NTNU
Norwegian University of
Science and Technology

The Lattice Boltzmann Method for Steady State Sediment Transport.

Thor Olaf Klemsdal

Master of Science in Physics and Mathematics
Submission date: June 2016
Supervisor: Alex Hansen, IFY
Co-supervisor: Jan Tveiten, Schlumberger

Norwegian University of Science and Technology
Department of Physics

Abstract

Simulating water behaviour in coastal areas is an important tool when predicting sedimentary processes over geological time. We implement a general lattice Boltzmann model for shallow water flow, that can handle having dry areas in the simulated system. The lattice Boltzmann model is a relative new numerical technique where direct computing between dry and wet nodes was only achieved in the last three years. In this thesis we tried to implement the model in a way that allows us to gain accurate results when working with realistic systems.

We found our simulator to be accurate in one dimension when simulating with bed friction and in two dimensions when simulating frictionless systems. We also observed some unphysical slowing of the conversion dry nodes to wet nodes and vice versa.

Sammendrag

Å kunne simulere vannbevegelser i kystområder er et viktig redskap når man prøver å forutse sedimentære prosesser over geologisk tid. Vi implementerer en generell gitter Boltzmann-modell for grunne vannstrømninger, som kan takle å ha tørre områder i det simulerte systemet. Gitter Boltzmann modellen er en relativt ny numerisk teknikk, hvor direkte beregning mellom tørre og våte noder først ble mulig i løpet av de siste tre årene. I denne oppgaven har vi prøvd å implementere modellen slik at vi får høy nøyaktighet i resultatet når vi simulerer realistiske systemer.

Vi fant at simulatoren var treffsikker i én dimensjon når vi simulerte med bunnfriksjon og i to dimensjoner når vi brukte friksjonsløse systemer. Vi observerte også ufysisk nedbremsing av overgangen fra tørre noder til våte, og omvendt.

Preface

This thesis is written as part of my Master of Science in Applied Physics and Mathematics degree. It is worth 30 ECTS credits and is being completed at the Norwegian University of Science and Technology, NTNU. The work has been done under the supervision of Prof. Alex Hansen at the Department of Physics and Jan Tveiten at Schlumberger.

I would like to thank Jan Øystein Bakke at Schlumberger for providing the interesting project. It has been both a rewarding and challenging task to work on. I would also like to thank Jan Tveiten for offering helpful suggestions and advice whenever I was stuck or confused. In addition, I would like to offer thanks to Michael Niebling at Schlumberger for helpful discussions, and Alex Hansen for organizing everything in addition to giving advice. I also want to offer my thanks to Schlumberger for providing me with a place to stay while I visited them in Stavanger.

Personally I would like to thank my family: Ingvild Tøgersen Kjellstadli, for providing unending love, support and proofreading, and my mother and father for trying to understand and help where possible. Lastly, I want to offer thanks to Jonas Tøgersen Kjellstadli for assisting with proofreading.

Trondheim, June 27, 2016

Thor Olaf Klemsdal

Contents

Abstract i
Sammendrag iii
Preface v

1 Introduction **1**
1.1 Shallow Water Equations 1
1.2 Lattice Boltzmann Method 2
1.3 Organization 2

2 Shallow water equations **3**
2.1 Introduction 3
2.2 Navier-Stokes equations 3
2.3 Shallow water equations 3

3 Lattice Boltzmann theory **13**
3.1 Introduction 13
3.2 Lattice gas automata 13
3.3 Lattice Boltzmann equation 14
3.4 Lattice pattern 15
3.5 Local equilibrium distribution function 16
3.6 Recovery of the Shallow Water Equations 19
3.7 Stability 20
3.8 Force Terms 21
3.9 Wet and dry boundary 24
3.10 Boundary Conditions 25
3.11 Algorithm 25

4 Results **29**
4.1 Still water over flat seabed 29
4.2 Still water over regular bed 30
4.3 Steady flow over a bump 30
4.4 Tidal flow over uneven bed 31
4.5 Still water over two-dimensional bump 34
4.6 Steady flow down a slope 35
4.7 Steady flow down a two-dimensional slope 36
4.8 Sloshing 36

5 Discussion **41**

6 Conclusion **43**
6.1 Future work 43

1 Introduction

Recently, there has been much work on developing tools to produce quantitative models of a reservoir or a basin based on geologic knowledge at Schlumberger Limited (SLB). The geologic process modeler (GPM) they developed models the structural and sedimentary processes through geologic time and can also be used to understand the different physical processes that worked in the past to create today's environment. GPM consists of a simulation module and of a pre- and postprocessor.

This allows the software to generate an initial system, send it forward to the simulator that simulates a result before it gets sent to a postprocessor for further study. To simulate over geological time, GPM runs through the simulation module several times. The module first simulates how the water behaves, then changes in sediments, before this new system is sent back to the water simulator for a new cycle. Each cycle is assumed to generate the equivalent of 0.5 years of changes to the system. As GPM is created to model over thousands and millions of years, this restricts the length each cycle can last.

The GPM is as of now single threaded, which causes the water simulator used to run slower than what is wanted. As such, we are looking to develop a new and faster water simulator that can take advantage of the recent improvements in multi threaded computing. Studying the needs of the simulator, it was decided to develop a Lattice Boltzmann (LB) simulation program as LB is inherently easy to parallelize.

The task of this thesis is thus to create a simulator that can solve steady state sediment transport using LB formulation. As the simulation of sediments is considered complete, the main task is to implement and verify the simulation of water.

1.1 Shallow Water Equations

All fluid flows need to obey Newton's second law of motion. The result of implementing conservation of mass and momentum for fluid flow is the Navier-Stokes equations (NSE). These equations, however, are very hard to utilize in an efficient fashion and is as such not very efficient to use in computational fluid dynamics (CFD). When working with flows with much greater horizontal than vertical scale, the NSEs can be integrated over their depth giving out the Shallow water equations (SWE). SWE are hyperbolic partial differential equations (PDE), and are also known as Saint Venant equations. They describe a flow running under a pressure surface, like the water surface. A side effect of operating under the condition of a small vertical scale compared to the horizontal scale is that the vertical velocity must be small to conserve mass. This condition also makes SWE inherently good for working with fluid flows in river and channel systems or in coastal areas.

Sadly, SWE are not easy to solve numerically and much work has gone into solving them. Different computational approaches like finite-difference method (FDM), finite-element method (FEM) and finite-volume method (FVM) have been tried to solve this with mixed results [1], [2] and [3]. The numerical complexity of those methods to correctly calculate the source terms caused issues with the accuracy of the simulations.

The LB method was, because of its ease of implementing source terms, suggested as an option to solve SWE. Several papers have also reported success with using LB method to simulate wind driven ocean circulation with SWE [4] and [5]. In 2011, a study of

the stability of using LB to solve SWE showed that as long as the model is stable and consistent it should converge [6].

1.2 Lattice Boltzmann Method

The lattice Boltzmann method (LBM) we see today is the result of continual improvement, beginning with the Cellular automata (CA) in the late 1940s. CA is a two-dimensional grid of cells which all have the same rules for evolution and for which time moves discretely. Each cell has its own boolean (alive or dead) initial state which it updates individually each time-step based on the given evolutionary rules. One of the more famous uses of CA is the *Game of Life* [7] which J. Conway created in 1970, which portrays how complex behaviour can be simple numerical steps.

In 1973, the HPP model [8] was invented by Hardy, de Pazzis and Pomeau. The CA then evolved into what is now referred to as lattice gas automata (LGA), when it was attempted to make CA conserve mass and momentum. With the square lattice used, it was impossible to fully implement NSE, which was the goal at the time. In 1987, the FHP model [9] was created by Frisch, Hasslacher and Pomeau. It was the first LGA that managed to simulate the NSE from a set of macroscopic equations, however, it suffered largely from numerical noise. To improve the method, LBM was developed.

LBM can be derived directly from the Boltzmann equation resulting in a simulation method where fluid flows are calculated from microscopic events. A side effect of this is that it can easily calculate two phase systems and other complex systems, where other methods derived from macroscopic events will face difficulties. Other important advantages with the LBM is that it is local in space so parallel computing is easily achieved, all the computational steps are simple to calculate, and it operates with only one unknown variable, lessening the load per cycle.

1.3 Organization

This thesis contains the following chapters: Chapter 1 is where we introduce the background for this thesis. We also give some brief information on SWE and LBM which are the primary subjects of this thesis. In chapter 2 we derive the SWE from the depth averaged NSE. The resulting equations are then used as the governing equations in the simulator. The third chapter is focused on deriving the LBM. We derive all the terms used in the method and establish their inherent accuracy. In this chapter we also cover how to apply boundary conditions, handle the wetting/drying front and proving that the method simulate the SWE. In chapter 4 we present the test cases used and the simulation results while computing these. In chapter 5 the results presented in chapter 4 are discussed. The final chapter contains the conclusions we arrived at based on the results in chapter 4 and discussions in chapter 5. It also contains suggestions for further work based on the shortcomings of the present simulator.

2 Shallow water equations

2.1 Introduction

Water, just like every other fluid flow, needs to obey conservation laws for momentum and mass. In theory this means that a complete solution to any flow can be found by solving the Navier-Stokes equations. Sadly, these equations are very costly to calculate, so it is often advised to find simpler solutions. Water can in most situations be treated as incompressible, which simplifies matters greatly. Further, as the horizontal scale in coastal regions, which is of most value for this paper, is of much greater magnitude than the vertical scale, the use of shallow water equations where deemed ideal. To derive the useful SWEs we will in this chapter follow the derivation done by Zhou [10].

2.2 Navier-Stokes equations

The three-dimensional fluid flow version of Newton's second law of motion for incompressible fluids are the Navier-Stokes equations. In tensor form they are

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.1}$$

and

$$\frac{\partial u_j}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = f_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}. \tag{2.2}$$

For this we have used Einstein summation convention and t is time, ν is the kinematic viscosity, p is pressure, f_i is body forces and ρ is the density of the fluid. Hence Eq. (2.1) becomes

$$\frac{\partial u_j}{\partial x_j} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{2.3}$$

where u is the flow velocity in x-direction, v is the flow velocity in y-direction and w is the flow velocity in z-direction. As the terms in NSEs have physical interpretations, they are all named after their physical role. The left side of Eq. (2.2) is called the inertia term, with the different parts on the right called respectively body force term, pressure term and viscous term. These equations have very few situations where analytical solutions are possible, which leaves numerical simulation needed to find stable solutions. This is one of the major reasons why computational fluid dynamics (CFD) will continue to grow as a field as we need to understand more and more complex flows.

2.3 Shallow water equations

Rivers and coastal regions are usually described more by horizontal motion rather than vertical motion. This is usually because the vertical scale everything works on is very small compared to the horizontal scale. With comparable small vertical motion it was decided it is more efficient to use the assumption of hydrostatic pressure, meaning the vertical acceleration can be ignored. This leads to SWEs of two and three dimensions, where the former is based on the depth-averaged velocity of a flow and the latter directly from Eq. (2.1) and (2.2). As both sets of equations suffer from inability to handle

separation vertically in a flow and the two-dimensional equations demands one degree less computing the general consensus has been to work with the two-dimensional SWEs.

When applying Eq. (2.1) and (2.2) on Earth the only body forces experienced is the Coriolis force and gravity. This leaves

$$f_x = f_c v, \qquad f_y = f_c u, \qquad f_z = -g, \tag{2.4}$$

where f_c is the Coriolis parameter and $g = 9.81 m/s^2$ is the gravitational acceleration. Given its weak influence on the systems, the Coriolis force will be neglected from the simulations, but will be included in this chapter for completeness. The Coriolis parameter is $f_c = 2\omega \sin\phi$, where ϕ is the latitude of the area and $\omega \approx 7.3 \times 10^{-5} rad/s$ is the angular velocity of the Earth's rotation.

For us to get the mass continuity Eq. (2.1), to the wanted dimensional order we integrate over depth and get,

$$\int_{z_b}^{h+z_b} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \mathrm{d}z = 0, \tag{2.5}$$

which simplifies to

$$\int_{z_b}^{h+z_b} \frac{\partial u}{\partial x} \mathrm{d}z + \int_{z_b}^{h+z_b} \frac{\partial v}{\partial y} \mathrm{d}z + w_s - w_b = 0, \tag{2.6}$$

where the vertical velocity at surface and channel bed is w_s and w_b. z_b is the bed elevation in the chosen frame of reference and h is water elevation relative to z_b, as shown in Fig. 2.1.

