Entropy stable discontinuous Galerkin methods for the shallow water equations with subcell positivity preservation

Xinhui Wu *[1], Nathaniel Trask[2], and Jesse Chan[1]

[1]Department of Computational Applied Mathematics and Operations Research, Rice University, Houston, TX
[2]Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania

Abstract

High order schemes are known to be unstable in the presence of shock discontinuities or under-resolved solution features, and have traditionally required additional filtering, limiting, or artificial viscosity to avoid solution blow up. Entropy stable schemes address this instability by ensuring that physically relevant solutions satisfy a semi-discrete entropy inequality independently of discretization parameters. However, additional measures must be taken to ensure that solutions satisfy physical constraints such as positivity. In this work, we present a high order entropy stable discontinuous Galerkin (ESDG) method for the nonlinear shallow water equations (SWE) on two-dimensional (2D) triangular meshes which preserves the positivity of the water heights. The scheme combines a low order positivity preserving method with a high order entropy stable method using convex limiting. This method is entropy stable and well-balanced for fitted meshes with continuous bathymetry profiles.

1 Introduction

The shallow water equations (SWE) are used to model fluid flows in rivers, open channels, lakes, and coastal regions. One of the most prominent practical applications of the shallow water equations is the numerical prediction of storm surges under extreme weather conditions like hurricanes near coastal regions [1, 2, 3]. All of these flows share the fact that vertical scales of the motion are much smaller than horizontal scales. Under certain assumptions [4], the incompressible Navier–Stokes equations of fluid dynamics can be simplified to the 2D shallow water equations:

$$h_t + (hu)_x + (hv)_y = 0, \tag{1.1}$$

$$(hu)_t + (hu^2 + gh^2/2)_x + (huv)_y = -ghb_x, \tag{1.2}$$

$$(hv)_t + (huv)_x + (hv^2 + gh^2/2)_y = -ghb_y, \tag{1.3}$$

where $h = h(x, y, t)$ denotes the height of the water measured from the bottom, $u = u(x, y, t)$ and $v = v(x, y, t)$ denote the velocity in the x and y directions respectively, g denotes the gravitational constant. The bathymetry height is denoted by $b = b(x, y)$. The subscript $(.)_t$ denotes the

*Email: philipwxh@gmail.com

time derivative, while the subscripts $(.)_x$ or $(.)_y$ denote directional derivatives along the x and y directions, respectively.

Researches on numerical methods for the solution of the shallow water equations have achieved considerable attention in the past two decades. Common numerical methods including the finite difference, the finite volume, and the finite element method have been applied to the shallow water system. This paper focuses on the discontinuous Galerkin (DG) method. Discontinuous Galerkin methods combine advantages of both finite element and finite volume methods, and have been successfully applied to a variety of settings [5]. Discontinuous Galerkin methods can achieve high order accuracy, parallel efficiency, and enjoy more straightforward hp-refinement [6, 7]. However, high order discontinuous Galerkin methods also introduce issues of stability for problems with shocks or under-resolved solutions. Recent works on entropy stable high order discontinuous Galerkin methods [8, 9, 10] provide a way to address such instabilities. Entropy stable discontinuous Galerkin methods can also be extended to curved meshes, which can be necessary when dealing with complex geometries.

One of the important numerical properties for the shallow water equations is the preservation of steady state solutions, also known as the "lake at rest" condition [3]:

$$H = \text{constant}, \qquad u = v = 0, \tag{1.4}$$

where $H = h + b$. A numerical scheme which preserves this steady state is said to be well-balanced [11, 12]. Schemes that are not well-balanced can generate spurious waves in the presence of varying bottom topographies. Accomplishing this discretely can be challenging because special discretizations of the source terms are required [8, 13, 14].

Another important issue often experienced in numerical simulations of the shallow water equations is the treatment of dry areas, where the water is absent. Dry areas are common in realistic applications of the shallow water equations, such as dam breaks, flooding, and shoreline simulations. Some numerical methods can fail when encountering zero or negative water heights. Various wetting and drying treatments for the shallow water equations have been developed [15, 16, 17, 18, 19]. Some use mesh adaption to track dry fronts and modify the meshes to fit the wet area. Others exclude dry elements from computation and include them when they are wet again; however, this approach may introduce oscillations and a loss of mass and momentum conservation. Positivity-preserving limiters that average the water height in partially dry elements have been also explored in [16].

In this paper, we introduce a high order entropy stable numerical method that preserves the positivity of water heights for the shallow water equations through convex limiting. The sections in this paper are organized as follows. We first present a high order entropy stable numerical scheme and entropy stable numerical fluxes for the shallow water equation in 1D and on 2D triangular meshes [9, 20, 21, 22, 23] in Sec. 2. Then, we present a low order positivity preserving method [24] which preserves the positivity of the water heights by using a graph viscosity in Sec. 3. We combine the high order scheme with the low order scheme through a technique called convex limiting [25] in Sec. 4. If the high order scheme produces negative water heights, the convex limiting procedure blends them with the low order solution to preserve positivity while maintaining high order accuracy in wet areas. Our approach is also different in the sense that we only limit in dry areas without limiting near shocks. We also prove this scheme is well-balanced for meshes which are fitted to dry-wet interfaces. We demonstrate our new method with five numerical experiments that illustrate different properties of our new method in Sec. 5. Conclusions are presented in Sec. 6.

2 High order entropy stable summation-by-parts formulations

2.1 Mathematical assumption and notations

We first introduce some underlying mathematical assumptions and notations for our DG method. For consistency, we reuse notations from [9], with slight modifications to provide a cleaner discrete formulation. We denote the triangular reference element by $\widehat{D}$ with boundary $\partial\widehat{D}$. In 2D, the reference element is a triangle with vertices at $(-1,-1)$, $(-1,1)$ and $(1,-1)$. We use $\widehat{n}_i$ to represent the ith component of the outward normal vector scaled by the face Jacobian on the boundary of the reference element. The space of polynomials up to degree N on the reference element is defined as

$$P^N(\widehat{D}) = \{\widehat{x}^i\widehat{y}^j, \quad (\widehat{x},\widehat{y}) \in \widehat{D}, \quad 0 \le i+j \le N\}. \tag{2.1}$$

Finally, we denote the dimension of P^N as $N_p = \mathrm{dim}(\mathrm{P^N}(\widehat{\mathrm{D}}))$.

2.2 SBP quadrature rules

We require SBP-quadrature rules to satisfy certain properties detailed in [21, 22]. In our numerical experiments, we consider two sets of SBP quadrature points. The first uses 1D Gauss-Legendre quadrature on the edges while the second uses 1D Gauss-Lobatto quadrature on edges. These are shown in the Fig. 1 and 2, respectively. The Gauss-Legendre SBP quadrature rules were introduced in [21], and the Gauss-Lobatto SBP quadrature rules were introduced in [22].

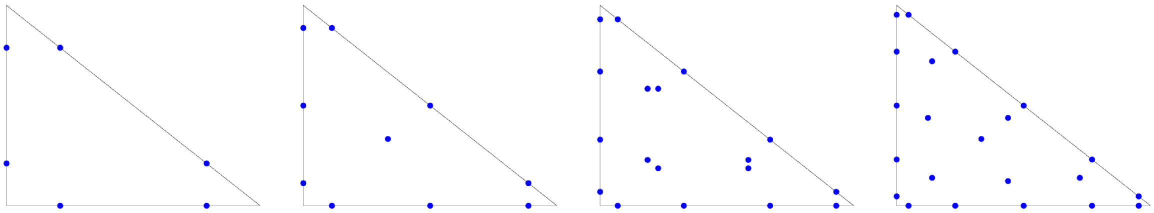


Figure 1: SBP nodes based on Gauss-Legendre quadrature for $N = 1, 2, 3, 4$

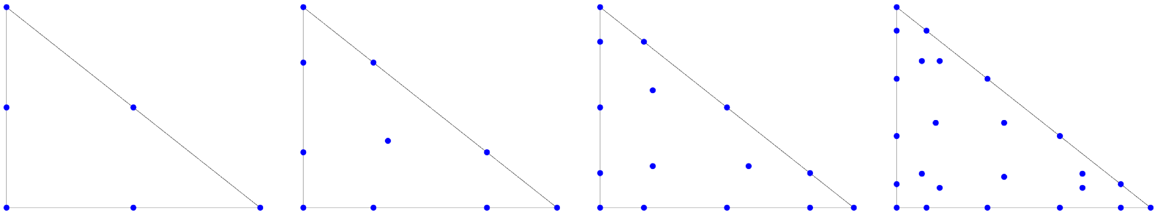


Figure 2: SBP nodes based on Gauss-Lobatto quadrature for $N = 1, 2, 3, 4$

2.3 Multi-dimensional SBP operators

We now describe properties satisfied by nodal SBP operators. We assume we are given a degree $2N-1$ quadrature rule for the polynomial space $P^N(\widehat{D})$, with N_q nodes, $\{\mathbf{x_i}\}_{i=1}^{N_q}$, and positive weights $\{w_i\}_{i=1}^{N_q}$. The face quadrature rule is degree $2N$ and is embedded in the volume nodes. The nodal values of the function $u(x)$ on the quadrature points is denoted by:

$$\mathbf{u} = [u(\mathbf{x}_1), ..., u(\mathbf{x}_{N_q})]^T. \tag{2.2}$$

The SBP mass matrix is defined as a diagonal matrix with quadrature weights on the diagonal.

$$\mathbf{M} = \operatorname{diag}\left([w_1, ..., w_{N_q}]\right). \tag{2.3}$$

We define $\mathbf{D}_x$ and $\mathbf{D}_y$ to be nodal differentiation matrices associated with the x and y derivatives [20, 22]. We also define $\mathbf{B}_x$ and $\mathbf{B}_y$ to be diagonal surface matrices whose entries are surface quadrature weights scaled by the x and y components of the outward normal vector.

We have the following definition [21].

Definition 2.1. *Consider the diagonal mass matrix consisting of quadrature weights*

$$\mathbf{M} = \operatorname{diag}\left([w_1, ..., w_{N_q}]\right). \tag{2.4}$$

A 2D operator $\mathbf{Q}_i$ is said to be high order accurate and have SBP property if for $i = x, y$, the following properties hold:

- *Let $\mathbf{D}_i = \mathbf{M}^{-1}\mathbf{Q}_i$. Then $(\mathbf{D}_i\mathbf{u})_j = \frac{\partial \mathbf{u}}{\partial x_i}\Big|_{x=x_j}$ for any $\mathbf{u} \in P^N(\widehat{D})$.*
- *$\mathbf{Q}_i + (\mathbf{Q}_i)^T = \mathbf{B}_i$.*

Given this equivalence, we can now construct an entropy conservative DG-SBP form:

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}} + \sum_{i=x,y} \left(2\mathbf{Q}_i \circ \mathbf{F}_i\right)\mathbf{1} + \mathbf{B}_i\left(\boldsymbol{f}_{S,i}\left(\mathbf{u}^+, \mathbf{u}\right) - \mathbf{f}_i(\boldsymbol{u})\right) = \mathbf{S}, \tag{2.5}$$

where the matrix operations $\circ$ denotes the Hadamard product. $\mathbf{F}_i$ denotes the flux matrix and each entry $(\mathbf{F}_i)_{jk} = \boldsymbol{f}_i(\boldsymbol{u}_j, \boldsymbol{u}_k)$, where $\boldsymbol{f}_i$ is an entropy conservative numerical flux that we will introduce in the next section. $\mathbf{S}$ denotes the source term. The construction of the high order operators are presented in [23]. The stability analysis of the SBP formulation follows from the results from [21], and the extension to curved elements can be found in [26].

2.4 Entropy conservative fluxes for the shallow water equations

The entropy function for the shallow water equations is the total energy of the system [13, 8]:

$$S(\boldsymbol{u}) = \frac{1}{2}h(u^2 + v^2) + \frac{1}{2}gh^2 + ghb. \tag{2.6}$$

We also define the entropy variable $\boldsymbol{v} = S'(\boldsymbol{u})$. The convexity of the entropy function guarantees that the mapping between $\boldsymbol{u}$ and $\boldsymbol{v}$ is invertible. The entropy variables for the shallow water equations are given explicitly as:

$$v_1 = \frac{\partial S}{\partial h} = g(h+b) - \frac{1}{2}u^2 - \frac{1}{2}v^2, \tag{2.7}$$

$$v_2 = \frac{\partial S}{\partial (hu)} = u, \tag{2.8}$$

$$v_3 = \frac{\partial S}{\partial (hv)} = v. \tag{2.9}$$

It can be shown as in [27] that there exists an entropy flux function $F(\boldsymbol{u})$ and entropy potential $\psi(\boldsymbol{u})$ such that

$$\boldsymbol{v}(\boldsymbol{u})^T\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \frac{\partial F(\boldsymbol{u})^T}{\partial \boldsymbol{u}}, \quad \psi(\boldsymbol{u}) = \boldsymbol{v}(\boldsymbol{u})^T\boldsymbol{f}(\boldsymbol{u}) - F(\boldsymbol{u}), \quad \psi'(\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u}). \tag{2.10}$$

We introduce numerical fluxes for the shallow water equations and describe an entropy conservation discrete formulation [28, 29, 21, 30]. To construct the entropy stable scheme in higher dimensions, we require entropy conservative fluxes as defined in [31].

Definition 2.2. *Let $\boldsymbol{f}_{S,i}(\boldsymbol{u}_L, \boldsymbol{u}_R)$ be a bivariate function which is symmetric and consistent with the flux function $\boldsymbol{f}_i(\boldsymbol{u})$, for $i = 1, ..., d$*

$$\boldsymbol{f}_{S,i}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{f}_{S,i}(\boldsymbol{v}, \boldsymbol{u}), \quad \boldsymbol{f}_{S,i}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}_i(\boldsymbol{u}). \tag{2.11}$$

The numerical flux $\boldsymbol{f}_{S,i}(\boldsymbol{u}_L, \boldsymbol{u}_R)$ is entropy conservative if, for entropy variables $\boldsymbol{v}_L = \mathbf{V}(\boldsymbol{u}_L)$ and $\boldsymbol{v}_R = \boldsymbol{v}(\boldsymbol{u}_R)$, $\boldsymbol{f}_{S,i}(\boldsymbol{u}_L, \boldsymbol{u}_R)$ satisfies

$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{f}_{S,i}(\boldsymbol{u}_L, \boldsymbol{u}_R) = \psi_{i,L} - \psi_{i,R}, \tag{2.12}$$

$$\psi_{i,L} = \psi_i(\boldsymbol{v}(\boldsymbol{u}_L)), \quad \psi_{i,R} = \psi_i(\boldsymbol{v}(\boldsymbol{u}_R)). \tag{2.13}$$

The flux $\boldsymbol{f}_{S,i}$ can be used to construct entropy conservative and entropy stable finite volume methods, as well as DG methods. Entropy conservative (EC) fluxes for the 2D shallow water equations [32, 33, 11, 3] are given by

$$\boldsymbol{f}_{x,S}(\boldsymbol{u}_L, \boldsymbol{u}_R) = \begin{bmatrix} \{\{hu\}\} \\ \{\{hu\}\}\{\{u\}\} + g\{\{h\}\}^2 - \frac{1}{2}g\{\{h^2\}\} \\ \{\{hu\}\}\{\{v\}\} \end{bmatrix}, \tag{2.14}$$

$$\boldsymbol{f}_{y,S}(\boldsymbol{u}_L, \boldsymbol{u}_R) = \begin{bmatrix} \{\{hv\}\} \\ \{\{hv\}\}\{\{u\}\} \\ \{\{hv\}\}\{\{v\}\} + g\{\{h\}\}^2 - \frac{1}{2}g\{\{h^2\}\} \end{bmatrix}. \tag{2.15}$$

These fluxes also yield a well-balanced scheme for continuous bathymetry, if the source terms are discretized appropriately [8]. The source terms $\mathbf{S}$ for the shallow water equations with continuous bottom geometry in the discrete form is defined as

$$\mathbf{S} = -g\begin{bmatrix} \mathbf{0} \\ \operatorname{diag}(\mathbf{h})\,\mathbf{Q}_x\mathbf{b} \\ \operatorname{diag}(\mathbf{h})\,\mathbf{Q}_y\mathbf{b} \end{bmatrix}. \tag{2.2}$$

By adding local Lax-Friedrichs interface penalization [8], we can make our numerical scheme entropy stable. The shallow water equations in 1D and 2D are presented in Appendix C, along with their flux and entropy functions.

3 Positivity preserving formulation

In this section, we first describe the steps for constructing a low order positivity preserving DG scheme from [25] for the shallow water equations. Then, we prove that the low order method preserve the positivity of the water heights. Last, we remark on the sparsity of the low order operators.

