} \varphi_N^{\bullet,\pm} \\ (\varphi \mathbf{u})_N^{\bullet,\pm} \end{bmatrix} = \begin{bmatrix} \max \{ \varphi_N^\pm + \tau_N^\pm - \tau_N^\bullet, 0 \} \\ \varphi_N^{\bullet,\pm} \mathbf{u}_N^\pm \end{bmatrix}, \quad (43)$$

where τ_N^\bullet is the reconstructed cell interface height

$$\tau_N^\bullet := \max \{ \tau_N^-, \tau_N^+ \}. \quad (44)$$

This step can be understood as an upwind evaluation of the bottom topography and can be physically motivated [3]. At the interface, the water column height which can transfer the discharge is limited by the jump in the bottom topography. We replace the numerical flux (21) with the hydrostatically reconstructed flux

$$\mathbf{F}_N^\bullet(\mathbf{q}_N^-, \mathbf{q}_N^+) := \mathbf{F}_N^*(\mathbf{q}_N^{\bullet,-}, \mathbf{q}_N^{\bullet,+}) + \frac{1}{2}((\varphi^-)^2 - (\varphi^{\bullet,-})^2) \begin{bmatrix} 0 \\ \hat{\mathbf{i}} \\ \hat{\mathbf{j}} \\ \hat{\mathbf{k}} \end{bmatrix}. \quad (45)$$

This flux is not single-valued, as is necessary for it to be well-balanced. If we insert the “lake at rest” solution, we recover the analytical flux $\mathbf{F}(\mathbf{q}^\pm)$ on both sides of the interface.

Finally, the last projection step in (31) also requires special treatment, as it is inexact. Again, we make use of the strong form and evaluate the difference between numerical and exact fluxes directly on the children nodes. To overcome this problem, we perform “gather” projections on the difference of the numerical and analytical fluxes:

$$\frac{1}{2} \mathcal{P}_1^0(\mathbf{F}_N^\bullet(\mathcal{P}_0^1 \mathbf{q}_N^0, \mathbf{q}_N^1) - \mathbf{F}_N(\mathcal{P}_0^1 \mathbf{q}_N^0)) + \frac{1}{2} \mathcal{P}_2^0(\mathbf{F}_N^\bullet(\mathcal{P}_0^2 \mathbf{q}_N^0, \mathbf{q}_N^2) - \mathbf{F}_N(\mathcal{P}_0^2 \mathbf{q}_N^0)). \quad (46)$$

For the “lake at rest” solution the difference of numerical and analytical fluxes is $\mathbf{0}$ and as such, the projection is also $\mathbf{0}$, regardless of its accuracy.

Proposition 3. Let \mathcal{G} be a balanced, non-conforming mesh in Ω and let $\bar{\varphi}_N$ be a “lake at rest” solution in the sense of (35). Then the modifications (40)–(46) yield a well-balanced evaluation of flux terms at the non-conforming interfaces.

Proof. Because the non-conformity of the discretization does not affect the evaluation of the volume terms, we only have to prove the well-balanced property of the fluxes.

We have already established that the “lake at rest” property is carried over onto the nodes of the children elements, when the first projection step is applied on the reconstructed water surface height. As such, we show that the evaluation of $\mathbf{F}_N^\bullet(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+)$ gives the same result as $\mathbf{F}_N(\bar{\mathbf{q}}_N)$. Hydrostatic reconstruction gives us $\varphi_N^{\bullet,-} = \varphi_N^{\bullet,+}$ due to $\bar{\varphi}_N^\pm + \tau_N^\pm = \varphi_0$. Consequently, evaluation of (45) yields

$$\mathbf{F}_N^\bullet(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+) = \mathbf{F}_N^*(\mathbf{q}_N^{\bullet,-}, \mathbf{q}_N^{\bullet,+}) + \frac{1}{2}((\varphi^+)^2 - (\varphi^{\bullet,+})^2) \begin{bmatrix} 0 \\ \hat{\mathbf{i}} \\ \hat{\mathbf{j}} \\ \hat{\mathbf{k}} \end{bmatrix} = \mathbf{F}_N(\bar{\mathbf{q}}_N),$$

where we have utilized the identity of the flux (6). Then for the final step we perform the “gather” projection steps on the flux differences $\mathbf{F}_N^\bullet(\bar{\mathbf{q}}_N^-, \bar{\mathbf{q}}_N^+) - \mathbf{F}_N(\bar{\mathbf{q}}_N)$ which are $\mathbf{0}$, as just demonstrated.

We would like to remark, that if adaptive mesh refinement is employed, we also need to ensure that refinement and coarsening of the mesh conserve the lake at rest property of \bar{q}_N . As these operations are projections, we use the same method above and utilize the reconstructed water surface variable σ_N for the projection and then convert back.

4. Treatment of wet/dry transitions

The introduction of dry areas requires a distinction between wet and dry areas. We call a node \mathbf{x}_i dry, if the water height $\varphi_N(\mathbf{x}_i)$ is smaller than a certain tolerance φ_{tol} . Consequently, a node is called wet if the opposite is true. In the context of discontinuous Galerkin methods, three situations can occur. If a cell D contains only dry nodes, we call it a *dry* cell and if it contains only wet nodes we call it a *wet* cell. In the case that the cell contains both wet and dry nodes, we call the cell *semi-dry* or *partly dry*. Such cells demand special treatment as they pose significant numerical challenges. These include the appearance of negative water heights, artificial pressure gradients and instabilities due to the discontinuity in the derivative of the state variables. To guarantee positivity, we propose to adapt the method presented in [12]. We then ensure well-balancedness by introducing a finite-difference type gradient operator which we will use only in the semi-dry cells. This operator is an efficient method of enforcing the boundary implicitly. As a result, the scheme becomes well-balanced and only weakly unstable. In a final step, we apply filtering to the semi-dry cells to stabilize the scheme.

4.1. Positivity preserving dG

A well-established method of guaranteeing positivity in finite volume schemes is to impose a timestep restriction, which guarantees positivity of the cell average φ_N^{avg} . In the context of a discontinuous Galerkin discretization, we need to make sure that each nodal value remains positive. Xing et al. [12] introduce a limiter, which rescales the solution around the positive cell-averages in a post-processing step. We give a brief introduction of the method, which we have adapted to curved grids.

Proposition 4. Let $\mathbf{q}_N(\mathbf{x}, t_n)$ denote the numerical solution at time t_n with positive water height on all nodes \mathbf{x}_i . Assuming exact integration of the integrals in (18), the cell-averaged water column height

$$\varphi_N^{\text{avg}} = \int_D \varphi_N(\mathbf{x}) \, d\mathbf{x} \quad (47)$$

remains positive after one Euler timestep (28), given that the timestep meets the CFL-like requirement

$$\frac{J_{\partial D}}{J_D} c \Delta t \leq \frac{\omega_1}{2} \quad (48)$$

everywhere.

Proof. The proof is given in Appendix A.

Even though this proof assumes exact numerical integration, which the scheme does not have, it gives an idea of how big the timestep should be to maintain positive cell-averages. In practice we do not experience any negative water height averages using half of the prescribed timestep. This property can also be retained with higher-order time integration methods if we utilize convex combinations of Euler timesteps. Strong stability preserving Runge–Kutta (SSPRK) methods possess this property and therefore preserve the positivity of the cell-averaged water height [12,26].

As we have established the positivity of the cell-averages, we correct the nodal values using the positivity limiter presented in [12]. After each timestep, we rescale the solution in each cell according to

$$\varphi_N^* = \theta(\varphi_N - \varphi_N^{\text{avg}}) + \varphi_N^{\text{avg}}, \quad (49)$$

$$\mathbf{u}_N^* = \theta(\mathbf{u}_N - \mathbf{u}_N^{\text{avg}}) + \mathbf{u}_N^{\text{avg}}, \quad (50)$$

where

$$\theta = \min \left\{ 1, \frac{\varphi_N^{\text{avg}}}{\varphi_N^{\text{avg}} - \varphi_N^{\min}} \right\}, \quad (51)$$

$$\varphi_N^{\min} = \min_{\mathbf{x}_i \in D} \{\varphi_N(\mathbf{x}_i)\}. \quad (52)$$

Here we have introduced the cell-averages φ_N^{avg} and $\mathbf{u}_N^{\text{avg}}$, which can be approximated using (15). This “positivity limiter” rescales the solution around the average water height while preserving conservation of mass. By using this limiter after each timestep, we guarantee positivity of the solution on all nodes. Finally, as we consider any node with water height below

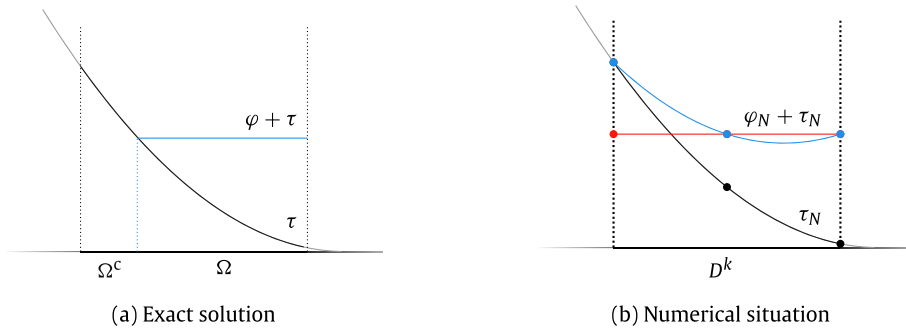


Fig. 4. Comparison of the exact and numerical “lake at rest” solutions. The polynomial approximation cannot have a vanishing slope unless unphysical, negative water heights were to be allowed.

the threshold φ_{tol} as a dry node, we have to make sure that the velocity remains $\mathbf{0}$ at these nodes. For this reason, we set $\varphi \mathbf{u} = \mathbf{0}$ at all nodes with water height $\varphi \leq \varphi_{\text{tol}}$. Clearly, this violates conservation of momentum but it is necessary in order to maintain stability and physical plausibility of the solution.

