Fourier Neural Operator-Based Models for Solving the Shallow Water Equations in Spherical Coordinates

Melissa Ulsøe Jessen (s194322)

Supervisor: DTU Compute
Co-supervisor: DTU Compute

Master Thesis
Mathematical Modelling and Computation

DTU Compute
Technical University of Denmark
February 2, 2025

Fourier Neural Operator-Based Models for Solving the Shallow Water Equations in Spherical Coordinates

Melissa Ulsøe Jessen (s194322)

October 2, 2024

Abstract

This master thesis focuses on the numerical and data-driven solutions for the Shallow Water Equations (SWE), which are fundamental in computational fluid dynamics. The project aims to solve the SWE using the Finite Volume Method (FVM) in 2D and spherical coordinates and subsequently exploit machine learning techniques, particularly the Fourier Neural Operator (FNO), to improve solution accuracy and efficiency. By the end of the project, the goal is to have a robust and efficient computational toolkit for solving the SWE, which will also be useful for simulating important geophysical processes like Kelvin and Rossby waves.

Notes

40 years ago Lax and Wendroff [116] proved mathematically that conservative numerical methods, if convergent, do converge to correct solutions of the equations. More recently, Hou and LeFloch [91] proved a complementary theorem which says that if a non-conservative method is used, then the wrong solution will be comptedm if this contains a shock wave.

Spurious oscillations are unavoidable if one uses linear methods of accuracy greater than one.

Gudonovs theorem: all (linear) schemes of accuracy greater than one will provide spurious oscillations. One must use non-linear methods, even when applied to linear problems.

A succesful new class of shock-capturing numerical methods are the so-called high-resolution methods. High-resolution methods = Oscillation-free near shock waves, and retain second-order accuracy in smooth parts of the flow.

Gudonov methods are shock-capturing upwind methods. He later proposed a very fast exact Riemann solver and also approximate Riemann solvers.

a: the generalisation of Gudonov's first-order method to second-order accuracy by van Leer. b: the development of new approximate Riemann solvers by Roe and others.

TVD = Total Variation Diminishing.

More advanced, not presented in this book are UNO, ENO and WENO methods. Also the discountinuous Galerkin finite element method, combining FEM and Gudonov theory.

Trento course: - Day 2: SWE is a non-linear scalar equation/ conservation law

Contents

Chapter 1

Introduction

This master thesis focuses on the numerical and data-driven solutions for the Shallow Water Equations (SWE), which are fundamental in computational fluid dynamics. The project aims to solve the SWE using the Finite Volume Method (FVM) in 2D and spherical coordinates and subsequently exploit machine learning techniques, particularly the Fourier Neural Operator (FNO), to improve solution accuracy and efficiency. By the end of the project, the goal is to have a robust and efficient computational toolkit for solving the SWE, which will also be useful for simulating important geophysical processes like Kelvin and Rossby waves.

The Shallow Water Equations (SWE) are a set of hyperbolic partial differential equations that describe the motion of a fluid in a shallow layer of water. In this project, we will derive the SWE in 1D, 2D and in spherical coordinates, to model the flow of water on the surface of the Earth. We will go through the Finite Volume Method (FVM) in 1D and 2D, which is a numerical method for solving partial differential equations by dividing the domain into small control volumes and integrating the equations over these volumes. The method is widely used in computational fluid dynamics to model the behavior of fluid flows. We will implement the FVM and use it to solve the SWE in 1D for two different dam break problems.

1.1 Motivation

1.2 Literature

1.3 Thesis overview

Chapter 2

Theory

2.1 Notation

Before deriving the shallow water equations (SWE), we will introduce the notation that will be used throughout this report. In both the 1D and 2D cases of SWE, we use cartesian coordinates (x, y, z) with time denoted by t. Given that linear algebra is a fundamental tool used in this report, we first establish the relevant notation. Lowercase bold letters represent vectors, while uppercase bold letters represent matrices. For instance, $\mathbf{a}$ is a vector of size $1 \times r, r \in \mathbb{R}$, and $\mathbf{A}$ is a matrix of size $m \times n, m, n \in \mathbb{R}$. The identity matrix, denoted by $\mathbf{I}$, is a square matrix with ones along the diagonal and zeros elsewhere. For example, the 3×3 identity matrix is given by:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We use the following notation for partial derivatives:

$$f_x = \frac{\partial f}{\partial x} = \partial x f, \quad f_y = \frac{\partial f}{\partial y} = \partial y f, \quad f_z = \frac{\partial f}{\partial z} = \partial z f.$$

The gradient operator, denoted by ∇, gives the gradient of a scalar function f as a vector:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{bmatrix}.$$

The divergence operator, represented as $\nabla\cdot$, gives the divergence of a vector $\mathbf{a}$ as:

$$\nabla \cdot \mathbf{a} = \frac{\partial a_1}{\partial x} + \frac{\partial a_2}{\partial y} + \frac{\partial a_3}{\partial z}.$$

Given two vectors $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$, their tensor product, denoted as $\mathbf{a} \otimes \mathbf{b}$, is a matrix where each element is the product of the elements of $\mathbf{a}$ and $\mathbf{b}$, i.e.,

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}.$$

2.2 The SWE with conservative variables

In this section we will derive the shallow water equations (SWE) in conservative form. The derivation follows four steps: First we consider the conservation laws for mass and momentum, then we consider the boundary conditions

for a free surface problem, and make some neccessary assumptions and finally we use the boundary conditions to integrate the conservation laws over depth. The derivation follows the methods outlined in [2] and [5].