3.1 The low order scheme

We build a low order positivity preserving scheme based on a connectivity graph and the solution of a graph Laplacian problem constructed from the connectivity graph. We present the details of the construction in Appendix A. We assume the same reference element, quadrature points, and discrete operators. We assume the forward Euler time stepping and let t^n denote the time at step $n \in \mathbb{N}$ and dt denote the current time step size. Let m_i denote the ith diagonal entry global mass matrix, which is diagonal. Suppose we can construct an inviscid approximation $\mathbf{u}^{n+1}$ of the shallow water equations at some time as follow [25]:

$$\frac{m_i}{dt}(\mathbf{u}_i^{n+1} - \mathbf{u}_i^n) + \sum_{j \in I(i)} \mathbf{f}(\mathbf{u}_j^n)^T (\mathbf{Q}^L)_{ij} = m_i \mathbf{S}(\mathbf{u}_i^n), \tag{3.1}$$

for any $i \in V$, the set of all nodes. For any i, the set $I(i)$ is the neighborhood of the node i. We assume that the following property holds: $j \in I(i)$ iff $i \in I(j)$. $\mathbf{Q}^L$ denotes the low order differentiation operator. Notice that $\mathbf{Q}^L$ in 2D consists of the differential operators in both x and y direction and $(\mathbf{Q}^L)_{ij}$ denotes the vector $[(\mathbf{Q}_x^L)_{ij}, (\mathbf{Q}_y^L)_{ij}]^T$. $\mathbf{Q}^L$ is described more in details in Sec. 3.2, and its construction is presented in Appendix A. The discrete form of the source term $\mathbf{S}$ is defined in Eq. 2.2, with replacing the high order differentiation operator with lower order differentiation operator. We also assume that the entries of the low order differentiation operator, $(\mathbf{Q}^L)_{ij}$, has the following properties:

$$(\mathbf{Q}^L)_{ij} = -(\mathbf{Q}^L)_{ji} \quad \text{and} \quad \sum_{j \in I(i)} (\mathbf{Q}^L)_{ij} = 0. \tag{3.2}$$

To handle shocks and discontinuous in the solutions, we introduce artificial dissipation using the graph Laplacian associated with the connectivity graph (V, E). Let $d_{ij}^{L,n}$ denote the scalar graph viscosity for $(i, j) \in E$ with the following properties:

$$d_{ij}^{L,n} = d_{ji}^{L,n} > 0 \quad \text{if} \quad i \neq j. \tag{3.3}$$

We define $d_{ii}^{L,n} = -\sum_{j \in I(i), j \neq i} d_{ij}^{L,n}$. Now, we can define the low order update $\mathbf{u}_i^{L,n+1}$ as follows:

$$\frac{m_i}{dt}(\mathbf{u}_i^{n+1} - \mathbf{u}_i^n) + \sum_{j \in I(i)} \boldsymbol{f}(\mathbf{u}_j^n)^T (\mathbf{Q}^L)_{ij} - \sum_{j \in I(i) \setminus \{i\}} d_{ij}^{L,n}(\mathbf{u}_j^n - \mathbf{u}_i^n) = m_i \mathbf{S}(\mathbf{u}_i^n), \tag{3.4}$$

for all $1 \leq i \leq N_q$. This scheme is conservative in the following sense [25].

Lemma 3.1. *Assume that $\boldsymbol{S} \equiv \mathbf{0}$, then the scheme (3.4) is conservative, such that the following identity holds for any $n \in \mathbb{N}$:*

$$\sum_{i \in V} m_i \mathbf{u}_i^{L,n+1} = \sum_{i \in V} m_i \mathbf{u}_i^{L,n}. \tag{3.5}$$

Proof. See [25]. □

To enforce the positivity preserving property, we define the graph viscosity coefficients $d_{ij}^{L,n}$ as follows:

$$d_{ij}^{L,n} = \max(\lambda_{\max}(\mathbf{u}_i^n, \mathbf{u}_j^n, \mathbf{n}_{ij}) \|(\mathbf{Q}^L)_{ij}\|_{\ell^2}, \lambda_{\max}(\mathbf{u}_j^n, \mathbf{u}_i^n, \mathbf{n}_{ij}) \|(\mathbf{Q}^L)_{ji}\|_{\ell^2}), \tag{3.6}$$

where

$$\lambda_{\max}(\mathbf{u}_i^n, \mathbf{u}_j^n, \mathbf{n}_{ij}) \tag{3.7}$$

is the upper bound for the maximum wave speed. For the shallow water equations, we use the estimate

$$\lambda_{\max}(\mathbf{u}_i^n, \mathbf{u}_j^n, \mathbf{n}_{ij}) = \max(|u_i \boldsymbol{n}_{ij} + v_i \boldsymbol{n}_{ij}| + \sqrt{gh_i}, |u_j \boldsymbol{n}_{ij} + v_j \boldsymbol{n}_{ij}| + \sqrt{gh_j}),$$

where u_i and v_i are the x and y velocity respectively at quadrature point i and $\boldsymbol{n}_{ij} = (\mathbf{Q}^L)_{ij}/\|(\mathbf{Q}^L)_{ij}\|$.

Theorem 3.1. *The numerical scheme (3.4) with $d_{ij}^{L,n}$ defined in (3.6) preserves the positivity of the water height for the shallow water equations for a time step size $dt < \min_i \frac{m_i}{2d_{ii}^{L,n}}$.*

Proof. Since we have $\sum_{j \in I(i)} \boldsymbol{f}(\mathbf{u}_i^n)^T (\mathbf{Q}^L)_{ij} = 0$ by property (3.2), we can rewrite the update on the water height term (3.4) as follows:

$$\frac{m_i}{dt}(h_i^{n+1} - h_i^n) + \sum_{j \in I(i) \backslash \{i\}} 2d_{ij}^{L,n} h_i^n + \left(\boldsymbol{f}_h(\mathbf{u}_j^n) - \boldsymbol{f}_h(\mathbf{u}_i^n)\right)^T (\mathbf{Q}^L)_{ij} - d_{ij}^{L,n}(h_j^n + h_i^n) = 0, \tag{3.8}$$

where $\boldsymbol{f}_h$ denotes the flux term that corresponds to the water height. We ignore the source term $\boldsymbol{S}(\mathbf{u}_i^n)$ because the topography or friction terms only affect the momentum equations but not water heights. Then we can introduce the intermediate bar states defined by

$$\bar{h}_{ij}^n = \frac{1}{2}(h_i^n + h_j^n) - \left(\boldsymbol{f}_h(\mathbf{u}_j^n) - \boldsymbol{f}_h(\mathbf{u}_i^n)\right)^T \frac{(\mathbf{Q}^L)_{ij}}{2d_{ij}^{L,n}}. \tag{3.9}$$

Notice that we have $\bar{h}_{ii}^n = h_i^n$. We first want to show that if we assume h_i^n is positive for all i, then $\bar{h}_i^n$ is also positive. Notice that, for the shallow water equations, the entropy stable numerical fluxes (2.14) are $\{\{hv\}\}$ for the water height, where v is the velocity of the flow. We can substitute the flux in (3.9) and rewrite the "bar state" as follow:

$$\bar{h}_{ij}^n = \frac{1}{2}(h_i^n + h_j^n) - \left((\boldsymbol{hv})_j^n - (\boldsymbol{hv})_i^n\right)^T \frac{(\mathbf{Q}^L)_{ij}}{2d_{ij}^{L,n}}. \tag{3.10}$$

By the definition of $d_{ij}^{L,n}$ and $\lambda_{\max}$, we have that $\lambda_{\max} > \max(|\boldsymbol{v}_i^T \cdot \mathbf{Q}_{ij}^L|, |\boldsymbol{v}_j^T \cdot \mathbf{Q}_{ij}^L|)$ where $\boldsymbol{v}_i$ is the velocity state, we know that $\left|\frac{\boldsymbol{v}_i^n (\mathbf{Q}^L)_{ij}}{2d_{ij}^{L,n}}\right| = c_i < 1$ and $\left|\frac{\boldsymbol{v}_j^n (\mathbf{Q}^L)_{ij}}{2d_{ij}^{L,n}}\right| = c_j < 1$. Since h_i^n and h_j^n are positive, we have

$$\bar{h}_{ij}^n \geq \frac{1}{2}(h_i^n + h_j^n - c_i h_i^n - c_j h_j^n) > 0. \tag{3.11}$$

Then we can rewrite the low order update on the water height with the bar state as:

$$h_i^{n+1} = \left(1 - \sum_{j \in I(i) \backslash \{i\}} \frac{2d_{ij}^{L,n} dt}{m_i}\right) h_i^n + \sum_{j \in I(i) \backslash \{i\}} \frac{2d_{ij}^{L,n} dt}{m_i} \bar{h}_{ij}^n \tag{3.12}$$

$$= \left(1 + 2dt \frac{d_{ii}^{L,n}}{m_i}\right) h_i^n + \sum_{j \in I(i) \backslash \{i\}} \frac{2d_{ij}^{L,n} dt}{m_i} \bar{h}_{ij}^n. \tag{3.13}$$

Recall that $d_{ii}^{L,n} = -\sum_{j\in I(i),j\neq i} d_{ij}^{L,n}$. If time step satisfies that $dt < \min_i \left(\frac{m_i}{2d_{ii}^{L,n}}\right)$, then, the term $1 + 2dt\frac{d_{ii}^{L,n}}{m_i} > 0$ and the terms in (3.13) form a convex combination of two positive water heights. □

3.2 Sparsity of the low order operators

One way to build a more accurate lower order numerical scheme is to remove excessive artificial dissipation from the method. In [34], the author constructed sparse low order operators for tensor product elements in 2D. In this paper, we construct sparse low order operators on triangular elements in 2D. Sparsification can be accomplished by using a sparse connectivity graph to construct low order operators. The sparsity of the graph adjacent matrix $\mathbf{A}$ directly dictates the sparsity of the differentiation operators $\mathbf{Q}^L$. With sparser low order operators, each quadrature node is connected to a fewer number of neighboring nodes. Sparsity reduces the amount of algebraic dissipation since the dissipation coefficient $d_{ij}^{L,n}$ is only non-zero for connected pairs of nodes in the graph. Details on the construction of sparse connectivity graphs are presented in Appendix A.I and numerical results from the low order scheme with different connectivity graphs are presented in Appendix B.

4 Convex limiting

In this section, we describe the application of convex limiting to combine the low order positivity preserving scheme with high order entropy stable discontinuous Galerkin and apply it to the shallow water equations. This scheme is aligned with the ideas presented in [25, 35, 36].

In Sec. 3, we showed that the low order solution satisfies the positivity preserving property. In particular, it preserves the positivity of water heights for the shallow water equations. However, it is low order accurate. To recover a high order accurate numerical scheme that also satisfies positivity requirements, we blend the high order and low order updates using "convex limiting". Convex limiting is only applied to elements where the high order scheme produces zero or negative water heights.

4.1 An overview of convex limiting

We use the superscript ^{H} to denote high order terms. By assuming using forward Euler time stepping, we begin by rewriting high order discontinuous Galerkin methods in the form

$$\frac{m_i}{dt}(\mathbf{u}_i^{H,n+1} - \mathbf{u}_i^n) + \sum_{j\in I(i)} \boldsymbol{F}_{ij}^H = m_i\mathbf{S}(\mathbf{u}_i^n), \tag{4.1}$$

where $\boldsymbol{F}_{ij}^H \in \mathbb{R}$ contains the flux contribution between quadrature points j and i. We also extend the results to higher order in time with strong-stability preserving Runge-Kutta (SSP-RK). Note that $\boldsymbol{F}_{ij}^H$ is skew-symmetric, $\boldsymbol{F}_{ij}^H = -\boldsymbol{F}_{ji}^H$. Subtracting (3.1) from (4.1), we have the following:

$$m_i\mathbf{u}_i^{H,n+1} = m_i\mathbf{u}_i^{L,n+1} + \sum_{j\in I(i)\backslash\{i\}} dt(\boldsymbol{F}_{ij}^L - \boldsymbol{F}_{ij}^H). \tag{4.2}$$

Let $\boldsymbol{A}_{ij} = dt(\boldsymbol{F}_{ij}^{L,n} - \boldsymbol{F}_{ij}^{H,n})$. Then, we can rewrite (4.1) as

$$m_i \mathbf{u}_i^{H,n+1} = m_i \mathbf{u}_i^{L,n+1} + \sum_{j \in I(i) \backslash \{i\}} \boldsymbol{A}_{ij}^n. \tag{4.3}$$

Note that we have $\boldsymbol{A}_{ij}^n = \boldsymbol{A}_{ji}^n$, which implies that $\sum_{i \in V} m_i \mathbf{u}_i^{L,n+1} = \sum_{i \in V} m_i \mathbf{u}_i^{H,n+1}$. Therefore, the high order and low order updates have the same mass. Our goal is to modify the high order update so that the modified high order update satisfies the positive water heights constraint. We introduce symmetric limiting parameters $l_{ij} = l_{ji} \in [0,1]$, for all $i, j \in V$. We define the limited update $\mathbf{u}_i^{n+1}$ as

$$m_i \mathbf{u}_i^{n+1} = m_i \mathbf{u}_i^{L,n+1} + \sum_{j \in I(i) \backslash \{i\}} l_{ij} \boldsymbol{A}_{ij}^n. \tag{4.4}$$

The limiting process is conservative for any choice of symmetric coefficients $l_{ij} = l_{ji}$ [25]. Notice that $\mathbf{u}_i^{n+1} = \mathbf{u}_i^{L,n+1}$ if $l_{ij} = 0$ for all $j \in I(i) \backslash i$ and $\mathbf{u}_i^{n+1} = \mathbf{u}_i^{H,n+1}$ if $l_{ij} = 1$ for all $j \in I(i) \backslash i$. Therefore, $h > 0$ when all $l_{ij} = 0$. To obtain a more accurate approximation, we want the limiting coefficients l_{ij} to be as close to 1 as possible while enforcing $h > 0$.

We can rewrite equation 4.4 as

$$m_i \mathbf{u}_i^{n+1} = \sum_{j \in I(i) \backslash \{i\}} l_{ij} \lambda_j (\mathbf{u}_i^{L,n+1} + l_{i,j} \boldsymbol{P}_{ij}^n), \quad \boldsymbol{P}_{ij}^n = \frac{1}{m_i \lambda_j} \boldsymbol{A}_{ij}^n, \tag{4.5}$$

where $\{\lambda_j\}_{j \in I(i) \backslash \{i\}}$ is any set of strictly positive convex coefficients such that $\sum_{j \in I(i) \backslash \{i\}} \lambda_j = 1$, $\lambda_j > 0$ for all $j \in I(i) \backslash \{i\}$. We choose $\lambda_j = \frac{1}{\text{card}(I(i))-1}$ as in [25]. To calculate l_{ij}, we first define l_j^i for every $i \in V$ and $j \in I(i)$:

$$l_j^i = \begin{cases} 1 & \text{if } h_i^{L,n+1} + P_{ij}^{n,h} \geq 0, \\ \max\{l \in [0,1] | h_i^{L,n+1} + l P_{ij}^{n,h}\} \geq 0 & \text{otherwise}, \end{cases} \tag{4.6}$$

where $P_{ij}^{n,h}$ corresponds to the update on the h component of the solution. We set the node-wise limiting factor $l_{ij} = l_{ji} = \min\{l_j^i, l_i^j\}$. With this definition, we can ensure that the water heights stay positive by Theorem 7.21 in [25]. While most of time, finding l_j^i require a line search, it remains simple for the shallow water equations since we can express l_j^i explicitly, due to the fact that h is a linear function, as follows:

$$l_j^i = \begin{cases} 1 & \text{if } h_i^{L,n+1} + P_{ij}^{n,h} \geq 0, \\ -\frac{h^{L,n+1}}{P_{ij}^{n,h}} & \text{otherwise}. \end{cases} \tag{4.7}$$

With convex limiting, we can ensure that our solution remains positive in water heights throughout a simulation.