To further solve this we use Leibnitz rule [11],

$$\int_a^b \frac{\partial f(x,y)}{\partial y} \mathrm{d}x = \frac{\partial}{\partial y} \int_a^b f(x,y) \mathrm{d}x - f(b,y) \frac{\partial b}{\partial y} + f(a,y) \frac{\partial a}{\partial y} \tag{2.7}$$

on Eq. (2.6) resulting in,

$$\begin{aligned} &\frac{\partial}{\partial x} \int_{z_b}^{h+z_b} u \mathrm{d}z + \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} v \mathrm{d}z \\ &+ [w_s - u_s \frac{\partial}{\partial x}(h + z_b) - v_s \frac{\partial}{\partial y}(h + z_b)] \\ &- (w_b - u_b \frac{\partial z_b}{\partial x} - v_b \frac{\partial z_b}{\partial y}) = 0. \end{aligned} \tag{2.8}$$

This leaves the kinematic condition to be,

$$w_s = \frac{\partial}{\partial t}(h - z_b) + u_s \frac{\partial}{\partial x}(h + z_b) + v_s \frac{\partial}{\partial y}(h + z_b) \tag{2.9}$$

and

$$w_b = \frac{\partial z_b}{\partial t} + u_b \frac{\partial z_b}{\partial x} + v_b \frac{\partial z_b}{\partial y} \tag{2.10}$$

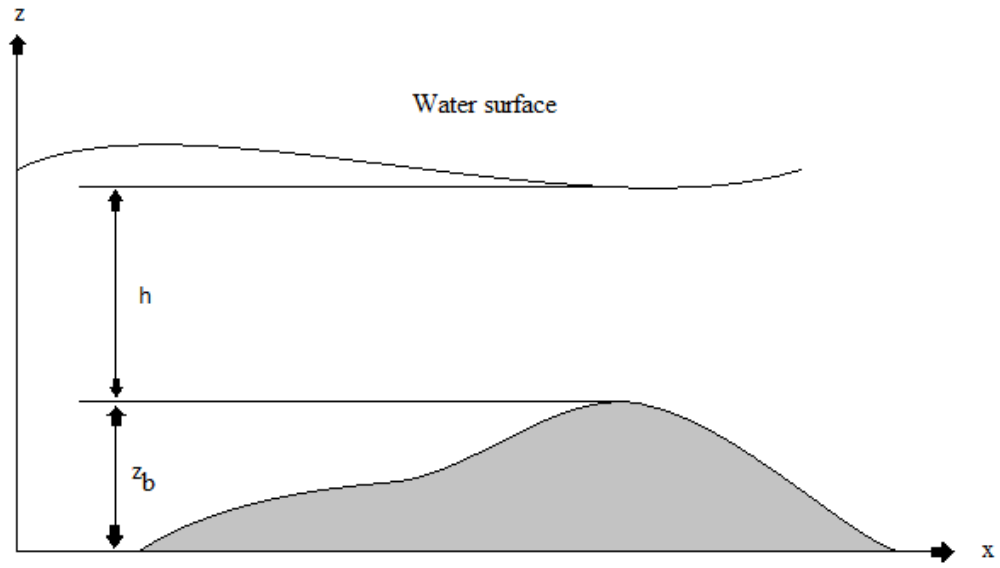


Figure 2.1: Sketch of water seen from the side.

at the free surface and channel bed. Inserting Eq. (2.9) and (2.10) into Eq. (2.8) results in

$$\frac{\partial h}{\partial t} + \frac{\partial (h\bar{u})}{\partial x} + \frac{\partial (h\bar{v})}{\partial y} = 0, \tag{2.11}$$

which is the mass continuity equation for shallow water flows. Here $\bar{u}$ and $\bar{v}$ is the depth-averaged velocity, defined as

$$\bar{u} = \frac{1}{h} \int_{z_b}^{h+z_b} u \mathrm{d}z, \tag{2.12}$$

and

$$\bar{v} = \frac{1}{h} \int_{z_b}^{h+z_b} v \mathrm{d}z. \tag{2.13}$$

Integrating over the depth in the x-component of Eq. (2.2) leaves

$$\begin{aligned} \int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (vu)}{\partial y} + \frac{\partial (wu)}{\partial z} \right] \mathrm{d}z = \int_{z_b}^{h+z_b} f_c v \mathrm{d}z \\ + \int_{z_b}^{h+z_b} \left[-\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \right] \mathrm{d}z. \end{aligned} \tag{2.14}$$

Of the terms on the left side of Eq. (2.14), the last term is the only one that can be evaluated by direct integration,

$$\int_{z_b}^{h+z_b} \frac{\partial(wu)}{\partial z} \mathrm{d}z = w_s u_s - w_b u_b. \tag{2.15}$$

The three other terms of the left side of Eq. (2.14) must be evaluated by use of Leibnitz rule (2.7),

$$\int_{z_b}^{h+z_b} \frac{\partial u}{\partial t} \mathrm{d}z = \frac{\partial}{\partial t} \int_{z_b}^{h+z_b} u \mathrm{d}z - u_s \frac{\partial(h+z_b)}{\partial t} + u_b \frac{\partial z_b}{\partial t}, \tag{2.16}$$

$$\int_{z_b}^{h+z_b} \frac{\partial(u^2)}{\partial x} \mathrm{d}z = \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} u^2 \mathrm{d}z - u_s^2 \frac{\partial(h+z_b)}{\partial x} + u_b^2 \frac{\partial z_b}{\partial x}, \tag{2.17}$$

$$\int_{z_b}^{h+z_b} \frac{\partial(vu)}{\partial y} \mathrm{d}z = \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} vu \mathrm{d}z - v_s u_s \frac{\partial(h+z_b)}{\partial y} + v_b u_b \frac{\partial z_b}{\partial y}. \tag{2.18}$$

The left side of Eq. (2.14) can now via Eq. (2.15), (2.16), (2.17) and (2.18) be written as

$$\begin{aligned} &\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial(u^2)}{\partial x} + \frac{\partial(vu)}{\partial y} + \frac{\partial(wu)}{\partial z} \right] \mathrm{d}z = \\ &\frac{\partial}{\partial t} \int_{z_b}^{h+z_b} u \mathrm{d}z + \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} u^2 \mathrm{d}z + \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} vu \mathrm{d}z \\ &+ u_s \left[w_s - \frac{\partial(h+z_b)}{\partial t} - u_s \frac{\partial(h+z_b)}{\partial x} - v_b \frac{\partial z_b}{\partial y} \right] \\ &- u_b \left(w_b - \frac{\partial z_b}{\partial t} - u_b \frac{\partial z_b}{\partial x} - v_b \frac{\partial z_b}{\partial y} \right). \end{aligned} \tag{2.19}$$

Inserting both w_s and w_b from Eq. (2.9) and (2.10), and $\bar{u}$ and $\bar{v}$ from Eq. (2.12) and (2.13) into Eq. (2.19) results in

$$\begin{aligned} &\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial(u^2)}{\partial x} + \frac{\partial(vu)}{\partial y} + \frac{\partial(wu)}{\partial z} \right] \mathrm{d}z = \\ &\frac{\partial(h\bar{u})}{\partial t} + \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} u^2 \mathrm{d}z + \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} vu \mathrm{d}z \end{aligned} \tag{2.20}$$

The second mean value theorem for integrals [12],

$$\int_a^b f(x) g(x) \mathrm{d}x = f(\zeta) \int_a^b g(x) \mathrm{d}x, \tag{2.21}$$

now allows us to rewrite

$$\int_{z_b}^{h+z_b} u^2 \mathrm{d}z = \check{u}_1 \int_{z_b}^{h+z_b} u \mathrm{d}z = \check{u}_1 h\bar{u}, \tag{2.22}$$

and

$$\int_{z_b}^{h+z_b} vu \mathrm{d}z = \check{u}_2 \int_{z_b}^{h+z_b} v \mathrm{d}z = \check{u}_1 h\bar{v}. \tag{2.23}$$

Using Eq. (2.21) is what causes the previously mentioned weakness of SWEs not being able to handle changes in the direction the water is flowing between the surface and the channel bed. This is because the second mean value theorem demands a continuous u and v that never change sign in the region of integration.

Assuming $\check{u}_1 = \theta_1\bar{u}$ and $\check{u}_2 = \theta_2\bar{u}$ we can with help from Eq. (2.22) and (2.23) simplify Eq. (2.20) into

$$\begin{aligned}\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial(u^2)}{\partial x} + \frac{\partial(vu)}{\partial y} + \frac{\partial(wu)}{\partial z}\right] \mathrm{d}z = \\ \frac{\partial(h\bar{u})}{\partial t} + \frac{\partial(\theta_1 h\bar{u}^2)}{\partial x} + \frac{\partial(\theta_2 h\bar{v}\bar{u})}{\partial y}.\end{aligned} \tag{2.24}$$

Based on Eq. (2.22) and (2.23), we can also determine θ_1 and θ_2, the momentum correction factors, to be

$$\theta_1 = \frac{1}{h\bar{u}^2} \int_{z_b}^{h+z_b} u^2 \mathrm{d}z, \tag{2.25}$$

and

$$\theta_2 = \frac{1}{h\bar{v}\bar{u}} \int_{z_b}^{h+z_b} vu \mathrm{d}z. \tag{2.26}$$

Repeating Eq. (2.14) to (2.24), but this time in y-direction results in

$$\begin{aligned}\int_{z_b}^{h+z_b} \left[\frac{\partial v}{\partial t} + \frac{\partial(uv)}{\partial x} + \frac{\partial(v^2)}{\partial y} + \frac{\partial(wv)}{\partial z}\right] \mathrm{d}z = \\ \frac{\partial(h\bar{v})}{\partial t} + \frac{\partial(\theta_2 h\bar{u}\bar{v})}{\partial x} + \frac{\partial(\theta_3 h\bar{v}^2)}{\partial y}.\end{aligned} \tag{2.27}$$

The third momentum correction factor θ_3 is found to be

$$\theta_3 = \frac{1}{h\bar{v}^2} \int_{z_b}^{h+z_b} v^2 \mathrm{d}z. \tag{2.28}$$

The Coriolis force term on the right side of Eq. (2.14) can easily be integrated to

$$\int_{z_b}^{h+z_b} f_c v \mathrm{d}z = f_c h \bar{v}. \tag{2.29}$$

To solve the pressure term of Eq. (2.14), we first look to the momentum equation (2.2). By solving the momentum equation in z-direction we quickly get

$$\frac{\partial p}{\partial z} = -\rho g. \tag{2.30}$$

This is a result of the vertical fluid acceleration being negligible compared to what is experienced horizontally, leading to $w \approx 0$. Integrating Eq. (2.30) results in

$$p = -\rho g z + C, \tag{2.31}$$

where C is an integration constant. At the surface $p = p_a$ and $z = h + z_b$ where p_a is the air pressure, which allows us to determine C as

$$C = \rho g(h + z_b) + p_a. \tag{2.32}$$

Inserting this into Eq. (2.31) gives

$$p = \rho g(h + z_b - z) + p_a. \tag{2.33}$$

Following the standard for coastal and hydraulic engineering [13], we set $p_a = 0$. As the differences in pressure along the surface of the water is negligible for all situations of interest, p_a is for all purposes a constant already, resulting in

$$p = \rho g(h + z_b - z), \tag{2.34}$$

which is the hydrostatic pressure approximation used in shallow water flows. By differentiating Eq. (2.34) with respect to x we now find that the pressure term in Eq. (2.14) is

$$\frac{\partial p}{\partial x} = \frac{\partial (h + z_b)}{\partial x}. \tag{2.35}$$

This form allows us to solve the integration of the pressure term given in Eq. (2.14), as it is not dependent on the z direction. It thus becomes

$$\int_{z_b}^{h+z_b} \frac{1}{\rho} \frac{\partial p}{\partial x} \mathrm{d}z = \frac{h}{\rho} \frac{\partial p}{\partial x}, \tag{2.36}$$

which with Eq. (2.35) inserted is

$$\int_{z_b}^{h+z_b} \frac{1}{\rho} \frac{\partial p}{\partial x} \mathrm{d}z = gh \frac{\partial (h + z_b)}{\partial x}. \tag{2.37}$$

To further solve Eq. (2.14) we now need to introduce two approximations for the first two parts of the viscous term as

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial x^2} \mathrm{d}z \approx \nu \frac{\partial^2 (h\bar{u})}{\partial x \partial x}, \tag{2.38}$$

and

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial y^2} \mathrm{d}z \approx \nu \frac{\partial^2 (h\bar{u})}{\partial y \partial y}. \tag{2.39}$$

As a result we now only need to calculate the last part of the viscous term to,

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial z^2} \mathrm{d}z = \left(\nu \frac{\partial u}{\partial z} \right)_s - \left(\nu \frac{\partial u}{\partial z} \right)_b. \tag{2.40}$$

Practice shows that the terms on the right side of Eq. (2.40) can be approximated by the shear stress inflicted by respectively the wind (former term) and the channel bed (latter term), hence

$$\left(\nu \frac{\partial u}{\partial z} \right)_s = \frac{\tau_{wx}}{\rho} \qquad \left(\nu \frac{\partial u}{\partial z} \right)_b = \frac{\tau_{bx}}{\rho}. \tag{2.41}$$

Inserting into Eq. (2.40) we now have

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial z^2} \mathrm{d}z = \frac{\tau_{wx}}{\rho} - \frac{\tau_{bx}}{\rho}. \tag{2.42}$$

Reconstructing Eq. (2.14) with the results of Eq. (2.24), (2.29), (2.37) - (2.39) and (2.42) gives us

$$\begin{aligned} \frac{\partial (h\bar{u})}{\partial t} + \frac{\partial (\theta_1 h \bar{u}^2)}{\partial x} + \frac{\partial (\theta_2 h \bar{u}\bar{v})}{\partial y} = \\ -g \frac{\partial}{\partial x} \left(\frac{h^2}{2} \right) + \nu \frac{\partial^2 (h\bar{u})}{\partial x \partial x} + \nu \frac{\partial^2 (h\bar{u})}{\partial y \partial y} \\ -gh \frac{\partial z_b}{\partial x} + f_c h \bar{v} + \frac{\tau_{wx}}{\rho} - \frac{\tau_{bx}}{\rho}, \end{aligned} \tag{2.43}$$

which is the momentum equation for shallow water flows in x-direction. The y-direction momentum equation can similarly be written as

$$\begin{aligned} \frac{\partial (h\bar{v})}{\partial t} + \frac{\partial (\theta_2 h \bar{u}\bar{v})}{\partial x} + \frac{\partial (\theta_3 h \bar{v}^2)}{\partial y} = \\ -g \frac{\partial}{\partial y} \left(\frac{h^2}{2} \right) + \nu \frac{\partial^2 (h\bar{v})}{\partial x \partial x} + \nu \frac{\partial^2 (h\bar{v})}{\partial y \partial y} \\ -gh \frac{\partial z_b}{\partial y} - f_c h \bar{u} + \frac{\tau_{wy}}{\rho} - \frac{\tau_{by}}{\rho}. \end{aligned} \tag{2.44}$$

Now we only need to determine the momentum correction factors, θ_1, θ_2 and θ_3 to have a fully functioning momentum equation. We could in theory determine them now based on their definition in Eq. (2.25), (2.26) and (2.28) however, barring a few simple situations, this is impossible as there is no correct universal velocity profile describing the flow. The solution has been to use an approximation where $\theta_1 = \theta_2 = \theta_3 = 1$ which has shown to deliver good numerical results in many situations [14], [15], [16].

