Well-Balanced Discontinuous Galerkin Method for Solving the Shallow Water Equations

by

Somayeh Fouladi

A research paper
presented to the University of Waterloo
in partial fulfillment of the
requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This report reviews positivity-preserving high-order well-balanced discontinuous Galerkin methods for the shallow water equations that are of interest in modeling, for example, flows in rivers and coastal areas. Well-balanced and positivity-preserving methods are constructed to exactly preserve the still water steady state, and non-negativity of the water height without loss of mass conservation.

In this report, the focus is on presenting both the derivation of well-balanced discontinuous Galerkin and positivity preserving schemes for the shallow water equations. We prove that, under a suitable Courant–Friedrichs–Lewy condition, the non-negativity of the water height at the next time step is preserved, provided that the current water height is non-negative. Finally, numerical experiments are performed in one dimension and then extended to two dimensions to demonstrate the theoretical properties of the scheme.

Acknowledgements

Love the loving and like to like, hate from abhorrence, be kind to affection, placate with peace and be apart from divorcement. –Zartosht

I express my deepest gratitude to my esteemed supervisors, Sander Rhebergen and David Del Rey Fernández, for their expert guidance, unwavering support, and significant assistance with editing during my academic research. Their invaluable insights and kindness enriched my journey profoundly.

Special thanks to Hans De Sterck, who previously served as our esteemed director and is currently the Chair of Applied Mathematics, for his consistent encouragement and support. Additionally, I extend my gratitude to George Labahn, our new director, for his invaluable assistance. I am also thankful to the administration team, Maureen, Alicia, and Tina, for their professionalism and support. To my dear friends, whose companionship and support made the challenges of distance from family more bearable, I express my sincere appreciation.

My heartfelt appreciation goes to my husband, my parents, my brother Behzad, for their unwavering love and support.

Dedication

I dedicate this achievement to the memory of my beloved brother, Behnam, whose spirit continues to inspire me.

Table of Contents

Chapter 1

Introduction

Flows in lakes, rivers, irrigation channels and near-shore oceanic flows are of great interest in hydrology, oceanography and climate modeling. Common to all of these flows is that vertical scales of motion are much smaller than horizontal scales. Using this observation, and the assumption of hydrostatic balance, the incompressible Navier–Stokes equations of fluid dynamics can be simplified to the well-known shallow water equations. In one dimension, these equations are given by the following hyperbolic system of partial differential equations (PDEs):

$$\partial_t \boldsymbol{u} + \partial_x \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{s}, \tag{1.1}$$

where $\boldsymbol{u} = [h, hu]^T$, $\boldsymbol{f}(\boldsymbol{u}) = [hu, hu^2 + \frac{1}{2}gh^2]^T$, and $\boldsymbol{s} = [0, -gh\partial_x b]^T$. Here h denotes the total water depth, u the depth-averaged flow velocity, g is the constant acceleration due to gravity, and b is the (fixed) elevation of the bottom topography. The free-surface of the flow resides at $z = h + b$. See Figure 1.1 for a depiction of the notation.

In this research paper we consider the numerical solution of the shallow water equations by the discontinuous Galerkin finite element method. In particular, we will study: (i) the numerical methods' abilities to preserve the steady state solution, also known as the *lake-at-rest* solution; and (ii) a positivity-preserving limiter for the numerical method. Below we expand on these two properties.

Lake-at-rest. As the name suggests, the lake-at-rest solution is a solution to the shallow water equations (1.1) in which there is no flow, i.e., $u = 0$. Setting $u = 0$ in (1.1), the shallow water equations reduce to

$$\partial_t h = 0 \qquad \text{and} \qquad gh\partial_x h = -gh\partial_x b. \tag{1.2}$$

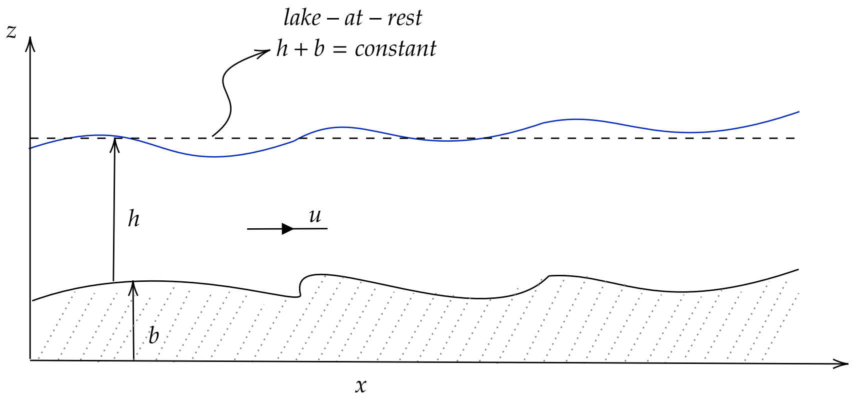


Figure 1.1: Shallow water flow profile.

The first of these equations implies that h is stationary, while the second equation implies that $h + b = c$ where c is some constant. In other words, the lake-at-rest solution is given by $u = 0$ and $h+b = c$ where c is some constant. Many numerical methods fail to preserve this steady state solution and introduce artificial flows. We will study an approach within the discontinuous Galerkin framework that is able to preserve the lake-at-rest solution.

Positivity-preservation. When solving the shallow water equations (1.1), the total water depth h may get arbitrarily close to zero, for example near the shore. Numerical methods have difficulty resolving such solutions, often resulting in negative approximations to h. A negative water depth is non-physical. Furthermore, negative h results in complex eigenvalues in the system of PDEs (1.1). To see this, we write the shallow water equations as

$$\partial_t \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \partial_x \begin{bmatrix} u_2 \\ \frac{u_2^2}{u_1} + \frac{1}{2} g u_1^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -g u_1 \partial_x b \end{bmatrix},$$

where $u_1 = h$ and $u_2 = hu$. The Jacobian of the flux function is given by

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \begin{bmatrix} 0 & 1 \\ -\frac{u_2^2}{u_1^2} + g u_1 & \frac{2u_2}{u_1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}.$$

Its eigenvalues are found by solving $\det(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} - \lambda I) = 0$. For the shallow water equations these are $\lambda_1 = u - \sqrt{gh}$ and $\lambda_2 = u + \sqrt{gh}$. Since $g > 0$ we observe that if h is negative then the eigenvalues are complex. This changes the shallow water equations from a hyperbolic system of PDEs to an elliptic problem. This will result in numerical instabilities. For this reason we will study how to preserve positivity of the total water depth h, again with the discontinuous Galerkin framework.

Chapter 2

The discontinuous Galerkin method for the shallow water equations

In this chapter we give a brief overview of the discontinuous Galerkin (DG) method for the shallow water equations (1.1) in one spatial dimension. In Section 2.1 we first consider the DG method for the shallow water equations over a flat bottom topography, i.e., $b = c$ where c is a constant. This implies that $\boldsymbol{s} = \mathbf{0}$ in (1.1). We proceed in Section 2.2 to extend the DG method to the shallow water equations with non-flat bottom topography. The main goal of this section will be to show how to maintain the lake-at-rest solution when using a DG method. Finally, in Section 2.3, we study an approach that is able to preserve the positivity of the total water depth h.

2.1 DG for the shallow water equations: flat bottom topography

2.1.1 Mesh and function spaces

For simplicity we will assume that we are solving the shallow water equations on a domain $\Omega = (0, 1)$ and time interval $I = [0, T]$ with T some final time. To discretize the shallow water equations we first partition the domain into N elements $K_j = [x_{j-1}, x_j]$, $j = 1, \ldots, N$, where $x_j = j\Delta x$ and $\Delta x = 1/N$. Furthermore, let $\Delta t = T/M$ be a time step and introduce the time levels $t_n = n\Delta t$.

For the discontinuous Galerkin method we approximate the solution to the shallow water equations in the following finite dimensional space:

$$\boldsymbol{V}_h := \{\boldsymbol{v}_h \in L^2(\Omega)^2 \,:\, \boldsymbol{v}_h \in P_k(K_j)^2 \; j = 1, \ldots, N\},$$

where P_k denotes the space of polynomials of degree k. For simplicity, in the section we derive the DG method only for $k = 1$. For $k = 1$, $\boldsymbol{V}_h$ denotes the space of piecewise linear polynomials that may be discontinuous across x_j, $j = 1, \ldots, N$.

2.1.2 Derivation of semi-discrete system

To derive the DG method for the shallow water equations, we first multiply (1.1) by a test function $\boldsymbol{v}_h \in \boldsymbol{V}_h$ in each element and integrate-by-parts to find:

$$\int_{K_j} \boldsymbol{v}_h^j \cdot \partial_t \boldsymbol{u}_h^j \, dx - \int_{K_j} \boldsymbol{f}(\boldsymbol{u}_h^j) \cdot \partial_x \boldsymbol{v}_h^j \, dx + \widehat{\boldsymbol{f}}_j \cdot \boldsymbol{v}_h^j(x_j) - \widehat{\boldsymbol{f}}_{j-1} \cdot \boldsymbol{v}_h^j(x_{j-1}) = 0, \tag{2.1}$$

where $\boldsymbol{u}_h^j$ is the restriction of $\boldsymbol{u}_h$ to element K_j and $\widehat{\boldsymbol{f}}_j := \boldsymbol{H}(\boldsymbol{u}_h^j(x_j, t), \boldsymbol{u}_h^{j+1}(x_j, t))$ is a numerical flux at node x_j that is introduced for stability reasons. The numerical flux is an approximation to the true flux $\boldsymbol{f}(\boldsymbol{u}(x_j))$.

