AMERICAN UNIVERSITY OF BEIRUT

CENTRAL SCHEME FOR SYSTEMS OF SHALLOW WATER EQUATIONS WITH WET AND DRY STATES

by
FARAH MOHAMAD KANBAR

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Mathematics
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
April 2017

AMERICAN UNIVERSITY OF BEIRUT

Central Scheme for Systems of Shallow Water Equations With Wet and Dry States

by

FARAH KANBAR

Approved by:

Dr. Nabil Nassif, Professor — Advisor
Mathematics

Dr. Issam El Lakkis, Associate Professor — Member of Committee
FEA

Dr. Sophie Moufawad, Assistant Professor — Member of Committee
Mathematics

Dr. Rony Touma, Associate Professor — Member of Committee
Mathematics, LAU

Date of thesis defense: April, 2017

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Kanbar Farah Mohamad
Last First Middle

☒ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☒ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

Signature May 5, 2017
Date

ACKNOWLEDGEMENTS

First, I would like to give my deepest appreciation to my advisor Prof. Rony Touma from the Lebanese American University (LAU) who was always supporting and motivating me throughout my whole work.

Impressed by Prof. Nabil Nassif, who stirred my interest in numerical analysis, I made the decision of deepening my knowledge in that field and trying to achieve such an accomplishment.

Moreover, I am grateful for having professors Sophie Moufawad and Issam El-Lakkis in my committee.

Finally, I cannot disregard the fact that my entire family and my friends have been by my side, igniting my potentials and pushing me to work harder and harder.

AN ABSTRACT OF THE THESIS OF

Farah Kanbar for Master of Science
Major: Mathematics

Title: Central Scheme for Systems of Shallow Water Equations with wet and Dry States

In this thesis we present a new well-balanced, non-oscillatory, second-order accurate central scheme for the numerical solution of the two-dimensional shallow water equations (SWE) with wet and dry states. The numerical scheme is a central scheme that follows a classical Riemann-free finite volume method and that evolves the numerical solution on a single Cartesian grid. Most numerical schemes generate numerical instabilities, such as negative water heights, when considered with wet and dry regions. The developed well-balanced numerical scheme is capable of maintaing, when necessary, the steady state requirement of SWE systems, along with a proper and clean interaction between wet and dry states whenever water run-ups are present. The developed scheme is then validated and the numerical solution of recent two-dimensional SWE problems is reported.

Contents

Chapter 1

Introduction

Central schemes are considered simple and efficient tools for the numerical solution of systems of hyperbolic conservation laws. Lax-Friedrichs scheme is a numerical method for the solution of hyperbolic partial differential equations. Based on this scheme, Nessyahu and Tadmor introduced a central scheme that evolves a piecewise linear numerical solution on two staggered grids in 1990 [1]. Later on, extensions of the NT scheme were developed [29-31]. These schemes were used to solve problems arising in aerodynamics and magnetohydrodynamics. In 1998, Jiang et al presented an unstaggered adaptation of the NT scheme [27]. A two-dimensional extension of this scheme is then presented in [28]. In this thesis we develop a new well-balanced unstaggered central scheme for the numerical solution of systems of two-dimensional shallow water equations with wet and dry states. The shallow water equations (SWE) system is a set of hyperbolic partial differential equations that describes the flow of a free surface fluid under gravity over variable bottom topography. The equations are derived by depth-integrating the Navier–Stokes equations in the case where the horizontal length scale is much greater than the vertical length scale. The system can be used to model waves in the atmosphere, rivers, lakes and oceans. It is also used to simulate tsunamis, dam breaches and others. The SWE system, written in its conservative from, is given by:

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} = \begin{pmatrix} 0 \\ -gh\frac{\partial z}{\partial x} \\ -gh\frac{\partial z}{\partial y} \end{pmatrix}$$

Where t is the time variable, x and y are the two spatial variables, (u, v) are the velocities in the x-direction and y-direction respectively, h is the water height function $h(x, y)$, z is the water bed function $z(x, y)$ and g is the gravitational constant.

Many Papers were recently published for the numerical solution of the shallow water equations using Riemann solvers [5,7-18]. Accurate numerical schemes were developed by balancing between the source term and the flux divergence [12,13]. A one-dimensional well-balanced scheme is derived in [4], where the discretization of the source term and the flux divergence were taken properly. The one-dimensional well-balanced scheme, derived in [4], was extended to a two-dimensional one in [6]. The scheme features the well-balancing of the source term with the flux divergence and the proper discretization of the water height function according to the surface gradient method discussed in [5].

Solving steady state problems is one of the main challenges in the simulation of the SWE problems. For this reason, new well-balanced central schemes for the shallow water equations were developed [4,6,13,19] to solve steady state problems. Another challenge is the development of well-balanced schemes that treat problems with wet and dry states. The appearance of such states usually result from the way the initial water height is defined or from the flow of water across the computational domain. So, we may have wet areas where $h > 0$ and dry areas where $h = 0$. Most numerical schemes break down when they are considered with wet and dry regions and generate negative water heights in the forward and the backward projection steps due to the steep gradient of the interpolant which performs a negative interpolated water height (i.e below the water bed). Therefore, our aim is to propose a treatment that yields clean forward and backward steps, and also maintain the conservation of water across the computational domain as well as the well-balanced property of the developed scheme. A list of numerical schemes were developed in [20-26] to satisfy both the lake at rest and the wet and dry constrain.

A new unstaggered, well-balanced, non-oscillatory, and second-order accurate central scheme for the one-dimensional system of shallow water equations with wet and dry treatment was proposed in [4]. The scheme is capable of handling lake at rest problems at the discrete level by the aid of the surface gradient method. Simultaneously, the scheme treats the numerical instabilities resulting from the wet and dry interactions by correcting the slope of the water height interpolant in the forward and backward projection steps.

In this thesis we develop a new well-balanced, non-oscillatory, and second order accurate central scheme for the two-dimensional system of shallow water equations on variable bottom topography with wet and dry states. It is an extension of the one-dimensional scheme presented in [4]. The extended scheme ensures the water conservation across the computational domain and maintain the positivity requirement of the water height function. Our numerical experiments confirm that the well-balanced property at the discrete level together with the surface gradient method is able to handle lake at rest problems. Besides, it is able to handle problems with wet and dry interactions. The good agreement between our numerical results and the ones proposed in the recent literature as well as the comparison with the one-dimensional experiments prove the efficiency of the developed scheme. The paper is organized as follows: In chapter 2, we present the one-dimensional well-balanced scheme that solves lake at rest problems by the aid of the surface gradient method as well as problems with wet and dry regions with a specific treatment. In chapter 3, we present the two-dimensional extension of the scheme in section 1-2. Furthermore, we develop a treatment of wet and dry states for the two-dimensional scheme in section 3 followed by numerical experiments to test the efficiency of the developed scheme. In chapter 4, we conclude with important remarks and future work.

Chapter 2

One-dimensional Well-balanced Central Scheme

In this section we present an overview of the previously developed one-dimensional central scheme and a list of experiments on which the well-balanced scheme is applied.

2.1 Well-balanced Central Scheme for the One-dimensional Shallow Water Equations

In the one-dimensional case, the SWE system is given by

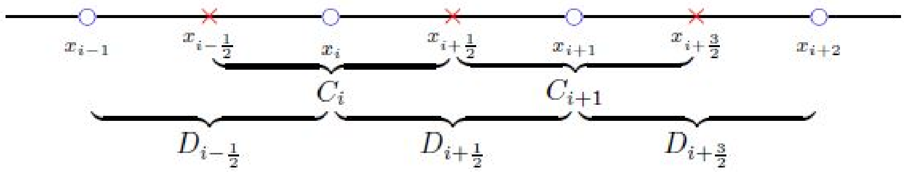
$$\begin{cases} \partial_t u + \partial_x f(u) = S(u,x), \qquad u = u(x,t),\ x \in \Omega \subset \mathbb{R},\ t > 0 \\ u(x,0) = u_0(x) \end{cases}$$

We will apply the central NT scheme [1] to the shallow water equations:

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hv \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} hv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} = \begin{pmatrix} 0 \\ -gh\frac{\partial z}{\partial x} \end{pmatrix}$$

Where $h(x,t)$ denotes the water height, $v(x,t)$ is the velocity in the x-direction, g is the gravitational constant, and $z(x)$ denotes the bottom topography function.

The computational domain Ω is an interval of the real line divided into control cells defined by $C_i = \left[x_{i-1/2}, x_{i+1/2}\right]$ of equal width $\Delta x = x_{i+1/2} - x_{i-1/2}$ and centered at x_i. We also define the dual cells $D_{i+\frac{1}{2}} = [x_i, x_{i+1}]$ with centers $x_{i+1/2} = x_i + \frac{\Delta x}{2}$. The time step will be denoted by Δt, and for a positive integer n we set $t^{n+1} = n\Delta t$.



We assume that the solution u_i^n at time t^n is known at the nodes x_i where $u_i^n = u(x_i, t^n)$, and is defined as a piecewise linear function over C_i. Our aim is to compute the solution u_i^{n+1} at time $t^{n+1} = t^n + \Delta t$.

We start by integrating $u_t + f(u)_x = S(u,x)$ over the domain $R^n_{i+\frac{1}{2}} = D_{i+\frac{1}{2}} \times [t^n, t^{n+1}]$

$$\iint_{R^n_{i+1/2}} [u_t + f_x(u)]dR = \iint_{R^n_{i+1/2}} S(u,x)dR \tag{2.1}$$

then we apply Green's formula to the double integral on the left-hand side of equation (2.1), which allows us to change the double integral into a line integral by the following formula:

$$\iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right)dxdy = \oint_R (Pdx + Qdy) \tag{2.2}$$

After applying this formula with $\frac{\partial Q}{\partial x} = f(u)_x$ and $\frac{\partial P}{\partial y} = -u_t$, Eq.(1.1) becomes:

$$\oint_{\partial R^n_{i+1/2}} (f(u)dt - udx) = \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(u,x)dxdt \tag{2.3}$$

where the boundary of the squared region $R^n_{i+1/2}$ is as follows:

$\partial R^n_{i+1/2} = [x_i, x_{i+1}] \cup [t^n, t^{n+1}] \cup [x_{i+1}, x_i] \cup [t^{n+1}, t^n]$

Dividing the line integral over the four segments, we get:

$$\int_{x_i}^{x_{i+1}} [f(u(x,t^n))dt - u(x,t^n)dx] + \int_{t^n}^{t^{n+1}} [f(u(x_{i+1},t))dt - u(x_{i+1},t)dx]$$
$$+ \int_{x_{i+1}}^{x_i} [f(u(x,t^{n+1}))dt - u(x,t^{n+1})dx] + \int_{t^{n+1}}^{t^n} [f(u(x_i,t))dt - u(x_i,t)dx]$$
$$= \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(u,x)dxdt$$

Splitting the integrals and rearranging them simplifies the equation to:

$$-\int_{x_i}^{x_{i+1}} u(x,t^n)dx + \int_{t^n}^{t^{n+1}} f(u(x_{i+1},t))dt + \int_{x_i}^{x_{i+1}} u(x,t^{n+1})dx - \int_{t^n}^{t^{n+1}} f(u(x_i,t))dt$$
$$= \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(u,x)dxdt \quad (2.4)$$

Next, we apply the Mean-Value theorem for integrals of the function $u(x,t)$, note that the theorem is applicable because the function is piecewise linear at the cell centers.

The integrals become:

$\int_{x_i}^{x_{i+1}} u(x,t^{n+1})dx = \Delta x u_{i+\frac{1}{2}}^{n+1}$

$\int_{x_i}^{x_{i+1}} u(x,t^n)dx = \Delta x u_{i+\frac{1}{2}}^{n}$

For flux integrals we apply the midpoint quadrature rule.

The integrals become:

$\int_{t^n}^{t^{n+1}} f(u(x_i,t))dt \approx f(u_i^{n+\frac{1}{2}}).\Delta t$

$\int_{t^n}^{t^{n+1}} f(u(x_{i+1},t))dt \approx f(u_{i+1}^{n+\frac{1}{2}}).\Delta t$

We Substitute these 4 integrals in Eq.(1.4) and we divide by Δx to obtain:

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^{n} - \frac{\Delta t}{\Delta x}[f(u_{i+1}^{n+\frac{1}{2}}) - f(u_i^{n+\frac{1}{2}})] + \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} S(u,x)dxdt. \quad (2.5)$$

where $u_i^{n+\frac{1}{2}}$ are the predicted values at the intermediate time $t^{n+\frac{1}{2}}$ and they are approximated using a first-order Taylor expansion in time:

$u(x,t) = u(x,a) + u_t(x,a)(t-a) \quad$ for any a and t.

$u(x_i,t) = u(x_i,a) + u_t(x_i,a)(t-a)$

Let $a = t^n$,

$u(x_i,t) = u(x_i,t^n) + u_t(x_i,t^n)(t-t^n)$

Let $t = t^{n+\frac{1}{2}}$

$u(x_i,t^{n+\frac{1}{2}}) = u(x_i,t^n) + u_t(x_i,t^n)(\frac{\Delta t}{2})$

$u_i^{n+\frac{1}{2}} = u_i^n + \frac{\Delta t}{2}(u_t)_i^n$

$u_i^{n+\frac{1}{2}} = u_i^n + \frac{\Delta t}{2}[-f_x(u_i^n) + S(u_i^n, x_i)]$

But the partial derivatives of the flux are calculated using the chain rule, Then:

$u_i^{n+\frac{1}{2}} = u_i^n + \frac{\Delta t}{2\Delta x}[-f' + S_i^n.\Delta x]$

The integral of the source term is being discretized using the midpoint quadrature rule with respect to time and space:

$$\begin{aligned}
\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} S(u(x,t),x)dxdt &= \int_{x_i}^{x_{i+1}} \Delta t S(u(x,t^{n+\frac{1}{2}}),x)dx \\
&= \frac{\Delta x\Delta t}{2}\left[S(u(x_{i+1},t^{n+\frac{1}{2}}),x_{i+1}) + S(u(x_i,t^{n+\frac{1}{2}}),x_i)\right] \\
&= \frac{\Delta x\Delta t}{2}\left[S(u_{i+1}^{n+\frac{1}{2}},x_{i+1}) + S(u_i^{n+\frac{1}{2}},x_i)\right] \\
&= \frac{\Delta t\Delta x}{2}\left[\begin{pmatrix} 0 \\ -gh_{i+1}^{n+\frac{1}{2}}(\frac{\partial z}{\partial x})_{i+1}\end{pmatrix} + \begin{pmatrix} 0 \\ -gh_i^{n+\frac{1}{2}}(\frac{\partial z}{\partial x})_i\end{pmatrix}\right] \\
&= \frac{\Delta t\Delta x}{2}\begin{pmatrix} 0 \\ -g\left[h_{i+1}^{n+\frac{1}{2}}(\frac{\partial z}{\partial x})_{i+1} + h_i^{n+\frac{1}{2}}(\frac{\partial z}{\partial x})_i\right]\end{pmatrix} \\
&= \Delta t\Delta x\begin{pmatrix} 0 \\ -g\frac{h_{i+1}^{n+\frac{1}{2}}+h_i^{n+\frac{1}{2}}}{2}(\frac{z_{i+1}+z_i}{\Delta x})\end{pmatrix}
\end{aligned}$$

So,

$$\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} S(u(x,t),x)dxdt \approx \Delta t\Delta x S(u_i^{n+\frac{1}{2}}, u_{i+1}^{n+\frac{1}{2}})$$

with

$$S(u_i^{n+\frac{1}{2}}, u_{i+1}^{n+\frac{1}{2}}) = \begin{pmatrix} 0 \\ -g\frac{h_{i+1}^{n+\frac{1}{2}}+h_i^{n+\frac{1}{2}}}{2}(\frac{z_{i+1}+z_i}{\Delta x})\end{pmatrix} \tag{2.6}$$