4.2. Well-balanced shores

While the previously described combination of a timestep restriction and positivity limiter handle dry areas, the outcome is not well-balanced. We analyze this situation in one dimension. Fig. 4 depicts the “lake at rest” solution and its numerical approximation. In a semi-dry cell, we cannot represent $\varphi_N + \tau_N = \text{const.}$ numerically, due to the requirement of positive water heights. Consequently, the numerical “lake at rest” solution (35) has a non-zero slope. This leads to artificial pressure gradients $\varphi_N \nabla(\varphi_N + \tau_N)$, which are not physical, since the dry domain acts on the wet domain even though there should be no interaction between these two domains. As a consequence, unphysical waves are created at the shores and pollute the domain of interest.

We propose a simple yet effective approach to mitigate this problem. To eliminate the influence of the dry subdomain, we switch to a local evaluation of the gradient. Let us introduce the finite difference operator

$$\mathcal{D}_\xi f(\xi_i) := \begin{cases} \frac{f(\xi_{i+1}) - f(\xi_{i-1})}{\xi_{i+1} - \xi_{i-1}} & \text{if } \xi_{i+1}, \xi_i, \xi_{i-1} \text{ are wet nodes} \\ \frac{f(\xi_{i+1}) - f(\xi_i)}{\xi_{i+1} - \xi_i} & \text{if } \xi_{i+1}, \xi_i \text{ are wet and } \xi_{i-1} \text{ is dry or } i - 1 < 1 \\ \frac{f(\xi_i) - f(\xi_{i-1})}{\xi_i - \xi_{i-1}} & \text{if } \xi_i, \xi_{i-1} \text{ are wet and } \xi_{i+1} \text{ is dry or } i + 1 > N \\ 0 & \text{otherwise,} \end{cases} \quad (53)$$

which takes neighboring nodes into account only if they are considered wet. This finite difference operator (53) presents a way of implicitly imposing boundary conditions within the semi-dry element. As such, it is consistent with the physical situation as we do not have any information on what the water surface $\varphi + \tau$ should be in the dry regions to recover the correct gradient in the wet subdomain. We discard this information by using the local finite difference discretization to compute the gradient of the water surface.

In the physical domain, the operators translate to the finite difference approximation

$$\mathcal{D}_x := \frac{\partial \xi}{\partial x} \mathcal{D}_\xi + \frac{\partial \eta}{\partial x} \mathcal{D}_\eta, \quad (54)$$

in the x coordinate and correspondingly we define \mathcal{D}_y and \mathcal{D}_z . Using these operators we construct the gradient-like operator

$$\nabla_N := [\mathcal{D}_x \quad \mathcal{D}_y \quad \mathcal{D}_z]^T, \quad (55)$$

which replaces the conventional gradient operator for the computation of $\varphi_N \nabla_N(\varphi_N + \tau_N)$. With this definition we can show the well-balancedness of our modified discretization:

Proposition 5. Let $\bar{\mathbf{q}}_N$ again be the numerical “lake at rest” solution (35) including dry domains. Then, let $\mathbf{R}_N(\mathbf{q}_N)$, denote the right-hand side (39), where we have replaced $\nabla(\varphi_N + \tau_N)$ with $\nabla_N(\varphi_N + \tau_N)$. The scheme is well-balanced, in spite of dry regions in the solution.

Proof. Following along the lines of the proof of Proposition 2, it is sufficient to show the final step

$$\varphi_N \nabla_N \varphi_N = -\varphi_N \nabla_N \tau_N,$$

for $\bar{\mathbf{q}}_N$. Due to the linearity of ∇_N , this is equivalent to $\varphi_N \nabla_N(\varphi_N + \tau_N) = 0$. Plugging in $\bar{\varphi}_N$ yields

$$\begin{aligned} \mathcal{D}_\xi(\max\{\varphi_0 - \tau_N(\mathbf{x}(\xi_i, \eta_j)), 0\} + \tau_N(\mathbf{x}(\xi_i, \eta_j))) \\ = \mathcal{D}_\xi(\varphi_0 - \tau_N(\mathbf{x}(\xi_i, \eta_j)) + \tau_N(\mathbf{x}(\xi_i, \eta_j))) = 0. \end{aligned}$$

The first step is permissible as any water height $\varphi \leq \varphi_{\text{tol}}$ will be ignored by \mathcal{D}_ξ anyway. The latter step is true as any finite difference of the constant function φ_0 will yield 0. This is also true for \mathcal{D}_η , and consequently for \mathcal{D}_x , \mathcal{D}_y and \mathcal{D}_z . This in turn means that

$$\nabla_N(\bar{\varphi}_N + \tau_N) = \mathbf{0},$$

which concludes our proof.

We emphasize that this operator is only used for the computation of the gradient of the water surface in the semi-dry cells. As such, the lower accuracy introduced by this operator is restricted to the semi-dry cells, in which one cannot expect high-order accuracy.

4.3. A few notes on stability

Finally, we would like to note a few things on the stability of the scheme. Numerical experiments show that the computation of the fluxes in the semi-dry cells can be unstable if performed in the wrong manner. While it might seem attractive to construct the flux term $\frac{\partial}{\partial x} \varphi u^2$ exactly using the derivatives $\frac{\partial}{\partial x} u$ and $\frac{\partial}{\partial x} \varphi$, this leads to an unstable scheme. In general, the velocity u is not continuous at the wet/dry interface and as a consequence the evaluation of $\frac{\partial}{\partial x} u$ leads to Gibbs oscillations. For this reason, we evaluate the derivatives of the flux variables $\frac{\partial}{\partial x} \varphi u^2$, $\frac{\partial}{\partial x} \varphi uv$, ... directly, which is not exact as flux variables are first approximated as polynomials but avoids the Gibbs oscillations associated with u . We would like to point out that this step introduces an error, as we first approximate the flux using an N -th order polynomial and then evaluate its derivatives.

Finally, high-order dG schemes oftentimes require some sort of artificial dissipation in order to stabilize the scheme in the presence of strong gradients. In the context of the shallow water equations, this happens in the vicinity of the shorelines as shallow water waves start to build up. To stabilize the scheme, we employ filtering, which modifies the solution according to

$$\mathbf{q}_N^F = \sum_{i,j=0}^N \sigma_i \sigma_j \hat{\mathbf{q}}_{ij} P_i(\xi) P_j(\eta), \quad (56)$$

where P_i , P_j denote the basis functions of the one-dimensional Legendre modal basis and $\hat{\mathbf{q}}_{ij}$ is the respective coefficient of the solution in the modal basis. The filter weights σ_i damp high-order modes, which has a stabilizing effect on the numerical scheme. For our purpose, the exponential filter weights

$$\sigma_i = \exp(-\alpha(i/N)^s) \quad (57)$$

are chosen and the filter parameters are set to $\alpha = 30$ and $s = 10$. The updated solution is then computed in the following manner:

- Step 1. Compute the immediate solution $\mathbf{q}_N(\mathbf{x}, t_{n+1})$ at time t_{n+1} .
- Step 2. Transform solution into water surface variables (40).
- Step 3. In the semi-dry cells apply filtering on these variables.
- Step 4. Apply the positivity-preserving limiter (49) and set $\mathbf{u} = \mathbf{0}$ on all dry nodes.

By using the reconstructed water surface (40) we ensure that filtering is performed in a well-balanced manner. Moreover, we restrict the use of the filter to semi-dry cells to maintain the high-order accuracy of our method in the fully wet areas.