There exist many numerical fluxes, see for example [13]. For the flat bottom topography case we consider the usual Lax–Friedrichs flux which, at node x, is given by:

$$\boldsymbol{H}(\boldsymbol{a}, \boldsymbol{b}) = \tfrac{1}{2}(\boldsymbol{f}(\boldsymbol{a}) + \boldsymbol{f}(\boldsymbol{b}) - \tfrac{\alpha}{2}(\boldsymbol{b} - \boldsymbol{a}), \tag{2.2}$$

where α is the maximum of the absolute eigenvalues computed over all elements in our mesh.

A function $\boldsymbol{u}_h \in \boldsymbol{V}_h$ can we expressed as

$$\boldsymbol{u}_h(x, t) = \boldsymbol{u}_j^{(0)}(t)\psi_0(x) + \boldsymbol{u}_j^{(1)}(t)\psi_1(x), \qquad \text{for } x \in K_j,$$

where $\psi_0(x)$ and $\psi_1(x)$ are basis functions. We choose Legendre basis functions. For this, let $\widehat{K} = (-1, 1)$ be a reference element and define the mapping $F_{K_j} : \widehat{K} \to K_j$ as $\xi \mapsto x = \frac{1}{2}(x_{j-1} + x_j) + \frac{1}{2}\Delta x \xi$. Define $\widehat{\psi}_0(\xi) = 1$ and $\widehat{\psi}_1(\xi) = \xi$. Then $\psi_m(x) = \hat{\psi}_m(F_{K_j}^{-1}(x))$ for $m = 0, 1$. With these basis functions $\boldsymbol{u}_j^{(0)}$ is a approximation to the mean of $\boldsymbol{u}(x, t)$ on element K_j and $\boldsymbol{u}_j^{(1)}$ is a slope. Since $\psi_0(x) = 1$ we will from now on write

$$\boldsymbol{u}_h(x, t) = \boldsymbol{u}_j^{(0)}(t) + \boldsymbol{u}_j^{(1)}(t)\psi(x), \qquad \text{for } x \in K_j. \tag{2.3}$$

Substituting this expression for the trial function $\boldsymbol{u}_h$, and similarly for the test function $\boldsymbol{v}_h$, in (2.1), we obtain the following semi-discrete system of ordinary differential equations (ODEs): for $j = 1, ..., N$ we have

$$\tfrac{d}{dt}\boldsymbol{u}_j^{(0)}(t) = \frac{1}{\Delta x}\left[\widehat{\boldsymbol{f}}_{j-1} - \widehat{\boldsymbol{f}}_j\right] \tag{2.4a}$$

$$\tfrac{d}{dt}\boldsymbol{u}_j^{(1)}(t) = \frac{3}{\Delta x}\left[\int_{-1}^{1} \boldsymbol{f}(\boldsymbol{u}_h(x(\xi), t))\,\mathrm{d}\xi - \left(\widehat{\boldsymbol{f}}_j + \widehat{\boldsymbol{f}}_{j-1}\right)\right]. \tag{2.4b}$$

2.1.3 Runge–Kutta time stepping

To solve the semi-discrete system of ODEs (2.4), we will use an explicit Runge–Kutta (RK) time stepping method. To introduce the RK method we write the DG method as:

$$\tfrac{d}{dt}\boldsymbol{u}_h(t) = \boldsymbol{L}(\boldsymbol{u}_h(t)). \tag{2.5}$$

We use the high-order total variation diminishing (TVD) Runge–Kutta method as presented in [12] (see also [7]). At time step n, this RK method is given as follows. Set $\boldsymbol{u}_h^{[0]} = \boldsymbol{u}_h^n$. For $i = 1, .., r$, where r is the order of the scheme, compute

$$\boldsymbol{u}_h^{[0]} = \sum_{m=0}^{i-1} \alpha_{im}\boldsymbol{u}_h^{[m]} + \beta_{im}\Delta t^n \boldsymbol{L}_h(\boldsymbol{u}_h^{[m]}), \tag{2.6}$$

then set $\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^{[r]}$. The coefficients α_{im} and β_{im} in (2.6) depend on the order r. These are given by:

If $r = 2$: $\alpha_{10} = \beta_{10} = 1, \quad \alpha_{20} = \alpha_{21} = \beta_{21} = \frac{1}{2}, \quad \beta_{20} = 0,$

If $r = 3$: $\alpha_{21} = \beta_{21} = \frac{1}{4} \quad \alpha_{30} = \frac{1}{3} \quad \beta_{30} = \alpha_{31} = \beta_{31} = 0 \quad \alpha_{32} = \beta_{32} = \frac{2}{3}.$

For the RK method given in (2.6) to be TVD, we will require that $\lambda := \Delta t/\Delta x$ satisfies the following CFL condition (see [7]):

$$\lambda \leq \lambda_r = \lambda_0 \min_{im} \frac{\alpha_{im}}{\beta_{im}}, \tag{2.7}$$

where, for $r = 2$ and $r = 3$, $\lambda_r = 1$.

2.1.4 Slope limiters

As is typical on non-linear hyperbolic conservation laws, a solution to the SWE may become discontinuous. Discontinuous solutions respresent hydraulic jumps and over-turning waves. However, approximating discontinuous solutions is difficult for numerical methods and typically result in large over- or under-shoots. These in turn affect stability of the discretization. To address these over- and under-shoots we use a slope-limiter. In particular, we use the corrected minmod limiter from [11].

To define the corrected minmod slope-limiter we first consider the standard minmod slope-limiter (see [7]). We introduce the following function:

$$m(a_1, a_2, a_3) = \begin{cases} s \cdot \min_{1 \leq i \leq 3} |a_i|, & \text{if } \operatorname{sign}(a_1) = \operatorname{sign}(a_2) = \operatorname{sign}(a_3) = s, \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

Consider now elements K_{j-1}, K_j, K_{j+1}. On these elements we have, see (2.3),

$$\begin{aligned} \boldsymbol{u}_h|_{K_{j-1}} &= \boldsymbol{u}_{j-1}^{(0)} + \boldsymbol{u}_{j-1}^{(1)}\psi(x), \\ \boldsymbol{u}_h|_{K_j} &= \boldsymbol{u}_j^{(0)} + \boldsymbol{u}_j^{(1)}\psi(x), \\ \boldsymbol{u}_h|_{K_{j+1}} &= \boldsymbol{u}_{j+1}^{(0)} + \boldsymbol{u}_{j+1}^{(1)}\psi(x). \end{aligned}$$

On element K_j we also define $\Delta_+ v_j = v_{j+1} - v_j$ and $\Delta_- v_j = v_j - v_{j-1}$. The minmod slope-limiting procedure for the DG discretization is given as follows: After each Runge–Kutta stage we replace $\boldsymbol{u}_h|_{K_j} = \boldsymbol{u}_j^{(0)} + \boldsymbol{u}_j^{(1)}\psi(x)$ on element K_j by

$$\boldsymbol{u}_h^{(mod)}|_{K_j} = \boldsymbol{u}_j^{(0)} + \boldsymbol{m}(\boldsymbol{u}_j^{(1)}, \Delta_+\boldsymbol{u}_j^{(0)}, \Delta_-\boldsymbol{u}_j^{(0)})\psi(x), \tag{2.9}$$

where

$$\boldsymbol{m}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3) = \begin{bmatrix} m((\boldsymbol{a}_1)_1, (\boldsymbol{a}_2)_1, (\boldsymbol{a}_3)_1) \\ m((\boldsymbol{a}_1)_2, (\boldsymbol{a}_2)_2, (\boldsymbol{a}_3)_2) \end{bmatrix}. \tag{2.10}$$

The minmod slope-limiter, however, is know to lose accuracy. To avoid a loss in accuracy, we will consider the corrected minmod slope limiter. This is defined by replacing $m(\cdot, \cdot, \cdot)$ by $\bar{m}(\cdot, \cdot, \cdot)$ which is defined as:

$$\bar{m}(a_1, a_2, a_3) = \begin{cases} a_1, & \text{if } |a_1| \leq M\Delta x^2, \\ m(a_1, a_2, a_3), & \text{otherwise.} \end{cases} \tag{2.11}$$

Here $M = \frac{2}{3}M_2$ where M_2 is an upper bound for $|u_{xx}|$ if $u \in C^2$.

An alternative to applying the minmod slope limiter component-wise, as described by (2.10), is to consider a characteristic approach [6]. We will apply the characteristic approach in our numerical examples in Chapter 3.

2.2 DG for the shallow water equations: non-flat bottom topography

In this section we consider the DG discretization of the SWE with non-flat bottom topography. A naive DG discretization of the SWE (1.1) will not satisfy the lake-at-rest conditions (1.2) numerically. Therefore, we will extend and modify the DG method presented in Section 2.1 in such a way that the lake-at-rest conditions (1.2) are satisfied numerically. Such a DG method is called well-balanced. This will, for example, require a modification of the numerical flux (2.2). We will follow the framework presented in [1, 10].

2.2.1 Naive DG discretization

We first show that a naive DG discretization does not satisfy the lake-at-rest conditions. For this, define $q = hu$. The momentum equation of the SWE can then be written as:

$$\partial_t q + \partial_x f(\boldsymbol{u}) = s(h, b), \tag{2.12}$$

where $f(\boldsymbol{u}) = hu^2 + \frac{1}{2}gh^2$ and $s(h, b) = -gh\partial_x b$.