2.2.1 Conservation laws

The conservation laws for mass and momentum can be expressed generally as follows (see eq. (2.1) and (2.2) in [2]):

$$\rho_t + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{2.2.1}$$

$$(\rho \mathbf{v})_t + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + p\mathbf{I} - \mathbf{T}) = \rho \mathbf{g}, \tag{2.2.2}$$

where ρ is the fluid density, $\mathbf{v} = \begin{bmatrix} u & v & w \end{bmatrix}$ is the fluid velocity in the x, y and $z-$direction respectively; p is the pressure, $\mathbf{I}$ is the identity matrix, and vector $\mathbf{g} = \begin{bmatrix} g_1 & g_2 & g_3 \end{bmatrix}$ represents body forces including gravity. The matrix $\mathbf{T}$ is the viscous stress tensor, given by

$$\mathbf{T} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}.$$

However, in this project the viscous stress tensor $\mathbf{T}$ is neglected. The matrix $\mathbf{v} \otimes \mathbf{v}$ represents the tensor product of the velocity vector $\mathbf{v}$ with itself, i.e.,

$$\mathbf{v} \otimes \mathbf{v} = \begin{bmatrix} u^2 & uv & uw \\ vu & v^2 & vw \\ wu & wv & w^2 \end{bmatrix}.$$

Note that $\mathbf{v} \otimes \mathbf{v} = \mathbf{v}^\top \mathbf{v}$, allowing us to rewrite (2.2.2) as

$$(\rho \mathbf{v})_t + \nabla \cdot (\rho \mathbf{v}^\top \mathbf{v} + p\mathbf{I}) - \rho \mathbf{g} = 0. \tag{2.2.3}$$

For incompressible fluids, the fluid density ρ is independent of the pressure p. Although the density is not constant, it is assumed to depend only on temperature and salinity. Therefore, the equations (2.2.1) and (2.2.3), splitted in x, y and $z-$directions, simplify to

$$u_x + v_y + w_z = 0, \tag{2.2.4}$$

$$u_t + uu_x + vu_y + wu_z = -\frac{1}{\rho} p_x, \tag{2.2.5}$$

$$v_t + uv_x + vv_y + wv_z = -\frac{1}{\rho} p_y, \tag{2.2.6}$$

$$w_t + uw_x + vw_y + ww_z = -\frac{1}{\rho} p_z - g. \tag{2.2.7}$$

2.2.2 Boundary conditions

In this project, we consider the flow of water with a free surface. To solve the SWE, it is essential to impose boundary conditions at both the bottom of the water column and the free surface. We assumme the bottom boundary is defined by a function

$$z = b(x, y),$$

meaning that the bottom is dependent on x and y, but not on time. The free surface is defined by

$$z = s(x, y, t) \equiv b(x, y) + h(x, y, t),$$

where $h(x, y, t)$ is the water depth at time t. The following illustration helps to visualize the setup:

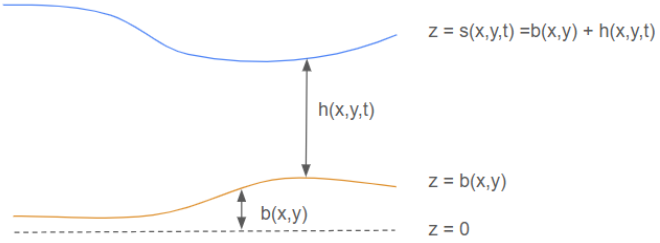

Figure 2.1: Illustration of a water column with a free surface.

We impose boundary conditions at the bottom and the free surface, adressing both kinematic and dynamical conditions. Assume that a boundary is described by the function

$$\psi(x, y, z, t) = 0.$$

For the free surface, this boundary is given by

$$\psi(x, y, z, t) \equiv z - s(x, y, t) = 0, \tag{2.2.8}$$

and for the bottom, it is described by

$$\psi(x, y, z, t) \equiv z - b(x, y) = 0. \tag{2.2.9}$$

In the kinematic condition, we assume that fluid particles on the boundary remain on the boundary over time. It is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi(x, y, z, t) = \psi_t + u\psi_x + v\psi_y + w\psi_z = 0. \tag{2.2.10}$$

