Finite Volume Methods and Adaptive Refinement for Tsunami Propagation and Inundation

David L. George

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2006

Program Authorized to Offer Degree: Applied Mathematics

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

David L. George

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Randall J. LeVeque

Reading Committee:

Randall J. LeVeque

Christopher S. Bretherton

Vasilli V. Titov

Date:

Signature____________________________

Date________________________________

University of Washington

Abstract

Finite Volume Methods and Adaptive Refinement for Tsunami Propagation and Inundation

David L. George

Chair of the Supervisory Committee:
Professor Randall J. LeVeque
Department of Applied Mathematics

The shallow water equations are a commonly accepted governing system for tsunami propagation and inundation. In their most generally valid form, the equations are a set of hyperbolic integral conservation laws—a general class of systems for which an extensive body of numerical theory exists. In this thesis, finite volume wave propagation methods—high resolution Godunov-type methods—are extended to this form of the shallow water equations in the context of tsunami modeling. A novel approximate Riemann solver is developed in order to handle the diverse flow regimes exhibited by tsunamis. This solver provides well-balanced source term inclusion required for accurate resolution of near steady state solutions—a necessity when modeling transoceanic tsunami propagation. The solver also preserves nonnegative water depths and accurately captures discontinuities and moving shorelines, making it appropriate for inundation modeling. Adaptive refinement algorithms are extended to this application. These algorithms allow evolving sub-grids of various resolutions to move with features in the solution. Extending the adaptive algorithms to tsunami modeling requires some new interpolation and integrating strategies in order to preserve steady states. Finally, the methods are extended to solution on a sphere or idealized earth-fitted reference ellipsoids. Together, the methods developed allow modeling transoceanic tsunami propagation as well as coastal inundation in single global-scale simulations.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

My thesis advisor Randall LeVeque has a well-deserved reputation among his students for being an excellent mentor, and I would like to thank him for his guidance. His unique combination of patience and high expectations provided the encouragement necessary to complete this work. I would also like to thank him for his generous support and encouragement to attend many conferences and pursue collaboration with researchers in diverse fields.

I would like to sincerely thank my Reading and Supervisory Committee, Christopher Bretherton, Vasily Titov and Richard Anderson, for their valuable advice as well as accommodating my schedule during the busy summer months. I would also like to express gratitude to Professor Harry Yeh of Oregon State University for his support and advice. His expertise and knowledge of tsunamis guided much of this work. Professor Marsha Berger of the Courant Institute at NYU developed and wrote the core algorithms for the adaptive mesh refinement, and her generous assistance was always only a phone call or email away. I would also like to thank Roger Denlinger of the U.S.G.S. for his advice and interest in this work.

I am grateful to the current and former students of Professor LeVeque, as well as everyone in the Applied Mathematics department for input and collaboration. I owe thanks to Damon Toth for collaboration and discussion along the way, and for assisting me with my final defense petition while I was away.

Lastly, I would like to thank my family and friends, and Rebecca and her family for patience and encouragement.

Chapter 1

INTRODUCTION

1.1 Statement of Purpose

Whenever a mathematical model is used to represent or predict the behavior of a real physical system there are two possible sources of error that should be carefully distinguished. First, the set of equations assumed to govern the system will always oversimplify the true physical system. Second, given the complexity of most physical applications, the governing equations will rarely be exactly solved—either analytical approximations or numerical computations will only be estimates of solutions to the already oversimplified system. Interpreting the source of error when models fail to represent reality is not always easy, yet, the first source of error is typically more appreciated in the scientific community. This thesis primarily concerns the latter issue.

The purpose of the work leading to this thesis was to develop accurate, robust and efficient numerical methods for solving the shallow water equations when they are used for tsunami modeling. Tsunamis exhibit a wide variety of diverse fluid dynamical features and no single set of governing equations approximated by a numerical method will ideally model all of the features. However, general patterns and important characteristics of tsunamis can be predicted by various sets of governing equations—the shallow water equations being one important and commonly used example. This thesis describes a particular class of numerical methods that is well equipped for dealing with some of the mathematical features exhibited by the shallow water equations. Further, it describes the development, application and modification of algorithms to confront some of the unique difficulties presented by tsunamis. It must be emphasized that this work is centered around the numerical issues involved when solving a set of equations in a certain context. It is not a set of numerical experiments for the scientific investigation of tsunamis, although some comparisons with physical data are presented in Chapter 9 and Chapter 10.

1.2 Tsunami Modeling with the Shallow Water Equations

Computational modeling has been commonly accepted for decades by the geological and oceanographic communities as serving an integral role in tsunami science. Numerical and mathematical models of actual or potential tsunamis serve several purposes. First, they can be used for potential inundation mapping, which is important for hazard mitigation such as emergency planning and coastal infrastructure construction. Second, they can be used for scientific studies aimed at revealing the physical processes of tsunami initiation, propagation and inundation. These scientific studies then allow better predictions of potential tsunami events. Third, numerical simulations can be used to establish seismic criteria for issuing tsunami warnings in the event of an actual tsunami, and they might eventually be used for real-time simulations for issuing warnings. (For more information on these topics, see, for instance, the National Oceanic and Atmospheric Administration (NOAA) tsunami research site [79].)

Because tsunamis begin as very long wavelength disturbances, they are commonly, and have been historically, modeled with the shallow water equations—a system of nonlinear hyperbolic PDEs described in detail in this thesis. Like tsunamis themselves, solutions to the shallow water equations exhibit diverse flow features and the equations serve as an approximation to the global propagation regime, the near shore regime (which can exhibit discontinuities such as breaking waves and propagating bores) and the inundation regime (the flooding of dry land).

Mathematical and physical studies of wave runup on sloping beaches and other simplified shore geometries, for the purpose of investigating tsunami inundation, have long made use of numerical and analytical approximations to the shallow water equations. See for instance the work of Carrier and Greenspan [13], Hibberd and Peregrine (*e.g.* [80, 44]), Yeh (*e.g.* [104, 105, 108, 111, 106]), Briggs, Synolakis, and Liu (*e.g.* [11, 90, 72]) and Fujima (*e.g.* [23, 24]). For early studies of isolated flow features such as runup, numerical methods were largely based on finite difference approximations based on Taylor series, such as the Lax-Wendroff [60] and similar methods. For the purpose of computing real tsunami simulations with global-scale propagation or runup, more specific and sophisticated finite difference methods were developed in the 1990s that continue to evolve. See, for instance, Titov and Synolakis [93, 97, 94, 98, 99, 96], Liu and Cho (*e.g.* [72, 71]), Mader (*e.g.* [73]), Goto and Sato [36], Imamura [48], as well as Kowalik (*e.g.* [56]).

The shallow water equations are found in several different forms, some more generally valid than others. The different forms of the equations present different sets of numerical difficulties, and often certain flow regimes make one form of the equations more or less tractable. For instance, for the global propagation regime, a *primitive* form of the equations is the most tractable and provides the most reliable solutions, given standard numerical methods. This form is typically used for modeling global propagation, for which other forms of the equations can be problematic. However, the validity of this form of the equations breaks down in the near shore regime if steepening or discontinuities exist, and furthermore, standard numerical methods often exhibit spurious results near such features. Only the most generally valid integral form of the shallow water equations is valid near discontinuities.

The shallow water equations, in their most fundamental physically relevant form, are hyperbolic integral conservation laws. This mathematical class of equations arises in diverse applications with wave-like behavior, and is notorious for exhibiting difficulties for numerical solution, such as non-uniqueness of solutions as well as propagating shock-waves and similar features that generate numerical oscillations. Nevertheless, a large body of theory and numerical methods have been dedicated to overcoming these difficulties. Such work has largely been in the field of compressible gas dynamics, where systems such as the Euler equations exhibit propagating shock-waves and large variations of physical scales in the solution, requiring adaptive refinement. One class of methods that has been particularly successful for such applications is the *Godunov-type* finite volume methods and the related *high-resolution methods.* In recent years these methods have been increasingly used for solving the shallow water equations for various applications. See, for instance, Kurganov and Levy (*e.g.* [59]), Gallouet Herard and Seguin (*e.g.* [26]), Chinnayya and LeRoux (*e.g.* [15]), Garcia-Navarro *et al* (*e.g.* [27]) and the text by Toro [101]. The aim of the research presented in this thesis has been to extend modern developments in this field to tsunami modeling. By taking this approach, several new difficulties are encountered. However, by using the most generally valid form of the shallow water equations—and methods equipped at dealing with that form—the potential to accurately and robustly model all solution regimes might be realized. Although using this general class of numerical method for the shallow water equations is by no means novel, tsunamis present a collection of unique features that must be overcome. The methods described here present a unified approach at dealing with these diverse difficulties when using Godunov-type finite volume methods for hyperbolic conservation laws.

1.3 Numerical Challenges

We will categorize the numerical difficulties associated with tsunami modeling, as belonging to two types—*diverse flow regimes* and *diverse spatial scales.*

1.3.1 Flow Regimes

Tsunamis exhibit diverse flow characteristics that demand diverse properties from a numerical method. For instance, a tsunami begins as a tiny perturbation to the motionless steady state, since the amplitude of a deep ocean tsunami is on the order of centimeters and ocean depths can approach thousands of meters. Accurately modeling global propagation therefore demands resolving this tiny deviation from the background steady state. Since the energy of the deviation is focused and concentrated as the waves approach shore, small errors can be amplified—much like the tsunami itself. This problem is pronounced with the conservative integral form of the shallow water equations, since, in this form, the steady state arises from the nontrivial balance of possibly large gradients.

As a tsunami wave approaches shallow coastal waters, its wavelength, and hence energy is compressed and its amplitude increases. In this region nonlinearities lead to steepening waves, or even breaking waves and propagating bores. Modeling this regime accurately with the shallow water equations requires a numerical method that can handle steep gradients and discontinuities robustly.

Ultimately, the inundation of dry land is of the most interest, since this is where tsunamis unleash their destructive capacity. Modeling inundation requires a method that can accurately track or capture the shoreline boundary, as well as be robust to the appearance of dry regions. Preserving positivity of depth is a well known difficulty for these types of methods when used in shallow and drying regions. Additionally, accurately modeling the moving shoreline is problematic given realistic finite grid spacings, and even determining the right governing assumptions can be difficult. However, the integral form of the shallow water equations is valid when applied to a region that contains wet and dry regions and this conservative form can be used to capture moving shorelines.

An integral component of the numerical methods described in this thesis is the *Riemann solver*. Here, a novel Riemann solver is presented that is designed to solve many of the above difficulties simultaneously.

1.3.2 Spatial Scales

The diverse flow regimes of tsunamis occur at extremely different spatial scales. For instance, in the deep ocean, tsunami wavelength is on the order of hundreds of kilometers. Modeling this propagation requires global-scale grids, which necessitates coarse grid spacings. Luckily these can accurately resolve the long wavelength tsunamis; however, as tsunamis shoal in shallow waters, their wavelengths are compressed to the meter scale. In the inundation regime, bathymetry and local coastal features (even manmade structures) can influence and focus the inundation patterns in unpredictable ways. These features can occur at the spatial scale of several meters. Modeling all of these flow features must involve some form of grid refinement. Refining all coastal areas to an ideal scale is prohibitively costly; however, rarely does a given tsunami impact every potential shoreline—allowing selective refinement. Since bathymetry focuses tsunami energy in unpredictable ways, knowing which areas to refine *a priori* is problematic.

For these reasons this thesis presents the use of adaptive refinement methods for tsunami modeling. Algorithms previously developed for hyperbolic problems generally are modified and extended to this application.

1.4 Thesis Overview

In Chapter 2 some mathematical properties of hyperbolic conservation laws and related hyperbolic PDEs will be discussed in general terms. This discussion will provide the proper mathematical context for the numerical methods described in this thesis. Chapter 3 introduces Godunov-type finite volume methods and Riemann solver techniques as well as the specific *wave propagation algorithms* that underly all of the numerical methods described in later chapters. For the most part, Chapter 2 and Chapter 3 provide background material for readers less familiar with hyperbolic conservation laws in a general setting. However, Section 3.5 on the vacuum problem and Section 3.6 on source term integration might introduce some new concepts to readers familiar with hyperbolic conservation laws in other contexts. Chapter 4 provides a detailed look at the shallow water equations, specifically some analytical problems that will be important for later chapters. Chapter 5 provides an introduction into some aspects of numerical tsunami modeling. This chapter is intended to provide a background on tsunami modeling, as well as emphasize some of the difficulties that motivate later numerical techniques. Chapter 6 describes the approximate Riemann solver that is intended to overcome many of the difficulties of this application in a unified manner. Chapter 7 describes the adaptive mesh refinement techniques—specifically the modifications that were necessary for this application. Chapter 8 describes how these numerical methods can be generalized to modeling on the curved Earth. The following two chapters show various numerical results, meant to validate and legitimize the mathematical methods described. A comprehensive comparison of numerical experiments with field work data and empirical tsunami data is not shown here. However, as mentioned in the Chapter 11, extensive comparison with empirical tsunami data is one of the future goals for applying this work.

Chapter 2

HYPERBOLIC CONSERVATION LAWS AND RELATED SYSTEMS

This chapter introduces hyperbolic conservation laws and addresses the mathematical properties of these and related systems of equations before a discussion of numerical methods in Chapter 3. It begins with an introduction to the shallow water equations—a commonly accepted approximation governing tsunamis as well as other types of fluid flows.

2.1 The Shallow Water Equations

The shallow water equations are a system of PDEs for depth and momentum

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \tag{2.1a}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) + \frac{\partial}{\partial y}(huv) = -gh\frac{\partial b}{\partial x}, \tag{2.1b}$$

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}(\frac{1}{2}gh^2 + hv^2) = -gh\frac{\partial b}{\partial y}, \tag{2.1c}$$

where g is the gravitational constant, $h(x, y, t)$ is the nonnegative fluid depth, $u(x, y, t)$ is the fluid velocity in the x-direction, $v(x, y, t)$ is the fluid velocity in the y-direction and $b(x, y)$ is the elevation of the bottom surface (see Figure 2.1). The system (2.1) governs free surface flows where the fluid acceleration in the vertical z-direction is zero and horizontal velocities are uniform in the vertical direction. These simplifying assumptions will be discussed and the equations (2.1) derived in Chapter 4. In this chapter we will confine our attention to solutions to systems such as (2.1), and save any discussion regarding the relevance of the shallow water approximation until Chapter 4 and Chapter 5.

For theoretical examination, frequently it will be convenient to discuss the one dimensional form of the shallow water equations

$$h_t + (hu)_x = 0, \tag{2.2a}$$

$$(hu)_t + (hu^2 + \frac{1}{2}gh^2)_x = -ghb_x, \tag{2.2b}$$

where subscripts denote partial derivatives ($()_t \equiv \frac{\partial}{\partial t}$). If the bottom elevation is flat, $b_x \equiv 0$, (2.2) reduces to the homogeneous shallow water equations:

$$h_t + (hu)_x = 0, \tag{2.3a}$$

$$(hu)_t + (hu^2 + \frac{1}{2}gh^2)_x = 0. \tag{2.3b}$$

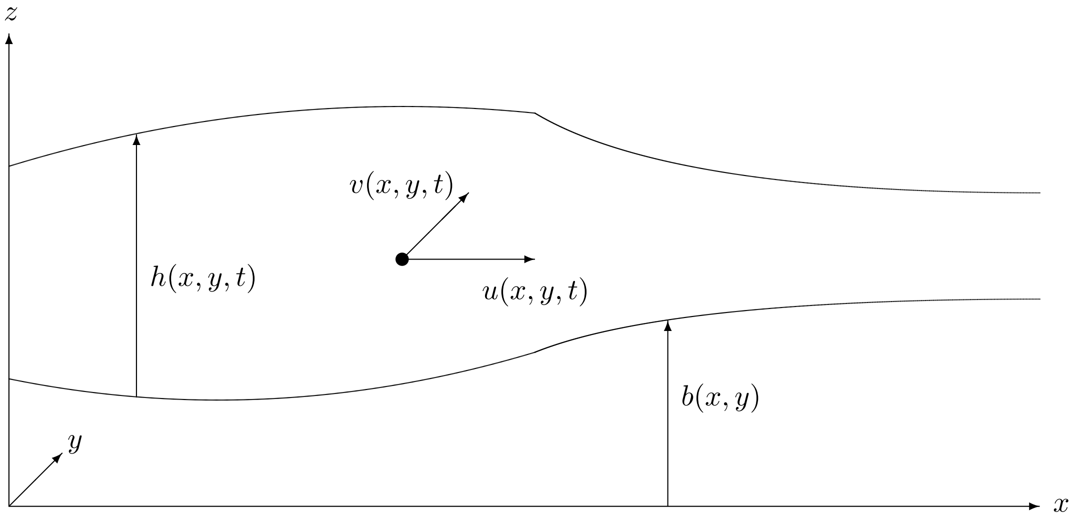


Figure 2.1: Cross section of a free surface flow governed by the shallow water equations (2.1). The fluid depth is $h(x, y, t)$, the fluid velocities are $u(x, y, t)$ and $v(x, y, t)$, and $b(x, y)$ is the bottom elevation. The moving free surface is at $z = h(x, y, t) + b(x, y)$.

2.2 Hyperbolic Conservation Laws

2.2.1 Conservation Laws

The shallow water equations belong to a more general class of conservation laws

$$\frac{\partial q(x,t)}{\partial t} = \sum_{j=1}^{n} \frac{\partial f_j(q)}{\partial x_j} = \psi(q,x), \quad x = (x_1, \ldots, x_n)^T \in \mathrm{I\!R}^n, \tag{2.4}$$

where $q \in \mathrm{I\!R}^m$ is a vector of m *state variables*, $f_j(q) \in \mathrm{I\!R}^m$ is a vector of the corresponding *fluxes* in the j^{th} direction, and $\psi(q, x) \in \mathrm{I\!R}^m$ is a vector of *source terms*. If $\psi(q, x)$ is nonzero, then (2.4) is often referred to as a *balance law*, however, we will use *conservation law* to refer to all such systems, and use *homogeneous conservation law* if we wish to emphasize that $\psi(q, x) \equiv 0$. The terminology follows from the fact that, if $\psi(q, x) \equiv 0$, the system (2.4) implies the conservation of the m state variables or *conserved quantities*. That is, in any region $\Omega \subseteq \mathrm{I\!R}^n$, the change in the amount of q in that region is due only to the flux through the boundary $\partial\Omega$, plus any contribution from a nonzero source term in the region. This can be seen by integrating (2.4) over Ω and applying the divergence theorem, which gives

$$\frac{d}{dt}\int_\Omega q\,dx = \sum_{j=1}^{n} \int_{\partial\Omega} f_j \omega_j\,dS = \int_\Omega \psi\,dx, \tag{2.5}$$

where $\omega = (\omega_1, \ldots, \omega_n)^T \in \mathrm{I\!R}^n$ is the unit outward normal vector on the boundary $\partial\Omega$.

2.2.2 Hyperbolic Conservation Laws

The system (2.4) is said to be *hyperbolic* if the flux functions satisfy certain properties.

Definition 2.1. Hyperbolicity. *Let*

$$\frac{\partial f_j}{\partial q}$$

denote the Jacobian matrix of the function $f_j(q)$, and consider the matrix

$$A(q,\omega) = \sum_{j=1}^{n} \omega_j \frac{\partial f_j}{\partial q}$$

where $\omega = (\omega_1, \ldots, \omega_n)^T$ is a unit vector. The system (2.4) is hyperbolic*, if $A(q,\omega)$ has m real eigenvalues $\lambda^p(q,\omega)$, $p = 1,\ldots,m$, and m linearly independent eigenvectors $r^p(q,\omega)$, $p = 1,\ldots,m$, for all $\omega \in \mathbb{R}^n$, $|\omega| = 1$. Further, the system (2.4) is* strictly hyperbolic *at a state q_0 if the eigenvalues $\lambda^p(q_0,\omega)$, $p = 1,\ldots,m$, are distinct.*

The integral conservation law (2.5) will also be referred to as hyperbolic, if the corresponding system of PDEs (2.4) is hyperbolic. Note that the matrix $A(q,\omega)$ is the Jacobian of the flux in the direction ω. The arbitrariness of ω implies that hyperbolicity is independent of any particular choice of coordinate directions.

The shallow water equations are hyperbolic conservation laws. As we will see, solutions to hyperbolic conservation laws will share certain properties, and by identifying the shallow water equations with this broader class of equations, we will be able to exploit the body of theory and numerical methods previously developed for these mathematically challenging systems.

2.3 Discontinuities and Weak Solutions

Although hyperbolic conservation laws were introduced as a system of PDEs (2.4) that implied the integral system (2.5), in most applications the integral system is derived from first principles and is the more fundamental governing system. The two are equivalent if the solution is continuously differentiable, however, the integral system (2.5) admits nonsmooth, even discontinuous solutions for which the PDEs are not defined in the classical sense. We seek solutions to the more fundamental integral system, which can be shown to be equivalent to *weak solutions* to (2.4)—more generalized nonclassical solutions in the sense of distributions. See, for intstance, [22] for a general treatment of distribution theory and generalized solutions to PDEs.

2.3.1 The Rankine-Hugoniot Jump Conditions

The weak solutions that arise in physical applications are characterized by regions of smooth variation separated by propagating jump discontinuities. We can therefore isolate weak solutions by requiring that they satisfy the PDEs where smooth, and meet appropriate jump conditions across discontinuities derived by the integral form. If we apply the integral conservation law (2.5) to a region around a jump discontinuity, it is easy to derive (*e.g.* [67, 32]) the following condition:

$$|s|\left(q^{+}-q^{-}\right)=\sum_{j=1}^{n}\hat{s}_j\left(f_j(q^{+})-f_j(q^{-})\right), \tag{2.6}$$

where $|s|$ is the instantaneous speed of the propagating discontinuity, $\hat{s}$ is a unit vector in the direction of the propagation, and the superscripts denote the limiting value of the solution on either side of the discontinuity. The condition (2.6) on discontinuous solutions is known as the Rankine-Hugoniot jump condition. It is easy to show that (2.6) is unaffected by a source term that is bounded in the neighborhood of the discontinuity.

For a one-dimensional system, (2.6) is simply

$$s\,[[q]]=[[f(q)]]\,, \tag{2.7}$$

where s is the speed of the discontinuity, and $[[\cdot]]$ denotes the jump in the quantity across the discontinuity.

2.3.2 Entropy Conditions

While uniqueness is guaranteed for smooth classical solutions to the hyperbolic PDEs (2.4), uniqueness of discontinuous solutions to the integral conservation laws (2.5) is not guaranteed, and additional admissability conditions based on physical considerations are required to isolate the physically relevant solution. These admissability conditions are called *entropy conditions*, owing to the Euler equations of gas dynamics—a hyperbolic system for which entropy plays this role. An *entropy function* $\nu(q)$ is an additional conserved quantity when the solution is smooth, obeying another conservation law

$$\nu(q)_t+\psi(q)_x=0, \tag{2.8}$$

where $\psi(q)$ is an *entropy flux*—some function based on physical considerations. For the shallow water equations, $\nu(q)$ is the mechanical energy, $\psi(q)$ is the energy flux, and (2.8) can be shown to be equivalent to the shallow water PDEs (2.2). For discontinuous solutions, (2.8) becomes an inequality based on the physical admissability considerations. For the shallow water equations, mechanical energy must decrease in a region containing a discontinuity, and the equality in (2.8) is replaced by an inequality, $\leq$, for more general solutions.

An alternative formulation of entropy conditions can be found for systems satisfying certain mathematical properties, of which the shallow water equations belong. That formulation will be more convenient for our purposes, and will be discussed further below.

2.3.3 Steepening Waves

It should be realized that the admissability of discontinuous solutions to the integral form of hyperbolic conservation laws is not merely a mathematical technicality. The existence of smooth initial conditions by no means implies that the solution to a hyperbolic system will remain smooth, and depending on the form of the flux function, various smooth initial data leads to steepening waves—eventually leading to a shock discontinuity. Therefore, for most applications, the existence of discontinuous solutions cannot be ignored by merely considering smooth initial data.

2.4 Linear and Quasilinear Hyperbolic Systems

2.4.1 One-Dimensional Linear Systems

Some basic properties of solutions to hyperbolic conservation laws are elucidated by considering the simple linear homogeneous case in one dimension, where the flux $f(q) = Aq$ implies the system

$$q_t + Aq_x = 0. \tag{2.9}$$

In order for (2.9) to be hyperbolic, the constant matrix $A \in \mathbb{R}^{m\times m}$ must have real eigenvalues $\lambda^1, \ldots, \lambda^m$, and a corresponding set of linearly independent eigenvectors $r^1, \ldots, r^m$. Since A is therefore diagonalizable, (2.9) is equivalent to the diagonal system

$$w_t + \Lambda w_x = 0, \tag{2.10}$$

where $w = R^{-1}q$, R is the matrix of eigenvectors, and $\Lambda = R^{-1}AR$ is the diagonal matrix of eigenvalues. Since (2.10) is a decoupled system of m scalar advection equations, we note that the initial profile of each characteristic variable w^p, $p = 1, \ldots, m$, is advected at the constant speed of the corresponding eigenvalue λ^p. (For a review of the solution to scalar advection equations, see for instance [67]). The characteristic curves of (2.9) in the x–t plane are therefore straight lines of unvarying slope, carrying the “weight” of each eigenvector at the speed of an associated eigenvalue.

2.4.2 Quasilinear Hyperbolic Systems

In general $f(q)$ is a nonlinear function of q, but we can write the one dimensional conservation law

$$q_t + f(q)_x = 0, \tag{2.11}$$

as a quasilinear system

$$q_t + f'(q)q_x = 0, \tag{2.12}$$

where $f'(q)$ denotes the Jacobian matrix of the flux. Although hyperbolicity implies that $f'(q)$ is diagonalizable, the eigenvalues and eigenvectors are generally functions of q. As with linear systems, however, all solution information is carried at the local speed of the eigenvalues of $f'(q)$. The characteristics associated with a particular eigenpair $\{\lambda^p, r^p\}$ will be referred to as the p^{th} *characteristic family* or p^{th} *characteristic field.*

The conservative system (2.12) belongs to a larger class of hyperbolic systems of PDEs

$$q_t + B(q, x)q_x = 0, \tag{2.13}$$

where $B(q,x) \in \mathbb{R}^{m\times m}$ is a nonlinear diagonalizable matrix with real eigenvalues. In general, however, $B(q,x)q_x$ need not be a full derivative of any function, and so (2.13) is not necessarily a conservation law. Nevertheless, many of the properties of solutions to (2.12) and the larger class of systems (2.13), are shared. For instance, the wave-like behavior of solutions to hyperbolic conservation laws is due to the fact that information propagates at the finite speed of the eigenvalues. This remains true for (2.13).

2.5 Riemann Problems

We will be interested in weak solutions to one-dimensional hyperbolic conservation laws, and hyperbolic systems generally, given a specific form of initial data that happens to be tractable to analytical solution techniques. Consider the one-dimensional system

$$q_t + f(q)_x = 0, \quad x \in \mathbb{R},\ t > 0, \tag{2.14a}$$

with piecewise constant initial conditions

$$q(x,0) = \begin{cases} q_l & \text{if} \quad x < 0, \\ q_r & \text{if} \quad x > 0. \end{cases} \tag{2.14b}$$

The initial value problem (2.14) is known as the *Riemann problem.*

2.5.1 Linear Waves

If $f(q)$ happens to be linear, we know that the initial profile of each characteristic variable w^p translates at the speed of the corresponding eigenvalue λ^p. This implies that the solution to (2.14) in the linear case is a set of constant states separated by m discontinuities propagating away from the origin at the speed of the eigenvalues. The jump $(q^+ - q^-)$ across the p^{th} discontinuity must be proportional to the p^{th} eigenvector, since it represents the initial jump in the characteristic variable, w^p, where $q = Rw$. The solution to the linear Riemann problem can therefore be determined by decomposing the initial jump into the eigenvectors r^p of $f'(q) = A$,

$$q_r - q_l = \sum_{p=1}^{m} \alpha^p r^p, \tag{2.15}$$

and noting that each discontinuity $\alpha^p r^p$ propagates at the speed λ^p. We will denote the discontinuity $\mathcal{W}^p = \alpha^p r^p$, and refer to it as the p^{th}*−wave.* Note that each wave satisfies the Rankine-Hugoniot conditions (2.7) for discontinuities since $f(q) = Aq$.

2.5.2 Nonlinear Waves

For the nonlinear case, the solution also consists of a set of transitions in each characteristic family that connect the states q_l and q_r. These transitions separate regions where the solution is constant, however, the transitions are not necessarily discontinuous. Three types of transitions can be distinguished: *rarefactions*, *shock waves*, and *contact discontinuities.*

A rarefaction is a type of smooth differentiable transitions in a particular characteristic field, that if parameterized, is an integral curve of the associated eigenvector in phase space. That is, the solution satisfies $q(x,t) = \tilde{q}(\xi(x,t))$ throughout the wave, where

$$\tilde{q}'(\xi) = \alpha(\xi) r^p(\xi), \tag{2.16}$$

for some function $\xi(x,t)$, where $\alpha(\xi)$ is some scalar depending on the particular function ξ. A solution with such properties is known as a *simple wave*, of which, a rarefaction is one type arising in Riemann problems. Shock waves and contact discontinuities are both discontinuities satisfying (2.7). In order to distinguish the two, the following definitions are required:

Definition 2.2. *The p^{th} characteristic field is called genuinely nonlinear if*

$$\nabla\lambda^p(q) \cdot r^p(q) \neq 0, \quad \forall q \in \Omega. \tag{2.17a}$$

Alternatively, the p^{th} characteristic field is called linearly degenerate if

$$\nabla\lambda^p(q) \cdot r^p(q) \equiv 0, \quad \forall q \in \Omega. \tag{2.17b}$$

Shock waves are discontinuities in a genuinely nonlinear field. A contact discontinuity conversely, is a jump in a linearly degenerate field. If a field is linearly degenerate, the eigenvalue would be unchanged through any simple wave satisfying (2.16), because of (2.17b). The field is similar to a linear field then, in the sense that the characteristic curves have a constant slope through any transition that remains proportional to the p^{th} eigenvector. A contact discontinuity in the p^{th} field may be thought of as a jump from one point to another on an integral curve of r^p. In this case, $s = \lambda^p(q^+) = \lambda^p(q^-)$, where s is the speed of the discontinuity. For each of the three types of waves, the characteristics of the corresponding field are shown in Figure 2.2.

It can be shown (*e.g.* [32]) that the Riemann problem (2.14) has a unique entropy satisfying weak solution if all of the m characteristic fields are either genuinely nonlinear or linearly degenerate, provided that $|q_r - q_l|$ is sufficiently small. Furthermore, the solution is a similarity solution depending only on (x/t), where $(m+1)$ constant states are separated by waves—shock waves, rarefactions or contact discontinuities. Solutions to a linear and nonlinear Riemann problem with $m = 2$ are depicted in Figure 2.3.

2.5.3 Entropy

For strictly hyperbolic conservation laws where each field is genuinely nonlinear, an alternative formulation of the entropy condition holds that does not explicitly require an entropy function (2.8). This is known as the *Lax entropy condition*, and the following definition is borrowed from [67] .

Definition 2.3. Lax entropy condition. *A discontinuity separating states q_l and q_r, propagating at speed s, satisfies the Lax entropy condition if there is an index p such that*

$$\lambda^p(q_l) > s > \lambda^p(q_r), \tag{2.18a}$$

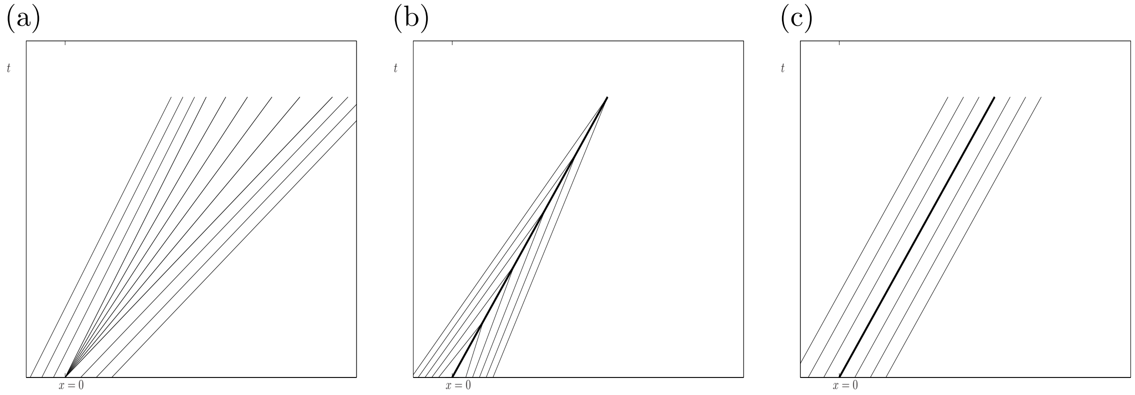


Figure 2.2: Characteristics of the three wave-types in Riemann solutions are shown in the x–t plane. (a) A rarefaction in the p^{th} field: the p-characteristics emanating from $x = 0$ move at a speed x/t. If the solution is parameterized through the rarefaction, it smoothly varies along an integral curve of r^p. (b) A shock in the p^{th} field: the p-characteristic impinge on the shock according to the Lax entropy condition (Definition 2.3). (c) A contact discontinuity in the p^{th} field: the discontinuity and p-characteristics on either side of the discontinuity all propagate at the same speed.

so that p-characteristics are impinging on the discontinuity, while the other characteristics are crossing the discontinuity,

$$\lambda^j(q_l) < s \quad \textit{and} \quad \lambda^j(q_r) < s \qquad \textit{for} \quad j < p, \tag{2.18b}$$

$$\lambda^j(q_l) > s \quad \textit{and} \quad \lambda^j(q_r) > s \qquad \textit{for} \quad j > p, \tag{2.18c}$$

where, in this definition, the eigenvalues are ordered so that $\lambda^1 < \lambda^2 < \cdots < \lambda^m$ in each state.

2.5.4 The Shallow Water Riemann Problem and Characteristic Structure

For the homogeneous shallow water equations in one dimension (2.3), we have

$$q = \begin{bmatrix} h \\ hu \end{bmatrix}, \qquad f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}. \tag{2.19}$$

(The flux and the following quantities derived from it will frequently be denoted as functions of the conserved variable q, but their formulas will be written in terms of the more familiar variables, h and u.) The Jacobian matrix (differentiated with respect to q) for the flux is

$$f'(q) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}, \tag{2.20}$$

which has eigenvalues

$$\lambda^1(q) = u - \sqrt{gh}, \qquad \lambda^2(q) = u + \sqrt{gh}, \tag{2.21}$$

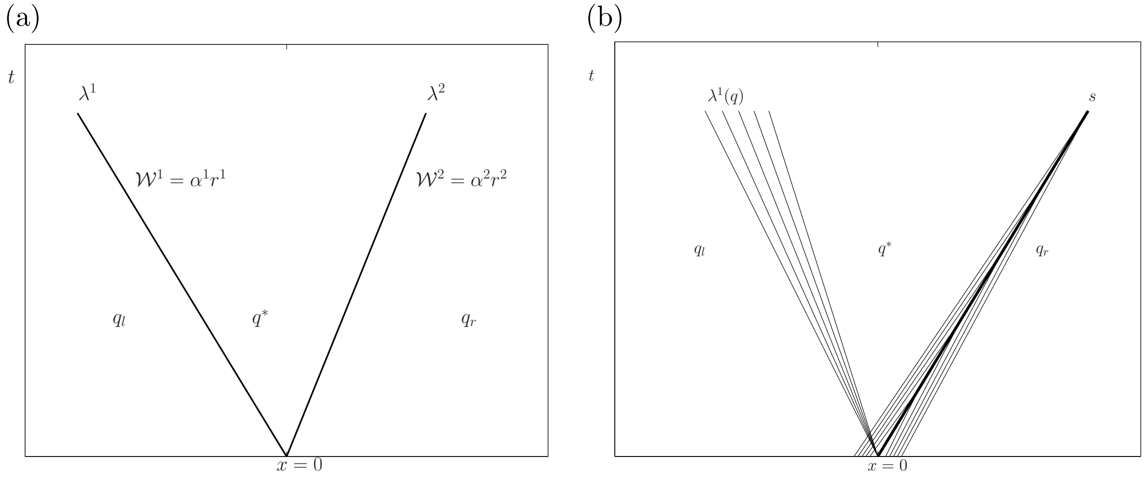


Figure 2.3: (a) The solution to a Riemann problem for a linear system in the x–t plane. The solution consists of m *waves*—each a jump discontinuity $\mathcal{W}^p$ proportional to the eigenvector r^p of A, moving at the speed of the corresponding eigenvalue λ^p. (b) An example of a solution to a nonlinear Riemann problem in the x–t plane showing each of the waves or characteristic transitions. The example shows 1-characteristics for a rarefaction in the first field, and shows a propagating discontinuity—a shock in the second field. In both (a) and (b) regions where the solution is constant are separated by the moving waves. For $m = 2$, there is a single constant middle state denoted q^*.

and corresponding eigenvectors

$$r^1(q) = \begin{bmatrix} 1 \\ \lambda^1 \end{bmatrix}, \qquad r^2(q) = \begin{bmatrix} 1 \\ \lambda^2 \end{bmatrix}. \tag{2.22}$$

It is easy to show that both characteristic fields are genuinely nonlinear. Therefore, the Riemann problem must consist of two waves—each wave being either a rarefaction or a shock—separating a single constant middle state from the adjacent state, q_l or q_r. The single constant middle state will be denoted q^*, and the region it occupies in the x–t plane will occasionally be referred to as the *middle region* (see Figure 2.3). The unique entropy satisfying weak solution to the Riemann problem for the shallow water equations can be determined, though this entails nonlinear root finding. This will be described in more detail in Chapter 4 (see also [64, 101]). Several concepts that help elucidate this process will be briefly introduced below.

It is possible to show that an integral curve in the phase plane of a genuinely nonlinear field corresponds to a constant contour of a function of q (*e.g.* [64, 32]). These functions are called *Riemann invariants* since the value of the function is invariant along an integral curve. For the shallow water equations, the Riemann invariant of the first and second field are

$$w^1(q) = u + 2\sqrt{gh}, \tag{2.23a}$$

$$w^2(q) = u - 2\sqrt{gh}. \tag{2.23b}$$

If a rarefaction connects the left state to the middle state, then $w^1(q_l) = w^1(q^*)$, and for a right state connected through a rarefaction, $w^2(q_r) = w^2(q^*)$.

It turns out that shocks can also be represented as smooth curves in phase space that connect two states. Recall that a discontinuous shock must satisfy the Rankine-Hugoniot jump conditions (2.7). If we consider the state on one side of the shock as a given, these conditions comprise m equations in $(m+1)$ variables—the m unknown state variables plus the speed of the shock s. It is therefore potentially possible to eliminate s and relate each state variable through a single parameter. The corresponding curve in phase space represents states that may be connected through a shock, and this curve, or the set of states on it, is known as a *Hugoniot locus*. In practice, multiple solutions for such parameterizations appear. For the shallow water equations, two such curves appear at a given point in phase space yet it is possible to identify each curve with a shock in one of the respective characteristic families. If we parameterize a curve by the depth h, then the Hugoniot loci for each family as a function of the velocity are

$$u(h) = u_l - (h - h_l)\sqrt{\frac{g}{2}\left(\frac{1}{h} + \frac{1}{h_l}\right)}, \tag{2.24a}$$

$$u(h) = u_r + (h - h_r)\sqrt{\frac{g}{2}\left(\frac{1}{h} + \frac{1}{h_r}\right)}. \tag{2.24b}$$

The curve defined by (2.24a) represents states that may be connected to q_l through a shock in the first family, or a *1-shock*, and (2.24b) represents states that may be connected to q_r through a *2-shock*. See, for example, [67] for details on determining (2.24).

If we know the type of wave that exists in each family, the middle state q^* is determined by the intersection of the appropriate curve in each respective family. This is explained in more detail in Chapter 4. Note that this is fundamentally the same as for a linear Riemann problem with $m = 2$, yet, in that case the curves are the eigenvectors in phase space. Figure 2.4 shows integral curves and Hugoniot loci for the shallow water equations.

2.5.5 Higher-Dimensional Normal Riemann Problems

Although we will only consider the Riemann problem in one dimension, it is possible to solve one-dimensional Riemann problems of higher dimensional systems. This problem naturally arises if one considers a higher-dimensional system with initial conditions that only vary in one direction. For instance, for the two-dimensional homogeneous shallow water equations, which we will write in the general form

$$q_t + f(q)_x + g(q)_y = 0, \tag{2.25}$$

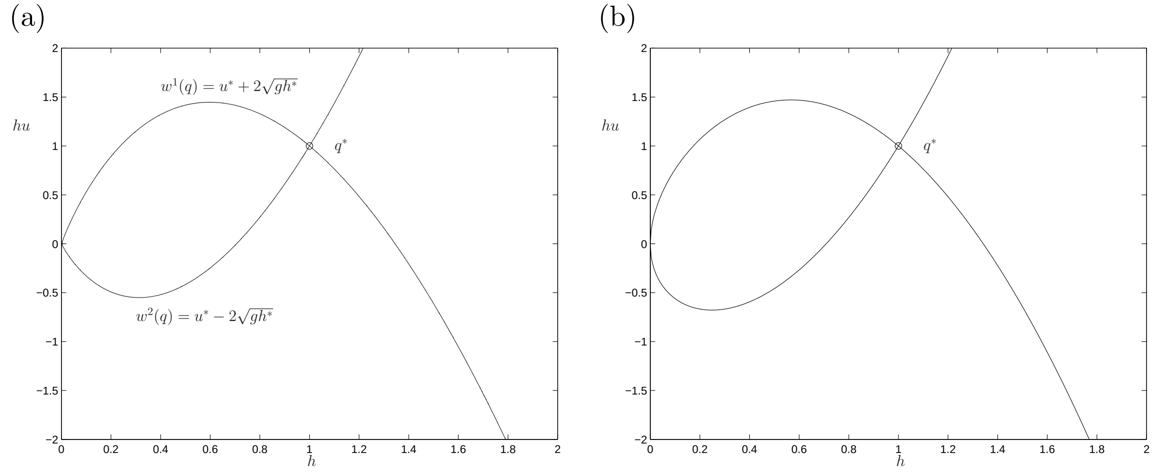


Figure 2.4: (a) Integral curves of the shallow water equations: each curve is tangent to one of the eigenvectors at each point in the phase plane and corresponds to a constant of a Riemann invariant $w^p(q)$. (b) Hugoniot loci of the shallow water equations: each curve represents states that can be connected through a shock in the 1st or 2nd family.

we consider the Riemann problem (2.14), with

$$q = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ hvu \end{bmatrix}. \tag{2.26}$$

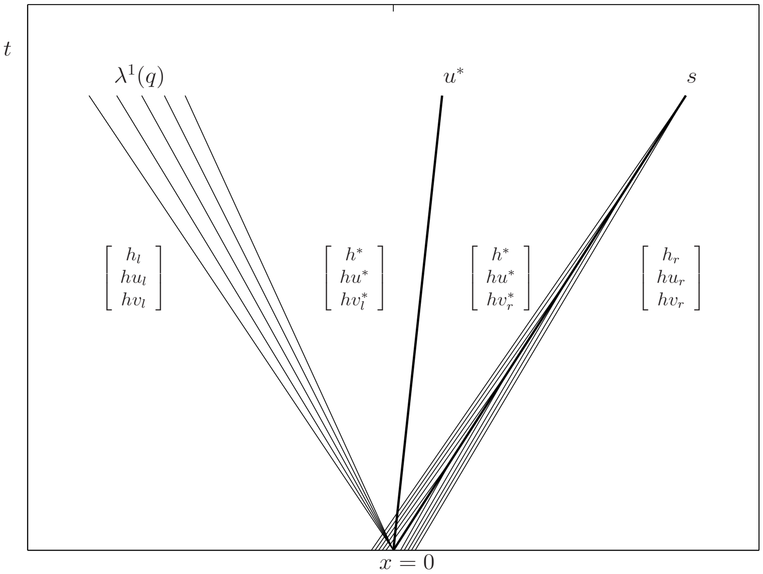
Since now $m = 3$, we expect the Riemann solution to contain an additional wave. The eigenvalues of $f'(q)$ are

$$\lambda^1(q) = u - \sqrt{gh}, \qquad \lambda^2(q) = u, \qquad \lambda^3(q) = u + \sqrt{gh}, \tag{2.27}$$

and corresponding eigenvectors

$$r^1(q) = \begin{bmatrix} 1 \\ \lambda^1 \\ v \end{bmatrix}, \qquad r^2(q) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad r^3(q) = \begin{bmatrix} 1 \\ \lambda^3 \\ v \end{bmatrix}. \tag{2.28}$$

Considering this eigenstructure, it is apparent that the solution to the Riemann problem for the first two variables, h and hu, is identical to that for the one-dimensional system. The new second field is linearly degenerate, and a contact discontinuity, moving at the intermediate speed of u^*, separates the fluid that was initially in contact at $x = 0$. On either side of the discontinuity the initial transverse velocities remain. Therefore, the transverse velocity can be thought of as a passive tracer that has no effect on the dynamics of the Riemann solution. For more details see [67] . The Riemann solution in the x–t plane is depicted in Figure 2.5.



t
$\lambda^1(q)$
u^*
s
$\begin{bmatrix} h_l \\ hu_l \\ hv_l \end{bmatrix}$
$\begin{bmatrix} h^* \\ hu^* \\ hv_l^* \end{bmatrix}$
$\begin{bmatrix} h^* \\ hu^* \\ hv_r^* \end{bmatrix}$
$\begin{bmatrix} h_r \\ hu_r \\ hv_r \end{bmatrix}$
$x = 0$

Figure 2.5: A solution to a one-dimensional normal Riemann problem of the two-dimensional shallow water equations. The example shown has 3 waves—a 1-rarefaction and a 3-shock, and the ubiquitous contact discontinuity in the 2^{nd} field, separating the fluid that was initially in contact. Only v varies across the contact discontinuity: h^* and hu^* are identical to the solution for the one-dimensional system. The speed of the propagating waves are shown above them.

2.6 Balance Laws and Source Terms

For most of this chapter we have neglected source terms, such as the one due to variable bathymetry for the shallow water equations. In general it is not possible to solve a Riemann problem exactly when it includes a source term. However, the characteristic structure of a hyperbolic conservation law due to the eigenstructure of the flux Jacobian is unchanged by the existence of a source term. The complication occurs since the solution information carried on a characteristic is literally changed by the source term.

If we consider a linear conservation law with a source term

$$q_t + Aq_x = \psi, \tag{2.29}$$

we can still decouple the system into the characteristic variables $w = R^{-1}q$, which leads to the following system of decoupled scalar advection equations with source terms:

$$w_t + \Lambda w_x = R^{-1}\psi. \tag{2.30}$$

The characteristics are straight lines in the x–t plane as before, carrying the weight w^p of the p^{th} eigenvector at the constant speed of the eigenvalue λ^p. Along a characteristic curve,

$X(t) = x_0 + \lambda^p t$, each component obeys the ODE

$$\frac{d}{dt}(w^p(X(t), t)) = (R^{-1}\psi)^p, \tag{2.31}$$

suggesting that the decomposed source term is carried on characteristics as well. For instance, if the source term ψ happens to be proportional to an eigenvector r^p, then it affects only the w^p characteristic variable.