Discritization of the source term:

we define the sensor function s_i as follows:

$$s_i = \begin{cases} -1, & \text{if } h_i^{'} = \theta\frac{h_i^n - h_{i-1}^n}{\Delta x} \\ 1, & \text{if } h_i^{'} = \theta\frac{h_{i+1}^n - h_i^n}{\Delta x} \\ 0, & \text{if } h_i^{'} = 0 \\ 2, & \text{if } h_i^{'} = \frac{h_{i+1}^n - h_{i-1}^n}{2\Delta x} \end{cases}$$

where $1 \leq \theta \leq 2$ is the MC-θ parameter used in the gradients limiting. This parameter will force the discretization of dz/dx in the source term to follow the discretization of dh/dx. Then, the source term is discretized as follows:

$S_i^n = S_{i,L}^n + S_{i,R}^n + S_{i,C}^n$

where

$$S_{i,L} = s_i^2\frac{1-s_i}{6}(2-s_i)\begin{pmatrix} 0 \\ -gh_i^n\theta\frac{z_i - z_{i-1}}{\Delta x} \end{pmatrix}$$

$$S_{i,R} = s_i^2\frac{1+s_i}{2}(2-s_i)\begin{pmatrix} 0 \\ -gh_i^n\theta\frac{z_{i+1} - z_i}{\Delta x} \end{pmatrix}$$

$$S_{i,C} = s_i\frac{(s_i+1)(s_i-1)}{6}(2-s_i)\begin{pmatrix} 0 \\ -gh_i^n\frac{z_{i+1} - z_{i-1}}{2\Delta x} \end{pmatrix}$$

The source term is then given by:

$$S_i = \begin{cases} \begin{pmatrix} 0 \\ \\ -gh_i^n \theta \frac{z_i - z_{i-1}}{\Delta x} \end{pmatrix}, & \text{if} \quad s_i = -1 \\ \begin{pmatrix} 0 \\ \\ -gh_i^n \theta \frac{z_{i+1} - z_i}{\Delta x} \end{pmatrix}, & \text{if} \quad s_i = 1 \\ \quad 0, & \text{if} \quad s_i = 0 \\ \begin{pmatrix} 0 \\ \\ -gh_i^n \frac{z_{i+1} - z_{i-1}}{2\Delta x} \end{pmatrix}, & \text{if} \quad s_i = 2 \end{cases}$$

To sum up, the steps of the numerical scheme can be summarized as follows:

- Calculate the predicted values $u_i^{n+\frac{1}{2}}$ using the following equation:

$$u_i^{n+\frac{1}{2}} = u_i^n + \frac{\Delta t}{2\Delta x}[-f' + S_i^n.\Delta x]$$

- Calculate the forward projected values $u_{i+\frac{1}{2}}^n$ using Taylor expansion of $u(x, t^n)$ in space

$$u_{i+\frac{1}{2}}^n = \frac{1}{2}(u_i^n + u_{i+1}^n) + \frac{\Delta x}{8}((u_i^n)' - (u_{i+1}^n)') \tag{2.7}$$

  where $(u_i^n)'$ is the derivative of $u(x_i, t^n)$ calculated using the MC-θ limiter defined in [5].

- Calculate the solution at time t^{n+1} on the staggered dual cells using this formula:

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n - \frac{\Delta t}{\Delta x}[f(u_{i+1}^{n+\frac{1}{2}}) - f(u_i^{n+\frac{1}{2}})] + \Delta t S(u_i^{n+\frac{1}{2}}, u_i^{n+\frac{1}{2}}) \tag{2.8}$$

- Project the solution $u_{i+\frac{1}{2}}^{n+1}$ back into the original grid using Taylor expansions in space:

$$u_i^{n+1} = \frac{1}{2}(u_{i-\frac{1}{2}}^{n+1} + u_{i+\frac{1}{2}}^{n+1}) + \frac{\Delta x}{8}((u_{i-\frac{1}{2}}^{n+1})' - (u_{i+\frac{1}{2}}^{n+1})'). \tag{2.9}$$

The Steady State:

Where the surface of the liquid is initially at rest corresponding to initial velocity zero.(i.e. $h + z = cst$)

In this case the variable, the flux, and the source term vector reduce to

$$u = \begin{pmatrix} h \\ \\ 0 \end{pmatrix} \quad f = \begin{pmatrix} 0 \\ \\ \frac{1}{2}gh^2 \end{pmatrix} \quad S = \begin{pmatrix} 0 \\ \\ -gh\frac{\partial z}{\partial x} \end{pmatrix}$$

The solution for the steady state problems of the shallow water equations at time t^{n+1} should be equal to that at time t^n and the equality $h_i + z_i = cst$ should be satisfied at each time step and at every node in the computational domain.

We will prove now that in the steady state case the following equalities hold:

- $u_i^{n+\frac{1}{2}} = u_i^n$
- $u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n$

Proof of the first equality:

$u_i^{n+\frac{1}{2}} = u_i^n + \frac{\Delta t}{2\Delta x}(-(f')_i^n + S_i^n.\Delta x)$

Consider the case where $h_i' = \theta\frac{h_i - h_{i+1}}{\Delta x}$(Similar proof for the remaining cases)

by the definition of the sensors s_i, the source will be defined as:

$$S_i^n = \begin{pmatrix} 0 \\ \\ -gh_i^n\theta\frac{z_i - z_{i-1}}{\Delta x} \end{pmatrix}$$

and

$$F_i' = \begin{pmatrix} 0 \\ \\ gh_ih_i^{'} \end{pmatrix}$$

The equation becomes:

$$\begin{aligned} u_i^{n+\frac{1}{2}} =& u_i^n + \frac{\Delta t}{2\Delta x}\left[\begin{pmatrix} 0 \\ \\ -gh_i^n(h')_i^n - gh_i^n\theta(z_i - z_{i-1}) \end{pmatrix}\right] \\ &= u_i^n + \frac{\Delta t}{2\Delta x}\left[\begin{pmatrix} 0 \\ \\ -gh_i^n[(h')_i^n + \theta(z_i - z_{i-1})] \end{pmatrix}\right] \\ &= u_i^n + \frac{\Delta t}{2\Delta x}\left[\begin{pmatrix} 0 \\ \\ -gh_i^n[\theta(h_i^n + z_i) - \theta(h_{i-1}^n + z_{i-1})] \end{pmatrix}\right] \end{aligned}$$

but we're given that $h_i^n + z_i = cst$ over all the computational domain.

hence,

$$u_i^{n+\frac{1}{2}} = u_i^n$$

proof of the second equality:

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n - \frac{\Delta t}{\Delta x}[f(u_{i+1}^{n+\frac{1}{2}}) - f(u_i^{n+\frac{1}{2}})] + \Delta t S(u_i^{n+\frac{1}{2}}, u_i^{n+\frac{1}{2}})$$

where,

$$S(u_i^{n+\frac{1}{2}}, u_{i+1}^{n+\frac{1}{2}}) = \begin{pmatrix} 0 \\ -g\frac{h_i^{n+\frac{1}{2}}+h_{i+1}^{n+\frac{1}{2}}}{2}\left(\frac{z_{i+1}-z_i}{\Delta x}\right) \end{pmatrix}$$

Then we get,

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n - \frac{\Delta t}{\Delta x}\left[\begin{pmatrix} 0 \\ \frac{1}{2}g[h_i^{n+\frac{1}{2}} + h_{i+1}^{n+\frac{1}{2}}][(h_{i+1}^{n+\frac{1}{2}} + z_{i+1}) - (h_i^{n+\frac{1}{2}} + z_i)] \end{pmatrix}\right]$$

Again, $h + z = cst$ and from 1. we have $u_i^{n+\frac{1}{2}} = u_i^n$

this leads to

$$(h_i^{n+\frac{1}{2}} + z_i) - (h_{i+1}^{n+\frac{1}{2}} + z_{i+1}) = (h_i^n + z_i) - (h_{i+1}^n + z_{i+1}) = 0$$

Therefore,

$$u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n$$

In this section we presented the derivations of the well-balanced scheme. In general, the well-balanced scheme fails to solve lake at problems as we will show in the numerical experiments section. Therefore, a new reformulation of the scheme is needed to remedy this situation as we will present in the next section.

2.2 The Surface Gradient Method for the One-dimensional Shallow Water Equations System

In order for the developed scheme to satisfy the steady state requirement i.e $(h+z=c, v=0)$, some reformulations on the well-balanced central scheme is to be applied.

As proved in the previous section, the following equalities:

$$\begin{cases} u_i^{n+\frac{1}{2}} = u_i^n \\ \\ u_{i+\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^n \end{cases}$$

hold at every node of the computational domain. Therefore a proper reformulation of the scheme is to be done for this equality $u_i^{n+1} = u_i^n$ to hold at every node x_i in the original cells. We will follow the surface gradient method discussed in [4] and calculate the numerical derivative of the water height component in terms of the water level function $H(x,t) = h(x,t) + z(x)$.

The surface gradient method can be summarized by the following steps:

First, we define the bottom topography function $z(x)$ at the centers of the dual cells, i.e, $z_{i+\frac{1}{2}}$ then we find its values on the centers of the control cells using linear interpolations:

$$z(x) = z_i + \frac{1}{\Delta x}(z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}})(x - x_i)$$

At x_i, we define

$$z_i = \frac{1}{2}(z_{i+\frac{1}{2}} + z_{i-\frac{1}{2}})$$

We linearize the water level function $H(x)$ and then use the relation $h(x) = H(x) - z(x)$ to linearize the water height function. So,

$$h_i^{'} = H_i^{'} - z_i^{'}$$

where $H_i^{'}$ is calculated using MC-θ limiter and $z_i^{'}$ is calculated using central difference.

Then,

$$h_i^{'} = H_i^{'} - \frac{1}{\Delta x}(z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}}) \tag{2.10}$$

Substituting $h_i^{'}$ by its value in Eq. (2.7), we get:

$$h^n_{i+\frac{1}{2}} = \frac{1}{2}(h^n_i + h^n_{i+1}) + \frac{\Delta x}{8}[(H'_i - \frac{z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}}}{\Delta x}) - (H'_{i+1} - \frac{z_{i+\frac{3}{2}} - z_{i+\frac{1}{2}}}{\Delta x})] \tag{2.11}$$

It is essential to mention that in the steady state $H(x)$ is constant and $H^{'} = 0$. Moreover, $z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}} = 2(z_{i+\frac{1}{2}} - z_i)$ and $z_{i+\frac{3}{2}} - z_{i+\frac{1}{2}} = 2(z_{i+1} - z_{i-\frac{1}{2}})$

This reduces Eq.(2.11) to:

$$h^n_{i+\frac{1}{2}} = \frac{1}{2}(h^n_i + h^n_{i+1}) + \frac{1}{2}(z_{i+\frac{1}{2}} - \frac{z_i + z_{i+1}}{2}) \tag{2.12}$$

Similar reformulations is to be applied on the backward projection step Eq.(2.9)

We have the relation $H_i = h_i + z_i$ over the control cells $C_i = \left[x_{i-1/2}, x_{i+1/2}\right]$. Similarly, we have:

$$\tilde{H}^{n+1}_{i+\frac{1}{2}} = h^{n+1}_{i+\frac{1}{2}} + \tilde{z}_{i+\frac{1}{2}}. \tag{2.13}$$

where $\tilde{z}_{i+\frac{1}{2}}$ is defined by the following formula due to the fact that the bottom topography function is linear inside the control cells C_i, but not on the dual cells $D_{i+\frac{1}{2}} = [x_i, x_{i+1}]$:

$$\tilde{z}_{i+\frac{1}{2}} = z_{i+\frac{1}{2}} - \frac{1}{2}(z_{i+\frac{1}{2}} - \frac{z_i + z_{i+1}}{2}) \tag{2.14}$$

Then,

$$(h^{n+1}_{i+\frac{1}{2}})' = (\tilde{H}^{n+1}_{i+\frac{1}{2}})' - \frac{1}{\Delta x}(z_{i+1} - z_i). \tag{2.15}$$

Substituting Eq.(2.15) in Eq.(2.9), we obtain:

$$h^{n+1}_i = \frac{1}{2}(h^{n+1}_{i-\frac{1}{2}} + h^{n+1}_{i+\frac{1}{2}}) + \frac{\Delta x}{8}(H^{'}_{i-\frac{1}{2}} - \frac{z_i - z_{i-1}}{\Delta x} - H^{'}_{i+\frac{1}{2}} + \frac{z_{i+1} - z_i}{\Delta x}) \tag{2.16}$$

By applying the surface gradient method we end our reformulations of the well-balanced scheme.

Now, let's prove that steady state is conserved after applying the described reformulations, i.e, $u^{n+1}_i = u^n_i$.

We have proved that it is conserved on the staggered values ($u^{n+1}_{i+\frac{1}{2}} = u^n_{i+\frac{1}{2}}$). That means, it is enough to prove that projecting the staggered values to the nonstaggered ones returns the same values we started with. In the case of steady state the velocity is zero so we will only consider the first component h. Because H_i is constant, then $H_i^{'} = 0$ and Eq. (2.11) becomes:

$$\begin{aligned} h^n_{i+\frac{1}{2}} &= \frac{1}{2}(h^n_i + h^n_{i+1}) + \frac{\Delta x}{8}[(H'_i - \frac{z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}}}{\Delta x}) - (H'_{i+1} - \frac{z_{i+\frac{3}{2}} - z_{i+\frac{1}{2}}}{\Delta x})] \\ &= \frac{1}{2}(h^{n+1}_{i-\frac{1}{2}} + h^{n+1}_{i+\frac{1}{2}}) - \frac{1}{2}(z_{i+\frac{1}{2}} - \frac{z_i + z_{i+1}}{2}) \end{aligned}$$

using the relations $z_{i+\frac{1}{2}} - z_{i-\frac{1}{2}} = 2(z_{i+\frac{1}{2}} - z_i)$ and $z_{i+\frac{3}{2}} - z_{i+\frac{1}{2}} = 2(z_{i+1} - z_{i+\frac{1}{2}})$ and substituting $h^n_{i+\frac{1}{2}}$ by its value in Eq. (2.13), we get that $\tilde{H}^{n+1}_{i+\frac{1}{2}}$ is constant and so $(\tilde{H}^{n+1}_{i+\frac{1}{2}})' = 0$. Now,

$$h^{n+1}_i = \frac{1}{2}(h^{n+1}_{i-\frac{1}{2}} + h^{n+1}_{i+\frac{1}{2}}) + \frac{\Delta x}{8}(H^{'}_{i-\frac{1}{2}} - \frac{z_i - z_{i-1}}{\Delta x} - H^{'}_{i+\frac{1}{2}} + \frac{z_{i+1} - z_i}{\Delta x})$$

By taking into account the values $h^n_{i+\frac{1}{2}}$ and the fact that $H^{'}_{i+\frac{1}{2}} = 0$, we get $h^{n+1}_i = h^n_i$. In this way we end our proof and the steady state is preserved at the centers of the original cells.

2.3 Treatment of Wet and Dry States

The proposed well-balanced scheme leads to numerical instabilities such as negative water heights when SWE problems with wet and dry rejoins are considered. The appearance of such instabilities is due to the gradient limiters that give an interpolated numerical solution falling below the water bed on the dual cells in the forward projection step Eq.(2.7), or an interpolated water height lying below the water bed function on the original cells in the backward projection step Eq.(2.9). Therefore, additional treatment is required to remedy this situation.

The water height interpolant at time t^n on the cell C_i is defined by

$\eta_i^n(x) = h_i^n + (x - x_i)(h_i^n)'$

In situations where the water height is positive or zero, the slope $(h_i^n)'$ of this interpolant may lead to negative water height $h^n_{i+\frac{1}{2}}$

Let us consider a case where $h_{i+1} = 0$ and h_i and h_{i-1} are strictly positive.

