Numerical Solution of the Shallow Water Equations

G14PJA

Mathematics 3rd Year Project

Autumn 2014/15

School of Mathematical Sciences

University of Nottingham

Will Feavyour

Supervisor: Dr. J Billingham

Research group: Industrial and Applied Mathematics

Project code: JB P1

Assessment type: Investigation

I have read and understood the School and University guidelines on plagiarism. I confirm that this work is my own, apart from the acknowledged references.

Contents

1 Introduction **3**

2 Shallow Water Equations in One Dimension **4**

2.1 Navier-Stokes Equations 4

2.2 SWEs from Navier-Stokes 5

2.3 SWEs over a flat bed 6

2.4 Wave Speeds and Characteristics 7

3 Analytical Solution to Dam Break **8**

4 Lax-Friedrichs Method **10**

4.1 Derivative Approximations 10

4.2 Lax-Friedrichs Derivation 11

4.3 Accuracy and Stability 11

4.4 Dry Bed Issues 12

5 Kurganov-Tadmor Method **13**

5.1 Principals and Origin 13

5.2 Fully Discrete Conditions 15

6 Errors in Numerical Methods **15**

6.1 Errors Over Space 15

6.2 Errors Over Time 18

7 Tsunamis and SWEs **19**

7.1 Relevance of Application 19

7.2 SWEs for an Uneven Bed 19

7.3 Numerical Solutions 20

7.4 Comments on Accuracy 20

8 Conclusions **23**

Appendix A Matlab code: KT Dam Break **26**

Appendix B Matlab code: LxF Tsunami Model **27**

1 Introduction

The shallow water equations (SWEs) model the flow of a fluid where the wavelength of disturbances is much longer than its depth. They are also called the Saint Venant Equations after their 18th century originator, and can be derived from different conservation laws or their offspring. An understanding is gained in section 2.2 by working from the Navier-Stokes equations.

SWEs are commonly used to model dam breaks, tsunamis, open channel flow and similar scenarios. The dam break is the foremost scenario examined throughout the majority of this paper. Dam break models show the results when still water is released from behind a vertical dam to flow onto a dry bed. This set up gives enough initial conditions that, by imposing certain simplifications, an analytical solution can be found. Here, this analytical result is derived in section sec:analytical. Reading on, it is compared to two numerical solutions; garnered using the Lax-Friedrichs and Kurganov-Tadmor methods. These central schemes depend on discretizing time and space to approximate the derivatives within the wave equations. An aim is to implement the numerical approximations into a computer program for running different situations quickly. Through doing this in Matlab, it is seen that the Kurganov-Tadmor method is far more accurate, although the merits of each are discussed.

Uses for modelling tsunamis are examined less thoroughly. Using the Lax-Friedrichs scheme yields results which do not compare well with true expectations. As the field of study is so broad, there are many extensions which could be made using SWEs to the application in this report.

2 Shallow Water Equations in One Dimension

2.1 Navier-Stokes Equations

The shallow water equations (SWEs) are used to model waves where the depth of the water is much less than the wavelength of a given surface disturbance. They can be derived from the Navier-Stokes (NS) equations which describe viscous flow of fluids; arising from an application of Newton's Second Law. The origins of the NS equations mean that we get equations for both the conservation of mass and that of momentum, 2.1 and 2.2 respectively.

$$\nabla \cdot \mathbf{V} = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}\rho\mathbf{V} + \nabla \cdot (\rho\mathbf{V} \cdot \mathbf{V}) = -\nabla p + \rho\mathbf{g} \tag{2.2}$$

Where $\mathbf{V} = \binom{u}{v}$ with u and v being the horizontal and vertical velocity fields respectively. Also, ρ is the fluid density, p is the pressure, and $\mathbf{g} = \binom{0}{g}$ is the gravitational constant. The surface tension terms have been neglected in equation 2.2. Variables which describe the height of the fluid are defined as in figure 2.1.

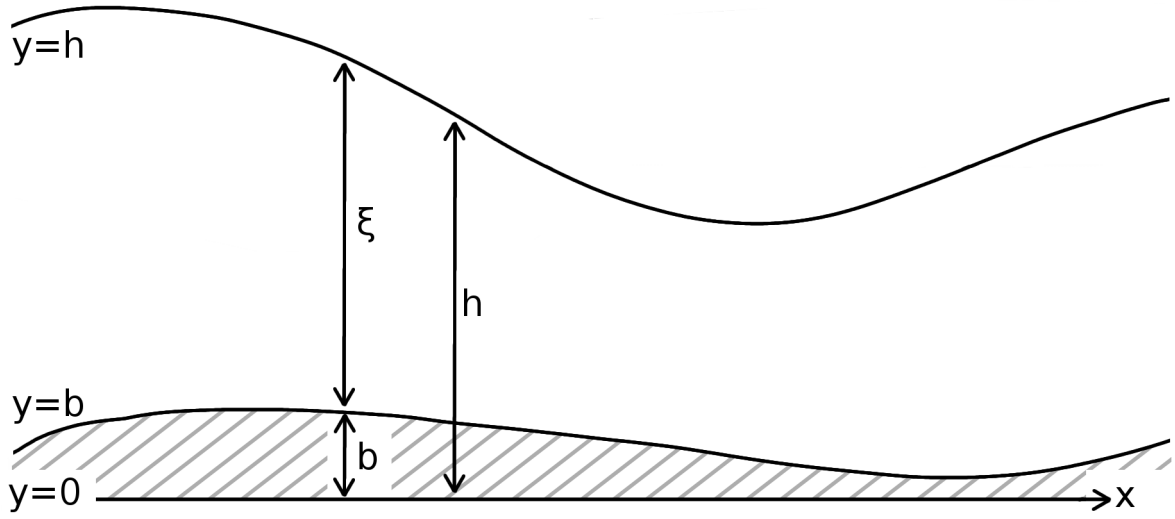


Figure 2.1: Definitions of quantities in SWEs.

2.2 SWEs from Navier-Stokes

Taking the vertical component of equation 2.2 we get 2.3

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + g \tag{2.3}$$

Neglecting vertical accelerations as per our assumptions about shallow water we get equation 2.4, showing that the pressure is purely hydrostatic as shown in [1, p. 110]. We then depth integrate to get the pressure, p, by equation 2.5.

$$-\frac{1}{\rho}\frac{\partial p}{\partial y} + g = 0 \tag{2.4}$$

$$p = p_0 + \rho g(h - y) \tag{2.5}$$

With p_0 as the surface pressure at $y = \xi$. Next, we take the horizontal component of equation 2.2, giving equation 2.6. Substituting the value of p gained in 2.5 and bearing in mind that u has no y dependence, we can simplify to get 2.8.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{2.6}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial}{\partial x}(p_0 + \rho g(h - y)) = 0 \tag{2.7}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} = 0 \tag{2.8}$$

For the last stage in the construction of the SWEs we use the Leibniz integral rule, 2.9, and integrate equation 2.1 over depth.