5. Results in one dimension

In the following, we present results in one dimension. For the analysis of the results, we use the relative L^2 error

$$\mathcal{E}_{L^2, \Omega} = \frac{\|\mathbf{q}_N - \mathbf{q}\|_{L^2(\Omega)}}{\|\mathbf{q}\|_{L^2(\Omega)}}, \quad (58)$$

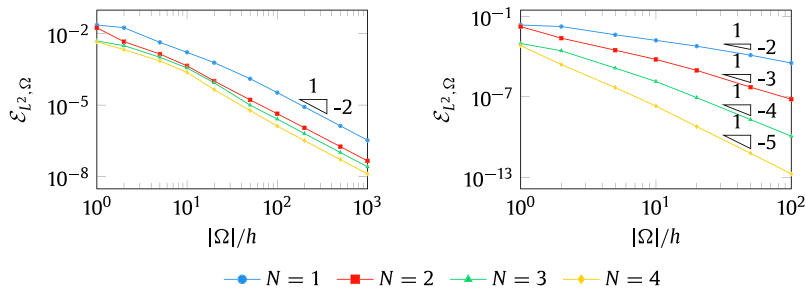


Fig. 5. Convergence behavior for the standing wave solution. We compare the accuracy of the wetting/drying flux discretization on the left to the accuracy of the regular discretization on the right.

where Ω is not necessarily the entire solution domain. The solution \mathbf{q} denotes the exact solution approximated by the numerical solution \mathbf{q}_N . As there is not always an analytical solution available, we also introduce the relative mass error

$$\mathcal{E}_{\varphi, \Omega} = \frac{\int_{\Omega} \varphi_N \, dx - \int_{\Omega} \varphi \, dx}{\int_{\Omega} \varphi \, dx} \quad (59)$$

and energy error

$$\mathcal{E}_{E, \Omega} = \frac{\int_{\Omega} E(\mathbf{q}_N) \, dx - \int_{\Omega} E(\mathbf{q}) \, dx}{\int_{\Omega} E(\mathbf{q}) \, dx} \quad (60)$$

where $E(\mathbf{q})$ denotes the total energy

$$E(\mathbf{q}) := \frac{1}{2g} (\varphi \|\mathbf{u}\|^2 + \varphi^2 + \varphi \tau). \quad (61)$$

For the conservation errors, we can replace the reference solution \mathbf{q} with the initial condition \mathbf{q}_0 , as both mass and energy are conserved by the shallow water equations.

5.1. Manufactured solution

Before we move on to more common solutions, we would like to investigate the accuracy of our wetting/drying method. By adding an extra source term, we enforce the standing wave solution

$$\begin{aligned} \varphi &= \varphi_0 + \varphi_A \cos(\omega t) \sin(\kappa x), \\ \varphi u &= -\varphi_A \frac{\omega}{\kappa} \sin(\omega t) \cos(\kappa x), \end{aligned}$$

with $\varphi_0 = 0.3$, $\varphi_A = 0.1$, $\omega = \pi$, $\kappa = \pi$. The domain is set to $\Omega = [-0.5, 0.5]$ and exact boundary conditions are used. We compare the convergence behavior of the regular flux discretization to the wetting/drying flux discretization (55). To this purpose, we use the wetting/drying discretization everywhere, even though there are no dry areas in the solution. The comparison of the two in Fig. 5 shows that our wetting/drying method is able to recover second order accuracy in the case of sufficiently smooth functions. This is to be expected, because the discretization of the flux term remains unchanged if we choose the finite difference approximation for a first order discontinuous Galerkin method on a wet domain. Therefore we can be confident that this change of flux discretization does indeed converge to the exact solution, provided that it is sufficiently smooth. In the general case, the transition from wet to dry areas is continuous but not differentiable and we cannot expect more than first order accuracy using a polynomial approximation. As such, the switch to a second order accurate method in the wetting/drying domain will not harm accuracy considerably.

5.2. Water at rest on a sloping beach

In the next test, we verify that the one-dimensional scheme is well-balanced. To do this, we use the “lake at rest” solution on a sloping bed $b(x) = 0.5x$ and set $g = 9.81$, $\Omega = [-1.5, 1.5]$, $\varphi_0 = 0.1005g$. We run simulations with filtering turned off and using the parameters $N = 3$, $K = 100$, $\Delta t = 5 \cdot 10^{-5}$, $\varphi_{\text{tol}} = 10^{-4}$. The boundaries are treated as regular cell interfaces using the exact solution on the opposing side of the interface. In Fig. 6, we compare the numerical results obtained with our method, which uses the finite difference-type volume discretization (53) in the semi-dry cell, to results obtained with the conventional dG volume discretization using the polynomial derivatives. The latter corresponds to the positivity-preserving discontinuous Galerkin discretization presented in [12]. We would like to note that we made sure that the wet/dry interface does not coincide with a cell interface, thus creating a semi-dry cell. With the conventional method,

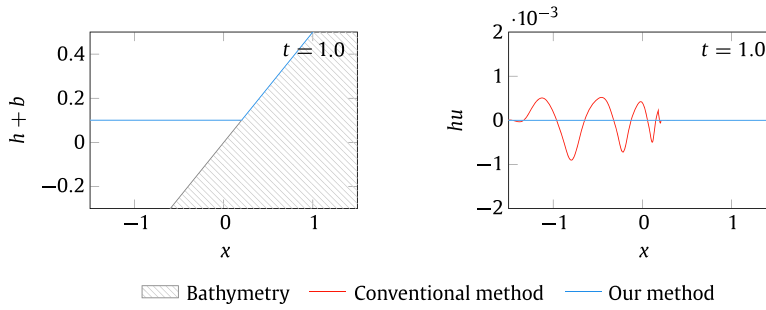


Fig. 6. Lake at rest solution on a sloping bed at $t = 1$. We compare the conventional dG method to our well-balanced, wetting/drying method.

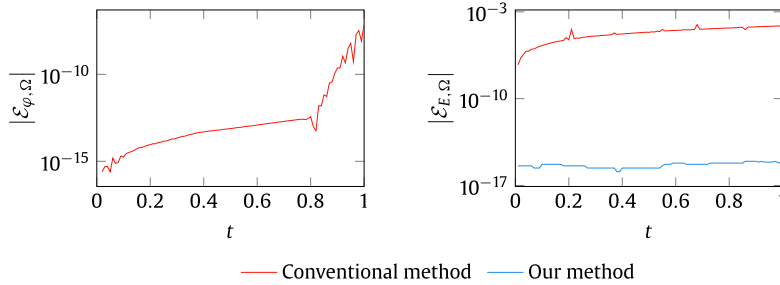


Fig. 7. Comparison of the conservation errors of our method to the conventional dG discretization. Mass conservation errors are shown on the left in logarithmic scale. Due to the logarithmic scale, the error of our method cannot be displayed as it is exactly 0. The energy conservation errors are depicted on the right.

we observe artificial waves being created in the semi-dry cell that then propagate into the solution. These are especially noticeable in the discharge plot of Fig. 6, which shows non-zero discharges from the semi-dry cell to the left. This is caused by the artificial pressure gradients, as discussed in Section 4.2. While the magnitude of the waves might seem negligible in this one-dimensional test, this is not the case in more dimensions. It is clear that the semi-dry cells require special treatment to make the wet/dry transition well-balanced.

Our wetting/drying discretization (53) achieves just that and we cannot observe any spurious movement of the water surface. We can verify that it is indeed well-balanced, by analyzing the relative mass and energy errors. The well-balanced property is in fact equivalent to the exact conservation of mass and energy in the special case of the “lake at rest” solution. Fig. 7 compares the relative mass and energy errors of our scheme to the regular dG discretization. We can clearly see that the conventional method does neither conserve energy nor mass. While the mass errors are initially small, one can see energy errors being accumulated, which is caused by the artificial pressure gradients. Once these waves start interacting with the boundary at $t \approx 0.8$, the rate at which mass conservation is violated starts to increase. The wetting/drying method that we propose, on the other hand, conserves both mass and energy for the “lake at rest” solution, as it is well-balanced. In fact, the mass error is exactly 0 and therefore cannot be plotted in logarithmic scale. The energy error is 0 within machine precision and remains stable throughout the entire simulation.

