ON NUMERICAL SOLUTIONS OF THE SHALLOW WATER WAVE EQUATIONS

John Jakeman

October 2006

A thesis submitted in partial fulfilment of a Degree of
Bachelor of Science with Honours

Abstract

This thesis examines numerical solutions of the one-dimensional shallow water wave equations. A mathematical derivation and investigation of the equations is provided with emphasis on the important assumptions of shallow water theory and possible difficulties inherent in analytical and numerical calculations. An overview of the different numerical methods used to solve the shallow water wave equations is also given. Unlike most texts this review considers both finite volume and finite element methods and discusses their differences and similarities, advantages and disadvantages. A classification of these schemes is attempted to facilitate the differentiation of the various methods used to simulate shallow water flow. Emphasis was placed on a simple synthesis of appropriate techniques and where possible implementation issues specific to modelling the shallow water wave equations.

A second-order central-upwind Godunov-type finite volume method for solving the one-dimensional shallow water wave equations is constructed. This scheme can easily be extended to simulate two-dimensional flows and advantageously involves no Riemann solvers or characteristic decomposition. A high resolution is achieved due to the small amount of numerical dissipation introduced. The local propagation speeds of the rarefaction fans at each cell interface are used to estimate the size of the fan. The solution in these discontinuous regions are then averaged and combined with the estimates of the solutions in the smooth regions to obtain a final approximation. A fractional-step method is used to reduce the non-homogeneous equations into a sequence of augmented problems. The solution is then evolved through time using a second-order Runge-Kutta method. Another advantage of this method is that it can easily be extended to model two-dimensional flow. The scheme is second-order accurate when modelling smooth flows and first-order accurate when modelling flow characterised by shocks.

The central-upwind scheme presented in the thesis has been implemented in a modularised object-oriented software package written in `Python`. This code was developed from scratch, to facilitate greater understanding and more comprehensive evaluation of the numerical technique in question. Use of pre-built software may not have identified some of the numerical difficulties discussed below.

Numerical experiments found that the second-order central-upwind scheme requires a time discretisation of the corresponding order. When the spatial and temporal discretisations are inconsistent, spurious oscillations pollute the solution and the amplitude of these oscillations

increases, as the computational grid is refined. The second-order time stepping method implemented is less computationally efficient. Nevertheless the increase in accuracy and the inaccuracies of the first-order temporal discretisation justifies the increased computational effort.

A range of limiters were tested for the second-order central-upwind scheme. Of the six limiters investigated, three produced extremely poor results when modelling flow over dry beds. The alternative three limiters performed well in all situations, but underestimated the velocity of the advancing wet of a dry dam-break. Further work is needed to find an appropriate method to adjust the limiting procedure at wet and dry interfaces.

Employing the most accurate van Leer limiter, numerical approximations were compared against three types of analytical solutions to establish the veracity of the model. Excellent resolution of shocks, rarefaction fans and steady flow over rapidly varying topography is obtained. Further work to improve the simulation of the wetting and drying of topography is required. An initial attempt, based on expressions relating fluid area to the free surface elevation in partially wetted cells, did not provide an increase in accuracy. A more accurate prediction of the velocity in partially wetted cells is needed.

Contents

Acknowledgements

The work for this thesis was undertaken at the Department of Mathematics, The Australian National University, with financial assistance from the Australian Partnership for Advanced Computing. I would particularly like to thank Steve Roberts for his unerring support and guidance. His suggestions of research topics were stimulating and highly appreciated.

Special thanks must also go to Ole Nielsen. His enthusiasm was infectious and his advice and experience invaluable.

Finally I must thank my family, to whom I am indebted. Without their sacrifices and unconditional support not only during my honours year, but also throughout my three years of study previous, the work I have completed would not have been possible.

Acknowledgement of Sources and Claims to Originality

Except where otherwise indicated this thesis is my own original work.

John Jakeman

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Fluid mechanics, the study of the movement of fluid, can be used to describe and model a large number of complex systems, from jet streams expelled from black holes to the flow of water down the kitchen sink. The basic equations governing fluid motion are built on the principles of conservation of mass, momentum and energy. Numerous variations of these equations can be constructed from varying assumptions regarding the nature of the fluid, such as its viscosity and compressibility, and the dimensions and properties of the domain in which the fluid is situated. In fact it is often necessary to make limiting assumptions, that are acceptable for a particular application, in order to obtain approximate descriptions of flow that allow tractable analytical and numerical solutions.

The description of shallow water flow by what are known as the shallow water wave equations (SWWE) is one such approximation. These equations are used to represent the flow of water in a region in which the horizontal dimension of the water body, say wave length, greatly exceeds the depth. These equations arise from the basic equations of fluid mechanics for an inviscid and incompressible fluid. Generally a two-dimensional approximation of flow is used that assumes negligible vertical acceleration and a hydrostatic pressure distribution under which pressure increases linearly with depth. If the flow is characterised by some vertical acceleration then it is necessary to relax the hydrostatic pressure assumption. If a non-hydrostatic distribution is assumed variations of the shallow water wave equations, known as Boussinesq equations, can be obtained.

Shallow water flows are characterised by flow regions with a free surface, an impermeable

bottom topography, such as a sea floor, and horizontal velocity that dominates the flow field. Flow may be generated by gravitational forces associated with sloped beds, wind that creates stress on the free surface of the fluid or an applied pressure gradient, such as a tidal influence, a disturbance in the impermeable bottom, say from an earthquake, or a combination of two or all three.

The Shallow Water Wave equations have been used for many applications including:

- Dam Breaks — Water impoundment resulting from dam breaks poses a potential risk to life and infrastructure. Such events are often characterised by rapidly varying flows and shock waves. Zoppou and Roberts [60] use a shallow water model to simulate the sudden collapse of supply reservoirs in urban areas.

- Tsunami Propagation — Tsunamis pose a large risk to coastal communities all over the world. This has become agonizingly evident in the aftermath of the now-called Boxing-day Tsunami of 2004. For these reasons there is an increased focus on tsunami hazard mitigation (tsunami detection, forecasting and emergency preparedness) to limit loss of life and economic fallout resulting from such an event [52]. The shallow water wave equations or the Boussinesq-type variations are common descriptions used to model tsunami propagation [28, 56].

- Storm Surges — Storm surges are caused by the large wind velocities, pressure and high tides associated with large storms, especially tropical cyclones. For similar reasons given for tsunami events it is important to be able to model storm surges to identify areas of high risk and areas that may be protected by artificial barriers such as break walls. Examples of modelling storm surges can be found in [11, 17].

- Solute Transport — Prediction of solute transport concentrations in various fluid flows is important for many environmental and industrial applications. When developing ecological models of wetlands and estuaries information regarding the movement of pollutants and other forms of macro solutes which are often food sources is invaluable. Chertock et al. [12] detail the use of the shallow water wave equations for modelling the transport of passive pollutants. Other examples of can be found in [18, 19]

- River Flows and Inundation — Flow in rivers over floodplains is very accurately represented by shallow water flow. Ogink et al. [42] simulate such flows to determine flow

patterns and river heights during flooding. Goutal and Maurel [23] use the shallow water flow theory to model steady flow in rivers.

- Ecological Models — Macro particles found in rivers and oceans are often food sources for small marine animals. The concentration of these particles can thereby have an effect on the population size and distribution of these organisms and in turn on the ecosystem as a whole. Consequently shallow water modelling can be used to simulate the movement of these particles and organisms and used to determine and predict important ecological markers. Nixon and Noye [41], for example, use a shallow water model coupled with a dispersion model to predict the distribution of prawn larvae in the Spencer Gulf.

The shallow water wave equations form a non-linear hyperbolic system. The equations often admit discontinuous solutions even when the initial data is smooth and analytical solutions are limited to very few idealised cases. Computational methods therefore must be used to model most practical problems.

The majority of shallow water flow of practical interest is two-dimensional. However, two-dimensional numerical models of these flows are more complex and difficult to implement than the corresponding one-dimensional representations. Although few 'real' flows are truly one-dimensional, most computational methods for simulating two-dimensional flows are strongly based on the numerical techniques used to model one-dimensional flows [31]. In addition the simpler representation of one-dimensional models allows easier identification of computational issues that cause inaccurate simulation of flows, such as the violation of the entropy condition and spurious oscillations in the solution. Consequently this thesis focuses almost entirely on one-dimensional numerical techniques used to model shallow water flow.

1.2 Outline of the Thesis

The shallow water wave equations can be derived in a number of ways with varying initial assumptions and mathematical complexity. Chapter 2 presents a derivation of the two-dimensional shallow water wave equations. It aims to emphasise the important assumptions of shallow water theory. The chapter is divided into three sections: (i) a derivation of the fundamental equations of continuity and motion for a perfect inviscid fluid; (ii) a derivation of the shallow water wave equations from these fundamental equations; and (iii) a brief discussion of higher-order Boussinesq-type equations also used to describe shallow water flow.

Chapter 3 investigates the non-linear hyperbolic nature of the shallow water wave equations, using one-dimensional flow as a concrete example. Particular attention is given to characteristic theory, the Riemann problem and three idealised analytical solutions. Characteristic theory is used to examine the propagation of information in shallow water and introduce principles necessitated by the subsequent discussion of the Riemann problem and exact solutions. The next section describes the Riemann problem central to many hydrodynamical numerical techniques. And the final section presents three analytical solutions describing: (i) flows resulting from dam-breaks; (ii) oscillating planar flow in a parabolic canal; and (iii) steady flow over an obstacle. These problems were selected to offer further insight into the behaviour of the shallow water equations and provide means to establish the veracity of the numerical schemes presented later.

Due to the non-linear nature of the shallow water equations, numerical methods are required to solve most practical problems. The numerical techniques used to simulate shallow water flow fall into three categories: finite volume, finite element and spectral methods. The principles of both finite volume and finite element methods are discussed in Chapter 4. A discussion of the less popular spectral element methods is omitted here.

Chapter 5 develops a central-upwind Godunov-type finite volume method for solving the one-dimensional shallow water wave equations, based on the work of Kurganov et al. [30]. The scheme presented attempts to avoid some of the numerical difficulties inherent in the alternative approaches discussed in Chapter 4. Much attention is given to slope limiters, fractional step methods, temporal discretisation, and boundary conditions. Implementation considerations and the extension of the central-upwind scheme to model two-dimensional flows are also considered.

Model performance is addressed in Chapter 6. Specifically the analytical solutions presented in Chapter 3 are compared against the numerical approximations of the central-upwind method, in order to establish the veracity of the model. Attention is given both to the rate at which the numerical solution converges to the true solution and the robustness of certain frequently-used slope limiters. This chapter also addresses the need for consistency in the order of spatial and temporal discretisations of the central-upwind scheme and the resulting trade-off between accuracy and performance.

Finally, in Chapter 7, we attempt to improve the procedure used to model the wetting and drying of varying topography. The method presented is based on the work of Begnudelli and Sanders [3] and uses equations that relate the fluid area in a one-dimensional cell to the free surface in that cell.

Conclusions are presented in Chapter 8.

Chapter 2

Deriving The Shallow Water Wave Equations

Literature on the derivation of both the fundamental equations of fluid mechanics and the shallow water wave equations is comprehensive. Numerous approaches have been used, some adopting a more rigorous mathematical approach, others appealing to physical intuition and many more adopting a combination of both these methods. This chapter presents a derivation of the two-dimensional shallow water wave equations in a manner hoped to highlight the important assumptions required, that appeals to both the mathematical and physically minded, and best suits the demands of the numerical methods discussed later in this thesis.

Specifically this chapter first derives the fundamental equations of continuity and motion for a perfect inviscid fluid, which constitute two of the three well-known Euler equations. The Euler equation based on the conservation of energy is also presented, however its derivation is neglected as it is not needed in the derivation of the shallow water wave equations found here. From these two equations the various assumptions of shallow water theory are discussed and used to construct the depth-averaged shallow water equations for one-dimensional flow, in an open channel of regular cross-section with a horizontal bed. The last section of this chapter relaxes some of the assumptions of shallow water theory and introduces the resulting Boussiensq equations.

2.1 The Fundamental Equations of Fluid Mechanics

The governing equations of fluid flow, including shallow water flow, are based on the conservation of mass, momentum, and energy and the second law of thermodynamics. The principle of conservation requires that the three aforementioned fundamental quantities are neither created nor destroyed. The consideration of the conservation of mass is simple but the conservation of energy and momentum is more difficult, owing to the interrelationship between these quantities and the numerous phenomena that influence these quantities. For simplicity the following discussion, as with many classical texts [31, 13], begins by eliminating all influences on momentum and energy, save for redistribution and pressure, by initially only considering inviscid flows of perfect fluids free of forces except for pressure. Various forces can then be added to describe more complex flows.

The equation of motion and continuity are generally derived by adopting either the Lagrangian or Eulerian coordinate systems. The Lagrangian approach uses an evolving coordinate system that follows the trajectory of each fluid particle within a chosen region as it flows downstream. The boundary of this region continually evolves so as not to permit any fluid particle crossing it. Intuitively one can see that this allows a simple method of conserving mass and momentum within these regions. The Lagrangian method gives rise to equations that specify the location and the basic flow quantities, such as velocity and pressure, of each fluid particle within the flow field. This means that the resulting equations are expressed in basic flow quantities, such as velocity and pressure, of each fluid particle within the flow field in a coordinate system that moves with the fluid flow. In contrast the Eulerian approach describes fluid motion in terms of the fluxes of the various fluid quantities across the surface of a control volume that is fixed with respect to the coordinate system being used. This method does not consider the position of individual particles occupying a certain space, but rather evaluates basic flow quantities, such as mass and momentum, as a function of the coordinates of a point and time [37].

This chapter adopts the Lagrangian approach to derive the equation of continuity and the Eulerian approach to construct the equation of motion to emphasize the inherent differences in these two approaches and to enable better understanding of the Eulerian and Lagrangian numerical methods discussed in Section 4.

2.1.1 The Conservation of Mass

The conservation of mass is the first principle used to develop the basic equations of fluid mechanics. Simply the conservation of mass implies that the total mass of a closed system in a region Ω is constant and mass can neither be created nor destroyed.

Let us adopt the Eulerian approach and consider a region in space $\Omega \in \mathbb{R}^3$ in which a body of fluid is situated. Employing this method we denote the velocity of the fluid moving through the point $\vec{x} \in \Omega$ by $\vec{u}(\vec{x}, t) := (u, v, w)$. In addition assume that for each time t the fluid has a well-defined mass density $\rho(\vec{x}, t)$, where the functions $\vec{u}$ and ρ exist and are smooth enough so that the operations of calculus used in this thesis can be performed on them.

For any subregion $W_0 \in \Omega$, fixed in space with volume W_0, the mass of fluid in W_0 at time t is given by

$$m_t(W_0) = \int_{W_0} \rho \, dV$$

Note the subscript t does not denote differentiation. The rate of change of mass in W_0 is

$$\frac{d}{dt} m_t(W_0) = \frac{d}{dt} \int_{W_0} \rho \, dV \tag{2.1}$$

We must now consider the state of the fluid a short period in time later. At this time some of the fluid that occupied the region at time t will have passed out of W_0 and a portion of the fluid originally outside the region at time t will replace what has left.

Consider an infinitesimally small region on the surface ∂W_0 of the region W_0, as shown in Figure 2.1. The outflow of mass from W_0 is a function only of the velocity of the fluid on the surface ∂W_0, because mass is advected with the fluid leaving W_0. So the mass that flows out from a small area on the surface of W_0 in time Δt is $\rho \vec{u} \cdot \vec{n} \, dA \Delta t$. Here $\vec{n}$ is the outward unit normal to the area dA. Summing all such areas over the entire surface yields an expression for the rate of change of mass crossing ∂W_0 in the outward direction:

$$\frac{dm_t}{dt} = \lim_{\Delta t \to \infty} \frac{m_{out} - m_{in}}{\Delta t} = \frac{1}{\Delta t} \int_{\partial W_0} \rho \vec{u} \cdot \vec{n} \Delta t \, dA \tag{2.2}$$

Conservation of mass requires that the the rate of change of the mass of W_0 must be equal to the rate at which mass passes through ∂W_0 in the inward direction. That is

$$\int_{W_0} \frac{\partial}{\partial t} \rho \, dV = - \int_{\partial W_0} \rho \vec{u} \cdot \vec{n} \, dA \tag{2.3}$$

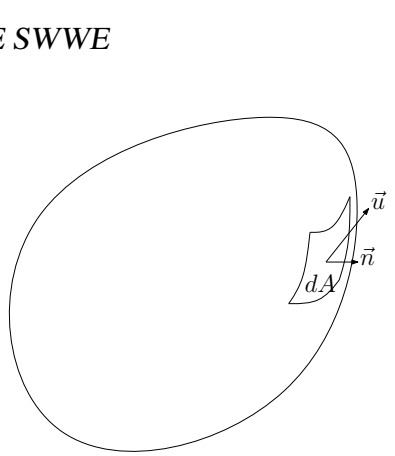


Figure 2.1: The mass outflow from an incremental area on the region W_0 over a small period in time Δt

Recall that the region W_0 is fixed in space and so the limits of integration of the integral on the left hand side do not change with time. This means that the derivative sign can be placed within the integrand.

Using the divergence theorem

$$\int_V \vec{\nabla} \cdot \vec{u}\, dV = \int_{\partial V} \vec{u} \cdot \vec{n}\, dA \tag{2.4}$$

equation (2.3) becomes

$$\int_{W_0} \left[\frac{\partial}{\partial t}\rho + \vec{\nabla} \cdot (\rho \vec{u}) \right] dV = 0 \tag{2.5}$$

Thus far the limits of the control volume have not been defined. So the control volume can be chosen as any non-zero region of the integrand, which violates (2.5). Therefore the integrand must be zero everywhere and

$$\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.6}$$

Equation (2.6) is known as Euler's equation of continuity.

2.1.2 The Conservation of Momentum

Now let us set aside the Eulerian approach and adopt the Lagrangian coordinate system to construct Euler's equations of motion from the principle of conservation of momentum. To do this we must define some new notation. The particles of fluid within the domain travel well-defined trajectories. Let $\vec{\phi}(\vec{x}, t)$ be the trajectory followed by a particle that originated from the

position $\vec{x}_0 \in \Omega$ at time $t = 0$. This particle travels with the velocity $\vec{u}(\vec{x}, t) = \frac{\partial}{\partial t}\vec{\phi}(\vec{x}_0, t)$ as it moves through the point $\vec{x} \in \Omega$ at time t. Again assume that for all t, that is for $0 \leq t \leq \infty$, the fluid within the domain has a well-defined mass density $\rho(\vec{x}, t)$.

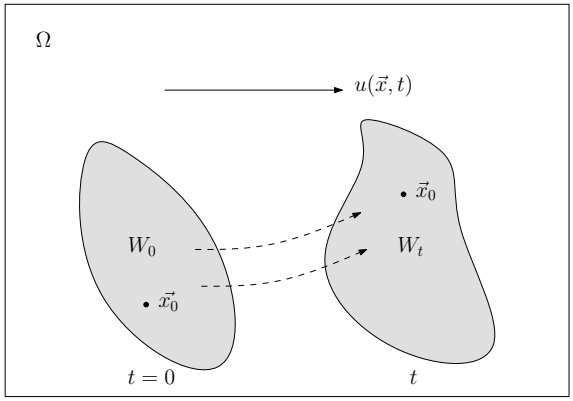


Figure 2.2: The trajectory of particles, initially contained in a region W_0 at $t = 0$, through a larger region Ω flowing with velocity $u(\vec{x}, t)$. W_t is an image of W_0 at some $t > 0$.

A Lagrangian coordinate system follows the trajectory of each fluid particle within a chosen region $W(t)$ as it flows downstream. The boundary of this region continually evolves so as not to permit any fluid particle crossing it. Refer to Figure 2.2. This means, that unlike the region W_0 considered when using an Eulerian coordinate system, W must be a function of time and the total momentum in a region W_t is given by

$$\int_{W_t} \rho(\vec{x}, t)\vec{u}(\vec{x}, t)\, d\vec{x}$$

We will follow this region as it evolves with the velocity field of the fluid, defining W_t so that no fluid particles enter or leave the region at any time.

Newton's law of motion states that the rate of change in momentum is equal to the applied force $\vec{F}$. That is

$$\vec{F} = \frac{d}{dt}\int_{W_t} \rho(\vec{x}, t)\vec{u}(\vec{x}, t)\, d\vec{x}$$

and by changing variables we see that

$$\begin{aligned} \vec{F} &= \int_{W_0} \frac{d}{dt}\rho(\vec{\phi}(\vec{x}_0, t), t)\vec{u}((\vec{x}_0, t), t)J(\vec{x}_0, t)\, d\vec{x}_0 \\ &= \int_{W_0} J(\vec{x}_0, t)\frac{d}{dt}\rho\vec{u}\, d\vec{x}_0 + \int_{W_0} \rho\vec{u}\frac{\partial}{\partial t}J(\vec{x}_0, t)\, d\vec{x}_0 \end{aligned} \tag{2.7}$$

Here $J = det(\nabla\vec{\phi})$ is the Jacobian of the transformation $\vec{\phi}$ and we have dropped the arguments of u and ρ for clarity. Whenever associated with $J(\vec{x}_0, t)$ these two quantities are functions of $(\vec{\phi}(\vec{x}_0, t), t)$. Otherwise they are functions of (x, t).

To simplify this equation consider the following lemma provided by Chorin and Marsden [13] that states

Lemma 2.1.1.

$$\frac{\partial}{\partial t}J(\vec{x}_0, t) = J(\vec{x}_0, t)\nabla \cdot u(\vec{\phi}(\vec{x}_0, t), t)$$

Applying this result and the chain rule to (2.7) yields

$$\begin{aligned}\vec{F} &= \int_{W_0} J(\vec{x}_0, t)\frac{\partial}{\partial t}\rho\vec{u}\, d\vec{x}_0 + \int_{W_0} \vec{u} \cdot \nabla(\rho\vec{u})J(\vec{x}_0, t)\, d\vec{x}_0 \\ &\quad + \int_{W_0} (\rho\vec{u}\nabla \cdot \vec{u})J(\vec{x}_0, t)\, d\vec{x}_0\end{aligned}$$

Noting that

$$\begin{aligned}\vec{u} \cdot \nabla(\rho\vec{u}) + \rho\vec{u}\nabla \cdot \vec{u} &= u\frac{\partial\rho\vec{u}}{\partial x} + v\frac{\partial\rho\vec{u}}{\partial y} + w\frac{\partial\rho\vec{u}}{\partial z} + \rho\vec{u}\frac{\partial u}{\partial x} + \rho\vec{u}\frac{\partial v}{\partial y} + \rho\vec{u}\frac{\partial w}{\partial z} \\ &= \frac{\partial\rho\vec{u}\vec{u}}{\partial x} + \frac{\partial\rho\vec{u}\vec{u}}{\partial y} + \frac{\partial\rho\vec{u}\vec{u}}{\partial z} \\ &= \nabla \cdot (\rho\vec{u}\vec{u})\end{aligned}$$

where $\vec{u}\vec{u}$ is a dyadic product $(u_i u_j)$ we obtain

$$\begin{aligned}\vec{F} &= \int_{W_0} \left[\frac{\partial}{\partial t}\rho\vec{u} + \nabla \cdot \rho\vec{u}\vec{u}\right] J(\vec{x}_0, t)\, d\vec{x}_0 \\ &= \int_{W_t} \left[\frac{\partial}{\partial t}\rho\vec{u} + \nabla \cdot \rho\vec{u}\vec{u}\right] d\vec{x}\end{aligned}$$

Using the continuity equation we see that

$$\begin{aligned}\frac{\partial}{\partial t}\rho\vec{u} + \nabla \cdot (\rho\vec{u}\vec{u}) &= \frac{\partial}{\partial t}\rho u_i + \frac{\partial}{\partial x_j}\rho u_i u_j \\ &= u_i\frac{\partial\rho}{\partial t} + \rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} + u_i\frac{\partial}{\partial x_j}\rho u_j \qquad (2.8) \\ &= \rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} \\ &= \rho\frac{D}{Dt}\vec{u} \qquad (2.9)\end{aligned}$$

where $\frac{D}{Dt}$ is the material derivative which takes into account the fact the fluid is moving and the positions of the particles within the fluid change with time.

So finally

$$\vec{F} = \int_{W_t} \rho \frac{D}{Dt} \vec{u}\, d\vec{x}$$

The forces acting on a region of fluid are of only three types. The first is the pressure force acting on the boundary of a region of fluid. The second is the external or body forces which exert a force per unit volume on the fluid, and the third type are known as stress forces that act on the surface of the fluid. For now we will assume that there are no stress forces acting on our fluid.

To evaluate the pressure force exerted across the surface of a region of fluid let us again consider a region of fluid $W(t)$ at time t contained by the surface ∂W. Now define S to be an infinitessimally small area on $\partial W(t)$. The pressure force on S per unit area in the direction of the outward unit normal $\vec{n}$ is $p\vec{n}$. The total force $\vec{F_p}$ exerted on $W(t)$ by pressure p across its surface $\vec{F_p}$ is obtained by summing all such areas over the entire surface. That is

$$\vec{F_p} = -\int_{\partial W_t} p\vec{n}\, dA = -\int_{W_t} \nabla p\, dV$$

We must also consider the total body force acting on W_t is given by

$$F_b = \int_{W_t} \rho \vec{b}\, dV$$

where b is the body force per unit mass. Putting all this together we obtain the equations of motion

$$\int_{W_t} \left[\frac{\partial}{\partial t} \rho\vec{u} + \rho(\vec{u} \cdot \vec{\nabla})\vec{u} \right] dV = -\int_{W(t)} \left[\vec{\nabla}\, p + \rho\vec{b} \right] dV$$

Finally assuming that gravity $\vec{g}$ provides the only body force acting on our fluid (i.e. $\vec{b} = \vec{g}$) Euler's equations of motion in differentiable form are, in the x-direction,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{2.10}$$

in the y-direction

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y} \tag{2.11}$$

and in the z-direction

$$\frac{\partial v}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g \tag{2.12}$$

2.1.3 Conservation of Energy

Conservation of energy states that the total amount of energy (including potential energy) in an isolated system remains constant. An increase in energy in space and time must be accompanied by an equal decrease in energy at another point. In one dimension the change in total energy in $[x_1, x_2]$ in the interval $[t_1, t_2]$ is equal to the net energy flow through the boundaries of $[x_1, x_2]$ over the same time plus the net energy change due to pressure on the boundaries of $[x_1, x_2]$. That is

$$\begin{aligned}\int_{x_1}^{x_2} &[\rho(x,t_2)e_T(x,t_2) - \rho(x,t_1)e_T(x,t_1)]\ dx = \\ - &\int_{t_1}^{t_2} [\rho(x_2,t)u(x_2,t)e_T(x_2,t) - \rho(x_1,t)u(x_1,t)e_T(x_1,t)]\ dt \\ - &\int_{t_1}^{t_2} [p(x_2,t)u(x_2,t) - p(x_1,t)u(x_1,t)]\ dt\end{aligned}$$

where $e_T(x,t)$ is the total energy per unit mass, $\rho(x,t)e_T(x,t)$ is the total energy per unit volume and $\rho(x,t)u(x,t)e_T(x,t)$ is the instantaneous total energy flux.

2.2 The Depth-Averaged Shallow Water Wave Equations

There are several approaches that can be used to develop the depth-averaged shallow water wave equations. The following most closely follows the text of Ligget [37] which begins with the aforementioned hydrodynamic equations to stress the necessary assumptions and approximations of shallow water theory. An extended discussion is given to the hydrostatic pressure assumption, essential to shallow water theory, and the appropriate boundary conditions needed to derive the equations and applying a hydrostatic pressure assumption. The limits of integration used to average the flow in shallow water are also changed to best suit the numerical method discussed later.

Shallow water theory makes three important assumptions regarding the nature of the fluid being studied. The fluid is assumed to be incompressible, irrotational and inviscid. The latter two assumptions were made when deriving Euler's equations. The construction of the depth-averaged shallow water equations also requires three main assumptions about the spatial domain. Shallow water flows only exist in bodies of fluid with a vertical extent D of Ω that is 'much smaller' than the horizontal extent L. By 'much smaller' we mean the ratio of horizontal and vertical extent

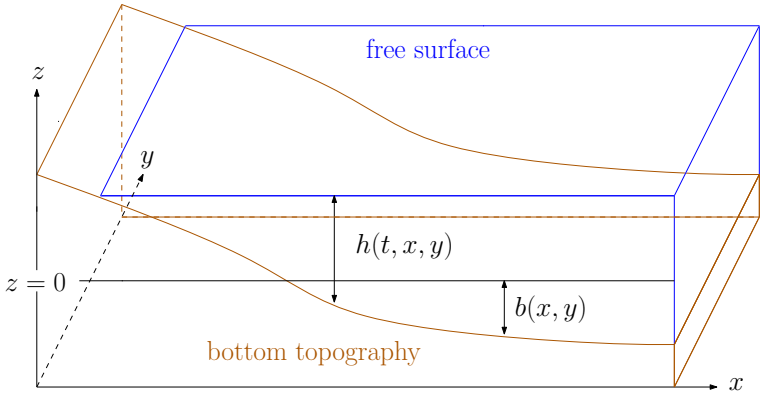


Figure 2.3: The Cartesian coordinate system used when defining the shallow water wave equations. $b(x, y)$ is the elevation of the impermeable bottom topography and $h(t, x, y)$ is the depth of the water.

is much smaller than one. That is $\epsilon = D/L \ll 1$. In addition pressure is assumed to increase linearly with depth and there is assumed to be negligible vertical acceleration within the fluid.

2.2.1 Coordinate System

When dealing with the shallow water wave equations the "β-plane" coordinate system is usually assumed [32]. In such a situation the x-y plane is taken tangent to the Earth at the latitude under consideration. The x-axis is taken in the direction of the Earth's latitude and the y-axis in the direction of the Earth's longitude. The z-axis lies along the outward normal to the Earth's surface. The free surface occurs at $z = b + h$ where $z \in [b(x, y), b(x, y) + h(t, x, y)]$. Here b is the elevation of the impermeable bottom of the domain relative to a particular datum, h is the depth of water , and t is the non-negative time variable — see Figure 2.3. Ignoring the curvature of the Earth the plane $z = 0$ is assumed to be the free surface of the fluid body when it is at rest.

The domain Ω consists of a single region, such as an ocean or experimental fluid tank. The vertical boundaries of this region are denoted by $\partial\Omega$ and consist of "open" boundaries, for which the inflow and outflow must be specified, or "closed" boundaries such as a solid wall.

2.2.2 Boundary Conditions

Before deriving the shallow water wave equations we must first specify the boundary conditions of the problem domain. There are three types of boundaries that must be considered: the "open" or "closed" vertical surfaces of the domain $\partial\Omega$, the free surface, and the impermeable bottom of the domain.

The boundary conditions on $\partial\Omega$ will not be dealt with in this section, as the conditions on these boundaries are problem-dependent and do not need to be taken into account when deriving the shallow water wave equations. Initial conditions are also problem dependent and as such are not addressed here.

Consider the free surface defined by the equation

$$z = f(x, y, t)$$

The velocity of this free surface is given by $\vec{v} = (0, 0, \partial f/\partial t)$ and the velocity of a particle within the fluid is given by $\vec{u} = (u, v, w)$. It is reasonable to assume that fluid particles cannot cross the free surface of Ω so that

$$\vec{v} \cdot \vec{n} = \vec{u} \cdot \vec{n}$$

where $\vec{n}$ is the unit outward normal vector to the free surface given by

$$\vec{n} = \left(-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, 1 \right)$$

Equating terms yields

$$\frac{\partial f}{\partial t} = -u\frac{\partial f}{\partial x} - v\frac{\partial f}{\partial y} + w$$

Noting that $f(x, y, t) = b(x, y) + h(x, y, t)$ gives rise to the surface condition

$$\frac{\partial h}{\partial t} + \hat{u}\frac{\partial}{\partial x}(b + h) + \hat{v}\frac{\partial}{\partial y}(b + h) - \hat{w} = 0 \tag{2.13}$$

Similarly fluid cannot cross the impermeable bottom of Ω so

$$\breve{u}\frac{\partial b}{\partial x} + \breve{v}\frac{\partial b}{\partial y} - \breve{w} = 0 \tag{2.14}$$

Here the hat indicates that the quantities are evaluated at the free surface of the fluid ($z = b + h$) and the breve indicates that the quantities are evaluated at the impermeable bottom of Ω ($z = b$). We have also assumed a non-uniform vertical velocity distribution.

Across the impermeable bottom the "non-slip" condition can be applied, that is

$$\breve{u} = 0 \quad , \quad \breve{v} = 0 \quad , \quad \breve{w} = 0$$

If friction had an important effect on flow then this last condition would need to be changed and the resulting formulation of the equations would differ sightly.

2.2.3 Conservation of Mass

Consider the assumption that the density of the fluid is constant throughout Ω. We then note that the continuity equation (2.6) contains a substantial derivative that shows the rate of change in reference to an individual particle. The physical interpretation of the equation implies that the change of density of a fluid particle is equal to the expansion of the fluid. For an incompressible fluid, the change in the density of a fluid particle is zero — i.e. $d\rho/dt = 0$. Using this result (2.6) becomes

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.15}$$

Using the coordinate system defined in Section 2.2.1 and integrating this equation over the vertical extent of the fluid yields

$$\int_b^{b+h} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dz = 0 \tag{2.16}$$

where $\vec{u}(\vec{x}, t) = (u(x, y, z, t), v(x, y, z, t), w(x, y, z, t))$. Next we use need to reverse the order of integration and differentiation in (2.16).