For a nonlinear system, the effect of the source term is considerably more complex since the eigenvalues and eigenvectors are functions of the solution. The solution to a Riemann problem with a source term would not, in general, be a similarity solution—a function of (x/t) only—since the characteristics can curve with the changing eigenvalues. Similarly, if the source term is bounded, the instantaneous speed of a shock still obeys the Rankine-Hugoniot conditions, however, the value of the solution on either side of a shock can change due to the source term, curving the shock in the x–t plane.

An alternative view of the effect of a source term is provided by considering a concentrated point source. Suppose that the source term acts at a single point, $\psi(q, x) = \Psi\delta(x)$, where $\delta(x)$ is the delta function defined by the functional

$$f(0) = \int f(x)\delta(x)\, dx. \tag{2.32}$$

By applying conservation to a vanishingly small neighborhood around $x = 0$, it is easy to show (*e.g.* [67]) that the flux must satisfy

$$f(q(0^+, t)) - f(q(0^-, t)) = \Psi. \tag{2.33}$$

Away from the concentrated source the conservation law is essentially homogeneous, since $\psi(q, x) = 0$. The solution to a Riemann problem with a source term at the initial discontinuity is therefore a similarity solution. Like the homogeneous conservation law, a set of waves propagate away from the interface into regions with no source term, yet, a contact discontinuity remains at $x = 0$ where the flux obeys (2.33). This view will help elucidate numerical methods where the source term is idealized as a point source $\psi(q, x) = \Psi\Delta x\delta(x)$, where Δx is some interval that the point source "acts for." That is, $\Psi \approx \psi(q(0, t), 0)$ so that $\int_{x_1}^{x_2} \psi(q, x)\, dx \approx \Psi\Delta x$, where $\Delta x = (x_r - x_l)$.

Chapter 3

FINITE VOLUME METHODS AND WAVE PROPAGATION ALGORITHMS

This chapter gives an overview of a class of numerical methods developed for hyperbolic conservation laws (2.4). Originally, these methods were developed largely in the context of solving the Euler equations of compressible gas dynamics, however, they are generally applicable to other systems such as the shallow water equations. Additionally, we will consider *wave propagation algorithms*—finite volume methods that are applicable to more general hyperbolic systems (2.13). Ultimately, as will be described in this and later chapters, the more general wave propagation algorithms will be used to solve the shallow water equations. This chapter describes the basic methods—modifications specific to tsunami modeling will be taken up in later chapters.

3.1 Finite Volume Methods for Conservation Laws

For numerically solving systems that obey integral conservation laws, finite volume methods provide a natural intuitive approach. In this approach, the numerical solution can be viewed as a piecewise constant function in space rather than a set of values at discrete points. The spatial domain is divided into a set of *grid cells* where the numerical solution has a given value at each time step. In one dimension, the grid cells are simply intervals $\mathcal{C}_i = [x_{i-1/2}, x_{i+1/2}]$ of length Δx, around the point $x_i = x_0 + i\Delta x$, where $x_{i\pm1/2} = x_i \pm \frac{1}{2}\Delta x$, $i \in \mathbb{Z}_c \subseteq \mathbb{Z}$. (For simplicity we will assume that the intervals are all of equal length Δx, however, this is not strictly necessary.) The numerical solution $Q_i^n \in \mathrm{I\!R}^m$ is then an approximation to the average value of the true solution in the grid cell

$$Q_i^n \approx \frac{1}{\Delta x}\int_{\mathcal{C}_i} q(x,t^n)\,dx. \tag{3.1}$$

(For analysis of smooth solutions, note that $Q_i^n \approx q(x_i,t^n)$.) If we wish to approximate a homogeneous conservation law, the numerical solution can be updated from time t^n to t^{n+1}, with $\Delta t = t^{n+1} - t^n$, by the explicit *flux-differencing scheme*

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[F_{i+1/2}^n - F_{i-1/2}^n\right], \tag{3.2}$$

where $F_{i\pm1/2}^n$ are the *numerical fluxes*—approximations to the fluxes at the grid cell boundaries. Note that (3.2) is a direct approximation to the integral conservation law applied to the domain $\mathcal{C}_i$, which integrated from time t^n to t^{n+1} and divided by Δx gives

$$\begin{aligned}\frac{1}{\Delta x}\int_{\mathcal{C}_i} q(x,t^{n+1})\,dx = \frac{1}{\Delta x}&\int_{\mathcal{C}_i} q(x,t^n)\,dx \\ &- \frac{1}{\Delta x}\left[\int_{t^n}^{t^{n+1}} f(q(x_{i+1/2},t))\,dt - \int_{t^n}^{t^{n+1}} f(q(x_{i-1/2},t))\,dt\right]. \end{aligned} \tag{3.3}$$

Considering (3.1) and (3.3), note that the numerical fluxes in (3.2) should be approximations to the time averaged fluxes at the cell boundaries:

$$F^n_{i\pm1/2} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(q(x_{i\pm1/2},t))\,dt. \tag{3.4}$$

Because (3.3) is exact, the accuracy and stability of the numerical method (3.2) is determined by the approximation (3.4). Developing an explicit flux differencing scheme of the form (3.2), therefore, entails developing consistent and stable numerical fluxes.

Note that (3.2) is a *conservative numerical scheme* regardless of the form of $F^n_{i\pm1/2}$. That is, the numerical solution is conserved on the computing domain (aside from any contribution from the boundaries), since the same $F^n_{i-1/2}$ is used to update Q^n_{i-1} and Q^n_i. This implies that whatever quantity enters one grid cell must leave another, as with the true conservation law. It turns out that numerical conservation is a necessary condition for converging to the correct discontinuous solutions to conservation laws [47].

3.2 Godunov-Type Methods

A class of methods that provide a stable and consistent approximation to the numerical fluxes (3.4), are the Godunov-type methods. The original Godunov method [33], is a first-order upwind-type scheme, where Riemann problems are solved at each grid cell interface before each time step to determine the numerical flux for that time step. That is, at time t^n, $\forall i \in \mathbb{Z}_c$, we solve the Riemann problem (2.14), with data $q_l = Q^n_{i-1}$ and $q_r = Q^n_i$. If we denote the similarity solution to the Riemann problem, described in the previous chapter, as $\tilde{q}(x/t, q_l, q_r)$, then our numerical flux $F^n_{i-1/2}$ becomes $f(\tilde{q}(0, Q^n_{i-1}, Q^n_i))$—the actual flux function evaluated with the solution to the Riemann problem at $x = 0$. At first glance, solving the Riemann problem seems like an overly involved procedure for estimating the flux (3.4), however, it turns out that it provides several essential properties. First, the explicit time update in (3.2) requires proper upwinding of the flux estimate for stability. For a first order method, the Riemann solution provides the proper upwinding, even when there are m characteristic curves. Second, the Godunov method converges to entropy satisfying discontinous weak solutions, provided that the Riemann solutions satisfy entropy conditions. The standard Godunov method is only first order accurate but standard second order methods, based on the validity of Taylor series, break down near discontinuities. If fact, it can be shown that no linear second order method can converge to discontinuous solutions in general (where linearity is in the sense of the abstract updating function $Q^{n+1}_i = \mathcal{L}(\Delta x, \Delta t, Q^n_1, \dots, Q^n_i, \dots)$). See, for example, [32, 67, 100] for a full discussion of these topics. Later in this chapter we will discuss the extension of the standard Godunov method to higher order methods for smooth solutions that maintain the ability to converge to discontinuities.

In practice, it is often sufficient to approximate the solution to the Riemann problem, and various *approximate Riemann solvers* have been developed specifically for Godunov-type methods. Some standard approaches will be discussed in the next section.

3.3 Approximate Riemann Solvers

Approximate Riemann solvers are typically based on using a solution to some related linear problem at each grid cell interface that approximates the true nonlinear Riemann solution. Recall that a solution to a linear Riemann problem consists of a set of discontinuities or *waves* propagating at speeds equal to the eigenvalues of the constant Jacobian. All of the approximate solvers discussed in this section will be of that same basic form. That is, the approximate solutions consist of a set of M_w waves, $\mathcal{W}^p \in \mathbb{R}^m$, propagating at corresponding speeds s^p, $p = 1, \ldots, M_w$, where

$$q_r - q_l = \sum_{p=1}^{M_w} \mathcal{W}^p. \tag{3.5}$$

The new notation M_w is used for the number of waves, and s^p for the speeds, since, for an approximate solution, it is not strictly necessary that $M_w = m$, nor that s^p correspond to actual eigenvalues λ^p of some Jacobian.

One approach for creating an approximating linear solution is to use a local linearization of the Jacobian, where $f'(q)$ is replaced with a constant matrix $A(q_l, q_r)$. The resulting linear Riemann problem is then solved to determine the numerical flux. In order to maintain hyperbolicity, $A(q_l, q_r)$ must have real eigenvalues and a full set of linear independent eigenvectors. Additionally, it must be consistent in the following sense: $A(q_l, q_r) \to f'(q)$, as $q_l, q_r \to q$.

If the approximate solution is conservative, that is, if it obeys the conservation law when applied to a region surrounding the Riemann solution, it must satisfy

$$f(q_r) - f(q_l) = \sum_{p=1}^{M_w} s^p \mathcal{W}^p. \tag{3.6}$$

The right hand side represents the time rate of change of the solution due to the moving jump discontinuities $\mathcal{W}^p$, and the left hand side represents the flux evaluated at the far sides of the changing region. Not all linearizations will satisfy (3.6).

3.3.1 The Roe Solver

A particular linearization that is endowed with several nice properties is the *Roe linearization*, or the *Roe solver.* This was originally developed for the Euler equations by Roe [82], but it has since been extended to the shallow water equations. The Roe solver is based on evaluating the flux Jacobian $f'(q)$ with a special average of q_r and q_l in the Riemann problem. If we denote this matrix $\hat{A}$, we have

$$\hat{A}(q_l, q_r) = f'(\hat{q}(q_l, q_r)), \tag{3.7}$$

where $\hat{q}(q_l, q_r)$ denotes the special Roe average of q_l and q_r. Because the Roe matrix is determined by evaluating the true flux Jacobian with a valid state $\hat{q}$, all of the essential hyperbolic eigenstructure is maintained. The essential and unique property of the average $\hat{q}$ is that the resulting linear Riemann solution is conservative with respect to the original flux. That is, the following property is satisfied:

$$f(q_r) - f(q_l) = \hat{A}(q_l, q_r)\,(q_r - q_l)\,. \tag{3.8}$$

By considering a decomposition of $(q_r - q_l)$ into the eigenvectors of $\hat{A}(q_l, q_r)$, it is easy to see that if (3.8) is satisfied, the solution to a linear Riemann problem with Jacobian $\hat{A}$ satisfies the conservation condition (3.6). The actual Roe average $\hat{q}(q_r, q_l)$ depends on the particular system of equations. For the shallow water equations,

$$\hat{h}(q_l, q_r) = \frac{1}{2}\left(h_l + h_r\right), \tag{3.9a}$$

$$\hat{u}(q_l, q_r) = \frac{u_l\sqrt{h_l} + u_r\sqrt{h_r}}{\sqrt{h_l} + \sqrt{h_r}}, \tag{3.9b}$$

where, as usual, the formulas have been written in terms of the more familiar variables h and u rather than q. Roe averages and the functions evaluated with them will be denoted with a hat throughout this thesis.

A beneficial feature of the Roe solver is the approximate solution that it produces in the neighborhood of a shock. Suppose that the true solution to the Riemann problem is a single shock wave propagating at speed s satisfying the Rankine-Hugoniot conditions. The Roe solver produces the exact solution in this case, following from its conservative properties:

$$s(q_r - q_l) = f(q_r) - f(q_l) = \hat{A}(q_l, q_r)(q_r - q_l). \tag{3.10}$$

The first equality in (3.10) is the Rankine-Hugoniot condition governing the true solution, and the second equality follows from (3.8). Recall that the solution to a linear problem can be determined by decomposing the initial jump $(q_r - q_l)$ into the eigenvectors of the Jacobian and translating the resulting waves at the speed of the corresponding eigenvalues. Since (3.10) implies that $(q_r - q_l)$ is an eigenvector of $\hat{A}(q_l, q_r)$ with eigenvalue s, clearly the Roe solver produces the exact solution in this case. This can also be interpreted in the following way: if the data q_l and q_r lie on the same Hugoniot locus of the p^{th} characteristic family, the vector $(q_r - q_l)$ is the eigenvector $\hat{r}^p$ of $\hat{A}(q_l, q_r)$.

3.3.2 The HLL and HLLE solvers

The HLL solver, introduced by Harten, Lax and Van Leer [42], provides a simple conservative alternative to the Roe linearization. The HLL solver does not use an explicit linearization of the Jacobian, but rather constructs a conservative solution by requiring that two discontinuities propagate at predetermined speeds, s^1 and s^2, $s^2 > s^1$, where the speeds are based somehow on q_l and q_r. (The solver actually uses two speeds even when the system has $m > 2$ equations.) The middle state q^* between the two discontinuities is then determined by applying the conservation law to the region

$$s^1(q^* - q_l) + s^2(q_r - q^*) = f(q_r) - f(q_l), \tag{3.11a}$$

$$\Rightarrow q^* = \frac{1}{s^1 - s^2}\left(s^1 q_l - s^2 q_r + f(q_r) - f(q_l)\right). \tag{3.11b}$$

The properties of the HLL solver are strongly tied to the pre-chosen speeds s^1 and s^2. For the shallow water equations, where $m = 2$, if s^1 and s^2 correspond to the Roe eigenvalues—the eigenvalues of $f'(q)$ evaluated at $\hat{q}$—then the HLL solver is identical to the Roe solver, which follows from conservation.

The HLLE solver is an HLL type of solver that arises when a particular choice of speeds are used:

$$s^1 = \min_p \min(\lambda^p(q_l), \hat{\lambda}^p(q_l, q_r)), \tag{3.12a}$$

$$s^2 = \max_p \max(\lambda^p(q_r), \hat{\lambda}^p(q_l, q_r)), \tag{3.12b}$$

where $\hat{\lambda}(q_l, q_r)^p$ are the Roe eigenvalues, or the *Roe speeds*. The speeds (3.12) were advocated by Einfeldt [20], and we will refer to them as the *Einfeldt speeds*, and denote them $s^1_{\mathcal{E}}$ and $s^2_{\mathcal{E}}$. For the shallow water equations, it is easy to show that (3.13) is simply

$$s^1_{\mathcal{E}} = \min(\lambda^1(q_l), \hat{\lambda}^1(q_l, q_r)), \tag{3.13a}$$

$$s^2_{\mathcal{E}} = \max(\lambda^2(q_r), \hat{\lambda}^2(q_l, q_r)). \tag{3.13b}$$

The Einfeldt speeds endow the approximate solver for the shallow water equations with some special properties that will be discussed further below and in Section 3.5. See also [20, 21].

3.3.3 Entropy Fixes

As mentioned previously, the Godunov-type methods converge to the correct entropy satisfying discontinuous solutions, provided that the Riemann solutions are the correct entropy satisfying weak solutions. Since approximate Riemann solvers are discontinuous, even when the true solution is a smooth rarefaction, they can generate convergence to discontinuous entropy violating solutions. This well known problem occurs when the true solution to a Riemann problem contains a transonic rarefaction—a rarefaction in the p^{th} field where λ^p passes through zero. In this case the true solution has a wave spreading out in both directions. Since an approximate solver uses a single discontinuity to represent this rarefaction, it allows convergence to the pathological stationary *expansion shock*—a stationary discontinuity at a cell interface that violates entropy. The Roe solver is prone to this problem, however, numerous entropy fixes have been developed for the Roe solver (*e.g.* [19, 40, 41]).

The HLLE method provides a natural fix to this problem, since, in the case of a transonic rarefaction, the speeds (3.12) bound the true speeds of the spreading rarefaction.

3.4 Wave Propagation Algorithms

The standard Godunov-methods are flux differencing methods (3.2), where exact or approximate solvers are used to determine a numerical flux (3.4), sometimes called an *interface flux.* These methods are conservative regardless of the form of the approximate Riemann solution. Alternatively, the structure of an approximate Riemann solution can be used directly to update the numerical solution. That is, the effect of moving waves $\mathcal{W}^p \in \mathbb{R}^m$ can be directly re-averaged onto the computational grid cells. This is the idea behind the wave propagation algorithms, such as those developed by LeVeque [65, 67]. For instance, an approximate Riemann solution at $x_{i-1/2}$ with data $q_l = Q_{i-1}$ and $q_r = Q_i$, produces a set of waves:

$$Q_i - Q_{i-1} = \sum_{p=1}^{M_w} \mathcal{W}^p_{i-1/2}, \tag{3.14}$$

and a corresponding set of speeds $s^p_{i-1/2}$, $p = 1, \ldots, M_w$. If these waves are re-averaged onto the adjacent grid cells, they produce a change of the average value to the right

$$Q_i^{n+1} - Q_i^n = -\frac{\Delta t}{\Delta x} \sum_{\left\{p:\, s^p_{i-1/2}>0\right\}} s^p_{i-1/2} \mathcal{W}^p_{i-1/2}, \tag{3.15}$$

and to the left

$$Q_{i-1}^{n+1} - Q_{i-1}^n = -\frac{\Delta t}{\Delta x} \sum_{\left\{p:\, s^p_{i-1/2}<0\right\}} s^p_{i-1/2} \mathcal{W}^p_{i-1/2}. \tag{3.16}$$

The net effect of the two Riemann problems on the boundaries of $\mathcal{C}_i$ is then

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left[\sum_{\left\{p:\, s^p_{i-1/2}>0\right\}} s^p_{i-1/2} \mathcal{W}^p_{i-1/2} + \sum_{\left\{p:\, s^p_{i+1/2}<0\right\}} s^p_{i+1/2} \mathcal{W}^p_{i+1/2} \right]. \tag{3.17}$$

Adopting the notation from [67] , we will denote the sum in (3.15) as $\mathcal{A}^+\Delta Q_{i-1/2}$ and the sum in (3.16) as $\mathcal{A}^-\Delta Q_{i-1/2}$, and refer to them as *fluctuations.* The update (3.17) can then be written as

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2} \right). \tag{3.18}$$

The wave propagation method is more generally applicable than the flux differencing approach (3.2), since (3.18) can be applied to non-conservative hyperbolic systems of the form (2.13). In fact, (3.18) is a generally valid approach even when the Riemann solution has a discontinuity at $x = 0$, or in the case of the numerical update, at a cell interface. If (3.18) is used for a conservation law, the following property is required to maintain conservation

$$f(Q_i) - f(Q_{i-1}) = \mathcal{A}^-\Delta Q_{i-1/2} + \mathcal{A}^+\Delta Q_{i-1/2}. \tag{3.19}$$

Note that if approximate solvers of the form described in Section 3.3 are used, (3.19) is equivalent to the requirement

$$f(Q_i) - f(Q_{i-1}) = \sum_{p=1}^{M_w} s^p_{i-1/2}\mathcal{W}^p_{i-1/2}, \tag{3.20}$$

which is equivalent to (3.6)—the condition that an approximate solution obeys conservation.

3.4.1 The f-wave Method

Decomposing the solution at a grid cell interface (3.14), and then requiring (3.20), suggests a simpler approach where the jump in the flux is simply decomposed into some set of vectors,

$$f(Q_i) - f(Q_{i-1}) = \sum_{p=1}^{M_w} \mathcal{Z}^p_{i-1/2}, \tag{3.21}$$

where $\mathcal{Z}^p_{i-1/2} \in \mathbb{R}^m$ is called an *f-wave*, since it carries the same dimension as the flux, like $s^p_{i-1/2}\mathcal{W}^p_{i-1/2}$ (see [6]). As before, some set of speeds $s^p_{i-1/2}$, $p = 1, \ldots, M_w$ are associated with the waves. The fluctuations can then be defined by

$$\mathcal{A}^-\Delta Q_{i-1/2} = \sum_{\left\{p:\, s^p_{i-1/2}<0\right\}} \mathcal{Z}^p_{i-1/2}, \tag{3.22a}$$

$$\mathcal{A}^+\Delta Q_{i-1/2} = \sum_{\left\{p:\, s^p_{i-1/2}>0\right\}} \mathcal{Z}^p_{i-1/2}. \tag{3.22b}$$

If all of the speeds are nonzero, decomposing the flux into any set of vectors will maintain conservation since, by (3.21) and (3.22), (3.19) is automatically satisfied. However, for consistency, as with the solution the flux should typically be decomposed into the eigenvectors of the Jacobian. This follows from the fact that through any rarefaction of the p^{th} family, $f(q)_x = f'(q)q_x = \lambda^p r^p$. Additionally, across a shock in the p^{th} family, $[[f(q)]] = s\,[[q]]$. Note that if the Roe eigenvectors and speeds are used, the f-wave approach is the same as the standard Roe solver, and each flux wave is exactly $\mathcal{Z}^p_{i-1/2} = \hat{\lambda}^p(Q_{i-1}, Q_i)\mathcal{W}^p_{i-1/2}$, where the waves $\mathcal{W}^p_{i-1/2}$ are defined by a Roe decomposition of $(Q_i - Q_{i-1})$. This follows from the conservative property of the Roe solver (3.8).

In general, in order to ensure conservation, special measures have to be taken if one of the speeds $s^p_{i-1/2}$ happens to be zero, such as splitting the wave into two waves moving in either direction. However, this is not the case if $\mathcal{Z}^p_{i-1/2}$ is also zero. If a flux wave $\mathcal{Z}^p_{i-1/2}$ is nonzero, and it is stationary, $s^p_{i-1/2} = 0$, this represents a jump in the flux that does not change the solution. For certain systems, such as systems with a source term, this may be desired and in fact we will exploit this property in Chapter 6.

3.4.2 The CFL Condition

Recall that the CFL condition is a necessary condition for stability, and it requires that the numerical domain of dependence include the true domain of dependence. Since the wave propagation method uses the adjacent values to any grid cell to determine an explicit numerical update, it is required that the time step satisfy

$$\frac{\max_p(|s^p_{i-1/2}|)\Delta t}{\Delta x} \leq 1, \tag{3.23}$$

$\forall i \in \mathbb{Z}_c$. Recall that the left hand side of (3.23) is the *Courant number*.

3.5 Depth Positivity and Vacuum States

For some applications, certain properties might be desired in the approximate Riemann solution that are guaranteed in a true weak solution to the Riemann problem, such as positivity of one of the conserved variables. For the shallow water equations, the depth satisfies $h \geq 0$, and it might be desired that approximate Riemann solution always reflect this. However, the appearance of nonphysical negative values for the depth in the shallow water equations as well as the density in the Euler equations of gas dynamics, is a well known problem. This problem is sometimes called the *vacuum state problem*, named from the latter application.

3.5.1 The Roe Solver

Unfortunately the Roe solver fails to maintain positivity of the depth h for the shallow water equations. (This same difficulty is encountered for the density in the Euler equations.) Since the Roe decomposition for the shallow water equations

$$q_r - q_l = \sum_{p=1}^{2} \alpha^p \hat{r}^p = \sum_{p=1}^{2} \mathcal{W}^p \tag{3.24}$$

corresponds to two discontinuities on either side of a constant middle state q^*, the single middle state depth

$$h^* = h_l + \alpha^1(\hat{r}^1)^1 = h_r - \alpha^2(\hat{r}^2)^1 \tag{3.25}$$

corresponds to the intersection of the vectors $\hat{r}^1$ and $\hat{r}^2$ in phase space. In general, this is not necessarilly positive—particularly if the initial data induces large rarefactions, such as when $hu_r >> hu_l$. This type of Riemann problem will be described in detail in Chapter 4. Figure 3.1 compares the approximate middle states for the Roe and HLLE solver—the intersection of constant vectors—with the true middle state—the intersection of the two integral curves.

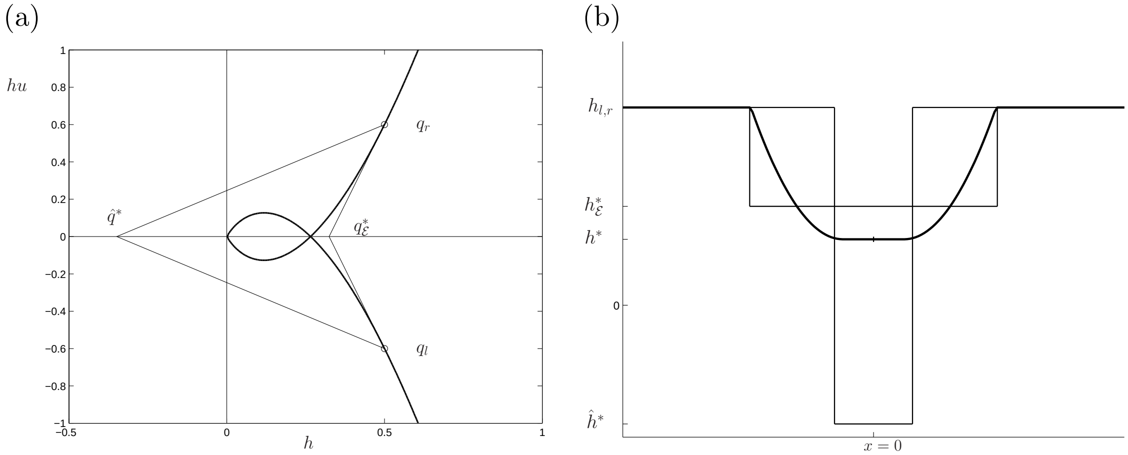


Figure 3.1: Comparison of the Roe and HLLE solvers for a Riemann problem with two strong rarefactions. (a) The true middle-state solution is the intersection of the two integral curves connected to the left and right states, q_l and q_r. The intersection of the Roe eigenvectors $\hat{r}^1$ and $\hat{r}^2$ occurs at $\hat{q}^*$, which corresponds to a negative depth. In this case of symmetrical rarefactions, the HLLE solution can be shown as the intersection of two vectors tangent to the integral curves at q_l and q_r. The intersection at $q^*_{\mathcal{E}}$ corresponds to the positive depth $h^*_{\mathcal{E}}$. (b) The depth of the Riemann solutions at some time $t > 0$. The true solution has two symmetrical rarefactions spreading outward, and the two approximate solutions have jump discontinuities of different strengths moving at different speeds. All of the solutions are conservative implying that the total mass of all three solutions is the same. Since the wave speeds of the Roe solution are much slower, a negative depth results in order to conserve mass.

3.5.2 The HLLE Method

One of Einfeldt's motivations for suggesting the speeds (3.12), was that the net positivity of density is then maintained for the Euler equations [21]. The HLLE method also guarantees positivity of h^* in the shallow water equations by using the Einfeldt speeds. This is easily proved, and as in [29] it will be stated as a theorem for later reference.

Theorem 3.1. *The middle state Einfeldt depth* $h^*_{\mathcal{E}}$

$$h^*_{\mathcal{E}} = \frac{(hu)_l - (hu)_r + s^2_{\mathcal{E}} h_r - s^1_{\mathcal{E}} h_l}{s^2_{\mathcal{E}} - s^1_{\mathcal{E}}}, \tag{3.26}$$

defined by using (3.11b) and (3.13) for the shallow water equations, is always non-negative.

Proof. Since the Roe speeds are such that $\hat{\lambda}^2(q_l, q_r) > \hat{\lambda}^1(q_l, q_r)$ for any states q_l and q_r, the denominator of (3.26) is always positive. For the numerator,

$$\begin{aligned} h^*_{\mathcal{E}} &= (hu)_l - (hu)_r + s^2_{\mathcal{E}} h_r - s^1_{\mathcal{E}} h_l \\ &\geq h_l u_l - h_r u_r + (u_r + \sqrt{gh_r})h_r - (u_l - \sqrt{gh_l})h_l \\ &= h_r\sqrt{gh_r} + h_l\sqrt{gh_l} \\ &\geq 0. \end{aligned} \tag{3.27}$$

□

Theorem 3.1 is intuitively clear if one considers the nature of the Einfeldt speeds. These speeds are intended to bound the maximum and minimum speeds of waves arising out of the true Riemann solution. If the true Riemann solution maintains positivity of h for all (x/t), and it is replaced by a single constant state extending beyond the lower and upper values of (x/t) where the true solution varies—conservation ensures that the approximate solution is also nonnegative. In fact, note from the proof of Theorem 3.1, that if the Eindeldt speeds were replaced by speeds $\tilde{s_{\mathcal{E}}}^{1,2}$ that bound them from below and above:

$$\tilde{s_{\mathcal{E}}}^1 \leq s^1_{\mathcal{E}} < s^2_{\mathcal{E}} \leq \tilde{s_{\mathcal{E}}}^2, \tag{3.28}$$

then positivity would still be maintained.

3.5.3 The f-wave Method

In the f-wave approach, the flux, not the solution is not decomposed into waves. A middle state solution q^* is therefore not explicitly constructed since the jumps across the waves $\mathcal{Z}^p$ do not correspond directly to jumps in q. However, for the shallow water equations, if one considers the numerical effect of the f-waves in defining the fluctuations (3.22), it is possible to define what we will call the *effective middle state depth(s).* That is, in the standard decomposition of the solution, one has

$$h^* = h_l + (\mathcal{W}^1)^1 = h_r - (\mathcal{W}^2)^1. \tag{3.29}$$

The fluctuations that update the solution are then constructed by multiplying the waves $\mathcal{W}^p$ by the corresponding speed s^p. In the f-wave approach, the fluctuations are defined by using $\mathcal{Z}^p$ directly and these are associated with a speed s^p. Note that if we decomposed the flux directly, and then defined the fluctuations in the standard way, using $\mathcal{W}^p = \mathcal{Z}^p/s^p$, this would produce exactly the same numerical result as if $\mathcal{W}^p$, representing jumps in the solution, were determined by some other means.

Definition 3.1. Effective middle-state depths. *If the fluctuations used in (3.18) are defined by a decomposition of the (shallow water) flux according to (3.22), then the effective middle state depths are given by*

$$h^*_l = h_l + \frac{(\mathcal{Z}^1)^1}{s^1}, \tag{3.30a}$$

$$h^*_r = h_r - \frac{(\mathcal{Z}^2)^1}{s^2}. \tag{3.30b}$$

More generally, given any wave-type approximate Riemann solver based on a flux decomposition, if the resulting numerical update is equivalent to that from a wave-type solver based on a solution decomposition, the middle-state depths of the latter will be referred to as the effective middle-state depths of the former.

In general, $h_r^* \neq h_l^*$, and for the f-wave approach, one may imagine a jump in the solution at the cell interface, even if it is not explicitly constructed. (If the Roe eigenstructure is used, it is easy to show that $h_l^* = h_r^*$, so a jump at the cell interface is due to a deviation from the Roe eigenstructure.) If two middle states were explicitly constructed using (3.30), and the fluctuations were defined in the standard way, the update to the mass equation would be identical to that of the f-wave decomposition.

Considering the positive preserving properties of the HLLE solver, it might seem that by using some set of vectors with corresponding speeds equivalent to the Einfeldt speeds, that positivity of the f-wave method should be maintained based on conservation. However, this is not the case since two effective middle state depths exist. It can be shown that the sum

$$s_{\mathcal{E}}^2 h_r^* - s_{\mathcal{E}}^1 h_l^*, \tag{3.31}$$

representing the net mass (divided by Δt) of the Riemann solution between the two waves would be positive for such a method. However, there is no guarantee in general that both h_l^* and h_r^* are positive with the f-wave method.

3.5.4 Depth-Positive-Definite Riemann Solvers

We will summarize some of these ideas as a definition for later reference.

Definition 3.2. Depth-positive-definite Riemann Solvers. *An approximate Riemann solver for the shallow water equations with initial data $h_l, h_r \geq 0$, that produces only nonnegative middle states, or effective middle states as defined by Definition 3.1, will be referred to as a depth-positive-definite Riemann solver.*

It might also simply be said that the Riemann solver *maintains positivity* of the solution. It should be noted that using a depth positive definite Riemann solver is not the only requirement for maintaining a nonnegative numerical solution. However, if the first order update (3.18) is used in one dimension, and if the Courant number is restricted to 1/2 (1/4 in two dimensions), it is sufficient. An additional fix is discussed in Section 3.9 if these restrictions are not met.

3.6 Numerical Treatment of Source Terms

3.6.1 The Fractional Step Method

The numerical methods described so far have been based on solving a homogeneous conservation law. When a source term exists, the numerical method described for homogeneous conservation laws can be implemented unaltered, by incorporating the source term with a separate integration step. This is called the *fractional step method*, or *fractional stepping.* The idea is to first solve the homogeneous conservation law

$$q_t + f(q)_x = 0, \tag{3.32}$$

followed by an integration of the set of ODEs

$$q_t = \psi(q, x) \tag{3.33}$$

in each time step. The effect of alternating between these two steps incorporates the source term. Accuracy of the individual methods can be inherited by the overall method (yet only up to second order, see [67]) given specific formulations of the order and size of the time steps (see Strang [88]).

The popularity of this approach is due to the fact that it is very simple and is very robust for many applications. The approach fails for applications where there is a prevalent steady state that exists due to a nontrivial balance $f(q)_x \approx \psi$. In this case it is well known (see, for instance, [66, 6]) that fractional stepping fails to preserve the steady state, or resolve small deviations to it. This is quite intuitive, since if $f(q)_x \approx \psi(q, x)$, and both terms are large, the distinct steps for (3.32) and (3.33) must nearly and precisely undo one another, while resolving the small deviation. This situation is precisely what occurs with the shallow water equations, where the "lake at rest" steady state over variable bathymetry is due to a nontrivial balance $\frac{1}{2}g(h^2)_x = -ghb_x$. This situation is exacerbated for tsunami modeling, where the tsunami is actually a tiny, nearly infinitesimal deviation, considering the background steady state. This will be discussed in more detail in Chapter 5.

3.6.2 An f-wave Approach to Source Terms

The wave propagation methodology of LeVeque uses the waves arising from a Riemann solver to directly update the solution. This same methodology can be applied to source terms as well; rather than treating source and flux terms separately, the effect of the source term can be added directly to the updating waves in the fluctuations $\mathcal{A}^{\pm}\Delta Q$. This is the approach suggested by Bale *et al* in [6]. This is accomplished, and is consistent with the original conservation law, if the source term is subtracted from the flux difference before performing the decomposition

$$f(Q_i) - f(Q_{i-1}) - \Delta x \Psi_{i-1/2} = \sum_{p=1}^{M_w} \mathcal{Z}^p_{i-1/2}, \tag{3.34}$$

where $\Psi_{i-1/2}\Delta x$ is some approximation to $\int_{x_{i-1}}^{x_{i+1}} \psi(q, x)\, dx$. The fluctuations at grid cell interfaces are then defined in the standard way (3.22), and the standard updating formula (3.18) can be used. If the source term is interpreted as a point source at the grid cell interface, $\psi(q, x) = \Psi_{i-1/2}\Delta x \delta(x - x_{i-1/2})$, equation (3.34) amounts to subtracting the jump in the flux at the grid cell interface

$$f(q(x^+_{i-1/2}, t)) - f(q(x^-_{i-1/2}, t)) = \Delta x \Psi_{i-1/2} \tag{3.35}$$

from the flux difference before decomposing into the propagating waves. If the Riemann problem arises near a steady state where $f(q)_x \approx \psi(q,x)$, then

$$\frac{f(Q_i) - f(Q_{i-1})}{\Delta x} \approx \Psi_{i-1/2}, \tag{3.36}$$

and nonzero waves in the decomposition (3.34) will be due to the deviation from the steady state.

Equation (3.34) is said to *upwind* the source term, in the sense that the source term is decomposed into the eigenvectors of the Jacobian and propagated at the wave speeds. This approximates the true solution, as was pointed out for linear problems in Section 2.6. In fact, this form of upwinding has been commonly advocated (*e.g.* [83, 52]).

3.7 High Resolution Methods

The Godunov-type methods, as described so far, converge to discontinuous entropy satisfying weak solutions, however, they are only first-order accurate and numerically diffusive. Various second-order methods, such as the Lax-Wendroff method, can be derived based on Taylor series expansions of the PDEs—requiring smoothness of the solution. These methods not only fail to be second-order accurate near discontinuities, they exhibit large spurious oscillations near such features and are overly dispersive in general near steep gradients—problematic since hyperbolic equations typically exhibit steepening. This problem is circumvented by the *high-resolution methods*—nonlinear methods that use standard second-order formulas where the solution is smooth yet resort to the first-order Godunov methods near shocks or steep gradients. This is accomplished by adding second-order corrections to the first-order method, together with the use of a *limiter function* ϕ that measures the local smoothness of the solution.

The wave propagation algorithm (3.18) can be extended to a second-order accurate method

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2}\right) + \frac{\Delta t}{\Delta x}(\tilde{F}_{i+1/2} + \tilde{F}_{i-1/2}), \tag{3.37}$$

where $\tilde{F}_{i\pm1/2}$ are limited second-order *correction fluxes.* These corrections can be defined in terms of the first-order waves,

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{M_w} |s_{i-1/2}^p| \left(1 - \frac{\Delta t}{\Delta x}|s_{i-1/2}^p|\right) \widetilde{\mathcal{W}}_{i-1/2}^p, \tag{3.38}$$

where $\widetilde{\mathcal{W}}_{i-1/2}^p = \phi\mathcal{W}_{i-1/2}^p$ is a limited wave. For the f-wave approach, the corresponding correction flux must be

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{M_w} \mathrm{sgn}(s_{i-1/2}^p) \left(1 - \frac{\Delta t}{\Delta x}|s_{i-1/2}^p|\right) \widetilde{\mathcal{Z}}_{i-1/2}^p, \tag{3.39}$$

where $\widetilde{\mathcal{Z}}_{i-1/2}^p = \phi\mathcal{Z}_{i-1/2}^p$. If the limiters are absent (or ϕ=1), (3.37) is an extension of the second-order Lax-Wendroff method for systems (see [63, 67] for details). Note that in the case of a source term, if the f-waves are defined by (3.34), then it is the deviations from the steady state that get limited.

Limiter functions interpret the smoothness of the solution by comparing the jump in the solution in neighboring grid cells, and in the case of a potential shock, limit the waves in (3.38) or (3.39). Ideally, the limiter functions should isolate when a steep gradient or shock exists in one characteristic family and limit that corresponding wave. This is accomplished by comparing the wave $\mathcal{W}^p_{i-1/2}$ with $\mathcal{W}^p_{I-1/2}$ where

$$I = \begin{cases} i-1 & \text{if} \quad s^p_{i-1/2} > 0, \\ i+1 & \text{if} \quad s^p_{i-1/2} < 0. \end{cases} \tag{3.40}$$

Some variable θ that measures the local smoothness in the p^{th} field at $x_{i-1/2}$ is then defined. For instance,

$$\theta^p_{i-1/2} = \frac{\mathcal{W}^p_{I-1/2} \cdot \mathcal{W}^p_{i-1/2}}{|\mathcal{W}^p_{i-1/2}|} \tag{3.41}$$

is small if $|\mathcal{W}^p_{i-1/2}| >> |\mathcal{W}^p_{I-1/2}|$, or if the p^{th} eigenvector varies significantly. A limiter function $\phi(\theta)$ is then chosen that approaches zero if $\theta \rightarrow 0$, and is otherwise bounded by a small constant. For instance, the minmod function

$$\phi(\theta) = \max(0, \min(1, 0)) \tag{3.42}$$

produces a method that varies between the diffusive Godunov method and the dispersive Lax-Wendroff method. Another choice of limiter, the MC limiter

$$\phi(\theta) = \max(0, \min(1+\theta)/2, 2, 2\theta) \tag{3.43}$$

produces a method that varies between Fromm's method ($\phi = 2$), the Lax-Wendroff method ($\phi = 1$) and the first-order Godunov method ($\phi = 0$) (see [67]). For finite grid spacings the MC limiter produces a more accurate method than the minmod limiter (*e.g.* [84]), though both are formally second order accurate for smooth solutions.

It can be shown that the high resolution methods with the limiters (3.42) or (3.43), and a reasonable definition of θ, produce a method that is *total variation diminishing*. For proof and details see the work by Harten [40] and Sweby [89]. The total variation of a numerical solution is defined by

$$TV(Q^n) = \sum_{i \in \mathbb{Z}_c} |Q^n_i - Q^{n+1}_i|. \tag{3.44}$$

A total variation diminishing method, or *TVD method*, satisfies $TV(Q^n_i) \leq TV(Q_i)$. This in turn allows convergence to shocks without spurious oscillations.

3.8 Extension to Multiple Dimensions

The wave propagation algorithms can be extended to higher dimensions by solving, at each grid cell interface, one-dimensional normal Riemann problems—described in the last section of Section 2.5. High resolution can again be accomplished by including limited flux corrections at each interface (though the total variation diminishing property is not strictly accomplished in more that one dimension [67]). Additionally, for full second-order accuracy, some transverse terms must be included that approximate cross derivatives.

We consider the two-dimensional conservations law (neglecting the source term)

$$q_t + f(q)_x + g(q)_y = 0. \tag{3.45}$$

The numerical solution Q^n_{ij} is an approximation

$$Q^n_{ij} \approx \frac{1}{\Delta x \Delta y} \int_{\mathcal{C}_{ij}} q(x, y, t^n)\, dxdy, \tag{3.46}$$

where $\mathcal{C}_{ij} = \big\{[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]\big\}$ is the rectangular grid cell. A first-order extension to the wave propagation algorithm (3.18) can be written

$$\begin{aligned} Q^{n+1}_{ij} = Q^n_{ij} - \frac{\Delta t}{\Delta x}(\mathcal{A}^+\Delta Q_{i-1/2,j} + \mathcal{A}^-\Delta Q_{i+1/2,j}) \\ - \frac{\Delta t}{\Delta y}(\mathcal{B}^+\Delta Q_{i,j-1/2} + \mathcal{B}^-\Delta Q_{i,j+1/2}), \end{aligned} \tag{3.47}$$

where the fluctuations $\mathcal{A}^\pm \Delta Q_{i\pm1/2,j}$ are defined as in Section 3.4, using the solution to one-dimensional normal Riemann problems at the two cell interfaces of $\mathcal{C}_{i,j}$ at $x = x_{i\pm1/2}$, and $\mathcal{B}^\pm\Delta Q_{i,j\pm1/2}$ are defined in the same logical way at the other two cell interfaces.

If we include the limited correction fluxes we can write the two dimensional method as

$$\begin{aligned} Q^{n+1}_{ij} &= Q^n_{ij} \\ &- \frac{\Delta t}{\Delta x}(\mathcal{A}^+\Delta Q_{i-1/2,j} + \mathcal{A}^-\Delta Q_{i+1/2,j}) - \frac{\Delta t}{\Delta y}(\mathcal{B}^+\Delta Q_{i,j-1/2} + \mathcal{B}^-\Delta Q_{i,j+1/2}) \\ &- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta x}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right), \end{aligned} \tag{3.48}$$

where $\tilde{G}_{i,j\pm1/2}$ are defined in the same logical way as $\tilde{F}_{i\pm1/2,j}$, described in Section 3.7.

The method (3.48), as so far described, is not formally second-order accurate for smooth solutions, since a Taylor expansion shows that cross derivatives (q_{xy}) are neglected. An alternative way to view this same problem, is that, in multiple dimensions waves should generally propagate at an angle to the grid, and in a single time step, the solution in all of the cells $\mathcal{C}_{i\pm1,j\pm1}$ effects the solution in $\mathcal{C}_{i,j}$. This is discussed in detail in [67] . The effect of these cross derivatives can be included in (3.48), by introducing new terms to $\tilde{F}_{i\pm1/2,j}$ and $\tilde{G}_{i,j-1/2}$. It turns out that this can be done in the context of the wave propagation methods by splitting the right- and left-going fluctuations $\mathcal{A}^\pm\Delta Q_{i\pm1/2,j}$ into additional up- and down-going components, and the up- and down-going components $\mathcal{B}^\pm\Delta Q_{i,j\pm1/2}$ into additional right- and left-going components. This procedure requires *transverse Riemann solvers*, which are similar to normal Riemann solvers, yet, the fluctuations not the solution are decomposed into transversely propagating waves. This procedure also improves the stability of the two-dimensional methods, allowing Courant numbers up to 1 (like the one-dimensional methods). The details are omitted and can be found in [67] .

3.9 Preserving Positivity by Limiting

Even when depth-positive-definite Riemann solvers are used, it is possible to generate non-physical negative depths for the shallow water equations due to the correction fluxes (second order terms) in the wave propagation method (3.48). This can be circumvented by introducing additional limiting to these terms to maintain nonnegativity when near a dry state. Since interface fluxes affect more than one cell, it is necessary to limit the gross outward mass fluxes, even when the net flux does not generate a negative depth. That is, if the gross outward mass flux from a grid cell is more than the remaining mass, those mass flux terms must be limited even if inward mass flux at another boundary prevents a negative depth. This is because, if limiting was based only on the net mass flux, preventing a negative depth in one cell might actually generate a negative depth on an adjacent cell, by limiting its inward mass flux. Therefore, only the outward mass fluxes from a given grid cell may be considered. This is demonstrated below.

The second order method (3.48) can be written as

$$Q_{ij}^{n+1} = Q_{ij}^{n\,*} - \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta x}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right), \tag{3.49}$$

where $Q_{ij}^{n\,*}$ represents the solution due only to the first order update:

$$\begin{aligned} Q_{ij}^{n\,*} &= Q_{ij}^{n} \\ &- \frac{\Delta t}{\Delta x}(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}) - \frac{\Delta t}{\Delta y}(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}). \end{aligned} \tag{3.50}$$

Now, let P_{ij} be the total gross mass-flux out of the cell $\mathcal{C}_{ij}$, due to the first component of the second order fluxes:

$$\begin{aligned} P_{ij} = \Delta y \Delta t &\left(\max(0, \tilde{F}^{1}_{i+1/2,j}) - \min(0, \tilde{F}^{1}_{i-1/2,j})\right) \\ &+ \Delta x \Delta t \left(\max(0, \tilde{G}^{1}_{i,j+1/2}) - \min(0, \tilde{G}^{1}_{i,j-1/2})\right). \end{aligned} \tag{3.51}$$

Let M_{ij} be the amount of mass (the first component of Q_{ij} times the area of $\mathcal{C}_{ij}$) remaining in the cell after the first-order update:

$$M_i = \Delta x \Delta y (Q_{ij}^{n\,*})^{1}. \tag{3.52}$$

We now define the limiter

$$\phi_{i,j} = \begin{cases} 1 & \text{if} \quad P_{ij} = 0, \\ \min(1, M_{ij}/P_{ij}) & \text{if} \quad P_{ij} \neq 0. \end{cases} \tag{3.53}$$

We then redefine the fluxes using the limiter from the cell that those fluxes take mass away from,

$$\tilde{F}_{i-1/2,j} \rightarrow \phi_{i,j}\tilde{F}_{i-1/2,j} \quad \text{if} \quad \tilde{F}^1_{i-1/2,j} < 0, \tag{3.54a}$$

$$\tilde{F}_{i-1/2,j} \rightarrow \phi_{i-1,j}\tilde{F}_{i-1/2,j} \quad \text{if} \quad \tilde{F}^1_{i-1/2,j} > 0, \tag{3.54b}$$

$$\tilde{G}_{i,j-1/2} \rightarrow \phi_{i,j}\tilde{G}_{i,j-1/2} \quad \text{if} \quad \tilde{G}^1_{i,j-1/2} < 0, \tag{3.54c}$$

$$\tilde{G}_{i,j-1/2} \rightarrow \phi_{i,j-1}\tilde{G}_{i,j-1/2} \quad \text{if} \quad \tilde{G}^1_{i,j-1/2} > 0. \tag{3.54d}$$

It is easy to show that this will prevent generating negative depths from the second order correction fluxes. When Courant numbers are allowed to approach 1, it might be necessary to include the first order fluxes (easily determined from the fluctuations [67]) in this procedure, even when using a depth-positive-definite Riemann solver. However, note that since the procedure is overly agressive (fluxes might be limited even when the net flux does not generate a negative value), use of a depth-positive-definite Riemann solver is nonetheless advantageous for accuracy.