Using this we can now write the momentum equations as

$$\begin{aligned} \frac{\partial(h\bar{u})}{\partial t} + \frac{\partial(h\bar{u}^2)}{\partial x} + \frac{\partial(h\bar{u}\bar{v})}{\partial y} = \\ -g\frac{\partial}{\partial x}\left(\frac{h^2}{2}\right) + \nu\frac{\partial^2(h\bar{u})}{\partial x \partial x} + \nu\frac{\partial^2(h\bar{u})}{\partial y \partial y} \\ -gh\frac{\partial z_b}{\partial x} + f_c h\bar{v} + \frac{\tau_{wx}}{\rho} - \frac{\tau_{bx}}{\rho}, \end{aligned} \tag{2.45}$$

and

$$\begin{aligned} \frac{\partial(h\bar{v})}{\partial t} + \frac{\partial(h\bar{u}\bar{v})}{\partial x} + \frac{\partial(h\bar{v}^2)}{\partial y} = \\ -g\frac{\partial}{\partial y}\left(\frac{h^2}{2}\right) + \nu\frac{\partial^2(h\bar{v})}{\partial x \partial x} + \nu\frac{\partial^2(h\bar{v})}{\partial y \partial y} \\ -gh\frac{\partial z_b}{\partial y} - f_c h\bar{u} + \frac{\tau_{wy}}{\rho} - \frac{\tau_{by}}{\rho}. \end{aligned} \tag{2.46}$$

We now have the governing shallow water equations, Eq, (2.11), (2.45) and (2.46), and can write them in tensor form as

$$\frac{\partial h}{\partial t} + \frac{\partial h u_j}{\partial x_j} = 0 \tag{2.47}$$

and

$$\frac{\partial h u_j}{\partial t} + \frac{\partial h u_i u_j}{\partial x_j} = -g\frac{\partial}{\partial x_i}(\frac{h^2}{2}) + \nu\frac{\partial^2(h u_i)}{\partial x_j \partial x_j} + F_i, \tag{2.48}$$

where the overbars are omitted for convenience. F_i is the force term, which is defined as

$$F_i = -gh\frac{\partial z_b}{\partial x_i} + \frac{\tau_{wi}}{\rho} - \frac{\tau_{bi}}{\rho} + \Omega_i. \tag{2.49}$$

Here the Coriolis term, Ω_i, is determined to be

$$\Omega_i = \begin{cases} f_c h v, & i = x, \\ -f_c h u, & i = y. \end{cases} \tag{2.50}$$

The bed shear stress in direction i, τ_{bi}, is determined by the depth-averaged velocities as

$$\tau_{bi} = \rho C_b u_i \sqrt{u_j u_j}, \tag{2.51}$$

where the bed friction coefficient, C_b, is determined by

$$C_b = \frac{g}{C_z^2}. \tag{2.52}$$

C_z is the Chezy coefficient, which in turn is decided by the Manning equation

$$C_z = \frac{h^{\frac{1}{6}}}{n_b}, \tag{2.53}$$

where n_b is the Manning's coefficient at the bed. This means Eq. (2.52) is

$$C_b = \frac{g n_b^2}{h^{\frac{1}{3}}}. \tag{2.54}$$

The stress caused by the wind, τ_{wi} is defined as

$$\tau_{wi} = \rho_a C_a u_{wi} \sqrt{u_{wj} u_{wj}} \tag{2.55}$$

Where ρ_a is the air density, u_{wi} is the speed of air in i-direction and C_a is the resistance coefficient of the air.

3 Lattice Boltzmann theory

3.1 Introduction

The lattice Boltzmann method is a numerical method that uses a few simple steps to simulate real physical flows. The first step is an equation that explains how particles move, which is the Boltzmann equation. The second step is to determine particle direction based on the pattern of the lattice. The third and final step is to find the distribution function based on the governing flow equations. The governing flow equations in this thesis are the SWEs derived in chapter 2. With these three simple steps, LBM is capable of simulating complicated flows where other CFD methods fail.

This chapter will fully explain each step of LBM and show how they came to be. To derive the lattice Boltzmann equation we follow in large part of the work done by Zhou [10].

3.2 Lattice gas automata

LGA is, as mentioned in the introduction of this thesis, historically the precursor of LBM. The LGA is fundamentally a way to imagine particle interaction in a gas where time, position and velocity is discrete. By fixing the positions available for a particle to nodes on a grid and applying evolutionary rules it was successful at creating a model where particles could collide based entirely on local nodes. To keep track of particle occupation in nodes, a Boolean variable was defined as $n_\beta(\mathbf{x}, t)$, where $\beta = 1, \cdots, M$ and M is the number of possible velocity directions at any given node [17]. A generalized LGA evolution equation would then be on the form of

$$n_\beta(\mathbf{x} + \mathbf{e}_\beta, t + 1) = n_\beta(\mathbf{x}, t) + \Omega_\beta(n(\mathbf{x}, t)), \tag{3.1}$$

where Ω is a collision operator and $\mathbf{e}_\beta$ is the velocity vector to the nearest node in β direction. The right side of the equation is now the collision step of the method while the left side shows how particles stream to neighboring nodes. One major weakness of this method was that no particle could ever stay in the same position as another particle. This was done to increase memory efficiency and had the added benefit of having a universal Fermi-Dirac local equilibrium distribution [9].

Another weakness of LGA was that it suffered from statistical noise. In 1988, it was suggested to change the particle occupation variable, the Boolean, n_β into $f_\beta = \langle n_\beta \rangle$, which is the real-valued particle distribution function. Combining this with ignoring particle-particle correlation and individual particle motion allows the process to eliminate the statistical noise [18].

By using the BGK collision operator [19], it was found that the resulting equation had improved flexibility and efficiency [20]. The BGK collision operator is named after Bhatnagar, Gross and Krook and is defined by

$$\Omega_\beta = -\frac{1}{\lambda}(f - f^{eq}), \tag{3.2}$$

where f^{eq} is the equilibrium distribution function and λ is a relaxation time.

Thus the LGA evolved into

$$f_\beta(\mathbf{x} + \mathbf{e}_\beta, t + 1) - f_\beta(\mathbf{x}, t) = -\frac{1}{\lambda}(f_\beta(\mathbf{x}, t) - f_\beta^{eq}(\mathbf{x}, t)). \tag{3.3}$$

Assuming the left side is a time derived function where Δt is the time step, the left side can easily be replaced with $\frac{f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t)-f_\beta(\mathbf{x},t)}{\Delta t}$. Thus the LGA transformed fully into the LBM,

$$f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t) - f_\beta(\mathbf{x},t) = -\frac{1}{\tau}(f_\beta(\mathbf{x},t) - f_\beta^{eq}(\mathbf{x},t)), \tag{3.4}$$

given $\tau = \lambda/\Delta t$.

3.3 Lattice Boltzmann equation

The LB equation can also be derived more analytically from the Boltzmann BGK equation [19],

$$\frac{\partial f}{\partial t} + \mathbf{e}\cdot\nabla f = -\frac{1}{\lambda}(f - f^{eq}), \tag{3.5}$$

where the nabla operator is $\nabla = \frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j}$ and f^{eq} is the Maxwell-Boltzmann equilibrium distribution function. It is defined as

$$f^{eq} = \frac{\rho}{(2\pi/3)^{D/2}} exp\left(-\frac{3}{2}(\mathbf{e}-\mathbf{V})^2\right), \tag{3.6}$$

with D as the spatial dimension, ρ as the fluid density, $\mathbf{e}$ the particle velocity and $\mathbf{V}$ the fluid velocity. Both $\mathbf{e}$ and $\mathbf{V}$ are normalized by $\sqrt{3RT}$ resulting in the speed of sound at $U_s = \frac{1}{\sqrt{3}}$ [17]. The T mentioned in the normalization is the temperature, while R is the ideal gas constant. Further ρ and $\mathbf{V}$ are both variables that can be recovered from the distribution function as

$$\rho = \int f d\mathbf{e}, \qquad \mathbf{V}\rho = \int \mathbf{e} f d\mathbf{e}. \tag{3.7}$$

When $|\mathbf{V}|$ is much smaller than U_s, we can Taylor expand Eq. (3.6) into

$$f^{eq} = \frac{\rho}{(2\pi/3)^{D/2}} exp\left(-\frac{3}{2}\mathbf{e}^2\right)\left[1 + 3(\mathbf{e}\cdot\mathbf{V}) + \frac{9}{2}(\mathbf{e}\cdot\mathbf{V})^2 - \frac{3}{2}\mathbf{V}\cdot\mathbf{V}\right] \tag{3.8}$$

resulting in second order accuracy [21]. As LBM is a discrete model, we only wish to handle a discrete amount of particle velocities reducing $\mathbf{e}$ to $\mathbf{e}_\beta$ where $\beta = 1, \cdots, M$ and M is the number of directions in the lattice. With discretization, Eq. (3.5) can be written as

$$\frac{\partial f_\beta}{\partial t} + \mathbf{e}_\beta\cdot\nabla f_\beta = -\frac{1}{\lambda}(f_\beta - f_\beta^{eq}). \tag{3.9}$$

Both f_β and f_β^{eq} now depend on $\mathbf{e}_\beta$ i.e. $f_\beta = f(\mathbf{x}, \mathbf{e}_\beta, t)$. Since left side of Eq. (3.9) is a Lagrangian time derivative we can rewrite it as

$$\frac{\partial f_\beta}{\partial t} + \mathbf{e}_\beta\cdot\nabla f_\beta = \frac{f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t) - f_\beta(\mathbf{x},t)}{\Delta t} \tag{3.10}$$

With simple substitution and inserting the single relaxation time, $\tau = \lambda/\Delta t$, we end up with the lattice Boltzmann equation,

$$f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t) = f_\beta(\mathbf{x},t) - \frac{1}{\tau}(f_\beta(\mathbf{x},t) - f_\beta^{eq}(\mathbf{x},t)), \tag{3.11}$$

where the left side of the equation is the streaming term and the right side is the collision term. Whenever we deal with outer forces that work on the flow, we add a force term to the end of the equation [10],

$$f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t) = f_\beta(\mathbf{x},t) - \frac{1}{\tau}(f_\beta(\mathbf{x},t) - f_\beta^{eq}(\mathbf{x},t)) + \frac{\Delta t}{N_\beta e^2} e_{\beta i} F_i(\mathbf{x},t), \tag{3.12}$$

where Einstein summation convention is used. F_i is the i-component of the force and the lattice size is $e = \Delta x/\Delta t$. N_β is a constant that is defined as

$$N_\beta = \frac{1}{e^2}\sum_\beta e_{\beta i} e_{\beta i}. \tag{3.13}$$

3.4 Lattice pattern

Choosing lattice pattern is important to both determining the particle velocities and position of the grid points. It is also vital to calculating the constant N_β. The two most common two-dimensional patterns are the square lattice and the hexagonal lattice, both of which have several different particle speed configurations possible to choose among. However, careful analytical studies have shown that only the 9-speed square model and the 7-speed hexagonal model contain sufficient lattice symmetry necessary to recover the necessary flow equation [22].

Further study has shown that the square 9-speed model, later called D2Q9 [23], is both more accurate [24] and easier to apply boundary conditions (BC) to [25] than the hexagonal model. Because of this, the focus onward will only be on the D2Q9 model.

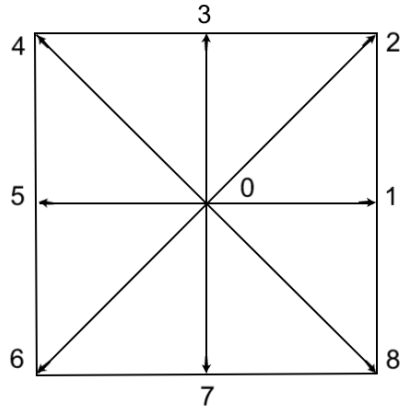


Figure 3.1: Sketch of the D2Q9 lattice pattern with $\mathbf{e}_\beta$, the particle speeds, numbered.

With the D2Q9 model we now have 9 particle velocities in each node. The velocity numbered 0, indicate zero particle speed, while 1 - 8 have velocity enough to move particles one lattice unit in each direction as seen in Fig. 3.1. The particle velocity vectors are

defined as

$$\mathbf{e}_\beta = \begin{cases} (0,0), & \beta = 0, \\ e\left[\cos(\frac{(\beta-1)\pi}{4}), \sin(\frac{(\beta-1)\pi}{4})\right], & \beta = 1,3,5,7, \\ \sqrt{2}e\left[\cos(\frac{(\beta-1)\pi}{4}), \sin(\frac{(\beta-1)\pi}{4})\right], & \beta = 2,4,6,8. \end{cases} \tag{3.14}$$

Using the inherent basic features of the D2Q9 lattice,

$$\sum_\beta e_{\beta i} = \sum_\beta e_{\beta i} e_{\beta j} e_{\beta k} = 0, \tag{3.15}$$

$$\sum_\beta e_{\beta i} e_{\beta j} = 6e^2 \delta_{ij}, \tag{3.16}$$

$$\sum_\beta e_{\beta i} e_{\beta j} e_{\beta k} e_{\beta l} = 4e^2(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) - 6e^4 \Delta_{ijkl} \tag{3.17}$$

with

$$\Delta_{ijkl} = \begin{cases} 1, & i = j = k = l, \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

Using Eq. (3.14) when calculating Eq. (3.13) results in

$$N_\beta = \frac{1}{e^2} \sum_\beta e_{\beta x} e_{\beta x} = \frac{1}{e^2} \sum_\beta e_{\beta y} e_{\beta y} = 6. \tag{3.19}$$

Thus Eq. (3.12) becomes the most common variant of the LBM for simulation of fluid flows, simply called the lattice Boltmzann equation (LBE),

$$f_\beta(\mathbf{x} + \mathbf{e}_\beta \Delta t, t + \Delta t) = f_\beta(\mathbf{x}, t) - \frac{1}{\tau}(f_\beta(\mathbf{x}, t) - f_\beta^{eq}(\mathbf{x}, t)) + \frac{\Delta t}{6e^2} e_{\beta i} F_i(\mathbf{x}, t). \tag{3.20}$$

3.5 Local equilibrium distribution function

The determination of the local equilibrium function plays an integral role in the LBM. With the right choice of f_β^{eq} it becomes possible to simulate different types of flow equations. The standard equilibrium function inherited from LGA was the Maxwell-Boltzmann equilibrium distribution function. However, the use of this function limited the use of the solver to NSE only. To overcome this limitation, the equilibrium function is treated as a power series in the macroscopic velocity u [26] on the form

$$f_\beta^{eq} = A_\beta + B_\beta e_{\beta i} u_i + C_\beta e_{\beta i} e_{\beta j} u_i u_j + D_\beta u_i u_i. \tag{3.21}$$

As the equilibrium function has the same symmetry as the lattice used, we can state from Fig. 3.1 that

$$A_1 = A_3 = A_5 = A_7 = \bar{A}, \qquad A_2 = A_4 = A_6 = A_8 = \tilde{A}. \tag{3.22}$$

This principle also gives us equivalent relations for B_β, C_β and D_β, which allow us to rewrite Eq. (3.21) as

$$f_\beta^{eq} = \begin{cases} A_0 + D_0 u_i u_i, & \beta = 0, \\ \bar{A} + \bar{B} e_{\beta i} u_i + \bar{C} e_{\beta i} e_{\beta j} u_i u_j + \bar{D} u_i u_i, & \beta = 1,3,5,7, \\ \tilde{A} + \tilde{B} e_{\beta i} u_i + \tilde{C} e_{\beta i} e_{\beta j} u_i u_j + \tilde{D} u_i u_i, & \beta = 2,4,6,8. \end{cases} \tag{3.23}$$

Considering that the equilibrium distribution function must obey the same laws as the SWE, we can find the coefficients in Eq. (3.23) from mass and momentum conservation. Thus we can state [10]

$$\sum_\beta f_\beta^{eq}(\mathbf{x}, t) = h(\mathbf{x}, t), \tag{3.24}$$

$$\sum_\beta e_{\beta i} f_\beta^{eq}(\mathbf{x}, t) = h(\mathbf{x}, t) u_i(\mathbf{x}, t), \tag{3.25}$$

$$\sum_\beta e_{\beta i} e_{\beta j} f_\beta^{eq}(\mathbf{x}, t) = \frac{1}{2} g h^2(\mathbf{x}, t) \delta_{ij} + h(\mathbf{x}, t) u_i(\mathbf{x}, t) u_j(\mathbf{x}, t). \tag{3.26}$$

We see directly from Eq. (3.25) that the macroscopic variable u_i can be calculated via

$$u_i(\mathbf{x}, t) = \frac{1}{h(\mathbf{x}, t)} \sum_\beta e_{\beta i} f_\beta^{eq}(\mathbf{x}, t) \tag{3.27}$$

and that the macroscopic variable h is found simply from Eq. (3.24).