Applying this to the free surface by substituting (2.2.8) into the kinematic condition (2.2.10) yields

$$(s_t + us_x + vs_y - w)|_{z=s} = 0. \tag{2.2.11}$$

Similarly, for the bottom, substitung (2.2.9) into the kinematic condition (2.2.10) gives

$$(b_t + ub_x + vb_y - w)|_{z=b} = 0. \tag{2.2.12}$$

The dynamical condition is related to the pressure distribution at the free surface. We assume that the pressure at the free surface is equal to the pressure in the air above the surface, that is, the atmospheric pressure. Since absolute pressure levels are irrelevant (we are primarily concerned with pressure differences), we set the pressure at the free surface to zero. This leads to the following expression for the pressure at the free surface:

$$p(x, y, z, t)|_{z=s(x,y)} = p_{atm} = 0. \tag{2.2.13}$$

This condition, known as the dynamical condition, relates to the forces acting on the boundaries of the fluid.

2.2.3 Assumptions

To derive the SWE it is neccessary to make some assumptions. The shallow water equations are an approximation to the full free-surface problem and result from the assumption that the vertical compononent of the acceleration

is negligible. We begin by assuming that the vertical acceleration, represented by the total derivate of the vertical velocity component w with respect to time, is negligible. This assumption leads to the condition

$$\frac{\mathrm{d}w}{\mathrm{d}t} = w_t + uw_x + vw_y + ww_z = 0.$$

Here, $\frac{\partial w}{\partial t}$ is the partial derivative of w with respect to t, while the total derivate $\frac{\mathrm{d}w}{\mathrm{d}t}$ accounts for both the direct change of w with t and the changes due to the movement of the fluid in the x, y and z directions. Recall, that mathematically, the total derivative of w wrt. t is given by

$$\begin{aligned}\frac{\mathrm{d}w}{\mathrm{d}t} &= \frac{\partial w}{\partial t} + \frac{\mathrm{d}x}{\mathrm{d}t}\frac{\partial w}{\partial x} + \frac{\mathrm{d}y}{\mathrm{d}t}\frac{\partial w}{\partial y} + \frac{\mathrm{d}z}{\mathrm{d}t}\frac{\partial w}{\partial z} \\ &= w_t + uw_x + uw_y + ww_z.\end{aligned}$$

Applying $\mathrm{d}\ w/\mathrm{d}\ t = 0$ in the $z-$momentum conservation equation (2.2.7) simplifies it to

$$p_z = -\rho g.$$

This implies the pressure distribution follows

$$p = \int_z^s -\rho g\ \mathrm{d}z = \rho g(s - z), \tag{2.2.14}$$

where s is the surface height. Differentiating (2.2.14) with respect to x and y yields

$$p_x = \rho g s_x, \quad p_y = \rho g s_y.$$

Substituting these expressions into the $x-$ and $y-$momentum conservation equations (2.2.5) and (2.2.6) leads to

$$u_t + uu_x + vu_y + wu_z = -gs_x, \tag{2.2.15}$$

and

$$v_t + uv_x + vv_y + wv_z = -gs_y. \tag{2.2.16}$$

These are the simplified momentum equations for the shallow water equations. We realize that p_x and p_y are both independent of z, implying that $\mathrm{d}\ u/\mathrm{d}\ t$ and $\mathrm{d}\ v/\mathrm{d}\ t$ are also independent of z. Hence $u_z = v_z = 0$, implying that (2.2.15) and (2.2.16) can be simplified to

$$u_t + uu_x + vu_y = -gs_x, \tag{2.2.17}$$

and

$$v_t + uv_x + vv_y = -gs_y. \tag{2.2.18}$$

2.2.4 Integration over depth

The next step in deriving the SWE is to integrate the equations over the vertical direction. We integrate the mass conservation equation (2.2.4) and the momentum conservation equations (2.2.5), (2.2.6) from the bottom, $z = b(x, y)$ to the free surface, $z = s(x, y, t)$. Starting with the mass conservation equation (2.2.4), we have

$$\int_b^s u_x + v_y + w_z\ \mathrm{d}z = 0,$$

implying that, using linearity of the integral:

$$\int_b^s u_x\ \mathrm{d}z + \int_b^s v_y\ \mathrm{d}z + w|_{z=s} - w|_{z=b} = 0. \tag{2.2.19}$$

We will use Leibniz's integral rule [6], which is stated as follows:

$$\frac{\mathrm{d}}{\mathrm{d}x}\int_{a(x)}^{b(x)} f(x,t)\ \mathrm{d}t = \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x,t)\ \mathrm{d}t + f(x,b(x))\frac{\mathrm{d}}{\mathrm{d}x}b(x) - f(x,a(x))\frac{\mathrm{d}}{\mathrm{d}x}a(x), \tag{2.2.20}$$

to integrate the first two terms on the right hand side in (2.2.19), which yields

$$\int_b^s u_x \mathrm{d}z = \frac{\mathrm{d}}{\mathrm{d}x}\int_b^s u\ \mathrm{d}z - u|_{z=s}\frac{\mathrm{d}s}{\mathrm{d}x} + u|_{z=b}\frac{\mathrm{d}b}{\mathrm{d}x}$$

and

$$\int_b^s v_y\ \mathrm{d}z = \frac{\mathrm{d}}{\mathrm{d}y}\int_b^s v\ \mathrm{d}z - v|_{z=s}\frac{\mathrm{d}s}{\mathrm{d}y} + u|_{z=b}\frac{\mathrm{d}b}{\mathrm{d}y}.$$