First, we project b into the finite element space

$$V_h := \{v_h \in L^2(\Omega) \,:\, v_h \in P_k(K_j),\ j = 1, ..., N\}.$$

We denote this projection by b_h. Note that if $h + b =$ constant, then $h_h + b_h =$ constant. Augmenting the DG discretization of the momentum equation in (2.1) with a discretization of the source term, we obtain:

$$\int_{K_j} v_h \partial_t q_h \, dx - \int_{K_j} f(\boldsymbol{u}_h)\partial_x v_h \, dx + \widehat{f}_j v_h(x_j^-) - \widehat{f}_{j-1} v_h(x_{j-1}^+) = \int_{K_j} s(h_h, b_h) v_h \, dx. \tag{2.13}$$

(Note that, compared to (2.1), the superscripts j have been removed from the test and trial functions. This has been done for notations purposes. Note, furthermore, that $v_h(x_j^-) = v_h^j(x_j)$ and $v_h(x_{j-1}^+) = v_h^j(x_{j-1})$.) We will first show that this discretization does not satisfy the lake-at-rest conditions. Let us define the residual at time-level n as

$$\mathcal{R}^n = -\int_{K_j} f(\boldsymbol{u}_h)\partial_x v_h \, dx + \widehat{f}_j v_h(x_j^-) - \widehat{f}_{j-1} v_h(x_{j-1}^+) - \int_{K_j} s(h_h, b_h) v_h \, dx.$$

To preserve the lake-at-rest at time-level $n+1$, we require $\mathcal{R}^n = 0$. Assuming the discrete solution satisfies lake-at-rest at time-level n, we write out the definition of $f(\cdot)$ and $s(\cdot,\cdot)$, and find (using that $u_h^n = 0$):

$$\mathcal{R}^n = -\int_{K_j} \tfrac{1}{2} g h_h^2 \partial_x v_h \, dx + \widehat{f}_j v_h(x_j^-) - \widehat{f}_{j-1} v_h(x_{j-1}^+) + \int_{K_j} g h_h \partial_x b_h v_h \, dx.$$

Integration by parts gives us:

$$\mathcal{R}^n = \int_{K_j} \partial_x(\tfrac{1}{2} g h_h^2) v_h \, dx + (\widehat{f}_j - \tfrac{1}{2} g h_h^2) v_h(x_j^-) - (\widehat{f}_{j-1} - \tfrac{1}{2} g h_h^2) v_h(x_{j-1}^+) + \int_{K_j} g h_h \partial_x b_h v_h \, dx.$$

Since the solution at time level n satisfies the lake-at-rest condition, we have from (1.2) that $\partial_x(\frac{1}{2} g h_h^2) = -g h_h \partial_x b_h$ and so the residual simplifies to:

$$\mathcal{R}^n = (\widehat{f}_j - \tfrac{1}{2} g h_h^2) v_h(x_j^-) - (\widehat{f}_{j-1} - \tfrac{1}{2} g h_h^2) v_h(x_{j-1}^+).$$

Since $\mathcal{R}^n$ is not zero in general, the solutions to a "naive" DG discretization of the momentum equation will not satisfy the lake-at-rest conditions at time-level $n+1$.

2.2.2 Well-balanced DG discretization

Changing the numerical flux $\widehat{\boldsymbol{f}}$ in (2.13), we consider the following modified DG discretization of the SWE:

$$\int_{K_j} \boldsymbol{v}_h \cdot \partial_t \boldsymbol{u}_h \, dx - \int_{K_j} \boldsymbol{f}(\boldsymbol{u}_h) \cdot \partial_x \boldsymbol{v}_h \, dx + \widehat{\boldsymbol{f}}_j^l \cdot \boldsymbol{v}_h(x_j^-) - \widehat{\boldsymbol{f}}_{j-1}^r \cdot \boldsymbol{v}_h(x_{j-1}^+) = \int_{K_j} \boldsymbol{s}(h_h, b_h) \cdot \boldsymbol{v}_h \, dx, \tag{2.14}$$

where the "left" and "right" numerical fluxes, $\widehat{\boldsymbol{f}}_j^l$ and $\widehat{\boldsymbol{f}}_{j-1}^r$ respectively, are different from the Lax–Friedrichs flux introduced in (2.2). These left and right fluxes will be defined such that the DG method will satisfy the lake-at-rest conditions.

We write (2.14) as

$$\begin{aligned} &\int_{K_j} \boldsymbol{v}_h \cdot \partial_t \boldsymbol{u}_h \, dx - \int_{K_j} \boldsymbol{f}(\boldsymbol{u}_h) \cdot \partial_x \boldsymbol{v}_h \, dx + \widehat{\boldsymbol{f}}_j \cdot \boldsymbol{v}_h(x_j^-) - \widehat{\boldsymbol{f}}_{j-1} \cdot \boldsymbol{v}_h(x_{j-1}^+) \\ &\qquad = \int_{K_j} \boldsymbol{s}(h_h, b_h) \cdot \boldsymbol{v}_h \, dx + (\widehat{\boldsymbol{f}}_j - \widehat{\boldsymbol{f}}_j^l) \cdot \boldsymbol{v}_h(x_j^-) - (\widehat{\boldsymbol{f}}_{j-1} - \widehat{\boldsymbol{f}}_{j-1}^r) \cdot \boldsymbol{v}_h(x_{j-1}^+). \end{aligned} \tag{2.15}$$

The left hand side is now the same as the left hand side of (2.1) while the right hand side in (2.15) is our discretization for the source term.

Consider now the residual $\boldsymbol{\mathcal{R}}^n$ for the modified DG scheme (2.15) when at rest:

$$\boldsymbol{\mathcal{R}}^n = -\int_{K_j} \boldsymbol{f}(\boldsymbol{u}_h) \cdot \partial_x \boldsymbol{v}_h \, dx + \widehat{\boldsymbol{f}}^l_j \cdot \boldsymbol{v}_h(x_j^-) - \widehat{\boldsymbol{f}}^r_{j-1} \cdot \boldsymbol{v}_h(x_{j-1}^+) - \int_{K_j} \boldsymbol{s}(h_h, b_h) \cdot \boldsymbol{v}_h \, dx. \quad (2.16)$$

As before, to preserve the lake-at-rest properties, we require $\mathcal{R}^n = 0$. To achieve this, we follow the approach of Audusse et al [1].

At each node x_j we compute $\boldsymbol{u}^{\pm}_{h,j} := \boldsymbol{u}_h(x_j^{\pm})$ and define

$$h^{*,\pm}_{h,j} = \max\left(0, h^{\pm}_{h,j} + b^{\pm}_{h,j} - \max\left(b^+_{h,j}, b^-_{h,j}\right)\right), \quad (2.17)$$

and

$$\boldsymbol{u}^{*,\pm}_{h,j} = \begin{bmatrix} h^{*,\pm}_{h,j} \\ (h^* u)^{\pm}_{h,j} \end{bmatrix}. \quad (2.18)$$

The definition of $h^{*,\pm}_{h,j}$ ensures that at steady state $h^+_{h,j} + b^+_{h,j} = h^-_{h,j} + b^-_{h,j}$. With these definitions we now define the left and right fluxes as follows:

$$\widehat{\boldsymbol{f}}^l_j = \boldsymbol{H}\left(\boldsymbol{u}^{*,-}_{h,j}, \boldsymbol{u}^{*,+}_{h,j}\right) + \begin{bmatrix} 0 \\ \frac{g}{2}\left(h^-_{h,j}\right)^2 - \frac{g}{2}\left(h^{*,-}_{h,j}\right)^2 \end{bmatrix}, \quad (2.19a)$$

$$\widehat{\boldsymbol{f}}^r_{j-1} = \boldsymbol{H}\left(\boldsymbol{u}^{*,-}_{h,j-1}, \boldsymbol{u}^{*,+}_{h,j-1}\right) + \begin{bmatrix} 0 \\ \frac{g}{2}\left(h^+_{h,j-1}\right)^2 - \frac{g}{2}\left(h^{*,+}_{h,j-1}\right)^2 \end{bmatrix}. \quad (2.19b)$$

Here $\boldsymbol{H}$ is once again the Lax–Friedrichs flux (2.2).

Lemma 1. *The modified DG method (2.14) with fluxes (2.19), which we call the well-balanced DG method, is able to preserve the lake-at-rest solution.*

Proof. We will confirm that $\boldsymbol{\mathcal{R}}^n = 0$ provided the lake-at-rest conditions at time-level n are satisfied. (The lake-at-rest conditions at time-level n are $h_h + b_h =$ constant and $u_h = 0$.) We start with integration-by-parts of (2.16):

$$\boldsymbol{\mathcal{R}}^n = \int_{K_j} \partial_x \boldsymbol{f}(\boldsymbol{u}_h) \cdot \boldsymbol{v}_h \, dx + (\widehat{\boldsymbol{f}}^l_j - \boldsymbol{f}^-_j) \cdot \boldsymbol{v}_h(x_j^-) - (\widehat{\boldsymbol{f}}^r_{j-1} - \boldsymbol{f}^+_{j-1}) \cdot \boldsymbol{v}_h(x^+_{j-1}) - \int_{K_j} \boldsymbol{s}(h_h, b_h) \cdot \boldsymbol{v}_h \, dx. \quad (2.20)$$

But, for the lake-at-rest conditions,

$$\partial_x \boldsymbol{f}(\boldsymbol{u}_h) - \boldsymbol{s}(h_h, b_h) = \begin{bmatrix} 0 \\ g h_h \partial_x h_h \end{bmatrix} - \begin{bmatrix} 0 \\ -g h_h \partial_x b_h \end{bmatrix} = \begin{bmatrix} 0 \\ g h_h \partial_x (h_h + b_h) \end{bmatrix} = \boldsymbol{0}. \quad (2.21)$$

Therefore,

$$\boldsymbol{\mathcal{R}}^n = (\widehat{\boldsymbol{f}}^l_j - \boldsymbol{f}^-_j) \cdot \boldsymbol{v}_h(x^-_j) - (\widehat{\boldsymbol{f}}^r_{j-1} - \boldsymbol{f}^+_{j-1}) \cdot \boldsymbol{v}_h(x^+_{j-1}). \tag{2.22}$$