By combining Eq. (3.23) and (3.24) we end up with

$$\begin{aligned} &A_0 + D_0 u_i u_i \\ &+4\bar{A} + \sum_{\beta=1,3,5,7} \bar{B} e_{\beta i} u_i + \sum_{\beta=1,3,5,7} \bar{C} e_{\beta i} e_{\beta j} u_i u_j + 4\bar{D} u_i u_i \\ &+4\tilde{A} + \sum_{\beta=2,4,6,8} \tilde{B} e_{\beta i} u_i + \sum_{\beta=2,4,6,8} \tilde{C} e_{\beta i} e_{\beta j} u_i u_j + 4\tilde{D} u_i u_i = h \end{aligned} \tag{3.28}$$

Evaluating this equation by inserting the particle velocities $\mathbf{e}_\beta$ from Eq. (3.14) while separating the results based on h and $u_i u_i$ results in

$$A_0 + 4\bar{A} + 4\tilde{A} = h \tag{3.29}$$

and

$$D_0 + 2e^2 \bar{C} + 4e^2 \tilde{C} + 4\bar{D} + 4\tilde{D} = 0 \tag{3.30}$$

Further, by inserting Eq. (3.23) into (3.25) we get

$$\begin{aligned} &A_0 e_{\beta i} + D_0 e_{\beta i} u_j u_j \\ &+ \sum_{\beta=1,3,5,7} (\bar{A} e_{\beta i} + \bar{B} e_{\beta i} e_{\beta j} u_i + \bar{C} e_{\beta i} e_{\beta j} e_{\beta k} u_i u_j + \bar{D} e_{\beta i} u_i u_i) \\ &+ \sum_{\beta=2,4,6,8} (\tilde{A} e_{\beta i} + \tilde{B} e_{\beta i} e_{\beta j} u_i + \tilde{C} e_{\beta i} e_{\beta j} e_{\beta k} u_i u_j + \tilde{D} e_{\beta i} u_i u_i) = h u_i, \end{aligned} \tag{3.31}$$

which we can separate based on u_i and Eq. (3.15) into

$$2e^2\bar{B} + 4e^2\tilde{B} = h. \tag{3.32}$$

If we then insert Eq. (3.23) into (3.26) we get

$$\begin{aligned}\sum_{\beta=1,3,5,7} &(\bar{A}e_{\beta i}e_{\beta j} + \bar{B}e_{\beta i}e_{\beta j}e_{\beta k}u_i + \bar{C}e_{\beta i}e_{\beta j}e_{\beta k}e_{\beta l}u_iu_j + \bar{D}e_{\beta i}e_{\beta j}u_iu_i)\\ + \sum_{\beta=2,4,6,8} &(\tilde{A}e_{\beta i}e_{\beta j} + \tilde{B}e_{\beta i}e_{\beta j}e_{\beta k}u_i + \tilde{C}e_{\beta i}e_{\beta j}e_{\beta k}e_{\beta l}u_iu_j + \tilde{D}e_{\beta i}e_{\beta j}u_iu_i)\\ &= \frac{1}{2}gh^2\delta_{ij} + hu_iu_j.\end{aligned} \tag{3.33}$$

To simplify this, we use Eq. (3.14) to get

$$\begin{aligned}&2\bar{A}e^2\delta_{ij} + 2\bar{C}e^4u_iu_i + 2\bar{D}e^2u_iu_i + 4\tilde{A}e^2\delta_{ij}\\ &+8\tilde{C}e^4u_iu_j + 4\tilde{C}e^4u_iu_i + 4\tilde{D}e^2u_iu_i = \frac{1}{2}gh^2\delta_{ij} + hu_iu_j,\end{aligned} \tag{3.34}$$

which can be separated to four different relations,

$$2e^2\bar{A}\ 4e^2\tilde{A} = \frac{1}{2}gh^2, \tag{3.35}$$

$$8e^4\tilde{C} = h, \tag{3.36}$$

$$2e^4\bar{C} = h \tag{3.37}$$

and

$$2e^2\bar{D} + 4e^2\tilde{D} + 4e^4\tilde{C} = 0. \tag{3.38}$$

From Eq. (3.36) and (3.37) it is obvious that

$$4\tilde{C} = \bar{C} \tag{3.39}$$

which implies, because of lattice symmetries, that we can reasonably assume

$$4\tilde{A} = \bar{A}, \qquad 4\tilde{B} = \bar{B} \qquad \text{and} \qquad 4\tilde{D} = \bar{D}. \tag{3.40}$$

Solving Eq. (3.29), (3.30), (3.32) and (3.35)-(3.38) with these assumptions results in

$$A_0 = h - \frac{5gh^2}{6e^2}, \qquad D_0 = -\frac{2h}{3e^2} \tag{3.41}$$

$$\bar{A} = \frac{gh^2}{6e^2}, \qquad \bar{B} = \frac{h}{3e^2}, \qquad \bar{C} = \frac{h}{2e^4}, \qquad \bar{D} = -\frac{h}{6e^2}, \tag{3.42}$$

$$\tilde{A} = \frac{gh^2}{24e^2}, \qquad \tilde{B} = \frac{h}{12e^2}, \qquad \tilde{C} = \frac{h}{8e^4}, \qquad \tilde{D} = -\frac{h}{24e^2}. \tag{3.43}$$

By inserting the above equations into Eq. (3.23) we get,

$$f_\beta^{eq} = \begin{cases} h - \frac{5gh^2}{6e^2} - \frac{2h}{3e^2}u_iu_i, & \beta = 0,\\ \frac{gh^2}{6e^2} + \frac{h}{3e^2}e_{\beta i}u_i + \frac{h}{2e^4}e_{\beta i}e_{\beta j}u_iu_j - \frac{h}{6e^2}u_iu_i, & \beta = 1,3,5,7,\\ \frac{gh^2}{24e^2} + \frac{h}{12e^2}e_{\beta i}u_i + \frac{h}{8e^4}e_{\beta i}e_{\beta j}u_iu_j - \frac{h}{24e^2}u_iu_i, & \beta = 2,4,6,8, \end{cases} \tag{3.44}$$

which is the equilibrium distribution function used when solving SWEs.

3.6 Recovery of the Shallow Water Equations

To prove that the macroscopic variables found in Eq. (3.24) and (3.27) for velocities and height are in fact the solutions to the SWEs, we use a Chapman-Enskog expansion of LBE. Our goal is to recover the SWEs Eq, (2.47) and (2.48). We start by expressing Eq. (3.20) as

$$f_\beta(\mathbf{x}+\mathbf{e}_\beta\epsilon, t+\epsilon) - f_\beta(\mathbf{x},t) = -\frac{1}{\tau}(f_\beta(\mathbf{x},t) - f_\beta^{eq}(\mathbf{x},t)) + \frac{\epsilon}{6e^2}e_{\beta i}F_i(\mathbf{x},t) \tag{3.45}$$

where it is assumed $\Delta t = \epsilon$ and that ϵ is small.

Taylor expansion around $(\mathbf{x}, t)$ for the first term on the left hand side results in

$$\begin{aligned}\epsilon\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)f_\beta + \frac{1}{2}\epsilon^2\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)^2 f_\beta + O(\epsilon^3) \\ = -\frac{1}{\tau}\left(f_\beta - f_\beta^{(0)}\right) + \frac{\epsilon}{6e^2}e_{\beta j}F_j.\end{aligned} \tag{3.46}$$

We now expand f_β around f_β^{eq} while setting $f_\beta^{(0)} = f_\beta^{eq}$, getting

$$f_\beta = f_\beta^{(0)} + \epsilon f_\beta^{(1)} + \epsilon^2 f_\beta^{(2)} + O(\epsilon^3). \tag{3.47}$$

Inserting Eq. (3.47) into (3.46) gives

$$\begin{aligned}\epsilon\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)\left(f_\beta^{(0)} + \epsilon f_\beta^{(1)} + \epsilon^2 f_\beta^{(2)} + O(\epsilon^3)\right) + \\ \frac{1}{2}\epsilon^2\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)^2\left(f_\beta^{(0)} + \epsilon f_\beta^{(1)} + \epsilon^2 f_\beta^{(2)} + O(\epsilon^3)\right) + O(\epsilon^3) \\ = -\frac{1}{\tau}\left(\epsilon f_\beta^{(1)} + \epsilon^2 f_\beta^{(2)} + O(\epsilon^3)\right) + \frac{\epsilon}{6e^2}e_{\beta j}F_j.\end{aligned} \tag{3.48}$$

When we separate this equation with respect to ϵ and ϵ^2 while ignoring all ϵ^3 terms we get

$$\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)f_\beta^{(0)} = -\frac{1}{\tau}f_\beta^{(1)} + \frac{1}{6e^2}e_{\beta j}F_j \tag{3.49}$$

and

$$\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)f_\beta^{(1)} + \frac{1}{2}\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)^2 f_\beta^{(0)} = -\frac{1}{\tau}f_\beta^{(2)}. \tag{3.50}$$

When we now insert Eq. (3.49) into (3.50) we end up with

$$\left(1 - \frac{1}{2\tau}\right)\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)f_\beta^{(1)} = -\frac{1}{\tau}f_\beta^{(2)} - \frac{1}{2}\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)\frac{1}{6e^2}e_{\beta k}F_k \tag{3.51}$$

If we now calculate the sum over Eq. (3.49) and ϵ multiplied with Eq. (3.51) we get

$$\frac{\partial}{\partial t}\left(\sum_\beta f_\beta^{(0)}\right) + \frac{\partial}{\partial x_j}\left(\sum_\beta e_{\beta j}f_\beta^{(0)}\right) = -\frac{\epsilon}{12e^2}\frac{\partial}{\partial x_j}\left(\sum_\beta e_{\beta j}e_{\beta k}F_k\right). \tag{3.52}$$

When solving this equation using Eq. (3.44) and (3.14), and applying first order accuracy for the force term, we get

$$\frac{\partial h}{\partial t} + \frac{\partial h u_j}{\partial x_j} = 0, \tag{3.53}$$

which is the continuity SWE, (2.47), derived in section (2.3).

To find the momentum equation for SWE, we start by taking $\sum e_{\beta i}\,[3.49 + \epsilon \times 3.51]$ which is

$$\begin{aligned} &\frac{\partial}{\partial t}\left(\sum_\beta e_{\beta i} f_\beta^{(0)}\right) + \frac{\partial}{\partial x_j}\left(\sum_\beta e_{\beta i} e_{\beta j} f_\beta^{(0)}\right) \\ &\qquad + \epsilon\left(1 - \frac{1}{2\tau}\right)\frac{\partial}{\partial x_j}\left(\sum_\beta e_{\beta i} e_{\beta j} f_\beta^{(1)}\right) \\ &= F_j \delta_{ij} - \epsilon \frac{1}{2}\sum_\beta e_{\beta i}\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)\frac{1}{6e^2} e_{\beta j} F_j \end{aligned} \tag{3.54}$$

If we simplify this equation the same way as we did when going from Eq. (3.52) to (3.53), we end up with

$$\frac{\partial (h u_i)}{\partial t} + \frac{\partial (h u_i u_j)}{\partial x_j} = -\frac{g}{2}\frac{\partial (h^2)}{\partial x_i} - \frac{\partial \Lambda_{ij}}{\partial x_{ij}} + F_i. \tag{3.55}$$

Here Λ_{ij} is defined as

$$\Lambda_{ij} = \frac{\epsilon}{2\tau}(2\tau - 1)\sum_\beta e_{\beta i} e_{\beta j} f_\beta^{(1)}, \tag{3.56}$$

which with Eq. (3.44) and (3.14), some algebra and referencing Eq. (3.49) can be simplified to

$$\Lambda_{ij} \approx -\nu\left[\frac{\partial (h u_i)}{\partial x_j} + \frac{\partial (h u_j)}{\partial x_i}\right]. \tag{3.57}$$

By inserting this expression into (3.55) we recover Eq. (2.48), the momentum equation for SWE,

$$\frac{\partial h u_j}{\partial t} + \frac{\partial h u_i u_j}{\partial x_j} = -g\frac{\partial}{\partial x_i}\left(\frac{h^2}{2}\right) + \nu\frac{\partial^2 (h u_i)}{\partial x_j \partial x_j} + F_i. \tag{3.58}$$

It is also worth noting that the kinematic viscosity is defined as by

$$\nu = \frac{e^2 \Delta t}{6}(2\tau - 1). \tag{3.59}$$

As we used only the first order force term to derive SWE from LBE the solution is only accurate to first order in time.

