Computers and Fluids 288 (2025) 106493

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A high-order explicit Runge-Kutta approximation technique for the shallow water equations☆

Jean-Luc Guermond [a], Matthias Maier [a], Eric J. Tovar [b,*]

[a] Department of Mathematics, Texas A&M University 3368 TAMU, College Station, TX 77843, United States of America
[b] Theoretical Division, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM, 87545, United States of America

ARTICLE INFO

Dataset link: https://github.com/conservation-laws/ryujin

MSC:
65M60
65M12
35L50
35L65
76M10

Keywords:
Shallow water equations
Well-balanced
Invariant-domain-preserving
Explicit Runge–Kutta
High-order accuracy
Convex limiting
Positivity-preserving

ABSTRACT

We introduce a high-order space–time approximation of the Shallow Water Equations with sources that is invariant-domain preserving (IDP), well-balanced with respect to rest states, and employs a novel explicit Runge–Kutta (ERK) introduced in Ern and Guermond (SIAM J. Sci. Comput. 44(5), A3366–A3392, 2022) for systems of non-linear conservation equations. The resulting method is then numerically illustrated through verification and validation.

1. Introduction

Shallow water models are widely used for applications in coastal hydraulics, in-land flooding, and climate prediction. The development of efficient and accurate discretization techniques for the shallow water equations (and variations thereof) is therefore important. In addition to accuracy, robustness is also an important criterion. Here, we say that a method is robust if it can handle dry states and if it can preserve important equilibrium states which could be either the rest state [1,2] or time-independent solutions with nonzero velocity [3–5]. Numerical methods that preserve such equilibrium states are said to be well-balanced. The reader is referred to the book of [6] for a review on issues related to well-balancing. Finally, to be useful for practitioners, numerical methods for solving the shallow water equations must be versatile; in particular, one must be able to implement them on unstructured meshes. But, achieving robustness with respect to dry states, well-balancing, and high-order accuracy in space and time on unstructured meshes is challenging. The task becomes even more complex when external source terms besides topography are added.

Developing well-balanced methods that are robust with respect to dry states is an active topic of research; see [7–12]. Recent works on high-order schemes for the shallow water equations without external source terms that are well-balanced and robust with respect to dry states have been proposed in [13] for finite volumes and central weighted essentially non-oscillatory schemes on structured meshes and in [14] for continuous finite elements on unstructured meshes. The reader is also referred to [5,15–17] for other recent works that consider the inclusion of external source terms such as friction and rain effects. With the advancement of computing architectures, there has also been

☆ **Funding:** This material is based upon work supported in part by the National Science Foundation grants DMS-1619892 and DMS-2110868 (JLG), DMS-1912847 and DMS-2045636 (MM), by the Air Force Office of Scientific Research, under grant/contract number FA9550-23-1-0007 (JLG, MM), the Army Research Office, under grant number W911NF-19-1-0431 (JLG), the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contracts B640889, B641173 (JLG). ET acknowledges the support from the U.S. Department of Energy's Office of Applied Scientific Computing Research (ASCR) and Center for Nonlinear Studies (CNLS) at Los Alamos National Laboratory (LANL) under the Mark Kac Postdoctoral Fellowship in Applied Mathematics.

* Corresponding author.
E-mail address: ericjtovar@lanl.gov (E.J. Tovar).

https://doi.org/10.1016/j.compfluid.2024.106493
Received 6 April 2024; Received in revised form 29 October 2024; Accepted 19 November 2024
Available online 26 November 2024
0045-7930/Published by Elsevier Ltd.

some development on efficient implementation of numerical methods for the shallow water equations; see e.g., [18–21].

The goal of this work is to present an explicit approximation of the shallow water equations with topography and external sources that is well-balanced and high-order accurate in space and time. Our theoretical and algorithmic work is supplemented with a high performance implementation suitable for high fidelity simulations that is made freely available as part of the `ryujin` project [22,23].[1] The purpose of this work is to provide a stepping stone for various multi-physics extensions of the shallow water equations that require an implicit–explicit (IMEX) time discretization. Such variations include the Serre–Green–Naghdi Equations [24,25] for dispersive water waves and the coupling of the shallow water equations to subsurface models such as Richard's equation. The starting point for this work are the approximation techniques introduced in [17,26] for the shallow water equations. Unfortunately, the methodology discussed in [17,26] has two drawbacks: (i) the high-order spatial approximation is not fully well-balanced when a shoreline is present (the shoreline must coincide with the mesh for the method to maintain well-balancing). (ii) the time-discretization is limited in accuracy and efficiency due to the use of explicit strong stability preserving (SSP) explicit Runge–Kutta (ERK) methods which are known to be limited to fourth-order accuracy (see [27, Thm. 4.1]) and generally have an efficiency ratio significantly smaller than one [28, Def. 1.1]. Here, we provide solutions to these drawbacks. Specifically, we revisit the low-order method proposed in [26, §3] and construct a high-order version thereof that is unconditionally well-balanced with respect to the mesh geometry and therefore more robust with respect to dry states than the ones outlined in [26, §4] and [17, §5&6]. Compared to the previous works, the novelty of the discretization introduced in this work is fourfold: (i) we introduce modified auxiliary states (see Eq. (3.5)) that act as local Riemann averages for hydrostatic reconstructed left/right states; (ii) we rewrite the low-order method as a convex combination of these auxiliary states and external source terms (see Lemma 3.6); (iii) we introduce novel local bounds in space and time that control the magnitude of the velocity from above thereby avoiding blow-up at the shoreline and unnecessary time-step restrictions (see Lemma 3.13); (iv) we use ERK methods that have efficiency 1. These modifications allow the final limited update to be high-order accurate in space and time, invariant-domain preserving (IDP), and well-balanced with respect to rest states without any restrictions on the underlying mesh.

The paper is organized as follows. We briefly present the mathematical model and relevant properties in Section 2. In Section 3, we introduce a discretization-independent high-order spatial approximation to the shallow water equations with forward Euler time stepping and a convex limiting procedure. The main results of this section are Lemma 3.6, Propositions 3.9, 3.18 and 3.19. Then, using the convex limiting methodology for the high-order spatial discretization, we introduce in Section 4 a high-order in time invariant-domain preserving explicit Runge–Kutta method (IDP-ERK). Finally, we verify and validate the numerical method in Section 5. For the sake of completeness, we detail the implementation of boundary conditions for our method in Appendix.

2. The model problem

Let D be a polygonal domain in $\mathbb{R}^d$, $d \in \{1,2\}$, occupied by a body of water whose evolution in time under the action of gravity is modeled by the shallow water equations (also known as the Saint-Venant equations). Let $\boldsymbol{x} \in D$ be the position vector and $t > 0$ be the time variable. Let $\boldsymbol{u} := (\mathsf{h}, \boldsymbol{q})^\mathsf{T} \in \mathbb{R}^{d+1}$ be the dependent variable of the system where $\mathsf{h}(\boldsymbol{x},t)$ is the water depth and $\boldsymbol{q}(\boldsymbol{x},t) \in \mathbb{R}^d$ is the depth-averaged momentum vector of the fluid, also called discharge. Let $z(\boldsymbol{x})$ be the known topography mapping. We henceforth assume that z is in $W^{1,\infty}(D;\mathbb{R})$ to make sure that ∇z is a bounded function and thereby avoiding the need of properly defining $\mathsf{h}\nabla z$ when z is discontinuous; see Remark 2.5.

The goal of this work is to solve the following system of partial differential equations in the weak sense:

$$\begin{aligned} &\partial_t \boldsymbol{u} + \nabla\cdot\mathbb{f}(\boldsymbol{u}) = \mathbb{b}(\boldsymbol{u}, \nabla z(\boldsymbol{x})) \quad && \text{for æ } t > 0,\ \boldsymbol{x} \in D, \\ &\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}) && \text{for æ } \boldsymbol{x} \in D, \end{aligned} \tag{2.1}$$

with $\mathbb{f}(\boldsymbol{u}) := (\boldsymbol{q}, \boldsymbol{q}\otimes\boldsymbol{v} + \frac{1}{2}g\mathsf{h}^2\mathbb{I}_d)^\mathsf{T}$ and $\mathbb{b}(\boldsymbol{u}, \nabla z(\boldsymbol{x})) := (0, -g\mathsf{h}\nabla z)^\mathsf{T}$ where $\boldsymbol{v} := \mathsf{h}^{-1}\boldsymbol{q} \in \mathbb{R}^d$ is the (depth-averaged) velocity vector, g is the gravitational acceleration constant, and $\mathbb{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix. For the sake of completeness, we state a few properties regarding (2.1) that will be useful when constructing physically relevant approximations at the discrete level.

Definition 2.1 (*Invariant Set*)**.** The following convex domain is an *invariant set* (in the sense of [29, Def. 2.3]) for the shallow water Eqs. (2.1):

$$\mathcal{A} := \big\{\boldsymbol{u} = (\mathsf{h},\boldsymbol{q})^\mathsf{T} \in \mathbb{R}^{d+1} \mid \mathsf{h} > 0\big\}. \tag{2.2}$$

When the fluid is at rest, i.e., $\boldsymbol{q} \equiv \boldsymbol{0}$, the Shallow Water Equations (2.1) reduce to $g\mathsf{h}\nabla(\mathsf{h}+z) = \boldsymbol{0}$, which motivates to introduce the following terminology.

Definition 2.2 (*Problem at Rest*)**.** A solution $\boldsymbol{u}(\boldsymbol{x},t)$ to the shallow water equations is said to be *at rest* at time t if

$$\begin{aligned} &\boldsymbol{q}(x,t) = \boldsymbol{0} && \text{for æ } \boldsymbol{x} \in D, \quad \text{and} \\ &(\mathsf{h}+z)(\boldsymbol{x},t) = \text{const.} && \text{æ on connected components of } \big\{\, \boldsymbol{x} \in D \ : \\ & && \mathsf{h}(\boldsymbol{x},t) > 0 \,\big\}. \end{aligned}$$

Definition 2.3 (*Entropy Pairs and Entropy Solutions*)**.** The pair $\big(E(\boldsymbol{u}), \boldsymbol{F}(\boldsymbol{u})\big)$:

$$E(\boldsymbol{u}) := \tfrac{1}{2}g\mathsf{h}^2 + \tfrac{1}{2}\mathsf{h}\|\boldsymbol{v}\|^2 + gz\mathsf{h}, \boldsymbol{F}(\boldsymbol{u}) := \boldsymbol{v}\big(E(\boldsymbol{u}) + \tfrac{1}{2}g\mathsf{h}^2\big), \tag{2.3}$$

is an entropy pair for the shallow water equations (2.1) i.e., it satisfies $\nabla\cdot\boldsymbol{F}(\boldsymbol{u}) = \big(\nabla_{\boldsymbol{u}}E(\boldsymbol{u})\big)^\mathsf{T}(\nabla\cdot\mathbb{f}(\boldsymbol{u}) - \mathbb{b}(\boldsymbol{u},\nabla z(\boldsymbol{x})))$. We call $\boldsymbol{u}(\boldsymbol{x},t)$ an *entropy solution* to (2.1) if it is a weak solution to (2.1) and additionally satisfies the following inequality in the weak sense: $\partial_t E(\boldsymbol{u}) + \nabla\cdot\boldsymbol{F}(\boldsymbol{u}) \le 0$.

Remark 2.4 (*Entropy Pair for Flat Topography*)**.** When there is no influence due to topography ($z(\boldsymbol{x}) \equiv 0$), the entropy pair (2.3) simplifies to:

$$E_{\text{flat}}(\boldsymbol{u}) := \tfrac{1}{2}g\mathsf{h}^2 + \tfrac{1}{2}\mathsf{h}\|\boldsymbol{v}\|^2, \boldsymbol{F}_{\text{flat}}(\boldsymbol{u}) := \boldsymbol{v}\big(E_{\text{flat}}(\boldsymbol{u}) + \tfrac{1}{2}g\mathsf{h}^2\big). \tag{2.4}$$

and satisfies $\nabla\cdot\boldsymbol{F}_{\text{flat}}(\boldsymbol{u}) = \big(\nabla_{\boldsymbol{u}}E_{\text{flat}}(\boldsymbol{u})\big)^\mathsf{T}\nabla\cdot\mathbb{f}(\boldsymbol{u})$. □

For various applications, the system (2.1) is augmented with external source terms. Some examples include forcing due to bottom friction and source/sink of the water depth [5], Coriolis force [30] and many others. In this work, we only consider time-independent sources. It could be a mass source due for instance to rainfall, $R(\boldsymbol{x})$, and a discharge source due for instance to the Gauckler-Manning friction force $-gn^2\mathsf{h}^{-\frac{4}{3}}\boldsymbol{q}\|\boldsymbol{v}\|_{\ell^2}$. We set

$$\boldsymbol{S}(\boldsymbol{u}) = (R(\boldsymbol{x}), -gn^2\mathsf{h}^{-\frac{4}{3}}\boldsymbol{q}\|\boldsymbol{v}\|_{\ell^2})^\mathsf{T}, \tag{2.5}$$

where $R(\boldsymbol{x}) > 0$ and n is the Gauckler-Manning roughness coefficient.

Remark 2.5 (*Discontinuous Topography*)**.** The assumption $z \in W^{1,\infty}(D)$ is unrealistic for some applications, but giving a precise meaning to the product $\mathsf{h}\nabla z$ when both z and h are discontinuous at the same location is a nontrivial question. We refer the interested reader to

[1] https://github.com/conservation-laws/ryujin.

[31–33] and the references therein where this question addressed. Although some tests reported in the paper fall in this category, this important question is out of the scope of the paper. Note that, as we are using continuous finite elements to represent the topography (and the solution), the gradient of the approximate topography mapping is always well defined. □

3. Well-balanced forward Euler method

In this section, we introduce a forward Euler approximation to the shallow water equations that is high-order accurate in space, well-balanced and invariant-domain preserving. This section lays the foundation for the high-order explicit Runge–Kutta methodology introduced in Section 4.

3.1. Approximation details

Let $(0,T)$ be a chosen time interval for (2.1). Let $\{t^n\}_{n\in\{0:N\}}$ be a discretization of $(0,T)$ with the convention that $N\geq 1$, $t^0=0$, and $t^N=T$. The spatial approximation is discretization independent and can be either finite differences, finite volumes, continuous or discontinuous finite elements. Letting t^n be the current discrete time, we assume that the spatial approximation of $\boldsymbol{u}(\cdot,t^n)$ is entirely defined by the collection of states $\mathbf{U}^n := \{\mathbf{U}_i^n\}_{i\in\mathcal{V}}$, where $\mathcal{V} := \{1:I\}$ is the index set for the spatial degrees of freedom and $I := \mathrm{card}(\mathcal{V})$. We also set $\mathbf{U}_i^n := (\mathsf{H}_i^n, \mathbf{Q}_i^n)^\intercal \in \mathbb{R}^{d+1}$. Here, H_i^n and $\mathbf{Q}_i^n$ represent approximations of the water depth and discharge associated with the ith degree of freedom at time t^n. To be able to refer to the water depth and the discharge of an arbitrary state $\tilde{\mathbf{U}}=(\tilde{\mathsf{H}},\tilde{\mathbf{Q}})$ in $\mathbb{R}^{d+1}$, we introduce the linear mappings $\mathsf{H}:\mathbb{R}^{d+1}\to\mathbb{R}$ and $\mathbf{Q}:\mathbb{R}^{d+1}\to\mathbb{R}^d$ so that $\mathsf{H}(\tilde{\mathbf{U}})=\tilde{\mathsf{H}}$ and $\mathbf{Q}(\tilde{\mathbf{U}})=\tilde{\mathbf{Q}}$. We assume that the topography mapping is approximated by the collection of states: $\mathsf{Z} := \{\mathsf{Z}_i\}_{i\in\mathcal{V}}$. We assume that for every $i\in\mathcal{V}$, there exists a subset $\mathcal{J}(i)\subsetneq\mathcal{V}$ that collects the local degrees of freedom that interact with i, which we call stencil at i. Let $\mathcal{J}^*(i) := \mathcal{J}(i)\backslash\{i\}$. We assume that the underlying spatial discretization provides the following three quantities for all $i\in\mathcal{V}$ and all $j\in\mathcal{J}(i)$:

(i) An invertible low-order mass matrix $\mathbb{M}^{\mathrm{L}}_{ij}=m_i\delta_{ij}$ where $m_i>0$ is called the mass associated with the ith degree of freedom;
(ii) An invertible, symmetric high-order matrix with entries $\mathbb{M}^{\mathrm{H}}_{ij}=m_{ij}$ such that: $(\mathbb{M}^{\mathrm{H}}\mathsf{X})_i=\sum_{j\in\mathcal{J}(i)}m_{ij}\mathsf{X}_j$ for all $\mathsf{X}\in\mathbb{R}^I$;
(iii) A vector $\boldsymbol{c}_{ij}\in\mathbb{R}^d$ that approximates the gradient operator on average: i.e., there exists a shape function φ_i so that $\int_D\varphi_i(\boldsymbol{x})\nabla X(\boldsymbol{x})\,\mathrm{d}x\approx\sum_{j\in\mathcal{J}(i)}\mathsf{X}_j\boldsymbol{c}_{ij}$, where $\{\mathsf{X}_j\}_{j\in\mathcal{J}(i)}$ are pointwise approximations of $\boldsymbol{X}$ over the stencil $\mathcal{J}(i)$.

The local stencil $\mathcal{J}(i)$ satisfies $(j\notin\mathcal{J}(i))\implies(\boldsymbol{c}_{ij}=\mathbf{0}$ and $m_{ij}=0)$. We further assume that $\boldsymbol{c}_{ij}=-\boldsymbol{c}_{ji}$ whenever i or j is not a boundary degree of freedom, and $\sum_{j\in\mathcal{J}(i)}\boldsymbol{c}_{ij}=0$ which is necessary for mass conservation. The high-order mass matrix is related to the low-order mass matrix through the relation $m_i=\sum_{j\in\mathcal{J}(i)}m_{ij}$ to guarantee that $\mathbb{M}^{\mathrm{L}}$ and $\mathbb{M}^{\mathrm{H}}$ carry the same mass: $\sum_{i\in\mathcal{V}}m_i\mathbf{U}_i^n=\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{J}(i)}m_{ij}\mathbf{U}_j^n$. Examples of discretization techniques satisfying the above assumptions are described in [34].