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\left(\int_{\beta(\theta)}^{\alpha(\theta)} f(\phi,\theta)\mathrm{d}\phi\right) = \int_{\beta(\theta)}^{\alpha(\theta)} \frac{\partial f(\phi,\theta)}{\partial\theta}\,\mathrm{d}\phi + f(\alpha,\theta)\cdot\alpha' - f(\beta,\theta)\cdot\beta' \tag{2.9}$$

$$0 = \int \nabla\cdot\mathbf{V}\,\mathrm{d}y = \int_b^h \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\,\mathrm{d}y \tag{2.10}$$

$$0 = \int_b^h \frac{\partial u}{\partial x}\,\mathrm{d}y + v|_{y=\xi} - v|_{y=-b} \tag{2.11}$$

$$0 = \frac{\mathrm{d}}{\mathrm{d}x}\left(\int_b^h u\,\mathrm{d}y\right) - u|_{y=h}\cdot\frac{\partial h}{\partial x} + u|_{y=b}\cdot\frac{\partial b}{\partial x} + v|_{y=h} - v|_{y=b} \tag{2.12}$$

Using boundary conditions for the system as stated in [5, p. 18] we know that, at the static floor $y = b$, 2.13 holds with normal velocity vanishing.

$$u \cdot \frac{\partial b}{\partial x} - v = 0 \tag{2.13}$$

Also, at the free surface $y = h$ where the surface may move, we get the similar but time dependent equation 2.14; here applying the condition that the *relative* normal velocity vanishes.

$$\frac{\partial h}{\partial t} + u \cdot \frac{\partial h}{\partial x} - v = 0 \tag{2.14}$$

As the original SWE conditions assume that the horizontal velocity u does not vary greatly in the vertical direction, we can average it over the entire depth and define it as, say, $\tilde{u}$ as in 2.15.

$$\tilde{u} = \frac{1}{\xi} \int_b^h u \, \mathrm{d}y \tag{2.15}$$

Now, drawing the BCs together with 2.15 and 2.12 we get 2.16. Also recalling 2.8 we get the pair of SWEs for fluid with a free surface over an uneven, frictionless, bed.

$$\frac{\partial h}{\partial t} + \frac{\partial (\tilde{u}\xi)}{\partial x} = 0 \tag{2.16}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} = 0 \tag{2.17}$$

As the depth averaged horizontal velocity assumption will hold for all further uses of $\tilde{u}$, for simplicity let $\tilde{u} := u$.

2.3 SWEs over a flat bed

For a large part of the following sections, we are only interested in using these equations in situations where the bed topography is completely flat and horizontal. So letting $b = 0$ we have that $h = \xi$, and the general SWEs are 2.18 and 2.19.

$$\frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial x} = 0 \tag{2.18}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} = 0 \tag{2.19}$$

By substituting the relations in equations 2.18 and 2.19 into a product rule expansion of $\frac{\partial}{\partial t}(uh)$ we get a more suitable equation to use in conjunction with 2.18.

$$\begin{aligned}\frac{\partial(uh)}{\partial t} &= u\frac{\partial h}{\partial t} + h\frac{\partial u}{\partial t} \\ &= u \cdot \left(-\frac{\partial(uh)}{\partial x}\right) + h \cdot \left(-u\frac{\partial u}{\partial x} - g\frac{\partial h}{\partial x}\right) \\ &= -\frac{\partial}{\partial x}(u^2 h) - \frac{\partial}{\partial x}(\tfrac{1}{2}gh^2) \\ &\Rightarrow \frac{\partial(uh)}{\partial t} + \frac{\partial}{\partial x}(u^2 h + \tfrac{1}{2}gh^2) = 0 \end{aligned} \tag{2.20}$$

Now we have two equations in the same form. For speed, let us define the system of equations 2.18 and 2.20 as vectors, with $q = uh$ they become 2.22.

$$U = \begin{bmatrix} h \\ q \end{bmatrix}, \; F(U) = \begin{bmatrix} q \\ \frac{q^2}{h} + \frac{1}{2}gh^2 \end{bmatrix} \tag{2.21}$$

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{2.22}$$

2.4 Wave Speeds and Characteristics

By the theory of linear shallow water wavers, it can be seen that the local wave speed in terms of 2.18 and 2.19 is $c = \sqrt{gh}$. If this is substituted to frame the equations in terms of c, they become

$$\frac{1}{g}\frac{\partial c^2}{\partial t} + \frac{\partial}{\partial x}\left(\frac{uc^2}{g}\right) = 0 \tag{2.23}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial}{\partial x}\left(\frac{c^2}{g}\right) = 0 \tag{2.24}$$

Then we can simplify and combine to obtain 2.27.

$$(2.23) \Rightarrow 2\frac{\partial c}{\partial t} + 2u\frac{\partial c}{\partial x} + c\frac{\partial u}{\partial x} = 0 \tag{2.25}$$

$$(2.24) \Rightarrow \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + 2c\frac{\partial c}{\partial x} = 0 \tag{2.26}$$

$$\frac{\partial}{\partial t}(u \pm 2c) + (u \pm c)\frac{\partial}{\partial x}(u \pm 2c) \tag{2.27}$$

We can re-write 2.25 and 2.26 in the form of the system 2.28 with $A(B)$ and B as defined in 2.29. Then we can determine the eigenvalues of A to better understand the system.

$$\frac{\partial B}{\partial t} + A(U)\frac{\partial B}{\partial x} = 0 \tag{2.28}$$

$$A(U) = \begin{bmatrix} u & \frac{c}{2} \\ 2c & u \end{bmatrix}, \; B = \begin{bmatrix} c \\ u \end{bmatrix} \tag{2.29}$$

$$\det(A - \lambda I) = 0 \Rightarrow (u - \lambda)^2 - c^2 = 0 \tag{2.30}$$

$$\lambda_{\pm} = u \pm c = u \pm \sqrt{gh} \tag{2.31}$$

Given that $h \geq 0$ holds for all sensible real life interpretations, A is diagonalizable with real eigenvalues $\lambda_{\pm}$. As stated in [6], these eigenvalues are surface velocities of the fluid. It's worth noting that in dry areas, i.e. $h = 0$, they are equal. Now, $\lambda_{\pm} = \mathrm{d}X_{\pm}/\mathrm{d}t$ are characteristics; along which there are constant Riemann invariants, say $R_{\pm}$. It can easily be seen in 2.27 that these are $R_{\pm} = u \pm 2\sqrt{gh}$.

3 Analytical Solution to Dam Break

For some specific initial conditions, solutions to the system 2.22 can be determined analytically. One such scenario is the instantaneous disappearance of a vertical dam holding back a semi-infinite expanse of level fluid. The case where the dam overlooks a dry level bed is shown in figure 3.1. If this dam were to be removed, or "break", then we would expect a wave form to propagate in the positive x direction.

The solution to this problem is attributed originally to Ritter in 1892 [2]. The derivation can be understood by analysing the characteristics. As justified by Billingham and King (BK), there are 3 distinct regions governed by the characteristics [1, , p. 271].

The first is the region of undisturbed water. We define $c_0 = \sqrt{gh_0}$ as the initial linear wave speed, where h_0 is some positive constant. At $t = 0$, $h = h_0$ and we have still water behind the dam; $u = 0$. These conditions will hold whenever the characteristics originating from $t = 0$ intersect because of the coinciding values of $R_{\pm}$. The invariants

