Finite Volume Evolution Galerkin Methods for the Shallow Water Equations with Dry Beds

Andreas Bollermann[1,*], Sebastian Noelle[1], and Maria Lukáčová-Medvid'ová,[2]

[1] *IGPM, RWTH Aachen, Templergraben 55, 52062 Aachen, Germany.*
[2] *Department of Mathematics, University of Technology Hamburg, Schwarzenbergstraße 95, 21073 Hamburg, Germany*

Abstract. We present a new Finite Volume Evolution Galerkin (FVEG) scheme for the solution of the shallow water equations (SWE) with the bottom topography as a source term. Our new scheme will be based on the FVEG methods presented in (Lukáčová, Noelle and Kraft, *J. Comp. Phys.* 221, 2007), but adds the possibility to handle dry boundaries. The most important aspect is to preserve the positivity of the water height. We present a general approach to ensure this for arbitrary finite volume schemes. The scheme is also well-balanced and a new entropy fix improves the reproduction of sonic rarefaction waves.

AMS subject classifications: 65M08, 76B15, 76M12, 35L50

PACS: 02.60.Cb, 47.11.Df, 92.10.Sx
Key words: Well-balanced schemes, Dry boundaries, Shallow water equations, Evolution Galerkin schemes, Source terms

1 Introduction

The shallow water equations (SWE) are a model for the movement of water under the action of gravity. Mathematically spoken, they form a set of hyperbolic conservation laws, which can be extended by source terms like the influence of the bottom topography, friction or wind forces. In this case, we will speak of a balance law. This work will consider the variation of the bottom as the only source term.

Many important properties of the model rely on the fact that the water height is strictly positive. Despite this, typical relevant problems include the occurrence of dry areas, like dam break problems or the run-up of waves at a coast, with tsunamis as the most impressive example. So for simulations of these problems, we have to develop numerical schemes that can handle the (possibly moving) shoreline in a stable and efficient way.

*Corresponding author. *Email address:* bollermann@igpm.rwth-aachen.de (A. Bollermann)

Another crucial point in solving balance laws is the treatment of the source terms. For precise solutions, it is necessary to evaluate the source term in such a way that certain steady states are kept numerically. This problem has been solved for several schemes (see e.g. [2,8,12,17,18,20]), but not all of them take dry areas into account.

The finite volume evolution Galerkin (FVEG) methods developed by Lukáčová, Morton and Warnecke, cf. [11–13], have been successfully applied to the SWE in [12]. They are based on the evaluation of so called *evolution operators* which predict values for the finite volume update. Thanks to these operators, the schemes take into account all directions of wave propagation, enabling them to precisely catch multidimensional effects even on Cartesian grids. These schemes show a very good accuracy even on relatively coarse meshes compared to other state of the art schemes and they are also competitive in terms of efficiency.

However, the existing FVEG schemes are not able to deal with dry boundaries. Thus in this work we will present a new, positivity preserving FVEG scheme that can handle the interface between dry and wet regions. Our approach is very general and carries over to arbitrary finite volume scheme. Our FVEG method will preserve the *well-balancing* property, i.e. the numerical flux and the numerical source term cancel each other exactly for equilibrium solutions. In addition, we present a new entropy fix for the FVEG that improves the reproduction of sonic rarefaction waves.

We start our paper with a short presentation of the SWE in Section 2. Section 3 describes the FVEG method we will start from. The arising difficulties by introducing dry areas and means to overcome them are described in Section 4, which is the main part of the paper. Finally, in Section 5, we will show selected numerical test cases that demonstrate the performance of our schemes.

2 The Shallow Water Equations

2.1 Balance Law Form

We consider the shallow water system in balance form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}}(\mathbf{u}) = -\boldsymbol{\mathcal{S}}(\mathbf{u},\vec{x}). \tag{2.1}$$

The conserved variables and the flux are given by

$$\mathbf{u} = \begin{pmatrix} h \\ hv_1 \\ hv_2 \end{pmatrix}, \quad \boldsymbol{\mathcal{F}}(\mathbf{u}) = (\boldsymbol{\mathcal{F}}_1(\mathbf{u})\boldsymbol{\mathcal{F}}_2(\mathbf{u})) = \begin{pmatrix} hv_1 & hv_2 \\ hv_1^2 + g\frac{h^2}{2} & hv_1v_2 \\ hv_1v_2 & hv_2^2 + g\frac{h^2}{2} \end{pmatrix}, \tag{2.2}$$

where h denotes the relative water height, $\vec{v}=(v_1,v_2)^T$ the flow speed and g the (constant) gravity acceleration. The source term $\boldsymbol{\mathcal{S}}(\mathbf{u},\vec{x})$ is given by

$$\boldsymbol{\mathcal{S}}(\mathbf{u},\vec{x}) = gh \begin{pmatrix} 0 \\ \frac{\partial b(\vec{x})}{\partial x_1} \\ \frac{\partial b(\vec{x})}{\partial x_2} \end{pmatrix} \tag{2.3}$$

with $b(\vec{x})$ the local bottom height. We also introduce the *free surface level*, or total water height,

$$H(\vec{x}) = h(\vec{x}) + b(\vec{x}) \tag{2.4}$$

and the *speed of sound*

$$c = \sqrt{gh}. \tag{2.5}$$

2.2 Quasi-linear Form

For the derivation of the evolution operators in Section 3.2, it is helpful to rewrite (2.1) in primitive variables. The system then takes the form

$$\mathbf{w}_t + \mathbf{A}_1(\mathbf{w})\mathbf{w}_{x_1} + \mathbf{A}_2(\mathbf{w})\mathbf{w}_{x_2} = \mathbf{t} \tag{2.6}$$

with

$$\mathbf{w} = \begin{pmatrix} h \\ v_1 \\ v_2 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{pmatrix} v_1 & h & 0 \\ g & v_1 & 0 \\ 0 & 0 & v_1 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} v_2 & 0 & h \\ 0 & v_2 & 0 \\ g & 0 & v_2 \end{pmatrix} \tag{2.7}$$

and the source term

$$\mathbf{t} = \begin{pmatrix} 0 \\ -gb_{x_1} \\ -gb_{x_2} \end{pmatrix}. \tag{2.8}$$

For each angle $\theta \in [0,2\pi)$ we define the direction $\vec{\xi}(\theta) := (\cos\theta, \sin\theta)$. As system (2.1) is hyperbolic, for each of these directions and a fixed $\mathbf{w}$ the matrix

$$\mathbf{A}(\mathbf{w}) = \vec{\xi}_1 \mathbf{A}_1(\mathbf{w}) + \vec{\xi}_2 \mathbf{A}_2(\mathbf{w}) \tag{2.9}$$

has real eigenvalues

$$\lambda_1 = \vec{v}\cdot\vec{\xi} - c, \quad \lambda_2 = \vec{v}\cdot\vec{\xi}, \quad \lambda_3 = \vec{v}\cdot\vec{\xi} + c \tag{2.10}$$

and a full set of linearly independent eigenvectors

$$r_1 = \begin{pmatrix} -1 \\ g\frac{\cos\theta}{c} \\ g\frac{\sin\theta}{c} \end{pmatrix}, \quad r_2 = \begin{pmatrix} 0 \\ \sin\theta \\ -\cos\theta \end{pmatrix}, \quad r_3 = \begin{pmatrix} 1 \\ g\frac{\cos\theta}{c} \\ g\frac{\sin\theta}{c} \end{pmatrix}. \tag{2.11}$$

2.3 Lake at Rest

A trivial, but nevertheless important solution to (2.1) is the lake at rest situation, where the water is steady and the free surface level is constant, i.e. we have

$$\vec{v} = (0,0)^T \text{ and } H(\vec{x}) = H_0. \tag{2.12}$$

From (2.4) we immediately get

$$\nabla h = -\nabla b \tag{2.13}$$

and therefore (with (2.1) – (2.3) and $\vec{v} = (0,0)^T$)

$$\begin{pmatrix} 0 \\ g\frac{h^2}{2} \\ 0 \end{pmatrix}_{x_1} + \begin{pmatrix} 0 \\ 0 \\ g\frac{h^2}{2} \end{pmatrix}_{x_2} = -gh \begin{pmatrix} 0 \\ \frac{\partial b(\vec{x})}{\partial x_1} \\ \frac{\partial b(\vec{x})}{\partial x_2} \end{pmatrix}. \tag{2.14}$$

A scheme fulfilling a discrete analogon of (2.14) exactly is called *well-balanced*.

3 FVEG Schemes

Finite volume schemes are very popular for solving hyperbolic conservation laws for several reasons. They represent the underlying physics in a natural way and can be implemented very efficiently. Nevertheless, nearly all of them are based on the solution of one-dimensional Riemann problems and therewith a dimensional splitting. This introduces some sort of a bias: Wave propagation aligned with the grid is very well represented, whereas waves oblique to the grid cannot be caught as accurate.

In the last decade Lukáčová *et.al.* developed a class of finite volume evolution Galerkin schemes, see e.g. [10, 13, 15]. The FVEG scheme is a predictor-corrector method. In the predictor step a multidimensional evolution is done, the correction step is a finite volume update.

In this section we will recall the second order scheme presented in [12]. This method will be the starting point for our extensions for computations including dry beds in Section 4. Therefore we concentrate on the properties playing a role in this context and limit ourselves to the main ideas otherwise.

3.1 Finite Volume Update

For our computations, we use Cartesian grids, i.e. we divide our computational domain Ω in rectangular cells C_i, separated by edges E. On the edges, we have quadrature points $\vec{x}_k$. The subscript i will always refer to a cell, whereas k as a subscript is used as a global index for quadrature points. If we talk about the local quadrature points on a single edge, we use the index j instead.

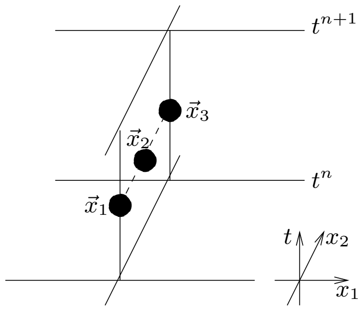

Figure 1: Quadrature points $\vec{x}_j$ for finite volume update

On each cell we define the initial value as

$$\mathbf{u}_i^0 := \mathbf{u}_i(0) \approx \frac{1}{|C_i|} \int_{C_i} \mathbf{u}(\vec{x}) d\vec{x} \tag{3.1}$$

where we use a Gaussian quadrature to approximate the integral. Integrating (2.1) on each cell, we can then define the update as

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{1}{|C_i|} \int_{t^n}^{t^{n+1}} \left(\int_{\partial C_i} \boldsymbol{\mathcal{F}}(\mathbf{u}(\vec{x},t)) \cdot \vec{n} d\vec{x} dt + \int_{C_i} \boldsymbol{\mathcal{S}}(\mathbf{u}(\vec{x}),\vec{x}) d\vec{x} \right) \tag{3.2}$$

using the Gauss theorem. Here $\mathbf{u}_i^n$ denotes cell average in C_i at time t^n and $\vec{n}$ is the outer normal.