3.7 Stability

The Lattice Boltzmann method is, like all numerical methods, stable under some conditions and unstable under others. While the theoretical conditions are not entirely established, a few general conditions are known. When simulating actual water it is necessary

to simulate some kind of diffusion. Thus the kinematic viscosity can never have a negative value, i.e.

$$\nu = \frac{e^2 \Delta t}{6}(2\tau - 1) > 0. \tag{3.60}$$

As a result the single relaxation constant must be

$$\tau > \frac{1}{2}. \tag{3.61}$$

Another condition that limits the LBM is that streaming only goes between nodes immediately beside each other. This limits the velocity to values smaller than the lattice propagation (size divided by the time step used),

$$\frac{u_j u_j}{e^2} < 1 \qquad \text{given} \qquad e = \frac{\Delta x}{\Delta t}. \tag{3.62}$$

All shallow water flows have a wave speed of $\sqrt{gh}$ which also cannot be faster than the propagation of information in the lattice, hence

$$\frac{gh}{e^2} < 1. \tag{3.63}$$

Inserting Eq. (3.62) into (3.63) and finding the square root of the result lets us find

$$F_r = \frac{\sqrt{u_j u_j}}{\sqrt{gh}} < 1, \tag{3.64}$$

where F_r is called the Froude number. Froude number is a tool to see if we are dealing with a supercritical flow ($F_r > 1$), a critical flow ($F_r = 1$) or a subcritical flow ($F_r < 1$). Hence LBM is only usable for subcritical flows. The LBM is mostly of use in coastal, river or channel systems where low Froude numbers are the norm allowing this condition to mostly be satisfied. The conditions in Eq. (3.60) to (3.62) on the other hand can all be satisfied with appropriate choice of variables.

3.8 Force Terms

To be able to accurately simulate any actual flow it is important to be able to calculate the force terms. Without force terms one can no longer accurately simulate systems that contain friction with the ground, seabed with any inclination, wind shear effect on water, or that contain the Coriolis effects. In this thesis we have ignored both wind shear stress and the Coriolis effects as they are very dependent on system location and was deemed not relevant.

The force term as seen in Eq. (2.49) is

$$F_i = -gh\frac{\partial z_b}{\partial x_i} + \frac{\tau_{wi}}{\rho} - \frac{\tau_{bi}}{\rho} + \Omega_i. \tag{3.65}$$

When calculating this, the first term on the right hand side contain a derivative. This is not practical for efficient computing and as such one other way for calculating this was developed by Zhou [27],

$$\frac{g\bar{h}}{6e^2}\left[z_b(\mathbf{x} + \mathbf{e}_\beta \Delta t) - z_b(\mathbf{x})\right] = gh\frac{\partial z_b}{\partial x_i}. \tag{3.66}$$

Which means the LBE Eq. (3.20), now can be written as,

$$\begin{aligned} f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t) = f_\beta(\mathbf{x},t) - \frac{1}{\tau}(f_\beta(\mathbf{x},t) - f_\beta^{eq}(\mathbf{x},t)) \\ -\frac{g\bar{h}}{6e^2}[z_b(\mathbf{x}+\mathbf{e}_\beta\Delta t) - z_b(\mathbf{x})] + \frac{\Delta t}{6e^2} e_{\beta i} F_i(\mathbf{x},t), \end{aligned} \tag{3.67}$$

where $\bar{h} = \frac{1}{2}[h(\mathbf{x},t) + h(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t)]$ and the force term is redefined to

$$F_i = \frac{\tau_{wi}}{\rho} - \frac{\tau_{bi}}{\rho} + \Omega_i. \tag{3.68}$$

To make sure this is accurate, we perform a Chapman-Enskog analysis of Eq. (3.67) where we start by assuming $\Delta t = \epsilon$ and that it is small. Thus we get

$$\begin{aligned} f_\beta(\mathbf{x}+\mathbf{e}_\beta\epsilon, t+\epsilon) = f_\beta(\mathbf{x},t) - \frac{1}{\tau}(f_\beta - f_\beta^{eq}) \\ -\frac{g\bar{h}}{6e^2}[z_b(\mathbf{x}+\mathbf{e}_\beta\Delta t) - z_b(\mathbf{x})] + \frac{\epsilon}{6e^2} e_{\beta i} F_i, \end{aligned} \tag{3.69}$$

which allows us to use Taylor expansion in time and space around $(\mathbf{x}, t)$, on the left hand side which turns into

$$\epsilon\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right) f_\beta + \frac{1}{2}\epsilon^2\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right)^2 f_\beta + O(\epsilon^3) \tag{3.70}$$

Using the Chapman-Enskog procedure allows us to expand f_β around $f_\beta^{(0)} = f_\beta^{eq}$, getting Eq. (3.47). By Taylor expanding the third term in Eq. (3.69) we find that it becomes

$$\frac{g}{6e^2}\left[h + \frac{\epsilon}{2}\left(\frac{\partial h}{\partial t} + \mathbf{e}_{\beta j}\frac{\partial h}{\partial x_j}\right)\right]\left(\epsilon\mathbf{e}_{\beta j}\frac{\partial z_b}{\partial x_j} + \frac{\epsilon^2}{2}\mathbf{e}_{\beta i}\mathbf{e}_{\beta j}\frac{\partial^2 z_b}{\partial x_i \partial x_j}\right) + O(\epsilon^3). \tag{3.71}$$

The last term in Eq. (3.69), the force term, should be used with the centered scheme, as Zhou proved it is second order accurate [10],

$$F_j = F_j\left(\mathbf{x} + \frac{1}{2}\mathbf{e}_\beta\epsilon, t + \frac{1}{2}\epsilon\right). \tag{3.72}$$

This again can be Taylor expanded into

$$F_j + \frac{\epsilon}{2}\left(\frac{\partial F_j}{\partial t} + \mathbf{e}_{\beta i}\frac{\partial F_j}{\partial x_i}\right) + O(\epsilon^3). \tag{3.73}$$

If we now insert Eq. (3.47), (3.70), (3.71) and (3.73) into Eq. (3.69) we end up with

$$\left(\frac{\partial}{\partial t} + e_{\beta j}\frac{\partial}{\partial x_j}\right) f_\beta^{(0)} = -\frac{f_\beta^{(1)}}{\tau} - \frac{gh\mathbf{e}_{\beta j}}{6e^2}\frac{\partial z_b}{\partial x_j} + \frac{\mathbf{e}_{\beta j}F_j}{6e^2}, \tag{3.74}$$

to order ϵ^1 and

$$\begin{aligned}\left(\frac{\partial}{\partial t}+e_{\beta j}\frac{\partial}{\partial x_j}\right)f_\beta^{(1)}+\frac{1}{2}\left(\frac{\partial}{\partial t}+e_{\beta j}\frac{\partial}{\partial x_j}\right)^2 f_\beta^{(0)}=\\ -\frac{f_\beta^{(2)}}{\tau}-\frac{g\mathbf{e}_{\beta j}}{12e^2}\left(\frac{\partial h}{\partial t}+e_{\beta j}\frac{\partial h}{\partial x_i}\right)\frac{\partial z_b}{\partial x_j}\\ -\frac{gh\mathbf{e}_{\beta i}\mathbf{e}_{\beta j}}{12e^2}\frac{\partial^2 z_b}{\partial x_i\partial x_j}+\frac{\mathbf{e}_{\beta j}}{12e^2}\left(\frac{\partial F_j}{\partial t}+e_{\beta i}\frac{\partial F_j}{\partial x_i}\right),\end{aligned} \tag{3.75}$$

to order ϵ^2. If we now insert Eq. (3.74) into (3.75) we get the more compact expression

$$\left(1-\frac{1}{2\tau}\right)\left(\frac{\partial}{\partial t}+e_{\beta j}\frac{\partial}{\partial x_j}\right)f_\beta^{(1)}=-\frac{f_\beta^{(2)}}{\tau}. \tag{3.76}$$

By evaluating $\sum_\beta[3.74+\epsilon\times 3.76]$ we find

$$\frac{\partial}{\partial t}\sum_\beta f_\beta^{(0)}+\frac{\partial}{\partial x_j}\sum_\beta \mathbf{e}_{\beta j}f_\beta^{(0)}=0, \tag{3.77}$$

which with the equilibrium function from Eq. (3.44) easily reduces to the continuum equation of the SWE, Eq. (2.47), and is accurate to second order.

If we now evaluate $\sum_\beta \mathbf{e}_{\beta i}[3.74+\epsilon\times 3.76]$ we end up with

$$\begin{aligned}\frac{\partial}{\partial t}\sum_\beta \mathbf{e}_{\beta i}f_\beta^{(0)}+\frac{\partial}{\partial x_j}\sum_\beta \mathbf{e}_{\beta i}\mathbf{e}_{\beta j}f_\beta^{(0)}\\ +\epsilon\left(1-\frac{1}{2\tau}\right)\frac{\partial}{\partial x_j}\sum_\beta \mathbf{e}_{\beta i}\mathbf{e}_{\beta j}f_\beta^{(1)}=-gh\frac{\partial z_b}{\partial x_i}+F_i,\end{aligned} \tag{3.78}$$

which with the equilibrium function from Eq. (3.44) can be rewritten into the momentum equation of the SWE, Eq. (2.48), and is also accurate to the second order.

Thus we know that avoiding the differentiation when there are forces resulting from elevation differences of the seabed is both possible and second order accurate. In practice this is of no use considering it is implicit in time, however a semi-implicit form,

$$\bar{h}=\frac{1}{2}[h(\mathbf{x},t)+h(\mathbf{x}+\mathbf{e}_\beta\Delta t,t)], \tag{3.79}$$

has shown to provide accurate solutions [27]. This form is numerically quick and easy to use and thus is for all intent preferred in practice.

Tests during this thesis have shown that writing the LBE on the form of

$$\begin{aligned}f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t,t+\Delta t)=f_\beta(\mathbf{x},t)-\frac{1}{\tau}(f_\beta-f_\beta^{eq})\\ -\frac{g\bar{h}P_\beta}{6e^2}[z_b(\mathbf{x}+\mathbf{e}_\beta\Delta t)-z_b(\mathbf{x})]+\frac{\Delta t P_\beta}{3e^2}e_{\beta i}F_i,\end{aligned} \tag{3.80}$$

leads to stable and accurate results given

$$P_\beta = \begin{cases} 0, & \beta = 0, \\ 1, & \beta = 1,3,5,7, \\ \frac{1}{4}, & \beta = 2,4,6,8. \end{cases} \tag{3.81}$$

To calculate a more realistic flow when using the no-slip BC mentioned in section 3.10 we can include an extra friction vector in the force term,

$$F_i = \frac{\tau_{wi}}{\rho} - \frac{\tau_{bi}}{\rho} - \frac{\tau_{fi}}{\rho} + \Omega_i, \tag{3.82}$$

where

$$\tau_{fi} = -\rho C_f |\mathbf{V}_\tau| \mathbf{V}_\tau. \tag{3.83}$$

$\mathbf{V}_\tau$ is the velocity parallel to the wall and since there should be no velocity component normal to the wall we can write

$$\mathbf{V}_\tau \approx \mathbf{V}. \tag{3.84}$$

C_f is found by Manning's equation to be

$$C_f = g \frac{n_f^2}{h_f^{\frac{1}{3}}}, \tag{3.85}$$

where n_f is Manning's coefficient of the wall and h_f is the distance from wall to node. Thus wall friction is

$$\tau_{fi} = -\rho g \frac{n_f^2}{h_f^{\frac{1}{3}}} u_i \sqrt{u_j u_j}. \tag{3.86}$$

3.9 Wet and dry boundary

Until recently, LBM has had issues with computing how and when a dry node becomes wet as a result of the flow developing to cover dry areas. In a paper [28], Liu and Zhou derive how they accomplished this by use of Chapman-Enskog analysis and Taylor expansion. They find that the unknown distribution functions at dry nodes that become wet can be calculated from

$$\begin{aligned} f_\beta = -\frac{gh\tau P_\beta}{6e^2}[z_b(\mathbf{x} + \mathbf{e}_\beta \Delta t) - z_b(\mathbf{x})] - \frac{\Delta t \tau P_\beta}{3e^2} \mathbf{e}_{\beta i} C_b u_i \sqrt{u_j u_j} \\ -\tau \left(f_\beta^{eq}(\mathbf{x} + \mathbf{e}_\beta \Delta t) - f_\beta^{eq}(\mathbf{x}) \right), \end{aligned} \tag{3.87}$$

given the particle direction is toward a wet node. If the direction is toward a dry node or is $\beta = 0$ the distribution function becomes the average of neighbouring nodes

$$f_\beta = \frac{1}{8} \sum_{\alpha=1}^{8} f_\beta(\mathbf{x} + \mathbf{e}_\beta \Delta t). \tag{3.88}$$

3.10 Boundary Conditions

One of the reasons for choosing D2Q9 model over the hexagonal 7-speed model was its simplicity in dealing with BC. Using the correct BC easily influences the accuracy of the solutions acquired as can be seen in Zou's article [29].

The BC used in this thesis was proposed by Zhou in [10]. When dealing with a solid boundary, like going from wet to dry node, no-slip or slip scheme was used. The no-slip BC works by implementing a full bounce-back scheme where the edge between wet and dry is in the middle of the two nodes.

The slip condition, on the other hand, works by implementing a zero gradient for the distribution function across the boundary. Thus we achieve that all flow goes along the boundary and not across it.

When dealing with an open boundary, that is a flow between the outside of the simulation area and the inside of the simulation area, there are two ways to implement BC.

If the velocity and height of the flow are known, we can calculate the distribution functions streamed in from the boundary directly. From [30], we find that given open boundary on the left side of the simulation area, the distribution functions become

$$\begin{aligned} f_1 &= f_5 + \frac{2h\mathbf{u}}{3e}, \\ f_2 &= f_6 + \frac{h\mathbf{u}}{6e} + \frac{h\mathbf{v}}{2e} + \frac{f_7 - f_3}{2}, \\ f_8 &= f_4 + \frac{h\mathbf{u}}{6e} - \frac{h\mathbf{v}}{2e} + \frac{f_3 - f_7}{2}, \end{aligned} \tag{3.89}$$

where the choice of f_β is decided from Fig. 3.2.