5.3. Dam break on a dry domain

Our next test case is an ideal dam break on a dry bed. Initially there is a water reservoir of height $\varphi_l = 0.1g$ on the left half-plane and a dry domain on the right side. The dam break is assumed to be instantaneous and the bottom is assumed to be flat, i.e. $\tau = 0$. The analytical solution to this problem is given by

$$\varphi(x, t) = \begin{cases} \varphi_l & x \leq x_A(t) \\ \frac{4}{9}(\sqrt{\varphi_l} - \frac{x-x_0}{2t})^2 & x_A(t) < x < x_B(t) \\ 0 & x \geq x_B(t) \end{cases}$$

and

$$u(x, t) = \begin{cases} 0 & x \leq x_A(t) \\ \frac{2}{3}(\sqrt{\varphi_l} + \frac{x-x_0}{t}) & x_A(t) < x < x_B(t) \\ 0 & x \geq x_B(t) \end{cases},$$

where $x_A(t) = x_0 - t\sqrt{\varphi_l}$ and $x_B(t) = x_0 + 2t\sqrt{\varphi_l}$ are the positions of the discontinuities [34]. This case is challenging due to the presence of the wetting process, which requires the scheme to maintain positivity, while correctly resolving

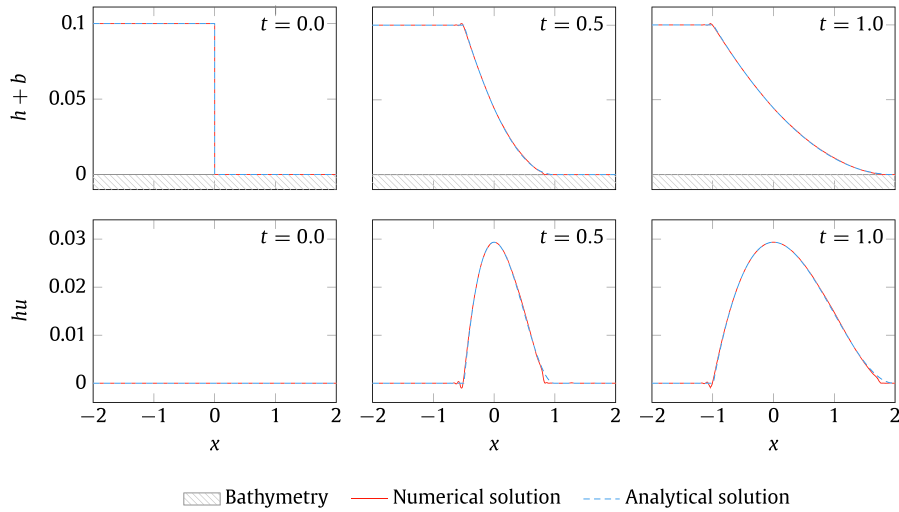


Fig. 8. Dam break on a dry bed at different times. Comparison of the numerical solution with $N = 4$ and $K = 100$ to the analytical solution.

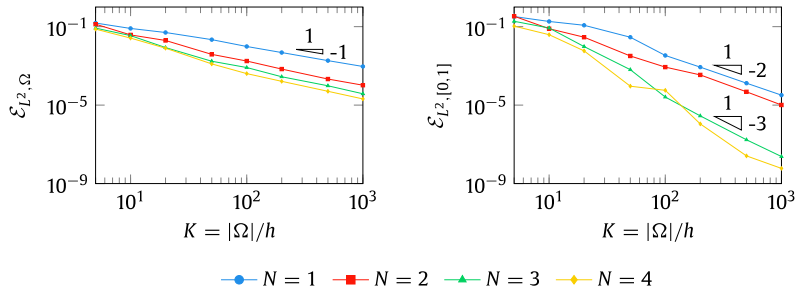


Fig. 9. Convergence for the dam break solution on a dry bed. We compare the L^2 error on Ω on the left to the L^2 error in the smooth region $[0, 1]$ on the right.

the moving shore. Along with these challenges comes the problem of the discontinuity in the derivative at the shore point, which induces Gibbs oscillations near the shoreline. These have to be controlled, otherwise they spread into the solution through the rarefaction wave. As such, we use both filtering and the positivity-preserving limiter and set the dry tolerance to $\varphi_{\text{tol}} = 10^{-6}$. Fig. 8 depicts our numerical solution computed with $N = 4$, $K = 100$, exact boundary conditions and $\Delta t = 5 \cdot 10^{-5}$ in comparison to the analytical solution. It is clear that the scheme is able to predict the shoreline correctly and closely matches the exact solution.

To investigate the accuracy of our method, we perform a convergence analysis of the L^2 error. The solution is initialized at $t = 0.1$ and filtering is only used in the semi-dry cell. The simulation is run until $t = 1$ with $\Delta t = 5 \cdot 10^{-5}$ for varying polynomial orders N and element widths h . The wet/dry tolerance φ_{tol} needs special attention as it can severely limit accuracy if it is set too high or make the scheme unstable if set too low. We found a reciprocal linear relation $\varphi_{\text{tol}} = 10^{-4}/N$ to give satisfying results for our convergence study. This relation is not necessarily optimal as we have observed much smaller tolerances to be possible for small mesh widths. Fig. 9 shows the results of the convergence analysis in which we compare the convergence of the L^2 error on the whole domain $\Omega = [-2, 2]$ to the convergence of the error in the smooth subdomain $[0, 1]$. This is an interesting test as the latter domain is initially dry but fully wet at $t = 1$. Moreover the solution is a third order polynomial, such that we can expect up to third order accuracy if we ignore the discontinuities in the wet/dry transitions. While the global convergence rates are limited due to the low regularity of the solution, we do observe up to third order convergence if we evaluate the error exclusively in the smooth part. The convergence rates are not optimal however, and cubic ansatz functions should theoretically be able to achieve machine precision accuracy in the smooth part if the scheme was exact. The latter is not possible in practice due to the presence of the wetting/drying process and the low regularity associated with it. The suboptimal convergence rates on the other hand are caused by the choice of flux discretization in the volume terms (19), which is not exact. Nonetheless, these results are encouraging as they demonstrate that high-order accuracy can be achieved in the presence of a wet/dry transition.

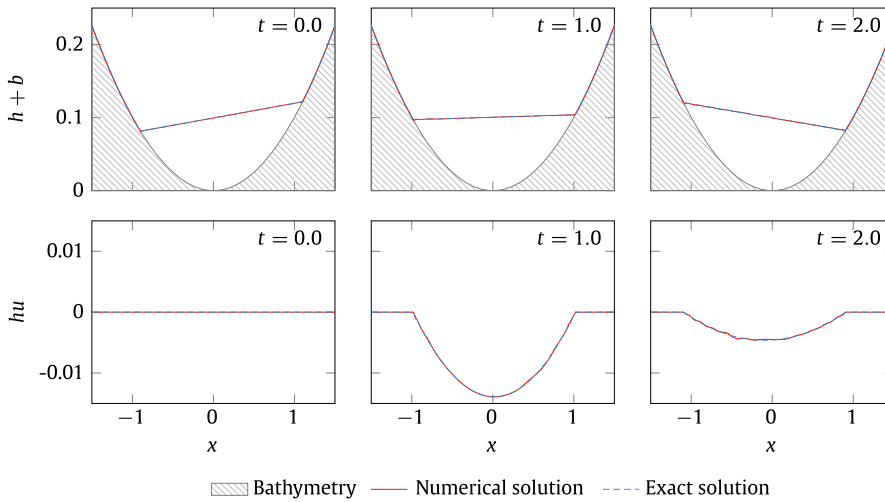


Fig. 10. Oscillating lake in a parabolic bed. Comparison of the exact solution to the numerical solution computed with $N = 2$ and $K = 100$ at different times.

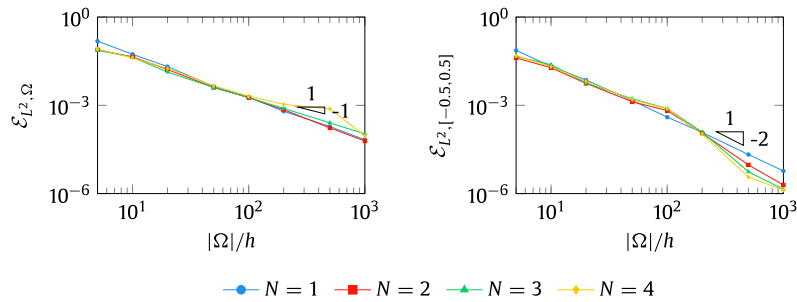


Fig. 11. Convergence for the oscillating lake in a parabolic channel. We compare the global L^2 error on the left to the error in the wet subdomain on the right.

5.4. Oscillating lake

Our final, one-dimensional test case is the oscillating planar surface

$$\varphi(x, t) = \max \left\{ \varphi_0 + 2\varphi_0\alpha \cos(\omega t) \left(x - \frac{\alpha}{2} \cos(\omega t) \right) - \varphi_0 x^2, 0 \right\},$$

$$(\varphi u)(x, t) = -\varphi(x, t)\alpha\omega \sin(\omega t),$$

on a parabolic bed $\tau(x) = \varphi_0 x^2$, where $\omega = \sqrt{2\varphi_0}$ is the frequency of the oscillation [35]. We choose the parameters $\varphi_0 = 0.1005g$, $\alpha = 0.1$ and set the domain to $[-1.5, 1.5]$. Boundary conditions can be chosen arbitrarily, as the two shores do not reach the boundaries of the domain. In this example, filtering is directly performed on the conserved variables \mathbf{q}_N as there is no meaningful water surface to reconstruct. Fig. 10 compares the numerical solution, obtained with $N = 2$, $K = 100$, $\Delta t = 1 \cdot 10^{-4}$ and $\varphi_{\text{tol}} = 10^{-4}$ to the analytical solution. We observe that wet/dry transitions are resolved properly and that the scheme remains stable even though the solution continuously undergoes both wetting and drying. Again, we perform a convergence analysis of the given problem, which can be seen in Fig. 11. To this end, we compare the relative L^2 error, evaluated on the whole domain, to the relative L^2 error on $[-0.5, 0.5]$. As in the dam break example, we observe that the convergence of the L^2 error on the entire domain is only slightly higher than 1. Again, this is because the global errors are dominated by the spatial approximation errors at the wet/dry interface, where one cannot expect more than first order accuracy. In the smooth part however, the solution lies in the ansatz space \mathbb{V} and high-order convergence can be achieved. The analytical solution is a quadratic polynomial in space and we observe convergence up to second order. After that point we cannot achieve errors of machine precision magnitude due to the presence of the wetting/drying process. This is consistent with the dam break example, where we have observed exactly the same behavior.