The lake-at-rest conditions imply $\boldsymbol{u}^{*,-}_{h,j-1} = \boldsymbol{u}^{*,+}_{h,j-1}$ and $\boldsymbol{u}^{*,-}_{h,j} = \boldsymbol{u}^{*,+}_{h,j}$ and therefore $\boldsymbol{H}\left(\boldsymbol{u}^{*,-}_{h,j-1}, \boldsymbol{u}^{*,+}_{h,j-1}\right) = \boldsymbol{f}(\boldsymbol{u}^{*,+}_{h,j-1})$ and $\boldsymbol{H}\left(\boldsymbol{u}^{*,-}_{h,j}, \boldsymbol{u}^{*,+}_{h,j}\right) = \boldsymbol{f}(\boldsymbol{u}^{*,+}_{h,j})$. Consider the first term on the right hand side of (2.22). Then

$$\begin{aligned} \widehat{\boldsymbol{f}}^l_j - \boldsymbol{f}^-_j &= \boldsymbol{f}(\boldsymbol{u}^{*,+}_{h,j}) - \boldsymbol{f}^-_j + \begin{bmatrix} 0 \\ \frac{g}{2}\left(h^-_{h,j}\right)^2 - \frac{g}{2}\left(h^{*,-}_{h,j}\right)^2 \end{bmatrix} \\ &= \begin{bmatrix} (h^*u)^+_{h,j} - (hu)_{h,j} \\ \frac{1}{2}g(h^{*,+}_{h,j})^2 - \frac{1}{2}g(h^-_{h,j})^2 + \frac{1}{2}g(h^-_{h,j})^2 - \frac{1}{2}g(h^{*,-}_{h,j})^2 \end{bmatrix} = \boldsymbol{0}. \end{aligned} \tag{2.23}$$

We have shown that $\boldsymbol{\mathcal{R}}^n = \boldsymbol{0}$ and so the DG method (2.14) with fluxes (2.19) will result in a discretization of the SWE that is able to preserve the lake-at-rest solution. □

2.2.3 Slope limiting

To preserve the lake-at-rest conditions also with slope-limiting, the authors of [1, 16] propose the following approach. The slope-limiter is first applied to $(h_h + b_h, q_h)^T$. This step is used to flag the elements that require slope-limiting. In the second step, the slope limiter is then applied to $(h_h, q_h)^T$ on the flagged elements.

2.3 Positivity preservation

To simplify notation in this section, instead of writing $h^{(0)}_j$ for the mean approximation to h in element K_j, as introduced in (2.3), we will write $\bar{h}_j$. Furthermore, we define $h^{n,-}_{j-1} = h_h(x^-_{j-1}, t^n)$ and $h^{n,+}_{j+1} = h_h(x^+_j, t^n)$. It was then shown in [15] that the well-balanced DG scheme of the previous section, but using forward Euler time-stepping, preserves the positivity of the mean of the water depth at time level $n+1$, i.e., that $\bar{h}^{n+1}_j \geq 0$, under the conditions that $\bar{h}^n_j \geq 0$, $h^{n,-}_{j-1} \geq 0$, $h^{n,+}_{j+1} \geq 0$ and $\lambda\alpha \leq 1/2$. However, instead of preserving only the positivity of the mean, we are also interested in preserving the DG approximation $h_h(x)$. Furthermore, we want to consider higher-order Runge–Kutta time-stepping. To achieve positivity for RK-DG for $h_h(x)$, Xing, Zhang, and Shu [15] introduced a positivity-preserving limiter, as we describe next.

Let $\boldsymbol{u}_j^n(x) := \boldsymbol{u}_h^n|_{K_j}$. Then $\boldsymbol{u}_j^n(x) = (h_j^n(x), q_j^n(x))$. Denote the means of $h_j^n(x)$ and $q_j^n(x)$ in element K_j by $\bar{h}_j^n$ and $\bar{q}_j^n$, respectively, and define $\bar{\boldsymbol{u}}_j^n = (\bar{h}_j^n, \bar{q}_j^n)$. The positivity-preserving limiter is then defined as:

$$\tilde{\boldsymbol{u}}_j^n(x) = \theta(\boldsymbol{u}_j^n(x) - \bar{\boldsymbol{u}}_j^n) + \bar{\boldsymbol{u}}_j^n, \tag{2.24}$$

where

$$\theta = \min\left(1, \frac{\bar{h}_j^n}{\bar{h}_j^n - m_j}\right), \qquad \text{where} \qquad m_j = \min_{x \in K_j} h_j^n(x).$$

To apply this limiter, we proceed as follows: After every Runge–Kutta stage, we first apply the modified slope-limiter as discussed in Section 2.2.3 and then the positivity-preserving limiter (2.24). If the mean of h_h is negative, we halve the time step.

Chapter 3

Numerical Examples

Prior to presenting the numerical results, we first provide an explanation of a few types of boundary conditions (BCs) that are typically considered when solving the SWEs. We then consider numerical examples demonstrating the well-balancedness and positivity preservation of the DG method. Although we only presented the DG method for $k = 1$ and for one spatial dimension, we will present numerical results for $k = 1, 2, 3$ and for both one and two spatial dimensions. For the numerical results the constant acceleration due to gravity is set to $g = 9.812\ m/s^2$. We present the results under the CFL condition $\frac{\lambda}{\hat{w}_1}\alpha \leq \frac{1}{k+1}$, $\alpha = \max\left(|u| + \sqrt{gh}\right)$ where the maximum is taken over the whole domain.

3.1 Boundary conditions

In this section we discuss the various boundary conditions (BCs) considered in this section and how to impose them numerically. Specifically, we use three types of BCs, transmissive, reflective and periodic [2] which are described in the following sections.

Transmissive Boundary Conditions

Transmissive BCs at the left end of the domain are given as

$$h(x_0^-, t) := h(x_0^+, t), \qquad hu(x_0^-, t) := hu(x_0^+, t),$$

i.e., the arguments to the numerical fluxes are

$$H(h(x_0^+, t), h(x_0^+, t)), \qquad H(hu(x_0^+, t), hu(x_0^+, t)),$$

while on the right end they are

$$h(x_{N_e}^{+}, t) := h(x_{N_e}^{-}, t), \qquad hu(x_{N_e}^{+}, t) := hu(x_{N_e}^{-}, t),$$

i.e., the numerical fluxes should be

$$H(h(x_{N_e}^{-}, t), h(x_{N_e}^{-}, t)), \qquad H(hu(x_{N_e}^{-}, t), hu(x_{N_e}^{-}, t)).$$

The above simulate the condition when the water freely moves from left and right.

Reflective Boundary Conditions

Now we consider the case in which there exist a wall at the boundary. In this scenario, we expect that when water wave reaches the wall it returns into the domain with the same speed, but with the opposite direction. This is imposed as follows. At the left end of the domain we have

$$h(x_0^{-}, t) := h(x_0^{+}, t), \qquad hu(x_0^{-}, t) := -hu(x_0^{+}, t),$$

i.e., the arguments to the numerical fluxes are

$$H(h(x_0^{+}, t), h(x_0^{+}, t)), \qquad H(-hu(x_0^{+}, t), hu(x_0^{+}, t)).$$

At the right end of the domain they are

$$h(x_{N_e}^{+}, t) := h(x_{N_e}^{-}, t), \qquad hu(x_{N_e}^{+}, t) := -hu(x_{N_e}^{-}, t),$$

i.e., the numerical fluxes should be

$$H(h(x_{N_e}^{-}, t), h(x_{N_e}^{-}, t)), \qquad H(hu(x_{N_e}^{-}, t), -hu(x_{N_e}^{-}, t)).$$

Note that all the values are the same as the transmissive case but the momentum has an opposite sign.

Periodic Boundary Conditions

The last case is periodic BCs. For this condition, at the left end of the domain, we set

$$h(x_0^{-}, t) := h(x_{N_e}^{-}, t), \qquad hu(x_0^{-}, t) := hu(x_{N_e}^{-}, t),$$

i.e., the arguments to the numerical fluxes are

$$H(h(x_{N_e}^-, t), h(x_0^+, t)), \qquad H(hu(x_{N_e}^-, t), hu(x_0^+, t)).$$

At the right end the boundary conditions are:

$$h(x_{N_e}^+, t) := h(x_0^+, t), \qquad hu(x_{N_e}^+, t) := hu(x_0^+, t),$$

i.e., the arguments to the numerical fluxes are

$$H(h(x_{N_e}^-, t), h(x_0^+, t)), \qquad H(hu(x_{N_e}^-, t), hu(x_0^+, t)).$$

3.2 One-dimensional examples

3.2.1 Accuracy test: Over a flat bottom

We first consider the accuracy and order of convergence of the well-balanced RKDG method described in Section 2.2. Despite considering a flat bottom, this tests the preservation of accuracy when using the modified fluxes presented in Section 2.2.2. We solve (1.1) with $\boldsymbol{s} = (f_1(x,t), f_2(x,t)$ and subject to the following initial conditions

$$h(x,0) = 2 + e^{\cos(2\pi x)}, \qquad u(x,0) = \sin^2(\cos(2\pi x)). \tag{3.1}$$

The functions $f_1(x,t)$ and $f_2(x,t)$ are given functions such that

$$h(x,t) = 2 + e^{\cos(2\pi(x-t))}, \qquad u(x,t) = \sin^2(\cos(2\pi(x-t))), \tag{3.2}$$

are the exact solutions of (1.1).

We present L_2 errors and rates of convergence for h and hu in Table 3.1 for $k = 1, 2, 3$. We observe that the rates of convergence are $k+1$.

3.2.2 Accuracy test: Over a non-flat bottom

We now consider the accuracy and order of convergence of the well-balanced RKDG method described in Section 2.2 for the non-flat bottom topography case. We solve (1.1) with $\boldsymbol{s} = (Z_1(x,t), -gh\partial_x b + Z_2(x,t)$ and subject to the following initial conditions (3.1) and periodic boundary conditions on domain $[0,1]$. The bottom topography is defined as $b(x) = \sin^2(2\pi x)$. The functions $Z_1(x,t)$ and $Z_2(x,t)$ are given such that the exact solution of the problem is given by (3.2). The errors and rates of convergence are reported in Table 3.2. We again observe that the rates of convergence are $k+1$.