An alternative limiting scheme, called element-wise limiting, is also considered in this paper. In this scheme, we have a uniform limiting factor across the same element. We define the element-wise limiting factor $l_e = \min_{i,j}\{l_{ij}\}$. This scheme produces solutions that are closer to the low order positivity preserving scheme than the node-wise limiting scheme because it always has smaller local

$$\boldsymbol{F}^H = \left[\begin{array}{ccc|ccc} f^H_{11} & f^H_{12} & f^H_{13} & & & f^H_{16} \\ f^H_{21} & f^H_{22} & f^H_{23} & & & \\ f^H_{31} & f^H_{32} & f^H_{33} & f^H_{34} & & \\ \hline & & f^H_{43} & f^H_{44} & f^H_{45} & f^H_{46} \\ & & & f^H_{54} & f^H_{55} & f^H_{56} \\ f^H_{61} & & & f^H_{64} & f^H_{65} & f^H_{66} \end{array}\right] \qquad \boldsymbol{F}^L = \left[\begin{array}{ccc|ccc} f^L_{11} & f^L_{12} & & & & f^L_{16} \\ f^L_{21} & & f^L_{23} & & & \\ & f^L_{32} & f^L_{33} & f^L_{34} & & \\ \hline & & f^L_{43} & f^L_{44} & f^L_{45} & \\ & & & f^L_{54} & & f^L_{56} \\ f^L_{61} & & & & f^L_{65} & f^L_{66} \end{array}\right]$$

Figure 3: Illustration of high order and low order flux matrices for two one-dimensional elements.

$$\boldsymbol{A} = \left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & & & a_{16} \\ a_{21} & a_{22} & a_{23} & & & \\ a_{31} & a_{32} & a_{33} & a_{34} & & \\ \hline & & a_{43} & a_{44} & a_{45} & a_{46} \\ & & & a_{54} & a_{55} & a_{56} \\ a_{61} & & & a_{64} & a_{65} & a_{66} \end{array}\right] = \left[\begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & & & \\ a_{21} & a_{22} & a_{23} & & & \\ a_{31} & a_{32} & a_{33} & & & \\ \hline & & & a_{44} & a_{45} & a_{46} \\ & & & a_{54} & a_{55} & a_{56} \\ & & & a_{64} & a_{65} & a_{66} \end{array}\right]$$

Figure 4: Illustration of the flux correction matrix $\boldsymbol{A} = dt(\boldsymbol{F}^L - \boldsymbol{F}^H)$ for two one-dimensional elements.

limiting factors. In practice, solutions from the element-wise limiting scheme are usually bounded further away from 0 and therefore, produce smaller maximum wave speed on the shallow water equations. This has added benefit of allowing us to take larger time steps. We can also prove that, in contrast to the node-wise limiting scheme, the element-wise limiting scheme also preserves the entropy stability of the blended scheme [37].

4.2 Local implementation

One important and efficient property of our numerical scheme is that it only requires computing limiting factors l_{ij} for the nodes i, j within the same element, which was first observed in [34]. We use the same entropy stable interface numerical fluxes with a Lax-Friedrichs dissipation in both high order and low order schemes. The resulting fluxes on the boundaries of each element are the same. For example, consider the high order and low order flux matrices for a simple periodic domain in 1D with two elements shown in Fig. 3. Suppose there are three nodes in each element. The non-zero pattern of the matrix $\boldsymbol{A} = dt(\boldsymbol{F}^L - \boldsymbol{F}^H)$ from Eq. (4.3) is shown in Fig. 4. We have non-zero entries from the off diagonal block due to the fluxes calculated on the boundaries of each element, namely the terms a_{16}, a_{34}, a_{43}, and a_{61}. However, these boundary fluxes are the same for both high order and low order updates; therefore corresponding entries in the difference matrix $\boldsymbol{A}$ vanish.

As a result, we do not need to implement the convex limiting procedure for the pairs of boundary nodes on neighboring elements. Convex limiting is only required for the pairs of nodes within the same element. With this key property, the implementation of convex limiting is purely local, simplifying the resulting numerical scheme.

5 Numerical results

In this section, we present some numerical experiments and results in both 1D and 2D to demonstrate the accuracy and stability of the convex limited entropy stable DG scheme. We also add a local Lax-Friedrichs penalization [8] to all interface fluxes.

The first experiment is a "lake-at-rest" condition to test the well-balancedness of our scheme in 1D. The second experiment is the parabolic bowl problem in 1D, designed to investigate the convergence rate when positivity preservation is activated. The third experiment is a dam break simulation in 2D from [3]. The last experiment is a wave over a bump in 2D [18].

All numerical experiments utilize a second order strong-stability preserving Runge-Kutta (SSP-RK) scheme (e.g., Heun's method) [38] for time integration. Following the derivation of stable timestep restrictions in [39], we define the timestep dt^H for the high-order scheme to be

$$dt^H = CFL \times \frac{h}{C_N}, \qquad C_N = \frac{(N+1)(N+2)}{2}, \tag{5.1}$$

where C_N is the degree dependent constant in the trace inequality space [40], and CFL is a user-defined constant. We use CFL $= 0.125$ for all experiments. For the low order scheme, the timestep dt^L is calculated via

$$dt^L = \min_{i \in V} \frac{w_i}{2\lambda_{\max}}, \tag{5.2}$$

where $\lambda_{\max}$ is the maximum wave speed calculated with Eq. (3.7) over the entire domain. We check for violation of the low order time-step during each stage of the Runge-Kutta method, and recompute the step with a smaller time-step if necessary. We define the timestep $dt = \min\{dt^H, dt^L\}$.

To avoid division by zero, we enforce the water height to be above a small positive tolerance $tol = 10^{-14}$ in dry areas. We use $g = 1$ unless otherwise noted in the description.

5.1 Lake at rest

The test problem is chosen to verify that the convex limited DG scheme indeed preserves the still water steady state with a non-flat bottom containing a wet/dry interface. The bottom topography is given by the depth function [16, 41]:

$$b(x) = \max(0, -20(x - 1/8)(x + 1/8) + 2) \quad \text{on } -1 \leq x \leq 1. \tag{5.3}$$

The initial data is the stationary solution:

$$h(x) + b(x) = \max(2, b), \quad hu = 0, \quad \text{on } -1 \leq x \leq 1. \tag{5.4}$$

A periodic boundary condition is used in this experiment. We compute the solution until T = 1 using 128 uniform cells. The computed surface level $h + b$ and the bottom b are plotted in Fig. 5. In order to demonstrate that the lake at rest solution is preserved up to round-off error, we compute the L^2 error of the solution. In double precision, the L^2 error is 3.56E-15.

We note that well-balancedness is not preserved if the mesh element boundary is not aligned with the wet/dry interface, and that designing a sub-cell well-balanced scheme remains challenging. For example, [42] presents two numerical schemes, and the scheme which preserves sub-cell well-balanceness is restricted to second order accuracy.

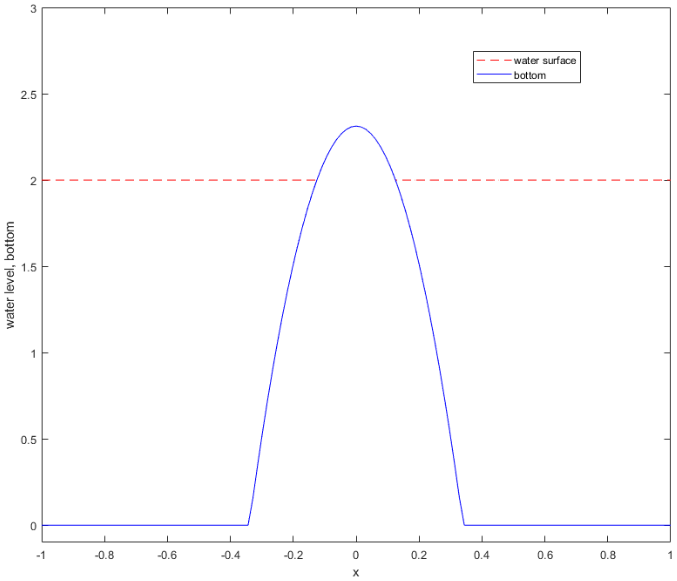


Figure 5: The surface level h + b and the bottom b for the lake at rest

5.2 Translating vortex

We now consider the translating vortex test. We set the domain to be $[-10, 10] \times [-5, 5]$. The exact solution for the vortex at any time t is given by [43, 44]:

$$h = h_\infty - \frac{\beta^2}{32\pi^2} e^{-2(r^2-1)}, \tag{5.5}$$

$$u = u_\infty - \frac{\beta}{2\pi} e^{-2(r^2-1)} y_t, \tag{5.6}$$

$$v = v_\infty + \frac{\beta}{2\pi} e^{-2(r^2-1)} x_t, \tag{5.7}$$

$$b = 0, \tag{5.8}$$

where

$$x_t = x - x_c - u_\infty t, \tag{5.9}$$

$$y_t = y - y_c - v_\infty t, \tag{5.10}$$

$$r^2 = x^2 + y^2. \tag{5.11}$$

In this example, we set

$$h_\infty = 1, \quad \beta = 5, \quad g = 2 \quad \text{and} \quad (u_\infty, v_\infty) = (1, 0). \tag{5.12}$$

Initially the vortex is located at $(x_c, y_c) = (0, 0)$ as shown in Fig. 6. In this setup, the vortex propagates to the right along the x-axis. The domain size is chosen to be large enough such that periodic boundary conditions can be used without affecting accuracy.

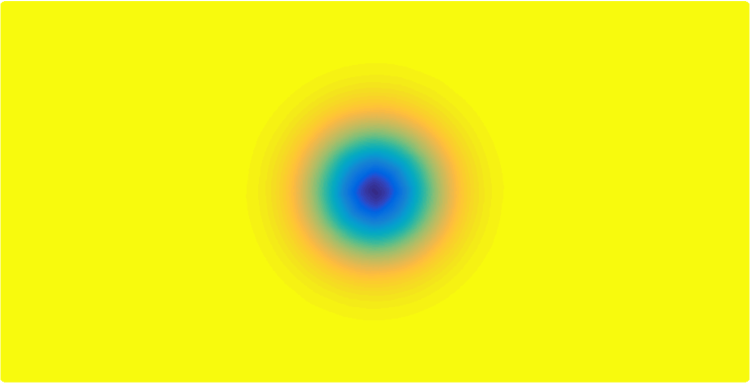


Figure 6: A translating vortex in 2D

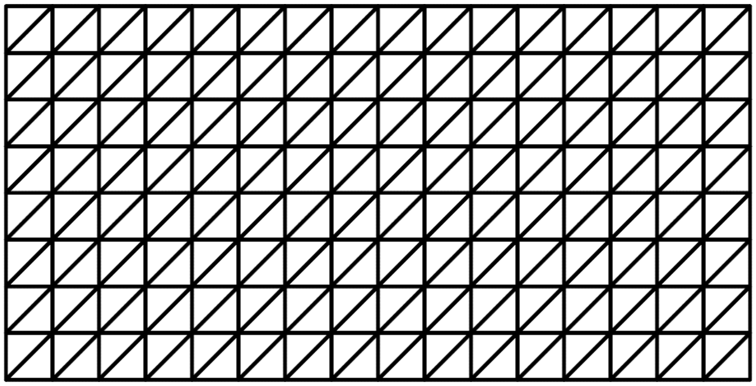


Figure 7: Meshes in 2D for translating vortex problem

We use affine triangular meshes for this experiment as shown in Fig. 7. We compute the L^2 error for the SBP formulation using operators based on Gauss-Legendre quadrature nodes. Recall that the SBP-DG discretization does not correspond to a polynomial approximation space. Thus, to calculate L^2 errors for the SBP-DG discretization, we first project the final numerical solution to polynomials of degree N. We then use this projection to evaluate the L^2 error using a quadrature rule which is exact for at least degree $2N+2$ polynomials. The convergence results are presented in Fig. 8. We observe an $O(h^{N+1})$ rate of convergence for the high order method on affine triangular meshes. The method can also be extended to curved triangular meshes [22].

5.3 Parabolic bowl

The second numerical experiment is used to verify the accuracy and convergence rate of the convex limited DG scheme. For the one-dimensional shallow water equations with a parabolic bottom topography, analytic solutions have been derived by Sampson et al. [45]. For this example, the parabolic bottom is given by

$$b(x) = h_0(x/a)^2, \tag{5.13}$$

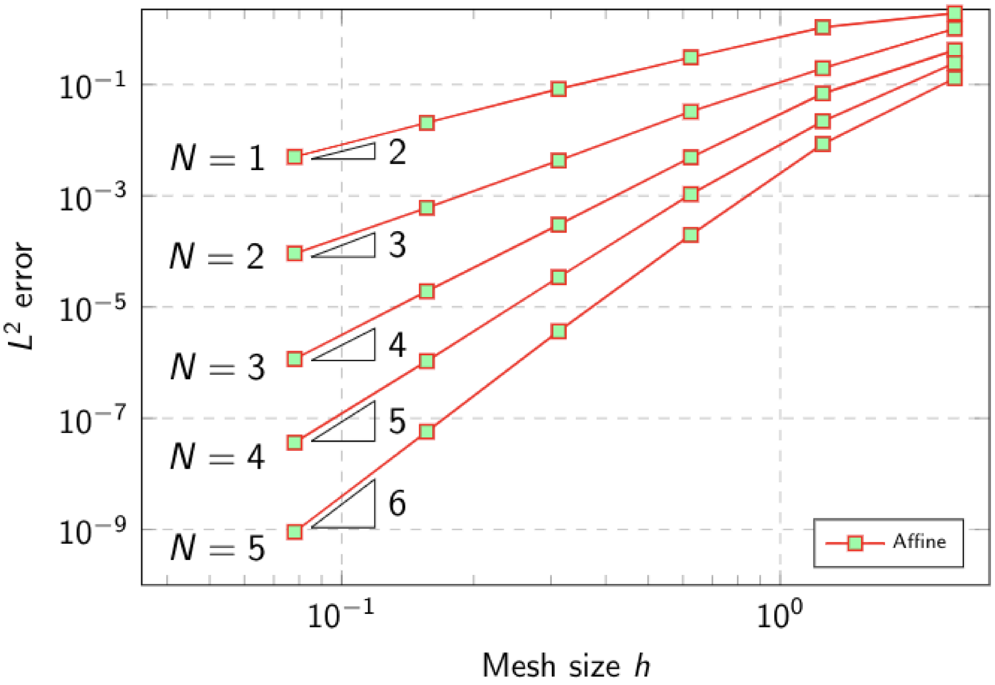


Figure 8: L^2 errors for the translating vortex after 0.5 seconds using nodal DG and Gauss-Legendre SBP operator schemes on various mesh sizes and polynomial degrees.

where we set $h_0 = 8$ and $a = 2$. The computational domain is $[-5, 5]$. The analytical water height for the shallow water equations is given by

$$h(x,t) + b(x) = h_0 - \frac{B^2}{4g}\cos(2\omega t) - \frac{B^2}{4g} - \frac{Bx}{2a}\sqrt{\frac{8h_0}{g}}\cos(\omega, t), \tag{5.14}$$

where $\omega = \sqrt{2gh_0}/a$ and $B = 2$. The exact location of the wet/dry fronts take the form

$$x_0 = -\frac{B\omega a^2}{2gh_0}\cos(\omega t) \pm a \tag{5.15}$$

The initial water height is then defined by (5.14). The velocity is initialized to zero. We can use either periodic or wall boundary conditions for this experiment because the water will never reach the boundary. We set the polynomial degree to $N = 3$ and run the simulation until $T = 1$ with various numbers of uniform elements. We plot the water surface at different times using the node-wise limiting and the element-wise limiting in Fig. 9 and Fig. 10, respectively. We also include the analytical solution to provide a comparison. We observe that the numerical solution matches well with the analytic solution. The element-wise limiting scheme produces smoother solutions than the node-wise limiting scheme. The node-wise limiting scheme produces some spurious oscillations near the dry front, though these oscillations are localized and vanish as we refine the mesh.