On a wet cell $\eta_i^n(x) \approx h(x, t^n)$ for x$\in C_i$ and $\eta_i^n(x_i) = h_i^n > 0$.

On a dry cell we set $\eta_i^n(x) = 0 \quad \forall x \in C_i$.

If for example $\eta_i^n(x_{i+\frac{1}{4}}) < 0$, then $h_{i+\frac{1}{2}} = \frac{1}{2}(\eta_i^n(x_{i+\frac{1}{4}}) + \eta_i^n(x_{i+\frac{3}{4}})) < 0$ because $\eta_i^n(x_{i+\frac{3}{4}}) < |\eta_i^n(x_{i+\frac{1}{4}}|$.

Hence, the idea is to correct this slope in order to make sure that the forward projection step is not executing a negative water height.

If the forward projection step $h^n_{i+\frac{1}{2}}$ leads to negative water heights, we set it to be zero and correct the slopes of the water height interpolants in the cells C_i and C_{i+1} to ensure water conservation across the computational domain. In addition, we extend the definition of the sensor function as well as the discretization of the source term in order to maintain the well-balanced property of the developed scheme.

The treatment can be summarized by the following steps:

- If $h^n_{i+\frac{1}{2}} < 0$, we set it 0.

- Correct the slope of the water height interpolants in the cells C_i and C_{i+1}, two cases arise:

  1. **If** $h_i^n > \epsilon$ we re-linearize the water height function and update the definition of the sensor functions,

     $(h_i^n)' = -\frac{h_i^n}{\frac{dx}{4}}$ and $s_i = -4$

     Otherwise $h_i^n = 0$ and $(h_i^n)' = 0$ and $s_i = 0$.

  2. **If** $h_{i+1}^n > \epsilon$ we re-linearize the water height function and update the definition of the sensor functions,

     $(h_{i+1}^n)' = \frac{h_{i+1}^n}{\frac{dx}{4}}$ and $s_i = 4$

     Otherwise $h_{i+1}^n = 0$ and $(h_{i+1}^n)' = 0$ and $s_i = 0$.

- Update the discretization of the source term:

$$S_i = \begin{cases} \begin{pmatrix} 0 \\ -gh_i^n \theta \frac{z_i - z_{i-1}}{dx} \end{pmatrix}, & \text{if} \quad s_i = -1 \\ \begin{pmatrix} 0 \\ -gh_i^n \theta \frac{z_{i+1} - z_i}{dx} \end{pmatrix}, & \text{if} \quad s_i = 1 \\ 0, & \text{if} \quad s_i = 0 \\ \begin{pmatrix} 0 \\ -gh_i^n \theta \frac{z_{i+1} - z_{i-1}}{2dx} \end{pmatrix}, & \text{if} \quad s_i = 2 \\ \begin{pmatrix} 0 \\ -gh_i^n \frac{z_{i+1/2} - z_i}{\frac{dx}{2}} \end{pmatrix}, & \text{if} \quad s_i = -4 \\ \begin{pmatrix} 0 \\ -gh_i^n \frac{z_{i+1} - z_{i+1/2}}{\frac{dx}{2}} \end{pmatrix}, & \text{if} \quad s_i = 4 \end{cases}$$

Similar treatment is to be applied on the backward projection step:

If $h_i^{n+1} < 0$ we set it 0.

The proposed treatment ensures non-negative water height values in the forward and the backward projection steps, and maintain the well-balanced property of the scheme as well as the conservation of water across the computational domain.

Remarks:

1.

$$\Delta t = \text{CFL} * \frac{\Delta x}{\max(\max(|\text{L1}|, |\text{L2}|))}$$

- L1 and L2 are two vectors containing the eigenvalues of $\frac{\partial f}{\partial u}$ at each node.
- CFL is considered 0.485 in all computations.

2. All simulations have been done using MATLAB.

2.4 Numerical Experiments

Numerical experiments are carried out to test the efficiency of the developed scheme. The gravitational constant for all experiments is $g = 9.8m/s^2$.

Note that the surface gradient method (sec 2.2) cannot be applied in the experiments with Wet/Dry interactions(sec.2.1). In these cases we stick to the well-balanced scheme.

2.4.1 Lake at rest over variable bottom topography

In the first experiment we consider lake at rest over variable bottom topography defined as:

$$z(x) = \begin{cases} 0 & x < 0.25 \\ 1 + x & 0.25 < x < 0.5 \\ 1 - x & 0.5 < x < 0.75 \\ 0 & x > 0.75 \end{cases}$$

we discretize the computational domain [0,1] using 100 grid points. The initial conditions are $h(x,0) + z(x) = 5$ and $v(x,0) = 0 \ \forall x \in [0,1]$. Figure 2.1 illustrates the water level at time t=0.07 where we compare the water level obtained with (solid line) and without (dashed line) applying the surface gradient method to the numerical base scheme. When applying the surface gradient method, the oscillations disappear and the lake remains at rest. This confirm the well-balanced property of the developed scheme. Table 2.1 report the L_1 error together with its order using 100, 200 and 400 grids.

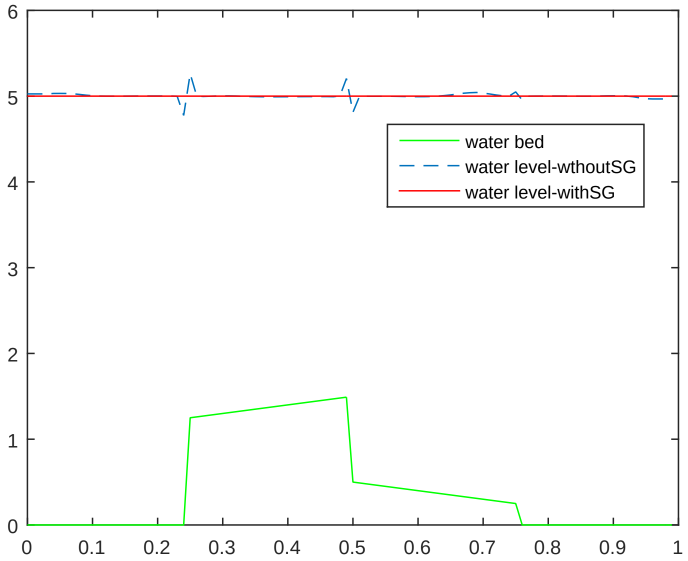


Figure 2.1: One-dimensional lake at rest problem: water level obtained using the well-balanced scheme at time t=0.07 with (solid line) and without (dashed line) SG.

2.4.2 Dam break over rectangular bump

For our second experiment we consider a dam break problem over a rectangular bump. The waterbed function is given by

$$z(x) = \begin{cases} 5 & |x - \frac{1500}{2}| < 1500/4 \\ 0 & \text{otherwise} \end{cases}$$

The water level function is initially

$$H(x) = \begin{cases} 20 & x < \frac{1500}{2} \\ 15 & \text{otherwise} \end{cases}$$

$h(x,0) = H(x) - z(x)$ and $v(x,0) = 0$. The computational domain [0,1500] is discretized using 600 grid points. Figure 2.2 describes the profile of the water level at time t=10 after applying the well-balanced scheme with (solid line) and without (dashed line) applying the

Table 2.1: L_1 error of convergence for the lake at rest problem using different grids N with respect to N=600.

N	L_1 error	order
100	0.42×10^{-2}	
200	1.7×10^{-3}	1.31
400	4.16×10^{-4}	2.03

surface gradient method (SG).

2.4.3 Two symmetric dam break problems

Symmetric dam break problem over triangular bottom topography is considered in this experiment. The computational domain is [-2,2] which is discretized using 100 grid points. The water bed function is defined as:

$$z(x) = \begin{cases} 1 & x < -1 \\ 2 + x & -1 \leq x \leq 0 \\ 2 - x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

the water height is initially defined as:

$$h(x) = \begin{cases} 2 & x < -1 \\ 0 & -1 \leq x \leq 1 \\ 2 & x > 1 \end{cases}$$

and the initial velocity is zero. The results are presented in figure 2.3 where we apply the proposed scheme. Reflecting boundary conditions are applied on both sides of the domain.

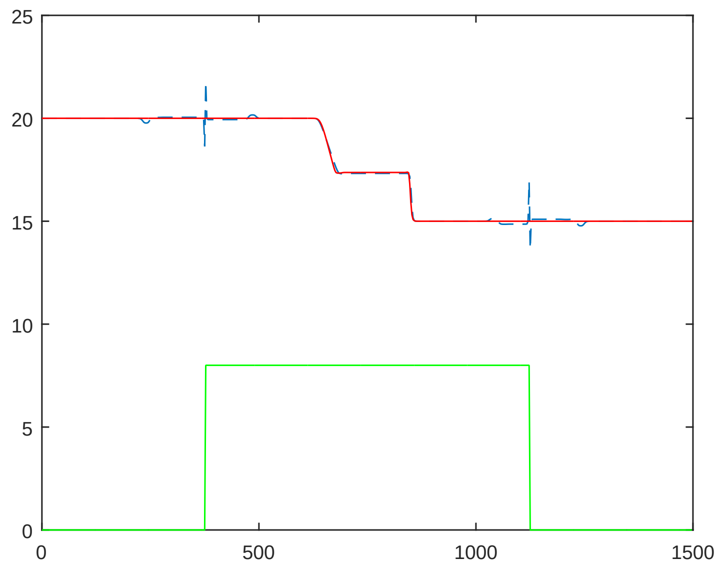


Figure 2.2: Dam break problem: water level obtained at time t=10 using the well-balanced scheme with (dashed line) and without (solid line) SG.

2.4.4 Parabolic bowl

In this test case we consider a parabolic bottom topography and initial water height functions defined as: $z(x) = h_0(\frac{x}{a})^2$ and $h(x,0) = h_0 - \frac{B^2}{2g} - \frac{Bx}{2a}\sqrt{\frac{8h_0}{g}} - z(x)$. $h_0 = 10, a = 3000, B = 5$ are constants. The computational domain is the interval [-5000,5000] and is discretized using 200 grid points. The water is located between the two points $x_1 = -a - \frac{B\omega a^2}{2gh_0}\cos(\omega t)$ and $x_2 = a - \frac{B\omega a^2}{2gh_0}\cos(\omega t)$ where $\omega = \frac{\sqrt{2gh_0}}{a}$. Results are reported in figure 2.4 where the computed water height (green circles) is in perfect match with the analytic water height function (red line) of equation:

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

2.4.5 V-shape

we consider the V-shape problem mentioned in [24] where the water is released from the left part of the V-shape bottom topography.

The water bed is given by the following formula:

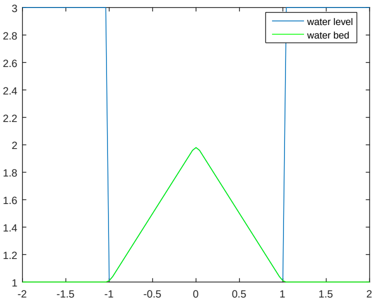
$$z(x) = \frac{1}{\sqrt{3}}|x-1|$$

The water has initially a parabolic profile together with a zero velocity:

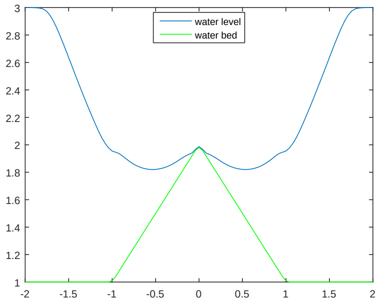
$$h(x,0) = \max(0, -1.5(x-0.3)(x-0.7))$$

$$v(x,0) = 0$$

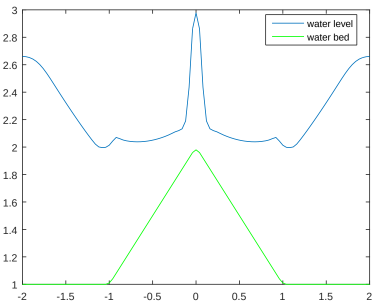
the computational domain is the interval [0,2] which is partitioned using 400 grid points. The results are reported in the figure 2.5 where we see the water propagates across the domain and reaches the right side of the bottom at the final time t=0.7.



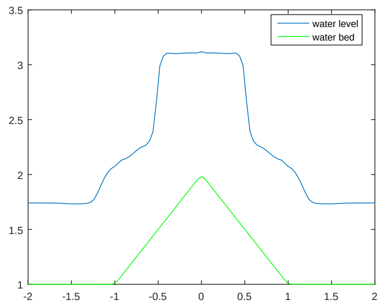
(a) water height at t=0



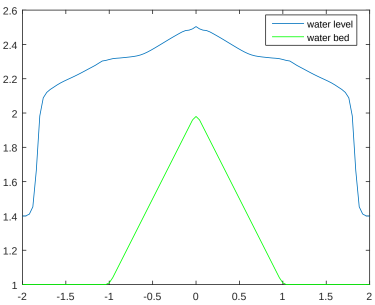
(b) water height at t=0.15



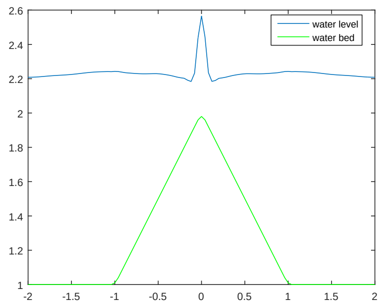
(c) water height at t=0.25



(d) water height at t=0.49

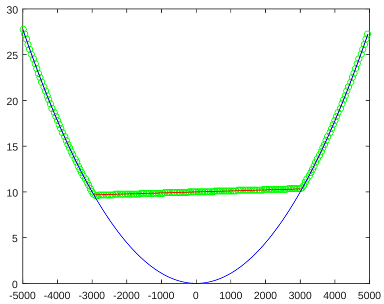


(e) water height at t=0.82

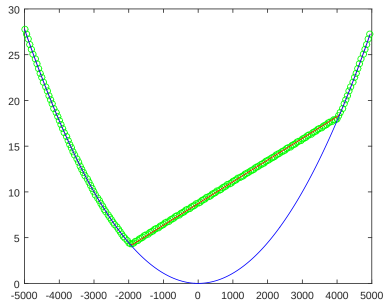


(f) water height at t=1.5

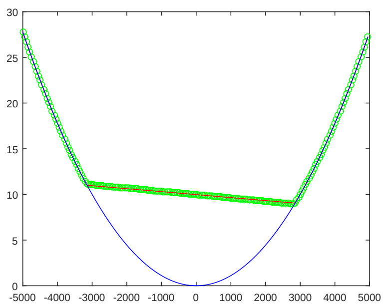
Figure 2.3: Two symmetric dam break problems over triangular bump



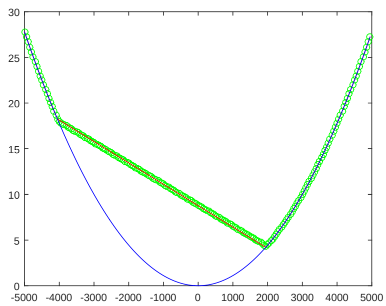
(a) water height at t=1000



(b) water height at t=2000



(c) water height at t=3000



(d) water height at t=4000

Figure 2.4: Water height over parabolic bottom topography

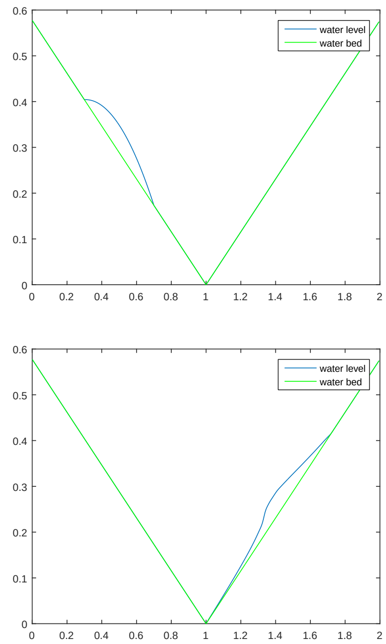


Figure 2.5: Water profile at t=0(up) and t=0.7(down)

Chapter 3

Two-dimensional Well-balanced Central Scheme

3.1 Well-balanced Central Scheme for the Two-dimensional Shallow Water Equations System

The one-dimensional scheme we developed in chapter 2 will be extended to the two-dimensional case. The SWE system is given by:

$$\begin{cases} \partial_t U + \partial_x F(U) + \partial_y G(U) = S(U, x, y), \ (x, y) \in \Omega \subset \mathbb{R} \times \mathbb{R}, \ t > 0 \\ U(x, 0) = U_0(x) \end{cases}$$

we will apply the central scheme mentioned in [7] to this system

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} = \begin{pmatrix} 0 \\ -gh\frac{\partial z}{\partial x} \\ -gh\frac{\partial z}{\partial y} \end{pmatrix}$$

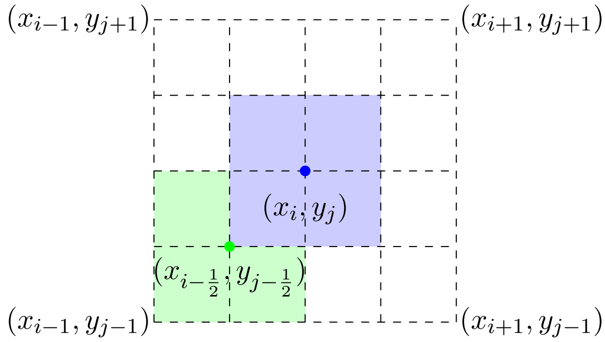


Figure 3.1: $C_{i,j}$(blue cell) and $D_{i-\frac{1}{2},j-\frac{1}{2}}$(green cell)

Where t is the time variable, x and y are the two spatial variables, (u,v) are the velocities in the x-direction and y-direction respectively, h is the water height function $h(x, y, t)$, z is the water bed function $z(x, y)$ and g is the gravitational constant.