Note that since a change in x does not affect the y-component of the bottom or surface, we have that $\mathrm{d}s/\mathrm{d}x = s_x$ and $\mathrm{d}b/\mathrm{d}x = b_x$, and correspondingly for s_y and b_y. Likewise we can substitute $\frac{\mathrm{d}}{\mathrm{dx}}$ with $\frac{\partial}{\partial x}$ in the integrals, since the integrals are with respect to z. Inserting these results in the above equations gives

$$\int_b^s u_x\ \mathrm{d}z = \frac{\partial}{\partial x}\int_b^s u\ \mathrm{d}z - u|_{z=s}s_x + u|_{z=b}b_x \tag{2.2.21}$$

and

$$\int_b^s v_y\ \mathrm{d}z = \frac{\partial}{\partial y}\int_b^s v\ \mathrm{d}z - v|_{z=s}s_y + u|_{z=b}b_y. \tag{2.2.22}$$

Inserting these results in (2.2.19) gives

$$\frac{\partial}{\partial x}\int_b^s u\ \mathrm{d}z - u|_{z=s}s_x + u|_{z=b}b_x + \frac{\partial}{\partial y}\int_b^s v\ \mathrm{d}z - v|_{z=s}s_y + u|_{z=b}b_y + w|_{z=s} - w|_{z=b} = 0. \tag{2.2.23}$$

From (2.2.12) we have

$$w|_{z=b} = (ub_x + vb_y)|_{z=b}, \tag{2.2.24}$$

and from (2.2.11) we have

$$w|_{z=s} = (s_t + us_x + vs_y)|_{z=s}. \tag{2.2.25}$$

Realizing that $s = b + h$ and hence $s_t = h_t$, as the bottom is fixed. Recall that u and v are independent of z, and the water depth is $h = s - b$, meaning we have

$$\int_b^s u\ \mathrm{d}z = u(s-b) = hu, \quad \int_b^s v\ \mathrm{d}z = v(s-b) = hv.$$

Putting it all together (2.2.23) simplifies to

$$h_t + (hu)_x + (hv)_y = 0, \tag{2.2.26}$$

which is also the first equation in the SWE in conservative form. When integrating the momentum equations (2.2.17) and (2.2.18) over the vertical direction, we see that since the equations are independent of z, the resulting equations are simply

$$h(u_t + uu_x + vu_y + gs_x) = 0, \tag{2.2.27}$$

$$h(v_t + uv_x + vv_y + gs_y) = 0. \tag{2.2.28}$$

We multiply (2.2.26) with u and v respectively, and add the two equations to (2.2.27) and (2.2.28) correspondingly. Recall that $s = h + b$. By using the product rule for differentiation and collecting terms, we obtain the momentum equations in conservative form:

$$(hu)_t + (hu^2 + \frac{1}{2}gh^2)_x + (huv)_y = -ghb_x,$$

and

$$(hv)_t + (huv)_x + (hv^2 + \frac{1}{2}gh^2)_y = -ghb_y.$$

2.2.5 The SWE in vector form for 1D and 2D

The SWE, which consist of three partial differential equations, can be written in differential conservation law form as a vector equation

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{S}(\mathbf{U}), \tag{2.2.29}$$

where

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix} \quad \text{and} \quad \mathbf{S}(\mathbf{U}) = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}.$$

We call $\mathbf{U}$ the vector of conserved variables, $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$ the flux vectors in the x and y direction, and $\mathbf{S}(\mathbf{U})$ the source term vector.

In this project, we will begin by considering the homogenneous one-dimensional case, where the flow is only in the x-direction:

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0,$$

where

$$\mathbf{U} = \begin{bmatrix} h \\ hu \end{bmatrix}, \quad \text{and} \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}.$$

2.2.6 SWE in Spherical Coordinates

TBD.

Chapter 3

Methodology

In this section we will describe the finite volume method for solving the shallow water equations (SWE) and the numerical implementation of the method in Python. In the Finite Volume Method, we discretize the domain into cells or control volumes. Then we solve the local Riemann problem at the cell interface to obtain the fluxes. Using the computed fluxes, we update the solution in each cell. This way, the FVM allows for discountinuous solutions, as we solve the Riemann problem at the cell interfaces. Therefore it is well suited for hyperbolic conservation laws, such as the shallow water equations.