Chapter 4

THE SHALLOW WATER EQUATIONS

In this chapter we will take a closer look at the shallow water equations before addressing tsunami modeling specifically in Chapter 5. First we will look at the shallow water assumptions and a derivation of the equations, followed by a closer investigation of solutions to the Riemann problem. Finally, we will look at the source term due to variable bathymetry and consider some properties of analytical steady state solutions.

4.1 The Shallow Water Approximation and Derivation

The shallow water equations can be derived in a number of ways, all of which are consistent with a hydrostatic assumption about the flow—that fluid parcels are in vertical equilibrium due to a balance of gravity and the vertical pressure gradient. Once this assumption is made, the shallow water equations can be derived from the full three dimensional equations for an incompressible inviscid free surface flow. However, if the hydrostatic assumption and its implications are accepted *a priori*, a more generally valid system of equations in conservative form can be derived from first principles.

We start with an incompressible inviscid fluid of uniform density ρ and a flow velocity $\mathbf{u} \equiv (u, v, w)$, with a free (unknown) upper boundary $z = \eta(x, y, t)$ and a fixed bottom boundary $z = b(x, y)$. The flow is governed by the incompressibility condition

$$\nabla \cdot \mathbf{u} = 0, \tag{4.1}$$

and the Euler equations for an inviscid fluid

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P - g\hat{\mathbf{z}}, \tag{4.2}$$

where

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \tag{4.3}$$

is the *material derivative*, $g\hat{\mathbf{z}}$ is the gravitational force in the z-direction and $P(x, y, z, t)$ is the pressure. The PDEs (4.1) and (4.2) are complemented with kinematic and dynamic boundary conditions on $z = b(x, y)$ and $z = \eta(x, y, t)$ (*e.g.* [103, 87]). Determining $\eta(x, y, t)$ is one of the goals of the problem. The crucial simplifying assumption for shallow water theory is that the pressure is strictly hydrostatic,

$$\frac{\partial P(x, y, z, t)}{\partial z} = -\rho g, \tag{4.4}$$

which upon integrating vertically implies

$$P(x, y, z, t) = \rho g \left(\eta(x, y, t) - z\right). \tag{4.5}$$

From (4.1) and (4.2), this assumption implies (*e.g.* [103, 87]) that the velocity field is independent of z,

$$u_z \equiv v_z \equiv w_z \equiv 0, \tag{4.6}$$

and that $w \equiv 0$. Clearly no realistic dynamic flow is strictly hydrostatic, however, we are interested in the leading order behavior of flows that approximately satisfy (4.5).

The relevant question is then—what types of flows are approximately hydrostatic? Using (4.1), (4.2) and boundary conditions on the free surface $\eta(x,y,t)$ and bottom surface $b(x,y)$, a perturbation procedure can be developed for all quantities using relevant physical scales of a particular problem. For flows where the relevant horizontal length scale L is long compared to a characteristic depth H, the hydrostatic condition is the leading order approximation to the pressure $P(x,y,z,t)$. That is, if the equations are developed using the nondimensional quantity

$$\epsilon = \frac{H}{L}, \tag{4.7}$$

and the pressure is written as an asymptotic series,

$$P(x,y,z,t) = P_0(x,y,z,t) + \epsilon P_1(x,y,z,t) + \epsilon^2 P_2(x,y,z,t) + \cdots, \tag{4.8}$$

it can be shown (*e.g.* [50, 87, 103])that

$$P_0(x,y,z,t) = \rho g \left(\eta(x,y,t) - z\right). \tag{4.9}$$

For shallow water theory then, (4.9) is taken to be the pressure, and higher order corrections are ignored. No assumption is made about the amplitude of shallow water waves relative to the depth—all such nonlinearities are retained in shallow water theory.

If we take the hydrostatic pressure assumption and its consequences for granted, we can derive the shallow water equations directly from conservation. Since the velocity field is independent of z, we can integrate out the vertical dimension and reduce the problem to two dimensions. Consider a three dimensional control volume:

$$\mathcal{V} = \left\{(x,y,z):\ (x,y) \in \Omega,\ b(x,y) \le z \le \eta(x,y,t)\right\}, \tag{4.10}$$

where $\Omega \subseteq \mathbb{R}^2$ is an arbitrary region of the x–y plane (see Figure 4.1). Conservation of mass requires

$$\frac{\partial}{\partial t}\int_{\mathcal{V}} \rho = -\int_{\partial\mathcal{V}} \rho \mathbf{u}\cdot\mathbf{n}, \tag{4.11}$$

where $\mathbf{n}$ is the unit outward normal vector to the surface $\partial\mathcal{V}$. It should be clear that there is no contribution to the surface integral in (4.11) from the top and bottom boundaries:

$$\partial\mathcal{V}_\eta = \left\{(x,y,z):\ (x,y) \in \Omega,\ z = \eta(x,y,t)\right\} \qquad \text{and,} \tag{4.12a}$$

$$\partial\mathcal{V}_b = \left\{(x,y,z):\ (x,y) \in \Omega,\ z = b(x,y)\right\}. \tag{4.12b}$$

Integrating out the vertical dimension and dropping the constant density gives

$$\frac{\partial}{\partial t}\int_{\Omega} h = -\int_{\partial\Omega} h\mathbf{u}_\perp \cdot \mathbf{n}_\perp, \tag{4.13}$$

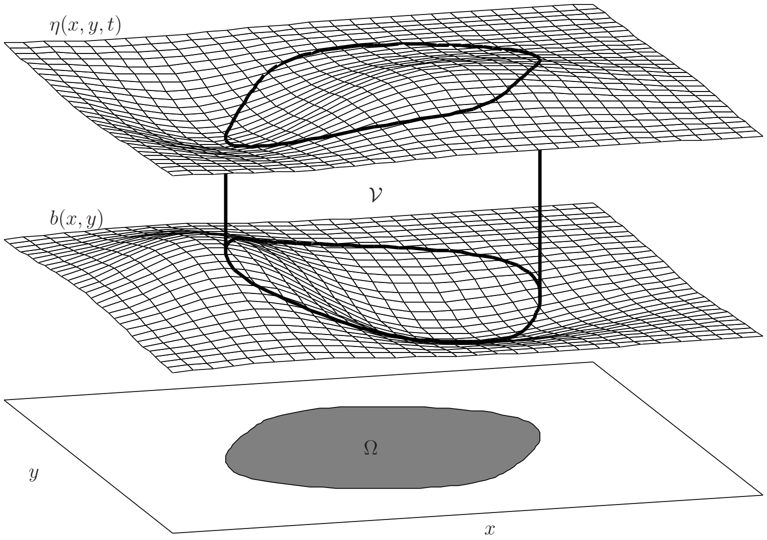


Figure 4.1: Deriving the shallow water equations from conservation principles. The three dimensional control volume $\mathcal{V}$ and the two dimensional region Ω that remains after the vertical dimension is integrated out, are shown. The shallow water equations are a hyperbolic conservation law in the two dimensional x-y plane.

where $\mathbf{u}_\perp = (u, v)^T$ is the horizontal velocity field, $\mathbf{n}_\perp = (n_x, n_y)^T$—the outward direction normal to $\partial\Omega$—and $h(x, y, t) = \eta(x, y, t) - b(x, y)$. Note that if $\mathbf{u}_\perp$ is considered a depth averaged velocity field, (4.13) requires none of the shallow water assumptions.

Conservation of momentum on the control volume $\mathcal{V}$ requires

$$\frac{\partial}{\partial t}\int_{\mathcal{V}} \rho\mathbf{u}_\perp = -\int_{\partial\mathcal{V}_s} (\rho\mathbf{u}_\perp)\mathbf{u}\cdot\mathbf{n} - \int_{\partial\mathcal{V}_s} P\mathbf{n}_\perp - \int_{\partial\mathcal{V}_b} P\mathbf{n}_\perp, \tag{4.14}$$

where $\partial\mathcal{V}_s$ denotes the vertical boundaries of $\mathcal{V}$:

$$\partial\mathcal{V}_s = \partial\mathcal{V} - \{\partial\mathcal{V}_\eta \cup \partial\mathcal{V}_b\}. \tag{4.15}$$

All of the vertical forces have been neglected from (4.14) since the hydrostatic assumption implies that they balance. The third equation for ρw has therefore been neglected from the system since it reduces it to $(\rho w)_t = 0$. The first surface integral in (4.14) is due to the advection of momentum across the boundaries, and the last two integrals are simply the net horizontal force due to the pressure on the boundaries. The surface integral of the pressure at the top surface $z = \eta(x, y, t)$ is zero since the pressure is defined to be zero there. Using (4.5) for P and then integrating out the vertical dimension in (4.14) gives,

$$\begin{aligned}\frac{\partial}{\partial t}\int_{\Omega} \rho h\mathbf{u}_\perp &= -\int_{\partial\mathcal{V}_s} \rho\mathbf{u}_\perp(\mathbf{u}\cdot\mathbf{n}) - \int_{\partial\mathcal{V}_s} \rho g(\eta - z)\mathbf{n}_\perp - \int_{\partial\mathcal{V}_b} \rho g(\eta - z)\mathbf{n}_\perp, \\ &= -\int_{\partial\Omega} \rho h\mathbf{u}_\perp(\mathbf{u}_\perp\cdot\mathbf{n}_\perp) - \int_{\partial\Omega} \rho\frac{1}{2}gh^2\mathbf{n}_\perp - \int_{\Omega} \rho gh\nabla_\perp b, \end{aligned} \tag{4.16}$$

where $\nabla_\perp \equiv (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^T$. The final integral results from parameterizing the surface integral on $\partial \mathcal{V}_b$, and noting that $\eta - z = h$ on that surface. The constant density can again be dropped from (4.16).

We can write (4.13) and (4.16) in the more familiar form

$$\frac{\partial}{\partial t}\int_\Omega q \quad + \int_{\partial\Omega} f(q)n_x + \int_{\partial\Omega} g(q)n_y = \int_\Omega \psi, \tag{4.17a}$$

where

$$q = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad g(q) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix},$$

and

$$\psi = \begin{bmatrix} 0 \\ -ghb_x \\ -ghb_y \end{bmatrix}. \tag{4.17b}$$

In deriving the shallow water equations from conservation principles, note that no assumptions were made about smoothness of the top boundary $z = \eta(x, y, t)$. The conservative integral form (4.17a) admits solutions with discontinuities in the solution as discussed more generally for hyperbolic conservation laws in Chapter 2. This is important since steepening can lead to discontinuities given any form of shallow water equations. The physically relevant conservative form of the equations provides the proper governing system for these discontinuities.

4.2 The Riemann Problem

In Chapter 2 and Chapter 3 Riemann problems were discussed in the context of hyperbolic conservation laws and finite volume methods. In this section we will look more closely at the Riemann problem for the homogeneous shallow water equations,

$$h_t + (hu)_x = 0, \tag{4.18a}$$

$$(hu)_t + (hu^2 + \frac{1}{2}gh^2)_x = 0, \tag{4.18b}$$

with

$$q(x,0) = \begin{cases} q_l & \text{if } x < 0, \\ q_r & \text{if } x > 0, \end{cases} \quad \text{where} \quad q = \begin{bmatrix} h \\ hu \end{bmatrix}. \tag{4.18c}$$

Recall that the solution to (4.18) is a similarity solution with two *waves* or characteristic transitions emanating from the initial discontinuity at $x = 0$. Since the characteristic fields for the shallow water equations are genuinely nonlinear, the waves can be either smooth rarefactions defined by integral curves or shock discontinuities. For the purpose of easing discussion in this section, zero-strength transitions (or lack of a wave in a family) will be considered a *rarefaction*. For the shallow water equations the Riemann invariants

$$w^1(q) \equiv u + 2\sqrt{gh}, \tag{4.19a}$$

$$w^2(q) \equiv u - 2\sqrt{gh}, \tag{4.19b}$$

hold through an integral curve of the first and second characteristic fields respectively. Discontinuities must obey the Rankine-Hugoniot jump conditions (2.7), which take the form

$$s \left[\left[\begin{bmatrix} h \\ hu \end{bmatrix}\right]\right] = \left[\left[\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}\right]\right], \tag{4.20}$$

where $[[\cdot]]$ indicates the difference across the jump discontinuity and s is the propagation speed of the discontinuity. Recall that the Rankine-Hugoniot conditions can be used to derive the Hugoniot loci, or curves in phase space that can connect two states through a shock. For the shallow water equations, states q^* that can be connected to q_l through a 1-shock satisfy

$$u^* + h^*\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_l}\right)} = u_l + h_l\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_l}\right)}, \tag{4.21a}$$

and states q^* that can be connected to q_r through a 2-shock satisfy

$$u^* - h^*\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_r}\right)} = u_r - h_r\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_r}\right)}. \tag{4.21b}$$

Discontinuities must also obey entropy conditions, such as the Lax entropy condition— Definition 2.3— which applies to strictly hyperbolic genuinely nonlinear systems such as the shallow water equations. For the shallow water equations, simple admissability criteria can be derived from the Lax entropy condition.

Theorem 4.1. *A solution to (4.18) is the correct entropy satisfying solution if and only if it has the following properties:*

(i). A 1-shock connects q_l to q^* if and only if $h^* > h_l$.

(ii). A 2-shock connects q_r to q^* if and only if $h^* > h_r$.

(iii). Otherwise rarefactions defined by smooth integral curves of the first or second characteristic field connect q^* to q_l or q_r respectively.

Proof. This is a simple consequence of the Lax entropy condition with (4.19) and (4.21). Since the homogeneous shallow water equations are strictly hyperbolic and genuinely nonlinear, q^* is connected to q_l by either a 1-shock or a rarefaction in the first characteristic field, and q^* is connected to q_r by either a 2-shock or a rarefaction in the second characteristic field. It will therefore suffice to prove the contrapositive to *(i)* and *(ii).*

(⇒) Suppose that a 2-wave connecting q_r to q^* is a rarefaction. By (4.19) and the Lax entropy condition,

$$\begin{aligned} 0 &\leq \lambda^2(q_r) - \lambda^2(q^*) \\ &= u_r + \sqrt{gh_r} - \left(u^* + \sqrt{gh^*}\right) \\ &= w^2(q_r) + 3\sqrt{gh_r} - \left(w^2(q^*) + 3\sqrt{gh^*}\right) \\ &= 3(\sqrt{gh_r} - \sqrt{gh^*}), \end{aligned} \tag{4.22}$$

implying $h^* \leq h_r$.

(⇐) Conversely, suppose that $h^* \leq h_r$. If a 2-shock connects q^* to q_r, then the Lax entropy condition implies: $u^* + \sqrt{gh^*} > u_r + \sqrt{gh_r}$,

$$\begin{aligned} \Rightarrow u^* - u_r &> \sqrt{gh_r} - \sqrt{gh^*}, \\ &\geq 0 \quad \text{by supposition.} \end{aligned} \tag{4.23}$$

Additionally, the two states must lie on the Hugoniot loci for a 2-shock, implying that

$$u^* - h^*\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_r}\right)} = u_r - h_r\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_r}\right)}, \tag{4.24}$$

$$\begin{aligned} \Rightarrow u^* - u_r &= (h^* - h_r)\sqrt{\frac{g}{2}\left(\frac{1}{h^*} + \frac{1}{h_r}\right)} \\ &\leq 0 \quad \text{by supposition.} \end{aligned} \tag{4.25}$$

However, (4.23) and (4.25) are contradictory since they imply $0 \geq u^* - u_r > 0$. Therefore, if $h^* \leq h_r$, then a rarefaction not a 2-shock must connect q_r to q^*. An entirely analogous proof can be used for 1-waves.

□

Theorem 4.1 allows us to uniquely determine the middle solution q^* by using (4.19) and (4.21) such that the properties *(i)-(iii)* are satisfied. If we define the function

$$\varphi^1(h; h_l) = \begin{cases} 2\left(\sqrt{gh} - \sqrt{gh_l}\right) & \text{if } h \leq h_l, \\ (h - h_l)\sqrt{\frac{g}{2}\left(\frac{1}{h} + \frac{1}{h_l}\right)} & \text{if } h > h_l, \end{cases} \tag{4.26}$$

then we have

$$u(h) = u_l - \varphi^1(h; h_l), \tag{4.27}$$

for u and h through any 1-wave connected to u_l. Similarly, if we define the function

$$\varphi^2(h; h_r) = \begin{cases} 2\left(\sqrt{gh} - \sqrt{gh_r}\right) & \text{if } h \leq h_r, \\ (h - h_r)\sqrt{\frac{g}{2}\left(\frac{1}{h} + \frac{1}{h_r}\right)} & \text{if } h > h_r, \end{cases} \tag{4.28}$$

then through any 2-wave we have

$$u(h) = u_r + \varphi^2(h; h_r). \tag{4.29}$$

The middle state must occur in phase space where the curves defined by (4.27) and (4.29) intersect. If we subtract the right hand side of (4.27) from (4.29) we have

$$f(h) = \varphi^1(h; h_l) + \varphi^2(h; h_r) + \Delta u, \quad h > 0, \tag{4.30}$$

where $\Delta u \equiv u_r - u_l$. A root of $f(h)$ determines an intersection in phase space,

$$f(h^*) = \varphi^1(h^*; h_l) + \varphi^2(h^*; h_r) + \Delta u = 0, \tag{4.31}$$

and hence corresponds to a middle state depth h^*. If a root h^* is found to $f(h)$, then u^* can be found by evaluating either (4.27) or (4.29).

Clearly $f(h)$ is nonlinear, however it has several nice properties in terms of root finding. These are stated as a theorem for clarity.

Theorem 4.2. *The function $f(h)$ defined by (4.30) has the following properties:*

(i). It is a monotonically increasing function of h.

(ii). It is continuous and has continuous derivatives, even at $h = h_l$ and $h = h_r$.

(iii). It is convex: $f''(h) < 0$.

Theorem 4.2 can be demonstrated through straightforward analysis. See Toro [101] for more details.

By using Theorem 4.1 and Theorem 4.2, a complete characterization of the Riemann structure can be determined, even without root finding [101]. That is, we can determine if each respective wave in the Riemann solution is a shock or rarefaction, simply by evaluating $f(h_l)$ and $f(h_r)$. To see this, let

$$h_{min} = \min(h_l, h_r), \tag{4.32a}$$
$$h_{max} = \max(h_l, h_r). \tag{4.32b}$$

Now, by Theorem 4.1 and the fact that $f(h)$ is monotonically increasing we have

$$\left.\begin{array}{lll} \text{CASE 1:} & f(h_{min}) \geq 0 & \Leftrightarrow \text{ two rarefactions,} \\ \text{CASE 2:} & f(h_{max}) \geq 0 > f(h_{min}) & \Leftrightarrow \text{ one shock, one rarefaction,} \\ \text{CASE 3:} & f(h_{max}) < 0 & \Leftrightarrow \text{ two shocks,} \end{array}\right\}. \tag{4.33}$$

The basis of (4.33) is clear when one considers where the root h^* must lie relative to h_{min} and h_{max}. Determining whether CASE 2 has a 1-shock or 2-shock simply depends on which is greater h_l or h_r. Figure 4.2 shows an example of $f(h)$ for each of the three cases. Note that since $f(h)$ is only defined for $h > 0$, it is possible that no root will exist. This will be discussed in detail in the next section, when we consider dry states $h = 0$.

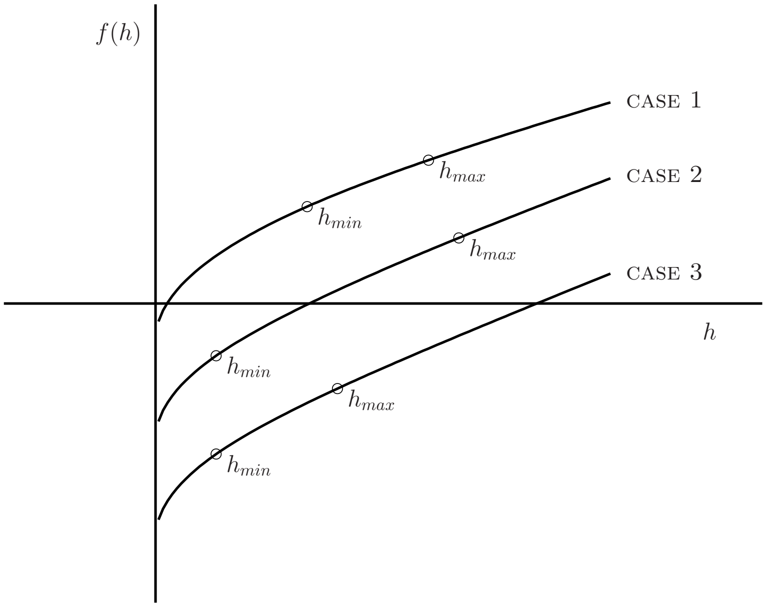


Figure 4.2: Characterization of the Riemann problem for the shallow water equations by function evaluation. The middle state depth h^* is the root of $f(h)$. The graph shows an example of $f(h)$ for each of the three possible cases in (4.33). Note that Δu shifts the curve for $f(h)$ up or down. Large values of Δu can therefore ensure CASE 1, or may even eliminate a positive root h^*

The fact that $f(h)$ is convex will allow us to exploit various root finding techniques when estimating h^*. This will be discussed in Chapter 6 when we discuss approximate Riemann solvers in detail. Note that $f(h)$ is linear in $\sqrt{gh}$ when $h \leq h_{min}$. This means that the root can be determined exactly in the case of a double rarefaction—when h^* is less than h_{min}. Solving for h^* in (4.31) gives

$$h^* = \frac{1}{16g}\left(2\left(\sqrt{gh_l} + \sqrt{gh_r}\right) - \Delta u\right)^2. \tag{4.34}$$

Note that if

$$\Delta u > 2\left(\sqrt{gh_l} + \sqrt{gh_r}\right), \tag{4.35}$$

then the quantity squared in (4.34) is negative, and (4.34) is no longer valid. This is called the *incipient dry bed* or *incipient cavitation* problem [101], and will be discussed in the next section. Following Toro [101], we will denote

$$(\Delta u)_{crit} = 2\left(\sqrt{gh_l} + \sqrt{gh_r}\right), \tag{4.36}$$

and refer to

$$(\Delta u)_{crit} > \Delta u, \tag{4.37}$$

as the *depth positivity condition*.

4.3 Dry State Riemann Problems

4.3.1 Incipient Cavitation

If $\Delta u > (\Delta u)_{crit}$, then $f(h)$ does not have a physical root, and the middle state h^* cannot be determined by (4.31). Physically this occurs in the case of a strong double rarefaction that leads to a dry middle state depth h^*. Figure 4.3 shows the depth from an incipient cavitation Riemann problem, where initially $h_l = h_r = 1$, and $u_r = -u_l = 8$. Mathematically, cavitation occurs when the integral curves in the first and second fields reach $h = 0$ before intersecting in the phase plane. Figure 4.4 shows the integral curves in the h–hu and h–u phase planes for the same Riemann problem. The Riemann invariants hold through these integral curves as $h \longrightarrow 0$. A dry middle state $h^* = 0$ is still physically admissable, since conservation is maintained throughout the smooth integral curve.

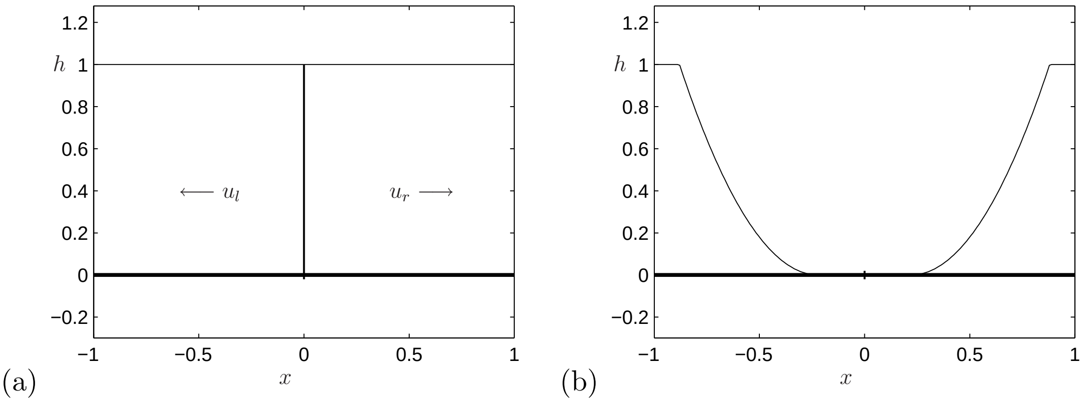


Figure 4.3: An incipient cavitation Riemann problem. (a) The Riemann problem at $t = 0$, with $\Delta u > (\Delta u)_{crit}$. (b) The Riemann problem at $t = 3$. The strong double rarefaction creates an expanding dry middle region.

The speed of the dry/wet interfaces can be determined by the integral curves. For instance, to determine the speed of the wet/dry interface to the left of the dry bed, the 1-Riemann invariant holds throughout the integral curve connecting the left state q_l to the state just to the left of the dry region, denoted q^{*-}. Since $h^{*-} = 0$, applying (4.19) gives

$$u^{*-} = u_l + 2\sqrt{gh_l}. \tag{4.38}$$

The wet/dry interface is moving at the speed of these particles,

$$\lambda^1(q^{*-}) = u^{*-} = u_l + 2\sqrt{gh_l}. \tag{4.39}$$

By similar analysis, the speed of the wet/dry interface to the right of the dry region is

$$\lambda^2(q^{*+}) = u^{*+} = u_r - 2\sqrt{gh_r}. \tag{4.40}$$

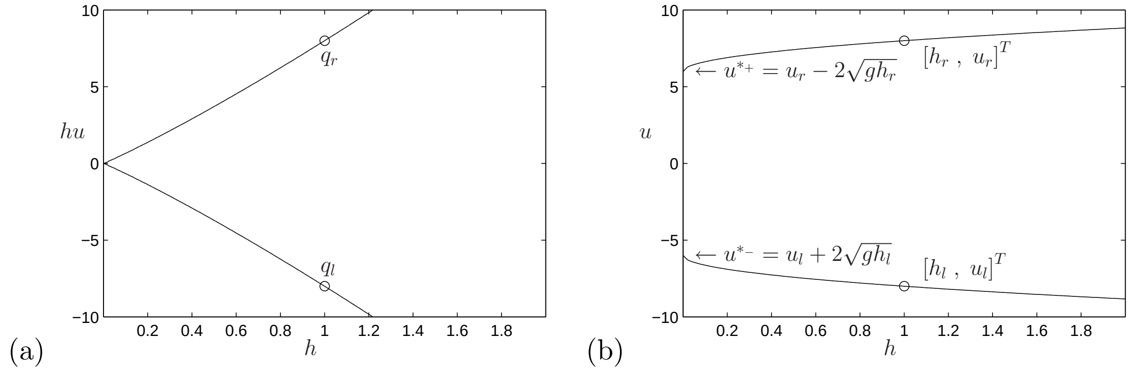


Figure 4.4: Integral curves for the incipient cavitation problem with $\Delta u > (\Delta u)_{crit}$. (a) Curves plotted in the h–hu phase plane. The curves reach the origin $h = 0$ before intersecting. (b) Curves in the h–u plane. The r^1 integral curve approaches $h = 0$ at the point $u^{*-} = u_l + 2\sqrt{gh_l}$ and the r^2 integral curve approaches $h = 0$ at $u^{*+} = u_r - 2\sqrt{gh_r}$. For the solution shown in these figures $g = 1$.

Figure 4.5 shows the characteristics in x-t plane of the rarefactions surrounding the expanding dry region.

Note that $\Delta u > (\Delta u)_{crit}$ implies that

$$\lambda^2(q^{*+}) > \lambda^1(q^{*-}), \tag{4.41}$$

which is consistent with existence of a growing dry region. In fact, $\Delta u = (\Delta u)_{crit}$ precisely when

$$\lambda^2(q^{*+}) = \lambda^1(q^{*-}), \tag{4.42}$$

and the dry region appears as a single point. At a wet/dry interface each respective characteristic field becomes linearly degenerate, since at this point in the phase plane $\nabla\lambda^{1,2}(q)$ are undefined. Because of this, it is useful to think of each wet/dry interface as a contact discontinuity in its respective characteristic field, and note that the fluid at each respective interface was initially in contact at $x = 0$ in the Riemann problem.

4.3.2 Initial Dry Bed

Another type of dry state Riemann problem occurs when initially either $h_l = 0$ or $h_r = 0$. In this type of Riemann problem, the wet state must be connected to the dry state through some set of waves. For instance, consider Figure 4.6, where initially $h_l > 0$ and $h_r = 0$. Conceivably some combination of a 1-wave and 2-wave could connect q_l to q^* and q^* to q_r respectively. However, if $h^* > 0$, Theorem 4.1 requires that a 2-shock connect q^* to the dry

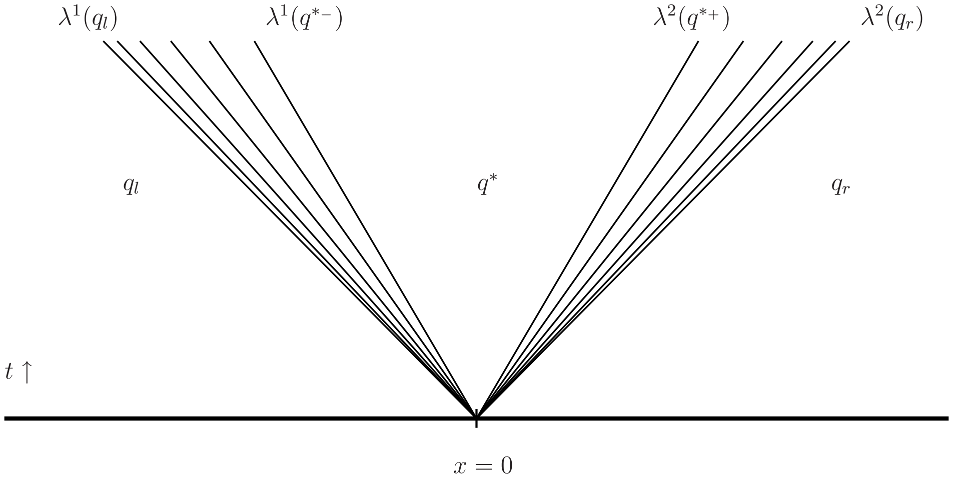


Figure 4.5: The incipient cavitation Riemann problem in the x-t plane. A strong double rarefaction leads to a dry middle state $h^* = 0$. The wet/dry interface on the left moves with speed $\lambda^1(q^{*-}) = u_l + 2\sqrt{gh_l}$ and the wet/dry interface on the right moves with speed $\lambda^2(q^{*+}) = u_r - 2\sqrt{gh_r}$.

state q_r. But this impossible since the Rankine-Hugoniot jump conditions require

$$s\,(h_r - h^*) = hu_r - hu^*, \tag{4.43a}$$

$$s\,(hu_r - hu^*) = \left(hu^2 + \frac{1}{2}gh^2\right)_r - \left(hu^2 + \frac{1}{2}gh^2\right)^*, \tag{4.43b}$$

where s is the speed of the shock. Since $h_r = 0$, eliminating s and solving for h^* in (4.43) gives $h^* = 0$. Clearly then, the middle state must also be a dry state, $h^* = 0$, and the 2-wave is nonexistent (see also [101]). Since the left state q_l is connected to the dry middle state through a 1-wave, Theorem 4.1 implies that the 1-wave is a rarefaction.

Since the 1-Riemann invariants hold through the r^1 integral curve, the speed of the wet/dry interface can be found to be

$$\lambda^1(q^{*-}) = u^{*-} = u_l + 2\sqrt{gh_l}. \tag{4.44}$$

This is exactly analogous to the incipient cavitation problem described previously, except that only the left rarefaction exists. For the initial dry bed problem where $h_r > 0$ and $h_l = 0$, only the 2-rarefaction exists, and the speed of the wet/dry interface is

$$\lambda^2(q^{*+}) = u^{*+} = u_r - 2\sqrt{gh_r}. \tag{4.45}$$

Figure 4.7 shows the characteristic structure for dry bed Riemann problems. If q_l and q_r are the same as in the incipient cavitation problem, the 1-waves when $h_r = 0$ and the 2-waves when $h_l = 0$ are identical to the corresponding waves in the incipient cavitation problem.

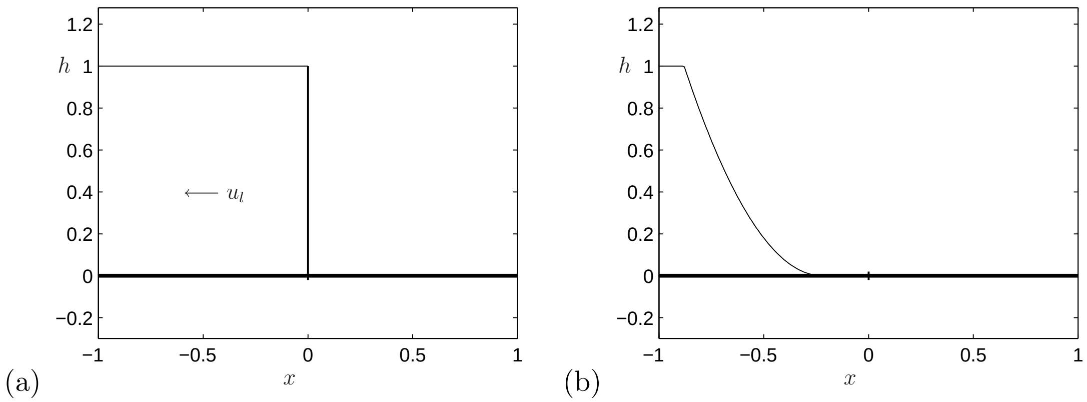


Figure 4.6: An initial dry bed Riemann problem. (a) The Riemann problem initially has a dry state $h_r = 0$ adjacent to a wet state h_l. (b) The solution at $t = 3$ where q_l is identical to that shown in Figure 4.3. The rarefaction has the same structure as the 1-rarefaction in Figure 4.3.

This is physically intuitive, since, for the incipient cavitation problem, the structure of each rarefaction wave should not depend on the other wave since they are separated by a dry region, and the waves should be identical to the corresponding single rarefaction waves in the dry bed problems. Figure 4.7 shows the individual rarefaction waves for the two respective dry bed problems.

Of course, in the case of an initial dry bed, a rarefaction connects the wet state to the dry state regardless of the initial velocity in the wet state. Contrary to the examples shown in the Figure 4.7, the wet/dry front can advance at considerable speeds, $u_{l,r} \pm 2\sqrt{gh_{l,r}}$, which can be large if the wet state has significant depth. In fact, the advancing speeds are much larger than standard shock speed estimates in the case of very shallow or vanishing depth Riemann problems. For instance, the Roe speed estimates for Riemann problems where $h_{r,l} \to 0$ are $u_{l,r} \pm \frac{1}{\sqrt{2}}\sqrt{gh_{l,r}}$. See [101] for a further discussion about the possibility of underestimating the speed of the advancing wet/dry front.

4.4 Steady States and Source Terms

In this section we will discuss steady state solutions to the homogeneous and nonhomogeneous shallow water equations. We begin with a theorem about hyperbolic systems in general that applies to the homogeneous shallow water equations.

Theorem 4.3. *The only smooth (differentiable) steady state solution of a strictly hyperbolic fully nonlinear system*

$$q_t + f(q)_x = 0, \tag{4.46}$$

is the trivial steady state $q_x \equiv 0$.

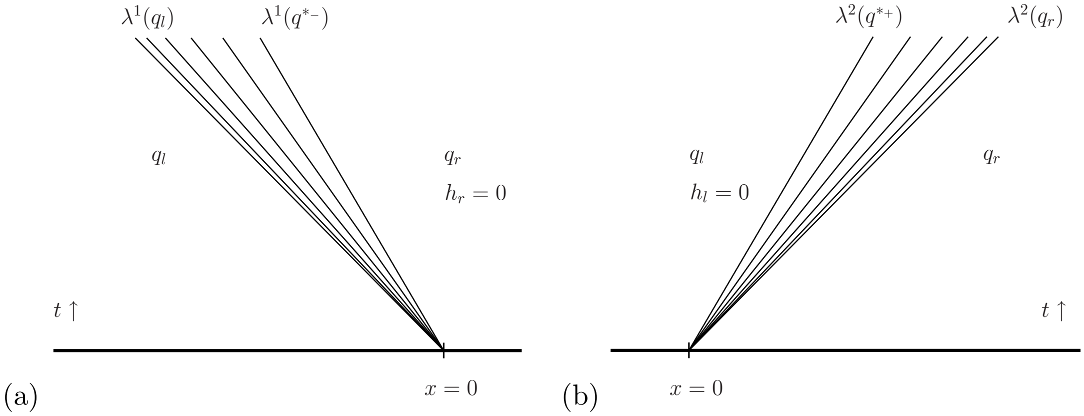


Figure 4.7: The initial dry bed Riemann problems in the x-t plane. (a) The 1-wave when $h_r = 0$. The wet/dry interface moves with speed $\lambda^1(q^{*-}) = u_l + 2\sqrt{gh_l}$. (b) The 2-wave when $h_l = 0$. The wet/dry interface moves with speed $\lambda^2(q^{*+}) = u_r - 2\sqrt{gh_r}$.

Proof. A steady state solution $q(x, \cdot)$ to (4.46) implies that

$$f(q)_x = f'(q)q_x = 0. \tag{4.47}$$

If q_x is nonzero, then (4.47) implies that q_x is an eigenvector of $f'(q)$ with eigenvalue equal to zero. This implies that the solution is a simple wave

$$\tilde{q}'(\xi) = \alpha(\xi) r^p\left(\tilde{q}(\xi)\right), \tag{4.48}$$

where $\tilde{q}(\xi) \equiv q(x, \cdot)$, and

$$\lambda^p(\tilde{q}(\xi)) \equiv 0. \tag{4.49}$$

However, the simple wave is an integral curve of r^p, a fully nonlinear field, which requires that $\nabla\lambda^p(\tilde{q}) \cdot r^p(\tilde{q}) \neq 0$. Therefore λ^p cannot be identically zero through the simple wave, and it must be that $q_x \equiv 0$. □

Theorem 4.3 means that steady state solutions to systems such as the homogeneous shallow water equations, are either trivial uniformly constant functions, or perhaps piecewise constant functions with stationary jump discontinuities. If stationary jump discontinuities exist, they are governed by the Rankine-Hugoniot conditions

$$s\left[[q]\right] = \left[[f(q)]\right]. \tag{4.50}$$

Since $s = 0$,

$$f(q^+) = f(q^-), \tag{4.51}$$

across the discontinuity, where q^+ and q^- are the constant states to the immediate left and right of the jump. For the shallow water equations, (4.51) implies that hu and the momentum flux $(hu^2 + \frac{1}{2}gh^2)$ do not vary across a stationary discontinuity. Since the momentum flux is a nonlinear function of h and hu, in general it is possible to have a stationary jump in h. This jump must also satisfy entropy conditions.

4.4.1 Source Terms

More complex steady states exist when source terms are included,

$$q_t + f(q)_x + \psi, \tag{4.52}$$

since the balance of ψ and the flux gradient $f(q)_x$ imply that $q_t = 0$. Note that for the shallow water equations over variable bathymetry,

$$h_t + (hu)_x = 0, \tag{4.53a}$$

$$hu_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x = -ghb_x, \tag{4.53b}$$

this occurs when

$$(hu)_x = 0, \tag{4.54a}$$

$$\left(hu^2 + \frac{1}{2}gh^2\right)_x = -ghb_x. \tag{4.54b}$$

As with the homogeneous shallow water equations, (4.54a) implies that hu is always constant for steady states, otherwise mass would accumulate in certain regions. However, (4.54b) implies that the momentum flux must balance the source term for steady states. Since the momentum flux is only a function of h and hu, h must vary through a steady state where $b_x \neq 0$. If we differentiate the momentum flux in (4.54b), while holding hu constant, we have

$$\left(-u^2 + gh\right) h_x = -ghb_x. \tag{4.55}$$

Since the eigenvalues for the shallow water equations satisfy $-\lambda^1\lambda^2 = -u^2 + gh$, it should be noted that a smooth steady state over sloping bathymetry cannot have a sonic point. Therefore, for regions with monotonically varying bathymetry, steady-state flow is either entirely subsonic or entirely supersonic. If the flow is subsonic $\left(-u^2 + gh > 0\right)$, h_x and b_x are opposite in sign, with the opposite being true if the flow is supersonic. The transition can only occur at an extremum of the bathymetry where $b_x = 0$.

In section Section 2.3, it was mentioned that the Rankine-Hugoniot conditions are unchanged in the presence of a bounded source term. Therefore, for a steady state in the form of a stationary jump discontinuity, (4.51) still holds as long as b_x is defined continuously, or if b_x is bounded in a neighborhood of the jump. However, for developing numerical methods, it is useful to consider $b(x)$ as a piecewise constant function, and allow b_x to be a sum of generalized distributions equal to a delta function at each jump discontinuity,

$$b_x \equiv \sum_{x_j \in \mathcal{X}} \left(b(x_j^+) - b(x_j^-)\right) \delta(x - x_j), \tag{4.56}$$

where $\mathcal{X}$ contains the locations of the jump discontinuities in $b(x)$, and $\delta(x)$ is the delta function defined by

$$\int \delta(x) f(x) = f(0). \tag{4.57}$$

The source term in the shallow water equations, $-ghb_x$, is not necessarily well defined by (4.56), even in the sense of a distribution, since it involves h, which is possibly (in fact necessarily for steady states) discontinuous at $x_j \in \mathcal{X}$. This creates a problem for the theory of distributions involving overlapping discontinuities. See also [26], and for some discussion of overlapping discontinuities and distribution theory, see [18]. However, we suspend uncertainty temporarily and assume that the source term is a sum of point sources at the locations x_j. We write the source term as a sum of distributions

$$-ghb_x = \psi = \sum_{x_j \in \mathcal{X}} \Psi_j \delta(x - x_j), \tag{4.58}$$

which integrated across a jump discontinuity gives

$$\int_{x_j^-}^{x_j^+} \psi \, dx = \Psi_j. \tag{4.59}$$

If we then apply the integral conservation law to a neighborhood around the single discontinuity at x_j, we have

$$(hu)^+ = (hu)^-, \tag{4.60a}$$

$$\left(hu^2 + \frac{1}{2}gh^2\right)^+ - \left(hu^2 + \frac{1}{2}gh^2\right)^- = \Psi_j, \tag{4.60b}$$

where for notational ease the superscripts $(\cdot)^+, (\cdot)^-$ denote evaluation at $x_j^{+,-}$. The conditions (4.60) are similar to the Rankine-Hugoniot conditions for a stationary discontinuity over continuous bathymetry (4.51), except now the infinite source term contributes a right hand side to (4.60b). This follows physical intuition, since if the jump in momentum flux did not balance the point source of momentum, the solution would quickly become unbounded in the neighborhood of the discontinuity. As with (4.51), the momentum hu is constant and so a jump in momentum flux implies that only h varies across the stationary discontinuity. This is what causes the difficulty in defining the source term by (4.56) and (4.57), since

$$\int -gh(x)\left(b(x_j^+) - b(x_j^-)\right)\delta(x - x_j), \tag{4.61}$$

is not clearly defined when h is discontinuous at x_j. Note however that (4.60) defines Ψ_j in terms of the left and right states, $q^{-,+}$, on either side of the discontinuity.

It is informative to consider the steady state problem with zero velocity, where $\eta \equiv h + b \equiv 0$ everywhere. Based on physical intuition, this is obviously the correct limiting steady state solution even when $b(x)$ is a piecewise constant function. Evaluating (4.60) using $u^+ = u^- = 0$ and $h = -b$ gives

$$\begin{aligned}\Psi_j &= \left(\frac{1}{2}gh^2\right)^+ - \left(\frac{1}{2}gh^2\right)^- \\ &= \frac{g}{2}\left(h^+ + h^-\right)\left(h^+ - h^-\right) \\ &= -g\frac{\left(h^+ + h^-\right)}{2}\left(b^+ - b^-\right) \\ &= -g\bar{h}_j\left(b^+ - b^-\right), \end{aligned} \tag{4.62}$$

where $\bar{h}_j$ is the arithmetic average of h to either side of the discontinuity. For this steady state problem then, note that the source term could be defined as a distribution using (4.56) and the definition

$$\int f(x)\delta(x) \equiv \frac{1}{2}\left(f(0^+) + f(0^-)\right), \tag{4.63}$$

when f is piecewise continuous. However, as will be shown, this is not correct for more general cases, since (4.60) requires a more complex average of h involving the velocities on either side of the discontinuity.

4.4.2 Conditions on Smooth Steady States

When discussing a point source approximation for $\int -ghb_x\,dx$, what we are really seeking is the correct limiting behavior of smooth solutions over variable bathymetry as b approaches a piecewise constant function, so that we can determine a value for $\int -ghb_x\,dx$ when the solution is only known outside of regions where b_x is nonzero. It is not immediately clear that this will be possible, however, it is possible to make some precise statements regarding all differentiable steady state solutions over regions of variable bathymetry. If we allow b to vary only in isolated regions then, we can analyze the difference in the solution across these regions.

Suppose that b is a slightly smoothed version of a piecewise constant that is differentiable everywhere, so that b_x is classically defined everywhere, and is nonzero only in neighborhoods $[x_j - \epsilon_j, x_j + \epsilon_j]$ of isolated points x_j. In the smoothed version, b varies monotonically from $b(x_j - \epsilon)$ to $b(x_j + \epsilon)$, so that b_x does not change sign in the interval $[x_j - \epsilon, x_j + \epsilon]$, and of course

$$\int_{x_j-\epsilon_j}^{x_j+\epsilon_j} b_x\,dx = (\Delta b)_j, \tag{4.64}$$

where $(\Delta b)_j$ is the magnitude of the jump at x_j in the piecewise constant version. (Details of such a construction—a smoothed infinitely differentiable function satisfying the above properties—can be found in [22].) For simplicity, in the following discussion we will assume that b varies on only one interval around $x = 0$, $I_\epsilon = [-\epsilon, \epsilon]$, and that $b(x) = b_l$ for $x < -\epsilon$ and $b(x) = b_r$ for $x > \epsilon$. Figure 4.8 shows an example of a smoothed function b with such properties.