Table 3.1: Errors and spatial computational order of the well-balanced positivity-preserving RKDG method for the following shallow water equation over a flat bottom, see Section 3.2.1.

		h		hu	
	Δx	L_2 error	rate	L_2 error	rate
$k = 1$	$\frac{1}{10}$	2.8312×10^{-2}	–	1.5251×10^{-1}	–
	$\frac{1}{20}$	7.0927×10^{-3}	1.9970	3.8672×10^{-2}	1.9795
	$\frac{1}{40}$	1.7743×10^{-3}	1.9991	9.6933×10^{-3}	1.9962
	$\frac{1}{80}$	4.4343×10^{-4}	2.0005	2.4208×10^{-3}	2.0015
	$\frac{1}{160}$	1.1084×10^{-4}	2.0002	6.0492×10^{-4}	2.0006
$k = 2$	$\frac{1}{10}$	2.5735×10^{-3}	–	1.5315×10^{-2}	–
	$\frac{1}{20}$	3.4088×10^{-4}	2.9164	2.0569×10^{-3}	2.8963
	$\frac{1}{40}$	4.3876×10^{-5}	2.9578	2.6267×10^{-4}	2.9692
	$\frac{1}{80}$	5.5412×10^{-6}	2.9852	3.3002×10^{-5}	2.9926
	$\frac{1}{160}$	6.9464×10^{-7}	2.9959	4.1303×10^{-6}	2.9982
$k = 3$	$\frac{1}{10}$	2.3345×10^{-4}	–	1.3247×10^{-3}	–
	$\frac{1}{20}$	1.3910×10^{-5}	4.0691	8.3361×10^{-5}	3.9902
	$\frac{1}{40}$	8.5951×10^{-7}	4.0165	5.1789×10^{-6}	4.0087
	$\frac{1}{80}$	5.3579×10^{-8}	4.0038	3.2309×10^{-7}	4.0026
	$\frac{1}{160}$	3.3466×10^{-9}	4.0009	2.0183×10^{-8}	4.0007

Table 3.2: Errors and spatial computational order of the well-balanced positivity-preserving RKDG method for the following shallow water equation over a non-flat bottom, see Section 3.2.2.

		h		hu	
	Δx	L_2 error	rates	L_2 error	rates
$k = 1$	$\frac{1}{10}$	2.9100×10^{-2}	–	1.2031×10^{-1}	–
	$\frac{1}{20}$	7.2507×10^{-3}	2.0048	2.9853×10^{-2}	2.0108
	$\frac{1}{40}$	1.8431×10^{-3}	1.9760	7.4829×10^{-3}	1.9962
	$\frac{1}{80}$	4.6519×10^{-4}	1.9862	1.8770×10^{-3}	1.9952
	$\frac{1}{160}$	1.1652×10^{-4}	1.9973	4.6932×10^{-4}	1.9998
$k = 2$	$\frac{1}{10}$	2.7483×10^{-3}	–	1.3174×10^{-2}	–
	$\frac{1}{20}$	5.4298×10^{-4}	2.3396	2.0200×10^{-3}	2.7053
	$\frac{1}{40}$	1.3277×10^{-4}	2.0320	3.2952×10^{-4}	2.6159
	$\frac{1}{80}$	3.3006×10^{-5}	2.0081	6.4236×10^{-5}	2.3589
	$\frac{1}{160}$	8.2394×10^{-6}	2.0021	1.4641×10^{-5}	2.1334
$k = 3$	$\frac{1}{10}$	4.7474×10^{-4}	–	1.5295×10^{-3}	–
	$\frac{1}{20}$	3.5716×10^{-5}	3.7325	9.3668×10^{-5}	4.0293
	$\frac{1}{40}$	2.3223×10^{-6}	3.9430	5.9328×10^{-6}	3.9808
	$\frac{1}{80}$	1.4657×10^{-7}	3.9858	3.7225×10^{-7}	3.9944
	$\frac{1}{160}$	9.1830×10^{-9}	3.9965	2.3289×10^{-8}	3.9985

3.2.3 Riemann problem on a flat bottom

Here, we consider the shallow water equations (1.1) with $\boldsymbol{s} = \mathbf{0}$ subject to the following initial conditions:

$$h(x,0) = \begin{cases} h_l, & x < x_0, \\ h_r, & x > x_0, \end{cases} \qquad u(x,0) = \begin{cases} u_l, & x < x_0, \\ u_r, & x > x_0. \end{cases}$$

We consider transmissive boundary conditions on h and hu. The Riemann problem, also known as the dam-break problem over a flat bottom, has the following exact solutions [4]:

$$\sqrt{gh(x,t)} = \begin{cases} \sqrt{gh_l}, & (x-x_0) < \left(u_l - \sqrt{gh_l}\right)t, \\ \frac{1}{3}\left[u_l + 2\sqrt{gh_l} - \frac{x-x_0}{t}\right], & \left(u_l - \sqrt{gh_l}\right)t < (x-x_0) < S_l t, \\ 0, & S_l t < (x-x_0) < S_r t, \\ \frac{1}{3}\left[\frac{x-x_0}{t} - u_r + 2\sqrt{gh_r}\right], & S_r t < (x-x_0) < \left(u_r + \sqrt{gh_r}\right)t, \\ \sqrt{gh_r}, & x - x_0 > \left(u_r + \sqrt{gh_r}\right)t, \end{cases}$$

and

$$u(x,t) = \begin{cases} u_l, & (x-x_0) < \left(u_l - \sqrt{gh_l}\right)t, \\ \frac{1}{3}\left[u_l + 2\sqrt{gh_l} + 2\frac{x-x_0}{t}\right], & \left(u_l - \sqrt{gh_l}\right)t < (x-x_0) < S_l t, \\ 0, & S_l t < (x-x_0) < S_r t, \\ \frac{1}{3}\left[2\frac{x-x_0}{t} + u_r - 2\sqrt{gh_l}\right], & S_r t < (x-x_0) < \left(u_r + \sqrt{gh_r}\right)t, \\ u_r, & x - x_0 > \left(u_r + \sqrt{gh_r}\right)t, \end{cases}$$

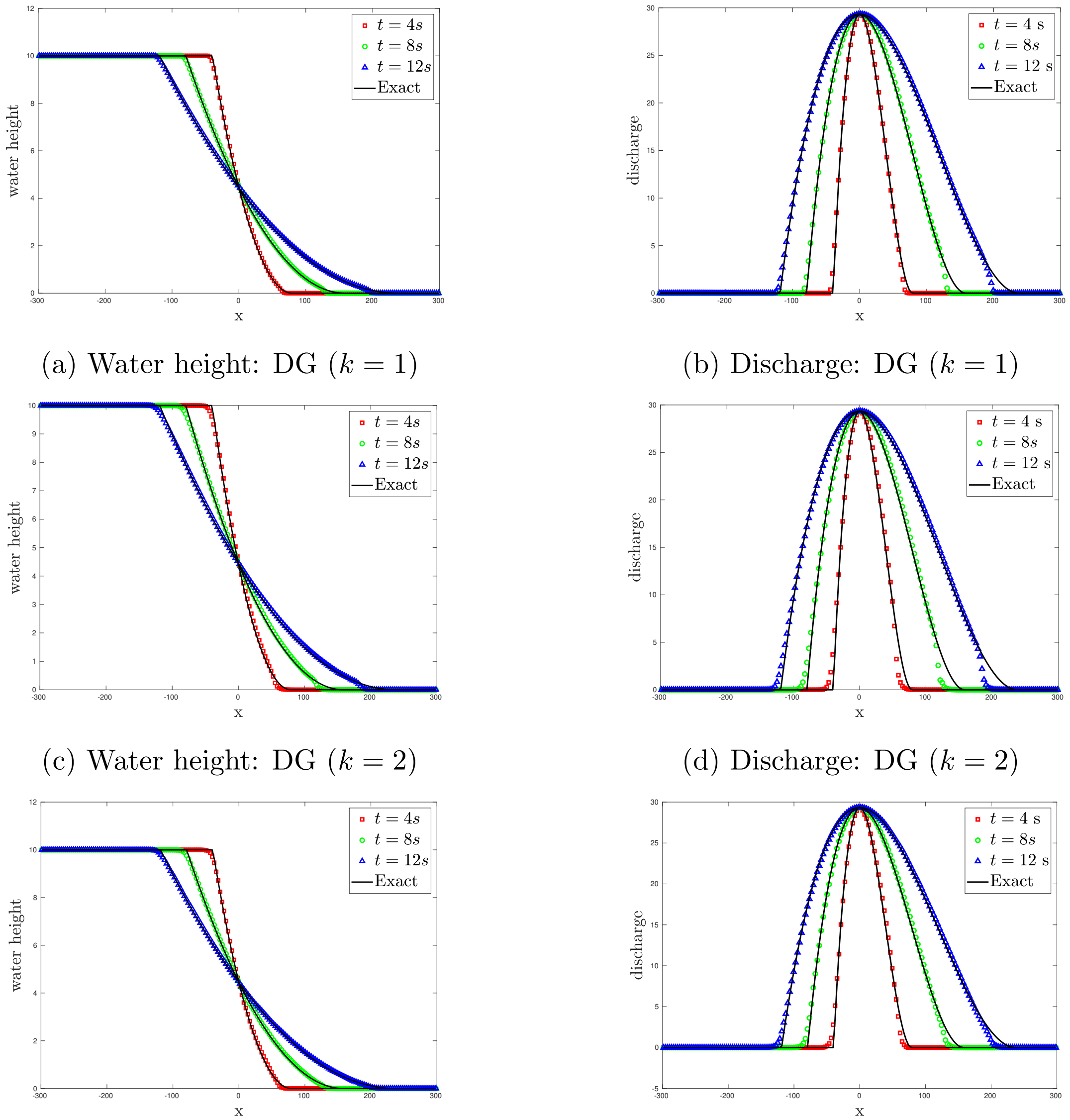
where $S_l = u_l + 2\sqrt{gh_l}$ and $S_r = u_r - 2\sqrt{gh_r}$. Let $x_0 = 0$. We consider the following two sets of initial conditions:

$$h(x,0) = \begin{cases} 10, & x \le 0, \\ 0, & x > 0, \end{cases} \qquad u(x,0) = \begin{cases} 0, & x \le 0, \\ 0, & x > 0, \end{cases} \tag{3.3}$$

and

$$h(x,0) = \begin{cases} 5, & x \le 0, \\ 10, & x > 0, \end{cases} \qquad u(x,0) = \begin{cases} 0, & x \le 0, \\ 40, & x > 0. \end{cases} \tag{3.4}$$

We compute the numerical solution on a mesh with 200 uniform elements. In Figures 3.1 and 3.2 we compare the exact and numerical approximations at different time levels. In Figure 3.1 we used the initial conditions in (3.3) and in Figure 3.2 we used the initial conditions in (3.4). We observe that the numerical solution compares well with the exact solution. We furthermore observe that the DG method is able to handle well the zero depth height of initial condition pair (3.3).