The other way to implement an open boundary is to assume that nothing changes outside of the boundary. Then we can simply assume

$$\begin{aligned} f_1(\mathbf{x}) &= f_1(\mathbf{x} + \mathbf{e_1}\Delta t), \\ f_2(\mathbf{x}) &= f_2(\mathbf{x} + \mathbf{e_2}\Delta t), \\ f_8(\mathbf{x}) &= f_8(\mathbf{x} + \mathbf{e_8}\Delta t), \end{aligned} \tag{3.90}$$

which is very fast and simple to implement.

3.11 Algorithm

The core routine the simulator goes through can be visualized in Fig. 3.3. The procedure can be described as following:

1. It starts by generating input by defining the macroscopic variables h and $\mathbf{U}$. From these it can then generate the equilibrium distribution function from Eq. (3.44) before it sets $f_\beta = f_\beta^{eq}$. This way to generate the particle distribution function has been shown to be very efficient.

2. Creates/updates controls that know where the dry and wet areas are.

3. Calculates body forces for each node.

4. Calculates collision from Eq. (3.80).

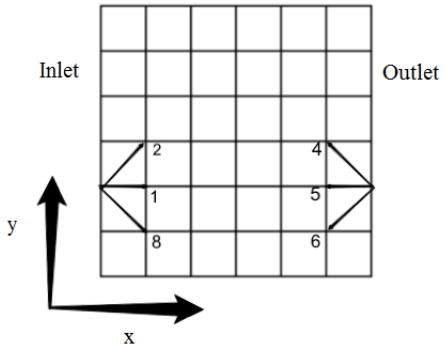


Figure 3.2: Figure of lattice showing unknown distribution functions when dealing with open boundary.

5. Checks if the collision will cause some dry nodes to become wet.
6. Performs the streaming as seen in Eq. (3.80).
7. Calculates f_β for the unknown f_β if nodes become wet, then updates unknown f_β in the newly wet node. If nodes did not become wet, applies proper boundary condition.
8. Finds new h and $\mathbf{U}$ from Eq. (3.24) and (3.27).
9. Computes f_β^{eq} again from Eq. (3.44).
10. Returns to step 2 until solution is found.

At the time of writing this thesis, the simulator is written as a single thread. However, parallelization of every component of the method is possible without much issue. If the author had managed to solve every issue faced in this thesis, the final step would have been to make the code parallel and as such increase the efficiency of the code.

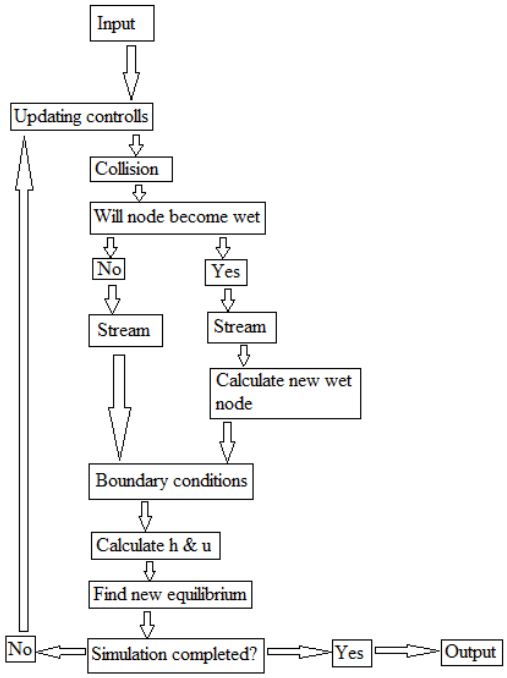


Figure 3.3: Flowchart of main components of the simulation algorithm.

4 Results

In this chapter we will present the results of simulating several test cases that are designed to portray the accuracy of the different terms of the lattice Boltzmann method. By doing these tests, we aim to find out which parts of the simulating module are running correctly and which parts need further improvement before it can be used as a universal steady state lattice Boltzmann simulator.

4.1 Still water over flat seabed

In this test we wish to show that the solver is fundamentally stable. By choosing to simulate still water over a flat seabed we can easily establish whether the governing LBM is stable without interference from any force terms. As can be seen from Fig. 4.1, the flow is stable over the region. By observing Fig. 4.2, we can also establish that the method does not generate artificial currents. Considering that the total velocity $U = \sqrt{u_x^2 + u_y^2}$ is less than the machine precision, this is a good conclusion.

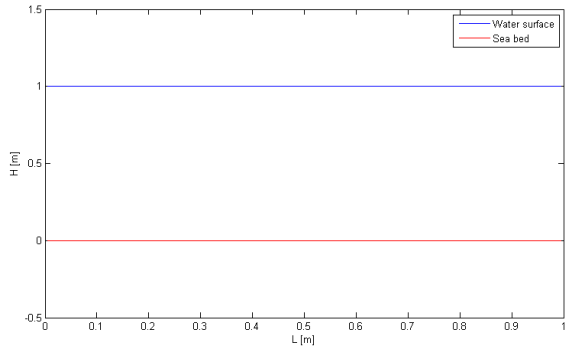


Figure 4.1: Cross section of still water over flat seabed, portraying altitude.

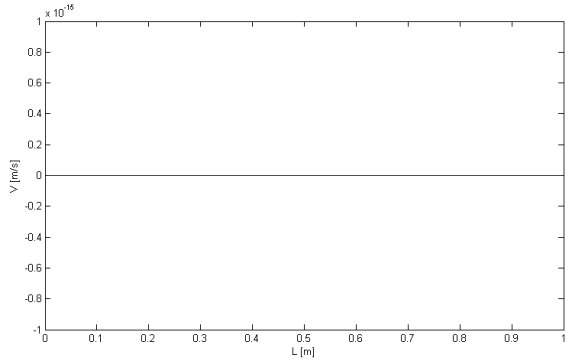


Figure 4.2: Cross section of still water over flat seabed, portraying total velocity.

4.2 Still water over regular bed

In this test, we have still water resting over a seabed with a bump in it. The goal of this test is to showcase the validity of the term that applies seabed changes to the LBM. In Fig. 4.3 we have a two-dimensional surface that can be projected down to one dimension showing the flat water surface resting over the seabed. Had there been any issues with the weighting of different terms against the changes of the seabed, it would have been readily apparent here. For further confirmation, we see that the total velocity of the flow, observed in Fig. 4.4, is still in the same magnitude as the machine precision.

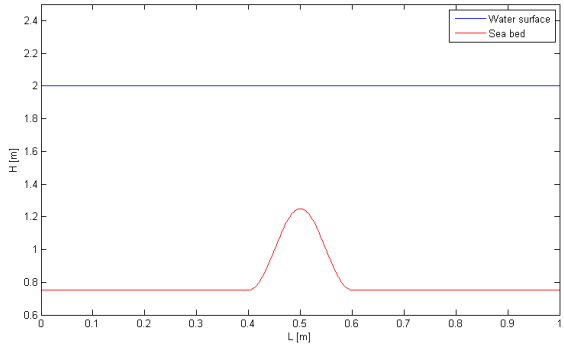


Figure 4.3: Cross section of still water over bumpy seabed, portraying altitude.

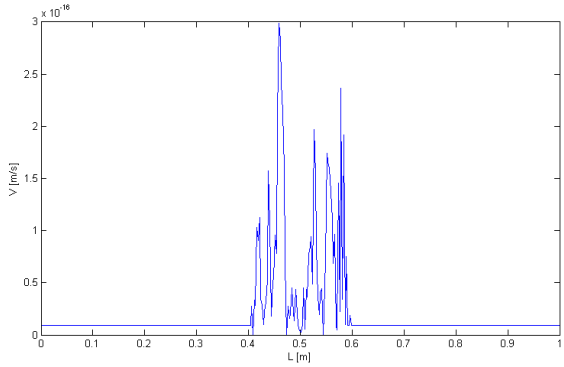


Figure 4.4: Cross section of still water over bumpy seabed, portraying total velocity.

4.3 Steady flow over a bump

With confirmation that the basic algorithm is stable, the next test is to make sure it is stable with some initial velocity in the system. When a stable subcritical flow goes over a bump, the surface will drop in the area above the bump. As all flows must stay subcritical for LBM to stay stable, this was not an issue. The simulated flow is, for all intents and

purposes, a one-dimensional flow, which can be achieved with slip BC on the northern and southern sides of the simulated areas, while keeping the western and eastern open.

In this test we have used no friction, as the analytical answer is derived for a frictionless flow. The seabed, z_b is defined as

$$z_b(x) = \begin{cases} 0.2 - 0.05(x-10)^2, & \text{when } 8 < x < 12, \\ 0, & \text{otherwise,} \end{cases} \tag{4.1}$$

while the rest of the initial conditions are

$$u_{\text{initial}}(x) = 2.21m/s \qquad h_{\text{initial}} = 2m - z_b(x). \tag{4.2}$$

We used a $25m$ long simulation area with $lx = 500$ (number of nodes in x direction), $\tau = 1.5$ and $e = 15m/s$ when performing this simulation.

This is a common test for new simulators, as it has a distinct shape and a clearly defined analytic altitude over the bump. The shape can be observed in Fig. 4.5 where the flow surface has dropped down to $h = 1.7074m$. Compared to the analytical solution [31], the water depth has a relative error of 0.0029% at the lowest point.

In the initial conditions, we had decided on an initial velocity and water depth. This means we have an initial discharge, $Q = 4.42m^3/s$ since discharge is volume water per second. Theoretically this value should stay constant, which is what we experienced in Fig. 4.6.

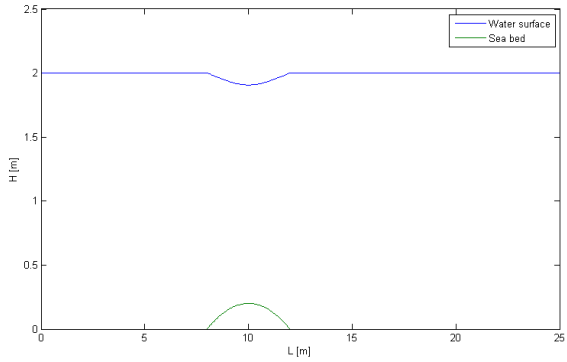


Figure 4.5: Cross section of steady flow over bump, portraying altitude.

4.4 Tidal flow over uneven bed

Here we consider a one-dimensional tidal flow where the water comes in from the left side and acts as if the right side is closed. This is fairly similar to something that can happen in coastal regions even though we have ignored friction with the seabed. The test case was used by Bermudez and Vázquez [1] to verify their solution of the bed slope source term by use of upwind discretization. In our case we use it to make sure the test can handle large areas of changing currents and that the currents generated by water surface altitude

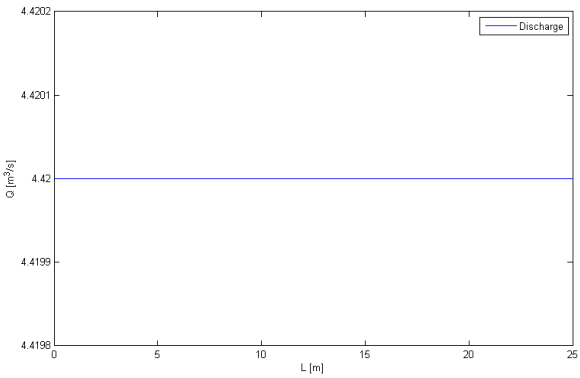


Figure 4.6: Cross section of steady flow over bump, portraying discharge over the area.

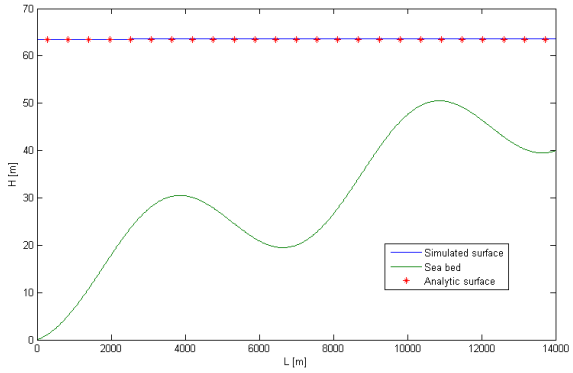


Figure 4.7: Analytic and simulated free surface of tidal flow at $t = 9117.5s$.

difference are computed properly. This is also another test to verify the implementation of bed slope source term in the simulator.

The topography of the test can be observed in Fig. 4.7 and is defined as

$$H(x) = 50.5 - \frac{40x}{L} - 10\sin\left[\pi\left(\frac{4x}{L} - \frac{1}{2}\right)\right], \tag{4.3}$$

where $L = 14000m$ is the length of the simulation system. $H(0)$ is the surface height of the flow over a reference level. Since the water is level at the start of the system, the seabed is defined as

$$z_b = H(0) - H(x), \tag{4.4}$$

and the initial flow surface has a height and velocity of

$$h_{\text{initial}}(x) = H(x) \qquad u_{\text{initial}}(x) = 0. \tag{4.5}$$

As mentioned above, the right side of the simulated area has a closed boundary forcing $u(L,t) = 0m/s$. To simulate the proper tidal flow we include an open boundary with a forced surface height of

$$h(0,t) = H(0) + 4 - 4\sin\left[\pi\left(\frac{4t}{86400} + \frac{1}{2}\right)\right], \tag{4.6}$$

which simulates a 24 hour tidal wave. Under these conditions, the analytical solution [1] is found to be

$$h(x,t) = H(x) + 4 - 4\sin\left[\pi\left(\frac{4t}{86400} + \frac{1}{2}\right)\right] \tag{4.7}$$

and

$$u(x,t) = \frac{\pi(x-L)}{5400h(x,t)}\cos\left[\pi\left(\frac{4t}{86400} + \frac{1}{2}\right)\right]. \tag{4.8}$$

After simulation to $t = 9117.5s$ we find the flow surface as seen in Fig. 4.7, which has a maximum relative error of 0.17% as can be observed in Fig. 4.8. In the simulation, we used the parameters $e = 200m/s$, $lx = 800$ and $\tau = 0.6$ as Zhou [10] suggested.