3.2. Well-balancing preliminaries

Before introducing the low- and high-order spatial approximations, we first define the discrete velocity and what it means to be well-balanced with respect to rest states. Given a state $(\mathsf{H}_i^n,\mathbf{Q}_i^n)^\intercal$ with nonzero water depth, $\mathsf{H}_i^n>0$, the velocity is defined to be the ratio $\mathbf{Q}_i^n/\mathsf{H}_i^n$. To be robust with respect to dry states, we adopt the regularization technique from [10] that avoids the division by zero when $\mathsf{H}_i^n\to 0$. For this purpose, we introduce a small dimensionless parameter ϵ and a characteristic length scale $\mathsf{h}_{\max}$ that scales like the average of the water depth in the problem, and we set

$$\mathbf{V}_i^n := \frac{2\mathsf{H}_i^n}{(\mathsf{H}_i^n)^2+\max(\mathsf{H}_i^n,\epsilon\mathsf{h}_{\max})^2}\mathbf{Q}_i^n. \tag{3.1}$$

Notice that $\mathbf{V}_i^n=\mathbf{Q}_i^n/\mathsf{H}_i^n$ when $\mathsf{H}_i^n\geq\epsilon\mathsf{h}_{\max}$; that is, the regularization is active only when $\mathsf{H}_i^n\leq\epsilon\mathsf{h}_{\max}$. All the numerical simulations reported in the paper are done with $\epsilon=10^{-12}$ and using double precision floating point arithmetic. The well-balancing techniques in this work adopt the methodology proposed in [7,35] known as of the hydrostatic reconstruction water depth.

Definition 3.1 (*Hydrostatic Reconstruction*)**.** Let $i\in\mathcal{V}$ and $j\in\mathcal{J}(i)$. Let $\mathbf{U}_i$ be a state with water depth $\mathsf{H}_i>0$. The *hydrostatic reconstruction* between i and j of H_i and $\mathbf{U}_i$ are defined as follows:

$$\mathsf{H}_i^{j,*} := \max(0,\mathsf{H}_i+\mathsf{Z}_i-\max(\mathsf{Z}_i,\mathsf{Z}_j)), \tag{3.2a}$$

$$\mathbf{U}_i^{j,*} := \begin{pmatrix}\mathsf{H}_i^{*,j}\\ \mathbf{V}_i\mathsf{H}_i^{*,j}\end{pmatrix}. \tag{3.2b}$$

Definition 3.2 (*Discrete States at Rest*)**.** A given set of discrete numerical states $\{(\mathsf{H}_i^n,\mathbf{Q}_i^n)\}_{i\in\mathcal{V}}$ is said *to be at rest* if the approximate momentum $\mathbf{Q}_i^n$ is zero for all $i\in\mathcal{V}$, and if the approximate water depth H_i^n and the approximate bathymetry map Z_i satisfy the following property for all $i\in\mathcal{V}$: $\mathsf{H}_i^{n,j,*}=\mathsf{H}_j^{n,i,*}$ for all $j\in\mathcal{J}(i)$.

Remark 3.3. Note that we have adopted the condition $\mathsf{H}_i^{n,j,*}=\mathsf{H}_j^{n,i,*}$ to be the discrete analog to $(\mathsf{h}+z)(\boldsymbol{x},t)=const.$ in Definition 2.2 instead of the natural looking identity $\mathsf{H}_i^n+\mathsf{Z}_i=\mathsf{H}_j+\mathsf{Z}_j$. The reason behind this choice is the fact that Definition 3.2 does not need to distinguish between dry and wet states. For example, assume that $\mathsf{H}_j=0$ and $\mathsf{Z}_i>\mathsf{Z}_j$ for all $j\in\mathcal{J}(i)$. Then, $\mathsf{H}_i^{n,j,*}=\max(0,0+\mathsf{Z}_i-\mathsf{Z}_i)=0$ and $\mathsf{H}_j^{n,i,*}=\max(0,0+\mathsf{Z}_j-\mathsf{Z}_i)=0$, which gives $\mathsf{H}_i^{n,j,*}=\mathsf{H}_j^{n,i,*}$. However, the condition $\mathsf{H}_i^n+\mathsf{Z}_i=\mathsf{H}_j+\mathsf{Z}_j$ breaks down in this case since $\mathsf{H}_j=0$ for all $j\in\mathcal{J}(i)$ would imply that the topography mapping should be constant which is not the case. □

Finally, the following result, established in [26, Lem. 3.1&3.3], explains why the hydrostatic reconstruction is useful.

Lemma 3.4. *Assume that the approximation is done with Lagrange finite elements with global shape functions $\{\varphi_i\}_{i\in\mathcal{V}}$. Let $\mathrm{g}(\boldsymbol{u}) := (\boldsymbol{q},\boldsymbol{q}\otimes\boldsymbol{v})^T$ be the gas dynamics flux.* (i) *The expression $\sum_{j\in\mathcal{J}(i)}\frac{1}{2}((H_j^{*,i})^2-(H_i^{*,j})^2)\boldsymbol{c}_{ij}$ is a first-order approximation of the flux $\int_{\mathcal{J}(i)}(\nabla(\frac{1}{2}h^2)+h\nabla z)\varphi_i\,dx$.* (ii) *The quantity $\sum_{j\in\mathbb{I}(i)}(\mathrm{g}(\mathbf{U}_j^{*,i})+\mathrm{g}(\mathbf{U}_i^{*,j}))\cdot\boldsymbol{c}_{ij}$ is a first-order approximation of $\int_i\nabla\cdot\mathrm{g}(\boldsymbol{u})\varphi_i\,dx$ away from the shoreline if the mesh is centro-symmetric.* (iii) *Assume the discrete numerical states $\{(H_i^n,\mathbf{Q}_i^n)\}_{i\in\mathcal{V}}$ are at rest, then $\sum_{j\in\mathbb{I}(i)}(\mathrm{g}(\mathbf{U}_j^{*,i})+\mathrm{g}(\mathbf{U}_i^{*,j}))\cdot\boldsymbol{c}_{ij}+\frac{1}{2}(0,(H_j^{*,i})^2-(H_i^{*,j})^2)\boldsymbol{c}_{ij})^T=\mathbf{0}$ for all $i\in\mathcal{V}$.*

3.3. Low-order spatial approximation

We now discuss a low-order method that will serve as safeguard to the high-order method. We essentially follow [26, Sec. 3] which introduced a formally first-order consistent approximation of the shallow water equations when the spatial discretization consists of continuous, linear finite elements. But departing from [26] we introduce a different definition of the auxiliary states (see (3.5) below) and introduce a different convex combination of these auxiliary states to reconstruct the low order update; see Lemma 3.6.

Let t^n be the current time and let $\tau := t^{n+1}-t^n$ denote the current time step. Recalling [26, Eq. (3.3)], the low-order approximation with forward Euler time-stepping is then given by:

$$\frac{m_i}{\tau}(\mathbf{U}_i^{\mathrm{L},n+1}-\mathbf{U}_i^n)=\sum_{j\in\mathcal{G}(i)}\mathbf{F}_{ij}^{\mathrm{L},n},\tag{3.3a}$$

$$\begin{aligned}\mathbf{F}_{ij}^{\mathrm{L},n}:=\quad&-\left(\mathbf{U}_j^{n,i,*}\otimes\mathbf{V}_j^n+\mathbf{U}_i^{n,j,*}\otimes\mathbf{V}_i^n\right)\boldsymbol{c}_{ij}+d_{ij}^{\mathrm{L},n}\left(\mathbf{U}_j^{n,i,*}-\mathbf{U}_i^{n,j,*}\right)\\&-\begin{pmatrix}0\\ g\boldsymbol{c}_{ij}\left(\frac{1}{2}(\mathsf{H}_j^{n,i,*})^2-\frac{1}{2}(\mathsf{H}_i^{n,j,*})^2+(\mathsf{H}_i^n)^2\right)\end{pmatrix}.\end{aligned}\tag{3.3b}$$

Here, $d_{ij}^{\mathrm{L},n}\geq 0$ is the graph-viscosity coefficient that makes the method invariant-domain preserving:

$$d_{ij}^{\mathrm{L},n}:=\max\Big(\lambda_{\max}(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*},\boldsymbol{n}_{ij})\|\boldsymbol{c}_{ij}\|_{\ell^2},\lambda_{\max}(\mathbf{U}_j^{n,i,*},\mathbf{U}_i^{n,j,*},\boldsymbol{n}_{ji})\|\boldsymbol{c}_{ji}\|_{\ell^2},\Big),\tag{3.4}$$

where $\boldsymbol{n}_{ij}:=\boldsymbol{c}_{ij}/\|\boldsymbol{c}_{ij}\|_{\ell^2}$ and $\lambda_{\max}(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*},\boldsymbol{n}_{ij})$ is a guaranteed upper bound on the maximum wave speed in the local Riemann problem with left and right states $(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*})$ and flux $\mathbb{f}(\cdot)\boldsymbol{n}_{ij}$. Analytical expressions for $\lambda_{\max}$ are given in [26, Lem. 3.8]. Note that $d_{ij}^{\mathrm{L},n}=d_{ji}^{\mathrm{L},n}$ which is necessary for mass conservation. By convention, we set $d_{ii}^{\mathrm{L},n}:=-\sum_{j\in\mathcal{G}^*(i)}d_{ij}^{\mathrm{L},n}$. Observe that $\mathbf{F}_{ij}^{\mathrm{L},n}=-\mathbf{F}_{ji}^{\mathrm{L},n}$ when the topography is flat.

Lemma 3.5 (*Well-balancing and Conservation*). *The scheme $\mathbf{U}^n\mapsto\mathbf{U}^{\mathrm{L},n+1}$ defined in (3.3) is mass-conservative and well-balanced.*

Proof. See [26, Prop. 3.9]. □

We now introduce auxiliary states that are meant to be thought of as averages of the self-similar solution to the local Riemann problem for the pair $(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*})$ in the direction $\boldsymbol{n}_{ij}$. We do this to rewrite the low-order scheme (3.3) as a convex combination of these auxiliary states and source terms to extract local bounds in space and time. These bounds are needed for the convex limiting procedure described in Section 3.5. The importance of such auxiliary states for nonlinear conservation laws has been established in the literature and we refer the reader to [36,37] and the references therein.

Let $i\in\mathcal{V}$. For every $j\in\mathcal{G}(i)$, we define the following auxiliary states for the pair (i,j) as follows:

$$\overline{\mathbf{U}}_{ij}^n:=\frac{1}{2}\{\mathbf{U}_i^{n,j,*}+\mathbf{U}_j^{n,i,*}\}-\frac{1}{2d_{ij}^{\mathrm{L},n}}\{\mathbb{f}(\mathbf{U}_j^{n,i,*})-\mathbb{f}(\mathbf{U}_i^{n,j,*})\}\boldsymbol{c}_{ij},\tag{3.5}$$

with the convention that $\overline{\mathbf{U}}_{ii}^n:=\mathbf{U}_i^n$. We recall that $\overline{\mathbf{U}}_{ij}^n$ coincides with the exact space average over $[-\frac{1}{2},\frac{1}{2}]$ at time $\frac{\|\boldsymbol{c}_{ij}\|_{\ell^2}}{2d_{ij}^{\mathrm{L},n}}$ of the solution to the Riemann problem with left and right states $(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*})$ and with flux $\mathbb{f}(v)\boldsymbol{n}_{ij}$ provided the viscosity $d_{ij}^{\mathrm{L},n}$ is large enough so that $\lambda_{\max}(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*},\boldsymbol{n}_{ij})\|\boldsymbol{c}_{ij}\|_{\ell^2},\leq d_{ij}^{\mathrm{L},n}$. Note that the bar states here differ from those in [26, Prop. 3.11]. We also define an affine shift needed for the convex combination update:

$$\mathbf{B}_i^{\mathrm{L},n}:=\sum_{j\in\mathcal{G}(i)}\mathbf{B}_{ij}^{\mathrm{L},n},\qquad\mathbf{B}_{ij}^{\mathrm{L},n}:=-2(d_{ij}^{\mathrm{L},n}+\mathbf{V}_i^n\cdot\boldsymbol{c}_{ij})(\mathbf{U}_i^{n,j,*}-\mathbf{U}_i^n).\tag{3.6}$$

Lemma 3.6 (*Stability*). *Assume that $\mathbf{U}_i^n\in\mathcal{A}$ for all $i\in\mathcal{V}$. Assume also that the time step satisfies the restriction $\tau\leq\min_{i\in\mathcal{V}}\frac{m_i}{2|d_{ii}^{\mathrm{L},n}|}$. Then,*

(i) *The following convex combination holds:*

$$\mathbf{U}_i^{\mathrm{L},n+1}=\Big(1+\frac{2\tau d_{ii}^{\mathrm{L},n}}{m_i}\Big)(\mathbf{U}_i^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n})+\sum_{j\in\mathcal{G}^*(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}(\overline{\mathbf{U}}_{ij}^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}).\tag{3.7}$$

(ii) *The water depth of $\mathbf{U}^{\mathrm{L},n+1}$ is positive, i.e., $\mathbf{U}_i^{\mathrm{L},n+1}\in\mathcal{A}$ for all $i\in\mathcal{V}$.*

Proof. In order to show 3.3 we add and subtract $\mathbf{B}_i^{\mathrm{L},n}$ in (3.3) and then rearrange:

$$\frac{m_i}{\tau}(\mathbf{U}_i^{\mathrm{L},n+1}-\mathbf{U}_i^n)=\mathbf{B}_i^{\mathrm{L},n}+\sum_{j\in\mathcal{G}(i)}\left[\mathbf{F}_{ij}^{\mathrm{L},n}-\mathbf{B}_{ij}^{\mathrm{L},n}\right].$$

Recalling that $\mathbb{f}(\boldsymbol{u}):=(\boldsymbol{q},\boldsymbol{q}\otimes\boldsymbol{v}+\frac{1}{2}g\mathsf{h}^2\mathbb{I}_d)^{\mathsf{T}}$ and $\sum_{j\in\mathcal{G}(i)}\boldsymbol{c}_{ij}=\mathbf{0}$, we have

$$\begin{aligned}\sum_{j\in\mathcal{G}(i)}\mathbf{F}_{ij}^{\mathrm{L},n}-\mathbf{B}_{ij}^{\mathrm{L},n}&=\sum_{j\in\mathbb{I}(i)}-\{\mathbb{f}(\mathbf{U}_j^{n,i,*})-\mathbb{f}(\mathbf{U}_i^{n,j,*})\}\boldsymbol{c}_{ij}+d_{ij}^{\mathrm{L},n}\{\mathbf{U}_i^{n,j,*}+\mathbf{U}_j^{n,i,*}\},\\&=\sum_{j\in\mathcal{G}(i)}2d_{ij}^{\mathrm{L},n}\overline{\mathbf{U}}_{ij}^n.\end{aligned}$$

Then, recalling that $\overline{\mathbf{U}}_{ii}^n=\mathbf{U}_i^n$, and $\sum_{j\in\mathcal{G}(i)}d_{ij}^{\mathrm{L},n}=0$ holds true by virtue of the definition of $d_{ii}^{\mathrm{L},n}$, we infer that

$$\begin{aligned}\mathbf{U}_i^{\mathrm{L},n+1}&=\mathbf{U}_i^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}+\sum_{j\in\mathcal{G}(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}\overline{\mathbf{U}}_{ij}^n,\\&=\mathbf{U}_i^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}+\frac{2\tau d_{ii}^{\mathrm{L},n}}{m_i}\mathbf{U}_i^n+\sum_{j\in\mathcal{G}^*(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}\overline{\mathbf{U}}_{ij}^n\\&\quad+\underbrace{\sum_{j\in\mathcal{G}(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}(\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n})}_{=0},\\&=\mathbf{U}_i^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}+\frac{2\tau d_{ii}^{\mathrm{L},n}}{m_i}(\mathbf{U}_i^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n})+\sum_{j\in\mathcal{G}^*(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}\\&\quad\times(\overline{\mathbf{U}}_{ij}^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}),\\&=\Big(1+\frac{2\tau d_{ii}^{\mathrm{L},n}}{m_i}\Big)(\mathbf{U}_i^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n})+\sum_{j\in\mathcal{G}^*(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}(\overline{\mathbf{U}}_{ij}^n+\frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}).\end{aligned}$$

The above decomposition is a genuine convex combination under the CFL condition $1+\frac{2\tau|d_{ii}^{\mathrm{L},n}|}{m_i}\geq 0$.

For 3.3 we recall that the water depth of the sate $\mathbf{B}_i^{\mathrm{L},n}$ is given by $\mathsf{H}(\mathbf{B}_i^{\mathrm{L},n}):=\sum_{j\in\mathcal{G}(i)}2(d_{ij}^{\mathrm{L},n}+\mathbf{V}_i^n\cdot\boldsymbol{c}_{ij})(\mathsf{H}_i^n-\mathsf{H}_i^{n,j,*})$.

It is established in [26, Prop. 3.7] that $d_{ij}^{\mathrm{L},n}+\mathbf{V}_i^n\cdot\boldsymbol{c}_{ij}\geq 0$ and by definition we have $(\mathsf{H}_i^n-\mathsf{H}_i^{n,j,*})\geq 0$. Hence, $\mathsf{H}(\mathbf{B}_i^{\mathrm{L},n})>0$. Moreover, the definition of $d_{ij}^{\mathrm{L},n}$ implies that $\mathsf{H}(\overline{\mathbf{U}}_{ij}^n)\geq 0$ for all $j\neq i\in\mathcal{G}^*(i)$. The assertion now follows directly from the combination (3.7) which is convex provided the CFL time step restriction holds true. □

Remark 3.7 (*Source Terms*). In order to incorporate external source terms into the low-order scheme, we augment (3.3a) with a suitable low-order approximation of $\boldsymbol{S}(\boldsymbol{u})$ given by (2.5):

$$\frac{m_i}{\tau}(\mathbf{U}_i^{\mathrm{L},n+1}-\mathbf{U}_i^n)=\sum_{j\in\mathcal{G}(i)}\mathbf{F}_{ij}^{\mathrm{L},n}+m_i\boldsymbol{S}_i^n,\quad\text{with}\tag{3.8}$$

$$\boldsymbol{S}_i^n:=\left(R(\boldsymbol{a}_i),-gn^2(\mathscr{H}_i^n)^{-1}\boldsymbol{Q}_i^n\|\mathbf{V}_i\|_{\ell^2}\right)^{\mathsf{T}},\tag{3.9}$$

$$\text{where }\mathscr{H}_i^n:=\frac{1}{2}\left[(\mathsf{H}_i^n)^{4/3}+\max((\mathsf{H}_i^n)^{4/3},2gn^2\tau\|\mathbf{V}_i^n\|_{\ell^2})\right].$$

The expression $\mathscr{H}_i^n$ is introduced for regularizing the term $\mathsf{h}^{-4/3}$ as in [17, Eq. (3.3)]. Here, $\boldsymbol{a}_i$ denotes a collocation point associated with the ith degree of freedom. Note that the results in Lemma 3.6 still hold, provided that one replaces $\mathbf{B}_i^{\mathrm{L},n}$ by a modified affine shift $\mathbf{B}_i^{\mathrm{L},n}+m_i\boldsymbol{S}_i^n$ throughout. □

Lemma 3.8. *Consider the entropy pair $(E_{\mathrm{flat}},\boldsymbol{F}_{\mathrm{flat}})$ defined in (2.4). Let $\overline{\mathbf{U}}_{ij}^n$ be given by (3.5) with $d_{ij}^{\mathrm{L},n}$ defined in (3.4). Then,*

$$E_{\mathrm{flat}}(\overline{\mathbf{U}}_{ij})\leq\frac{1}{2}\Big(E_{\mathrm{flat}}(\mathbf{U}_i^{n,j,*})+E_{\mathrm{flat}}(\mathbf{U}_j^{n,i,*})\Big)\tag{3.10}$$

$$- \frac{1}{2d_{ij}^{\mathrm{L},n}}\Big(\boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_j^{n,i,*}) - \boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_i^{n,j,*})\Big)\boldsymbol{c}_{ij}.$$

Proof. Let $\boldsymbol{n}$ a unit vector in $\mathbb{R}^d$. Let $\boldsymbol{v}(\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)(x,t)$ be the solution to the Riemann problem with flux $\mathbb{f}(\cdot)\boldsymbol{n}$ and with left and right states, $\boldsymbol{v}_L$ and $\boldsymbol{v}_R$, respectively. Let $\overline{\boldsymbol{v}}(t,\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R) := \int_{-1/2}^{1/2}\boldsymbol{v}(\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)(x,t)\,\mathrm{d}x$ be the average of the Riemann solution at time $t>0$. Assume that $t\lambda_{\max}(\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R) \le \frac{1}{2}$. Then, invoking [29, Lem. 2.2], we have that $\eta(\overline{\boldsymbol{v}}(t,\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)) \le \frac{1}{2}\left(\eta(\boldsymbol{v}_L)+\eta(\boldsymbol{v})_R\right) - t\left(\boldsymbol{q}(\boldsymbol{v}_R)\cdot\boldsymbol{n} - \boldsymbol{q}(\boldsymbol{v}_L)\cdot\boldsymbol{n}\right)$ for every entropy pair $(\eta,\boldsymbol{q})$ of the flux $\mathbb{f}$.