3.1 Riemann problem

The Riemann problem is a generalisation of the so-called dam-break problem. The difference is that in the Riemann problem the particle velocity components, u_L, u_R, v_L and v_R, are allowed to be distinct from zero, whereas in the dam-break problem, they must be zero.

We consider the case where the water depth is positive everywhere, i.e., what we call a wet bed. A dry bed is then the case if the water depth is zero in some cells. The special Riemann problem where parts of the bed are dry is dealing with the so-called dry fronts or wet/dry fronts, which are challenging to handle numerically. We will leave these cases for now, and only consider the wet bed problems.

Later we will consider approximate solutions to the Riemann problem, which are intended for local use in Godunov-type methods. But for now, we consider the exact Riemann solvers for wet bed problems.

The Riemann problem is defined as the initial-value problem (IVP)

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0, \tag{3.1.1}$$

$$\mathbf{U}(x,0) = \begin{cases} \mathbf{U_L}, & x < 0, \\ \mathbf{U_R}, & x > 0. \end{cases} \tag{3.1.2}$$

We consider the x-split, two-dimensional shallow water equations with the initial states:

$$\mathbf{U_L} = \begin{bmatrix} h_L \\ h_L u_L \\ h_L v_L \end{bmatrix}, \quad \mathbf{U_R} = \begin{bmatrix} h_R \\ h_R u_R \\ h_R v_R \end{bmatrix},$$

which represents the conditions at time $t = 0$ in the left and right states of $x = 0$, respectively. The vectors are

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}.$$

In the Riemann problem the particle velocity components u_L, u_R, v_L and v_R are allowed to be distinct from zero, whereas in the dam-break problem, they must be zero.

3.1.1 Wave solution of the Riemann problem/something about waves..

Some background about waves.

In the solution of the Riemann problem (3.1.1) there are four possible wave patterns outcomes, which are combinations of shock waves and rarefaction waves. In each case there are three waves, the left and right waves correspond to the one-dimensional SWE, and the middle wave aries from the $y-$momentum equation in (3.1.1) and is always a shear wave. The left and right waves are either shock waves or rarefaction waves. The four possible wave patterns are illustrated in Figure XX and are as follows:

1. Left rarefaction, right shock
2. Left shock, right rarefaction
3. Both left and right rarefaction
4. Both left and right shock

Hence the structure of the solution in general is shown in the figure below. From the figure we see that the solution consists of three waves, a left wave, a middle wave and a right wave, which together seperate four regions, described by the vector

$$\mathbf{W} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}.$$

The four regions are describes by $\mathbf{W}_L$ (left data), $\mathbf{W}_R$ (right data), $\mathbf{W}_{*L}$ and $\mathbf{W}_{*R}$, which both denote star region data. We are interested in the star region data, since these are the unknowns. We know that the left and right waves are always either a shock wave or a rarefaction wave, and the middle wave is always a shear wave. Based on the given initial conditions, we must determine the types of waves. Second, it is known that across the left and right waves, both h and u change but v remains constant. Whereas across the middle wave, v changes but h and u remain constant. Thus the water depth and particle velocity are constant in the star region and are denoted by h_* and u_*, respectively.

3.1.2 Exact Riemann solver

The exact Riemann solver presented in this section is very efficient and leads to Gudonov methods, that are only slightly more expensive than those based on approximate Riemann solvers [2]. Important to find the exact solution to the Riemann problem in the early stage of development, before moving on to more complex applications.

In the solution of the Riemann problem (3.1.1), there are four possible wave patterns outcomes, which are combinations of shock waves and rarefaction waves. In each case there are three waves, the left and right waves

correspond to the one-dimensional SWE, and the middle wave aries from the $y-$momentum equation in (3.1.1). The left and right waves are either shock waves or rarefaction waves, and the middle wave is always a shear wave.

But for now, we will consider the case where the solution consists of a single non-trivial wave and all other waves are assumed to have zero strenght. This is enough to solve the Riemann problem as it is always possible to solve the Riemann problem by considering one wave at a time.

We denote the constant values of the water depth and particle velocity in the star region by h_* and u_*, respectively.

3.1.3 Approximate Riemann solvers

We will study two approximate Riemann solvers, that is, the HLL and Roe.

HLL solver

The HLL (Harten, Lax and van Leer) approach assumes a two-wave structure of the Riemann problem, see Figure. The solver is based on the data $\mathbf{U}_L \equiv \mathbf{U}_i^n, \mathbf{U}_R \equiv \mathbf{U}_{i+1}^n$ and fluxes $\mathbf{F}_L \equiv \mathbf{F}(\mathbf{U}_L), \mathbf{F}_R \equiv \mathbf{F}(\mathbf{U}_R)$.