We divide the computational domain Ω into cells $C_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{i-1/2}, y_{i+1/2}]$ centered at (x_i, y_j) We also define the staggered cells

$D_{i+\frac{1}{2},j+\frac{1}{2}} = [x_i, x_{i+1}] \times [y_i, y_{i+1}]$ centered at $(x_{i+1/2}, y_{i+1/2})$ where $x_{i+1/2} = x_i + \frac{\Delta x}{2}$ and $y_{i+1/2} = y_i + \frac{\Delta y}{2}$.

We assume that the solution $U^n_{i,j}$ at time t^n is known at the nodes (x_i, y_j). We want to compute the solution $U^{n+1}_{i,j}$ at time $t^{n+1} = t^n + \Delta t$. The solution will be obtained first at the staggered cells and projected back to the original cell to avoid Riemann problems arising at the cell interfaces.

We integrate $\partial_t U + \partial_x F(U) + \partial_y G(U) = S(U, x, y)$ over the domain $R^n_{i+\frac{1}{2},j+\frac{1}{2}} = D_{i+\frac{1}{2},j+\frac{1}{2}} \times [t^n, t^{n+1}]$

$$\iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} [\partial_t U + \partial_x F(U) + \partial_y G(U)]dR = \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U, x, y)dR.$$

$$\iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} \partial_t U + \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} \partial_x F(U) + \partial_y G(U)]dR = \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U, x, y)dR.$$

The integral $\iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} \partial_t U$ can be reduced to:

$\int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} (U^{n+1} - U^n) dxdy$

Next we apply the Mean-Value Theorem for integrals, taking into account that $U(x, y, t)$ is assumed to be a piecewise linear function defined at the cell centers:

$\int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} U^{n+1} dxdy \approx \Delta x \Delta y U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$.

$\int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} U^{n} dxdy \approx \Delta x \Delta y U^{n}_{i+\frac{1}{2},j+\frac{1}{2}}$.

Then we have:

$$\begin{aligned}\Delta x \Delta y U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} &= \Delta x \Delta y U^{n}_{i+\frac{1}{2},j+\frac{1}{2}} \\ &- \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} \partial_x F(U) + \partial_y G(U)] dR \\ &+ \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U, x, y) dR\end{aligned} \tag{3.1}$$

For the flux integrals we apply the divergence theorem that changes the volume integral into surface integral.Equation (3.1) becomes then:

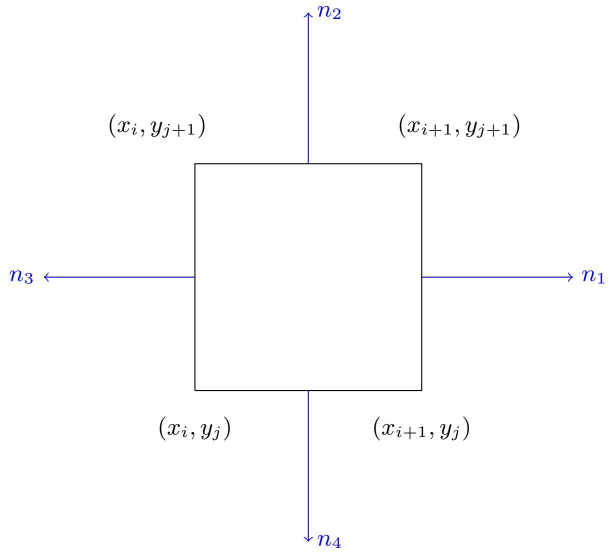
$$\begin{aligned}\Delta x \Delta y U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \Delta x \Delta y U^{n}_{i+\frac{1}{2},j+\frac{1}{2}} - \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} F(U).n_x dAdt \\ - \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} G(U).n_y dAdt + \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U, x, y) dR\end{aligned} \tag{3.2}$$

where $R_{xy} = [x_i, x_{i+1}] \times [y_i, y_{i+1}]$ and n_x, n_y are the unit normal vectors to ∂R_{xy}(the boundary of R_{xy}).

Dividing both sides of equation (3.2) by $\Delta x \Delta y$ we get:

$$\begin{aligned}U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = U^{n}_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} F(U).n_x dxdydt \\ - \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} G(U).n_y dxdydt + \frac{1}{\Delta x \Delta y} \iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U, x, y) dR\end{aligned} \tag{3.3}$$

Next, we approximate the integrals $I = \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} F(U).n_x dxdydt$ and $J = \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} G(U).n_y dxdydt$



$$\begin{aligned} I &= \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} F(U).n_x dxdydt \\ &= \int_{t^n}^{t^{n+1}} \int_{y_j}^{y_{j+1}} F(U(x_{i+1}, y, t)).(1,0,0)dy \\ &+ \int_{t^n}^{t^{n+1}} \int_{x_{i+1}}^{x_i} F(U(x, y_{j+1}, t)).(0,1,0)dx \\ &+ \int_{t^n}^{t^{n+1}} \int_{y_{j+1}}^{y_j} F(U(x_i, y, t)).(-1,0,0)dy \\ &+ \int_{t^n}^{t^{n+1}} \int_{x_i}^{x_{i+1}} F(U(x, y_j, t)).(0,-1,0)dx \end{aligned}$$

An approximation of each integral using the midpoint rule leads to:

$I = \Delta t \Delta y [\frac{F(U(x_{i+1}, y_j, t^{n+\frac{1}{2}})) + F(U(x_{i+1}, y_{j+1}, t^{n+\frac{1}{2}}))}{2}] - \Delta t \Delta y [\frac{F(U(x_i, y_j, t^{n+\frac{1}{2}})) + F(U(x_i, y_{j+1}, t^{n+\frac{1}{2}}))}{2}]$

$I = \Delta t \Delta y [\frac{F(U_{i+1,j}^{n+\frac{1}{2}}) + F(U_{i+1,j+1}^{n+\frac{1}{2}}) - F(U_{i,j}^{n+\frac{1}{2}}) - F(U_{i,j+1}^{n+\frac{1}{2}})}{2}]$

Similar approximation for J implies:

$J = \int_{t^n}^{t^{n+1}} \int_{\partial R_{xy}} G(U).n_y dxdydt$

$J = \Delta t \Delta x [\frac{G(U_{i,j+1}^{n+\frac{1}{2}}) + G(U_{i+1,j+1}^{n+\frac{1}{2}}) - G(U_{i,j}^{n+\frac{1}{2}}) - G(U_{i+1,j}^{n+\frac{1}{2}})}{2}]$

Hence equation (3.2) becomes:

$$
\begin{aligned}
U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} &= U^{n}_{i+\frac{1}{2},j+\frac{1}{2}} \\
&- \frac{\Delta t}{2\Delta x}[F(U^{n+\frac{1}{2}}_{i+1,j}) + F(U^{n+\frac{1}{2}}_{i+1,j+1}) - F(U^{n+\frac{1}{2}}_{i,j}) - F(U^{n+\frac{1}{2}}_{i,j+1})] \\
&- \frac{\Delta t}{2\Delta y}[G(U^{n+\frac{1}{2}}_{i,j+1}) + G(U^{n+\frac{1}{2}}_{i+1,j+1}) - G(U^{n+\frac{1}{2}}_{i,j}) - G(U^{n+\frac{1}{2}}_{i+1,j})] \\
&+ \frac{1}{\Delta x \Delta y}\iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U,x,y)dR
\end{aligned}
\tag{3.4}
$$

where $U^{n+\frac{1}{2}}_{i,j}$ are the predicted values at the time $t^{n+\frac{1}{2}}$ and they are approximated using Taylor's expansion:

$$
U^{n+\frac{1}{2}}_{i,j} = U^{n}_{i,j} + \frac{\Delta t}{2}(U_t)^n_{i,j} \tag{3.5}
$$

now substitute $\partial_t U = -\partial_x F(U) - \partial_y G(U) + S(U,x,y)$ in Eq.(3.5):

$U^{n+\frac{1}{2}}_{i,j} = U^{n}_{i,j} + \frac{\Delta t}{2}[-\partial_x F(U_{i,j}) - \partial y G(U_{i,j}) + S(U_{i,j}, x_i, y_j)]$

then we calculate the flux partial derivatives using the chain rule as follows:

- $\partial_x F(U^n_{i,j}) = (\frac{\partial F}{\partial U})^n_{i,j}.\frac{\partial \delta^n_{i,j}}{\partial x}$ with $\frac{\delta^n_{i,j}}{\Delta x}$ approximates the partial derivative of U with respect to x.
- $\partial_x G(U^n_{i,j}) = (\frac{\partial G}{\partial U})^n_{i,j}.\frac{\partial \sigma^n_{i,j}}{\partial y}$ with $\frac{\sigma^n_{i,j}}{\Delta y}$ approximates the partial derivative of U with respect to y.

Eq.(3.5) becomes:

$U^{n+\frac{1}{2}}_{i,j} = U^{n}_{i,j} + \frac{\Delta t}{2}[-\frac{F'_{i,j}}{\Delta x} - \frac{G'_{i,j}}{\Delta y} + S^n_{i,j}]$ with $F'_{i,j} = \frac{\partial F}{\partial U}.\frac{\delta_{i,j}}{\Delta x}$ and $G'_{i,j} = \frac{\partial G}{\partial U}.\frac{\sigma_{i,j}}{\Delta y}$.

The integral of the source term is being discretized using the midpoint quadrature rule with respect to time and space:

$\iiint_{R_{i+\frac{1}{2},j+\frac{1}{2}}} S(U,x,y)dR \approx \Delta t \Delta x \Delta y S(U^{n+\frac{1}{2}}_{i,j}, U^{n+\frac{1}{2}}_{i+1,j}, U^{n+\frac{1}{2}}_{i,j+1}, U^{n+\frac{1}{2}}_{i+1,j+1})$

with $S(U^{n+\frac{1}{2}}_{i,j}, U^{n+\frac{1}{2}}_{i+1,j}, U^{n+\frac{1}{2}}_{i,j+1}, U^{n+\frac{1}{2}}_{i+1,j+1}) =$

$$
\frac{1}{2}\begin{pmatrix} 0 \\ -g\frac{h^{n+\frac{1}{2}}_{i+1,j}+h^{n+\frac{1}{2}}_{i,j}}{2}(\frac{z_{i+1,j}-z_{i,j}}{\Delta x}) - g\frac{h^{n+\frac{1}{2}}_{i+1,j+1}+h^{n+\frac{1}{2}}_{i,j+1}}{2}(\frac{z_{i+1,j+1}-z_{i,j+1}}{\Delta x}) \\ -g\frac{h^{n+\frac{1}{2}}_{i,j+1}+h^{n+\frac{1}{2}}_{i,j}}{2}(\frac{z_{i,j+1}-z_{i,j}}{\Delta y}) - g\frac{h^{n+\frac{1}{2}}_{i+1,j+1}+h^{n+\frac{1}{2}}_{i+1,j}}{2}(\frac{z_{i+1,j+1}-z_{i+1,j}}{\Delta y}) \end{pmatrix}
$$

Eq.(3.4) becomes:

$$
\begin{aligned}
U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} &= U^{n}_{i+\frac{1}{2},j+\frac{1}{2}} \\
&- \frac{\Delta t}{2\Delta x}[F(U^{n+\frac{1}{2}}_{i+1,j}) + F(U^{n+\frac{1}{2}}_{i+1,j+1}) - F(U^{n+\frac{1}{2}}_{i,j}) - F(U^{n+\frac{1}{2}}_{i,j+1})] \\
&- \frac{\Delta t}{2\Delta y}[G(U^{n+\frac{1}{2}}_{i,j+1}) + G(U^{n+\frac{1}{2}}_{i+1,j+1}) - G(U^{n+\frac{1}{2}}_{i,j}) - G(U^{n+\frac{1}{2}}_{i+1,j})] \\
&+ \Delta t.S(U^{n+\frac{1}{2}}_{i,j}, U^{n+\frac{1}{2}}_{i+1,j}, U^{n+\frac{1}{2}}_{i,j+1}, U^{n+\frac{1}{2}}_{i+1,j+1})
\end{aligned}
\tag{3.6}
$$

Discritization of the source term:

As in the one-dimensional case we define the two sensor functions s_i and t_j as follows:

$$
s_i = \begin{cases} -1 & h_x = \theta \frac{h^n_{i,j} - h^n_{i-1,j}}{\Delta x} \\ 1 & h_x = \theta \frac{h^n_{i+1,j} - h^n_{i,j}}{\Delta x} \\ 0 & h_x = 0 \\ 2 & h_x = \frac{h^n_{i+1,j} - h^n_{i-1,j}}{2\Delta x} \end{cases}
\qquad
t_j = \begin{cases} -1 & h_y = \theta \frac{h^n_{i,j} - h^n_{i,j-1}}{\Delta y} \\ 1 & h_y = \theta \frac{h^n_{i,j+1} - h^n_{i,j}}{\Delta y} \\ 0 & h_y = 0 \\ 2 & h_y = \frac{h^n_{i,j+1} - h^n_{i,j-1}}{2\Delta y} \end{cases}
$$

These parameters will force the discretization of $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ in the source term to follow the discretization of $\frac{\partial h}{\partial x}$ and $\frac{\partial h}{\partial y}$ respectively.$(1 \leq \theta \leq 2)$.