For an approximation of (3.2), on each edge we define three quadrature points $\vec{x}_j, j = 1,2,3$, see Fig. 1. These quadrature points are located on the vertices ($j=1,3$) and the centre ($j=2$) of an edge. The flux over the edge is approximated by using midpoint rule in time and Simpson's rule in space, hence we will use the evolution operators from Section 3.2 to predict point values at the quadrature points at time $t^{n+1/2}$. The flux over an edge is then defined as

$$\boldsymbol{\mathcal{F}}_E := \sum_{j=1}^{3} \alpha_j \boldsymbol{\mathcal{F}}(\mathbf{u}_j^{n+1/2}) \cdot \vec{n} \approx \frac{1}{\Delta t |E|} \int_{t^n}^{t^{n+1}} \int_E \boldsymbol{\mathcal{F}}(\mathbf{u}(\vec{x},t)) \cdot \vec{n} d\vec{x} dt. \tag{3.3}$$

$\Delta t = t^{n+1} - t^n$ is the time step, the α_j represent the weights of Simpson's rule, i.e. we have $\alpha_{1,3} = \frac{1}{6}$ and $\alpha_2 = \frac{2}{3}$. Finally the source term is discretised as

$$\boldsymbol{\mathcal{S}}_i := g \sum_{j=1}^{3} \alpha_j \begin{pmatrix} 0 \\ \frac{1}{2}(\hat{h}_j^r + \hat{h}_j^l)(b_j^r - b_j^l) \\ \frac{1}{2}(\hat{h}_j^t + \hat{h}_j^b)(b_j^t - b_j^b) \end{pmatrix} \approx \frac{1}{\Delta x \Delta t} \int_{t^n}^{t^{n+1}} \int_{C_i} \boldsymbol{\mathcal{S}}(\mathbf{u}) d\vec{x}. \tag{3.4}$$

Here $\hat{h}_j$ represents the first component of $\mathbf{u}_j^{n+1/2}$ and $b_j = b(\vec{x}_j)$. The superscripts stand for the edges surrounding the cell, namely the **r**ight, **l**eft, **t**op and **b**ottom edge. Eqs. (3.2)–

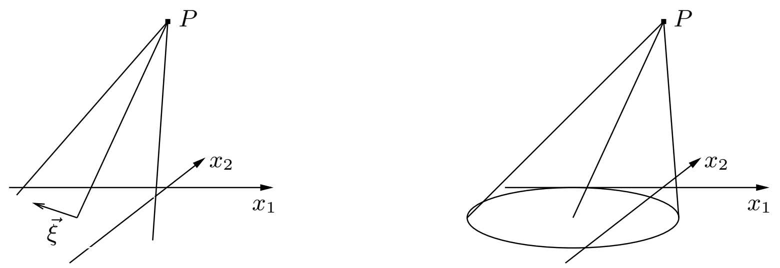

Figure 2: Bicharacteristic decomposition. Left: Bicharacteristic curves for a fixed direction $\vec{\xi}$. Right: Bicharacteristic cone.

(3.4) lead to the fully discrete scheme

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x}\left[\left(\sum_{E, E\subset\partial C_i} \boldsymbol{\mathcal{F}}_E\right) + \boldsymbol{\mathcal{S}}_i\right]. \tag{3.5}$$

The time step is chosen as

$$\Delta t = \mu \min_i \frac{\Delta x}{\max_k |\lambda_k|} \tag{3.6}$$

with λ_k the eigenvalues from (2.10) and $\mu < 1$ a CFL number.

3.2 Evolution Operators

As mentioned before, we use so-called evolution operators to predict point values of the solution for the quadrature points in (3.3). Indeed, a solution of (2.6) can be seen as a superposition of waves. So for a fixed point $P = (\vec{x}, t)$, we want to identify all the waves that contribute to the solution there. This section will describe shortly the evolution operator, present an exact formulation and give an example of a suitable approximation allowing an efficient implementation.

The derivation of evolution operators is based on the quasi-linear form of the system (2.6). For any given point P, we identify a suitable average value $\bar{\mathbf{w}}$ and linearise the system around P:

$$\mathbf{w}_t + \mathbf{A}_1(\bar{\mathbf{w}})\mathbf{w}_{x_1} + \mathbf{A}_2(\bar{\mathbf{w}})\mathbf{w}_{x_2} = \mathbf{t}. \tag{3.7}$$

The waves we are looking for propagate along the characteristics of this system. Thus for a fixed direction $\vec{\xi}(\theta)$, we apply an one-dimensional characteristic decomposition of the linearised system. This allows us to identify different wave propagations corresponding to the eigenvalues (2.10), the *bicharacteristics*. The left side of Fig. 2 shows an illustration. Integrating the decomposed system along the bicharacteristics, we get an integral representation of the solution at point P. At this point, the solution still depends on a particular direction $\vec{\xi}(\theta)$ and therefore does not respect waves coming from other directions. Thus we perform the decomposition for all angles $\theta \in [0, 2\pi)$ and average the solution at

P over θ. This yields the exact evolution operator of (3.7). The combination of all bicharacteristics yields the *bicharacteristic cone* shown in the right picture of Fig. 2. We introduce the following notation for the peak $P=(\vec{x},t^n+\tau)$ and points on the sonic cone:

$$Q_0 := (x_1+\tau\bar{v}_1, x_2+\tau\bar{v}_2, t^n) \tag{3.8}$$
$$\tilde{Q}_0 := (x_1+(t^n+\tau-\tilde{t})\bar{v}_1\cos\theta, x_2+(t^n+\tau-\tilde{t})\bar{v}_2\sin\theta, \tilde{t}) \tag{3.9}$$
$$Q := (x_1+\tau(\bar{c}+\bar{v}_1)\cos\theta, x_2+\tau(\bar{c}+\bar{v}_2)\sin\theta, t^n) \tag{3.10}$$
$$\tilde{Q} := (x_1+(t^n+\tau-\tilde{t})(\bar{c}+\bar{v}_1)\cos\theta, x_2+(t^n+\tau-\tilde{t})(\bar{c}+\bar{v}_2)\sin\theta, \tilde{t}) \tag{3.11}$$

Q_o is the centre of the sonic circle at time $t=t^n$, $\tilde{Q}_0$ denotes a point on the inner bicharacteristic connecting P and Q_0, Q is a point on the perimeter of the sonic circle at time $t=t^n$ and $\tilde{Q}$ denotes a point on the mantle of the sonic cone at an arbitrary time $\tilde{t}\in[t^n,t^n+\tau]$.

After some tedious calculations, see e.g. [12], we get the evolution operators for the SWE:

$$\begin{aligned}
h(P) = & \frac{1}{2\pi}\int_0^{2\pi} h(Q) - \frac{\bar{c}}{g}(v_1(Q)\cos\theta + v_2(Q)\sin\theta)\, d\theta \\
& + \frac{\bar{c}}{2\pi}\int_t^{t+\tau}\int_0^{2\pi}(b_{x_1}(\tilde{Q})\cos\theta + b_{x_2}(\tilde{Q})\sin\theta)\, d\theta d\tilde{t} \\
& - \frac{1}{2\pi}\int_t^{t+\tau}\frac{1}{t+\tau-\tilde{t}}\int_0^{2\pi}\frac{\bar{c}}{g}(v_1(\tilde{Q})\cos\theta + v_2(\tilde{Q})\sin\theta)\, d\theta d\tilde{t}
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
v_1(P) = & \frac{1}{2}v_1(Q_0) + \frac{1}{2\pi}\int_0^{2\pi} -\frac{g}{\bar{c}}h(Q)\cos\theta + v_1(Q)\cos^2\theta + v_2(Q)\sin\theta\cos\theta\, d\theta \\
& - \frac{g}{2}\int_t^{t+\tau} h_{x_1}(\tilde{Q}_0) + b_{x_1}(\tilde{Q}_0)\, d\tilde{t} \\
& - \frac{g}{2\pi}\int_t^{t+\tau}\int_0^{2\pi}(b_{x_1}(\tilde{Q})\cos^2\theta + b_{x_2}(\tilde{Q})\sin\theta\cos\theta)\, d\theta d\tilde{t} \\
& + \frac{1}{2\pi}\int_t^{t+\tau}\frac{1}{t+\tau-\tilde{t}}\int_0^{2\pi}(v_1(\tilde{Q})\cos 2\theta + v_2(\tilde{Q})\sin 2\theta)\, d\theta d\tilde{t}
\end{aligned} \tag{3.13}$$

$$\begin{aligned}
v_2(P) = & \frac{1}{2}v_2(Q_0) + \frac{1}{2\pi}\int_0^{2\pi} -\frac{g}{\bar{c}}h(Q)\sin\theta + v_1(Q)\sin\theta\cos\theta + v_2(Q)\sin^2\theta\, d\theta \\
& - \frac{g}{2}\int_t^{t+\tau} h_{x_2}(\tilde{Q}_0) + b_{x_2}(\tilde{Q}_0)\, d\tilde{t} \\
& - \frac{g}{2\pi}\int_t^{t+\tau}\int_0^{2\pi}(b_{x_1}(\tilde{Q})\cos\theta\sin\theta + b_{x_2}(\tilde{Q})\sin^2\theta)\, d\theta d\tilde{t} \\
& + \frac{1}{2\pi}\int_t^{t+\tau}\frac{1}{t+\tau-\tilde{t}}\int_0^{2\pi}(v_1(\tilde{Q})\sin 2\theta + v_2(\tilde{Q})\cos 2\theta)\, d\theta d\tilde{t}.
\end{aligned} \tag{3.14}$$

For efficient computations these operators have to be simplified. In [12], the authors follow [11] and present approximations of (3.12) – (3.14) that provide exact solutions of some one-dimensional Riemann problems. For piecewise constant data, these approximations read

$$\begin{aligned} h(P) = & \frac{1}{2\pi}\int_0^{2\pi} [H(Q) - \frac{\bar{c}}{g}(v_1(Q)\,\mathrm{sgn}(\cos\theta) + v_2(Q)\,\mathrm{sgn}(\sin\theta))]\,d\theta \\ & - b(P) + \frac{\tau}{2\pi}\int_0^{2\pi} (\bar{v}_1 b_{x_1}(Q) + \bar{v}_2 b_{x_2}(Q))\,d\theta, \end{aligned} \tag{3.15}$$

$$\begin{aligned} v_1(P) = & \frac{1}{2\pi}\int_0^{2\pi} \Big[-\frac{g}{\bar{c}} H(Q)\,\mathrm{sgn}(\cos\theta) \\ & + v_1(Q)\left(\cos^2\theta + \frac{1}{2}\right) + v_2(Q)\sin\theta\cos\theta \Big]\,d\theta, \end{aligned} \tag{3.16}$$

$$\begin{aligned} v_2(P) = & \frac{1}{2\pi}\int_0^{2\pi} \Big[-\frac{g}{\bar{c}} H(Q)\,\mathrm{sgn}(\sin\theta) \\ & + v_1(Q)\sin\theta\cos\theta + v_2(Q)\left(\sin^2\theta + \frac{1}{2}\right) \Big]\,d\theta. \end{aligned} \tag{3.17}$$