For smooth steady state solutions, Theorem 4.3 implies that the solution is uniform outside of $(-\epsilon, \epsilon)$, and we denote $q(x,t) = q_l$ for $x < -\epsilon$, and $q(x,t) = q_r$ for $x > \epsilon$. We

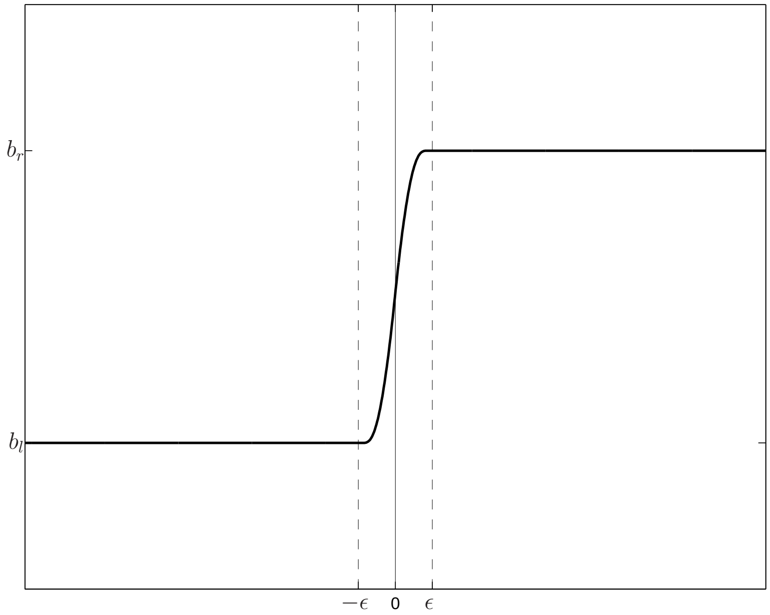


Figure 4.8: A smoothed version of piecewise constant bathymetry. The function shown is $C^\infty(\mathbb{R})$, and b_x has compact support on $[-\epsilon, \epsilon]$. Additionally the function is symmetric, $b(0) = (b_r + b_l)/2$.

can derive some conditions relating the left and right states, q_l and q_r, and the "jump" in bathymetry

$$\Delta b = (b_r - b_l). \tag{4.65}$$

(In many of the equations that follow, the operator Δ will refer to differencing the quantities from the right to the left $\Delta(\cdot) = (\cdot)_r - (\cdot)_l$, where the subscripts refer to evaluating the quantities in the constant region to the right and left respectively.) First, $(hu)_x = 0$ everywhere, so

$$\Delta hu = hu_r - hu_l = 0. \tag{4.66}$$

Using this fact, we can relate the jump in the momentum flux,

$$\Delta\left(hu^2 + \frac{1}{2}gh^2\right) = \left(hu^2 + \frac{1}{2}gh^2\right)_r - \left(hu^2 + \frac{1}{2}gh^2\right)_l, \tag{4.67}$$

to the difference or jump in the depth

$$\Delta h = (h_r - h_l). \tag{4.68}$$

We rearrange (4.60b),

$$
\begin{aligned}
\Delta\left(hu^2+\frac{1}{2}gh^2\right) &= \left(\frac{(hu)^2}{h_r}-\frac{(hu)^2}{h_l}\right)+\frac{g}{2}\left(h_r+h_l\right)\left(h_r-h_l\right)\\
&= \left(\frac{|(hu)_r(hu)_l|}{h_r}-\frac{|(hu)_l(hu)_r|}{h_l}\right)+\frac{g}{2}\left(h_r+h_l\right)\left(h_r-h_l\right)\\
&= \left(h_l|u_r u_l|-h_r|+u_l u_r|\right)+\frac{g}{2}\left(h_r+h_l\right)\left(h_r-h_l\right)\\
&= \Delta h\left(-|u_r u_l|+\frac{g}{2}\left(h_r+h_l\right)\right),
\end{aligned}
\tag{4.69}
$$

where $h \geq 0$ and $hu_r = hu_l$ together imply that $u_r u_l = |u_r u_l|$. If we denote the averages

$$\bar{h} = \tfrac{1}{2}(h_r+h_l), \tag{4.70a}$$

$$\widetilde{u^2} = |u_r u_l|, \tag{4.70b}$$

we can rewrite (4.69) as

$$\Delta\phi = \left(-\widetilde{u^2}+g\bar{h}\right)\Delta h, \tag{4.71}$$

where ϕ is the momentum flux $\phi = (hu^2+\frac{1}{2}gh^2)$. This notation for the momentum flux will be used frequently in the remaining chapters. The ratio implied by (4.71) is essentially a discrete version of the partial derivative of the momentum flux ϕ with respect to h, while holding hu constant,

$$\left(\frac{\partial\phi}{\partial h}\right)_{hu} = -u^2+gh, \tag{4.72}$$

and requires the special averages (4.70) of the left and right states when Δq is not infinitesimal. Equation (4.71) may also be viewed as an exact first order Taylor expansion, where the partial derivative is evaluated at the intermediate state defined by (4.70).

We can also relate Δh to Δb for smooth solutions. Consider (4.55), which is exactly true for differentiable steady state solutions. Dividing by h gives

$$\left(-\frac{(hu)^2}{h^3}+g\right)h_x = -gb_x. \tag{4.73}$$

By noting that $(hu)_x \equiv 0$ for steady states, it can be seen that (4.73) is equivalent to

$$\left(\frac{1}{2}\frac{(hu)^2}{h^2}+gh+gb\right)_x = 0, \tag{4.74}$$

which implies that $\left(\frac{1}{2}u^2+gh+gb\right)$ is a constant for a differentiable steady state. So it must be true that

$$\left(\frac{1}{2}u^2+gh+gb\right)_r = \left(\frac{1}{2}u^2+gh+gb\right)_l. \tag{4.75}$$

Multiplying (4.75) by $\bar{h} = \frac{1}{2}(h_r + h_l)$ and rearranging gives,

$$\begin{aligned}
-g\bar{h}\Delta b - g\bar{h}\Delta h &= \left(\frac{h_r + h_l}{2}\right)\left(\frac{u_r^2 - u_l^2}{2}\right) \\
&= \left(\frac{1}{2}\right)^2 \left(h_r u_r^2 + h_l u_r^2 - h_r u_l^2 - h_l u_l^2\right) \\
&= \left(\frac{1}{2}\right)^2 \left(\left(2h_r u_r^2 - h_r u_r^2\right) + h_l u_r^2 - h_r u_l^2 - \left(2h_l u_l^2 - h_l u_l^2\right)\right) \\
&= \left(\frac{1}{2}\right)^2 \left(\left(2h_l u_l u_r - h_r u_r^2\right) + h_l u_r^2 - h_r u_l^2 - \left(2h_r u_r u_l - h_l u_l^2\right)\right) \\
&= \left(\frac{1}{2}\right)^2 \left(\left(h_l u_l^2 + 2h_l u_l u_r + h_l u_r^2\right) - \left(h_r u_l^2 + 2h_r u_l u_r + h_r u_r^2\right)\right) \\
&= \left(\frac{1}{2}\right)^2 \left(h_l \left(u_r + u_l\right)^2 - h_r \left(u_r + u_l\right)^2\right) \\
&= -\left(\frac{u_r + u_l}{2}\right)^2 \Delta h, \qquad (4.76)
\end{aligned}$$

where the fourth equality makes use of $h_l u_l = h_r u_r$. Equation (4.76) implies the relationship

$$\left(-\bar{u}^2 + g\bar{h}\right)\Delta h = -g\bar{h}\Delta b, \qquad (4.77)$$

where $\bar{u}$ is the arithmetic average $\frac{1}{2}\left(u_l + u_r\right)$. Note that this is a discrete representation of (4.55), or an exact first order Taylor expansion for steady states, using the arithmetic averages of h and u.

Since (4.71) relates Δh to $\Delta\phi$, equation (4.77) allows us to relate $\Delta\phi$ to Δb,

$$\Delta\phi = -g\bar{h}\Delta b\left(\frac{-\widetilde{u^2} + g\bar{h}}{-\bar{u}^2 + g\bar{h}}\right). \qquad (4.78)$$

By applying conservation,

$$\Delta\phi = \int_{I_\epsilon} -ghb_x\,dx, \qquad (4.79)$$

the right hand side of equation (4.78) provides the correct formulation of the source term for all smooth steady states across an interval of varying bathymetry. For these solutions, it must be true that

$$\int_{I_\epsilon} -ghb_x\,dx = -S_-^+ g\bar{h}\Delta b. \qquad (4.80)$$

where,

$$S_-^+ = \left(\frac{-\widetilde{u^2} + g\bar{h}}{-\bar{u}^2 + g\bar{h}}\right). \qquad (4.81)$$

The ratio S_-^+ is a nondimensional quantity that provides the correct source term integral when multiplied by $-g\bar{h}\Delta b$, for smooth steady states across any region of varying bathymetry $b_r - b_l = \Delta b$. (The notation is meant to resemble an integral, since the product $-S^+_- g\bar{h}\Delta b$ is equal to the value of the integral $-\int_{I_\epsilon} ghb_x\, dx$.) Note that for smooth solutions, if $q_l \to q_r$, then $S^+_- \to 1$. It might at first seem curious that S^+_- could be negative, even if b_x does not change sign, or worse, the denominator might vanish. However, recall that we have found that steady flow cannot have a sonic point when b varies monotonically by (4.55), therefore, such flows must be entirely subsonic or supersonic. By requiring the Froude number, $|u/gh|$, to be either strictly less than or more than one for all x, it is easy to demonstrate that the ratio (4.81) is always positive for these flows. In fact, since

$$\bar{u}^2 \geq u_r u_l, \qquad \forall u_l, u_r \in \mathbb{R}, \tag{4.82}$$

with equality only when $u_r = u_l$, it follows that S^+_- is less than unity for supersonic flow, and greater than or equal to unity for subsonic flow, with equality only for stationary flow with zero velocity—following from the fact that $u_l = u_r$ implies that $h_l = h_r$ unless $u_l = u_r = 0$. (Equality of depth across variable bathymetry is a contradiction because of (4.55).) For $u \equiv 0$, the ratio $S^+_- = 1$ produces the desired result (4.62) for the source term.

Equation (4.55) relating h_x and b_x for smooth steady flows, allows further characterization of $-g\bar{h}\Delta b S^+_-$. Since the flow in question has no sonic points, h_x cannot change sign, and h must vary monotonically from h_l to h_r across the interval where b varies. Therefore, the integral for the source term can be bounded,

$$\begin{aligned} |S^+_- g\bar{h}\Delta b| &= |\int_{I_\epsilon} ghb_x\, dx| \\ &\leq g\max(h_l, h_r)|\int_{I_\epsilon} b_x\, dx| \\ &= g\max(h_l, h_r)|\Delta b|, \end{aligned} \tag{4.83}$$

and similarly

$$|S^+_- g\bar{h}\Delta b| \geq g\min(h_l, h_r)|\Delta b|. \tag{4.84}$$

We can therefore state

$$\min(h_l, h_r) \leq S^+_- \bar{h} \leq \max(h_l, h_r), \tag{4.85}$$

for smooth steady states.

4.4.3 Steady State Data

In this final section, the properties of smooth steady state solutions over monotonically varying bathymetry will be summarized by presenting them in a simple unified manner. A discontinuous steady state will be briefly discussed as well. We will consider only a continuously differentiable bathymetry function $b(x)$, such as that discussed in the previous section and depicted in Figure 4.8, where $b(x) = b_l$ to the left of an interval I_ϵ and $b(x) = b_r$ to the right of the interval, and $b(x)$ varies monotonically in the interval. It has been shown that this implies that the steady state solution must also be uniform to the left and right of the interval, and h varies monotonically in the interval. Concisely summarizing the properties that must be satisfied by the left and right states is the goal of this section.

It will be convenient to define some of the averages discussed above, as operators acting on two states q_l and q_r. First, we will define the operators

$$\overline{\lambda^1\lambda^2}(q_l, q_r) = -\left(-\left(\frac{u_l + ur}{2}\right)^2 + g\left(\frac{h_r + h_l}{2}\right)\right), \tag{4.86}$$

and

$$\widetilde{\lambda^1\lambda^2}(q_l, q_r) = -\left(-u_l u_r + g\left(\frac{h_r + h_l}{2}\right)\right). \tag{4.87}$$

The negative signs are introduced in the right hand sides of (4.86) and (4.87), so that the operators represent an approximation to the product of the eigenvalues of the shallow water equations $\lambda^1(q)\lambda^2(q) = (u^2 - gh)$. We also introduce the operator

$$\mathcal{S}_-^+(q_l, q_r) = -\frac{g}{2}\left(h_r + h_l\right)\frac{\widetilde{\lambda^1\lambda^2}(q_l, q_r)}{\overline{\lambda^1\lambda^2}(q_l, q_r)}, \tag{4.88}$$

which is equivalent to (4.81), except that we have absorbed the $-g\bar{h}$. The operator then satisfies the following property,

$$\mathcal{S}_-^+(q_l, q_r)\Delta b = -\int ghb_x\, dx, \tag{4.89}$$

if it is evaluated by left and right states that correspond to a smooth steady state. We now state some results as a theorem, for emphasis and for convenient reference in the following chapters.

Theorem 4.4. *Given a monotonically varying smooth bathymetry function such as that described above, the following property must be satisfied for all smooth steady state data, q_l and q_r,*

$$\Delta\begin{bmatrix} h \\ hu \\ \phi \\ b \end{bmatrix} = \Delta b \begin{bmatrix} \frac{\mathcal{S}_-^+(q_l,q_r)}{-\widetilde{\lambda^1\lambda^2}(q_l,q_r)} \\ 0 \\ \mathcal{S}_-^+(q_l, q_r) \\ 1 \end{bmatrix}, \tag{4.90}$$

where Δ implies subtracting the quantities evaluated with data q_l from those evaluated with q_r.

Proof. The theorem follows from simply considering the ratios developed in the previous section, and the fact that $(hu)_x \equiv 0$. □

It is interesting to consider what happens if we allow a discontinuity in the steady state over monotonically varying smooth bathymetry. For instance, suppose that the solution to the left of the interval I_ϵ, $q(x < -\epsilon, t) = q_l$, is joined to solution to the right of I_ϵ, $q(x > \epsilon, t) = q_r$, through a smooth transition on either side of a discontinuity at $x = 0$, with $q(0^-, t) = q^-$ and $q(0^+, t) = q^+$. Recall that when b is continuously differentiable, a discontinuous solution is still governed by the Rankine-Hugoniot (4.50) conditions for a shock. For a steady state, the shock is stationary, and the fluxes must be constant across the discontinuity (4.51). So we have, assuming that $b(x)$ is continuously differentiable,

$$\Delta \begin{bmatrix} h \\ hu \\ \phi \\ b \end{bmatrix} = \Delta h \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.91}$$

where Δ is the difference just across the discontinuity.

It should be noted that the relation (4.71) is valid across the discontinuity, and because the flux is constant it must be that $\widetilde{\lambda^1\lambda^2}(q^-, q^+) = 0$. In fact, $\widetilde{\lambda^1\lambda^2}(q^-, q^+) = 0$ for any stationary discontinuity where q^- and q^+ are the states to the left and right of the discontinuity, which follows from relating (4.55) to the Rankine-Hugoniot conditions governing a stationary shock.)

Conservation applied around the interval I_ϵ still requires that

$$f(q_r) - f(q_l) = -\int_{I_\epsilon} ghb_x \, dx. \tag{4.92}$$

Since the right hand side of (4.92) is nonzero, $f(q^-) = f(q^+)$, and hu does not vary, h *must* vary through the smooth regions and the total change $(h_r - h_l)$ must be nonzero. Therefore, over a region of monotonically varying bathymetry, q_l *cannot* be connected to q_r with a single stationary shock. Since (4.71) still holds using q_l and q_r, it cannot be that $\widetilde{\lambda^1\lambda^2}(q_l, q_r) = 0$. Therefore, the total jump $\Delta h = (h_r - h_l)$ is well defined by (4.71). Since a stationary shock implies that the flow is supersonic to one side of the shock and subsonic to the other, it is not clear if a general statement can be made about the sign of $\widetilde{\lambda^1\lambda^2}(q_l, q_r)$, however, it is apparent that h_x is positive to one side of the discontinuity and negative to the other because of (4.55). We can summarize these properties most easily by first defining the augmented state variable

$$\tilde{q}(q, b) = \begin{bmatrix} h \\ hu \\ \phi \\ b \end{bmatrix}. \tag{4.93}$$

For all steady state data, q_l and q_r, to the left and right of a region of monotonically smoothly varying bathymetry, $\tilde{q}(q_r, b_r)$ and $\tilde{q}(q_l, b_l)$ can be connected by a linear combination of the two vectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \frac{-\int ghb_x}{-\widetilde{\lambda^1\lambda^2}(q_l, q_r)} \\ 0 \\ -\int ghb_x \\ b_r - b_l \end{bmatrix}, \tag{4.94}$$

where $\widetilde{\lambda^1\lambda^2}(q_l, q_r)$ is guaranteed to be nonzero. Additionally, when the solution is continuously differentiable

$$-\int ghb_x \, dx = \mathcal{S}_-^+(q_l, q_r)\Delta b. \tag{4.95}$$

Chapter 5

NUMERICAL TSUNAMI MODELING

Physical aspects of tsunamis will be addressed in this chapter, and the resulting challenges that must be confronted by numerical modelers will be described. We begin by first considering the generation of tsunamis and governing approximations for tsunami flows.

5.1 Tsunami Generation and Governing Assumptions

Tsunamis can be generated by a variety of geophysical disturbances—most commonly earthquakes, landslides, or some combination of the two. Because of the diverse physical nature of these disturbances, tsunamis can vary greatly in their size and extent, however, all can be destructive and deadly. Typically, landslide generated tsunamis are more localized, yet, large submarine landslides can generate oceanic tsunamis impacting an entire region. For instance, the Papua New Guinea Tsunami of 1998, which killed over 2000 people, is believed to have been caused by a submarine landslide following an earthquake. A more localized tsunami due to an underwater landslide occurred in 1994 at Skagway Harbor, Alaska, where a cruise ship wharf undergoing construction collapsed [57].

The most notable tsunamis are the large earthquake-generated global-scale tsunamis that can affect coastlines around an entire oceanic region, such as the recent Indian Ocean Tsunami. The generating mechanism for these tsunamis is believed to be the vertical uplift of the water column due to the displacement of the ocean floor, not the seismic waves or the horizontal momentum of the fault plane. It is therefore the geometry of a fault mechanism that determines whether a seismic event is tsunamigenic. If the vertical displacement of the water column is the principle tsunamigenic mechanism, then tsunami waves begin as *long waves*—waves governed by the shallow water equations. This is because the horizontal length scales of most fault displacements are far greater than the depth of the ocean, and additionally, vertical displacement of the ocean floor creates initial conditions that are invariant in the vertical direction—the entire water column is lifted and remains hydrostatic. Recently, it has been argued by some (*e.g.* Rivera [81]) that the horizontal motion of the fault plane is a significant ingredient in tsunami initiation by imparting momentum to the water column. If this is true, then vertically varying horizontal fluid momentum would play a significant role in tsunamigenesis, and these effects would not be governed by the shallow water approximation. However, it is more traditionally believed that such motion is not transferred to the long-wave gravity waves of tsunami propagation. See [92] for a discussion of tsunamigenic sea-floor deformations.

There is growing evidence that dispersive effects may be more important in tsunami propagation than previously thought. See, for instance, [34],[46] and [58] for some empirical arguments. The largest terms neglected in the shallow water equations are dispersive terms, which might be desirable for more accurate modeling of tsunamis in certain cases.

Various equations can be derived that include these terms, by making some assumptions about the *flow regime* (depth, wavelength, amplitude, etc.). Solutions to such equations are often called *weakly-dispersive*, and exist due to a balance of nonlinearities, of the kind present in the shallow water equations, and weakly nonlinear to linear dispersive terms involving third-order spatial derivatives. See a theoretical water-wave treatment such as [50] or [103] for derivations of the classic types of these equations and properties of their solutions. Boussinesq-type equations that include various nonlinear dispersive terms have been used as governing equations for tsunami modeling, however, they are generally valid only for smooth differentiable solutions in particular flow regimes where wavelength and depth are predictably related. Since advection nonlinearities, of the type present in the shallow water equations, dominate as waves shoal in coastal waters, potential steepening and breaking render Boussinesq systems problematic in these regions. Additionally, steep gradients can pose a difficulty for the finite difference solution of high-order derivatives present in Boussinesq systems. However, despite the difficulties, Boussinesq methods continue to be a valuable approach for ocean modeling in certain regimes. See Peregrine [80] for the classic derivation of the conventional depth averaged Boussinesq model, or for more recent advances that are more broadly applicable see [1], [31], [53], [74], [75] or [78].

Like the PDE form of the shallow water equations, Boussinesq equations are free-surface depth-averaged equations, where the effects of nontrivial vertical flow dynamics appear as additional terms to the shallow water equations. An interesting alternative approach for incorporating dispersion is taken by Walters in [102], where the effects of vertical flow acceleration are incorporated directly into the shallow water equations by essentially absorbing nonhydrostatic effects into a spatially and temporally varying gravitational constant. Since the nonlinear shallow water equations can be maintained in this formulation, near-shore flows where nonlinearity dominates could potentially be modeled by methods that can converge to discontinuous weak solutions of the shallow water equations.

Occasionally, modelers have used full three dimensional equations for free surface flows for the purpose of tsunami modeling. These are sometimes referred to as Navier-Stokes solutions, however, often simple elliptic equations are solved for the incompressible internal flow, with appropriate boundary conditions for the free surface. These methods are often too expensive for global-scale tsunami modeling, but rather serve as a test case, or “reality check” on the more efficient approximate governing equations, or for detailed studies of a specific area or event. See, for instance, Mader [73], Gisler *et al* [30], Fujima, Masamura and Goto [25].

Although other governing equations are important for tsunami modeling in certain cases, the shallow water equations describe the general characteristics of tsunamis surprisingly well in many of the important flow regimes—from global propagation to inundation in coastal areas. Moreover, the additional computational cost accrued by solving more computationally expensive systems must be weighed carefully against the added information, even if an accurate numerical solution to such systems can be taken for granted. Certainly for some cases, these benefits will warrant the costs, however, because of their wide applicability and proven empirical success, solutions to the shallow water equations will continue to serve as an invaluable asset for tsunami modelers. Furthermore, the purpose of this chapter is not to advocate the use of a particular set of governing equations, but to demonstrate some properties that are required of numerical methods used for solving the shallow water equations for tsunami modeling. Other governing systems present their own set of numerical challenges—a modern approach for solving Boussinesq systems can be found in [86], and the use of Navier-Stokes equations for tsunami modeling is covered in [73].

5.2 The Diverse Regimes of Tsunami Flow

In this section, we will take the governing equations—the shallow water equations—for granted, and discuss the diversity of flow regimes exhibited by solutions to these equations in the context of tsunamis. These diverse flow regimes demand diverse properties of numerical methods, and it must be emphasized that not all consistent numerical methods for the shallow water equations will produce accurate solutions for flows in these regimes.

In the deep ocean, the wavelength of a typical tsunami wave is very long—on the order of several hundred kilometers—while the depth of the ocean is typically only a few kilometers. If we define the parameter δ to be the ratio of depth divided by wavelength, the shallow water approximation comes from assuming that δ is small resulting in the implication that vertical variation in the flow is negligible. These properties are what allow tsunami energy to propagate thousands of miles away from a source before being dissipated. (This is another argument for the shallow water approximation—large vertical variations in the water column are more quickly dissipated, and not transmitted in the transoceanic waves. Certainly on the sea floor near a rupturing fault, complex, even turbulent flows must exist, but this energy is quickly dissipated, and the potential energy associated with the vertical uplift is what is carried away with the tsunami.) Contrasted to the wavelength, the amplitude of tsunami waves in the deep ocean are very small—typically less than a meter, sometimes only a few centimeters. Given the destructive capacity of tsunamis, this seems peculiar until one considers the potential energy associated with a surface deviation, even less than a meter, spread over hundreds of kilometers. Figure 5.1 shows a cross section of the Bay of Bengal from a simulation of the 2004 Sumatra-Andaman tsunami, demonstrating the scales involved.

As tsunami waves enter shallower waters, or *shoal*, the diffuse energy associated with the long wavelength is compressed and focused. Recall that the wave speeds for the shallow water equations are $u \pm \sqrt{gh}$. In the deep ocean then, the wave speeds are roughly proportional to the square root of depth. As the leading edge of the waves slow in shallow coastal regions, the faster moving tails of the long-waves catch up, and the waves are compressed. As the energy is compressed, the flow regime can change dramatically becoming much more violent. For instance, the waves can steepen dramatically leading to breaking waves and propagating bores, as described in section Section 5.4. The flow is further complicated as waves inundate dry land, and interact with complex shoreline features.

5.3 Deep Ocean Propagation

If one considers the length scales shown in Figure 5.1, a deep ocean tsunami can be thought of as a tiny perturbation to the steady state, albeit spread over vast areas. Consider the common steady state—a motionless pool of water over variable bathymetry—with an

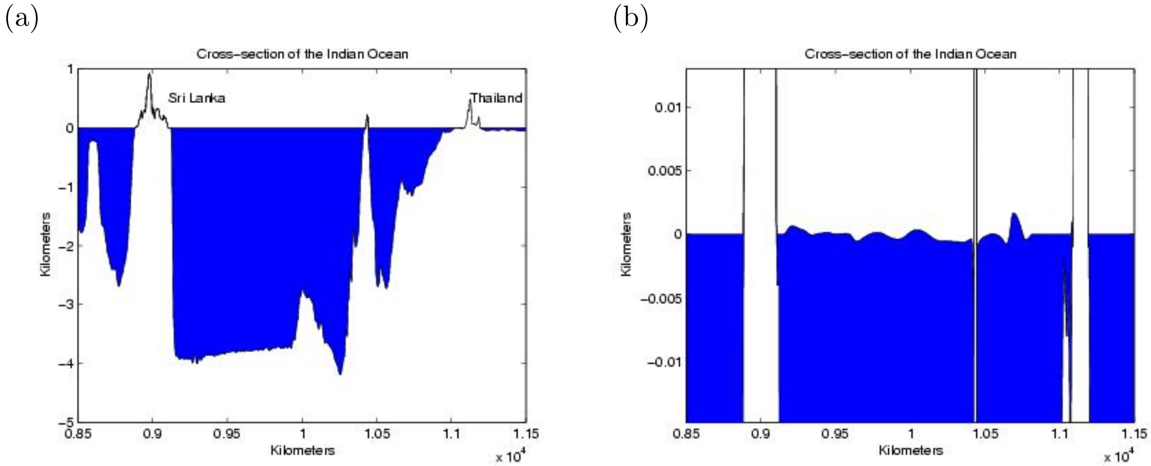


Figure 5.1: Tsunami scales: cross-section of the Bay of Bengal from a simulation of the 2004 Sumatra-Andaman Tsunami. (a) Cross-section showing the entire depth of the ocean—the tsunami is a small (invisible) perturbation to the steady state. (b) Cross-section of the surface for the same simulation. Note the scales of the axes—the wavelength is much greater than the depth, and the amplitude is less than a few meters.

undisturbed depth $H = -b$ (see Figure 5.2). The motionless steady state arises from a nontrivial balance of the source term due to variable bathymetry, and the gradient of the pressure term in the momentum flux,

$$\begin{aligned}\left(hu^2 + \frac{1}{2}gh^2\right)_x &= gHH_x \\ &= -gHb_x. \end{aligned} \tag{5.1}$$

If we consider a small deviation η to the steady state depth,

$$h(x,t) = H(x) + \eta(x,t), \tag{5.2}$$

with some velocity $u(x,t)$, then we can rewrite the shallow water equations

$$h_t + (hu)_x = 0, \tag{5.3a}$$

$$(hu)_t + (hu^2)_x + gh\eta_x = 0. \tag{5.3b}$$

Note that the change in the solution comes from the gradient of the small deviations η and u (small compared to the conserved variables h and hu). When modeling this flow regime numerically, we are attempting to resolve the tiny deviations from the steady state since $H(x) = -b(x)$ is a large fixed constant. If we numerically solve the shallow water equations in conservative form, the flux gradient ghh_x and the source term $-ghb_x$ are magnitudes larger than the actual deviation of interest. These terms must be treated precisely or the

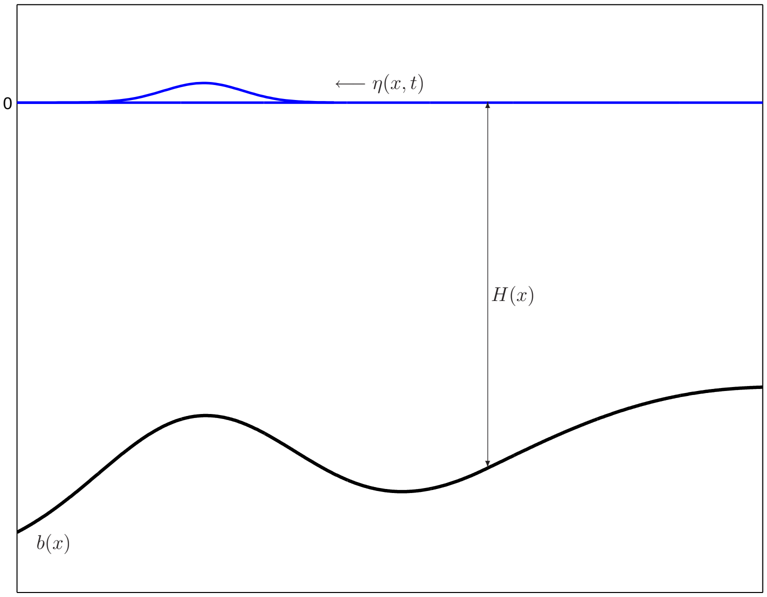


Figure 5.2: Deviation from the physically common steady state: a small perturbation of amplitude η propagates against the background state with depth $H(x) = -b(x)$. The figure does not depict a realistic scale, since for a real tsunami $|\eta| << H$.

solution of interest would be completely lost in noise. For instance, use of the fractional step methods described in Section 3.6, where the homogeneous shallow water equations are solved by standard methods followed by an integration of the source term, would fail miserably for modeling this regime accurately—the numerical noise for typical grid spacings might be larger than the actual tsunami that we are trying to model!

It has been more standard practice in global tsunami modeling to use an alternative form of the shallow water equations, using the non-conserved primitive variables,

$$\eta_t + ((\eta - b)\, u)_x = 0, \tag{5.4a}$$

$$u_t + uu_x + g\eta_x = 0, \tag{5.4b}$$

where $\eta \equiv h + b$. The system (5.4) can be easily derived by manipulating the shallow water equations in their conservative form. Note that for the motionless steady state, all of the gradients in (5.4) are zero. The motionless steady state balance has essentially been removed from the equations (5.4) by canceling terms after taking derivatives. Therefore, numerically preserving the motionless steady state, or resolving small deviations to the steady state is much more easily accomplished. In fact, most standard finite difference methods based on

(5.4) will produce accurate solutions in this regime. The drawback to using (5.4) arises in the other flow regimes of tsunamis.

5.4 Shoaling: Bores and Discontinuities

As the tsunami waves enter shallow waters, the wave energy is compressed and focused leading to more violent flow characteristics. The nonlinearities in the shallow water equations dominate in this flow regime, which can lead to breaking waves and turbulent bores. Figure 5.3 depicts the shoaling of a deep water wave, showing the compression. (Recall the steepening phenomenon present in hyperbolic systems leading to shock-waves, described at the end of Section 2.3.) Standard finite difference solutions to the shallow water equations can be problematic in this regime, leading to spurious numerical artifacts. In fact, the actual governing equations (5.4) are no longer valid near shock waves. In this regime we seek weak solutions to the PDEs (2.1), or solutions to the physically relevant integral form of the shallow water equations.

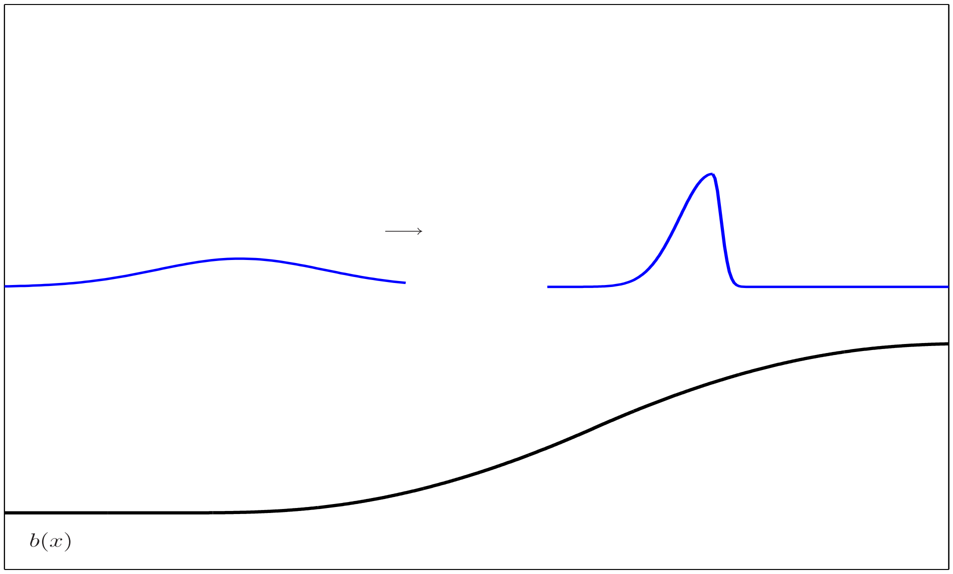


Figure 5.3: Tsunami shoaling: a deep ocean wave is compressed and steepens as the leading edge of the wave slows before the trailing edge. Further steepening would lead to a breaking wave followed by a turbulent bore.

Here we must digress briefly to draw a careful distinction between what we will call a *shock-wave*—a valid discontinuous mathematical solution to the shallow water equations, and a *turbulent bore*—a physical phenomenon prevalent in the near shore region. This semantic distinction seems to be the source of a large amount of misconception and misunderstanding. There are two issues at hand here: first, for modelers, the issue of computing a true shock-wave solution to the shallow water equations is a difficulty, and second, it is not clear how well a shock-wave solution to the shallow water equations represent the physical turbulent bore or breaking wave. The resolution of the latter issue seems to be that shock-waves represent turbulent bores surprisingly well, if one considers only the macroscopic properties of the bore, such as when it forms, its propagation speed, and how it affects inundation (see [44, 97, 98, 108]). Certainly the shallow water equations do not describe the flow features right at the edge of the turbulent region of the bore. However, if the bore is represented mathematically as a discontinuity, the shallow water equations with appropriate entropy conditions provide the correct conservation principles for hydrostatic flow surrounding the bore. For the standard mathematical treatment of this topic, see [50],[87] or [103]. The empirical success of the shallow water equations is nicely summarized by Titov and Synolakis in [97]: "... further manifestation of the well-documented but unexplained ability of the shallow water wave equations to provide quantitatively correct runup results even in parameter ranges where the underlying assumptions of the governing equations are violated."

This work is more concerned with the first issue—numerically converging to correct discontinuous solutions of the shallow water equations. (Regardless of the crudeness of the shock-wave approximation to a true physical bore, certainly a correct solution to the shallow water equations is a better approximation to the bore than a spurious one.) Shock-waves pose a difficulty for numerical solution for several reasons. First, PDE forms that are convenient for many of the flow regimes, such as (5.4), are no longer valid near discontinuities—only the physically relevant integral form of the shallow water equations provides the correct formulation for a shock-wave. Second, nonconservative numerical methods, even those based on the physically conserved quantities, can converge to spurious discontinuous solutions propagating at incorrect speeds. And finally, discontinuities and steep gradients can generate spurious oscillations in numerical methods that destroy accuracy, especially in standard finite difference methods that are based on the validity of Taylor series. It is necessary to use a conservative numerical method that is based on the integral form of the shallow water equations, and use appropriate up-winding near shocks, such as that provided by solving Riemann problems, if we are to converge to correct weak solutions to the shallow water equations. This creates a difficulty for modeling global propagation, as described in the previous section.

5.5 Shorelines and Inundation

Ultimately, the goal of tsunami modeling is to determine what happens at the shoreline, since this is where tsunamis unleash their energy. The difficulties posed by the diverse regimes already described are further exacerbated by the existence of dry regions and shorelines, especially if one attempts to model inundation. Inundation modeling poses several numerical difficulties. First, the shoreline itself is a discontinuity, and methods based on PDEs must use some form of appropriate up-winding at this interface. Second, these regions have vanishing depths, and spurious negative solutions are a ubiquitous problem for numerical methods in this regime. Third, it is not obvious how to determine the movement of the shoreline based on the flow variables and bathymetry near the shore, and rectifying accurate shore movement with positivity is difficult.

There are various types of approaches that have been used for treating the shoreline, the most common of which is to simply avoid inundation modeling by treating the shoreline as a fixed wall boundary. Inundation can then be estimated by runup at the boundary. This approach is certainly not without merit, though it is less common recently since various methods for modeling inundation have become available. Most shoreline treatments might be referred to as *shoreline-tracking* methods. These methods attempt to track the shoreline explicitly in some way, usually by a heuristic argument. Appropriate boundary conditions, or up-winding for finite difference methods, are then implemented near the shoreline. Some of the early shoreline-tracking methods can be found in [2] and [44]. A method developed by Titov and Synolakis [97], extrapolates the velocity of the fluid at the shoreline to determine the movement and the position of the shoreline at the next time-step. Grid points are added or removed appropriately, so that the computational grid covers the wet domain, with points right at the estimated shoreline. This method produces good comparison with analytical solutions [97], and surprisingly good comparison with actual tsunami data [99, 96, 98]—another testament to the use of the shallow water equations for inundation modeling. Another shoreline tracking method used by Knight, found in [70], extrapolates the surface elevation of the water near the shoreline in the direction perpendicular to linearly sloping bathymetry, as if water existed beyond the shoreline below the land. The location where the water surface intersects the land elevation after the time-step determines the shoreline location for the next time-step.

We will take a different approach, described more fully in the next chapter, by attempting to *capture* the shoreline rather than explicitly track it. This is similar in philosophy to the shock-capturing property of Godunov-type methods, where shocks are captured automatically by solving Riemann problems appropriately near these features.

Chapter 6

APPROXIMATE AUGMENTED RIEMANN SOLVERS FOR THE SHALLOW WATER EQUATIONS AND TSUNAMI MODELING

Specialized Riemann solvers are one of the components that allow the finite volume methods and the wave propagation algorithms to satisfy the requirements demanded by tsunami modeling—described in Chapter 5. For instance, Riemann solvers can be used to incorporate the source term from the shallow water equations directly into the updating waves, allowing accurate near-steady-state calculations required for transoceanic propagation modeling. Additionally, Riemann solvers allow convergence to entropy satisfying discontinuous solutions—the main impetus for which they were originally developed—allowing convergence to solutions in regions that exhibit shocks and bores. Finally, dry regions and shorelines can be handled by developing Riemann solvers that maintain positivity, as well as model the moving wet-dry interface directly—allowing inundation modeling. Typical Riemann solvers, such as those introduced in Chapter 3, often deal with one or several of the above difficulties but often fail to handle all of the above difficulties simultaneously.

In this chapter, a novel Riemann solver will be described that we have developed to deal with the simultaneous difficulties described above. The Riemann solver will be shown to be *well-balanced*, a concept that will be described in Section 6.4. This will allow modeling the global propagation regime accurately, since the property implies that the waves used in defining the numerical update represent deviations from the delicate steady state. The Riemann solver will also be shown to be depth-positive-definite as well as accurate at solving the dry-state Riemann problem, making it appropriate for inundation modeling and shoreline capturing. Further, the solver is well suited for shock-capturing, and in fact will be shown to be equivalent to the Roe solver near shocks. Lastly, the solver does not require an entropy fix such as the Roe solver. It should be emphasized that although this particular solver is novel, recently there has been considerable research directed toward developing solvers with these properties (*e.g.* [26, 5]). However, the form of this solver is particularly well suited for use with the wave propagation algorithms described in Chapter 3.

6.1 Augmented Systems

We can write the shallow water equations as a homogeneous hyperbolic system, by defining the vector $\tilde{q}$ to be the original conserved quantities q augmented by the bathymetry function b,

$$\tilde{q} = \begin{bmatrix} h \\ hu \\ b \end{bmatrix}. \tag{6.1}$$

(In this thesis $\tilde{q}$ will be used to denote various different augmented state variables.) The shallow water equations (in PDE form) are then equivalent to the following larger system,

$$\tilde{q}_t + B(\tilde{q})\tilde{q}_x = 0, \tag{6.2}$$

where

$$B(\tilde{q}) = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & gh \\ 0 & 0 & 0 \end{bmatrix}. \tag{6.3}$$

The system (6.2) is no longer a conservation law since $B(\tilde{q})$ is not a Jacobian, but it is a nonlinear hyperbolic system. The nonlinear matrix $B(\tilde{q})$ has a full set of eigenvectors—two are the same as the eigenvectors for the homogeneous shallow water equations (with a zero third component for b),

$$r^1 = \begin{bmatrix} 1 \\ u - \sqrt{gh} \\ 0 \end{bmatrix}, \quad r^2 = \begin{bmatrix} 1 \\ u + \sqrt{gh} \\ 0 \end{bmatrix}, \tag{6.4}$$

with the associated eigenvalues

$$\lambda^1 = u - \sqrt{gh}, \quad \lambda^2 = u + \sqrt{gh}. \tag{6.5}$$

The third eigenvector of the matrix (6.3)

$$r^3 = \begin{bmatrix} -gh \\ 0 \\ -u^2 + gh \end{bmatrix} \tag{6.6}$$

has a zero eigenvalue $\lambda^3 \equiv 0$. The characteristic field associated with r^3 is therefore linearly degenerate and the characteristics are straight vertical lines in the x–t plane that carry the variation in b. We will refer to this characteristic field as the *steady state field*, since a smooth steady state is equivalent to variation in this field

$$\begin{aligned} &\tilde{q}_t = 0 \\ \Rightarrow &B(\tilde{q})\tilde{q}_x = 0, \end{aligned} \tag{6.7}$$

(we know from Theorem 4.3 in Section 4.4, that smooth variation cannot coexist with an identically zero eigenvalue in one of the other nonlinear characteristic fields). It is interesting to note that the eigenspace of $B(\tilde{q})$ becomes degenerate if one of the eigenvalues λ^1 or λ^2 becomes zero. However, recall from Section 4.4 that this can only occur where $b_x = 0$, and hence it is irrelevant since all variation in the augmented state vector occurs in the phase direction of the standard fields r^1 and r^2 at that point in phase space.

Although the system (6.2) is only valid for smooth solutions, the Riemann problem for the system can be analyzed by including the Rankine-Hugoniot relations for shock solutions from the standard shallow water equations. See, for example, [16] or [26] for a discussion of some properties of systems such as (6.2). Of course, unlike Riemann problems for homogeneous integral conservation laws, (6.2) is not classically defined for problems where part of the solution ($b(x)$) is non-differentiable. For this system, the Riemann problem might better be thought of as an approximation to a slightly smoothed problem. For a more rigorous discussion involving more generalized functions, see [35]. Augmented systems that eliminate the source term by introducing an additional variable and nonconservative products are described more generally in [10]. Some methods for the shallow water equations with bathymetry that include additional solution variables and transformations of the solution variables are described in [26].

An additional augmented system can be derived by first considering a governing equation for the flux of a hyperbolic conservation law. Consider a general hyperbolic conservation law (with source term)

$$q_t + f(q)_x = \psi(q, x). \tag{6.8}$$

Multiplying (6.8) by the Jacobian $f'(q)$ gives

$$f'(q)q_t + f'(q)f(q)_x = f'(q)\psi(q, x). \tag{6.9}$$

By noting that $f'(q)q_t = f(q)_t$, equation (6.9) implies a hyperbolic system for the flux $f(q)$,

$$f_t + f'(q)f_x = f'(q)\psi(q, x), \tag{6.10}$$

where the dependence of f on q has been dropped. For the shallow water equations, where $q^2 = f^1$, the first equation of (6.10) simply reproduces the momentum equation for the shallow water equations. The second equation in (6.10) is a governing equation for $f^2 = \phi = (hu^2 + \frac{1}{2}gh^2)$, which can be written as

$$\phi_t + (-u^2 + gh)(hu)_x + (2u)\phi_x + (2ugh)b_x = 0. \tag{6.11}$$

Of course, the flux is entirely determined by the solution to the shallow water equations, but (6.11) will provide some theoretical insight into how the flux behaves. By defining the additional augmented state variable (first introduced in Section 4.4)

$$\tilde{q} = \begin{bmatrix} h \\ hu \\ \phi \\ b \end{bmatrix}, \tag{6.12}$$

and including equation (6.11), the shallow water equations can be written as an overdetermined, augmented homogeneous system

$$\tilde{q}_t + \tilde{B}(\tilde{q})\tilde{q}_x = 0, \tag{6.13}$$

where

$$\tilde{B}(\tilde{q}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -u^2 + gh & 2u & 0 & gh \\ 0 & -u^2 + gh & 2u & 2ugh \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6.14}$$

The matrix $\tilde{B}(\tilde{q})$ has the eigenvectors

$$r^1 = \begin{bmatrix} 1 \\ u - \sqrt{gh} \\ \left(u - \sqrt{gh}\right)^2 \\ 0 \end{bmatrix}, \quad r^2 = \begin{bmatrix} 1 \\ u + \sqrt{gh} \\ \left(u + \sqrt{gh}\right)^2 \\ 0 \end{bmatrix}, \quad r^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad r^4 = \begin{bmatrix} \frac{-gh}{-u^2+gh} \\ 0 \\ -gh \\ 1 \end{bmatrix}, \tag{6.15}$$

with associated eigenvalues

$$\lambda^1 = u - \sqrt{gh}, \quad \lambda^2 = u + \sqrt{gh}, \quad \lambda^3 = 2u, \quad \lambda^4 = 0. \tag{6.16}$$

Note that r^1 and r^2 correspond to the nonlinear fields associated with the standard shallow water equations. The field associated with r^4 is the linearly degenerate steady state field described in the preceding discussion, yet now the vector also dictates how the flux ϕ should vary through the steady state field. The eigenvector r^3 has a nonzero component only in the "direction" (in phase space) of ϕ. This field is physically irrelevant, since the flux is only a function of h and hu, and cannot vary independently. Since the system (6.13) is valid for smooth solutions, it must be that solutions to the shallow water equations are never simple waves corresponding to this field. This is confirmed if we note that through any simple wave

$$f(q)_x = f'(q)q_x = c(x)\lambda^p r^p, \tag{6.17}$$

for $p = 1, 2$, and some scalar function $c(x)$. Considering the eigenstructure for the shallow water equations, (6.17) implies that through a simple wave of the first or second family,

$$\frac{\partial}{\partial x} \begin{bmatrix} h \\ hu \\ \phi \end{bmatrix} = c(x) \begin{bmatrix} 1 \\ \lambda^p \\ (\lambda^p)^2 \end{bmatrix}, \tag{6.18}$$

where $(\lambda^p)^2$ is the p^{th} eigenvalue squared. Although nonphysical, we might heuristically interpret r^3 as a field that would carry variation in the flux ϕ if it was another variable independent of h and hu.

6.2 An Approximate Riemann Solver Based on an Augmented System

In this section we will introduce an approximate Riemann solver, before describing its features more fully in the following sections while showing its relation to various other Riemann solvers. It is hoped that this solver will be viewed as unifying many of the strengths of several solvers with diverse properties. The solver is introduced first in a simplified basic form, before describing some corrections in the sections that follow that supply some added properties.