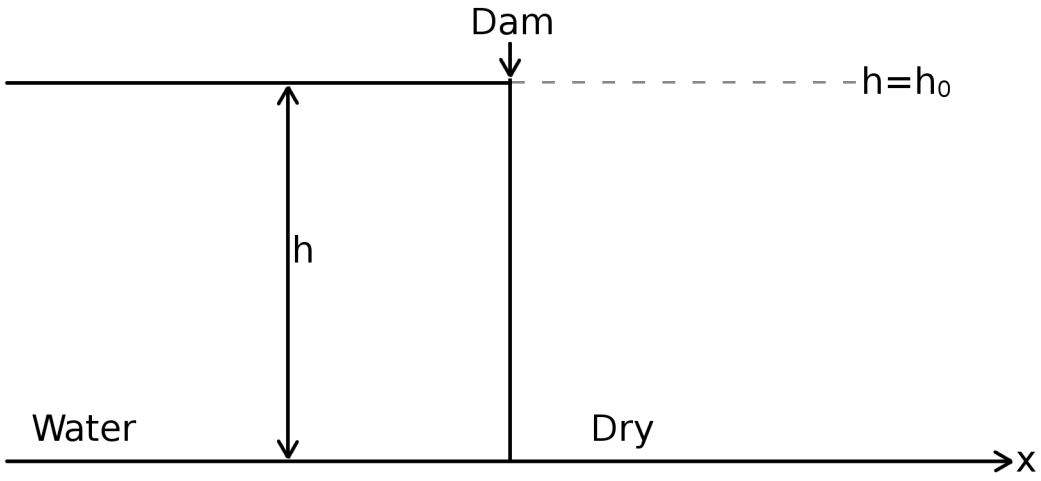


Figure 3.1: Initial conditions for the dam break scenario.

along the intersecting lines are $R_{\pm} = \pm 2c_0$, meaning that $u + 2c = 2c_0$ and $u - 2c = -2c_0$ must be concurrent at intersections, and so our initial conditions are held. Furthermore, $\mathrm{d}X_{\pm}/\mathrm{d}t = \pm c_0$ meaning that characteristics are straight lines. Because of this, the region of stillness is defined for $x < -c_0 t$.

The most interesting region is the propagating wave form in $-c_0 t < x < 2c_0 t$. Extracting information about the characteristics, we know that $\mathrm{d}X_-/\mathrm{d}t$ is constant, and as there is no fluid in $x > 0$ at $t = 0$, we integrate to get $X_- = (u - \sqrt{gh})t$. Also it must be assumed that the $\mathrm{d}X_+/\mathrm{d}t$ characteristics fill the region. By combining the line equation $x = (u - \sqrt{gh})t$ and the invariant $R_+ = u + 2\sqrt{gh} = 2c_0$, we can get a picture of what happens after the break. This culminates in the pair of equations 3.1.

$$h = \frac{h_0}{9}\left(2 - \frac{x}{c_0 t}\right)^2, \; u = \frac{2}{3}\left(c_0 + \frac{x}{t}\right) \tag{3.1}$$

In Ritter's equations for a simplified dam break, the minimum h is at $x = 2c_0 t$. BK's short analysis of the λ_+ characteristic shows that the upper boundary for the wave region coincides with this point. However, it can also be understood somewhat intuitively that this model would not have the wave form rising up past this minimum point in some sort of leading secondary wave.

Combining all of this, we get a resulting set of equations 3.2 which describe the three

distinct sections within the wave form.

$$\left.\begin{aligned} h = h_0, u = 0 \qquad & \text{for } x \leq -c_0 t \\ h = \frac{h_0}{9}\left(2 - \frac{x}{c_0 t}\right)^2, u = \frac{2}{3}\left(c_0 + \frac{x}{t}\right) \qquad & \text{for } -c_0 t \leq x \leq 2c_0 t \\ h = 0, u = 0 \qquad & \text{for } x \geq 2c_0 t \end{aligned}\right\} \tag{3.2}$$

4 Lax-Friedrichs Method

4.1 Derivative Approximations

There are many ways to approximate a derivative. The Lax-Friedrichs (LxF) method uses central differencing for the spatial derivative and forward differencing in time. To see how the central differencing approach works, we first must look at two Taylor series 4.1 and 4.2 for some spatial function $g(x)$, with a small change in x being Δx.

$$g(x + \Delta x) = g(x) + g'(x)\Delta x + g''(x)\frac{(\Delta x)^2}{2} + \mathcal{O}((\Delta x)^3) \tag{4.1}$$

$$g(x - \Delta x) = g(x) - g'(x)\Delta x + g''(x)\frac{(\Delta x)^2}{2} + \mathcal{O}((\Delta x)^3) \tag{4.2}$$

By subtracting these two formulas and rearranging in terms of the first derivative, we get equation 4.3. As the error is proportional to $(\Delta x)^2$, this is a second order approximation [7, p. 184].

$$g'(x) = \frac{g(x + \Delta x) - g(x - \Delta x)}{2 \cdot \Delta x} + \mathcal{O}((\Delta x)^2) \tag{4.3}$$

The time derivative is found using the forward Euler method which uses forward differencing. It is gained by looking again at Taylor series and rearranging for the derivative, as in equations 4.4 and 4.5.

$$f(t + \Delta t) = f(t) + f'(t)\Delta t + \mathcal{O}((\Delta t)^2) \tag{4.4}$$

$$f'(t) = \frac{f(t + \Delta t) - f(t)}{\Delta t} + \mathcal{O}((\Delta t)) \tag{4.5}$$

4.2 Lax-Friedrichs Derivation

The LxF method is FTCS (Forward-Time Central-Space); a term first coined by Patrick Roache [8], meaning that it uses the two derivative approximations discussed in 4.1. The use of this first order time approximation limits the LxF method, overall, to a be first order approximation in itself. Taking discrete time steps Δt, we must define our x axis as discrete points x_i with $x_{i+1} = x_i + \Delta x$. We define the shorthand $U_i^t = U(x_i, t)$ as the instantaneous value for U. Now 2.22 can be approximated as above, giving us equation 4.6 which is the beginnings of the LxF method.

$$\frac{U_i^{t+\Delta t} - U_i^t}{\Delta t} + \frac{F(U_{i+1}^t) - F(U_{i-1}^t)}{2\Delta x} = 0 \tag{4.6}$$

Rearranging, we get the time-iterative form 4.7. However, this is inherently unstable for discontinuous waves because of its reliance on U_i^t. To avoid this, we can employ a central difference approximation for this term by averaging the neighbouring values U_{i+1}^t and U_{i-1}^t. This results in the generalised form 4.8 of the LxF method as stated in [3, p. 125].

$$U_i^{t+\Delta t} = U_i^t - \frac{\Delta t}{2\Delta x}(F(U_{i+1}^t) - F(U_{i-1}^t)) \tag{4.7}$$

$$U_i^{t+\Delta t} = \frac{1}{2}(U_{i+1}^t + U_{i-1}^t) - \frac{\Delta t}{2\Delta x}(F(U_{i+1}^t) - F(U_{i-1}^t)) \tag{4.8}$$

4.3 Accuracy and Stability

Once implemented into Matlab code we can observe the system working for some arbitrary initial conditions; such as those in figures 4.1 and 4.2. In these figures we can see that the size of Δx is important, and in fact when it is too large we lose fidelity in the approximation of the shape of the fluid. When it is suitably small though, the method does gives a reasonable approximation to the true behaviour. It's also worth noting that the size of Δt must be proportional to that of Δx. This is because, to satisfy the Courant-Friedrichs-Lewy (CFL) condition for convergence, it must satisfy $\Delta t \leq |\Delta x / 2c|$ where c is the maximum local wave speed [4]. This condition varies over time as c is not constant, so it must be re-evaluated at every time step to ensure stability of the numerical solution.