Then the source term is given by:

$$
S^n_{i,j} = \begin{pmatrix} 0 \\ -gh^n_{i,j}\frac{\partial z}{\partial x} \\ -gh^n_{i,j}\frac{\partial z}{\partial y} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix}
$$

$S_1 = 0$

$S_2 = S^n_{2,L} + S^n_{2,R} + S^n_{2,C}$

where

$$
S_{2,L} = s_i^2 \frac{1 - s_i}{6}(2 - s_i)\left(-gh^n_{i,j}\theta \frac{z_{i,j} - z_{i-1,j}}{\Delta x}\right)
$$

$$
S_{2,R} = s_i^2 \frac{1 + s_i}{2}(2 - s_i)\left(-gh^n_{i,j}\theta \frac{z_{i+1,j} - z_{i,j}}{\Delta x}\right)
$$

$$
S_{2,C} = s_i \frac{(s_i + 1)(s_i - 1)}{6}(2 - s_i)\left(-gh^n_{i,j} \frac{z_{i+1,j} - z_{i-1,j}}{2\Delta x}\right)
$$

Hence,

$$S_2 = \begin{cases} -gh^n_{i,j}\theta\frac{z_{i,j}-z_{i-1,j}}{\Delta x} & s_i = -1 \\ -gh^n_{i,j}\theta\frac{z_{i+1,j}-z_{i,j}}{\Delta x} & s_i = 1 \\ 0 & s_i = 0 \\ -gh^n_{i,j}\theta\frac{z_{i+1,j}-z_{i-1,j}}{2\Delta x} & s_i = 2 \end{cases}$$

and $S_3 = S^n_{3,L} + S^n_{3,R} + S^n_{3,C}$

$$S_{3,L} = t_j^2\frac{1-t_j}{6}(2-t_j)\left(-gh^n_{i,j}\theta\frac{z_{i,j}-z_{i,j-1}}{\Delta y}\right)$$

$$S_{3,R} = t_j^2\frac{1+t_j}{2}(2-t_j)\left(-gh^n_{i,j}\theta\frac{z_{i,j+1}-z_{i,j}}{\Delta y}\right)$$

$$S_{3,C} = t_j\frac{(t_j+1)(t_j-1)}{6}(2-t_j)\left(-gh^n_{i,j}\frac{z_{i,j+1}-z_{i,j-1}}{2\Delta y}\right)$$

Hence,

$$S_3 = \begin{cases} -gh^n_{i,j}\theta\frac{z_{i,j}-z_{i,j-1}}{\Delta y} & t_j = -1 \\ -gh^n_{i,j}\theta\frac{z_{i,j+1}-z_{i,j}}{\Delta y} & t_j = 1 \\ 0 & t_j = 0 \\ -gh^n_{i,j}\theta\frac{z_{i,j+1}-z_{i,j-1}}{2\Delta y} & t_j = 2 \end{cases}$$

Now, the forward projection step $U^n_{i+\frac{1}{2},j+\frac{1}{2}}$ is being calculated using linear interpolations in 2D:

Notice that the linear approximation in 2D is given by the following equation:

$f(x,y) = f(a,b) + \frac{\partial f}{\partial x}_{(a,b)}.(x-a) + \frac{\partial f}{\partial y}_{(a,b)}.(y-b)$

So, for every $(x,y) \in C_{i,j}$

$$U_{C_{i,j}}(x,y) = U^n_{i,j} + (x-x_i)\frac{\delta_{i,j}}{\Delta x} + (y-y_j)\frac{\sigma_{i,j}}{\Delta y} \tag{3.7}$$

Using Linear interpolations the forward projection step is defined as follows:

$$\begin{aligned} U^n_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{4}U_{C_{i,j}}(x_i + \frac{1}{4}\Delta x, y_j + \frac{1}{4}\Delta y, t^n) \\ &+ \frac{1}{4}U_{C_{i+1,j}}(x_{i+1} - \frac{1}{4}\Delta x, y_j + \frac{1}{4}\Delta y, t^n) \\ &+ \frac{1}{4}U_{C_{i,j+1}}(x_i + \frac{1}{4}\Delta x, y_{j+1} - \frac{1}{4}\Delta y, t^n) \\ &+ \frac{1}{4}U_{C_{i+1,j+1}}(x_{i+1} - \frac{1}{4}\Delta x, y_j - \frac{1}{4}\Delta y, t^n) \end{aligned} \tag{3.8}$$

Using Eq.(3.7), we get:

$$\begin{aligned} U_{C_{i,j}}(x_i + \frac{1}{4}\Delta x, y_j + \frac{1}{4}\Delta y, t^n) =& U^n_{C_{i,j}}(x_i + \frac{1}{4}\Delta x, y_j + \frac{1}{4}\Delta y) \\ &= U^n_{i,j} + (x_i + \frac{1}{4}\Delta x - x_i)\frac{\delta_{i,j}}{\Delta x} + (y_j + \frac{1}{4}\Delta y - y_j)\frac{\sigma i, j}{\Delta y} \\ &= U^n_{i,j} + \frac{1}{4}\delta_{i,j} + \frac{1}{4}\sigma_{i,j} \end{aligned} \tag{3.9}$$

In the same way we calculate,

$U_{C_{i+1,j}}(x_{i+1} - \frac{1}{4}\Delta x, y_j + \frac{1}{4}\Delta y, t^n) = U^n_{i+1,j} - \frac{1}{4}\delta_{i,j} + \frac{1}{4}\sigma_{i+1,j}$.

$U_{C_{i,j+1}}(x_i + \frac{1}{4}\Delta x, y_{j+1} - \frac{1}{4}\Delta y, t^n) = U^n_{i,j+1} + \frac{1}{4}\delta_{i,j+1} - \frac{1}{4}\sigma_{i,j+1}$.

$U_{C_{i+1,j+1}}(x_{i+1} - \frac{1}{4}\Delta x, y_{j+1} - \beta\Delta y, t^n) = U^n_{i+1,j+1} - \frac{1}{4}\delta_{i+1,j+1} - \frac{1}{4}\sigma_{i+1,j+1}$.

Replacing and rearranging the previous equations in Eq.(3.8) we get,

$$\begin{aligned} U^n_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{4}(U^n_{i,j} + U^n_{i+1,j} + U^n_{i,j+1} + U^n_{i+1,j+1}) \\ &+ \frac{1}{16}(\delta_{i,j} + \delta_{i,j+1} - \delta_{i+1,j} - \delta_{i+1,j+1}) \\ &+ \frac{1}{16}(\sigma_{i,j} - \sigma_{i,j+1} + \sigma i + 1, j - \sigma_{i+1j+1}) \end{aligned} \tag{3.10}$$

Similarly, we derive the backward projection step.

Therefore, the numerical scheme can be summarized by the following steps:

- Start with $U^n_{i,j}$.
- Find the predicted values as time $t^{n+\frac{1}{2}}$ using the derived formula:

$$U^{n+\frac{1}{2}}_{i,j} = U^n_{i,j} + \frac{\Delta t}{2}[-\frac{F'_{i,j}}{\Delta x} - \frac{G'_{i,j}}{\Delta y} + S^n_{i,j}] \tag{3.11}$$

- Find the forward projection step:

$$\begin{aligned} U^n_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{4}(U^n_{i,j} + U^n_{i+1,j} + U^n_{i,j+1} + U^n_{i+1,j+1}) \\ &+ \frac{1}{16}(\delta_{i,j} + \delta_{i,j+1} - \delta_{i+1,j} - \delta_{i+1,j+1}) \\ &+ \frac{1}{16}(\sigma_{i,j} - \sigma_{i,j+1} + \sigma_{i+1,j} - \sigma_{i+1,j+1}) \end{aligned} \tag{3.12}$$

- Find the solution at time n+1:

$$\begin{aligned} U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} &= U^n_{i+\frac{1}{2},j+\frac{1}{2}} \\ &- \frac{\Delta t}{2\Delta x}[F(U^{n+\frac{1}{2}}_{i+1,j}) + F(U^{n+\frac{1}{2}}_{i+1,j+1}) - F(U^{n+\frac{1}{2}}_{i,j}) - F(U^{n+\frac{1}{2}}_{i,j+1})] \\ &- \frac{\Delta t}{2\Delta x}[G(U^{n+\frac{1}{2}}_{i,j+1}) + G(U^{n+\frac{1}{2}}_{i+1,j+1}) - G(U^{n+\frac{1}{2}}_{i,j}) - G(U^{n+\frac{1}{2}}_{i+1,j})] \\ &+ \Delta t.S(U^{n+\frac{1}{2}}_{i,j}, U^{n+\frac{1}{2}}_{i+1,j}, U^{n+\frac{1}{2}}_{i,j+1}, U^{n+\frac{1}{2}}_{i+1,j+1}) \end{aligned} \tag{3.13}$$

- Find the backward projection step using the following equation:

$$\begin{aligned} U^{n+1}_{i,j} &= \frac{1}{4}(U^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} + U^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} + U^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} + U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}) \\ &+ \frac{1}{16}(\delta_{i-\frac{1}{2},j-\frac{1}{2}} + \delta_{i-\frac{1}{2},j+\frac{1}{2}} - \delta_{i+\frac{1}{2},j-\frac{1}{2}} - \delta_{i+\frac{1}{2},j+\frac{1}{2}}) \\ &+ \frac{1}{16}(\sigma_{i-\frac{1}{2},j-\frac{1}{2}} - \sigma_{i-\frac{1}{2},j+\frac{1}{2}} + \sigma_{i+\frac{1}{2},j-\frac{1}{2}} - \sigma_{i+\frac{1}{2},j+\frac{1}{2}}) \end{aligned} \tag{3.14}$$

Notice that the way we're discretizing the source gives the following two relations:

- $U^{n+\frac{1}{2}}_{i,j} = U^n_{i,j}$
- $U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = U^n_{i+\frac{1}{2},j+\frac{1}{2}}$

see proof in [6]. But it is not the case on the original cells i.e $U^{n+1}_{i,j} \neq U^n_{i,j}$.

As in 1D, the numerical scheme fails to satisfy the lake at rest constrain unless we apply the surface gradient method which will be described in the following section.

3.2 The Surface Gradient Method for the Two-dimensional Shallow Water Equations

As mentioned above, the well-balanced 2D scheme does not satisfy the steady state requirement i.e$(h + z = c, u = 0, v = 0)$ that was initially imposed. So, as we did in 1D, the surface gradient method described in [6] will be applied for the two-dimensional well-balanced scheme. The first step in the surface gradient method is to define the bottom topography function on the staggered cells and not on the original ones. Then, we find its values on the centers of the original cells using linear interpolations:

$$z_{i,j} = \frac{z_{i-\frac{1}{2},j-\frac{1}{2}} + z_{i-\frac{1}{2},j+\frac{1}{2}} + z_{i+\frac{1}{2},j-\frac{1}{2}} + z_{i+\frac{1}{2},j+\frac{1}{2}}}{4}$$

Using the 2D unstaggered central scheme, the water height h and the bottom topography function are considered to be linear on each original cell $C_{i,j}$. So, we first linearize the water level function $H(x, y)$ and the water height function using the relation $h(x, y) = H(x, y) - z(x, y)$.

Then, the partial derivatives of the water height h in the forward and the backward projection steps will be calculated from the partial derivatives of the water level function $H(x, y)$. Now,

$$h^n_{i,j} = H^n_{i,j} - z_{i,j} \qquad \text{for all} \quad i, j$$

Then,

$$h_x|^n_{i,j} = H_x|^n_{i,j} - z_x|_{i,j} \qquad\qquad h_y|^n_{i,j} = H_y|^n_{i,j} - z_y|_{i,j}$$

where H_x and H_y being calculated using MC-Θ limiter and z_x , z_y are obtained using central differences.

Hence,

$$h_x|^n_{i,j} = H_x|^n_{i,j} - \frac{1}{\Delta x}\left(\frac{z_{i,j} + z_{i+1,j}}{2} - \frac{z_{i-1,j} + z_{i,j}}{2}\right) \tag{3.15}$$

$$h_y|^n_{i,j} = H_y|^n_{i,j} - \frac{1}{\Delta y}\left(\frac{z_{i,j} + z_{i,j+1}}{2} - \frac{z_{i,j-1} + z_{i,j}}{2}\right) \tag{3.16}$$

Substituting these two equations in the first component of the forward projection step (Eq.(3.10)) i.e $h_{i+\frac{1}{2},j+\frac{1}{2}}$, we get:

$$
\begin{aligned}
h^n_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{4}(h^n_{i,j} + h^n_{i+1,j} + h^n_{i,j+1} + h^n_{i+1,j+1}) \\
&+ \frac{\Delta x}{16}\left[H_x|^n_{i,j} - \frac{\frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta x} \right] \\
&+ \frac{\Delta x}{16}\left[H_x|^n_{i,j+1} - \frac{\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{3}{2}}}{2} - \frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{3}{2}}}{2}}{\Delta x} \right] \\
&- \frac{\Delta x}{16}\left[H_x|^n_{i+1,j} - \frac{\frac{z_{i+\frac{3}{2},j-\frac{1}{2}}+z_{i+\frac{3}{2},j+\frac{1}{2}}}{2} - \frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta x} \right] \\
&- \frac{\Delta x}{16}\left[H_x|^n_{i+1,j+1} - \frac{\frac{z_{i+\frac{3}{2},j+\frac{1}{2}}+z_{i+\frac{3}{2},j+\frac{3}{2}}}{2} - \frac{z_{i+\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{3}{2}}}{2}}{\Delta x} \right] \\
&+ \frac{\Delta y}{16}\left[H_y|^n_{i,j} - \frac{\frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j-\frac{1}{2}}}{2}}{\Delta y} \right] \\
&+ \frac{\Delta y}{16}\left[H_y|^n_{i+1,j} - \frac{\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{3}{2},j+\frac{1}{2}}}{2} - \frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{3}{2},j-\frac{1}{2}}}{2}}{\Delta y} \right] \\
&- \frac{\Delta y}{16}\left[H_y|^n_{i,j+1} - \frac{\frac{z_{i-\frac{1}{2},j+\frac{3}{2}}+z_{i+\frac{1}{2},j+\frac{3}{2}}}{2} - \frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta y} \right] \\
&- \frac{\Delta y}{16}\left[H_y|^n_{i+1,j+1} - \frac{\frac{z_{i+\frac{1}{2},j+\frac{3}{2}}+z_{i+\frac{3}{2},j+\frac{3}{2}}}{2} - \frac{z_{i+\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{3}{2},j+\frac{1}{2}}}{2}}{\Delta y} \right]
\end{aligned}
\tag{3.17}
$$

A similar reformulation is to be applied to the first component of the backward projection step $h^{n+1}_{i,j}$.

For each (x_i, y_j), we have $H_{i,j} = h_{i,j} + z_{i,j}$ we will redefine the bottom topography function on the staggered nodes because z(x,y) is linear $\forall(x, y) \in C_{i,j}$ but not in $D_{i+\frac{1}{2},j+\frac{1}{2}}$. Hence,

$$
\tilde{z}_{i+\frac{1}{2},j+\frac{1}{2}} = z_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{1}{2}\left(z_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{z_{i,j} + z_{i+1,j} + z_{i,j+1} + z_{i+1,j+1}}{4} \right) \tag{3.18}
$$

The water height function on the staggered cells is then defined as:

$$
h^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \tilde{H}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - \tilde{z}_{i+\frac{1}{2},j+\frac{1}{2}}
$$

The partial derivatives of the water height function will be calculated from that of the water level function.

$$
h_x|_{i+\frac{1}{2},j+\frac{1}{2}} = H_x|_{i+\frac{1}{2},j+\frac{1}{2}} - \left[\frac{\frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta x} \right] \tag{3.19}
$$

$$h_y|_{i+\frac{1}{2},j+\frac{1}{2}} = H_y|_{i+\frac{1}{2},j+\frac{1}{2}} - \left[\frac{\frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j-\frac{1}{2}}}{2}}{\Delta y}\right] \tag{3.20}$$

Finally, we substitute these two formulas in $h_{i,j}^{n+1}$'s formula to get:

$$\begin{aligned}
h_{i,j}^{n+1} &= \frac{1}{4}(h^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} + h^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} + h^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} + h^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}) \\
&+ \frac{\Delta x}{16}\left[H_x|^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} - \frac{\frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j-\frac{1}{2}}}{2} - \frac{z_{i-\frac{3}{2},j-\frac{1}{2}}+z_{i-\frac{1}{2},j-\frac{1}{2}}}{2}}{\Delta x}\right] \\
&+ \frac{\Delta x}{16}\left[H_x|^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} - \frac{\frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{3}{2},j+\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta x}\right] \\
&- \frac{\Delta x}{16}\left[H_x|^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} - \frac{\frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{3}{2},j-\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j-\frac{1}{2}}}{2}}{\Delta x}\right] \\
&- \frac{\Delta x}{16}\left[H_x|^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{3}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta x}\right] \\
&+ \frac{\Delta y}{16}\left[H_y|^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} - \frac{\frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{3}{2}}+z_{i-\frac{1}{2},j-\frac{1}{2}}}{2}}{\Delta y}\right] \\
&+ \frac{\Delta y}{16}\left[H_y|^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} - \frac{\frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2} - \frac{z_{i+\frac{1}{2},j-\frac{3}{2}}+z_{i+\frac{1}{2},j-\frac{1}{2}}}{2}}{\Delta y}\right] \\
&- \frac{\Delta y}{16}\left[H_y|^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} - \frac{\frac{z_{i-\frac{1}{2},j+\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{3}{2}}}{2} - \frac{z_{i-\frac{1}{2},j-\frac{1}{2}}+z_{i-\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta y}\right] \\
&- \frac{\Delta y}{16}\left[H_y|^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{3}{2}}}{2} - \frac{z_{i+\frac{1}{2},j-\frac{1}{2}}+z_{i+\frac{1}{2},j+\frac{1}{2}}}{2}}{\Delta y}\right]
\end{aligned} \tag{3.21}$$

These reformulations on the forward and the backward projection steps complete the well-balanced numerical scheme and solve the steady state problem. As one can look for the proof in [6] where the equality $U_{i,j}^{n+1} = U_{i,j}^{n}$ is obtained for all (x_i, y_j) in the computational domain.

