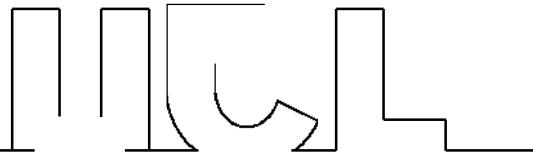


Accurate methods for computing rotation-dominated flows

Martin Büttner, University College London
MSci Theoretical Physics

Supervisor: Prof. Edward Johnson

March 30, 2015



Submission of coursework for Physics and Astronomy course PHASM201/PHAS3400

Please sign, date and return this form with your coursework by the **specified deadline** to the Departmental Office.

DECLARATION OF OWNERSHIP

I confirm that I have read and understood the guidelines on plagiarism, that I understand the meaning of plagiarism and that I may be penalised for submitting work that has been plagiarised.

I confirm that all work will also be submitted electronically and that this can be checked using the JISC detection service, Turnitin®.

I understand that the work cannot be assessed unless both hard copy and electronic versions of the work are handed in.

I declare that all material presented in the accompanying work is entirely my own work except where explicitly and individually indicated and that all sources used in its preparation and all quotations are clearly cited.

Should this statement prove to be untrue, I recognise the right of the Board of Examiners to recommend what action should be taken in line with UCL's regulations.

Signed _____

PrintName _____

Dated _____

Title	Date Received	Examiner	Examiner's Signature	MARK

Abstract

Atmospheric and oceanic flows are often modelled using the Shallow Water Equations (SWEs) with the addition of source terms to account for varying topography and the rotation of the Earth. Naive numerical methods fail to preserve steady states in which these additional effects are large but exactly cancelled by pressure gradients in the system. This is a significant shortcoming, as large-scale real-world flows are usually very close to such a balance at all times. This report investigates two existing methods to obtain so-called *well-balanced* solvers. The author was not able to use one of the methods as presented originally, but was able to derive an alternative formulation which did work. New solvers for the SWEs under rotation have been developed based on these methods and implemented with the software package *Clawpack*. The new solvers were shown to be able to preserve these steady states at least to machine precision.

Contents

1	Introduction	2
1.1	Literature Overview	2
1.2	Report Outline	4
2	Theory	5
2.1	The Shallow Water Equations	5
2.2	Hyperbolic Systems	8
2.3	Roe's Approximate Riemann Solver	12
2.4	Balanced Method: LeVeque	15
2.5	Balanced Method: Rogers et al.	19
3	Implementation	25
3.1	Clawpack	25
3.2	Test Harness	26
3.3	Notes on LeVeque Solver	29
4	Results	33
4.1	Still Water	33
4.2	Wave through Still Water	35
4.3	Geostrophic Equilibrium	35
4.4	Wave through Geostrophic Equilibrium	40
4.5	Uniform Flow	40
4.6	Execution Time	44
5	Conclusion	45
5.1	LeVeque's Method	45
5.2	Rogers's Method	45
5.3	Future Work	46
A	Notes on method by Rogers et al.	47
	Bibliography	49

Chapter 1

Introduction

The shallow water equations (SWEs) are a simplified but very effective model for incompressible fluid flow. Despite the name, this model has been particularly successful in modelling large-scale geophysical flows of atmospheric and oceanic currents — here, “shallow” merely means that the fluid depth is small compared to the horizontal length scales of the system. Considering flows on the scale of the Earth, which can easily extend across several thousand kilometres, the roughly 4 kilometre deep ocean is indeed comparably shallow.

In the context of these applications, two effects are particularly important to model: the Coriolis force, due to the rotation of the Earth, as well as varying topography¹. This report is concerned with simulating such models using *finite volume methods*. These will usually include the above effects as source terms. Computational difficulties arise in steady flow when these very large forces balance pressure gradients exactly. This is referred to as a geostrophic balance, and large-scale real-world currents are close to this balance at all times. Therefore, the interest of this report lies in numerical methods which work around these computational problems such that they preserve geostrophic flows exactly and can reliably compute small perturbations of these steady states.

Two existing methods to account for such a balance in the presence of bathymetry are reviewed and extended to include the Coriolis terms as well. Both new methods are then evaluated on various systems to allow a comparison of their individual merits.

1.1 Literature Overview

The SWEs are a hyperbolic system of conservation laws. A lot of effort has been put into studying and solving these systems of equations numerically. Several textbooks about the numerical methods considered in this report exist, including [1, 2, 3], as well as [4] which focuses on the SWEs in particular. In addition, there is a textbook on the rotating SWEs by Zeitlin [5], Chapter 4 of which focuses on numerical methods.

¹Undersea topography is usually referred to as *bathymetry*. This term will be used predominantly throughout this report.

Such hyperbolic systems can be solved using finite volume methods in which the domain is divided up into a (not necessarily regular) grid of control volumes, and the conserved quantities are discretised by assuming that they are constant across each such volume. Note that “volume” is being used in a generalised sense here — for a two-dimensional system like the SWEs, the grid cells are actually areas. The most popular of these methods is due to Godunov [6]² and is simply known as Godunov’s method — in fact, today a whole family of Godunov methods has been developed based on the concepts derived in this original paper.

A fairly recent review of Godunov-type methods was conducted by Toro and Garcia-Navarro [7] and an older review can be found in [8].

The piecewise-constant discretisation reduces the problem to a number of step functions at the cell boundaries. The initial-value problem consisting of a single step between two constant values is known as a Riemann problem (see [2], Section 2.2.2, [3], Section 3.8, or [7], Section 2.1) and can be solved exactly for many systems. The discretised SWEs can be solved by solving each of the Riemann problems at the cell boundaries. The other insight of Godunov’s method is that each hyperbolic system has a number of characteristic waves which only propagate in certain directions at finite speeds, which allows to simplify the computation by only looking at waves that can propagate *into* each cell. This technique is known as *upwinding* (see e.g. [3], p. 72).

While it is possible to solve the Riemann problem exactly for many systems (including the SWEs), this dominates the computations required to solve each time step. Therefore, approximate solvers have been developed, the most popular one being due to Roe [9].

However, these methods generally have been developed for homogeneous systems. A simple approach to account for source terms like bathymetry and the Coriolis force is to compute these independently of the homogeneous system in a separate step. This leads to problems in steady or quasi-steady scenarios, where the flux terms arising from the homogeneous system and the source terms are balanced at all times. According to Toro and Garcia-Navarro [7], the first authors to recognise this were Glimm et al. [10]. Preserving such a balance requires that a time step in the homogeneous system is cancelled exactly by the corresponding time step for the source terms. Since these terms can in principle be very large, even for balanced systems, due to different methods being employed and cancelling terms being computed in separate steps, this is practically impossible. Hence, equilibria cannot be modelled accurately, even when the system is as simple as a still lake. The numerical errors create spurious oscillations which may even be amplified in future time steps. Furthermore, small perturbations away from equilibrium would be completely dominated by said numerical errors. These problems are particularly relevant for large-scale geophysical flows, which are usually very close to geostrophic balance — an instance of balanced flux and source terms — at all times (see e.g. [11], Section 7.6).

²The author could not obtain an English translation of this Russian paper, but the method developed by Godunov has been extensively reiterated in many papers and textbooks. Therefore, the explanation of the method in this report is based on those secondary sources.

Therefore, a lot of research has been conducted over the past two decades to develop so-called *well-balanced* methods which are able to preserve these equilibria exactly. To the best of the author’s knowledge, Greenberg and LeRoux [12] were the first to use the term “well-balanced”.

Subsequently, dozens of well-balanced methods have been developed, including [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. The most recent articles the author could find are by Zhang et al. [25] and Chertock et al. [26], the latter being particularly interesting here, as their assumptions align with those made in this report. Furthermore, Section 4.4 of [5] presents a long list of other well-balanced methods applicable to the rotating SWEs and refers to the method discussed in [21] and related works as “the most classical [well-balanced] method”. With [27], there is also a textbook focusing primarily on these methods.

Many of these methods deal with very specific models which are beyond the scope of this report. In particular, some address the use of geometric source terms to model the equations on an irregular grid. Especially more recent papers have largely focused on methods which are capable modelling dry states. Hence, two methods were chosen to be investigated in detail in this report: the method presented by LeVeque [13], which balances the terms by introducing additional Riemann problems, as well as the method due to Rogers et al. [20], which employs a change of variables.

Other methods were considered for closer investigation, in particular [15] and [26]. However, these essentially develop unsplit balanced methods, which require a considerably different computational framework. Hence, the scope of this report is limited to the above two methods, both of which can be implemented with relatively little effort by adapting existing Godunov-type methods in wave-propagation form.

1.2 Report Outline

The remainder of this report is structured as follows. Chapter 2 constitutes the main part of this report and introduces the theory behind these numerical methods and extends them to include the Coriolis terms. Subsequently, Chapter 3 describes the test framework which was set up to evaluate these methods as well as implementation details of the methods themselves. Chapter 4 shows the results obtained from these implementations. Lastly, Chapter 5 draws some conclusions and suggests ways in which further research could improve on the work presented here.

Chapter 2

Theory

This chapter introduces the system of equations used throughout this report and recapitulates the theory of hyperbolic conservation laws and Godunov method's. It then proceeds to introduce the balanced methods investigated and extends them to the relevant systems.

2.1 The Shallow Water Equations

The two-dimensional SWEs are a system of three partial differential equations in three conserved quantities: the water depth, h , and the two Cartesian components of the momentum, hu and hv (where u and v are the components of the velocity). The water depth can be viewed as the difference between the water surface, h_s , and the bathymetry (or bed elevation), B , i.e. $h = h_s(x, t) - B(x)$. Throughout this report, $h_s > B$ will be assumed for all x and t . The PDEs can be written as:

$$h_t + (hu)_x + (hv)_y = 0 \quad (2.1a)$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}gh^2 \right)_x + (huv)_y = -ghB_x + fhv \quad (2.1b)$$

$$(hv)_t + (huv)_x + \left(hv^2 + \frac{1}{2}gh^2 \right)_y = -ghB_y - fhu, \quad (2.1c)$$

where subscripts denote partial differentiation, g is acceleration due to gravity and f is the Coriolis coefficient. These can be obtained from the Navier–Stokes equations by assuming that the depth of the water is small compared to some significant horizontal length-scale and by depth-averaging the flow variables. For a full derivation see [28].

Numerically, two-dimensional systems can be solved to a good approximation by applying a dimensional splitting. This refers to solving the equations on a grid along slices of constant y first, and then solving them along slices of constant x , for each time step. During each of those stages, variation along the orthogonal direction is completely ignored. This amounts to setting $\partial/\partial y = 0$ for solving the equations only in the x -direction (and vice versa). Hence, two-dimensional methods can easily be built on top of one-dimensional

solvers. As LeVeque notes in [13], full second-order order accuracy can be achieved by adding so-called transverse solvers. However, these do not affect the approach to the balancing of source terms. So without loss of generality, this report is only concerned with these x -split equations, which reduce the SWEs to a system in only one spatial dimension:

$$h_t + (hu)_x = 0 \quad (2.2a)$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}gh^2 \right)_x = -ghB_x + fhv \quad (2.2b)$$

$$(hv)_t + (huv)_x = -fhu \quad (2.2c)$$

It is common practice in fluid dynamics to use dimensionless variables in order to reduce the system to a minimal amount of free parameters — all additional parameters, like individual length scales, then merely give similarity solutions. To do so, we introduce a typical horizontal length scale L (the width of the domain), a vertical length scale H (a typical depth of the water), a wave speed c and a time scale T . A reasonable choice for T is L/c , which is the time taken for a wave to travel across the domain. We can also define the wave speed as $c = \sqrt{gH}$, which is the speed of linear gravity waves at depth H . Using these parameters, we can rewrite the variables in terms of dimensionless quantities:

$$t = T\bar{t}, \quad x = L\bar{x}, \quad y = L\bar{y}, \quad h = H\bar{h}, \quad u = c\bar{u}, \quad v = c\bar{v}, \quad B = H\bar{B} \quad (2.3)$$

These can be substituted into each of the SWEs. Making use of $T = L/c$, the h -equation gives:

$$h_t + (hu)_x = 0 \quad (2.4)$$

$$\Rightarrow \frac{H}{T}\bar{h}_{\bar{t}} + \frac{Hc}{L}(\bar{h}\bar{u})_{\bar{x}} = 0 \quad (2.5)$$

$$\Rightarrow \bar{h}_{\bar{t}} + (\bar{h}\bar{u})_{\bar{x}} = 0, \quad (2.6)$$

so it remains unchanged. For the hu -equation, using both $T = L/c$ and $c^2 = gH$:

$$(hu)_t + \left(hu^2 + \frac{1}{2}gh^2 \right)_x = -ghB_x + fhv \quad (2.7)$$

$$\Rightarrow \frac{Hc}{T}(\bar{h}\bar{u})_{\bar{t}} + \frac{1}{L} \left(Hc^2\bar{h}\bar{u}^2 + \frac{1}{2}gH^2\bar{h}^2 \right)_{\bar{x}} = -\frac{gH^2}{L}\bar{h}\bar{B}_{\bar{x}} + fHc\bar{h}\bar{v} \quad (2.8)$$

$$\Rightarrow (\bar{h}\bar{u})_{\bar{t}} + \left(\bar{h}\bar{u}^2 + \frac{1}{2}\bar{h}^2 \right)_{\bar{x}} = -\bar{h}\bar{B}_{\bar{x}} + \frac{fL}{c}\bar{h}\bar{v}. \quad (2.9)$$

This equation depends on a single parameter $K \equiv fL/c$, which measures the strength of the rotation. Similarly, for the hv -equation:

$$(hv)_t + (huv)_x = -fhu \quad (2.10)$$

$$\Rightarrow \frac{Hc}{T}(\bar{h}\bar{v})_{\bar{t}} + \frac{Hc^2}{L}(\bar{h}\bar{u}\bar{v})_{\bar{x}} = -fH\bar{c}\bar{h}\bar{u} \quad (2.11)$$

$$\Rightarrow (\bar{h}\bar{v})_{\bar{t}} + (\bar{h}\bar{u}\bar{v})_{\bar{x}} = -\frac{fL}{c}\bar{h}\bar{u} = -K\bar{h}\bar{u} \quad (2.12)$$

From here on, the bars will be omitted, as the dimensionless quantities will be used throughout the report. To obtain dimensional results from the dimensionless quantities, Eqs. 2.3 can be used. In summary, the dimensionless SWEs are

$$h_t + (hu)_x = 0 \quad (2.13a)$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}h^2 \right)_x = -hB_x + Khv \quad (2.13b)$$

$$(hv)_t + (huv)_x = -Khu. \quad (2.13c)$$

The development of well-balanced methods is motivated by the desire to model states which are either in equilibrium or are small perturbations about equilibrium. Hence, it is worth examining which equilibrium states exist for Eqs. 2.13. By definition, all time derivatives of an equilibrium state are zero, such that the equations reduce to

$$(hu)_x = 0 \quad (2.14a)$$

$$\left(hu^2 + \frac{1}{2}h^2 \right)_x = -hB_x + Khv \quad (2.14b)$$

$$(huv)_x = -Khu. \quad (2.14c)$$

The simplest equilibrium, which exists regardless of the value of K is the so-called *still water* or *still lake* equilibrium, defined by $u = v = 0$ and h_s being a constant. For simplicity, we will assume that $h_s = 1$, such that the dimensional water level is H . In this case, $h = 1 - B(x)$ and hence $h_x = -B_x$, which fulfills Eq. 2.14b. All other terms in the equations are zero. This is the equilibrium addressed in most papers, including [13] and [20] which are the focus of this report.

For non-zero K , there exists a less trivial, and geophysically much more relevant equilibrium state. Given the right velocity profile, any arbitrary (continuously differentiable) water surface profile can be maintained. This is called a geostrophic equilibrium, and most large-scale flows on Earth are close to such an equilibrium at all times. The condition on v for a given profile can easily be derived from Eqs. 2.14. We assume that $u = 0$ and $h = h_s(x) - B(x)$. Then only the x -momentum equation is non-zero and gives:

$$\left(\frac{1}{2}h^2\right)_x = -hB_x + Khv \quad (2.15)$$

$$\Rightarrow hh_x = -hB_x + Khv \quad (2.16)$$

$$\Rightarrow h(h_s - B)_x = -hB_x + Khv \quad (2.17)$$

$$\Rightarrow v = \frac{(h_s)_x}{K} \quad (2.18)$$

There are other steady states, in particular those which involve non-zero u , but these depend on the given bathymetry and are beyond the scope of this report. See [29] for an analysis of the phase space of steady rotating flow over a ridge.