6. Results on the sphere

We now consider the surface of the rotating sphere. In all of the following simulations, we set the physical constants to Earth's values: $R = 6.37122 \cdot 10^6$ m, $g = 9.80616$ m/s² and $\omega = 7.29 \cdot 10^{-5}$ rad/s. The bottom topography $\tau = gb(\mathbf{x})$ is

Table 1Relative errors for the still water solution on the sphere at $t = 5$ d and $t = 10$ d with various polynomial orders.

N	L^2 error $\mathcal{E}_{L^2, \Omega}$		Mass error $\mathcal{E}_{\varphi, \Omega}$		Energy error $\mathcal{E}_{E, \Omega}$	
	$t = 5$ d	$t = 10$ d	$t = 5$ d	$t = 10$ d	$t = 5$ d	$t = 10$ d
1	$2.428 \cdot 10^{-15}$	$5.015 \cdot 10^{-15}$	0.0	0.0	0.0	0.0
2	$7.592 \cdot 10^{-13}$	$1.641 \cdot 10^{-12}$	$-3.324 \cdot 10^{-15}$	$6.980 \cdot 10^{-14}$	$-1.000 \cdot 10^{-13}$	$-1.239 \cdot 10^{-13}$
3	$8.388 \cdot 10^{-13}$	$1.828 \cdot 10^{-12}$	$2.350 \cdot 10^{-14}$	$2.085 \cdot 10^{-15}$	$-6.134 \cdot 10^{-14}$	$-1.023 \cdot 10^{-13}$
4	$6.957 \cdot 10^{-13}$	$2.858 \cdot 10^{-13}$	$-1.360 \cdot 10^{-14}$	$-5.247 \cdot 10^{-14}$	$-5.674 \cdot 10^{-14}$	$-7.318 \cdot 10^{-14}$

generated through piecewise linear interpolation of the ETOPO1 Earth Relief dataset [36]. Moreover, we set the water surface globally to $\varphi_0 = 0$ m relative to sea level, thus ignoring tidal effects. This approximation is acceptable as tidal effects are largely irrelevant for the propagation of tsunamis.

We use icosahedral meshes as described in [18] and refine the mesh uniformly until a prescribed refinement level of N_{uni} is reached. Then, we define a region of interest

$$\{\mathbf{x} \in S^2(R) \mid d_{S^2}(\mathbf{x}, \mathbf{x}_0) \leq \rho\},$$

where $d_{S^2}(\cdot, \mathbf{x}_0)$ is the great-circle distance with respect to a point \mathbf{x}_0 on the sphere and ρ is an angle which specifies the radius of the area. In this region, the grid is further refined locally until a desired refinement level of N_{loc} is reached. After static mesh refinement, the bottom topography data is re-read in order to obtain maximally accurate topography data. The filtering parameters and the dry tolerance are set to $s = 10$, $\alpha = 30$ and $\varphi_{\text{tol}} = g \cdot 10$ m in all following test cases.

6.1. Lake at rest solution

As a first test we seek to verify the theory developed in Sections 3 and 4 by verifying that the method is well-balanced on non-conforming, curved meshes. To do so, a non-conforming mesh is constructed by refining the icosahedral mesh in a circular region with radius $\rho = 40^\circ$ and center \mathbf{x}_0 at -10° longitude and -10° latitude. On this mesh, we initialize the “lake at rest” solution and run the simulation until $t = 10$ d is reached, which exceeds the timescale of tsunami events. We present our results obtained with various polynomial degrees at $t = 5$ d and $t = 10$ d in Table 1. For all the tested polynomial orders we can see that errors stay within the magnitude of accumulated roundoff errors. This confirms that the method is indeed able to preserve the water at rest solution on curved, non-conforming grids with bottom topography and dry cells in the domain.

6.2. Adaptive mesh refinement

Next, we would like to demonstrate that the methods developed for non-conforming discretizations allow the use of adaptive mesh refinement. We utilize the same mesh as for the “lake at rest” solution and generate a tsunami at 0° longitude and 0° latitude. For the initial shape we choose a Gaussian of the form

$$\varphi + \tau = \varphi_0 + \varphi_a \exp\left(- (d_{S^2}(\mathbf{x}, \mathbf{x}_0) / \rho_d)^2\right),$$

with a wave height of $\varphi_a = g \cdot 10$ m and a width of $\rho_d = 0.1$ rad. We utilize adaptive mesh refinement every 50 time steps using a velocity criterion. According to this criterion, refinement or coarsening is performed whenever the absolute velocity $\|\mathbf{u}\|_2$ exceeds or falls below the threshold of 0.01 m/s. Fig. 12 depicts the numerical solution at three different times. We observe that mesh refinement is handled without introducing any visible artifacts at the non-conforming interfaces. This allows for dynamically refined meshes, which allow an increased efficiency as computational effort can be directed towards regions, that require a higher resolution.

6.3. 2011 Tohoku tsunami event

In our last example, we validate the capability of the algorithm through a numerical simulation of the Tohoku tsunami event, which occurred off the Japanese coast in 2011. As an initial mesh, we use the unrefined icosahedral mesh and locally refine the area of interest in the Pacific Ocean. We select the point \mathbf{x}_0 at -177° longitude, 12° latitude and refine the initial grid within a radius of $\rho = 55^\circ$ until a refinement level of $N_{\text{loc}} = 5$ is reached. This amounts to a grid containing 14238 elements, most of which are located within the Pacific Ocean. The mesh is displayed in Fig. 13. Using this method, we can automatically generate meshes that are adapted to any region of interest and avoid boundary conditions at the cost of a few extra elements on the back side of the sphere.

The initial condition is generated using the Okada model [37] and the fault parameters of model III presented in [38]. We assume an instantaneous slip and calculate the final displacement of the bottom topography, reached 200 s after the earthquake has occurred. This bathymetry displacement is directly translated to a displacement of the initial water surface.

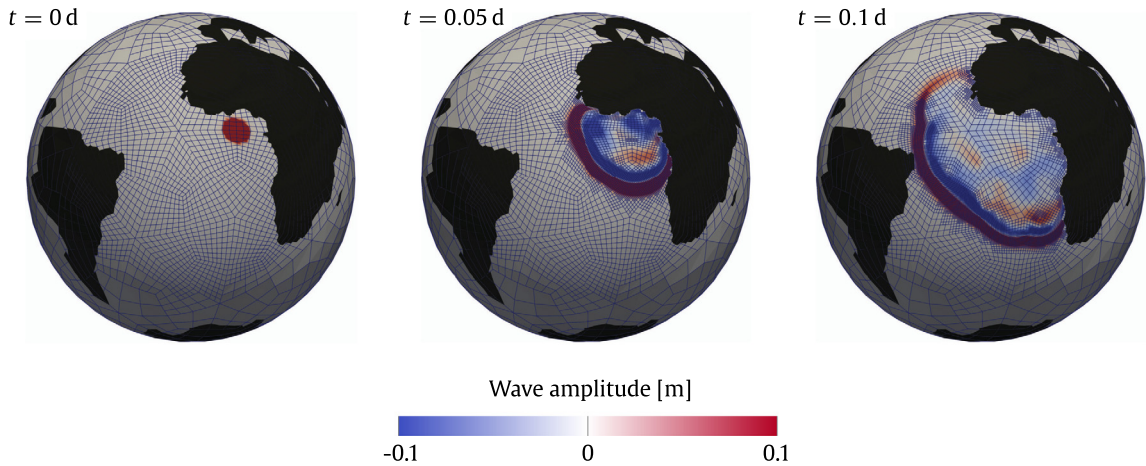


Fig. 12. Adaptive simulation of a solitary wave at 0° longitude and 0° latitude. The mesh is refined adaptively using our well-balanced mesh refinement.

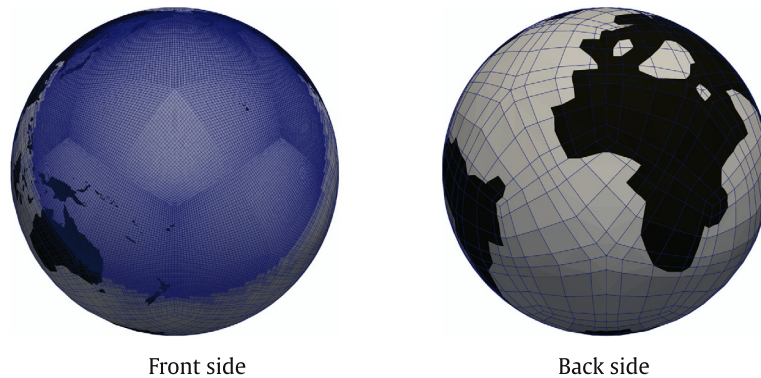


Fig. 13. Mesh for the Tohoku tsunami simulation using biquartic polynomials i.e. $N = 4$. The mesh contains 14238 elements with 25 collocation points in each element.