Theorem 2.2.1. *Leibniz' Theorem*

$$\frac{\partial}{\partial t} \int_{a(y,t)}^{b(y,t)} f(x, y, t)\, dx = \int_{a(y,t)}^{b(y,t)} \frac{\partial f}{\partial t}\, dx - f(a, y, t)\frac{\partial a}{\partial t} + f\ (b, y, t)\frac{\partial b}{\partial t}$$

Note again that we will neglect the arguments of functions in the following to improve clarity. Arguments are still included when clarification may be needed. Applying Leibniz' theorem

$$\begin{aligned}\int_b^{b+h}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)dz &= \frac{\partial}{\partial x}\int_b^{b+h} u\,dz-\hat{u}\frac{\partial}{\partial x}(b+h)+\breve{u}\frac{\partial b}{\partial x}\\ &\quad+\frac{\partial}{\partial y}\int_b^{b+h} v\,dz-\hat{v}\frac{\partial}{\partial y}(b+h)+\breve{v}\frac{\partial b}{\partial y}\\ &\quad+\int_b^{b+h}\frac{\partial w}{\partial z}\,dz\\ &= 0 \end{aligned} \tag{2.17}$$

Now define the average velocity over the depth of vertical extent of Ω in the x and y directions to be

$$\bar{u}=\frac{1}{h}\int_b^{b+h} u\,dz \quad \text{and} \quad \bar{v}=\frac{1}{h}\int_b^{b+h} v\,dz$$

respectively. Substituting these quantities into (2.17) yields

$$\begin{aligned}\int_b^{b+h}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)dz &= \frac{\partial}{\partial x}h\bar{u}+\frac{\partial}{\partial y}h\bar{v}-\hat{u}\frac{\partial}{\partial x}(b+h)-\hat{v}\frac{\partial}{\partial y}(b+h)\\ &\quad+\breve{u}\frac{\partial b}{\partial x}+\breve{v}\frac{\partial b}{\partial y}+\hat{w}-\breve{w}\\ &= 0\end{aligned}$$

Substituting in the boundary conditions at the surface (2.13) and along the bottom of the domain (2.14), the above equation simplifies to give the first of three two-dimensional depth-averaged shallow water wave equations

$$\frac{\partial}{\partial t}h+\frac{\partial}{\partial x}h\bar{u}+\frac{\partial}{\partial y}h\bar{v}=0 \tag{2.18}$$

2.2.4 Conservation of Momentum

Euler's equations of motion can be used to obtain the remaining two-dimensional shallow water wave equations. Integrating the momentum equation in the x-direction (2.10) over the vertical extent of the fluid yields

$$\int_b^{b+h}\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial x}\right)dz=0 \tag{2.19}$$

Using (2.15)

$$\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} = u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}$$

and substituting this expression into (2.19) yields

$$\int_b^{b+h} \left(\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial x} \right) dz = 0 \tag{2.20}$$

For ease of representation let us evaluate each term of the integrand separately. Starting with the time derivative term we use Leibniz's' theorem to obtain

$$\begin{aligned} \int_b^{b+h} \frac{\partial u}{\partial t}\, dz &= \frac{\partial}{\partial t}\int_b^{b+h} u\, dz + \breve{u}\frac{\partial b}{\partial t} - \hat{u}\frac{\partial}{\partial t}(b+h) \\ &= \frac{\partial}{\partial t}\int_b^{b+h} u\, dz - \hat{u}\frac{\partial}{\partial t}(b+h) \\ &= \frac{\partial}{\partial t}(\bar{u}h) - \hat{u}\frac{\partial}{\partial t}h \end{aligned}$$

Similarly

$$\int_b^{b+h} \frac{\partial}{\partial x}u^2\, dz = \frac{\partial}{\partial x}\int_b^{b+h} u^2\, dz + \breve{u}^2\frac{\partial b}{\partial x} - \hat{u}^2\frac{\partial}{\partial x}(b+h) \tag{2.21}$$

By averaging the fluid velocities in the vertical direction we can relax the assumption that limits the shallow water equations to fluid flow with no vertical acceleration. However by defining the above quantities the shallow water wave equations no longer conserve momentum exactly. With this in mind let us define the momentum correlation factors

$$\beta_{xx} = \frac{\int_b^{b+h} u^2\, dz}{\bar{u}^2 h} \quad \text{and} \quad \beta_{xy} = \frac{\int_b^{b+h} uv\, dz}{\bar{u}\bar{v}h}$$

which enables us to simplify (2.21) and the similar equation of the $\frac{\partial uv}{\partial y}$ term of (2.20). Thus

$$\int_b^{b+h} \frac{\partial}{\partial x}u^2\, dz = \frac{\partial}{\partial x}(\beta_{xx}\bar{u}^2 h) + \breve{u}^2\frac{\partial b}{\partial x} - \hat{u}^2\frac{\partial}{\partial x}(b+h) \tag{2.22}$$

and

$$\int_b^{b+h} \frac{\partial}{\partial x}uv\, dz = \frac{\partial}{\partial y}(\beta_{xy}\bar{u}\bar{v}h) + \breve{u}\breve{v}\frac{\partial b}{\partial y} - \hat{u}\,\hat{v}\frac{\partial}{\partial y}(b+h) \tag{2.23}$$

The third term of (2.20 is)

$$\int_b^{b+h} \frac{\partial}{\partial x} uw \, dz = \hat{u}\hat{w} - \breve{u}\breve{w} \tag{2.24}$$

We must now make the important assumption that the pressure of the fluid within Ω increases linearly with depth. That is

$$p = \rho g(z_1 - z_0)$$

where $z_1 - z_0$ is the vertical distance between two water levels. This condition is known as the hydrostatic pressure assumption and is derived from the basic premise of shallow water theory that restricts flow to situations where the fluid depth D is much smaller than the characteristic horizontal dimensions L. That is $\epsilon = D/L \ll 1$.

Let us denote U and W to be the characteristic fluid velocities in the horizontal $x - y$ plane and in the vertical direction respectively, and define the following dimensionless quantities accordingly:

$$x^\star = \frac{x}{L} \quad y^\star = \frac{y}{L} \qquad z^\star = \frac{z}{D} \quad u^\star = \frac{u}{U} \quad v^\star = \frac{v}{U} \quad w^\star = \frac{w}{W}$$
$$t^\star = \frac{Ut}{L} \quad p^\star = \frac{p}{\rho U^2} \quad g^\star = \frac{gU^2}{D}$$

Substituting these dimensionless quantities into the continuity equation gives

$$\frac{U}{L}\frac{\partial u^\star}{\partial x^\star} + \frac{U}{L}\frac{\partial v^\star}{\partial y^\star} + \frac{W}{D}\frac{\partial w^\star}{\partial z^\star} = 0$$

For this equation to be satisfied $W \approx O(UD/L)$. To show that pressure is hydrostatic we must consider the dimensionless form of the vertical momentum equation

$$\frac{U^2 D}{L^2}\left(\frac{\partial w^\star}{\partial t^\star} + u^\star\frac{\partial w^\star}{\partial x^\star} + v^\star\frac{\partial w^\star}{\partial y^\star} + w^\star\frac{\partial w^\star}{\partial z^\star}\right) = -\frac{U^2}{D}\frac{\partial p^\star}{\partial z^\star} - \frac{D}{U^2}g^\star$$

Multiplying through by D/U^2 yields

$$\frac{D^2}{L^2}\left(\frac{\partial w^\star}{\partial t^\star} + u^\star\frac{\partial w^\star}{\partial x^\star} + v^\star\frac{\partial w^\star}{\partial y^\star} + w^\star\frac{\partial w^\star}{\partial z^\star}\right) = -\frac{\partial p^\star}{\partial z^\star} - \frac{D^2}{U^4}g^\star \tag{2.25}$$

All the terms on the left hand side of (2.25) are all of order $\epsilon^2 = (D/L)^2$ and can be neglected. When observing shallow water flow in an ocean with a characteristic depth of 5×10^3m and a characteristic width of 1×10^7m, these terms are of the order 10^{-8}. This means that the pressure derivative can be approximated as

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

Integrating from the surface to some depth z gives

$$p = \rho g(b + h - z) + p(t, x, y, b + h)$$

where $p(t, x, y, b + h)$ is the pressure at the surface of the free surface of the fluid. Only the gradient components of the pressure appear in the equations of motion. So assuming that the atmospheric pressure is constant at the surface we can choose this pressure to be zero without complication. We have now arrived at the hydrostatic pressure assumption which states that at any vertical elevation within the fluid z the pressure is given by

$$p = \rho g(b + h - z)$$

Using this result we can now integrate the last term of (2.20) to obtain

$$\begin{aligned}
\int_b^{b+h} \frac{\partial p}{\partial x}\, dz &= \rho g \int_b^{b+h} \frac{\partial}{\partial x}(h + b - z)\, dz \\
&= \rho g \left(\frac{\partial}{\partial x} \int_b^{b+h} (h + b - z)\, dz + h\frac{\partial b}{\partial x} \right) \\
&= \rho g \left(\frac{1}{2}\frac{\partial}{\partial x} h^2 + h\frac{\partial b}{\partial x} \right) \qquad (2.26)
\end{aligned}$$

Substituting (2.22), (2.23), (2.24), and (2.26) into (2.20) yields

$$\begin{aligned}
\frac{\partial}{\partial t}(\bar{u}h) &- \hat{u}\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(\beta_{xx}\bar{u}^2 h) + \breve{u}^2\frac{\partial b}{\partial x} - \hat{u}^2\frac{\partial}{\partial x}(b + h) \\
&+ \frac{\partial}{\partial y}(\beta_{xy}\bar{u}\bar{v}h) + \breve{u}\breve{v}\frac{\partial b}{\partial y} - \hat{u}\hat{v}\frac{\partial}{\partial y}(b + h) + \hat{u}\hat{w} - \breve{u}\breve{w} \\
&+ g\left(\frac{1}{2}\frac{\partial}{\partial x} h^2 + h\frac{\partial b}{\partial x} \right) = 0
\end{aligned}$$

By applying the boundary conditions at the free surface (2.13) and the impermeable bottom (2.14) we obtain

$$\frac{\partial}{\partial t}(\bar{u}h) + \frac{\partial}{\partial x}(\beta_{xx}\bar{u}^2 h) + \frac{\partial}{\partial y}(\beta_{xy}\bar{u}\bar{v}h) + g\frac{1}{2}\frac{\partial h^2}{\partial x} = -gh\frac{\partial b}{\partial x}$$

Setting the momentum correlation factors to unity yields

$$\frac{\partial}{\partial t}(\bar{u}h) + \frac{\partial}{\partial x}(\bar{u}^2 h) + \frac{\partial}{\partial y}(\bar{u}\bar{v}h) + g\frac{1}{2}\frac{\partial h^2}{\partial x} = -gh\frac{\partial b}{\partial x} \tag{2.27}$$

and

$$\frac{\partial}{\partial t}(\bar{u}h) + \frac{\partial}{\partial x}(\bar{u}\bar{v}h) + \frac{\partial}{\partial y}(\bar{v}^2 h) + g\frac{1}{2}\frac{\partial h^2}{\partial y} = -gh\frac{\partial b}{\partial y} \tag{2.28}$$

These two equations in combination with (2.18) constitute one form of the depth-averaged shallow water wave equations. By setting the momentum correlation factors to one we have implicitly made the assumption that the horizontal velocities are uniform throughout the depth of the fluid. Nevertheless these correlation factors can take on other values and become very important when considering non-uniform vertical velocity distributions [29].

The exact nature of the shallow water wave equations depends on the number and types of forces assumed to affect shallow water flow. Here we have only considered the influence of gravity on the motion of fluids. However many other forces affect shallow water flow including friction, wind shear, atmospheric pressure gradients, and Coriolis. Frequently terms are simply added to the right hand side of (2.27) and (2.28) to model the effects of these forces. For example Zoppou and Roberts [61] add an additional source term to the right hand side of(2.27)

$$S_{f_x} = \frac{u\eta\sqrt{u^2+v^2}}{h^{4/3}}$$

and similarly to (2.28)

$$S_{f_y} = \frac{v\eta\sqrt{u^2+v^2}}{h^{4/3}}$$

to model flow affected by friction. Here η is the Manning resistance coefficient. The derivation and proprieties of these terms and other similar terms will not be considered in this thesis. Instead, throughout the remainder of this thesis we will consider the general form of the shallow water wave equations

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \\ vh \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} vh \\ uvh \\ v^2h + gh^2/2 \end{bmatrix} = S \tag{2.29}$$

where S is a generic source term representing the external forces affecting shallow water flow and is given by

$$S = \begin{bmatrix} 0 \\ -ghb_x \\ -ghb_y \end{bmatrix} \tag{2.30}$$

unless otherwise stated.

2.3 Boussinesq-Type Equations

The Boussinesq equations are an extension of the shallow water wave equations. As stated previously, depth-averaged shallow water theory assumes that the ratio of water depth to wavelength is much less than one and that there is a hydrostatic pressure distribution and negligible vertical acceleration. Boussinesq descriptions of shallow water flow relax the two latter assumptions and weaken the first so that they can be used in slightly deeper water and rapidly varying flows.

There is, however, no unique Boussinesq equation. Many different forms of the equations can be derived based on varying factors regarding the choice of horizontal velocity variable to be used, the types of equations used in the derivation process, and the fact that spatial and time derivatives can be to an extent interchanged due to linear theory.

Shocks describe the situation where the fluid motion undergoes a sudden transition in level between two different uniform flows of water. Both the shallow water and Boussinesq equations can be used to model these discontinuous flows, but only the latter can reflect the observed oscillations near the discontinuity. Generally it does not matter that the shallow water equations do not capture these oscillations as the behaviour of surface waves near this region is not important to the prediction of the downstream flow or the shock front. However in some cases these 'ripples' may be important and the Boussinesq equations (and Euler equations) provide the only means of studying this phenomenon.

The following one-dimensional equations are widely considered as the 'standard form' of the Boussinesq equations and were derived by Peregrine [44] in 1967

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(h\bar{u}) = 0$$

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} - \frac{h^2}{3}\frac{\partial^3 \bar{u}}{\partial x^2 \partial t} = -g\frac{\partial}{\partial x}(b+h)$$

These equations use the depth-averaged velocity $\bar{u}$ but are only applicable to a restricted set of conditions. Much work has been undertaken to extend these equations to achieve various goals such as removing the flat bed restriction of Peregrine's equations. A number of enhanced and higher-order Boussinesq equations have been formulated to improve both the linear and nonlinear properties of the equations and various numerical techniques have been employed to solve these equations [38, 1, 21]. In general numerical modelling of Boussinesq equations is more difficult than the modelling of the shallow water equations because of their added complexity.

Chapter 3

Properties of the Shallow Water Wave Equations

This chapter investigates the important mathematical properties of the shallow water wave equations. The following is aimed to provide insight into the behaviour of the shallow water equation and introduce concepts central to the numerical schemes presented in later chapters. In particular we analyse the characteristic structure of the shallow water wave equations and study solutions of the Riemann problem which are central to many numerical methods. The remainder of this chapter is devoted to developing a number of analytical solutions that are used to establish the veracity of numerical schemes in Chapter 6.

3.1 Behaviour of the Shallow Water Wave Equations

The general conservative form of the one-dimensional shallow water wave equations is given by

$$\frac{\partial}{\partial t}q(x,t) + \frac{\partial}{\partial x}f(q(x,t)) = S \tag{3.1}$$

where q is the vector of conserved variables, f is the flux vector in the x direction and S represents the source vector. The vectors q and f can be expressed in terms of the primary variables, u and h, as

$$q = \begin{bmatrix} h \\ uh \end{bmatrix}, \quad f = \begin{bmatrix} uh \\ u^2h + gh^2/2 \end{bmatrix} \tag{3.2}$$

Here u is the x component of the depth-averaged fluid velocity, h is the water depth, and g is the acceleration due to gravity. Following the derivation in Chapter 2 the source term S is given by

$$S = \begin{bmatrix} 0 \\ ghS_{0_x} \end{bmatrix} \tag{3.3}$$

where S_0 is the bed slope. This source term can be modified to include other driving forces such as friction and wind shear. However let us temporarily neglect the source terms and restrict our analysis to the homogeneous system

$$q_t + f(q)_x = 0 \tag{3.4}$$

Alternatively these equations can be written in the quasilinear form

$$q_t + f'(q)q_x = 0 \tag{3.5}$$

where the Jacobian matrix $f'(q)$ of f with respect to q is

$$f'(q) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}$$

The homogeneous system (3.4) is strictly hyperbolic since the Jacobian matrix $f'(q)$ is diagonalisable and has real distinct eigenvalues

$$\lambda_1 = u - \sqrt{gh} \quad \text{and} \quad \lambda_2 = u + \sqrt{gh}$$

with the corresponding eigenvectors

$$r_1 = \begin{bmatrix} 1 \\ u - \sqrt{gh} \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ u + \sqrt{gh} \end{bmatrix}$$

For hyperbolic systems such as (3.4) these eigenvalues correspond to the propagation velocities of the various components of the wave and are often called characteristic speeds. In the case of shallow water flow these velocities are dependent on the velocity of the flow u and the depth of the water h. Specifically the eigenvalues correspond to the gradients of the two families of characteristic curves associated with shallow water flow

$$C_1 : \frac{dx}{dt} = u - \sqrt{gh} \quad \text{and} \quad C_2 : \frac{dx}{dt} = u + \sqrt{gh} \tag{3.6}$$

The curvature of these curves depends on the point through which these curves pass at $t = 0$. The curves are constant in regions where u and h are constant. For a system of m equations there

will be m characteristic families. The functions $u \pm 2\sqrt{gh}$ are constant along these curves and are known as Riemann invariants and give rise to the following relations

$$u - 2\sqrt{gh} = k_1 \quad \text{and} \quad u + 2\sqrt{gh} = k_2 \tag{3.7}$$

which for constants m, k_1, and k_2 are constant along the curves C_1 and C_2, respectively. Given an initial condition for u and h, (3.6) and (3.7) can be completely determined to give a solution to the problem. Note that k_1 and k_2 will be different on curves originating at different points in space at $t = 0$.

Consider a one-dimensional problem for which $u(x,t)$ and $h(x,t)$ are known for all values of x at $t = 0$. To determine the solution for $t > 0$ assume that

$$u(x,0) = d(x)$$
$$\sqrt{gh(x,0)} = f(x)$$

where u and f are known functions. We can now approximate the values of u and gh at some small time in the future Δt. This is achieved (Figure 3.1) by selecting points, $1, 2, 3$ and 4, on the x-axis at a small distance Δx apart, for which u and gh are known, and calculating the slopes of C_1 and C_2. Straight line segments can then be drawn tangentially to the characteristic curves at these points until they intersect at the points $5, 6$ and 7. These points will provide good approximations of the position of the intersections of the characteristics, provided Δx is sufficiently small. From (3.7) we can determine the values of x and t at points $5, 6$ and 7 and approximate u and gh using

$$u - 2\sqrt{gh} = d - 2f$$
$$u + 2\sqrt{gh} = d + 2f$$

along C_1 and C_2 respectively. Once u and gh are known the slopes of C_1 and C_2 can be determined from (3.6) and the previous process can be repeated to obtain estimates of points 8 and 9 (see Figure 3.2). Now consider the point 10 fixed in space and time. The above discussion has shown that the solution $q(X,T)$ at this point can only be influenced by points at earlier times, particularly the points 1, 2, 3 and 4. Furthermore point $10 = (X,t)$ is not influenced by all points at these earlier times because information only travels at finite wave speeds. Consequently a point in the x-t plane is only influenced by points in a finite domain of dependence. For non-linear shallow water equations the domain of dependence is bounded by the waves with the greatest and

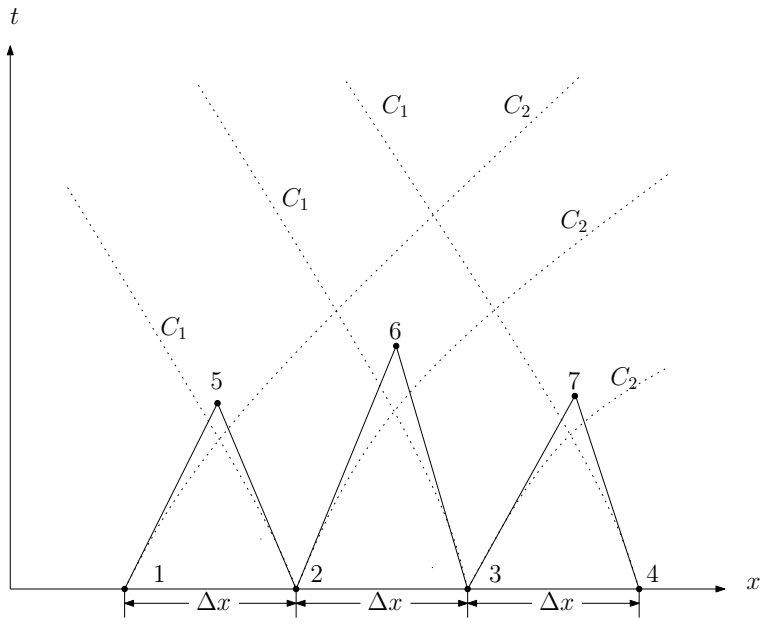


Figure 3.1: Linear approximations of characteristic curves.

least velocities. Specifically the solution $q(X,T)$ depends only on the initial data in the interval $[X-\lambda_2 T, X-\lambda_1 T]$. In a similar manner a point in the x-t plane can only influence points at later times. The range of influence of this point is again bounded by the characteristic curves corresponding to the smallest and largest eigenvalues. The domain of dependence and the range of influence of a point (X,T) are shown in Figure 3.3.

It can be proved mathematically that as $\Delta x \to 0$ the aforementioned process will converge to the exact solution corresponding to the given initial conditions. It must be noted, however, that we have assumed that the two families of characteristics form a regular curvilinear coordinate system over the entire x-t plane. As such we must consider when and where these characteristics cease to have this property and the physical meaning of this phenomenon, which leads neatly to the Rankine-Hugoniot and entropy conditions.

3.1.1 The Rankine-Hugoniot Condition

In deriving the partial differential equation form of the shallow water equations (3.1) the solution was assumed to be smooth. This is often not the case so we must use the integral form of the conservation law. When two or more characteristics of the same family intersect a discontinuity known as a shock will occur. The speed of the shock $s(t)$ can be determined

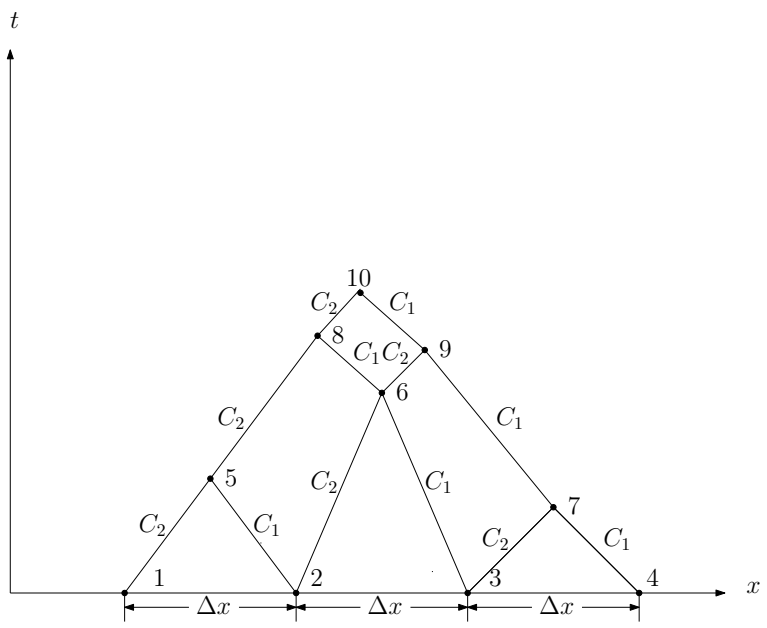


Figure 3.2: Estimating the solution of the shallow water wave equations using finite differences approximations of characteristic curves.

at any time in terms of the states $q_l(t)$ and $q_r(t)$ immediately to the left and right of the shock respectively.

As a solution containing a discontinuity evolves the shock will propagate with some speed $s(t)$. At any time t this shock speed can be determined in terms of the states $q_l(t)$ and $q_r(t)$ immediately to the left and the right of the shock. Consider a small region in space $[x_1, x_1 + \Delta x]$ and a small time period $[t^n, t^n + \Delta t]$. In this domain the shock speed and the states to the left and right of the shock are constant (refer to Figure 3.4). Applying the conservation law in this equation yields

$$\begin{aligned}\int_{x_1}^{x_1+\Delta x} q(x, t_1 + \Delta t)\, dx \;&-\; \int_{x_1}^{x_1+\Delta x} q(x, t_1)\, dx \\ &= \int_{t_1}^{t_1+\Delta t} f(q(x_1, t))\, dt - \int_{t_1}^{t_1+\Delta t} f(q(x_1 + \Delta x, t))\, dt\end{aligned}$$

Because q is essentially constant on each side of the shock in the rectangular domain, the fluxes to the left and right of the shock are also constant and the above equation becomes

$$\Delta x(q_l - q_r) = \Delta t(f(q_l) - f(q_r))$$

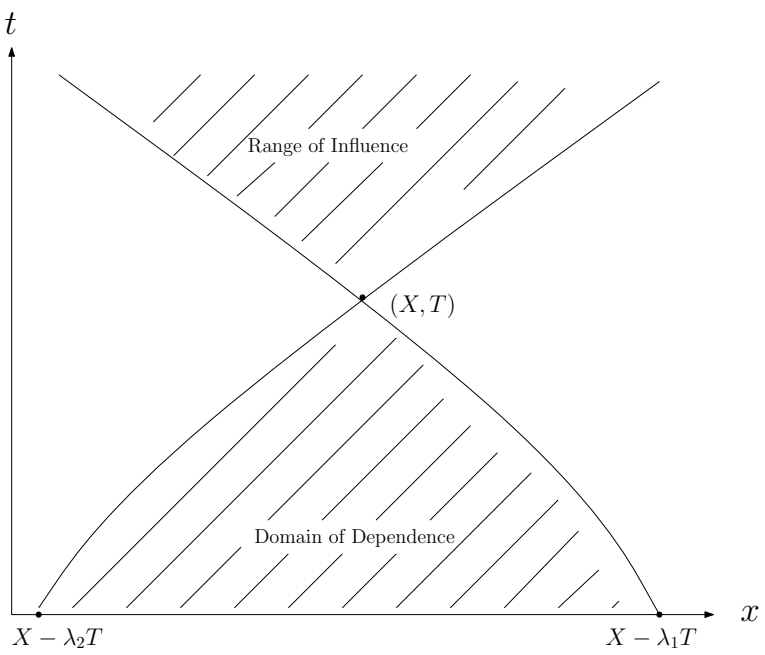


Figure 3.3: The domain of dependence and range of influence of the point (X, T) for a non-linear hyperbolic system of two equations with $\lambda_1 < 0 < \lambda_2$.

Since the shock is propagating at speed s, then $\Delta x = s\Delta t$. Using this definition, dividing by Δt and taking the limit as $\Delta t \to 0$ we can show that any shock must satisfy the condition

$$s(q_r - q_l) = f(q_r) - f(q_l) \tag{3.8}$$

This is known as the Rankine-Hugoniot condition and any solution that satisfies this condition across a shock is a weak solution of the conservation law.

3.1.2 The Lax Entropy Condition

The Rankine-Hugoniot condition does not guarantee unique solutions that are physically correct. In these situations additional constraints, often referred to as admissibility conditions, must be added [34]. The Lax entropy condition can be used to determine whether a weak solution is indeed the physically correct solution. The Lax entropy condition is satisfied if characteristics converge such that, for the 1-shock,

$$\lambda_1(q_l) > s > \lambda_1(q_r)$$

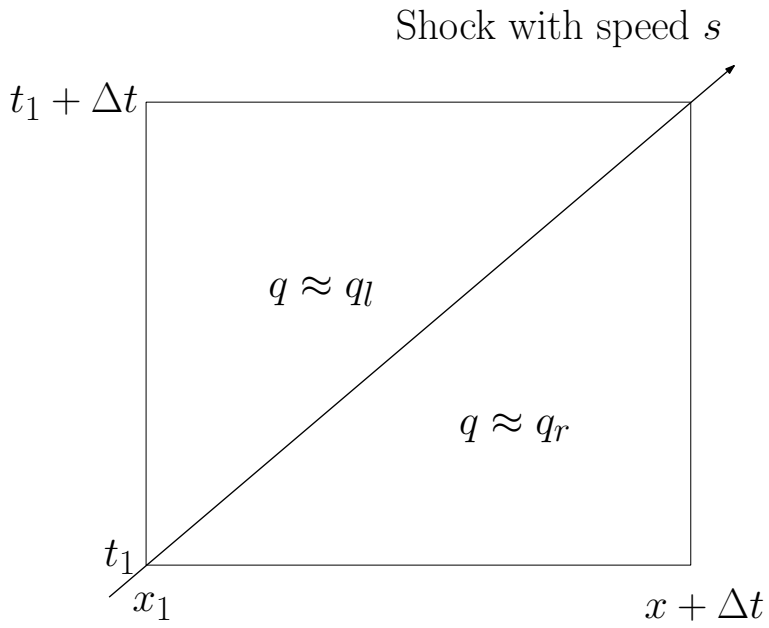


Figure 3.4: The infinitessimal region in the x-t plane used to derive the Rankine-Hugoniot condition.

and, for the two shock,

$$\lambda_2(q_l) > s > \lambda_2(q_r) \tag{3.9}$$

If the Rankine-Hugoniot and Lax Entropy conditions hold, the energy in the fluid will decrease across the shock ensuring is physically correct.

3.2 The Riemann Problem

The Riemann problem describes flow initiated with uniform initial conditions upstream and downstream of a single jump discontinuity. There are two main reasons for discussing the Riemann problem. Firstly, the Riemann problem can be solved analytically and so provides a neat test for numerical schemes. And secondly, many numerical schemes schemes used to simulate shallow water flow, such as those found in [61, 10, 40, 20], incorporate exact or approximate solutions of the Riemann problem to accurately capture wave propagation. Although the central-upwind Godunov scheme presented in Chapter 5 does not adopt this approach, a discussion of this problem is still extremely valuable.

Considered the Riemann problem centred at $x = x_0 = 0$ and $t = t_0 = 0$ with the initial

conditions

$$h(x,t_0) = \begin{cases} q_l & x < x_0 \\ q_r & x > x_0 \end{cases} \tag{3.10}$$

where q_l and q_r are constant vectors. The solution to (3.10) stretches uniformly in space as time increases so that the solution depends only on the single variable x/t rather than on x and t separately. This reduction in the number of independent variables allows analytical solutions to be found more easily. However finding solutions to the general Riemann problem for the shallow water wave equations is still difficult. Consequently the following only discusses the qualitative structure of the solution. Nevertheless Section 3.3.1 presents the analytical solution of an idealised Riemann problem, specifically the Riemann problem with the limiting assumption that fluid on either side of the jump discontinuity is initially at rest.

As the solution evolves, some of the water on the left side of the discontinuity is accelerated instantaneously causing it to flow over the slower moving fluid on the right side of the discontinuity, and cause a shock. For now let us assume that the shock moves with constant speed. Then at some time in the future t_1 the solution can be decomposed into three regions: the undisturbed regions to the left l, and right r, of the disturbance emanating from the point of the initial discontinuity; and an intermediate region m connecting these two quiescent states. The intermediate region can be further decomposed into another two regions. If the shock moves at constant speed the state immediately behind the shock will remain constant for all time. However this state cannot continue upstream indefinitely since the velocity of this region is non-zero and a section of the upstream domain will still be undisturbed and thus have no velocity. This means that there must be some form of depression wave connecting these two constant states.

Figure 3.5(a) shows the two families of characteristic curves of a dam-break producing subcritical flow and how they change within each region. The shock occurs when the 2-characteristic curves of differing slope meet. This is referred to as a 2-shock. A 1-shock is created when 1-characteristic curves converge. The regions containing constant states are indicated by characteristic curves with the same constant slope. In this solution the intermediate constant state and the upstream constant state have been connected with a rarefaction fan. These occur immediately to the left and to the right of the shock and to the left of the rarefaction fan. This region could also consist of a shock propagating with a speed s determined by the Rankine-Hugoniot condition (3.8). However this solution is not physically plausible and violates the entropy condition (3.9).

The non-linear nature of the shallow water equations makes deriving a mathematical expression of the general Riemann problem very difficult and as a result such a discussion is beyond the

scope of this thesis. However solutions to certain idealised Riemann problems are more easily obtained. An analytical solution of one such problem is given in Section 3.3.1.

3.3 Analytical Solutions

It is very difficult to comprehensively analyse numerical models of physical systems. However analytical solutions provide means to at least test a model's ability to simulate certain idealised situations. This section presents a number of analytical solutions that describe both unsteady and steady shallow water flow. The exact solutions selected are intended to provide insight into the behaviour of the shallow water equations and provide a valuable means to establish the veracity of the numerical schemes presented later.