Nevertheless, we are interested in setting up systems with a uniform background flow, in order to test the numerical methods for states which are not the known equilibria. In these cases, the Coriolis term requires some practical considerations. Let $B = 0$ for now and consider uniform flow with $u = U$, $v = 0$ and $h_s = 1$ at $t = 0$. In this case, the momentum equations of 2.13 become

$$u_t = Kv \quad (2.19)$$

$$v_t = -Ku \quad (2.20)$$

The solution to this system is circular motion with constant speed U . Hence, even without complicated bathymetry or an initial surface profile, this system cannot maintain uniform flow. This can be rectified by introducing a transversal pressure gradient, $+KhU$ which balances the Coriolis force due to the background flow. From a practical point of view, this is equivalent to having the water at rest (in the rotating frame, i.e. in solid body rotation), towing an obstacle through the water and changing into the rest frame of the obstacle. The full equations for uniform background flow are thus

$$h_t + (hu)_x = 0 \quad (2.21a)$$

$$(hu)_t + \left(hu^2 + \frac{1}{2}h^2\right)_x = -hB_x + Khv \quad (2.21b)$$

$$(hv)_t + (huv)_x = KhU - Khu \quad (2.21c)$$

2.2 Hyperbolic Systems

This section reviews the relevant theory of hyperbolic conservation laws and Godunov methods.

Conservation laws are systems of partial differential equations (PDEs) which, in one dimension, can be written in the form:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{s}(\mathbf{q}, x). \quad (2.22)$$

Here, \mathbf{q} is a vector of density functions of conserved quantities, \mathbf{f} is a *flux vector*, while \mathbf{s} stands for a number of *source terms*. For the components q_i to be conserved means that the integral $\int_{-\infty}^{\infty} (q_i - s_i) dx$ is independent of time. The flux terms describe how the quantities \mathbf{q} are transported through the domain. Apart from actual sources or sinks the source terms \mathbf{s} may be used to model a variety of physical and geometric effects.

For the purpose of this project only the above bathymetry and Coriolis source terms will be considered, but more advanced treatment of shallow water systems might include further terms to model other physical effects. Examples include bed friction, surface stresses and eddy viscosity. If the SWEs are discretised on an irregular grid, geometric source terms might also be used which represent properties of the grid cells.

Such a system is called *hyperbolic* if the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{q}$ has real eigenvalues.

Associating the dimensionless SWEs (Eqs. 2.13) with Eq. 2.22, the SWEs can be written in vector form using

$$\mathbf{q} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \equiv \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}h^2 \\ huv \end{pmatrix} = \begin{pmatrix} q_2 \\ q_2^2/q_1 + \frac{1}{2}q_1^2 \\ q_2q_3/q_1 \end{pmatrix}, \quad \mathbf{s} = \begin{pmatrix} 0 \\ -hB_x + Khv \\ -Khu \end{pmatrix}.$$

The Jacobian of this matrix is

$$\mathbf{A} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \begin{pmatrix} 0 & 1 & 0 \\ -(q_2/q_1)^2 + q_1 & 2q_2/q_1 & 0 \\ -q_2q_3/q_1^2 & q_3/q_1 & q_2/q_1 \end{pmatrix} \quad (2.23)$$

$$= \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix}, \quad (2.24)$$

where $c = \sqrt{h}$ is the wave speed. Note that a dimensional wave speed can be recovered by multiplying by \sqrt{gH} , giving the familiar result $c = \sqrt{gHh}$. This Jacobian has eigenvalues

$$\lambda_1 = u - c, \quad \lambda_2 = u, \quad \lambda_3 = u + c \quad (2.25)$$

All of these are real, and hence the SWEs are indeed a hyperbolic system of conservation laws. For completeness and future reference, the corresponding right eigenvectors are given by

$$r_1 = \begin{pmatrix} 1 \\ u - c \\ v \end{pmatrix}, \quad r_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad r_3 = \begin{pmatrix} 1 \\ u + c \\ v \end{pmatrix} \quad (2.26)$$

Godunov's method has been studied thoroughly for homogeneous hyperbolic conservation laws, where $\mathbf{s} = 0$, and is based on the integral form of such systems:

$$\frac{d}{dt} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(x) dx = \mathbf{f}(\mathbf{q}(x_{i-1/2}, t)) - \mathbf{f}(\mathbf{q}(x_{i+1/2}, t)), \quad (2.27)$$

where $x_{i-1/2}$ and $x_{i+1/2}$ are the boundaries of a control volume centred at x_i . As opposed to the differential form, this integral form admits discontinuities, like hydraulic jumps.

The numerical method can be implemented in various mathematically equivalent forms. In this report the *wave propagation form* is used. For a full derivation, the reader is referred to [3], which first derives the method for scalar equations and subsequently generalises it for conservation law systems, nonlinear equations and ultimately nonlinear systems of equations. Here, only the result is quoted, using the same notation as the textbook.

The core insight of the method is that information in a hyperbolic system propagates at finite speeds. By discretising the domain and replacing the conserved variables by a piecewise constant function, the problem is reduced to a series of step functions. The initial-value problem of an isolated step between two constant values for a conservation law is referred to as a *Riemann problem* and can be solved exactly for many systems (including the SWEs). Each such Riemann problem generates a finite number of waves which propagates into the adjacent cells and updates the conserved variables there. Provided the time steps are sufficiently small, these waves will only propagate into one cell (as opposed to moving past the next Riemann problem), such that the problem becomes entirely local. This condition on the time steps is called the CFL condition (after Courant, Friedrich and Lewy) and can be expressed in terms of the Courant number

$$C \equiv \left| \frac{c\Delta t}{\Delta x} \right|, \quad (2.28)$$

where c is the maximum wavespeed emanating from the Riemann problem, Δt is the time step and Δx is the cell width, i.e. the distance to the next Riemann problem. For all methods presented in this report, the CFL condition is $C \leq 1$.

The equations are discretised as follows. The domain is divided into a regular grid of N cells of width Δx . The positions of the cell centres will be denoted as x_i for $i \in \{0, 1, \dots, N-1\}$, and the cell boundaries at $x_{i\pm1/2}$. At $t_0 = 0$, the conserved quantity \mathbf{q} is replaced by a piecewise constant function, which has value \mathbf{Q}_i^0 in cell i , where

$$\mathbf{Q}_i^0 = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(x, t=0) dx \quad (2.29)$$

is the average of \mathbf{q} across the cell. The values of the \mathbf{Q}_i for the next time level t_{n+1} can now be computed based on time level t_n with a method of the form

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta \mathbf{Q}_{i-1/2} + \mathcal{A}^- \Delta \mathbf{Q}_{i+1/2}), \quad (2.30)$$

where $\Delta t = t_{n+1} - t_n$. The final two terms are called *fluctuations* and are defined as

$$\mathcal{A}^- \Delta \mathbf{Q}_{i+1/2} = \mathbf{f}(\mathbf{Q}_{i+1/2}^\downarrow) - \mathbf{f}(\mathbf{Q}_i) \quad (2.31a)$$

$$\mathcal{A}^+ \Delta \mathbf{Q}_{i-1/2} = \mathbf{f}(\mathbf{Q}_i) - \mathbf{f}(\mathbf{Q}_{i-1/2}^\downarrow), \quad (2.31b)$$

where $\mathbf{Q}_{i-1/2}^\downarrow$ denotes the value $\mathbf{q}(x_{i-1/2}, t_n < t < t_{n+1})$, which can be obtained by solving the nonlinear Riemann problem between \mathbf{Q}_{i-1} and \mathbf{Q}_i (the Riemann problem has a similarity solution which is constant along rays of x/t , hence this value is well-defined). These fluctuations represent the waves propagating into the cell from the step functions at the cell's boundaries.

Boundary conditions are usually implemented by surrounding the domain with so-called *ghost cells*, at x_{-2} , x_{-1} , x_N and x_{N+1} (the exact number of ghost cells depends on the method used, but two on each side are sufficient for all methods presented in this report). These are set manually before each time step to reflect the chosen boundary condition and read from when updating the cells near the boundary. For the purpose of this project only outflow boundary conditions were used which let all waves leave the domain cleanly. Normally, these are implemented simply by copying the variables from the outermost cells into the ghost cells, thereby using zeroth-order extrapolation. However, in the presence of non-zero bathymetry gradients at the boundaries special care has to be taken. The approach suggested in [13], Section 6.1, has been used successfully, which still copies the values of hu and hv into the ghost cells, but sets h according to

$$h_i = h_0 + B_0 - B_i, \quad \text{where } i \in \{-2, -1\} \quad (2.32a)$$

$$h_i = h_{N-1} + B_{N-1} - B_i, \quad \text{where } i \in \{N, N+1\}. \quad (2.32b)$$

This requires the bathymetry to be known outside of the domain, which was the case for all systems investigated in this report.

The previous discussion assumed homogeneous systems. However, the interest of this project does not lie in homogeneous systems, but in conservation laws with source terms. Similar to how a dimensional splitting can be applied, traditionally, the simplest way to solve hyperbolic systems with source terms is by splitting the system into two parts. The homogeneous hyperbolic PDEs:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0.$$

And a set of ordinary differential equations (ODEs) for the source terms:

$$\mathbf{q}_t = \mathbf{s}(\mathbf{q}, x).$$

The appeal of this approach is that the homogeneous system can be solved using well-studied Godunov-type methods, and the source terms can be solved independently by a simple integration in time, also using established methods like Runge-Kutta (originally developed by Runge [30] and Kutta [31]; see [32], Sections 8.3 and 8.4 for a modern account). See [4], Section 12.2.2 or [3], Sections 17.2.2 to 17.5, for instance.

However, as noted earlier, in systems at or close to equilibrium when $\mathbf{q}_t \sim 0$, the terms $\mathbf{f}(\mathbf{q})_x$ and $\mathbf{s}(\mathbf{q}, x)$ may actually both be large (and approximately equal). In this case, the two separate numerical methods will both apply a substantial update to \mathbf{q} . In theory, these updates should cancel, but in practice they will almost never cancel completely. This causes numerical noise which means that equilibria cannot be preserved exactly.

2.3 Roe's Approximate Riemann Solver

The SWEs are a nonlinear system, for which obtaining the full solution to each Riemann problem can be computationally expensive. Furthermore, only little information about the full solution is actually used (only the value along the cell edge). For this reason, several approximate Riemann solvers have been developed. A common approach is to linearise the problem at each cell boundary in the form

$$\mathbf{q}_t + \hat{\mathbf{A}}_{i-1/2}\mathbf{q}_x = 0, \quad (2.33)$$

where $\hat{\mathbf{A}}_{i-1/2}$ is an approximation to the true flux Jacobian $\partial\mathbf{f}/\partial\mathbf{q}$ evaluated at $x_{i-1/2}$. For a linear problem the fluctuations can be written

$$\mathcal{A}^-\Delta\mathbf{Q}_{i+1/2} = \sum_{p=1}^{M_w} (\lambda_{i+1/2}^p)^- \mathcal{W}_{i+1/2}^p \quad (2.34a)$$

$$\mathcal{A}^+\Delta\mathbf{Q}_{i-1/2} = \sum_{p=1}^{M_w} (\lambda_{i-1/2}^p)^+ \mathcal{W}_{i-1/2}^p, \quad (2.34b)$$

where the sum is over the characteristic waves of the Jacobian, $\lambda_{i-1/2}^p$ are the wavespeeds of the p th characteristic (which are the eigenvalues of $A_{i-1/2}$) and $\mathcal{W}_{i-1/2}^p$ are the waves, which are proportional to the right eigenvectors $r_{i-1/2}^p$ and decompose the step at $x_{i-1/2}$ such that $Q_i - Q_{i-1} = \sum_{p=1}^{M_w} \mathcal{W}_{i-1/2}^p$. The superscripts + and - are defined as follows:

$$w^+ = \max(0, w) \quad (2.35)$$

$$w^- = \min(0, w) \quad (2.36)$$

One of the most popular approximations is the solver due to Roe [9]. The following derivation and notation closely follows section 15.3 of [3]. However, LeVeque only derived the solver for the one-dimensional SWEs (without transverse momentum), whereas this section presents a derivation for the x -split equations used throughout this report.

The basic idea is to perform an invertible change of variables $\mathbf{z} = \mathbf{z}(\mathbf{q})$, and parametrise this variable between the cell values surrounding the boundary in question:

$$\mathbf{z}(\xi) = \mathbf{Z}_{i-1} + (\mathbf{Z}_i - \mathbf{Z}_{i-1})\xi \quad (2.37)$$

Then one can obtain two matrices from the integrals:

$$\hat{\mathbf{B}}_{i-1/2} = \int_0^1 \frac{d\mathbf{q}(\mathbf{z}(\xi))}{d\mathbf{z}} d\xi \quad (2.38a)$$

$$\hat{\mathbf{C}}_{i-1/2} = \int_0^1 \frac{d\mathbf{f}(\mathbf{z}(\xi))}{d\mathbf{z}} d\xi. \quad (2.38b)$$

The approximate flux Jacobian is then:

$$\hat{\mathbf{A}}_{i-1/2} = \hat{\mathbf{C}}_{i-1/2} \hat{\mathbf{B}}_{i-1/2}^{-1} \quad (2.39)$$

The purpose of the change of variables is to make the integrals more easily solvable. If one tried to parametrise \mathbf{Q} and integrate the flux Jacobian directly, the integrand would contain rational functions of ξ . With a suitable choice for $\mathbf{z}(\mathbf{q})$, one can simplify the integrands to polynomials.