3.3 Treatment of Wet and Dry States

As in the one-dimensional case, when the developed well-balanced unstaggered central scheme is considered with wet and dry regions, numerical instabilities may arise. These instabilities appear as negative water heights in the forward and the backward projection steps.

Going back to the way we are calculating the forward and the backward projection steps:

Forward projection step:

$$\begin{aligned} U^n_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{4}(U^n_{i,j} + U^n_{i+1,j} + U^n_{i,j+1} + U^n_{i+1,j+1}) \\ &+ \frac{1}{16}(\delta_{i,j} + \delta_{i,j+1} - \delta_{i+1,j} - \delta_{i+1,j+1}) \\ &+ \frac{1}{16}(\sigma_{i,j} - \sigma_{i,j+1} + \sigma_{i+1,j} - \sigma_{i+1,j+1}) \end{aligned} \tag{3.22}$$

Backward projection step:

$$\begin{aligned} U^{n+1}_{i,j} &= \frac{1}{4}(U^{n+1}_{i-\frac{1}{2},j-\frac{1}{2}} + U^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}} + U^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}} + U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}) \\ &+ \frac{1}{16}(\delta_{i-\frac{1}{2},j-\frac{1}{2}} + \delta_{i-\frac{1}{2},j+\frac{1}{2}} - \delta_{i+\frac{1}{2},j-\frac{1}{2}} - \delta_{i+\frac{1}{2},j+\frac{1}{2}}) \\ &+ \frac{1}{16}(\sigma_{i-\frac{1}{2},j-\frac{1}{2}} - \sigma_{i-\frac{1}{2},j+\frac{1}{2}} + \sigma_{i+\frac{1}{2},j-\frac{1}{2}} - \sigma_{i+\frac{1}{2},j+\frac{1}{2}}) \end{aligned} \tag{3.23}$$

The water height interpolant on the cell $C_{i,j}$ at time t^n is defined as:

$$\eta^n_{i,j} = h^n_{i,j} + (x - x_i)(h_x|^n_i) + (y - y_j)((h_y|^n_j), (x, y) \in C_{i,j}$$

any negative water height in the forward or the backward projection step is due to the steep gradient of the interpolant which performs a negative interpolated water height (i.e below the water bed). Therefore, our aim is to propose a treatment that yields clean forward and backward steps, and also maintain the conservation of water across the computational domain as well as the well-balanced property of the developed scheme. Consider the case where $h^n_{i,j}$ is such that $h^n_{k,l} > 0$ for k$\in \{i, i-1\}$ and l$\in \{j, j-1\}$.

The interpolant $\eta^n_{i,j}$ approximates the water height $h(x, y)$ at time t^n for every couple $(x_i, y_j) \in C_{i,j}$ (the wet cell in our case). For the dry cells we will simply set $\eta^n_{i,j} = 0$.

In this case the forward projection step may execute a negative water height value $h^n_{i+\frac{1}{2}}$ if the interpolant has steep gradient.

for example, when $\eta^n_{i,j}(x)$ is too steep so that $\eta^n_{i+\frac{1}{4},j+\frac{1}{4}} < 0$ and

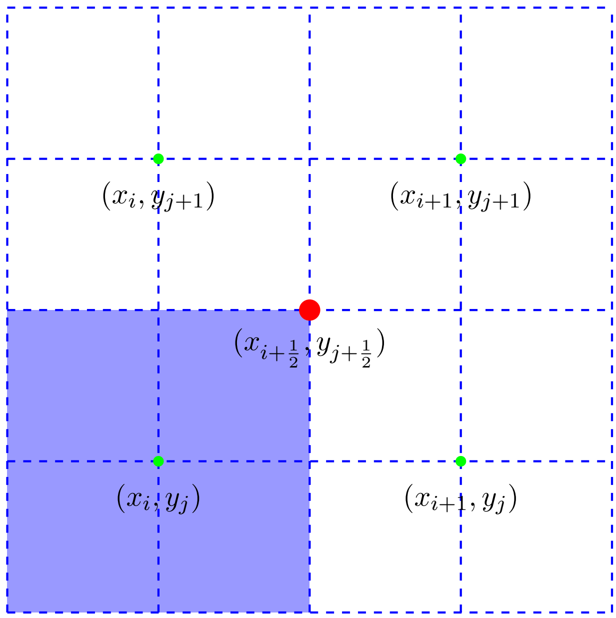


Figure 3.2: The four cells $C_{i,j}, C_{i+1,j}, C_{i,j+1}, C_{i+1,j+1}$ in the case $C_{i,j}$ is a wet cell and the rest are dry.

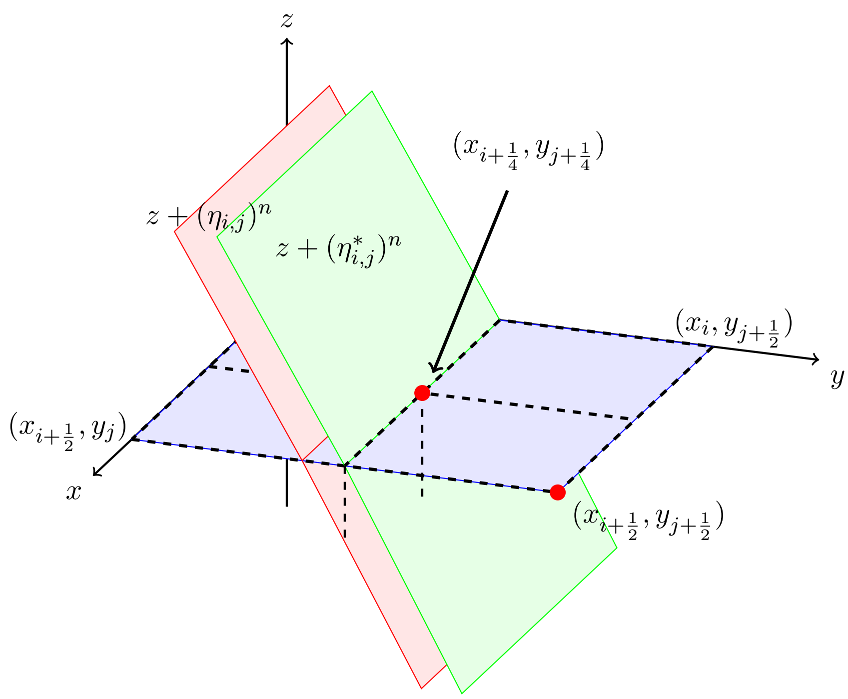


Figure 3.3: The forward projection step leading to negative $h^n_{i+\frac{1}{2}}$ (red plane) and the corrected one leading to $h^n_{i+\frac{1}{2}}$=0 (green plane).

$$\eta^n_{i+\frac{3}{4},j+\frac{1}{4}} < |\eta^n_{i+\frac{1}{4},j+\frac{1}{4}}|$$

$$\eta^n_{i+\frac{1}{4},j+\frac{3}{4}} < |\eta^n_{i+\frac{1}{4},j+\frac{1}{4}}|$$

$$\eta^n_{i+\frac{3}{4},j+\frac{3}{4}} < |\eta^n_{i+\frac{1}{4},j+\frac{1}{4}}|$$

Then,

$$h^n_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}(\eta^n_{i+\frac{1}{4},j+\frac{1}{4}} + \eta^n_{i+\frac{3}{4},j+\frac{1}{4}} + \eta^n_{i+\frac{1}{4},j+\frac{3}{4}} + \eta^n_{i+\frac{3}{4},j+\frac{3}{4}}) < 0$$

The treatment we impose on the well-balanced scheme can be summarized by the following steps:

- If $h_{i+\frac{1}{2},j+\frac{1}{2}} < 0$, we set it to 0.
- Correct the gradient of the water height interpolants in the cells $C_{i,j}, C_{i+1,j}, C_{i,j+1}, C_{i+1,j+1}$, four cases arise:

  1. **If** $h^n_{i,j} \geq \epsilon$ we re-linearize the water height function and update the definition of the sensor functions,

     $$h_x|^n_{i,j} = -\frac{h^n_{i,j}}{\frac{\Delta x}{4}}, \qquad s_i = 3$$

     and

     $$h_y|^n_{i,j} = -\frac{h^n_{i,j}}{\frac{\Delta y}{4}}, \qquad t_j = 3$$

     Otherwise, we set $h^n_{i,j} = 0$ and $h_x|^n_{i,j} = h_y|^n_{i,j} = 0$, $s_i = t_j = 0$

  2. **If** $h^n_{i+1,j} \geq \epsilon$ we re-linearize the water height function and extend the definition of the sensor functions,

     $$h_x|^n_{i+1,j} = \frac{h^n_{i+1,j}}{\frac{\Delta x}{4}}, \qquad s_i = 4$$

and

$$h_y|_{i,j}^n = -\frac{h_{i+1,j}^n}{\frac{\Delta y}{4}}, \qquad t_j = 4$$

Otherwise, we set $h_{i+1,j}^n = 0$ and $h_x|_{i+1,j}^n = h_y|_{i+1,j}^n = 0$, $s_i = t_j = 0$

3. **If** $h_{i,j+1}^n \geq \epsilon$ we re-linearize the water height function and update the definition of the sensor functions,

$$h_x|_{i,j+1}^n = -\frac{h_{i,j+1}^n}{\frac{\Delta x}{4}}, \qquad s_i = 5$$

and

$$h_y|_{i,j+1}^n = \frac{h_{i,j+1}^n}{\frac{\Delta y}{4}}, \qquad t_j = 5$$

Otherwise, we set $h_{i,j+1}^n = 0$ and $h_x|_{i,j+1}^n = h_y|_{i,j+1}^n = 0$, $s_i = t_j = 0$

4. **If** $h_{i+1,j+1}^n \geq \epsilon$ we re-linearize the water height function and update the definition of the sensor functions,

$h_x|_{i+1,j+1}^n = \frac{h_{i+1,j+1}^n}{\frac{\Delta x}{4}}, \qquad s_i = 6$

and

$h_y|_{i+1,j+1}^n = \frac{h_{i+1,j+1}^n}{\frac{\Delta y}{4}}, \qquad t_j = 6$

Otherwise, we set $h_{i+1,j+1}^n = 0$ and $h_x|_{i+1,j+1}^n = h_y|_{i+1,j+1}^n = 0$, $s_i = t_j = 0$

- Update the discretization of the source term accordingly:

$$S_1 = 0$$

$$S_2 = \begin{cases} -gh^n_{i,j}\theta\frac{z_{i,j}-z_{i-1,j}}{\Delta x} & \text{if} \quad s_i = -1 \\ -gh^n_{i,j}\theta\frac{z_{i+1,j}-z_{i,j}}{\Delta x} & \text{if} \quad s_i = 1 \\ 0 & \text{if} \quad s_i = 0 \\ -gh^n_{i,j}\theta\frac{z_{i+1,j}-z_{i-1,j}}{2\Delta x} & \text{if} \quad s_i = 2 \\ -gh^n_{i,j}\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}-z_{i,j}}{2\Delta x} & \text{if} \quad s_i = 3 \\ -gh^n_{i,j}\frac{z_{i,j+1}-z_{i+\frac{1}{2},j+\frac{1}{2}}}{2\Delta x} & \text{if} \quad s_i = 4 \\ -gh^n_{i,j}\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}-z_{i+1,j}}{2\Delta x} & \text{if} \quad s_i = 5 \\ -gh^n_{i,j}\frac{z_{i+1,j+1}-z_{i+\frac{1}{2},j+\frac{1}{2}}}{2\Delta x} & \text{if} \quad s_i = 6 \end{cases}$$

$$S_3 = \begin{cases} -gh^n_{i,j}\theta\frac{z_{i,j}-z_{i,j-1}}{\Delta y} & \text{if} \quad t_j = -1 \\ -gh^n_{i,j}\theta\frac{z_{i,j+1}-z_{i,j}}{\Delta y} & \text{if} \quad t_j = 1 \\ 0 & \text{if} \quad t_j = 0 \\ -gh^n_{i,j}\theta\frac{z_{i,j+1}-z_{i,j-1}}{2\Delta y} & \text{if} \quad t_j = 2 \\ -gh^n_{i,j}\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}-z_{i,j}}{2\Delta y} & \text{if} \quad t_j = 3 \\ -gh^n_{i,j}\frac{z_{i+\frac{1}{2},j+\frac{1}{2}}-z_{i,j+1}}{2\Delta y} & \text{if} \quad t_j = 4 \\ -gh^n_{i,j}\frac{z_{i+1,j}-z_{i+\frac{1}{2},j+\frac{1}{2}}}{2\Delta y} & \text{if} \quad t_j = 5 \\ -gh^n_{i,j}\frac{z_{i+1,j+1}-z_{i+\frac{1}{2},j+\frac{1}{2}}}{2\Delta y} & \text{if} \quad t_j = 6 \end{cases}$$

A similar treatment is to be applied for the backward projection step:

If $h^n_{i,j} < 0$ we set to 0.

The proposed treatment will lead to clean forward and backward projection steps with positive water heights. In addition, it maintains the well-balanced property of the unstaggered numerical scheme by updating the source term as we mentioned above. The scheme will also ensure the conservation of water across the computational domain as we prove in the numerical experiment's section.

Remarks:

1.

$$\Delta t = \min(\Delta t_1, \Delta t_2).$$

- $\Delta t_1 = \text{CFL} * \frac{\Delta x}{\max(\max(\text{LF})}$ where LF is the matrix containing the eigenvalues of

$\frac{\partial F}{\partial U}$.

- $\Delta t_2 = \text{CFL} * \frac{\Delta y}{\max(\max(\text{LG})}$ where LG is the matrix containing the eigenvalues of $\frac{\partial G}{\partial U}$.

- CFL is considered 0.485 in all computations.

2. All simulations have been done using MATLAB.

3.4 Numerical Experiments

In this section we test the designed well-balanced scheme on several problems. The gravitational constant is $g = 9.8m/s^2$.