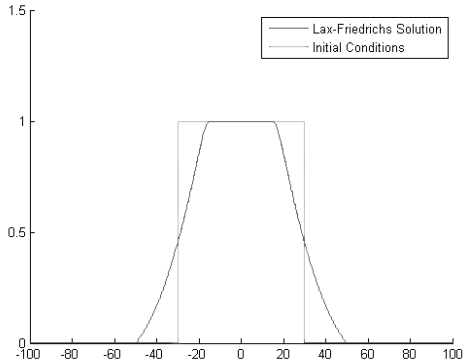


Figure 4.1: $t = 4$, $\Delta x = 0.1$

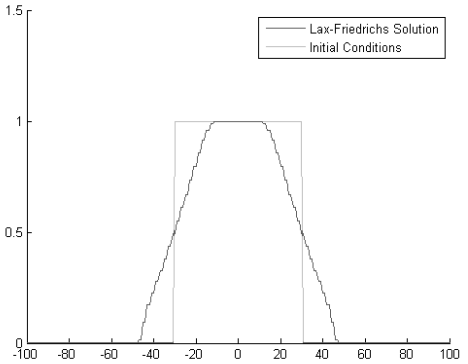


Figure 4.2: $t = 4$, $\Delta x = 1$

As this is just the minimum requirement, when implementing into code it is reassuring to use a safety scaling of Δt, say by 0.95, to ensure that the condition is satisfied irrespective of small floating point errors and such.

For the dam break scenario, we found in section 2.4 that the wave speeds are $u \pm \sqrt{gh}$, so the condition on Δt can be specified more. We must have $\Delta t \leq |\Delta x/2(u + \sqrt{gh})|$. In the dam break, u is always non-negative, so only the positive root is taken because $\max\{|u + \sqrt{gh}|, |u - \sqrt{gh}|\} = u + \sqrt{gh}$ in this case.

4.4 Dry Bed Issues

In the analytical solution outlined, we assume that in the positive x direction past the dam, the region is completely devoid of fluid and $h = 0$. However, for computational reasons this cannot be the case when using numerical methods. This is because there would be divide by zero errors when computing $F(U)$ as defined in 2.22. In avoiding this error, the depth is set to be very small; preserving accuracy to the true representation.

There is an interesting aside to observe, and that is when the region actually has some notable fluid depth over it. When the depth is $\sim 10^{-8}$ or smaller, the issues aren't noticeable at a reasonable scale. However, at orders of magnitude even slightly more similar to those of the dammed region's initial depth such as $\sim 10^{-4}$ or larger, there starts to be very apparent shocks forming at the wave front. An example can be seen

in figure 4.3, where the affect has been exaggerated with the initial fluid depth at 10^{-2}. This steep front propagates forward with the continued pressure from the wave behind it. It can be seen in the physical experiments in [9], as well as Stoker's generalised wet-bed form of Ritter's equations [6], that it does in fact reflect true to life. More complicated models such as those proposed by Hwung et al. [10] and experimental observations show that these waves broke in front of the steep front. The models used in this investigation do not allow for such wave forms.

With this note on shocks aside, we herein ensure that the dry bed depth is negligible and only non-zero where necessary to avoid failure in the numerical methods.

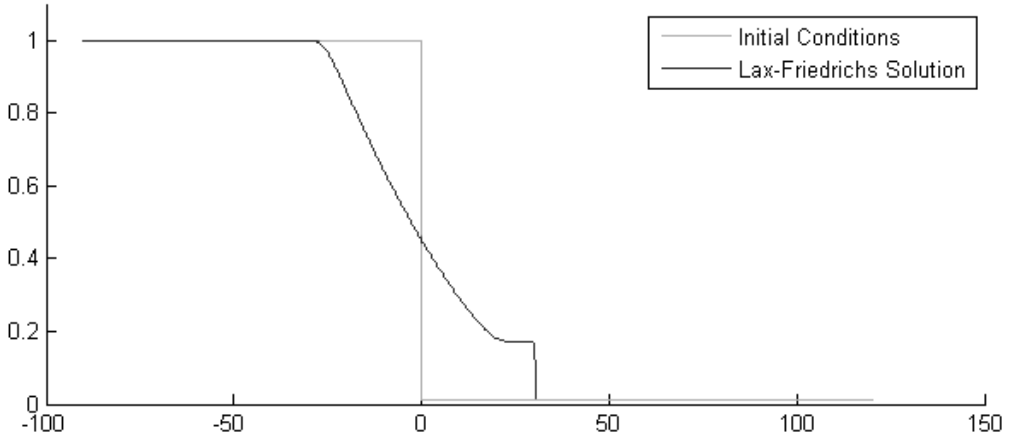


Figure 4.3: Shock forming at wave front of LxF approximation.

5 Kurganov-Tadmor Method

5.1 Principals and Origin

There are many different numerical methods which can be used to solve the SWEs. One central difference scheme developed on from LxF which offers greatly improved accuracy is that which was proposed by Kurganov and Tadmor (KT, [12]). Unlike the LxF method,

it doesn't suffer from such a great amount of dissipation and smoothing. Also it has the special property that it can be used in both a fully discrete or, leaving the time derivative in tact, semi discrete form.

Calculating approximate derivatives, as discussed in section 4.1, is an important component of these numerical solvers. The KT method utilises the *minmod limiter* function 5.1, taking into account the two approximate derivatives from neighbouring cells in a discrete grid. It outputs zero if they have opposing signs, and the minimum absolute derivative otherwise. For shorthand, the notation 5.2 is used; denoting numerical differentiation with respect to x at grid space $x_i = i\Delta x$.

$$\text{minmod}(a, b) := \frac{1}{2}(\text{sgn}(a) + \text{sgn}(b)) \cdot \min(|a|, |b|) \tag{5.1}$$

$$(U_x)_i = \text{minmod}\left(\frac{U_i - U_{i-1}}{\Delta x}, \frac{U_{i+1} - U_i}{\Delta x}\right) \tag{5.2}$$

The method is an extension of the Nessyahu-Tadmor method [13], with some minor but important differences. Unlike its predecessor, it uses additional information about the local propagation speeds at the boundaries of each cell. $U^+_{i+1/2}$ and $U^-_{i+1/2}$ are piecewise linear reconstructions of the boundaries [14], as defined by equations 5.3 and 5.4. For the application to SWEs, the maximum propagation speeds $a_{i+1/2}$ are found at these boundaries, and calculated by 5.5

$$U^+_{i+1/2} = U_{i+1} - \tfrac{\Delta x}{2}(u_x)_{i+1}(t) \tag{5.3}$$

$$U^-_{i+1/2} = U_i + \tfrac{\Delta x}{2}(u_x)_i(t) \tag{5.4}$$

$$a_{i+1/2} = \max\left\{\rho\left(\frac{\partial F}{\partial U}(U^-_{i+1/2}(t))\right), \rho\left(\frac{\partial F}{\partial U}(U^+_{i+1/2}(t))\right)\right\} \tag{5.5}$$

Where $\rho(M)$ is the spectral radius of M. For SWEs, we know that the $\rho(\partial F/\partial U) = \max\{|u + \sqrt{gh}|, |u - \sqrt{gh}|\}$. The method focusses on calculating averages between these narrower, non-uniform boundaries, before projecting values back onto a uniform grid which can be used for computation. The conservative, semi-discrete form of the KT method can be summarised by equation 5.6, with the numerical flux terms as defined in

equation 5.7.

$$\frac{\mathrm{d}}{\mathrm{d}t}U_i(t) = -\frac{H_{i+1/2}(t) - H_{i-1/2}(t)}{\Delta x} \tag{5.6}$$

$$H_{i+1/2}(t) = \frac{F(U^+_{i+1/2}(t)) + F(U^-_{i+1/2}(t))}{2} - \frac{a_{i+1/2}(t)}{2}[U^+_{i+1/2}(t) - U^+_{i+1/2}(t)] \tag{5.7}$$

5.2 Fully Discrete Conditions

Currently the KT method, as defined using the above components, is second order accurate and avoids diffusion issues faced by the LxF method. To retain the high accuracy which we gain from the method, it is noted ([12]) that high-order time discretizations are required. If desired, many different stability-preserving time discretization methods may be used, such as those discussed in [15]. For this investigation, the same forward Euler method is used as was in the LxF method; although there are different conditions on the size of Δt. Now, to guarantee that the value of h remains above zero, we must satisfy the CFL condition $\Delta t \leq \Delta x/4c$. This requires time steps half as large, which would obviously lead to more expensive computation. It is suggested in [16] that this is overcome by always satisfying the more loose condition $\Delta t \leq \Delta x/2c$, and only being more strict if this results in any negative h. It's also stated that it would be uncommon that the loose condition isn't satisfied, so it would be rare to have to double up on processing in any given time step.