3.3.1 Dam-Break Problem

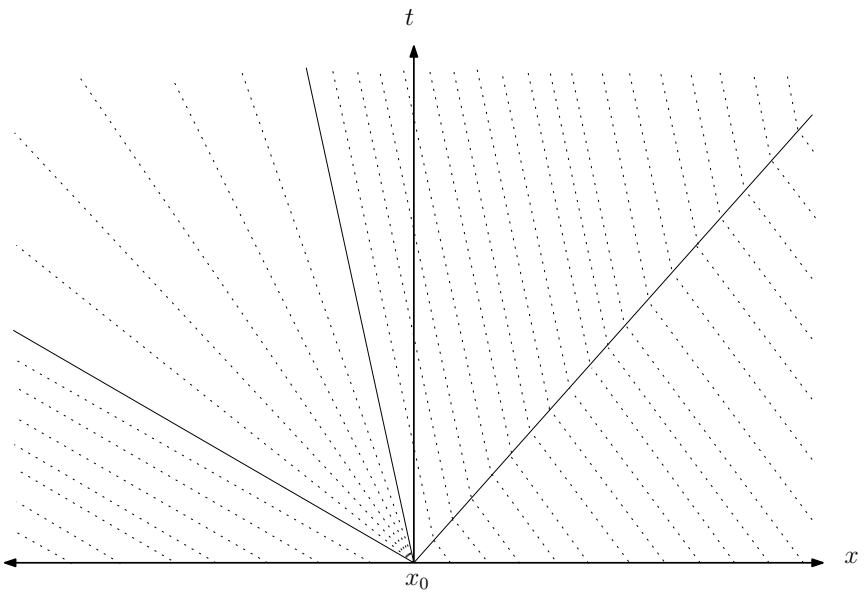
One-dimensional dam-break problems represent an idealized case of the Riemann problem discussed above with the limiting assumption that fluid on either side of the jump discontinuity is initially at rest. Physically they represent the instantaneous collapse of an infinitesimally thin dam wall in a wide infinitely long horizontal channel. These problems admit discontinuous solutions and so provide a tough test of the ability of a numerical model to resolve discontinuities and rarefaction fans. The following presents analytical solution of dam-breaks in channels with finite water depth everywhere and in channels with a dry bed downstream of the dam wall.

Finite Water Depth

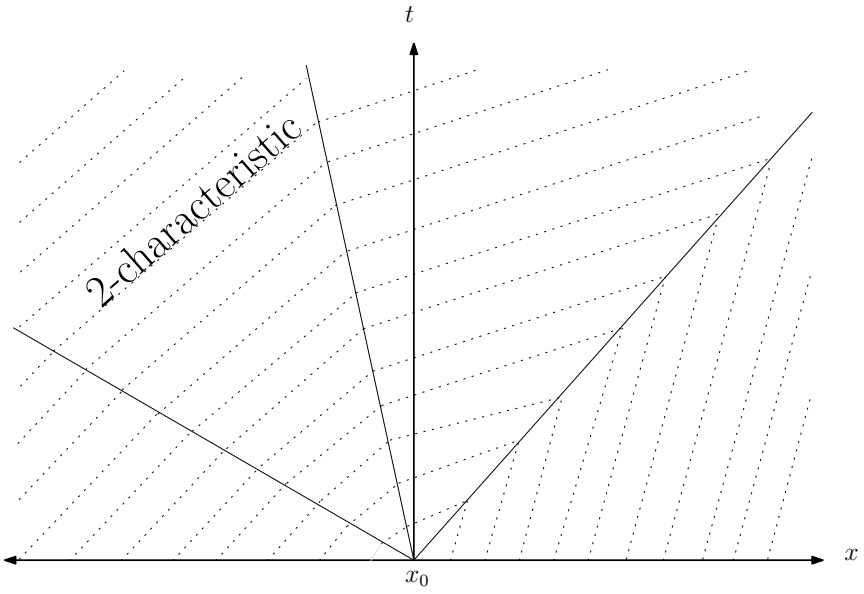
Stoker [50] constructed analytical expressions of the free surface and velocity of shallow water flow resulting from a dam-break . Consider the dam-break with the following initial conditions

$$u(x,0) = 0 \quad \text{and} \quad h(x,0) = \begin{cases} h_0 & x > 1000 \\ h_1 & x \leq 1000 \end{cases} \tag{3.11}$$

At time $t = 0$ the dam wall is immediately removed and at any subsequent time t the spatial domain can be divided into four regions (see Figure 3.6). A shock forms to the right of the initial disturbance as faster moving water, accelerated by removing the dam, hits the stationary water downstream. Behind this shock a constant region (2) forms. This region (2) is connected to the



(a)



(b)

Figure 3.5: Solution of the dam-break Riemann problem (3.10). The solid lines indicate the position of the shock wave and edges of the rarefaction fan and the dotted lines show (a) the 1-characteristics and (b) the 2-characteristics.

undisturbed water upstream (1) by a rarefaction fan with a parabolic free surface. The point at which these two regions meet is determined by the characteristic given by

$$x = -c_1 t$$

The characteristic separating the rarefaction fan (3) and the constant wave region (2) is given by

$$x = (u_2 - c_2)t$$

Finally the position of the shock, travelling at the speed S_2, is given by

$$x = S_2 t$$

Here $c_0 = \sqrt{gh_0}$, $c_1 = \sqrt{gh_1}$ and $c_2 = \sqrt{gh_2}$. Applying the conservation of mass to the column of water surrounding the shock yields

$$h_0(S_2 - u_0) = h_2(S_2 - u_2) \tag{3.12}$$

and since $u_0 = 0$m/s

$$\frac{c_0^2}{c_2^2} = \frac{h_0}{h_2} = \frac{S_2 - u_2}{S_2} \tag{3.13}$$

Similarly applying the conservation of momentum to the column of water surrounding the shock yields

$$\frac{1}{2}(c_2^2 - c_0^2) = h_2(S_2 - u_2)(u_2 - u_0)$$

Note that applying the conservation of mass and momentum across the shock is not a trivial process. Again noting $u_0 = 0$ and substituting u_2 as defined in (3.12) into the above equation yields

$$\frac{1}{2}(c_2^2 - c_0^2) = h_2(S_2 - u_2)S_2\frac{h_2 - h_0}{h_2}$$

and after rearrangement

$$c_2^2 + c_0^2 = 2S_2(S_2 - u_2) \tag{3.14}$$

By eliminating c_2 from (3.14) using (3.13) we obtain the quadratic equation

$$2S_2(S_2 - u_2) - c_0^2(S_2 - u_2) - c_0^2 S_2^2 = 0$$

Solving for u_2 gives

$$u_2 = S_2 - \frac{c_0}{4S_2}\left(1 + \sqrt{1 + \frac{8S_2^2}{c_0^2}}\right) \tag{3.15}$$

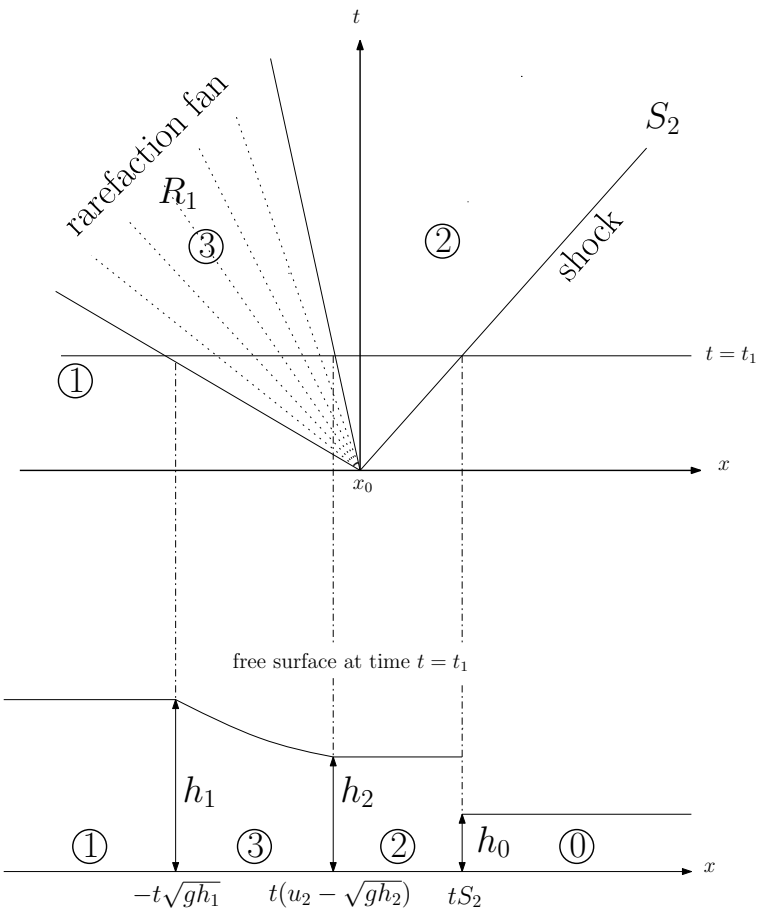


Figure 3.6: Typical solution of the dam-break problem with undisturbed finite water depth downstream at some time $t_1 > 0$.

Here, noting that u_2 and S_2 are always positive, the positive sign of the radical was used to ensure that $u_2 - S_2$ and $-S_2$ have the same sign. Next we eliminate u_2 from (3.15) using (3.13) and obtain

$$\frac{c_2^2}{c_0^2} = \frac{1}{2}\sqrt{1 + \frac{8S_2^2}{c_0^2}} - \frac{1}{2} \tag{3.16}$$

Equations (3.15) and (3.16) provide expressions of the velocity u_2 and the wave speed c_2 in the constant region behind the shock as functions of the shock speed S_2 and the wave speed c_0 in the undisturbed region (0). Consider the rarefaction fan in region (3). Along each of the characteristics in this region (Figure 3.6) the Riemann invariant $u + 2c$ is constant, so on the left extreme of the rarefaction fan

$$u + 2c = 2c_1$$

and on the right extreme

$$u + 2c = u_2 + 2c_2$$

Then combining these two equations yields

$$2c_0 = u_2 + 2c_2$$

and substituting the values of u_2 and c_2 as determined by (3.15) and (3.16) yields a nonlinear expression for the shock speed S_2

$$c_1 = \frac{S_2}{2} - \frac{c_0^2}{8S_2}\left(1 + \sqrt{1 + \frac{8S_2^2}{c_0}}\right) + \left(\frac{c_0^2}{2}\sqrt{1 + \frac{8S_2^2}{c_0}} - \frac{c_0^2}{2}\right)^{1/2}$$

or in a more convenient form

$$S_2 = 2c_1 + \frac{c_0^2}{4S_2}\left(1 + \sqrt{1 + \frac{8S_2^2}{c_0^2}}\right) - \left(2c_0^2\sqrt{1 + \frac{8S_2^2}{c_0^2}} - 2c_0^2\right)^{1/2}$$

Now that the shock speed has been determined we can evaluate u_2 and c_2 and hence determine the nature of the solution in region (3). In this region the characteristic curves are determined by

$$\frac{dx}{dt} = \frac{x}{t} = u_3 - c_3$$

Adopting a similar argument to that previously, we again note that along each of the characteristics in this region (Figure 3.6) the Riemann invariants $u + 2c$ is constant, so on the left extreme of the rarefaction fan

$$u + 2c = 2c_1$$

and along the rarefaction fan

$$u + 2c = u_3 + 2c_3$$

So

$$2c_1 = u_3 + 2c_3$$

Finally solving these equations yields

$$u_3 = \frac{2}{3}\left(c_1 + \frac{x}{t}\right)$$

and

$$c_3 = \frac{2}{3}c_1 - \frac{x}{3t}$$

For given initial conditions, we can now determine the free surface profile and velocity at any time $t > 0$ for the idealised dam break with zero initial velocity and non-zero finite water depths. Specifically the analytical solution of (3.11) is

$$h(x) = \begin{cases} h_1 & x \leq -t\sqrt{gh_1} \\ h_3 = \frac{4}{9g}\left(\sqrt{gh_1} - \frac{x}{2t}\right)^2 & -t\sqrt{gh_1} \leq x \leq t(u_2 - \sqrt{gh_2}) \\ h_2 = \frac{h_0}{2}\sqrt{1 + 8\left(\frac{2h_2}{h_2 - h_0}\frac{\sqrt{h_1} - \sqrt{h_2}}{\sqrt{h_0}}\right)^2} - \frac{h_0}{2} & t(u_2 - \sqrt{gh_2}) < x < tS_2 \\ h_0 & tS_2 \leq x \end{cases} \tag{3.17}$$

$$u(x) = \begin{cases} 0 & x \leq -t\sqrt{gh_1} \\ u_3 = \frac{2}{3}\left(\sqrt{gh_1} + \frac{x}{t}\right) & -t\sqrt{gh_1} \leq x \leq\ t(u_2 - \sqrt{gh_2}) \\ u_2 = S_2 - \frac{gh_0}{4S_2}\left(1 + \sqrt{1 + \frac{8S_2^2}{gh_0}}\right) & t(u_2 - \sqrt{gh_2}) < x < tS_2 \\ 0 & tS_2 \leq x \end{cases} \tag{3.18}$$

where the shock speed is given by

$$S_2 = 2\sqrt{gh_1} + \frac{gh_0}{4S_2}\left(1 + \sqrt{1 + \frac{8S_2^2}{gh_0}}\right) - \left(2gh_0\sqrt{1 + \frac{8S_2^2}{gh_0}} - 2gh_0\right)^{1/2}$$

Unfortunately this solution requires solving two non-linear equations to find the velocity in the constant region behind the shock u_2 and the shock speed S_2. However recently Wu et al. [57] derived a solution to the same problem that only requires solving one non-linear equation

$$h_2 = \frac{h_0}{2}\left[\sqrt{1 + 8\left(\frac{2h_2}{h_2 - h_0}\frac{\sqrt{gh_1} - \sqrt{gh_2}}{\sqrt{gh_0}}\right)^2} - 1\right]^{1/2}$$

The resulting value for h_2 is then used to find

$$S_2 = \frac{2h_2}{h_2 - h_0}(\sqrt{gh_1} - \sqrt{gh_2})$$

and

$$u_2 = 2(\sqrt{gh_1} - \sqrt{gh_2})$$

Values of h_3 and u_3 are found using the corresponding equations found in (3.17) and (3.18)

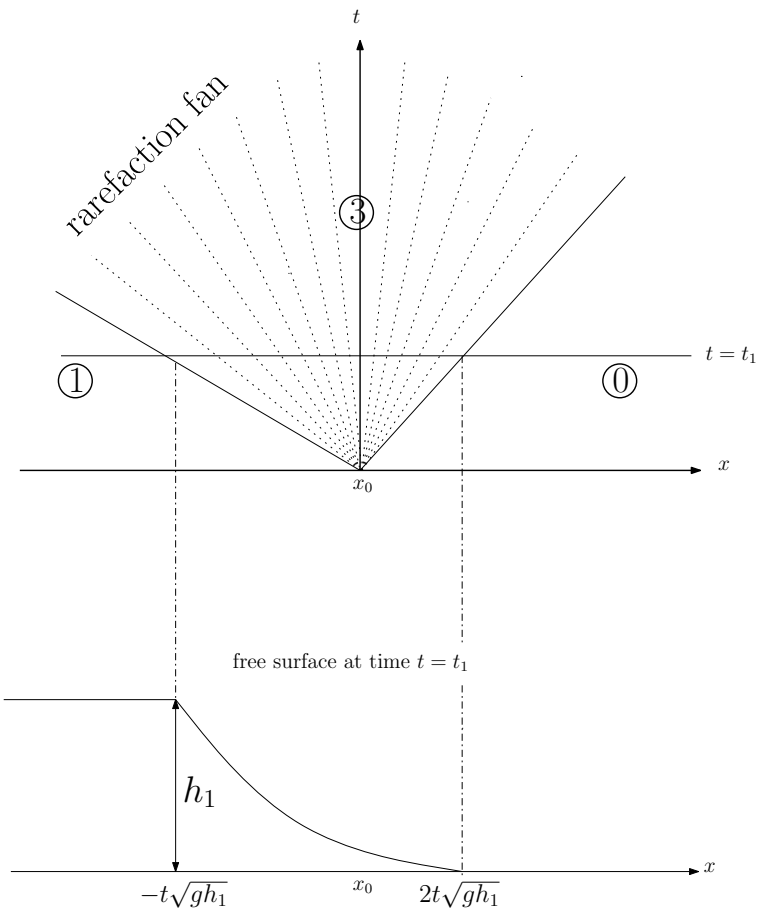


Figure 3.7: Typical solution of the dam-break problem on to a dry bed at some time $t_1 > 0$.

Dry Bed Flow

The ability to simulate flow over dry beds is a very desirable property of any model used to simulate shallow water flows. Consider a dam-break problem with a finite water depth upstream and a dry bed $h_0 = 0$m downstream. In this situation a shock is no longer created and the constant state with depth h_2 and velocity u_2 disappears. Under these circumstances the solution consists of only three regions, the undisturbed upstream and downstream regions and the rarefaction connecting these constant states (see Figure 3.7).

The analytical solution to this problem is presented by Ritter [45] and is given by

$$u(x) = \begin{cases} 0 & x \leq -t\sqrt{gh_1} \\ u_3 = \frac{2}{3}\left(\sqrt{gh_1} + \frac{x}{t}\right) & -t\sqrt{gh_1} < x \leq 2t\sqrt{gh_1} \\ 0 & 2t\sqrt{gh_1} \leq x \end{cases} \tag{3.19}$$

$$h(x) = \begin{cases} h_1 & x \leq -t\sqrt{gh_1} \\ h_3 = \frac{4}{9g}\left(\sqrt{gh_1} - \frac{x}{2t}\right)^2 & -t\sqrt{gh_1} < x \leq 2t\sqrt{gh_1} \\ h_0 & 2t\sqrt{gh_1} \leq x \end{cases} \tag{3.20}$$

3.3.2 Oscillating Planar Free Surface in a Frictionless Parabolic Canal

Flows in parabolic canals and basins admit an interesting class of analytical solutions which can be used to test the accuracy of numerical models of shallow water flow. Thacker [54] developed an analytical description of flow oscillating along the entire length of a uniform frictionless channel with a parabolic bed profile. The free surface of the water in the channel remains planar along the entire length of the channel for all time.

Let us assume that the basin profile is given by

$$z_b = z_\infty \left(\frac{x^2}{L_x^2} + \frac{y^2}{L_y^2}\right)$$

Here L_x and L_y are the lengths of the equilibrium water surfaces in the x and y directions , respectively and z_∞ is the maximum water depth at equilibrium (refer to Figure 3.8). For this problem we assume that $L_y \gg L_x$ so $v = 0$ and $dz/dy = 0$

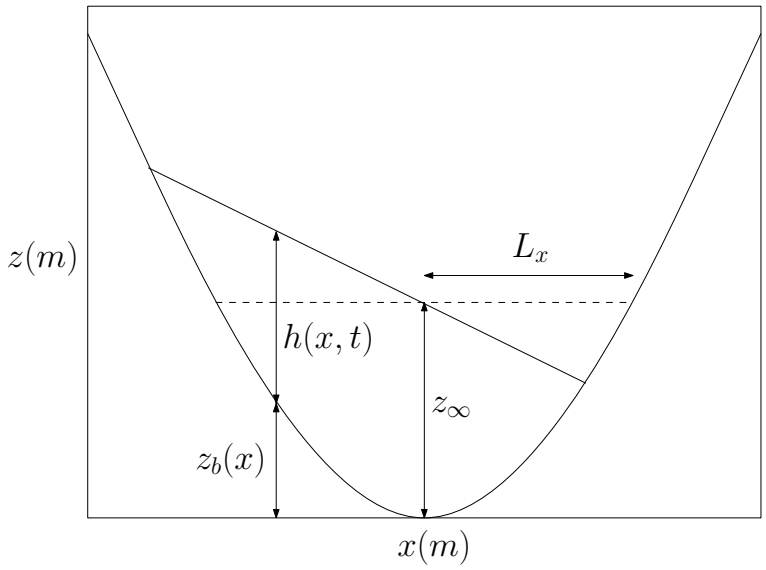


Figure 3.8: A typical representation of a flat, initially stationary, water surface in a parabolic canal.

The water surface profile of an oscillating planar free surface in a parabolic canal is given by

$$z(x,t) = z_\infty + 2A_0\frac{z_\infty}{L_x}\cos\omega t\left(\frac{x}{L_x} - \frac{A_0}{2L_x}\cos\omega t\right) \tag{3.21}$$

and the velocity of the flow is

$$u = -A_0\omega\sin\omega T \tag{3.22}$$

Here A_0 determines the amplitude of the oscillations and the angular frequency of the oscillation is given by

$$\omega = \sqrt{f^2 + \frac{2gz_\infty}{L_x^2}}$$

Note that the solution presented here is based on the inhomogeneous form of the shallow water wave equations with a source term given by $S = -ghb_x$. The dam-break problems above can also be modelled using this equation. However because the dam-breaks are simulated over horizontal beds the source terms becomes zero.

3.3.3 Steady Flow

A robust numerical method must be able to accurately simulate unsteady and steady flows. Consequently it must be able to accurately reproduce steady flow profiles and converge to the corresponding solutions given certain initial conditions. Replicating surface profiles of stationary, undisturbed regions of fluid with varying topography provides a simple yet effective test. However steady flow over a change in the bed of a channel provides a more interesting and more strenuous trial of model performance.

Many of the analytical solutions available describe steady flow in wide, rectangular, frictionless channels with a single obstruction or depression somewhere along their length. The exact nature of these solutions is dependent on the height and velocity of the fluid upstream and downstream of the obstacle and on the size of the structure itself. Here we derive an analytical expression for steady flow in a horizontal frictionless channel of constant width, with bottom elevation $z_b(x)$, that contains a parabolic obstruction somewhere along its length. Flow in this channel is governed by the shallow water wave equations

$$h_t + (uh)_x = 0 \tag{3.23}$$

$$(uh)_t + (u^2h + \frac{1}{2}gh^2)_x = -gh(z_b)_x \tag{3.24}$$

Assuming that the derivatives of u and h are smooth, expanding the derivatives in (3.24) yields

$$uh_t + hu_t + u(uh)_x + uhu_x + gh(h_x + z_b)_x = 0$$

Then using (3.23) to replace the h_t term and dividing by h we obtain

$$\frac{1}{h}uh_t + u_t + uu_x + g(h_x + z_b)_x = 0$$

Under steady flow conditions the water depth h and velocity u throughout the channel does not change with time, so the continuity and momentum equations, (3.23) and (3.24), become

$$(uh)_x = 0$$

$$uu_x + g(h_x + z_b)_x = 0$$

Integrating with respect to x gives

$$uh = q$$

and

$$\frac{1}{2g}u^2 + h + z_b = C \tag{3.25}$$

where q is the volume flux per unit width of the channel and C is a constant. These equations are valid throughout the entire length of the channel. Thus, far from the obstacle, where $h(x) \to h_0$, $z_b(x) = 0$ and $u(x) = u_0$ and we can write

$$uh = u_0h_0 \tag{3.26}$$

and

$$\frac{1}{2g}u^2 + h + z_b = \frac{u_0^2}{2g} + h_0 \tag{3.27}$$

Here h_0 and u_0 refer to the flow conditions at a point upstream unaffected by the obstruction. See Figure 3.9. By eliminating u from (3.27) using (3.26) we obtain

$$\frac{u_0^2h_0^2}{2gh^2} + h + z_b = \frac{u_0^2}{2g} + h_0$$

Recalling $Fr_0 = u_0/\sqrt{gh_0}$ and dividing by h_0 the above equation becomes

$$\frac{Fr_0^2}{2}\frac{h_0^2}{h^2} + \frac{z_b}{h_0} + \frac{h}{h_0} = \frac{Fr_0^2}{2} + 1$$

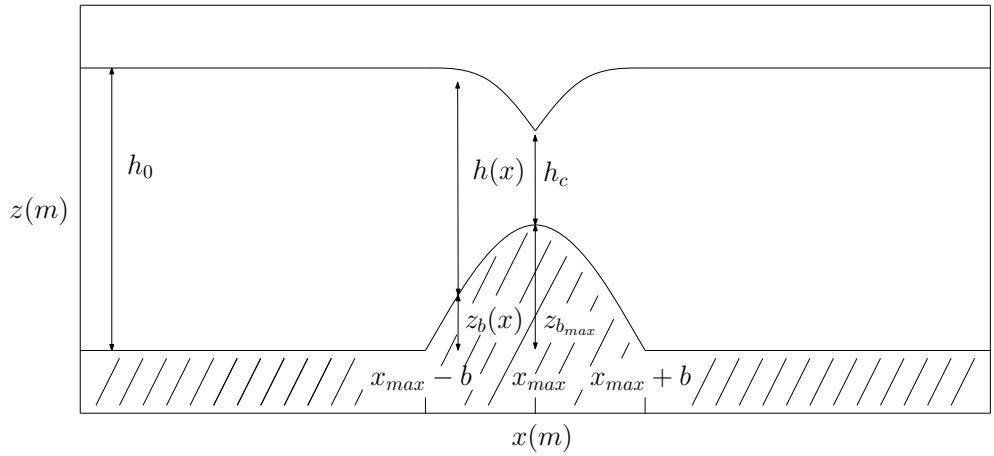


Figure 3.9: The water surface profile for flow over a parabolic obstruction in a frictionless wide rectangular canal conveying a constant flow.

Then rearranging and substituting the dimensionless quantities $D = h/h_0$ and $H = z_b/h_0$ we can write

$$\frac{Fr_0^2}{2D^2} + D = \frac{Fr_0^2}{2} - H + 1$$

We can now find the water surface profile by solving the cubic equation

$$D^3 + D^2\left(H - 1 - \frac{Fr_0^2}{2}\right) + \frac{Fr_0^2}{2} \tag{3.28}$$

3.4 Conclusion

In this chapter we have seen that the non-linear hyperbolic nature of the shallow water wave equations makes finding analytical solutions difficult. The equations admit both continuous and discontinuous solutions even when the initial conditions are smooth. A detailed derivation of the analytical solution of an idealised dam-break was given to indicate the difficulties faced even when describing relatively simple flows. A number of other analytical solutions were also presented as examples of other types of shallow water flows.

Chapter 4

Numerical Methods for solving the Shallow Water Wave Equations: An Overview

As mentioned in Chapter 3, the non-linear hyperbolic nature of the shallow water wave equations makes finding analytic solutions to these equations difficult. The equations admit both continuous and discontinuous solutions even when initial conditions are smooth. Consequently numerical schemes are needed to solve most practical problems. This chapter presents some of the major numerical methods used to solve the shallow water wave equations. The approaches described here can be divided into two categories – finite volume methods and finite element methods. Spectral methods have also been used to solve the shallow water equations, but much less frequently than the other aforementioned approaches. Finite volume methods can be further divided into three categories – naive, upwind and high-resolution schemes. Other finite volume techniques have been developed but are not considered here. Finite element methods used to solve the shallow water equations can be subdivided into two categories – discontinuous Galerkin and streamline diffusion methods.

The treatment of numerical methods found here differs from reviews of traditional texts [31, 34, 33] in several ways. Firstly, unlike most texts this review considers both finite volume and finite element methods and discusses their differences and similarities, advantages and disadvantages. A classification of these schemes is attempted to facilitate the differentiation of the various methods used to simulate shallow water flow. Emphasis was placed on a simple synthesis of appropriate techniques and where possible implementation issues specific to modelling the shallow water wave equations. Furthermore, the literature was often difficult to follow, partly because of the piecemeal accrual of results. So an attempt has been made to present relevant

concepts more simply. The remainder of this chapter considers the Lagrangian formulation of the shallow water equations and associated numerical schemes.

There is no single correct classification of numerical methods used to solve the shallow water wave equations. In fact even in the following, various methods do not lie completely within a particular class. Often these methods combine techniques common to a number of approaches. This particular classification was developed to better describe the numerical techniques used to simulate shallow water flow and to introduce the concepts relevant to the finite volume method(s) constructed later in this thesis.

To enhance the value of the following chapters the remainder of this chapter discusses the numerical techniques used to solve systems of equations belonging to the broader class of hyperbolic conservation laws to which the shallow water equations belong, viz.

$$\frac{\partial}{\partial t}q(x,t) + \frac{\partial}{\partial x}f(q(x,t)) = S \tag{4.1}$$

For the shallow water wave equations

$$q = \begin{bmatrix} h \\ uh \end{bmatrix} f(q) = \begin{bmatrix} uh \\ u^2h + \frac{1}{2}gh^2 \end{bmatrix} S = \begin{bmatrix} 0 \\ -ghb_x \end{bmatrix}$$

For simplicity the following will focus on solving the homogeneous problem ($S = 0$). However methods used to model the inhomogeneous equations are discussed in detail in Chapter 5.

4.1 Finite Volume Method

Finite volume methods are based on the integral form of the conservation law

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)\,dx + f(q(x_2,t)) - f(q(x_1,t)) = 0 \tag{4.2}$$

It is this integral form of the conservation law that ensures that the approximated solution of a finite volume method is conservative. This is particularly important when trying to accurately resolve shock waves. Simply, finite volume methods subdivide the spatial domain into cells, known as the finite volumes, in which approximations to the integrals of the conserved variables are calculated. At each time step the flux through the boundaries of these cells is approximated and the conserved values within these regions are updated accordingly. From equation (4.2) an

explicit conservative time-stepping algorithm can be found by integrating from t^n to t^{n+1} (see Figure 4.1), rearranging and dividing by Δx

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right) \tag{4.3}$$

Here Q_j^n approximates the average value of the quantity q at time $t^n := n\Delta t$ and $F_{j+1/2}^n$ is an approximation of the average flux across the cell interface at $x = x_{j+1/2}$. That is

$$Q_j^n \approx \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} q(x,t^n)\,dx \quad \text{and} \quad F_{j+1/2}^n \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(q(x,t^n))\,dt$$

Rearranging equation (4.3) yields

$$\frac{Q_j^{n+1} - Q_j^n}{\Delta t} + \frac{F_{j+1/2}^n - F_{j-1/2}^n}{\Delta x} = 0$$

This equation can equally be interpreted as a finite difference approximation, or as a finite volume approximation to the conservation law $q_t + f(q)_x = 0$, and illustrates the interrelationship between these two methods.

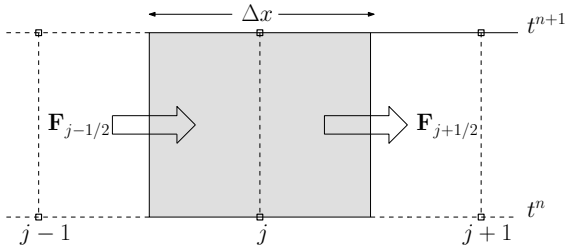


Figure 4.1: Control volume for cell j and the associated intercell fluxes.

The majority of finite volume methods can be classified into three broad categories – naive schemes, upwind schemes and high-resolution schemes. These differ in the way the numerical flux of equation (4.3) is evaluated. Naive schemes do not directly consider the properties of the governing equations, whereas upwind methods do and as a result are slightly more complicated. High-resolution schemes employ different methods for avoiding the artificial oscillations introduced by the higher-order methods used.

4.1.1 Naive Schemes

Finite volume methods can be partitioned into explicit and implicit schemes. Explicit schemes calculate the state of a system at a later time from the state of the system at the current time. Implicit schemes approximate the state of the system at the later time by solving an equation involving both the current state of the system and the later one. For non-linear equations such as the shallow water equations the discretised equations resulting from an implicit scheme are also non-linear. Solving these equations is a computationally costly procedure and so explicit schemes are generally used. The following examines some explicit finite difference schemes.

The classical first-order Lax-Friedrichs scheme can be used to solve the one-dimensional shallow water equations. This method uses the explicit updating scheme in (4.3) where the numerical flux is given by

$$F_{j+1/2} = \frac{1}{2}(F_{j+1/2} - F_{j-1/2}) - \frac{\Delta x}{2\Delta t}(Q_{j+1} - Q_j) \tag{4.4}$$

However, the scheme is highly diffusive and generates approximate solutions that are excessively smoothed. This property is typical of first-order methods and can only be addressed by heavy refinement of the grid used.

The second-order two-step MacCormack scheme has also been used to simulate shallow water flow [62]. This scheme approximates the solution at time t^{n+1} by first calculating an intermediate value for the average cell values (first step)

$$\bar{Q}_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x}(F_{j+1}^n - F_j^n) \tag{4.5}$$

This value is then used to find the final cell values (second step)

$$Q_j^{n+1} = \frac{1}{2}(Q_j^n + \bar{Q}_j^{n+1}) - \frac{\Delta t}{2\Delta x}(\bar{F}_j^{n+1} - \bar{F}_{j-1}^{n+1}) \tag{4.6}$$

where $\bar{F} = F(\bar{Q})$. This scheme does not perform well in the presence of shocks and introduces non-physical oscillations into the solution. This problem is typical of second-order naive difference methods.

4.1.2 Upwind Schemes

Hyperbolic conservation laws (4.1) propagate information along characteristics as waves at different speeds, and possibly in different directions. Upwind methods utilise this knowledge

to obtain approximations of the numerical flux functions, which are more robust than naive approximations. Upwind methods gather information in the direction from which the information is coming [34].

The upwind scheme proposed by Bermudez and Vazquez [5] employs a spatial discretisation based on the propagation properties of the conservation law, which is typical of upwind finite difference methods. The method involves constructing a matrix G that satisfies $f(q) = Gq$. For the shallow water equations

$$G = P\Lambda P^{-1} = \begin{bmatrix} 0 & 1 \\ -u^2 + gh^2/2 & 2u \end{bmatrix} \quad P = \begin{bmatrix} 1 & 1 \\ u + \sqrt{gh/2} & u - \sqrt{gh/2} \end{bmatrix}$$

and

$$\Lambda = \begin{bmatrix} u + \sqrt{gh/2} & 0 \\ 0 & u - \sqrt{gh/2} \end{bmatrix}$$

Using these matrices the flux matrix G can be decomposed into fluxes dependent on the direction of flow so that $F = F^+ - F^-$ and the governing equations become $Q_t + (F^+)_x + (F^-)_x = 0$. The propagation properties of the system are exploited by considering the direction of flow explicitly. Backwards differencing is used to discretise $(F^+)_x$ and forward differencing is used to discretise $(F^-)_x$ so that $F^+ = (F + |F|)/2$ and $F^- = (F - |F|)/2$. Assuming $|F| = |G|q$ and $|G| = P|\Lambda|P$ the inter-cell flux becomes

$$F_{j+1/2} = \frac{1}{2}(F_{j+1/2} + F_j) + \frac{1}{2}(|F_{j+1/2}| - |F_j|)$$

Unfortunately this scheme produces entropy-violating conditions at the transition from subcritical to supercritical flow [62]. There are many other upwind methods which can be used to solve the shallow water equations, but many of these also suffer from entropy-violating conditions. On many occasions an entropy 'fix' can be used to improve the simulated profile near the entropy violating point. However this entropy-violating point may not be removed completely (Zoppou and Roberts [62]).