Following the derivation for the one-dimensional SWEs in section 15.3.3 of [3], a Roe solver can be derived for the x -split SWEs by the following choice for \mathbf{z} :

$$\mathbf{z} = h^{-1/2} \mathbf{q} \Rightarrow \begin{pmatrix} z^1 \\ z^2 \\ z^3 \end{pmatrix} = \begin{pmatrix} \sqrt{h} \\ \sqrt{h}u \\ \sqrt{h}v \end{pmatrix} \quad (2.40)$$

Inverting this relation:

$$\mathbf{q} = \begin{pmatrix} (z^1)^2 \\ z^1 z^2 \\ z^1 z^3 \end{pmatrix} \Rightarrow \frac{dq}{dz} = \begin{pmatrix} 2z^1 & 0 & 0 \\ z^2 & z^1 & 0 \\ z^3 & 0 & z^1 \end{pmatrix} \quad (2.41)$$

Further, writing \mathbf{f} in terms of the components of \mathbf{z} , the Jacobian can be found:

$$\mathbf{f} = \begin{pmatrix} z^1 z^2 \\ (z^2)^2 + \frac{1}{2}(z^1)^4 \\ z^2 z^3 \end{pmatrix} \Rightarrow \frac{df}{dz} = \begin{pmatrix} z^2 & z^1 & 0 \\ 2(z^1)^3 & 2z^2 & 0 \\ 0 & z^3 & z^2 \end{pmatrix} \quad (2.42)$$

Now, let $z_k = Z_{i-1}^k + (Z_i^k - Z_{i-1}^k)\xi$ for $k = 1, 2, 3$ and perform the integrals in Eqs. 2.38. As for the one-dimensional SWEs, the linear terms become

$$\frac{1}{2}(Z_{i-1}^k + Z_i^k) \equiv \bar{Z}^k \quad (2.43)$$

and the cubic term becomes

$$\frac{1}{2}(Z_{i-1}^1 + Z_i^1) \frac{1}{2}((Z_{i-1}^1)^2 + (Z_i^1)^2) \equiv \bar{Z}^1 \bar{h}. \quad (2.44)$$

Hence, the intermediate matrices are

$$\hat{\mathbf{B}}_{i-1/2} = \begin{pmatrix} 2\bar{Z}^1 & 0 & 0 \\ \bar{Z}^2 & \bar{Z}^1 & 0 \\ \bar{Z}^3 & 0 & \bar{Z}^1 \end{pmatrix} \quad (2.45)$$

$$\hat{\mathbf{C}}_{i-1/2} = \begin{pmatrix} \bar{Z}^2 & \bar{Z}^1 & 0 \\ 2\bar{Z}^1 \bar{h} & 2\bar{Z}^2 & 0 \\ 0 & \bar{Z}^3 & \bar{Z}^2 \end{pmatrix} \quad (2.46)$$

and using Eq. 2.39, the approximate flux Jacobian is found to be

$$\hat{\mathbf{A}}_{i-1/2} = \begin{pmatrix} 0 & 1 & 0 \\ \bar{h} - (\bar{Z}^2/\bar{Z}^1)^2 & 2\bar{Z}^2/\bar{Z}^1 & 0 \\ -\bar{Z}^2 \bar{Z}^3 / (\bar{Z}^1)^2 & \bar{Z}^3/\bar{Z}^1 & \bar{Z}^2/\bar{Z}^1 \end{pmatrix} \quad (2.47)$$

$$= \begin{pmatrix} 0 & 1 & 0 \\ \bar{h} - \hat{u}^2 & 2\hat{u} & 0 \\ -\hat{u}\hat{v} & \hat{v} & \hat{u} \end{pmatrix}, \quad (2.48)$$

where

$$\hat{u} = \frac{\sqrt{h_{i-1}}u_{i-1} + \sqrt{h_i}u_i}{\sqrt{h_{i-1}} + \sqrt{h_i}} \quad (2.49)$$

$$\hat{v} = \frac{\sqrt{h_{i-1}}v_{i-1} + \sqrt{h_i}v_i}{\sqrt{h_{i-1}} + \sqrt{h_i}} \quad (2.50)$$

are special weighted averages, called *Roe averages*. Note that, comparing this result with Eq. 2.24, just as in the one-dimensional case this is simply the flux Jacobian of the SWEs evaluated at this special Roe-averaged state, with average wave speed, $\hat{c} = \sqrt{\bar{h}}$. This gives right eigenvectors

$$r_{i-1/2}^1 = \begin{pmatrix} 1 \\ \hat{u} - \hat{c} \\ \hat{v} \end{pmatrix}, \quad r_{i-1/2}^2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad r_{i-1/2}^3 = \begin{pmatrix} 1 \\ \hat{u} + \hat{c} \\ \hat{v} \end{pmatrix} \quad (2.51)$$

Finally, the waves $\mathcal{W}_{i-1/2}^p = \alpha_{i-1/2}^p r_{i-1/2}^p$ can be found by inverting the matrix of right eigenvectors (to obtain a matrix of left eigenvectors) and multiplying it into the step across $x_{i-1/2}$:

$$\alpha_{i-1/2} = \frac{1}{2\hat{c}} \begin{pmatrix} \hat{u} + \hat{c} & -1 & 0 \\ -2\hat{c}\hat{v} & 0 & 2\hat{c} \\ -\hat{u} + \hat{c} & 1 & 0 \end{pmatrix} (\mathbf{Q}_i - \mathbf{Q}_{i-1}) \quad (2.52)$$

This, together with the eigenvalues and eigenvectors provides all the information needed to implement a method based on the fluctuations in Eqs. 2.34. This method will be referred to as the *unbalanced method* throughout the rest of the report.

2.4 Balanced Method: LeVeque

One of the earlier well-balanced methods, which is also based on the wave propagation form of Godunov's method, is due to LeVeque [13]. LeVeque introduces the general theory behind the method, which this report will give a short overview of, and then applies it to the one- and two-dimensional SWEs with bathymetry, showing that it is well-balanced for perturbations about still water. In this report, based on LeVeque's work, a solver for x -split one-dimensional SWEs with Coriolis terms is derived, which is also well-balanced for geostrophic equilibria.

The basic idea is to implement the source terms by introducing additional Riemann problems at the cell centres, thereby replacing each cell value \mathbf{Q}_i with two different values \mathbf{Q}_i^- and \mathbf{Q}_i^+ . To ensure that the method is still conservative, the cell average needs to be maintained, i.e.

$$\mathbf{Q}_i = \frac{1}{2}(\mathbf{Q}_i^- + \mathbf{Q}_i^+). \quad (2.53)$$

Furthermore, the step is chosen such that

$$\mathbf{f}(\mathbf{Q}_i^+) - \mathbf{f}(\mathbf{Q}_i^-) = \mathbf{s}(\mathbf{Q}_i, x_i)\Delta x. \quad (2.54)$$

This condition means that the waves arising from the new Riemann problem are exactly equal and opposite to the effect of the source term in this cell, which implies that neither the source terms nor this Riemann problem have to be solved in the numerical method. The resulting method is simply

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x}(\mathcal{A}^+ \Delta \tilde{\mathbf{Q}}_{i-1/2} + \mathcal{A}^- \Delta \tilde{\mathbf{Q}}_{i+1/2}), \quad (2.55)$$

where the $\mathcal{A}^\pm \Delta \tilde{\mathbf{Q}}_{i-1/2}$ are analogous to the $\mathcal{A}^\pm \Delta \mathbf{Q}_{i-1/2}$ from the unbalanced method, but based on the modified cell values \mathbf{Q}_{i-1}^+ and \mathbf{Q}_i^- instead. Note that this refers to the fluctuations from the exact solution of the Riemann problems. However, approximate solvers can be employed in exactly the same way as for the unbalanced method. The benefit of this method is that the displacement of the cell values leads to small or vanishing Riemann problems at the edges when the system is near equilibrium.

Difficulties in implementing this method arise in determining the values of \mathbf{Q}_i^\pm for nonlinear systems like the SWEs.

2.4.1 A Solver for the SWEs with Bathymetry and Rotation

LeVeque [13] shows how these can be found if bathymetry terms included. The following presents a derivation of a new method which also supports the Coriolis terms.

First, note that the h -equation of the SWEs, Eq. 2.13a, does not contain any source terms, so by Eq. 2.54,

$$(hu)_i^+ = (hu)_i^- = (hu)_i \equiv m_i, \quad (2.56)$$

i.e. the x -momentum remains unchanged. To ensure Eq. 2.53 is satisfied, the new values can be chosen as equal and opposite offsets from the cell average:

$$h_i^\pm = h_i \pm \delta_i \quad (2.57)$$

$$(hv)_i^\pm = (hv)_i \pm \epsilon_i, \quad (2.58)$$

such that only δ_i and ϵ_i need to be found from Eq. 2.54. For the hu -equation, Eq. 2.13b, this condition yields

$$\left(hu^2 + \frac{1}{2} h^2 \right)_i^+ - \left(hu^2 + \frac{1}{2} h^2 \right)_i^- = (-h_i(B_x)_i + K(hv)_i) \Delta x \quad (2.59)$$

$$\left(\frac{m_i^2}{h_i^+} + \frac{1}{2} (h_i^+)^2 \right) - \left(\frac{m_i^2}{h_i^-} + \frac{1}{2} (h_i^-)^2 \right) = (-h_i(B_x)_i + K(hv)_i) \Delta x \quad (2.60)$$

$$m_i^2 \left(\frac{1}{h_i + \delta_i} - \frac{1}{h_i - \delta_i} \right) + \frac{1}{2} ((h_i + \delta_i)^2 - (h_i - \delta_i)^2) = (-h_i(B_x)_i + K(hv)_i) \Delta x \quad (2.61)$$

$$m_i^2 \left(\frac{1}{h_i + \delta_i} - \frac{1}{h_i - \delta_i} \right) + 2h_i\delta_i = (-h_i(B_x)_i + K(hv)_i) \Delta x, \quad (2.62)$$

which gives a cubic equation for δ_i . As in LeVeque's paper, if $u = 0$, which is the case for both still water and geostrophic equilibria, the solution is simply

$$\delta_i = \frac{\Delta x}{2} \left(-(B_x)_i + K \frac{(hv)_i}{h_i} \right). \quad (2.63)$$

If $u \neq 0$, solving the cubic can be more difficult. LeVeque proposes using a few iterations of the Newton-Raphson method (see any undergraduate textbook covering numerical methods, e.g. [33], pp. 990–992), using Eq. 2.63 as an initial guess. The author of this report has also attempted solving the cubic exactly, although this raises the question of which root to choose if there are multiple real roots. See Chapter 4 for evaluation of the results of the choices.

Once δ_i has been found, the step condition 2.54 imposed on the hv -equation, Eq. 2.13c, can be used to obtain ϵ_i :

$$(huv)^+ - (huv)^- = -K(hu)_i \Delta x \quad (2.64)$$

$$m_i \left(\frac{(hv)_i^+}{h_i^+} - \frac{(hv)_i^-}{h_i^-} \right) = -K m_i \Delta x \quad (2.65)$$

$$\frac{(hv)_i + \epsilon_i}{h_i + \delta_i} - \frac{(hv)_i - \epsilon_i}{h_i - \delta_i} = -K \Delta x \quad (2.66)$$

$$((hv)_i + \epsilon_i)(h_i - \delta_i) - ((hv)_i - \epsilon_i)(h_i + \delta_i) = -K \Delta x (h_i^2 - \delta_i^2) \quad (2.67)$$

$$2\epsilon_i h_i - 2(hv)_i \delta_i = -K \Delta x (h_i^2 - \delta_i^2) \quad (2.68)$$

$$\epsilon_i = -\frac{K}{2h_i} \Delta x (h_i^2 - \delta_i^2) + \frac{(hv)_i \delta_i}{h_i} \quad (2.69)$$

Recall that in the presence of a uniform background flow, an additional source term appears in the hv -equation (see Eq. 2.21). For this case, a similar result can be derived:

$$\epsilon_i = \frac{K}{2} \left(\frac{U}{m_i} - \frac{1}{h_i} \right) \Delta x (h_i^2 - \delta_i^2) + \frac{(hv)_i \delta_i}{h_i} \quad (2.70)$$

Note that this diverges if the x -momentum is zero anywhere in the domain.

LeVeque mentions in the paper that the equations simplify slightly if the discretised bathymetry is defined on the cell edges as opposed to the cell centres. It should be stressed

that this is *necessary*, for the method to be perfectly well-balanced. The author has attempted cell-centred schemes as well, and while they generate considerably less noise than an unbalanced method they will not preserve equilibria exactly. Recall that for geostrophic equilibria, another x -derivative enters the source terms at equilibrium (Eq. 2.18). This means that h_s also has to be discretised on the cell edges in order to maintain geostrophic equilibria exactly. This can be shown explicitly:

Assume that the bathymetry is discretised as $B_{i-1/2}$ and the initial surface profile as $(h_s)_{i-1/2}$. Furthermore, the discretised x -derivatives at the cell centres should be defined via the central differences of the corresponding cell edges:

$$(B_x)_i = \frac{B_{i+1/2} - B_{i-1/2}}{\Delta x} \quad (2.71)$$

$$((h_s)_x)_i = \frac{(h_s)_{i+1/2} - (h_s)_{i-1/2}}{\Delta x} \quad (2.72)$$

Similarly, the initial condition for the h_i and $(hv)_i$ should be based on the averages of the cell edge values:

$$h_i = \frac{1}{2}((h_s)_{i-1/2} + (h_s)_{i+1/2}) - \frac{1}{2}(B_{i-1/2} + B_{i+1/2}) \quad (2.73)$$

$$(hv)_i = \frac{h_i((h_s)_x)_i}{K} = \frac{h_i((h_s)_{i+1/2} - (h_s)_{i-1/2})}{K\Delta x} \quad (2.74)$$

Using these values, Eq. 2.63 gives a depth offset of

$$\delta_i = \frac{\Delta x}{2}(((h_s)_x)_i - (B_x)_i) = \frac{1}{2}((h_s)_{i+1/2} - (h_s)_{i-1/2} - B_{i+1/2} + B_{i-1/2}). \quad (2.75)$$

One can now compare h_{i-1}^+ and h_i^- :

$$h_{i-1}^+ = h_{i-1} + \delta_{i-1} \quad (2.76)$$

$$= (h_s)_{i-1/2} - B_{i-1/2} \quad (2.77)$$

$$h_i^- = h_i - \delta_i \quad (2.78)$$

$$= (h_s)_{i-1/2} - B_{i-1/2}. \quad (2.79)$$

Hence, all the Riemann problems at the cell edges vanish.¹ Note that the value taken by h at the cell edges is essentially the exact value $h_{i-1/2}$. As the Riemann problems vanish, all geostrophic equilibria are preserved exactly (to machine precision). The author

¹This is a slight oversimplification, because a similar result does not hold for $(hv)_{i-1}^+$ and $(hv)_i^-$, i.e. there will still be steps in the hv variable. However, Δhv across a step is only used in updating hv itself, which is advected with speed u . But u is zero for geostrophic equilibria and hence hv is also preserved exactly.

has verified this for several different systems. See Chapter 4 for details.

2.5 Balanced Method: Rogers et al.

A different well-balanced method has been developed by Rogers et al. [18] and then generalised in [20]. As opposed to LeVeque's solver, this method is developed specifically for approximate Riemann solvers based on a quasi-linearisation of the system (such as Roe's solver).

The basic idea is to subtract the equilibrium system from the equations, which results in a change of variables where the vector \mathbf{q} is replaced by deviations from the chosen equilibrium. Note that this implies that the resulting solver will only be well-balanced for a single equilibrium. While it is possible to derive a class of solvers for several equilibria (e.g. arbitrary geostrophic equilibria), which can easily be parametrised, one has to choose a particular equilibrium before using the solver on any given system.

Unlike the previous section, this one is only loosely based on the corresponding paper. The author attempted implementing the method as presented in the paper, but was not able to make it work. This section presents an alternative derivation — the author's original work — which yields a slightly different, but still well-balanced method which the author found to be working. This is still based on the idea of Rogers et al. of subtracting off the equilibrium system, so the resulting solver will be referred to as Rogers's solver. For details about the discrepancies with [20], see Appendix A.

Consider again the general form of a conservation law, Eq. 2.22:

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{s}(\mathbf{q}, x)$$

One can evaluate this equation at some equilibrium state \mathbf{q}^{eq} :

$$\mathbf{q}_t^{\text{eq}} + \mathbf{f}(\mathbf{q}^{\text{eq}})_x = \mathbf{s}(\mathbf{q}^{\text{eq}}, x) \quad (2.80)$$

$$\text{or } \mathbf{q}_t^{\text{eq}} + \mathbf{f}_x^{\text{eq}} = \mathbf{s}^{\text{eq}} \quad (2.81)$$

Subtracting this equation off the conservation law, deviatoric quantities can be defined:

$$(\mathbf{q} - \mathbf{q}^{\text{eq}})_t + (\mathbf{f} - \mathbf{f}^{\text{eq}})_x = \mathbf{s} - \mathbf{s}^{\text{eq}} \quad (2.82)$$

$$\text{or } \mathbf{q}'_t + \mathbf{f}'_x = \mathbf{s}' \quad (2.83)$$

Now, the quasi-linearisation can be applied:

$$\mathbf{q}'_t + (\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}^{\text{eq}}))_x = \mathbf{s}' \quad (2.84)$$

$$\mathbf{q}'_t + \mathbf{f}(\mathbf{q})_x - \mathbf{f}(\mathbf{q}^{\text{eq}})_x = \mathbf{s}' \quad (2.85)$$

$$\mathbf{q}'_t + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \mathbf{q}_x - \frac{\partial \mathbf{f}(\mathbf{q}^{\text{eq}})}{\partial \mathbf{q}^{\text{eq}}} \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (2.86)$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}_x - \mathbf{A}^{\text{eq}} \mathbf{q}_x^{\text{eq}} = \mathbf{s}', \quad (2.87)$$

where \mathbf{A} is the Jacobian of \mathbf{f} and \mathbf{A}^{eq} is the same Jacobian, evaluated at the equilibrium state \mathbf{q}^{eq} . This can be further manipulated, by subtracting and adding a cross term:

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}_x - \mathbf{A} \mathbf{q}_x^{\text{eq}} + \mathbf{A} \mathbf{q}_x^{\text{eq}} - \mathbf{A}^{\text{eq}} \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (2.88)$$

$$\mathbf{q}'_t + \mathbf{A}(\mathbf{q}_x - \mathbf{q}_x^{\text{eq}}) + (\mathbf{A} - \mathbf{A}^{\text{eq}})\mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (2.89)$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x = \mathbf{s}' - \mathbf{A}' \mathbf{q}_x^{\text{eq}}, \quad (2.90)$$

where $\mathbf{A}' \equiv \mathbf{A} - \mathbf{A}^{\text{eq}}$. This has the same form as a quasi-linearised method of the original equations, where \mathbf{q} and \mathbf{s} have been replaced by their deviatoric pendants, but the flux Jacobian remains unchanged, and an additional source term has been added. These source terms can be computed with the same source splitting as used in the unbalanced method. This makes it very easy to derive solvers based on this method: apart from choosing an equilibrium state, only $\mathbf{A}' \mathbf{q}_x^{\text{eq}}$ has to be computed — apart from that the unbalanced solver can be adapted trivially.