Using the discrete equivalent of the numerical flux in 5.7, one fully discrete form of the KT method would be equation 5.8.

$$U_i^{t+\Delta t} = \frac{1}{2}(U^t_{i+1} + U^t_{i-1}) - \frac{\Delta t}{\Delta x}(H^t_{i+1/2} - H^t_{i-1/2}) \tag{5.8}$$

6 Errors in Numerical Methods

6.1 Errors Over Space

Figures 6.1 and 6.2 show the LxF and KT methods respectively by comparison to the analytical solution to the dam break scenario. Both give reasonable approximations to

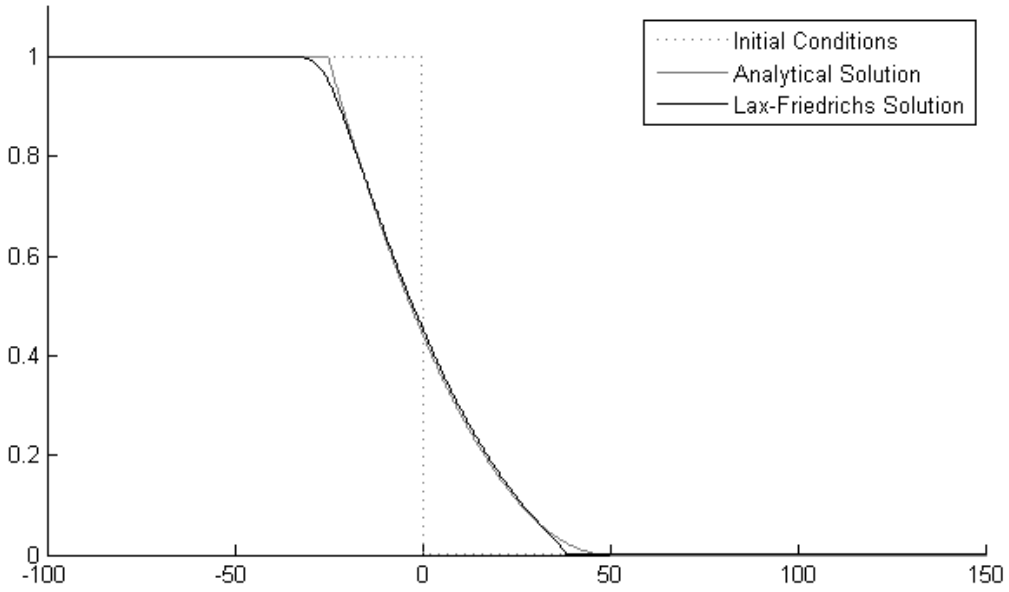


Figure 6.1: LxF approximation for dam break, $t = 10, \Delta x = 0.1$

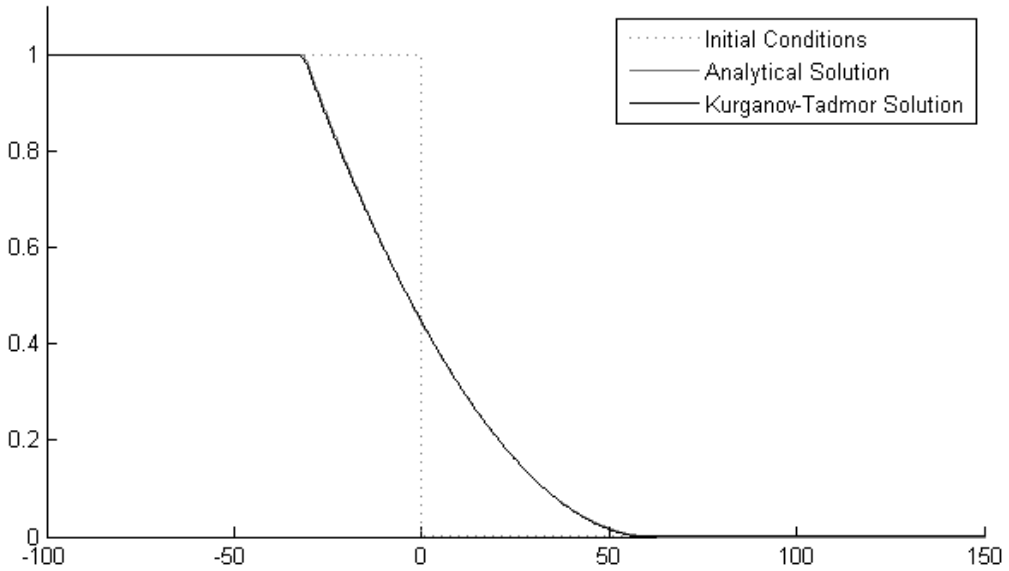


Figure 6.2: KT approximation for dam break, $t = 10, \Delta x = 0.1$

the shape and propagation of the wave, with the KT method giving one almost indistinguishable at this scale from the analytical solution. Figures 6.3 and 6.4 show the errors in wave height for both methods on the same scale.

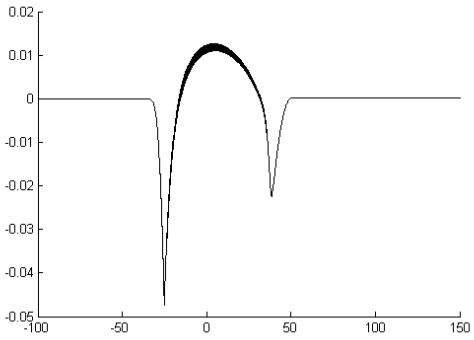


Figure 6.3: LxF errors $t = 10$, $\Delta x = 0.1$

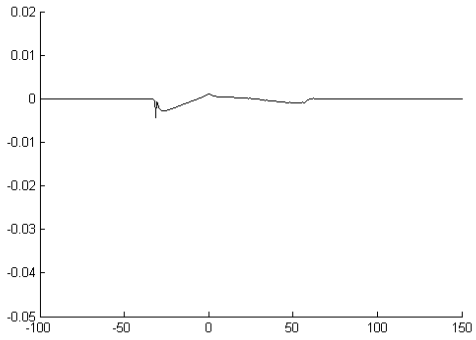


Figure 6.4: KT errors $t = 10$, $\Delta x = 0.1$

There are two main attributes to the errors curve for LxF. The first is the sharp downward peaks which show large relative under-estimates (of between 1% and 3%) of the actual solution. The second is the over-estimate in the centre of the curve, which becomes more notable where the wave should have the tightest radius. Both of these issues arise from the smoothing effect that the LxF method causes. Too much dissipation means that the sharp fidelity at these points is lost.

The KT method has errors an order of magnitude smaller, although they do share some of the same traits just described within the LxF errors. For instance, the greatest errors still occur at the top of the wave and in the centre. The motivation behind not employing higher order time discretization for the KT method is to allow a comparison of the two methods on their merits for numerical differentiation over space. It is clear that the KT method is far superior for this solution on that alone. Better utilising its semi-discrete attributes, it would only be even better.