Starting 200 s after the initial earthquake at 2011-3-11 05:49:04 UTC, we simulate 12 h of tsunami propagation. The simulation is run twice; once with biquadratic polynomials $N = 2$ and a second time with biquartic polynomials $N = 4$. Running on a single core of an Intel Xeon E5-2643 v3 processor at 3.40 GHz, the simulations take 1 h 01 m and 12 h 36 m respectively. Fig. 14 depicts our results using biquartic polynomials. We conclude that our algorithm is capable of simulating 12 h of tsunami propagation in a stable manner. Moreover, visual comparison of the biquadratic method to the biquartic method show no significant differences and we expect our method to be consistent.

This is further reinforced by our comparison with buoy data. In Fig. 15 we show a comparison of real-world buoy data to wave amplitudes extracted from our numerical simulation at the 10 buoy positions marked in Fig. 14. The buoy data was collected by the DART tsunami monitoring network [22], which uses pressure data to infer the wave amplitude at the water surface. The oscillations recorded early on can be ignored as they are caused by seismic waves traveling through the Earth's crust. Moreover, we do not take tidal effects into account and compensate by subtracting the time-averaged water height from the time series of the buoy data. Both of our simulations are able to predict arrival times accurately over long distances. At longer propagation times, we observe a slight shift of both signals compared to the buoy data. This effect is caused by unresolved bathymetry and becomes noticeably smaller with the higher resolution simulation. Likewise, the time series of the buoy data is matched more accurately by the higher-order simulation and the gradual change in the time series suggests that the method is consistent. Table 2 summarizes the arrival times and states the absolute and relative error in the amplitudes of the initial waves for the biquartic solution. As we have already observed in the time series data, arrival times are quite accurate in the near field but start to diverge in the far field. The largest error, however, is smaller than 8 min, which is quite accurate considering the propagation time of more than 7 h. The errors in the predicted wave amplitudes do not exceed 0.3 m even though the waves are propagated over thousands of kilometers. We notice that for both arrival times and wave amplitudes, the largest errors occur in the signals of DART 52403 and DART 51407. As previously stated, this is most likely caused by underresolved bathymetry and uncertainties in the initial condition. DART 51407 is positioned close to the coast of Hawaii and the tsunami has to travel along the Hawaiian-Emperor seamount chain in order to reach it. Because the bathymetry is underresolved, small islands in the path of the tsunami are missing. Although these islands are relatively small, they would cause reflections, which can have a noticeable effect on the solution. Nevertheless, the time

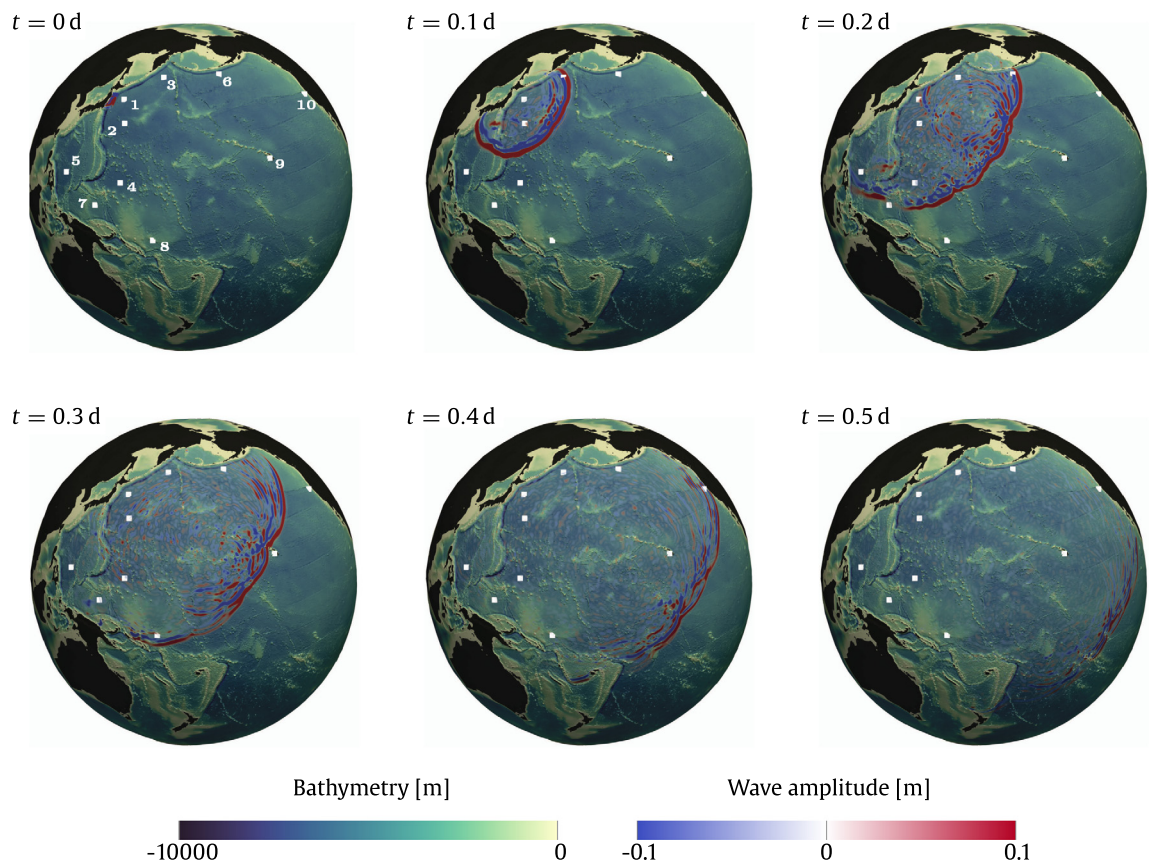


Fig. 14. High-fidelity simulation of the Tohoku tsunami event with $K = 14238$ and $N = 4$. The initial condition is taken to be a static initial water surface displacement from the “lake at rest” steady state. The buoys are numbered in ascending order with respect to the tsunami arrival times: 1 – DART 21418, 2 – DART 21413, 3 – DART 21416, 4 – DART 52402, 5 – DART 52405, 6 – DART 46408, 7 – DART 52403, 8 – DART 52406, 9 – DART 51407, 10 – DART 46411.

Table 2
Comparison of forecasted tsunami arrival times to recorded arrival times. Absolute and relative errors in the height of the initial tsunami wave are also stated. The results are extracted from the refined simulation using biquartic polynomials i.e. $N = 4$.

Buoy	Arrival time	Forecast	Amplitude error	Relative error
DART 21418	06:13:04	06:13:04	−0.279 m	−0.149
DART 21413	06:59:16	06:59:16	0.297 m	0.382
DART 21416	07:59:16	07:58:05	0.008 m	0.024
DART 52402	09:17:52	09:15:28	−0.079 m	−0.242
DART 52405	09:26:52	09:20:52	−0.016 m	−0.195
DART 46408	10:20:16	10:18:28	0.054 m	0.251
DART 52403	10:31:40	10:25:40	−0.062 m	−0.505
DART 52406	12:24:48	12:15:28	−0.033 m	−0.186
DART 51407	13:25:40	13:17:52	−0.129 m	−0.415
DART 46411	14:57:28	14:50:16	0.020 m	0.110

series data in Fig. 15 indicates that the method is able to capture the physical effects that are necessary to accurately predict the long-term evolution of tsunami waves.

We would like to point out that the proposed method can be adapted to allow well-balanced p -refinement. In this way, we can fully leverage the flexibility of hp -adaptivity that the discontinuous Galerkin method offers. This opens up the possibility of using elements with a low polynomial order in coastal areas, where h -refinement is preferable due to the restricted accuracy. High-order approximations, on the other hand, retain their advantages [1,39] in the wet parts of the domain, where the solutions are smooth and can be approximated well by polynomials.