Given a Riemann problem for the shallow water equations, we consider the corresponding Riemann problem for the augmented system (6.13),

$$\tilde{q}_t + \tilde{B}(\tilde{q})\tilde{q}_x = 0, \tag{6.19a}$$

with piecewise constant initial data

$$\tilde{q}(x,0) = \begin{cases} \tilde{q}_l & \text{if} \quad x < 0 \\ \tilde{q}_r & \text{if} \quad x > 0 \end{cases}, \tag{6.19b}$$

where, of course,

$$\tilde{q}_l = \begin{bmatrix} h_l \\ (hu)_l \\ \phi_l \\ b_l \end{bmatrix}, \quad \text{and} \quad \tilde{q}_r = \begin{bmatrix} h_r \\ (hu)_r \\ \phi_r \\ b_r \end{bmatrix}. \tag{6.19c}$$

We then approximate the solution to (6.19), by the decomposition

$$\Delta\tilde{q} = \tilde{q}_r - \tilde{q}_l = \sum_{p=1}^{4} \alpha^p \tilde{r}^p, \tag{6.20a}$$

where $\tilde{r}^p$ are local approximations to the eigenvectors of $\tilde{B}$,

$$\tilde{r}^1 = \begin{bmatrix} 1 \\ s^1_{\mathcal{E}} \\ (s^1_{\mathcal{E}})^2 \\ 0 \end{bmatrix}, \quad \tilde{r}^2 = \begin{bmatrix} 1 \\ s^2_{\mathcal{E}} \\ (s^2_{\mathcal{E}})^2 \\ 0 \end{bmatrix}, \quad \tilde{r}^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \tilde{r}^4 = \begin{bmatrix} \frac{\mathcal{S}^+_-(q_l,q_r)}{-\widetilde{\lambda^1\lambda^2}(q_l,q_r)} \\ 0 \\ \mathcal{S}^+_-(q_l,q_r) \\ 1 \end{bmatrix}. \tag{6.20b}$$

The values $s^{1,2}_{\mathcal{E}}$ are the Einfeldt speeds defined in Section 3.3, and $\mathcal{S}^+_-(q_l,q_r)$ and $\widetilde{\lambda^1\lambda^2}(q_l,q_r)$ are the averages defined in Section 4.4. We then propagate the waves defined by the decomposition at speeds

$$s^1 = s^1_{\mathcal{E}}, \quad s^2 = s^2_{\mathcal{E}}, \quad s^3 = \frac{1}{2}(s^1_{\mathcal{E}} + s^2_{\mathcal{E}}), \quad s^4 = 0. \tag{6.20c}$$

The third eigenpair, r^3 and s^3, are only one of several options presented here for completeness. Properties of this eigenpair and other options will be discussed in Section 6.7 when we discuss entropy fixes. We will refer to (6.20) as an *augmented approximate Riemann solver.*

Note that we only need two components of the waves defined by (6.20) in order to update the solution. In order to maintain conservation, we will use the waves associated with the flux components to update the solution directly, based on the f-wave method described in Chapter 3. This will be described more in Section 6.4.

A Riemann solver that decomposes an augmented variable, such as that defined by (6.20), can be related to simple relaxation schemes such as those described in [49]. This connection is described by LeVeque and Pelanti [69] and also in the text [10]. These connections are not further explored here.

6.3 Shock Preservation and Relation to the Roe Solver

Solutions to the shallow water equations, and hyperbolic systems in general, typically consist of regions that are smoothly varying punctuated by abrupt jump discontinuities corresponding to shocks. Numerically then, Riemann problems that are typically encountered are of two types: those that correspond to a discretization of smoothly varying data, and those with data corresponding to a shock discontinuity. The latter case was one of the original motivations for solving Riemann problems to begin with. For this latter case, where the true solution is a jump discontinuity, a Riemann solver that approximates the true solution with jump discontinuities should at least produce the true shock. This was one of the strengths of the Roe solver and the Roe averages introduced in Chapter 3, and we would like to ensure that our Riemann solver maintains this property. That is, given initial data q_l and q_r that lie on a Hugoniot locus, our Riemann solver should produce the exact solution. Recall that the Roe solver accomplished this by ensuring conservation, which must correspond to the Rankine-Hugoniot conditions in the case of a single discontinuity. That is, the set of relations

$$s(q_r - q_l) = f(q_r) - f(q_r) = \hat{A}(q_r - q_l), \tag{6.21}$$

imply that $(q_r - q_l)$ is an eigenvector of $\hat{A}$ with eigenvalue s. The Roe decomposition therefore reproduces the exact solution. Considering the shallow water equations, (6.21) implies that the following relationship is satisfied by the exact solution

$$\begin{bmatrix} h_r \\ (hu)_r \\ \phi(q_r) \end{bmatrix} - \begin{bmatrix} h_l \\ (hu)_l \\ \phi(q_l) \end{bmatrix} = (h_r - h_l) \begin{bmatrix} 1 \\ \hat{\lambda}^p \\ (\hat{\lambda}^p)^2 \end{bmatrix} = (h_r - h_l) \begin{bmatrix} 1 \\ s^p \\ (s^p)^2 \end{bmatrix}. \tag{6.22}$$

Also, the fact that the Roe eigenvalue $\hat{\lambda}^p$ corresponds to the true shock speed implies that the Einfeldt speed $s^p_{\mathcal{E}}$ will be the Roe speed $\hat{\lambda}^p$. This follows from considering the characteristic structure of a shock wave in either family, and recalling that entropy satisfying shocks must have the characteristics impinging on the shock wave. Therefore, in the case where the Riemann data lies on a Hugoniot locus and the true solution corresponds to a single shock, the decomposition (6.20) will produce the exact solution since

$$\begin{bmatrix} h_r \\ (hu)_r \\ \phi(q_r) \\ b_r \end{bmatrix} - \begin{bmatrix} h_l \\ (hu)_l \\ \phi(q_l) \\ b_l \end{bmatrix} = (h_r - h_l) \begin{bmatrix} 1 \\ s^p_{\mathcal{E}} \\ (s^p_{\mathcal{E}})^2 \\ 0 \end{bmatrix}. \tag{6.23}$$

More typically near a shock, the Riemann data will have a large jump that corresponds to the shock wave, and very small deviations that correspond to the small transitions in the other fields. For instance, a small variation in bathymetry from adjacent grid cells will produce weak waves for r^k, $k \neq p$. However, the large jump that corresponds to the shock is propagated correctly.

If we ignore variable bathymetry, note that if both Einfeldt speeds $s^{1,2}_{\mathcal{E}}$ correspond to the Roe speeds $\hat{\lambda}^{1,2}$, the the approximate solver (6.20) corresponds to the Roe solver, regardless of the form of r^3. This is because the jump in $\tilde{q}$ lies in the two dimensional space spanned by

$$\begin{bmatrix} 1 \\ \hat{\lambda}^1 \\ (\hat{\lambda}^1)^2 \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 1 \\ \hat{\lambda}^2 \\ (\hat{\lambda}^2)^2 \\ 0 \end{bmatrix}, \tag{6.24}$$

which follows from the conservative property of the Roe solver:

$$f(q_r) - f(q_l) = \hat{A}(q_r - q_l) = \hat{A}\sum_{p=1}^{2} \alpha^p \hat{r}^p = \sum_{p=1}^{2} \alpha^p \hat{\lambda}^p \hat{r}^p. \tag{6.25}$$

Considering the shallow water equations, (6.25) implies that

$$\begin{bmatrix} h_r \\ (hu)_r \\ \phi(q_r) \end{bmatrix} - \begin{bmatrix} h_l \\ (hu)_l \\ \phi(q_l) \end{bmatrix} = \sum_{p=1}^{2} \alpha^p \begin{bmatrix} 1 \\ \hat{\lambda}^p \\ (\hat{\lambda}^p)^2 \end{bmatrix}. \tag{6.26}$$

Therefore, the decomposition (6.20) would result in $\alpha^3 = 0$. For typical Riemann problems that correspond to discretized smooth data, or perhaps shocks, we expect $|\alpha^3| << |\alpha^p|$, $p \neq 3$, since, if one of the Einfeldt speeds is not the Roe speed the two will typically be very close. An exception occurs in the case of a strong rarefaction, where either $u_l - \sqrt{gh_l} << \hat{\lambda}^1$ or $u_r + \sqrt{gh_r} >> \hat{\lambda}^2$. This will be explored more fully when we discuss entropy fixes and strong rarefactions in Section 6.7.

6.4 Source Terms and Well-Balancing

In Section 5.3 it was shown that accuracy for tsunami modeling will require our numerical method to perfectly preserve certain steady states, and more importantly, resolve tiny deviations to the steady state. The steady state arises from a nontrivial balance of the source term and the flux gradient, as described in Chapter 4 and Chapter 5. At first glance it might seem that the Riemann solver defined by (6.20) does not include the source term, since it does not appear in the variables decomposed into the propagating waves. However, recall that our numerical update comes from the re-averaging effect of waves moving into the grid cells and second order corrections based on those waves. The wave associated with the linearly degenerate steady state field, $\tilde{r}^4$, is a contact discontinuity that remains at $x = 0$ producing the contribution of the source term.

6.4.1 Relation to the f-wave Method

In Section 3.3 the f-wave method was introduced as an approach for dealing with problems where the solution is discontinuous at the cell interfaces, either due to a variable coefficient flux function, or in the case described in Section 3.6, due to a point source terms concentrated at the cell interfaces. In this latter case, recall that the point source term is accounted for by subtracting it from the flux function before performing the decomposition

$$f(q_r) - f(q_l) - \Psi\Delta x = \sum_{p=1}^{M_w} \beta^p r^p, \tag{6.27}$$

where $\Psi\Delta x$ is some approximation of the source term, $\int_{x_l}^{x_r} \psi \, dx$, and r^p, $p = 1, \cdots, M_w$, are some set of suitable approximations to the eigenvectors of the Jacobian. The resulting waves $\mathcal{Z}^p = \beta^p r^p$ moving into the surrounding grid cells then update the solution directly. For the shallow water equations, (6.27) takes the form

$$\begin{bmatrix} (hu)_r - (hu)_l \\ \phi_r - \phi_l - \overline{\int_{x_l}^{x_r} \psi \, dx} \end{bmatrix} = \sum_{p=1}^{2} \beta^p \begin{bmatrix} 1 \\ \lambda^p \end{bmatrix}, \tag{6.28}$$

where $\overline{\int_{x_l}^{x_r} \psi \, dx}$ is some approximation to $-\int_{x_l}^{x_r} ghb_x \, dx$, based on q_l,b_l,b_r and q_r, and λ^p are approximations to the eigenvalues of the Jacobian. If we now consider the augmented decomposition (6.20), it must be that $\alpha^4 = (b_r - b_l)$, since this is the only eigenvector with a nonzero fourth component. Therefore, ultimately the decomposition is equivalent to that obtained by first subtracting $(b_r - b_l)\tilde{r}^4$ from $\tilde{q}$, followed by a decomposition of the first three components of $\tilde{q}$ onto the first three components of the remaining eigenvectors. If we consider just the flux components of such a decomposition, which are what get used for the numerical update, we have

$$\begin{bmatrix} (hu)_r - (hu)_l \\ \phi_r - \phi_l - \mathcal{S}_-^+(q_l, q_r)(b_r - b_l) \end{bmatrix} = \sum_{p=1}^{2} \alpha^p s_{\mathcal{E}}^p \begin{bmatrix} 1 \\ s_{\mathcal{E}}^p \end{bmatrix} + \beta^3 \tilde{r}^3. \tag{6.29}$$

Since $\mathcal{S}_-^+(q_l, q_r)(b_r - b_l)$ is an approximation to $\int_{x_l}^{x_r} -ghb_x \, dx$, as shown in Section 4.4, equation (6.29) elucidates the relation between the standard f-wave approach, equation (6.28), and the augmented decomposition (6.20). Note that the form of $\tilde{r}^4$ and the first component of $\tilde{q}$—the depth h—ultimately determine the difference since they affect the decomposition of the flux into the three vectors in (6.29) rather than the two in (6.28).

6.4.2 Well-Balancing and Preservation of Steady States

A *well-balanced* method generally is one that perfectly preserves certain discrete steady state data. That is, if we take a smooth steady state solution to a hyperbolic conservation law with a source term, and use some relevant spatial discretization of that solution for an initial condition, a well-balanced method will have an identically zero numerical update. Often a method might be referred to as well-balanced even if it preserves only certain important steady states. For instance, various well-balanced schemes have been developed for the "lake at rest" Riemann problem for the shallow water equations, where $u_l = u_r = 0$, and $h_l + b_l = h_r + b_r$ (e.g. [5, 12, 27, 77]). For a discussion of well-balanced schemes in the more general context of hyperbolic systems with source terms see the work of Greenberg, LeRoux *et al* [38, 39], or the recent text [10] by Bouchut.

The augmented Riemann solver (6.20) is well-balanced with respect to the lake at rest problem, as well as the steady states with nonzero velocities described in Section 4.4. Note that if Riemann data comes from evaluating any smooth steady state at two points, x_l and x_r, surrounding monotonically varying bathymetry, the Riemann solver (6.20) will produce no updating waves. This follows immediately from Theorem 4.3, and the form of the *steady state vector*, $\tilde{r}^4$, which must span the initial jump $(\tilde{q}_r - \tilde{q}_l)$. Such steady state Riemann data is preserved as a contact discontinuity at $x = 0$, which has no effect on the solution. Figure 6.1 depicts the depths and bathymetry of a Riemann solution given steady state data for a motionless steady state as well as a steady flow where $hu_r = hu_l \neq 0$. This property of the solver (6.20) when given Riemann data that approximates a smooth steady state might be thought of as analogous to the Roe solver in the event of a single shock. Note however, in the former case the Riemann solution is numerically the correct approximation to a relevant smooth steady state, whereas in the latter case the approximate Riemann solution is exactly the true solution to the Riemann problem.

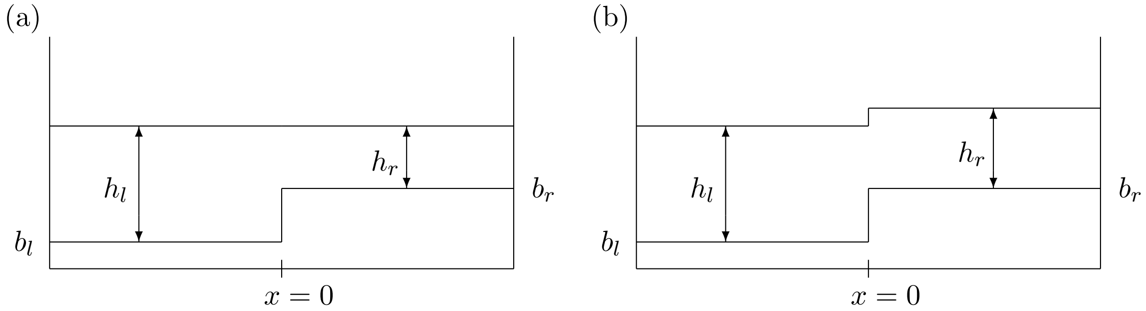


Figure 6.1: Depths of steady state Riemann solution, for the motionless steady state and steady flow with velocity. (a) The motionless steady state problem, $u_l = u_r = 0$. The jump in the depth $h_r - h_l$ satisfies $h_r + b_r = h_l + b_l$. This jump is maintained as a contact discontinuity in the approximate augmented Riemann solver (6.20), due to the form of $\tilde{r}^4$. (b) A steady state problem where $hu_l = hu_r \neq 0$. The steady state is maintained again as a contact discontinuity, due to the form of $\tilde{r}^4$. However, in this case, the nonzero velocities dictate that $h_r + b_r \neq h_l + b_l$.

For smooth steady states surrounding regions of monotonically varying bathymetry, we know that the averages $\mathcal{S}_-^+(q_l, q_r)$ and $\frac{\mathcal{S}_-^+(q_l,q_r)}{-\widetilde{\lambda^1\lambda^2}(q_l,q_r)}$ are well defined because of the results of Section 4.4. (In fact, the quantities are well defined even when b has an extremum whenever $b_r - b_l \neq 0$, yet $\mathcal{S}_-^+$ might be zero if there is a sonic point where $b_x = 0$.) We also know the bound

$$-g\max(h_l, h_r) \leq \mathcal{S}_-^+(q_l, q_r) \leq -g\min(h_l, h_r), \tag{6.30}$$

for such problems. Of course, the initial Riemann data, $\tilde{q}_l$ and $\tilde{q}_r$ might not be steady state data, in which case we must ensure that $\tilde{r}^4$ is defined appropriately, and produces a consistent Riemann solver. This can be accomplished by enforcing the bound (6.30) on the third component of the steady state vector, $(\tilde{r}^4)^3$, which is only guaranteed for the smooth steady state Riemann data. Considering (6.28) and (6.29), this is equivalent to requiring

$$g\min(h_l, h_r)|b_r - b_l| \leq |\Psi|\Delta x \leq g\max(h_l, h_r)|b_r - b_l| \tag{6.31}$$

in the f-wave method, where typically $\Psi\Delta x = -g\bar{h}(b_r - b_l)$, where $\bar{h}$ is some consistent average of h_l and h_r. When viewed in this light, enforcing the bounds (6.30) is simply equivalent to ensuring that $(\tilde{r}^4)^3$ is a consistent approximation to $-\int_{x_l}^{x_r} ghb_x\,dx$ based on h_l and h_r, even when the Riemann data is far from a steady state.

We also must ensure that the first component of the steady state vector, $(\tilde{r}^4)^1$, which corresponds to the jump in depth across the contact discontinuity at $x = 0$, is bounded appropriately when the initial Riemann data is far from a steady state. This will be taken up in Section 6.5, where bounds on this term will be introduced to ensure positivity of the Riemann solution. Recall that for the steady state problems discussed in Section 4.4, this term must be negative for problems that are entirely subsonic, and positive for problems that are entirely supersonic.

Typically $(\tilde{r}^4)^1$ and $(\tilde{r}^4)^3$ are well defined simply by the averages $\mathcal{S}^{+}_{-}(q_l, q_r)$ and $\widetilde{\lambda^1\lambda^2}(q_l, q_r)$, and bounds on these terms will not change there value. An exception occurs for Riemann problems that have a sonic point, either as a transonic rarefaction or a stationary shock. The simple pratical fix for this type of problem will be taken up Section 6.6.

6.5 Preserving Positivity and Relation to the HLLE Solver

6.5.1 Relation to the HLLE solver

In Section 3.3 the HLL type of approximate solvers were introduced. Recall that these solvers approximate the Riemann solution with two simple discontinuities propagating at predetermined speeds. Conservation then determines the middle state q^*. Recall also that the HLLE solver was a type of HLL solver that arises when the Einfeldt speeds $s_{\mathcal{E}}^{1,2}$ are used for the propagation speeds. This solver had several nice properties such as providing a simple entropy fix as well as preserving positivity of the middle state depth—the *Einfeldt depth* $h^*_{\mathcal{E}}$. In this section we will show that the augmented solver (6.20) is equivalent to the HLLE solver for the mass update when the source term is neglected, implying that, over constant bathymetry, the solver is depth positive definite in the sense of Definition 3.2. For variable bathymetry, we will introduce some bounds on the first component of the steady state vector, $\tilde{r}^4$, that will maintain this property in the presence of a source term.

Consider a simple Riemann solver for the homogeneous shallow water equations, where the initial jump in the conserved quantities is decomposed into the two vectors

$$\begin{bmatrix} h_r \\ (hu)_r \end{bmatrix} - \begin{bmatrix} h_l \\ (hu)_l \end{bmatrix} = \sum_{p=1}^{2} \alpha^p \begin{bmatrix} 1 \\ s_{\mathcal{E}}^p \end{bmatrix}, \tag{6.32}$$

and the resulting waves $\mathcal{W}^p = \alpha^p r^p$ are propagated at the speeds $s_{\mathcal{E}}^p$. This Riemann solver is not necessarily conservative, except for the case that $s_{\mathcal{E}}^{1,2} = \hat{\lambda}^{1,2}$, resulting in the conservative

Roe solver. However, considering the re-averaging effect of the first component of the propagating waves $\mathcal{W}^{1,2}$, conservation of mass requires

$$\alpha^1 s^1_{\mathcal{E}} + \alpha^2 s^2_{\mathcal{E}} = (hu)_r - (hu)_l. \tag{6.33}$$

Since this condition is exactly the second equation of (6.32), the method must conserve mass. Note also that since the second component of each vector r^p corresponds to the speed of the wave, using the jump in depth multiplied by speed of the moving discontinuity to define the update to the mass equation—as in the standard wave propagation algorithm—must be the same as using the jumps in the momentum directly to update the mass—as in the f-wave approach. Therefore, the two effective middle state depths (Definition 3.1) implied by the second equation in (6.32) are equivalent to the actual single middle state depth implied by the first equation. Stated more broadly, a consistent use of the first or second equation in (6.32) to update the mass equation produces the same numerical result.

Since the HLLE method conserves mass with only two jump discontinuities propagating at the same speeds as (6.32), the HLLE method and (6.32) must be identical for the mass component. (This can be further verified—the mass decomposition in (6.32) implies a middle state depth h^*, where, $\alpha^1 = h_r - h^*$ and $\alpha^2 = h^* - h_l$. Therefore, (6.33) is equivalent to (3.26)—the equation that defines $h^*_{\mathcal{E}}$ from Theorem 3.1.) Since we saw that the HLLE method preserves positivity of depth from Theorem 3.1, the method implied by (6.32) must also preserve positivity in the sense of Definition 3.1.

6.5.2 Preserving Positivity with the Augmented Decomposition

For the augmented decomposition (6.20), the initial jump $\tilde{q}_r - \tilde{q}_l$ is ultimately decomposed into four vectors, representing four propagating discontinuities or waves. Writing the decomposition in (6.20) as

$$\begin{bmatrix} h_r \\ (hu)_r \\ * \\ * \end{bmatrix} - \begin{bmatrix} h_l \\ (hu)_l \\ * \\ * \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & \frac{\mathcal{S}^{+}_{-}(q_l,q_r)}{-\widetilde{\lambda^1\lambda^2}(q_l,q_r)} \\ s^1_{\mathcal{E}} & s^2_{\mathcal{E}} & 0 & 0 \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \alpha^3 \\ \alpha^4 \end{bmatrix}, \tag{6.34}$$

allows us to concentrate on the depth and momentum components of the decomposition. For the case of constant bathymetry, where $b_r - b_l = \alpha^4 = 0$, it is clear that (6.34) is identical to the decomposition (6.32) for depth and momentum. Like the HLLE method then, when there is constant bathymetry our method (6.20) must preserve positivity of depth in the sense of Definition 3.2. Of course, once variable bathymetry is introduced to the Riemann problem, there exists a jump in depth at $x = 0$ due to a nonzero α^4, thereby changing the terms α^1 and α^2, and creating two middle state depths—h^*_l and h^*_r. See Figure 6.2 depicting such an approximate Riemann solution. Recall from Section 3.3, where it was shown that the existence of two effective middle state depths, h^*_l and h^*_r, prevented maintaining effective positivity when using the f-wave approach, where the momentum decomposition alone updated the mass equation. However, decomposing both the depth and momentum in the augmented decomposition (6.20), will allow us to control the effective middle state depths that come from the momentum decomposition, by controlling the actual middle state depths h_l^* and h_r^*. That is, note that the jumps in the second component of the decomposition (6.34) are precisely the jumps in the first component multiplied by the speeds of the propagating waves. Therefore, the effective middle state depths defined by the second component of (6.34) are exactly the same as the actual depths defined by the first component. Therefore, if we can ensure that the actual middle state depths remain non-negative, then (6.20) will be depth positive definite in the sense of Definition 3.2.

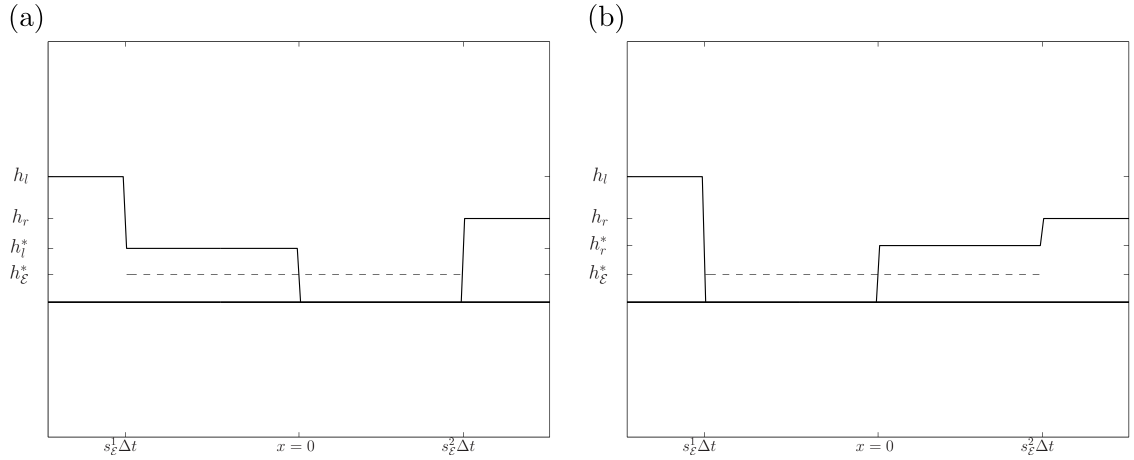

(a)
h_l
h_r
h_l^*
$h_\mathcal{E}^*$
$s_\mathcal{E}^1 \Delta t$
$x = 0$
$s_\mathcal{E}^2 \Delta t$
(b)
h_l
h_r
h_r^*
$h_\mathcal{E}^*$
$s_\mathcal{E}^1 \Delta t$
$x = 0$
$s_\mathcal{E}^2 \Delta t$

Figure 6.2: Preserving the depth positive definite property of the augmented solver when variable bathymetry exists. The depth of an approximate Riemann solution to a subsonic problem with two rarefactions is shown at some time $t > 0$. The single middle depth $h_\mathcal{E}^*$ that would arise if there was no source term is shown as a dashed line. If a momentum source term from variable bathymetry exists, it implies that there is a jump in the depth at the interface $x = 0$. Since there is no source of mass, the amount of mass between the two propagating discontinuities is fixed, regardless of the jump in depth at $x = 0$. Since this total amount of mass is equivalent to that from the HLLE solver, positivity of the actual and effective depths—h_l^* and h_r^*—are maintained as long as the jump at the interface is bounded. (a) The maximum jump $(h_r^* - h_l^*)$ that allows conservation together with $h_r^* \geq 0$. For subsonic flow, this situation implies $b_r - b_l > 0$. (b) A similar case when $b_r - b_l < 0$.

Since the second component of (6.34) ensures conservation of mass, we can relate h_l^* and h_r^* to the single conservative middle state depth that would arise if there was no jump in depth at $x = 0$. We will denote this depth, $h_\mathcal{E}^*$, since it corresponds to the HLLE depth. For a subsonic problem, where $s_\mathcal{E}^1 < 0 < s_\mathcal{E}^2$, conservation of mass requires

$$s_\mathcal{E}^2 \left(h_r - h_\mathcal{E}^*\right) + s_\mathcal{E}^1 \left(h_\mathcal{E}^* - h_l\right) = s_\mathcal{E}^2 \left(h_r - h_r^*\right) + s_\mathcal{E}^1 \left(h_l^* - h_l\right), \tag{6.35}$$

implying intuitively that as we increase either h_l^* or h_r^* the other must decrease. If we increase h_r^* until $h_l^* = 0$, (6.35) implies that

$$h_r^* = \frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^2} h_{\mathcal{E}}^*. \tag{6.36}$$

Similarly, if $h_r^* = 0$, we have

$$h_l^* = \frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{-s_{\mathcal{E}}^1} h_{\mathcal{E}}^*. \tag{6.37}$$

If we prevent h_l^* or h_r^* from becoming negative, then (6.36) and (6.37) imply a bound on $h_r^* - h_l^*$,

$$\frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^1} h_{\mathcal{E}}^* \leq h_r^* - h_l^* \leq \frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^2} h_{\mathcal{E}}^*. \tag{6.38}$$

Since, the method (6.20) implies

$$h_r^* - h_l^* = (b_r - b_l)(\tilde{r}^4)^1, \tag{6.39}$$

(6.38) would be satisfied if

$$\frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^1} h_{\mathcal{E}}^* \leq (b_r - b_l)(\tilde{r^4})^1 \leq \frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^2} h_{\mathcal{E}}^*. \tag{6.40}$$

Note that the first term in (6.40) is negative and the last is positive for subsonic flow. Since $(\tilde{r}^4)^1$ is an approximation to $\big(-gh/(-\lambda^1\lambda^2)\big)$, we expect this term to be negative for subsonic flow and positive for supersonic flow. For steady state problems, we know that $(\tilde{r}^4)^1$ will be appropriately defined by

$$(\tilde{r^4})^1 = \frac{\mathcal{S}_-^+(q_l, q_r)}{\widetilde{-\lambda^1\lambda^2}(q_l, q_r)}. \tag{6.41}$$

However, for subsonic problems that are far from a steady state, we may need to bound $(\tilde{r}^4)^1$ by the conditions,

$$\frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^1} \frac{h_{\mathcal{E}}^*}{(b_r - b_l)} \leq \left(\tilde{r}^4\right)^1 \leq -1 \quad \text{if} \quad b_r - b_l > 0, \tag{6.42a}$$

$$\frac{(s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1)}{s_{\mathcal{E}}^2} \frac{h_{\mathcal{E}}^*}{(b_r - b_l)} \leq \left(\tilde{r}^4\right)^1 \leq -1 \quad \text{if} \quad b_r - b_l < 0, \tag{6.42b}$$

These bounds will ensure that the middle state depths are positive in the case of subsonic flow.

For supersonic flow, we expect $(\tilde{r}^4)^1$ to change sign, however, bounding this term is not necessary for guaranteeing positivity since all of the waves move in one direction. This is because conservation of mass ensures that multiple middle state depths arising from the multiple waves integrate to the single nonnegative middle state depth, $h_{\mathcal{E}}^*$. Since all of the waves move in the same direction, the numerical effect of a negative middle state depth is balanced by a positive one. In fact, the term $(\tilde{r}^4)^1$ has no effect on the numerical solution at all for supersonic flow, at least when considering the first order update. (Since limiters applied to second-order corrections are based on comparing individual wave families from adjacent cells, bounding the term $(\tilde{r}^4)^1$ for supersonic flow may have some unforeseen benefit, since it would change the weights in the decomposition (6.20) into different wave families. This has not yet been explored, however, the bounds that prevent negative middle state depths are easily determined.) To maintain positivity of all of actual and effective middle state depths, using arguments similar to those above for subsonic problems, the following bounds suffice:

$$0 \leq \left(\tilde{r}^4\right)^1 \leq \frac{s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1}{s_{\mathcal{E}}^1} \frac{h_{\mathcal{E}}^*}{(b_r - b_l)} \quad \text{if} \quad b_r - b_l > 0, \tag{6.43a}$$

$$0 \leq \left(\tilde{r}^4\right)^1 \leq \frac{-h_l}{(b_r - b_l)} \quad \text{if} \quad b_r - b_l < 0, \tag{6.43b}$$

when $0 < s_{\mathcal{E}}^1 < s_{\mathcal{E}}^2$, and

$$0 \leq \left(\tilde{r}^4\right)^1 \leq \frac{h_r}{(b_r - b_l)} \quad \text{if} \quad b_r - b_l > 0, \tag{6.44a}$$

$$0 \leq \left(\tilde{r}^4\right)^1 \leq \frac{s_{\mathcal{E}}^2 - s_{\mathcal{E}}^1}{s_{\mathcal{E}}^2} \frac{h_{\mathcal{E}}^*}{(b_r - b_l)} \quad \text{if} \quad b_r - b_l < 0, \tag{6.44b}$$

when $s_{\mathcal{E}}^1 < s_{\mathcal{E}}^2 < 0$.

6.6 The Resonant Problem and Near-Sonic Problems

In defining the steady state vector $\tilde{r}^4$ for the augmented solver, it has been assumed that all of the averages used for the components are well defined. For problems that are subsonic, $\overline{\lambda^1\lambda^2}(q_l, q_r)$ and $\widetilde{\lambda^1\lambda^2}(q_l, q_r)$ are negative, and $\mathcal{S}_-^+(q_l, q_r)$ is a consistent approximation for $-gh$. The same is true for supersonic problems, accept that $\overline{\lambda^1\lambda^2}(q_l, q_r)$ and $\widetilde{\lambda^1\lambda^2}(q_l, q_r)$ are each positive. This reflects the fact that for steady subsonic problems, h_x and b_x have opposite signs and for steady supersonic problems they have the same sign. However, it is possible to have Riemann problems where either $\overline{\lambda^1\lambda^2}(q_l, q_r)$ and $\widetilde{\lambda^1\lambda^2}(q_l, q_r)$ are opposite in sign or perhaps one is zero. This problem can be better understood by considering the augmented system discussed in Section 6.1. Note that if one of the eigenvalues $\lambda^1(q)$ or $\lambda^2(q)$ passes through zero, the corresponding eigenvector $r^1(q)$ or $r^2(q)$ (in (6.4)) is identical to the steady state eigenvector $r^3(q)$ (in (6.6)), and a full set of linearly independent eigenvectors no longer exists. Therefore, at such a point, there is no eigenvector with a component allowing variation in the additional variable b. This is a more general problem for augmented systems where a nonhomogeneous system is converted to a homogeneous one with nonconservative products, such as (6.2). This is known as the *resonant problem*, and it is discussed in more detail in [16]. In Section 6.1, it was pointed out that this is not relevant for exact physical solutions, since a sonic point cannot occur over variable bathymetry, and must occur at an extremum in b. However, it is possible to encounter Riemann problems with data that results in a critical point in the homogeneous solution, together with a jump in $(b_r - b_l)$. For instance, if a steady state with a critical point at a local extremum in b is discretized, such a Riemann problem might result. For such a problem, recall from Section 4.4, that b_x and h_x are of the same sign to one side of the critical point and opposite sign to the other. The same relationship exists for a stationary shock over smoothly varying bathymetry. For the resulting Riemann problems, the relationship between $(h_r - h_l)$ and $(b_r - b_l)$, and hence the sign of $(\tilde{r}^4)^1$, is unclear.

We can deal with problems where the standard definitions used in $\tilde{r}^4$ are inconsistent, by simply checking if the problem has data that reflects a possible sonic point. In such a case, $(\tilde{r}^4)^1$ is set to zero, and $\mathcal{S}_-^+(q_r, q_l)$ is defined as $-g\bar{h}$. A simple sonic check can be performed by simply looking at the quantities, $\lambda^1(q_l)\lambda^1(q_r)$, $\lambda^2(q_l)\lambda^2(q_r)$ and $\widetilde{\lambda^1\lambda^2}(q_l, q_r)\overline{\lambda^1\lambda^2}(q_l, q_r)$. If any of these quantities is less than or equal to zero, the Riemann data contains a sonic point. In practice it makes sense to compare the sign of $\overline{\lambda^1\lambda^2}$ and $\widetilde{\lambda^1\lambda^2}$ with the sign of $s_{\mathcal{E}}^1 s_{\mathcal{E}}^2$, so that for subsonic or supersonic problems, all of the definitions used in the eigenvectors are consistent with the direction that the waves propagate.

Perhaps fortuitously, the two dimensional space spanned by $\tilde{r}^4$ and $\tilde{r}^k$ does not depend on $(\tilde{r}^4)^1$, when $s_{\mathcal{E}}^k = 0$. Therefore, setting $(\tilde{r}^4)^1 = 0$ has no effect on the approximate solution for problems where $s_{\mathcal{E}}^k$ is zero—a portion of the jump $(\tilde{q}_r - \tilde{q}_l)$ is projected onto this two-dimensional space, and these two waves remain at the interface. Considering (4.94), the space spanned by $\tilde{r}^4$ and $\tilde{r}^k$ approximates the true jump in the steady state solution when there is a stationary shock in the k^{th} family over variable bathymetry.

6.7 Rarefactions and Entropy Fixes

In this section we will discuss alternatives for the approximate eigenvector $\tilde{r}^3$ used in the Riemann solver (6.20). Although the Einfeldt speeds provide an entropy fix for the pathological expansion shock, which can result from trancritical rarefactions, these large speeds can produce approximate solutions with poor accuracy for these and similar types of Riemann problems. It will be shown that the vector $\tilde{r}^3$ can be used to produce better numerical approximation for Riemann problems with strong, possibly transcritical rarefactions. In this section we will consider Riemann solutions to the homogeneous shallow water equations with no variable bathymetry, since Riemann problems with large transitions in the two nonlinear fields are our main concern here.

6.7.1 The Commonly Encountered Riemann Problem and Weak Transitions

When the augmented decomposition (6.20) was introduced in Section 6.2, it included

$$\tilde{r}^3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T, \tag{6.45}$$

as an additional eigenvector in the decomposition of $(\tilde{q}_r - \tilde{q}_l)$. This is in fact an eigenvector of the augmented system (6.3), but as pointed out in Section 6.1, this eigenvector is nonphysical since the flux ϕ is a function only of h and hu. For smooth solutions, $f(q)_x = f'(q)q_x$, therefore, at any point the vector

$$\begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_x, \tag{6.46}$$

must lie in the two dimensional space spanned by

$$\left\{ \begin{bmatrix} 1 \\ \lambda^1(q) \\ (\lambda^1(q))^2 \end{bmatrix}, \begin{bmatrix} 1 \\ \lambda^2(q) \\ (\lambda^2(q))^2 \end{bmatrix} \right\}. \tag{6.47}$$

As described in Section 6.3, solutions to hyperbolic conservation laws typically have regions with smooth variation in the solution, punctuated by abrupt shock wave discontinuities. If we consider a discretized smooth solution, evaluated at two points x_l and x_r with $x_r - x_l = \Delta x$, using Taylor series it is easy to show that

$$\begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_r - \begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_l = \sum_1^2 \alpha^p \begin{bmatrix} 1 \\ \bar{\lambda}^p \\ (\bar{\lambda}^p)^2 \end{bmatrix} + \mathcal{O}(\Delta x^2), \tag{6.48}$$

where $\bar{\lambda}^p$ is any consistent approximation to $\lambda^p(q)$ based on q_l and q_r. For typical Riemann problems in smooth varying regions, $s_{\mathcal{E}}^p \approx \lambda^p(q_l) \approx \lambda^p(q_r)$, and for the decomposition

$$\begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_r - \begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_l = \begin{bmatrix} 1 & 0 & 1 \\ s_{\mathcal{E}}^1 & s_{\mathcal{E}}^2 & 0 \\ (s_{\mathcal{E}}^1)^2 & (s_{\mathcal{E}}^2)^2 & 1 \end{bmatrix} \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{bmatrix}, \tag{6.49}$$

we expect α^3 to be much smaller in magnitude than α^1 and α^2. This type of Riemann problem has weak or small transitions in the characteristic families, whether they be small shocks or slight rarefactions. This is depicted in Figure 6.3 (a). However, in the presence of a physical shock wave in the solution, the Riemann problem might contain much larger jumps $(\tilde{q}_r - \tilde{q}_l)$, and because of nonsmoothness, (6.48) is no longer valid. However, as described in Section 6.3, in the presence of a shock wave, $\alpha^3 \approx 0$ in the decomposition (6.49). In fact, because of the conservative properties of the Roe solver, if $s_{\mathcal{E}}^{1,2} = \hat{\lambda}^{1,2}$, then $\alpha^3 = 0$ in (6.49).

For typical Riemann problems (6.45) is therefore an appropriate choice for $\tilde{r}^3$, due to several reasons. First, it allows maintaining the positive preserving properties described previously. Second, as described in Section 6.5, by setting the first two components of $\tilde{r}^3$ to zero, the propagating discontinuities implied by (6.20) are conservative with respect to mass, even when the Roe speeds are not used. Third, we need a complete set of linearly independent eigenvectors for the approximate Riemann solver, and preferably a well-conditioned $\tilde{R}$.

6.7.2 Strong Rarefactions

Although much less common for typical grid spacings, we might encounter an exceptional type of Riemann problem that contains a large rarefaction in one or both of the characteristic families. For instance, if we have a strong rarefaction in the first characteristic family, as

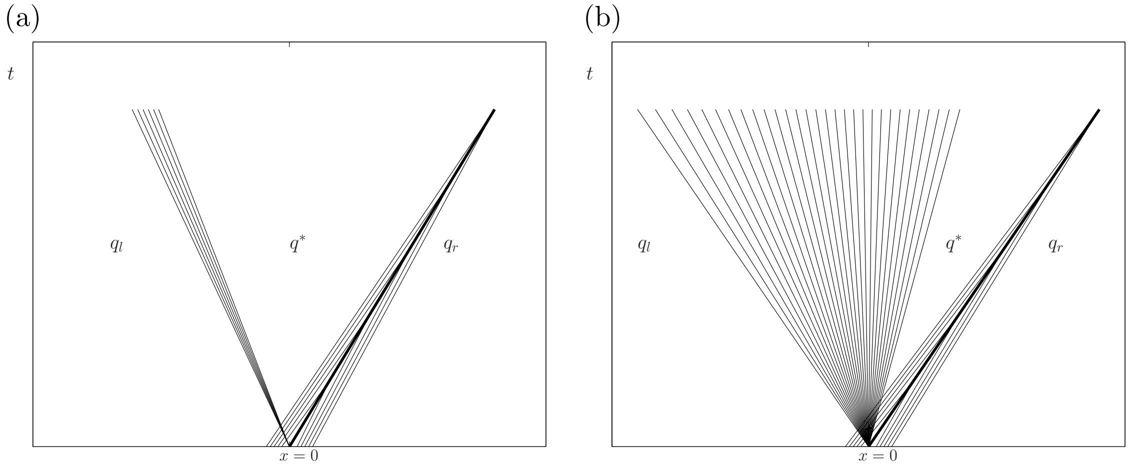


Figure 6.3: A common Riemann problem compared to one with a large rarefaction. Both Riemann problems depict a small shock in the second family, and a rarefaction in the first family. The 2-characteristics are shown only near the shock wave, and the 1-characteristics are shown through the rarefaction in both figures. (a) A typical Riemann problem that arises for regions where the solution is smoothly varying, showing that the characteristic transitions are very small in both families. For these types of problems, $q_l \approx q_r$, implying that $\lambda^p(q_l) \approx \lambda^p(q_r) \approx s_{\mathcal{E}}^p$. (b) An exceptional Riemann problem with a large transonic rarefaction in the first characteristic family. Note that $s_{\mathcal{E}}^1 = \lambda^1(q_l) << \lambda^1(q^*)$.

shown in Figure 6.3 (b), it is no longer true that $s^1_{\mathcal{E}} \approx \lambda^1(q^*)$. If we consider such a Riemann solution in *h*-*hu* phase plane as shown in Figure 6.4, the rarefaction traces out a smooth integral curve that is not well approximated by the vector

$$\begin{bmatrix} 1 \\ s^1_{\mathcal{E}} \end{bmatrix}, \tag{6.50}$$

which is tangent to the integral curve at one of its endpoints. In fact, for a strong rarefaction, the vectors

$$\begin{bmatrix} 1 \\ \lambda^1(q_l) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 \\ \lambda^1(q^*) \end{bmatrix}, \tag{6.51}$$

might be far from parallel in phase space. The decomposition (6.49) approximates the first wave with only one vector based on $\lambda^1(q_l) = s^1_{\mathcal{E}}$. Since the augmented decomposition (6.20) provides a larger set of vectors, we can better approximate the solution in phase space as a linear combination of vectors, by using $\tilde{r}^3$ as another approximation to the integral curve. For instance, the decomposition

$$\begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_r - \begin{bmatrix} h \\ hu \\ \phi \end{bmatrix}_l = \begin{bmatrix} 1 & 1 & 1 \\ s^1_{\mathcal{E}} & s^2_{\mathcal{E}} & \lambda^1(q^*) \\ (s^1_{\mathcal{E}})^2 & (s^2_{\mathcal{E}})^2 & (\lambda^1(q^*)) \end{bmatrix} \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{bmatrix}, \tag{6.52}$$

would produce a much better approximation to the actual Riemann problem in phase space when $\lambda^1(q^*) >> s^1_{\mathcal{E}}$. See Figure 6.4.

It therefore makes sense to use an approximation to

$$\begin{bmatrix} 1 \\ \lambda^k(q^*) \\ (\lambda^k(q^*))^2 \\ 0 \end{bmatrix}, \tag{6.53}$$

for $\tilde{r}^3$ in (6.20), when there is a strong rarefaction in the k^{th} characteristic family. Recall from section Section 4.2 that the entire Riemann structure can be determined by the simple evaluation (4.33). If a rarefaction exists in either family, the Riemann invariants (4.19) hold through those rarefactions, and we can use that to determine, or at least to estimate, the characteristic speed on the inside of a rarefaction. For instance, suppose that there is a rarefaction in the first characteristic family, so that the first Riemann invariant holds for the middle star region:

$$u_l + 2\sqrt{gh_l} = u^* + 2\sqrt{gh^*}. \tag{6.54}$$

Since $\lambda^1(q^*) = u^* - \sqrt{gh^*}$, we can subtract $3\sqrt{gh^*}$ from both sides of (6.54), giving

$$\lambda^1(q^*) = u_l + 2\sqrt{gh_l} - 3\sqrt{gh^*}. \tag{6.55a}$$

Similarly, for a rarefaction in the second family we have,

$$\lambda^2(q^*) = u_r - 2\sqrt{gh_r} + 3\sqrt{gh^*}. \tag{6.55b}$$

Moreover, the differences

$$\lambda^1(q^*) - \lambda^1(q_l) = 3\left(\sqrt{gh_l} - \sqrt{gh^*}\right), \tag{6.56a}$$

$$\lambda^2(q_r) - \lambda^2(q^*) = 3\left(\sqrt{gh_r} - \sqrt{gh^*}\right), \tag{6.56b}$$

indicate the strength of the rarefaction in the first and second field respectively. If we have an estimate h^*, denoted h_0^*, then we can estimate (6.55) and (6.56). Recall that in the case of two rarefactions, the middle state depth could be exactly determined, and $h_0^* = h^*$. If only one rarefaction exists, considering Theorem 4.2 and (4.33), a single secant iteration provides an estimate $h_0^* \geq h^*$. This provides a conservative estimate for (6.55a) or (6.55b), and for the strength of the single rarefaction.

6.7.3 Choosing the Vector $\tilde{r}^3$

Allowing various forms for the eigenvector $\tilde{r}^3$ presents an additional task when developing algorithms for an approximate Riemann solver, however, it also endows the Riemann solver (6.20) with flexibility that will allow it to accommodate diverse types of Riemann problems. The choice can be systematically made, based on some practical considerations and observations.

First, note that in the augmented Riemann solver (6.20), when using a vector of the form

$$\tilde{r}^3 = \begin{bmatrix}1 & s & (s)^2 & 0\end{bmatrix}^T, \tag{6.57}$$

where s is the speed of the wave, the effective depths defined by the second component of the decomposition are equivalent to the actual depths defined by the first component. Therefore, positivity of the effective depths can be easily checked after the decomposition. If any are negative, the decomposition can be re-performed with (6.45) for $\tilde{r}^3$, ensuring positivity. In practice this may only be necessary for Riemann problems near a potential dry state.

For problems where there is no strong rarefaction, the vectors in (6.51) might become nearly co-linear, and (6.52) might become poorly conditioned, suggesting that (6.45) is a better choice for $\tilde{r}^3$. For problems with two large rarefactions, positivity can be an issue due to incipient cavitation—described in Section 4.3—and so (6.45) might again be the appropriate choice. As one last caveat, it is possible to have a very large rarefaction where $\lambda^1(q_0^*)$ approaches $s_{\mathcal{E}}^2$, or similarly for $\lambda^2(q_0^*)$, however, this possibility is reduced by the estimate $h_0^* \geq h^*$, since the inequality produces conservative estimates for the speeds (6.55). Therefore, if this occurs, it must be due to inadequate estimations of $s_{\mathcal{E}}^1$ or $s_{\mathcal{E}}^2$ in the case of a shock. This problem is pronounced in regions of vanishing depths, and will be discussed in detail in the next section where a unified solution will produce a robust solver for the various types of Riemann problems.