Note that both $\mathbf{s}' = \mathbf{s}(\mathbf{q}) - \mathbf{s}(\mathbf{q}^{\text{eq}})$ and $\mathbf{A}' = \mathbf{A}(\mathbf{q}, x) - \mathbf{A}(\mathbf{q}^{\text{eq}}, x)$ vanish when $\mathbf{q} = \mathbf{q}^{\text{eq}}$. In this case, one also has $\mathbf{q}' = 0$ everywhere and so $\mathbf{q}'_x = 0$. That is, when the system is in equilibrium there are neither flux terms nor source terms to compute and the method is perfectly well-balanced even though the system is still split into flux and source equations.

The system can be solved using any approximate Riemann solver to estimate the flux Jacobian \mathbf{A} at the cell edges. For simplicity, the Roe solver derived for the unbalanced method will be used for this report.

2.5.1 A Solver for the Still Water Equilibrium

Consider the x -split SWEs with bathymetry and Coriolis terms, Eqs. 2.13 and its Jacobian, Eq. 2.24. In the following, a well-balanced solver for the still water equilibrium is derived.

Without loss of generality, the still water level can be taken as $h_s = 1$. Define the equilibrium water depth as $h_0 \equiv 1 - B$. Hence, $\mathbf{q}^{\text{eq}} = (h_0, 0, 0)$, and $\mathbf{q}' = (\eta, hu, hv)$, where $\eta \equiv h - h_0$ is the deviation from the still water depth. Note that $(h_0)_x = -B_x$.

The terms in Eq. 2.90 are easily computed:

$$\mathbf{s}' = \mathbf{s}(\mathbf{q}) - \mathbf{s}(\mathbf{q}^{\text{eq}}) \quad (2.91)$$

$$= \begin{pmatrix} 0 \\ -hB_x + Khv \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 \\ -h_0B_x + 0 \\ -0 \end{pmatrix} \quad (2.92)$$

$$= \begin{pmatrix} 0 \\ -(h - h_0)B_x + Khv \\ -Khu \end{pmatrix} \quad (2.93)$$

$$= \begin{pmatrix} 0 \\ -\eta B_x + Khv \\ -Khu \end{pmatrix}. \quad (2.94)$$

And:

$$\mathbf{A}' = \mathbf{A}(\mathbf{q}) - \mathbf{A}(\mathbf{q}^{\text{eq}}) \quad (2.95)$$

$$= \begin{pmatrix} 0 & 1 & 0 \\ h - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ h_0 - 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (2.96)$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ (h - h_0) - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \quad (2.97)$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ \eta - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \quad (2.98)$$

The source terms of Eq. 2.90 are then

$$\mathbf{s}' - \mathbf{A}'\mathbf{q}_x^{\text{eq}} = \begin{pmatrix} 0 \\ -\eta B_x + Khv \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ \eta - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \cdot \begin{pmatrix} h_0 \\ 0 \\ 0 \end{pmatrix}_x \quad (2.99)$$

$$= \begin{pmatrix} 0 \\ -\eta B_x + Khv \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ \eta - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \cdot \begin{pmatrix} -B_x \\ 0 \\ 0 \end{pmatrix} \quad (2.100)$$

$$= \begin{pmatrix} 0 \\ -\eta B_x + Khv \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 \\ -\eta B_x + u^2 B_x \\ uv B_x \end{pmatrix} \quad (2.101)$$

$$= \begin{pmatrix} 0 \\ Khv - u^2 B_x \\ -Khu - uv B_x \end{pmatrix} \quad (2.102)$$

Note that in this case the boundary condition must not be applied as in Eqs. 2.32, because the corresponding variable η is independent of the bathymetry profile. Hence, for this solver all variables are set using zeroth-order extrapolation into the ghost cells.

2.5.2 A Solver for Geostrophic Equilibria

Similarly, a solver can be derived for any particular geostrophic equilibrium, as defined by Eq. 2.18, given its surface profile $h_s(x)$.

The equilibrium state is $\mathbf{q}^{\text{eq}} = (h_0, 0, (hv)_0)$, where $h_0(x) = h_s(x) - B(x)$ and $(hv)_0 = h_0(h_s)_x/K$. This gives $\mathbf{q}' = (\eta, hu, \chi)$ where $\eta = h - h_0$ and $\chi = hv - (hv)_0$. Now the source terms can be calculated:

$$\mathbf{s}' = \mathbf{s}(\mathbf{q}) - \mathbf{s}(\mathbf{q}^{\text{eq}}) \quad (2.103)$$

$$= \begin{pmatrix} 0 \\ -h B_x + Khv \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 \\ -h_0 B_x + K(hv)_0 \\ -0 \end{pmatrix} \quad (2.104)$$

$$= \begin{pmatrix} 0 \\ -(h - h_0) B_x + K(hv - (hv)_0) \\ -Khu \end{pmatrix} \quad (2.105)$$

$$= \begin{pmatrix} 0 \\ -\eta B_x + K\chi \\ -Khu \end{pmatrix}. \quad (2.106)$$

And:

$$\mathbf{A}' = \mathbf{A}(\mathbf{q}) - \mathbf{A}(\mathbf{q}^{\text{eq}}) \quad (2.107)$$

$$= \begin{pmatrix} 0 & 1 & 0 \\ h-u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ h_0-0 & 0 & 0 \\ 0 & \frac{(hv)_0}{h_0} & 0 \end{pmatrix} \quad (2.108)$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ (h-h_0)-u^2 & 2u & 0 \\ -uv & v-\frac{(hv)_0}{h_0} & u \end{pmatrix} \quad (2.109)$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ \eta-u^2 & 2u & 0 \\ -uv & v-\frac{(h_s)_x}{K} & u \end{pmatrix}. \quad (2.110)$$

The full source terms are then:

$$\mathbf{s}' - \mathbf{A}'\mathbf{q}_x^{\text{eq}} = \begin{pmatrix} 0 \\ -\eta B_x + K\chi \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ \eta-u^2 & 2u & 0 \\ -uv & v-\frac{(h_s)_x}{K} & u \end{pmatrix} \cdot \begin{pmatrix} h_0 \\ 0 \\ (hv)_0 \end{pmatrix}_x \quad (2.111)$$

$$= \begin{pmatrix} 0 \\ -\eta B_x + K\chi \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 \\ (\eta-u^2)(h_0)_x \\ -uv(h_0)_x + u((hv)_0)_x \end{pmatrix} \quad (2.112)$$

$$= \begin{pmatrix} 0 \\ -\eta B_x + K\chi \\ -Khu \end{pmatrix} - \begin{pmatrix} 0 \\ (\eta-u^2)((h_s)_x - B_x) \\ -uv(h_0)_x + u((hv)_0)_x \end{pmatrix} \quad (2.113)$$

$$= \begin{pmatrix} 0 \\ -\eta(h_s)_x + K\chi + u^2(h_0)_x \\ -Khu + uv(h_0)_x - u((hv)_0)_x \end{pmatrix} \quad (2.114)$$

Note that $(h_0)_x$ and $((hv)_0)_x$ could be further expanded in terms of h_s , B and K , but no further useful cancellations will occur, so it is generally easier and more accurate to implement the method based on discrete versions of the equilibrium quantities themselves.

As for the still water solver, the bathymetry terms in Eqs. 2.32 should not be used.

It should also be noted that one recovers the still water solver from the previous section by setting $h_0 = 1$.

Another important point is that for uniform background flow KhU cannot simply be added before going through these steps (which would yield a $K\eta U$ term in hv -equation), as the chosen states are no longer equilibria of the modified system. It appears that a correct (i.e. converging) method is obtained by simply adding the full KhU term to the balanced solver, but the author has not been able to find a theoretical justification for this. In general, it is probably not desirable to use a geostrophic equilibrium solver for uniform

background flow in the first place, as the system will never be near the chosen equilibrium. It would instead be more useful to derive separate solvers based on the steady subcritical, transcritical or supercritical flow to be used with these systems.² For an overview of the phase space of steady rotating flow over a ridge, see [29].

²“Subcritical”, “transcritical” and “supercritical” refer to the Froude numbers found in the system. The Froude number F is defined as $|u/c|$, where u is the actual velocity of fluid particles in the system, and c is a typical wavespeed. If $F < 1$ everywhere, the flow is subcritical, if $F > 1$ everywhere, the flow is supercritical. If the flow passes through $F = 1$ or contains a shock from $F > 1$ to $F < 1$, the flow is transcritical.

Chapter 3

Implementation

This chapter outlines how the methods derived in the previous section were implemented and evaluated. First, the software package which was used is introduced briefly. Afterwards, the chapter focuses on the framework the author wrote in order to validate and assess the different methods.

3.1 Clawpack

The methods presented in the previous chapter have been implemented using the software package *Clawpack*¹ (**C**onservation **l**aw **p**ackage), version 5.2.2, or more specifically its *Pyclaw* package. Clawpack was originally written by Randall J. LeVeque and is now maintained and further developed by a small group of researchers (including LeVeque).

The basic principle of Clawpack is that the user writes a Riemann solver, which receives the grid of cell averages along with optional auxiliary data and calculates the fluctuations (Eqs. 2.31) for each cell. The package then uses this to implement Godunov's method in wave propagation form, Eq. 2.30. The parameters of the solver, like the grid size, boundary conditions, gravity or the strength of the Coriolis force, as well as the initial conditions and auxiliary data like bathymetry can specified separately. Clawpack also provides the option to specify a separate function to be solved after each time step which can be used to implement source terms via source splitting.

Traditionally, solvers were written in Fortran (FORTRAN 77 originally, Fortran 90 as of Clawpack 5) and the system was configured through data files. The more recent Pyclaw package provides Python wrappers around the core framework (which is itself written in Fortran). Pyclaw now allows the system to be configured programmatically using Python, version 2.7. When using Pyclaw, solvers can be written either in Fortran or in Python. However, it is advisable to use Fortran-solvers, which are in general significantly faster. The solver's code usually dominates the computation, so this an important concern when simulating fine grids, long time scales or multiple dimensions.

For this project, four solvers were implemented in Fortran:

¹<http://www.clawpack.org/>

- An unbalanced solver, based on the Roe solver derived in Section 2.3, together with a source splitting.
- The LeVeque solver derived in Section 2.4.1.
- The Rogers solver for still water systems derived in Section 2.5.1.
- The Rogers solver for geostrophic equilibria derived in Section 2.5.2.

The code for these was based on a one-dimensional SWE solver provided with Clawpack.² In the case of the LeVeque solver, some code was also adapted from LeVeque's original implementation.³

3.2 Test Harness

In order to evaluate these solvers, the author wrote a test framework based on Pyclaw. The full source code can be found in a *git*⁴ repository at <https://github.com/mbuettner/balanced-swe-solvers>.

The framework is set up via a single configuration file, which contains nine parameters:

- Solver to be tested
- Bathymetry profile
- Initial condition of the system
- N , the number of grid cells
- First time level to be plotted (solving always starts at $t = 0$)
- Final time level to be solved and plotted
- Number of frames to plot
- K , the Coriolis parameter
- U , the background flow velocity (only used if the initial condition is uniform flow)

Due to the use of dimensionless equations, the domain can be fixed to $x \in [-0.5, 0.5]$, the bathymetry will always be in $B(x) \in [0, 1]$ and the initial conditions are scaled such that background surface level at the boundaries is $h_s = 1$. All systems use outflow boundary conditions to focus entirely on the dynamics due to the initial conditions.

²https://github.com/clawpack/riemann/blob/ff047a75e0122828fa4e725998f170b93b1d47e1/src/rp1_shallow_roe_with_efix.f90

³http://faculty.washington.edu/rjl/clawpack/shallow/shallow_id/rp/rp1swt.f

⁴<http://git-scm.com/>

The bathymetry profile and initial condition can be selected from a number of pre-defined settings, which were selected to test how well the methods perform on several important systems.

The following seven bathymetry profiles are supported:

- Flat: $B = 0$ everywhere.
- Sloped: $B = 0.4 + 0.8x$. See Fig. 3.1a.
- Gaussian ridge: $B = \frac{1}{2} \exp(-128x^2)$. See Fig. 3.1b.
- Cosine ridge: $B = \frac{1}{2} \cos(4\pi x)^2$, for $|x| < \frac{1}{8}$, $B = 0$, otherwise. See Fig. 3.1c. This is the bathymetry used in [13].
- Parabolic ridge: $B = \frac{1}{2} - 32x^2$, for $|x| < \frac{1}{8}$, $B = 0$, otherwise. See Fig. 3.1d.
- Parabolic bowl: $B = 2x^2$. See Fig. 3.1e.
- Cliff: $B = \frac{1}{4}(1 + \tanh(100x))$. See Fig. 3.1f.

These profiles are discretised by evaluating them on the cell edges and then averaging each cell's edge values to estimate the bathymetry at the cell centres, as required for the LeVeque solver. The other solvers do not impose any condition on the discretisation, so this scheme was used for all solvers.

The three ridge profiles may seem very similar qualitatively, but have been included for different reasons. The cosine ridge and parabolic ridge have compact support, which means that the regions of the domain affected by the bathymetry can be clearly distinguished from those which are not. Likewise, when waves travel across the ridge, it is clear at which time step the wave is first affected by the ridge. The difference between them is that the cosine ridge has a continuous derivative, while the parabola is not differentiable at $|x| = \frac{1}{8}$.

The Gaussian ridge does not have compact support but is still valuable, because this profile exactly matches the surface profile of the geostrophic equilibrium chosen (see below). By using the same profile for both water surface and bathymetry, the flux terms vanish completely (since h becomes a constant), and the two source terms balance each other. This balance does not seem to be addressed usually, presumably because a naïve unbalanced method with source split will still be well-balanced with respect to the source terms themselves (as they are computed in the same step). However, it is worth verifying whether this balance is still preserved for methods which modify how source terms are treated.

Bathymetries with non-zero slope at the domain boundaries (i.e. sloped bathymetry and the parabolic bowl) were a good test for the chosen boundary conditions.