4.1.3 High-Resolution Methods

Upwind methods generally preserve monotonicity, but in doing so can lose accuracy. In contrast second-order accurate methods such as the two-step MacCormack scheme (4.5,4.6) give much better resolution of smooth regions than upwind methods but introduce oscillations near discontinuities. High-resolution methods attempt to combine the desirable properties of these

two methods, by obtaining higher-order accuracy where possible, but do not enforce this higher accuracy in regions where the numerical solution might behave poorly. Simply, high-resolution methods attempt to increase the amount of numerical dissipation near discontinuities to combat the artificial oscillations introduced by higher-order methods.

To dampen the oscillations created by high-order methods a slope limiter, flux limiter, flux correction or some form of dissipation must be used [62]. Slope limiters impose constraints on the gradients of the dependent variables. Flux limiters impose constraints on the gradients of the flux functions. Flux correction involves updating an upwind flux with an anti-diffusion flux and dissipation methods introduce artificial viscosity to the basic schemes to reduce oscillations in the numerical solution.

Total variation provides a way of measuring oscillations in a solution. For a grid function Q^n total variation is defined as $TV(Q^n) = \sum_{j=-\infty}^{\infty} |Q_j - Q_{j-1}|$. We can attempt to avoid oscillations by requiring that the total variation of Q^n not increase with time. Methods that uphold this condition are known as Total Variation Diminishing (TVD) schemes and preserve monotonicity. A scheme is said to be total variation diminishing, if for any set of data Q^n, the values Q^{n+1} computed by the scheme satisfy $TV(Q^{n+1}) \leq TV(Q^n)$. Unfortunately Godunov's theorem states that monotone preserving linear schemes can at best be first-order accurate [22]). However second-order TVD schemes can resolve smooth regions with the higher order of accuracy whilst reducing to first-order accuracy close to shocks.

Artificial Viscosity

The simplest means to reduce oscillation is to introduce artificial dissipation into a second-order numerical scheme such as the MacCormack scheme (4.5,4.6). For example Davis [16] proposed the following TVD scheme

$$Q_j^{n+1} = Q_j^{Mac} + (D_{j+1/2} - D_{j-1/2})$$

where Q_j^{mac} is the MacCormack solution and $D_{j\pm1/2}$ are the artificial dissipative terms. These non-linear terms depend on the Courant number and are limited to suppress oscillations. When employed to solve the shallow water equations this scheme generally produces excellent resolution of shocks and refraction fans, but oscillations in the solution are not completely removed.

Flux-Corrected Methods

First-order TVD schemes are also used in conjunction with higher-order numerical schemes to create a method that exhibits higher-order accuracy in smooth regions and robust first-order behaviour around discontinuities. This framework was first proposed by Boris and Book [8] and involves updating an upwind flux (see section 4.1.2) with an anti-diffusion flux. This flux is limited so that existing extrema are not accentuated and no new extrema are introduced [62]. Yee [58] proposes a TVD Lax-Wendroff scheme where the flux is given by

$$F_{j+1/2} = F^{UP}_{j+1/2} + \phi_j(F^{SO}_{j+1/2} - F^{UP}_{j+1/2}) \tag{4.7}$$

where $F^{UP}_{j+1/2}$ is a first-order conservative upwind flux, $F^{SO}_{j+1/2}$ is a second-order conservative Lax-Wendroff flux and ϕ is a flux limiter that ensures the solution is TVD. Generally these flux limiters are a function of the parameter

$$\theta = \frac{\bar{q}_j - \bar{q}_{j-1}}{\bar{q}_{j+1} - \bar{q}_j}$$

One such commonly used limiter is the van Leer limiter given by

$$\phi_j(\theta) = \frac{|\theta| + \theta}{1 + \theta}$$

When used to solve the shallow water wave equations such a scheme may not introduce enough artificial diffusion to completely remove all oscillations [62]. These schemes can also produce entropy violating conditions (refer to Section 3.1.2), caused by the upwind methods they employ (see section 4.1.2).

Godunov-Type Methods

There are two types of Godunov schemes: central and upwind. Upwind Godunov-type methods are a generalisation of the upwind finite difference schemes discussed previously in which the local characteristic structure is used by solving a Riemann problem rather than diagonalising the Jacobian matrix. These schemes use exact or approximate Riemann solvers to solve the Riemann problems at each cell interface. Central Godunov-type methods instead use exact evolution of the solution by averaging over the local Riemann fans. A detailed description of central schemes is given in Section 5.

Upwind Godunov-type methods involve solving a sequence of local Riemann problems

$$q(x,0) = \begin{cases} q_l & x < x_{j-1/2} \\ q_r & x > x_{j-1/2} \end{cases} \tag{4.8}$$

at each cell interface. The numerical fluxes used in (4.3) are found by evaluating the true flux function with the solution of the Riemann problem (4.8) at each cell interface so that

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x}\left[f(q_{j+1/2}^n) - f(q_{j-1/2}^n)\right]$$

where $q_{j\pm1/2}^n$ are the solutions to the Riemann problem at $x_{j\pm1/2}$.

At each cell interface the states q_l and q_r are estimated from the reconstruction polynomial $Q_j(x) = P_j(x)$. It is the order of this polynomial that determines the order of the scheme. A piecewise constant reconstruction produces a first-order method where the left and right states are given by $q_l = Q_{j-1}^n$ and $q_r = Q_j^n$, respectively. A linear reconstruction produces a second-order method and so on. Once the polynomial $P_j(x)$ has been constructed the left and right states at each cell interface are limited to avoid oscillations in the solution near discontinuities. An exact or approximate Riemann solver is then used to calculate the inter-cell flux. Finally at the end of a time-step the data is re-averaged and the process is begun again.

Exact Riemann solvers can be used to solve the local Riemann problems, but this method is complex and case specific. Consequently many approximate Riemann solvers have been developed, each with various benefits and disadvantages. The nature of the Riemann problem is investigated in Section 3.2, however a discussion of approximate Riemann solvers is beyond the scope of this thesis, but can be found in [62, 34]. Examples of solving the shallow water wave equations using upwind Godunov-type methods can be found in [23, 10, 49, 40].

Slope-Limiter Methods

The Upwind Godunov methods described in the previous section reconstruct the solution within each cell with a polynomial of degree s. When $s \geq 1$ these higher-order reconstructions introduce artificial oscillations (Figure 4.2(a)) . These local extrema can be dampened using slope limiters (Figure 4.2(b)) so that the reconstruction in cell j is given by

$$q(x) = q(x_j) + \sigma_j(x - x_j)$$

One choice is the minmod slope limiter

$$\sigma_i^n = \text{minmod}\left(\frac{Q_j^n - Q_{j-1}^n}{\Delta x}, \frac{Q_{j+1}^n - Q_j^n}{\Delta x}\right) \tag{4.9}$$

which preserves second-order accuracy whilst still satisfying the TVD property. Here the minmod function of two arguments is given by

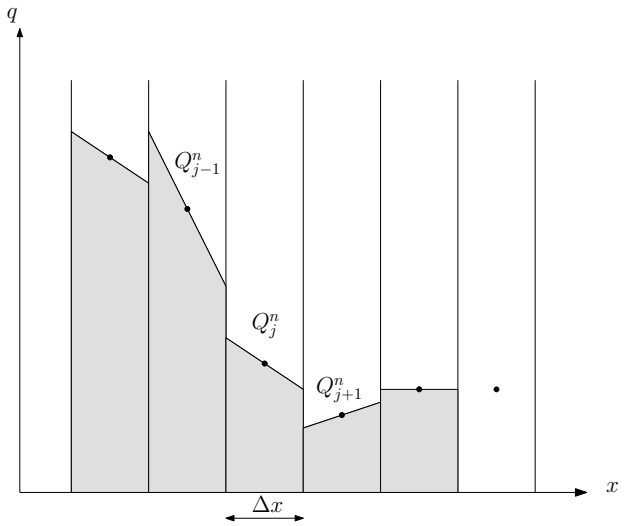
$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases}$$

More sophisticated limiters have been developed, including the Superbee, van Leer, van Albada, and monotonized central-difference slope limiters, but no one limiter performs the best in all situations. Refer to Sections 5.2 and 6.2 for a detailed discussion of slope limiters and their performance.

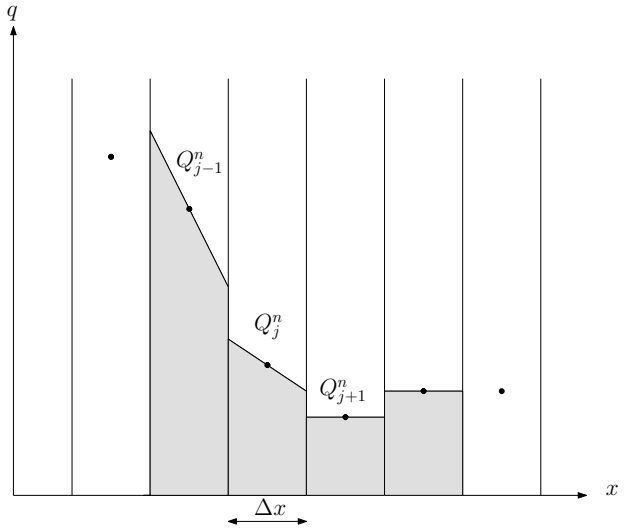
Essentially Non-Oscillatory (ENO) Schemes

Finite difference-type schemes achieve high-order accuracy by using large stencils and hence they may loose accuracy near shocks (pollute) [14]. This problem of pollution may be overcome with the use of essentially non-oscillatory (ENO) schemes and an adaptive grid approach. The large stencils, however, will still cause difficulties when dealing with irregular geometries and/or complicated boundary conditions. Unlike TVD schemes non-oscillatory schemes are not required to dampen the values of each local extrema at each time step. Instead local extremism can be accentuated occasionally and the increase in amplitude recovered at the next time step [24].

ENO schemes were developed to provide a means of automatically dealing with solutions that are not smooth. Consider the piecewise linear upwind Godunov method described above. New local extrema were avoided by limiting the left and right states at the cell interfaces using a slope limiter. ENO methods can be extended to polynomials of arbitrary degree s. The interpolation points used to determine the polynomial reconstruction of the solution within each cell are chosen so as to minimise the amount of oscillation introduced into the solution. In essence ENO methods choose the optimal stencil when performing the reconstruction [47]. Leveque [34] provides the following example: Start with the linear function passing through $q(x_j)$ and $q(x_{j+1})$ to define the first-order polynomial $P_j^{(1)}(x)$ where the superscript indicates the order of the polynomial. This polynomial can be extended to a quadratic polynomial $P_j^{(2)}(x)$ by calculating the two divided differences based on the points $\{q(x_{j+1/2}), q(x_{j-1/2}), q(x_{j-3/2})\}$ and $\{q(x_{j+3/2}), q(x_{j+1/2}), q(x_{j-1/2})\}$ and selecting the one with the smallest magnitude. This process is repeated until a polynomial of degree s is chosen.



(a)



(b)

Figure 4.2: Piecewise linear reconstruction using (a) upwind slopes (b) gradients subject to the minmod slope limiter.

4.2 Finite Element Method

Standard finite element methods do not work well in cases where the exact solution is not smooth. If a discontinuity is present in the exact solution then the finite element solution will generally be characterised by large spurious oscillations even far from the jump itself [27]. This is an obvious concern for hyperbolic equations, in particular the shallow water equations, for which exact solutions are often discontinuous. Two modified finite element methods have been developed to overcome these difficulties. These methods are known as the streamline diffusion method and the discontinuous Galerkin method and are discussed in the following section.

4.2.1 Streamline Diffusion Method

Streamline Diffusion (SD) methods can be viewed as variants of the standard Galerkin finite element method. Simply the test functions of the standard Galerkin method are modified in a manner that results in added stability without sacrificing accuracy [27]. The basic SD method was first applied to non-linear, stationary, hyperbolic conservation laws by Hughes and Tezduyar [26] in 1984. This method provides a large improvement from standard Galerkin methods. In some cases, however, irregularities around shocks may still exist [27]. From this basic formulation Hughes et al. [25] developed a shock-capturing SD method for hyperbolic conservation laws. This method dealt with discontinuities present in the solution by adding a 'shock-capturing' term that introduces 'crosswind' control near these shocks. This method has the qualitative properties of a finite difference scheme that is second-order in smooth regions and first-order close to shocks [27].

The shock capturing streamline diffusion method is used to solve the time-dependent hyperbolic system of conservation laws in $\mathbb{R}^d$

$$u_t + f(u)_x, \quad t > 0 \tag{4.10}$$

For simplicity let us considers Burger's equation

$$\begin{aligned} u_t + uu_x &= 0 \quad , x \in \mathbb{R}(-\infty, \infty),\, t > 0 \\ u(x,0) &= u_0(x) \quad , x \in \mathbb{R} \end{aligned}$$

The method begins by defining a finite element space. To construct this space consider a sequence

of time levels $0 = t_0, t_1, \ldots$ and the 'slabs' $S_n = \mathbb{R}^d \times I_n$. In addition for $h > 0$ let T_h^n be a quasi-uniform triangulation of S_n and define

$$V_n^h = \left\{ v \in H^1(S_n) : v|_k \in P_k(K), K \in T_h^n \right\}$$

where $P_k(K)$ denotes the polynomials of degree at most k. An approximate solution $U = U_h$ is then found in the space $V_h = \Pi_{n \geq 0} V_h^n$. The functions in V_h are continuous in x and possibly discontinuous in t at the discrete time levels t_n.

Using this foundation the basic SD method is formulated as follows: find $U \in V_h$ such that for $n = 1, 2, \ldots$,

$$\int_{S_n} (U_t + UU_x)(v + \delta(v_t + Uv_x))\, dxdt + \int_R (U_+^n - U_-^n) v_+^n dx = 0, \quad \forall v \in V_h^n \tag{4.11}$$

where $\delta = h$, $U_-^0 = u_0$ and

$$v_\pm^n = \lim_{s \to 0^\pm} v(t_n + s)$$

The above formulation of the basic SD method has used the standard Galerkin test functions plus a bias term. The shock-capturing SD method introduces an additional bias term to obtain a better approximation of the solutions to (4.10) near discontinuities. Using the notation $\beta = \beta(U) = (1, U)$ and $\nabla v = (v_t, v_x)$ the modified test function in (4.11) can be written

$$v + \delta(v_t + Uv_x) = v + \delta\beta \cdot \nabla v$$

where $\beta(U) \cdot \nabla v = v_t + Uv_x$. In comparison the modified test functions in the shock-capturing method are given by

$$v + \delta\beta \cdot \nabla v + \bar{\delta}\bar{\beta} \cdot \nabla v$$

Here $\bar{\delta}$ is another parameter with $\bar{\delta} \sim h$ and

$$\bar{\beta} = \frac{\beta \cdot \nabla U}{|\nabla U|^2} \nabla U$$

Finally the shock-capturing method finds $U \in V_h$ such that for $n = 1, 2, \ldots,$

$$\int_{S_n} (U_t + UU_x)(v + \delta\beta(U) \cdot \nabla v + \bar{\delta}\bar{\beta}(U) \cdot \nabla v)\, dx\, dt + \int_R (U_+^n - U_-^n) v_+^n dx = 0, \quad \forall v \in V_h^n$$

The streamline diffusion method is not used as frequently to solve the shallow water equations as its discontinuous Galerkin counterpart in Section 4.2.2. However Bova and Carey [9] detail the use of an SD scheme to simulate shallow water in a channel.

4.2.2 Discontinuous Galerkin Method

Discontinuous Galerkin (DG) methods are a class of finite element methods that employ the use of completely discontinuous basis functions, usually chosen as piecewise polynomials [47]. The discontinuous nature of these basis functions provides flexibility not present in typical finite element methods. This additional flexibility is typified by the following: The degree of the piecewise polynomials in each element can be changed independently of its neighbours (p-adaptivity). Regardless of the order of the scheme DG methods are characterised by a local data structure in which elements only communicate with immediately adjacent elements. DG methods can be applied to unstructured non-conforming grids, with hanging nodes and elements refined and unrefined (h-adaptivity).

The main advantage of the discontinuous Galerkin Method is its local, element-by-element enforcement of the mass and momentum conservation laws [6]. The unstructured nature of the grid enables steep fronts and gradients to be captured. DG methods have been shown to be second-order accurate over linear triangles and are highly parallelisable. A parallel efficiency is usually more than 99% for a fixed mesh [47]. However one of the major disadvantages of DG methods is that there are no continuity restrictions between elements and so a high number of degrees of freedom are required [6].

Similar to finite volume methods a DG scheme approximates an integral form of the conservation law (4.2). The computational domain is divided into finite elements Δ_j. Triangular elements are a common choice for two-dimensional equations. Within each element the numerical solution is approximated by a polynomial of degree r and due to the p-adaptivity of the scheme the two polynomials along the interface of a cell need not satisfy any continuity condition. This is one of the main differences between the DG method and the streamline diffusion method (Section 4.2.1). The p-adaptivity of the scheme means that there are $(r+1)(r+2)/2$ degrees of freedom within each finite element which are used as the coefficients of the local polynomial. The high number of degrees of freedom is a major drawback of the DG method, however computations can be simplified by choosing a local orthogonal basis to expand the polynomial.

The discontinuous Galerkin method is obtained by choosing a test function $v(x, y)$ which is a polynomial of degree r within the cell, multiplying the conservation equation (4.1) by this test function, and integrating by parts over the cell Δ_j. This yields

$$\frac{d}{dt}\int_{\Delta_j} u(x,y,t)v(x,y)\,dxdy - \int_{\Delta_j} F(u)\cdot\nabla v\,dxdy + \int^{\partial\Delta_j} F(u)\cdot nv\,ds = 0 \qquad (4.12)$$

Here n is the unit normal to the element boundary $\partial\Delta_j$.

The second volume integral in the above equation is typically discretised by a sufficiently high-order numerical quadrature as is the line integral. For problems such as the shallow water wave equations a total variation bounded (TVB) method (see Laney [31]) may be needed to limit the solution at interfaces between elements.

Cockburn and Shu [15] have developed a method to solve non-linear time-dependent problems for hyperbolic conservation laws. Their method consists of three parts:

(i) The conservation law is discretised using a DG discretisation in space with polynomials of order r. A discontinuous approximation of the solution u_h is found such that, when it is restricted to the element K it belongs to the finite dimensional space $U(K)$ defined by imposing the equation (4.12) $\forall v_h \in U(K)$. In this step approximate Riemann solvers or other numerical flux methods are used to approximate fluxes across cell interfaces.

(ii) The resulting system of differential equations $\frac{d}{dt}u_h = L(u_h)$ is discretised using an explicit total variation bounded (TVB) Runge-Kutta time-stepping scheme of order $r+1$. The following is an example of such time-stepping methods:

Set $u_h^{(0)} = u_h^n$
For $i = 1, \ldots, k$ compute the intermediate functions

$$u_h^{(i)} = \sum_{l=1}^{i-1} \alpha_{il} w_h^{il}, \quad w_h^{il} = u_h^{(l)} + \frac{\beta_{il}}{\alpha_{il}} \Delta t^n L_h(u_h^{(l)})$$

Set $u_h^{n+1} = u_h^k$

(iii) Finally a generalised slope limiter is used to minimise undershoots and overshoots of the solution from cell to cell. These limiters are used to provide non-oscillatory solutions containing shocks. The slope limiter is a non-linear projection operator devised so that for some function v_h if $u_h^{(l)} = \Lambda\Pi_h v_h$, then the mapping $u_h^{(l)} \mapsto w_h^{il}$ is stable. In this situation the time marching algorithm becomes:

Set $u_h^{(0)} = u_h^n$ where u_h^n is the discretised solution at the nth time step. For $i = 1, \ldots, k$ compute the intermediate functions

$$u_h^{(i)} = \sum_{l=1}^{i-1} \Lambda\Pi_h \left(\alpha_{il} w_h^{il}\right), \quad w_h^{il} = u_h^{(l)} + \frac{\beta_{il}}{\alpha_{il}} \Delta t^n L_h(u_h^{(l)})$$

Set $u_h^{n+1} = u_h^k$

Much work has been undertaken using DG methods to simulate shallow water flow. For examples refer to the work by Blain et al. [6] and Bokhove [7]. This discontinuous Galerkin method can be thought of as a generalisation of the central-upwind Godunov-type method presented in the next chapter.

4.3 Lagrangian Method

Lagrangian descriptions of the shallow water wave equations are employed much less frequently than their equivalent Eulerian descriptions. The Lagrangian approach has been neglected for different reasons associated with its practical implementation. Typically, in the past, only Eulerian measurements of a system being modelled were made, and the resulting Eulerian statistics were not easily translated into Lagrangian coordinates [53]. In addition, a Lagrangian framework is only more beneficial in certain circumstances and, although conceptualisation may be made easier, cumbersome analysis is still required [39]. Finally, the moving grid associated with the Lagrangian coordinates of the fluid particles can become twisted, thereby severely complicating any numerical method [39].

Recently Lagrangian models of shallow water flow, particularly those describing ocean phenomena such as tsunamis and storm surges, have become more practical. This can be largely attributed to the advances in satellite monitoring and the increase in the number of measurement stations that drift across the Earth's oceans, making Lagrangian data more accessible. However the 'distortion' of the Lagrangian grid still remains and is today the most pertinent issue with Lagrangian solutions of the shallow water wave equations.

To overcome this problem Mead (2004) [39] uses the Lagrangian grid to represent the initial position of the particles. This grid is stationary and so is not affected by problems of distortion. Instead of following an evolving grid the positions of particles are calculated by solving a highly non-linear partial differential equation. This PDE is associated with the coordinate transformation that maps the particles from their initial position to their position at time t. This mapping is similar to that used to derive the equations of motion in Section 2.1.2 and is a precise record of particles movements.

There is also a theoretical advantage in adopting a Lagrangian description of the shallow water wave equation. This benefit is found when considering the well-posedness of the equa-

tions in open domains. Ideally open domains (domains with open boundaries) do not introduce boundary layer phenomena and their boundaries govern interior flow as if they were not there at all. When Eulerian domains are adopted they lead, almost inevitably, to ill-posed mixed initial-boundary-value problems [43]. This is because flow at the open boundaries can vary vertically between subcritical and supercritical. This can be overcome by applying boundary conditions for each type of flow, however this may not be practical. In contrast Bennett and Chua [4] have shown that solutions of the shallow water wave equations are guaranteed to be unique in the open domains if the equations are solved on moving particles, for example under a Lagrangian framework.

Wang et al. [55] demonstrate the use of a pure Lagrangian numerical method to model dam-break flows using the one-dimensional shallow water equations. This method is based on the smoothed particle hydrodynamics originally developed for astrophysical fluid dynamics and is grid-free. It therefore avoids the grid distortion methods encountered in grid-based Lagrangian methods such as that proposed by Mead [39].

4.3.1 Semi-Lagrangian Method

Methods that adopt a hybrid of Eulerian and Lagrangian techniques have also been used to model shallow water flow (see [35, 59]). These hybrid methods assume that a fluid particle arrives at a mesh point at the end of each time step. At the beginning of each time step the value of the conserved variables (e.g. water depth) is known and is assumed to remain constant during each time step. The position and value of the particle at the beginning of the time step is obtained by backtracking from a mesh point and interpolating the values at the surrounding mesh points. The method is explicit, but is not restricted by the Courant-Friedrichs-Lewy (CFL) criterion of Eulerian schemes [59]. These problems are also not plagued by grid distortion.

4.4 Conclusion

In this chapter a framework was used to discuss and clarify the properties of some of the frequently used and well-known numerical techniques employed specifically to simulate shallow water flow. Unlike most texts, finite volume and finite element methods were presented concurrently and an investigation of their differences and similarities investigated. Emphasis was placed on a simple synthesis of appropriate techniques and where possible implementation is-

sues specific to modelling the shallow water wave equations. Alternative descriptions of shallow water flow, specifically Boussinesq type equations and the Lagrangian formulation of the shallow water wave equations, are also presented.

This chapter was also intended as an introduction to hydrodynamic modelling, in general, and to demonstrate the variety of algorithms available. Accordingly, a number techniques presented above will not be used in later chapters, but appropriate references have been made to enable readers to investigate further, if they so desire.

Chapter 5

A Central-Upwind Godunov Method for Solving the 1D Shallow Water Wave Equations

The shallow water wave equations (2.29) form a hyperbolic non-linear system. Finding analytical solutions to these equations is often difficult and numerical schemes must be employed to solve many practical problems. There is no one method that best models shallow water flow. The choice of numerical scheme depends on the purpose for which it is intended.

The shallow water equations admit both smooth and discontinuous solutions even in the cases in which the initial conditions are smooth. This property provides the greatest challenge for numerical models. For this reason an important feature of any numerical method is its ability to accurately resolve shocks. In many applications shallow water flow, such as that arising from dam-breaks in urban areas, often encounters obstacles. In addition dam-break flow and tsunami propagation occur over many different topographical regions which are both dry and wet. Thus a numerical method must be robust enough to deal with flow over dry beds and the appearance of dry states within previously wet regions. A numerical method should also be able to accurately describe steady state flows and small perturbations from these steady states over rapidly-varying topography.

This chapter develops a high-resolution Godunov-type method based on the work of Kurganov et al. [30] for solving the one-dimensional shallow water equations. A fractional-step method is used to discretise the problem in time and space. It reduces the non-homogeneous equations into a sequence of augmented problems and a first or second-order Runge-Kutta method is used to

evolve the solution through time. There is no single method that performs better than all others in all situations. Instead one aims to develop a method that performs best in addressing the intended objectives. The major advantages of this model include its ability to model wetting and drying of beds, accurately resolve shocks and reproduce planar surfaces. In addition the underlying principle is novel and simple and, although constructed here for one-dimensional flow, can be easily extended to two-dimensional flow (Section 5.6). Flexible meshes can also be used when problems involve complex and rapidly-varying geometries.

5.1 Kurganov's Central-Upwind Scheme

The following will derive a central-upwind Godunov-type numerical scheme for solving one-dimensional conservation laws such as the shallow water wave equations. This work is based on the work of Kurganov et al [30]. The central-upwind scheme is a finite volume Godunov-type method that does not require exact or approximate Riemann solvers or characteristic analysis, which upwind Godunov-type methods require. However it is upwind in nature in the sense that one-sided information is used to estimate the size of the rarefaction fan. The main advantages of the proposed scheme are the higher resolution, due to less numerical dissipation than previous methods, and its generic nature.

For simplicity let us restrict our analysis to a uniform grid. This grid is broken up into intervals $I_j := \left[x_{j-1/2}, x_{j+1/2}\right]$, of length Δx across which the numerical solution $q_j^n := q(x_j, t^n)$ evolves with small time increments Δt. Here we have adopted the notation: $x_j := j\Delta x$, $x_{j\pm1/2} := (j \pm 1/2)\Delta x$, and $t^n = n\Delta t$.

We start by finding the cell average of $q(., t^n)$ over the interval $I_j := \left[x_{j-1/2}, x_{j+1/2}\right]$ at time level t^n given by

$$Q_j^n(t) = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j-1/2}} q(x, t^n)\, dx$$

Using these average values we can construct a piecewise polynomial reconstruction of the solution in each cell as

$$\tilde{q}(x, t^n) = p_j^n(x), \quad x_{j-1/2} < x < x_{j+1/2} \quad \forall j$$

This piecewise-polynomial initial condition can be used to build a piecewise polynomial reconstruction of the entire solution at time t^n, with possible discontinuities at the cell interfaces $\{x_{j\pm1/2}\}$. Consult Figure 5.1(a). Centered at each discontinuity is a rarefaction fan, which must

be dealt with. Unlike upwind Godunov methods that treat this problem by solving exactly or approximating the corresponding initial Riemann problems, we utilize the left and right speeds of these Riemann fans to further decompose the computational domain into rectangular domains of varying size. These domains break up the numerical solution into continuous and discontinuous sections. This is achieved by considering the one-sided local speeds of the Riemann fan, which are estimated by

$$\begin{aligned} a^+_{j+1/2} &= \max\left\{\lambda_N\left(\frac{\partial f}{\partial q}(q^-_{j+1/2})\right), \lambda_N\left(\frac{\partial f}{\partial q}(q^+_{j+1/2})\right), 0\right\}, \\ a^-_{j+1/2} &= \min\left\{\lambda_1\left(\frac{\partial f}{\partial q}(q^-_{j+1/2})\right), \lambda_1\left(\frac{\partial f}{\partial q}(q^+_{j+1/2})\right), 0\right\} \end{aligned}$$

Here $\lambda_1 < \ldots < \lambda_N$ are the eigenvalues of the Jacobian $\frac{\partial f}{\partial q}$ and

$$q^+_{j+1/2} = p_{j+1}(x_{j+1/2}) \quad \text{and} \quad q^-_{j+1/2} = p_j(x_{j+1/2})$$

These speeds can be used to construct non-equal rectangular domains (see Figure 5.1(a))

$$[x^n_{j-1/2,r}, x^n_{j+1/2,l}] \times [t^n, t^{n+1}] \quad \text{and} \quad [x^n_{j+1/2,l}, x^n_{j+1/2,r}] \times [t^n, t^{n+1}] \tag{5.1}$$

with $x^n_{j+1/2,l} := x_{j+1/2} + \Delta t a^-_{j+1/2}$ and $x^n_{j+1/2,r} := x_{j+1/2} + \Delta t a^+_{j+1/2}$

By utilising the propagation speeds of the Riemann fan we have ensured that the Riemann fan at each discontinuity is contained within the cell $[x^n_{j-1/2,r}, x^n_{j+1/2,l}]$ between t^n and t^{n+1}. Unlike upwind Godunov methods (Section 4.1.3) no Riemann problems are solved. At each time step the Riemann problems are centered at $x_{j\pm1/2}$ and do not affect the values at x_j provided the Courant number is less than one (refer to Section 5.4.3).

The cell averages

$$\begin{aligned} \bar{w}^{n+1}_j &= \frac{1}{x^n_{j+1/2,l} - x^n_{j-1/2,r}} \int_{x^n_{j-1/2,r}}^{x^n_{j+1/2,l}} p^n_j(x)\, dx \\ &- \frac{1}{x^n_{j+1/2,l} - x^n_{j-1/2,r}} \int_{t^n}^{t^{n+1}} \left[f(q(x^n_{j+1/2,l}, t) - f(q(x^n_{j-1/2,r}, t))\right] dt \end{aligned}$$

$$\begin{aligned} \bar{w}^{n+1}_{j+1/2} &= \frac{1}{x^n_{j+1/2,r} - x^n_{j+1/2,l}} \left[\int_{x^n_{j+1/2,l}}^{x^n_{j+1/2}} p^n_j(x)\, dx + \int_{x^n_{j+1/2}}^{x^n_{j+1/2,r}} p^n_{j+1}(x)\, dx\right] \\ &- \frac{1}{x^n_{j+1/2,r} - x^n_{j+1/2,l}} \int_{t^n}^{t^{n+1}} \left[f(q(x^n_{j+1/2,r}, t) - f(q(x^n_{j+1/2,l}, t))\right] dt \end{aligned}$$

are obtained by integrating over the corresponding domains in (5.1) and are used to reconstruct a non-oscillatory piecewise polynomial interpolant

$$\tilde{w}^{n+1}(x) = \sum_j \left(\tilde{w}_j^{n+1}(x)\chi[x^n_{j-1/2,r}, x^n_{j+1/2,l}] + \tilde{w}_{j+1/2}^{n+1}(x)\chi[x^n_{j+1/2,l}, x^n_{j+1/2,r}] \right)$$

Here the χ's are the characteristic functions and $\{\tilde{w}_{j+1/2}^{n+1}(x), \tilde{w}_j^{n+1}(x)\}$ are the polynomial pieces associated with the corresponding intervals of the characteristic functions they multiply. It is the order of these polynomial pieces that determines the accuracy of the method. A piecewise constant reconstruction results in a first-order method, a piecewise linear reconstruction gives rise to a second-order method and so on.