Let $\overline{\mathbf{U}}_{ij}^n$ be defined in (3.5) with $d_{ij}^{\mathrm{L},n}$ defined in (3.4). Consider the Riemann problem with flux $\mathbb{f}(\cdot)\boldsymbol{n}_{ij}$, $\boldsymbol{n}_{ij} := \frac{\boldsymbol{c}_{ij}}{\|\boldsymbol{c}_{ij}\|_{\ell^2}}$, and left and right states $(\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*})$. Let us set $t_{ij} := \frac{\|\boldsymbol{c}_{ij}\|_{\ell^2}}{2d_{ij}^{\mathrm{L},n}}$. Then the definition of $d_{ij}^{\mathrm{L},n}$ implies that $t_{ij}\lambda_{\max}(\boldsymbol{n}_{ij},\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*}) \le \frac{1}{2}$, which in turn implies that $\overline{\mathbf{U}}_{ij}^n = \overline{\boldsymbol{v}}(t,\boldsymbol{n}_{ij},\mathbf{U}_i^{n,j,*},\mathbf{U}_j^{n,i,*})$. Then (3.10) is a consequence of the inequality proved above. □

Proposition 3.9 (*Entropy Inequality*)**.** *Assume that the time step satisfies the CFL condition* $\tau \le \min_{i\in\mathcal{V}}\frac{m_i}{2|d_{ii}^{\mathrm{L},n+1}|}$. *Then, the low-order update satisfies the following discrete entropy inequality for every* $i\in\mathcal{V}$*:*

$$\frac{m_i}{\tau}\big(E_{\mathrm{flat}}(\mathbf{U}_i^{\mathrm{L},n+1}) - E_{\mathrm{flat}}(\mathbf{U}_i^n)\big) + \sum_{j\in\mathcal{G}(i)}\Big\{\big(\boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_j^{n,i,*}) - \boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_i^{n,j,*})\big)\boldsymbol{c}_{ij} - d_{ij}^{\mathrm{L},n}\big(E_{\mathrm{flat}}(\mathbf{U}_j^{n,i,*}) - E_{\mathrm{flat}}(\mathbf{U}_i^{n,j,*})\big)\Big\} \le \nabla_{\boldsymbol{u}}E_{\mathrm{flat}}(\mathbf{U}_i^{\mathrm{L},n})\cdot\mathbf{B}_i^{\mathrm{L},n} + \mathcal{O}(\tau). \tag{3.11}$$

When there is no influence due to topography, the entropy inequality reduces to:

$$\frac{m_i}{\tau}\Big(E_{\mathrm{flat}}(\mathbf{U}_i^{\mathrm{L},n+1}) - E_{\mathrm{flat}}(\mathbf{U}_i^n)\Big) + \sum_{j\in\mathcal{G}(i)}\Big\{\big(\boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_j) - \boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_i)\big)\boldsymbol{c}_{ij} - d_{ij}^{\mathrm{L},n}\big(E_{\mathrm{flat}}(\mathbf{U}_j) - E_{\mathrm{flat}}(\mathbf{U}_i)\big)\Big\} \le 0 \tag{3.12}$$

Proof. We first rewrite the convex combination (3.7) as follows:

$$\mathbf{U}_i^{\mathrm{L},n+1} - \frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n} = \Big(1+\frac{2\tau d_{ii}^{\mathrm{L},n}}{m_i}\Big)\mathbf{U}_i^n + \sum_{j\in\mathcal{G}^*(i)}\frac{2\tau d_{ij}^{\mathrm{L},n}}{m_i}\overline{\mathbf{U}}_{ij}^n.$$

Invoking Lemma 3.8 to bound $E_{\mathrm{flat}}(\overline{\mathbf{U}}_{ij}^n)$ from above and exploiting the convexity of the functional E_{flat} gives the inequality:

$$\frac{m_i}{\tau}\Big(E_{\mathrm{flat}}\big(\mathbf{U}_i^{\mathrm{L},n+1} - \frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}\big) - E_{\mathrm{flat}}(\mathbf{U}_i^n)\Big) + \sum_{j\in\mathcal{G}(i)}\Big\{\big(\boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_j^{n,i,*}) - \boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_i^{n,j,*})\big)\boldsymbol{c}_{ij} - d_{ij}^{\mathrm{L},n}\big(E_{\mathrm{flat}}(\mathbf{U}_j^{n,i,*}) - E_{\mathrm{flat}}(\mathbf{U}_i^{n,j,*})\big)\Big\} \le 0 \tag{3.13}$$

As a last ingredient for showing (3.11) we use the following Taylor series expansion:

$$\begin{aligned}\frac{m_i}{\tau}E_{\mathrm{flat}}\big(\mathbf{U}_i^{\mathrm{L},n+1} - \frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}\big) &= \frac{m_i}{\tau}E_{\mathrm{flat}}\big(\mathbf{U}_i^{\mathrm{L},n+1}\big) - \nabla_{\boldsymbol{u}}E_{\mathrm{flat}}(\mathbf{U}_i^{\mathrm{L},n+1})\cdot\mathbf{B}_i^{\mathrm{L},n} + \mathcal{O}(\tau)\\ &= \frac{m_i}{\tau}E_{\mathrm{flat}}\big(\mathbf{U}_i^{\mathrm{L},n+1}\big) - \nabla_{\boldsymbol{u}}E_{\mathrm{flat}}(\mathbf{U}_i^{\mathrm{L},n})\cdot\mathbf{B}_i^{\mathrm{L},n} + \mathcal{O}(\tau).\end{aligned}$$

Finally, (3.13) readily reduces to (3.12) for the case of constant bathymetry. □

3.4. High-order spatial approximation

We now present a high-order spatial approximation of the problem. For simplicity, we follow [28, Sec. 3.1] and make the following assumptions: (i) The low-order and high-order methods use the same spatial degrees of freedom; (ii) The low-order and high-order fluxes are computed over the same stencil (see also Remark 3.21). For $i\in\mathcal{V}$ and $j\in\mathcal{G}(i)$, we define the following high-order flux:

$$\mathbf{F}_{ij}^{\mathrm{H},n} := -\Big(\mathbf{U}_j^n\otimes\mathbf{V}_j^n + \mathbf{U}_i^n\otimes\mathbf{V}_i^n\Big)\boldsymbol{c}_{ij} + d_{ij}^{\mathrm{H},n}\Big(\mathbf{U}_j^{n,i,*} - \mathbf{U}_i^{n,j,*}\Big) - \begin{pmatrix}0\\ g(\mathsf{H}_i^n\mathsf{H}_j^n + \mathsf{H}_i^n(\mathsf{Z}_j-\mathsf{Z}_i))\boldsymbol{c}_{ij}\end{pmatrix}, \tag{3.14}$$

where $d_{ij}^{\mathrm{H},n} = d_{ji}^{\mathrm{H},n}$ is a high-order graph viscosity. The symmetry of $d_{ij}^{\mathrm{H},n}$ implies that $\mathbf{F}_{ij}^{\mathrm{H},n} = -\mathbf{F}_{ji}^{\mathrm{H},n}$ holds true when the topography is flat. We set $\mathbf{F}_i^{\mathrm{H},n} := \sum_{j\in\mathcal{G}(i)}\mathbf{F}_{ij}^{\mathrm{H},n}$. The high-order graph viscosity coefficient $d_{ij}^{\mathrm{H},n}$ is defined as follows:

$$d_{ij}^{\mathrm{H},n} := d_{ij}^{\mathrm{L},n}\frac{\alpha_i^n+\alpha_j^n}{2} \quad \text{for } i\neq j, \qquad d_{ii}^{\mathrm{H},n} := \sum_{j\in\mathcal{G}^*(i)}d_{ij}^{\mathrm{H},n}. \tag{3.15}$$

Here, $\alpha_i^n\in[0,1]$ is an indicator for entropy production and is defined as follows for each $i\in\mathcal{V}$:

$$\alpha_i^n := \frac{|N_i^n|}{D_i^n + \epsilon D_{\max}}, \tag{3.16}$$

$$N_i^n := \sum_{j\in\mathbb{I}(i)}\big\{\boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_j^n) - (\nabla_{\boldsymbol{u}}E_{\mathrm{flat}}(\mathbf{U}_i^n))^{\mathsf{T}}\nabla\cdot\mathbb{f}(\mathbf{U}_j^n)\big\}\boldsymbol{c}_{ij}, \tag{3.17}$$

$$D_i^n := \Big|\sum_{j\in\mathbb{I}(i)}\boldsymbol{F}_{\mathrm{flat}}(\mathbf{U}_j^n)\boldsymbol{c}_{ij}\Big| + \Big|\sum_{j\in\mathbb{I}(i)}(\nabla_{\boldsymbol{u}}E_{\mathrm{flat}}(\mathbf{U}_i^n))^{\mathsf{T}}\nabla\cdot\mathbb{f}(\mathbf{U}_j^n)\boldsymbol{c}_{ij}\Big|. \tag{3.18}$$

The small number $\epsilon D_{\max} := \epsilon\times\sqrt{g\mathsf{h}_{\max}}\frac{1}{2}g\mathsf{h}_{\max}^2$ is meant to avoid division by zero when either the water depth or the velocity is zero or when the entropy is constant.

As the high-order approximation requires estimating the inverse of the high-order mass matrix to reduce the dispersion effects, we proceed as in [38, §3.4] to approximate $(\mathbb{M}^{\mathrm{H}})^{-1}$. Using the expression $(\mathbb{M}^{\mathrm{H}})^{-1} = (\mathbb{M}^{\mathrm{L}})^{-1}\big(\mathbb{I}-(\mathbb{M}^{\mathrm{L}}-\mathbb{M}^{\mathrm{H}})(\mathbb{M}^{\mathrm{L}})^{-1}\big)^{-1}$ and setting $\mathbb{B} := (\mathbb{M}^{\mathrm{L}}-\mathbb{M}^{\mathrm{H}})(\mathbb{M}^{\mathrm{L}})^{-1}$, we approximate $(\mathbb{M}^{\mathrm{H}})^{-1}$ by $(\mathbb{M}^{\mathrm{L}})^{-1}(\mathbb{I}+\mathbb{B})$. This expansion is shown in [39, Prop. 3.1] to be superconvergent and to remove the dispersion error for the approximation of the linear transport equation with piecewise linear continuous finite elements on uniform meshes. Let $\{b_{ij}\}_{j\in\mathcal{G}(i)}$ for all $i\in\mathcal{V}$ be the entries of $\mathbb{B}$, i.e., $b_{ij} = \delta_{ij} - \frac{m_{ij}}{m_j}$ and $b_{ji} = \delta_{ij} - \frac{m_{ji}}{m_i}$. Then, recalling that $\sum_{j\in\mathcal{G}(i)}b_{ji} = 0$, the provisional high-order approximation with forward Euler time-stepping is given by:

$$\frac{m_i}{\tau}(\mathbf{U}_i^{\mathrm{H},n+1} - \mathbf{U}_i^n) = \sum_{j\in\mathcal{G}(i)}\mathbf{F}_{ij}^{\mathrm{H},n} + b_{ij}\mathbf{F}_j^{\mathrm{H},n} - b_{ji}\mathbf{F}_i^{\mathrm{H},n}. \tag{3.19}$$

As the provisional high-order update $\mathbf{U}_i^{\mathrm{H},n+1}$ defined above is not always well-balanced and also not guaranteed to be invariant-domain preserving, we present in the next section a convex limiting technique that combines the low-order update and the provisional high-order update in a way that makes the final update at t^{n+1} high-order accurate, well-balanced, and invariant-domain preserving.

Remark 3.10 (*Well-balancing*)**.** The high-order flux (3.14) is such that

$$\mathbf{F}_i^{\mathrm{H},n} = \sum_{j\in\mathcal{G}(i)} -\Big(\mathbf{U}_j^n\otimes\mathbf{V}_j^n\Big)\boldsymbol{c}_{ij} - \begin{pmatrix}0\\ g\mathsf{H}_i^n(\mathsf{H}_j^n+\mathsf{Z}_j-\mathsf{H}_j^n-\mathsf{Z}_i)\boldsymbol{c}_{ij}\end{pmatrix} + d_{ij}^{\mathrm{H},n}\Big(\mathbf{U}_j^{n,i,*} - \mathbf{U}_i^{n,j,*}\Big).$$

Hence, the high-order scheme (3.19) is well-balanced when the water depth is bounded from below away from zero (since $\mathbf{U}_j^{n,i,*} = \mathbf{U}_i^{n,j,*}$ and $\mathsf{H}+\mathsf{Z} = const$ in this case). This property is important to properly approximate the dynamics of the system under the action of small perturbations around states that are at rest. This property is numerically illustrated in Section 5.2.4. □

Remark 3.11 (*Spatial Accuracy*)**.** The spatial accuracy of the method (3.19) delivers close to optimal accuracy for smooth solutions for

the underlying spatial discretization. For example, assuming the discretization is based on continuous linear finite elements, the method is formally second-order accurate. □

Remark 3.12 (*High-order Graph-viscosity*)**.** As convex limiting keeps the approximation in bounds, it might be tempting to set the high-order graph-viscosity coefficient, $d_{ij}^{\mathrm{H},n}$, to zero so that the high-order method is just the "Galerkin" approximation to the PDE. However, we have demonstrated in the past that this idea is not robust and sometimes outright wrong as this may let the approximation converge to a weak solution that is not physical. We refer the reader to Section 7.2 in [17] and the literature cited in this section where this claim is substantiated. □

3.5. Convex limiting procedure

We now detail the convex limiting procedure. The methodology is loosely based on [17,40] and follows the common FCT ideology (see e.g., [41–43]). The novelty of the approach proposed in the paper resides in the definition of the local bounds and the incorporation of sources in the convex combination introduced in Lemma 3.6.

3.5.1. Local bounds

For each $i \in \mathcal{V}$, we let $\{\lambda_j\}_{j\in\mathcal{J}^*(i)}$ be any set of positive coefficients that sum up to 1. In the numerical illustrations reported at the end of the paper we use $\lambda_i = \frac{1}{\mathrm{Card}(\mathcal{J}^*(i))}$. Subtracting (3.3) from (3.19) we obtain

$$(\mathbf{U}_i^{\mathrm{H},n+1} - \mathbf{U}_i^{\mathrm{L},n+1}) = \sum_{j\in\mathcal{J}^*(i)} \lambda_i \mathbf{P}_{ij}^n, \tag{3.20}$$

with

$$\mathbf{P}_{ij}^n := \frac{\tau}{m_i\lambda_i}\{\mathbf{F}_{ij}^{\mathrm{H},n} - \mathbf{F}_{ij}^{\mathrm{L},n} + b_{ij}\mathbf{F}_j^{\mathrm{H},n} - b_{ji}\mathbf{F}_i^{\mathrm{H},n}\}. \tag{3.21}$$

Notice that the coefficients $\mathbf{P}_{ij}^n$ are skew-symmetric. The key principle of the convex limiting strategy is as follows: For all $i \in \mathcal{V}$ and all $j \in \mathcal{J}^*(i)$, we look for a set of symmetric limiting coefficients $\ell_{ij}^n \in [0,1]$ such that the limited update $\mathbf{U}_i^{\mathrm{L},n+1} + \sum_{j\in\mathcal{J}^*(i)} \ell_{ij}^n \lambda_i \mathbf{P}_{ij}^n$ satisfies reasonable properties and is well-balanced. After finding this collection of limiting coefficients, we define the final update to be

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^{\mathrm{L},n+1} + \sum_{j\in\mathcal{J}^*(i)} \ell_{ij}^n \lambda_i \mathbf{P}_{ij}^n. \tag{3.22}$$

Notice that the final update can be equivalently written as:

$$\mathbf{U}_i^{n+1} = \sum_{j\in\mathcal{J}^*(i)} \lambda_i \left(\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij}^n \mathbf{P}_{ij}^n\right). \tag{3.23}$$

which is just a convex combination of limited states. We now explain how we define the local bounds which are used to define the limiting coefficients. The following details differ from our previous work [17, 40]. Taking inspiration from the convex combination (3.7) for all $i \in \mathcal{V}$ and all $j \in \mathcal{J}(i)$, we set

$$\overline{\mathbf{W}}_{ij}^n := \overline{\mathbf{U}}_{ij}^n + \frac{\tau}{m_i}\mathbf{B}_i^{\mathrm{L},n}. \tag{3.24}$$

We then define the minimum and maximum local bound on the water depth by setting:

$$\mathsf{H}_{i,\min}^n = \min_{j\in\mathcal{J}(i)} \mathsf{H}(\overline{\mathbf{W}}_{ij}^n), \qquad \mathsf{H}_{i,\max}^n = \max_{j\in\mathcal{J}(i)} \mathsf{H}(\overline{\mathbf{W}}_{ij}^n). \tag{3.25}$$

In order to control the potential blow-up of the velocity in dry states, we introduce the quantity $\mathsf{H}(\mathbf{U})^2\mathsf{V}_{i,\max}^{2,n} - \|\mathbf{Q}(\mathbf{U})\|_{\ell^2}^2$ where

$$\mathsf{V}_{i,\max}^{2,n} = \max_{j\in\mathcal{J}(i)} \|\mathbf{V}(\overline{\mathbf{W}}_{ij}^n)\|_{\ell^2}^2. \tag{3.26}$$

Here, $\mathbf{V}(\mathbf{U})$ is the regularized version of the velocity. In [17, p. A3889], the authors propose a limiting technique based on the kinetic energy ($\frac{1}{2}\|\mathbf{Q}(\mathbf{U})\|_{\ell^2}^2/\mathsf{H}(\mathbf{U})$), but we found that the approach we propose here is more robust with respect to dry states. A control on the velocity via limiting is also adopted in [14, Sec. 3.2]. The following result is essential to establish the validity of the limiting process.