The HLL flux is given by

$$\mathbf{F}_{1+\frac{1}{2}} = \begin{cases} \mathbf{F}_L & \text{if } S_L \geq 0, \\ \mathbf{F}^{hll} \equiv \frac{S_R\mathbf{F}_L - S_L\mathbf{F}_R + S_L S_R(\mathbf{U}_R - \mathbf{U}_L)}{S_R - S_L} & \text{if } S_L \leq 0 \leq S_R, \\ \mathbf{F}_R & \text{if } S_R \leq 0. \end{cases}$$

The wave speeds S_L and S_R must be estimated in some way, and one possibility is to use

$$S_L = u_L - a_L q_L, \quad S_R = u_R + a_R q_R.$$

Roe solver

Consider the non-linear Riemann problem in (3.1.1):

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x \equiv \mathbf{U}_t + \mathbf{A}\mathbf{U}_x = 0,$$

where $\mathbf{A}$ is the Jacobian matrix of $\mathbf{F}$. The Roe solver is based on an approximation of the Jacobian matrix $\mathbf{A}$ by a Roe matrix $\tilde{\mathbf{A}}$, which is a constant coefficient matrix, to obtain the linear system

$$\mathbf{U}_t + \tilde{\mathbf{A}}\mathbf{U}_x = 0.$$

This means that the Roe solver solves the approximated Riemann problem

$$\mathbf{U}_t + \tilde{\mathbf{A}}\mathbf{U}_x = 0.$$
$$\mathbb{U}(x,0) = \begin{cases} \mathbf{U}_L, & x < 0, \\ \mathbf{U}_R, & x > 0. \end{cases}$$

exact. That is, the original non-linear conservation laws are replaced by a linearised system with constant coefficients.

The main idea in the Roe solver is to find average values $\tilde{h}, \tilde{a}, \tilde{u}$ and $\tilde{\psi}$ for the depth h, the celerity a (??), the velocity component u and the scalar ψ. The method thus use the following Roe averages:

$$\begin{cases} \tilde{h} = \sqrt{h_L h_R}, \\ \tilde{u} = \dfrac{u_L\sqrt{h_L} + u_R\sqrt{h_R}}{\sqrt{h_L} + \sqrt{h_R}}, \\ \tilde{a} = \sqrt{\dfrac{1}{2}(a_L^2 + a_R^2)}, \\ \tilde{\psi} = \dfrac{\psi_L\sqrt{h_L} + \psi_R\sqrt{h_R}}{\sqrt{h_L} + \sqrt{h_R}}. \end{cases} \tag{3.1.3}$$

The average eigenvalues (of what?) are

$$\tilde{\lambda}_1 = \tilde{u} - \tilde{a}, \quad \tilde{\lambda}_2 = \tilde{u}, \quad \tilde{\lambda}_3 = \tilde{u} + \tilde{a},$$

with the correspinding right eigenvectors

$$\tilde{\mathbf{R}}^{(1)} = \begin{bmatrix} 1 \\ \tilde{u} - \tilde{a} \\ \tilde{\psi} \end{bmatrix}, \quad \tilde{\mathbf{R}}^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{R}}^{(3)} = \begin{bmatrix} 1 \\ \tilde{u} + \tilde{a} \\ \tilde{\psi} \end{bmatrix}.$$

The wave strenghts $\tilde{\alpha}_j$ described by Roe averages are given by

$$\begin{cases} \tilde{\alpha}_1 = \dfrac{1}{2}\left[\Delta h - \dfrac{\tilde{h}}{\tilde{a}}\Delta u\right], \\ \tilde{\alpha}_2 = \dfrac{1}{2}\left[\tilde{h}\Delta\psi\right], \\ \tilde{\alpha}_3 = \dfrac{1}{2}\left[\Delta h + \dfrac{\tilde{h}}{\tilde{a}}\Delta u\right]. \end{cases} \tag{3.1.4}$$

Applying theory of linear systems with constant coefficients. The numerical flux is

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_L + \mathbf{F}_R\right) - \frac{1}{2}\sum_{j=1}^{3}\tilde{\alpha}_j \left|\tilde{\lambda}_j\right| \tilde{\mathbf{R}}^{(j)}. \tag{3.1.5}$$

The Roe flux (3.1.5) is used in the explicit conservative scheme to solve the SWE in 1D.

Entropy fix?

3.2 Finite volume method for Shallow Water Equations

In this section we will describe the finite volume method for solving the shallow water equations (SWE) and the numerical implementation of the method in Python. In the Finite Volume Method, we discretize the domain into cells or control volumes. Then we solve the local Riemann problem at the cell interface to obtain the fluxes. Using the computed fluxes, we update the solution in each cell. This way, the FVM allows for discountinuous solutions, as we solve the Riemann problem at the cell interfaces. Therefore it is well suited for hyperbolic conservation laws, such as the shallow water equations.