Constructing this piecewise interpolant ensures that the domain can be split up into regions that allow the cell averages at the next time step to be evaluated without the difficulties originally posed by the discontinuities at the cell interfaces. The cell averages at time t^{n+1} are given by

$$Q_j^{n+1} = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \tilde{w}_j^{n+1}(x)\, dx$$

This framework leads to a fully discretised Godunov-type central upwind scheme. We start with the time derivative of $Q_j(t)$ which is expressed as

$$\frac{d}{dt}Q_j(t) = \lim_{\Delta t \to \infty} \frac{Q_j^{n+1} - Q_j^n}{\Delta t} = \lim_{\Delta t \to \infty} \frac{1}{\Delta t} \left[\frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} \tilde{w}^{n+1}(x)\, dx - Q_j^n \right] \tag{5.2}$$

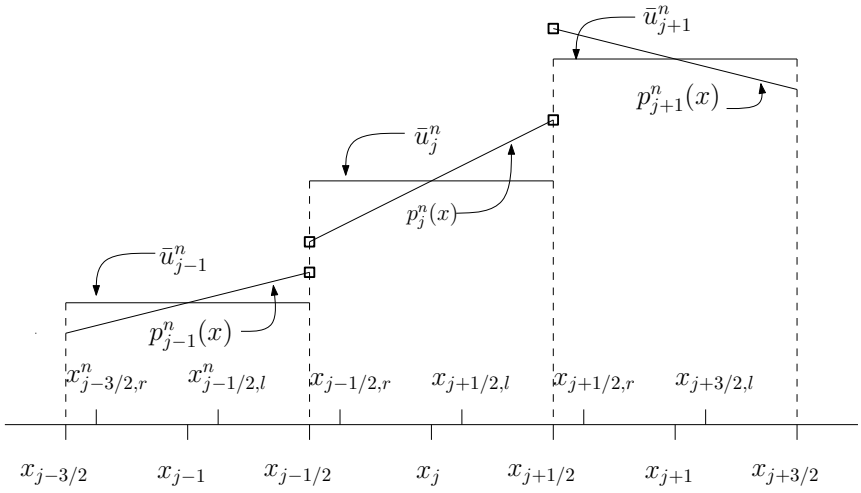
Assuming that the slopes of $\tilde{w}_{j\pm1/2}^{n+1}$ are uniformly bounded, independently of Δt, and because the width of the Riemann fan is bounded by $\left(a^+_{j+1/2} - a^-_{j+1/2}\right)\Delta t$ (Figure 5.1(b))

$$\tilde{w}_{j\pm1/2}^{n+1}(x) = \bar{w}_{j\pm1/2}^{n+1} + O(\Delta t) \quad \forall x \in [x^n_{j\pm1/2,l}, x^n_{j\pm1/2,r}] \tag{5.3}$$

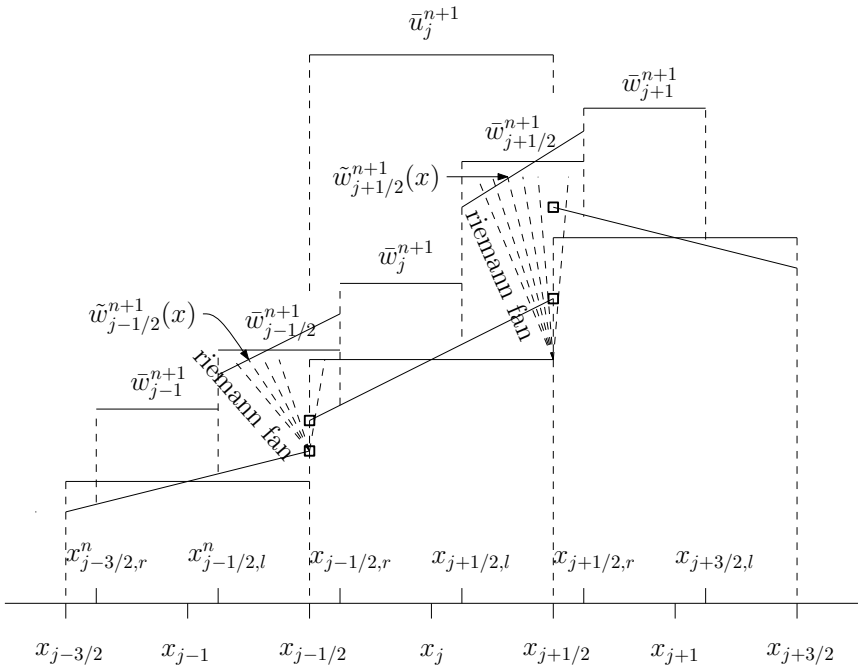
In the domain $[x^n_{j-1/2,r}, x^n_{j+1/2,l}]$ no discontinuity will be present and

$$\frac{1}{x^n_{j+1/2,l} - x^n_{j-1/2,r}} \int_{x^n_{j-1/2,r}}^{x^n_{j+1/2,l}} \tilde{w}_j^{n+1}(x)\, dx = \bar{w}_j^{n+1} \tag{5.4}$$

Using equations (5.3) and (5.4), and substituting into (5.2) with the definitions of $x^n_{j-1/2,r}$ and $x^n_{j+1/2,l}$, yields



(a)



(b)

Figure 5.1: Spatial discretisation and second-order polynomial reconstruction of the central-upwind method.

$$\begin{aligned}\frac{d}{dt}Q_j(t) &= \frac{a^+_{j-1/2}}{\Delta x}\lim_{\Delta t\to\infty}\bar{w}^{n+1}_{j-1/2}+\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\left(\frac{x^n_{j+1/2,l}-x^n_{j-1/2,r}}{\Delta x}\bar{w}^{n+1}_j-Q^n_j\right)\\ &\qquad -\frac{a^-_{j+1/2}}{\Delta x}\lim_{\Delta t\to\infty}\bar{w}^{n+1}_{j+1/2}\\ &= \frac{1}{\Delta x}\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\left[\int_{x_{j-1/2}}^{x_{j-1/2,r}}\tilde{w}^{n+1}_{j-1/2}\,dx+\int_{x_{j-1/2,r}}^{x_{j+1/2,l}}\tilde{w}^{n+1}_j\,dx\right.\\ &\qquad \left.+\int_{x_{j+1/2,l}}^{x_{j+1/2,r}}\tilde{w}^{n+1}_{j+1/2}\,dx-\Delta xQ^n_j\right]\end{aligned}$$

Generating a piecewise constant reconstruction in each region yields

$$\begin{aligned}\frac{d}{dt}Q_j(t) &= \frac{1}{\Delta x}\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\left[[x_{j-1/2,r}-x_{j-1/2}]\bar{w}^{n+1}_{j-1/2}+[x_{j+1/2,l}-x_{j-1/2,r}]\bar{w}^{n+1}_j\right.\\ &\qquad \left.+[x_{j+1/2}-x_{j+1/2,l}]\bar{w}^{n+1}_{j+1/2}-\Delta xQ^n_j\right]\end{aligned}$$

Then using $x^n_{j+1/2,l}:=x_{j+1/2}+\Delta ta^-_{j+1/2}$ and $x^n_{j+1/2,r}:=x_{j+1/2}+\Delta ta^+_{j+1/2}$ we obtain

$$\begin{aligned}\frac{d}{dt}Q^j(t) &= \frac{1}{\Delta x}\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\left[[(x_{j-1/2}+\Delta ta^+_{j-1/2})-x_{j-1/2}]\bar{w}^{n+1}_{j-1/2}-\Delta xQ^n_j\right.\\ &\qquad \left.+(x_{j+1/2,l}-x_{j-1/2,r})\bar{w}^{n+1}_j+[x_{j+1/2}-(x_{j+1/2}+\Delta ta^-_{j+1/2})]\bar{w}^{n+1}_{j+1/2}\right]\\ &= \frac{a^+_{j-1/2}}{\Delta x}\lim_{\Delta t\to\infty}\bar{w}^{n+1}_{j-1/2}+\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\left(\frac{x^n_{j+1/2,l}-x^n_{j-1/2,r}}{\Delta x}\bar{w}^{n+1}_j-Q^n_j\right)\\ &\qquad -\frac{a^-_{j+1/2}}{\Delta x}\lim_{\Delta t\to\infty}\bar{w}^{n+1}_{j+1/2}\end{aligned}\tag{5.5}$$

Treating each term separately we find

$$\begin{aligned}\lim_{\Delta t\to\infty}\frac{1}{\Delta t}\left(\frac{x^n_{j+1/2,l}-x^n_{j-1/2,r}}{\Delta x}\bar{w}^{n+1}_j-Q^n_j\right) &= \frac{a^-_{j+1/2}q^-_{j+1/2}-a^+_{j-1/2}q^+_{j-1/2}}{\Delta x}\\ &\qquad -\frac{f(q^-_{j+1/2})-f(q^+_{j-1/2})}{\Delta x}\end{aligned}$$

and

$$\lim_{\Delta t\to\infty}\bar{w}^{n+1}_{j+1/2}=\frac{a^+_{j+1/2}a^+_{j+1/2}-a^-_{j+1/2}a^-_{j+1/2}}{a^+_{j+1/2}-a^-_{j+1/2}}-\frac{f(q^+_{j+1/2})-f(q^-_{j+1/2})}{a^+_{j+1/2}-a^-_{j+1/2}}$$

where $q^{\pm}_{j+1/2}$ are the left and right values of the piecewise polynomial interpolant constructed at time t^n. Substituting these limits into (5.5) we obtain the conservative semi-discrete central-upwind scheme given by

$$\frac{d}{dt}Q_j(t) = -\frac{F_{j+1/2}(t) - F_{j-1/2}(t)}{\Delta x} \tag{5.6}$$

Here the numerical fluxes $F_{j+1/2}$ are given by

$$F_{j+1/2}(t) = \frac{a^+_{j+1/2} f(q^-_{j+1/2}) - a^-_{j+1/2} f(q^+_{j+1/2})}{a^+_{j+1/2} - a^-_{j+1/2}} + \frac{a^+_{j+1/2} a^-_{j+1/2}}{a^+_{j+1/2} - a^-_{j+1/2}} \left[q^+_{j+1/2} - q^-_{j+1/2} \right] \tag{5.7}$$

5.2 Slope Limiters

When constructing the central-upwind Godunov method (Section 5.1) a piecewise polynomial initial condition was used to build a non-oscillatory piecewise polynomial $\tilde{q}(x, t^n)$ reconstruction of the entire solution at time t^n, with possible discontinuities at the cell interfaces $\{x_{j\pm1/2}\}$. This polynomial was used to determine the left and right states at cell interfaces. If the conserved quantities are assumed to be constant in each cell then a first-order approximation to the inter-cell fluxes is obtained. To achieve second-order accuracy a linear estimate of the left and right states is necessary.

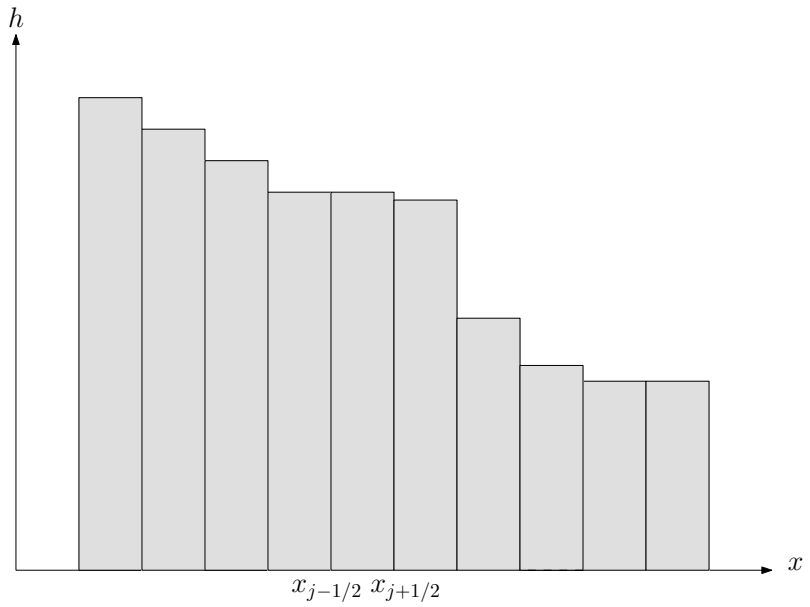
For a given cell j the piecewise polynomial has the form

$$\tilde{q}(x, t^n) = Q^n_j + \sigma^n_j (x - x_j) \quad , x_{j-1/2} < x < x_{j+1/2}$$

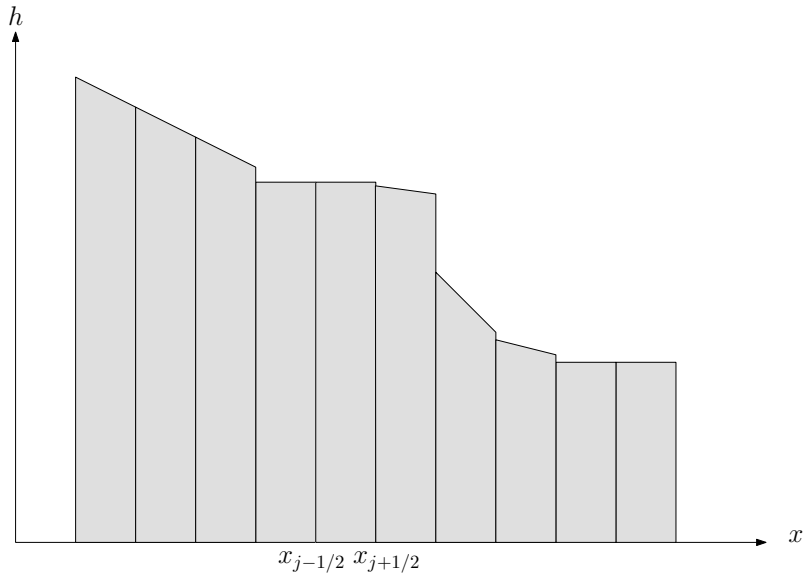
where σ^n_j is the slope of the jth cell. Specifically the left and right states of a cell interface are approximated by

$$q^-_{j+1/2} = Q^n_j + \frac{\Delta x}{2}\sigma^n_j \quad \text{and} \quad q^+_{j+1/2} = Q^n_{j+1} + \frac{\Delta x}{2}\sigma^n_{j+1}$$

Choosing slopes $\sigma^n_j = 0$ results in a first-order method (Figure 5.2(a)). Choosing non-zero slopes results in a second-order method. Numerical oscillations are common with these second and higher-order numerical schemes. These oscillations can be controlled by limiting the slopes of the linear polynomial reconstructions within each cell, thereby dampening any new local extrema introduced.



(a)



(b)

Figure 5.2: Polynomial reconstructions of depth h employed in first and second-order schemes: (a) piecewise constant reconstruction (b) piecewise linear reconstruction with gradients subject to the minmod slope limiter.

Numerous limiter functions have been developed, including the simple minmod limiter

$$\sigma_i^n = \text{minmod}\left(\frac{Q_j^n - Q_{j-1}^n}{\Delta x}, \frac{Q_{j+1}^n - Q_j^n}{\Delta x}\right)$$

Here the minmod of two arguments is defined by

$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases}$$

This limiter reduces the slope of a cell severely near discontinuities. Consult Figure 5.2(b). Less diffuse approximations of discontinuities can be obtained through more sophisticated limiters. Kurganov et al. [30] advocate the limiter function

$$\sigma_j = \text{minmod}\left(\beta\frac{Q_j^n - Q_{j-1}^n}{\Delta x}, \frac{Q_{j+1}^n - Q_{j-1}^n}{2\Delta x}, \beta\frac{Q_{j+1}^n - Q_j^n}{\Delta x}\right) \tag{5.8}$$

This limiter ensures the central-upwind Godunov method is TVD (consult Section 4.1.3) for $1 \leq \beta \leq 2$ [30]. When $\beta = 2$, (5.8) becomes the well known monotonized central-difference limiter (MC limiter). Another possibility is the Superbee limiter proposed by Roe [46]

$$\sigma_j = \text{maxmod}(\sigma^{(1)}, \sigma^{(2)})$$

Here

$$\begin{aligned} \sigma^{(1)} &= \text{minmod}\left(\frac{Q_{j+1}^n - Q_j^n}{\Delta x}, 2\frac{Q_j^n - Q_{j-1}^n}{\Delta x}\right) \\ \sigma^{(2)} &= \text{minmod}\left(2\frac{Q_{j+1}^n - Q_j^n}{\Delta x}, \frac{Q_j^n - Q_{j-1}^n}{\Delta x}\right) \end{aligned}$$

and

$$\text{maxmod}(a,b) = \begin{cases} a & \text{if } |a| > |b| \text{ and } ab > 0 \\ b & \text{if } |b| > |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases}$$

Two limiters originally intended for flux limiting are the van Leer limiter

$$\sigma_j = \frac{|a|\,a + |b|\,b}{|a| + |b|}$$

and the van Albada limiter

$$\sigma_j = \frac{a^2 b + ab^2}{a^2 + b^2}$$

Here a and b are the upwind and downwind slopes given by

$$a = \frac{Q_{j+1}^n - Q_j^n}{\Delta x} \quad \text{and} \quad b = \frac{Q_j^n - Q_{j-1}^n}{\Delta x}$$

The final limiter considered here was obtained from a open sources software package called ANUGA developed jointly by Geoscience Australia and the Australian National University. This Pyvolution limiter adopts a slightly different technique to dampen slopes than the aforementioned limiters. Before the limiter can be applied the gradients in each cell must be calculated using central-differences on the internal cells and upwind or downwind differences at the boundaries. Once the gradients have been applied and estimates of the states at the left and right states, $\hat{q}_l$ and $\hat{q}_r$, have been obtained the limiter is applied to the gradient terms so that the new estimate of the $\hat{q}_l$ is

$$q_l = Q_j^n + \sigma_j \nabla Q_j^n$$

Here the non-linear limiter is given by

$$\sigma_j = \max[\min(\beta r_j, 1), \min(r_j, \beta)]$$

where $0 \leq \sigma_j \leq 1$ and

$$r_j = \begin{cases} (Q_j^{\max} - Q_j)/(\hat{q}_l - Q_j) & \hat{q}_l > Q_j \\ (Q_j^{\min} - Q_j)/(\hat{q}_l - Q_j) & \hat{q}_l < Q_j \\ 1 & \hat{q}_l = Q_j \end{cases}$$

and

$$Q_j^{\min} = \min(Q_j, Q_i) \quad \text{and} \quad Q_j^{\max} = \min(Q_j, Q_i) \quad i = j-1, j+1$$

This slope limiter adds an extra constraint to ensure that if $Q_i > Q_j$, then $q_l < q_r$. Without this extra condition very small values of h produce very large values for $u = uh/h$.

Of course there are many other slope limiters beside those mentioned above. The functions selected were chosen to highlight the varying nature and complexity of slope limiters available. The benefits, disadvantages and performance of these limiters are detailed in Section 6.2.

5.3 Treatment of the Source Term

The shallow water wave equations are used to solve many practical problems that necessitate adding a source term to the homogeneous equations. The derivation provided in Chapter 2 derives a source term based on variable bottom topography. Other influences on this source term

include Coriolis, friction and wind shear. Various assumptions have to be made when these terms are included, but these are not considered here.

5.3.1 Fractional-Step Method

The fractional-step method is a standard way to deal with source terms. It replaces

$$q_t + f(q)_x = S \tag{5.9}$$

with the two problems

$$q_t + f(q)_x = 0 \tag{5.10}$$

$$q_t = S \tag{5.11}$$

which are solved sequentially. The solution to problem (5.9) Q^{n+1} at time t^{n+1} is obtained by first solving the homogeneous equation (5.10) at time t^n to give an intermediate solution $\bar{Q}^{n+1}$. The ordinary differential equation or inhomogeneous problem (5.11) is then solved using $\bar{Q}^{n+1}$ to find the complete solution Q^{n+1}. This splitting can be expressed as

$$Q^{n+1} = O^{(\Delta t)} H^{(\Delta t)} Q^n$$

in which $H^{(t)}$ is the homogeneous solution operator and $O^{(t)}$ is the ordinary differential equation operator.

Fractional step methods are occasionally unable to accurately reproduce certain physically trivial situations such as undisturbed motionless water over topography, time independent steady-flows and small perturbations from such steady states. In these situations $q_t \approx 0$ and so the flux term must approximately balance the source terms $f(q)_x \approx S$. In situations where the flux and source terms are large, resulting say from time-independent flow over steep topography, the solutions obtained by the fractional steps of (5.10) and (5.11) must counteract each other.

Other techniques have been developed that specifically address this problem, however these are not as simple to conceptualise and implement as the fractional step method described here and so are neglected. One such method based on the central-upwind method described above is proposed by Kurganov and Levy [36].

5.4 Temporal Evolution

Thus far we have only discussed the spatial discretisation used to simulate shallow water flow. However the central-upwind Godunov-type method described in Section 5.1 also requires a discretisation of time. The semi-discrete scheme proposed in (Section 5.1) written in the conservative form

$$\frac{d}{dt}Q_j(t) = -\frac{F_{j+1/2}(t) - F_{j-1/2}(t)}{\Delta x} \tag{5.12}$$

is a system of time-dependent ODEs for which many stable numerical solvers are known. The following focuses on the explicit Runge-Kutta methods, introduced by Shu and Osher [48], which lead to step sizes under which the numerical process is TVD. An important feature of this class of time discretisations is that they are convex combinations of first-order forward Euler steps. Hence they maintain strong stability properties in any semi-norm of the forward Euler step [47].

5.4.1 First-order Method

The first-order Runge-Kutta temporal discretisation is given by

$$Q^{n+1} = Q^n + \Delta t F(Q, t^n) \tag{5.13}$$

where $F_j(Q_j, t^n) := F_{j-1/2} - F_{j-1/2}$. This algorithm is simple and computationally more efficient than any higher-order methods. However the first-order accuracy of this approximation is not consistent with the second-order accuracy of the spatial discretisation. For this reason it is important to consider second-order temporal discretisations.

5.4.2 Second-order Method

The second-order Runge-Kutta method considered here is a two-stage method given by

$$Q_j^{(1)} = Q_j^n + \Delta t F_j^n$$

$$\begin{aligned} Q_j^{n+1} &= \frac{1}{2}Q_j^n + \frac{1}{2}Q_j^{(1)} + \frac{\Delta t}{2}F_j^{(1)} \\ &= Q_j^n + \frac{\Delta t}{2}(F_j^n + F_j^{(1)}) \end{aligned} \tag{5.14}$$

which is also known as the improved Euler approximation. Simply, this method produces an initial guess $Q^{(1)}$ to the solution at the next time step and then uses (5.14) to refine this guess.

Simply extending the first-order temporal approximation (5.13) to second-order accuracy is insufficient. The fractional step method presented in Section 5.3.1 is formally only first-order accurate in time and so this must also be addressed. This increase in accuracy can be achieved by adopting the following splitting scheme

$$Q^{n+1} = O^{(\Delta t/2)} H^{(\Delta t)} O^{(\Delta t/2)} Q^n$$

Again the inhomogeneous shallow water equations

$$q_t + f(q)_x = S$$

must be replaced with the two problems

$$\begin{aligned} q_t + f(q)_x &= 0 \\ q_t &= S \end{aligned}$$

At each time step the inhomogeneous equation is solved over half a time step. This result is then used as initial data for the homogeneous problem which is solved over a full time step. Finally this data is used to again solve the inhomogeneous equation over another half time step. The resulting approximation is second-order accurate provided both the homogeneous and inhomogeneous equations are solved to at least second-order accuracy. This approach was developed by Gilbert Strang [51] and is often referred to as Strang splitting.

5.4.3 The Courant-Friedrichs-Lewy (CFL) Condition

The CFL condition is a necessary condition of stability for finite volume or finite difference methods. The approximations provided by such methods will only converge to the exact solution as the grid is refined if the CFL condition is satisfied. Simply the CFL condition states that information, propagating at the speeds determined by the eigenvalues of the flux Jacobian matrix $f'(q)$, does not skip any cell in a single time step.

For the one-dimensional shallow water equations information propagates at two speeds $\lambda_1 = u - \sqrt{gh}$ and $\lambda_2 = u + \sqrt{gh}$. These values are used to define the Courant number, also known as the CFL number, given by

$$\nu = \frac{1}{2} \frac{\Delta t}{\Delta x} \max_p |\lambda_p|$$

Explicit methods used to solve hyperbolic equations, such as the Godunov method defined above, will only be stable if the Courant number is less than one, that is $\nu < 1$. For the first and second-order time-stepping methods defined above

$$\Delta t = \min_{j=1,\ldots,N} C_r \frac{1}{2} \frac{\Delta x_j}{\max_p |\lambda_{j,p}|} \tag{5.15}$$

where $\max_p |\lambda_{j,p}|$ is the maximum absolute wave speed in cell j with length Δx_j. C_r is a constant which, for the test problems presented here, is set to one unless otherwise specified. The factor of a half ensures that the local rarefaction fans at each cell interface do not interact. Consult Figure 5.3. It must be stressed that the CFL condition is a necessary but not sufficient condition of stability and numerical methods may still exhibit instability.

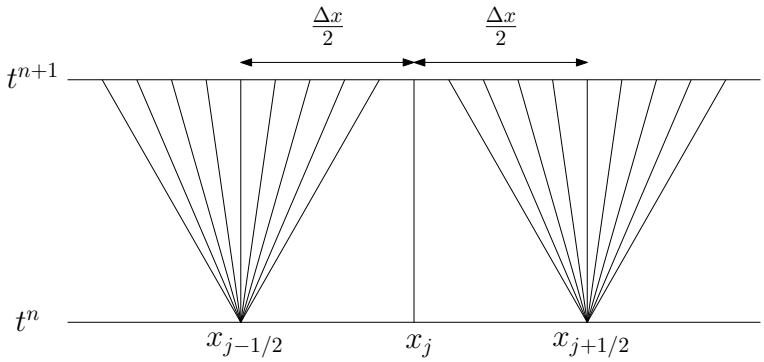


Figure 5.3: Local rarefaction fans at the boundaries of a cell j. To ensure stability information must not travel a distance further than $\Delta x/2$.

5.5 Boundary Conditions

Thus far we have assumed that the information needed to update cell averages Q_j^n is available in neighbouring cells. However all numerical approximations are computed on a bounded domain with a finite set of cells. Cells that lie along the boundary will not have enough information to be accurately updated. There are two main approaches used to update these cells. The first approach involves extending the computational domain to include additional cells connected to those that originally lie along the boundary. The value of these so-called ghost cells are set at the beginning of each time step and are based on the boundary conditions and occasionally the solution in the interior domain. These cells provide the necessary information for the cells originally adjacent to the boundary.

In one dimension only the cells at the front and end of the domain will require the additional information carried by these ghost cells. If information is only required from immediately adjacent cells then only one ghost cell need be added to each end of the computation domain. If a method is used that uses information from two cells upstream and/or downstream then two ghost cells must be added to each end of the physical domain.

The second approach constructs formulas that are case-specific to deal with the special conditions that occur near the boundaries. These formulas depend on the type of method and boundary condition employed. Ghost cells provide a novel approach to dealing with boundary conditions that is conceptually simple, however due to the limited scope of this thesis, the more easily implemented formulaic approach was adopted. Boundary conditions are very easy to implement when using this approach and the central-upwind Godunov method described above. A cell on the boundary will have a state $q = [h, uh]^T$, which is constant over a single time step. Only information from cells immediately adjacent to these cells is needed so only the unknown state on the other side of the boundary need be specified.

For a reflective boundary this unknown state is given by

$$Q_b = \begin{bmatrix} h \\ -uh \end{bmatrix}$$

and for a transmissive boundary

$$Q_b = Q \tag{5.16}$$

Along a boundary, flow may vary between supercritical and subcritical. For this reason Oliger and Sundstrom [43] state that the boundary value problem of the shallow water wave equations is not well-posed in open domains. Such open domains do not occur naturally but are used in small-scale simulations and are bounded by conditions, such as (5.16), which determine interior flow as if they were not there at all. To ensure that a solution is unique boundary conditions must be applied separately for subcritical and supercritical flow.

5.6 Modelling Two-Dimensional Shallow Water Flow

The section follows the work of Zoppou and Roberts [60, 61].

The general conservative form of the two-dimensional (2D) shallow water wave equation is given by

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = S \tag{5.17}$$

where q is the vector of conserved variables, f and g are the flux vectors in the x and y directions and S represents the source vector. The vectors q, f and g can be expressed in terms of the primary variables, u, v and h as

$$q = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix}, f = \begin{bmatrix} uh \\ u^2h + gh^2/2 \\ uvh \end{bmatrix} \text{ and } g = \begin{bmatrix} vh \\ uvh \\ v^2h + gh^2/2 \end{bmatrix}$$

Here u is the x component of the depth-averaged fluid velocity, v is the y component of the depth-averaged fluid velocity, h is the water depth, and g is the acceleration due to gravity.

The conservative integral form of the 2D shallow water equation is obtained by integrating over an arbitrary element V_i (Figure 5.4) and is given by

$$\frac{\partial}{\partial t}\int_{V_i} q\, dV + \oint_{\partial V_i} e \cdot \vec{n}\, dS = \int_{V_i} S\, dV \tag{5.18}$$

where $e = (f, g)$, $\vec{n}$ is the unit outward vector normal to the boundary ∂V_i, the integrand $e \cdot \vec{n}$ is the normal flux across a surface with normal $\vec{n}$ and dV and dS are the small area and arc elements, respectively.

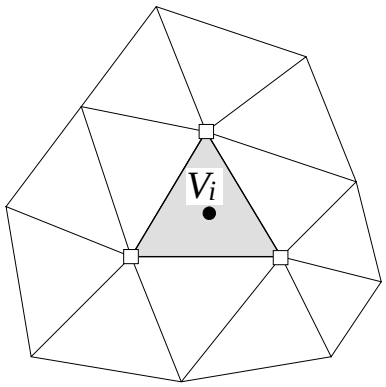


Figure 5.4: Triangular elements in the 2D finite volume method.

Now we can make use of the rotational invariance property of the shallow water equations by defining the rotation matrix

$$\mathbf{T}_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_1 & n_2 \\ 0 & -n_2 & n_1 \end{bmatrix}$$

that aligns the normal $\vec{n} = (n_1, n_2)^T$ with the x-axis. Using this matrix we can write

$$e(q) \cdot \vec{n} = \mathbf{T}_n^{-1} f(\mathbf{T}_n q) \tag{5.19}$$

Substituting (5.19) into (5.18) yields

$$\frac{\partial}{\partial t} \int_{V_i} q \, dV + \oint_{\partial V_i} \mathbf{T}_n^{-1} f(\mathbf{T}_n q) \, dS = \int_{V_i} S \, dV \tag{5.20}$$

Here we have assumed that the conserved quantities $\mathbf{U}$ are constant within each cell and the flux across each edge j of the element i is determined by the states of the cells partitioned by the edge j. Discretising (5.20) gives the basic framework of a finite volume method for solving the 2D shallow water equations

$$\frac{dQ_i}{dt} + \frac{1}{A_i} \sum_{j \in N(i)} \mathbf{T}_{n_{i,j}}^{-1} \tilde{F}(\mathbf{T}_{n_{i,j}} Q_i, \mathbf{T}_{n_{i,j}} Q_j) L_{i,j} = S_i \tag{5.21}$$

Here $N(i)$ is the set of all elements that share a common edge with element i. If triangular cells are used the number of elements in $N(i)$ will be three, and the index j corresponding to the sides of cell i will be in the range $[1, 3]$. In addition A_i is the area of cell i, $L_{i,j}$ is the arc length of side j of element i and S_i is the source term associated with the cell i. Finally $\tilde{F}(\mathbf{T}_{n_{i,j}} Q_i, \mathbf{T}_{n_{i,j}} Q_j)$ is the estimated numerical flux across the edge separating two neighbouring elements. Equation (5.21) is then used to update the solution.

The problem now involves the solution of a local one-dimensional problem in the direction normal to the element interface for which the one-dimensional central-upwind Godunov method, described above, can be used.

5.7 Model Implementation

This section concerns the implementation of the aforementioned central-difference scheme. The software developed is written using the object-oriented approach of `Python`. The code was developed from scratch, to facilitate greater understanding and evaluation of the numerical technique in question. Use of pre-built software may not have identified some of the numerical difficulties discussed below. The code constructed is too large to be included in its entirety, however a brief discussion of the method used to initialise and run a scenario is included in Appendix A.

The `Python` language was chosen because it allows code to be written quickly and easily changed. Unfortunately this language is not as computationally efficient as lower level languages such as `C`. However `Python` provides means for inserting `C` routines into python programs allowing more computationally expensive functions to be written in `C` to decrease execution time. However this was beyond the scope of this thesis.

To implement a scenario the user specifies the geometry, the initial free surface elevation (stage), momentum, boundary conditions (reflective or transmissive) and source terms. A grid is then created which divides the study area into N cells with $2N$ vertices. At each vertex i of each cell j the coordinates x_{ji} are specified to allow grids with cells of equal or varying length. The bed elevation z_{ji}, free surface elevation w_{ji} and momentum xm_{ji} associated with these vertices are also stored. This information is then used to define the value of these three quantities at the cell centroids. The centroid bed elevation is given by $zc_{ji} := (z_{j1} + z_{j2})/2$ and the centroid stage by $wc_{ji} := (w_{j1} + w_{j2})/2$. Furthermore integer mapping arrays are created to define the neighbouring cells and vertices of each cell in addition to the outward unit normal vectors of each cell vertex. These maps reference cells from left to right.

When the model is run, the grid is converted into domain object which represents the study area, quantities, boundaries and source terms with the domain object also containing methods for time-stepping, flux calculations, and all other numerical operations pertinent to the model. This information is specified by a shallow water domain that inherits its main structure from a more generic domain. This allows the code to be more easily extended to other systems of conservation laws such as Euler's equations.

Once the initial conditions have been specified the simulation is run using the routine evolve:

```
for t in domain.evolve(yieldstep, duration):
    <update solution>
```

The parameter duration specifies the time period over which evolve operates, and control is passed to the body of the for-loop at each fixed yieldstep. This decouples the internal time-stepping from the overall time-stepping so that outputs may be interrogated. The length of each time step is determined from the maximal speeds encountered and the sizes of each cell so that the Courant-Friedrichs-Lewy (CFL) condition is not violated. This ensures that no information will skip any cell in one time step.