Lemma 3.13. *Assume that $\mathbf{U}_i^n \in \mathcal{A}$ for all $i \in \mathcal{V}$. Assume also that the time step satisfies the restriction $\tau \le \min_{i\in\mathcal{V}} \frac{m_i}{2|d_{ii}^{\mathrm{L},n}|}$. Then,*

$$H_{i,\min}^n \le H(\mathbf{U}_i^{\mathrm{L},n+1}) \quad \text{and} \quad H(\mathbf{U}_i^{\mathrm{L},n+1}) \le H_{i,\max}^n, \tag{3.27}$$

$$\|\mathbf{Q}(\mathbf{U}_i^{\mathrm{L},n+1})\|_{\ell^2}^2 \le H(\mathbf{U}_i^{\mathrm{L},n+1})V_{i,\max}^{2,n} \qquad \text{if } \epsilon h_{\max} \le H_{i,\min}^n. \tag{3.28}$$

Proof. The bound (3.27) is a direct consequence of the combination (3.7) being convex and the mapping $\mathbf{U} \to \mathsf{H}(\mathbf{U})$ being linear. We now prove the second bound. We first observe that the mapping $\mathbb{R}_{>0}\times\mathbb{R}^d \ni \mathbf{U} \to \|\mathbf{Q}(\mathbf{U})\|_{\ell^2}^2/\mathsf{H}(\mathbf{U}) \in \mathbb{R}$ is convex. Then, since $\mathbf{U} \to \mathsf{H}(\mathbf{U})$ is linear, the mapping $\mathbb{R}_{>0}\times\mathbb{R}^d \ni \mathbf{U} \to \|\mathbf{Q}(\mathbf{U})\|_{\ell^2}^2/\mathsf{H}^2(\mathbf{U}) \in \mathbb{R}$ is quasi-convex due to [34, Lem. 7.4]. An application of [34, Lem. 7.2] yields: $\|\mathbf{V}(\mathbf{U}_i^{\mathrm{L},n+1})\|_{\ell^2}^2 \le \mathsf{V}_{i,\max}^{2,n}$. As we assumed that $\mathsf{H}_{i,\min}^n \ge \epsilon\mathsf{h}_{\max}$, we infer that $\mathsf{H}(\overline{\mathbf{W}}_{ij}^n) \ge \epsilon\mathsf{h}_{\max} > 0$ and $\mathsf{H}(\mathbf{U}_i^{\mathrm{L},n}) \ge \epsilon\mathsf{h}_{\max} > 0$. This implies in particular that $\mathbf{Q}(\overline{\mathbf{W}}_{ij}^n) = \mathsf{H}(\overline{\mathbf{W}}_{ij}^n)\mathbf{V}(\overline{\mathbf{W}}_{ij}^n)$ and $\mathbf{Q}(\mathbf{U}^{\mathrm{L},n}) = \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n})\mathbf{V}(\mathbf{U}^{\mathrm{L},n})$. Hence,

$$\|\mathbf{V}(\mathbf{U}_i^{\mathrm{L},n+1})\|_{\ell^2}^2 \le \mathsf{V}_{i,\max}^{2,n} \implies \|\mathbf{Q}(\mathbf{U}_i^{\mathrm{L},n+1})\|_{\ell^2}^2 \le \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n+1})^2\mathsf{V}_{i,\max}^{2,n}.$$

This completes the proof. □

Remark 3.14 (*Positivity-preserving vs. Invariant-domain Preserving*)**.** The local bounds defined above combined with the convex limiting methodology gives a method that goes a little beyond the traditional "positivity-preserving" approach. More precisely, for every $i \in \mathcal{V}$, positivity-preserving methods locally enforce $0 < \mathsf{h}(\mathbf{U}_i)$, whereas the proposed approach yields

$$0 < \mathsf{H}_{i,\min}^n \le \mathsf{h}(\mathbf{U}_i) \le \mathsf{H}_{i,\max}^n, \qquad \|\boldsymbol{q}(\mathbf{U}_i)\|_{\ell^2}^2 \le \mathsf{h}(\mathbf{U}_i)^2\mathsf{V}_{i,\max}^{2,n}. \tag{3.29}$$

Note that the first inequality above implies that scheme can also be classified as a "positivity-preserving" scheme. □

Remark 3.15 (*Bounds Relaxation*)**.** To achieve optimal accuracy in L^p-norms, $p \ge 1$, for smooth solutions, the bounds defined above must be relaxed. For the sake of brevity, we refer the reader to [34, Sec. 7.6] where this is discussed in detail. □

Remark 3.16 (*Source Terms*)**.** In order to incorporate the external source term $\boldsymbol{S}(\boldsymbol{u})$ given by (2.5) into the high-order scheme and the subsequent convex limiting procedure we change the definition of $\mathbf{P}_{ij}$ as follows:

$$\begin{aligned}\mathbf{P}_{ij}^n := \frac{\tau}{m_i\lambda_i}\Big\{ &\mathbf{F}_{ij}^{\mathrm{H},n} - \mathbf{F}_{ij}^{\mathrm{L},n} + b_{ij}\mathbf{F}_j^{\mathrm{H},n} - b_{ji}\mathbf{F}_i^{\mathrm{H},n} \\ &+ m_{ij}\boldsymbol{S}_j^n - m_{ij}\boldsymbol{S}_i^n + b_{ij}\Big(\sum_{k\in\mathcal{J}(j)} m_{jk}\boldsymbol{S}_k^n\Big) - b_{ji}\Big(\sum_{k\in\mathcal{J}(i)} m_{ik}\boldsymbol{S}_k^n\Big)\Big\},\end{aligned} \tag{3.30}$$

where $\boldsymbol{S}_i^n$ is again given by (3.9). □

3.5.2. Optimal limiting coefficient

We now detail the process for finding near optimal limiting coefficients ℓ_{ij}. We introduce the functionals:

$$\Psi_1(\mathbf{U}) := \mathsf{H}(\mathbf{U}) - \mathsf{H}_i^{n,\min}, \qquad \Psi_2(\mathbf{U}) := \mathsf{H}_i^{n,\max} - \mathsf{H}(\mathbf{U}), \tag{3.31a}$$

$$\Psi_3(\mathbf{U}) := \mathsf{H}(\mathbf{U})^2\mathsf{V}_{i,\max}^{2,n} - \|\mathbf{Q}(\mathbf{U})\|_{\ell^2}^2. \tag{3.31b}$$

The strategy is as follows: for each $k \in \{1,2,3\}$, we find $\ell \in [0,1]$ such that $\Psi_k(\mathbf{U}_i^{\mathrm{L},n+1} + \ell\mathbf{P}_{ij}^n) \ge 0$ in a sequential manner over k.

We first limit the water depth. To ensure robustness with respect to dry states, we introduce for i in $\mathcal{V}$:

$$\ell_j^{i,\mathsf{h}} = \begin{cases} \min\Big(\frac{\left|\mathsf{H}_{i,\min}^n - \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n+1})\right|}{\left|\mathsf{P}_{ij}^{\mathsf{h}}\right| + \epsilon \mathsf{H}_{i,\max}^n}, 1\Big), & \text{if } \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n+1}) + \mathsf{P}_{ij}^{\mathsf{h}} < \mathsf{H}_{i,\min}^n, \\ 1, & \mathsf{H}_{i,\min}^n \le \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n+1}) + \mathsf{P}_{ij}^{\mathsf{h}} \le \mathsf{H}_{i,\max}^n, \\ \min\Big(\frac{\left|\mathsf{H}_{i,\max}^n - \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n+1})\right|}{\left|\mathsf{P}_{ij}^{\mathsf{h}}\right| + \epsilon \mathsf{H}_{i,\max}^n}, 1\Big), & \text{if } \mathsf{H}_{i,\max}^n < \mathsf{H}(\mathbf{U}_i^{\mathrm{L},n+1}) + \mathsf{P}_{ij}^{\mathsf{h}}. \end{cases} \tag{3.32}$$

This process guarantees that $\Psi_1(\mathbf{U}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^n) \ge 0$ and $\Psi_2(\mathbf{U}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^n) \ge 0$ for all $\ell \in [0, \ell_j^{i,\mathsf{h}}]$. This enforces a local minimum principle and a local maximum principle on the water depth. As a corollary this also enforces positivity of the water depth H_i^{n+1}.

After limiting the water depth, we limit the velocity based on the bound (3.28). Notice that the functional $\Psi_3(\mathbf{U}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^n)$ is quadratic in ℓ:

$$\Psi_3(\mathbf{U}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^n) = (\mathsf{H}_i^{\mathrm{L},n+1} + \ell \mathsf{P}_{ij}^h)^2 \mathsf{V}_{i,\max}^{2,n} - \|\mathbf{Q}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^{\boldsymbol{q}}\|_{\ell^2}^2. \tag{3.33}$$

Thus, one can find the root $\ell_j^{i,\boldsymbol{\nu}} \in [0, \ell_j^{i,\mathsf{h}}]$ of $\Psi_3(\mathbf{U}_i^{\mathrm{L},n+1} + \ell_j^{i,\boldsymbol{\nu}} \mathbf{P}_{ij}^n) = 0$ by either solving the quadratic equation as in [17, Eq. (6.33)-(6.34)] or simply employing a quadratic Newton algorithm. We refer the reader to [22, Alg. 3] for a description of the quadratic newton algorithm implemented in the code used for the numerical illustrations. This process guarantees that $\Psi_3(\mathbf{U}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^n) \ge 0$ for all $\ell \in [0, \ell_j^{i,\boldsymbol{\nu}}]$. This enforces a local maximum principle on the quantity $\|\boldsymbol{\nu}\|_{\ell^2}^2$. As a corollary, this also enforces that the final solution is well-balanced with respect to rest states.

Finally, we set the optimal limiting coefficient to

$$\ell_{ij}^n := \min(\ell_j^{i,\boldsymbol{\nu}}, \ell_i^{j,\boldsymbol{\nu}}), \qquad \text{for all } i \in \mathcal{V} \text{ and } j \in \mathcal{J}(i). \tag{3.34}$$

The symmetry ensures that the limiting is conservative.

3.5.3. Conservation, invariant-domain preservation and well balancing

We now formalize results for the convex limiting procedure concerning conservation, invariant-domain preservation and well balancing.

Proposition 3.17 (*Conservation*). *The update given by (3.22) is conservative, i.e., the total mass is conserved if the mass flux at the boundary is zero, and the total discharge is conserved if the bathymetry is flat and the boundary flux is zero.*

Proof. Recalling the definitions introduced in Section 3.1, we let $\mathsf{H}(\mathbf{P}_{ij}^n)$ be the mass component of $\mathbf{P}_{ij}^n$. After inspection of $\mathbf{F}_{ij}^{\mathrm{L},n}$ and $\mathbf{F}_{ij}^{\mathrm{H},n}$, we observe that $\mathsf{H}(\mathbf{F}_{ij}^{\mathrm{L},n}) = -\mathsf{H}(\mathbf{F}_{ji}^{\mathrm{L},n})$ and $\mathsf{H}(\mathbf{F}_{ij}^{\mathrm{H},n}) = -\mathsf{H}(\mathbf{F}_{ji}^{\mathrm{H},n})$, whence $\mathsf{H}(\mathbf{P}_{ij}^n) = -\mathsf{H}(\mathbf{P}_{ji}^n)$. Given that the limiter ℓ_{ij} is symmetric, we infer from (3.22) that $\sum_{i\in\mathcal{V}} m_i \mathsf{H}_i^{n+1} = \sum_{i\in\mathcal{V}} m_i \mathsf{H}_i^n$, i.e., the total mass is conserved. Let us now assume that the bathymetry is flat and there are no contributions from external source terms so that the total discharge is conserved by the PDE system (2.1). Then, arguing as above we observe that $\mathbf{Q}(\mathbf{P}_{ij}) = -\mathbf{Q}(\mathbf{P}_{ji})$, which in turn implies that $\sum_{i\in\mathcal{V}} m_i \mathbf{Q}_i^{n+1} = \sum_{i\in\mathcal{V}} m_i \mathbf{Q}_i^n$, i.e., the total discharge is conserved on the discrete level. □

Proposition 3.18 (*Invariant-domain Preserving*). *Let $n \ge 0$. Assume that the $\mathbf{U}_i^n \in \mathcal{A}$ for all $i \in \mathcal{V}$. Then the update $\mathbf{U}_i^{n+1}$ given by (3.22) with the limiting coefficient (3.34) is invariant-domain preserving under the time-step restriction $\tau \le \min_{i\in\mathcal{V}} \frac{m_i}{2|d_{ii}^{\mathrm{L},n}|}$.*

Proof. Suppose that the time-step restriction $\tau \le \min_{i\in\mathcal{V}} \frac{m_i}{2|d_{ii}^{\mathrm{L},n}|}$ holds. Then, the combination in Lemma 3.6 is convex and the local bounds in Lemma 3.13 hold true. Then, by construction of the limiter (3.32), we have that:

$$\mathsf{H}(\mathbf{U}_i^{n+1}) = \mathsf{H}\Big(\sum_{j\in\mathcal{J}(i)} \lambda_i (\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij}^n \mathbf{P}_{ij}^n)\Big) \ge \mathsf{H}_{i,\min}^n > 0.$$

Thus, $\mathbf{U}_i^{\mathrm{L},n+1} \in \mathcal{A}$ for all $i \in \mathcal{V}$. □

Proposition 3.19 (*Well Balancing*). *Let $n \ge 0$ and assume that the given state $\{\mathbf{U}_i^n\}_{i\in\mathcal{V}} \subset \mathcal{A}$ is at rest as formalized in Definition 3.2. Then, the update $\mathbf{U}_i^{n+1}$ given by (3.22) with the limiting coefficient (3.34) is at rest under the time-step restriction $\tau \le \min_{i\in\mathcal{V}} \frac{m_i}{2|d_{ii}^{\mathrm{L},n}|}$. (This means that the scheme is well-balanced with respect to rest states.)*

Proof. Assume that at t^n the discrete state, $\{\mathbf{U}_i^n\}_{i\in\mathcal{V}}$, is at rest in the sense of Definition 3.2. By Lemma 3.5, the low-order update $\mathbf{U}^{\mathrm{L},n+1}$ is at rest. By assumption, we have that $\mathbf{V}_i^n = \mathbf{0}$ for all $i \in \mathcal{V}$ and so $\mathsf{V}_{i,\max}^{2,n} = 0$. Then, the limiting strategy for $\Psi_3(\mathbf{U}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^n) = 0$ reduces to finding ℓ such that $\|\mathbf{Q}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^{\boldsymbol{q}}\|_{\ell^2}^2 = 0$. As $\mathbf{Q}_i^{\mathrm{L},n+1} = \mathbf{0}$, we infer that $\ell^2 \|\mathbf{P}_{ij}^{\boldsymbol{q}}\|_{\ell^2}^2 = 0$. If $\|\mathbf{P}_{ij}^{\boldsymbol{q}}\|_{\ell^2}^2 = 0$, every value $\ell \in [0,1]$ gives $\|\mathbf{Q}_i^{\mathrm{L},n+1} + \ell \mathbf{P}_{ij}^{\boldsymbol{q}}\|_{\ell^2}^2 = 0$, otherwise one must have $\ell = 0$ and the same conclusion holds. Thus, the final update (3.22) reduces to the low-order solution $\mathbf{U}^{\mathrm{L},n+1}$ which is well-balanced. □

Remark 3.20 (*Well-balancing*). There are two key ingredients to achieve well-balancing of the high-order method: (i) The low-method (3.3) is well-balanced whether the shoreline coincides with mesh interfaces or not; (ii) Limiting on the squared norm of the velocity (3.31b)&(3.33) maintains the well-balancing of the high-order method (if the low-order method is at rest, then the high-order method stays at rest). In our previous works [17,40], the low-order method was well-balanced only if the shoreline coincided with the mesh. This is no longer the case with the present method. □

Remark 3.21 (*More General Limiting*). The assumptions made at the beginning of this section that the low-order and high-order approximations have the same stencil and use the same degrees of freedom can be relaxed by proceeding as in [44] or Zhang and Shu [45]. In [44], the low-order and high-order global shape functions are different and the stencil of the low-order approximation reduces to the next neighbors. The limiting of each degree of freedom is based on a convex combination using two sums: one sum is done over the degrees of freedom listed in the high-order stencil and one sum is done over the cells supporting the high-order shape function associated with the degree of freedom being limited. When using a discontinuous Galerkin approximation as in [45], the limiting is done cell-wise over the cells supporting the high-order shape functions. □

4. High-order IDP time-stepping: algorithmic and implementation details

In order to achieve higher order accuracy in time, the convex-limiting strategy (3.22) based on the forward Euler time stepping is now used as a building block in a higher order explicit Runge Kutta scheme. To ensure robustness of the method it is crucial that the high-order Runge–Kutta update be also invariant domain preserving. A widely used family of Runge–Kutta schemes achieving this are strong stability preserving (SSP) explicit Runge–Kutta (ERK) methods introduced by Shu and Osher [46]; see also [4,47]. Here, we choose a slightly different approach by using a family of invariant-domain preserving (IDP) explicit Runge–Kutta methods [28] that have the distinct advantage of having a milder time step size restriction than SSP-ERK methods. We refer the reader to [28] for a detailed discussion about derivation and design of IDP-ERK methods. We also refer the reader to Kuzmin et al. [48, Sec. 3.3] where a similar idea is discussed that revolves around

ensuring stability properties through limiting of ERK stages. The work proposed here focuses on maximally efficient ERK methods as opposed to general ERK methods in [48, Sec. 3.3].