7. Concluding remarks

We have proposed a discontinuous Galerkin method for the simulation of large-scale tsunami events on the surface of the rotating sphere. This required the development of a well-balanced, nodal discontinuous Galerkin method for the shallow

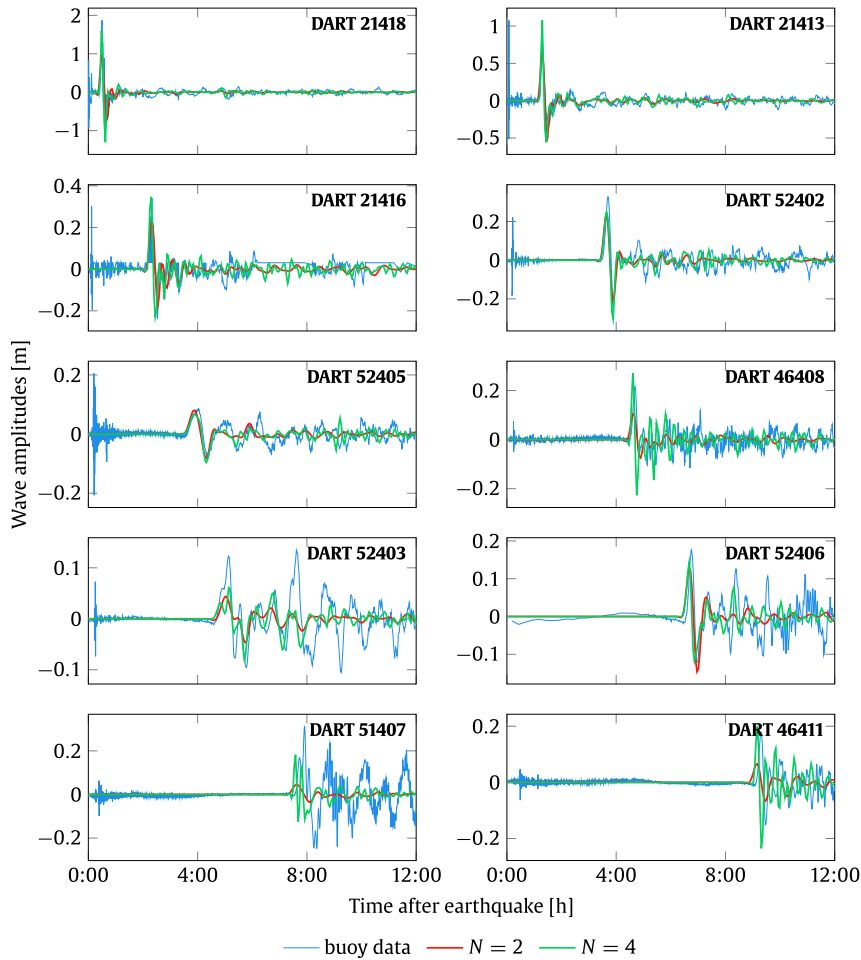


Fig. 15. Comparison of wave amplitudes extracted from DART buoy data to amplitudes extracted from the simulations.

water equations with wetting and drying on curved, non-conforming meshes. To achieve the well-balanced property, we have exploited the strong form of the discontinuous Galerkin formulation, which provides similar discretizations for source and flux terms. This makes the construction of a well-balanced scheme possible, without relying on integration by parts, which requires exact numerical integration. The strong form also provides a splitting of the well-balancing condition, which allows independent construction of well-balanced flux discretizations for non-conforming meshes and well-balanced volume discretizations to handle wetting/drying. When wet/dry transitions are introduced, special care is required to retain the well-balanced property, as partly dry elements would otherwise generate artificial waves at the shores. We have solved this issue by changing the differentiation operator in the partly dry cells to a local finite difference operator, which disregards the solution in the dry areas. Without adding any computational cost, we are implicitly treating the shoreline as a boundary of the physical domain, which makes it consistent with the original formulation of the shallow water equations. As such, the method is able to handle wetting/drying robustly at any polynomial order, while maintaining the well-balanced property.

The method is validated on a number of numerical tests. Using one-dimensional tests with exact solutions, we have studied the accuracy of our wetting/drying method. We could show that high-order convergence is achieved in the smooth parts of the solution even if the solution has less regularity at the shores. Moreover, we have demonstrated that it is well-balanced, even if dry areas are part of the computational domain. Ultimately, we show that our method provides an efficient and accurate way of simulating the shallow water equations on the rotating sphere. The accuracy of the method was demonstrated by performing simulations of the 2011 Tohoku tsunami and comparing the results to real-world buoy data. The inclusion of dry areas into the domain and non-conforming refinement allow for efficient generation of meshes based on the cubed sphere or the icosahedron. Boundary conditions are not required, which comes at the cost of a few extra elements lying in regions that are not of interest. The efficiency of the model was demonstrated by running a 12 h simulation of the Tohoku tsunami in either 1 h (using biquadratics) or 12 h (with biquartics), while producing accurate forecasts. Even after propagating the tsunami across the ocean over very long distances, we are able to obtain accurate predictions of arrival times and wave amplitudes. This indicates that the model captures all physical phenomena, relevant

to the long-time evolution of tsunamis. We believe that the wetting/drying method plays an important role, as it is able to correctly capture the interaction of waves with land masses.

The accuracy of our method makes it a suitable candidate for tsunami modeling in early warning systems and coastal risk assessment. The simple algorithmic structure and the parallelism of the discontinuous Galerkin method promise even faster tsunami simulations, which is crucial for the aforementioned applications. Beyond these applications, our method of splitting the well-balancing condition with the strong form can be extended to construct well-balanced schemes for other partial differential equations. Likewise, we can imagine our wetting/drying approach to be applicable to problems which involve free boundaries such as multi-phase flows. These situations are similar to wet/dry transitions and typically, one wants to avoid explicit tracking of the boundary. Our method offers a physically meaningful, yet implicit way of handling such boundaries, which could prove beneficial to these applications.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful and constructive comments, which helped us to improve our manuscript. F.X. Giraldo acknowledges support of the Office of Naval Research (PE-0602435N and previous support from the Computational Mathematics program).

Appendix A. Positive cell-averages

Proof. Under the assumption of exact numerical integration, we recover the evolution of cell averages by inserting $L_i = 1$ and the Euler timestep discretization in (18):

$$\int_D \mathbf{q}_N(\mathbf{x}, t_{n+1}) d\mathbf{x} = \int_D \mathbf{q}_N(\mathbf{x}, t_n) d\mathbf{x} + \Delta t \int_D \mathbf{S}_N d\mathbf{x} - \Delta t \oint_{\partial D} \hat{\mathbf{n}} \cdot \mathbf{F}_N^* d\mathbf{x}.$$

As we assume exact numerical integration, the strong form and the weak form are equivalent. We choose the weak form, which reduces the evolution of cell-averages to the boundary of the domain D . Then, by replacing the integrals with quadrature rules we get

$$\mathcal{Q}_D[\varphi_N(\mathbf{x}, t_{n+1})] = \mathcal{Q}_D[\varphi_N(\mathbf{x}, t_n)] - \Delta t \mathcal{Q}_{\partial D}[\hat{\mathbf{n}} \cdot \mathbf{F}_\varphi^*(\mathbf{q}_N^-(\mathbf{x}, t_n), \mathbf{q}_N^+(\mathbf{x}, t_n))]$$

for the cell-averaged water height. Here we have introduced \mathbf{F}_φ^* , which denotes the component of the numerical flux acting on the water height. We split the quadrature into sums over the edges and the sum over the interior points

$$\begin{aligned} \mathcal{Q}_D[\varphi_N(\mathbf{x}, t_{n+1})] &= \mathcal{Q}_D[\varphi_N(\mathbf{x}, t_n)] - \Delta t \mathcal{Q}_{\partial D}[\hat{\mathbf{n}} \cdot \mathbf{F}_\varphi^*(\mathbf{q}_N^-(\mathbf{x}, t_n), \mathbf{q}_N^+(\mathbf{x}, t_n))] \\ &= \sum_{i,j=2}^N \varphi_N(\xi_i, \eta_j, t_n) J_D(\xi_i, \eta_j) \omega_i \omega_j \\ &\quad + \sum_{i=1}^{N+1} \varphi_N(\xi_i, -1, t_n) J_D(\xi_i, -1) \gamma_i \omega_i \omega_1 - \Delta t \hat{\mathbf{n}} \cdot \mathbf{F}_\varphi^*(\mathbf{q}_N^-(\xi_i, -1, t_n), \mathbf{q}_N^+(\xi_i, -1, t_n)) J_{\partial D}(\xi_i, -1) \omega_i \\ &\quad + \sum_{i=1}^{N+1} \varphi_N(\xi_i, 1, t_n) J_D(\xi_i, 1) \gamma_i \omega_i \omega_N - \Delta t \hat{\mathbf{n}} \cdot \mathbf{F}_\varphi^*(\mathbf{q}_N^-(\xi_i, 1, t_n), \mathbf{q}_N^+(\xi_i, 1, t_n)) J_{\partial D}(\xi_i, 1) \omega_i \\ &\quad + \sum_{j=1}^{N+1} \varphi_N(-1, \eta_j, t_n) J_D(-1, \eta_j) \gamma_j \omega_1 \omega_j - \Delta t \hat{\mathbf{n}} \cdot \mathbf{F}_\varphi^*(\mathbf{q}_N^-(-1, \eta_j, t_n), \mathbf{q}_N^+(-1, \eta_j, t_n)) J_{\partial D}(-1, \eta_j) \omega_j \\ &\quad + \sum_{j=1}^{N+1} \varphi_N(1, \eta_j, t_n) J_D(1, \eta_j) \gamma_j \omega_N \omega_j - \Delta t \hat{\mathbf{n}} \cdot \mathbf{F}_\varphi^*(\mathbf{q}_N^-(1, \eta_j, t_n), \mathbf{q}_N^+(1, \eta_j, t_n)) J_{\partial D}(1, \eta_j) \omega_j \end{aligned}$$

where

$$\gamma_i = \begin{cases} \frac{1}{2} & \text{if } i = 1 \text{ or } i = N + 1, \\ 1 & \text{otherwise.} \end{cases}$$