3.4.1 Dam break over a rectangular bump

In this experiment, we consider a dam break over a rectangular water bed [6] over the computational domain [0,1500]× [0,1500] which is discretized using 600 grid points on the x-axis and 11 on the y-axis. The initial velocity components are zero. The water bed is given by this equation:

$$z(x,y) = \begin{cases} 8 & \text{if}, |x - \frac{1500}{2}| < \frac{1500}{4} \\ \\ 0 & \text{otherwise.} \end{cases}$$

The water level function at $t = 0$,

$$H(x,y,0) = \begin{cases} 20, & \text{if} \quad x < \frac{1500}{2} \\ \\ 15, & \text{otherwise.} \end{cases}$$

After applying the well-balanced scheme together with the surface gradient method,the water is propagating across the domain without any oscillations. Figure 3.4 shows the water profile at the final time.

3.4.2 Circular dam-break problem

In this section, we test our scheme on a circular dam break over flat bottom topography. Our computational domain is [-100,100]× [-100,100]and it is partitioned into 100×100 cells.The dam has a cylindrical profile defined as:

$$h(x,y,0) = \begin{cases} 10, & \text{if} \quad x^2 + y^2 \leq 60^2 \\ \\ 0, & \text{otherwise.} \end{cases}$$

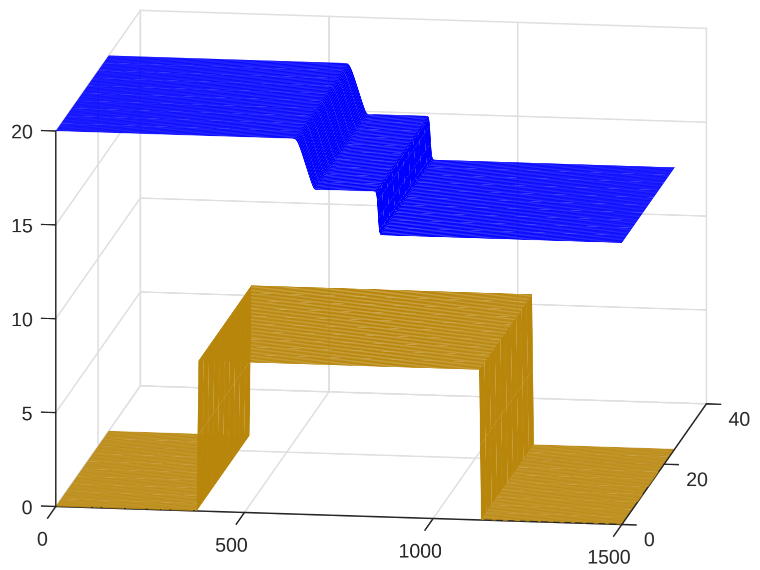


Figure 3.4: Dam break problem: water level obtained at time $t = 10$, using $600{\times}11$ grid points.

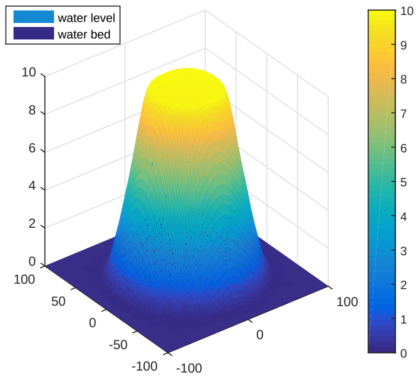
The velocity components are initially zero.

Figure 3.5 presents the water level surface and the contours at the final time $t = 1.75$ where we see a symmetric solution with no oscillations that agrees with the solution presented in [25].

3.4.3 Oscillating lake

We consider the two dimensional oscillating lake tested in [26]. The computational domain is [-2,2]×[-2,2] and the exact water height is given by this formula:

$$h(x, y, t) = \max(0, \frac{\sigma h_0}{a^2}(2x\cos(\omega t) + 2y\sin(\omega t) - \sigma) + h_0 - b)$$



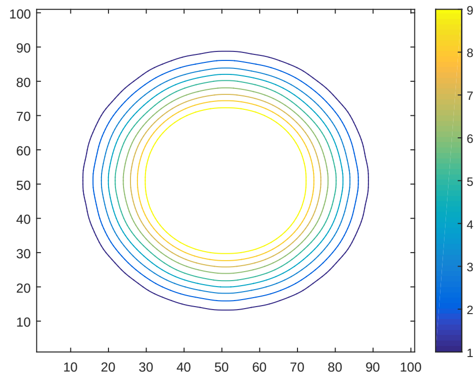


Figure 3.5: Circular Dam-Break: water level surface (left) and contours of the surface (right) at time $t = 1.75$.

and the parabolic topography by this formula,

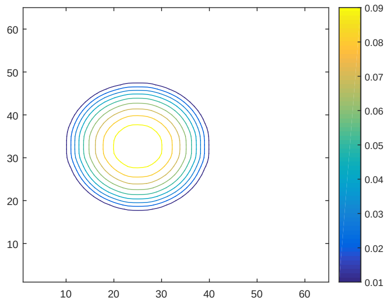
$$z(x, y) = h_0 \frac{x^2 + y^2}{a^2}$$

where ω is the frequency $\omega = \frac{\sqrt{2gh_0}}{a}$, a=1, $\sigma = 0.5$ and $h_0 = 0.1$.

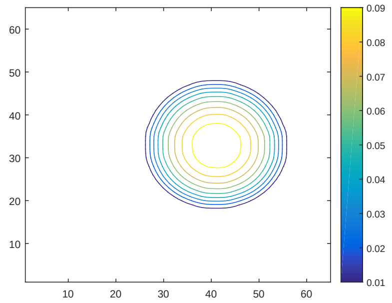
The initial velocities are zero and the initial water height is obtained for t=0. We impose reflecting boundary conditions throughout the boundaries. The solution is computed and compared to the exact solution until the final time T=$\frac{2\pi}{\omega}$.Figure 3.6 illustrates a comparison between contours of the numerical and the analytic solution at two different times where we see a good agreement between them. In figure 3.7, we draw the contours of the water height error between the exact and the numerical solution at time $t = \frac{T}{2}$.

3.4.4 Two symmetric dam break problems

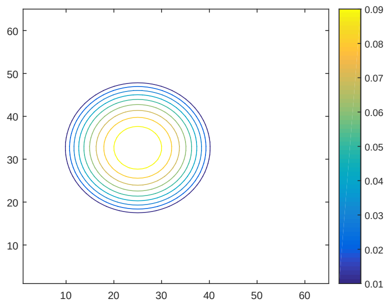
An extension of the one-dimensional experiment tested in [4] is to be tested here. The bottom topography and the initial water height are defined as:



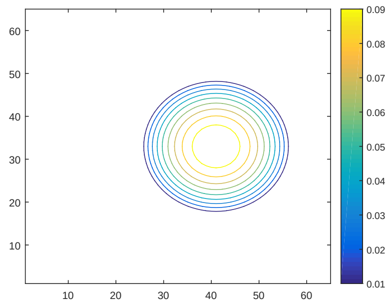
(a) contours of the water level at the time $t = \frac{T}{2}$.



(b) contours of the water level at the final time $t = T$.



(c) contours of the water level at the time $t = \frac{T}{2}$.



(d) contours of the water level at the final time $t = T$.

Figure 3.6: Oscillating lake: numerical solution (up) vs exact solution (down)

$$z(x,y) = \begin{cases} 1 & x < -1 \\ 2+x & -1 \leq x \leq 0 \\ 2-x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

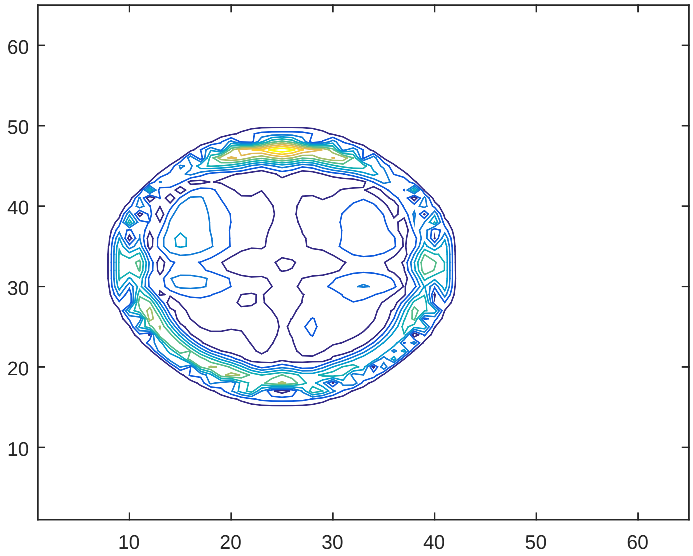


Figure 3.7: Oscillating Lake: contours of the water height error at time $t = \frac{T}{2}$.

$$h(x, y, 0) = \begin{cases} 2 & x < -1 \\ 0 & -1 \leq x \leq 1 \\ 2 & x > 1 \end{cases}$$

the water will flow along the x-axis. The initial velocity is zero. The domain is the rectangle $[-2, 2] \times [-2, 2]$ and is discritized over 100 points. Reflecting boundary conditions are implemented on the two sides of the x-axis. The results are reported in figure 3.8, where the water profile match the one-dimensional experiment. A cross section of the 2D profile is to be compared with a 1D profile in figure 3.9 at time t=1.0179 where we see the perfect match. Moreover, the water is conserved across the computational domain as shown in figure 3.10, where we compute the water volume c(t) and divide it by the initial water volume c_0.The ratio $c(t)/c_0$ is plotted at each time step until the final time $t = 6$ using 50,100,200 grid points.

3.4.5 Dam break over a plane

Again an extension of the one-dimensional experiment mentioned in [4] will be used to test the validity of the numerical scheme. We will consider the dam break over 3 inclined planes. The bottom topography is

$$z(x, y) = x \tan \alpha$$

with $\alpha = 0$, $\frac{\pi}{60}$, and $\frac{-\pi}{60}$ representing the three planes.The computational domain is [-15,15]×[-15,15] discretized using 200 grid points. The initial velocity components are zero and the initial water height is defined as:

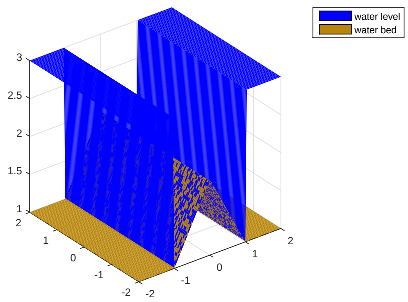
$$\begin{cases} 1 - z(x, y), & \text{if} \quad x \leq 0 \\ 0, & \text{otherwise.} \end{cases}$$

The water level is represented at the final time $t = 2$ in figure 3.11. In figure 3.12 we plot on the same graph the water level and the velocity in the x-direction obtained from a two-dimensional cross section where they are compared to the ones obtained from the one-dimensionl experiment at the final time $t_f = 2$ (left column). On the right column of the figure we compare the front position obtained from a two dimensional cross section to the one obtained form the one-dimensional experiment and also to the exact front position given by the following formula:

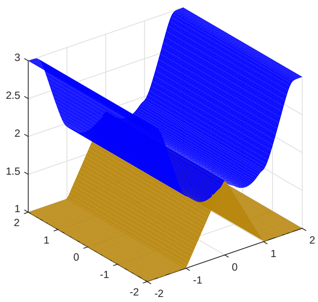
$$x_f(t) = 2t\sqrt{g \cos(\alpha)} - 0.5gt^2 \tan(\alpha)$$

while the numerical front position is the first cell-center where the water level exceeds the value $\epsilon = 10^{-9}$.

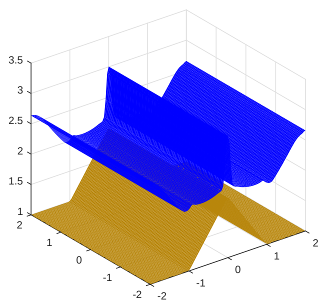
Moreover, we consider a dam break on the y-axis. The result is reported in figure 3.13 that illustrates the perfect agreement between the cross section along the x-axis and the cross section along the y-axis for $\alpha = 0$ at the final time t_f=2.



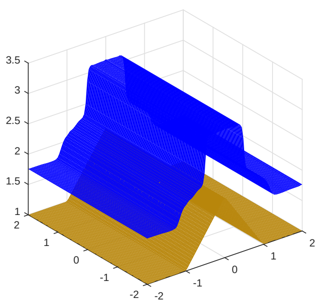
(a) water height at t=0.



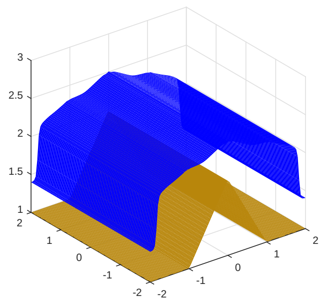
(b) water height at t=0.15.



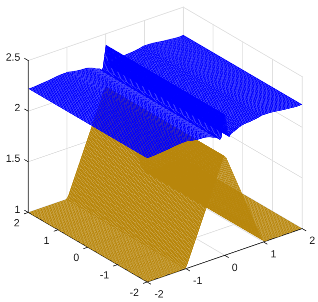
(c) water height at t=0.25.



(d) water height at t=0.49.



(e) water height at t=0.82.



(f) water height at t=1.5.

Figure 3.8: Two symmetric dam break problems over triangular bump.

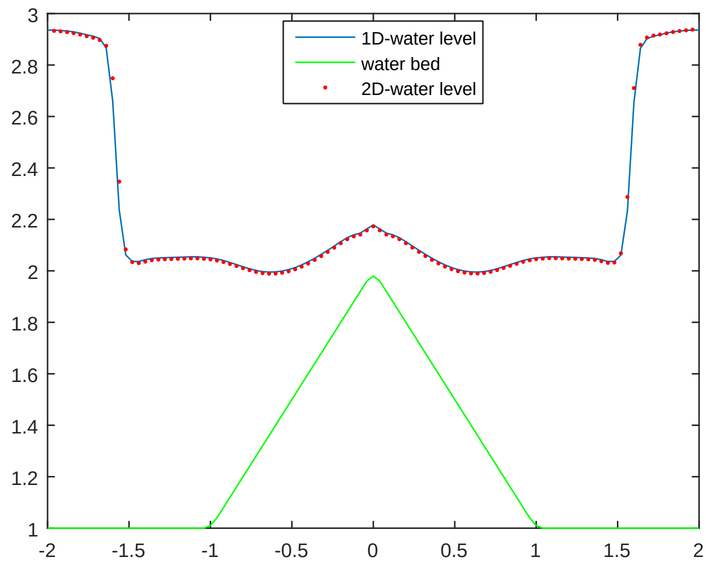


Figure 3.9: Two Symmetric Dam Break Problems: comparison between generated water level in the one-dimensional experiment and a cross section from the two dimensional experiment at time t=1.0179.