At each time step the conserved quantities are update by the numerical scheme. Firstly, the slopes of each conserved quantities for each cell are calculated using a specified slope limiter and

used to extrapolate the centroid value to the cell vertices. The flux of each conserved quantity is then computed using the corresponding values, q_l and q_r, found at the two vertices of neighbouring cells sharing the same coordinates. These fluxes are rotated according to the normals of each cell vertex to ensure the correct orientation. Water depth h is always used to compute the amount of water flowing across a cell interface. So if one is using stage as the conserved variable, the vertex heights needed for the flux calculations are found using $h_{ji} = w_{ji} - z_{ji}$. If the depth on either side of the cell face is less than a tolerance $\delta_h = 1e^{-3}$ the velocity, $u = xm/h = uh/h$, reconstructed from the momentum, is set to zero on that side before fluxes are computed. Similarly if depths on both sides of the cell interface are less than this tolerance, the fluxes are set to zero.

Once fluxes have been computed, source terms are evaluated. The values of each conserved quantity are updated at each cell centroid using either the first or second-order time-stepping scheme discussed in Section 5.4. Boundary conditions are then updated. The process is repeated until the final solution is computed. It must be noted that, in order to limit the scope of the thesis, varying topography was assumed to be the only source term affecting shallow water flow. Problems involving terms arising from phenomena such as friction and wind shear were not considered. This meant that source terms can be updated explicitly with the fluxes.

Chapter 6

Results I: Validation

A suite of procedures need to be considered when validating numerical models of 'real' systems. It should address the plausibility of the model properties and the satisfactory reproduction of observed behaviour including the reproduction of the variability and sensitivity of the system [2]. The previous chapters have addressed the choice and plausibility of the central-upwind Godunov type method (Chapter 5) for modelling shallow water flow, but to this point have not considered model performance.

Unfortunately numerical models of physical systems can seldom be fully analysed due to the shear amount of computational power needed to undertake comprehensive sensitivity and error assessment. Numerous approaches can be adopted but are well beyond the scope of this thesis. Instead here we consider the use of analytical solutions to help establish the veracity of our numerical model.

This chapter compares the numerical solution provided by the central-upwind Godunov-type method against a number of problems with both smooth and discontinuous solutions. Specifically, performance will be assessed on its ability to simulate three types of problems: (i) dam-breaks (ii) oscillating planner flow in a frictionless parabolic channel, and (iii) steady flow in wide frictionless channels. The analytical solutions to these problems are discussed in detail in Section 3.3.

Together these problem represent a wide range of shallow water flows and test the model's ability to deal with the sonic points, the transition from subcritical to supercritical flow, the wetting and drying of cells, resolution of shocks and rarefaction fans, source terms arising from varying topography, and the simulation of planar surfaces.

A robust model must produce solutions that converge to the exact solution as the computa-

tional grid is refined. The following investigates the error E between the simulated and exact water depth and momentum profiles for both steady flows and unsteady discontinuous flows, i.e dam-breaks. Throughout the remainder of this thesis the L^1 absolute error given by

$$E = \frac{1}{N}\sum_{j=1}^{N} |q(x_j) - Q_j| \tag{6.1}$$

is used to quantify error. Absolute error rather than relative error is used so that the errors occurring at small values of depth and momentum do not dominate the solution.

All the numerical experiments below are produced on a evenly-spaced grid using second-order spatial and temporal discretisations unless otherwise specified.

6.1 First and Second-Order Temporal and Spatial Discretisation

This section concerns the practical implications of first and second-order spatial discretisations. Specifically the following investigates the performance of the first-order central-upwind scheme with first-order time-stepping, the second-order central-upwind scheme with first-order time-stepping and the second-order central-upwind scheme with second-order time-stepping. Such a discussion is absent from most texts, but as will be seen is extremely important.

6.1.1 Accuracy

Model performance is evaluated on the ability to simulate unsteady subcritical flow resulting from a dam-break. Consider the dam-break problem in a channel of length $L = 2000$m subdivided into 100 cells of equal length with an upstream water depth of $h_1 = 10$m and a downstream water depth of $h_0 = 5$m. The exact depth and momentum profiles of the resulting flow are given by (3.17) and (3.18). The first-order piecewise constant numerical solution obtained at $t = 30$s using first-order forward Euler time-stepping is shown in Figure 6.1. This scheme is highly diffusive and highlights the need for a higher order of accuracy.

In contrast the second-order numerical solution obtained at $t = 30$s using first-order forward Euler time-stepping and the simple minmod slope limiter (Figure 6.2) is much less diffusive and the simulation of the shock is greatly enhanced. In addition the absolute error in the water

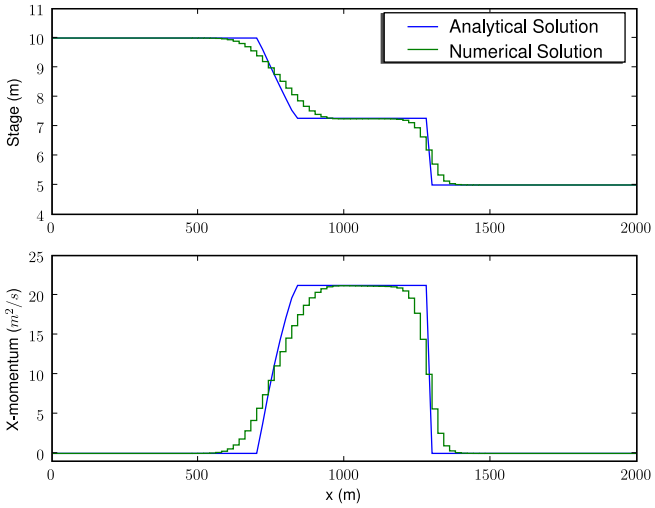


Figure 6.1: Numerical solution of the dam-break problem using 100 cells where $h_1 = 10$m and $h_0 = 5$m obtained at $t = 30$s with first-order spatial discretisation and first-order time-stepping.

profile has also decreased from 0.08950199 to 0.03412096 and the absolute error in the momentum from 0.72740844 to 0.28541721. However the second-order spatial reconstruction has introduced some large artificial oscillations. Refining the computational grid does not dampen these oscillations sufficiently. In fact as cell size is decreased the severity of these oscillations increases. This is highlighted in Figure 6.3(a) which compares the numerical solution against the analytical solution when the channel is divided into $25,600$ cells. The severity of the oscillations becomes so great that, at small enough grid sizes, a further reduction in cell lengths will increase the absolute error in the solution. Furthermore the position of the rarefaction fan is not accurately captured at these finer grid sizes, as shown in Figure 6.3(a). It was also found that the different choices of slope limiters did not produce more accurate results. In fact all the limiters detailed in Section 5.2 produced worse results. This is highlighted in Figure 6.3(b).

The reader can be assured that great care was taken to ensure that these two sources of error are indeed artifacts of the numerical method presented here and not due to problematic software. The important discovery made was that the second-order spatial method presented here requires a time discretisation of the corresponding order.

Figures 6.4(a) and 6.4(b) compare the analytical solution of the dam-break at $t = 30$s using both second-order spatial and time discretisation. The oscillations present in Figures 6.2, 6.3(a),

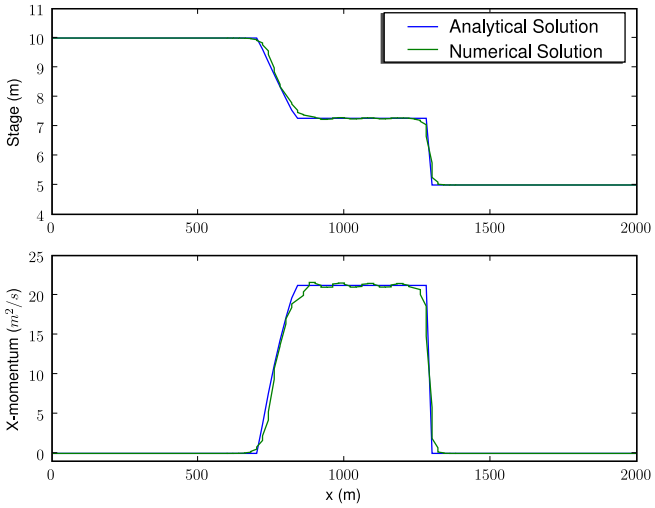
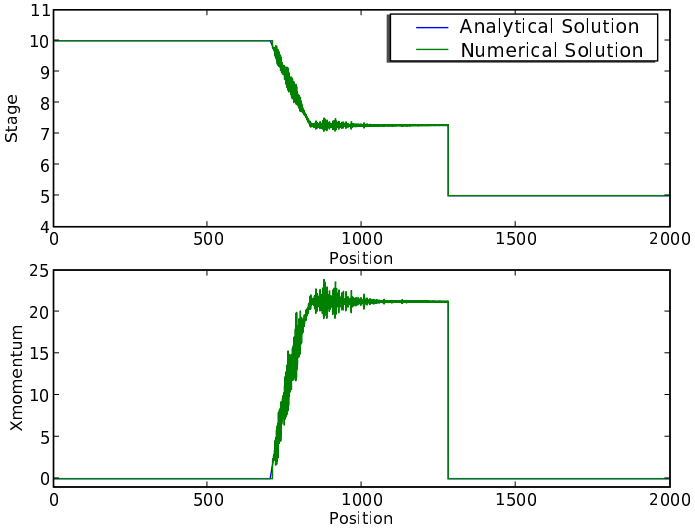


Figure 6.2: Numerical solution of the dam-break problem using 100 cells where $h_1 = 10$m and $h_0 = 5$m obtained at $t = 30$s with second-order spatial discretisation and first-order time-stepping and minmod slope limiting.

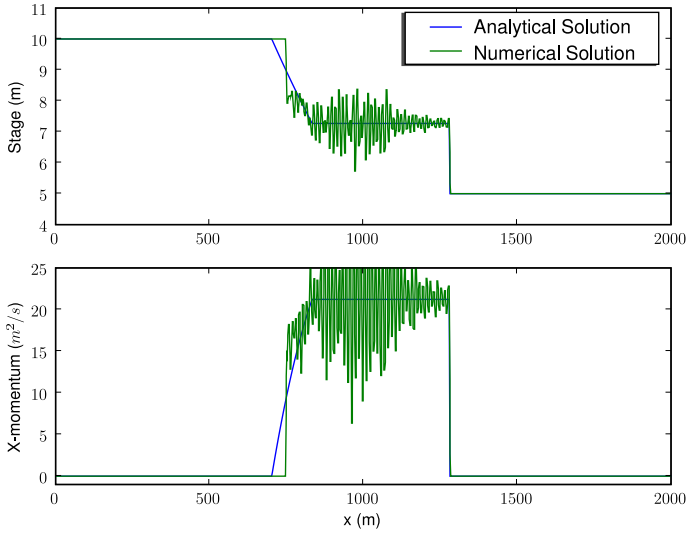
and 6.3(b) have been eliminated. The scheme has become slightly more diffusive. However less diffusive schemes can be constructed by using more sophisticated slope limiters. When the domain is divided into 100 cells the absolute error in the water profile is 0.05531952 and the absolute error in the momentum is 0.44613966. This is slightly larger than the error obtained using first-order time-stepping, but the error obtained using smaller cell sizes is much smaller using second-order temporal evolution than when first-order time-stepping is used. For example when the computational domain is divided into 1,600 cells the absolute error in the water depth is 0.0057452598 and 0.0039501822 for first and second-order time-stepping respectively and the absolute error in the momentum is 0.0503602339 and 0.0344664612.

6.1.2 Efficiency

Table 6.1 contains the wall time required to solve the subcritical solution above, at $t = 30$s. The average of ten runs for each grid cell is given to ensure these times are an accurate representation of the true running time. The first-order spatial and temporal method is much faster than the second-order methods. The second-order central upwind-scheme with first-order time-stepping is approximately twice as slow and the completely second-order method is another



(a)



(b)

Figure 6.3: Numerical solution of the dam-break problem where $h_1 = 10$m and $h_0 = 5$m obtained at $t = 30$s with first-order time-stepping and using (a) 25,600 cells and Minmod slope limiting (b) 800 cells and Superbee limiting.

Number of Cells	First-Order	First/Second-Order	Second-Order
100	0.63 s	1.39 s	4.75 s
200	2.45 s	5.38 s	18.61 s
400	9.74 s	21.09 s	73.65 s
800	39.59 s	84.37 s	294.94 s
1600	155.31 s	335.37 s	1176.22 s

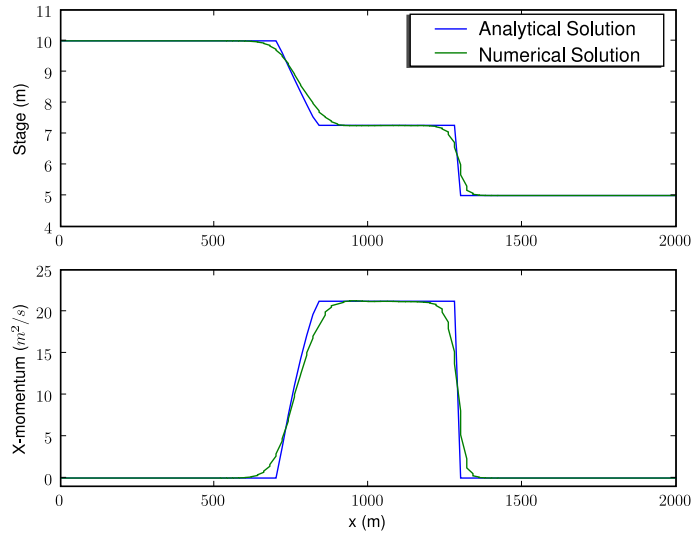
Table 6.1: Average wall time (10 runs per grid size) taken to simulate subcritical dam-break flow at $t = 30$s.

three times as slow. We can also see from Table 6.1 that the running time of all three models increases by a factor of four every time the cell size is halved. This suggests that the algorithm central-upwind scheme is an $O(n^2)$ algorithm.

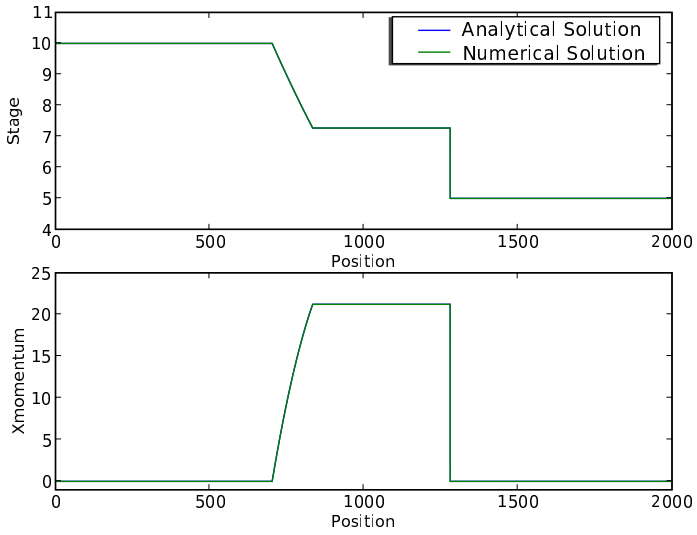
Unfortunately the increased accuracy of second-order time-stepping is accompanied by a reduction in computational efficiency. This is an unavoidable limitation, as the first-order central-upwind scheme is too diffusive and the second-order scheme with first-order time-stepping is unstable. However it may be possible to reduce the running time of the purely second-order method to only twice that of the method using first-order time-stepping. Currently at the beginning of each time step the second-order temporal method computes the upcoming time step based on the fluxes of the newly updated conserved quantities, and then undertakes two further flux computations before updating the numerical solution. In contrast the first-order temporal evolution requires only one set of flux computations and so is approximately three times as fast. The number of flux computations involved with second-order time-stepping can be reduced by approximating the time step size by the time step size at the previous time step. However situations may arise for which this estimate is very poor. Rigorous testing needs to be carried out to investigate the validity of this step and was not attempted here.

6.2 Dam-Break Problems

One-dimensional dam-break problems represent the idealized instantaneous collapse of an infinitessimally-thin dam wall in a wide, infinitely-long horizontal channel. These problems admit discontinuous solutions and so provide a tough test of the ability of a numerical model to resolve discontinuities and rarefaction fans and, in certain cases, flow over dry beds. These prob-



(a)



(b)

Figure 6.4: Numerical solution of the dam-break problem where $h_1 = 10$m and $h_0 = 5$m obtained at $t = 30$s with second-order time-stepping and minmod slope limiting using (a) 100 cells (b) 25,600 cells.

lems are also the most appropriate tests of the robustness of the cell reconstructions of different slope limiters. To accurately simulate a dam-break the slope limiter must reconstruct linear and quadratic polynomials, discontinuities and the interface between wet and dry regions. The following compares the numerical solutions of the central-upwind Godunov, utilising various slope limiters, against the corresponding exact solution for dam-breaks in channels with finite water depth everywhere and in channels with a dry bed downstream of the dam wall.

6.2.1 Subcritical Flow

Consider the dam-break problem in a channel of length $L = 2,000$m and conveying only subcritical flow, subdivided into 400 cells of equal length with an upstream water depth of $h_1 = 10$m and a downstream water depth of $h_0 = 5$m. Figure 6.5 shows the depth and momentum profiles of the flow at $t = 30$s using the Superbee limiter. Here the analytical and numerical solutions are nearly indistinguishable. The position of both the shock and rarefaction fan has been captured accurately and only a small amount of diffusion is present. Very similar results are produced by all the limiters defined in Section 5.2.

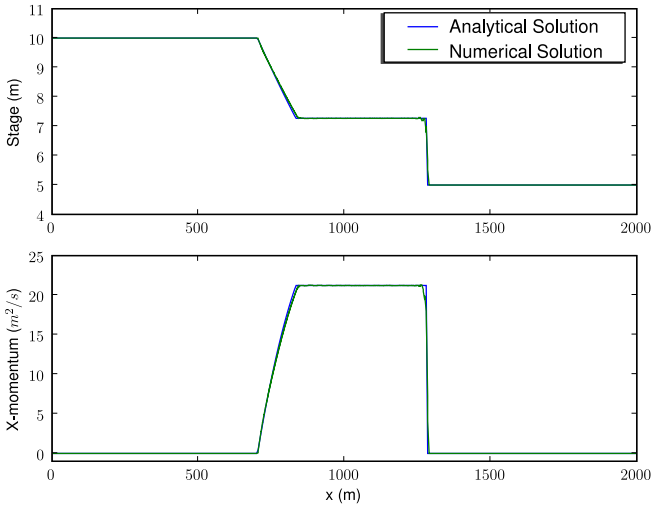


Figure 6.5: Numerical solution of the dam-break problem using 400 cells where $h_1 = 10$m and $h_0 = 5$m obtained at $t = 30$s using second-order central-upwind Godunov-type method with second-order time-stepping and Superbee slope limiting.

Table 6.2 contains the absolute error in the depth E_h and momentum E_{uh} for each limiter.

We see that the minmod limiter provides the least accurate reconstruction of both the depth and momentum and the Superbee limiter provides the best approximation. However, with the exception of the van Albada limiter, the performance of the other limiters is very similar.

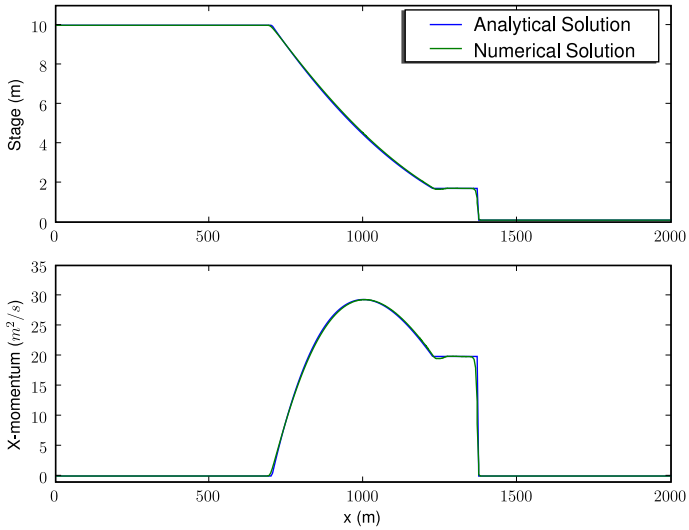
Slope Limiter	E_h	E_{uh}
van Leer	0.0109987392467	0.0895727806613
van Albada	0.0123154434756	0.100902704015
Minmod	0.0148438095397	0.122137908823
Kurganov minmod	0.0105356558074	0.0852279046185
Superbee	0.00945161500114	0.0767785816693
Pyvolution	0.0105356558074	0.0852279046185

Table 6.2: Absolute error in numerical approximation of subcritical dam-break flow using the second-order central-upwind method, second-order time-stepping and six different slope limiters.

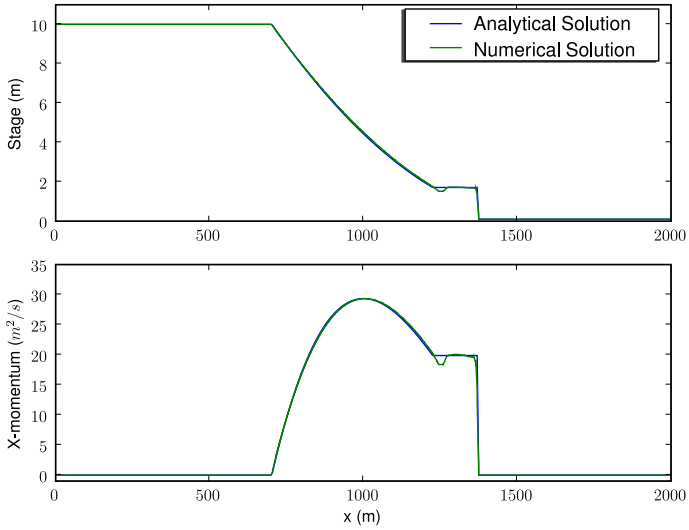
6.2.2 Supercritical Flow

The problem posed above only generates subcritical flow and so is not a rigorous test of a numerical scheme. The dam-break problem involving supercritical flow tests the model's ability to deal with the transition from subcritical to supercritical flow. Some methods produce entropy-violating solution at these so-called sonic points because the scheme is unable to establish the direction of the flow at a point with zero horizontal momentum.

By simply changing the downstream depth of the previous problem to $h_0 = 0.1$m we can produce an analytical solution describing supercritical flow. Table 6.3 contains the absolute error in the depth and momentum for each limiter. We see that the minmod limiter provides the least accurate reconstruction of both the depth and momentum, but the Superbee limiter does not reconstruct the solution to the same degree of accuracy as when used to model subcritical dam-break flow. Refer to Figure 6.6(b). The undershoot at the bottom of the rarefaction fan is typical of the limiter producing the other poor results. The van Leer limiter now provides the best approximation. See Figure 6.6(a).



(a)



(b)

Figure 6.6: Numerical solution of the dam-break problem using 400 cells where $h_1 = 10$m and $h_0 = 0.1$m obtained at $t = 30$s using second-order central-upwind Godunov-type method with second-order time-stepping and the (a) van Leer limiter (b) Superbee limiter.

Slope Limiter	E_h	E_{uh}
van Leer	0.0180311350992	0.0933302021086
van Albada	0.0186836509548	0.101657530186
Minmod	0.0206630247901	0.119619309589
Kurganov Minmod	0.0186711830305	0.0953878733192
Superbee	0.0204303287425	0.107340156459
Pyvolution	0.0186711830305	0.0953878733192

Table 6.3: Absolute error in numerical approximation of supercritical dam-break flow using the second-order central-upwind method, second-order time-stepping and six different slope limiters.

6.2.3 Dry Bed Flow

The ability to simulate flow over dry beds is a very desirable property of any model used to simulate shallow water flows. Consider a dam-break problem with a finite water depth upstream and a dry bed $h_0 = 0$m downstream. The analytical solution to this problem is given by (3.19) and (3.20).

Table 6.4 contains the absolute error in the depth and momentum for each limiter. The Kurganov minmod, Superbee, and Pyvolution produce very poor reconstructions. These limiters are entirely unable to deal with the interface between wet and dry cells. Refer to Figures 6.7(a), 6.7(b) and 6.8(a).

Slope Limiter	E_h	E_{uh}
van Leer	0.0175769943708	0.0973820780907
Van Albada	0.0183464833293	0.107163330128
Minmod	0.0213394124517	0.138865621084
Kurganov Minmod	0.0667542741838	0.648555806348
Superbee	0.160237677428	1.50142285659
Pyvolution	0.109509661578	1.08420278974

Table 6.4: Absolute error in numerical approximation of dry dam-break flow using the second-order central-upwind method, second-order time-stepping and six different slope limiters.

Again the van Leer limiter produces the most accurate results. However even this limiter, and

the minmod and van Albada limiters, are unable to deal with the wet-dry interface perfectly. Both the minmod and van Albada limiters underestimate the speed of the advancing wet front over the dry bed. The van Leer limiter also underestimates this speed but not to the same extent. This limiter does however introduce an artificial amount of water downstream of the advancing front. This latter phenomena was eliminated by reducing the grid size of the computational domain. The inaccuracy occurring at the interface between the wet and dry cells indicates that further work needs to be done to produce better approximations of water depth and velocity in these regions.

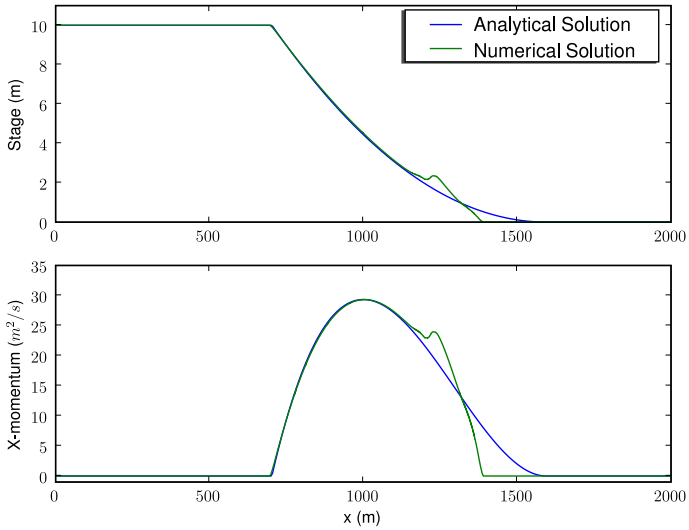
The above discussion has shown that the van Leer limiter produces the best approximations of both the water depth and momentum profiles resulting from a number of dam-break situations. It is able to accurately capture the position of shocks and rarefaction fans, smooth regions and to a lesser, but still reasonable, degree advancing wet fronts. The van Albada and minmod limiters also perform well in all situations.

The analysis also highlights the weakness of the Kurganov minmod, Superbee and Pyvolution limiters when modelling flow over dry beds. Although these slope limiters accurately resolve flow with finite-water depth everywhere, their dramatic failure reproducing dry bed flows prevent them from being used in any robust numerical scheme.

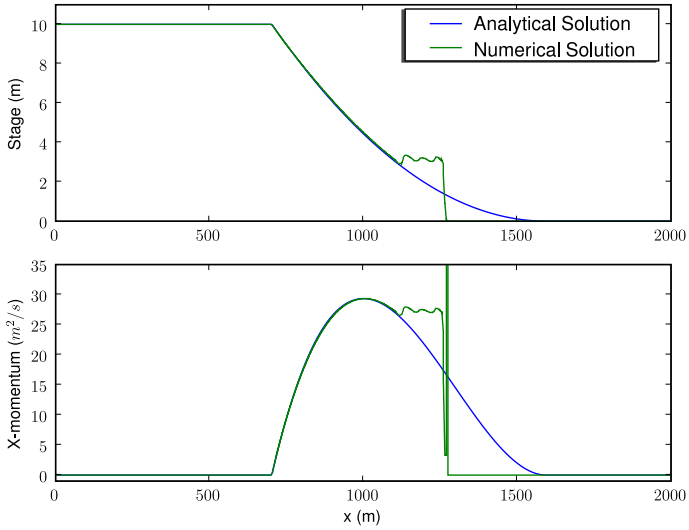
6.2.4 Convergence

A robust model must produce solutions that converge to the exact solution as the computational grid is refined. The errors between the simulated water depth E_h and momentum E_{uh} were calculated for the three hypothetical dam-break problems above at $t = 30$s for successively smaller cell sizes. Recall that absolute error rather than relative error is used so that the error occurring at small values of depth and momentum do not dominate the solution. This is especially important for the dry dam-break problem which produces the worst approximation of flow at the interface between wet and dry cells. At this point the analytical solution assumes very small values and for even small absolute errors this will produce large relative error.

Consider the dam-break problem in a channel of length $L = 2000$m conveying subcritical flow with an upstream water depth of $h_1 = 10$m and a downstream water depth of $h_0 = 5$m. Figures 6.10(a) and 6.10(b) show the L^1 error of the numerical solution for decreasing cell sizes. Here the straight line represents the exact first-order convergence rate. Although a second-order method has been used these figures clearly demonstrate a first-order convergence rate. However

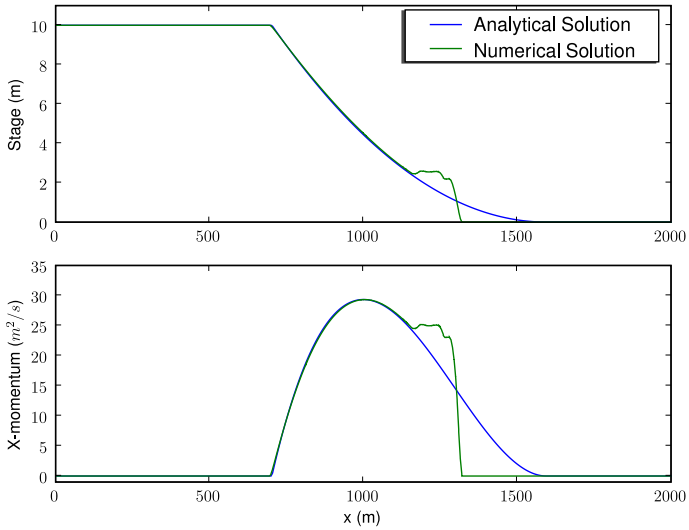


(a)

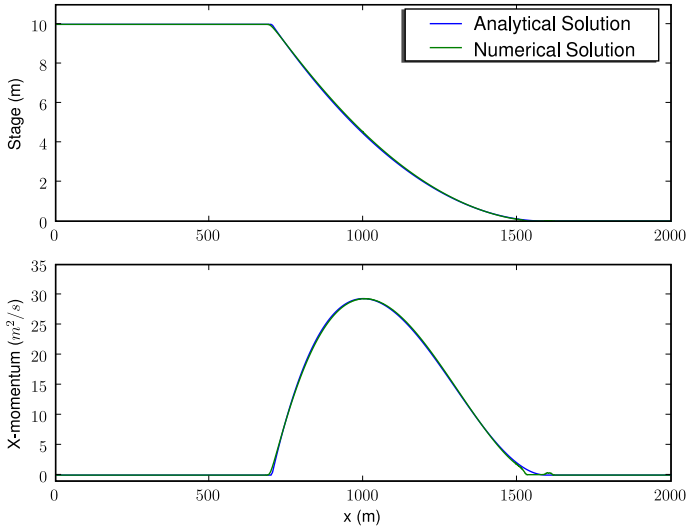


(b)

Figure 6.7: Numerical solution of the dam-break problem using 400 cells where $h_1 = 10$m and $h_0 = 0.0$m obtained at $t = 30$s using second-order central-upwind Godunov-type method with second-order time-stepping and the (a) Kurganov minmod limiter (b) Superbee limiter.

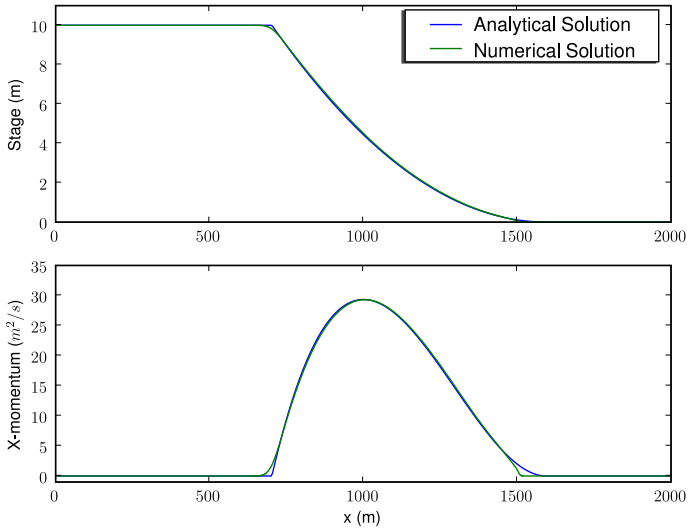


(a)

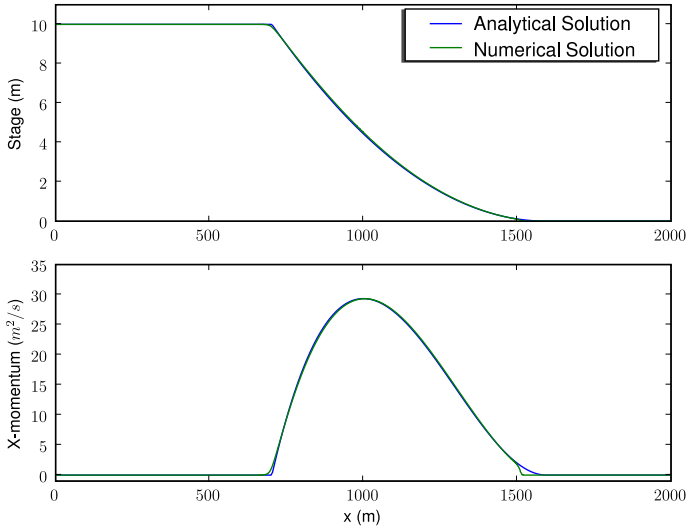


(b)

Figure 6.8: Numerical solution of the dam-break problem using 400 cells where $h_1 = 10$m and $h_0 = 0.0$m obtained at $t = 30$s using second-order central-upwind Godunov-type method with second-order time-stepping and the (a) Pyvolution limiter (b) van Leer limiter.