(a) Water height: DG ($k = 1$)

(b) Discharge: DG ($k = 1$)

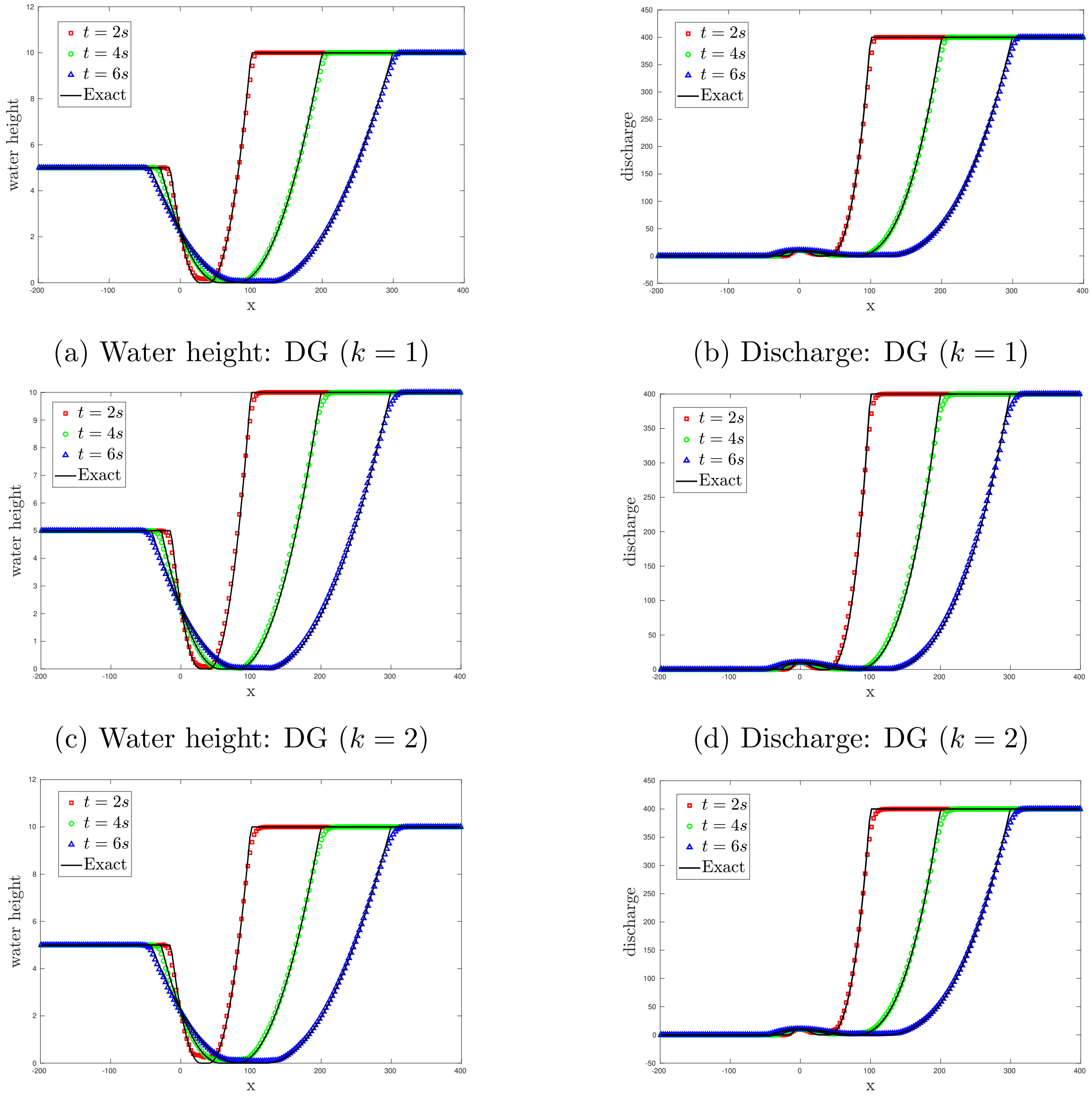
(c) Water height: DG ($k = 2$)

(d) Discharge: DG ($k = 2$)

(e) Water height: DG ($k = 3$)

(f) Discharge: DG ($k = 3$)

Figure 3.1: The numerical and exact solutions of the first Riemann problem with initial condition (3.3) at different times and for polynomial degrees $k = 1, 2, 3$.



(a) Water height: DG ($k = 1$)

(b) Discharge: DG ($k = 1$)

(c) Water height: DG ($k = 2$)

(d) Discharge: DG ($k = 2$)

(e) Water height: DG ($k = 3$)

(f) Discharge: DG ($k = 3$)

Figure 3.2: The numerical and exact solutions of the Riemann problem with initial condition (3.4) at different times and for polynomial degrees $k = 1, 2, 3$.

3.2.4 Drain on a non-flat bottom

For this example we choose a domain $\Omega = (0, 25)$. Consider (1.1) with the following initial condition

$$h(x, 0) = 0.5 - b(x), \qquad hu(x, 0) = 0,$$

and bottom topography given by

$$b(x) = \begin{cases} 0.2 - 0.05\,(x - 10)^2\,, & 8 \leq 12, \\ 0, & \text{otherwise.} \end{cases}$$

At $x = 0$ we consider reflective boundary conditions. Following [9] we impose $h(x^+_{N_e}, t) := 10^{-15}$ and $u(x^+_{N_e}, t) := u(x^-_{N_e}, t)$ at $x = 25$.

In our simulation we use 250 uniform elements and consider the solution for $k = 1$ and $k = 2$. Figure 3.3 illustrates the solutions at various times, specifically at $t = 10$, 20, and 100. After an extended period of time, the solution stabilizes, showing still water the left of the bump and a dry area on the right. The DG method is able effectively deal with the dry area.

3.2.5 Vacuum occurrence by a double rarefaction wave over a step

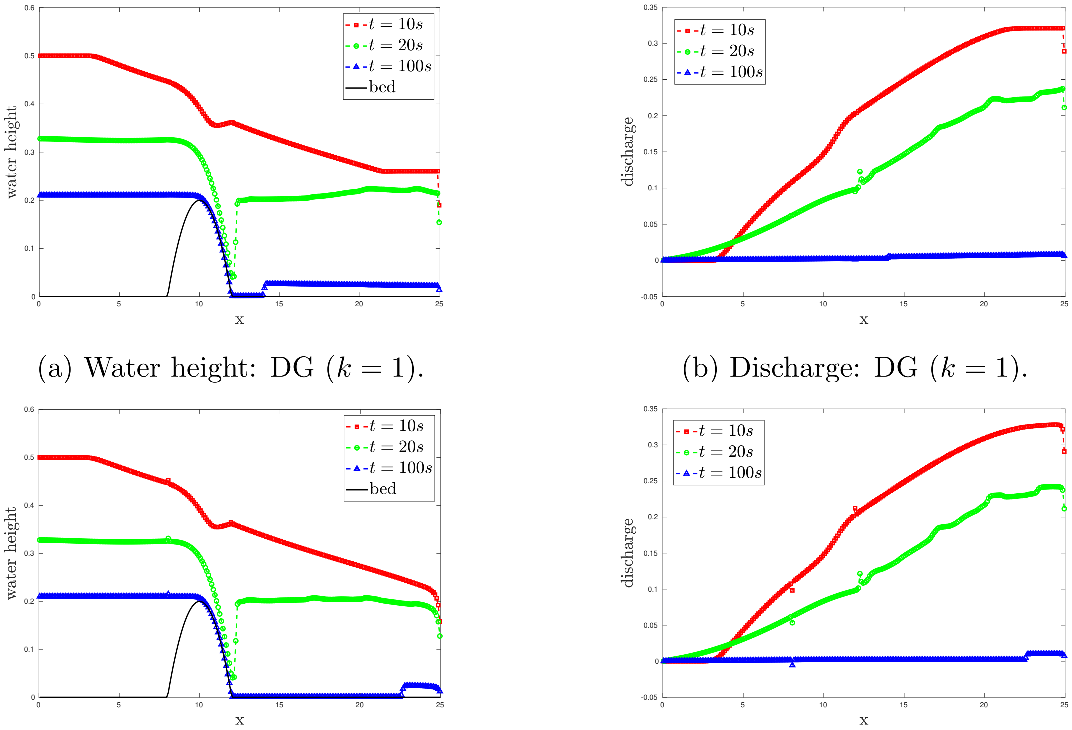
We consider a problem studied by Gallouet et al. [9] (see also [3, 14, 8]). Consider a domain $\Omega = (0, 25)$. We consider the SWE (1.1) in which the discontinuous bottom is defined as

$$b(x) = \begin{cases} 1, & \text{if } \frac{25}{3} \leq x \leq 12.5, \\ 0, & \text{otherwise.} \end{cases}$$

The initial water surface is $h + b = 10$ and the initial discharge is:

$$hu(x, 0) = \begin{cases} -350 & \text{if } x \leq \frac{50}{3}, \\ 350, & \text{otherwise.} \end{cases}$$

We impose transmissive BCs for h and hu. We plot the solution computed on a mesh with 250 element in Figure 3.4 for $k = 1, 2, 3$ at different time levels. The solutions compare well with those found in the literature, for example, [9, Figure 18]. We furthermore remark the ability of the DG method to deal with zero water height.