Independently of the bathymetry profile, one of five initial conditions can be chosen. The framework takes into account the chosen bathymetry when calculating the initial values of \mathbf{q} . The supported initial conditions are the following:

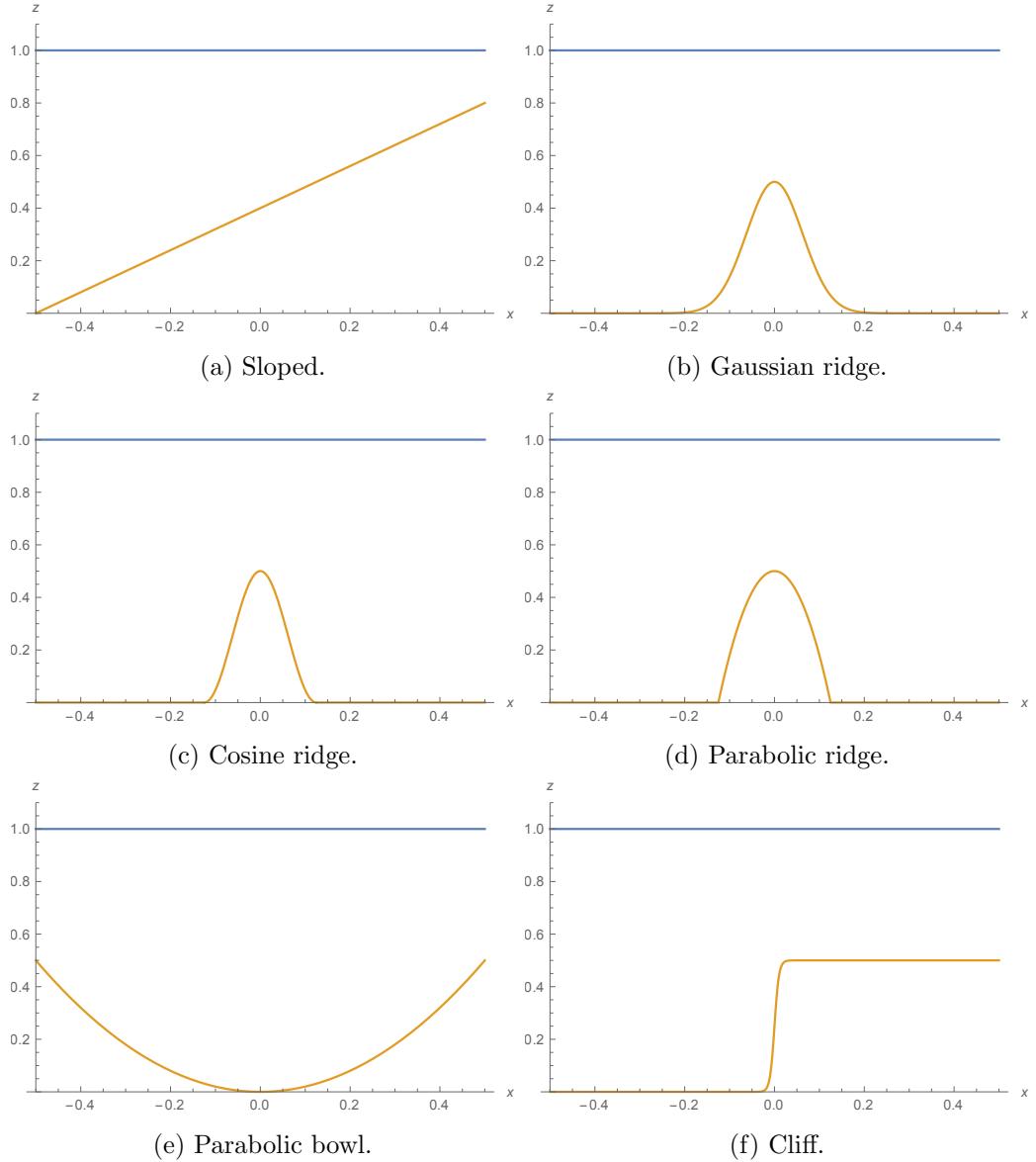


Figure 3.1: Supported bathymetry profiles. Flat profile, $B = 0$, not depicted. Blue lines are still water level, $h_s = 1$, shown for scale. Orange lines are bathymetry.

- Still water. Defined by $h_s = 1$, $u = v = 0$. Hence, the conserved variables are chosen to be $h = 1 - B$, $hu = hv = 0$. This is shown in Figs. 3.1.
- Wave through still water. This system is mostly identical to the still water case, apart from a small perturbation near one edge of the domain, such that $h_s = 1.05$ for $|x + 0.35| < 0.05$, and $h_s = 1$ otherwise. The basic behaviour is that the perturbation separates into two waves, one of which leaves the domain to the left and one of which traverses the domain and, in particular, the bathymetry where it may be partially reflected. The exact dynamics (especially in the presence of the Coriolis term) are more complicated, but this system allows a single wave to be studied in isolation. As an example, Fig. 3.2 shows the system for the cosine ridge.
- Geostrophic equilibrium. The surface level is fixed at $h_s = 1 + \frac{1}{2} \exp(-128x^2)$, such that h is chosen as $h_s - B$. Furthermore, $hu = 0$ and hv is determined by Eq. 2.18. This system requires $K \neq 0$. Figs. 3.3 show this initial condition for the flat and cliff bathymetries and several values of K . Note that hv depends both on the bathymetry and on K .
- Wave through geostrophic equilibrium. This is essentially a linear combination of the previous two systems. First, the geostrophic equilibrium is determined, and then a height perturbation of magnitude 0.05 is added to the interval $x \in [-0.4, -0.3]$. hv is unchanged from the geostrophic equilibrium. Fig. 3.4 shows the initial water level for the parabolic bowl bathymetry.
- Uniform flow. Defined by $h_s = 1$, $u = U$, $v = 0$, where U can be freely chosen. Unless the bathymetry is flat, this is not a steady state, but will settle into a steady system for many parameter combinations. U can be used to choose between sub-critical, transcritical and supercritical flows, as shown in [29]. Figs. 3.5 show initial momentum for two different bathymetries — the initial water level is indistinguishable from the still water case, and $hv = 0$ everywhere. Note that the momentum is not uniform, because h varies.

3.3 Notes on LeVeque Solver

As mentioned in Section 2.4.1, a cubic equation needs to be solved to determine the size of the new Riemann problem at the cell centres. Two different approaches have been implemented which can be chosen via a flag in the Fortran code:

- As implemented by LeVeque, the Newton–Raphson method can be used to improve an initial guess. If the solution does not converge within five iterations, the attempt is aborted and a warning is printed.

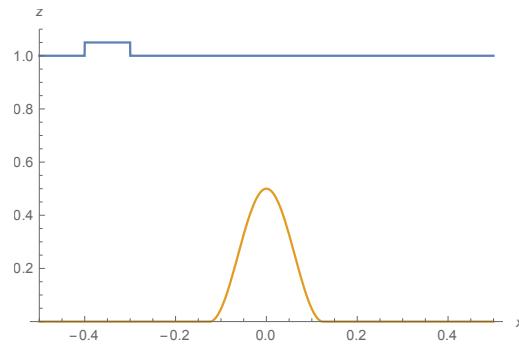


Figure 3.2: Initial condition for “wave through still water” system and cosine bathymetry. Orange is the bathymetry profile, B . Blue is the initial water level, $h_s = h + B$.

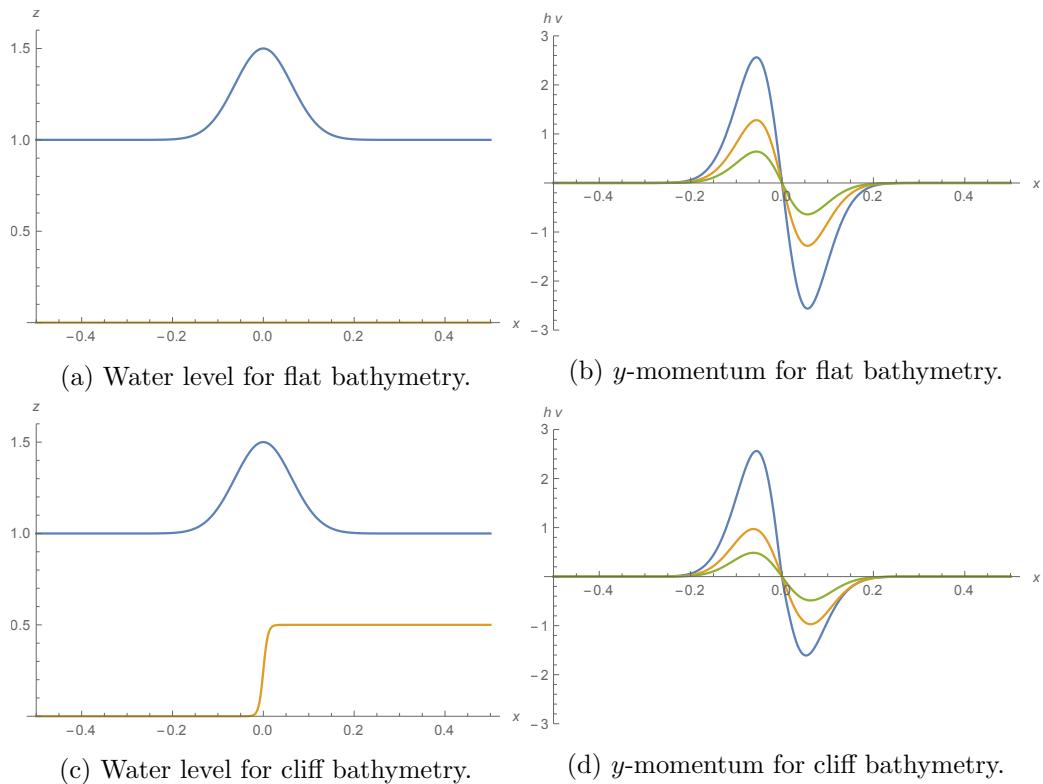


Figure 3.3: Initial conditions for geostrophic equilibria for two bathymetry profiles. In the hv plots, blue corresponds to $K = 2.5$, orange to $K = 5$ and green to $K = 10$.

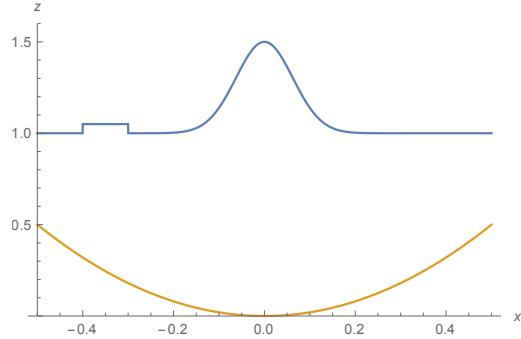


Figure 3.4: Initial condition for “wave through geostrophic equilibrium” system and parabolic bowl bathymetry. Orange is the bathymetry profile, B . Blue is the initial water level, $h_s = h + B$.

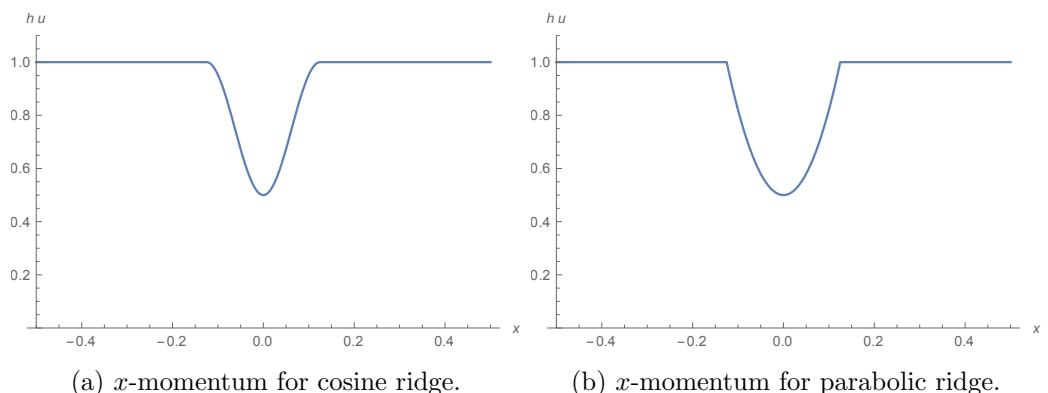


Figure 3.5: Initial conditions for uniform flow for two bathymetry profiles.

- Alternatively, the roots can be found using an explicit formula (see [34], pp. 178–179). However, when the discriminant is greater than 0, this yields three real roots and one of them has to be chosen. In this case, the author’s implementation chooses the root with the smallest absolute value in order to disturb the conserved variables as little as possible (and ensure that the positivity of h is preserved if possible).

Chapter 4

Results

This chapter presents results for the four solvers. For each of the five initial conditions, the solvers will be compared for a few representative bathymetries. Reference results were obtained using the unbalanced solver from Section 2.3 on a very fine grid of 10,000 cells. At the end, the execution time of the different solvers is compared as well.

Apart from the case of uniform flow, the explicit formula has been used for the cubic that appears in the LeVeque solver.

Note that, although the Rogers solver for still water systems is merely a special case of the Rogers solver for geostrophic equilibria, it is still included in the results. This is particularly important for the geostrophic equilibrium cases in which its results show how the geostrophic equilibrium solver would perform on a different equilibrium than the one chosen to balance it.

4.1 Still Water

The first system is simply still water over a cosine ridge, as shown in Fig. 3.1c. Figs 4.1 show the results at times $t \in \{0.1, 0.2, 0.5, 1.0\}$. Only one set of data has been included for the three balanced solvers, as they are indistinguishable in the plot. All sets of data have been obtained for 100 grid cells and with $K = 10$.

Of course, these plots show exactly the motivation for developing balanced solvers in the first place: the unbalanced solver develops quite substantial waves, about 1% of the water depth, and settles into a state which is not still with errors of about 0.5%. From inspection across different grid sizes it appears that the size of the waves is proportional to $1/N$, whereas the error in the settled state is proportional to $1/N^2$.

It should be noted that the data for the two Rogers solvers is exactly zero (since all terms that appear anywhere in the method are always zero), whereas the LeVeque solver is only zero to machine precision (since it relies on two separate computations to make the Riemann problems at the cell edges disappear).

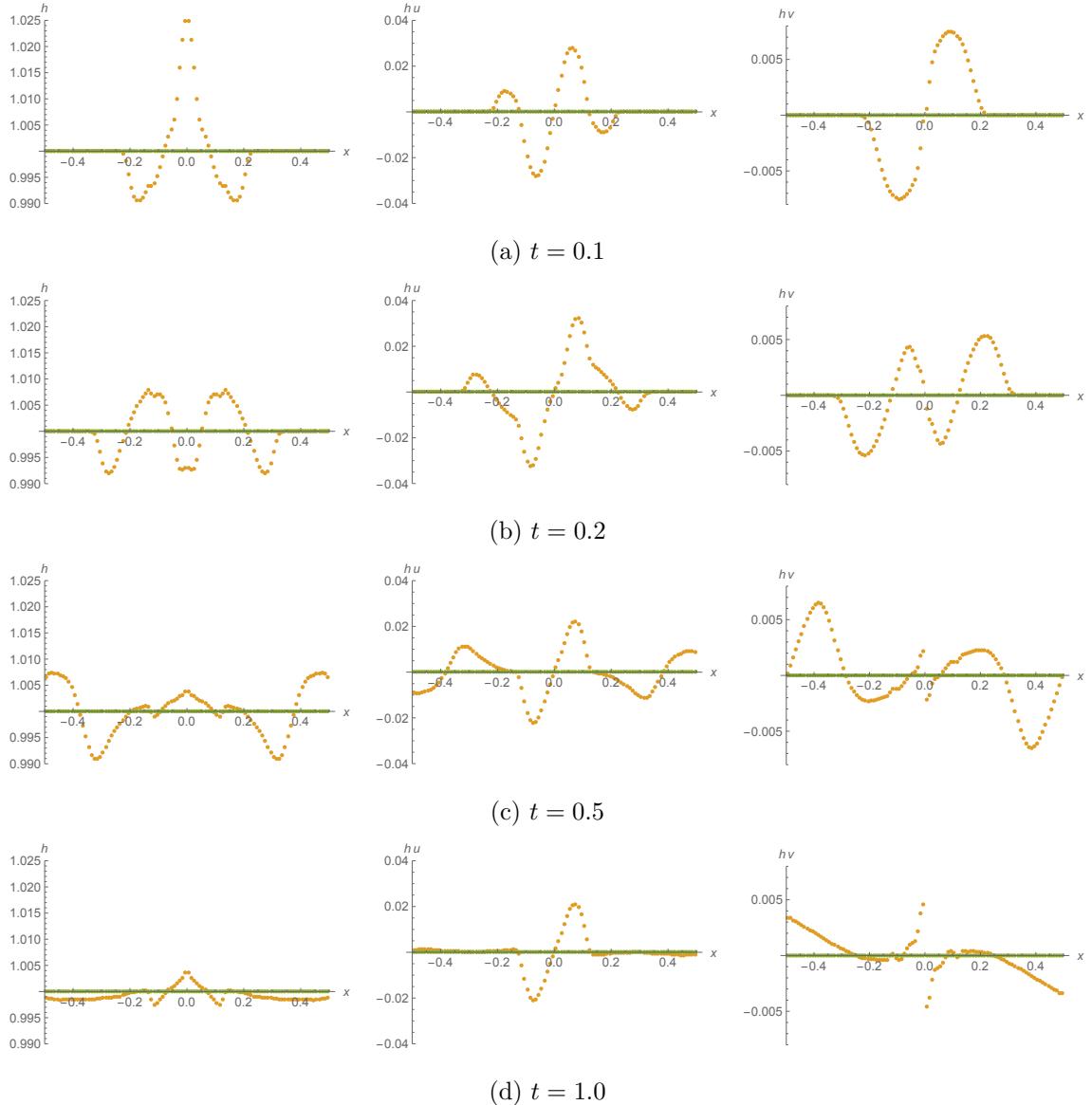


Figure 4.1: Results for still water over a cosine ridge. The exact solution is $h = 1$, $hu = hv = 0$ for all times. The orange data was obtained with the unbalanced solver over 100 grid cells. The green data corresponds to any of the three balanced solvers. The columns correspond to the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

4.2 Wave through Still Water

For this system a small perturbation is added to the surface profile of the water, near the left edge of the domain. The results for the unbalanced and LeVeque solver are shown in Figs. 4.2, those for the Rogers solvers in Figs. 4.3.