(a)



(b)

Figure 6.9: Numerical solution of the dam-break problem using 400 cells where $h_1 = 10$m and $h_0 = 0.0$m obtained at $t = 30$s using second-order central-upwind Godunov-type method with second-order time-stepping and the (a) Minmod limiter (b) Van Albada limiter.

the central-upwind scheme presented here can be considered quasi second-order. That is the scheme is second-order in smooth regions and first-order close to shocks. This becomes evident when looking at steady flows (Section 6.4). First-order convergence is also demonstrated by Figures 6.11(a) and 6.11(b) which were obtained simulating a dam-break with an upstream water depth of $h_1 = 10$m and a downstream water depth of $h_0 = 0.1$m.

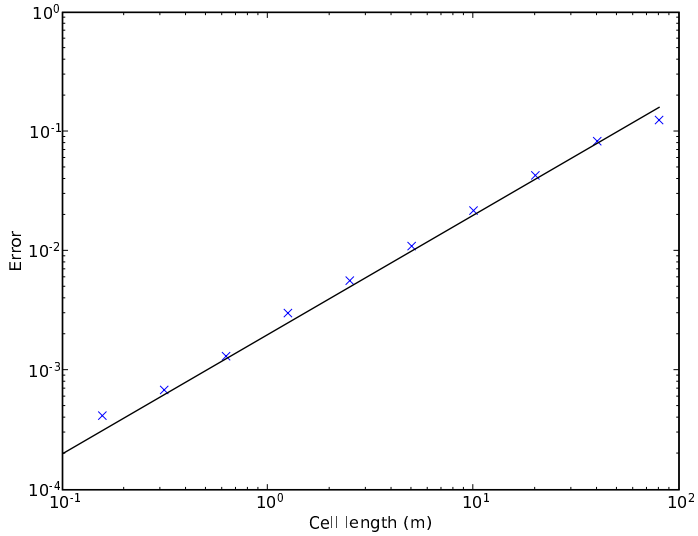
Finally consider the dry dam-break problem. As the grid is refined, the computational solution approaches the exact solution with approximately first-order accuracy. See Figures 6.12(a) and 6.12(b). At the finest resolution the convergence rate decreases due to the inability of the central-upwind scheme to completely resolve the advancing wet front. For coarser grid sizes this problem is masked by the larger errors in other areas. However as the grid is refined the error in these areas decreases faster than that at the wet/dry interface. Again this fact highlights the need for better approximations of water depth and velocity in this region.

6.3 Oscillating Planar Flow in a Frictionless Parabolic Canal

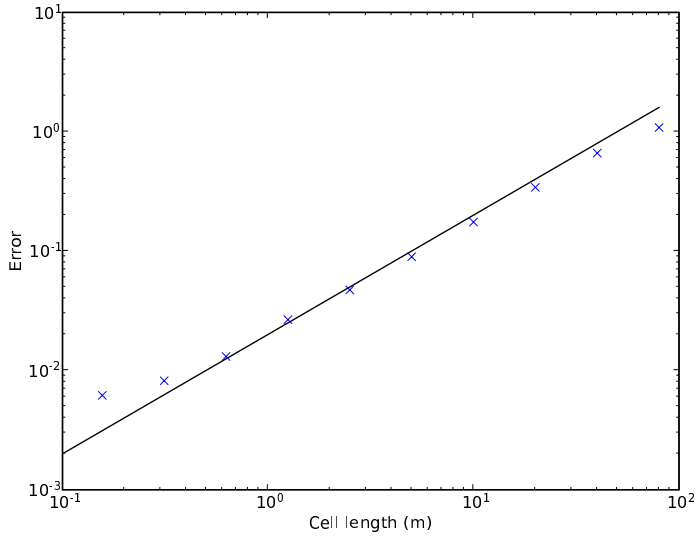
Flow in parabolic canals and basins admits an interesting class of analytical solutions which can be used to test the accuracy of numerical models of shallow water flow. Consider oscillating flow in a canal with no horizontal velocity in the y-direction, a maximum equilibrium water depth of $z_\infty = 10$m, and an equilibrium horizontal water surface length of $2L_x = 5,000$m. Here the free surface oscillates with an amplitude of $A_0 = L_x/2$ and a period of $T = 1122.0$ and the analytical solution of the free surface and water velocity is given by (3.21) and (3.22), respectively.

This problem proved to be the harshest test of the central-upwind scheme. Although the problem is smooth, the numerical scheme must constantly resolve the wetting and drying of cells and simulate the highly complex reversal of momentum that occurs when the planar surface ascending the side of the canal slows, stops and begins descending. When conserving stage (rather than water depth) the central-upwind scheme was very slow, taking an average time of 36.40 seconds to approximate the solution at $t = 841.5$s on a grid consisting of 100 cells.

Instead of reconstructing the elevation of the fluid surface by constructing limited slopes based on the value of stage at each cell centroid, one can equally reconstruct the free surface profile based on the water depth at the centroid. The resulting water surface profiles are markedly different as shown in Figures 6.13 and 6.14. When conserving depth the model runs approximately twice as fast as the scheme conserving stage. An average time of 16.28 seconds was

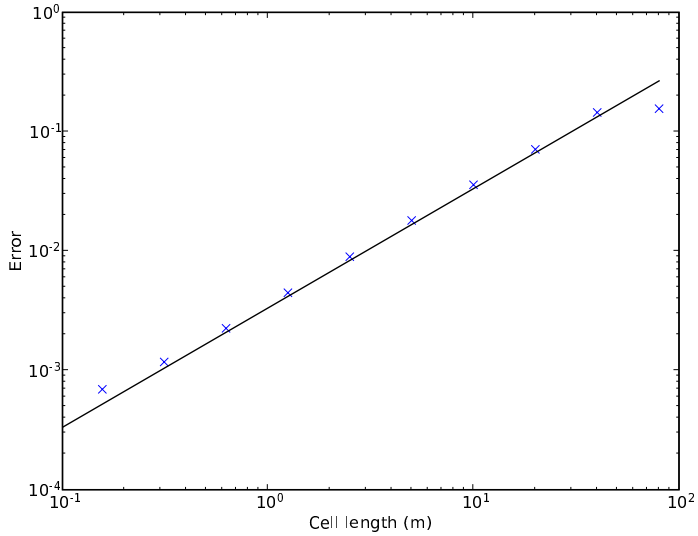


(a) Water depth (h).

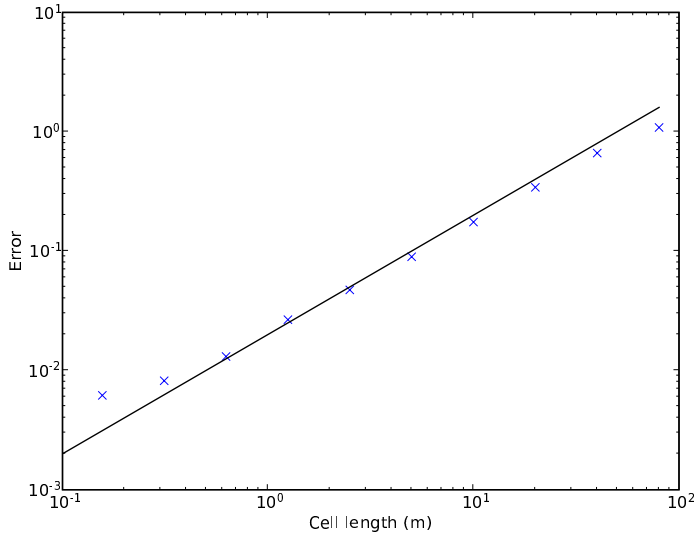


(b) Momentum (uh).

Figure 6.10: Convergence and accuracy of the numerical method when modelling a subcritical dam-break. The solid line is $\propto \Delta x$, and points $\times$ are the L_1 absolute error, given increasing computational resolution.

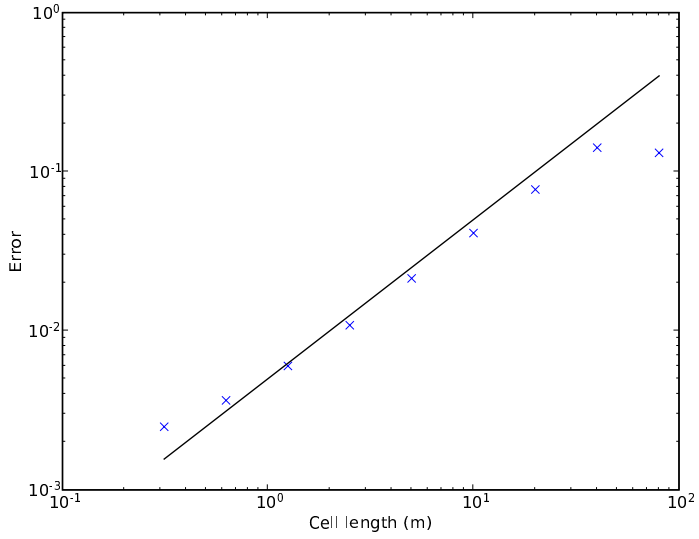


(a) Water depth (h).

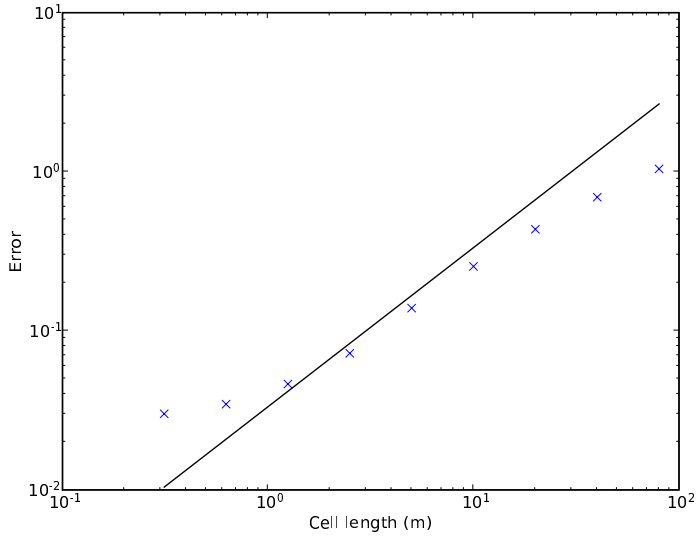


(b) Momentum (uh).

Figure 6.11: Convergence and accuracy of the numerical method when modelling a supercritical dam-break. The solid line is $\propto \Delta x$, and points $\times$ are the L_1 absolute error, given increasing computational resolution.



(a) Water depth (h).



(b) Momentum (uh).

Figure 6.12: Convergence and accuracy of the numerical method when modelling a dry dam-break. The solid line is $\propto \Delta x$, and points $\times$ are the L_1 absolute error, given increasing computational resolution.

Number of Cells	Stage w	Depth h
100	36.40 s	16.28 s
200	118.55 s	63.69 s
400	412.93 s	253.22 s
800	1481.12 s	1016.16 s
1600	5391.47 s	4062.42 s

Table 6.5: Average wall time (10 runs per grid size) taken to approximate the solution of oscillating planar flow in a parabolic canal at 841.5s when conserving stage and conserving depth.

required to simulate one oscillation on a grid consisting of 100 cells. The decrease in speed experienced when conserving stage w is a result of the small time steps generated in partially-wetted – cells with only one submerged vertex. Time step sizes are calculated using (5.15) which is inversely proportional to the maximum absolute wave speed found in the computational domain. In each cell the wave speed u is found by dividing the momentum uh in the cell by the depth h in the cell. When conserving stage the depth in partially-wetted cells can be small whilst the corresponding value of momentum is still relatively large, producing large absolute maximum speeds and thus small time steps. The frequency and severity of this phenomenon is reduced when depth is used to reconstruct the free surface profile. However as the computational grid is refined the differences between reconstructions of stage and depth become less evident and the model speeds converge accordingly. Consult Table 6.5. The two types of reconstruction produce results of similar accuracy.

The evolution of the numerical solution was simulated using the second-order central-upwind scheme on a grid consisting of 400 cells of equal length. The numerical solutions at $t = T/2$, $t = 3T/4$, $t = T$ and $t = 5T/4$ are shown in Figures 6.15(a), 6.15(b), 6.16(a) and 6.16(b). The form of the solution at the aforementioned times represents the most severe test of the central-upwind scheme this problem provides. At $t = T/2$, $t = T$ the water is at rest and the momentum of the fluid is changing direction. At the other times the magnitude of the fluid velocity is greatest. At all these times the numerical and analytical solutions of the water depth are indistinguishable, as are the numerical and analytical solutions of momentum at $t = 3T/4$ and $t = 5T/4$. The absolute error in the water depth E_h and momentum E_{uh} at each of these times is shown in Table 6.6. This table shows that the error in water depth is larger when the velocity is greater. However the error in the depth reconstruction decreases again with the velocity. This is not true of the

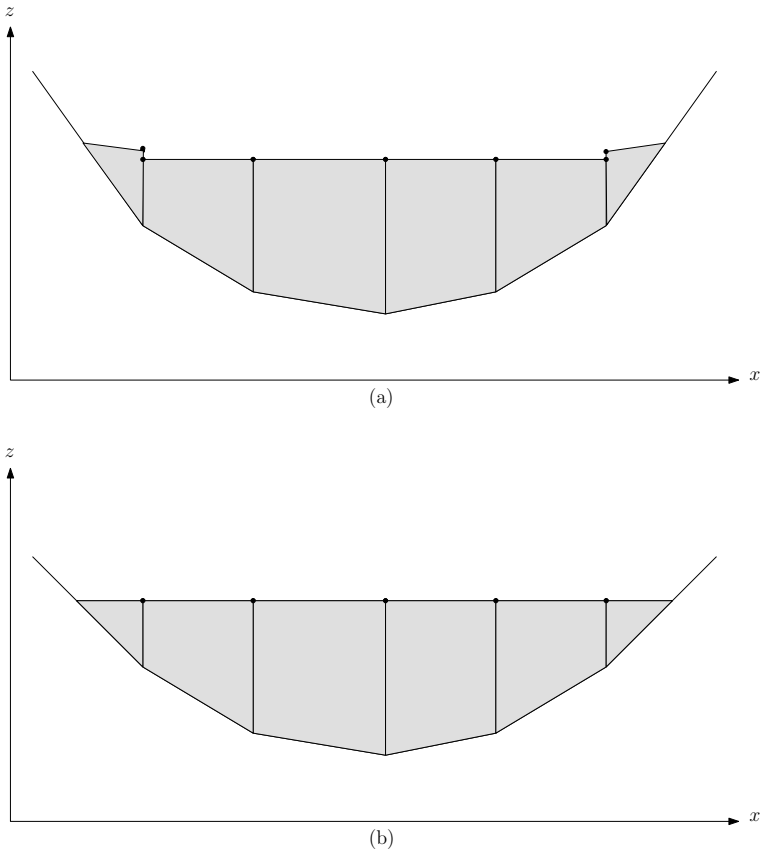


Figure 6.13: Second-order accurate free surface profiles with gradients, based on centroid stage values, subject to the minmod slope limiter with (a) stage at centroid of partially-wetted cell less than or equal to the associated bed elevation (b) stage in partially-wetted cells greater than the associated bed elevation.

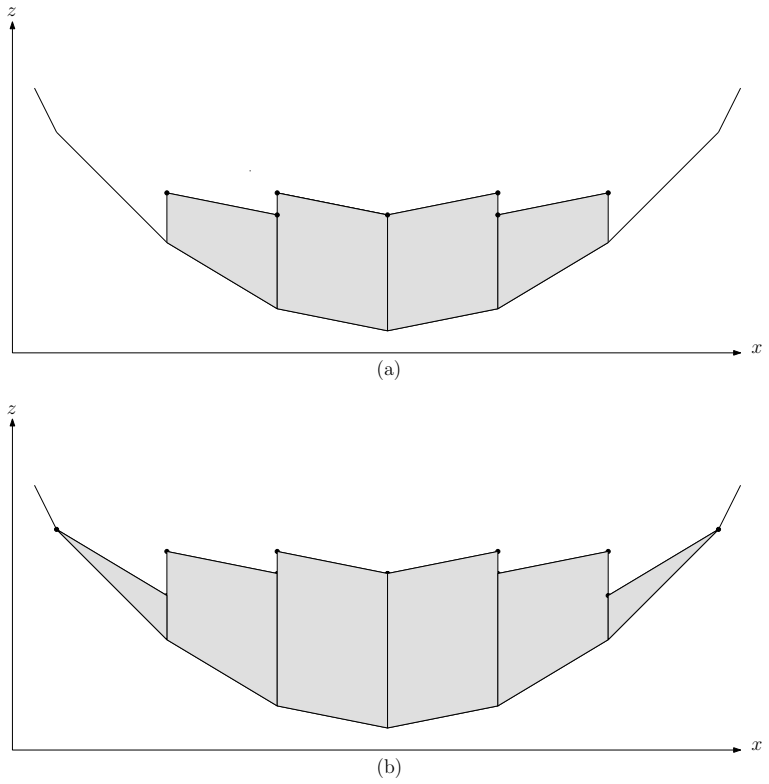


Figure 6.14: Second-order accurate free surface profile with gradients, based on centroid depth values, subject to the minmod slope limiter with (a) non-positive depth at centroid of partially-wetted cells (b) positive depth at centroid of partially-wetted cells.

error in the momentum profile which increases with time. Fortunately this error is bounded and is recovered with each oscillation.

t	E_h	E_{uh}
561.0s	0.0022157967	0.0335704661
841.5s	0.0049029748	0.0484488349
1122.0s	0.0028329655	0.0649583278
1402.5s	0.0074963810	0.0775239018

Table 6.6: Absolute error in numerical approximation of oscillating planar flow in a frictionless parabolic canal, at four different times, using the second-order central-upwind method, second-order time-stepping and the van Leer slope limiter.

6.4 Steady Flow

Fractional step methods are occasionally unable to accurately reproduce certain physically trivial situations such as undisturbed motionless water over topography, time-independent steady flows and small perturbations from such steady states. The simulation of flow at rest and flow over a bump enables one to check that the source terms are correctly evaluated.

Consider a horizontal frictionless channel, 50m in length, that contains a parabolic obstruction centred at $x_{max} = 25$m with a maximum elevation of $z_{b_{max}} = 0.2$m and width of $2b$. The bed elevation of this channel is given by

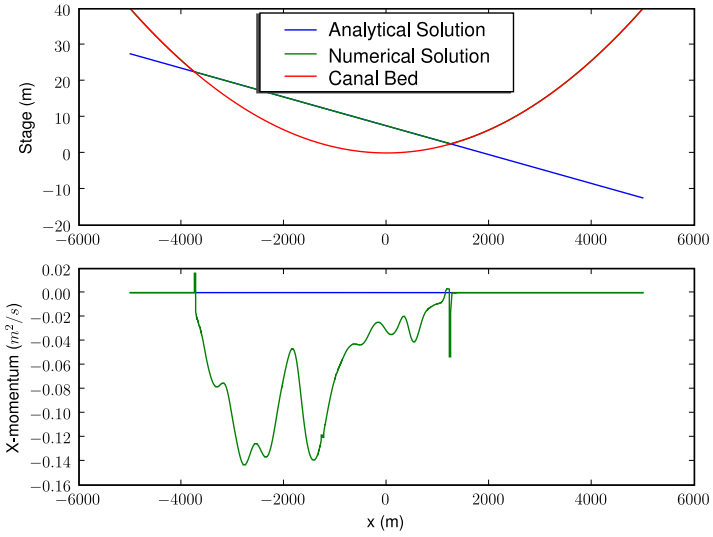
$$z_b(x) = \begin{cases} z_{b_{max}} \left[1 - \left(\frac{x - x_{max}}{b}\right)^2\right] & x_{max} - b < x < x_{max} + b \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

and, as shown in Section 3.3.3, the water surface profile can be calculated by solving

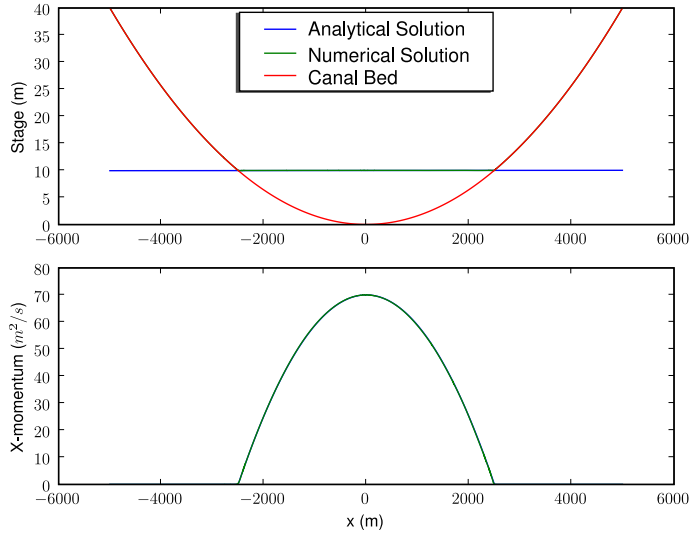
$$D^3 + D^2 \left(H - 1 - \frac{Fr_0^2}{2}\right) + \frac{Fr_0^2}{2} \tag{6.3}$$

where $h_0 = 0.5$m, $u_0 = 0.6$m/s, $D = h(x)/x$ and $Fr_0 = u_0/\sqrt{gh_0}$. The bisection method was used here.

The numerical solution to this problem was found using the second-order central-upwind scheme. Figure 6.17 compares the analytical and numerical solution of steady flow in a channel

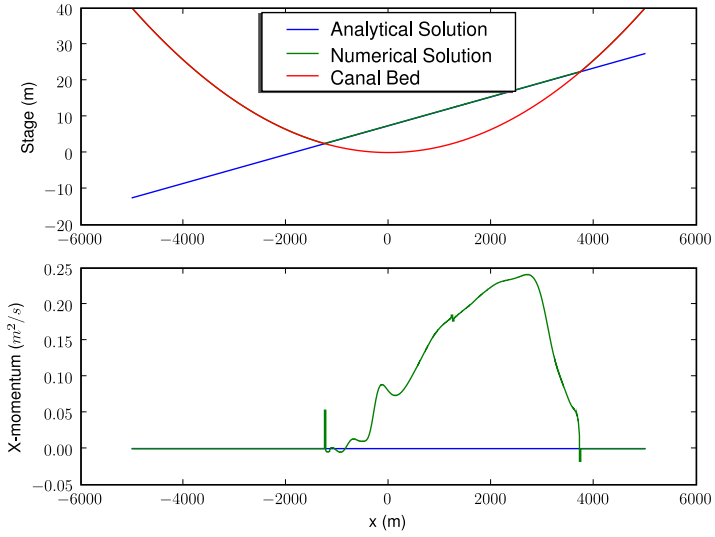


(a) $t = T/2$

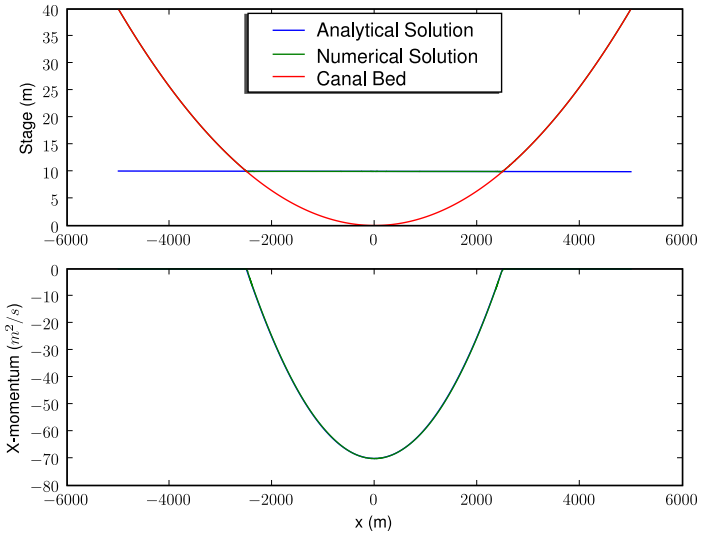


(b) $t = 3T/4$

Figure 6.15: Simulation of an oscillating free surface in a frictionless parabolic canal with a period of $T = 1122.0$s and $z_\infty = 10$m, $L_x = 2500$m, $A_0 = L_x/2$, $f = 0$, and $v = 0$



(a) $t = T$



(b) $t = 5T/4$

Figure 6.16: Simulation of an oscillating free surface in a frictionless parabolic canal with a period of $T = 1122.0$s and $z_\infty = 10$m, $L_x = 2500$m, $A_0 = L_x/2$, $f = 0$, and $v = 0$.

divided into 400 cells of equal length, $\Delta x = 0.125$. The analytical and steady numerical solutions obtained are indistinguishable. However we note that, at the discontinuities in the derivatives of the momentum profile peaks, in the error are seen. This is a phenomenon experienced by many numerical schemes and cannot be eliminated easily.

Demonstrating convergence to this exact solution is an important test for the numerical method, since it tests the ability of the scheme to simulate smooth solutions. Figures 6.18(a) and 6.18(b) shows convergence of the solution as the grid is refined. Figure 6.18(a) demonstrates the second-order accuracy of the method when approximating water depth. However the computational approximation of the water momentum converges to the true solution more slowly. This slower rate of convergence is caused by the peaks in the error found at the discontinuities in the derivatives of the momentum profile. The error at these points is large relative to the error in other regions and so dominates the total error. Thus because the larger errors only decrease at a rate proportional to Δx the same is true for the total L^1 error.

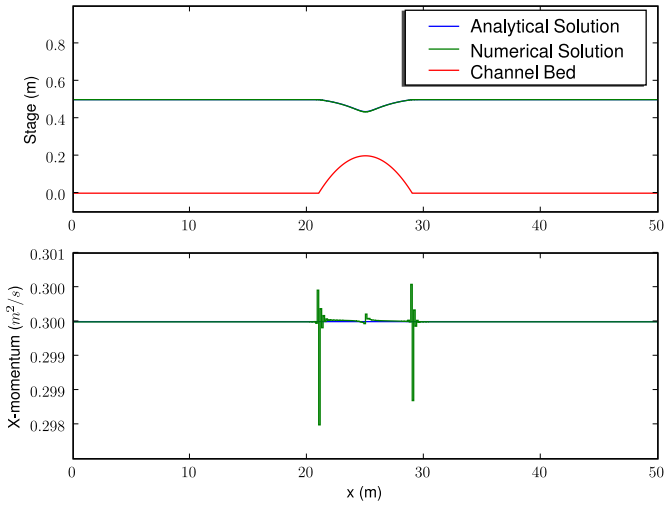
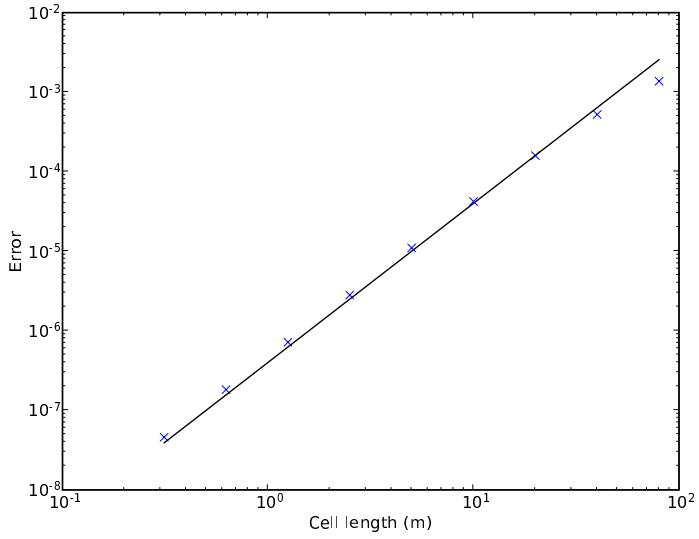
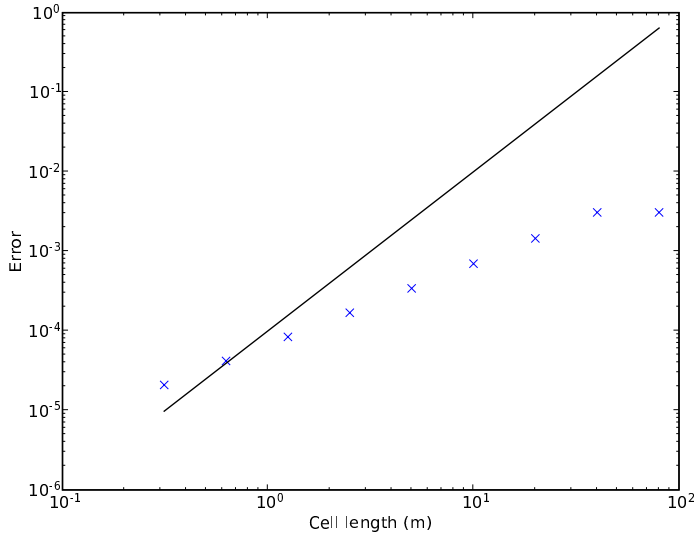


Figure 6.17: Steady-state water surface profile and velocity predicted at $t = 25$s in a horizontal frictionless channel, with a parabolic obstruction, conveying a subcritical flow of $q = 0.3m^2/s$. Results obtained using the second-order central-upwind scheme with $h_0 = h_1 = 0.5$m imposed at the boundaries.



(a) Water depth (h).



(b) Momentum (uh).

Figure 6.18: Convergence and accuracy of the numerical method when modelling steady flow. The solid line is $\propto \Delta x^2$, and points $\times$ are the L_1 absolute error, given increasing computational resolution.

6.5 Conclusion

This chapter investigated the veracity of the second-order central-upwind scheme by comparing the numerical model output against a number of analytical solutions. The model is able to accurately resolve shocks and simulate steady flow over rapidly-varying topography.

This second-order central-upwind scheme was initially implemented using a first-order forward Euler time-stepping method. It was found, however, that the second-order central-upwind scheme requires a time discretisation of the corresponding order. When the spatial and temporal discretisations are inconsistent, spurious oscillations pollute the solution and the amplitude of these oscillations increases, as the computational grid is refined. The second-order time-stepping method implemented is less computationally efficient. Nevertheless the increase in accuracy and the inaccuracies of the first-order temporal discretisation justify the increased computational effort.

A range of limiters were tested for the second-order central-upwind scheme. The Pyvolution, Superbee and Kurganov minmod limiters produce reasonable reconstruction of subcritical and supercritical flow from dam-breaks. But they produced extremely poor results when modelling dam-break over a dry bed. Due to the severity of the inaccuracies, use of these three limiters must be avoided. The van Leer limiter produced the most accurate results, followed by the van Albada limiter and the more diffusive minmod limiter. However all three limiters underestimated the velocity of the advancing wet of a dry dam-break. Further work is needed to find an appropriate method to adjust the limiting procedure at wet and dry interfaces.

The scheme is second-order accurate when modelling smooth flows and first-order accurate when modelling flow characterised by shocks. Subcritical flow over a parabolic obstruction produces smooth velocities and free surface profiles. As the computational grid is refined the numerical approximations of depth converge to the exact solution at a rate proportional to the square of the grid length. However the numerical approximation of momentum only converges at a rate proportional to the size of the grid length. This is caused by the larger errors found at the discontinuities in the derivatives, which dominate the solution and only decrease at a rate proportional to the grid length.

When finite water depth exists throughout the computational domain a dam-break will produce a shock. In this situation the central-upwind scheme produces numerical solutions that converge to the exact solution at a rate proportional to the cell length. When a dam-break occurs over a dry bed, a shock is not produced but the model still only produces first-order accurate so-

lutions. As the grid is refined the interface between the advancing wet front and the dry bed poses the most difficulties. Again this is related to the inaccuracies of the slope limiting procedure at the wet/dry interface.

Simulating oscillating planar flow in a parabolic canal proved the most severe test of the central-upwind scheme. Simulations conserving stage took significantly longer to execute than simulations conserving depth. The longer execution times are due to the smaller time steps required by the scheme conserving stage. The stage is reconstructed in a manner that can produce small values of depth and large values of momentum, resulting in large estimates of maximum wave speeds and consequently small time steps. The difference between execution times of the two schemes decreases as the computational grid is refined. Both schemes produce similar errors. The author believes that even more accurate results can be produced by employing a more plausible representation of the physical mechanisms of the wetting and drying of cells.

Chapter 7

Results II: Modifying the Wetting and Drying of Cells

As shown in the previous chapter, accurately modelling wetting and drying is one of the major challenges of simulating shallow water flow with finite volume methods. Finite volume methods relate the volume of fluid in a cell to the average depth in that cell. However in partially-wetted cells, in which only one vertex is submerged by water, this centroid value is an inaccurate representation of the average depth. For example, the linear reconstruction of the fluid surface in a wetted cell may produce a free surface elevation w_c that is below the bed elevation at the centroid z_c. See Figure 7.1.

If the surface of a partially submerged cell produces a surface elevation at the centroid less than the bed elevation at the centroid, the central-upwind described previously sets the stage at the centroid equal to the bed elevation at the centroid. If depth is being conserved rather than stage the water depth at the centroid is set to zero. Gradients are then calculated, and limited based on this new surface.