For the sake of completeness, we recall the general setup and formulas for ERK methods. Let us consider an ERK method composed of $s \geq 1$ stages and defined by its Butcher tableau:

$$\begin{array}{c|ccccc} c_1 & 0 & & & & \\ c_2 & a_{2,1} & 0 & & & \\ c_3 & a_{3,1} & a_{3,2} & 0 & & \\ \vdots & \vdots & & \ddots & \ddots & \\ c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \tag{4.1}$$

with $c_1 := 0$. We additionally set $c_{s+1} := 1$. We recall that the coefficients c_j define the intermediate time steps $t^{n,j} := t^n + c_j\tau_{\mathrm{ERK}}$ where τ_{ERK} is the time step at time t^n. Let $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u} = \mathscr{L}(t,\boldsymbol{u})$ denote a generic system of ordinary differential equations, then the s-stage ERK method for approximating this system is given by: $\boldsymbol{u}^{n,l} := \boldsymbol{u}^n + \tau_{\mathrm{ERK}}\sum_{j\in\{1:l-1\}} a_{l,j}\,\mathscr{L}(t^n + c_j\tau_{\mathrm{ERK}}, \boldsymbol{u}^{n,j})$ for all $l \in [1:s]$ and $\boldsymbol{u}^{n+1} := \boldsymbol{u}^n + \tau_{\mathrm{ERK}}\sum_{j\in\{1:s\}} b_j\,\mathscr{L}(t^n + c_j\tau_{\mathrm{ERK}}, \boldsymbol{u}^{n,j})$. The coefficients (4.1) satisfy various consistency criteria which we omit here for brevity (see e.g., [49, II.2] and [28, Sec. 2.3]).

Definition 4.1 (*Efficiency Ratio*)**.** Setting $l'(l) := \min\{k \in [1:l-1] | c_l - c_k \geq 0\}$ for all $l \in [2:s+1]$, the efficiency ratio of the method is defined to be

$$c_{\mathrm{eff}} := \frac{1}{s\Delta c^{\max}}, \quad \text{where} \quad \Delta c^{\max} := \max_{l\in\{2:s+1\}} \big(c_l - c_{l'(l)}\big). \tag{4.2}$$

In the following we focus our attention on a family of ERK methods that are second, third, and fourth order accurate and with optimal efficiency ratio [28]. In addition to $c_1 := 0$, these methods are such that $c_l - c_{l-1} := \frac{1}{s}$, giving that $c_{\mathrm{eff}} = 1$. These methods are optimal in the sense that if the low-order explicit Euler step is IDP for some time step τ, then it is shown in [28] the s-stage ERK method can also be made IDP as well with $\tau_{\mathrm{ERK}} = s\tau$. As in [28], we adopt the notation $\mathrm{RK}(s,p;c_{\mathrm{eff}})$ to differentiate the various ERK methods invoked in the paper.

We now present a reformulation of the IDP-ERK paradigm specialized to ERK methods that have an optimal efficiency ratio. This reformulation is particularly designed for high-performance computing. Given a state vector $\mathbf{U}^n$ at time t^n and a (single-step) time-step size τ_n satisfying the step size restriction of Lemma 3.6, we construct a sequence of updates as follows:

$$\mathbf{U}^n =: \mathbf{U}^{(1)} \xrightarrow{+\tau_n} \mathbf{U}^{(2)} \xrightarrow{+\tau_n} \cdots \xrightarrow{+\tau_n} \mathbf{U}^{(s)} \xrightarrow{+\tau_n} \mathbf{U}^{(s+1)} =: \mathbf{U}^{n+1}. \tag{4.3}$$

Notice that here we use the elementary time step τ_n instead of the global time step τ_{ERK}. Recall that $\tau_{\mathrm{ERK}} = s\tau_n$ when $c_{\mathrm{eff}} = 1$. Let us introduce the notation $a_{kk} := 0$ and $a_{s+1,k} := b_k$ for all $k \in \{1:s\}$. Then we define the weights w_{lk} for $l \in \{1:s\}$, $k \in \{1:l\}$ as follows:

$$w_{lk} := s\,(a_{l+1,k} - a_{l,k}). \tag{4.4}$$

We start with $\mathbf{U}^{(1)} := \mathbf{U}^n$. Then, for $l \in [1:s]$, we compute the $(l+1)$-th stage vector $\mathbf{U}^{(l+1)}$ at time $t^n + l\tau_n$ with the following procedure:

- Using the previous stage $\mathbf{U}^{(l)}$, we compute the low-order fluxes $\boldsymbol{F}_{ij}^{\mathrm{L},(l)}$ with Eq. (3.3b). Then we compute the low-order update $\mathbf{U}^{\mathrm{L},(l+1)}$ with (3.3a) and the time step size τ_n.
- Using the previous stage $\mathbf{U}^{(l)}$ again, we compute the high-order fluxes $\boldsymbol{F}_{ij}^{\mathrm{H},(l)}$ with Eqs. (3.14), store these fluxes with the previous ones, $\boldsymbol{F}_{ij}^{\mathrm{H},(1)}$, …, $\boldsymbol{F}_{ij}^{\mathrm{H},(l-1)}$, and set

$$\check{\mathbf{F}}_{ij} := \sum_{k=1}^{l} w_{lk}\mathbf{F}_{ij}^{\mathrm{H},(k)} \quad \text{and} \quad \check{\mathbf{F}}_i = \sum_{j\in\mathcal{J}(i)} \check{\mathbf{F}}_{ij}. \tag{4.5}$$

- Next we compute the fluxes $\check{\mathbf{P}}_{ij}$ as in (3.21):

$$\check{\mathbf{P}}_{ij} := \frac{\tau_n}{m_i\lambda_i}\big\{\check{\mathbf{F}}_{ij} - \mathbf{F}_{ij}^{\mathrm{L},(l)} + b_{ij}\check{\mathbf{F}}_j - b_{ji}\check{\mathbf{F}}_i\big\}. \tag{4.6}$$

- Finally, we compute the limiter coefficients $\check{\ell}_{ij}$ as outlined in Section 3.5 using $\check{\mathbf{P}}_{ij}$ into (3.22), and we define the high-order update $\mathbf{U}^{(l+1)}$ by setting

$$\mathbf{U}_i^{(l+1)} = \mathbf{U}_i^{\mathrm{L},(l+1)} + \sum_{j\in\mathcal{J}^*(i)} \check{\ell}_{ij}\lambda_i\check{\mathbf{P}}_{ij}. \tag{4.7}$$

The procedure described above inherits the properties listed in Propositions 3.17, 3.18, and 3.19 at every stage.

Remark 4.2 (*SSP vs. IDP-ERK*)**.** The three common features of SSP and IDP-ERK methods are a low-order flux, a high-order flux, and a limiting technique. The key difference between SSP and IDP-ERK methods are as follows: (i) Limiting in SSP methods is done at every forward Euler step, whereas limiting in IDP-ERK methods is done at every Runge–Kutta stage. (ii) Limiting in SSP methods is done on forward Euler fluxes, whereas limiting in IDP-ERK methods is done on the Butcher fluxes (see (4.5)). (iii) Every ERK method can be made IDP, whereas the SSP structure (making SSP methods IDP) is only enjoyed by a very small subclass of ERK methods. (iv) IDP-ERK methods can have efficiency ratios equal to 1 whereas it is not the case of most SSP methods. (v) SSP methods are easy to implement, whereas IDP-ERK methods require storing the high-order fluxes and recombining them at every stage (see (4.5)). □

5. Numerical illustrations

In this section, we illustrate the proposed method with various configurations including: (i) well-balancing tests; (ii) validation tests for convergence; (iii) verification with small-scale laboratory experiments; (iv) realistic flooding scenario with a digital elevation model.

5.1. Technical details

The numerical tests are conducted using the high-performance finite element code, `ryujin` [22,23]. The code uses continuous $\mathbb{Q}_1$ finite elements on quadrangular meshes for the spatial approximation and is built upon the `deal.II` finite element library [50].

To differentiate the temporal approximations, we use the notation $\mathrm{RK}(s,p;c_{\mathrm{eff}})$. The efficiency ratio for the IDP-ERK schemes introduced in Section 4 is $c_{\mathrm{eff}} = 1$. All the methods with optimal efficiency used in the paper are summarized in . For the sake of completeness, we also give the Butcher tableaux of these methods in . We are also going to make use the standard SSP-ERK method $\mathrm{RK}(2,2;\frac{1}{2})$ and $\mathrm{RK}(3,3;\frac{1}{3})$ (see [46, Eq. 2.16] and [46, Eq. 2.18], respectively).

The time step size τ_n is computed during the first stage of each time step using the expression

$$\tau_n := \mathrm{CFL}\max_{i\in\mathcal{V}} \frac{m_i}{2|d_{ii}^{\mathrm{L},n}|}, \tag{5.1}$$

where $\mathrm{CFL} \in (0,1]$ is a user-defined constant henceforth called Courant–Friedrichs–Lewy number. The global time step is computed using $\tau_{\mathrm{ERK}} := c_{\mathrm{eff}}s\tau_n$.

In all the simulations reported below, we take $g = 9.81\,\mathrm{m\,s^{-2}}$. To characterize the convergence properties of the method, we use the following consolidated error indicator for our tests:

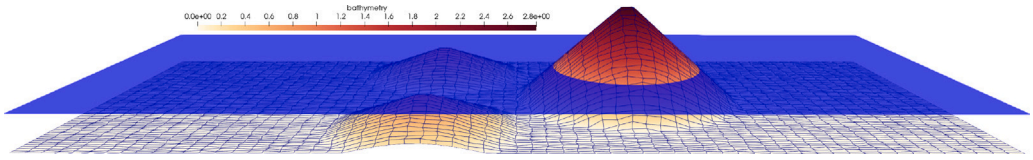


Fig. 1. Well-balancing configuration. Distorted mesh with 4225 $\mathbb{Q}_1$ degrees of freedom.

Table 1
Weights w_{ik} for the optimal IDP-ERK schemes RK(2, 2; 1), RK(3, 3; 1), RK(4, 3; 1), and RK(5, 4; 1).

(a) w_{ik} for RK(2, 2; 1)

	1	2
1	1	
2	−1	2

(b) w_{ik} for RK(3, 3; 1)

	1	2	3
1	1		1
2	−1	2	
3	3/4	−2	9/4

(c) w_{ik} for RK(4, 3; 1)

	1	2	3	4
1				
2	−1	2		
3	0	−1	2	
4	0	5/3	−10/3	8/3

(d) w_{ik} for RK(5, 4; 1)

	1	2	3	4
1	1.000000000000000			
2	0.303779113477746	0.696220886522255		
3	−2.596605007106260	3.860592821791782	−0.263987814685521	
4	2.373989715203703	−1.980102553333916	−3.819151895277756	4.425264733407969
5	−1.606747744309784	1.817291202624922	1.137969506889054	−2.114595709136266
				w_{55} = 1.766082743932075

Table 2
Butcher tableaux for the optimal IDP-ERK schemes RK(2, 2; 1), RK(3, 3; 1), RK(2, 2; 1), RK(4, 3; 1), and RK(5, 4; 1).

(a) RK(2, 2; 1)

0	0	
1/2	1/2	0
1	0	1

(b) RK(3, 3; 1)

0	0		0
1/3	1/3	0	
2/3	0	2/3	0
1	1/4	0	3/4

(c) RK(4, 3; 1)

0				
1/4	1/4	0		
2/4	0	2/4	0	
3/4	0	1/4	2/4	0
1	0	2/3	−1/3	2/3

(d) RK(5, 4; 1)

0	0			
1/5	0.2	0		
2/5	0.26075582269554909	0.13924417730445096	0	
3/5	−0.25856517872570289	0.91136274166280729	−0.05279756293710430	0
4/5	0.21623276431503774	0.51534223099602405	−0.81662794199265554	0.88505294668159373
1	−0.10511678454691901	0.87880047152100838	−0.58903404061484477	0.46213380485434047
				b_5 = 0.35321654878641495

$$\delta^q(T) := \frac{\|\mathsf{H} - \mathsf{h}_{\text{exact}}(T)\|_{L^q(D)}}{\|\mathsf{h}_{\text{exact}}(T)\|_{L^q(D)}} + \frac{\|\mathbf{Q} - \boldsymbol{q}_{\text{exact}}(T)\|_{L^q(D)}}{\|\boldsymbol{q}_{\text{exact}}(T)\|_{L^q(D)}},$$

where $q \in \{1, \infty\}$. In all the error tables, the symbol I denotes the number of spatial degrees of freedom, i.e., $I := \text{card}(\mathcal{V})$.

For the sake of brevity, we omit discussing the performance of the non-reflecting boundary conditions described in Appendix. Overall, the non-reflecting boundary conditions work well and as expected; no issues were observed regarding significant feedback or violation of the invariant-domain.

5.2. Well-balancing tests

In this section, we verify the well-balancing properties of the numerical method.

5.2.1. At rest

To verify the well-balancing at rest, we adopt the three conical bump topography configuration introduced in [51] and initialize the water depth to $\mathsf{H}_0(\boldsymbol{x}) = \max(1.5\,\text{m} - z(\boldsymbol{x}), 0)$ so that part of the topography is submerged and some is exposed creating a shoreline. The computational domain is set to $D = [0, 75\,\text{m}] \times [0, 30\,\text{m}]$ with slip boundary conditions. To make the problem slightly more challenging, we apply some distortion to the mesh since most realistic topographical data and respective meshes might not be uniform. We run the simulations until final time $T = 100\,\text{s}$ with CFL 0.9 using RK(3, 3; 1) and RK(3, 3; $\frac{1}{3}$). As shown in Fig. 1, no special treatment is done to align the shoreline with the mesh throughout the domain. We report the $L^\infty(T)$-norm of the error on the water depth for two meshes in Table 3. Inspection of the table shows that the method is indeed well-balanced even when the shoreline does not coincide with the mesh, which is a key improvement over the method proposed in [17, §5&6].

Table 3
At-rest well-balancing results.

I	RK(3, 3; 1)	RK(3, 3; $\frac{1}{3}$)
4225	1.33×10^{-15}	9.44×10^{-14}
16 641	1.33×10^{-15}	2.12×10^{-13}

Table 4
Steady flow over inclined plane well-balancing results.

I	RK(3, 3; 1)	RK(3, 3; $\frac{1}{3}$)
513	6.617×10^{-14}	5.300×10^{-13}
1025	1.642×10^{-14}	6.257×10^{-13}

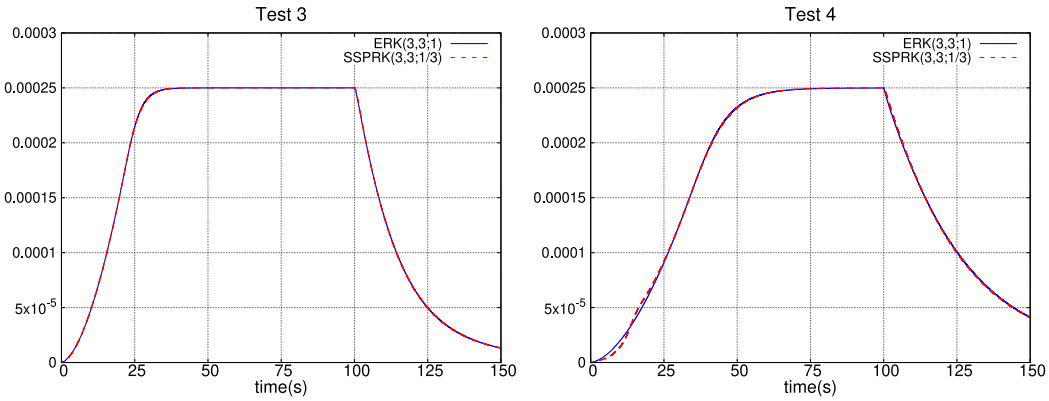


Fig. 2. Comparison of discharge at outlet boundary ($x = 2.5\,\mathrm{m}$) for the "Test 3" and "Test 4" configurations from Table 2 in [5].

5.2.2. Steady flow over inclined plane with friction

We now test the well-balancing property for a steady flow over an inclined plane with Gauckler-Manning friction. The specific configuration that we consider is that proposed in [5, Sec. 4.1] titled "Example 1" (Test 2). The domain is set to $D = (0, 25\,\mathrm{m})$ with Dirichlet conditions on the left for inflow, and non-reflecting boundary conditions on the right for dynamic outflow. The topography profile is defined by $z(x) = -bx$. The unit discharge is initialized with $q(x) = q_0$. The initial and exact solution for the depth is given by $\mathsf{h}(x) \equiv \mathsf{h}_0 = (\frac{n^2 q_0^2}{b})^{3/10}$ where n is the Gauckler-Manning friction coefficient, see (2.5) in [5]. The coefficients are set to $b = 0.01, q_0 = 0.1\,\mathrm{m}^2/\mathrm{s}, n = 0.02\,\mathrm{m}^{-1/3}\mathrm{s}$ which gives approximately $\mathsf{h}_0 \approx 0.095\,635\,\mathrm{m}$. We run the simulations until final time $T = 100\,\mathrm{s}$ with CFL 0.5 using RK$(3, 3; 1)$ and RK$(3, 3; \frac{1}{3})$. We report the $\delta^\infty(T)$ error for two meshes in Table 4. Well-balancing is again achieved in this case.

5.2.3. Rainfall over inclined plane with friction

We now test the method's ability to handle both rainfall and friction effects as sources. We again use the inclined plane bathymetry from the previous section, but now follow the configuration in [5, Sec. 4.1] titled "Example 3". Here, the initial configuration is set to dry with a constant rain source $R(x) = 1 \times 10^{-4}\,\mathrm{ms}^{-1}$ active in the interval $0 \le t \le 100\,\mathrm{s}$. The specific test cases we reproduce are "Test 3" and "Test 4" from Table 2 in [5]. The domain is set to $D = (0, 2.5\,\mathrm{m})$ with slip conditions on the left and do nothing boundary conditions on the right. For both tests, we run the simulations until final time $T = 150\,\mathrm{s}$ with CFL 0.5 using RK$(3, 3; 1)$ and RK$(3, 3; \frac{1}{3})$. The discharge is measured over time at the right boundary of the domain for each test. We report the time history of the discharge in Fig. 2. We observe results comparable to those reported in [5, Fig. 5]. For the results obtained with RK$(3, 3; \frac{1}{3})$ in Test 4, there are some slight oscillations at the beginning of the simulation.

5.2.4. Smooth water height perturbation over 2D smooth topography

We now test the scheme's ability to handle small perturbations around a state that is at rest. This particular benchmark is a modification of the one introduced in [52, Ex. 7.2] and reproduced in several papers (e.g., see [53,54] and references therein). The setup is as follows. The computational domain is defined to be $D = (-2, 2\,\mathrm{m}) \times (0, 1\,\mathrm{m})$ with "dynamic" conditions on the left/right boundaries and slip conditions on the top/bottom boundaries. The bathymetry is an ellipsoidal bump defined by $z(\boldsymbol{x}) = 0.8 \exp\left(-5(x - 0.9)^2 - (y - 0.5)^2\right)$. The initial water depth is defined to be mostly at rest with a smooth perturbation at the origin: $\mathsf{h}(\boldsymbol{x}) = 1 + 0.01 \exp(-200x^2) - z(\boldsymbol{x})$.