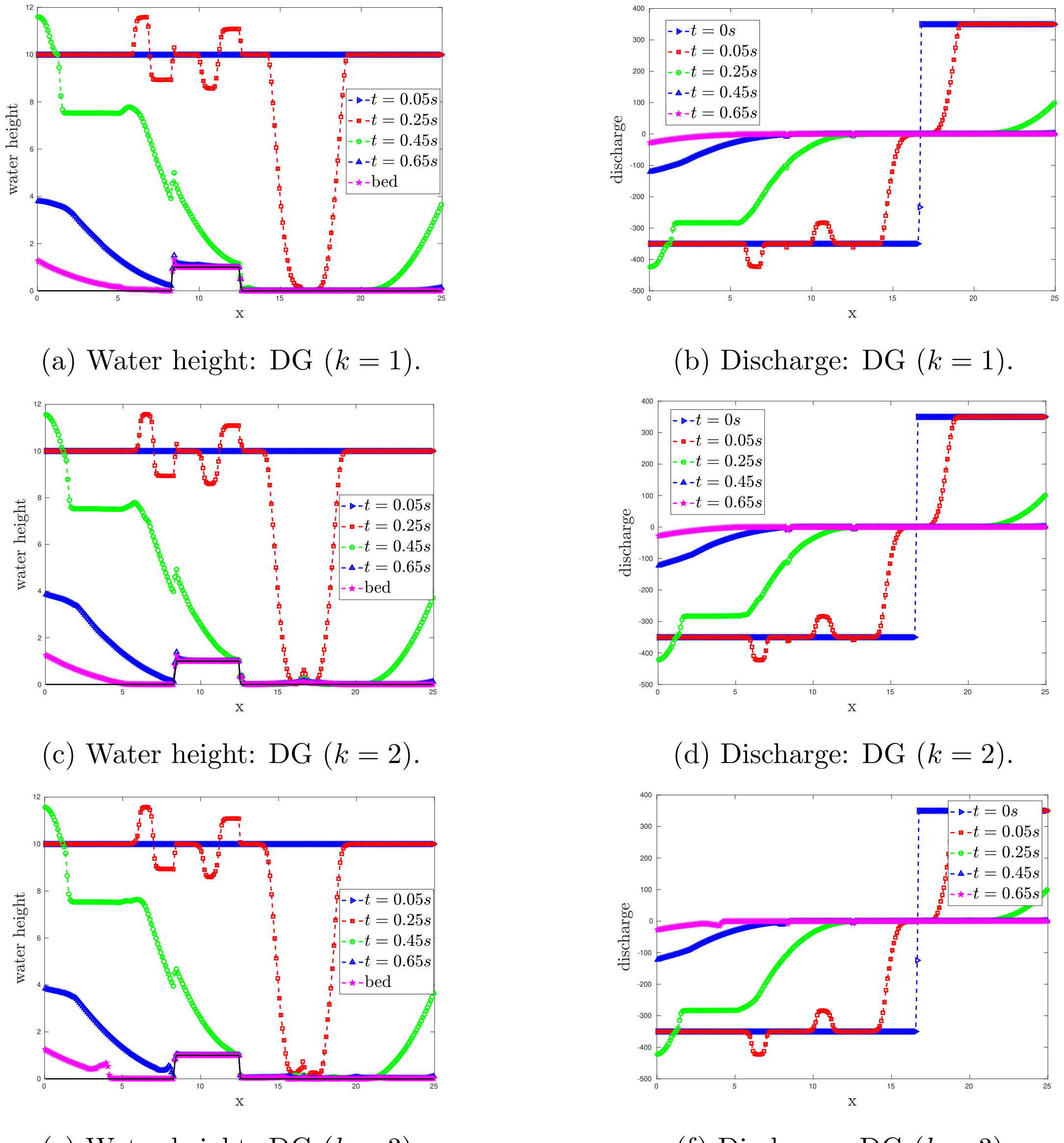
(a) Water height: DG ($k = 1$).

(b) Discharge: DG ($k = 1$).

(c) Water height: DG ($k = 2$).

(d) Discharge: DG ($k = 2$).

Figure 3.3: The numerical solution of the drain on a non-flat bottom example of Section 3.2.4 at different time levels and for polynomial degrees $k = 1, 2$.



(a) Water height: DG ($k = 1$).

(b) Discharge: DG ($k = 1$).

(c) Water height: DG ($k = 2$).

(d) Discharge: DG ($k = 2$).

(e) Water height: DG ($k = 3$).

(f) Discharge: DG ($k = 3$).

Figure 3.4: The numerical solution of the vacuum occurrence by a double rarefaction wave problem described in Section 3.2.5. The numerical solution is computed at different time levels on a mesh with 250 uniform elements. We consider polynomial approximations with $k = 1, 2, 3$.

3.3 Two-dimensional examples

The two-dimensional shallow water equations are given by:

$$\partial_t \boldsymbol{u} + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{s}, \tag{3.5}$$

where

$$\boldsymbol{u} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \boldsymbol{f}(\boldsymbol{u}) = \begin{bmatrix} hu & hv \\ hu^2 + \frac{1}{2}gh^2 & huv \\ huv & hv^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad \boldsymbol{s} = \begin{bmatrix} 0 \\ -gh\partial_x b \\ -gh\partial_y b \end{bmatrix}.$$

We refer to [5, 15] for a discretization of the 2D shallow water equations. All simulations in this section only consider $k = 1$ in our polynomial approximation.

3.3.1 An oscillating lake

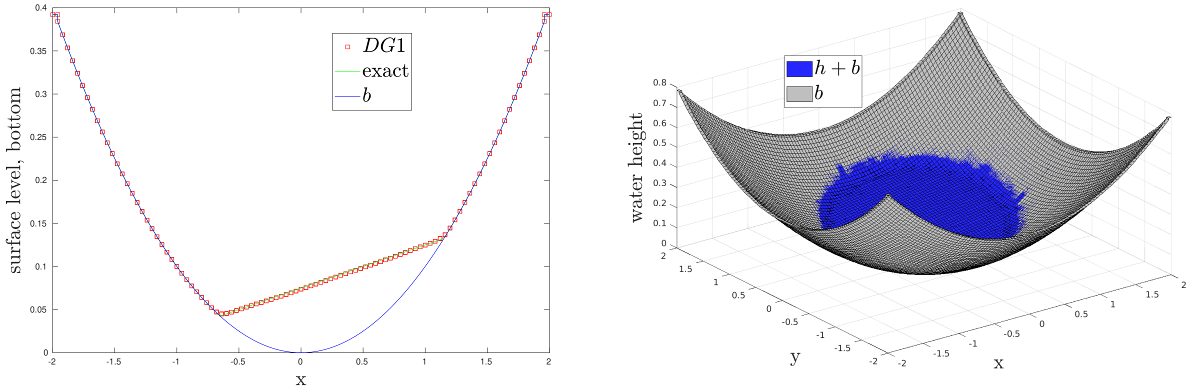
We consider the example from [15, Section 6.8]. For this we consider (3.5) on the domain $\Omega = (-2, 2)^2$. The initial conditions are such that the exact solution to our problem is given by:

$$\begin{aligned} h(x, y, t) &= \max\left(0, \frac{\sigma h_0}{a^2}(2x\cos(\omega t) + 2y\sin(\omega t) - \sigma) + h_0 - b(x, y)\right), \\ u(x, y, t) &= -\sigma\omega\sin(\omega t), \qquad v(x, y, t) = \sigma\omega\cos(\omega t), \end{aligned}$$

where $h_0 = 0.1$, $\sigma = 0.5$, $a = 1$, $\omega = \frac{\sqrt{2gh_0}}{a}$, $T = \frac{2\pi}{\omega}$ and the bottom topography is given by

$$b(x, y) = h_0 \frac{x^2 + y^2}{a^2}.$$

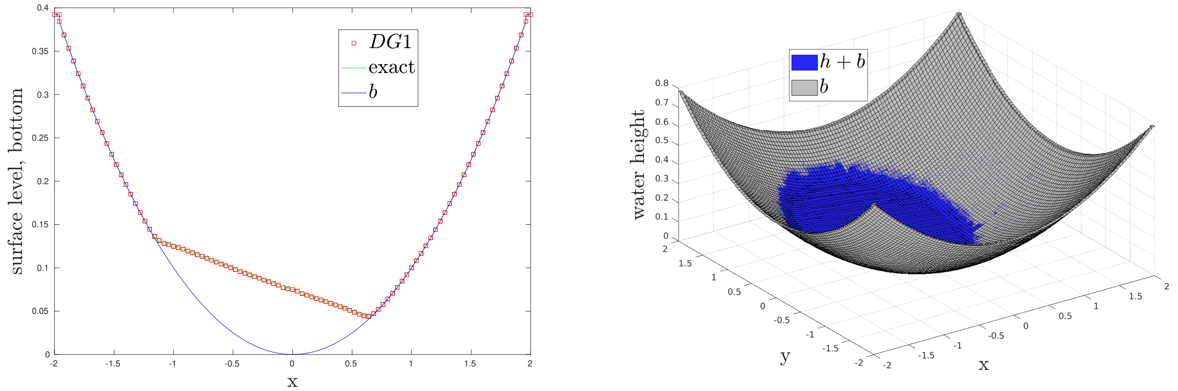
Given that the flow does not extend to the boundaries, we have the flexibility to choose any boundary conditions without influencing the outcomes of the numerical solutions. The simulation runs until time $\frac{T}{2}$ on a mesh using 100×100 uniform elements. We plot the water surface at different intervals in Figures 3.5, 3.6, and 3.7. In these figures we also plot the water surface along the line $y = 0$ and compare our numerical results with the exact solution. We observe that our numerical results compare well with the exact solution. Additionally, the DG method is able to correctly deal with the zero water depth.