To conclude this section, note that when a vector of the form (6.57) is used in the augmented Riemann solver for $\tilde{r}^3$, using the first two components of the decomposition and the speeds to define waves for the update is entirely equivalent to using the flux decomposition. The approximate Riemann solution must conserve the first two components of the solution then, even when the Roe speeds are not used. A decomposition of this form therefore unifies the f-wave approach and the standard approximate Riemann solvers, by introducing another wave. If maintaining this property is desired, even when linear dependence prevents using an approximation to (6.53), the vector

$$\begin{bmatrix}1 & \bar{s}_{\mathcal{E}} & (\bar{s}_{\mathcal{E}})^2 & 0\end{bmatrix}^T, \tag{6.58}$$

where $\bar{s_{\mathcal{E}}} = \frac{1}{2}(s^1_{\mathcal{E}} + s^2_{\mathcal{E}})$, provides an alternative that avoids linear dependence. This is also a suitable solution to the necessary fix required if $\tilde{r}^3$ is associated with a zero speed $\bar{s}_{\mathcal{E}}$.

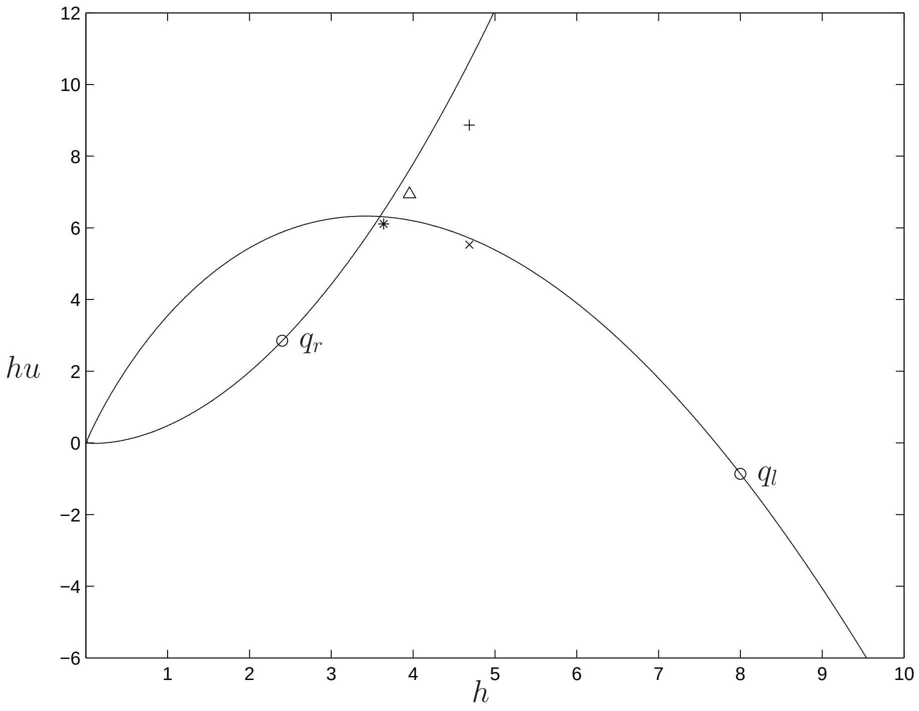


Figure 6.4: A Riemann problem in the h-hu phase plane, showing an integral curve in the first family when there is a large rarefaction. The intersection with the Hugoniot locus in the second family determines the true middle state. The vector $\begin{pmatrix}1 & s^1_{\mathcal{E}}\end{pmatrix}^T$ is tangent to the integral curve at q_l, and $\begin{pmatrix}1 & \lambda^1(q^*)\end{pmatrix}^T$ is tangent to the integral curve at the intersection q^*. The two vectors are far from parallel, and neither closely approximate the vector $q^* - q_l$ in phase space. The approximate middle states that result from using different Riemann solvers are shown as points in the plane. The decomposition of the solution into the eigenvectors associated with the Einfeldt speeds, produces the middle state marked with a plus $+$, and the actual HLLE middle state $q^*_{\mathcal{E}}$ is shown as a cross $\times$. Note that the depth of the two is the same, but conservation of the latter implies that the momenta must be different. The Roe eigenvector decomposition, shown as a triangle Δ, produces a better approximation since $\hat{r}^1$ is not tangent to the integral curve at its furthest point. The augmented decomposition produces two middle states since it has an extra wave, the one at $x = 0$ is shown as an asterisk $*$. Ultimately, it is the f-waves arising from the augmented decomposition that provide the numerical update, not the approximate middle state q^*, however, the figure demonstrates that one of the weakness of the HLLE solver can be overcome by allowing more waves.

6.8 Inundation

One of the added benefits of using finite volume methods for the integral form of the shallow water equations, is that the formal assumptions still hold when applied to regions surrounding wet and dry interfaces. Therefore, we can avoid special discretizations and specialized tracking or ad hoc treatments of the shorelines. In fact, we can accurately model inundation by simply solving Riemann problems between wet and dry cells—using the exact same formalism as for the rest of the domain. Of course, we must ensure that our approximate Riemann solver accurately and robustly deals with problems between wet cells and cells with zero or vanishing depths. In this last section of the chapter, we will describe modifications that will be made to the augmented Riemann solver (6.20) necessary for inundation modeling. We will first describe some modifications to the wave speed estimates and eigenvectors in the context of constant bathymetry. Finally, we will describe the complication that arises when there is variable bathymetry in a Riemann problem between a wet cell and a dry cell.

6.8.1 Wave Speeds at the Wet-Dry Front

Recall from Section 4.3 that the solution to a Riemann problem between a wet cell and a dry cell has a single rarefaction connecting the two states. The leading edge of this rarefaction, representing the tip of the shoreline, moves at a speed of

$$u_l + 2\sqrt{gh_l} \quad \text{if} \quad h_l > 0, \tag{6.59a}$$

or

$$u_r - 2\sqrt{gh_r} \quad \text{if} \quad h_r > 0. \tag{6.59b}$$

Note that these are the same speeds as (6.55a) and (6.55b), since in this case $h^* = 0$. For finite grid spacings where the depth is positive in cells adjacent to a dry region, standard estimates for the wave speeds underestimate the true speeds (6.59). Toro [101] discusses standard Riemann solvers for the shallow water equations in relation to this problem. To correctly model such Riemann problems, our Riemann solver should have waves that move at these speeds, capturing correctly the advancing shoreline.

In the previous section it was shown that the wave associated with $\tilde{r}^3$ could be used as an additional vector to represent an integral curve associated with a large rarefaction. We could use $\tilde{r}^3$ to represent the leading edge of the rarefaction in phase space, by using (6.53) where $\lambda^k(q^*)$ is given by (6.59), depending on which state is dry. However, we will take a slightly different approach that will allow the same definitions in the Riemann solver for all Riemann problems, so that Riemann problems with vanishing depths will be treated the same as those with exactly zero depths.

When one of the initial states in a Riemann problem is dry the eigenvalues of that state are undefined, so that $s^1_{\mathcal{E}}$ must be defined by the Roe speed $\hat{\lambda}^1$ when $h_l = 0$, and similarly

$s^2_{\mathcal{E}} = \hat{\lambda}^2$ if $h_r = 0$. The Roe speeds for these two cases are respectively

$$\hat{\lambda}^1 = u_r - \sqrt{\frac{gh_r}{2}} \quad \text{if} \quad h_l = 0, \tag{6.60a}$$

$$\hat{\lambda}^2 = u_l + \sqrt{\frac{gh_l}{2}} \quad \text{if} \quad h_r = 0. \tag{6.60b}$$

Comparing (6.60) and (6.59), we see that $\lambda^2(q^*) < \hat{\lambda}^1$, and $\hat{\lambda}^2 < \lambda^1(q^*)$ for the two respective

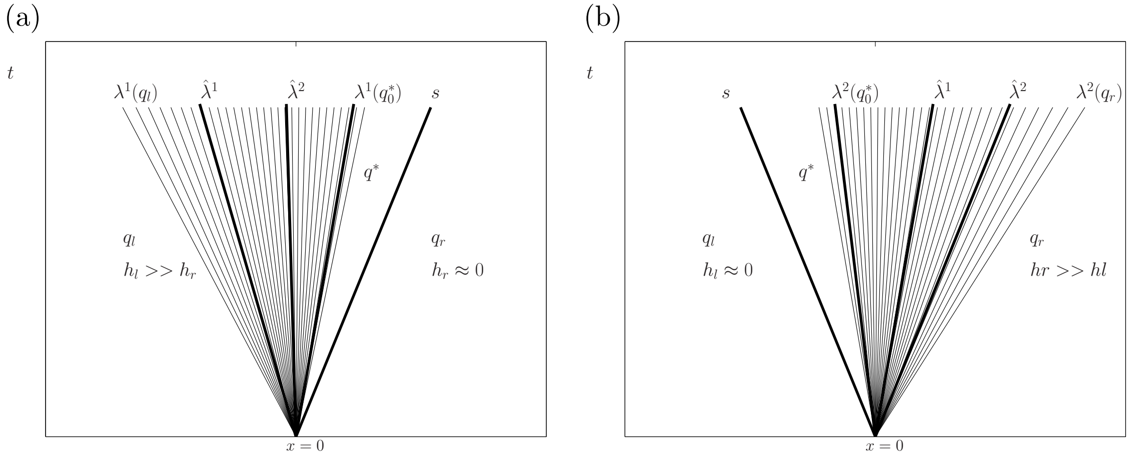


Figure 6.5: Insufficient speed estimates for near-dry-state Riemann problems. (a) Example of possible wave structure when $h_r \approx 0$. The right edge of the 1-rarefaction moves at a speed $\lambda^1(q^*)$ that is actually greater than the Roe estimate $\hat{\lambda}^2$ for the shock speed s. If $\lambda^2(q_r) < \lambda^1(q_0^*)$, then $s^2_{\mathcal{E}}$ is replaced by $\lambda^1(q_0^*)$. Note that $\lambda^1(q_0^*)$ is still less than $\lambda^1(q^*)$, which must be less than the true shock speed because of the Lax entropy condition (Definition 2.3). (b) The analogous situation when $h_l \approx 0$.

dry state problems—implying that the true single wave actually spreads faster than our estimate for the speed of the other characteristic family. This is not unique to the dry state Riemann problem however, but tends to occur even when one of the states is very shallow compared to the other, $h_l << h_r$ or $h_r << h_l$. In these problems, a large rarefaction in one field produces a speed $\lambda^{1,2}(q^*)$ that actually approaches or surpasses the estimate for the speed of the other field $s^{2,1}_{\mathcal{E}}$. Since the Eindfeldt speeds are intended to bound all of the characteristics emanating form the origin, and $\lambda^{1,2}(q_0^*)$ are conservative estimates for (6.55), this only occurs when $s^{1,2}_{\mathcal{E}}$ are inadequate estimates of the true shock speed. For instance, suppose that the estimate $\lambda^1(q_0^*)$ is greater than $s^2_{\mathcal{E}}$. Then $\lambda^2(q^*) > \lambda^1(q^*) > \lambda^1(q_0^*) > s^2_{\mathcal{E}} \geq u_r + \sqrt{gh_r}$, implying that we must have a shock in the second family, as guaranteed by the Lax entropy condition (which also guarantees that the true shock must move faster than $\lambda^1(q^*)$). See Figure 6.5 showing the characteristics and speed estimates for this situation.

It therefore makes sense to redefine the speeds $s^1_{\mathcal{E}}$ and $s^2_{\mathcal{E}}$ in the augmented solver (6.20),

$$\tilde{s_{\mathcal{E}}}^1 = \min(s^1_{\mathcal{E}}, \lambda^2(q^*_0)), \tag{6.61a}$$

$$\tilde{s_{\mathcal{E}}}^2 = \max(s^2_{\mathcal{E}}, \lambda^1(q^*_0)), \tag{6.61b}$$

whenever there is a rarefaction in the first or second characteristic field respectively. Note that when there is a rarefaction, because $h^*_0 \geq h^*$, the new definition (6.61) only decreases the Einfeldt speed $s^1_{\mathcal{E}}$ to a value still greater than the true speed of the 1-shock, and only increases $s^2_{\mathcal{E}}$ to a value still less than the true speed of the 2-shock.

Note that with the expanded definition (6.61), when we have a dry state problem, the integral curve is now approximated with $\tilde{r}^1$ and $\tilde{r}^2$, and the speeds of the waves associated with these vectors are the same as the far edges of the rarefaction, including the advancing (or receding) wet/dry front. For these problems, the vector $\tilde{r}^3$ should be defined by (6.45), which ensures positivity since the HLLE depth $h^*_{\mathcal{E}}$ is determined using the speeds (6.61). (See the comment after Theorem 3.1.)

It should be noted that for the practical issue of implementing the corrections described in this section, dry state problems need no special consideration. For instance, if a rarefaction exists in either family, say the first family, then the value h^* is quickly estimated and the speed of the rarefaction edge (6.55a) estimated. This determines the speed $\tilde{s_{\mathcal{E}}}^2$, which typically is the same as $s^2_{\mathcal{E}}$. If $\lambda^1(q^*_0)$ is sufficiently close to either $s^1_{\mathcal{E}}$ (as in the case of a weak rarefaction) or $s^2_{\mathcal{E}}$ (such as for a near dry state problem), then $\tilde{r}^3$ is defined by (6.45), and the associated wave moves at a speed $\bar{s}_{\mathcal{E}} = \frac{1}{2}(\tilde{s}^1_{\mathcal{E}} + \tilde{s}^2_{\mathcal{E}})$. If two strong rarefactions exist, then $\tilde{r}^3$ is also defined by (6.45) in order to ensure positivity. Otherwise, $\tilde{r}^3$ can be defined by

$$\tilde{r}^3 = \begin{bmatrix} 1 & \lambda^1(q^*_0) & (\lambda^1(q^*_0))^2 & 0 \end{bmatrix}^T . \tag{6.62}$$

It is also easy to show that defining $u_l = 0$, when $h_l = 0$, so that $\lambda^1(q_l) = 0$, will have no effect on the numerical solution, if the above procedure for the wave speeds with a dry state are taken.

6.8.2 Variable Bathymetry and Shorelines

As with most other approximate Riemann solvers, our wave speed estimates and approximate eigenvectors will not be altered by the existence of variable bathymetry and the resulting source term. However, we must treat the Riemann problem between a dry cell and a wet cell carefully when there is a jump in bathymetry between the two cells. For instance, suppose we are solving the Riemann problem depicted in Figure 6.6, where $h_r = 0$, and $h_l + b_l < b_r$. This situation arises naturally with any discretization near the shoreline (unless special care is taken to ensure that $h_l + b_l = b_r$). Solving such a Riemann problem with the methods described so far will be shown to produce a spurious result. This is most easily seen by considering the steady state problem, where we expect the Riemann solution to produce no waves, so that the shoreline remains unaffected. However, consider the initial

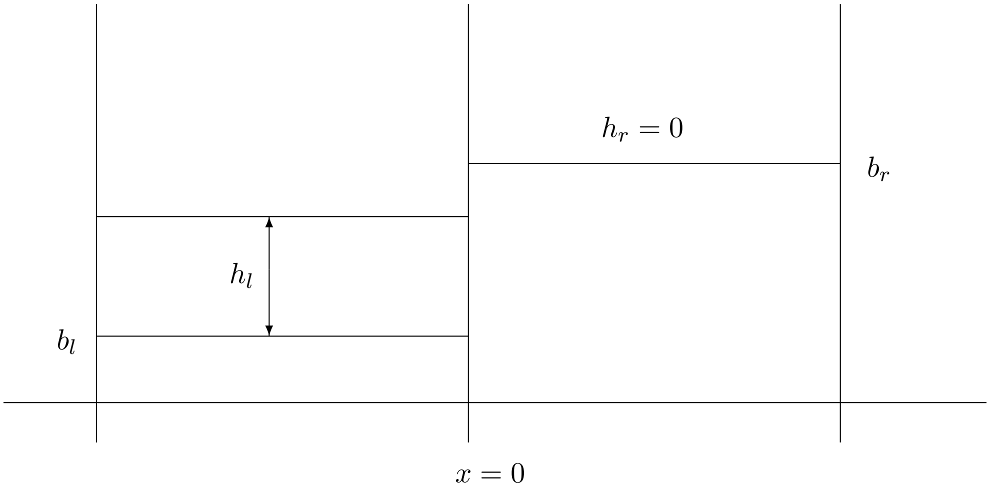


Figure 6.6: A Riemann problem at the shoreline, showing $h_l + b_l < b_r$. If this Riemann problem represents the steady state where $u_l = 0$, solving it without modification will result in spurious waves from the shoreline.

jump

$$\begin{bmatrix} h \\ hu \\ \phi \\ b \end{bmatrix}_r - \begin{bmatrix} h \\ hu \\ \phi \\ b \end{bmatrix}_l = \begin{bmatrix} -h_l \\ 0 \\ -\frac{g}{2}h_l^2 \\ b_r - b_l \end{bmatrix}. \tag{6.63}$$

If the steady state is to be maintained with no propagating waves, then (6.63) should be a multiple of the steady state vector $\tilde{r}^4$, which, if we consider the averages of the initial left and right states, is

$$\begin{bmatrix} -1 \\ 0 \\ -\frac{g}{2}h_l \\ 1 \end{bmatrix}. \tag{6.64}$$

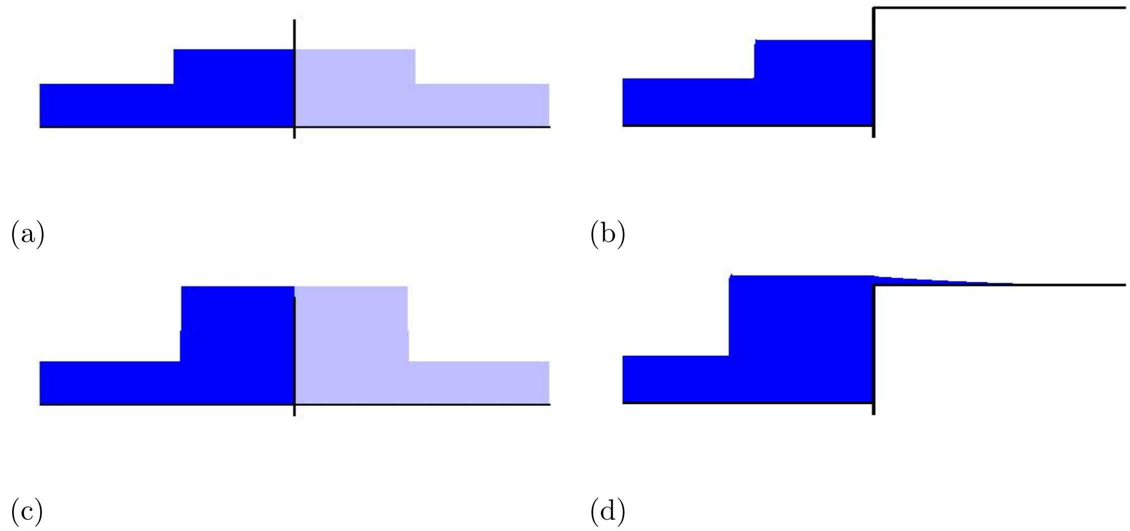
Preservation of the steady state would therefore require $h_l = b_r - b_l$, which would occur if the surface on the left was equal to the dry elevation on the right, which is not the case for this problem. This makes intuitive sense if one considers the problem from a physical standpoint—if the water is motionless, the elevation of the dry topography on the right is irrelevant if it is higher than $h_l + b_l$. If we solved the problem as is, spurious waves would be generated at the shoreline, which is clearly unacceptable. These waves can be quite large in fact for tsunami modeling, where a large domain requires large cells, so that often cells with depths approaching hundreds of meters or more are adjacent to dry cells.

An easy solution for this steady state problem would be to simply reduce the value b_r to $h_l + b_l$. However, consider what would happen if a small positive velocity was added to the cell on the left, so that $u_l > 0$. A resulting nonzero jump in $hu_r - hu_l$ would again cause propagating waves, and water would inundate the cell to the right. If b_r was actually significantly greater than $h_l + b_l$, then the inundation would be spurious. The same problem would again be repeated for the Riemann problem one cell to the right, and large spurious inundations would occur as a pure numerical artifact. It is imperative that these problems be treated correctly if inundation is to be correctly modeled.

The approach we take in this situation is to solve a preliminary Riemann problem, where we replace the values in the dry cell with *ghost cell values*—values used simply to determine the Riemann solution. (That is, we do not actually change the values stored in the dry cell.) Consider again the Riemann problem depicted in Figure 6.6, where $h_l + b_l > h_r$, but assume that u_l can have any value. We first solve the homogeneous Riemann problem with ghost cell values in the right cell that approximate a wall boundary condition:

$$\begin{bmatrix} h_r \\ (hu)_r \\ \phi_r \\ b_r \end{bmatrix} = \begin{bmatrix} h_l \\ -(hu)_l \\ \phi_l \\ b_l \end{bmatrix}. \tag{6.65}$$

The solution to such a Riemann problem is easy to determine, since it must be symmetrical about $x = 0$. Clearly, if $u_l < 0$, then we have a double rarefaction, with $h^* < h_l$. Conversely, if $u_l > 0$, then we have a double shock, with shocks propagating away at the same speeds with opposite signs, with $h^* > h_l$. Due to the symmetry of either case, it is easy to show that the decomposition (6.20) will produce only two waves with equal strength, $\mathcal{W}^1$ moving to the left and $\mathcal{W}^2$ moving to the right, regardless of the form of $\tilde{r}^3$. We then estimate $h*$, either by checking our approximate Riemann solution, or by approximating the root of the function f described in Section 4.2. (Since we are solving a homogeneous problem, h^* can be easily and confidently determined.) We then determine if h^* is large enough to inundate the "wall," *i.e.* the jump in bathymetry. That is if $h^* + b_l < b_r$, then no inundation would occur if the jump in bathymetry was actually a wall. In this case, the solution to the ghost cell problem serves to update q_l, by using the leftward moving waves from the Riemann solution. However, if $h^* + b_l \geq b_r$ then the jump in bathymetry does not form a large enough barrier to prevent inundation. In this case, the ghost problem is discarded, and the actual Riemann problem is solved with the actual values from the right cell. This ensures that the full source term is realized when the inundating water has enough momentum to overcome the entire slope. This approach in depicted in Figure 6.7, showing the ghost cell problem and the actual Riemann problem for the two possible cases.



(a)

(b)

(c)

(d)

Figure 6.7: The wall boundary condition Riemann problem as a test for the source term. The left panel shows the ghost cell Riemann problems for two different cases, and the right panel shows the update for the actual Riemann problem in the same cases. (a) The ghost cell problem has equal and opposite velocities in each cell, with $u_l > 0$, resulting in two shocks propagating away from the initial interface. The resulting middle depth h^* is not high enough to overtop the jump in bathymetry, $h^* < b_r - b_l$. (b) The actual Riemann problem corresponding to *(a)*. The leftward moving waves from the ghost cell problem get used to update the solution in the left cell. (c) A ghost cell problem similar to that shown in (a), yet with greater initial velocities, causing $h^* > b_r - b_l$. (d) The actual Riemann problem gets used to update the solution in both cells.

6.9 Extension to the Two-Dimensional Problem

Recall from Chapter 3 that we extend the basic wave propagation algorithms to multiple dimensions by solving normal one-dimensional Riemann problems corresponding to the higher-dimensional system. For the shallow water equations in two dimensions, recall from Section 2.5 that the Riemann problem is unchanged, except for an additional wave carrying the jump in the transverse velocity $(v_r - v_l)$, moving at the speed u^*. For an f-wave type of approach, where the transverse flux huv is decomposed into a set of waves, the two-dimensional Riemann problem is solved exactly as the one-dimensional problem, except that the waves defined in the one-dimensional problem have another component carrying the corresponding jumps in huv, and one additional wave carrying the remaining jump in huv, moving at an estimate to the speed u^*. For the augmented solver, the jump in huv across the wave $\mathcal{Z}^p$ (defined by the one-dimensional problem) is simply $v_l(\mathcal{Z}^p)^2$ if $s^p < u_0^*$, and $v_r(\mathcal{Z}^p)^2$ if $s^p > u_0^*$, where u_0^* is an estimate to the middle state velocity. The remaining jump

$$(huv)_r - (huv)_l - \sum_{\{p:s^p<u_0^*\}} v_l(\mathcal{Z}^p)^2 - \sum_{\{p:s^p>u_0^*\}} v_r(\mathcal{Z}^p)^2, \qquad (6.66)$$

then determines the additional f-wave moving at the speed u_0^*. Therefore, the only significant added task for the two-dimensional Riemann problem is to determine an estimate u_0^*. It is desired that this average have several properties. First, it should be a simple and consistent estimate for u^*. Second, in the case of a strong rarefaction such as in the initial dry state problem, the speed should not be intermediate to the speeds at the edges of the rarefaction wave. This is because the transverse contact discontinuity in such a case is to one side of the rarefaction. In fact, for the initial dry state problem, the transverse contact discontinuity is precisely at the advancing wet-dry front moving at the speed of the advancing edge of the single rarefaction. These properties are satisfied by using the value

$$u_0^* = \min(\lambda^2(q_0^*), \max(\lambda^1(q_0^*)), \bar{s}_{\mathcal{E}}), \qquad (6.67)$$

when a rarefaction exists, and $u_0^* = \bar{s}_{\mathcal{E}}$ otherwise, where $\bar{s}_{\mathcal{E}} = \frac{1}{2}(\tilde{s}_{\mathcal{E}}^1 + \tilde{s}_{\mathcal{E}}^2)$ using the definitions (6.61). Note that when there is no strong rarefaction, this corresponds to the speed of the wave associated with $\tilde{r}^3$. For a dry state Riemann problem, it corresponds to the speed of the advancing front.

Chapter 7

ADAPTIVE MESH REFINEMENT FOR TSUNAMI MODELING

So far our efforts have been directed at the difficulty of diverse flow *regimes* associated with tsunamis—our aim has been to develop algorithms that are reasonably robust and accurate for all of the regimes. Recall that these diverse regimes occur at extremely diverse *scales*, and we have not yet dealt with the obvious difficulty of simply being able to resolve these diverse scales in a single computation. In this chapter we confront that difficulty by describing adaptive mesh refinement algorithms. These algorithms will "bridge the gap" in the sense that they will allow the algorithms described previously to be simultaneously applied to the diverse spatial scales and flow regimes in a single computation. In fact, adaptive mesh refinement is one of the main motivations for designing algorithms that are suitable for all of the regimes of tsunami flow, otherwise it might be feasible to switch algorithms for different computations involving different flow regimes.

The initial profile of a tsunami is determined by the geometry and nature of the generating fault motion. For most global scale tsunamis the initial wavelength is hundreds of kilometers, and the disturbance can be spread over thousands of kilometers. (For the Sumatra-Andaman earthquake, the rupture zone encompassed a huge swath 400 km wide by 1200 km long—one of the largest single ruptures ever recorded [3, 43]). Since initial amplitudes are often less than a meter, tsunami energy is initially spread thin over large areas, and this energy does not easily dissipate until it is more locally concentrated. This allows tsunamis to travel across entire oceans without significant dissipation. Obviously modeling this propagation requires a large computing domain, at least big enough to include the source and coastlines of interest—sometimes in different hemispheres. Fortunately the wavelength of this propagation is large enough that we can capture it with a fairly coarse discretization.

As tsunami waves enter coastal shallow water their wavelength is compressed. This occurs due to the decreasing wave speeds—recall that the wave speeds are $u \pm \sqrt{gh}$. In the deep ocean this implies wave speeds of up to 200 m/s, or equivalently 720 km/hr. In the near shore the wave speeds decrease with roughly the square root of depth so the wave is dramatically compressed as the leading edge slows rapidly while the trailing edge is still in the deep ocean. This compression occurs continually as the waves approach the shore, halving the wavelengths many times over. This energy compression is what tsunamis owe their destructive capacity to—only at the shoreline is the energy dissipated. This decrease in wavelength near the shore requires an equivalent increase in computational grid resolution. If specific features of inundation are to be captured, necessary grid resolutions may be several meters—orders of magnitude smaller than in the deep ocean. With the energy more compressed the flow exhibits more violent and erratic features contributing to the unpredictability. These flow features add to the need for higher grid refinement.

A simple solution to this problem, and one frequently employed, is to use a set of fixed telescoping grids at the shoreline—typically only near a region or coastline of interest. This is a legitimate approach for many problems, particularly for problems where one is modeling a single community for future hazard mitigation, and can dedicate days or weeks of computational time to the problem. However, for global-scale modeling this is less than ideal for several reasons. First, it is inefficient when modeling the entire ocean. Although only a relatively coarse resolution is needed to capture deep ocean propagation, this resolution is still quite expensive if used throughout an entire ocean for an entire computation. Since tsunami waves propagate long distances, using fixed grids requires modeling an entire ocean at the level required to resolve the propagating tsunami. This means that a large amount of computing effort gets dedicated to steady-state calculations in regions where nothing is happening. Second, it requires either under-resolving the majority of coastal regions that might be impacted by a global-scale tsunami, or perhaps using some *a priori* knowledge of where inundation will be the most severe. Since depth and hence wave speeds are a function of the variable seafloor bathymetry, the compressing energy of a tsunami as it approaches shore is focused by bathymetry in complex and unpredictable ways. Often destruction varies greatly even along local stretches of coastline. Determining what communities or regions are vulnerable to a given fault event, using fixed grids, essentially requires a separate calculation for each region. This is certainly not the most efficient use of computational resources, nor is it practical for real-time tsunami modeling. Third, the fixed fine-scale grids are typically integrated for the entire computation, even long before the tsunami arrives. This again is computationally wasteful and ultimately decreases the degree to which the tsunami can be resolved due to misappropriation of resources.

7.1 Adaptive Mesh Refinement for Hyperbolic Systems

Adaptive refinement methods in general use variable sized grid spacings depending on the nature of the solution in different regions of the domain. The regions of refinement adapt to the changing solution based on some set of criteria relevant to the problem. These methods are particularly attractive for hyperbolic problems since interesting features in the solutions are often highly localized, yet, because of the wave-like nature of the solutions, the features often move throughout the domain. We have extended *Berger-Collela-Oliger* [9, 7] adaptive mesh refinement to the problem of tsunami modeling. These methods use multiple levels of rectangular Cartesian subgrids that use integer refinement ratios in space and time. Multiple subgrids at a given level may exist and are adaptively arranged in patches that cover important features in the solution (see Figure 7.1). (See Chapter 8 for a discussion of how these Cartesian computational grids can be used for more general geometries.) Since the size and location of the subgrids can change at any time depending on the level of refinement needed in a specific area, the grids can effectively move with important features in the solution. These methods were originally developed for hyperbolic systems such as the Euler equations of compressible gas dynamics, where often shock waves and very steep gradients are highly localized and need much higher levels of refinement than the rest of domain. Since the shock waves move throughout the domain, fixed telescoping grids would be ineffective.

These adaptive mesh refinement methods have previously been implemented in the gen-

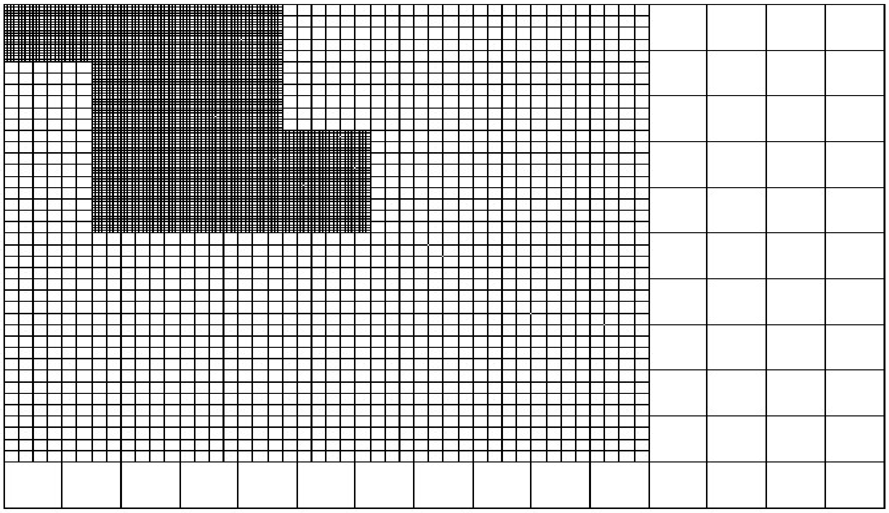


Figure 7.1: Example of adaptive mesh refinement with rectangular Cartesian grids. A portion of the domain is shown with three levels of grids refined by factors of 4. Typically many grids at a given level (> 1) may coexist refining important features in the solution with rectangular patches.

eral framework of the wave-propagation algorithms [8], and are a standard extension of the CLAWPACK software [61], called AMRCLAW. Since the adaptive subgrids are rectangular Cartesian grids, the wave propagation algorithms used for single grids can be applied virtually unchanged to each subgrid. However, special treatment of the subgrid boundaries is required, since these are in the interior of the domain, and adjacent to coarser grid cells. For a complete discussion of this, and adaptive mesh refinement in general, see [8, 7, 9]. For the purpose of tsunami modeling, some modifications had to be made to AMRCLAW in order to maintain the steady-states, and new interpolation strategies had to be developed for proper treatment of the shoreline. These modifications and the relevant aspects of the adaptive mesh refinement algorithms are discussed in the following sections.

7.2 Maintaining Conservation and Steady States for Tsunami Modeling

Since adaptive subgrids appear dynamically in the domain, the solution in the fine cells of a subgrid must be determined from the next coarser level through some type of interpolation. Additionally, when a subgrid moves or disappears, the fine grid cells that are eliminated must be averaged in some way to determine the value in the underlying coarser cell. It is important that the interpolation and averaging strategies do not destroy essential features of the underlying numerical method. For instance, since we have taken special care to ensure that our algorithms are conservative, we must require that interpolation and averaging maintain numerical conservation whenever possible. Additionally, we do not want interpolation to generate oscillations near discontinuities. For tsunami modeling, it is important that the delicate steady state is preserved when interpolating. Additionally, interpolation must be performed carefully near dry states and shorelines, as described in Section 7.3.

7.2.1 Conservation

Suppose that we have a level $(l+1)$ grid overlying the cell $\mathcal{C}_{ij}^{l}$ on the coarser level l grid. (Superscripts on variables will be used to denote their grid level.) Cell $\mathcal{C}_{ij}^{l}$ will contain an integer number of level $(l+1)$ grid cells

$$\mathcal{C}_{ks}^{l+1} = [x_{k-1/2}, x_{k+1/2}] \times [y_{s-1/2}, y_{s+1/2}] \subset \mathcal{C}_{ij}^{l}. \tag{7.1}$$

For the level l grid cell $\mathcal{C}_{ij}^{l}$, the set

$$\Gamma_{ij}^{l+1} = \left\{(k,s) : \mathcal{C}_{ks}^{l+1} \subset \mathcal{C}_{ij}^{l}\right\} \tag{7.2}$$

contains the indices of the level $(l+1)$ cells, and $\left|\Gamma_{ij}^{l+1}\right|$ will be used to denote the number of such cells, which depends on the spatial refinement ratios used from level l to level $(l+1)$. Figure 7.2 illustrates this situation, showing part of a level l grid and the level $(l+1)$ grid cells in the cell $\mathcal{C}_{ij}^{l}$. When the grid resolution changes in a region of the domain, it is important that conservation is maintained, *i.e.* that the conserved quantities integrated on the fine grid are equal to those on the coarse grid. This is accomplished if

$$Q_{ij}^{l} = \frac{1}{|\Gamma_{ij}^{l+1}|} \sum_{(k,s)\in\Gamma_{ij}^{l+1}} Q_{ks}^{l+1}, \tag{7.3}$$

where Q_{ks}^{l+1} is the solution in grid cell $\mathcal{C}_{ks}^{l+1}$ and

Maintaining conservation during interpolation from a coarse grid can be easily accomplished by using cell-centered linear interpolants in each coarse grid cell for the conserved variables. That is, the values in these finer grid cells are determined by evaluating, at their cell centers, a simple linear interpolant

$$L_{ij}(x,y) = Q_{ij}^{l} + \sigma_{ij}^{x}(x - x_i) + \sigma_{ij}^{y}(y - y_j), \tag{7.4a}$$

$$Q_{ks}^{l+1} = L_{ij}(x_k, y_s), \tag{7.4b}$$

where σ_{ij}^{x} and σ_{ij}^{y} are slopes in the x and y directions respectively, determined by some consistent evaluation of the surrounding coarse cells. If the interpolation (7.4) is used for all of the fine cells within a coarse cell (7.3) is maintained, as can easily be seen by considering symmetry of the interpolant around the coarse cell center. Note that not all forms of linear interpolation will maintain conservation. For instance, *bilinear interpolation*, which could be used to determine cell-centered values in each fine cell based on the values at the four nearest coarse cell centers, would not preserve conservation in general.

The slopes σ_{ij}^{x} and σ_{ij}^{y} could be determined in a variety of ways based on the cells surrounding cell $\mathcal{C}_{ij}^{l}$. In practice, we would like to avoid interpolation that might generate

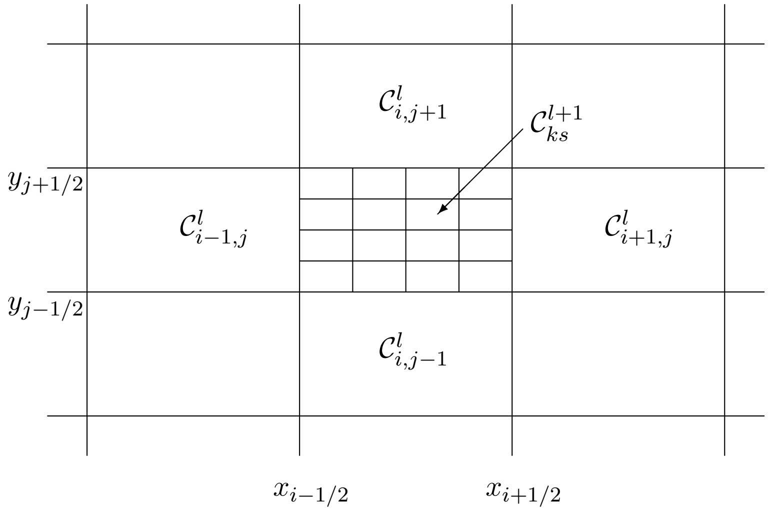


Figure 7.2: Interpolation and averaging between level l and $(l+1)$ grids. Using refinement ratios of 4 as an example, the 16 level $(l+1)$ grid cells within a single level l cell $\mathcal{C}^l_{ij}$ are shown. Linear interpolation to determine the fine grid cell values, based on (7.4), preserves conservation as can be seen by considering symmetry. Arithmetic averaging of the fine grid cell values to determine the coarse cell value also preserves conservation.

oscillations near steep gradients in the solution. This could occur if interpolation generates new extrema on the fine grid. Recall from Section **??**, where we considered the relationship between slope limiters and flux limiters, our goal was to limit the interpolating slopes in grid cells in order to prevent nonphysical oscillations in the solution by ensuring that new extrema were not generated. We can take the same approach when interpolating from coarse to fine grids—limited slopes can be used for σ^x_{ij} and σ^y_{ij} in order to prevent generating new extrema. For instance, using the minmod slopes

$$\sigma^x_{ij} = \text{minmod}\left(\frac{Q^l_{i+1,j} - Q^l_{ij}}{\Delta x}, \frac{Q^l_{ij} - Q^l_{i-1,j}}{\Delta x}\right), \tag{7.5a}$$

$$\sigma^y_{ij} = \text{minmod}\left(\frac{Q^l_{i,j+1} - Q^l_{ij}}{\Delta y}, \frac{Q^l_{ij} - Q^l_{i,j-1}}{\Delta y}\right), \tag{7.5b}$$

prevents the generation of new extrema even in two dimensions when using (7.4). Recall that the minmod function is defined by

$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0 & \text{if } ab \le 0. \end{cases} \tag{7.6}$$

Occasionally, areas of the domain covered by level $(l+1)$ grids will no longer need high levels of refinement, and the level $(l+1)$ grid cells will be averaged to determine a new level l coarse grid cell value. Clearly, conservation is maintained if the coarse cell value Q^l_{ij} is determined by the simple arithmetic average of the solution in the level $(l+1)$ grid cells within $\mathcal{C}^l_{ij}$, since then Q^l_{ij} is determined by (7.3).

7.2.2 Steady States

In the previous section we described how conservation is maintained when refining or coarsening grids, by linear interpolation and arithmetic averaging of the conserved numerical variables Q. For the shallow water equations, this implies interpolating and averaging the depth h and momenta hu and hv. However, interpolating h will upset the delicate steady states described in Section 5.3 and will destroy the well-balancing properties discussed in Section 6.4. To see this, consider a motionless body of water with a constant surface elevation $\eta \equiv h + b \equiv 0$ over nonlinear variable bathymetry. The bathymetry values on the coarse grid are averages of the true bathymetry values in each grid cell. When interpolating to a finer grid, the bathymetry is replaced by new averages of the true bathymetry within the finer grid cells. The true bathymetry is in general not linear, so it might vary differently than a simple linear interpolation of the coarse bathymetry would suggest. Since the depths in fine grid cells are determined by linear interpolation of the depths in the coarse cells, the new values of the surface elevation will no longer equal a constant—corresponding to the simple steady state problem. Figure 7.3 shows this for a simple one dimensional problem, where the depths in two fine grid cells are determined by minmod interpolation from the coarser grid. In this example, note that the minmod slope for the depth h would be $\sigma^x_i = \left(h^l_{i+1} - h^l_i\right)/\Delta x = -4/\Delta x$, giving values of $h^{l+1}_k = 19$ and $h^{l+1}_{k-1} = 21$ when evaluating (7.4). The bathymetry in these cells gives nonzero values of the surface elevation $\eta = h + b$.

The example shown in Figure 7.3 suggests a simple fix—interpolate η

$$L_{ij}(x,y) = \eta^l_{ij} + \sigma^x_{ij}(x - x_i) + \sigma^y_{ij}(y - y_j), \tag{7.7a}$$

$$\eta^{l+1}_{ks} = L_{ij}(x_k, y_s), \tag{7.7b}$$

where $\eta^l_{ij} = h^l_{ij} + b^l_{ij}$, and the slopes $\sigma^{x,y}_{ij}$ are the one-sided minmod slopes as before but determined by η in the coarse cells. The depth is then determined by

$$h^{l+1}_{ks} = \eta^{l+1}_{ks} - b^{l+1}_{ks}, \tag{7.8}$$

using the interpolated surface values and the fine grid bathymetry. This seems simple enough, but we must make sure that

$$\frac{1}{|\Gamma^{l+1}_{ij}|} \sum_{(k,s)\in\Gamma_{ij}} h^{l+1}_{ks} = h^l_{ij} \tag{7.9}$$

will still be satisfied for conservation of mass. As when interpolating depth, if linear interpolation (7.7) is used, the value of η^l_{ij} is the average of the surface values η^{l+1}_{ks} in the fine cells $\mathcal{C}^{l+1}_{ks} \subset \mathcal{C}^l_{ij}$,

$$\frac{1}{|\Gamma^{l+1}_{ij}|} \sum_{(k,s)\in\Gamma_{ij}} \eta^{l+1}_{ks} = \eta^l_{ij}. \tag{7.10}$$

Together with (7.8), (7.10) implies

$$\frac{1}{|\Gamma_{ij}^{l+1}|} \sum_{(k,s)\in\Gamma_{ij}} \left(h_{ks}^{l+1} + b_{ks}^{l+1}\right) = h_{ij}^{l} + b_{ij}^{l},$$

$$\Rightarrow \frac{1}{|\Gamma_{ij}^{l+1}|} \sum_{(k,s)\in\Gamma_{ij}} h_{ks}^{l+1} = h_{ij}^{l} + \left(b_{ij}^{l} - \frac{1}{|\Gamma_{ij}^{l+1}|} \sum_{(k,s)\in\Gamma_{ij}} b_{ks}^{l+1}\right). \tag{7.11}$$

We see that if the bathymetry b_{ij}^{l} in the coarse cell is the average of the bathymetry values b_{ks}^{l+1} in the fine cells $\mathcal{C}_{ks}^{l+1} \subset \mathcal{C}_{ij}^{l}$, which is the case if both come from averaging true bathymetry data, then the terms in parenthesis in (7.11) equal zero and (7.9) is satisfied.

We also want to conserve momentum when refining grids. Since the refined depth is determined by (7.7) and (7.8), it is not necessarily a simple linear function within each coarse cell. Following the idea of interpolating η, it might make sense to interpolate the velocities u_{ks}^{l+1} and v_{ks}^{l+1}, and then determine the momenta hu_{ks}^{l+1} and hv_{ks}^{l+1} by multiplying the interpolated velocities by h_{ks}^{l+1} in each fine grid cell. However, this will not conserve momentum in general, so we will simply interpolate momentum according to (7.4). This is not problematic for the most part, since the delicate background steady state that must be preserved has zero velocity. However, some precautions must be taken in very shallow regions, where the differences is depth between the interpolated cells are significant relative to the overall depth. This and other issues that arise in shallow regions and at shorelines will be explained and treated in the next section.

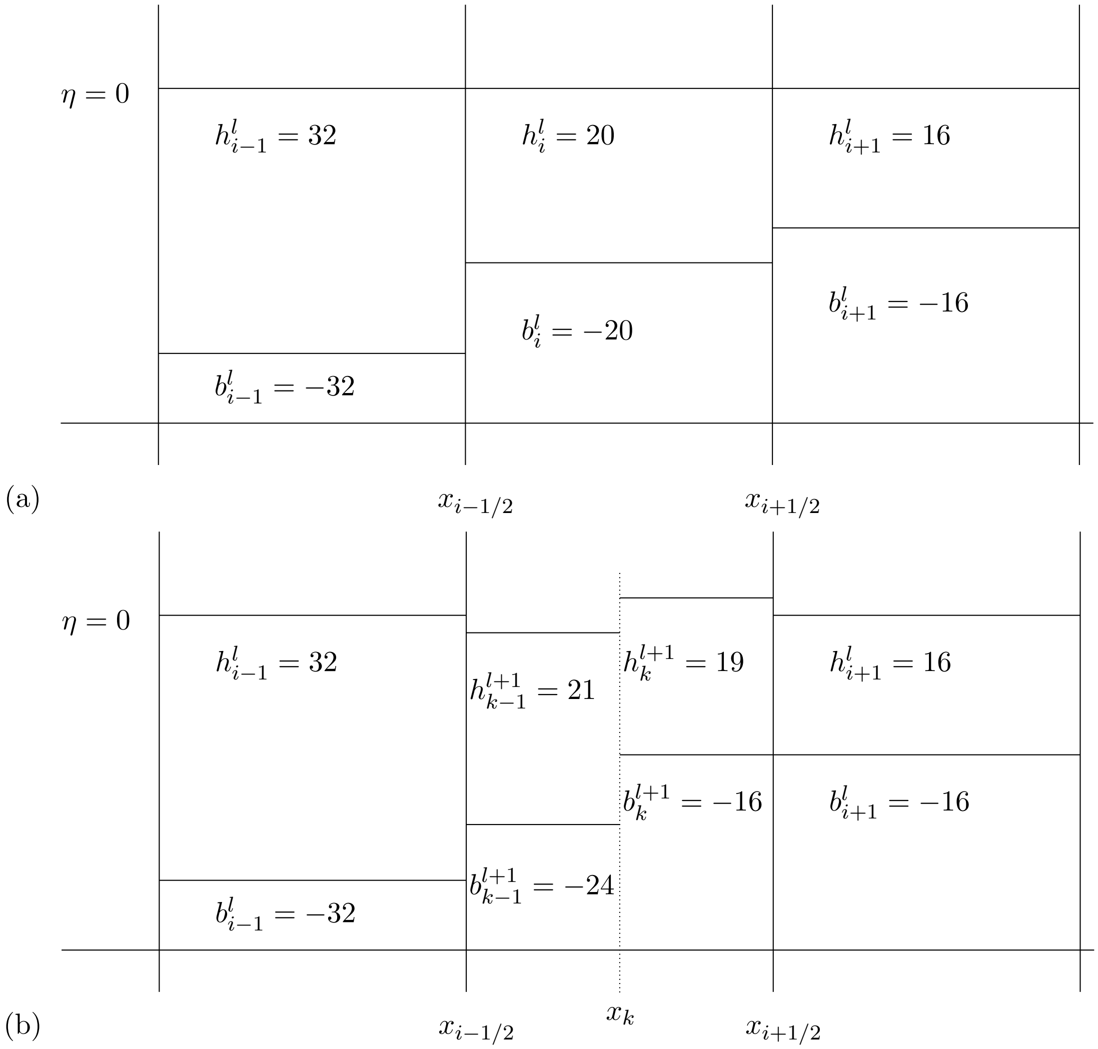


Figure 7.3: Interpolating the water depth from coarse grid cells to fine grid cells over nonlinear variable bathymetry. (a) A level l grid in one dimension for the common steady state problem of a motionless body of water with a flat surface $\eta = 0$ over nonlinear variable bathymetry. The numerical bathymetry is an average of the true bathymetry values in each grid cell. (b) Interpolation of h to a level $(l + 1)$ grid—refined by a factor of two—showing the two fine grid cells in the coarse cell $\mathcal{C}^l_i = [x_{i-1/2}, x_{i+1/2}]$. The bathymetry in the two fine grid cells reflects the average of the true bathymetry in those cells. Interpolating the depth h using minmod slopes, destroys the steady state.