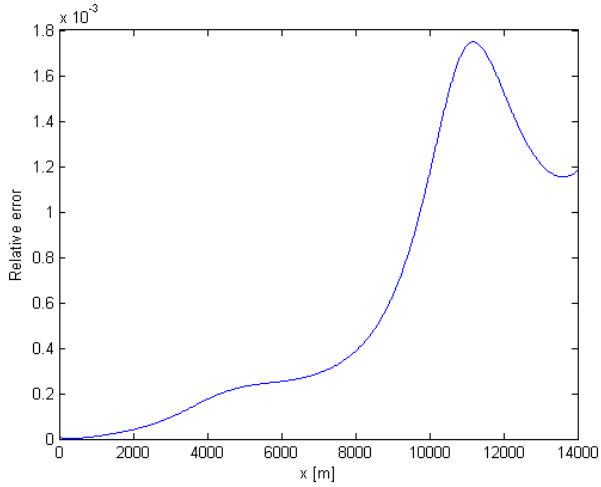


Figure 4.8: Relative error of the free surface of tidal flow at $t = 9117.5s$.

The simulated velocity at that time can be seen in Fig. 4.9 which follows very closely to the analytic velocity. The maximum relative error measured for the velocity is 0.26% for $x < 13116m$. As the boundary condition $u(L,t) = 0$ forces the velocity to go toward zero, the relative error will rapidly increase, making relative error not usable as a measurement of accuracy. The absolute error however, is at no point higher than $0.00021m/s$ at $x > 13116m$, showing that the velocity is highly accurate.

The calculation of the accuracy of the discharge also suffers from the same problem as the velocity. The discharge is highly accurate as observed in Fig. 4.10, with maximum relative error of 0.43% for $x < 13116m$, and maximum absolute error of $0.0049m^3/s$ for $x > 13116m$.

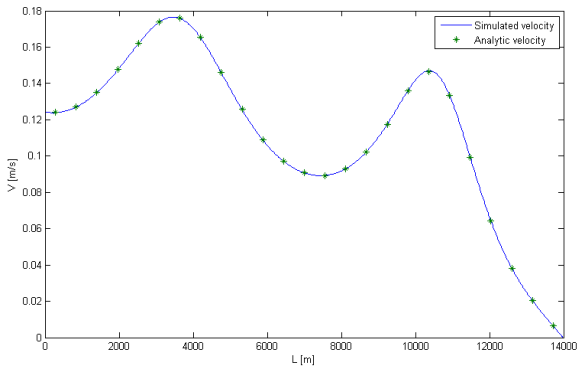


Figure 4.9: Analytic and simulated flow velocity of tidal flow at $t = 9117.5s$.

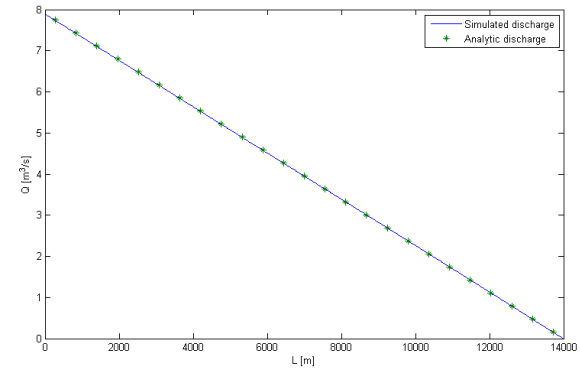


Figure 4.10: Analytic and simulated discharge of tidal flow at $t = 9117.5s$.

4.5 Still water over two-dimensional bump

In this test we check if any flow velocities are generated in a system where still water rests over a two-dimensional bump. The bump is defined as

$$z_b(x,y) = \begin{cases} 0.2\exp\left[-25(x-1)^2 - 50(y-0.5)^2\right], & \frac{1}{2} < x < \frac{3}{2}, \frac{1}{4} < y < \frac{3}{4} \\ 0, & \text{otherwise.} \end{cases} \tag{4.9}$$

Considering all previous tests have shown good accuracy in one dimension, this test is designed to show if the basic algorithm and bed slope term are accurate in two dimensions. The fluid surface is resting at $h = 2 - z_b$ and the parameters used is $lx = 200$, $ly = 100$, $\tau = 1.1$, $e = 15m/s$ with a two meter long and one meter wide simulation area.

The resulting fluid surface after simulation is resting a distance less than computer precision from the theoretical height. A cross section of the surface in one dimension can be observed in Fig. 4.11.

Similarly, the total velocity generated by the system is less than the computer precision

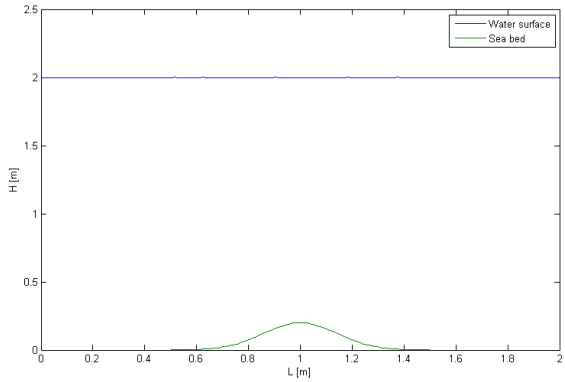


Figure 4.11: Cross section of simulated surface resting over a two-dimensional bump. The graph follows along the $y = 0.5m$ line

and can thus be safely ignored. A cross section of the velocity in one dimension can be observed in Fig. 4.12.

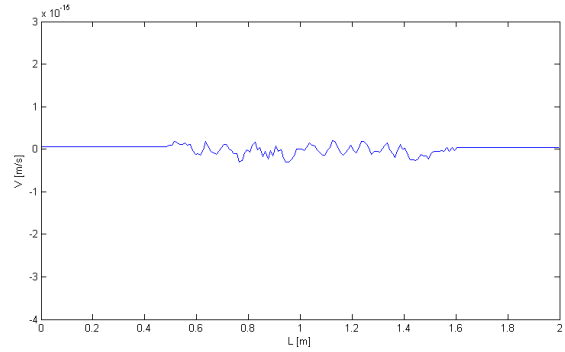


Figure 4.12: Cross section of simulated fluid velocity over a two-dimensional bump. The graph follows along the $y = 0.5m$ line

4.6 Steady flow down a slope

In this test we have a one-dimensional flow streaming down a slope. This is a typical example of an open channel flow where we can calculate the stable velocity of the flow based on the Manning's coefficient, water depth and slope rate. The goal is to let the flow stabilize around a velocity where the bed friction and gravity assisted acceleration down the slope reaches an equilibrium. To achieve this, we use $h = 1m$, $n = 0.013s/m^{\frac{1}{3}}$ and a slope rate of $0.001m/m$. The seabed and the simulated water depth can be observed in Fig. 4.13. The maximum relative error of the water depth was measured to $3.82 \times 10^{-12}\%$.

From Dr. Xing Fang's calculator [32], we find that the analytic velocity should stabilize

at $u = 2.4325m/s$. We see in Fig. 4.14 that the simulated velocity is very close to the theoretical velocity, which is calculated to have a maximum relative error of $9.52 \times 10^{-5}\%$.

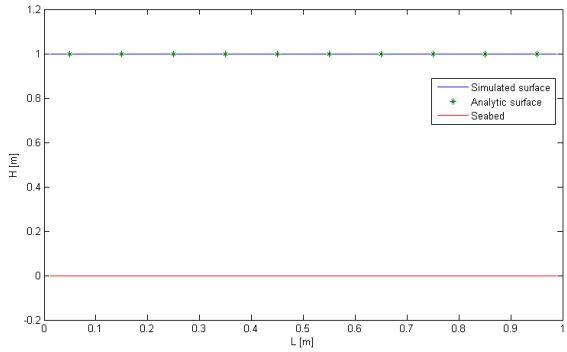


Figure 4.13: Water flowing down a slope with a slope rate of $0.001m/m$.

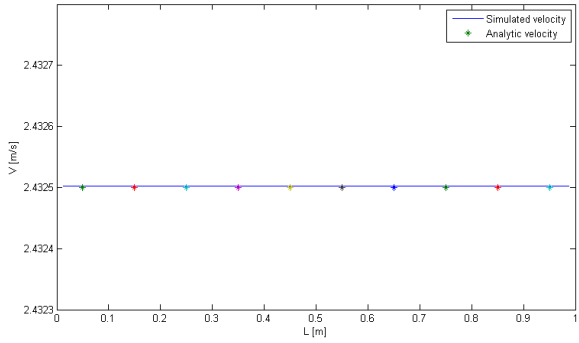


Figure 4.14: Simulated and analytic steady flow velocity of water flowing down a slope.

4.7 Steady flow down a two-dimensional slope

This test is a repeat of the previous test, however, this time it is computed for a two-dimensional surface. This slope is the same slope as the previous, although this time, it is turned 45°. By utilizing the same surface as previously, we already know the theoretically accurate flow velocity, giving us a clear target to aim for. From Fig. 4.15 we see that the calculated velocity, $u_{\text{total}} = 2.0482m/s$, is lower than the theoretical velocity. The maximum relative error in the velocity was calculated to 15.80%.

4.8 Sloshing

This test is created to test the implementation of wetting and drying of nodes. Since the LBM can only directly handle wet nodes, it is important to check that the boundary

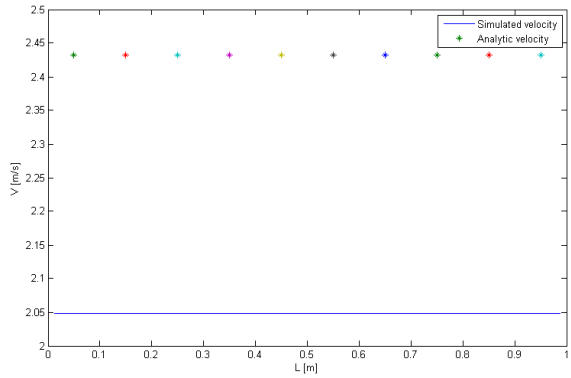


Figure 4.15: Simulated and analytic steady flow total velocity of water flowing down a two-dimensional slope.

between dry and wet nodes is handled correctly. In this test we recreate a scheme where there is a perturbed flow in a parabolic one-dimensional container. This was used by Sampson *et al.* in [33]. The bed topography is defined by

$$z_b(x) = h_0 \left(\frac{x^2}{a^2} \right), \tag{4.10}$$

with $h_0 = 10m$ and $a = 3000m$ chosen as constants. The bed friction coefficient is set to

$$C_b = \frac{h\tau_b}{|u|}, \tag{4.11}$$

where $\tau_b = 0.001s^{-1}$. To calculate the analytic water surface we need $p = \sqrt{8gh_0}/a$, which is the hump amplitude parameter. The hump amplitude parameter is then used in $s = \sqrt{p^2 - \tau_b^2}/2$. This is then used to calculate the analytic water surface [33],

$$\begin{aligned} h(x,t) = h_0 + \frac{a^2 B^2 \exp(-\tau_b t)}{8g^2 h_0} \left(-s\tau_b \sin(2st) + \left(\frac{\tau_b^2}{4} - s^2 \right) \cos(2st) \right) \\ - \frac{B^2 \exp(\tau_b t)}{4g} - \frac{\exp(\tau_b t/2}{g} \left(Bs\cos(st) + \frac{\tau_b B}{2} \sin(st) \right) x, \end{aligned} \tag{4.12}$$

where $B = 5m/s$ is a constant. The boundary between wet and dry nodes can be found from

$$x = -\frac{a^2 \exp(-\tau_b t/2)}{2gh_0} \left(Bs\cos(st) + \frac{\tau_b}{2} \sin(st) \right) \pm a. \tag{4.13}$$

By setting the initial water depth value to

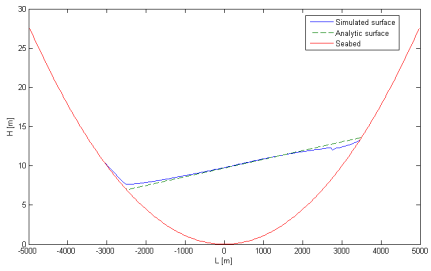
$$h_{\text{initial}} = h(x,0) \tag{4.14}$$

and

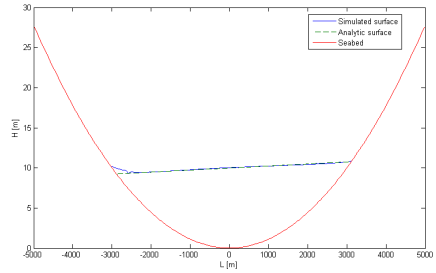
$$u(x,0) = 0 \tag{4.15}$$

we can now simulate the system and compare the simulated and analytical results. To do this we use a $10000m$ long system with $lx = 200$ and $e = 100m/s$. Studying Eq. (4.12) and (4.13) when $t \to \text{inf}$ we see that the flow will go to rest at $h(x, \text{inf}) = h_0$ and with wet/dry boundary at $x = \pm a$. This is the result of bed friction slowing the sloshing movement down until it is at rest.

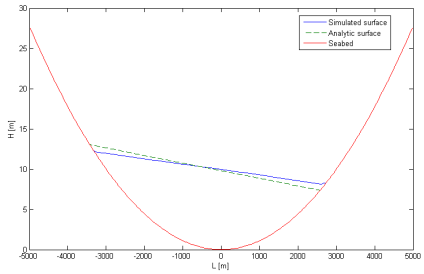
From Fig. 4.16 we observe the analytical and simulated water surface quickly reaching diverging results. The flow velocity can be observed from Fig. 4.17 where we see some unphysical velocity changes.