(a) Water height: $y = 0$.

(b) Water height

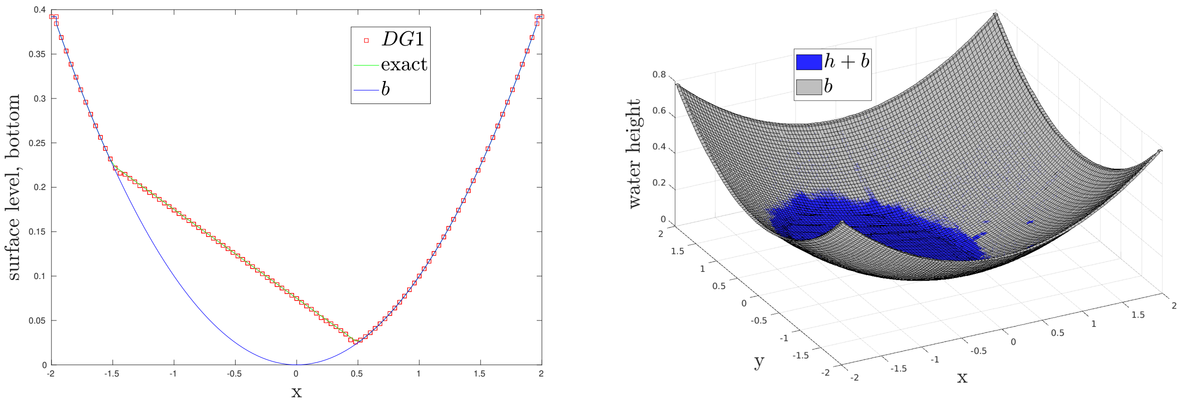
Figure 3.5: The water surface level in the two-dimensional oscillating lake problem at $t = \frac{T}{6}$: DG ($k = 1$) using 100×100 cell.



(a) Water height: $y = 0$.

(b) Water height

Figure 3.6: The water surface level in the two-dimensional oscillating lake problem at $t = \frac{T}{3}$: DG ($k = 1$) using 100×100 cell.



(a) $y = 0$.

(b) Water height

Figure 3.7: The water surface level in the two-dimensional oscillating lake problem at $t = \frac{T}{2}$: DG ($k = 1$) using 100×100 cell.

3.3.2 Flow over a non-flat bottom

In this final example we consider the shallow water equations (3.5) on the domain $\Omega = (-1, 1)^2$. We define the bottom topography as

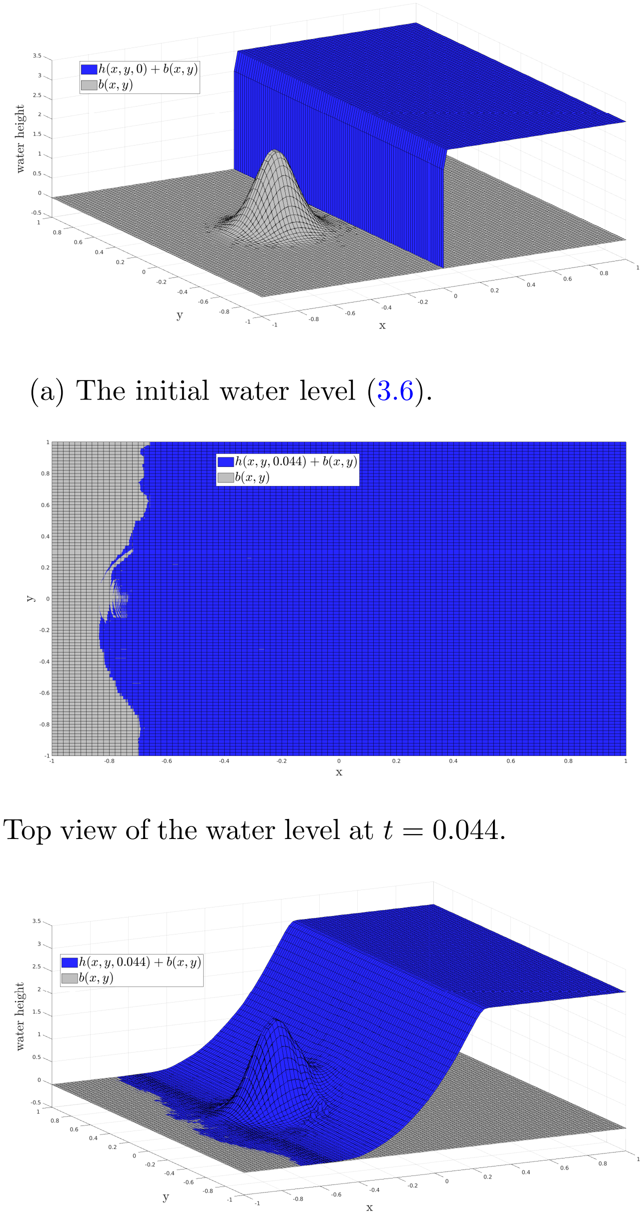
$$b(x, y) = 2e^{-\frac{(x-x_0)^2}{2c_0^2} - \frac{(y-y_0)^2}{2c_0^2}}, \qquad x_0 = -0.35,\ y_0 = 0,\ c_0 = 0.1,$$

and prescribe the following initial conditions:

$$h(x, y, 0) = \begin{cases} 3 - b(x, y), & x > 0, \\ 0, & x \leq 0 \end{cases}, \qquad hu(x, 0) = 0. \tag{3.6}$$

We furthermore impose reflective BCs.

We compute the solution on a mesh using $\Delta x = \Delta y = \frac{1}{50}$ until a final time $t = 0.044$. We plot the water surface at initial and final times in Figure 3.8. Once more we observe that the DG method is able to preserve the positivity of the water height.



(a) The initial water level (3.6).

(b) Top view of the water level at $t = 0.044$.

(c) Side view of the water level at $t = 0.044$.

Figure 3.8: The water level at $t = 0$ and $t = 0.044$ for the problem described in Section 3.3.2.

Chapter 4

Conclusions

In this research paper, we implemented a high-order well balanced discontinuous Galerkin method for the nonlinear shallow water equations with bottom topography. This report was divided into two parts, a theoretical part in Chapter 2 and a numerical simulations part in Chapter 3.

In Chapter 2 we first reviewed the DG method for the one-dimensional SWE with flat bottom topography. We discussed the DG discretization, Runge–Kutta time stepping, and slope limiters. We proceeded with reviewing the DG method for the SWE with non-flat bottom topography. We showed that a naive DG discretization is not well-balanced and then proceeded, following the approach of [1, 10], to derive a modified DG method that is well-balanced. This was achieved by modifying the numerical fluxes. Since positivity preservation of the water depth is crucial when solving the SWE, we finally followed the approach of [15] to define a positivity preserving limiter.

Next, in Chapter 3 we ran numerical simulations with the DG method. First we considered manufactured solutions to find the expected rates of convergence of $k + 1$ if polynomials of degree k are used in the approximation. We proceeded with a variety of one and two dimensional test cases and demonstrated that the DG method is well suited to solve the SWE even when the water depth approaches zero.

References

[1] 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 Journal on Scientific Computing*, 25(6):2050–2065, 2004.

[2] Stavros Avgerinos. All-Mach number solvers for the Euler equations and the Saint-Venant-Exner model. *Phd thesis*, 2019.

[3] Christophe Berthon and Fabien Marche. A positive preserving high order VFRoe scheme for shallow water equations: a class of relaxation schemes. *SIAM Journal on Scientific Computing*, 30(5):2587–2612, 2008.

[4] Onno Bokhove. Flooding and drying in discontinuous Galerkin finite-element discretizations of shallow-water equations. Part 1: one dimension. *Journal of Scientific Computing*, 22:47–82, 2005.

[5] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. the multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

[6] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.

[7] Bernardo Cockburn and Chi-Wang Shu. Tvb runge-kutta local projection discontinuous Galerkin finite element method for conservation laws. II. general framework. *Mathematics of Computation*, 52(186):411–435, 1989.

[8] José M Gallardo, Carlos Parés, and Manuel Castro. On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas. *Journal of Computational Physics*, 227(1):574–601, 2007.

[9] Thierry Gallouët, Jean-Marc Hérard, and Nicolas Seguin. Some approximate godunov schemes to compute shallow-water equations with topography. *Computers & Fluids*, 32(4):479–513, 2003.

[10] Sebastian Noelle, Normann Pankratz, Gabriella Puppo, and Jostein R Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *Journal of Computational Physics*, 213(2):474–499, 2006.

[11] Chi-Wang Shu. TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation*, 49(179):105–121, 1987.

[12] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[13] Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[14] Yulong Xing and Chi-Wang Shu. High-order finite volume WENO schemes for the shallow water equations with dry states. *Advances in Water Resources*, 34(8):1026–1038, 2011.

[15] 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.

[16] Jian G Zhou, Derek M Causon, Clive G Mingham, and David M Ingram. The surface gradient method for the treatment of source terms in the shallow-water equations. *Journal of Computational Physics*, 168(1):1–25, 2001.