It is notable that the unphysical waves of the balanced solver are comparable to the size of the real waves at this resolution, which makes the results very unreliable.

The balanced solvers are much better at approximating the real solution, and produce very good results, considering the low resolution. It is hard to compare the performance of the three balanced solvers in detail, but it seems that the LeVeque solver produces the least accurate result, whereas the Rogers solver for still water systems approximates the real solution most closely.

The discrepancy between the Rogers solvers is attributed to small differences in the implementation. In theory, they should be identical if the chosen geostrophic equilibrium is just the still water case.

4.3 Geostrophic Equilibrium

The results for geostrophic equilibrium (with Gaussian surface profile) over flat bathymetry is shown in Figs. 4.4. As opposed to the previous two systems, these plots show only deviations from the equilibrium state, since errors would be barely visible otherwise.

The unbalanced solver and the Rogers solver for still water systems give identical results. This was to be expected, as the Coriolis term is unchanged for this solver. The magnitude of the generated waves is comparable to the still water case. This is also not surprising, as the gradients of h are on the same order for both setups.

Both the LeVeque solver and the Rogers solver for geostrophic equilibria preserve the state exactly, as desired. Note that, as in the still water case, the deviations for the Rogers solver are exactly zero. The LeVeque solver preserves the equilibrium to machine precision.

As mentioned in the previous chapter, one can also look at the balance between bathymetry and Coriolis terms by choosing the same profile for bathymetry and water surface. In this case, the unbalanced LeVeque and Rogers solver for geostrophic equilibria preserve the balance. In the case of the unbalanced solver this is due to both terms being computed in the same step and cancelling to machine precision. For the other two solvers it is simply a consequence of being able to maintain the balance of any geostrophic equilibrium. The Rogers solver for still water systems on the other hand does not preserve this equilibrium as can be seen in Figs. 4.5. This is not surprising, since only one of the source terms is non-zero which is balanced by the flux term — i.e. this is the exact same problem observed for the unbalanced solver earlier. However, it appears that in this case the amplitude of Δh grows without bounds albeit slowly. The author has not been able to determine whether this is due to an implementation detail (e.g. an inconsistent discretisation of bathymetry and initial conditions) or an inherent problem of the method.

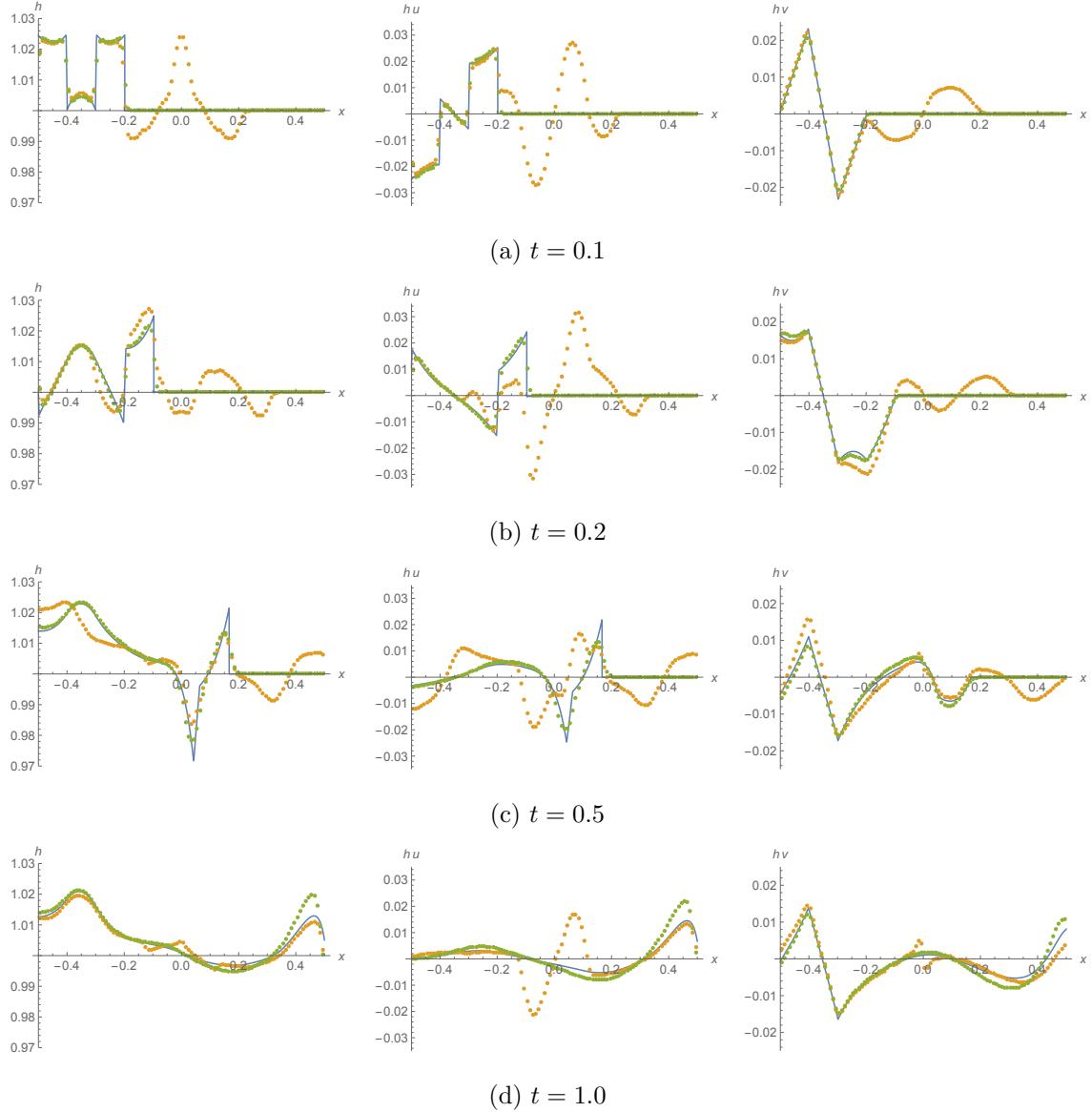


Figure 4.2: Results for wave through still water over a cosine ridge. The reference solution has been computed with the unbalanced solver on 10000 grid cells and is shown as a solid blue line. The orange data was obtained with the unbalanced solver over 100 grid cells. The green data corresponds to the LeVeque solver. The columns correspond to the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

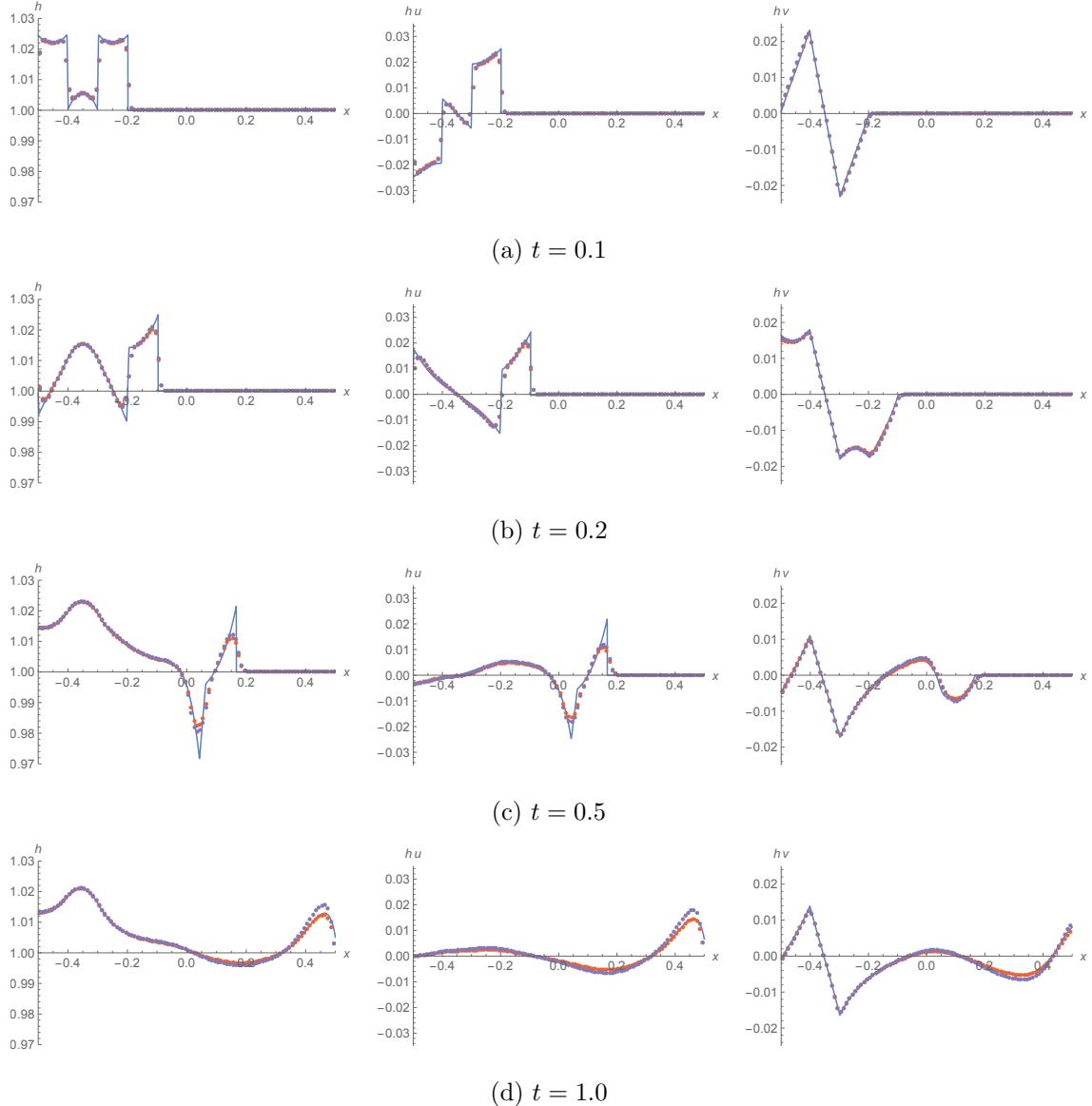


Figure 4.3: Results for wave through still water over a cosine ridge. The reference solution has been computed with the unbalanced solver on 10000 grid cells and is shown as a solid blue line. The red data was obtained with the Rogers solver for still water systems over 100 grid cells. The blue data corresponds to the Rogers solver for geostrophic equilibria. The columns correspond to the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

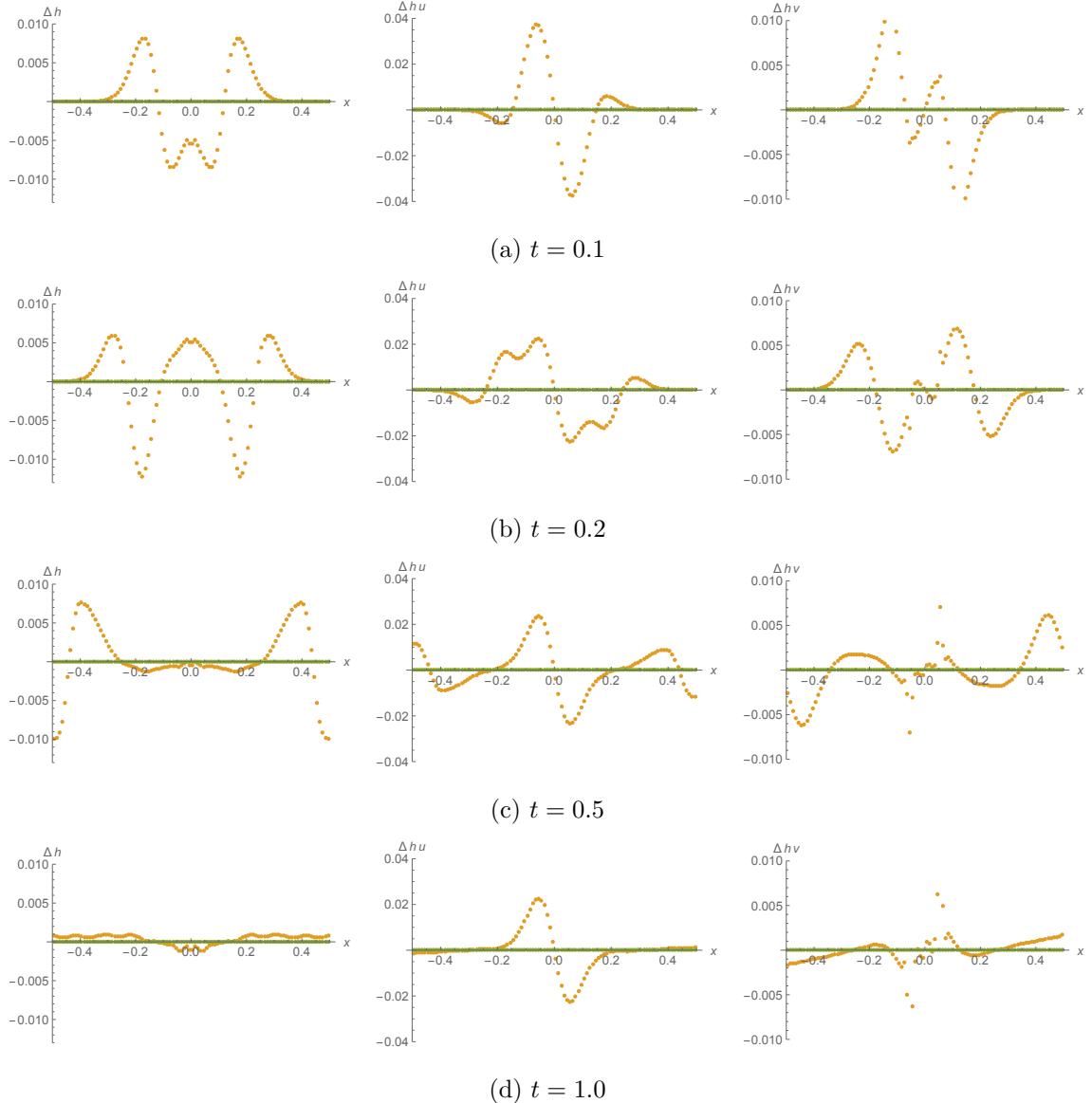


Figure 4.4: Results for geostrophic equilibrium. The equilibrium state has been subtracted off the plots, such that the exact solution is $\Delta h = \Delta hu = \Delta hv = 0$. The orange data corresponds to both the unbalanced solver and the Rogers solver for still water over 100 grid cells. The green data corresponds to both the LeVeque solver and the Rogers solver for geostrophic equilibria. The columns correspond to the deviations of the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