We test three different methods to illustrate the advantages of achieving well-balancing. (i) The first method is the IDP well-balanced limited scheme described in the paper; (ii) The second method is the same scheme with a modified high-order flux (3.14) that is not fully well-balanced but still employs the convex limiting technique, i.e., we replace $\sum_{j\in\mathcal{J}(i)} \boldsymbol{c}_{ij} \mathsf{H}_i^n \mathsf{H}_j^n$ in (3.14) by $\sum_{j\in\mathcal{J}(i)} \boldsymbol{c}_{ij} \frac{1}{2}(\mathsf{H}_j^n)^2$; (iii) The third scheme uses the modified high-order flux that is not fully well-balanced but no convex limiting is applied. We test each method on a single coarse mesh with mesh size equal to 1 cm in both directions as in [53, Sec. 4.2.3]. As a reference, we also run one simulation with the original scheme but on a mesh with mesh size 0.1 cm in the x-direction and 1 cm in the y-direction. The final time is 0.25 s. The simulations are done with the ERK$(4, 3; 1)$ time-stepping scheme with CFL 0.9. We show in the left panel of Fig. 3 the water elevation $\mathsf{h} + z$ along the segment $\{-2 \le x \le 2, y = 0.5\,\mathrm{m}\}$ for each method. We also show in the figure the scaled topography profile $z/200+0.9955$ to have a visual reference of the topography. The right panel in Fig. 3 shows a zoom of the four solutions close to maximum height of the topography map. We see that the well-balanced IDP solution (blue) is very close to the reference solution (red dashed) throughout the domain. The unlimited unbalanced solution (green) develops a spurious bump whose shape reproduces the topography. The limited and unbalanced solution (brown) is significantly closer to the reference solution than the unlimited one, but we nevertheless observe that the topography induces spurious velocities ahead of the right-moving wave which limiting (with relaxation) cannot entirely remove. This series of tests clearly demonstrate the benefits of using a high-order flux that is well-balanced in regions where the water height is bounded away from zero. Note that the shape of the left-moving wave is well approximated by the methods, thereby demonstrating that the differences observed on the right-moving wave are only due to the well-balancing of the flux or lack thereof. These tests also demonstrate the benefit of limiting the velocity when the flux is not well-balanced. In conclusion, the well-balanced IDP method proposed in the paper behaves exactly as advertised for this test case.

5.2.5. Smooth water height perturbation over 2D smooth topography with dry states

We now perform a modified version of the previous test case and introduce dry states in the problem. This will test the scheme's ability to handle small perturbations around dry states. The computational domain and bathymetry profile are the same as in the previous section. We modify the initial water depth as follows: $\mathsf{h}(\boldsymbol{x}) = \max(0.795 + 0.01\exp(-200(x - 0.5)^2) - z(\boldsymbol{x}), 0)$ so that the still water depth creates a shoreline with the bathymetry. We perform the test on the coarse mesh described above with the IDP well-balanced limited scheme described in the paper. The simulations are done with the ERK$(4, 3; 1)$ time-stepping scheme with CFL 0.9. Note that on this coarse mesh the height difference between the maximum of the topography bump and still water depth is $\frac{\Delta_x}{2}$ and the amplitude of the original perturbation is Δ_x (recall $\Delta_x = 1\,\mathrm{cm}$). We show in Fig. 4 the water elevation $\mathsf{h} + z$ along the segment $\{-0.5 \le x \le 2, y = 0.5\,\mathrm{m}\}$. For completeness, we compare the coarse solution with the solution on the finer mesh

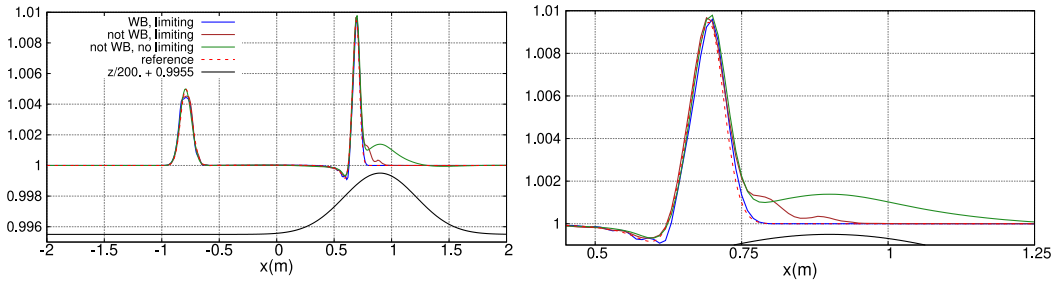


Fig. 3. Smooth perturbation over 2D smooth topography. Water elevation $\mathsf{h}+z$ along the segment $\{-2 \le x \le 2, y = 0.5\,\mathrm{m}\}$ at time $t = 0.25\,\mathrm{s}$ for different methods.

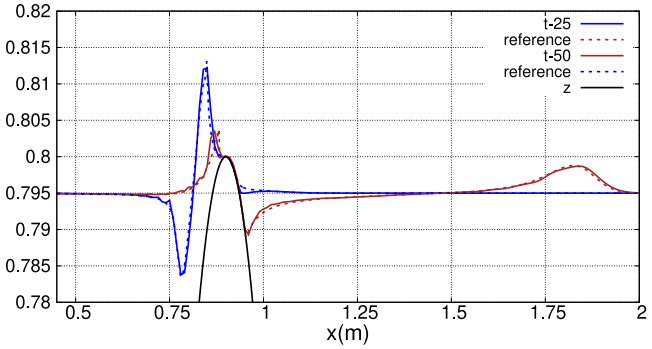


Fig. 4. Smooth perturbation over 2D smooth topography with dry states. Water elevation $\mathsf{h}+z$ along the segment $\{-2 \le x \le 2, y = 0.5\,\mathrm{m}\}$ at time $t = 0.25\,\mathrm{s}$ and $t = 0.50\,\mathrm{s}$.

Table 5
Error $\delta^1(T)$ and convergence rates for smooth vortex test with CFL 0.25.

I	RK(3, 3; 1)	Rate	RK(3, 3; $\frac{1}{3}$)	Rate
1089	3.579×10^{-3}	–	3.572×10^{-3}	–
4225	6.281×10^{-4}	2.51	6.274×10^{-4}	2.51
16 641	8.414×10^{-5}	2.90	8.399×10^{-5}	2.90
66 049	1.095×10^{-5}	2.94	1.094×10^{-5}	2.94
263 169	1.425×10^{-6}	2.94	1.424×10^{-6}	2.94
1 050 625	2.811×10^{-7}	2.34	1.919×10^{-7}	2.89

($\Delta_x = 0.1\,\mathrm{cm}$) as in the previous test case. We observed no numerical issues with perturbations around dry states and we can see that the overall behavior of the coarse solution matches closely to the reference fine solution.

5.3. Convergence tests

In this section, we verify the accuracy of the proposed method. For the sake of brevity, we only report results in two space dimensions since we observe similar behavior in one space dimension.

5.3.1. Smooth vortex

We now demonstrate the convergence of the method with a smooth analytical solution of the shallow water equations. This benchmark is a divergence-free vortex adapted (and slightly modified) from [12, Sec. 2.3] which mimics geophysical flows [53]. Let $(\mathsf{h}_\infty, \boldsymbol{v}_\infty)$ be the far-field state. Then, the analytical solution is defined as follows:

$$\mathsf{h}(\boldsymbol{x},t) = \mathsf{h}_\infty - \frac{1}{2\,g r_0^2}\psi(\overline{\boldsymbol{x}})^2, \tag{5.2a}$$

$$\boldsymbol{v} := \boldsymbol{v}_\infty + \delta\boldsymbol{v}, \tag{5.2b}$$

$$\delta\boldsymbol{v}(\boldsymbol{x},t) := \Big(\partial_{x_2}\psi(\overline{\boldsymbol{x}}), -\partial_{x_1}\psi(\overline{\boldsymbol{x}})\Big)^{\mathsf{T}}, \tag{5.2c}$$

Table 6
Error $\delta^1(T)$ and convergence rates for the test configuration with a planar surface flow in a paraboloid-shaped basin.

I	RK(3, 3; 1)	Rate	RK(3, 3; $\frac{1}{3}$)	Rate
1089	2.217×10^{-1}	–	2.268×10^{-1}	–
4225	6.328×10^{-2}	1.81	6.473×10^{-2}	1.81
16 641	1.723×10^{-2}	1.88	1.785×10^{-2}	1.86
66 049	5.106×10^{-3}	1.75	5.359×10^{-3}	1.74
263 169	1.740×10^{-3}	1.55	1.834×10^{-3}	1.55
1 050 625	7.041×10^{-4}	1.31	7.386×10^{-4}	1.31

Table 7
Efficiency comparison of various time-stepping schemes. We report the total number of cycles for a full Runge-Kutta step to reach final time $T = 0.5\,\mathrm{s}$ with CFL 0.2, the average throughput measured in million Q_1-mesh points per second (MQ/s) for a single Runge-Kutta substep (consisting of a single forward Euler step) and the total runtime to reach the final simulation time.

Method	# cycles	Throughput	Runtime
RK(2, 2; $\frac{1}{2}$)	1019	1.3564 MQ/s	99.64 s
RK(2, 2; 1)	510	1.2762 MQ/s	52.99 s
RK(3, 3; $\frac{1}{3}$)	1019	1.3578 MQ/s	149.20 s
RK(3, 3; 1)	340	1.2511 MQ/s	54.01 s
RK(4, 3; 1)	255	1.2312 MQ/s	54.91 s
RK(5, 4; 1)	204	1.1494 MQ/s	58.79 s

with $\overline{\boldsymbol{x}} := \boldsymbol{x} - \boldsymbol{x}^0 - \boldsymbol{v}_\infty t$ and $\psi(\boldsymbol{x}) := \frac{\beta}{2\pi}\exp(\frac{1}{2}(1 - \frac{\|\boldsymbol{x}\|^2}{r_0^2}))$. Here, $\boldsymbol{x}^0$ can be thought of as the center of the vortex, β the vortex strength and r_0 the radius of the vortex. The parameters are set to $\mathsf{h}_\infty = 2\,\mathrm{m}$, $\beta = 2, r_0 = 1\,\mathrm{m}$, $\boldsymbol{v}_\infty = (1,1)\mathrm{ms}^{-1}$. The computational domain is set to $D = (-6, 6\,\mathrm{m}) \times (-6, 6\,\mathrm{m})$ with Dirichlet boundary conditions. We set the final time to $T = 2\,\mathrm{s}$. The time-stepping is performed with RK(3, 3; 1) and RK(3, 3; $\frac{1}{3}$) with CFL 0.25. We report the consolidated $\delta^\infty(T)$ error and rates in Table 5. We observe close to third order accuracy in time and space. The super-convergence in space is compatible with the theoretical result from [39, Prop. A.1].

5.3.2. Planar surface flow in paraboloid-shaped basin

We now demonstrate the convergence of the method with Thacker's planar surface flow in paraboloid-shaped basin [55]. The problem consists of a free-surface moving in a periodic motion inside a paraboloid-shaped basin. The moving shoreline is circular at all times. The precise configuration we use is the one introduced in [56, Sec. 4.2.2] subsection "Planar surface in a paraboloid:" The computational domain is defined as $D = [0, 4\,\mathrm{m}] \times [0, 4\,\mathrm{m}]$ with slip boundary conditions. The theoretical period of the motion is $2\pi/\sqrt{2\,g\mathsf{h}_0}$ with $\mathsf{h}_0 = 0.1\,\mathrm{m}$. The final time is three periods, approximately $T = 13.457\,104\,40\,\mathrm{s}$. The time-stepping is performed with RK(3, 3; 1) and RK(3, 3; $\frac{1}{3}$) with CFL 0.5. We report the consolidated $\delta^1(T)$ error and rates in Table 6. We observe a convergence rate ranging from 1.8 to 1.3, which is consistent with what is reported in the literature, see e.g., [15, Sec. 4.3] and [57, Sec. 3.5].

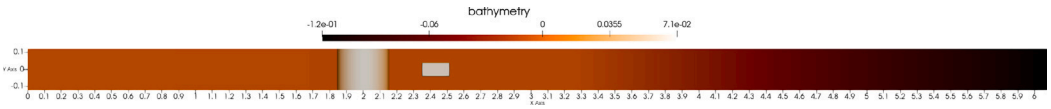


Fig. 5. Top view representation for the "G2-S.2" test case.

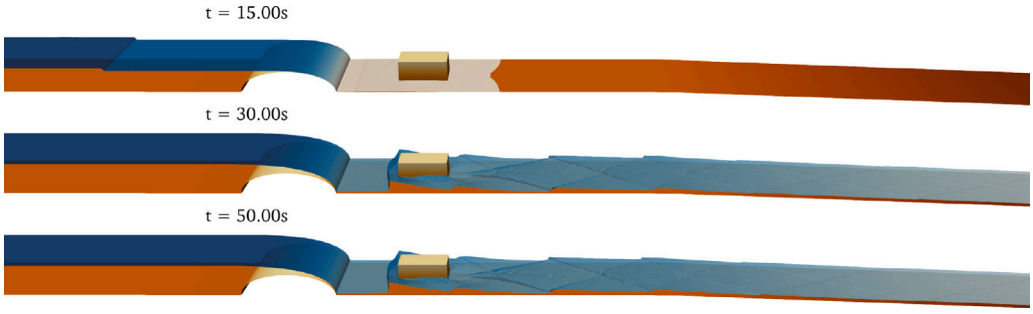


Fig. 6. Time snapshots for "G2-S.2" showing water elevation and bathymetry.

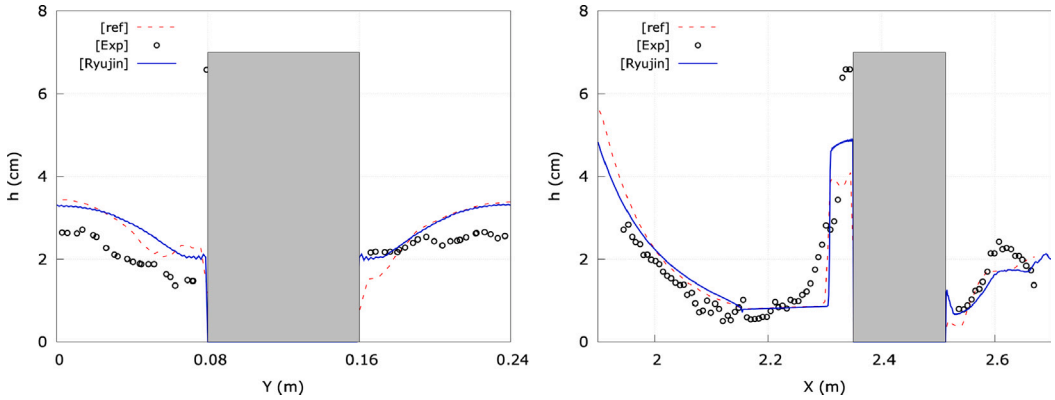


Fig. 7. Comparison of numerically computed water depth (solid blue line) for the "G2-S.2" configuration along the two sections with experimental data (black circles) and corresponding simulation data from [58] (red dashed line).

5.4. Small-scale laboratory experiments

In this section, we simulate two small-scale laboratory experiments described in [58]. The goal of the experiments was to provide validation data for shallow water solvers by studying complex steady and transient flume experiments. The experiments comprised of transcritical steady flow and dam-break flow around obstacles and complex beds. In this paper, we reproduce cases "G2-S.2" and "G3-D.1" described in Sections 4.3.1 and 4.4.2 in [58], respectively. We refer the reader to [58] for a detailed description of the experimental configuration. The setup can also be found in the source code for the `ryujin` software.

5.4.1. G2-S.2

The "G2-S.2" test case consists of a steady inflow discharge of $\mathsf{Q}_0 = 9.01\,\mathrm{m}^3/\mathrm{h}$ with the flume containing a semi-circular bump across its width followed by a rectangular obstacle placed at the center line of the flume. We give a top view representation of the setup in Fig. 5. Note that the discharge here is the volumetric flow rate. For our simulation, we use the unit flow discharge $q_0 = \mathsf{Q}_0/0.24\,\mathrm{m}$ which gives $q_0 = 0.0104\,\mathrm{m}^2/\mathrm{s}$ (here, the $0.24\,\mathrm{m}$ corresponds to the flume width). We reproduce this case using the computational domain $D = [0, 6.078\,\mathrm{m}] \times [-0.12, 0.12\,\mathrm{m}]$ with 928137 $\mathbb{Q}_1$ degrees of freedom (this corresponds to the mesh-size being roughly $1.25\,\mathrm{mm}$ in each direction). Note that we omit discretizing the tank reservoir since it is not needed for simulating steady inflow. The initial setup consists of a dry flume where the bottom/top boundaries are set to slip boundary conditions and the right boundary is set to non-reflecting boundary conditions for dynamic outflow. On the left boundary, we enforce the steady inflow discharge and do nothing boundary conditions for the water depth. We run the simulation with RK$(4, 3; 1)$ until $T = 50\,\mathrm{s}$ with CFL 0.9 to allow the flow to reach a steady state. We output four time snapshots for $t = \{1.5 \times 10^1\,\mathrm{s}, 3.0 \times 10^1\,\mathrm{s}, 5.0 \times 10^1\,\mathrm{s}\}$ in Fig. 6.

In the experiments, the water depth was measured at two different sections: $x = 2.40\,\mathrm{m}$ (across width of flume spanning the rectangular obstacle) and $y = 0\,\mathrm{m}$ (centerline of the flume). In Fig. 7, we compare the numerical output of our simulations along the sections and compare with the experimental data as well as the simulation data reported in [58]. Overall, our simulation compares well with the experiments and simulations from [58]. The discrepancies between the numerical simulations are due to mesh resolution differences. The discrepancies with the experiments show the shortcomings of the shallow water equations and that, short to solving the Navier–Stokes equations, a higher-fidelity model is required.

5.4.2. G3-D.1

The case "G3-D.1" consists of a dam-break flow with height $\mathsf{H}_0 = 0.055\,\mathrm{m}$ in the reservoir and the flume containing two semi-circular Venturi constriction elements followed by rectangular obstacle placed at the center line of the flume. We give a top view representation of the setup in Fig. 8. We reproduce this case using the computation domain $D = D_{\mathrm{res}} \cup D_{\mathrm{flume}}$ where $D_{\mathrm{res}} = [-1.58, 0\,\mathrm{m}] \times [-0.405, 0.405\,\mathrm{m}]$ and $D_{\mathrm{flume}} = [0, 6.078\,\mathrm{m}] \times [-0.12, 0.12\,\mathrm{m}]$ with 1753793 $\mathbb{Q}_1$ degrees of freedom (this corresponds to the mesh-size being roughly $1.25\,\mathrm{mm}$ in each direction). The flume is initially dry. The slip boundary condition

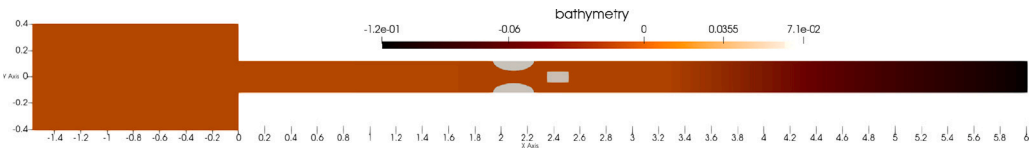


Fig. 8. Top view representation for "G3-D.1" case.