7.3 Shorelines and Shallow Regions

7.3.1 Depth

In the previous section it was shown that by interpolating the surface values η to determine h, steady states and conservation can be simultaneously maintained when refining grids. It was assumed that each of the coarse grid cells surrounding $\mathcal{C}_{ij}^l$ had a positive depth, yet, of course, in very shallow shoreline regions it's possible that some of the coarse cells are dry. Since the surface elevation is still defined in these cells by $\eta = b$, it's possible to interpolate as before using (7.7). This is not too problematic if the cell being interpolated is wet, $h_{ij}^l > 0$, and one or more of the surrounding cells are dry, however, it is still less than ideal. If $h_{ij}^l = 0$ simply interpolating η as before, with $\eta_{ij}^l = b_{ij}^l$, will cause spurious waves to be generated from the shore and must be avoided.

To illustrate this situation, consider again a simple one dimensional example where a coarse grid must be interpolated to determine two fine grid cell values, h_{k-1}^{l+1} and h_k^{l+1}, in each coarse cell $[x_{i-1/2}, x_{i+1/2}]$. Suppose first that $h_i^l = 0$ and the cell to the left is wet with water at rest (see Figure 7.4). This reflects the steady state situation that is encountered at the shoreline if no waves are present. When the grid is refined, the bathymetry will be adjusted to values that represent the average of the bathymetry in each fine cell. It is clear by looking at Figure 7.4 what the surface elevations and depths should be in the two fine grid cells if the steady state is to be maintained—the surface elevation should be constant across the interface $x_{i-1/2}$, implying that $\eta_{k-1}^{l+1} = 0$ or equivalently $h_{k-1}^{l+1} = -b_{k-1}^{l+1} = 1$, and the depth h_k^{l+1} should be zero. (Any other values would generate waves at the interfaces $x_{i-1/2}$ and $x_{k-1/2}$, and the steady state would not be maintained.) However, if the surface elevation η was interpolated according to (7.7) using the minmod slope from the left, $\sigma_i^x = 2/\Delta x$, the surface elevations would be $\eta_{k-1}^{l+1} = 1.5$ and $\eta_k^{l+1} = 2.5$. This would result in depths $h_{k-1}^{l+1} = -0.5$ and $h_k^{l+1} = 0.5$, which conserves mass, but obviously is not the desired result.

At first glance, it might seem that an easy fix to the shoreline situation is to simply define $\eta = 0$ in dry cells, however, a near motionless steady state can exist without being at sea level $\eta = 0$. For instance, when very long waves approach shore the sea level may rise or fall substantially in certain areas without a pronounced *wave*. Arriving waves might then be viewed as being deviations to this near steady state that is not actually at sea level. Defining $\eta_i^l = 0$ in a dry cell $\mathcal{C}_i^l$ at the shore, with no regard to the level of water in $\mathcal{C}_{i-1}^l$, could generate spurious waves by introducing a large discontinuity at $x_{i-1/2}$ upon interpolation.

A preferable strategy for defining η in dry cells that are being interpolated, is to use the surrounding wet cells. For example, if $h_{ij}^l = 0$, then the surface value η_{ij}^l in cell $\mathcal{C}_{ij}^l$ can be defined by

$$\eta_{ij}^l = \frac{1}{|\Omega_{ij}^l|} \sum_{(m,n)\in\Omega_{ij}^l} \eta_{i+m,j+n}^l, \tag{7.12}$$

where

$$\Omega_{ij}^l = \{(m,n) : m,n \in \{-1,0,1\}, h_{i+m,j+n} > 0\}, \tag{7.13}$$

and $|\Omega_{ij}^l|$ is the number of elements of Ω_{ij}^l—*i.e.* the number of wet cells in the cells surrounding $\mathcal{C}_{ij}$. When interpolating $\mathcal{C}^l_{ij}$, before determining slopes η should be defined in the same way for any of the surrounding dry cells

$$\eta^l_{i+\bar{m},j+\bar{n}} = \frac{1}{|\Omega^l_{ij}|} \sum_{(m,n)\in\Omega^l_{ij}} \eta^l_{i+m,j+n}, \tag{7.14}$$

where $\bar{m}, \bar{n} \in \{-1, 0, 1\}$ and $(\bar{m}, \bar{n}) \notin \Omega^l_{ij}$. Note that the surrounding dry cells are only defined by (7.14) for interpolating $\mathcal{C}^l_{ij}$. Interpolation can now be based on (7.7), yet using (7.12) and (7.14) for the surface elevation in cells that are dry. Additionally, when using the interpolated surface values to determine the depths, the depths should be simply held nonnegative

$$h^{l+1}_{ks} = \max\left(0, \eta^{l+1}_{ks} - b^{l+1}_{ks}\right). \tag{7.15}$$

By using this strategy several nice properties are realized. First, a motionless steady state at a shoreline is easily maintained. Second, the generation of new extrema in the water surface elevation is prevented. (This is because the average values used to define the surface elevation in dry cells always lie between the extremes in the wet cells. Note that if η was defined to be 0 or b in dry cells, then new extrema in wet cells could be generated since 0 or b could be an extreme value used in the interpolation.) Third, even when the shoreline is not at a steady state, discontinuities at the cell interfaces are made as small as possible since the dry cells have surface elevations based on their neighbors.

A similar modification must be made when fine grids are coarsened. (Coarsening was not mentioned in Section 7.2, since, averaging the depths or the surface elevations in fine cells produced the same result. This is because of conservation—(7.11)—and the fact that all of the cells had positive depth by assumption.) Consider again Figure 7.4. Suppose that the fine grid shown in Figure 7.4 (b) existed first and needed to be coarsened to determine the value h^l_i in Figure 7.4 (a). If the surface values $\eta^{l+1}_{k-1} = 0$ and $\eta^{l+1}_k = 5$ were averaged to determine the surface value $\eta^l_i = 2.5$, the resulting depth would be $h^l_i = 0.5$. This would generate a spurious wave at the shoreline. Averaging the fine cell depths would produce the same result. A rather obvious solution is to simply average the surface values in the wet cells,

$$\eta^l_i = \frac{1}{|\tilde{\Gamma}^{l+1}_{ij}|} \sum_{(k,s)\in\tilde{\Gamma}^{l+1}_{ij}} \eta^{l+1}_{ks}, \tag{7.16}$$

where $\tilde{\Gamma}^{l+1}_{ij} = \left\{(k,s) : \mathcal{C}^{l+1}_{ks} \subset \mathcal{C}^l_{ij}, h^{l+1}_{ks} > 0\right\}$, and $|\tilde{\Gamma}^{l+1}_{ij}|$ is the number of wet cells. Then the nonnegative depth can be determined by

$$h^l_i = \max\left(0, \eta^l_i - b^l_i\right). \tag{7.17}$$

It should be noted that mass is not actually conserved at the shoreline when grids are interpolated and averaged in this way. Consider Figure 7.4 where clearly mass is gained when the dry cell h^l_i is refined producing a wet cell h^{l+1}_{k-1}. However, this should be considered differently than a violation of conservation of mass in the deep water. In the former, violation of conservation would be due simply to a nonconservative interpolation strategy. In the latter, the violation is due to an actual refinement and more accurate representation of the shoreline. Consider a steady state problem for instance where a flat pool of water lies over variable bathymetry. Keeping the surface elevation of the water fixed and changing grid resolutions clearly changes the amount of mass present, since the area of the region under water changes. Figure 7.5 shows the same set of grids from Figure 7.1, being used to cover Sumatra. The dry cells over land have been erased showing only the wet cells. In more refined regions the shoreline is more accurately resolved, but the mass must clearly change as the shoreline is more accurately resolved. It should be pointed out that for the steady state problem, repeated interpolation and coarsening of the grid would have no net effect on mass.

7.3.2 Momentum

As might be expected, momentum must be treated carefully as well in shoreline regions. In Section 7.2 we showed that in order to conserve momentum we must interpolate momentum directly not velocity. However, in shoreline regions it is possible to end up with vanishing or zero depths in fine cells. If momentum is interpolated independently, this can lead to nonphysically large velocities in these cells since velocities are determined by dividing the momentum by the depth. (Note that this problem is independent of the issues described in the preceding discussion—even in regions that have positive depths in fine and coarse cells, nonlinear bathymetry can lead to vanishing depths independent of linearly interpolated momentum leading to unbounded velocities.) This must be avoided since the velocities are used in several formulas including the momentum flux.

Suppose that the coarse cell $\mathcal{C}_{ij}^{l}$ must be interpolated to determine values for the momentum hu in the fine cells $\mathcal{C}_{ks}^{l+1} \subset \mathcal{C}_{ij}^{l}$. Recall that the minmod slopes used for interpolation in Section 7.2 were employed to prevent extrema in the fine cells $\mathcal{C}_{ks}^{l+1} \subset \mathcal{C}_{ij}^{l}$, from surpassing the bounds of the neighboring coarse cells. Based on this idea, it is possible to limit velocity when interpolating while still actually interpolating momentum. This requires determining bounds for the $(l+1)$ velocities when interpolating,

$$\overline{u}_{ij}^{l} = \max_{(m,n)\in\mathcal{S}} \left(u_{(i+m,j+n)}\right), \tag{7.18a}$$

$$\underline{u}_{ij}^{l} = \min_{(m,n)\in\mathcal{S}} \left(u_{(i+m,j+n)}\right), \tag{7.18b}$$

where, $\mathcal{S} = \{\{-1,0,1\} \times \{-1,0,1\}\}$. (By considering the corner cells $\mathcal{C}_{i,j\pm1}^{l}$ and $\mathcal{C}_{i\pm1,j}$ the bounds are less restrictive and still prevent new local extrema in u). The momentum can then be interpolated by (7.4) as always, followed by a check of the new velocities $hu_{ks}^{l+1}/h_{ks}^{l+1}$ in all of the fine cells $\mathcal{C}_{ks}^{l+1} \subset \mathcal{C}_{ij}^{l}$. If any of these violate the bounds (7.18), then all of the fine cell momenta are defined by

$$hu_{ks}^{l+1} = h_{ks}^{l+1} u_{ij}^{l}, \qquad \forall (k,s) \in \Gamma_{ij}^{l+1}. \tag{7.19}$$

The fix (7.19) prevents new extrema in u, while still maintaining conservation if all of the fine cells are wet since,

$$\frac{1}{|\Gamma_{ij}^{l+1}|}\sum_{(k,s)\in\Gamma_{ij}^{l+1}} h_{ks}^{l+1}u_{ij}^{l} = u_{ij}^{l}\frac{1}{|\Gamma_{ij}^{l+1}|}\sum_{(k,s)\in\Gamma_{ij}^{l+1}} h_{ks}^{l+1} = u_{ij}^{l}h_{ij}^{l}, \tag{7.20}$$

where the last equality is due to conservation of mass which occurs when all of the fine cells are wet and $h_{ij}^{l} > 0$. Therefore, if momentum is gained or lost by employing (7.20), it must occur because mass is gained or lost when interpolating based on the procedures discussed in Section 7.3.1, and the momentum change is directly proportional to the mass change

$$\frac{\frac{1}{|\Gamma_{ij}^{l+1}|}\sum_{(k,s)\in\Gamma_{ij}^{l+1}} h_{ks}^{l+1}u_{ij}^{l}}{hu_{l}^{ij}} = \frac{\frac{1}{|\Gamma_{ij}^{l+1}|}\sum_{(k,s)\in\Gamma_{ij}^{l+1}} h_{ks}^{l+1}}{h_{l}^{ij}}. \tag{7.21}$$

Therefore, if momentum is gained or lost at the shoreline it is due directly to the gain or loss in mass. Trying to conserve momentum when mass is gained or lost at the shoreline would not be a stable approach.

The problem of unbounded velocities can also occur when coarsening a fine grid if some of the fine grid cells are dry. This occurs because of a loss of mass. Consider again Figure 7.4, assuming now that the shoreline is not at the steady state, but has momentum. Suppose that the fine cells in Figure 7.4 (b) must be averaged to determine hu_{i}^{l} in the coarse cell in Figure 7.4 (a). Suppose that hu_{k-1}^{l} is nonzero. Then averaging the momenta in the two fine cells would result in a coarse cell momentum of $hu_{i}^{l} = hu_{k-1}^{l+1}/2$. Obviously this is a problem since coarsening the depth based on (7.16) and (7.17) resulted in the value $h_{i}^{l} = 0$. An approach similar to that described above for interpolating momentum can be taken, where the momentum is not conserved but is related to the change in mass. The averaged momentum is multiplied by the ratio of mass in the fine and coarse cells,

$$hu_{ij}^{l} = \left(\frac{h_{i}^{l}}{\frac{1}{|\Gamma_{ij}^{l+1}|}\sum_{(k,s)\in\Gamma_{ij}^{l+1}} h_{ks}^{l+1}}\right)\frac{1}{|\Gamma_{ij}^{l+1}|}\sum_{(k,s)\in\Gamma_{ij}^{l+1}} hu_{ks}^{l+1}. \tag{7.22}$$

This prevents unbounded velocities that occur when the coarsened depth is far less than the average depth in the fine cells. Note that the ratio in (7.22) is always 1, unless dry fine cells exist.

Obviously the formulas and procedures discussed in this section apply to hv and v as well as hu and u. It should also be noted that the interpolation and averaging formulas presented in this section can be applied everywhere on the domain, since they reduce to the formulas in Section 7.2 when away from the shoreline.

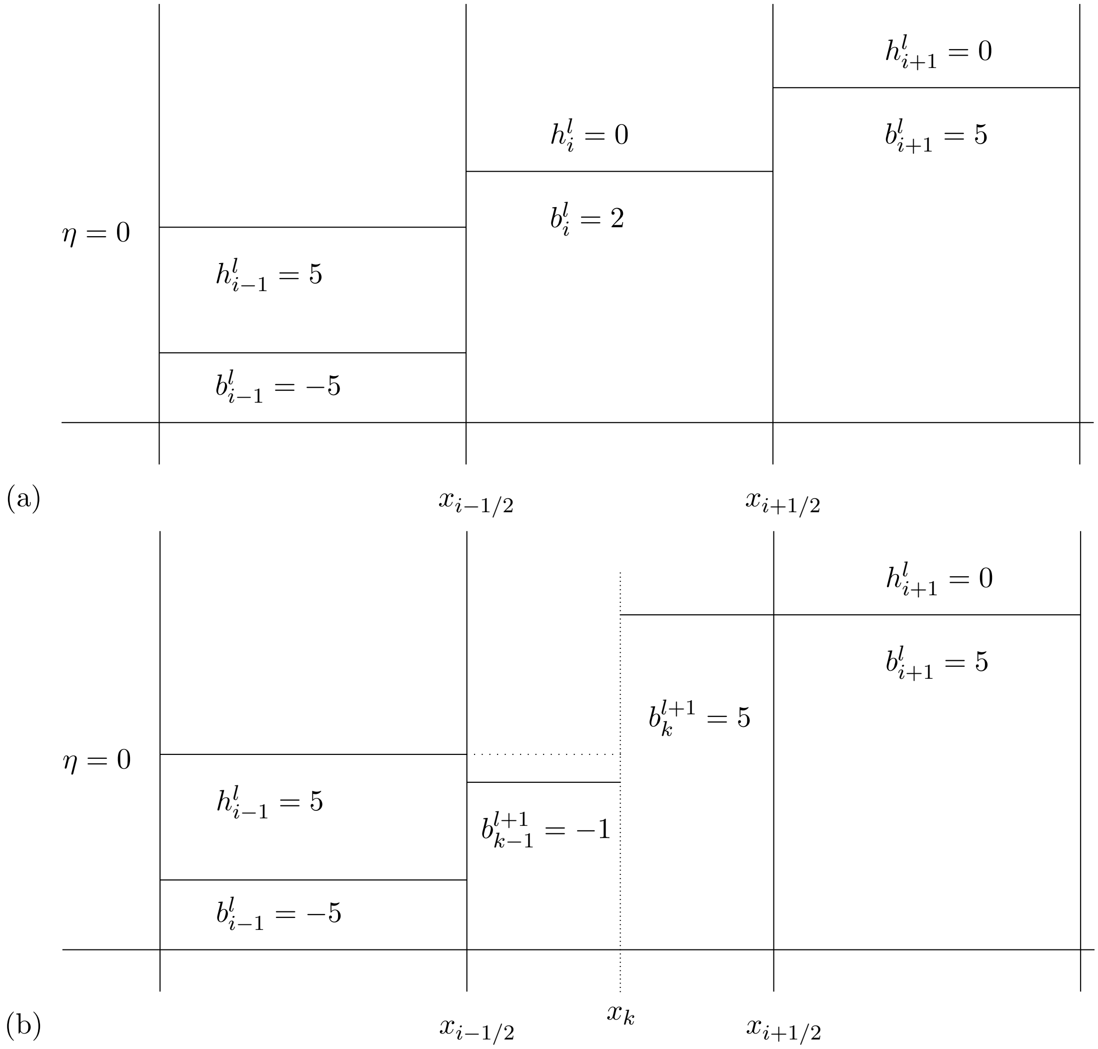


Figure 7.4: Interpolation at the shoreline. (a) Coarse level l grid at the shoreline where h^l_i=0 and the cell to the left is wet with $\eta^k_{i-1} = 0$. This reflects the common steady state problem at the shoreline. (b) Interpolation to fine grid cells within $[x_{i-1/2}, x_{i+1/2}]$. Note that to maintain the steady state the surface elevation should be constant across the interface $x_{i-1/2}$, implying that $h^{l+1}_{k-1} = 1$ and $h^{l+1}_k = 0$.

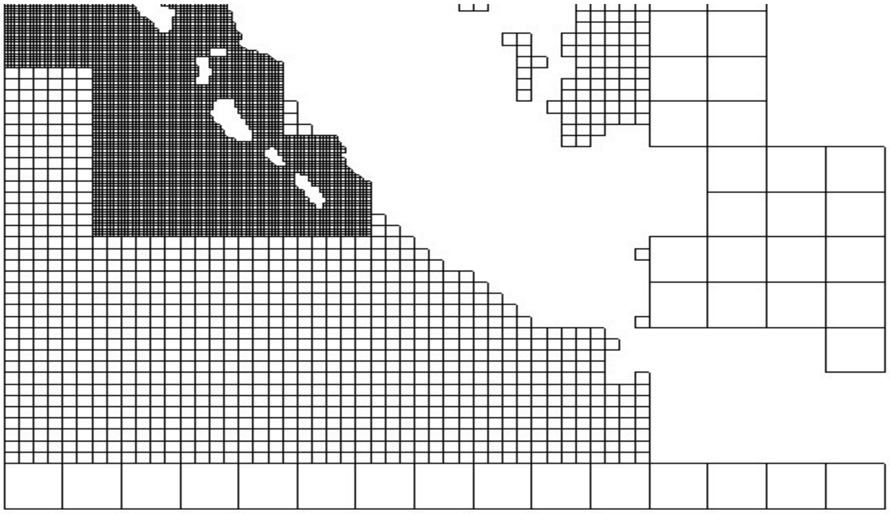


Figure 7.5: Adaptive mesh refinement around a shoreline. The grids are the same set of grids shown in Figure 7.1, covering Sumatra and some surrounding islands. The dry cells are erased to emphasize the shoreline, and show how refinement around the shoreline must necessarily change the overall mass if the true bathymetry is used and the steady state is to be maintained.

7.4 Time Refinement

The adaptive refinement methods discussed in this chapter typically employ refinement in time that is equal to refinement in space. That is, if a coarse level l grid is refined by a factor of n to a finer level $(l+1)$ grid, then n timesteps are taken on the fine grid for every timestep taken on the coarse grid. This is typically required for stability, assuming that the wave speeds on the fine grid and the coarser grids are similar. The CFL condition requires $s_x\Delta t < \Delta x$ and $s_y\Delta t < \Delta y$ for every grid cell, where s_x and s_y are the maximum wave speeds in the x and y directions respectively. If Δx or Δy are reduced by a factor of n, then Δt must be reduced as well. For gas dynamics—the application that these methods were originally developed for—this is not problematic since typically the wave speeds are as large in areas with the highest refinement—the areas with high pressure and shocks.

With tsunami modeling an interesting peculiarity arises—the wave speeds $u \pm \sqrt{gh}$ decrease with depth, and shallow regions are typically where the most refinement is needed. This means that temporal refinement equal to spatial refinement may be unnecessarily restrictive. Suppose that level l grids are restricted to shallow regions to refine inundating waves, and the deep ocean is covered by grids of levels $1, \cdots, l-1$. If level l grids are refined by a factor of n in space, timesteps reduced by a factor of n will typically have very small Courant numbers since the wave speeds in these shallow regions are much smaller than on level $l-1$ grids and lower. It might be possible to take far fewer timesteps on these finest grids, speeding the computation considerably. For instance, suppose that level l grids are restricted to regions less than 100 meters deep and refined by a factor of $n = 6$. A rough calculation (neglecting u) suggests that wave speeds will be faster by a factor of $\sqrt{4000/100} \approx 6$ on grid levels $l-1$ and less, since ocean depths can approach 4000 meters. This implies that timesteps on level l might approach that of time-steps on level $l-1$, however, the standard methods would take 6 steps on the level l grids for every step on level $l-1$. Of course, wave speeds are also related to water velocity, which can be quite large near the shoreline, so generally one must be more conservative than this simple calculation suggests, and we might refine by a factor of 2 at level l

All of the strategies discussed in this chapter have been successfully implemented in the CLAWPACK and AMRCLAW routines. We have employed the interpolation and averaging strategies discussed in Sections 7.2 and 7.3, and we have allowed for anisotropic time refinement, that allows specifying spatial and temporal refinement separately so that highly refined grids restricted to shallow regions can take less restrictive timesteps. These adaptive algorithms can also be implemented on more complex grids fitted to an idealized surface of the earth. The procedure of mapping rectangular Cartesian grids to more general geometries is discussed in the next chapter.

Chapter 8

MODELING TSUNAMIS ON THE EARTH

So far we have assumed that we are solving equations on a flat Cartesian grid in a rectangular domain, where the computational space is a direct discrete representation of the physical space. Of course, for global-scale tsunami modeling we are interested in solving the shallow water equations on the earth, or at least on the surface of an idealized sphere or ellipsoid. One approach to such a problem is to recast the shallow water PDEs in spherical coordinates, then use latitude and longitude as angular coordinates, however, this entails transforming derivatives—requiring assumptions about smoothness in the solution that we wish to avoid. It would be preferable to maintain the formalism that we have developed thus far, using the same general form of the equations and simply use a more complex grid that conforms to the physical domain. The integral conservation law can be applied to any grid cell and approximated numerically, so the basis of the finite volume approach is still valid on grid cells that are more complex than simple rectangles. This approach is generally applicable to conservation laws using quite general types of grids on two dimensional curved manifolds, and is described in detail in [67, 32, 85]. We will first consider this method on a flat domain, before considering more general curved earth-fitted grids.

8.1 General Quadrilateral Grids on a Flat Surface

We can use the wave propagation algorithm on a more general grid with cells other than uniform rectangles. We will consider only logically rectangular quadrilateral grids (ie. rectangular grids with cells in rows and columns that are labeled in a logical manner—cell (i, j) has four neighbors $(i \pm 1, j)$ and $(i, j \pm 1)$). The simplest implementation of the wave propagation algorithm on a nonuniform quadrilateral grid, is to use a uniform grid in *computational space*, and then incorporate an appropriate scaling on the numerical update that takes into account the true physical geometry of each computational cell. We will denote generalized coordinates in computational space by ξ and η. Figure 8.1 shows two such grid mappings for illustration. Both quadrilateral grids in physical space are mapped from a uniform rectangular grid in computational space shown on top. The grid on the bottom left is *orthogonal* whereas the grid on the bottom right is not.

For a finite volume method on a quadrilateral grid, the numerical solution Q_{ij} is an approximation to the average value of the solution in cell (i, j), regardless of the shape of the cell, and we can base a numerical method on the integral conservation law using interface fluxes just as in Chapter 3. Following the notation of LeVeque [67], such a method can be written as

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{|\mathcal{C}_{ij}|}\Big(h_{i+1/2,j}\breve{F}_{i+1/2,j} - h_{i-1/2,j}\breve{F}_{i-1/2,j} \\ +h_{i,j+1/2}\breve{G}_{i,j+1/2} - h_{i,j-1/2}\breve{G}_{i,j-1/2}\Big), \quad (8.1)$$

where

$$\begin{array}{rcl} |\mathcal{C}_{ij}| & = & \text{area of cell } (i,j), \\ h_{i\pm1/2,j} & = & \text{length of side between cells } (i\pm1,j) \text{ and } (i,j), \\ h_{i,j\pm1/2} & = & \text{length of side between cells } (i,j\pm1) \text{ and } (i,j), \\ \breve{F}_{i\pm1/2,j} & = & \text{flux normal to the edge between cells } (i,j) \text{ and } (i\pm1,j), \\ & & \text{per unit length per unit time} \\ \breve{G}_{i,j\pm1/2} & = & \text{flux normal to the between cells } (i,j) \text{ and } (i,j\pm1), \\ & & \text{per unit length per unit time.} \end{array}$$

For a uniform Cartesian grid in Euclidean space, the cell area $|\mathcal{C}_{ij}|$ is simply the product of the edge lengths $h_{i-1/2,j}$ and $h_{i,j-1/2}$, which are simply the increments of the computational coordinates Δx and Δy. Additionally the flux $\breve{F}_{i-1/2,j}$ is simply the flux in the x direction at the interface between cells $(i-1,j)$ and (i,j), denoted simply $F_{i-1/2,j}$, and similarly $\breve{G}_{i,j-1/2}$ is simply the flux in the y direction at the interface between cells (i,j) and $(i,j-1)$, denoted simply $G_{i,j-1/2}$. The more general numerical method (8.1) then reduces to the standard formula

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(G_{i+1/2,j} - G_{i-1/2,j}\right), \quad (8.2)$$

where the computational coordinates are simply x and y. For more general grids, we can write (8.1) in terms of computational coordinates following [67], if we set

$$\kappa_{ij} = \frac{|\mathcal{C}_{ij}|}{\Delta\xi\Delta\eta}, \quad (8.3a)$$

$$F_{i-1/2,j} = \left(\frac{h_{i-1/2,j}}{\Delta\eta}\right)\breve{F}_{i-1/2,j}, \quad (8.3b)$$

$$G_{i,j-1/2} = \left(\frac{h_{i,j-1/2}}{\Delta\xi}\right)\breve{G}_{i,j-1/2}, \quad (8.3c)$$

so that (8.1) becomes

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\kappa_{ij}\Delta\xi}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) - \frac{\Delta t}{\kappa_{ij}\Delta\eta}\left(G_{i+1/2,j} - G_{i-1/2,j}\right). \quad (8.4)$$

The formulas (8.3) relate the geometry of the computational grid to that of the physical grid. The area ratio κ_{ij} is sometimes called a *capacity function* since it relates the capacity or true physical area of a grid cell to the area of a cell in computational space. The length ratios that multiply the normal fluxes are *scale factors* and will be denoted

$$\gamma_{i-1/2,j} = h_{i-1/2,j}/\Delta\eta \quad (8.5a)$$

$$\gamma_{i,j-1/2} = h_{i,j-1/2}/\Delta\xi. \quad (8.5b)$$

The same approach can be extended to the more general wave propagation formula, which analogous to (8.4) becomes

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\kappa_{ij}\Delta\xi}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}\right) \\ - \frac{\Delta t}{\kappa_{ij}\Delta\eta}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}\right). \tag{8.6}$$

The fluctuations $\mathcal{A}^{\pm}\Delta Q$ and $\mathcal{B}^{\pm}\Delta Q$ are due to moving waves that carry the same dimensions as flux, and like the scaled fluxes $F_{i\pm1/2,j}$ and $G_{i,j\pm1/2}$ in (8.4), these waves should be properly scaled by the length ratios (8.5). This can be done by simply multiplying each of the f-waves by the appropriate scale factor when determining the fluctuations

$$\mathcal{A}^{-}\Delta Q_{i-1/2,j} = \sum_{s^p_{i-1/2,j}<0} \gamma_{i-1/2,j}\mathcal{Z}^p_{i-1/2,j} \tag{8.7a}$$

$$\mathcal{A}^{+}\Delta Q_{i-1/2,j} = \sum_{s^p_{i-1/2,j}>0} \gamma_{i-1/2,j}\mathcal{Z}^p_{i-1/2,j}, \tag{8.7b}$$

and similarly for $\mathcal{B}^{-}\Delta Q_{i,j-1/2}$ and $\mathcal{B}^{-}\Delta Q_{i,j-1/2}$ using $\gamma_{i,j-1/2}$. Since the waves are typically determined by solving normal Riemann problems at grid cell interfaces, any necessary rotation of the solution data or fluxes is required before determining the waves.

Second-order correction terms and transverse terms are modified by scale factors in a similar way, allowing the algorithm to retain its same essential form as on uniform Cartesian grids. See [67] for details.

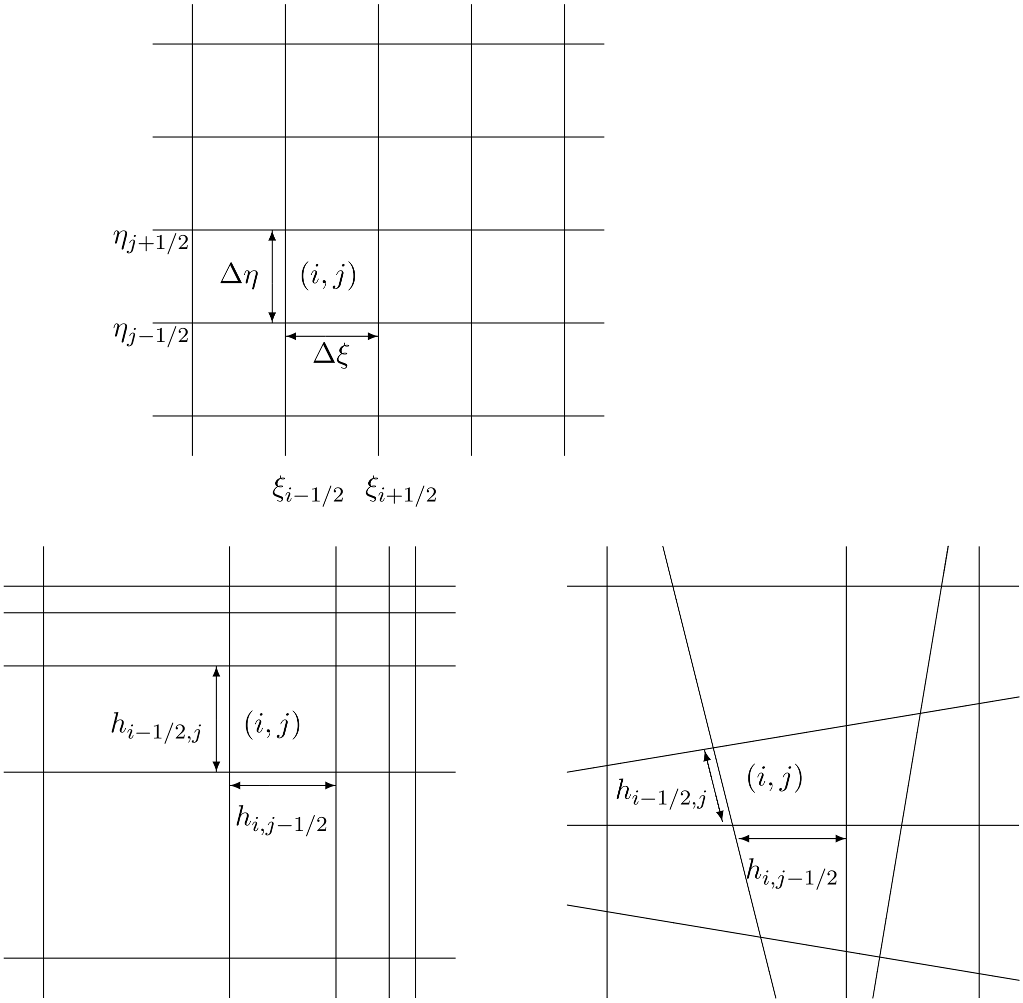


Figure 8.1: Logically rectangular quadrilateral grids. Top: a uniform rectangular grid in computational space. Bottom Left: a nonuniform orthogonal quadrilateral grid. Bottom Right: a nonuniform non-orthogonal quadrilateral grid.

8.2 The Shallow Water Equations on Latitude-Longitude Grids

Next we will consider the shallow water equations on the surface of a sphere or more general ellipsoid that approximates the surface of the earth. Recall that the shallow water equations can be derived by conservation and a hydrostatic assumption, meaning that the pressure gradient balances the gravitational force on a parcel of water at all times in these flows. Therefore, the equivalent assumption on an ellipsoid is that the gravitational force is balanced by hydrostatic pressure such that momentum is never in the direction normal to the surface and that pressure depends only on depth. Given the velocity of geophysical water flows like tsunamis, and the curvature of the earth, the validity of these assumptions is unchanged by the curved geometry. For higher velocity shallow flows on a more curved surface, where the pressure may depend appreciably on the velocity of the water column, a different set of equations may be necessary.

The same basic approach described in Section 8.1 can be used to solve conservation laws on smooth curved manifolds, assuming the conservation law is valid on the manifold. See [85] for a general treatment. For solving the shallow water equations on the surface of an idealized earth, the procedure is simplified from the general procedure for several reasons. First, the surface of an ellipsoid is easily parameterized by latitude and longitude coordinates, which means that we can use latitude and longitude to define a uniform quadrilateral grid in computational space that is logically rectangular. Second, latitude and longitude coordinates on an ellipsoid define locally orthogonal directions (ie. at any given point on the surface of the ellipsoid, infinitesimal increases in the two coordinate directions are orthogonal in the larger Euclidean space). This allows physical quantities in a grid cell to continue to be represented in a locally orthogonal basis, while simultaneously being aligned with the computational grid. This implies that Riemann problems and transverse corrections can be applied in the standard way without rotation of the data. Lastly, the pole singularity is avoided since the ice-covered poles do not need to be included in the domain. Figure 8.2 shows a portion of a uniform computational grid that is mapped to the surface of a reference ellipsoid for the earth.

For the special case of latitude-longitude grids, we will denote the computational coordinates by λ for latitude and θ for longitude, and to simplify the formulas we will assume that λ and θ are in radians. Since the fluxes are everywhere tangent to the ellipsoid surface, the relevant lengths, $h_{i\pm1/2,j}$ and $h_{i,j\pm1/2}$, are the arc lengths of the curved cell edges and the relevant area of the cell $|\mathcal{C}_{ij}|$ is the surface area of a curved surface in physical space. If we consider Euclidean space with the origin at the center of the ellipsoid, and represent the surface in Euclidean space by the parameterized vector, $\vec{S}(\theta,\lambda)$, the area of the cell (i,j) is

$$|\mathcal{C}_{ij}| = \int_{\lambda_{j-1/2}}^{\lambda_{j+1/2}} \int_{\theta_{i-1/2}}^{\theta_{i+1/2}} \left|\frac{\partial \vec{S}}{\partial \theta}\right| \left|\frac{\partial \vec{S}}{\partial \lambda}\right| d\theta d\lambda. \tag{8.8}$$

Similarly the cell interface lengths are

$$h_{i-1/2,j} = \int_{\lambda_{j-1/2}}^{\lambda_{j+1/2}} \left|\frac{\partial \vec{S}}{\partial \lambda}\right| d\lambda, \tag{8.9}$$

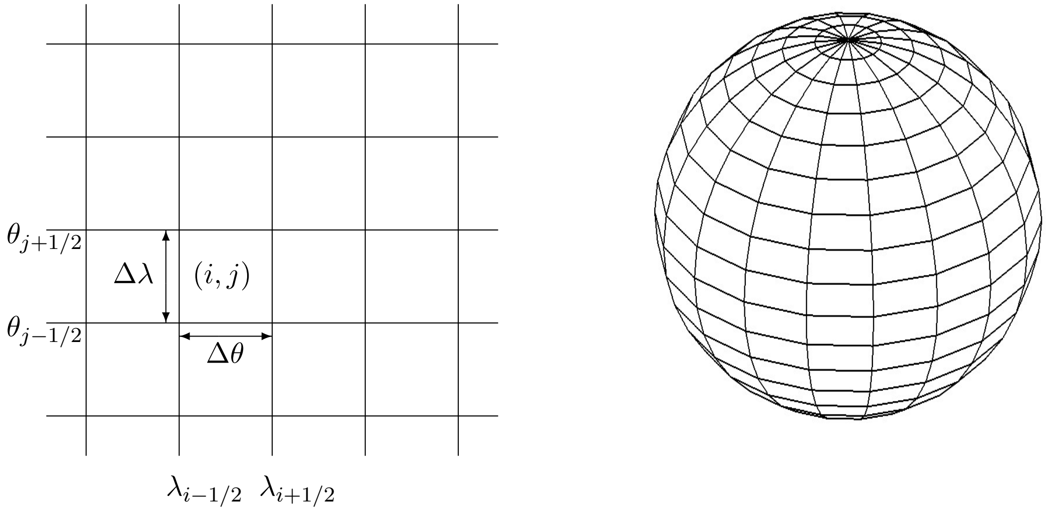


Figure 8.2: Earth-fitted grids: mapping of latitude λ and longitude θ coordinates in uniform computational space (left) to a curved grid on the surface of a reference ellipsoid (right). Coordinate Mapping is locally orthogonal.

and

$$h_{i,j-1/2} = \int_{\theta_{i-1/2}}^{\theta_{i+1/2}} \left| \frac{\partial \vec{S}}{\partial \theta} \right| d\theta. \tag{8.10}$$

For general ellipsoids the integrals (8.8)–(8.10) may need to be approximated.

8.2.1 A Perfect Sphere

For a sphere of radius R, using the standard parameterization

$$\vec{S}(\theta,\lambda) = (x,y,z) = (R\cos(\lambda)\cos(\theta), R\cos(\lambda)\sin(\theta), R\sin(\lambda)) \tag{8.11}$$

in (8.8)–(8.10) gives

$$\begin{aligned} |\mathcal{C}_{ij}| &= R^2 \left(\sin(\lambda_{j+1/2}) - \sin(\lambda_{j-1/2})\right) \Delta\theta, && \text{(8.12a)} \\ h_{i-1/2,j} &= R\Delta\lambda, && \text{(8.12b)} \\ h_{i,j-1/2} &= R\cos(\lambda_{i,j-1/2})\Delta\theta. && \text{(8.12c)} \end{aligned}$$

The capacity function and length scale ratios are then

$$\begin{aligned} \kappa_{ij} &= R^2 \big(\sin(\lambda_{j+1/2}) - \sin(\lambda_{j-1/2})\big)/\Delta\lambda, && \text{(8.13a)} \\ \gamma_{i-1/2,j} &= R, && \text{(8.13b)} \\ \gamma_{i,j-1/2} &= R\cos(\lambda_{i,j-1/2}). && \text{(8.13c)} \end{aligned}$$

Note that the length scale ratios relate the circumference of a great circle and a constant latitude circle to angular coordinates λ and θ respectively. Note also that as grid spacing decreases to zero

$$\kappa_{ij} = R^2 \frac{\left(\sin(\lambda_{j+1/2}) - \sin(\lambda_{j-1/2})\right)}{\Delta\lambda} \to R^2 \cos(\lambda_j), \tag{8.14}$$

resulting in the ratio relating the differential surface area of a sphere to latitude.

8.2.2 Geoids and Reference Ellipsoids

The true surface of the earth is of course not a perfect sphere, but is closely approximated by a *geoid*—an equipotential surface of the earth's own gravitational field (adjusted for rotation) that most closely approximates the mean sea level. A geoid is a complex surface that depends on inhomogeneities in the earth's gravitational field, so it is often described by its departure from a closely approximating *reference ellipsoid*. Reference ellipsoids are simple oblate spheroids (a flattened sphere) with a polar radius R_b and a single equatorial radius R_a. They are commonly described by a single equatorial radius and a *flattening ratio*

$$f = \frac{R_a - R_b}{R_a}, \tag{8.15}$$

that relates the two radii. (The reference ellipsoid in common use today—the GRS80—has an equatorial radius of 6,378,137 m and a flattening ratio of 1/298.25722.) Latitude-longitude coordinates are still locally orthogonal on an oblate spheroid, but now we must be precise about what latitude refers to—it is no longer the simple angular coordinate between the equatorial plane and the vertical axis. The *geographic latitude*, λ, of a given point on an oblate spheroid is the angle between the normal direction and the equatorial plane, which, depending on the flattening, is different than the angle between the point and the center of the spheroid—sometimes called the *geocentric latitude*, which we will denote by ϕ (see Figure 8.3). (Note: the difference is significant even for the nearly spherical earth—a geocentric latitude of 45° corresponds to a geographic latitude of 44.8°, a difference of more than 20 km). Since bathymetry and other data is in terms of geographic latitude (due to historical navigational methods), we will use the true geographic latitude λ as our computational coordinate. For an oblate spheroid longitude is unambiguous since there is a single equatorial radius.

An oblate spheroid can be parameterized in a variety of ways—for geophysical problems it is common to use latitude-dependent polar and equatorial radii, $N_b(\lambda)$ and $N_a(\lambda)$, which allows the surface to be parameterized like a spheroid with latitude and longitude playing the role of angular coordinates. However, this is unnecessarily burdensome when evaluating the integrals (8.8)–(8.10) for the scale factors, which are all we need for the numerical method. To determine these scale factors, we will use a more standard parameterization of an ellipsoid, then relate this to latitude and longitude coordinates. To do this we must introduce a third angular coordinate ψ, such that

$$\vec{S}(\theta, \psi) = (x, y, z) = (R_a \cos(\psi)\cos(\theta), R_a \cos(\psi)\sin(\theta), R_b \sin(\psi)) \tag{8.16}$$

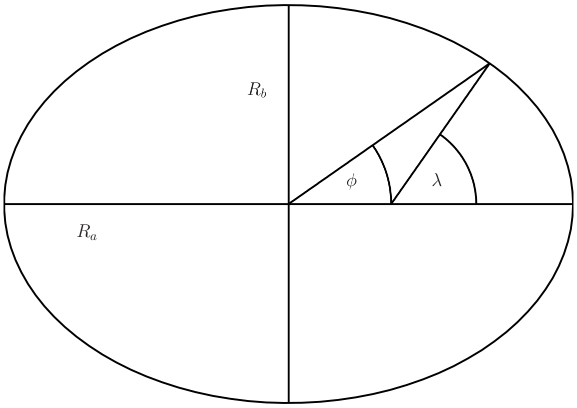


Figure 8.3: Geographic latitude on an oblate ellipse. The cross section of an oblate ellipse with a single equatorial radius R_a and smaller polar radius R_b is shown. The geographic latitude λ of a point is the angle between the normal direction and the equatorial plane, not the standard angular coordinate ϕ.

defines the surface. Note that ψ is neither the geocentric latitude ϕ nor the geographic latitude λ, but the three are related by the equation

$$\tan(\lambda) = \frac{1}{\sqrt{1-e^2}}\tan(\psi) = \frac{1}{(1-e^2)}\tan(\phi), \tag{8.17}$$

where e is the *ellipticity* defined by

$$e^2 = \frac{R_a^2 - R_b^2}{R_a^2}. \tag{8.18}$$

Using (8.16) in the integral (8.8) gives the surface area of a grid cell $\mathcal{C}_{ij}$

$$\begin{aligned}
|\mathcal{C}_{ij}| &= \int_{\psi(\lambda_{j-1/2})}^{\psi(\lambda_{j+1/2})} \int_{\theta_{i-1/2}}^{\theta_{i+1/2}} \left|\frac{\partial \vec{S}}{\partial \theta}\right| \left|\frac{\partial \vec{S}}{\partial \psi}\right| d\theta d\psi \\
&= R_a^2 \Delta\theta \int_{\psi(\lambda_{j-1/2})}^{\psi(\lambda_{j+1/2})} \cos(\psi)\sqrt{1-e^2\cos^2(\psi)} d\psi \\
&\approx R_a^2 \Delta\theta \int_{\psi(\lambda_{j-1/2})}^{\psi(\lambda_{j+1/2})} \cos(\psi)\left(1 - \frac{1}{2}e^2\cos^2(\psi)\right) d\psi \\
&= R_a^2 \Delta\theta \left[\sin(\psi)\left(1 - \frac{1}{3}e^2 - \frac{1}{6}e^2\cos^2(\psi)\right)\right]_{\psi=\psi(\lambda_{j-1/2})}^{\psi=\psi(\lambda_{j+1/2})}.
\end{aligned} \tag{8.19}$$

Although an exact closed form solution for (8.19) is possible, given the small value of the ellipticity of the earth, the $\mathcal{O}(e^4)$ terms are neglected in the third integral of (8.19), resulting in a less cumbersome formula. Using the implicit definition of $\psi(\lambda)$ from (8.17) in (8.19) gives

$$|\mathcal{C}_{ij}| = \left[\Delta\theta R_a^2\sqrt{1-e^2}\left(\frac{\sin(\lambda)}{\sqrt{1-e^2\sin^2(\lambda)}}\right)\left(1-\frac{1}{3}e^2-\frac{1}{6}\frac{e^2\cos^2(\lambda)}{\left(1-e^2\sin^2(\lambda)\right)}\right)\right]_{\lambda=\lambda_{i-1/2}}^{\lambda=\lambda_{i-1/2}}. \tag{8.20}$$

Similarly, using (8.16) and (8.17) to determine the length scale factors from (8.9) and (8.10) gives

$$h_{i,j-1/2} = \Delta\theta R_a \frac{\cos(\lambda_{i,j-1/2})}{\sqrt{1-e^2\sin^2(\lambda_{i,j-1/2})}} \tag{8.21a}$$

$$\begin{aligned} h_{i-1/2,j} &= R_a\int_{\psi(\lambda_{j-1/2})}^{\psi(\lambda_{j+1/2})}\sqrt{1-e^2\cos^2(\psi)}d\psi \\ &\approx R_a\int_{\psi(\lambda_{j-1/2})}^{\psi(\lambda_{j+1/2})}\left(1-\frac{1}{2}e^2\cos^2(\psi)\right)d\psi \\ &= R_a\left[\left(1-\frac{1}{4}e^2\right)\arctan(\sqrt{1-e^2}\tan(\lambda)) - \frac{1}{8}\frac{e^2\sqrt{1-e^2}\sin(\lambda)}{\left(1-e^2\sin^2(2\lambda)\right)}\right]_{\lambda=\lambda_{j-1/2}}^{\lambda=\lambda_{j+1/2}}. \end{aligned} \tag{8.21b}$$

We can now use (8.20) and (8.21) to determine the capacity function κ_{ij} and length scale ratios, $\gamma_{i-1/2,j}$ and $\gamma_{i,j-1/2}$, for (8.6) and (8.7).