As we know that the solution at t_n has a positive water height everywhere, the first term remains positive. Thus, it is sufficient to show that the four boundary sums stay positive. By inserting the Lax–Friedrichs flux (21), we can rewrite the boundary summands as

$$\varphi_N^- \left[J_D \gamma \omega_1 - \frac{1}{2} \Delta t J_{\partial D} (\hat{\mathbf{n}} \cdot \mathbf{u}_N^- + c) \right] + \varphi_N^+ \Delta t J_{\partial D} \left[\frac{c}{2} - \frac{1}{2} \hat{\mathbf{n}} \cdot \mathbf{u}_N^+ \right],$$

where we have dropped the indices for simplicity. Due to the definition of c , we have $\hat{\mathbf{n}} \cdot \mathbf{u}_N^\pm < c$. It follows that the second term is always positive and that the first term is positive under the condition

$$J_D \gamma \omega_1 \geq J_{\partial D} c \Delta t.$$

Rearranging and inserting the conservative value $\gamma = 1/2$ yields the desired result.

Appendix B. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.jcp.2018.02.008>.

References

- [1] J.S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods, Algorithms, Analysis, and Applications*, vol. 54, Springer Science & Business Media, New York, NY, 2008.
- [2] J.M. Greenberg, A.Y. Leroux, A well-balanced scheme for the numerical processing of source terms in hyperbolic equations, *SIAM J. Numer. Anal.* 33 (1) (2006) 1–16.
- [3] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, B. Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, *SIAM J. Sci. Comput.* 25 (6) (2004) 2050–2065.
- [4] A. Bermúdez, M.E. Vázquez, Upwind methods for hyperbolic conservation laws with source terms, *Comput. Fluids* 23 (1994) 1049–1071.
- [5] Y. Xing, C.-W. Shu, A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms, *Commun. Comput. Phys.* 1 (2006) 100–134.
- [6] S. Vater, N. Beisiegel, J. Behrens, A limiter-based well-balanced discontinuous Galerkin method for shallow-water flows with wetting and drying: one-dimensional case, *Adv. Water Resour.* 85 (2015) 1–13.
- [7] A. Meister, S. Ortleb, On unconditionally positive implicit time integration for the DG scheme applied to shallow water flows, *Int. J. Numer. Methods Fluids* 76 (2014) 69–94.
- [8] G. Kesserwani, Q. Liang, Well-balanced RKDG2 solutions to the shallow water equations over irregular domains with wetting and drying, *Comput. Fluids* 39 (2010) 2040–2050.
- [9] P. Chandrashekar, M. Zenk, Well-balanced nodal discontinuous Galerkin method for Euler equations with gravity, *J. Sci. Comput.* 71 (2017) 1062–1093.
- [10] N. Wintermeyer, A.R. Winters, G.J. Gassner, D.A. Kopriva, An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry, *J. Comput. Phys.* 340 (2017) 200–242.
- [11] O. Bokhove, Flooding and drying in discontinuous Galerkin finite-element discretizations of shallow-water equations. I. One dimension, *J. Sci. Comput.* 22/23 (2005) 47–82.
- [12] Y. Xing, X. Zhang, C.-W. Shu, Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations, *Adv. Water Resour.* 33 (2010) 1476–1493.
- [13] A. Ern, S. Piperno, K. Djadel, A well-balanced Runge–Kutta discontinuous Galerkin method for the shallow-water equations with flooding and drying, *Int. J. Numer. Methods Fluids* 58 (2008) 1–25.
- [14] Y. Xing, X. Zhang, Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes, *J. Sci. Comput.* 57 (2013) 19–41.
- [15] S. Bunya, E.J. Kubatko, J.J. Westerink, C. Dawson, A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations, *Comput. Methods Appl. Mech. Eng.* 198 (2009) 1548–1562.
- [16] B. van't Hof, E.A.H. Vollebregt, Modelling of wetting and drying of shallow water using artificial porosity, *Int. J. Numer. Methods Fluids* 48 (2005) 1199–1217.
- [17] T. Kärnä, B. de Brye, O. Gourgue, J. Lambrechts, R. Comblen, V. Legat, E. Deleersnijder, A fully implicit wetting–drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary, *Comput. Methods Appl. Mech. Eng.* 200 (2011) 509–524.
- [18] F.X. Giraldo, J.S. Hesthaven, T. Warburton, Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations, *J. Comput. Phys.* 181 (2002) 499–525.
- [19] R.D. Nair, S.J. Thomas, R.D. Loft, A discontinuous Galerkin transport scheme on the cubed sphere, *Mon. Weather Rev.* 133 (2005) 814–828.
- [20] S. Blaise, A. St-Cyr, A dynamic hp-adaptive discontinuous Galerkin method for shallow-water flows on the sphere with application to a global tsunami simulation, *Mon. Weather Rev.* 140 (2012) 978–996.
- [21] S. Blaise, A. St-Cyr, D. Mavriplis, B. Lockwood, Discontinuous Galerkin unsteady discrete adjoint method for real-time efficient tsunami simulations, *J. Comput. Phys.* 232 (2013) 416–430.
- [22] National Oceanic and Atmospheric Administration (2005), Deep-Ocean Assessment and Reporting of Tsunamis (DART(R)), National Centers for Environmental Information, NOAA, 2017.
- [23] R. Gandham, D. Medina, T. Warburton, GPU accelerated discontinuous Galerkin methods for shallow water equations, *Commun. Comput. Phys.* 18 (2015) 37–64.
- [24] A. Klöckner, T. Warburton, J. Bridge, J.S. Hesthaven, Nodal discontinuous Galerkin methods on graphics processors, *J. Comput. Phys.* 228 (2009) 7863–7882.
- [25] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, 2nd ed., Texts in Applied Mathematics, Springer Verlag, Berlin–Heidelberg, 2007.
- [26] S. Gottlieb, D. Ketcheson, C.-W. Shu, *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations*, World Scientific Publishing Co., 2010.
- [27] M.A. Kopera, F.X. Giraldo, Mass conservation of the unified continuous and discontinuous element-based Galerkin methods on dynamically adaptive grids with application to atmospheric simulations, *J. Comput. Phys.* 297 (2015) 90–103.
- [28] M.A. Kopera, F.X. Giraldo, Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations, *J. Comput. Phys.* 275 (2014) 92–117.
- [29] S. Marras, M.A. Kopera, F.X. Giraldo, Simulation of shallow-water jets with a unified element-based continuous/discontinuous Galerkin model with grid flexibility on the sphere, *Q. J. R. Meteorol. Soc.* 141 (2015) 1727–1739.
- [30] D.A. Kopriva, S.L. Woodruff, M.Y. Hussaini, Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method, *Int. J. Numer. Methods Eng.* 53 (2002) 105–122.

- [31] C. Sert, A. Beskok, Spectral element formulations on non-conforming grids: a comparative study of pointwise matching and integral projection methods, *J. Comput. Phys.* 211 (2006) 300–325.
- [32] D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, *J. Sci. Comput.* 26 (2006) 301–327.
- [33] S. Noelle, N. Pankratz, G. Puppo, J.R. Natvig, Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows, *J. Comput. Phys.* 213 (2006) 474–499.
- [34] O. Delestre, C. Lucas, P.-A. Ksinant, F. Darboux, C. Laguerre, T.N.T. Vo, F. James, S. Cordier, SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies, *Int. J. Numer. Methods Fluids* 72 (2013) 269–300.
- [35] W.C. Thacker, Some exact solutions to the nonlinear shallow-water wave equations, *J. Fluid Mech.* 107 (1981) 499–508.
- [36] C. Amante, B.W. Eakins, ETOPO1 1 Arc-Minute Global Relief Model, NOAA Technical Memorandum NESDIS NGDC-24, National Geophysical Data Center, NOAA, 2013.
- [37] Y. Okada, Surface deformation due to shear and tensile faults in a half-space, *Bull. Seismol. Soc. Am.* 75 (1985) 1135–1154.
- [38] G. Shao, X. Li, C. Ji, T. Maeda, Focal mechanism and slip history of the 2011 Mw 9.1 off the Pacific coast of Tohoku Earthquake, constrained with teleseismic body and surface waves, *Earth Planets Space* 63 (2011) 9–564.
- [39] D. Wirasaet, E.J. Kubatko, C.E. Michoski, S. Tanaka, J.J. Westerink, C. Dawson, Discontinuous Galerkin methods with nodal and hybrid modal/nodal triangular, quadrilateral, and polygonal elements for nonlinear shallow water flow, *Comput. Methods Appl. Mech. Eng.* 270 (2014) 113–149.