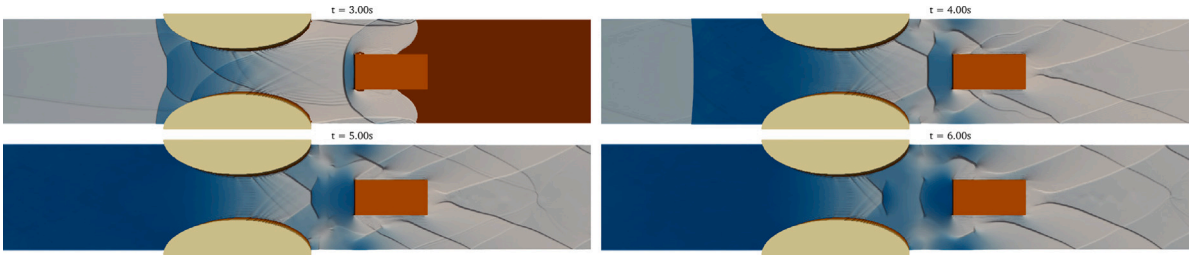


Fig. 9. Time snapshots for "G3-D.1" showing water elevation and bathymetry.

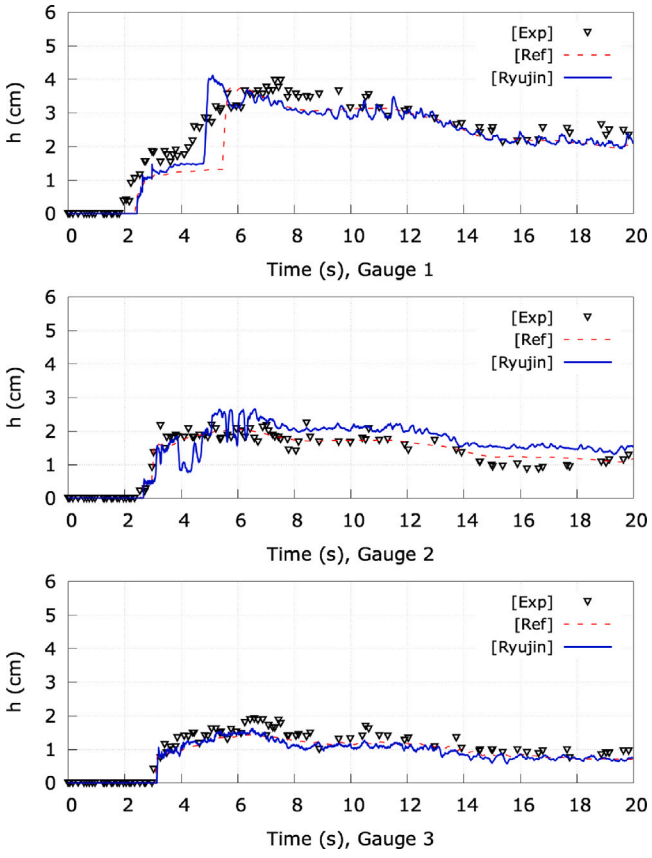


Fig. 10. Temporal series over $t \in [0, 20\,\mathrm{s}]$ comparing numerical water depth (blue solid line), experimental data (black triangles) and simulation data from [58] (red dashed line). From top to bottom: Gauge 1, Gauge 2, Gauge 3.

is enforced on all the boundaries except the right-most one. Non-reflecting boundary conditions are enforced on the right boundary for dynamic outflow. We run the simulation with RK$(4, 3; 1)$ until $T = 20\,\mathrm{s}$ with CFL 0.9. We output four time snapshots of the water elevation for $t = \{3\,\mathrm{s}, 4\,\mathrm{s}, 5\,\mathrm{s}, 6\,\mathrm{s}\}$ in Fig. 9.

In the experiments reported in [58], three wave gauges were placed in the basin to measure the water depth over the duration of the experiment. The specific locations of the gauges are given by: "Gauge 1" (225 cm, 12 cm); "Gauge 2" (240 cm, 20 cm); "Gauge 3" (260 cm, 12 cm). In Fig. 10, we compare the numerical output of our simulations with the experimental data and the simulation data reported in [58]. Overall, our simulation compares well with the experiment and simulations from [58].

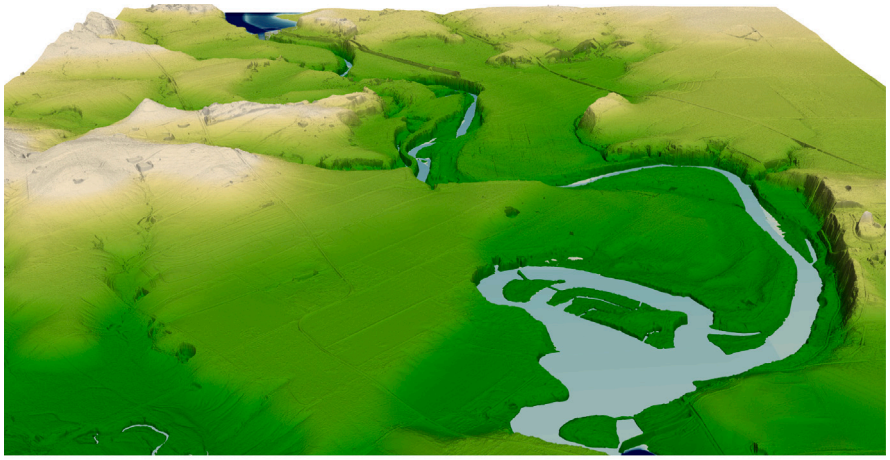
5.5. Efficiency tests

We now report a quick efficiency test to compare various time-stepping techniques. We choose the smooth vortex benchmark described in 5.3.1 and run until final time $T = 0.5\,\mathrm{s}$ with CFL 0.2 with the mesh composed of 66,049 $\mathbb{Q}_1$ degrees of freedom. Each simulation is performed on a single rank and single thread on a laptop computer. In Table 7, we report the results of our tests. We see that the overall efficiency of each method is directly proportional to the efficiency coefficient c_{eff}.

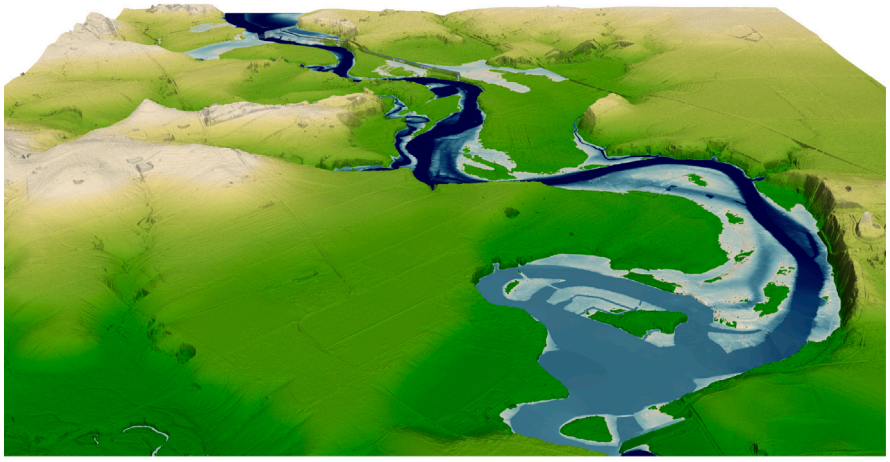
5.6. High-fidelity simulation

Lastly we perform a high-fidelity dam break simulation with the shallow water equations using realistic topography data. To this end we linked the `ryujin` software to the *Geospatial Data Abstraction Library* (`GDAL`)[2] in order to read in digital elevation models (DEMs). The DEM considered here was obtained from the *United States Geological Survey 3D Elevation Program* [59] via *OpenTopography*[3] and shows a portion of Lake Dunlap and the Guadalupe river in the state of Texas with a spatial resolution of $1\,\mathrm{m} \times 1\,\mathrm{m}$. We simulate the breaking of the dam at Lake Dunlap. The simulation was performed on the computational domain $D = [0, 7168\,\mathrm{m}] \times [0, 8192\,\mathrm{m}]$ which is the bounding box for the DEM data until final time $T = 2\,\mathrm{h}$. We use a Gauckler-Manning's friction source term with a roughness coefficient of $n = 0.025\,\mathrm{m}^{-1/3}\,\mathrm{s}$ and a gravity constant of $g = 9.81\,\mathrm{m\,s}^{-2}$. We set up an initial water column with $\mathsf{h} + z = 179.5\,\mathrm{m}$ (above sea level) for the upper basin and $\mathsf{h} + z = 163.5\,\mathrm{m}$ slowly sloping down to 161.3 m for the river bed; both with zero initial velocity. On the northern boundary of the domain we enforce Dirichlet conditions with $\mathsf{h} + z = 182.5\,\mathrm{m}$ which ensure that the upper basin is always filled. We emphasize that we chose this flow configuration purely for demonstration purposes and that it is not particularly realistic: we do not factor in the finite amount of water stored in the upper basin (as it is in fact not fully simulated); we make no attempt of creating a realistic initial configuration of the downstream river with correct water height and stream velocity; and the DEM does not contain bathymetry information of the river bed.

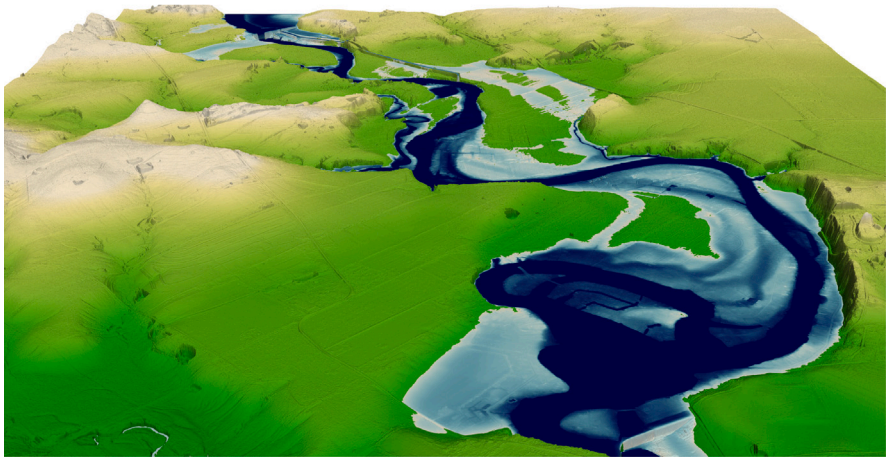
[2] https://gdal.org.
[3] https://opentopography.org.



(a) $t = 0h$



(b) $t = 1h$



(c) $t = 2h$

Fig. 11. Temporal snapshots for the high-fidelity simulation of a dam break with 58,735,617 $\mathbb{Q}_1$ degrees of freedom per component. The figures show a three dimensional rendering of the bathymetry $z(\boldsymbol{x})$ with a color scale ranging from dark green to light ocre, and the water surface $\mathsf{h} + z$ with a color scale ranging from dark blue (large h) to light blue (small h). The vertical direction is scaled by a factor 10.

The computation was performed with 58,735,617 $\mathbb{Q}_1$ degrees of freedom per component on 768 ranks (and 2 threads per rank) on the Whistler cluster at Texas A&M University using single-precision floating point arithmetic. We performed 152,637 RK$(3, 3; 1)$ steps with a chosen CFL number of 0.9 resulting in an average time step size of $\tau_{\text{ERK}} \approx 4.72 \times 10^{-2}$ s. The total CPU time summed over all ranks was

about 9820 h with an average per-rank throughput of about 1.52 MQ/s, where MQ/s stands for million $\mathbb{Q}_1$-mesh point updates per second for a single Runge–Kutta substep (consisting of a single forward-Euler step). We recorded a total runtime of approximately 6.43 h (wall time) which equates to a combined throughput over all ranks of about 1159.3 MQ/s.

We visualize in Fig. 11 the simulation results for temporal snapshots at initial time $t = 0$ h, at $t = 1$ h, and at $t = 2$ h. The figure shows a three dimensional rendering of the bathymetry $z(\boldsymbol{x})$ with a color scale ranging from dark green to light ocre where the bathymetry is scaled by a factor 10. Similarly, the water surface $\mathsf{h} + z$ is overlayed with the same scaling factor and with a color scale ranging from dark blue to light blue for large to small values of h.

6. Conclusion

In this work, we have provided a high-order space and time approximation of the shallow water equations with external sources on unstructured meshes. The numerical method was shown to be invariant-domain preserving and well-balanced with respect to rest states whether the shoreline is aligned with the mesh or not. The method was also shown to be robust with respect to external source terms. This work will be the stepping stone for various multi-physics extensions of the shallow water equations like the Serre–Green–Naghdi Equations which account for dispersive water waves or subsurface models such as Richard's equation.

CRediT authorship contribution statement

Jean-Luc Guermond: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Matthias Maier:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Eric J. Tovar:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank Sergio Martinez at the University of Zaragoza for his support in providing the data reported in [58] as well as the `Gnuplot` files for postprocessing the data. They also thank the two reviewers who, by their numerous constructives comments and questions, helped improving the manuscript significantly.

Appendix. Boundary conditions

In this section, we describe how the boundary conditions are enforced for the IDP explicit Runge–Kutta schemes above. Recall that the Shallow Water Equations with flat topography (and no external source terms) are equivalent to the isentropic compressible Euler Equations when the adiabatic index γ is 2. Thus, the details in this section are a direct modification of the work seen in [23] where it is shown how to enforce "reflecting" (i.e., slip or wall) and "non-reflecting" boundary conditions for the isentropic Euler Equations.

A.1. Preliminaries

Boundary conditions are enforced by post-processing the approximation at the end of each stage of the ERK-IDP algorithm. We consider two types of boundary conditions: (i) Reflecting conditions, also called "slip" or "wall": $\boldsymbol{v}\cdot\boldsymbol{n} = 0$; (ii) Non-reflecting conditions. Let $\partial D_{\rm r} \subset \partial D$ be the boundary where reflecting conditions are enforced. Let $\partial D_{\rm nr}$ denote the complement of $\partial D_{\rm r}$ in ∂D where non-reflecting conditions will be enforced. Let $\mathcal{V}_{\rm r}^{\partial} \subset \mathcal{V}^{\partial}$ be the collection of all the boundary degrees of freedom i such that $\varphi_{i|\partial D_{\rm r}} \not\equiv 0$. Let $\mathcal{V}_{\rm r}^{\partial} \subset \mathcal{V}^{\partial}$ be the collection of all boundary degrees of freedom i such that $\varphi_{i|\partial D_{\rm nr}} \not\equiv 0$. We define the normal vectors associated with the degrees of freedom in $\mathcal{V}_{\rm r}^{\partial}$ and $\mathcal{V}_{\rm nr}^{\partial}$, respectively:

$$\boldsymbol{n}_i^{\rm r} := \frac{\int_{\partial D_{\rm r}} \varphi_i \boldsymbol{n}\,{\rm d}s}{\|\int_{\partial D_{\rm r}} \varphi_i \boldsymbol{n}\,{\rm d}s\|_{\ell^2}}, \qquad \boldsymbol{n}_i^{\rm nr} := \frac{\int_{\partial D_{\rm nr}} \varphi_i \boldsymbol{n}\,{\rm d}s}{\|\int_{\partial D_{\rm nr}} \varphi_i \boldsymbol{n}\,{\rm d}s\|_{\ell^2}}. \tag{A.1}$$

In the following two sections, the symbol $\mathbf{U}$ denotes the state obtained at the end of each stage. The post-processed state is denoted by $\mathbf{U}^{\rm P}$.

A.2. Reflecting boundary conditions

Let $i \in \mathcal{V}_{\rm r}^{\partial}$ and let $\mathbf{U}_i = (\mathsf{H}_i, \mathbf{Q}_i)^{\mathsf{T}}$. Reflecting boundary conditions are enforced at the ith degree of freedom by setting:

$$\mathbf{U}_i^{\rm P} := (\mathsf{H}_i, \mathbf{Q}_i - (\mathbf{Q}_i\cdot\boldsymbol{n}_i^{\rm r})\boldsymbol{n}_i^{\rm r})^{\mathsf{T}}. \tag{A.2}$$

A.3. Non-reflecting boundary conditions

We now consider non-reflecting boundary conditions at $i \in \mathcal{V}_{\rm nr}^{\partial}$ based on Riemann invariants. We note that the idea of working with the Riemann invariants of the Shallow Water Equations for the use of boundary conditions is a common approach in the literature (see: [60, Sec. 6]). For notational simplicity, we assume the following states are at the ith degree of freedom and drop the subscript notation. Here, $\boldsymbol{n} := \boldsymbol{n}_i^{\rm nr}$.

Let $\mathsf{h} := \mathsf{h}(\mathbf{U})$, $\mathbf{Q} := \boldsymbol{q}(\mathbf{U})$ and set:

$$\mathbf{V} := \mathsf{h}^{-1}\mathbf{Q}, \qquad \mathsf{V}_n := \mathbf{V}\cdot\boldsymbol{n}, \qquad \mathbf{V}^{\perp} := \mathbf{V} - (\mathbf{V}\cdot\boldsymbol{n})\cdot\boldsymbol{n}, \qquad a := \sqrt{g\mathsf{h}}. \tag{A.3}$$

Assume that the topography is flat and that there are no external source terms. Then, the characteristic variables and characteristic speeds for the one-dimensional system, $\partial_t \boldsymbol{u} + \partial_x(\mathbb{f}(\boldsymbol{u})\boldsymbol{n}) = 0$, are:

$$\underbrace{\begin{cases}\lambda_1(\mathbf{U},\boldsymbol{n}) := \mathsf{V}_n - a\\ \mathsf{R}_1(\mathbf{U},\boldsymbol{n}) := \mathsf{V}_n - 2a,\end{cases}}_{\text{multiplicity } 1} \qquad \underbrace{\begin{cases}\lambda_2(\mathbf{U},\boldsymbol{n}) := \mathsf{V}_n\\ \mathbf{V}^{\perp},\end{cases}}_{\text{multiplicity } d-1}$$

$$\underbrace{\begin{cases}\lambda_3(\mathbf{U},\boldsymbol{n}) := \mathsf{V}_n + a\\ \mathsf{R}_3(\mathbf{U},\boldsymbol{n}) := \mathsf{V}_n + 2a.\end{cases}}_{\text{multiplicity } 1} \tag{A.4}$$

Note in passing there are only two Riemann invariants for the Shallow Water Equations, but we use the notation R_1 and R_3 so that they correspond directly to the eigenvalues. We consider four different cases depending on the type of flow at the boundary:

(i) torrential inflow: $\mathsf{V}_n < 0$ and $a < |\mathsf{V}_n|$ $\quad \lambda_1 \le \lambda_2 \le \lambda_3 < 0$,
(ii) torrential outflow: $0 \le \mathsf{V}_n$ and $a \le |\mathsf{V}_n|$ $\quad 0 \le \lambda_1 \le \lambda_2 \le \lambda_3$,
(ii) fluvial inflow: $\mathsf{V}_n < 0$ and $|\mathsf{V}_n| < a$ $\quad \lambda_1 \le \lambda_2 < 0 \le \lambda_3$,
(iv) fluvial outflow: $0 \le \mathsf{V}_n$ and $a < |\mathsf{V}_n|$ $\quad \lambda_1 < 0 \le \lambda_2 \le \lambda_3$.