Begnudelli and Sanders [3] advocate another approach which reconstructs the water surface in partially-wetted cells in a manner that more accurately represent the physical mechanisms of the wetting and drying of cells. The following is adapted from their work on two-dimensional shallow water flow on unstructured triangular grids. The method proposed is based on volume free surface relationships (VFRs) that relate the volume of fluid in a cell to the elevation of the free surface (stage) in that cell. In particular they use the fact that the average flow depth h_c in each cell is equal to the ratio of the volume of fluid V in each cell to the cell area A. In one dimension the average flow depth is equal to the ratio of the area of fluid in each cell and the

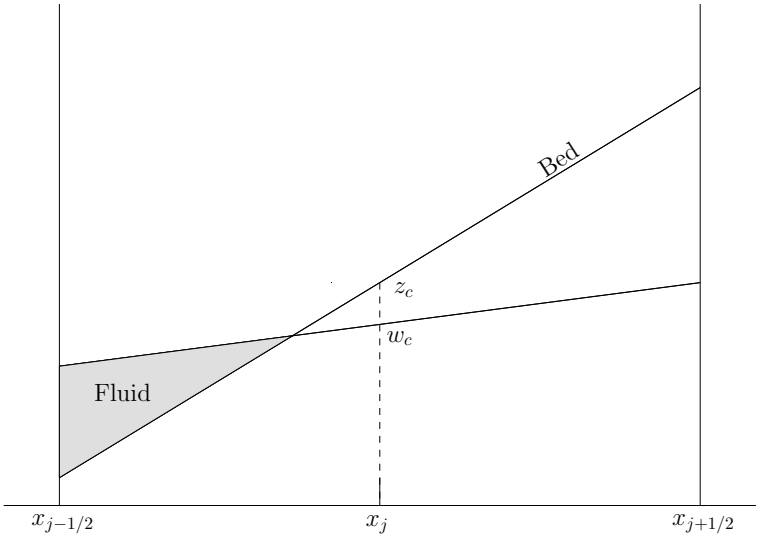


Figure 7.1: Surface profile of a partially wetted cell with a free surface elevation that is below the bed elevation at the centroid.

length of the cell L.

In wet cells, cells with both vertices submerged, the free surface elevation at the centroid is set to be $w_c = h_c + z_c$. However the same equality does not hold in partially-wetted cells and so the surface in partially-wetted cells must be updated differently. This is in contrast to the previously described method which updates all conserved quantities in a cell regardless of the amount of fluid present. For these cases Begnudelli and Sanders developed VFRs to determine the relationship between the stage w_c and the average depth h_c The following concerns the volume fluid ratios for one-dimensional domains.

7.1 Volume Free Surface Relationships

Let the vertex coordinates of the jth cell be (x_1, x_2) and the corresponding bed elevations and surface elevations be (z_1, z_2) and (w_1, w_2) respectively. The following assumes that $z_1 \leq z_2$, however similar expressions can be easily obtained for $z_1 \geq z_2$ by rearranging the vertex indices in an appropriate manner.

The VFR relating stage to depth is dependent on which cell vertex is submerged and the slope of the free surface in that cell. Trivially if $w_1 \leq z_1$ and $w_2 \leq z_2$ then no fluid is in the cell and $h_c = 0$. If $w_1 > z_1$ and $w_2 < z_2$ (Figure 7.2 (a)) the VFR is given by

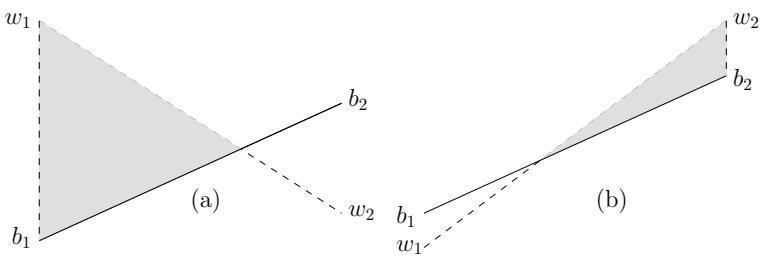


Figure 7.2: Water areas in partially-wetted cells when (a) $w_1 > z_1$ and $w_2 < z_2$ and (b) $w_1 < z_1 < z_2 < w_2$

$$h = \frac{A}{L} = \frac{(w_1 - z_1)(w_1 - z_1)}{2(z_2 - z_1 + w_1 - w_2)} \tag{7.1}$$

The above VFR is also used when $w_2 < z_1 < z_2 < w_1$. If $b_1 < w_1 < w_2 < b_2$, as shown in Figure 7.2 (b)

$$h = \frac{A}{L} = \frac{(w_2 - z_2)(z_2 - w_2)}{2(z_2 - z_1 + w_1 - w_2)} \tag{7.2}$$

The above VFR is also used when $w_1 < z_1$ and $z_2 < w_2$

These fluid volume ratios are used to determine the correct average depths in each cell given the free surface profile at $t = 0$. The initial stage profile is then used to obtain left and right states at each cell interface using $h_{ji} = w_{ji} - z_{ji}$, where h_{ji} is the water depth at the ith vertex of the jth cell. The central-upwind method is then used to evolve the solution through time to obtain new estimates of the average depth in each cell. Here the conserved quantities updated are water depth h and momentum uh. In partially-wetted cells the velocity is set to zero.

Once the values of depth and momentum have been updated at the centroids the free surface elevation at each centroid w_c must be computed using inverse VFRs that relate the updated average depth h_c in each cell to the stage w_c. This inverse problem is much more complex than the forward problem. At the start of the next time step the surface elevation in every cell is unknown as is the wet or dry status of each of cell. Although a cell maybe wet at the beginning of the previous time step it may have become partially wet or dry and, similarly, a dry cell may have become wet and a partially-wetted cell dry or completely wet.

We must first determine the stage at the vertices of every cell. In a particular cell the stage value at these vertices represents two unknowns that require two points of information – the average depth h_c (known) and the slope of the stage in that cell (unknown). The simplest solution is to approximate the slope by the slope in that cell at the previous times step. However when

implemented, and used to simulate oscillating planar flow in a canal, this approach was unstable and no result could be obtained. Alternatively the slope can be approximated using upwind, central, or downwind differencing. However this approach requires that the stage at each cell centroid be known, which it is not.

Begnudelli and Sanders [3] address this problem by assuming that the stage in partially-wetted cells is uniform. However, this assumption does not address the fact that the number and location of partially-wetted cells is not known. Instead we must initially assume that the surface in every cell is horizontal. The resultant forward and inverse VFRs are discussed below.

If $z_1 < z_2$ (refer to Figure 7.3(a)) the forward VFR is given by

$$h_c = \frac{1}{2}\frac{(w_c - z_1)^2}{(z_2 - z_1)} \tag{7.3}$$

and the associated inverse VFR is

$$w_c = \sqrt{2(z_2 - z_1)h_c} + z_1 \tag{7.4}$$

Similarly if $z_1 > z_2$ the forward VFR is given by

$$h_c = \frac{1}{2}\frac{(w_c - z_2)^2}{(z_2 - z_1)} \tag{7.5}$$

and the inverse VFR by

$$w_c = \sqrt{-2(z_2 - z_1)h_c} + z_2 \tag{7.6}$$

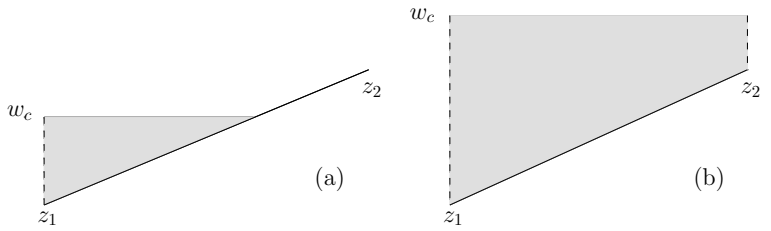


Figure 7.3: Water areas in (a) a partially-wetted cell $z_1 < w_c \leq z_2$ and (b) a wet cell $w_c < z_2$. Here stage is assumed to be uniform in each cell.

If $h_{cj} \leq 0$ the cell is dry. If the value of $h_{cj} = A/L \geq |z_2 - z_1|/2$ the cell is deemed wet and the stage at the centroid is given by $w_{cj} = h_{cj} + z_{cj}$. This is the maximum area that a horizontal surface can make with the bed elevation in the cell before both vertices are submerged. Refer to Figure 7.3 (b). Smaller values of h_{cj} indicate partially-wetted cells. The elevation of the free

surface w_{cj} in these cells is then calculated from the newly updated value of average water depth h_{cj} using (7.4) or (7.6). Unfortunately this classification is not perfect. Situations may arise for which a wet cell is wrongly classified partially wet because the average depth of the cell is less than $|z_2 - z_1|/2$. Refer to figure 7.4. A solution to this problem is not trivial and maybe intractable. Further intense scrutiny is required.

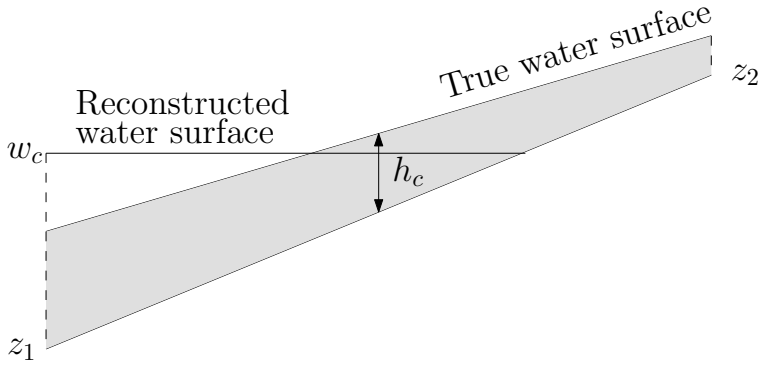


Figure 7.4: A Condition which results in the incorrect classification of a wet cell as partially-wetted.

Once the stage at the centroid of each cell has been determined the slopes of stage w and momentum uh are then computed and used to determine the left and right states at each cell interface. Note, that although water depth h is a conserved variable here, we calculate and limit slopes of the free surface elevation w to more accurately reconstruct planar surfaces.

Begnudelli and Sanders suggest that the centroid values be distributed to the cell vertices irrespective of the cells wet/dry status. However it was found that, at least for the central-upwind Godunov scheme, a cell's status must be taken into account. If slopes in partially-wetted cells are calculated and found to be non-zero then one can no longer ensure that the stage values at the vertices are updated in a way that preserves the VFR.

7.2 Overdraft

The time stepping methods described in section §5.4 may cause more water than the amount stored to be removed from a cell – overdraft. To ensure that centroid depths remain non-negative, i.e. $h_c \geq 0$. Begnudelli and Sanders propose that the excess amount of water removed be recovered from the appropriate neighbouring cell. This cell must satisfy the following two properties:

- Water flowed into the neighbouring cell from the over drafted cell during the time period when the overdraft occurred
- The storage in the neighbouring cell exceeds the overdraft.

The CFL condition provides another means of dealing with overdraft. Previously the central-upwind scheme was evolved with

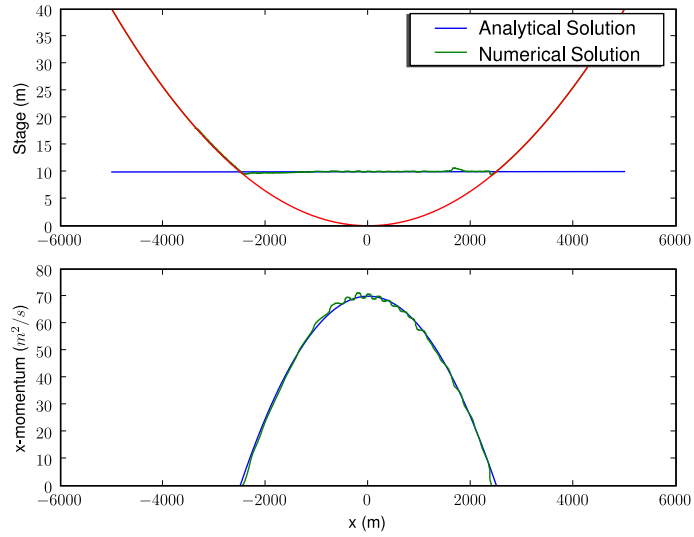
$$\Delta t = \min_{j=1,\ldots,N} C_r \frac{1}{2} \frac{\Delta x_j}{\max_p |\lambda_{j,p}|} \tag{7.7}$$

where $\max_p |\lambda_{j,p}|$ is the maximum absolute wave speed in cell j with length Δx_j. However we can ensure no overdraft occurs by setting Δx equal to the length of the cell submerged by water. In wet cells $\Delta x = x_2 - x_1$ and in partially-wetted cells $\Delta x = (-1)^i(x_i - x)$ where $i = 1, 2$ is the index of the submerged vertex and x is the point at which the stage intercepts the bed. Unfortunately, due to time constrains, software implementing this second approach was not finished.

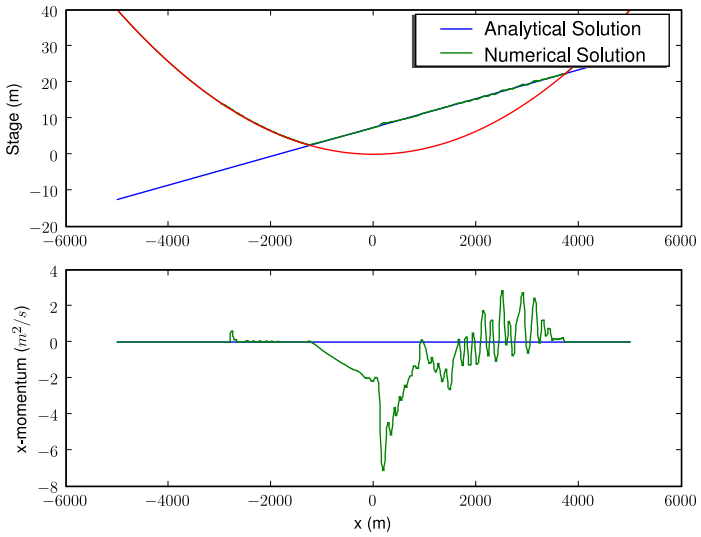
7.3 Oscillating Planar Flow in a Frictionless Parabolic Canal

Again consider the problem of oscillating flow in a canal with no horizontal velocity in the y-direction, a maximum equilibrium water depth of $z_\infty = 10$m, an equilibrium horizontal water surface length of $2L_x = 5000$m. Here the free surface oscillates with an amplitude of $A_0 = L_x/2$ and a period of $T = 1122.0$. This problem was solved using the second-order central-upwind scheme on a grid consisting of 400 cells of equal length. The numerical and analytical solutions at two different times are shown in Figures 7.5(a) and 7.5(b).

These plots show that the central-upwind scheme with the new wetting and drying procedure is less accurate than the solution obtained without the modification. The approximation of the fluid's momentum is extremely poor. Simply setting the velocity equal to zero in partially-wetted cells is inadequate. More accurate predictions of the velocities of wetting fronts is needed. Much greater accuracy can be obtained by using the analytical solution to reconstruct the momentum of the fluid at each time step and observing only the algorithms ability to simulate water depth. However due to the high correlation between momentum and depth this is not a very interesting or unexpected result. Various numerical experiments that used the analytical value of momentum in only partially-wetted cells were also carried out. However no benefits were noticeable.



(a)



(b)

Figure 7.5: Simulation of an oscillating free surface in a frictionless parabolic canal with a period of $T = 1122.0$s and $z_\infty = 10$m, $L_x = 2500$m, $A_0 = L_x/2$, $f = 0$, and $v = 0$, with new reconstructions in partially-wetted cells.

Here we conclude that the algorithm proposed by Begnudelli and Sanders is not as simple as their published work implies. Some necessary information is not given and this leads to the inconsistencies regarding cell classification and stage reconstruction discussed above. The basic premise of their work, however, is credible and worth exploring further. Here we conclude that the algorithm proposed by Begnudelli and Sanders is not as simple as their published work implies. Some necessary information is not given and this leads to the inconsistencies regarding cell classification and stage reconstruction discussed above. The basic premise of their work, however, is credible and worth exploring further.

Chapter 8

Conclusions

In this thesis a mathematical derivation of the two-dimensional shallow water wave equations has been given. We began with the simplest assumptions leading to Euler's equations for a perfect fluid. From these equations the shallow water wave equations are constructed under the following set of assumptions: that the waves are very long, there is no vertical acceleration, and the pressure distribution is hydrostatic. From the first assumption one may obtain conditions leading to the second and third assumptions. Relaxing these assumptions we can obtain the related Boussinesq-type equations. This class of equations can be used to model a range of flows, but is even more difficult to solve than the shallow water wave equations.

The shallow water wave equations form a hyperbolic non-linear system. The non-linear nature of this system renders analytical solutions difficult. The equations will often admit discontinuous solutions even when the initial conditions are smooth. These discontinuous solutions do not satisfy the partial differential equation form of the shallow water wave equations and so the integral form of the equations must be considered. This form can be used to determine the Rankine-Hugoniot condition that must hold across shocks. Any solution that satisfies this condition is a weak solution. Unfortunately the Rankine-Hugoniot condition does not guarantee unique solutions. Physically-admissible weak solutions must also satisfy the Lax entropy condition.

The non-linear behaviour of the shallow water equations also means that analytical solutions are restricted to a small set of idealised problems. Most practical problems necessitate the use of numerical techniques. Finite volume and finite element methods are the two main techniques used to simulate shallow water flow. Naive methods, which do not consider the properties of the system being modelled, are generally inadequate and often produce entropy-violating and/or

extremely diffusive solutions. As a result many schemes used to model the shallow water wave equations are complex and often involve, the computationally expensive, approximation of Riemann problems at each cell interface.

This thesis presented a central-upwind Godunov-type finite volume method for solving the shallow water wave equations, based on the work of Kurganov et al. [30]. This scheme involves no Riemann solvers or characteristic decomposition and achieves a high resolution due to the small amount of numerical dissipation introduced. The local propagation speeds of the rarefaction fans at each cell interface are used to estimate the size of the fan. The solution in these discontinuous regions are then averaged and combined with the estimates of the solutions in the smooth regions to obtain a final approximation. A fractional-step method is used to reduce the non-homogeneous equations into a sequence of augmented problems. The solution is then evolved through time using a second-order Runge-Kutta method. Another advantage of this method is that it can easily be extended to simulate two-dimensional flows.

The central-upwind scheme presented in this thesis has been implemented in a modularised object-oriented software package written in `Python`. This code was developed from scratch, to facilitate greater understanding and more comprehensive evaluation of the numerical technique in question. Use of pre-built software may not have identified some of the numerical difficulties discussed below.

This second-order central-upwind scheme was initially constructed using a first-order forward Euler time-stepping method. It was found, however, that the second-order central-upwind scheme requires a time discretisation of the corresponding order. When the spatial and temporal discretisations are inconsistent, spurious oscillations pollute the solution and the amplitude of these oscillations increases, as the computational grid is refined. The second-order time-stepping method implemented is less computationally efficient. Nevertheless the increase in accuracy and the inaccuracies of the first-order temporal discretisation necessitates the increased computational effort.

A range of limiters have been tested for the second-order central-upwind scheme. The Pyvolution, Superbee and Kurganov minmod limiters produce reasonable reconstructions of subcritical and supercritical flow from dam-breaks. But they produced extremely poor results when modelling dam-break over a dry bed. Due to the severity of the inaccuracies, use of these three limiters must be avoided. The van Leer limiter produced the most accurate results, followed by the van Albada limiter and the more diffusive minmod limiter. However all three limiters underestimated the velocity of the advancing wet of a dry dam-break. Further work is needed to find

an appropriate method to adjust the limiting procedure at wet and dry interfaces.

The veracity of this model was investigated by comparing the model output against a number of analytical solutions. The central-upwind scheme is able to accurately resolve shocks and model steady flow over rapidly-varying topography. The scheme is second-order accurate when modelling smooth flows and first-order accurate when modelling flow characterised by shocks.

Subcritical flow over a parabolic obstruction produces smooth velocities and free surface profiles. As the computational grid is refined the numerical approximations of depth converge to the exact solution at a rate proportional to the square of the grid length. However the numerical approximation of momentum only converges at a rate proportional to the size of the grid length. This is caused by the larger errors found at the discontinuities in the derivatives which dominate the solution and only decrease at a rate proportional to the grid length.

When finite water depth exists throughout the computational domain a dam-break will produce a shock. In this situation the central-upwind scheme produces numerical solutions that converge to the exact solution at a rate proportional to the cell length. When a dam-break occurs over a dry bed, a shock is not produced but the model still only produces first-order accurate solutions. As the grid is refined the interface between the advancing wet front and the dry bed poses the most difficulties. Again this is related to the inaccuracies of the slope limiting procedure at the wet/dry interface.

Simulating oscillating planar flow in a parabolic canal proved the most severe test of the central-upwind scheme. Simulations conserving stage took significantly longer to execute than simulations conserving depth. The longer execution times are due to the smaller time steps required by the scheme conserving stage. The stage is reconstructed in a manner that can produce small values of depth and large values of momentum resulting in large estimates of maximum wave speeds and consequently small time steps. The difference between execution times of the two schemes decreases as the computational grid is refined. Both schemes produce similar errors. The author believes that even more accurate results can be produced by employing a more plausible representation of the physical mechanisms of the wetting and drying of cells.

An attempt was made to adapt the procedure employed by Begnudelli and Sanders [3] to model the wetting and drying of two-dimensional shallow water flow on unstructured triangular grids. However the attempts documented here produced less accurate results. A more accurate prediction of the velocity in partially wetted cells is needed.

In this thesis we have developed a one-dimensional model for modelling shallow water flow. Although few real flows are truly one-dimensional, it is hoped we have demonstrated, through the

simpler 1D representation, many of the various mathematical and computational considerations central to computational hydraulics.

Appendix A

Simulating a Subcritical Dam-Break

The code developed to implement the central-upwind scheme presented in this thesis is too large to be included here in its entirety. The following presentation aims to provide insight into the process required to initialise and execute a scenario. Here we use the concrete example of the subcritical dam-break described in §3.3.1.

To set up a particular scenario the user specifies the geometry (bathymetry and topography), the initial water level, boundary conditions such as tide, and any forcing terms that may drive the system such as pressure gradients and friction. Here we construct the computation grid using the following commands

```
L = 2000.0 # Length of channel (m)
N = 100 # Number of computational cells
cell_len = L/N # Origin = 0.0
points = zeros(N+1,Float)
for i in range(N+1):
    points[i] = i*cell_len
domain = Domain(points)
```

When the model is run, the mesh is converted into a domain object that represents the study area, quantities, boundaries and forcing terms. The domain object also contains methods for time stepping, flux calculations, and all other numerical operations pertinent to the model.

The conserved quantities updated by the numerical scheme are stage w, through the conserved quantity h, and horizontal momentum. The initial values of the updated quantities is set using

```
domain.set_quantity('stage', stage)
domain.set_quantity('elevation', elevation)
```

Here the second argument of the `set_quantity` routine is a function that returns a vector specifying the values of the quantity of interest at each of the points in `domain`. In the case of the dam-break the function used to set the stage is

```
def stage(x):
   y = zeros(len(x),Float)
   for i in range(len(x)):
      if x[i]<=1000.0:
         y[i] = h1
      else:
         [i] = h0
   return y
```

A similar function is used to set the bathymetry elevation. Boundary conditions are bound to symbolic tags using

```
domain.set_boundary({'exterior':  Reflective_boundary(domain)})
```

This routine uses a lookup table implemented as a Python dictionary as input. The boundary objects are all assumed to be callable functions of the coordinate vector x. The software built implements two standard boundary types – Dirichlet and Reflective boundaries.

Once the physical properties of the domain have been set the following set of commands must be used to specify the order of the spatial and temporal discretisation, the C_r number controlling the time step size, the slope limiter to be used, and if necessary β required by some limiters such as the Kurganov Minmod limiter

```
domain.default_order = 1
domain.default_time_order = 1
domain.cfl = 1.0
domain.beta = 1.0
domain.limiter = "minmod"
```

The simulation is run using

```
yieldstep = 1.0
finaltime = 20.0
for t in domain.evolve(yieldstep = yieldstep, finaltime = finaltime):
    <update solution>
```

The parameter `finaltime` specifies the time period over which `evolve` operates. Control is passed to the body of the for-loop at each fixed yieldstep. This decouples the internal time stepping from the overall time-stepping so that outputs may be interrogated.

The nature of `<update solution>` depends on the temporal and spatial discretisation. If a first-order spatial and temporal discretisation is used the following process is executed at each time step.

```
domain.compute_fluxes()
domain.update_timestep(yieldstep, finaltime)
domain.compute_forcing_terms()
domain.update_conserved_quantities(self.timestep)
domain.distribute_to_vertices_and_edges()
domain.update_boundary()
```

If a second-order spatial and temporal discretisation is used then the following process is used.

```
domain.compute_timestep()
domain.solve_inhomogenous_second_order(yieldstep, finaltime)
domain.solve_homogenous_second_order(yieldstep,finaltime)
domain.solve_inhomogenous_second_order(yieldstep, finaltime)
```

Bibliography

[1] Y. Agnon, P.A. Madsen, and H.A. Schffer. A new approach to high-order Boussinesq models. *Journal of Fluid Mechanics*, 399:319–333, 1999.

[2] P. Bates and M. Anderson. Validation of hydraulic models. In *Model Validation: Perspectives in Hydrological Science*. John Wiley and Sons, Ltd., 2001.

[3] L. Begnudelli and B. Sanders. Unstructured grid finite volume algorithm for shallow water flow and scalar transport with wetting and drying. *Journal of Hydraulic Engineering*, 132(4):371–384, 2006.

[4] A.F. Bennett and B.S. Chua. Open boundary conditions for Lagrangian geophysical fluid dynamics. *Journal of Computational Physics*, 153(2):418–436, 1999.

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

[6] C.A. Blain and T.C. Massey. Application of a coupled discontinuous-continuous Galerkin finite element shallow water model to coastal ocean dynamics. *Ocean Modelling*, 10:283–315, 2005.

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

[8] J.P. Borris and D.L. Book. Flux-corrected transport, SHASTA: A fluid transport algorithm that works. *Journal of Computational Physics*, 11(1):38–69, 1973.

[9] S.W. Bova and G.F. Carey. An entropy variable formulation and applications for the two-dimensional shallow water equations. *International Journal For Numerical Methods in Fluids*, 23:29–46, 1996.

[10] D.M. Causon, C.G. Mingham, and D.M. Ingram. Advances in calculation methods for supercritical flow in splillway channels. *Journal of Hydraulic Engineering*, 125(10):1039–1050, 1999.

[11] H.M. Cekirge, R.W. Lardner, and R.J. Fraga. Adaptation of the solution of the two-dimensional tidal equations using the method of characteristics to wind induced currents and storm surges. *Computers and Mathematics with Applications*, 12(A):1081–1090, 1986.

[12] A. Chertock, A. Kurganov, and G. Petrova. Finite-volume-particle methods for models of transport of pollutant in shallow water. *Journal of Scientific Computing*, 27:189–199, 2006.

[13] A. J. Chorin and J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1979.

[14] B. Cockburn and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.

[15] B. Cockburn and C. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.

[16] S.F. Davis. A simplified TVD finite difference scheme via artificial viscosity. *SIAM Journal on Scientific and Statistical Computing*, 8:1–18, 1987.

[17] S.K. Dube, P.C. Sinha, and G.D. Roy. The numerical simulation of storm surges along the Bangla Desh coast. *Transactions of the American Mathematical Society*, 9:121–133, 1985.

[18] L. Fraccarollo and H. Capart. Riemann wave description of erosional dam-break flows. *Journal of Fluid Mechanics*, 461:183–228, 2002.

[19] L. Fraccarollo, H. Capart, and Y. Zech. A Godunov method for the computation of erosional shallow water transients. *International Journal for Numerical Methods in Fluids*, 41:951–976, 2003.

[20] L. Fraccarollo and E. Toro. Experimental and numerical assessment of the shallow water model for two dimensional dam break type. *Journal of Computational Physics*, 33(6):843–864, 1995.

[21] M.F. Gobbi, J.T. Kirby, and G. Wei. A fully nonlinear Boussinesq model for surface waves. Part 2. Extension to $o(kh)^4$. *Journal of Fluid Mechanics*, 405:181–210, 2000.

[22] S.K. Godunov. A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mathemayicsheski Sbornic*, 47:271–306, 1959. English translation available in U.S. Joint Publication Research Service.

[23] N. Goutal and F. Maurel. A finite volume solver for 1D shallow-water equations applied to an actual river. *International Journal for Numerical Methods in Fluids*, 38:1–19, 2002.

[24] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[25] T.J.R. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering*, 54:341–355, 1986.

[26] T.J.R. Hughes and T.E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular focus on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.

[27] C. Johnson, A .Szepessy, and P. Hansbo. On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. *Mathematics of Computation*, 54:107–129, 1990.

[28] A.B. Kennedy, Q. Chen, J.T. Kirby, and R.A. Dalrymple. Boussinesq modeling of wave transformation, breaking, and runup. *Journal of Waterway Port Coastal and Ocean Engineering*, 126(1):39–47, 2000.

[29] S. Koh. Honours thesis - theoretical and numerical simulation of rapidly-varying flows in open channels. The Australian National University, 2001.

[30] A. Kurganov, S. Noelle, and G. Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM Journal of Scientific Computing*, 23(3):707–740, 2001.

[31] C.B. Laney. *Computational Gas Dynamics*. Cambridge University Press, Cambridge, 1998.

[32] S. Latham. *A Free Surface Nonlinearity in the Simplified Steady-State Shallow Water Equations*. PhD thesis, The Australian National University, 1999.

[33] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Verlag, Basel, 1992.

[34] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, 2002.

[35] J.C. Levin, D.B. Haidvogel, B. Chua, A.F. Bennett, and M. Iskandarani. EulerLagrange equations for the spectral element shallow water system. *Ocean Modelling*, 12:348377, 2006.

[36] A. Kurganov D. Levy. Central-upwind schemes for the Saint-Venant system. *Mathematical Modelling and Numerical Analysis*, 36(3):397–425, 2002.

[37] J.A. Liggett. *Fluid Mechanics*. McGraw Hill, Inc., 1994.

[38] P.A. Madsen, H.B. Bingham, and H. Liu. A new Boussinesq method for fully nonlinear waves from shallow to deep water. *Journal of Fluid Mechanics*, 462:1–30, 2002.

[39] J.L. Mead. The shallow water equations in Lagrangian coordinates. *Journal of Computational Physics*, 200:654–669, 2004.

[40] C.G. Mingham and D.M. Causon. High-resolution finite-volume method for shallow water flows. *Journal of Hydraulic Engineering*, 124(6):605–614, 1998.

[41] J.B. Nixon and B.J. Noye. Prawn larvae dispersion modelling. In R.L. May and A.K. Easton, editors, *Computational Techniques and Applications*. Computational Mathematics Group, Australian Mathematical Society, World Scientific, 1996.

[42] H.J.M. Ogink, J.G. Grijsen, and A.J.H. Wijbenga. Aspects of flood level computations. In *International Symposium on Flood Frequency and Risk Analysis*, USA, 1986.

[43] J. Oliger and A. Sundstrom. Theoretical and practical aspects of some initial boundary value problems in fluid. *SIAM Journal of Applied Mathematics*, 35(3):419–446, 1978.

[44] D.H. Peregrine. Long waves on a beach. *Journal of Fluid Mechanics*, 27(4), 1967.

[45] A. Ritter. Die fortpflanzung der wasserwellen. *Zeitschrift des Vereines Deutscher Ingenieure*, 36(33):947–954, 1892.

[46] P.L. Roe. Some contributions to the modelling of discontinuous flows. *Lectures in Applied Mathematics*, 22:163–193, 1985.

[47] C. Shu. High-order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

[48] C. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *Journal of Computational Physics*, 77:4379–471, 1988.

[49] V. Singh and S.M. Bhallamundi. Conjective surface subsurace modeling of overland flow. *Advances in Water Resources*, 21(7):567–579, 1998.

[50] J.J. Stoker. *Water waves: the Mathematical Theory with Applications*. Interscience Publishers, New York, 1957.

[51] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal of Numerical Analysis*, 5:507–517, 1968.

[52] C. Synolakis, E. Okal, and E. Bernard. The megatsunami of December 26 2004. *National Academy of Engineering Publications*, 2005.

[53] H. Tennekes and J.L. Lumley. *A First Course in Turbulence*. Cambridge University Press, MIT Press, 1972.

[54] W. Thacker. Some exact solutions to the nonlinear shallow water wave equations. *Journal of Fluid Mechanics*, 107:499–508, 1981.

[55] Z. Wang and H.T. Shen. Lagrangian simulation of one-dimensional dam-break flow. *Journal of Hydraulic Engineering*, 125(11):1217–1221, 1999.

[56] P. Watts, S.T. Grilli, J.T. Kirby, G.J. Fryer, and D.R. Tappin. Landslide tsunami case studies using a Boussinesq model and a fully nonlinear tsunami generation model. *Natural Hazards and Earth System Sciences*, 3(5):391–402, 2003.

[57] C. Wu, G. Huang, and Y. Zheng. Theoretical solution of dam-break shock wave. *Journal of Hydraulic Engineering*, 125(11):1210–1215, 1999.

[58] H.C. Yee. Construction of explicit implicit symmetric TVD schemes and their applications. *Journal of Computational Physics*, 68(1):151–179, 1987.

[59] M.Y. Zhang, J. Monbaliu, and C.S. Yu. An Eulerian-Lagrangian method for the simulation of wave propagation. *Ocean Engineering*, 26:255276, 1999.

[60] C. Zoppou and S. Roberts. Catastrophic collapse of water supply reserviours in urban areas. *Journal of Hydraulic Engineering*, 125(7):686–695, July 1999.

[61] C. Zoppou and S. Roberts. Numerical solution of the two-dimensional unsteady dam break. *Applied Mathematical Modelling*, 24:457–475, 2000.

[62] C. Zoppou and S. Roberts. Explicit schemes for dam-break simulations. *Journal of Hydraulic Engineering*, 129(1):11–34, January 2003.