We plot the L^2 norm of the errors for different polynomial degrees and mesh sizes for the node-wise and element-wise limiting schemes in Fig. 11 and Fig. 12. We observe both schemes have a

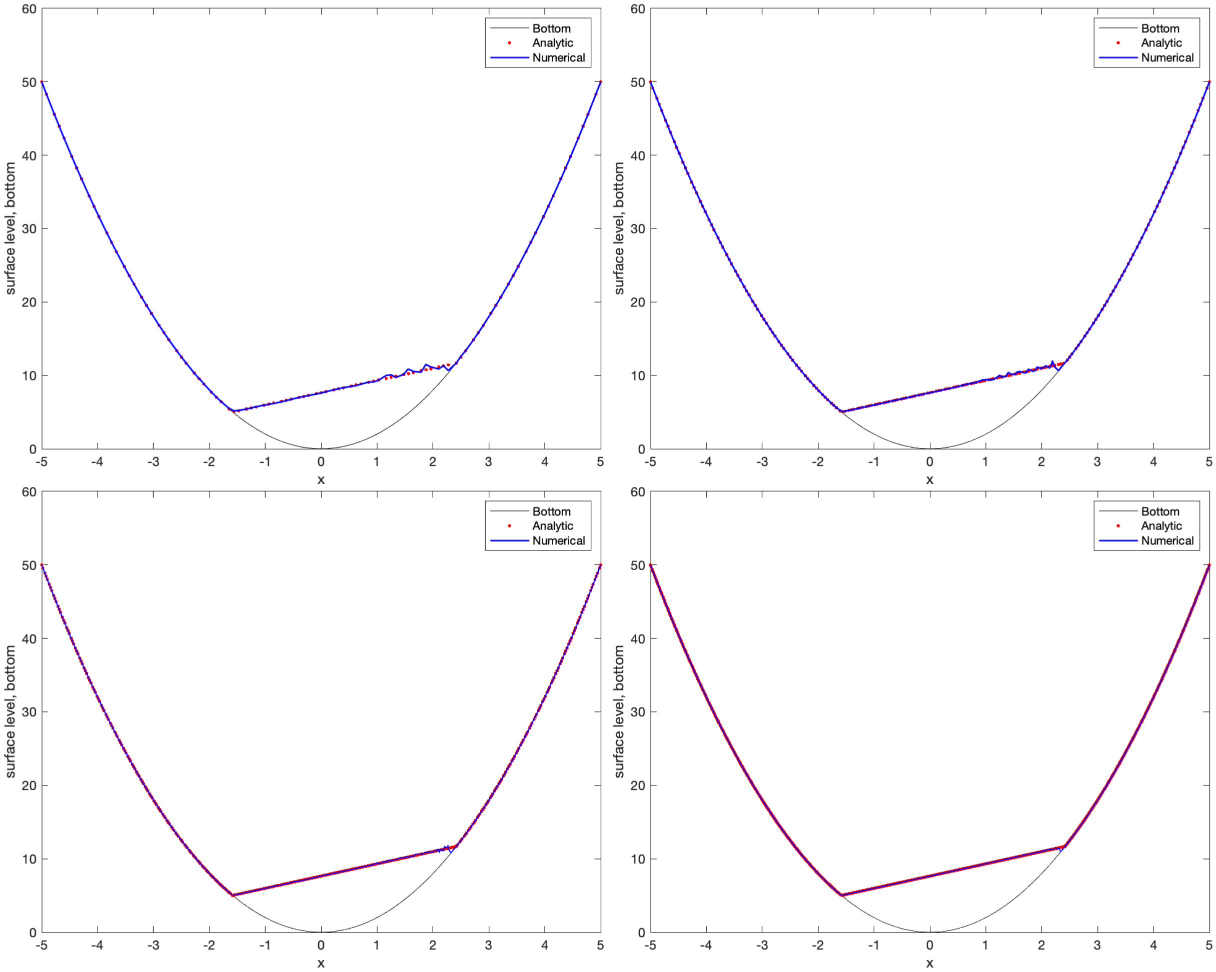


Figure 9: The surface level h + b and the bottom b for the parabolic bowl at $T = 1$ with node-wise convex limiting: $N = 3$; top left: $K = 32$; top right: $K = 64$; bottom left: $K = 128$; bottom right: $K = 256$

convergence rate of between $O(h)$ and $O(h^2)$. The node-wise limiting produces larger errors near the dry-wet interface, where the convex limiting scheme pushes towards the $O(h)$ low order scheme. The element-wise limiting, on the other hand, produces smaller errors throughout the entire wet domain for lower polynomial degree N. Element-wise limiting errors also converge slightly faster than node-wise limiting errors.

5.4 Dam break

This experiment is taken from [3, 46]. Similar dam break experiments and results can be found in [8, 13]. We utilize the same physical setting but use triangular meshes instead of quadrilateral meshes. The domain $[-1, 2] \times [-1, 1]$ is discretized using a 48×32 grid of quadrilaterals, and each of them is split into two triangular elements. We set $N = 3$ for the polynomial degree. The dam is modeled by imposing reflective boundary conditions along the y-axis with a break between

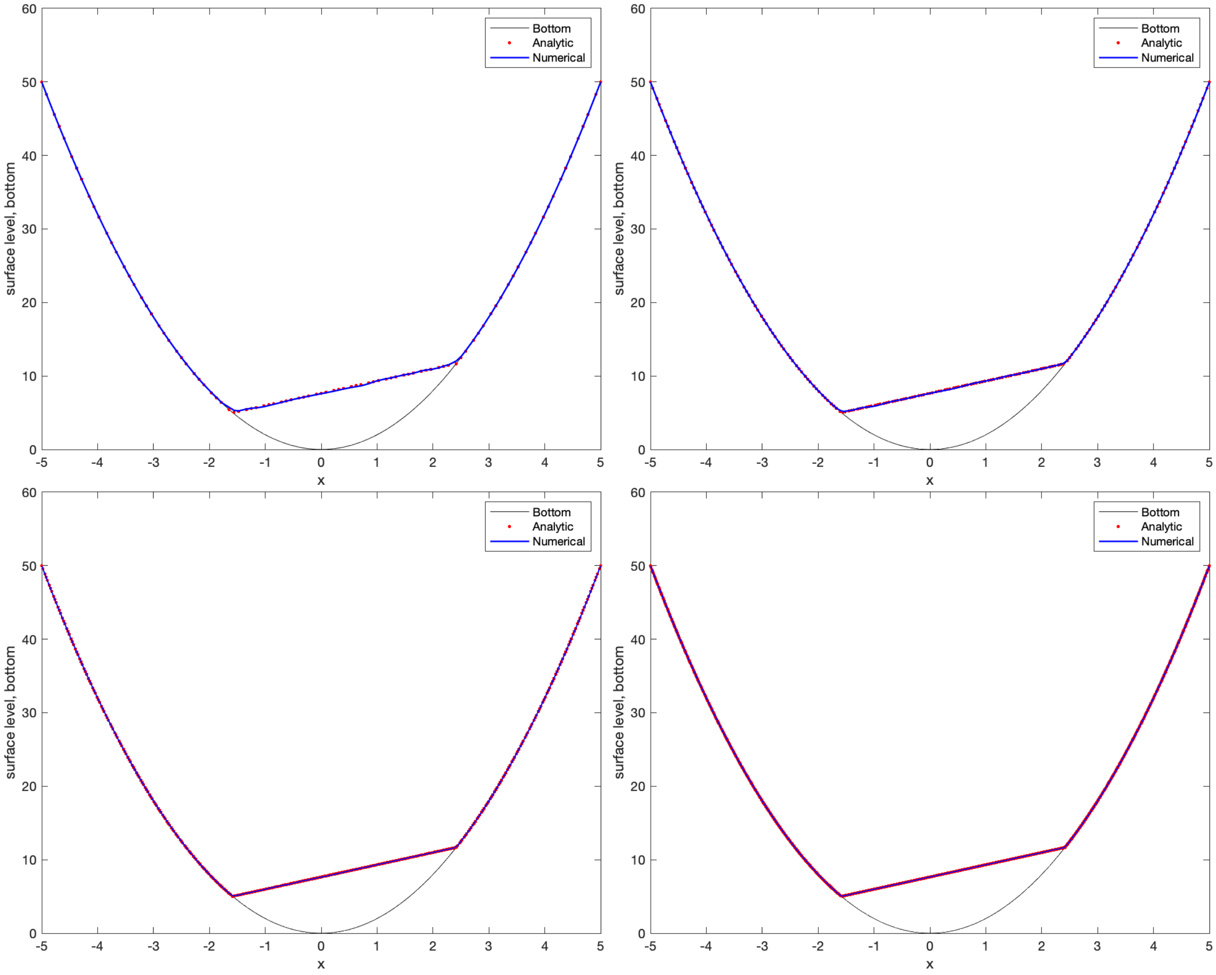


Figure 10: The surface level h + b and the bottom b for the parabolic bowl at $T = 1$: $N = 3$ with element-wise convex limiting; top left: $K = 32$; top right: $K = 64$; bottom left: $K = 128$; bottom right: $K = 256$

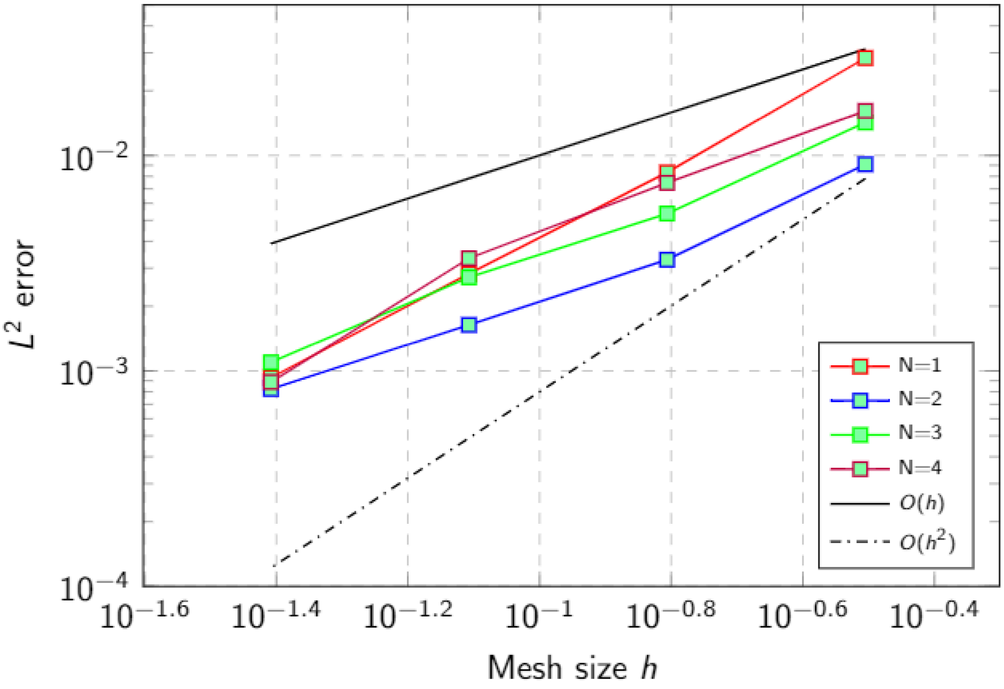


Figure 11: L^2 errors for the 1D oscillating bowl at T=0.5 using node-wise convex limiting with various mesh sizes and polynomial degrees

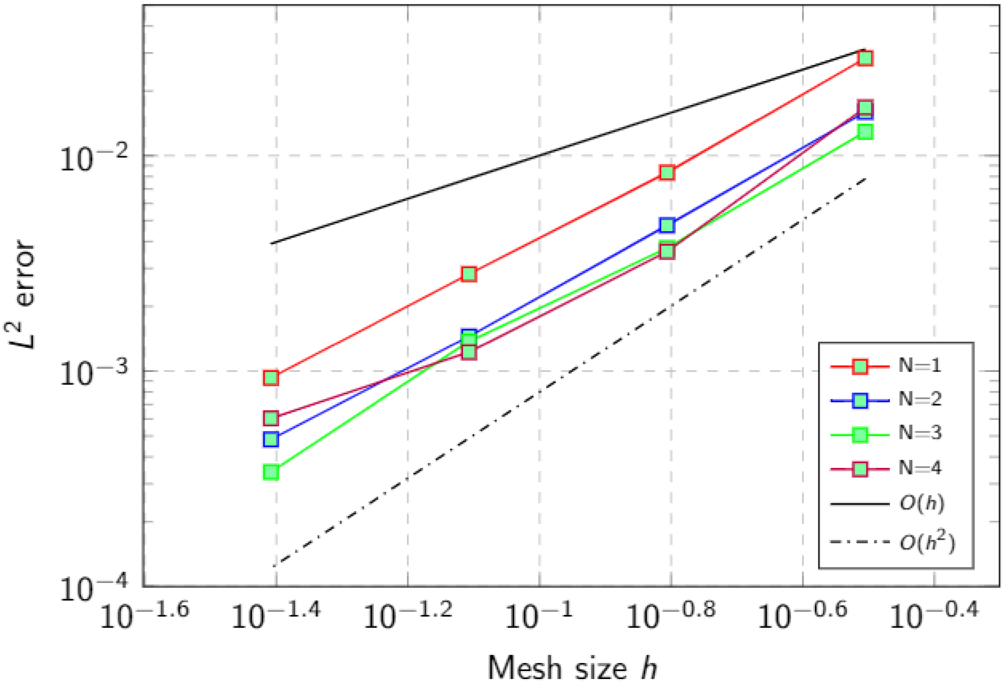


Figure 12: L^2 errors for the 1D oscillating bowl at T=0.5 using element-wise convex limiting with various mesh sizes and polynomial degrees

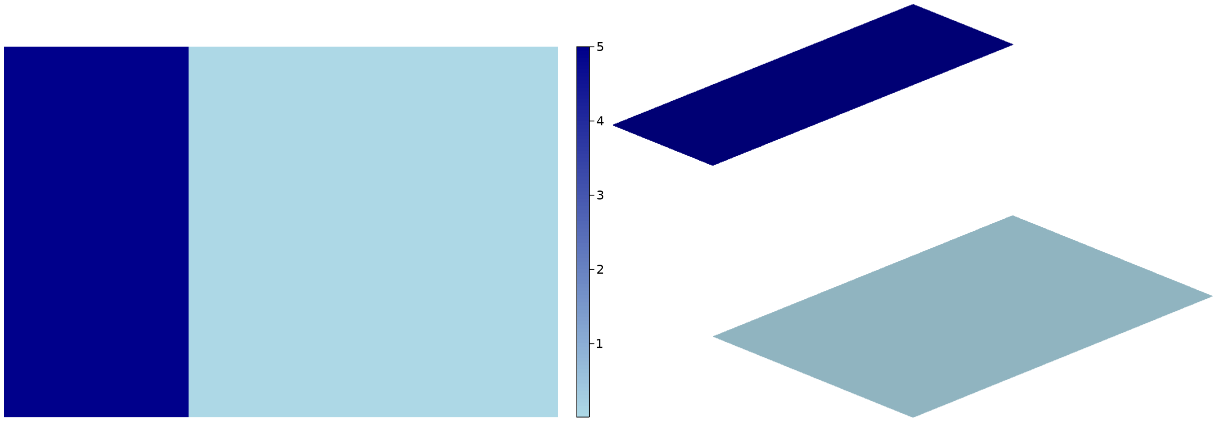


Figure 13: 2D dam break problem with initial water height

$y = -0.1$ and $y = 0.1$ to allow water to flow through, as shown in Fig. 13.

We start with a zero velocity and constant water heights on both sides of the dam. The bathymetry is set to $b = 0$ on both sides. We set the initial water height to be $h = 5$ on the left side of the dam and $h = tol$ on the right side as shown in Fig. 13. We plot the solutions from node-wise limiting, element-wise limiting, and the low order positivity preserving method at several different times in Fig. 14, Fig. 15, Fig. 16, and Fig. 17. We observe that the water falling from the left side of the dam to the right produces a wavefront in the lower half of the dam. This wave is discontinuous, so we observe some mesh dependent solution oscillations, but they are localized and do not cause the solution to blow up. The water heights remain positive during the entire numerical experiment. We have also tried simulations with polynomial degrees $N = 1$ and $N = 2$. Both of these simulations also remain stable throughout the experiment. The numerical solution of the parabolic dam break problem demonstrates that the entropy stable numerical method remains robust in the presence of shock discontinuities, while the use of convex limiting preserves positive water heights.

We compare the solutions from all three methods. We observe that the results from the node-wise limiting and the element-wise limiting schemes are very similar. They both become wavy near the right boundary. The results from the low order positivity preserving method are much more dissipative than the results from the convex limited schemes. The node-wise limiting scheme produces slightly larger oscillations; however, these oscillations may not be physical as they are not observed in other reference solutions in the literature [47, 3]. The numerical results from the element-wise limited scheme are more consistent with reference solutions from the literature. This may be due to the fact that an element-wise constant limiting factor preserves a semi-discrete entropy inequality [37].