The corresponding operators for piecewise (bi-)linear data are given as

$$\begin{aligned} h(P) = & H(Q_0)(1 - \frac{\pi}{2}) - b(P) + \frac{1}{4}\int_0^{2\pi} H(Q)\,d\theta \\ & - \frac{\bar{c}}{g\pi}\int_0^{2\pi} (v_1(Q)\cos\theta + v_2(Q)\sin\theta)\,d\theta \\ & + \frac{\tau}{2\pi}\int_0^{2\pi} (\bar{v}_1 b_{x_1}(Q) + \bar{v}_2 b_{x_2}(Q))\,d\theta, \end{aligned} \tag{3.18}$$

$$\begin{aligned} v_1(P) = & v_1(Q_0)(1 - \frac{\pi}{4}) + \frac{g}{\bar{c}\pi}\int_0^{2\pi} H(Q)\cos\theta\,d\theta \\ & + \frac{1}{4}\int_0^{2\pi} \big[v_1(Q)(1 + 3\cos^2\theta) + 3v_2(Q)\sin\theta\cos\theta\big]\,d\theta, \end{aligned} \tag{3.19}$$

$$\begin{aligned} v_2(P) = & v_2(Q_0)(1 - \frac{\pi}{4}) + \frac{g}{\bar{c}\pi}\int_0^{2\pi} H(Q)\sin\theta\,d\theta \\ & + \frac{1}{4}\int_0^{2\pi} \big[3v_1(Q)\sin\theta\cos\theta + v_2(Q)(1 + 3\sin^2\theta)\big]\,d\theta. \end{aligned} \tag{3.20}$$

In our schemes, these operators are evaluated at the quadrature points $\vec{x}_k$ of the finite volume update defined in (3.3). Thus all data contributing to the evolved values is derived from the cell values next to the quadrature point. We therefore define the *stencil* S_k of a quadrature point $\vec{x}_k$ as

$$S_k := \{C_i | \vec{x}_k \in \partial C_i\}. \tag{3.21}$$

An example of the intersection of the cone with grid cells and the resulting stencil is shown in Fig. 3. The suitable average value $\bar{\mathbf{w}}_k$ used in (3.7) is chosen as

$$\bar{\mathbf{w}}_k = \frac{1}{|S_k|}\sum_{i:C_i \in S_k} \mathbf{w}_i. \tag{3.22}$$

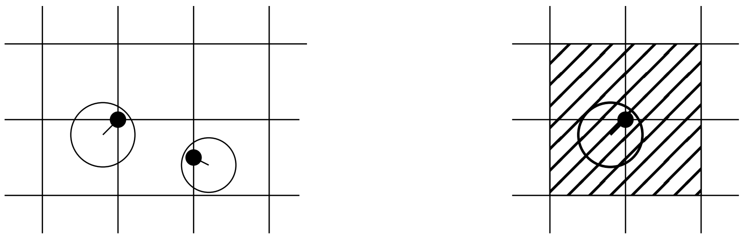

Figure 3: Left: Intersection of the sonic cone at quadrature points with grid cells. Right: Stencil of a quadrature point

We also tried a local Lax-Friedrichs update at the prediction points to get a better linearisation. The numerical results were almost exactly the same, so we chose the averaging procedure (3.22) for our computations.

3.3 Numerical Representation of the Bottom Topography

For finite volume schemes, the numerical representation of the bottom topography plays a crucial role in well-balancing as well as positivity of the scheme. In [2] Audusse *et.al.* use cell averages of the bottom for the computation of the free surface and reconstruct the free surface and the water height. The reconstruction of the bottom then results as the difference between slopes of H and h. In [8] Kurganov and Petrova propose to use a piecewise linear approximation of b instead of b itself by taking the values of b at cell corners. The cell average of b is then computed as the average of the corner values.

For the FVEG schemes it is necessary to define some value of b not only for cell averages and the reconstructed slopes, but also at the quadrature points where the evolution operators are evaluated. There is some freedom in doing this, as the source term discretisation (3.4) respects the well-balancing property independently of the reconstructed slopes of the bottom topography. As the evolution operators for the water height compute the free surface first and derive the actual water height via $h(P) = H(P) - b(P)$, the only necessary condition for $b(P)$ is $b(P) \leq H(P)$.

In this work, we will define the cell averages of b as in (3.1). For the quadrature points on cell corners, we set

$$b_k := b(\vec{x}_k) = \frac{1}{|S_k|} \sum_{i: C_i \in S_k} b_i. \tag{3.23}$$

The values of b_k at the centres of each edge are linearly interpolated from the neighbouring corners. While the latter condition has been derived in [3] to ensure well-balancing on adaptive grids, the formula for the corner points will turn out to be helpful for the dry bed case.

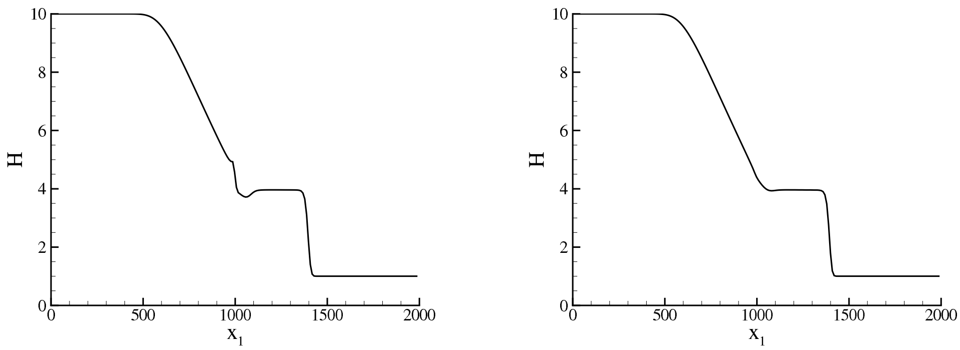

Figure 4: 1D dam break problem, solved with first order FVEG method. Left: solution without entropy fix. Right: solution with entropy fix from [14]

3.4 A Multidimensional Entropy Fix for the FVEG scheme

It is well known that the weak solution of a Riemann problem for conservation laws is not always unique, and an entropy condition is needed to single out the physically correct solution. This has its correspondence on the discrete level, where conservative numerical schemes may converge to entropy-violating solutions. This notorious difficulty seems to appear only near sonic rarefaction waves, where the flow changes from subcritical to supercritical velocity [21]. Various researchers have proposed so-called "entropy-fixes" for numerical schemes. In particular, we would like to mention Harten's and Hyman's entropy fix for the the Roe solver [6] (see also the discussion in [9]).

The FVEG schemes considered here make no exception, and may compute entropy violating solutions, see the left picture in Fig. 4. As is well known for classical finite volume methods, this effect is less visible (though still there) for second order schemes [21]. In order to make our point clear, we therefore focus on first order computations for the rest of this section.

In [14], Lukáčová and Tadmor proposed an entropy conservative variant for rarefaction waves computed by certain Riemann solvers, see also [24]. They applied this technique successfully to the finite volume corrector step of the FVEG scheme. They derived just the right amount of viscosity that one should add to the scheme to fulfil the entropy equality. Fig. 4 clearly shows the effectiveness of the scheme: While the standard FVEG scheme produces an entropy violating shock, the entropy conservative scheme clearly reproduces the correct rarefaction wave. Nevertheless, the scheme from [14] does not appear perfectly suitable for our needs. First, the proposed fix requires the characteristic decomposition of the jump of the conserved quantities across an edge. As the decomposition is not needed for the FVEG schemes, this is an undesired computational extra cost. In the context of dry boundaries, we should also mention that the decomposition matrix becomes very ill conditioned when $h \to 0$. The second point is that the scheme from [14] has been developed for the one-dimensional case. Although it can be applied dimension wise, this approach somewhat spoils the multidimensional spirit of the FVEG methods.

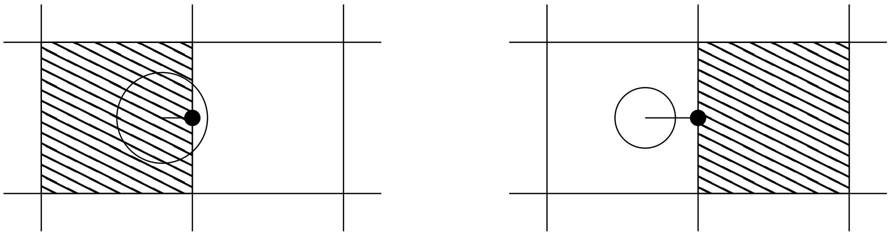

Figure 5: Position of sonic cones for the discrete sonic rarefaction with subsonic $\mathbf{u}_l$ and supersonic $\mathbf{u}_r$. Left: cone for $v_l < c_l$. Right: cone for $v_r > c_r$

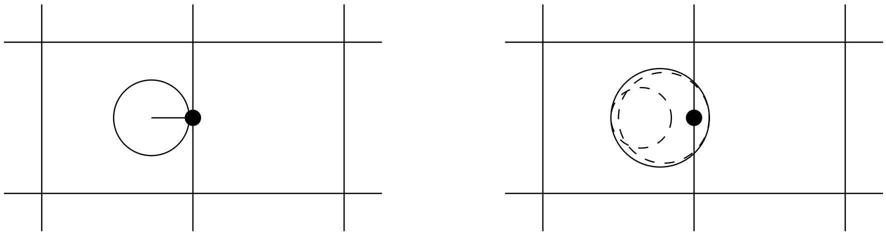

Figure 6: Position of sonic cones for the discrete sonic rarefaction with subsonic $\mathbf{u}_l$ and supersonic $\mathbf{u}_r$. Left: cone for averaged value $\bar{\mathbf{u}}$. Right: superscribed cone.

We therefore propose a new approach to solve the entropy problem. It is not based on a flux correction, but on the correct evaluation of the EG operators. To motivate our solution, we take a closer look on a discrete one-dimensional Riemann problem that should result in a transonic rarefaction, i.e. the flow is subsonic in upwind direction and supersonic in downwind direction. Thus let us assume we have two adjacent cells C_l and C_r with cell averaged data

$$\mathbf{u}_l = (h_l, v_l, 0), v_l < c_l \text{ and } \mathbf{u}_r = (h_r, v_r, 0), v_r > c_r. \tag{3.24}$$

Here c is the speed of sound defined in (2.5). To evaluate the evolution operators, we start with the sonic cones defined in (3.8) – (3.11). For simplicity, we limit ourselves to the quadrature point at the centre of the edge. The sonic cones resulting from $\mathbf{u}_l$ and $\mathbf{u}_r$ are sketched in Fig. 5. We can see that the cone resulting from $\mathbf{u}_r$ is located completely in C_l. Depending on the exact values of $\mathbf{u}_{l,r}$, this can also be the case for the sonic cone resulting from the averaging procedure (3.22). In other words: We use an evolution operator resulting from a supersonic linearisation in a regime that is subsonic. At least for the first order operators (3.15) – (3.17), this means that the predictor step exactly reproduces $\mathbf{u}_l$, which is then used for the flux evaluation. This corresponds to the generalised upwind method which is known to compute entropy violating solutions in some situations, cf. e.g. [9].