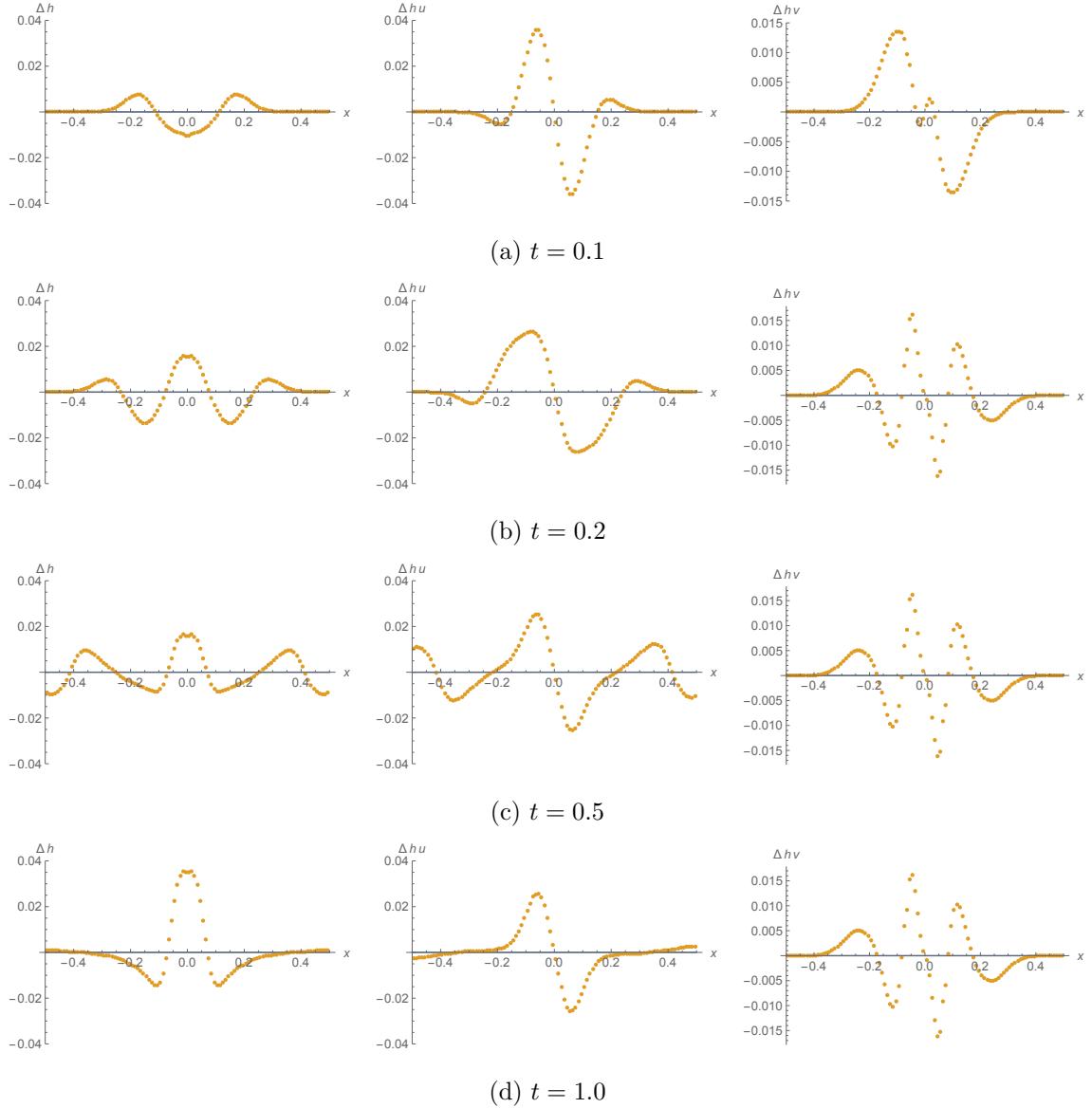


Figure 4.5: Results for geostrophic equilibrium over Gaussian bathymetry. The equilibrium state has been subtracted off the plots, such that the exact solution is $\Delta h = \Delta hu = \Delta hv = 0$. The data points correspond to the Rogers solver for still water systems over 100 grid cells. The columns correspond to the deviations of the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

Method	Time
Unbalanced	11 s
LeVeque (explicit)	16 s
LeVeque (Newton)	11 s
Rogers (still water)	12 s
Rogers (geostrophic equilibrium)	12 s

Table 4.1: Timing results for each method. Minimum of five runs. System: wave through geostrophic equilibrium over bathymetry. $N = 5000$. $K = 10$. $t_{\max} = 1$.

4.4 Wave through Geostrophic Equilibrium

This system is analogous to the wave through still water. It consists of a small perturbation added to the geostrophic equilibrium over flat bathymetry. The results for the unbalanced and LeVeque solver are shown in Figs. 4.6, those for the Rogers solvers in Figs. 4.7. As before, the reference solution has been computed with the unbalanced solver on a fine grid with $N = 10000$.

It can be seen that the solvers which can preserve the geostrophic equilibrium itself (LeVeque and Rogers for geostrophic equilibria) also perform very well on this perturbed system, whereas the other unphysical waves generated by the other two solvers are comparable with the real waves again, which makes the results unreliable.

4.5 Uniform Flow

When the system starts with uniform velocity in subcritical or supercritical flow, all methods were found to produce qualitatively correct (and converging) results, although the Rogers solvers appear to be less accurate than the other two at low resolution.

The interesting case is transcritical flow. While the unbalanced and Rogers solvers can handle these cases as well as any uniform flow, the LeVeque solver fails to compute them correctly. This has already been observed in [13]. LeVeque found that the Newton method diverges when the shock forms over the bathymetry and suggested using an unbalanced method until an almost steady state is reached, and then switching to the balanced method.

The author has observed the same divergences but has not attempted switching methods. Instead, the explicit formula mentioned in Section 3.3 was used to solve the cubic equation. While this stops the solution from breaking down, and the result roughly resembles the true solution, it contains too many shocks, as shown in Figs. 4.8. Presumably, this is due to choosing the wrong root when all three roots are real. For some parameters and grid sizes these unphysical shocks can also grow large enough to violate the assumption that $h > 0$, in which case the solution also breaks down (as the solver has not been developed to support dry states).

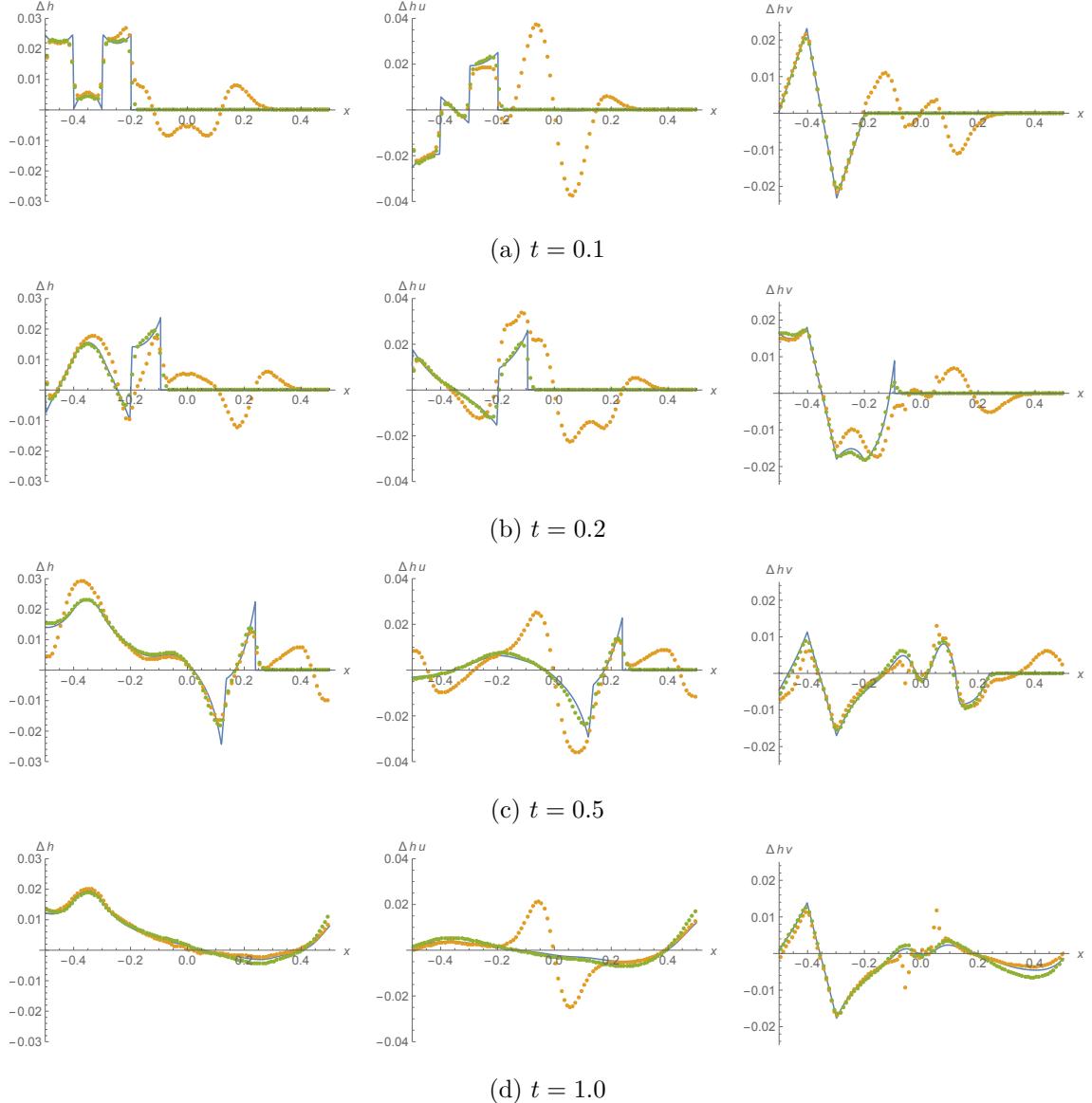


Figure 4.6: Results for wave through a geostrophic equilibrium. The equilibrium state has been subtracted off the plots. The reference solution has been computed with the unbalanced solver on 10000 grid cells and is shown as a solid blue line. The orange data was obtained with the unbalanced solver over 100 grid cells. The green data corresponds to the LeVeque solver. The columns correspond to the deviations of the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

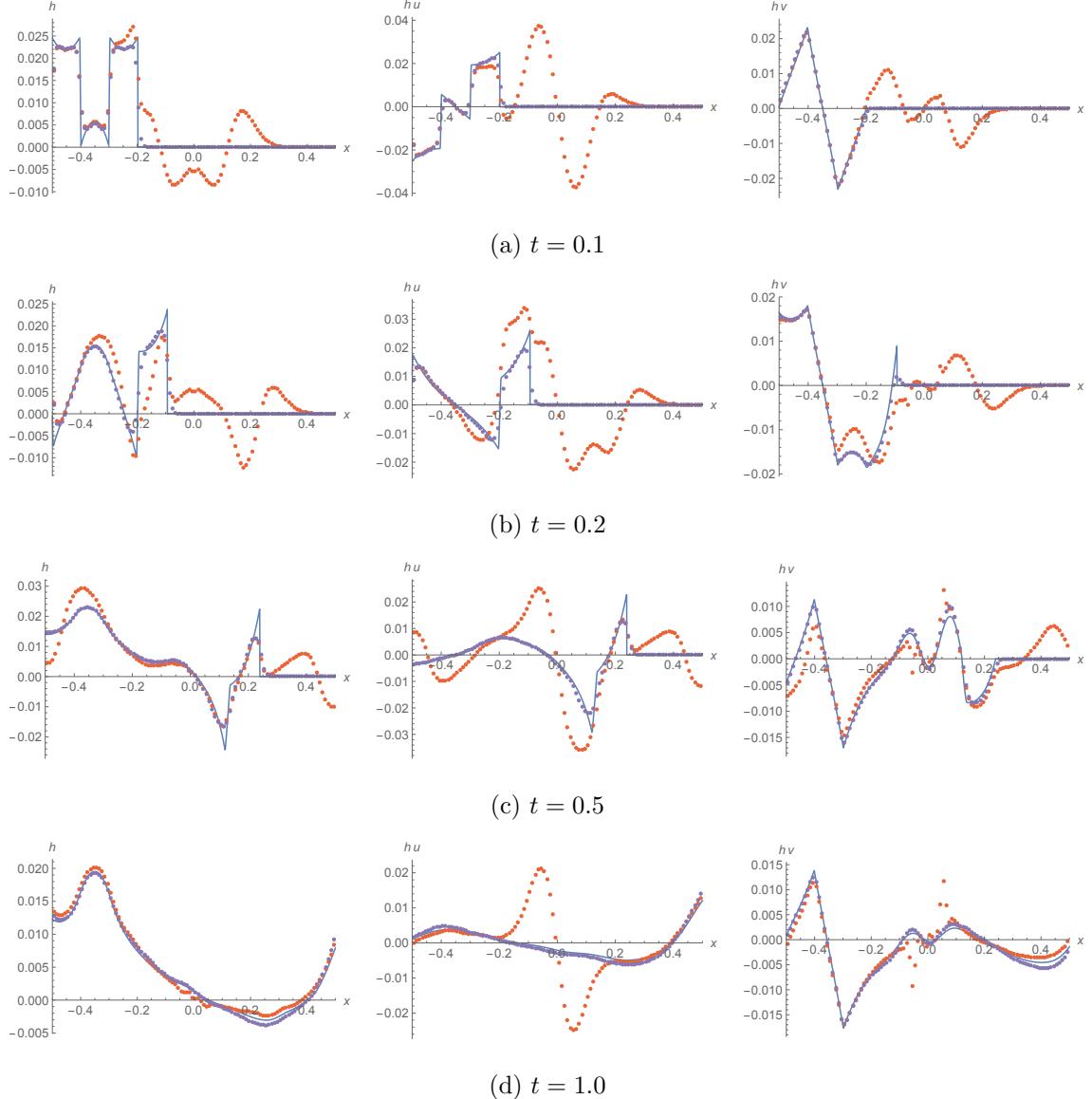


Figure 4.7: Results for wave through a geostrophic equilibrium. The equilibrium state has been subtracted off the plots. The reference solution has been computed with the unbalanced solver on 10000 grid cells and is shown as a solid blue line. The red data was obtained with the Rogers solver for still water systems over 100 grid cells. The blue data corresponds to the Rogers solver for geostrophic equilibria. The columns correspond to the deviations of the conserved variables h , hu and hv and the rows to different time levels. $K = 10$.

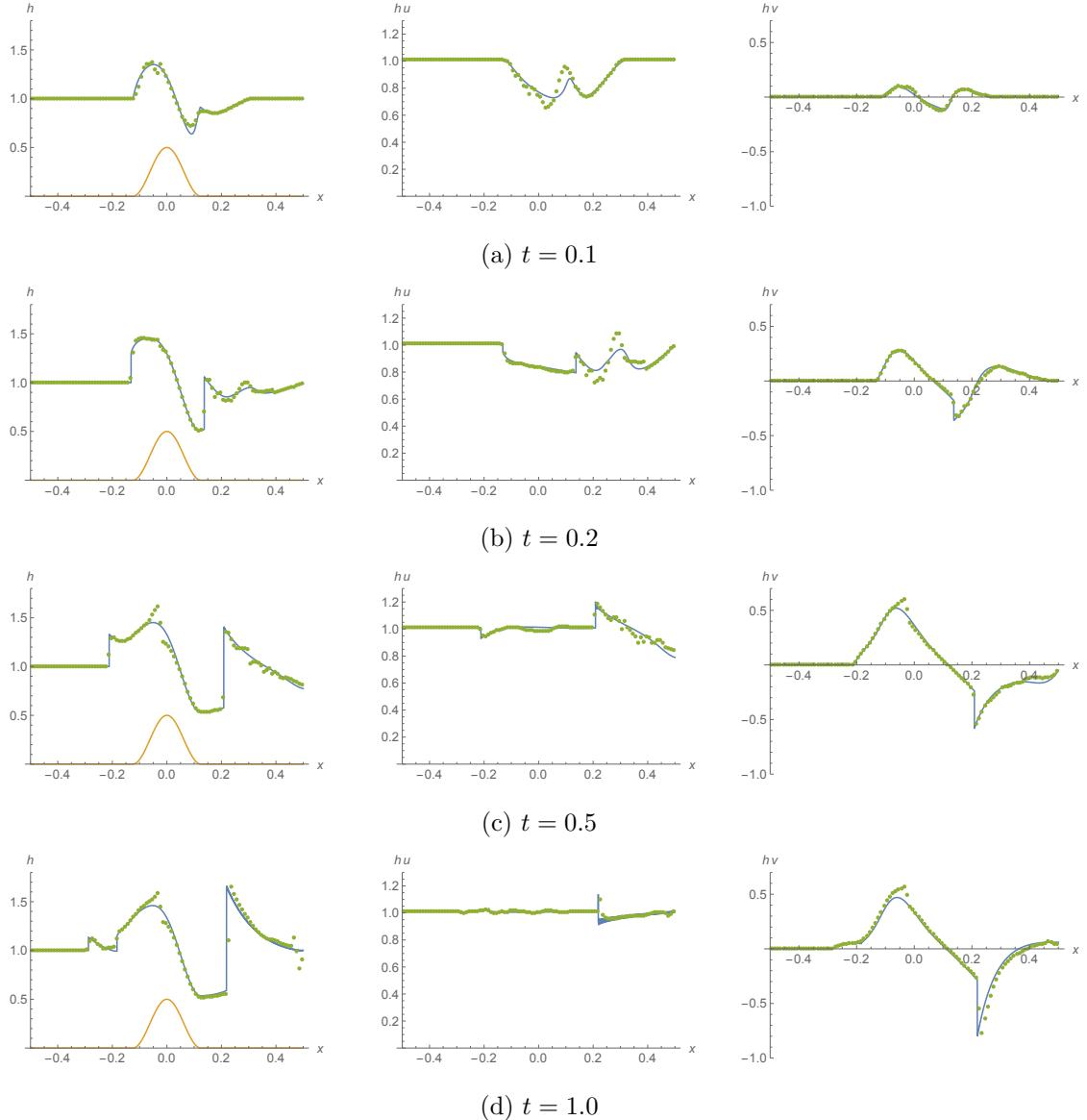


Figure 4.8: Results for transcritical flow over a cosine ridge. The reference solution has been computed with the unbalanced solver on 10000 grid cells and is shown as a solid blue line. The orange line represents the bathymetry. The green data was obtained with the LeVeque solver over 100 grid cells. The columns correspond to the deviations of the conserved variables h , hu and hv and the rows to different time levels. $K = 10$. $U = 1.01$.