5.5 Wave over a bump

The last experiment we perform is a 2D simulation of a wave passing over a bump. The bottom geometry is modeled by a Gaussian bump with a height such that the bump is never entirely submerged. This experiment provides a test of how our numerical solution handles a persistent dry area. We use polynomial degree $N = 3$ with a total of 8192 ($64 \times 64 \times 2$) elements. This experiment

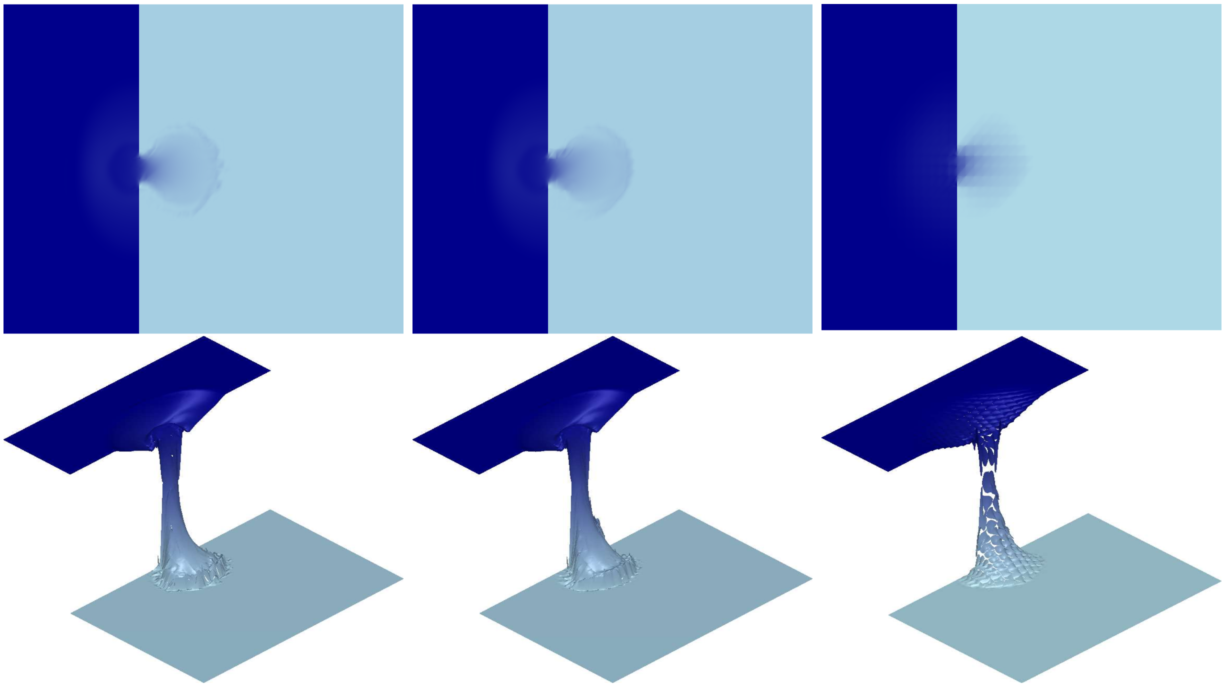


Figure 14: Top view (top row) and side view (bottom row) of the dam break problem at T = 0.25. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

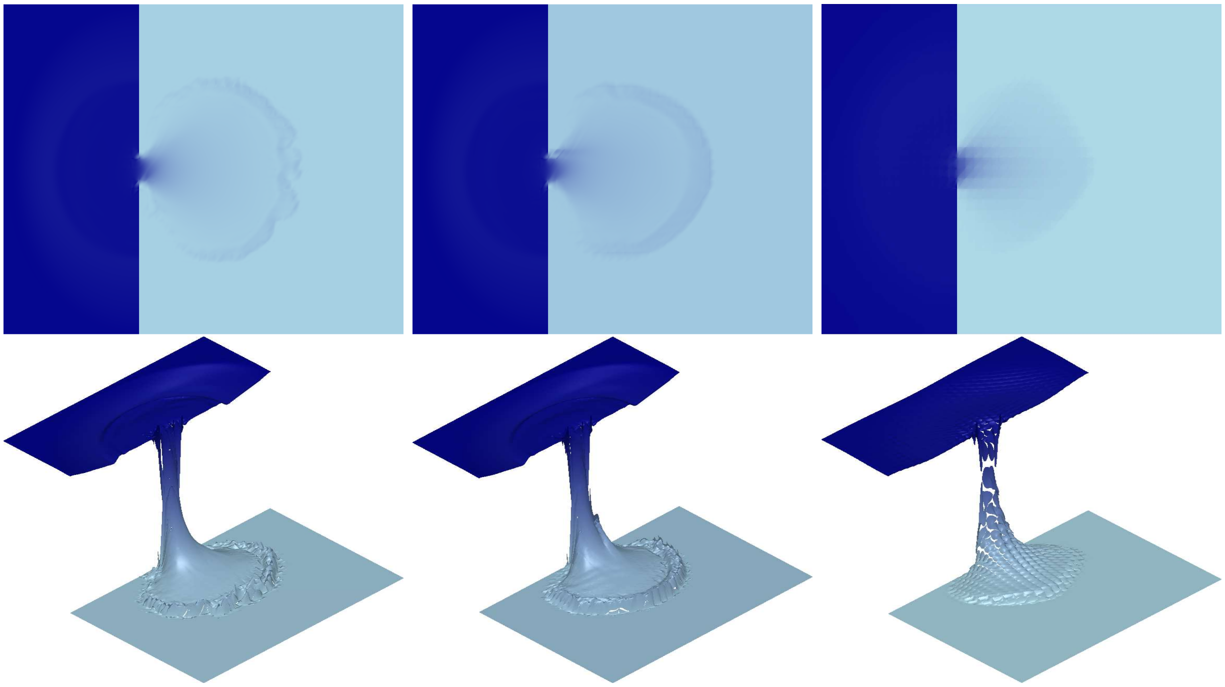


Figure 15: Top view (top row) and side view (bottom row) of the dam break problem at T = 0.5. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

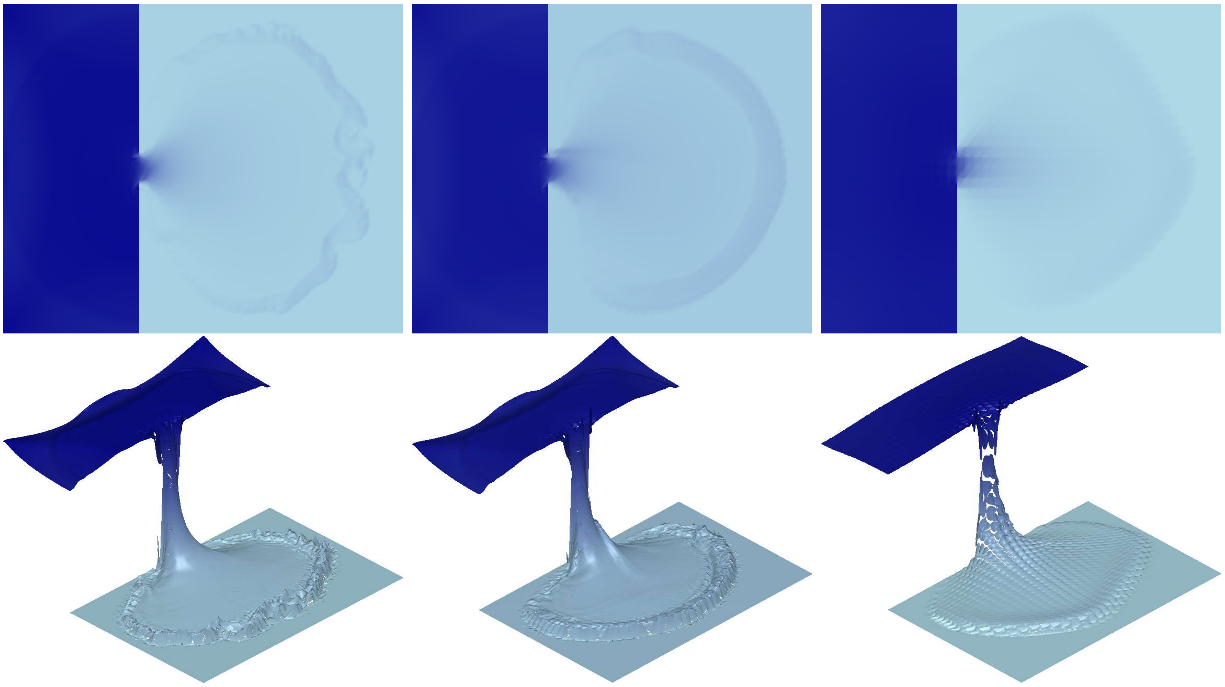


Figure 16: Top view (top row) and side view (bottom row) of the dam break problem at T = 0.75. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

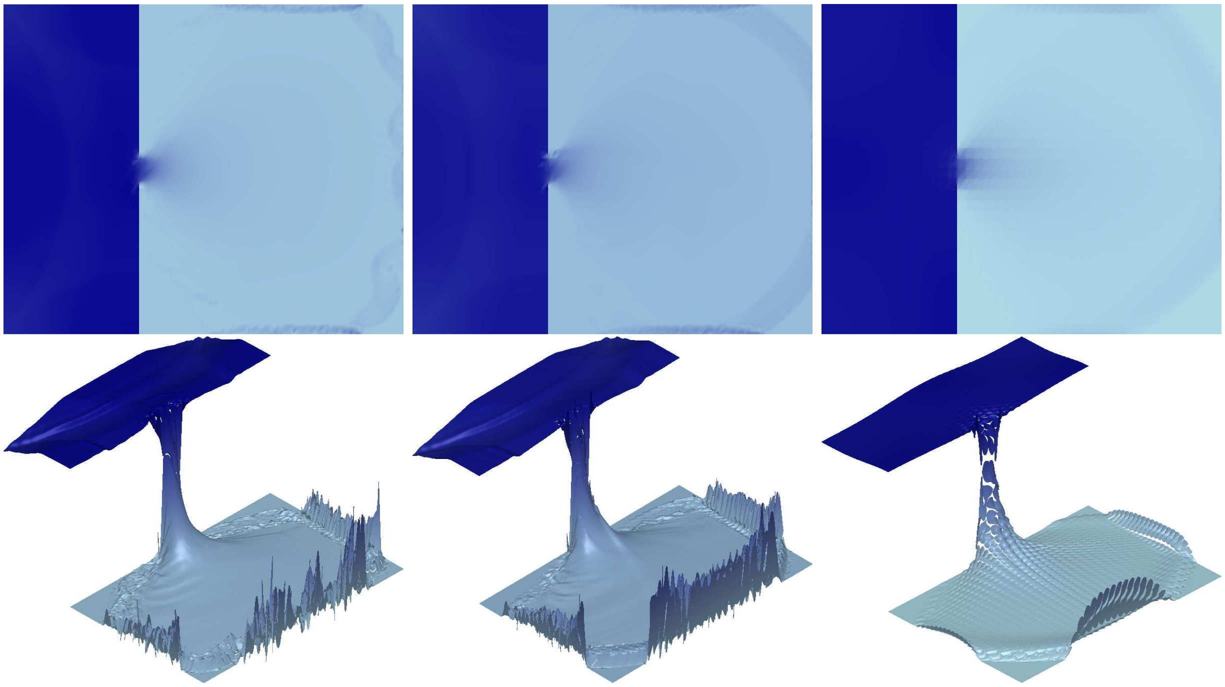


Figure 17: Top view (top row) and side view (bottom row) of the dam break problem at T = 1. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

runs on the domain $[-1,1]^2$ with periodic boundary conditions. The bottom topography is given by the following expression:

$$b(x,y) = 5\exp(-25(x^2+y^2)), \tag{5.16}$$

and the initial condition, as shown in Fig. 18, for this experiment is

$$h(x,y) = \max(0, \exp(-25((x.+0.5).^2)) + 2 - b), \tag{5.17}$$

$$u(x,y) = \begin{cases} 1 & \text{if } h > 2, \\ 0 & \text{otherwise,} \end{cases} \qquad v = 0. \tag{5.18}$$

We run this experiment up to time T = 1 and plot numerical solutions from three different schemes. In Fig. 19, Fig. 20, Fig. 21, and Fig. 22, we present the solutions from node-wise limiting scheme, element-wise limiting scheme, and the lower order positivity preserving scheme. For all schemes, we observe that the wave propagates along the x-axis, hits the bump and splits, and converges after passing it. We notice that our solution contains small oscillations at the wavefront but never blows up. The water height remains positive even in the presence of a dry area throughout the entire experiment. We observe that the results from the node-wise limiting scheme are very similar to the results from the element-wise limiting scheme for this problem.

5.6 Malpasset dam break

To validate the robustness of our positivity-preserving entropy stable DG method, we perform a large scale simulation of the flood resulting from the Malpasset dam failure, which occurred in southern France in 1959. The Malpasset dam was located in a narrow valley on Reyran River, about 12 km north of the town of Frejus. The dam failed catastrophically, flushing an enormous wave about 40 to 60 meters down the valley and causing massive destruction. We refer the reader to [48, 49, 50] for a detailed description of this problem.

This historical disaster provides a realistic and challenging setting in which to study the behavior of the proposed positivity-preserving method. We obtained the mesh and bathymetric data from [51]. The mesh consists of 18368 triangular elements, and the bathymetric data is given by 13541 measurements at points which are different from the mesh vertices. We interpolate the bathymetric data to quadrature points using nearest neighbor local least squares interpolation from [52]. Then, we project the data onto the degree N polynomial space to ensure that the bathymetry is continuous and approximated to high order accuracy. We follow a slightly simplified approach as in [37] and consider the dam to be a straight line between the points $(4701.18, 4143.41)$ and $(4655.5, 4392.1)$. The dam is marked in red in Fig. 23(a). The initial water heights in the reservoir are set to $\max\{100-b,\ tol\}$, where b is the bathymetry height. We then set the water heights in the ocean to be $\max\{0-b,\ tol\}$ where $b<0$.

To obtain more physically realistic solutions in this experiment, we also incorporate bottom friction in this simulation as in Eq. (3.5) in [53]. In particular, we add the following terms for the source terms for each node i:

$$\begin{bmatrix} 0 \\ -\frac{2gn^2(hu)_i||v||_2 m_i}{h_i^\gamma+\max\{h_i^\gamma, 2gn^2dt||v||_2\}} \\ -\frac{2gn^2(hv)_i||v||_2 m_i}{h_i^\gamma+\max\{h_i^\gamma, 2gn^2dt||v||_2\}} \end{bmatrix} \tag{5.19}$$

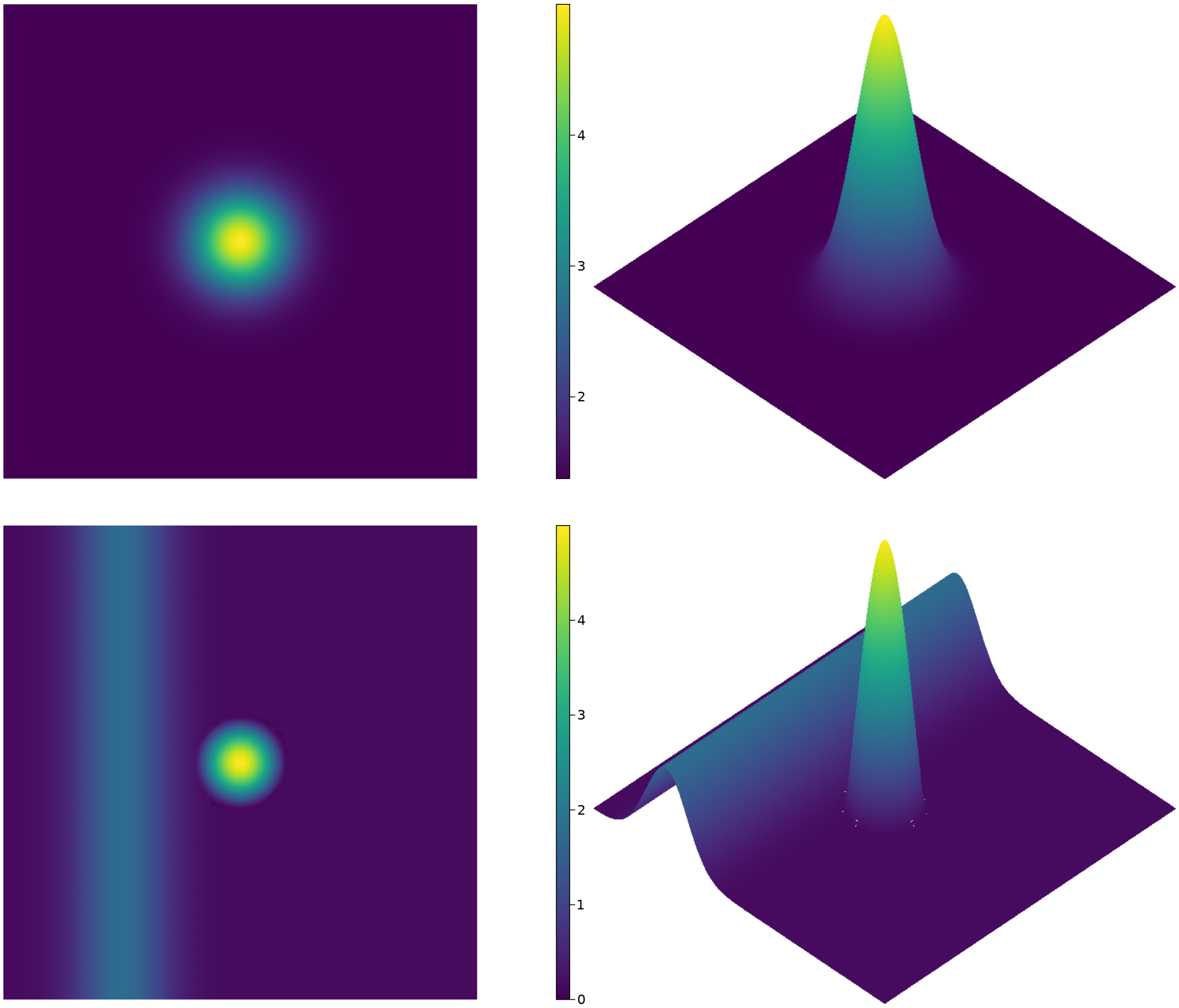


Figure 18: Bottom geometry and initial condition for wave over a bump: Top: top view and side view of bottom geometry. Bottom; top view and side view of initial water surface and dry area.