For instance, even at a low resolution, the KT method can give more accurate results than a high resolution LxF approximation; see figure 6.5. The higher computation time needed for the KT method can be combated by increasing Δx like this, giving comparable computation times to those using LxF.

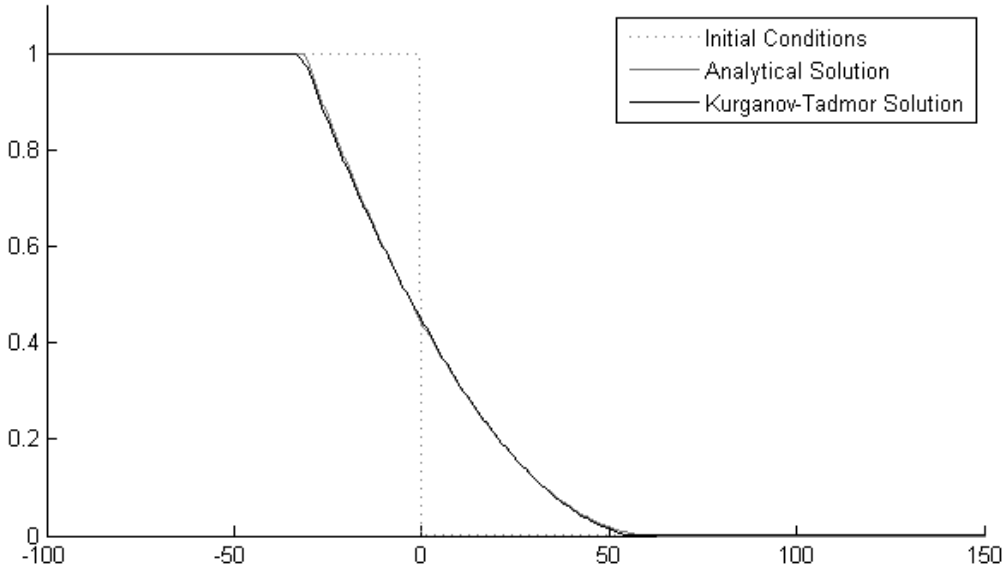


Figure 6.5: KT approximation for dam break, $t = 10, \Delta x = 0.5$

6.2 Errors Over Time

Over time, the transitions between the sections of the analytical solution 3.2 become less abrupt as the gradient of the wave section becomes less steep. Also, as the wave spreads out, its curvature lessens, and so it actually more closely matches the smoothed approximation. This can be seen in figures 6.6 and 6.7 which display the sum of the squared errors over the period $0 \leq t \leq 8$ for both methods. At any given point in this time though, the spacial errors reflect those seen above. The maximum errors contribute largely at the start, but as time goes on the errors both reduce in the extremities and overall. This analysis over time also further emphasises the strong convergence of the KT method over the LxF method, as these two error graphs are plotted on the same scale.

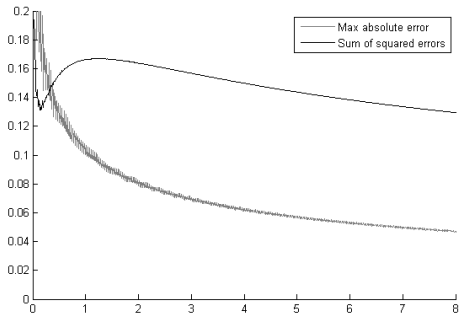


Figure 6.6: LxF errors over time

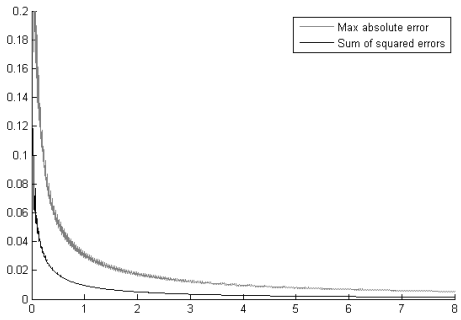


Figure 6.7: KT errors over time

7 Tsunamis and SWEs

7.1 Relevance of Application

Tsunamis are massive water waves, often generated by under-sea earthquakes. Despite having a large profile by comparison to regular oceanic waves, the depth of the ocean pales in comparison to the tsunami wavelengths which can reach 10s or 100s of km [11]. Because this fits with our notion of shallow water waves, we can apply one of the previously discussed methods to some given initial conditions, and see how a tsunami may propagate toward a beach.

7.2 SWEs for an Uneven Bed

Of course, as the bed topography will be non-horizontal for an ocean wave approaching a beach, we must use equations 2.16 and 2.17 which do not assume a flat bed. By substituting h for $\xi + b$ and simplifying in much the same way, we get equation 7.1.

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S \tag{7.1}$$

Where U and F are defined as in 7.2; a familiar form but with ξ replacing h, and now $q = u\xi$. Unlike in 2.22 though, the source term S (defined in 7.3) is preserved, accounting

for the bed being uneven.

$$U = \begin{bmatrix} \xi \\ q \end{bmatrix}, \; F(U) = \begin{bmatrix} q \\ \frac{q^2}{\xi} + \frac{1}{2} g \xi^2 \end{bmatrix} \tag{7.2}$$

$$S = \left[0, \; -gh \frac{\partial b}{\partial x} \right]^T \tag{7.3}$$

7.3 Numerical Solutions

To solve this, a method put forward as standard in [17] is to solve the homogeneous case first and then the source term equation 7.4. When used in a numerical solver, they can be solved independently and then combined at each time step. However, it is noted that this can lead to undesirable results around steady states and quickly lends itself to complex solutions when the source term is in the form of S.

$$\frac{\partial U}{\partial t} = S \tag{7.4}$$

A more simple approach is to use a backwards difference approximation for the bed gradient, and simply adding the resulting term on to our numerical approximation at each time step [18]. This is a valid approach for the LxF method, but can become more complicated for accordingly more complicated methods such as KT. Implementing it into the LxF equation 4.8 yields equation 7.5, with $\tilde{S}$ defined in 7.6.

$$U_i^{t+\Delta t} = \frac{1}{2}(U_{i+1}^t + U_{i-1}^t) - \frac{\Delta t}{2\Delta x}(F(U_{i+1}^t) - F(U_{i-1}^t)) + \Delta t \tilde{S} \tag{7.5}$$

$$\tilde{S} = \left[0, \; -g h_i^t \left(\frac{B_i^t - B_{i-1}^t}{\Delta x} \right) \right]^T \tag{7.6}$$

7.4 Comments on Accuracy

As this is only implemented as part of the LxF method here, accuracy is already dependent on small Δx. If the method is being run with fine enough discretization anyway, then the first order correction term for the uneven bed should still aid in giving a reasonable approximation to the behaviour of the wave. As this situation is naturally lends itself to

numerical solutions, a clear comparison with true solutions is difficult. However, a short analysis can be made of how its attributes reflect those which would be expected of a true tsunami.