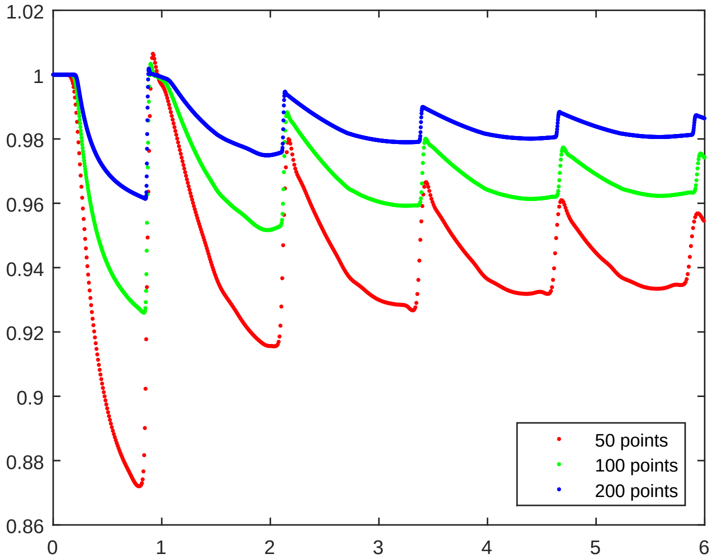
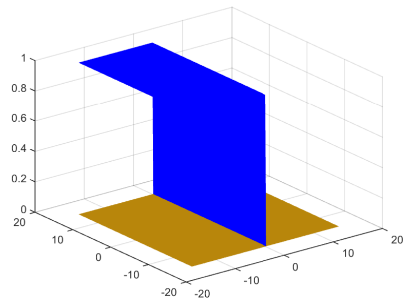
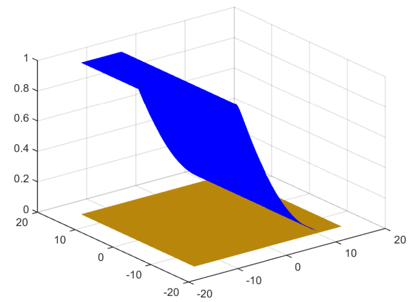


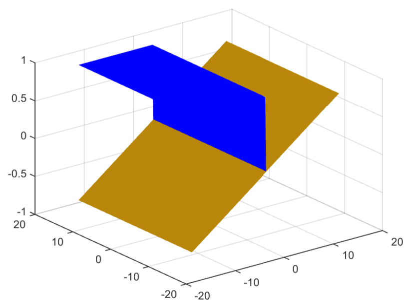
Figure 3.10: Two Symmetric Dam Break Problems: graph of the curve $c(t)/c_0$ on the time interval [0,6], obtained using 50, 100, and 200 grid points.



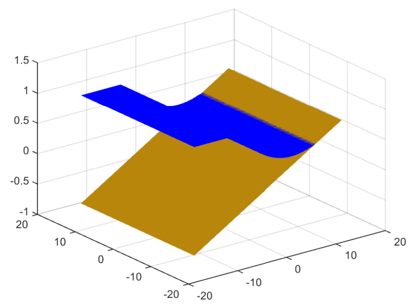
(a) $\alpha = 0, t = 0$.



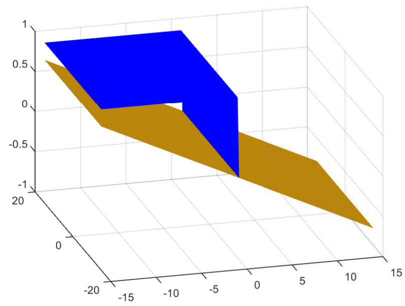
(b) $\alpha = 0, t_f = 2$.



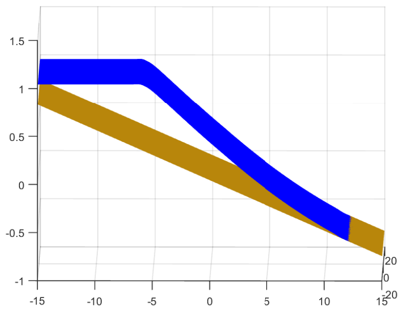
(c) $\alpha = \frac{\pi}{60}, t = 0$.



(d) $\alpha = \frac{\pi}{60}, t_f = 2$.

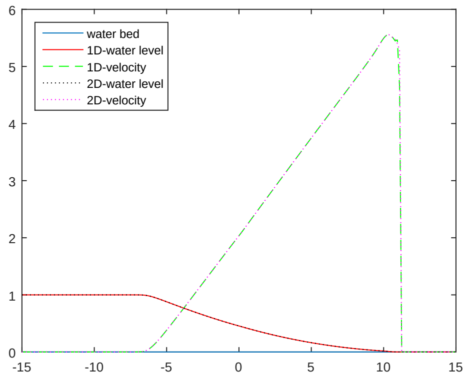


(e) $\alpha = -\frac{\pi}{60}, t = 0$.

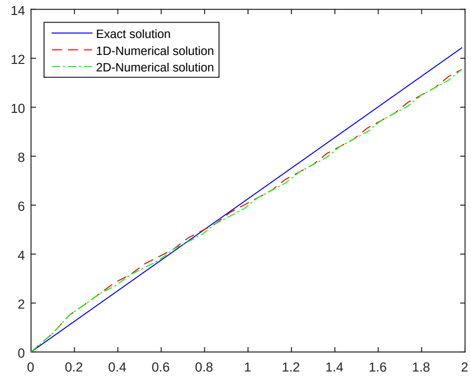


(f) $\alpha = -\frac{\pi}{60}, t_f = 2$.

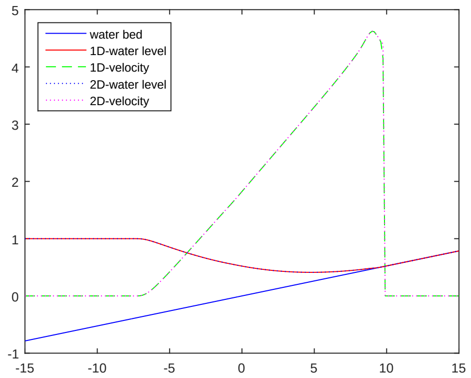
Figure 3.11: Dam Break over a plane: water level $h + z$ at $t = 0$ (left) and $t_f = 2$ (right).



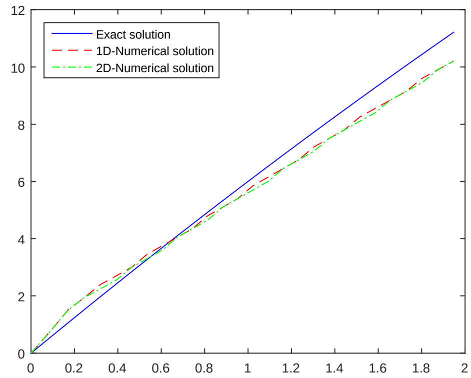
(a) $\alpha = 0$



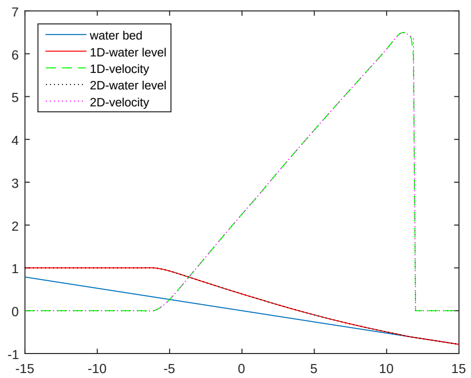
(b) $\alpha = 0$



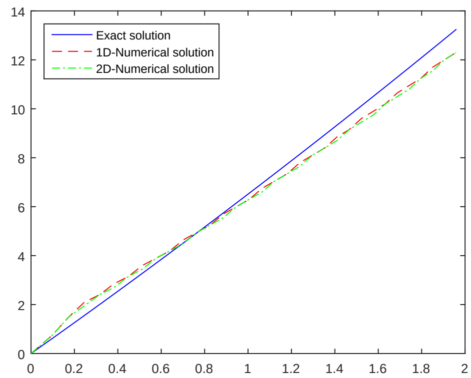
(c) $\alpha = \frac{\pi}{60}$



(d) $\alpha = \frac{\pi}{60}$



(e) $\alpha = -\frac{\pi}{60}$



(f) $\alpha = -\frac{\pi}{60}$

Figure 3.12: Dam Break over a plane: comparison between the one-dimensional water height and velocity with a two-dimensional cross section (left), comparison between the one-dimensional front positions with a two-dimensional cross section (right).

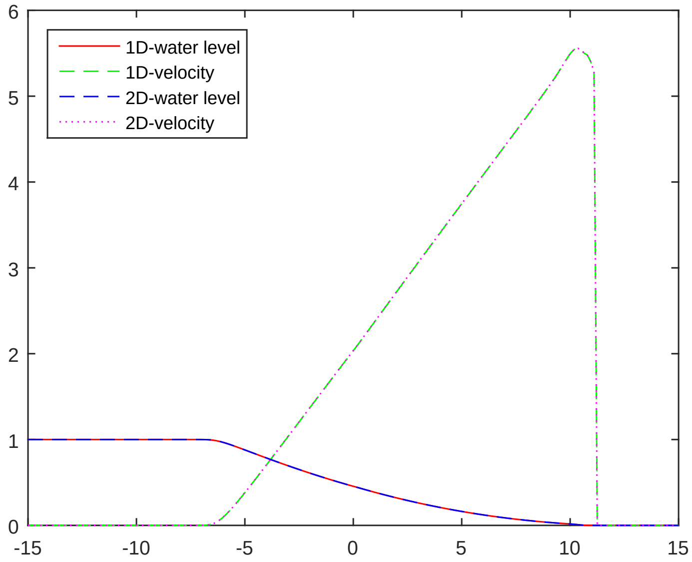


Figure 3.13: Dam break over plane: comparison between cross section along the x-axis and cross section along the y-axis..

Chapter 4

Conclusion

In this thesis, we present a well-balanced central scheme for systems of two-dimensional shallow water equations that treats wet and dry states and maintains the steady state requirement of the SWE systems when needed. In chapter 2 we present the one-dimensional well-balanced central scheme discussed in [4]. The one-dimensional system is first solved using the well-balanced scheme with the aid of the surface gradient method that introduce a new discretization of the water height function by first linearizing the water level function. In this case the steady state is successfully preserved. We then apply this scheme on several numerical experiments. An extension of the one-dimensional well-balanced scheme is presented in chapter 3 which was developed in [6]. As in the one-dimensional case, the scheme has the well-balanced property. This property results from discretizing the source term according to the flux divergence using sensor functions that force the discretization of the partial derivatives of the water bed function in the source term to follow that of the water height function. The two-dimensional well-balanced scheme is capable of maintaining a proper and clean interaction between wet and dry states whenever water run-ups are present. The negative water heights executed by the forward and the backward projection steps in this case are corrected by redefining the gradient components of the water height interpolant. Classical shallow water equations problems are solved with wet and dry states using the developed scheme. The proposed scheme is then validated in the numerical experiments' section. A very good agreement between our results and and the corresponding ones appearing in the literature. For future work, one may develop a new scheme for the two-dimensional SWEs that solves steady state problems with wet and dry interactions while preserving the well-balanced property of the numerical scheme.

Bibliography

[1] H. Nessyahu and E.Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, 1990, J. Comp. Phys., 87, 2, 408-463.

[2] N. Botta, R. Klein, S. Langenberg, and S. Lützenkirchen, *Well balanced finite volume methods for nearly hydrostatic flows*, 2004, J. Comp. Phys., 196, 2, 539-565.

[3] F. Bouchut, *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources*, 2004, Birkhäuser Verlag.

[4] R. Touma, *Well-balanced central schemes for systems of shallow water equations with wet and dry states (One-dimensional case)*, Applied Mathematical Modelling, 40 (4), 2929-2945.

[5] J.G.Zhou,D.M.Causon,C.G.Minigham,and D.M.Ingram, *The Surface Gradient Method for the Treatment of Source Terms in the Shallow-Water Equations*, Journal of computational physics 168,1-25(2001).

[6] R.Touma and S.khankan, *Well-balanced unstaggered central schemes for one and two-dimensional shallow water equation systems*, Applied Mathematics and computation 218(2012) 5948-5960.

[7] R.Touma, *Central Unstaggered Finite Volume Schemes for Hyperbolic Systems:Applications to Unsteady Shallow Water Equations*, Applied Mathematics and computation 213(2009) 47-59.

[8] P.Glaister, *Approximate Riemann solutions of the shallow water equations*, J.Hydraulic Res. 26,293(1988).

[9] F.Alcrudo and P.Garcia-Navarro, *A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations*, Int.J.Numer.Methods Fluids 16,489(1993).

[10] P.L.Roe, *Upwind difference schemes for the hyperbolic conservation laws with source terms*, in proceeding of the conference on Hyperbolic problems, edited by carasso, Raviart,and Serre,p.41(Springer-Verlag,New York,1986).

[11] P.Glaister, *Prediction of super critical flow in open channels*, Comput. and Math.Appl.Report 13/89,reading university,1989.

[12] M.E.Vazquez-Cendon, *Improved treatment of source terms in upwind schemes for the shallow water equations in channel with irregular geometry*, Journal of computational physics 148,497-526(1999).

[13] N. Crnjaric-Zic,S.Vukovic and L.Sopta, *Balanced central NT schemes for the shallow water equations*, Z. Drmac et al. (eds.), proceedings of the Conference on Applied Mathematics and Scientific Computing,part 2(Z.Drmac et. al., eds.)Springer(2005) 171-185.

[14] B.Neta and L.Lustman, *Parallel Conservative Scheme for Solving the Shallow Water Equations*, Noval postgraduate School, Department of Mathematics,Monterey,CA 93941.

[15] A.Arakawa and V.R.Lamb, *A Potential Enstrophy and Energy Conserving Scheme for the Shallow Water Equations*, Mon.Wea.Rev.,109(1981),18-36.

[16] I.M.Navon, *feudx; A two-stage high accuracy, finite-element Fortan program for solving shallow water equations*, Computers and Geosciences, 13(1987),255-285.

[17] B.Neta, R.Thanakij, *Finite Element Approximation of the shallow water equations on the Maspar*

[18] A.Stewart and P.J.Dellar, *Multilayer shallow water equations with complete Coriolis force* Part I:Derivation on a non-traditional beta-plane, OCIAM, Mathematical Institute, 24-29 St Giles',Oxford OX1 3LB, UK.

[19] G.S.jiang, D.Levy, C.T.Lin, S.Osher, E.Tadmor, *High-resolution non-oscillatory central schemes with non-staggered grids foe hyperbolic conservation laws*, SIAM J.Numer.Anal.35(6)(1998) 2147-2168.

[20] A.Bolzano *Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models*, Coastal Eng.34(1) (1998) 83-107.

[21] J.M.Gallardo, C.Pares and .M.Castro *On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas*, J.Comput.phys.227(1)(2007)574-601.

[22] M.J.Castro, A.F.Ferreiro, J.A.Garcia-Roidriguez , J.M.Gonzales-Vida, J.Macias,C.Pres,M.E.Vazquez-Cendon, *The numerical treatment of wet/dry fronts in shallow flows: application to one-layer and two-layer systems*, Math.Comput.Model.42(3)(2005) 419-439.

[23] J.M.Greenberg, A.Y.Leroux, *A well-balanced scheme for the numerical processing of source terms in hyperbolic equations*, SIAM J.Numer.Anal.33(1)(1996)1-16.

[24] A.Chertok,S.cui,A.kurganov,T.Wu, *Well-Balanced positivity preserving central up-wind scheme for the shallow water system with frictions terms*, Int. J. Numer. Meth. Fluids 2015; 78:355–383

[25] Y.Xing,X.zhang, *Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes*

[26] R.Gandham,D.Medina,T.Warburton, *GPU Accelerated Discontinuous Galerkin Methods for Shallow Water Equations*, arXiv:1403.1661v1 [math.NA] 7 Mar 2014

[27] G.-S.Jiang, D. Levy, C.-T. Lin, S. Osher, E. Tadmor, *High-resolution non oscillatory central schemes with non-staggred grids for hyperbolic conservation laws*, SIAM J.Numer. Anal. 35 (1998) 2157-2168.

[28] E.F. Toro, A.Hidalgo, M.Dumbser, *FORCE schemes on unstructured meshes I: Conservative Hyperbolic Systems*, J.Comput. Phys. 228 (2009) 3368-3389.

[29] A.N. Guarendi, A.J.Chandy, *Non-oscillatory central schemes for hyperbolic systems of conservation laws in three-space dimensions*, Sci. World J.2013 (2013)

[30] G.-S.Jiang, E.Tadmor *Non-oscillatory central schemes for multidimensional hyperbolic conservation laws*, SIAM J.Sci.Comput. 19(6)(1998) 1892-1917.

[31] R.Touma, P.Arminjon, *Central finite volume schemes with constrained transport divergence treatment for three-dimensional ideal MHD*, J. Comput. Phys. 212 (2) (2006) 617-636.