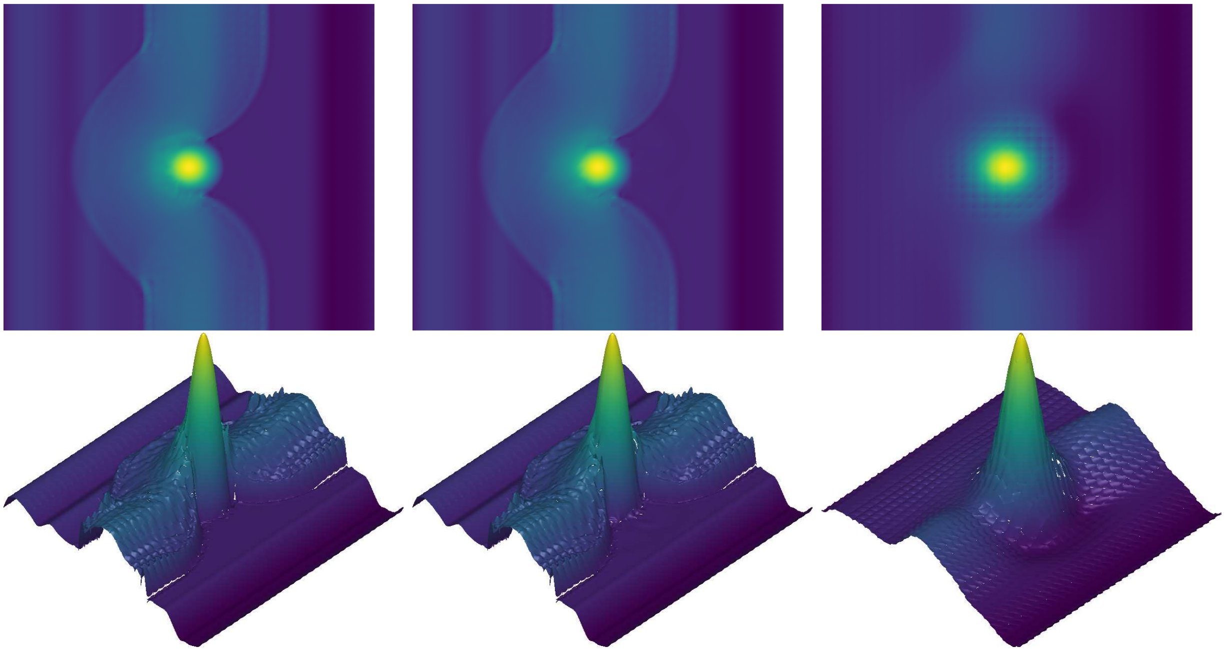


Figure 19: Top view (top row) and side view (bottom row) of water surface and dry area for wave over a bump at T = 0.25. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

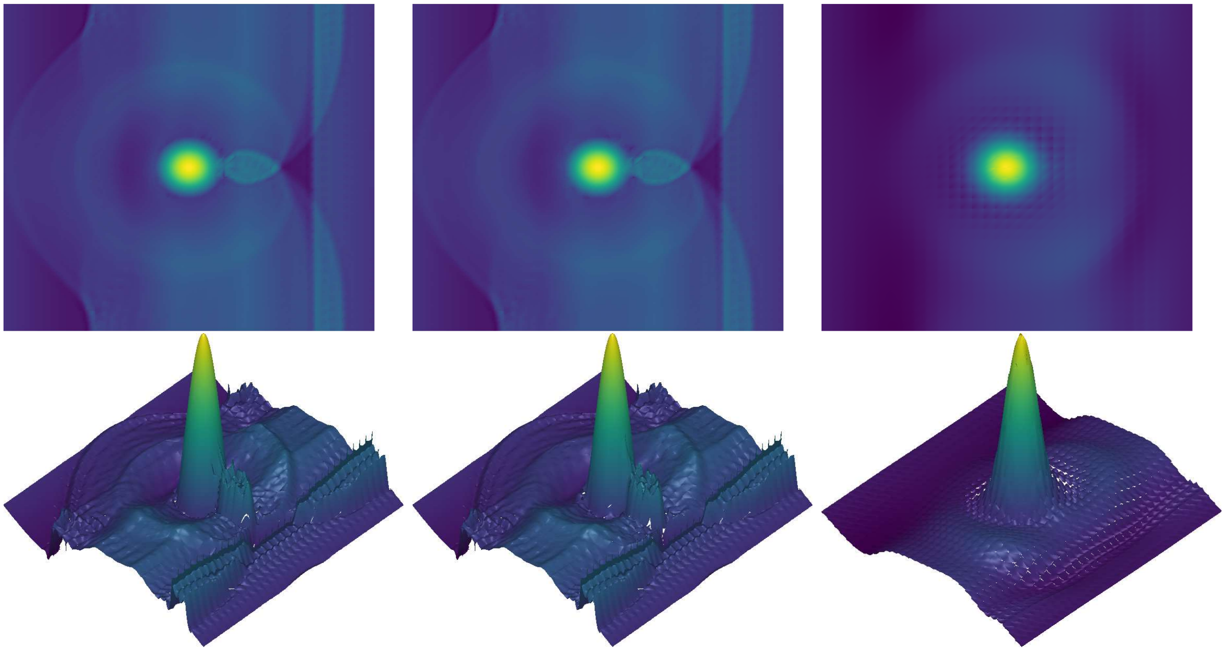


Figure 20: Top view (top row) and side view (bottom row) of water surface and dry area for wave over a bump at T = 0.5. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

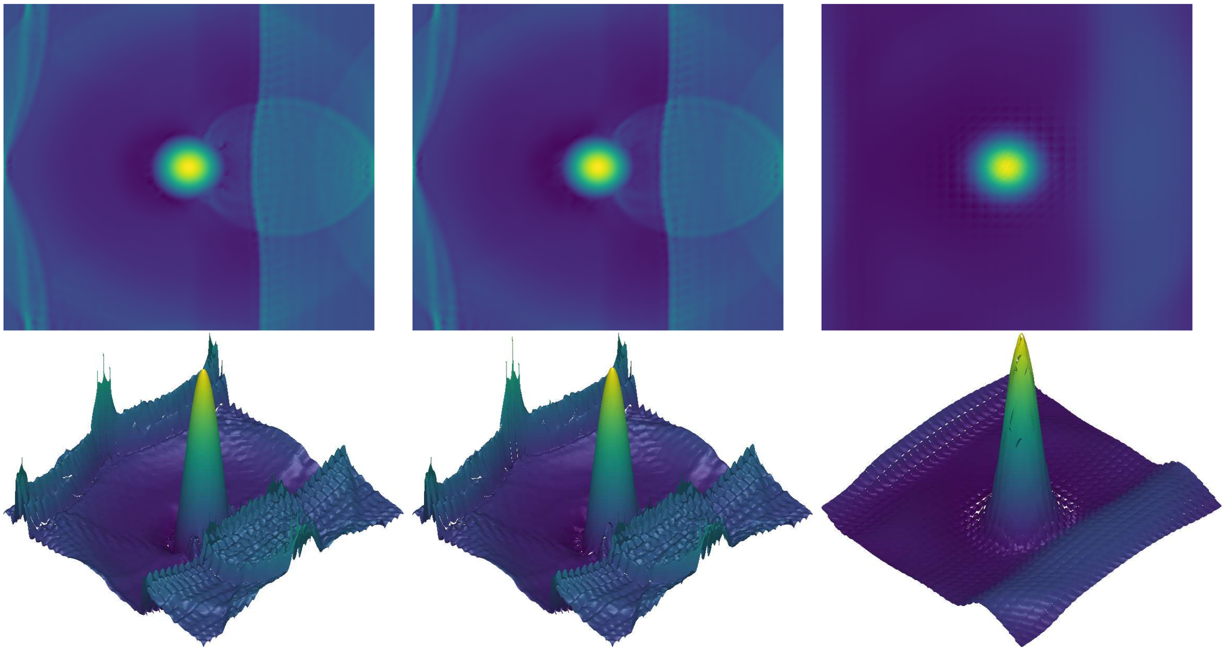


Figure 21: Top view (top row) and side view (bottom row) of water surface and dry area for wave over a bump at T = 0.75. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

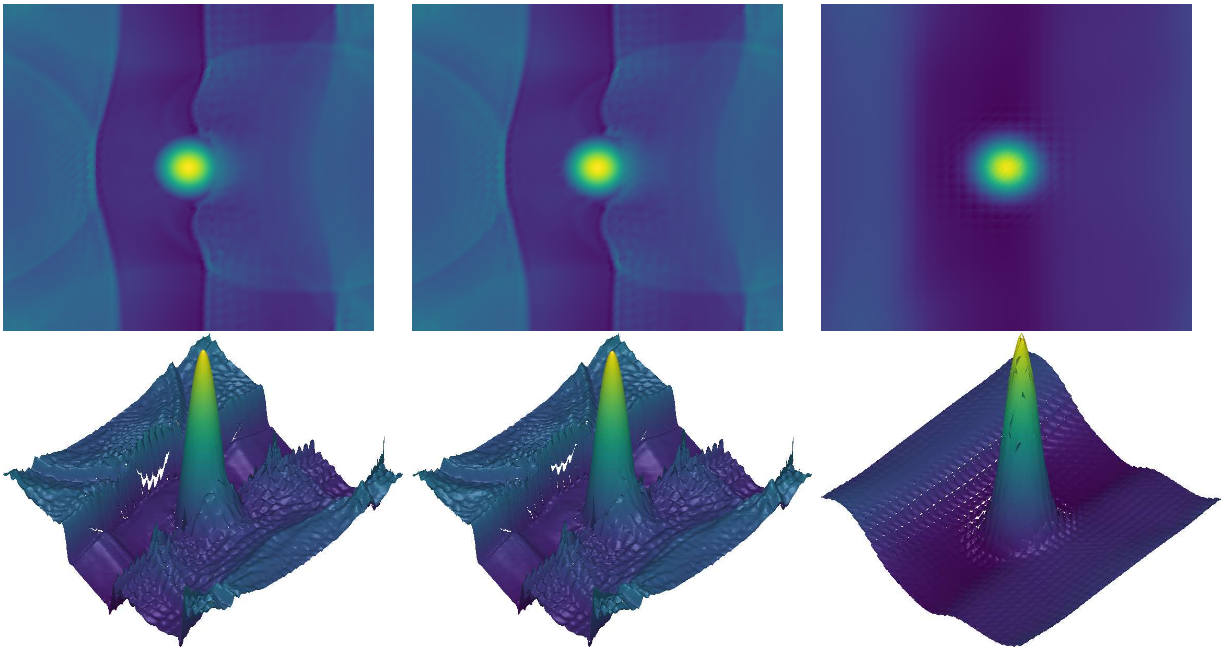


Figure 22: Top view (top row) and side view (bottom row) of water surface and dry area for wave over a bump at T = 1. Left: node-wise limiting. Middle: element-wise limiting. Right: low order scheme

where $g = 9.8$ denotes the gravitational acceleration, $||v||_2$ denotes the l^2 norm of the velocity, and dt denotes the time step. We use $\gamma = \frac{4}{3}$ and $n = 0.04$ for Manning's roughness coefficient on the entire domain. The bottom friction reduces the magnitude of the velocity of the flow and provides more physically realistic solutions in this simulation. Our numerical scheme is stable even in the absence of friction, but without friction the shallow water equations generate larger flow velocities that exceed typical water velocities in flash floods in nature [54].

Finally, we modify the element wise limiting strategy in this experiment to enforce a stronger *local* positivity condition, which improves robustness and enables larger time steps based on Theorem 3.1. We enforce the following conditions:

$$h \geq \epsilon h^L, \tag{5.20}$$

$$\ell_e := \ell_e h^H + (1 - \ell_e) h^L \geq \epsilon h^L, \tag{5.21}$$

where h^L denotes the low order solution, $\epsilon = 0.9$ and ℓ_e denotes the element-wise limiting factor. Not only are element-wise limiting factors cheaper to calculate, but this relaxed limiting scheme improves computational efficiency and bounds the water heights farther away from zero. This has the added advantage of avoiding small time step sizes due to spikes in the velocity, which can occur when computing the velocity by dividing the momentum by a small water height.

We use polynomials of degree $N = 1$ and $N = 2$ with final time $T = 3000$. We show in Fig. 23 and Fig. 24 the water heights at various times after the collapse of the dam. We observe that the water heights remain positive throughout the entire simulation. Our numerical schemes remain stable despite the initial discontinuity in the water heights at the dam, the presence of multiple wet/dry fronts, and highly variable bathymetric profiles. Our numerical results are comparable, in terms of the flood front and water heights, to those from [55, 48, 56, 57].

We notice that one of the most significant drawbacks of our numerical schemes is the small time steps it allows. We find that experiments with larger time steps either violate the positivity of the water heights or produce inaccurate and unrealistic solutions.

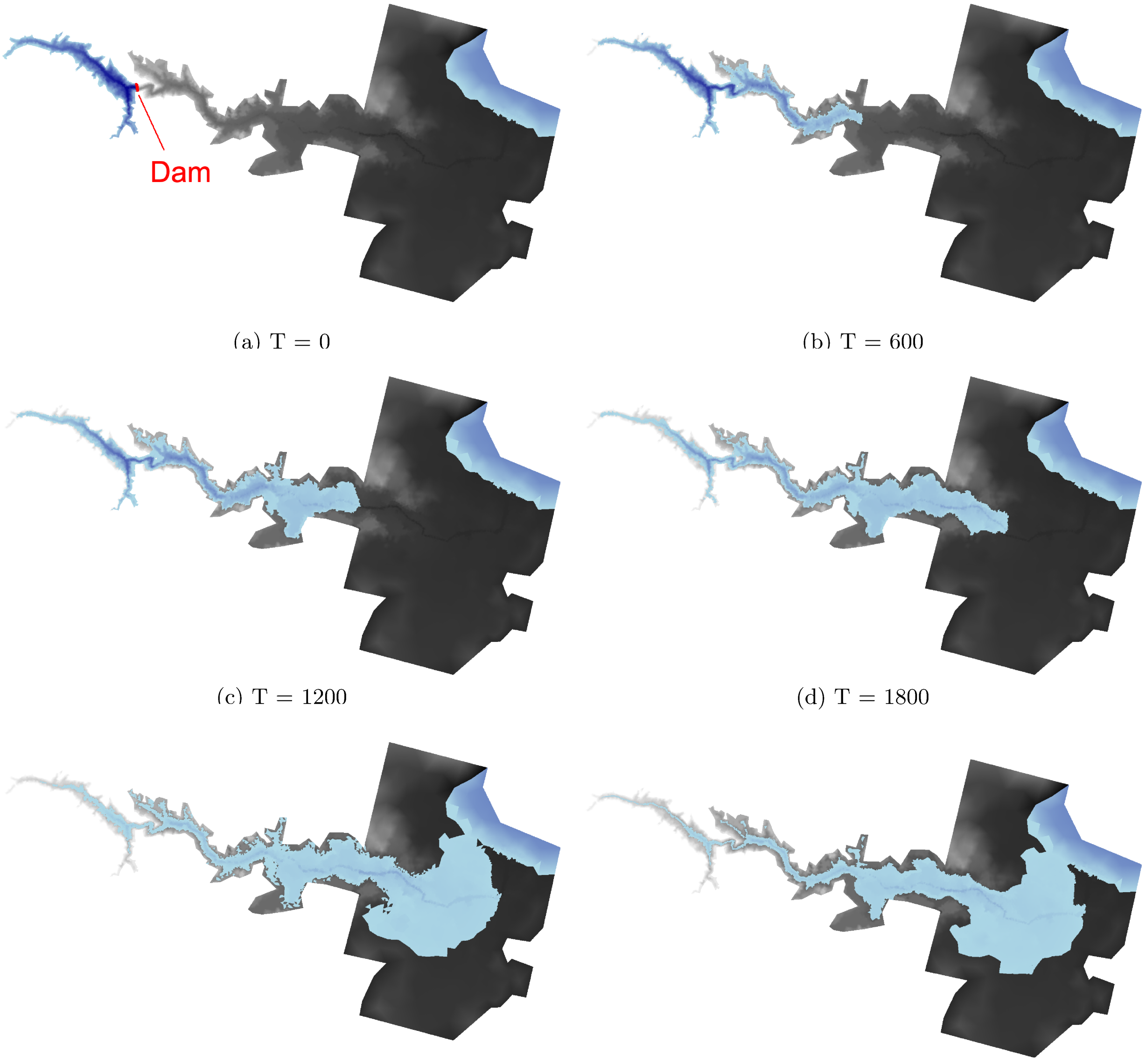
6 Conclusions

In this paper, we have presented a high order entropy stable positivity-preserving numerical scheme based on DG methods for the shallow water equations in 1D and on 2D with triangular meshes. This numerical method can preserve the positivity of water heights under an appropriate time step restriction. We apply the convex limiting technique on a low order positivity preserving method and a high order nodal SBP method to obtain this final scheme. This method can achieve high order of accuracy away from dry areas while remaining stable at the wet-dry front. We provide both 1D and 2D numerical experiments to demonstrate our theoretical findings.