Thus the core of the problems seems to be the wrong domain of dependence for the predictor step. In case of a sonic rarefaction, the sonic cone should always include both regions, the subsonic as well as the supersonic one. As this is not guaranteed, we modify

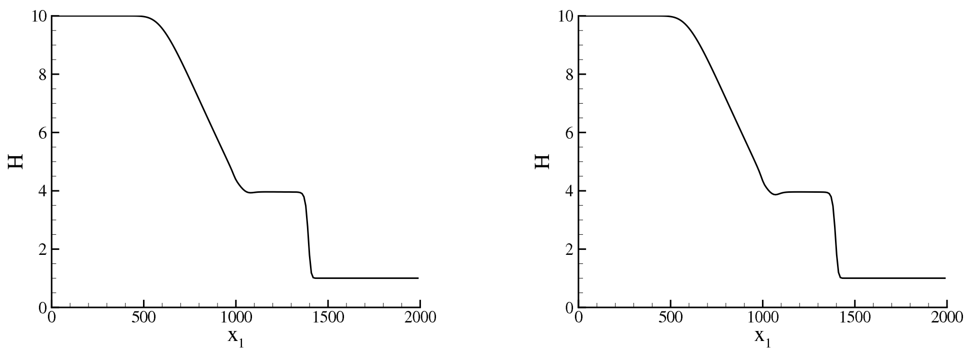

Figure 7: 1D dam break problem, solved with first order FVEG method. Left: solution with entropy fix from [14] Right: solution with new entropy fix

our method by extending the sonic cone if necessary. In a transonic situation we drop the sonic circle resulting from the averaging procedure (3.22). Instead, we use a circle which comprehends all the circles defined by the cell averaged values in the corresponding stencils, see Fig. 6 for an illustration. The exact formulation used for our schemes is as follows. Given two circles with midpoints $\vec{x}_i$ and radii r_i, $i=1{,}2$, we compute the new circle as

$$r = \frac{r_1 + r_2 + d}{2}$$
$$\vec{x} = \vec{x}_1 + (r - r_1)\frac{\vec{x}_2 - \vec{x}_1}{d}$$

with $d = \|\vec{x}_2 - \vec{x}_1\|_2$. However, if one circle comprehends the other one, we just choose the bigger circle as our new one. If the stencil of the evolution point consists of four cells, we apply the same formula for the neighbours on the two diagonals first and then again for the resulting circles.

The effectiveness of the new approach is shown in Fig. 7, where we compare the results of the entropy stable scheme from [14] and our new approach. They both clearly solve the entropy problem, with differences barely visible.

4 Dry Bed Modifications

To extend our schemes to computations including dry beds, we have to guarantee two properties: the positivity of the water height, and the well-balancing under the presence of dry areas. In literature, this is mainly achieved by two basic ingredients: a positivity preserving reconstruction and an additional time step constraint. Examples can be found in [2,8,20].

For the FVEG schemes, these measures fall short of the aims. The additional predictor step via the evolution operator prevents a direct proof of a positivity property. One reason for this is the extended stencil: the flux over an edge is computed using more cells than the direct neighbours (see Fig. 1 and 3). Another problem is the complex evaluation of the operators and with them the flux which makes an analysis of the positivity at least challenging if not impossible.

Regarding the well-balancing, a sophisticated reconstruction is not enough. From (3.4) it is obvious that the reconstruction does not directly affect the balancing of flux and source terms. The core of the problem is that the lake at rest described in (2.12) changes to

$$\vec{v} = (0,0)^T \text{ and } H(\vec{x}) = \begin{cases} H_0 & h(\vec{x}) > 0 \\ b(\vec{x}) & \text{else} \end{cases} \tag{4.1}$$

if dry areas are included. Thus for our schemes, we have to find evolution values for the water and bottom height that can handle properly the occurrence of this situation, i.e. which avoid the generation of spurious waves at the shoreline.

In this section, we will present an alternative approach to ensure the positivity of the water height as well as modifications of the finite volume update and the evolution operator to ensure the well-balancing property. We make sure that the changes do not affect the scheme away from dry regions.

4.1 A General Positivity Preserving FV Update

For the derivation of a positivity preserving scheme, we start with the first component of the finite volume update (3.5)

$$h_i^{n+1} = h_i^n - \frac{\Delta t}{\Delta x} \sum_{E, E \subset \partial C_i} \boldsymbol{\mathcal{H}}_E \tag{4.2}$$

with

$$\boldsymbol{\mathcal{H}}_E := \boldsymbol{\mathcal{F}}_E^h \tag{4.3}$$

the first component of the flux vector $\boldsymbol{\mathcal{F}}_E$ defined in (3.3). The condition for a positivity preserving scheme reads

$$h_i^n - \frac{\Delta t}{\Delta x} \sum_{E, E \subset \partial C_i} \boldsymbol{\mathcal{H}}_E \geq 0. \tag{4.4}$$

provided $h_i^n \geq 0 \forall i$.

As the last inequality does not necessarily hold, we will adjust the time step locally for those edges that are adjacent to cells violating (4.4). The basic idea is to reduce the time step only for the edges that contribute to the outflow of these cells. Therefore we start with separating the fluxes contributing to the inflow of a cell C from those contributing to the outflow by setting

$$\boldsymbol{\mathcal{H}}_E^+ := \max\{\boldsymbol{\mathcal{H}}_E, 0\} \tag{4.5}$$

$$\boldsymbol{\mathcal{H}}_E^- := \min\{\boldsymbol{\mathcal{H}}_E, 0\}. \tag{4.6}$$

This allows us to rewrite the update (4.2) as

$$h_i^{n+1} = h_i^n - \underbrace{\frac{1}{\Delta x} \sum_{E, E \subset \partial C_i} \Delta t_C^+ \boldsymbol{\mathcal{H}}_E^+}_{\text{outflow}} - \underbrace{\frac{1}{\Delta x} \sum_{E, E \subset \partial C_i} \Delta t \boldsymbol{\mathcal{H}}_E^-}_{\text{inflow}} \tag{4.7}$$

where initially Δt_C^+ is set to Δt. Whenever (4.4) is violated for $\Delta t_C^+ = \Delta t$, we replace it by the locally reduced time step

$$\Delta t_C^+ = \frac{\Delta x h_i^n}{\sum_{E, \boldsymbol{\mathcal{H}}_E^+ = \boldsymbol{\mathcal{H}}_E} \boldsymbol{\mathcal{H}}_E^+}. \tag{4.8}$$

Let us point out that both the nominator and the sum in the denominator are positive, so the resulting time step is positive.

Conservation requires that the outflow of a cell is the inflow of its neighbour. So for each edge, we define the local time step as

$$\Delta t_E := \min\left(\Delta t, \Delta t_{C^-(E)}^+\right) \tag{4.9}$$

where $C^-(E)$ determines the downwind cell of an edge. We can now rewrite our general update (3.5) as

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{1}{\Delta x} \sum_{E, E \subset \partial C_i} \Delta t_E \boldsymbol{\mathcal{F}}_E. \tag{4.10}$$

We dropped the source term temporarily for two reasons: First, it does not affect the water height and thus the positivity. The second reason is that the modification we present has impact on the well-balancing, cf. Section 4.2. We conclude the results of this section in the following theorem:

Theorem 1. *Assume we have a conservative finite volume scheme for the solution of the shallow water equations that can be written in the form (3.5). Then this scheme can be made positivity preserving via the steps described in algorithm 1 without any impact on the global time step.*

Proof. With definitions (4.9) and (4.8), the first component of the finite volume update (4.10) is non-negative by construction. Algorithm 1 ensures that all cells possibly affected by negative water heights are taken into account. Furthermore, the locally reduced $\Delta t_E < \Delta t$ only reflects the fact that the water height in the cell is 0 and therefore the flux out of the cell vanishes for times $t > t_n + \Delta t_E$. Hence, we still work with a global Δt for the problem. □

Algorithm 1.
- Identify all cells C_i violating (4.4) with the global Δt and store them in a list.
- Repeat the following steps until the list is empty:

- Take the first cell C_i in the list.
- Identify the outflow edges, i.e. all adjacent edges with $\boldsymbol{\mathcal{H}}_E = \boldsymbol{\mathcal{H}}_E^+$.
- Compute Δt_C^+ for cell C_i.
- Set $\Delta t_E = \Delta t_C^+$ for the outflow edges.
- Check if $h_k^{n+1} > 0$ according to (4.7) for all cells C_k adjacent to the outflow edges. If it is negative, add the cell C_k to the list.
- Remove the cell C_i from the list.

Remark 1. The choice of the local time step (4.8) is not unique. We tested a second possibility, namely

$$\Delta t_E^+ = \frac{\Delta x h_i^n - \sum_{E, \boldsymbol{\mathcal{H}}_E^- = \boldsymbol{\mathcal{H}}_E} \Delta t_E \boldsymbol{\mathcal{H}}_E^-}{\sum_{E, \boldsymbol{\mathcal{H}}_E^+ = \boldsymbol{\mathcal{H}}_E} \boldsymbol{\mathcal{H}}_E^+}. \tag{4.11}$$

In this case, we allow incoming fluxes to flow out in the same time step. While there are no visible differences in our numerical tests, this approach has some drawbacks. It violates the CFL condition as incoming mass should not cross the whole cell during a single time step. Furthermore, the dependency of Δt_E on the influx could lead to infinite loops in the algorithm above where the local time steps for neighbours are updated.