4.6 Execution Time

First, it should be noted that all algorithms have a time complexity of $O(N^2)$: for each time step, a fixed amount of computation has to be performed per grid cell, and due to the CFL condition, the time steps scale with the grid resolution. Hence, the time taken by each method is determined by that fixed amount of computation, and one can expect that the methods' timings will only differ by a constant scale factor.

Clawpack prints some diagnostic data when run, including exact time stamps of when the computation started and finished. This was used to get a rough idea of how efficient the different methods are. As a test the wave through a geostrophic equilibrium over a cosine ridge was chosen, using $N = 5000$ and $K = 10$, up to $t = 1$. Each method was timed solving this system five times, and the minimum time was recorded. For the purposes of these tests, the two versions of LeVeque's method were tested separately. The version which uses Newton's method was limited to five iterations. The results are shown in Tab. 4.1.

These results should be taken with a grain of salt and are only intended to give a very rough idea of the scale factors mentioned above. The timings were actually obtained to millisecond accuracy, but only reported to full seconds due to the fluctuations in the results (the tests have been run in a virtual machine). However, none of the implementations have been optimised for speed, so getting more accurate results in a more reliable environment would not yield very useful data at this stage.

Nevertheless, these results indicate a few features of the different solvers:

- The LeVeque solver using the explicit formula is the slowest. This difference could certainly be improved by optimising the implementation of the cubic solver. But at the same time, a more sophisticated algorithm in choosing the correct root might improve this method, which would slow this down even further. Regardless, a factor of approximately 1.5 does certainly not seem prohibitive in using this method.
- The use of Newton's method does not seem to have any noticeable impact on runtime compared to the unbalanced solver. Presumably, this is mostly due to the method converging within very few iterations. Not having to alternate with a source term solver (which was written in Python) might also make up for the additional time used to choose the new Riemann problems.
- While the Rogers solvers were consistently a little slower than the unbalanced solver, the author attributes this to a naïve implementation or fluctuations in the test results. In principle, their computation can be identical to the unbalanced solver but is simply performed on different variables.
- The complexity of the equilibrium chosen for a Rogers solver does not (significantly) affect its efficiency.

Chapter 5

Conclusion

For the purpose of accurately modelling rotating flows in or near geostrophic balance, two approaches have been investigated, based on [13] and [20]. After presenting an alternative form of Rogers's method (see Appendix A), new solvers have been derived based on these approaches and have been shown to be indeed well-balanced for geostrophic equilibria.

This chapter draws some conclusions about the advantages and disadvantages of each method and presents several avenues which could be pursued for future research.

5.1 LeVeque's Method

The main advantage of LeVeque's method is that solvers derived from it are automatically well-balanced for all steady states of the systems, since no information about a particular equilibrium is used in the derivation. This is particularly important if the equilibrium a system will settle into is not known a priori.

On the other hand, in some cases it is not obvious how the cubic should be solved which arises in the case of the SWEs. This has been shown to be problematic when shocks form in transcritical flows, where Newton's method does not converge and an explicit formula can choose the wrong root. Furthermore, while not substantial, this method involves a certain computational overhead for solving the cubic which should be kept in mind.

LeVeque's method seems to be a very good choice for complicated adjustment problems in which the final state of the system is not known up front.

5.2 Rogers's Method

Solvers based on this method are much easier to derive and implement and also do not necessarily incur any computational overhead. They also appear to produce very accurate results for systems near the chosen equilibrium, even at low resolutions. Furthermore, as opposed to the LeVeque solver, they can solve any system just as well as the unbalanced solver (as opposed to failing completely for some cases like transcritical flow).

The main limitation of this approach is that any solver's balance is tied to a particular

equilibrium. This not only means that the solver is essentially unbalanced for any other equilibrium, but it also requires the relevant equilibrium to be known *a priori*. Hence, this solver may not be suitable for adjustment problems where an unknown equilibrium is to be found.

These solvers seem particularly well suited to initial conditions which are small perturbations about a known equilibrium. They are also very useful if an unbalanced solver is already in use due to how easy an existing solver can be adapted.

5.3 Future Work

The field of computational geophysical fluid dynamics in general, and the research of balanced methods in particular is vast, and only a few very specific methods and problems could be investigated within the scope of this project. Hence, there are several interesting ways in which this work could be continued and extended.

Regarding the LeVeque solver, more effort could be put into solving the cubic equation: depending on the context of the problem it might be possible to determine the physically correct root, e.g. by making use of entropy conditions. This would allow the LeVeque solver to be used for arbitrary systems, even if they contain transcritical regions.

For Rogers's method, it is still unclear why the author's solvers differ from the theory presented in [20], yet both the author and Rogers et al. were able to obtain good results with their methods. This should be further investigated to reconcile the discrepancies in the two derivations. Additionally, the apparently unbounded growth of the still water solver for geostrophic equilibrium found in Section 4.3 should be analysed to determine whether this is a problem of the method or its implementation.

Once the theory behind this method is clarified, an even more general solver could be easily derived which could be well-balanced for equilibria with $hu \neq 0$, like the steady flows investigated in [29].

Of course, many more approaches for obtaining well-balanced solvers have been developed in the past, and some of these might be more suitable for balancing the Coriolis terms than the methods chosen in this project. Furthermore, one might investigate other numerical schemes which are not based on Riemann solvers, as Chertock et al. [26] have done in a recent paper.

Apart from looking into other approaches to well-balancing, another point of interest for future research is to look at more sophisticated models. For this report only the effects of bathymetry and the Coriolis force have been considered. However, more realistic models can be obtained by adding additional source terms to account for effects like eddy viscosity, bed friction and surface stresses (i.e. wind). Similarly, the methods used in this paper have assumed $h > 0$ at all times and were not able to deal with dry states. Many of the more recent well-balanced methods do allow for dry states, so adapting these to support the Coriolis terms could yield more powerful methods.

Appendix A

Notes on method by Rogers et al.

In this report, the derivation of balanced solvers based on Rogers et al. [20] has been presented differently from what Rogers et al. have shown in their paper, and subsequently the result has been a different method.

The author initially attempted to implement the method as given in the paper, but this computed unphysical flows when the system was not in equilibrium, whereas the methods based on the author's theory do converge.

The discrepancy arises in the manipulation of Eq. 2.84. Rogers at al. apply the chain rule directly to the $(\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}^{\text{eq}}))_x$ term, obtaining $\frac{\partial \mathbf{f}'}{\partial \mathbf{q}'} \mathbf{q}'_x$. Subsequently, they show that this deviatoric flux Jacobian is identical to the unmodified flux Jacobian $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, obtaining the method

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x = \mathbf{s}' \quad (\text{A.1})$$

which differs from the author's method by the term $-\mathbf{A}' \mathbf{q}^{\text{eq}}$ on the right-hand side.

While the result $\frac{\partial \mathbf{f}'}{\partial \mathbf{q}'} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ is undoubtedly correct, the first step seems to assume that $\mathbf{f}' = \mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{q}^{\text{eq}})$ is a function of $\mathbf{q}' = \mathbf{q} - \mathbf{q}^{\text{eq}}$ only — otherwise the chain rule would yield additional terms. It is not obvious why this should be the case, such that there is no dependence on \mathbf{q}^{eq} as well. This would be the most general case, since \mathbf{q} itself can be decomposed into \mathbf{q}' and \mathbf{q}^{eq} .

One can also go through this derivation without discarding this additional dependence:

$$\mathbf{q}'_t + \mathbf{f}'_x = \mathbf{s}' \quad (\text{A.2})$$

$$\mathbf{q}'_t + \frac{\partial \mathbf{f}'}{\partial \mathbf{q}'} \mathbf{q}'_x + \frac{\partial \mathbf{f}'}{\partial \mathbf{q}^{\text{eq}}} \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (\text{A.3})$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x + \left(\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}^{\text{eq}}} - \frac{\partial \mathbf{f}(\mathbf{q}^{\text{eq}})}{\partial \mathbf{q}^{\text{eq}}} \right) \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (\text{A.4})$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x + \left(\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{q}^{\text{eq}}} - \frac{\partial \mathbf{f}(\mathbf{q}^{\text{eq}})}{\partial \mathbf{q}^{\text{eq}}} \right) \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (\text{A.5})$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x + \left(\frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} - \frac{\partial \mathbf{f}(\mathbf{q}^{\text{eq}})}{\partial \mathbf{q}^{\text{eq}}} \right) \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (\text{A.6})$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x + (\mathbf{A} - \mathbf{A}^{\text{eq}}) \mathbf{q}_x^{\text{eq}} = \mathbf{s}' \quad (\text{A.7})$$

$$\mathbf{q}'_t + \mathbf{A} \mathbf{q}'_x = \mathbf{s}' - \mathbf{A}' \mathbf{q}_x^{\text{eq}}, \quad (\text{A.8})$$

where the results $\frac{\partial \mathbf{f}'}{\partial \mathbf{q}'} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ and $\frac{\partial \mathbf{q}}{\partial \mathbf{q}^{\text{eq}}} = 1$ have been used. This gives the same result as Eq. 2.90, including the additional source term.

The author has contacted Benedict Rogers and Alistair Borthwick about this discrepancy. However, while both of them replied very kindly, they do not seem to have addressed the problem with a justification for the initial manipulation — from (3.9) to (3.10) in their paper.

However, Rogers et al. have obtained good results with their method, so it seems likely that there is simply a different underlying assumption in author's work than in theirs, which is accounted for by the additional source term. This should certainly be investigated further.

Bibliography

- [1] Randall J LeVeque. *Numerical methods for conservation laws*, volume 132. Springer, 1992.
- [2] Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics*, volume 16. Springer, 1999.
- [3] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.
- [4] Eleuterio F Toro. *Shock-capturing methods for free-surface shallow flows*. Wiley, 2001.
- [5] Vladimir Zeitlin. *Nonlinear Dynamics of Rotating Shallow Water: Methods and Advances: Methods and Advances*, volume 2. Elsevier, 2007.
- [6] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.
- [7] Eleuterio F Toro and Pilar Garcia-Navarro. Godunov-type methods for free-surface shallow flows: A review. *Journal of Hydraulic Research*, 45(6):736–751, 2007.
- [8] Peter K Sweby. Godunov methods. In *Godunov Methods: Theory and Applications*, pages 879–898. Springer, 2001.
- [9] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.
- [10] James Glimm, Guillermo Marshall, and Bradley Plohr. A generalized riemann problem for quasi-one-dimensional gas flows. *Advances in Applied Mathematics*, 5(1):1–30, 1984.
- [11] Adrian E Gill. *Atmosphere-ocean dynamics*, volume 30. Academic press, 1982.
- [12] Joshua M Greenberg and Alain-Yves LeRoux. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis*, 33(1):1–16, 1996.

- [13] Randall J LeVeque. Balancing source terms and flux gradients in high-resolution godunov methods: the quasi-steady wave-propagation algorithm. *Journal of computational physics*, 146(1):346–365, 1998.
- [14] Pilar Garcia-Navarro and Maria Elena Vazquez-Cendon. On numerical treatment of the source terms in the shallow water equations. *Computers & Fluids*, 29(8):951–979, 2000.
- [15] Matthew E Hubbard and Pilar Garcia-Navarro. Flux difference splitting and the balancing of source terms and flux gradients. *Journal of Computational Physics*, 165(1):89–125, 2000.
- [16] J Burguete and P García-Navarro. Efficient construction of high-resolution tvd conservative schemes for equations with source terms: application to shallow water flows. *International Journal for Numerical Methods in Fluids*, 37(2):209–248, 2001.
- [17] Ll Gascón and JM Corberán. Construction of second-order tvd schemes for nonhomogeneous hyperbolic conservation laws. *Journal of computational physics*, 172(1):261–297, 2001.
- [18] Ben Rogers, Masayuki Fujihara, and Alistair GL Borthwick. Adaptive q-tree godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids*, 35(3):247–280, 2001.
- [19] Derek S Bale, Randall J LeVeque, Sorin Mitran, and James A Rossmanith. A wave propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM Journal on Scientific Computing*, 24(3):955–978, 2003.
- [20] Benedict D Rogers, Alistair GL Borthwick, and Paul H Taylor. Mathematical balancing of flux gradient and source terms prior to using roe’s approximate riemann solver. *Journal of Computational Physics*, 192(2):422–451, 2003.
- [21] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoît Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.
- [22] Ashwin Chinnayya, Alain-Yves LeRoux, and Nicolas Seguin. A well-balanced numerical scheme for the approximation of the shallow-water equations with topography: the resonance phenomenon. *Int. J. Finite Volumes*, 1:1–33, 2004.
- [23] QiuHua Liang and Alistair GL Borthwick. Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography. *Computers & Fluids*, 38(2):221–234, 2009.

- [24] Qiuhua Liang and Fabien Marche. Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in water resources*, 32(6):873–884, 2009.
- [25] HJ Zhang, JZ Zhou, S Bi, QQ Li, and Y Fan. A well-balanced numerical scheme for shallow water simulation on adaptive grids. In *Journal of Physics: Conference Series*, volume 495, page 012049. IOP Publishing, 2014.
- [26] Alina Chertock, Michael Dudzinski, Alexander Kurganov, and Mária Lukácová-Medvid'ová. Well-balanced schemes for the shallow water equations with coriolis forces. submitted 2014.
- [27] François Bouchut. *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws: And Well-Balanced Schemes for Sources*. Springer, 2004.
- [28] Paul J Dellar and Rick Salmon. Shallow water equations with a complete coriolis force and topography. *Physics of Fluids (1994-present)*, 17(10):106601, 2005.
- [29] J G Esler, O J Rump, and E R Johnson. Steady rotating flows over a ridge. *Physics of Fluids (1994-present)*, 17(11):116601, 2005.
- [30] Carl Runge. Über die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.
- [31] Wilhelm Kutta. Beitrag zur nähерungsweisen Integration totaler Differentialgleichungen. 1901.
- [32] Autar K Kaw, E Eric Kalu, and Glen Besterfield. *Numerical methods with applications*. Lulu. com, 2009.
- [33] Kenneth Franklin Riley, Michael Paul Hobson, and Stephen John Bence. *Mathematical methods for physics and engineering: a comprehensive guide*. Cambridge University Press, 2006.
- [34] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.