Acknowledgement

Authors Xinhui Wu and Jesse Chan gratefully acknowledge support from the National Science Foundation under awards DMS-1719818, DMS-1712639, and DMS-1943186.

Author Nathaniel Trask acknowledges funding under the Collaboratory on Mathematics and Physics-Informed Learning Machines for Multiscale and Multiphysics Problems (PhILMs) project



(a) T = 0

(b) T = 600

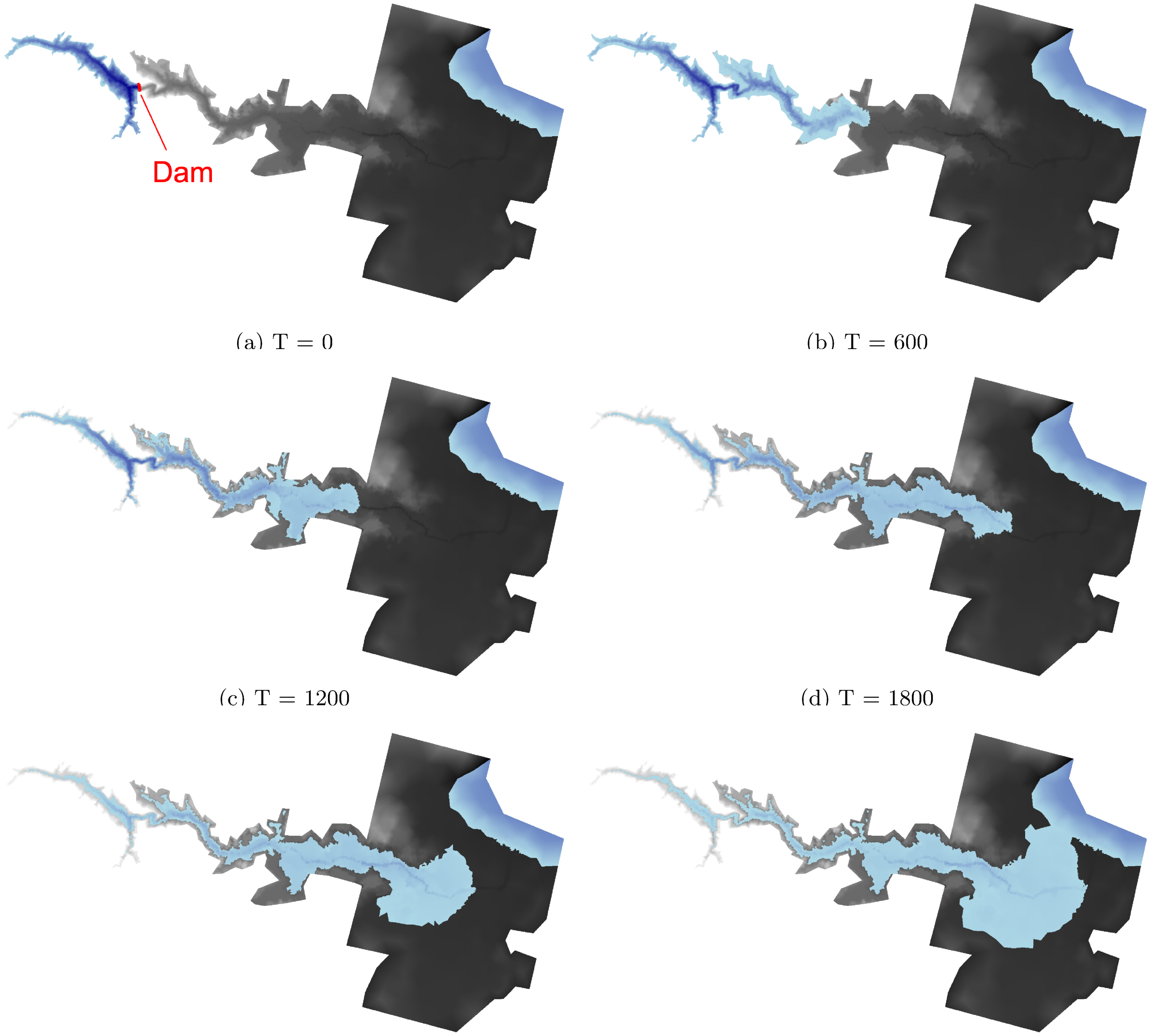
(c) T = 1200

(d) T = 1800

(e) T = 2400

(f) T = 3000

Figure 23: Top view of the water height for the Malpasset dam break problem at various times for $N = 1$. The color code is based on the initial water height.



(a) T = 0

(b) T = 600

(c) T = 1200

(d) T = 1800

(e) T = 2400

(f) T = 3000

Figure 24: Top view of the water height for the Malpasset dam break problem at various times for $N = 2$. The color code is based on the initial water height.

funded by DOE Office of Science (Grant number DE-SC001924) and the DOE Early Career program. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Appendix A: Construction of low order operators

In this appendix, we describe the procedures to compute lower order operators. We use a meshfree approach on the SBP nodes. We start by building a connectivity graph on the SBP nodes. Then, we solve a graph-Laplacian problem based on the connectivity graph to form the low order operators.

A.I Construction of sparse connectivity graph

To construct the low order operator $\mathbf{Q}_i^L$, we need to introduce a connectivity graph. This connectivity defines the adjacency matrix A from the previous section. We first use uniform radius $r = \alpha$ to define to determine the neighbors of each node, such that node $j \in I(i)$ if the distance between node j and i is less or equal to the radius r. However, this approach is less well suited to non-uniform node distributions, which is the case for the Gauss-Legendre quadrature node and Gauss-Lobatto quadrature nodes shown in Fig. 1 and Fig. 2, respectively. Then, we investigate the scaling of the radii calculated from coefficient α and the quadrature weight of the node raised to some power. So now the radius r_i for node i can be written as

$$r_i = \alpha(w_i)^p, \tag{A.1}$$

where w_i is the quadrature weight of the node i and p is some power. With our experimental approach, we test many different combinations of α and p and discover that $\alpha = 1.0$ and $p = \frac{1}{4}$ would produce sparsest connectivity graph. For illustration, we plot the circles with radii r_i on nodes with different quadrature weights in Fig. 25. The red circles define the neighborhoods of nodes. We observe that nodes with greater weights form neighborhoods with larger radii.

Notice that we still need the low order operator constructed from this connectivity graph to satisfy the requirements in (3.2). Therefore, α can not be arbitrarily small. In Fig. 26, we notice that the smaller radius can produce a much sparser low order operator by reducing the number of connected nodes. Sparsity can increase the accuracy of the algorithm. It also improves the computing speed because of the fewer number of non-zero terms in the matrix.

A.II Meshfree construction of low order summation-by-parts matrices

This setup defines an undirected finite graph (V, E), where the vertices consist of all the quadrature points. (j, k) is an edge of the graph, i.e., $(j, k) \in E$, iff $j \in I(k)$ and $k \in I(j)$. We use (V, E) to refer to the connectivity graph. Let r denote the radius of a ball. Then we define a adjacency matrix $\mathbf{A}$ where $\mathbf{A}_{jk} = 1$, for $j \neq k$ if the distance between node j and k is less or equal to r and $\mathbf{A}_{jk} = 0$ otherwise. $j \in I(k)$ iff $\mathbf{A}_{jk} = 1$, We let $\mathbf{A}_{jj} = |I(j)|$, the number of nodes adjacent to node j.

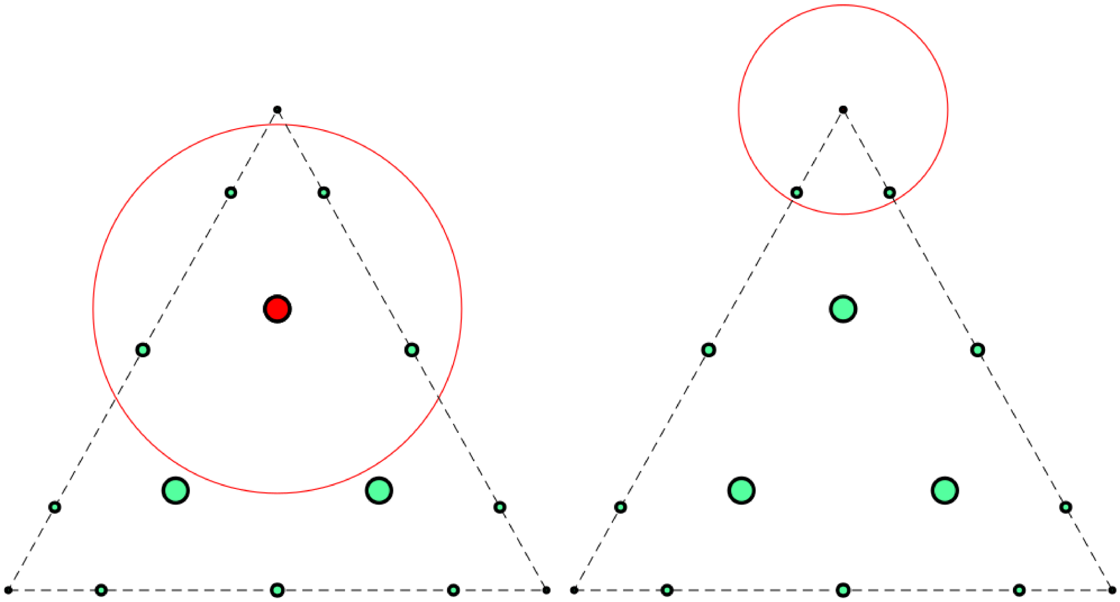


Figure 25: Left: neighborhood of a interior nodes with larger weights. Right: neighborhood of a boundary node with smaller weights. The sizes of the nodes are proportional to its quadrature weights. The central nodes of the neighborhoods are marked in red.

We first build a mesh-free low order differentiation operator $\mathbf{Q}_i^L$ in the ith coordinate direction with the summation-by-part property. We require that

$$\mathbf{Q}_i^L + (\mathbf{Q}_i^L)^T = \mathbf{B}_i = \text{diag}(\boldsymbol{n}_i \circ w_i^f), \tag{A.2}$$

$$\mathbf{Q}_i^L \mathbf{1} = \mathbf{0}, \tag{A.3}$$

We can rewrite Eq. (A.2) as

$$\mathbf{Q}_i^L = \boldsymbol{S} + \frac{1}{2}\mathbf{B}_i, \tag{A.4}$$

where $\boldsymbol{S}$ is skew-symmetric. We assume that $\boldsymbol{S}_{jk} = \phi_j - \phi_k$ for entries ϕ_j and ϕ_k of some “potential” vector Φ. Enforcing $\mathbf{Q}_i^L \mathbf{1} = \mathbf{0}$ then implies that

$$-\boldsymbol{S}\mathbf{1} = \frac{1}{2}\mathbf{B}_i\mathbf{1} \implies \underbrace{\sum_{k \in I(j)} (\phi_k - \phi_j)}_{-\boldsymbol{S}\mathbf{1}} = \frac{1}{2}(\mathbf{B}_i)_{jj} \tag{A.5}$$

This is equivalent to a Neumann-type graph-Laplacian problem in [58]. We can define the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal degree matrix whose entries are the degrees of each vertex. Then (A.5) can be written as $\mathbf{L}\boldsymbol{\Phi} = \frac{1}{2}\mathbf{B}\mathbf{1}$, and we solve the graph Laplacian problem subject to $\boldsymbol{\Phi}^T\mathbf{1} = \mathbf{0}$, where $\boldsymbol{\Phi}_j = \phi_j$. We solve the system with Lagrange multiplier and use the entries in vector $\boldsymbol{\Phi}$ to define the entries in $\boldsymbol{S}$. Last, we use Eq. (A.4) to obtain the differential operator $\mathbf{Q}_i^L$.

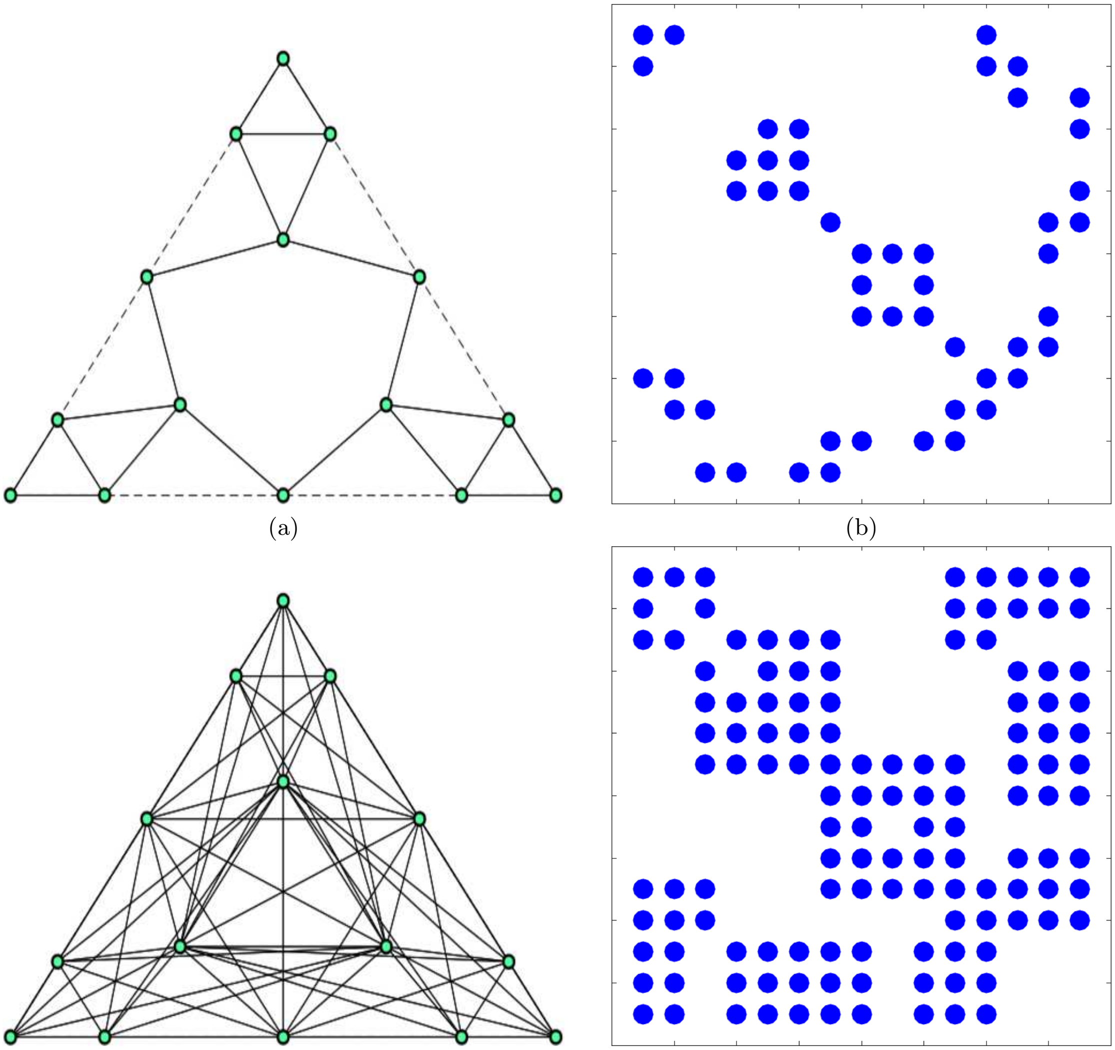


Figure 26: (a) Connectivity map of quadrature nodes with $\alpha = 1.0$. (b) Spy diagram for low order operator with $\alpha = 1.0$. (c) Connectivity map of quadrature nodes with $\alpha = 2.5$. (d) Spy diagram for low order operator with $\alpha = 2.5$.

Appendix B: Effect of sparsity of the low order operators on numerical dissipation

In this appendix, we demonstrate the effect of sparsity of low order operators on the quality of the low order positivity-preserving numerical solution. We test the low order scheme with low order operators constructed using different connectivity graphs. We utilize a sine wave initial condition on mesh defined on $[-1,1] \times [-1,1]$ with a 16×16 grid with periodic boundary conditions and a flat bottom topography. The initial conditions for this experiment are

$$h(x,y) = \sin(\pi x), \qquad hu(x,y) = hv(x,y) = 0. \tag{B.6}$$

We run each low order scheme up to $T = 0.1$ with different connection radii α and present the results in Fig. 27. We observe that the solutions become more dissipative as the radii α increases. The larger α becomes, the more connected the quadrature nodes are, resulting in graph viscosity being applied between more pairs of nodes. Choosing smaller values for α results in sparser low order operators, which sharpens the numerical solution and avoids introducing extraneous artificial dissipation.