3.2.1 Finite Volume Method for the 1D SWE

Recall that the homogeneous 1D SWE are given by

$$\begin{aligned} h_t + (hu)_x &= 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x &= 0. \end{aligned}$$

Explicit conservative formula 1D SWE

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+1/2}^n - \mathbf{F}_{i-1/2}^n\right) + \Delta t \mathbf{S}_i. \tag{3.2.1}$$

This approach (3.2.1) is called a finite volume scheme, since it is based on the integral conservation over finite volumes. Note that the main difference between the FVM and the FDM is that the FVM is based on the integral conservation over finite volumes, whereas the FDM is based on the differential conservation over finite differences. The main idea in the FVM is to define the numerical flux $\mathbf{F}_{i+1/2}^n$ at the cell inerface as a function of the cell averages $\mathbf{U}_i^n$ and $\mathbf{U}_{i+1}^n$. Since only the cell-averages solution is known. This also means that the FVM does not provide pointwise values of the solution, i.e., $\mathbf{U}(x,t)$, but only the cell-averages $\mathbf{U}_i^n$ over the control volume.

3.2.2 Finite Volume Method for the 1D SWE with source term

Consider the inhomoegeneous non-linear system

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{S}(\mathbf{U}),$$

where $\mathbf{S}(\mathbf{U})$ is a vector of sources.

3.2.3 Finite Volume Method for the 2D SWE

Follows the methods outlined in [3]. Consider a time-dependent two dimensional system of conservation laws

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0. \tag{3.2.2}$$

A numerical explicit finite volume scheme to solve (3.2.2) is given by

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^n + \frac{\Delta t}{\Delta x}(\mathbf{F}_{i-1/2,j} - \mathbf{F}_{i+1/2,j}) + \frac{\Delta t}{\Delta y}(\mathbf{G}_{i,j-1/2} - \mathbf{G}_{i,j+1/2}). \tag{3.2.3}$$

This is the unsplit finite volume method, meaning that, in a single step, the cell average $\mathbf{U}_{i,j}^n+$ is updated using the fluxes from all intercell boundaries.

3.2.4 Implementation of the FVM in 1D

The code to solve the SWE in 1D using FVM is based on the Godunov scheme with the exact Riemann solver. The exact solution of the Riemann problem is found by using the Riemann invariants and the Rankine-Hugoniot conditions [1].

3.2.5 The Godunov Upwind Method

We consider the Godunov Upwind method, which is a first-order accurate metod to solve non-linear systems of hyperbolic conservation laws [4].

Consider the initial-boundary value problem (IBVP) for a system of N nonlinear hyperbolic conservation (balance?) laws

$$\begin{cases} \text{PDEs:} & \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{S}(\mathbf{U}), \quad x \in [a,b], \quad t > 0, \\ \text{ICs:} & \mathbf{U}(x,0) = \mathbf{U}^{(0)}(x), \quad x \in [a,b], \\ \text{BCs:} & \mathbf{U}(a,t) = \mathbf{B}_L(t), \quad \mathbf{U}(b,t) = \mathbf{B}_R(t), \quad t \geq 0. \end{cases} \tag{3.2.4}$$

The vectors $\mathbf{B}_L(t)$ and $\mathbf{B}_R(t)$ denote the boundary conditions at the left and right boundaries, respectively. The Godunov Upwind method in conservative form (3.2.1) solves the IBVP (3.2.4).

3.2.6 Implementation of the FVM in 2D

Chapter 4

Numerical results

We have implemented the Finite Volume Method for solving the shallow water equations in 1D and tested it on two different dam break problems [1].

Method of manufactured solutions (MMS) was used to verify the implementation. We use a manufactured solution for $h(x,t)$ and $u(x,t)$. Choose a simple sine wave function for the water height h and a corresponding u that satisfies the shallow water equations. Consider

$$h(x,t) = h_0 + A\cos(\omega t - kx),$$
$$u(x,t) = \frac{A\omega}{kh_0}\cos(\omega t - kx),$$

where h_0 is the constant base depth, A is the amplitude of the wave, k is the wave number, and ω is the angular frequency.

We begin by computing the source terms S_h and S_u. First we compute the partial derivatives

$$h_t =,$$
$$u_t =$$

which gives (using the chain rule)

$$(hu)_x = h_x u + h u_x$$
$$=$$

4.1 The 1D Dam Break Problem

The numerical solution to the 1D Dam Break Problem using the FVM, together with the true solution can be seen in Figure 4.1.

[1] Code and small animations can be found at github, visit `https://github.com/MelissaJessen/Shallow-Water-Equations`.

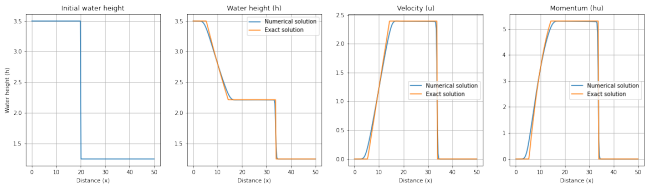

Figure 4.1: 1D dam break problem.