Note that the nomenclature "torrential" is equivalent to "supersonic" and "fluvial" is equivalent to "subsonic" in the context of gas dynamics. We assume that outside the domain D, we have at hand some Dirichlet data $\mathbf{U}^{\rm D} := (\mathsf{h}^{\rm D}, \mathbf{Q}^{\rm D})^{\mathsf{T}}$. Just as in [23], we are going to postprocess the solution $\mathbf{U}$ so that the characteristic variables of the post-processed

state $\mathbf{U}^{\mathrm{P}}$ associated with the in-coming eigenvalues match those of the prescribed Dirichlet data $\mathbf{U}^{\mathrm{D}}$, while leaving the out-going characteristics unchanged. More precisely, the strategy consists of finding $\mathbf{U}^{\mathrm{P}}$ so that the following holds:

$$\mathsf{R}_l(\mathbf{U}^{\mathrm{P}}) = \begin{cases} \mathsf{R}_l(\mathbf{U}^{\mathrm{D}}) & \text{if } \lambda_l(\mathbf{U},\boldsymbol{n}^{\mathrm{nr}}) < 0, \\ \mathsf{R}_l(\mathbf{U}) & \text{if } 0 \le \lambda_l(\mathbf{U},\boldsymbol{n}^{\mathrm{nr}}), \end{cases} \quad l \in \{1,3\}, \tag{A.5a}$$

$$(\mathbf{V}^{\mathrm{P}})^{\perp} = \begin{cases} (\mathbf{V}^{\mathrm{D}})^{\perp} & \text{if } \lambda_2(\mathbf{U},\boldsymbol{n}^{\mathrm{nr}}) < 0, \\ \mathbf{V}^{\perp} & \text{if } 0 \le \lambda_2(\mathbf{U},\boldsymbol{n}^{\mathrm{nr}}). \end{cases} \tag{A.5b}$$

We now solve the above system for each of the four flow configurations mentioned above.

Torrential inflow: Assume that $\lambda_1(\mathbf{U},\boldsymbol{n}) \le \lambda_2(\mathbf{U},\boldsymbol{n}) \le \lambda_3(\mathbf{U},\boldsymbol{n}) < 0$. Since all the characteristics are entering the computational domain, the postprocessing consists of replacing $\mathbf{U}$ by $\mathbf{U}^{\mathrm{D}}$:

$$\mathbf{U}^{\mathrm{P}} = \mathbf{U}^{\mathrm{D}}. \tag{A.6}$$

Torrential outflow: Assume that $0 \le \lambda_1(\mathbf{U},\boldsymbol{n}) \le \lambda_2(\mathbf{U},\boldsymbol{n}) \le \lambda_3(\mathbf{U},\boldsymbol{n})$. Since all the characteristics are exiting the computational domain, the postprocessing consists of doing nothing:

$$\mathbf{U}^{\mathrm{P}} = \mathbf{U}. \tag{A.7}$$

Fluvial inflow: Assume that $\lambda_1(\mathbf{U},\boldsymbol{n}) \le \lambda_2(\mathbf{U},\boldsymbol{n}) < 0 < \lambda_3(\mathbf{U},\boldsymbol{n})$. Then, $\mathbf{U}^{\mathrm{P}}$ is obtained by solving the following system:

$$\mathsf{R}_1(\mathbf{U}^{\mathrm{P}}) = \mathsf{R}_1(\mathbf{U}^{\mathrm{D}}), \qquad (\mathbf{V}^{\mathrm{P}})^{\perp} = (\mathbf{V}^{\mathrm{D}})^{\perp}, \qquad \mathsf{R}_3(\mathbf{U}^{\mathrm{P}}) = \mathsf{R}_3(\mathbf{U}). \tag{A.8}$$

This gives that $\mathsf{V}_n^{\mathrm{P}} = \frac{1}{2}\left(\mathsf{R}_1(\mathbf{U}^{\mathrm{D}}) + \mathsf{R}_3(\mathbf{U})\right)$ and $4a^{\mathrm{P}} = \left(\mathsf{R}_3(\mathbf{U}) - \mathsf{R}_1(\mathbf{U}^{\mathrm{D}})\right)$ $= \mathsf{V}_n + 2a - (\mathsf{V}_n^{\mathrm{D}} - 2a^{\mathrm{D}})$. Since in this flow configuration $\mathsf{V}_n + 2a > 0$, for a^{P} to be positive it must be that: $\mathsf{V}_n^{\mathrm{D}} \le 2a^{\mathrm{D}}$ which is an admissibility condition on the Dirichlet data. Finally, the postprocessing for a fluvial inflow boundary condition consists of setting the solution $\mathbf{U}^{\mathrm{P}}$ to:

$$\mathsf{h}^{\mathrm{P}} = \frac{1}{g}(a^{\mathrm{P}})^2 = \frac{1}{g}\left(\frac{\mathsf{R}_3(\mathbf{U}) - \mathsf{R}_1(\mathbf{U}^{\mathrm{D}})}{4}\right)^2, \tag{A.9a}$$

$$\mathbf{Q}^{\mathrm{P}} = \mathsf{h}^{\mathrm{P}} \times \left((\mathbf{V}^{\mathrm{D}})^{\perp} + \mathsf{V}_n^{\mathrm{P}}\boldsymbol{n}\right), \quad \text{with} \quad \mathsf{V}_n^{\mathrm{P}} = \frac{1}{2}\left(\mathsf{R}_1(\mathbf{U}^{\mathrm{D}}) + \mathsf{R}_3(\mathbf{U})\right). \tag{A.9b}$$

Fluvial outflow: Assume that $\lambda_1(\mathbf{U},\boldsymbol{n}) < 0 < \lambda_2(\mathbf{U},\boldsymbol{n}) \le \lambda_3(\mathbf{U},\boldsymbol{n})$. Then, $\mathbf{U}^{\mathrm{P}}$ is obtained by solving the following system:

$$\mathsf{R}_1(\mathbf{U}^{\mathrm{P}}) = \mathsf{R}_1(\mathbf{U}^{\mathrm{D}}), \qquad (\mathbf{V}^{\mathrm{P}})^{\perp} = \mathbf{V}^{\perp}, \qquad \mathsf{R}_3(\mathbf{U}^{\mathrm{P}}) = \mathsf{R}_3(\mathbf{U}). \tag{A.10}$$

Notice now that only one Dirichlet condition is prescribed on the first characteristic since $\lambda_1 < 0$. Again, we have that $\mathsf{V}_n^{\mathrm{P}} = \frac{1}{2}\left(\mathsf{R}_1(\mathbf{U}^{\mathrm{D}}) + \mathsf{R}_3(\mathbf{U})\right)$ and $4a^{\mathrm{P}} = \left(\mathsf{R}_3(\mathbf{U}) - \mathsf{R}_1(\mathbf{U}^{\mathrm{D}})\right)$. We also have the same admissibility condition on the Dirichlet data: $\mathsf{V}_n^{\mathrm{D}} \le 2a^{\mathrm{D}}$ for $a^{\mathrm{P}} > 0$. Finally, the postprocessing for a fluvial outflow boundary condition consists of setting the solution $\mathbf{U}^{\mathrm{P}}$ to:

$$\mathsf{h}^{\mathrm{P}} = \frac{1}{g}(a^{\mathrm{P}})^2 = \frac{1}{g}\left(\frac{\mathsf{R}_3(\mathbf{U}) - \mathsf{R}_1(\mathbf{U}^{\mathrm{D}})}{4}\right)^2, \tag{A.11a}$$

$$\mathbf{Q}^{\mathrm{P}} = \mathsf{h}^{\mathrm{P}} \times \left(\mathbf{V}^{\perp} + \mathsf{V}_n^{\mathrm{P}}\boldsymbol{n}\right), \quad \text{with} \quad \mathsf{V}_n^{\mathrm{P}} = \frac{1}{2}\left(\mathsf{R}_1(\mathbf{U}^{\mathrm{D}}) + \mathsf{R}_3(\mathbf{U})\right). \tag{A.11b}$$

Remark A.1 (*Conservation and Admissibility*)**.** The conservation and admissibility properties of the proposed boundary conditions are described in [23, Sec. 4.3.3]. □

Data availability

A high performance implementation of the algorithms discussed in this paper are freely available as part of the `ryujin` project [22, 23]. The source code repository is located at https://github.com/conservation-laws/ryujin. Parameter and configuration files for the numerical illustrations reported in Section 5 are made available upon request.

References

[1] Bermúdez A, Vázquez ME. Upwind methods for hyperbolic conservation laws with source terms. Comput & Fluids 1994;23(8):1049–71.

[2] Greenberg JM, Le Roux A-Y. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. SIAM J Numer Anal 1996;33(1):1–16.

[3] Noelle S, Xing Y, Shu C-W. High-order well-balanced finite volume WENO schemes for shallow water equation with moving water. J Comput Phys 2007;226(1):29–58.

[4] Xing Y, Shu C-W. A survey of high order schemes for the shallow water equations. J Math Study 2014;47(3):221–49.

[5] Chertock A, Cui S, Kurganov A, Wu T. Well-balanced positivity preserving central-upwind scheme for the shallow water system with friction terms. Internat J Numer Methods Fluids 2015;78(6):355–83.

[6] Bouchut F. Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources. Frontiers in mathematics, Basel: Birkhäuser Verlag; 2004.

[7] Audusse E, Bouchut F, Bristeau M-O, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. SIAM J Sci Comput 2004;25(6):2050–65.

[8] Bollermann A, Noelle S, Lukáčová-Medvidová M. Finite volume evolution Galerkin methods for the shallow water equations with dry beds. Commun Comput Phys 2011;10(2):371–404.

[9] Gallardo JM, Parés C, Castro M. On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas. J Comput Phys 2007;227(1):574–601.

[10] Kurganov A, Petrova G. A second-order well-balanced positivity preserving central-upwind scheme for the Saint–Venant system. Commun Math Sci 2007;5(1):133–60.

[11] Perthame B, Simeoni C. A kinetic scheme for the saint-venant system with a source term. Calcolo 2001;38(4):201–31.

[12] Ricchiuto M, Bollermann A. Stabilized residual distribution for shallow water simulations. J Comput Phys 2009;228(4):1071–115.

[13] Castro MJ, Semplice M. Third-and fourth-order well-balanced schemes for the shallow water equations based on the CWENO reconstruction. Internat J Numer Methods Fluids 2019;89(8):304–25.

[14] Hajduk H, Kuzmin D. Bound-preserving and entropy-stable algebraic flux correction schemes for the shallow water equations with topography. 2022, arXiv preprint arXiv:2207.07261.

[15] Liang Q, Marche F. Numerical resolution of well-balanced shallow water equations with complex source terms. Adv Water Resour 2009;32(6):873–84.

[16] Duran A, Marche F, Turpault R, Berthon C. Asymptotic preserving scheme for the shallow water equations with source terms on unstructured meshes. J Comput Phys 2015;287:184–206.

[17] Guermond J-L, Quezada de Luna M, Popov B, Kees C, Farthing M. Well-balanced second-order finite element approximation of the shallow water equations with friction. SIAM J Sci Comput 2018;40(6):A3873–901.

[18] Brodtkorb AR, Sætra ML, Altinakar M. Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation. Comput & Fluids 2012;55:1–12.

[19] Dietrich J, Zijlema M, Westerink J, Holthuijsen L, Dawson C, Luettich R, Jensen R, Smith J, Stelling G, Stone G. Modeling hurricane waves and storm surge using integrally-coupled, scalable computations. Coast Eng 2011;58(1):45–65.

[20] Delmas V, Soulaïmani A. Multi-GPU implementation of a time-explicit finite volume solver using CUDA and a CUDA-aware version of OpenMPI with application to shallow water flows. Comput Phys Comm 2022;271:108190.

[21] Caviedes-Voullième D, Morales-Hernández M, Norman MR, Özgen-Xian I. SERGHEI (SERGHEI-SWE) v1.0: A performance-portable high-performance parallel-computing shallow-water solver for hydrology and environmental hydraulics. Geosci Model Dev 2023;16(3):977–1008.

[22] Maier M, Kronbichler M. Efficient parallel 3D computation of the compressible Euler equations with an invariant-domain preserving second-order finite-element scheme. ACM Trans Parallel Comput 2021;8(3):16:1–30.

[23] Guermond J-L, Kronbichler M, Maier M, Popov B, Tomas I. On the implementation of a robust and efficient finite element-based parallel solver for the compressible Navier–Stokes equations. Comput Methods Appl Mech Engrg 2022;389:114250.

[24] Green AE, Laws N, Naghdi P. On the theory of water waves. Proc R Soc A 1974;338(1612):43–55.

[25] Serre F. Contribution à l'étude des écoulements permanents et variables dans les canaux. Houille Blanche 1953;39(6):830–72.

[26] Azerad P, Guermond J-L, Popov B. Well-balanced second-order approximation of the shallow water equation with continuous finite elements. SIAM J Numer Anal 2017;55(6):3203–24.

[27] Ruuth SJ, Spiteri RJ. Two barriers on strong-stability-preserving time discretization methods. J Sci Comput 2002;17:211–20.
[28] Ern A, Guermond J-L. Invariant-domain-preserving high-order time stepping: I. Explicit Runge–Kutta schemes. SIAM J Sci Comput 2022;44(5):A3366–92.
[29] Guermond J-L, Popov B. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. SIAM J Numer Anal 2016;54(4):2466–89.
[30] Chertock A, Dudzinski M, Kurganov A, Lukáčová-Medvid'ová M. Well-balanced schemes for the shallow water equations with Coriolis forces. Numer Math 2018;138:939–73.
[31] Alcrudo F, Benkhaldoun F. Exact solutions to the Riemann problem of the shallow water equations with a bottom step. Comput & Fluids 2001;30(6):643–71.
[32] Bernetti R, Titarev VA, Toro EF. Exact solution of the Riemann problem for the shallow water equations with discontinuous bottom geometry. J Comput Phys 2008;227(6):3212–43.
[33] Sun Z, Xing Y. On a numerical artifact of solving shallow water equations with a discontinuous bottom: Analysis and a nontransonic fix. 2023, URL https://arxiv.org/abs/2308.09265.
[34] Guermond J-L, Popov B, Tomas I. Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems. Comput Methods Appl Mech Engrg 2019;347:143–75.
[35] Audusse E, Bristeau M-O. A well-balanced positivity preserving "second-order" scheme for shallow water flows on unstructured meshes. J Comput Phys 2005;206(1):311–33.
[36] Harten A, Lax PD, Leer Bv. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. SIAM Rev 1983;25(1):35–61.
[37] Nessyahu H, Tadmor E. Non-oscillatory central differencing for hyperbolic conservation laws. J Comput Phys 1990;87(2):408–63.
[38] Guermond J-L, Nazarov M, Popov B, Yang Y. A second-order maximum principle preserving Lagrange finite element technique for nonlinear scalar conservation equations. SIAM J Numer Anal 2014;52(4):2163–82.
[39] Guermond J-L, Pasquetti R. A correction technique for the dispersive effects of mass lumping for transport problems. Comput Methods Appl Mech Engrg 2013;253:186–98.
[40] Guermond J-L, Popov B, Tovar E, Kees C. Robust explicit relaxation technique for solving the Green–Naghdi equations. J Comput Phys 2019;399:108917, 17.
[41] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. J Comput Phys 1979;31(3):335–62.
[42] Boris JP, Book DL. Flux-corrected transport. J Comput Phys 1997;135(2):172–86.
[43] Kuzmin D, Löhner R, Turek S. Flux-corrected transport: Principles, algorithms, and applications. Springer; 2012.
[44] Guermond J-L, Nazarov M, Popov B. Finite element-based invariant-domain preserving approximation of hyperbolic systems: Beyond second-order accuracy in space. Comput Methods Appl Mech Engrg 2024;418(part A). Paper No. 116470, 22.
[45] Zhang X, Shu C-W. Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms. J Comput Phys 2011;230(4):1238–48.
[46] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77(2):439–71.
[47] Gottlieb S, Shu C-W, Tadmor E. Strong stability-preserving high-order time discretization methods. SIAM Rev 2001;43(1):89–112.
[48] Kuzmin D, Quezada de Luna M, Ketcheson DI, Grüll J. Bound-preserving flux limiting for high-order explicit Runge–Kutta time discretizations of hyperbolic conservation laws. J Sci Comput 2022;91(1):21.
[49] Hairer E, Norsett S, Wanner G. Solving ordinary differential equations I: Nonstiff problems. Springer; 1993.
[50] Arndt D, Bangerth W, Bergbauer M, Feder M, Fehling M, Heinz J, Heister T, Heltai L, Kronbichler M, Maier M, Munch P, Pelteret J-P, Turcksin B, Wells D, Zampini S. The `deal.II` library, version 9.5. J Numer Math 2023;31(3):231–46.
[51] Kawahara M, Umetsu T. Finite element method for moving boundary problems in river flow. Internat J Numer Methods Fluids 1986;6(6):365–86.
[52] LeVeque RJ. Balancing source terms and flux gradients in high-resolution godunov methods: The quasi-steady wave-propagation algorithm. J Comput Phys 1998;146(1):346–65.
[53] Ricchiuto M, Abgrall R, Deconinck H. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes. J Comput Phys 2007;222(1):287–331.
[54] Seaïd M. Non-oscillatory relaxation methods for the shallow-water equations in one and two space dimensions. Internat J Numer Methods Fluids 2004;46(5):457–84.
[55] Thacker WC. Some exact solutions to the nonlinear shallow-water wave equations. J Fluid Mech 1981;107:499–508.
[56] Delestre O, Lucas C, Ksinant P-A, Darboux F, Laguerre C, Vo T-N-T, James F, Cordier S. SWASHES: A compilation of shallow water analytic solutions for hydraulic and environmental studies. Internat J Numer Methods Fluids 2013;72(3):269–300.
[57] Hou J, Simons F, Mahgoub M, Hinkelmann R. A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. Comput Methods Appl Mech Engrg 2013;257:126–49.
[58] Martínez-Aranda S, Fernández-Pato J, Caviedes-Voullième D, García-Palacín I, García-Navarro P. Towards transient experimental water surfaces: A new benchmark dataset for 2D shallow water solvers. Adv Water Resour 2018;121:130–49.
[59] United States Geological Survey. United States geological survey 3D elevation program: 1 meter digital elevation model. 2021, Distributed by OpenTopography.
[60] Bristeau M-O, Coussin B. Boundary conditions for the shallow water equations solved by kinetic schemes. Research report inria-00072305, INRIA; 2001.