4.2 Well-balancing at the Shoreline: the Finite Volume Update

In the derivation of the positivity preserving finite volume update, we so far neglected the source term. Its introduction to the new scheme (4.10) rises the question which time step should be used for the source term. To maintain the well-balancing, the source term and the gravity driven parts of the flux must be multiplied with the same time step. This is in contradiction to the definition of Δt_E, which may change for different edges of the same cell. On the other hand, the reduced time step is not necessary for the momentum equations. We therefore shift the gravity driven components of $\boldsymbol{\mathcal{F}}$ into the source term, i.e. we define

$$\boldsymbol{\mathcal{F}}^*(\mathbf{u}) := \begin{pmatrix} hv_1 & hv_2 \\ hv_1^2 & hv_1 v_2 \\ hv_1 v_2 & hv_2^2 \end{pmatrix} \text{ and } \boldsymbol{\mathcal{S}}^*(\mathbf{u}, \vec{x}) := \frac{g}{2} h \begin{pmatrix} 0 \\ \frac{\partial H(\vec{x})}{\partial x_1} \\ \frac{\partial H(\vec{x})}{\partial x_2} \end{pmatrix}. \tag{4.12}$$

By replacing $\boldsymbol{\mathcal{F}}$ with $\boldsymbol{\mathcal{F}}^*$ in (3.3) and changing (3.4) to

$$\boldsymbol{\mathcal{S}}_i^* := g \sum_{j=1}^{3} \alpha_j \begin{pmatrix} 0 \\ \frac{1}{2}(\hat{h}_j^r + \hat{h}_j^l)(H_j^r - H_j^l) \\ \frac{1}{2}(\hat{h}_j^t + \hat{h}_j^b)(H_j^t - H_j^b) \end{pmatrix} \tag{4.13}$$

we can rewrite (3.5) as

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{1}{\Delta x} \left[\left(\sum_{E, E \subset \partial C_i} \Delta t_E \boldsymbol{\mathcal{F}}_E^* \right) + \Delta t \boldsymbol{\mathcal{S}}_i^* \right]. \tag{4.14}$$

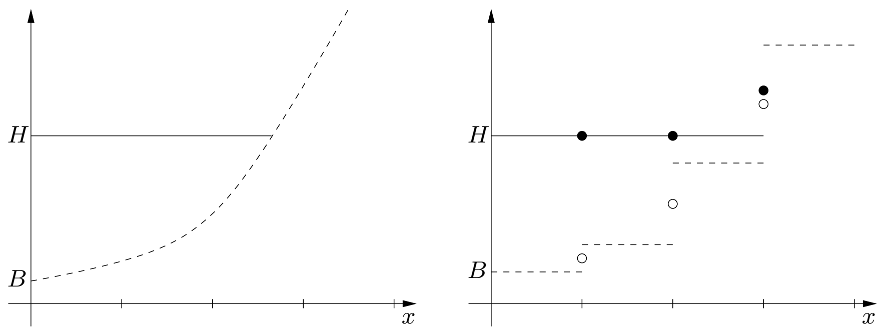

Figure 8: Lake at rest with dry boundaries. Solid line: free surface, dashed line: bottom topography, filled circles: H at evolution points, empty circles: B at evolution points. Left: Real situation. Right: Numerical Representation with evolution values.

This formulation ensures the well-balancing of the scheme even in the case of modified local time steps. Away from the shoreline we have $\Delta t_E = \Delta t \; \forall E$ and (4.14) equals the original update (3.5).

4.3 Well-balancing at the Shoreline: the Evolution Operator

The lake at rest situation with dry beds described in (4.1) is only preserved if the numerical flux and source terms in (4.13) are exactly balanced. This is not necessarily the case if the stencil of a quadrature point contains dry cells, as is demonstrated in Fig. 8. If $b_i > H_o$ for a cell in the stencil, the resulting bottom value from (3.23) can be higher than the free surface in the wet cells. Using the approximate evolution operators (3.15) and (3.18), it is easy to see that in the lake at rest case the evolved water height is also positive, leading to an even higher free surface at the quadrature point. In this case the combined flux and source term $\boldsymbol{\mathcal{S}}^*$ from (4.13) does not vanish anymore and introduces unphysical waves starting from the dry boundary.

To avoid the creation of these waves, we modify the data used for the predictor step at the interface. First, we replace the stencil S_k by S_k^* defined as

$$S_k^* := \{C_i | C_i \in S_k \wedge h_i > 0\} \tag{4.15}$$

which allows us to determine the maximal free surface level at $\vec{x}_k$ as

$$\bar{H}_k = \max_{S_k^*}(H_i). \tag{4.16}$$

Now in (3.22) and for the evaluation of the evolution operators we set

$$(H,b,\vec{v})_i = (\bar{H}_k, \bar{H}_k, 0) \quad \text{if } i \notin S_k^* \wedge b_i > \bar{H}_k. \tag{4.17}$$

In all other cases, we leave the values unchanged. The modification, which is illustrated in Fig. 9, ensures that the free surface is correctly represented in the source term computation (4.13). We also avoid an unphysical flooding of mounting slopes. In [5,20] a similar technique is used on triangles.

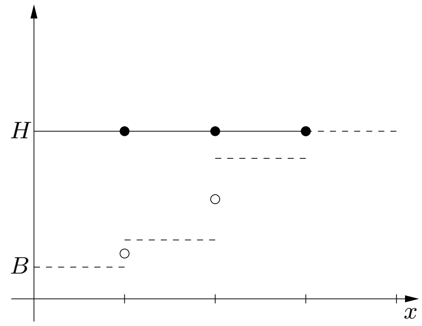

Figure 9: Lake at rest with dry boundaries, modification (4.17) for computation of evolution values. Symbols like in Fig. 8

Finally, even in the presence of wet, but nearly dry cells in the stencil, the expressions for $h(P)$ in (3.15) and (3.18) can become negative if h_i is small in the surrounding cells. This cannot necessarily be cured by a smaller time step, as substantial parts of the expressions do not depend on Δt. With this restriction in mind, we propose the simplest solution: Whenever we have $h(P) < 0$, we set $h(P) = v_1(P) = v_2(P) = 0$.

4.4 Treatment of Nearly Dry Cells

In our schemes, we consider a cell C_i to be dry when $h_i < \varepsilon_H$, where we have chosen $\varepsilon_h = 10^{-8}$. A well known problem occurs when h_i is close to that value: the velocity $v = hv/h$ can become singular due to small numerical errors in the conserved variables. This leads to very small time steps which in the end can basically stop the computation. This problem has been discussed e.g. in [8, 20], where different strategies have been proposed. In [8], Kurganov and Petrova propose to de-singularise v by multiplying it with a certain factor $f < 1$ whenever h falls below a certain threshold ε_v. In [20], the authors just set $v = 0$ whenever $h < \varepsilon_v$.

In this work, we use a different approach. As the solution at the dry boundary is always a rarefaction wave, the flow velocity will grow smoothly when water floods formerly dry areas. We will therefore limit the velocity in nearly dry regions depending on the velocity in flooded areas. We define the reference speed

$$v_{ref} := \max_{i:h_i > \varepsilon_v} \|\vec{v}\|_2 \tag{4.18}$$

where

$$\varepsilon_v = \frac{\Delta x}{L_{ref}}. \tag{4.19}$$

Whenever we have $h_i < \varepsilon_v$ and $\|\vec{v}_i\| > v_{ref}$, we set the new velocity to

$$v_i^* = v_{ref}\left(2 - \frac{v_{ref}}{\|\vec{v}_i\|}\right), \tag{4.20}$$

such that $\|\vec{v}\|$ is smoothly limited to a value between v_{ref} and $2v_{ref}$. The velocity components in C_i are then defined as

$$\vec{v}_i = v_i^* \vec{d} \tag{4.21}$$

with $\vec{d}$ the unit vector pointing in the same direction as the vector of discharge $(hv_1,hv_2)^T$. This approach appears us to be a better representation of the physics of the flow, as the velocity at the front is not necessarily vanishing.

4.5 The FVEG Algorithm

Before summarising the whole FVEG algorithm, we will spend a few words on the reconstruction needed to evaluate the evolution operators for piecewise linear data (3.18) – (3.20). As the operators are computed from the primitive variables $\mathbf{w}$, these are a natural choice for the reconstruction $R_{\Delta x}$. Thus in each cell we need the linear function

$$R_{\Delta x}(w_i)(\vec{x})_{|C_i} := \tilde{\mathbf{w}}_i(\vec{x}) = \mathbf{w}_i + \nabla \mathbf{w}_i \cdot (\vec{x} - \vec{x}_i) + (\mathbf{w}_i)_{x_1 x_2}(\vec{x} - \vec{x}_i)_1 (\vec{x} - \vec{x}_i)_2. \tag{4.22}$$

The derivatives $(\mathbf{w}_i)_{x_1}, (\mathbf{w}_i)_{x_2}$ and $(\mathbf{w}_i)_{x_1 x_2}$ are computed from the slopes between cell averages, cf. [13]. In this paper, we use the continuous, piecewise bilinear recovery described in [12] without any limiters. The piecewise bilinear functions are uniquely defined by the averages at the cell corners, which are already computed for the evaluation of the evolution operators, see (3.22). Although this reconstruction introduces some oscillations at steep fronts, it has a clear advantage in the vicinity of dry areas. As the averaged water height at the cell corners is non-negative via eqs. (4.15) – (4.17), the reconstruction is also non-negative by design. In dry cells, we set

$$\mathbf{w}_{x_1} = \mathbf{w}_{x_2} = \mathbf{w}_{x_1 x_2} = 0 \text{ if } h_i = 0. \tag{4.23}$$

We refer to [12] for all the details concerning the reconstruction strategy.

The complete algorithm then reads as follows:

Algorithm 2.

1. From given conservative data $\mathbf{u}_i^n$ and b_i at time t^n, compute the non-conservative variables $H_i^n, v_{1,i}^n, v_{2,i}^n$.
2. apply the reconstruction operator $R_{\Delta x}$ to $H_i^n, v_{1,i}^n, v_{2,i}^n$ and b_i.
3. compute the evolution operators
4. evaluate the advection fluxes $\boldsymbol{\mathcal{F}}_E^*$ from (4.12)
5. compute the gravity driven flux and source terms $\boldsymbol{\mathcal{S}}_i^*$ from (4.13)
6. perform the finite volume update (4.14)

We will finish the section by a proof of the well-balancing property of the scheme.

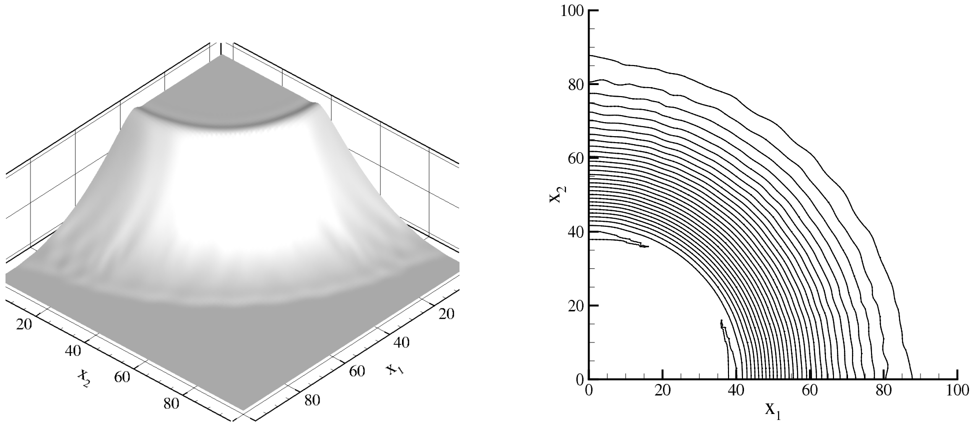

Figure 10: Circular dam break over dry bed, solution at t=1.75s. Left: 3D view. Right: 30 contour lines between $H=10.2$ and $H=0.2$

Theorem 2. *Suppose that we have a numerical solution respecting the lake at rest solution (4.1) with dry boundaries. Then the FVEG scheme (4.14) together with the modifications described in Section 4.3 preserves this state.*

Proof. For the lake at rest state with $\vec{v}=(0,0)^T$, the advective parts of the fluxes $\boldsymbol{\mathcal{F}}$ defined in (2.2) and $\boldsymbol{\mathcal{F}}^*$ defined in (4.12) are all zero. Thus for all edges we have $\Delta t_E=\Delta t$ and the original finite volume update (3.5) and the modified update (4.14) are the same. Then Theorem 2.1 from [12] states that the scheme is well-balanced provided that the predicted point values used for the flux evaluation also satisfy the lake at rest situation.