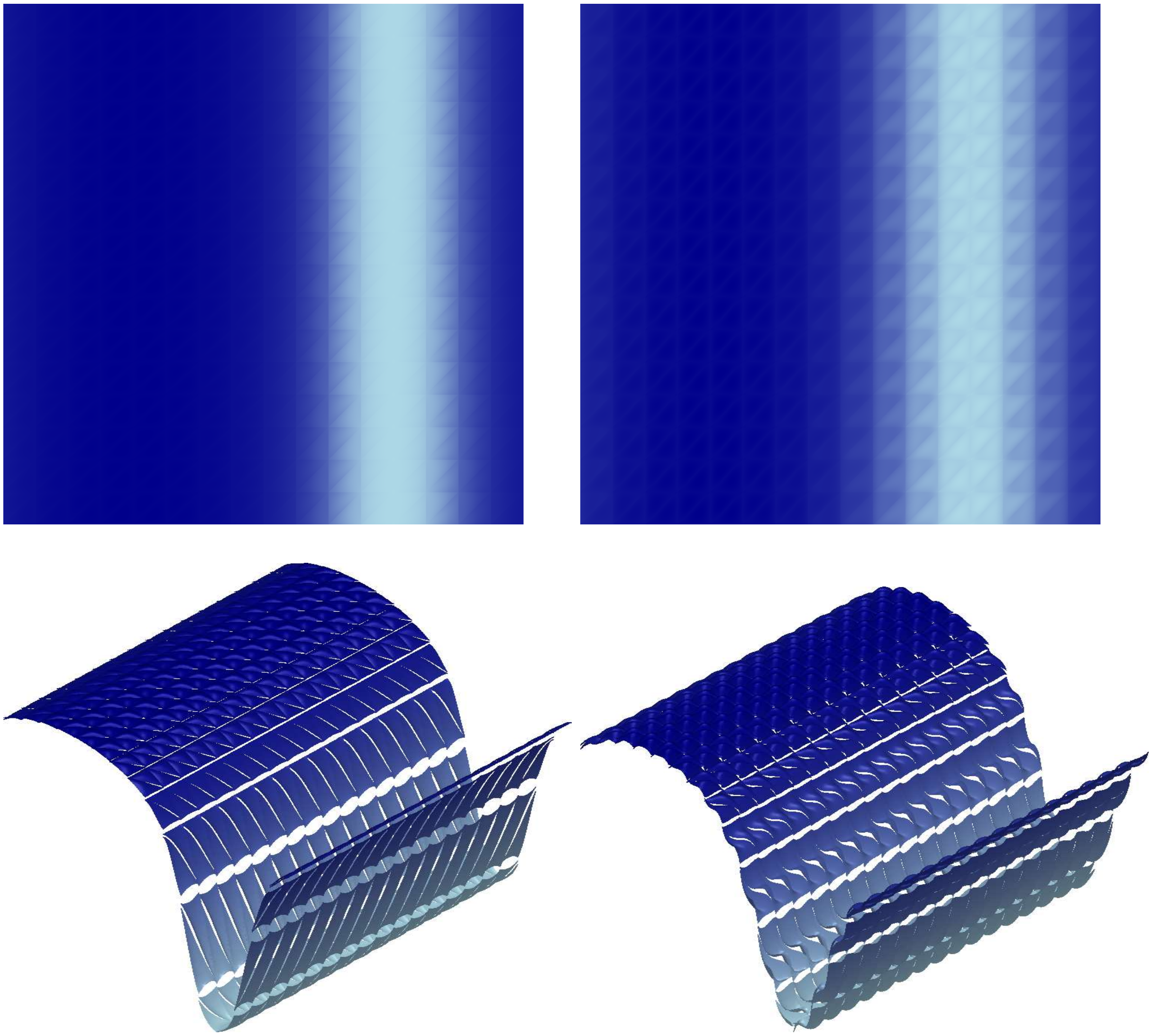


Figure 27: Sine wave on low order scheme with top view (top row) and side view (bottom row) at $T = 1$. Left: $\alpha = 1.0$. Right: $\alpha = 2.25$

Appendix C: Shallow water equations in 1D and 2D

In this appendix, we introduce the shallow water equations in 1D and 2D, as well as their flux and entropy functions.

For the 2D shallow water equations, the conservative variable is

$$\boldsymbol{u} = [h, hu, hv]^T, \tag{C.7}$$

and the flux functions are

$$\boldsymbol{f}_1 = [hu, hu^2 + gh^2/2, huv]^T \tag{C.8}$$

$$\boldsymbol{f}_2 = [hv, huv, hv^2 + gh^2/2]^T. \tag{C.9}$$

We can derive the 1D shallow water equations from the 2D shallow water equations by assuming a rectangular domain with length L_x and L_y in the x and y directions, respectively. If $L_y \ll L_x$ and wall boundary conditions are imposed, then we expect v to be small and h, u to be near-constant along the y-direction. These simplifications result in the 1D shallow water equations

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + gh^2/2 \end{bmatrix} = \boldsymbol{S}. \tag{C.10}$$

The mathematical entropy for the shallow water equations corresponds to total energy, and is given by

$$S(\boldsymbol{u}) = \frac{1}{2}\left(h \left\|U\right\|^2 + gh^2\right)$$

where $\left\|U\right\|^2 = u^2$ in 1D and $\left\|U\right\|^2 = u^2 + v^2$ in 2D. The entropy variables $\boldsymbol{v}$ in 2D are given by

$$v_1 = gh - \frac{1}{2}\left\|U\right\|^2, \qquad v_2 = u, \qquad v_3 = v.$$

In 1D, the entropy variables are simply $\boldsymbol{v} = [v_1, v_2]^T$. The inverse mapping in 2D is given by

$$h = \frac{v_1 + \frac{1}{2}\left\|U\right\|^2}{g}, \qquad hu = \frac{v_1 + \frac{1}{2}\left\|U\right\|^2}{g} v_2 = hv_2, \qquad hv = \frac{v_1 + \frac{1}{2}\left\|U\right\|^2}{g} v_3 = hv_3.$$

where we can compute $\left\|U\right\|^2 = v_2^2 + v_3^2$ in terms of the entropy variables. The inverse mapping in 1D follows by ignoring hv and setting $\left\|U\right\|^2 = v_2^2$.

The source terms of the shallow water equations that correspond to continuous bottom topographies are define as:

$$\boldsymbol{S} = \begin{bmatrix} 0 \\ -ghb_x \\ -ghb_y \end{bmatrix} \text{ in 2D} \quad \text{ and } \quad \boldsymbol{S} = \begin{bmatrix} 0 \\ -ghb_x \end{bmatrix} \text{ in 1D,} \tag{C.11}$$

where the subscripts $(.)_x$ or $(.)_y$ denote directional derivatives along x and y axis, respectively.

References

[1] Clint N Dawson, Ethan J Kubatko, Joannes J Westerink, Corey Trahan, Christopher Mirabito, Craig Michoski, and Nishant Panda. Discontinuous Galerkin methods for modeling hurricane storm surge. *Advances in Water Resources*, 34(9):1165–1176, 2011.

[2] Muhammad Akbar and Shahrouz Aliabadi. Hybrid numerical methods to solve shallow water equations for hurricane induced storm surge modeling. *Environmental modelling & software*, 46:118–128, 2013.

[3] Niklas Wintermeyer, Andrew R Winters, Gregor J Gassner, and Timothy Warburton. An entropy stable discontinuous Galerkin method for the shallow water equations on curvilinear meshes with wet/dry fronts accelerated by GPUs. *Journal of Computational Physics*, 375:447–480, 2018.

[4] Clint N Dawson and Christopher M Mirabito. The shallow water equations. *University of Texas, Austin*, 29, 2008.

[5] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[6] Clint Dawson and Jennifer Proft. Discontinuous and coupled continuous/discontinuous Galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 191(41-42):4721–4746, 2002.

[7] Francis X Giraldo, Jan S Hesthaven, and Tim Warburton. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181(2):499–525, 2002.

[8] Niklas Wintermeyer, Andrew R Winters, Gregor J Gassner, and David A Kopriva. An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. *Journal of Computational Physics*, 340:200–242, 2017.

[9] Jesse Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *Journal of Computational Physics*, 362:346–374, 2018.

[10] Xiao Wen, Wai Sun Don, Zhen Gao, and Yulong Xing. Entropy stable and well-balanced discontinuous galerkin methods for the nonlinear shallow water equations. *Journal of Scientific Computing*, 83:1–32, 2020.

[11] Gregor J Gassner, Andrew R Winters, and David A Kopriva. A well balanced and entropy conservative discontinuous Galerkin spectral element method for the shallow water equations. *Applied Mathematics and Computation*, 272:291–308, 2016.

[12] Sebastian Noelle, Yulong Xing, and Chi-Wang Shu. High-order well-balanced finite volume WENO schemes for shallow water equation with moving water. *Journal of Computational Physics*, 226(1):29–58, 2007.

[13] Ulrik S Fjordholm, Siddhartha Mishra, and Eitan Tadmor. Well-balanced and energy stable schemes for the shallow water equations with discontinuous topography. *Journal of Computational Physics*, 230(14):5587–5609, 2011.

[14] Yulong Xing. Exactly well-balanced discontinuous Galerkin methods for the shallow water equations with moving water equilibrium. *Journal of Computational Physics*, 257:536–553, 2014.

[15] Chris Nielsen and Colin Apelt. Parameters affecting the performance of wetting and drying in a two-dimensional finite element long wave hydrodynamic model. *Journal of Hydraulic Engineering*, 129(8):628–636, 2003.

[16] Yulong Xing, Xiangxiong Zhang, and Chi-Wang Shu. Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. *Advances in Water Resources*, 33(12):1476–1493, 2010.

[17] Emmanuel Audusse and Marie-Odile Bristeau. A well-balanced positivity preserving "second-order" scheme for shallow water flows on unstructured meshes. *Journal of Computational Physics*, 206(1):311–333, 2005.

[18] Yulong Xing and Xiangxiong Zhang. Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes. *Journal of Scientific Computing*, 57(1):19–41, 2013.

[19] Yi Jiang and Hailiang Liu. An invariant-region-preserving (IRP) limiter to DG methods for compressible Euler equations. In *XVI International Conference on Hyperbolic Problems: Theory, Numerics, Applications*, pages 71–83. Springer, 2016.

[20] Tianheng Chen and Chi-Wang Shu. Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes. *CSIAM Transactions on Applied Mathematics*, 1(1):1–52, 2020.

[21] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, 2017.

[22] Xinhui Wu, Ethan J Kubatko, and Jesse Chan. High-order entropy stable discontinuous Galerkin methods for the shallow water equations: curved triangular meshes and GPU acceleration. *Computers & Mathematics with Applications*, 82:179–199, 2021.

[23] Xinhui Wu and Jesse Chan. Entropy stable discontinuous Galerkin methods for nonlinear conservation laws on networks and multi-dimensional domains. *Journal of Scientific Computing*, 87(3):1–34, 2021.

[24] Jean-Luc Guermond and Bojan Popov. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 54(4):2466–2489, 2016.

[25] Jean-Luc Guermond, Bojan Popov, and Ignacio Tomas. Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 347:143–175, 2019.

[26] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, 2018.

[27] Michael S Mock. Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1):70–88, 1980.

[28] Travis C Fisher and Mark H Carpenter. High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. *Journal of Computational Physics*, 252:518–557, 2013.

[29] Gregor J Gassner, Andrew R Winters, Florian J Hindenlang, and David A Kopriva. The BR1 scheme is stable for the compressible Navier–Stokes equations. *Journal of Scientific Computing*, 77(1):154–200, 2018.

[30] Gregor J Gassner, Andrew R Winters, and David A Kopriva. Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. *Journal of Computational Physics*, 327:39–66, 2016.

[31] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987.

[32] Eitan Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numerica*, 12:451–512, 2003.

[33] Mark H Carpenter, Travis C Fisher, Eric J Nielsen, and Steven H Frankel. Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces. *SIAM Journal on Scientific Computing*, 36(5):B835–B867, 2014.

[34] Will Pazner. Sparse invariant domain preserving discontinuous Galerkin methods with subcell convex limiting. *Computer Methods in Applied Mechanics and Engineering*, 382:113876, 2021.

[35] Benoit Perthame and Youchun Qiu. A variant of Van Leer's method for multidimensional systems of conservation laws. *Journal of Computational Physics*, 112(2):370–381, 1994.

[36] Benoit Perthame and Chi-Wang Shu. On positivity preserving finite volume schemes for Euler equations. *Numerische Mathematik*, 73(1):119–130, 1996.

[37] Sebastian Hennemann, Andrés M Rueda-Ramírez, Florian J Hindenlang, and Gregor J Gassner. A provably entropy stable subcell shock capturing approach for high order split form DG for the compressible Euler equations. *Journal of Computational Physics*, 426:109935, 2021.

[38] Endre Süli and David F Mayers. *An Introduction to Numerical Analysis*. Cambridge university press, 2003.

[39] Jesse Chan, Zheng Wang, Axel Modave, Jean-François Remacle, and Tim Warburton. GPU-accelerated discontinuous Galerkin methods on hybrid meshes. *Journal of Computational Physics*, 318:142–168, 2016.

[40] Timothy Warburton and Jan S Hesthaven. On the constants in hp-finite element trace inverse inequalities. *Computer methods in applied mechanics and engineering*, 192(ARTICLE):2765–2773, 2003.

[41] Qiuhua Liang and Fabien Marche. Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in water resources*, 32(6):873–884, 2009.

[42] Pascal Azerad, Jean-Luc Guermond, and Bojan Popov. Well-balanced second-order approximation of the shallow water equation with continuous finite elements. *SIAM Journal on Numerical Analysis*, 55(6):3203–3224, 2017.

[43] Shinhoo Kang, Francis X Giraldo, and Tan Bui-Thanh. IMEX HDG-DG: A coupled implicit hybridized discontinuous Galerkin and explicit discontinuous Galerkin approach for shallow water systems. *Journal of Computational Physics*, 401:109010, 2020.

[44] Mario Ricchiuto and Andreas Bollermann. Stabilized residual distribution for shallow water simulations. *Journal of Computational Physics*, 228(4):1071–1115, 2009.

[45] Joe Sampson, Alan Easton, and Manmohan Singh. Moving boundary shallow water flow above parabolic bottom topography. *Anziam Journal*, 47:C373–C387, 2005.

[46] Mária Lukácová-Medvid'ová and Eitan Tadmor. On the entropy stability of the Roe-type finite volume methods. In *Proceedings of the twelfth international conference on hyperbolic problems, American Mathematical Society, eds. J.-G. Liu et al*, volume 67, 2009.

[47] Eitan Tadmor and Weigang Zhong. Energy-preserving and stable approximations for the two-dimensional shallow water equations. In *Mathematics and computation, a contemporary view*, pages 67–94. Springer, 2008.

[48] Jean-Michel Hervouet and Alain Petitjean. Malpasset dam-break revisited with two-dimensional computations. *Journal of hydraulic research*, 37(6):777–788, 1999.

[49] Alessandro Valiani, Valerio Caleffi, and Andrea Zanni. Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. *Journal of Hydraulic Engineering*, 128(5):460–472, 2002.

[50] DL George. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959). *International Journal for Numerical Methods in Fluids*, 66(8):1000–1018, 2011.

[51] Longxiang Li. NDG-FEM Malpasset 2D. `https://github.com/li12242/NDG-FEM/tree/master/Application/SWE/SWE2d/Benchmark/\%40Malpasset2d`, 2018.

[52] Emil Ljungskog. ScatteredInterpolation.jl. `https://github.com/eljungsk/ScatteredInterpolation.jl`, 2021.

[53] Jean-Luc Guermond, Manuel Quezada de Luna, Bojan Popov, Christopher E Kees, and Matthew W Farthing. Well-balanced second-order finite element approximation of the shallow water equations with friction. *SIAM Journal on Scientific Computing*, 40(6):A3873–A3901, 2018.

[54] Darren Lumbroso and Eric Gaume. Reducing the uncertainty in indirect estimates of extreme flash flood discharges. *Journal of hydrology*, 414:16–30, 2012.

[55] Chiara Biscarini, Silvia Di Francesco, Elena Ridolfi, and Piergiorgio Manciola. On the simulation of floods in a narrow bending valley: The malpasset dam break case study. *Water*, 8(11):545, 2016.

[56] Yuxin Huang, Ningchuan Zhang, and Yuguo Pei. Well-balanced finite volume scheme for shallow water flooding and drying over arbitrary topography. *Engineering Applications of Computational Fluid Mechanics*, 7(1):40–54, 2013.

[57] Gaurav Savant, Charlie Berger, Tate O McAlpin, and Jennifer N Tate. Efficient implicit finite-element hydrodynamic model for dam and levee breach. *Journal of Hydraulic Engineering*, 137(9):1005–1018, 2011.

[58] Nathaniel Trask, Pavel Bochev, and Mauro Perego. A conservative, consistent, and scalable meshfree mimetic method. *Journal of Computational Physics*, 409:109187, 2020.