4.2 2D idealised Circular Dam Break Problem

Consider the idealised circular dam break problem with a horizontal bottom. We assume there is an infinitely thin circular wall at radius $R = 2.5m$ is a square domain of size $40m \times 40m$ with centre at $(x_c, y_c) = (20m, 20m)$. The initial conditions are

$$h(x, y, 0) = \begin{cases} 2.5\ m, & \text{if } \sqrt{(x - x_c)^2 + (y - y_c)^2} \leq R, \\ 0.5\ m, & \text{otherwise,} \end{cases}$$
$$u(x, y, 0) = 0,$$
$$v(x, y, 0) = 0.$$

Use the WAF method (chapter 11) along with a second-order dimensional splitting scheme (chapter 12). CFL-condition: $C_{cfl} = 0.9$. van Leer limiter. Mesh: 200×200.

Solve the 1D inhomoegeneous system and compare.

4.3 Test cases

Test case 2:

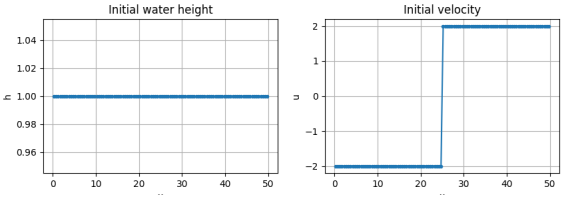

Figure 4.2: Initial conditions for the test case.

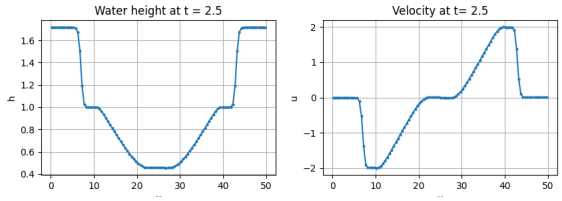

Figure 4.3: Final solution for the test case.

Test case 3:

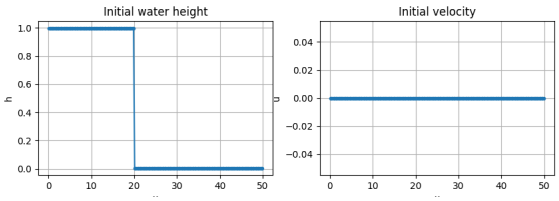

Figure 4.4: Initial conditions for the test case.

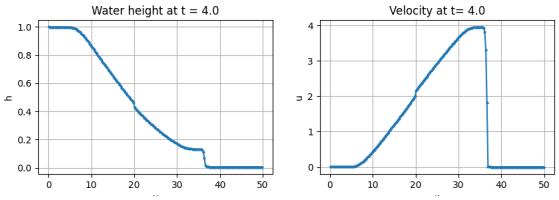

Figure 4.5: Final solution for the test case.

Test case 4:

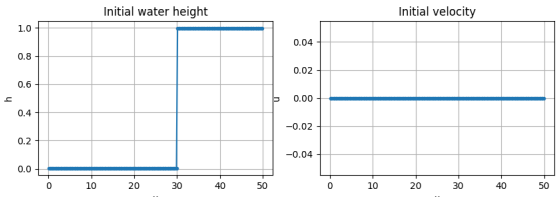

Figure 4.6: Initial conditions for the test case.

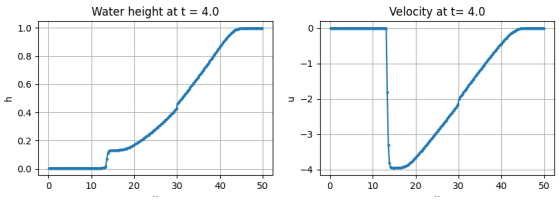

Figure 4.7: Final solution for the test case.

Test case 5:

Chapter 5

Discussion

In this project, we have implemented a Finite Volume Method to solve the shallow water equations in 1D, succesfully simulating the dam break problem.

Chapter 6

Further work

In this project, we have implemented a Finite Volume Method to solve the shallow water equations in 1D, succesfully simulating the dam break problem. Although we attempted to extend this method to 2D, time constraints prevented us from completing it. We also began adapting the 1D method to accommodate a curved bottom surface instead of a flat one but couldn't finish this implementation either. Both extensions are promising, and it would be interesting to observe how the model performs with a curved bottom. Another interesting extension is to implement the FVM in 3D and use it to solve the SWE in spherical coordinates.

Bibliography

[1] Ilya Peshkov. Advanced numerical methods for environmental modeling. `https://www.unitn.it/dricam/1116/advanced-numerical-methods-environmental-modeling`, 2023. Lecture notes.

[2] E.F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows.* Oxford University Press, 2001.

[3] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, 2009.

[4] E.F. Toro. *Computational Algorithms for Shallow Warer Equations.* Springer, 2024.

[5] C.B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow.* Kluwer Academic Publishers, 1994.

[6] Wikipedia. Leibniz integral rule.