We will now show that all data used for the reconstruction and for the evaluation of the predictor step satisfies the requirements of Theorem 3.1 from [12]. From definitions (4.15) and (4.16) we see that for all evolution points the averaged free surface is computed as H_0, which is the free surface level in all flooded cells. As all velocities are zero by (4.1) respectively (4.17), the averaging also returns zero. Finally, the reconstruction procedure is based on the averaged point values and therefore in all flooded cells we have $\mathbf{w}_{x_1}=\mathbf{w}_{x_2}=\mathbf{w}_{x_1x_2}=0$. In dry cells we have the same result by definition (4.23). Thus we can apply Theorem 3.1 from [12] and this concludes our proof. □

5 Numerical Results

5.1 Dam Break over Dry Bed

This is a classical test case where we simulate the complete break of a circular dam separating a basin filled with water from a dry area. The computational domain is $[0,100]^2$, the water filled basin is located at $r=\|\vec{x}\|\leq 60$. In the basin we set $H_0=10$ and elsewhere $H_0=0$. Reference solutions can be found in [1,20,22].

In Fig. 10, we see a 3D view and contour lines of the water height, Fig. 11 shows the

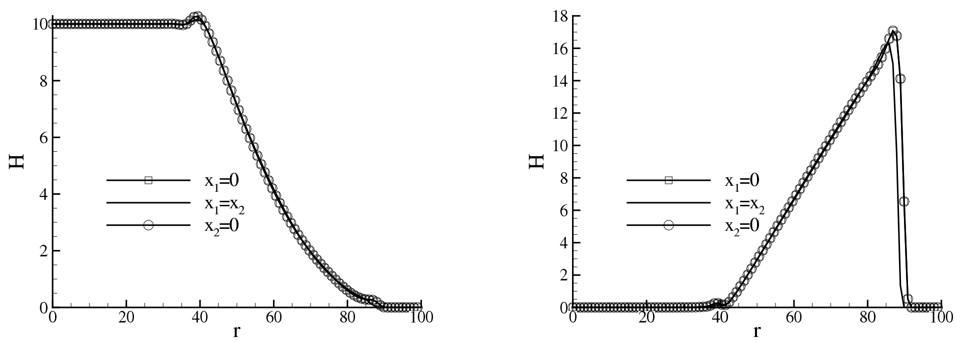

Figure 11: Circular dam break over dry bed, solution at t=1.75s with $r=\|\vec{x}\|_2$. Left: free surface. Right: velocity

water height and velocity at different lines through the domain. The solution is almost perfectly symmetric, the oscillation due to the non-limited reconstruction is restricted to three percent of the water height (we have $\max_i H_i = 10.269$). We see a small bump at the drying wetting front, but the front position and velocities are well represented. Thanks to the new entropy fix, there is no unphysical shock visible in the transcritical region.

5.2 Wetting/Drying on a Sloping Shore

This test case was proposed by Synolakis in [23] and computed in e.g. [16,20]. It describes the run-up and reflection of a wave on a mounting slope, with the initial solution given as

$$H_0(\vec{x}) = \max(f, b(\vec{x})), \quad \vec{v}_0(\vec{x}) = \left(\sqrt{\frac{g}{D}} H_0(\vec{x}), 0\right)^T \tag{5.1}$$

where

$$f(\vec{x}) = D + \delta \operatorname{sech}^2(\gamma(x_1 - x_a)). \tag{5.2}$$

As in [16,20], we set $D=1, \delta=0.019$ and

$$\gamma = \sqrt{\frac{3\delta}{4D}}, \quad x_a = \sqrt{\frac{4D}{3\delta}} \operatorname{arcosh}\left(\sqrt{\frac{1}{0.05}}\right). \tag{5.3}$$

For the bottom topography we have

$$b(\vec{x}) = b(x_1) = \begin{cases} 0 & x_1 < 2x_a \\ \dfrac{x_1 - 2x_a}{19.85} & \text{else.} \end{cases} \tag{5.4}$$

The computational domain is $\Omega = [0,80] \times [0,2]$ and the grid size $\Delta x = 0.04$.

In Fig. 12 we present the water height during the run-up and drying process. The run-up is excellently reproduced, with an accurate reflection of the wave. The scheme returns quickly to the lake at rest solution presented in the last picture. All in all, the results compare very well to the reference solutions presented in [16,20].

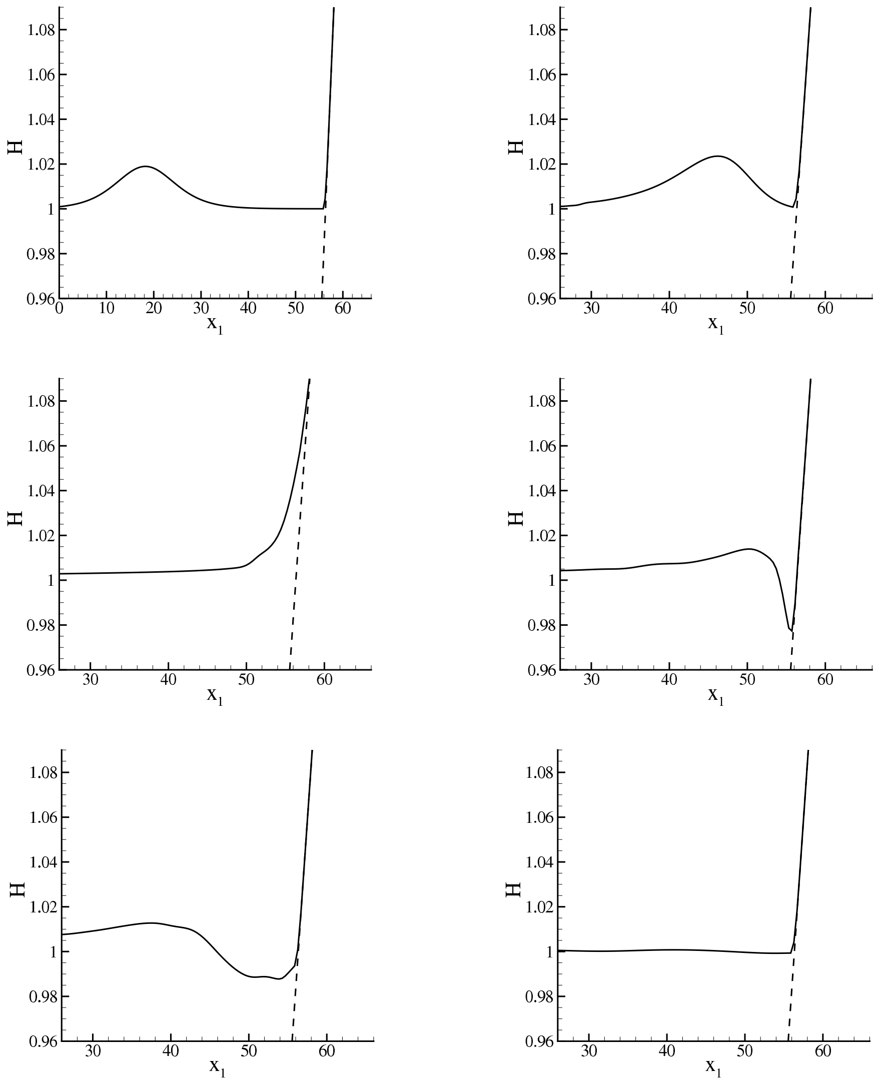

Figure 12: Drying/wetting on a sloping shore with free surface (solid line) and bottom height (dashed line) displayed. From top left to bottom right: Solutions at times $t=0$, $t=9$, $t=17$, $t=23$, $t=28$ and $t=80$.

# cells	L^∞	EOC	L^1	EOC	L^2	EOC
25 × 25	9.8038e-03		1.4526e-02		9.2884e-03	
50 ×50	3.6191e-03	1.44	3.9038e-03	1.90	2.7762e-03	1.74
100 ×100	1.5252e-03	1.25	1.3127e-03	1.57	9.5937e-04	1.53
200 ×200	1.1820e-03	0.37	4.6649e-04	1.49	3.8549e-04	1.32
400 ×400	5.3221e-04	1.15	1.7806e-04	1.39	1.4907e-04	1.37

Table 1: Experimental order of convergence (EOC) for Thackers curved solution. Error in water height in different norms.

5.3 Thacker's Periodic Solutions

We present two exact solutions of (2.1) proposed by Thacker in [25]. They both describe oscillations of a free surface in a parabolic basin with a free shoreline. The basin is defined as

$$b(\vec{x}) = b(r_c) = -H_0\left(1 - \frac{r_c^2}{a^2}\right). \tag{5.5}$$

r_c defines the distance from the basin's centre, H_0 the height of the centre and a is a parameter. We will define two functions $f(\vec{x},t)$ that describe solutions of (2.1) with $h(\vec{x},t)=\max(f(\vec{x},t)-b(\vec{x}),0)$. Both test cases will be computed on the domain $\Omega=[-2,2]^2$. The presented results have been performed with a grid size of $\Delta x = 0.4$. Reference solutions can be found in [16,19,20].

5.3.1 Thacker's Curved Solution

The first function results in a curved oscillation over b, it reads

$$f(r_c,t) = H_0\left(-1 + \frac{\sqrt{1-A^2}}{1-A\cos(\omega t)} - \frac{r_c^2}{a^2}\left(1 - \frac{1-A^2}{(1-A\cos(\omega t))^2}\right)\right). \tag{5.6}$$

Here, $\omega = \sqrt{8gH_0/a^2}$ is the frequency and for a given $r_0 > 0$, A is the shape parameter

$$A = \frac{a^2 - r_0^2}{a^2 + r_0^2}.$$

For the computation we set $a = 1$, $H_0 = 0.1$ and $r_0 = 0.8$, which results in an oscillating period of $T \approx 2.22$.