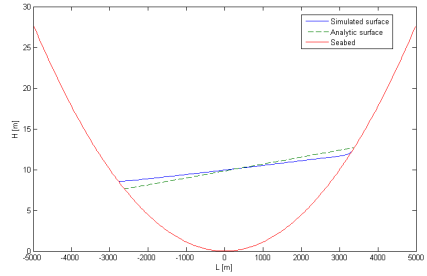
(a) $t = 500s$



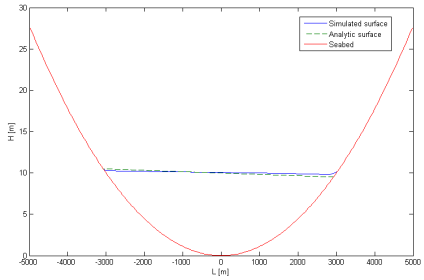
(b) $t = 1000s$



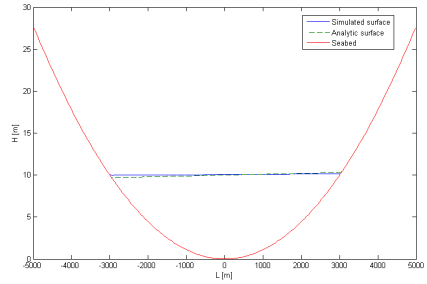
(c) $t = 1500s$



(d) $t = 2000s$

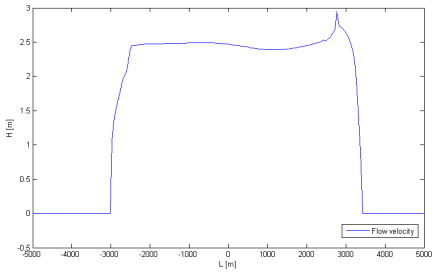


(e) $t = 3000s$

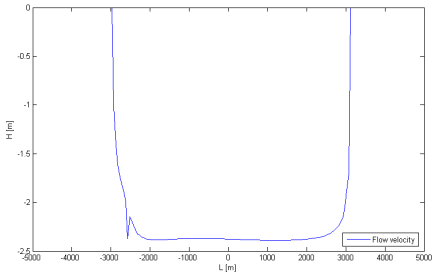


(f) $t = 6000s$

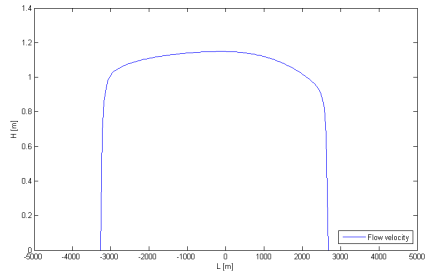
Figure 4.16: The simulated and analytical flow surface of sloshing motion.



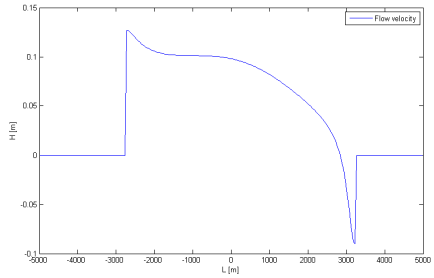
(a) $t = 500s$



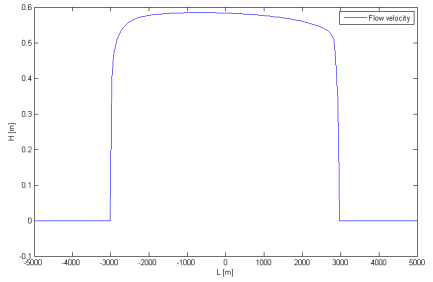
(b) $t = 1000s$



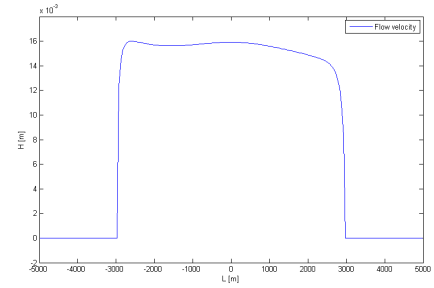
(c) $t = 1500s$



(d) $t = 2000s$



(e) $t = 3000s$



(f) $t = 6000s$

Figure 4.17: The simulated flow velocity of sloshing motion.

5 Discussion

We will here discuss the results gathered in chapter 4.

In section 4.1 to 4.6 we got highly accurate results when we tested the basic method with and without the bed slope source term. From this we can establish that the basic equation and the bed slope term are accurate in both one and two dimensions. By testing the bed force term in one dimension, we also find the friction term to be very accurate.

When we checked the bed friction term in two dimensions however, our simulation returned a large error in the velocity compared to the analytical result. This means there is something incorrect with the way bed friction is implemented. As the resulting velocity is lower than what it should be, we can conclude that the faulty implementation results in increased diagonal friction. When we consider the fact that this is not a known problem for LBM, we can state with a high degree of certainty that this flaw comes from poor implementation of LBM and not from flaws of the LBM itself.

The drying/wetting process observed in section 4.8 shows that the wetting mechanism used in this thesis is slowing down the front. This issue was not observed when Liu and Zhou worked on the same problem in [28]. This highly implies that the way this process was implemented in this thesis was faulty. It is worth noting, however, that this test does not include a steady state flow like the one the simulation module is intended to be used with.

With a steady source providing water, the slowing of the water at the front should theoretically not influence the resulting flow to any large degree. Our target is the flow after it has converged to a solution, which leaves the slowing of the dynamics to be assumed unimportant.

An issue that can cause greater problems, however, is the slow movement of water when water depth becomes sufficiently low. This problem can be spotted in Fig. 4.16a. On the left side of the figure, we can spot areas where the water slows behind the theoretical solution. We assume that this is caused by the increasing bed friction as the water depth decreases, resulting in a local slowing effect.

At this time, we are not certain how to correct this issue as it seems to be a fundamental issue with the method. This was, however, not experienced by Liu and Zhou [28] when they implemented the same test scenario.

Another issue that showed itself in section 4.8 is the unphysical spikes in the velocity observed in Fig. 4.17a and 4.17b. These spikes are the result of the model being unstable at high Froude numbers. The simulator actively suppresses the velocity to make sure it never goes past $F_r > 0.97$, which is what we observe happening in those two cases.

During the development of this thesis, we faced issues while implementing dry areas into the systems. These issues caused the boundaries between the dry and wet areas to create false velocities. At that time we used a slightly more complex bounce-back scheme which better simulated the distribution functions at the boundary layer. To solve the recurring issue we chose to implement the simplest bounce-back scheme to handle these problems.

One possible worry of this situation is that the simplest bounce-back scheme is too simple. If this is the case, then it can cause lower accuracy at the borders and thus be the root cause of both issues observed for the drying/wetting process.

All the problematic areas faced in this thesis have been correctly implemented previously by the leading scientists in this field, proving that the issues can be fixed. However

we did not manage to correctly implement these solutions within the limits of this thesis.

6 Conclusion

The goal of this thesis was to create a general lattice Boltzmann simulator that can solve the shallow water equations for steady flows. This module should be able to handle dry areas in the module as well as wet areas. The work in this thesis has only been toward building a functioning steady state water transport simulator, as the sediment simulation code is done in a separate module which is already deemed complete.

To accomplish this task, we created a simulator using the lattice Boltzmann method,

$$\begin{aligned} f_\beta(\mathbf{x}+\mathbf{e}_\beta\Delta t, t+\Delta t) &= f_\beta(\mathbf{x},t) - \frac{1}{\tau}(f_\beta - f_\beta^{eq}) \\ -\frac{g\bar{h}}{6P_\beta e^2}[z_b(\mathbf{x}+\mathbf{e}_\beta\Delta t) - z_b(\mathbf{x})] &+ \frac{\Delta t}{3P_\beta e^2} e_{\beta i} F_i, \end{aligned} \tag{6.1}$$

with

$$P_\beta = \begin{cases} 0, & \beta = 0, \\ 1, & \beta = 1,3,5,7, \\ 4, & \beta = 2,4,6,8, \end{cases} \tag{6.2}$$

and $\bar{h} = \frac{1}{2}[h(\mathbf{x},t) + h(\mathbf{x}+\mathbf{e}_\beta\Delta t, t)]$ and $F_i = -\frac{\tau_{bi}}{\rho}$. Lattice Boltzmann method is proved stable for all systems where the fluid has a Froude number lower than 1, single time relaxation constant larger than 0.5 and a Mach number lower than 1.

The simulator proved to be highly accurate when handling frictionless wet systems in two dimensions. It also proved accurate at simulating bed friction in one-dimensional systems.

In two-dimensional systems we faced an issue where the diagonal friction term was too large. When simulating the converting of dry nodes to wet nodes, the simulator did not manage to achieve the analytic speed of conversion and instead slowed down. The same issue is evident when converting wet nodes to dry, resulting in a flow that trails the analytic solution. These three areas are likely caused by imperfect implementation of the wetting/drying process and of the bed friction term.

The wetting/drying issues might be unproblematic when the solver is used as intended with steady state flows, however this needs further testing to be determined.

6.1 Future work

To create a complete general steady state solver, the force term must be correctly implemented. By correcting this issue, the simulator should be able to handle every system where all nodes in the system is wet.

Similarly, by fixing the wetting/drying process, the simulation module can begin simulating the correct transitions of nodes from dry to wet and opposite. This will allow the module to be useful for dynamic systems where the state of the system at any given time might be wanted. A common example of this is calculating and simulating tides on beaches.

Thirdly, the implemented bounce-back scheme is the simplest available, however, there is some uncertainty as to whether this causes some small errors in the distribution function at the boundaries. A study on the effect of utilizing a more complex bounce-back scheme is therefore advised.

When these issues have all been corrected, it is advised that the module is updated to handle parallel threads. This will create a significant speedup per cycle, allowing for much more efficient simulations. As all parts the general method are local in space, this should not prove too challenging.

References

[1] A. Bermudez and M. E. Vazquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids*, 23(8):1049–1071, 1994.

[2] P. K. Stansby and J. G. Zhou. Shallow-water solver with non-hydrostatic pressure: 2d vertical plane problems. *International Journal for Numerical Methods in Fluids*, 28(3):541–563, 1998.

[3] J. G. Zhou. Velocity-depth coupling in shallow water flows. *Journal of the Hydraulics Division*, 121(10):717–724, 1995.

[4] L. Zhong, Feng S., and Gao S. Wind-driven ocean circulation in shallow water lattice boltzmann model. *Advances in Atmospheric Sciences*, 22(3):349–358, 2005.

[5] R. Salmon. The lattice boltzmann method as a basis for ocean circulation modelling. *Journal of Marine Research*, 57(3):503–535, 1999.

[6] T. R. A. Uoane. Lattice boltzmann methods for shallow water flow applications. Master's thesis, University of the Witwatersrand, 2011.

[7] M. Gardner. The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223:120–123, 1970.

[8] J. O. Hardy, de Pazzis, and Y. Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical Review*, A13:120–123, 1976.

[9] U. Frisch, D. d'Humières, Hasslacher B., P. Lallemand, Y. Pomeau, and J. P. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649–707, 1987.

[10] J. G. Zhou. *Lattice Boltzmann Methods for Shallow Water Flows*. Springer-Verlag Berlin Heidelberg, 1st edition, 2004.

[11] I. S. Sokolnikoff and R. M. Redheffer. *Mathematics of Physics and Modern Engineering*. McGraw-Hill, New York, USA, 2st edition, 1966.

[12] M. Spivak. *Calculus*. Benjamin W. A Inc, New York, USA, 1967.

[13] M. B. Abbot and W. A. Price. *Coastal, Estuarial and Harbour Engineer's Reference Book*. Chapman and Hall, London, UK, 1993.

[14] A. G. L. Borthwick and G. A. Akponasa. Reservoir flow prediction by contra-variant shallow water equations. *Journal of the Hydraulics Division*, 123(5):432–439, 1997.

[15] C. B. Vreugdenhil and J. H. Wijbenga. Computation of flow patterns in rivers. *Journal of the Hydraulics Division*, 108(11):1296–1310, 1982.

[16] J. J. McGruirk and W. Rodi. A depth-average mathematical model for the field of the near side discharges into open-channel flow. *Journal of Fluid Mechanics*, 86:761–781, 1978.

[17] S. Chen and G. D. Doolen. Lattice boltzmann method for fluid flows. *Annual Reviews Fluid Mechanics.*, 30:329–364, 1998.

[18] G. R. McNamara and G. Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61:2332–2335, 1988.

[19] P. L. Bhatnagar, E.P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3):511–525, 1954.

[20] Y. H. Qian, D. d'Humières, and P. Lallemand. Lattice bgk models for navier-stokes equation. *Europhysics Letters*, 17(6):479–484, 1992.

[21] J. M. V. A. Koelman. A simple lattice boltzmann scheme for navier-stokes fluid flow. *Europhysics Letters*, 15(6):603–607, 1991.

[22] U. Frisch, Hasslacher B., and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(6):1505–1508, 1986.

[23] Y. H. Qian. *Lattice gas and lattice kinetic theory applied to the Navier-Stokes equations*. PhD thesis, University of Marie Curie, Paris, 1990.

[24] P. A. Skordos. Initial and boundary conditions for the lattice boltzmann method. *Physical Review E*, 48:4823–4842, 1993.

[25] J. G. Zhou. An elastic-collision scheme for lattice boltzmann methods. *Journal of Modern Physics C*, 12(3):387–401, 2001.

[26] D. H. Rothman and S. Zaleski. *Lattice-Gas Cellular Automata*. Cambridge University Press, London, 1st edition, 1997.

[27] J. G. Zhou. Enhancement of the labswe for shallow water flows. *Journal of Computational Physics*, 230:394–401, 2011.

[28] H. Liu and J. G. Zhou. Lattice boltzmann approach to simulating a wetting-drying front in shallow flows. *Journal of Fluid Mechanics*, 743:32–59, 2014.

[29] Q. Zou and X. He. On pressure and velocity boundary condition for the lattice boltzmann bgk model. *Physics of Fluids*, 35(6):1591–1598, 2002.

[30] M. B. Schlaffer. *Non-reflecting Boundary Conditions for the Lattice Boltzmann Method*. PhD thesis, Technische Universität München, 2013.

[31] N. Goutal and F. Maurel. *Proceedings of the second Workshop on Dam-break Wave Simulations*. EDF-DER, Paris, 1st edition, 1997.

[32] X. Fang. The open channel flow calculator. `http://www.eng.auburn.edu/~xzf0001/Handbook/Channels.html`, 2000. Online; accessed 2016-06-24.

[33] J. Sampson, A. Easton, and M. Singh. Moving boundary shallow water flow above parabolic bottom topography. *ANZIAM Journal*, 47:C373–C387, 2006.