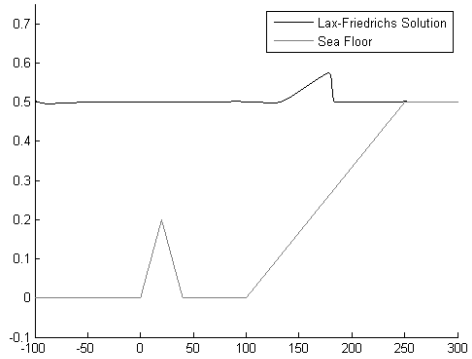


Figure 7.1: $t = 75$, $\Delta x = 0.1$

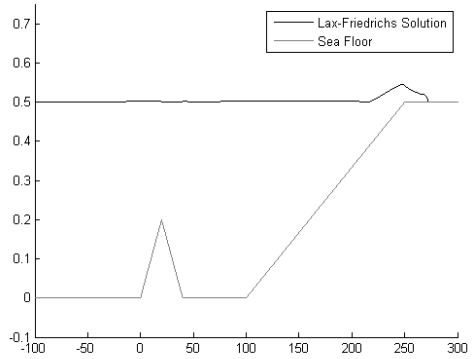


Figure 7.2: $t = 130$, $\Delta x = 0.1$

Figures 7.1 and 7.2 show the propagation of a wave caused by a sea-floor disturbance. The sharp peak in the sea floor is representative (albeit an extreme model) of a tectonic plate shift, or earthquake. The water directly above it was given an appropriate initial displacement. This caused a wave to travel in both directions, as expected from a symmetrical disturbance. The wave also breaks over the point of contact with land, dissipating its energy in a low front over the beach. Pugh [19] writes that experimental data and some theoretical calculations suggest that large oceanic waves normally break when the water depth is around 2.6 times the wave amplitude. This would suggest that the wave front reaching the actual shore line in the LxF model is somewhat unrealistic, although waves can break in a chaotic manor, from far out at sea to very close to land. The continuation of the wave over the land in the model at least shows some preservation of momentum and volume.

The wave in this model maintains a reasonably steady height. Different initial conditions can yield different outcomes in this area. As discussed by Klettner et al. [20], some tsunamis push a lot of water ahead of them making the tide appear to come in quickly. Others draw water towards them, gaining more energy and pulling water away

from the shoreline. It may be because of the dissipative nature of the LxF model that the affects of the wave do not reach the shore with much conviction. On approach to the beach, a common expectation is that shoaling effects will occur [21]. This involves the wave gaining height, and its wavelength shortening. Neither of these effects are visible in the LxF model.

8 Conclusions

The results from this investigation were varied in their success. A good understanding was gained about the origins of the SWEs, as well as what simplifications can be made under certain conditions.

The LxF method was implemented for use with the dam break problem and, later, further extended for use in a rudimentary analysis of a tsunami situation. The former of these uses showed great promise for such a simple method. It was quick to both write into code, and run repeatedly for different results. For observing the outcome of the dam break, it was a lightweight, flexible method which suited the purpose well. It has long been established as the default system for such problems.

KT was a harder method to grasp. Its construction is such that, once established, it is a fairly simple method defined over only a few lines. However, over analysis of the derivation can lead to over complication when coding or analysing. An example of the code written can be seen in A. Once understood, the method is a joy to use as it gives such brilliant accuracy. It avoids many obstacles which could make it less versatile, and gives satisfying results for many initial conditions and situations. Analysed here in comparison to the LxF method, it shows a great leap forward for a central scheme.

Tsunami modelling, whilst appropriate for SWEs, was not explored thoroughly. There is a lot of variation in expectations from source material, and the use of the dissipative LxF scheme meant that some understood effects were not seen. If there was an extension to this report, tsunamis could be explored in much greater depth. Varied solvers and bed topographies (under sea shelves, different beach profiles) may give a better understanding as to the extent which SWEs can be used in this situation.

Understanding what SWEs are, how they can be applied and developing numerical analysis skills have all been key outcomes of writing this report. It is a broad field, with lots of scope for personal and global development.

References

[1] J. Billingham and A. C. King *Wave Motion* Cambridge University Press (2000)

[2] A. Ritter *Die Fortpflanzung der Wasserwellen* Verein Deutsch. Ing. Vol. 36, Issue 33, pp. 947-954 (1892)

[3] R. J. LeVeque *Numerical Methods for Conservation Laws* Birkhuser (1992)

[4] G.S. Jiang, D. Levy, C.T. Lin, S. Osher, and E. Tadmor *High-Resolution Nonoscillatory Central Schemes With Nonstaggered Grids For Hyperbolic Conservation Laws* SIAM Journal of Numerical Analysis, Vol. 35 Issue 6, pp. 2147-2168 (1998)

[5] C.B. Vreugdenhil *Numerical Methods for Shallow-Water Flow* Springer Science & Business Media (1994)

[6] O. Delestre, C. Lucas, P.A. Ksinant, F. Darboux, C. Laguerre, et al. *SWASHES: a library of Shallow Water Analytical Solutions for Hydraulic and Environmental Studies* HAL: ID hal-00628246 (2011)

[7] Olver P.J. (2014) *Introduction to Partial Differential Equations* Springer

[8] Roache P.J. (1972) *Computational Fluid Dynamics* Hermosa

[9] Janosi I.M., Jan D., Szabo K.G., and Tel T. (2004) *Turbulent drag reduction in dam-break flows* Experiments in Fluids, Issue 37, pp. 219-229

[10] Hwung H.H., Yang R.Y., Tie Y.C., Hsu W.Y., Kuo P.C., and ShuganI.V. (2011) *Experimental and Theoretical Study on Flood Bore Propagation and Mushroom Jet Generation of Dam Break Flow* Visualization of Mechanical Processes: An International Online Journal, Vol. 1, Issue 3

[11] Eze C.L., Uko D.E., Gobo A.E.T, Sigalo F.B., and Israel-Cookey C. (2009) *Mathematical Evaluation of Tsunami Propagation* Research Journal of Applied Sciences, Vol. 4, Issue 6, pp. 213-216

[12] Kurganov A. and Tadmor E. (2000) *New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations* Journal of Computational Physics, Vol. 160 Issue 1, pp. 241-282

[13] Nessyahu H. and Tadmor E. (1990) *Non-oscillatory Central Differencing for Hyperbolic Conservation Laws* Journal of Computational Physics, Vol. 87, Issue 2, pp. 408-463

[14] Cavalli F., Naldi G., Puppo G., and Semplice M. (2006) *A comparison between relaxation and Kurganov-Tadmor schemes* arXiv: ID arXiv:math/0612236v1

[15] Gottlieb S., Shu C.W., and Tadmor E. (2001) *Strong Stability Preserving High-Order Time Discretization Methods* SIAM Review, Vol. 43, Issue 1, pp. 89-112

[16] Kurganov A. and Levy D. (2002) *Central-Upwind Schemes for the Saint-Venant System* Mathematical Modelling and Numerical Analysis, Vol. 36, pp. 397-425

[17] George D.L. (2004) *Numerical Approximation of the Nonlinear Shallow Water Equations with Topography and Dry Beds: A Godunov-Type Scheme* Unpublished manuscript, University of Washington

[18] Hudson J. (2001) *Numerical Techniques for the Shallow Water Equations* Unpublished manuscript, University of Reading

[19] Pugh D.T. (1996) *Tides, Surges and Mean Sea-Level* John Wiley & Sons

[20] Klettner C., Fernando H., Balasubramanian S., Voropayev S., Hunt J., and Eames I. (2012) *Draw-down and Run-up of Tsunami Waves on Sloping Beaches* Engineering and Computational Mechanics Vol. 165, Issue EM2, pp. 119129

[21] Truong H.V.P. (2012) *Wavee-Propagation Velocity, Tsunami Speed, Amplitudes, Dynamic Water-Attenuation Factors* Presented at World Conference on Earthquake Engineering, Lisbon, Portugal