5.3.2 Thacker's Planar Solution

The second solution is a planar surface rotating around the basin. The corresponding function is

$$f(\vec{x},t) = \frac{\eta H_0}{a^2}\left(-\eta + 2(\vec{x} - \vec{x}_C)\cdot(\cos(\omega t), \sin(\omega t))^{tr}\right) \tag{5.7}$$

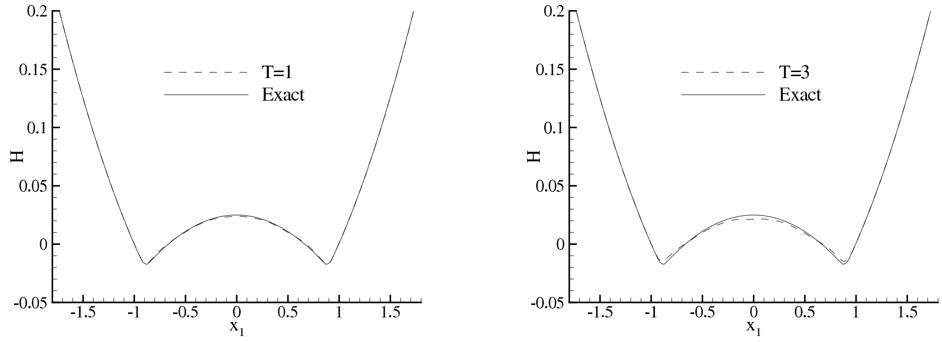

Figure 13: Thackers curved solutions. Left: $t=T$. Right: $t=3T$

# cells	L^∞	EOC	L^1	EOC	L^2	EOC
25 × 25	3.3855e-02		4.6507e-02		2.7148e-02	
50 ×50	1.7455e-02	0.96	1.8179e-02	1.36	1.1660e-02	1.22
100 ×100	1.0543e-02	0.73	1.0486e-02	0.79	6.6938e-03	0.80
200 ×200	8.2376e-03	0.36	8.3640e-03	0.33	5.4658e-03	0.29
400 ×400	7.1238e-03	0.21	7.8559e-03	0.09	5.1747e-03	0.08

Table 2: Experimental order of convergence (EOC) for Thackers planar solution. Error in water height in different norms.

with $\omega=\sqrt{2gH_0/a^2}$ the frequency and η another parameter. Here, we set $a=1$, $H_0=0.1$ and $\eta=0.5$. The resulting period is then $T\approx 4.44$.

We present the water height after one and three oscillations along the line $x_2=0$ in Fig. 13 for the curved solution and in Fig. 14 for the planar solution. The exact solution is very well reproduced, with only a slight smearing after three periods. There is no production of spurious waves at the dry boundary. In Tables 1 and 2 we present a convergence study for the two test cases. The experimental order of convergence for the curved solution is well better than one, which meets our expectations. The errors are slightly better than in [19].

For the planar solution, however, the order quickly drops to zero. The problem seems to raise from the boundary of the wetted domain, where we have supersonic velocities tangential to the bottom slope. So the problem might be related to the evolution operators, as they produce inexact solutions in other supersonic situations as well, see the discussion in Section 3.4 and 6.

5.4 Wave Run-up on a Conical Island

In this case we simulate the run-up of a solitary wave over a conical island. It has been performed experimentally at the U.S. Army Engineer Waterways Experiment Station, see [4]. The computational domain is $\Omega=[0,25]\times[0,30]$, the centre of the island is located

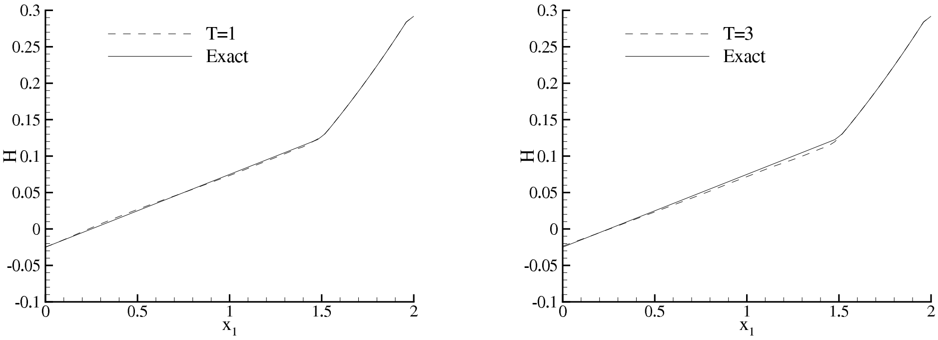

Figure 14: Thackers planar solutions. Left: $t = T$. Right: $t = 3T$

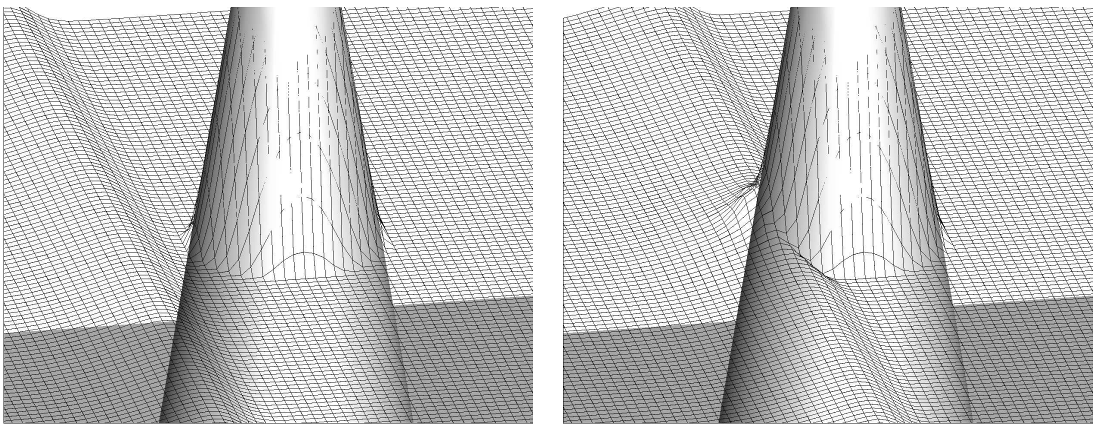

Figure 15: Run-up on a circular island. Left: Wave approaching island, $t = 7.9$. Right: run-up at front of the island, $t = 9.1$.

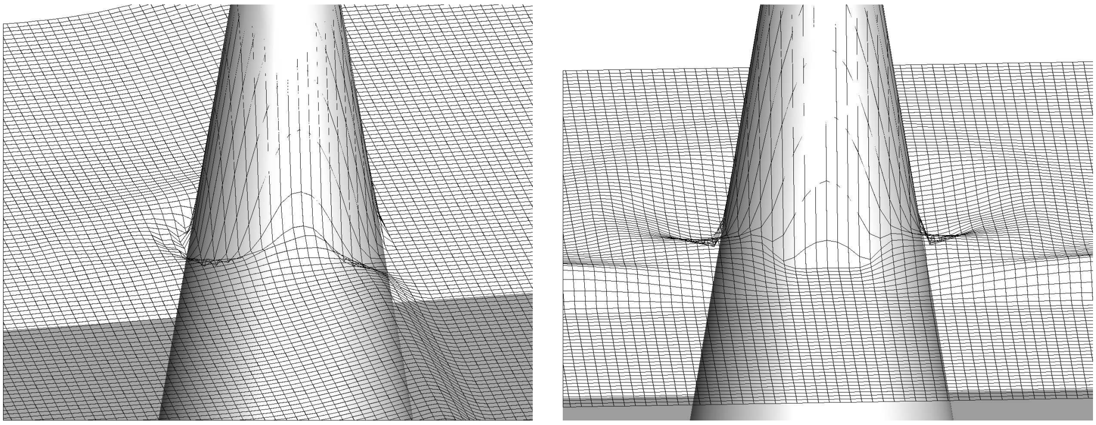

Figure 16: Run-up on a circular island. Left: lateral run-up, $t=10.7$. Right: symmetric waves around the island, $t=12.1$.

at $\vec{x}_C=(12.5,15)$ and with $r=\|\vec{x}-\vec{x}_C\|$ its shape is given by

$$b(r)=\begin{cases}0.625 & r\le 1.1\\ (3.6-r)/4 & r\le 3.6\\ 0 & \text{else.}\end{cases} \tag{5.8}$$

The initial free surface is given by $H_0=0.32$ and at time $t=0$ a wave enters the computational domain at $x_1=0$. The height of the wave is given by

$$H(0,y,t)=H_0+\alpha H_0\left(\frac{1}{\cosh(\xi\sqrt{gH_0/L}(t-3.5)}\right)^2$$

with $L=15$, $\alpha=0.1$ and $\xi=\sqrt{3\alpha(1+\alpha)L^2/(4H_0^2)}$, cf. [7,16].

We present 3D views of the solution in Fig. 15, 16 and 17. The vertical axis representing the free surface was scaled by a factor of 25 to emphasise the results. Moreover we show the evolution of the free surface at chosen points in Fig. 18. The position of the gages is given by $\vec{x}_3=(6.36,14.25)$, $\vec{x}_6=(8.9,15)$, $\vec{x}_9=(9.9,15)$, $\vec{x}_{16}=(12.5,12.42)$ and $\vec{x}_{22}=(15.1,15)$.

The features of the propagating wave are very well reproduced. We see the run-up when the wave hits the island, the formation of the symmetric waves surrounding it and finally the formation of the run-up behind the island. We also show that the scheme returns to the lake at rest. In Fig. 15 and 18, we can clearly see that the scheme is perfectly well-balanced in front of the wave. As in [20], the wave movement is shifted in time when compared to the data from [7], probably also due to the lack of friction terms. The run-down process is slightly underestimated, nevertheless we consider the results satisfying.

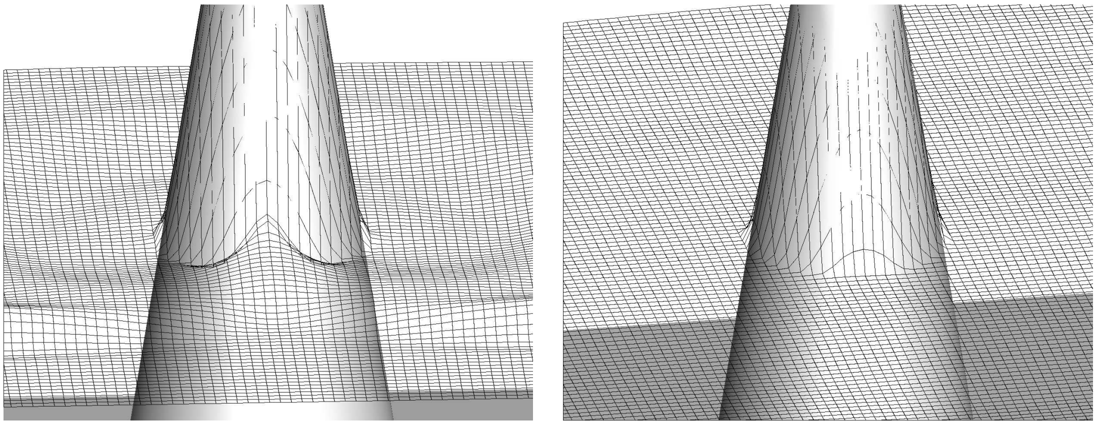

Figure 17: Run-up on a circular island. Left: run-up behind the island, $t = 13.7$. Right: reestablished lake at rest at $t = 40$.

6 Conclusions and Outlook

We presented an approach to ensure positivity of the water height for general finite volume schemes without affecting the global time step. This was achieved by limiting the outgoing fluxes of a cell whenever they would create negative water height. Physically, this corresponds to neglecting fluxes in the presence of vacuum. A splitting of advective and gravity driven parts of the flux preserved the well-balancing. In the context of FVEG schemes, we applied these techniques to develop a positivity preserving scheme which is well-balanced in the presence of dry areas. The scheme can also properly handle sonic rarefaction waves, thanks to a new entropy correction based on the evolution operators. We tested the scheme on a number of problems and in general obtained satisfying results.

However, the discussion of the entropy fix (see Section 3.4) revealed that in supersonic or transonic regimes the linearised wave cones used in the EG operator do not reflect the physical domain of dependence adequately. We conjecture that this is the origin of the loss of convergence for Thacker's planar solution (see Section 5.3.2), since here the velocities tangential to the boundary are larger than the (vanishing) gravitational speeds. Two issues should be analysed further. As mentioned above, the first is the linearisation strategy used in (3.7). With the entropy fix from 3.4 we made a first step towards a more sophisticated strategy adapted to the state of flow. The other issue is related to the approximation of the resulting linearized evolution operators. The approximations used here and in [12] are based on the approximations from [11], where they have been developed for the wave equations. Now for this system the second eigenvalue is always zero, so the sonic cone is never shifted in space with respect to the prediction point. An approximation taking this shift into account should give more accurate results in the critical regime.

Another possibility to improve the results is the introduction of friction terms. This could be helpful to control the velocities at the dry boundary by slowing down the waves

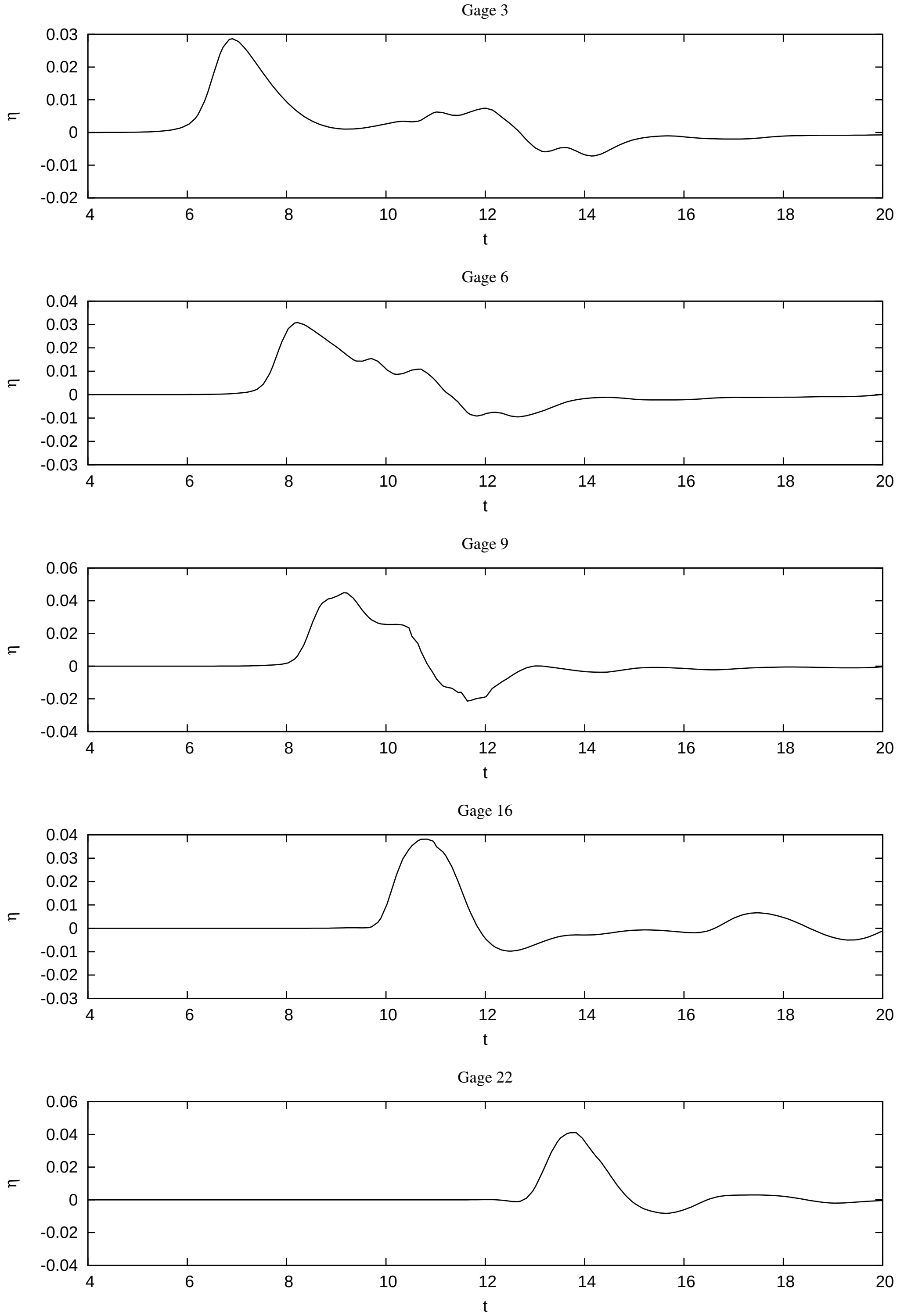

Figure 18: Run-up on a circular island. Time variation of the free surface $\eta = H - H_0$ at wave gages 6, 9, 16, and 22 of benchmark problem from [7].

near the shoreline. Finally we will combine the new scheme with the adaptation techniques presented in [3].

Acknowledgements

This work was supported by DFG-Grant "Adaptive semi-implicit FVEG methods for multidimensional systems of hyperbolic balance laws".

References

[1] Francisco Alcrudo and Pilar Garcia-Navarro, *A high-resolution godunov-type scheme in finite volumes for the 2d shallow-water equations*, Internat. J. Numer. Methods Fluids **16** (1993), no. 6, 489–505.

[2] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoît Perthame, *A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows*, SIAM J. Sci. Comput. **25** (2004), no. 6, 2050–2065.

[3] A. Bollermann, M. Lukáčová-Medvid'ová, and S. Noelle, *Well-balanced finite volume evolution galerkin methods for the 2d shallow water equations on adaptive grids*, ALGORITMY 2009, 18th Conference on Scientific Computing, 2009, pp. 81–90.

[4] Michael J. Briggs, Costas E. Synolakis, Gordon S. Harkins, and Debra R. Green, *Laboratory experiments of tsunami runup on a circular island*, Pure Appl. Geophys. **144** (1995), no. 3, 569–593.

[5] P. Brufau and P. García-Navarro, *Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique*, J. Comput. Phys. **186** (2003), no. 2, 503–526.

[6] Ami Harten and James M Hyman, *Self adjusting grid methods for one-dimensional hyperbolic conservation laws*, J. Comput. Phys. **50** (1983), no. 2, 235 – 269.

[7] Matthew E. Hubbard and Nick Dodd, *A 2d numerical model of wave run-up and overtopping*, Coastal Engineering **47** (2002), no. 1, 1–26.

[8] Alexander Kurganov and Guergana Petrova, *A second-order well-balanced positivity preserving central-upwind scheme for the saint-venant system*, Commun. Math. Sci. **5** (2007), no. 1, 133–160.

[9] Randall J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge University Press, 2002.

[10] M. Lukáčová-Medvid'ová, K. W. Morton, and G. Warnecke, *Evolution galerkin methods for hyperbolic systems in two space dimensions*, Math. Comput. **69** (2000), no. 232, 1355–1384.

[11] M. Lukáčová-Medvid'ová, K. W. Morton, and G. Warnecke, *Finite volume evolution galerkin methods for hyperbolic systems*, SIAM J. Sci. Comput. **26** (2004), no. 1, 1–30.

[12] M. Lukáčová-Medvid'ová, S. Noelle, and M. Kraft, *Well-balanced finite volume evolution galerkin methods for the shallow water equations*, J. Comput. Phys. **221** (2007), no. 1, 122–147.

[13] M. Lukáčová-Medvid'ová, J. Saibertova, and G. Warnecke, *Finite volume evolution galerkin methods for nonlinear hyperbolic systems*, J. Comput. Phys. **183** (2002), 533–562(30).

[14] M. Lukáčová-Medvid'ová and E. Tadmor, *On the entropy stability of the roe-type finite volume methods*, Hyperbolic Problems: Theory, Numerics and Applications (Eitan Tadmor, Jian-Guo Liu, and Athanasios E. Tzavaras, eds.), Proceedings of Symposia in Applied Mathematics, vol. 67, 2009, pp. 765–774.

[15] M. Lukáčová-Medvid'ová, G. Warnecke, and Y. Zahaykah, *On the boundary conditions for eg methods applied to the two-dimensional wave equation system*, ZAMM **84** (2004), no. 4, 237–251.

[16] F. Marche, P. Bonneton, P. Fabrie, and N. Seguin, *Evaluation of well-balanced bore-capturing schemes for 2d wetting and drying processes*, Internat. J. Numer. Methods Fluids **53** (2007), no. 5, 867–894.

[17] Sebastian Noelle, Normann Pankratz, Gabriella Puppo, and Jostein R. Natvig, *Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows*, J. Comput. Phys. **213** (2006), no. 2, 474–499.

[18] Sebastian Noelle, Yulong Xing, and Chi-Wang Shu, *High-order well-balanced finite volume weno schemes for shallow water equation with moving water*, J. Comput. Phys. **226** (2007), no. 1, 29–58.

[19] M. Ricchiuto and A. Bollermann, *Accuracy of stabilized residual distribution for shallow water flows including dry beds*, Hyperbolic Problems: Theory, Numerics and Applications (Eitan Tadmor, Jian-Guo Liu, and Athanasios E. Tzavaras, eds.), Proceedings of Symposia in Applied Mathematics, vol. 67, 2009, pp. 889–898.

[20] Mario Ricchiuto and Andreas Bollermann, *Stabilized residual distribution for shallow water simulations*, J. Comput. Phys. **228** (2009), no. 4, 1071–1115.

[21] P. L. Roe, *Sonic flux formulae*, SIAM J. Sci. Stat. Comput. **13** (1992), no. 2, 611–630.

[22] M. Seaïd, *Non-oscillatory relaxation methods for the shallow-water equations in one and two space dimensions*, Internat. J. Numer. Methods Fluids **46** (2004), no. 5, 457–484.

[23] Costas Emmanuel Synolakis, *The runup of solitary waves*, J. Fluid Mech. **185** (1987), 523–545.

[24] Eitan Tadmor, *Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems*, Acta Numerica **12** (2003), no. -1, 451–512.

[25] William Carlisle Thacker, *Some exact solutions to the nonlinear shallow-water wave equations*, J. Fluid Mech. **107** (1981), 499–508.