Appendix A Matlab code: KT Dam Break

This code shows how the dam break was analysed using Matlab and the KT method. A similar code was written using the more simple LxF solver, as seen in B, but the structure was the same.

```
function KT_SWE(L,N,tend,dtout)
    x = linspace(-L,L*1.5,N+1);
    g = 9.81;
    dx = x(2)-x(1);
    smallH = 1e-8;
    defaultH = 1.0;
    u_kt = [0.*x + smallH; 0.*x];
    u_kt(1,x<0) = defaultH;
    u_init = u_kt;
    u_anal = u_kt;
    t = 0; tout = 0;
    umx = zeros(2,N);
    a  = zeros(2,N-1);
    c0 = sqrt(g*defaultH);
    I = []; % Errors at each time step
    while t<tend
        % Kurganvov & Tadmor
        dt = dx/(4*max(abs(df(u_kt))));
        ux = [diff(u_kt(1,:))/dx; diff(u_kt(2,:))/dx];
        umx(:,2:N-1) = minmod(ux(:,2:N-1),ux(:,1:N-2));
        uplus = u_kt(:,2:N)-0.5*dx*umx(:,2:N);
        uminus = u_kt(:,1:N-1)+0.5*dx*umx(:,1:N-1);
        for k = 1:N-1
            a(:,k) = max(abs(df(uplus(:,k))),abs(df(uminus(:,k))));
        end
        H = 0.5*(f(uplus)+f(uminus))-0.5*a.*(uplus-uminus);
        u_kt(:,2:N-1) = u_kt(:,2:N-1)-(dt/dx)*(H(:,2:N-1)-H(:,1:N-2));

        % Analytic
        H_anal = u_anal(1,:);
        Q_anal = u_anal(2,:);
        H_anal(x<=-c0*t) = defaultH;
        Q_anal(x<=-c0*t) = 0;
        H_anal(x>-c0*t & x<= 2*c0*t) = ...
            (defaultH/9).*(2-(x(x>-c0*t & x<= 2*c0*t)./(c0*t))).^2;
        Q_anal(x>-c0*t & x<= 2*c0*t) = ...
            (2/3).*(c0+(x(x>-c0*t & x<= 2*c0*t)./t)).*H_anal(x>-c0*t & x<= 2*c0*t);
        H_anal(x>2*c0*t) = 0;
        Q_anal(x>2*c0*t) = 0;
        u_anal(1,:) = H_anal;
        u_anal(2,:) = Q_anal;

        % Errors
        if (t == 0)
            I = [sum((u_kt(1,:)-u_anal(1,:)).^2); ...
                max(abs(u_kt(1,:)-u_anal(1,:))); t];
        else
            I = [I(1,:), sum((u_kt(1,:)-u_anal(1,:)).^2); I(2,:), ...
                max(abs(u_kt(1,:)-u_anal(1,:))); I(3,:), t];
        end
```

```
        % Plotting
        t = t+dt;
        tout = tout+dt;
        if tout>dtout
            clf;
            subplot(2,1,1)
            hold on
            plot(x,u_init(1,:),'Color',[0.5, 0.5, 0.5],'LineStyle',':')
            plot(x,u_anal(1,:),'Color',[0.5, 0.5, 0.5])
            plot(x,u_kt(1,:),'k')
            legend('Initial Conditions','Analytical Solution','Kurganov-Tadmor Solution')
            hold off
            ylim([0,1.1]);
            subplot(2,1,2)
            hold on;
            %plot(x,u_kt(1,:)-u_anal(1,:),'k') %Spacial Errors
            plot(I(3,:),I(2,:), 'Color',[0.5, 0.5, 0.5]); %Max abs errors
            plot(I(3,:),I(1,:),'k');        % Sum errors squared
            hold off;
            ylim([0,0.2]);
            xlim([0,8]);
            legend('Max absolute error','Sum of squared errors')
            drawnow
            tout = tout-dtout;
            pause(0.2)
        end
    end

    function m = minmod(a,b)
        m = 0.5*(sign(a)+sign(b)).*min(abs(a), abs(b));
    end
    function f = f(u)
        %f = 0.5*u.^2; %Bergers'
        %f = u;         %Advection
        f = [u(2,:) ; (u(2,:).^2)./u(1,:) + 0.5*g*u(1,:).^2]; %SWE
    end

    function df = df(u)
        %df = u;        %Bergers'
        %df = ones(size(u(1,:))); %Advection
        df = u(2,:)./u(1,:) + sqrt(g*u(1,:)); %SWE
    end
end
```

Appendix B Matlab code: LxF Tsunami Model

This code uses a variable bed height, B, and an implementation of the Lax-Friedrichs method to model tsunami waves in a rudimentary way.

```
clear; clc;
%%%% SETUP %%%%
g = 9.81;       % Gravity
```

```
dt = 0.0;        % Increment for re-evaluation of equations
dtShow = 0.0;    % Counting toward showOn
showOn = 5.0;    % Time interval at which plotting occurs
simtime = 150.0;  % Real-world time for which simulation runs
t = 0.0;         % Current real-world time
pauseTime = 0.1;% length of time paused for between plots
dx = 0.1;        % Increment for x axis
LB = -100;       % Left Bound for simulation
RB = 300;        % Right Bound for simulation
colWidth = 30;  % Half width of inital column of water
defaultH = 0.5;     % Default height for dammed section
defaultH0 = 1e-8;  % Default height for low level water
steps = (RB - LB + 1)/dx;    % Num of steps in sim range
x = linspace(LB,RB,steps);  % x is array of discrete steps
q = 0.*x;        % Initialise 0s of same length as x. q = uh
u = q;           % Ditto
h = q;           % Ditto, then assign heights:
B = q;

X_bump = 40;
Bump_h = 0.2;
m = 2*Bump_h/X_bump;
X_beach= 100;
X_land = 250;
B(x>0 & x<=X_bump/2) = m*x(x>0 & x<=X_bump/2);
B(x>X_bump/2 & x<=X_bump) = (-m)*x(x>X_bump/2 & x<=X_bump) + m*X_bump;
a = 0.999;
B(x>X_beach) = (a*defaultH/(X_land-X_beach))*(x(x>X_beach)-X_beach);
B(x>X_land) = defaultH*a;

h(:) = defaultH - B(:);
h(x>0 & x<X_bump) = defaultH;
h0 = defaultH;% Create h0 for plotting a comparison during sim

%%%% SIMULATION %%%%
while (t < simtime)
    t = t + dt; % Increment real-world time
    dtShow = dtShow + dt;

    U = [h; q];
    R = [zeros(1,steps-2); g*U(1,2:steps-1).*(B(2:steps-1)-B(1:steps-2))];
    U(:,2:steps-1) = 0.5*(U(:,1:steps-2)+U(:,3:steps)) ...
        - (dt/(2*dx))*(f(U(:,3:steps))-f(U(:,1:steps-2))) - (dt/dx)*R;
    h = U(1,:);
    q = U(2,:);

    u = q./h;               % as q = uh by definition
    c = max(max(abs(u + sqrt(g.*h))),max(abs(u - sqrt(g.*h))));
    dt = 0.45*(dx / c);     % To satisfy dt < dx / c, with c as above.

    % Plot the resulting vectors
    if (dtShow >= showOn)
        pause(pauseTime);   % Add delay between plots
        clf;                % Clear figure from last plot
        hold on;
        plot(x,h+B,'k')     % Plot wave form
        plot(x,B,'Color',[0.5,0.5,0.5])
        hold off;
```

```
        legend('Lax-Friedrichs Solution','Sea Floor')
        ylim([-0.1,1.5*defaultH]);
        title(t)
        % Reset dtShow to start counting toward this loop
        dtShow = 0;
        drawnow()
    end
end
%%%% END OF MAIN PROGRAM %%%%
```