Geoid estimates allow an even closer approximation to the true surface of the earth, and could be incorporated into the surface area of grid cells by using locally scaled radii R_a and R_b. However, this would have a negligible effect for tsunami modeling—geoid deviations from the reference ellipsoid are on the order of tens of meters and vary little over hundreds of kilometers. While this is important for applications that require the exact Euclidean position of a point, such as GPS navigation, it would have little effect on the surface area and length-scales used to scale computational grid cells.

8.3 Coriolis Forces

So far in this chapter we have dealt with effects due entirely to the geometry or *curvature* of the earth. In this section we deal with Coriolis forces that are due to the *rotation* of the earth. Several pseudo-forces arise when Newton's laws are written with respect to a rotating coordinate system. When a particle confined to the surface of the earth changes latitude its radius of rotation changes so it must accelerate relative to the rotating coordinate system, if its momentum is conserved. This apparent acceleration is due to the Coriolis force. This pseudo-force can act on a wave as well, if it carries momentum to different latitudes.

The Coriolis acceleration of a particle moving at velocity $\vec{\mathbf{u}}$

$$\frac{\partial}{\partial t}\vec{\mathbf{u}} = -2\vec{\omega} \times \vec{\mathbf{u}}, \tag{8.22}$$

where $\vec{\omega}$ is the angular velocity vector of the earth, and the derivative is with respect to the rotating frame of reference. Since we are solving the shallow water equations on a latitude longitude grid, our momentum variables are conveniently represented in the rotating local orthogonal coordinates in the directions of increasing longitude and latitude, not a rotating Euclidean set of axes. If we denote the time dependent unit vectors in the latitude and longitude directions $\hat{\mathbf{e}}_\lambda$ and $\hat{\mathbf{e}}_\theta$ respectively, then our momentum vector is $\vec{\mathbf{hu}} \equiv (hu\,\hat{\mathbf{e}}_\theta + hv\,\hat{\mathbf{e}}_\lambda)$. The rate change in momentum in the rotating coordinate system due to the Coriolis acceleration (8.22) is then

$$\begin{aligned} \frac{\partial}{\partial t}\vec{\mathbf{hu}} &= -2|\omega|\left(\cos(\lambda)\,\hat{\mathbf{e}}_\lambda + \sin(\lambda)\,\hat{\mathbf{e}}_r\right) \times \left(hu\,\hat{\mathbf{e}}_\theta + hv\,\hat{\mathbf{e}}_\lambda\right), \\ &= 2|\omega|\left(hv\sin(\lambda)\,\hat{\mathbf{e}}_\theta - hu\sin(\lambda)\,\hat{\mathbf{e}}_\lambda + hu\cos(\lambda)\,\hat{\mathbf{e}}_r\right). \end{aligned} \tag{8.23}$$

Neglecting the acceleration in the radial direction, (8.23) gives the Coriolis force in the tangent plane

$$\frac{\partial}{\partial t}\begin{bmatrix} hu \\ hv \end{bmatrix} = 2|\omega|\sin(\lambda)\begin{bmatrix} hv \\ -hu \end{bmatrix}, \tag{8.24}$$

where as always the first and second components of the vectors are in the locally orthogonal directions, denoted by $\hat{\mathbf{e}}_\theta$ and $\hat{\mathbf{e}}_\lambda$ respectively. Equation (8.24) makes it clear that the Coriolis force is essentially another source term, similar to variable bathymetry and friction. That is, (8.24) can be written

$$\frac{\partial}{\partial t}q = \psi_\omega, \tag{8.25}$$

where

$$\psi_\omega = 2|\omega|\sin(\lambda)\begin{bmatrix} 0 \\ hv \\ -hu \end{bmatrix}. \tag{8.26}$$

Since there is no relevant steady state due to the Coriolis force, it can be integrated by fractional stepping, discussed in Section 3.6.

8.4 Frictional Forces

The shallow water equations are derived for a nonviscous fluid with a uniform velocity profile in the vertical direction so that there is no boundary layer between the fluid and the underlying substrate. The validity of this approximation becomes more and more tenuous as the fluid depth decreases and the velocity of the fluid increases. These flow characteristics are exhibited in the extreme inundation regimes, motivating the use of empirically derived friction terms in the shallow water equations when tsunami modeling. These source terms are often based on experimental studies of wave runup given various types of substrates (gravel, sand, concrete etc. *e.g.* [17]). Friction source terms should therefore be used cautiously and possibly only for depths below some cutoff value—although they are often very small for regions with any appreciable depth.

The shallow water equations with friction become

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \tag{8.27a}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) + \frac{\partial}{\partial y}(huv) = -gh\frac{\partial b}{\partial x} - \tau_x, \tag{8.27b}$$

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}(\frac{1}{2}gh^2 + hv^2) = -gh\frac{\partial b}{\partial y} - \tau_y, \tag{8.27c}$$

where

$$\tau_x = \frac{gn^2}{h^{7/3}} hu\sqrt{(hu)^2 + (hv)^2}, \tag{8.28}$$

and

$$\tau_y = \frac{gn^2}{h^{7/3}} hv\sqrt{(hu)^2 + (hv)^2}, \tag{8.29}$$

where n is the Manning coefficient—an empirically determined constant. Typical values for the Manning coefficient range from $n = 0.013$ to $n = 0.025$ depending on the application and the substrate (*e.g.* [17]). Since no relevant steady state exists due to friction for tsunamis (the case is very different for river flows), the source term due to friction can be included by fractional stepping. This is the same as for Coriolis forces, discussed in Section 8.3. However, note that an implicit numerical method should be used since the terms can become quite large and time-steps are determined by the CFL condition from the underlying hyperbolic part of the problem.

Chapter 9

MISCELLANEOUS NUMERICAL RESULTS

In this chapter several test problems will be shown using the methods described throughout this thesis. In some cases the results will be compared to other modern Godunov-type methods for the shallow water equations. A description of the other methods is not provided here, but can be found in the cited references. All of the numerical results using the methods described in this thesis are implemented in TsunamiClaw , described in Appendix A, which is an extension of LeVeque's clawpack software [61].

9.1 Some One-Dimensional Test Problems

9.1.1 Transcritical Flow with a Shock

This one dimensional test was borrowed from [10], and was originally computed in [37] and later discussed in [26]. The domain is $x \in [0, 25]$, $t \in [0, 25]$, with initial data $h(x, 0) \equiv 0.33$ and $hu(x, 0) \equiv 0.18$. Dirichlet boundary conditions must be imposed with $hu(0, t) = 0.18$ and $h(25, t) = 0.33$. The bottom topography is given by

$$b(x) = \begin{cases} 0.2 - 0.05\,(x-10)^2 & \text{if} \quad 8 < x < 12, \\ 0 & \text{otherwise.} \end{cases} \tag{9.1}$$

The goal of the problem is to investigate convergence to a non-stationary steady state. The eventual solution contains a critical point and stationary shock. (Stationary and slow moving shocks are notorious for generating spurious oscillations, even for shock-capturing schemes [4, 51].) Bouchut computed this problem using several modern numerical schemes, as described in [10]. These solutions are shown in Figure 9.1 (a), which is borrowed directly from [10]. Note that all but one of the methods is free from spurious oscillations. Figure 9.1 (b) shows the numerical solution using the augmented Riemann solver and wave propagation methods developed and described in this thesis. All of the numerical solutions shown were computed with 200 points. Figure 9.2 shows a close up of the solution at $t = 200$ in the critical region. First and second order versions of the methods described in [10] as well as the first and second order methods described in this thesis are compared in Figure 9.2.

Figure 9.3 shows the momentum hu for the various schemes for this same test problem. Note that the steady state should have hu constant throughout the domain. Since the shock occurs in the middle of a grid cell, the numerical solution in the single grid cell overlapping the shock has a different larger value for hu than what the true solution has. The data in this grid cell should lie on (or close to) the Hugoniot locus connecting the data to the right and left of the grid cell. Shock data moving into the grid cell from either side preserve the steady state in this cell. Therefore, the spike seen at the location of this shock remains even as the grid is refined and the solution converges.

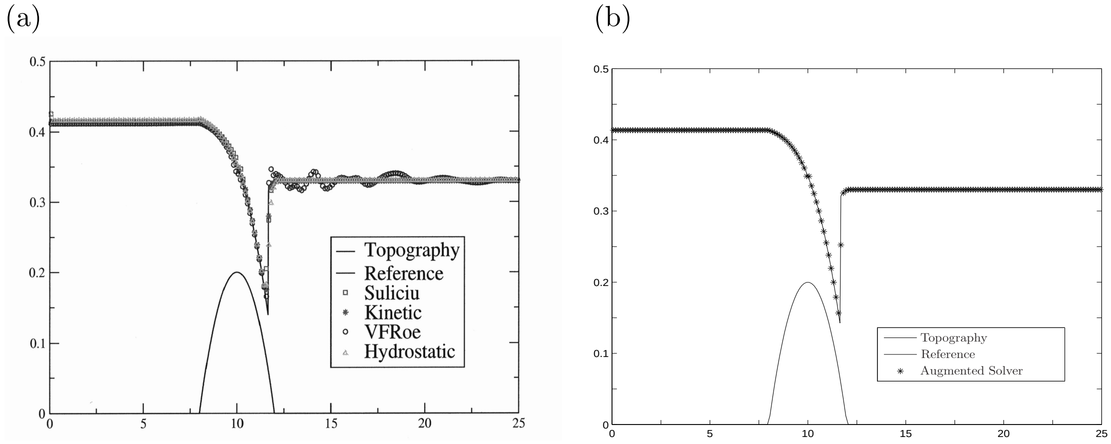


Figure 9.1: Transcritical flow with a shock. The water elevation is shown for the eventual steady state that is reached. (a) Various numerical solutions using methods described in [10] are shown. Figure is borrowed directly from Bouchut [10]. (b) The solution using the augmented approximate solver described in Chapter 6 and the wave propagation methods described in Chapter 3. Both figures show the solution at $t = 200$.

9.1.2 Vacuum Over Discontinuous Topography

This problem is again borrowed from Bouchut [10] and originated in [26]. The problem consists of a strong double rarefaction which leads to incipient cavitation over variable topography. The domain is again $x \in [0, 25]$ and the final time is $t = 0.25$. The topography is

$$b(x) = \begin{cases} 1 & \text{if} \quad 25/3 < x < 12.5, \\ 0 & \text{otherwise.} \end{cases} \tag{9.2}$$

The initial conditions are $h(x, 0) = 10$ and

$$hu(x, 0) = \begin{cases} -350 & \text{if} \quad x < 50/3, \\ 350 & \text{otherwise.} \end{cases} \tag{9.3}$$

Solutions using various methods are again borrowed from [10]. These are shown in Figure 9.4, and compared to solutions using the augmented solver and wave propagation methods in Figure 9.5. First and second order solutions at $t = 0.25$ are shown for all cases. A common property of many positive preserving methods near a dry front is illuminated in Figure 9.4. As described by Bouchut in [10], the "density is systematically underestimated on the front." Note that in Figure 9.4, especially for the second order schemes, the low depths preceding the actual dry front. This also occurs for the HLL method, and essentially results from the large speeds used that preserve positivity. See [10] for further discussion. Note that the augmented solver improves this weakness as can be seen in Figure 9.5.

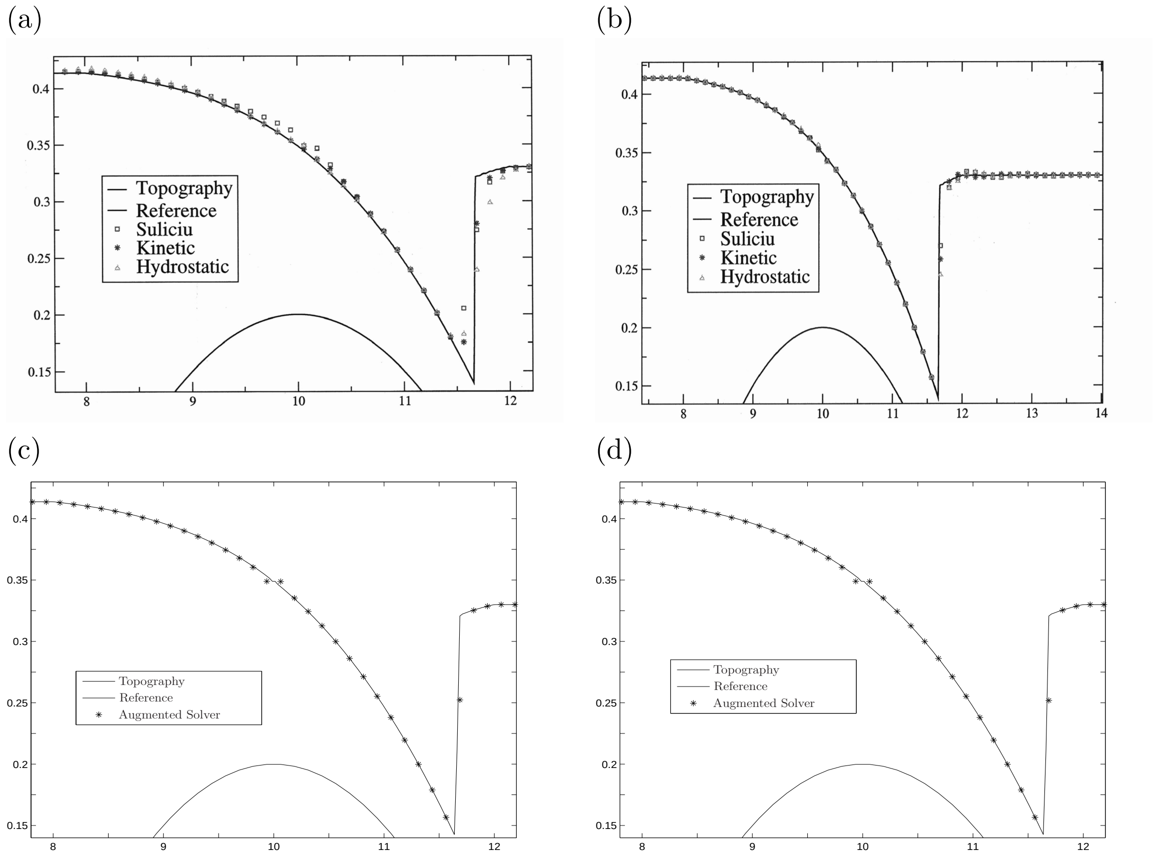


Figure 9.2: Close-up of the critical region for the transcritical flow problem. (a) The solutions for the first order schemes described in [10]. (b) Solutions to the same problem using the second order version of the same schemes. (c) The solution using the first order version of the methods described in this thesis. (d) Solutions using the second order corrections described in Chapter 3. (Panels (a) and (b) are borrowed from [10]).

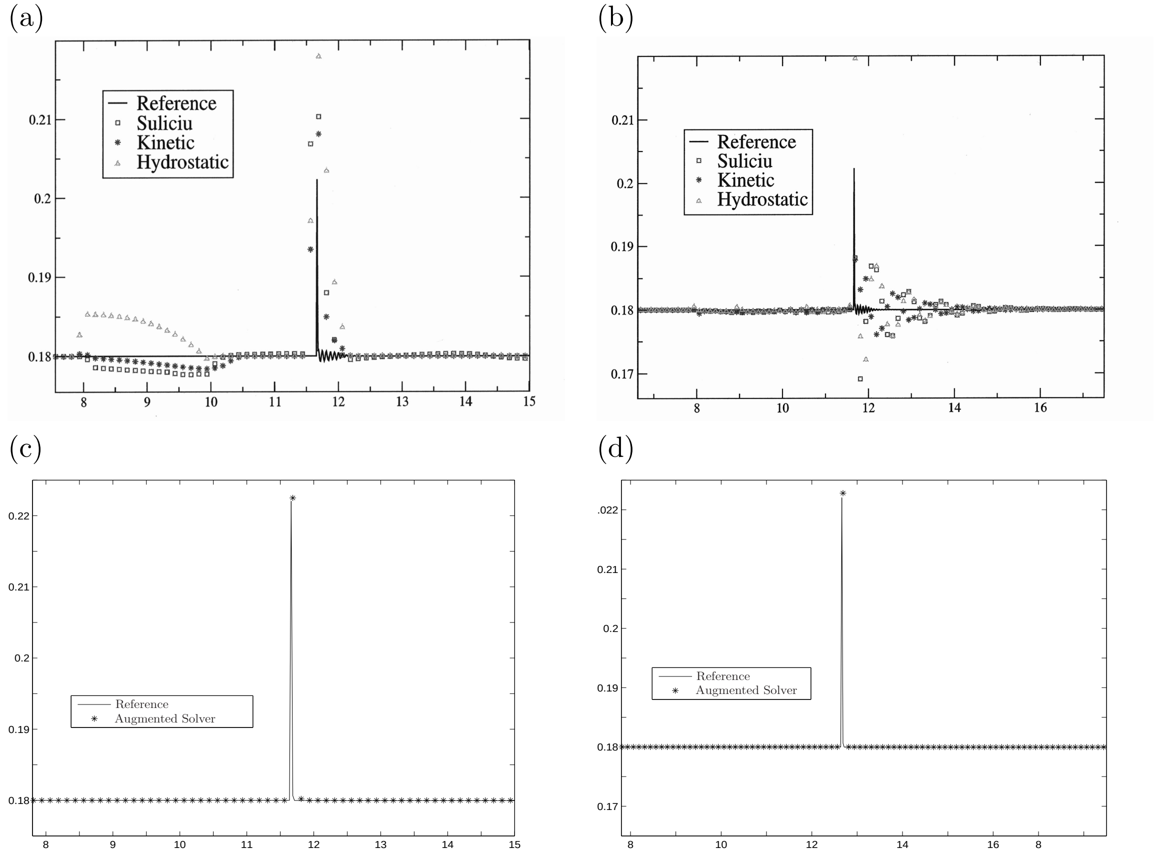


Figure 9.3: The momentum hu for the transcritical flow with a shock problem shown at $t = 200$. The true steady state solution should be constant $hu \equiv 0.18$. (a) The first order solution using the various schemes described in [10]. (b) The second order solutions to the same problem. (c) The solution using the augmented solver and the first order methods described in this thesis. (d) The same problem with second order corrections. The smooth steady state is nicely maintained. (Panels (a) and (b) are borrowed directly from [10]).

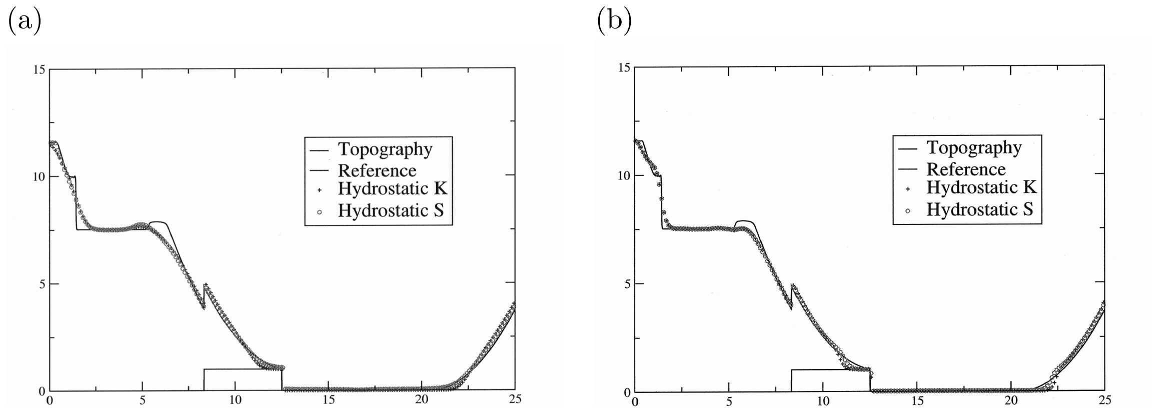


Figure 9.4: Various numerical solutions to the double rarefaction problem over discontinuous topography borrowed from [10]. (a) The first order methods. (b) The second order methods. Note the low depths preceding the true dry fronts on either side of the expanding vacuum region. Figures are borrowed directly from [10]

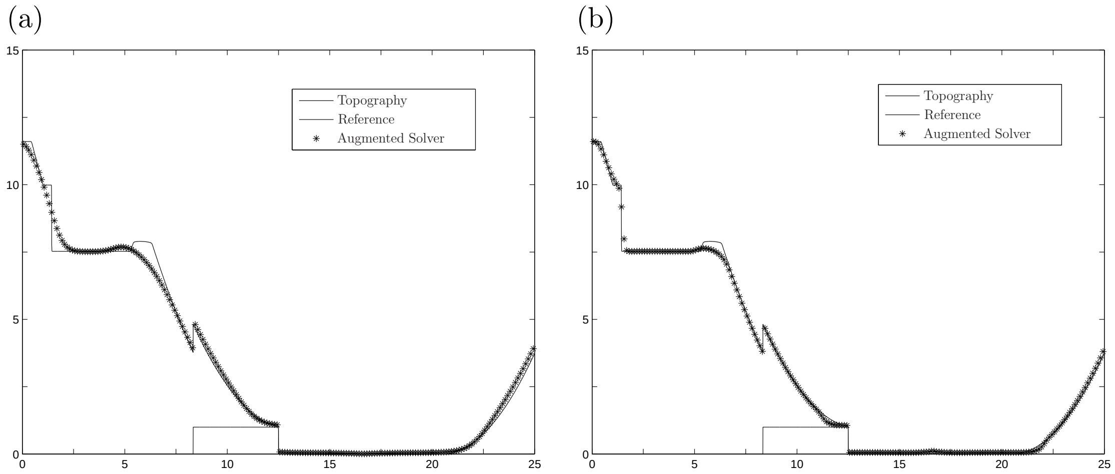


Figure 9.5: First and Second order solutions to the double rarefaction problem over discontinuous topography using the augmented Riemann solver and the wave propagation methods. (a) The first order method. (b) The same problem using the second order corrections. The dry front is more closely approximated.

9.2 Benchmark Problems from the 3rd International Longwave Workshop

This section describes problems from the 3^{rd} International Longwave Workshop, Catalina, June 2004. Some results from benchmark problems 1 and 2 are summarized. These results are additionally presented in our conference proceedings report [68].

9.2.1 Benchmark 1: Motion of the Shoreline

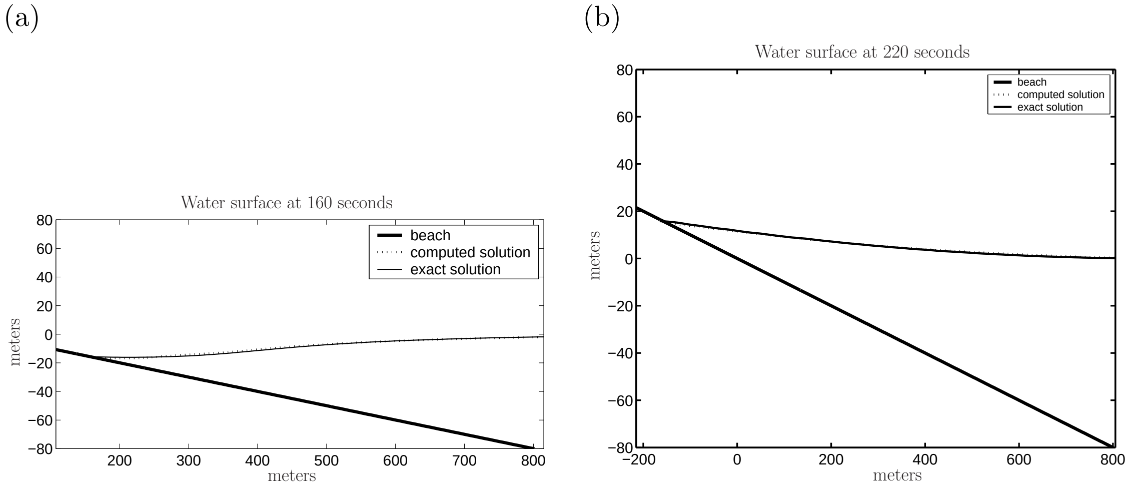


Figure 9.6: Benchmark 1 from the Long Wave Workshop [70]: The surface elevation and beach at two times. (a) Water surface elevation at $t = 160$ seconds. (b) Water surface elevation at $t = 220$ seconds.

The first benchmark problem, described in more detail in the conference proceedings [70], consists of an incoming wave from the right moving leftward toward a linear sloping beach on the left of the domain. The shoreline first recedes as the trough of the wave reaches the shore before advancing as the crest arrives. Figure 9.6 depicts the solution to the problem at two distinct times. One of the goals of the problem is to track the position and velocity of the single point where the water intersect the shore. This problem is solved analytically by Carrier and Yeh in [14], and serves as a validation tool for runup algorithms. The results of the shoreline movement, shown in Figure 9.7, are taken from our workshop results presented in [68]. The numerical solutions were therefore computed with a different Riemann solver, described in [68]. The solutions were computed with 1000 points on a much larger domain, with a mapping that clustered points near the shoreline. The area over which the shoreline moves, shown on the y–axis in Figure 9.7 (a), was covered with several hundred points.

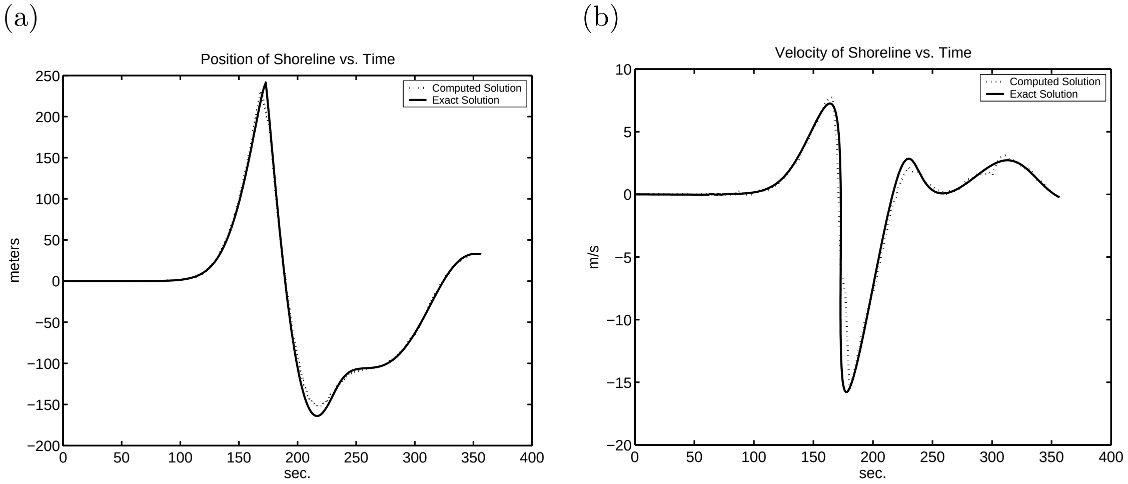


Figure 9.7: Benchmark 1: comparison of runup with an analytical solution from [14]. (a) Comparison of the numerical solution and the analytical solution for the position of the shoreline. (b) Comparison for the velocity of the shoreline.

9.2.2 Benchmark 2: Comparison to a Wave Tank Experiment

For the second benchmark problem, a scale model of Okushiri Island Japan was constructed in a wave tank basin. Okushiri Island was devastated by a tsunami in 1993, and suffered some unusually large runup patterns (*e.g.* [45]). An incoming wave was initiated by paddles at one end of the tank. One of the goals of the problem was to compare numerical solutions to the experimentally observed water surface elevations measured by wave gages at three distinct locations. Note that this is a test not only for the numerical method, but also for the shallow water equations in this context. The results are shown in Figure 9.8. More discussion and details can be found in our conference proceedings report [68], as well as comparisons to videos of inundation patterns.

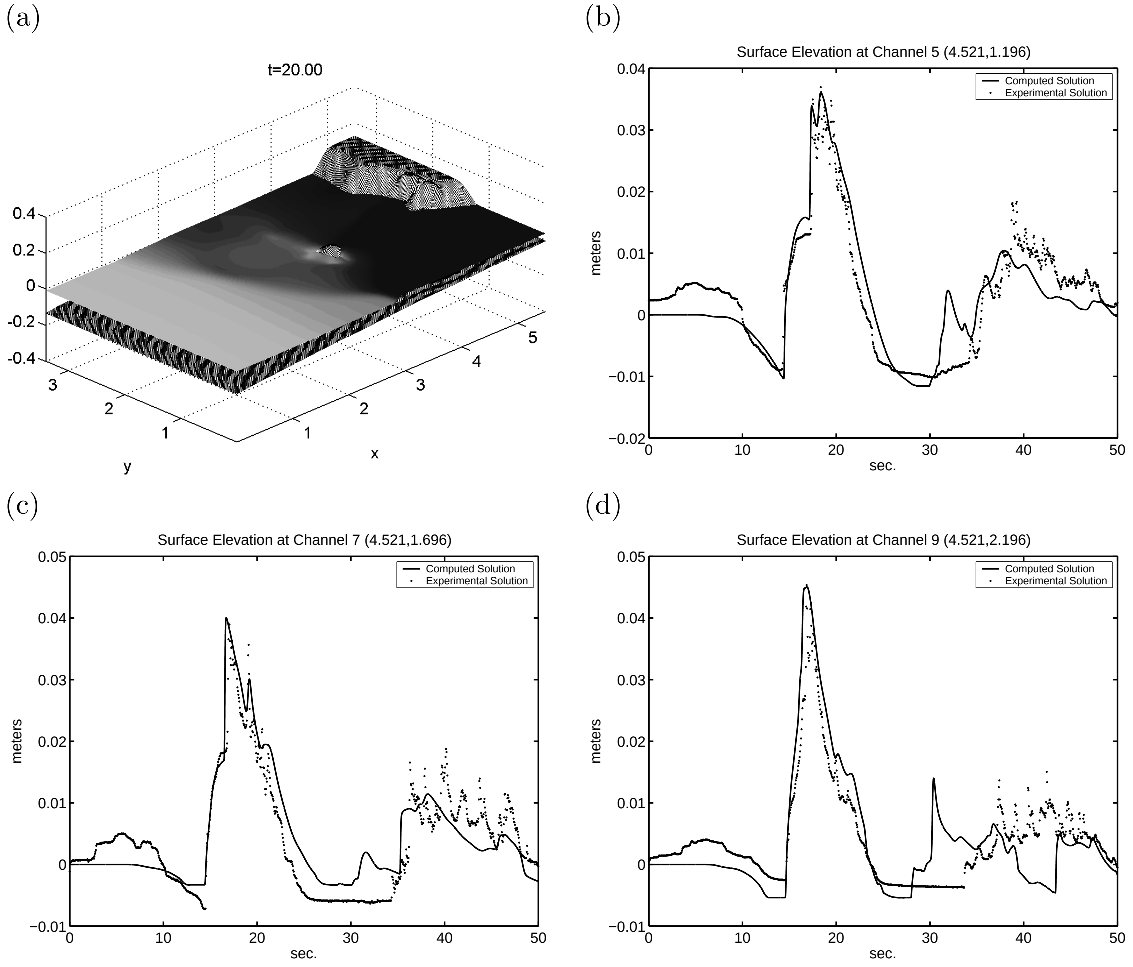


Figure 9.8: Benchmark 2 from the Long Wave Workshop [70]. Comparison to a wave tank experiment. (a) Snapshot from a simulation of the experiment showing Okushiri Island and the incoming wave. (b)-(d) Numerical and experimental solution at the location of 3 wave gages.

9.3 Adaptive Mesh Refinement for the Conical Island Test Problem

The problem in this section is taken from a wave tank experiment presented in Liu *et al* [72], where a solitary wave interacts with a circular or conical island model (see also). The problem was motivated largely by the unexpectedly large runup patterns observed on the lee side of islands during multiple tsunamis in the 1990s. See Yeh *et al* [110, 109, 112] and Titov *et al* [99, 96] for a discussion of tsunami runup patterns on islands. This section is also intended to introduce some features of using adaptive mesh refinement, described in Chapter 7.

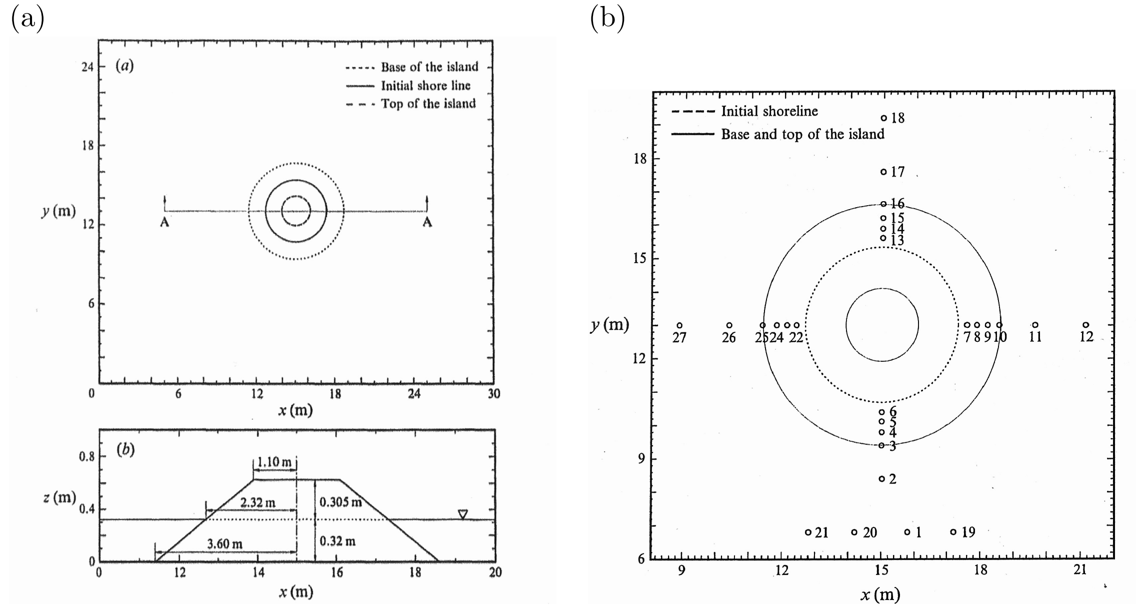


Figure 9.9: Diagram of the conical island wave tank experiment and the location of the gages, from Liu *et al* [72]. (a) Top: top view of the wave tank basin and island. Bottom: vertical view of the circular island on the cross section. (b) Location of the wave gages. The incoming soliton comes from $y = 0$ from the bottom of the figure. Figures are borrowed directly from [72].

Figure 9.9, borrowed directly from [72], shows the wave tank experiment set-up. The incoming wave was generated at the bottom of the tank $y = 0$, as described in [91]. TsunamiClaw was used for a numerical simulation of this experiment, using an undisturbed depth of 0.32 meters, and a soliton amplitude of 0.032 meters. The simulation was performed on a 50 meter long and 30 meter wide domain, in order to better propagate a soliton in the y direction, from $y < 0$. (Therefore, because of a discrepancy in reproducing the incoming wave that was actually generated in the wave tank experiment, the solution presented here will not be compared directly to the solutions in [72], but interesting features emerge.)

In the simulation shown in Figure 9.10, three grid levels were employed, with refinement ratios of 8. The coarsest grid had 1 meter grid cells. The second level tracks the incoming soliton, and third levels are reserved for capturing the runup at the island, as demonstrated in Figure 9.10. As the soliton wave wraps around the island, maximum runups are observed on the lee side of the island. In [72] this was experimentally reported, but the numerical results failed to reproduce that observed feature. Figure 9.11 shows the experimentally recorded water surface elevation at several gages in the experiment. The locations of the gages are shown in Figure 9.9. These figures are shown to demonstrate some general features in the recorded solution. They are not directly compared to the computational results of the simulation, since the incoming wave did not replicate the experimental wave exactly. Figure 9.12 shows the numerical solution of TsunamiClaw at locations corresponding to gages adjacent to the four sides of the island. Note that the solution at gage 13, located on the lee side of the island has the maximum runup. For the location of the gages, see Figure 9.9.

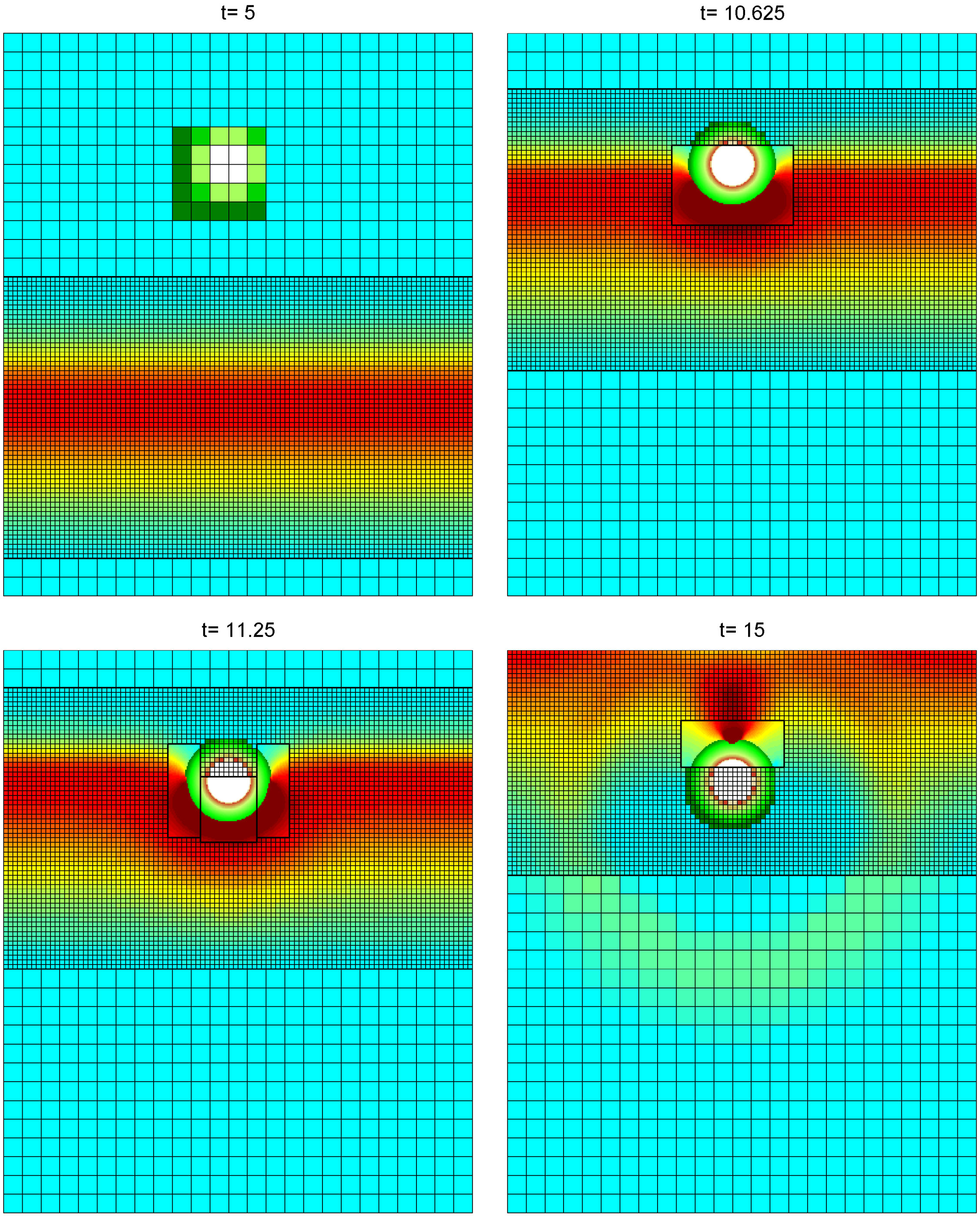


Figure 9.10: Simulation of the circular island experiment using adaptive mesh refinement. The incoming soliton is resolved on level 2 grids. As the soliton interacts with the island, level 3 grids resolve the runup.

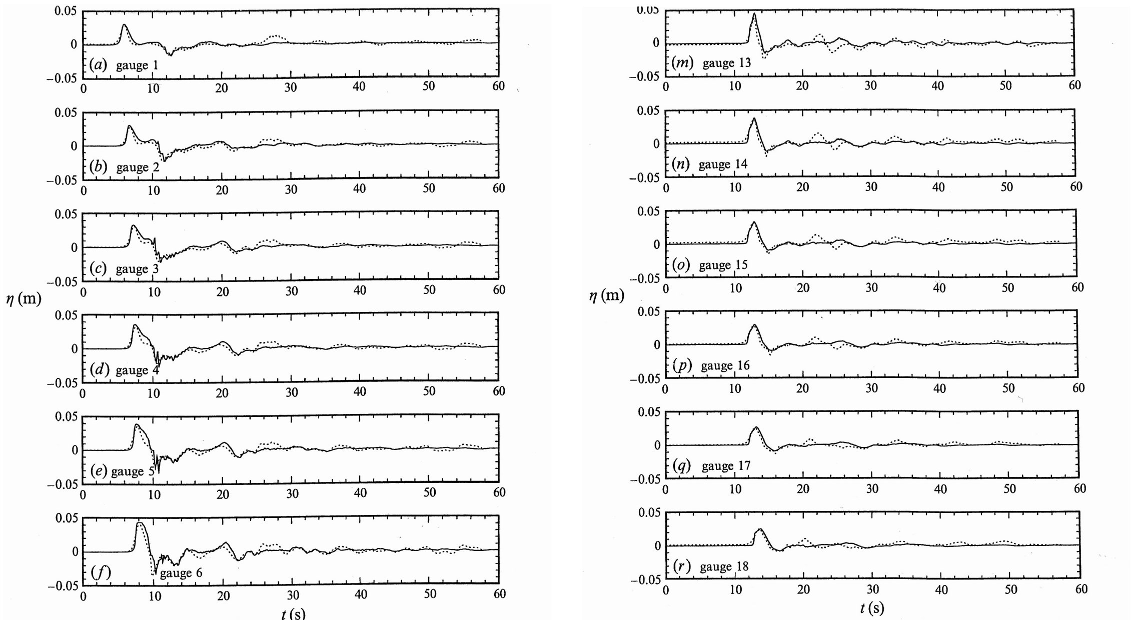


Figure 9.11: Surface elevation measured at some of the gages from the conical island experiment. Note that the surface elevation at gage 13, behind the island reaches a higher maximum that the elevation at gage 6, in front of the island. The figures are shown to indicate general patterns only since the incoming wave used in the computation described in this section does not replicate the experimental wave exactly. Figures are borrowed directly from [72].

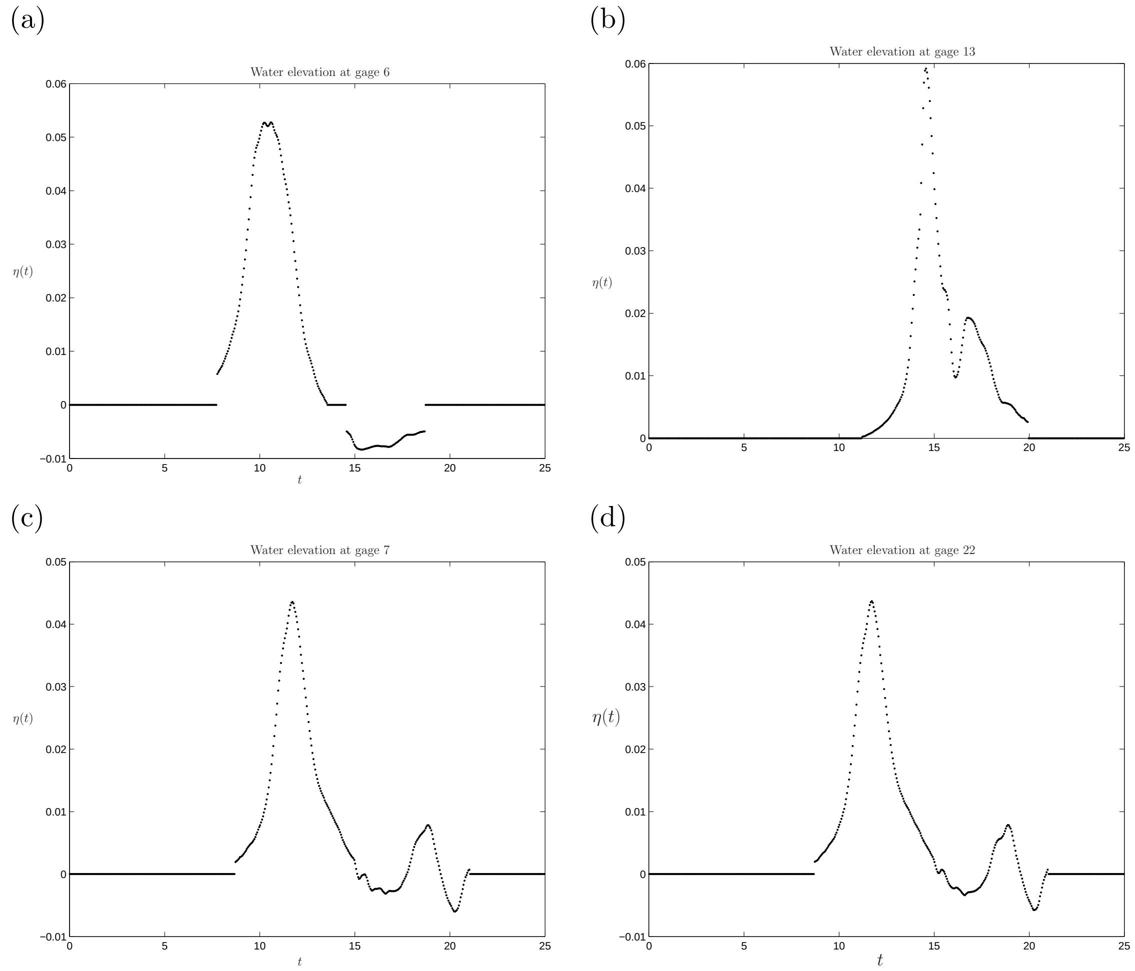


Figure 9.12: Runup at the four gages adjacent to each side of the circular island. See Figure 9.9 for the location of the gages. (a) Elevation at gage 6 on the near side of the island with respect to the incoming wave. (b) Elevation at gage 13 on the lee side of the island. Note that the runup at this gage is larger than at gage 6, as found in the wave tank experiments [72]. (c) & (d) Elevation at gages 7 and 22 on the two sides of the island. As expected from symmetry the solutions are identical.

Chapter 10

SIMULATIONS OF THE 2004 INDIAN OCEAN TSUNAMI

In this chapter several TsunamiClaw simulations of the 2004 Indian Ocean Tsunami will be shown. The goal of the chapter is to demonstrate the properties and features of adaptive mesh refinement in the context of a global scale transoceanic tsunami. In all of the simulations, dynamic motion of the seafloor bottom was used to generate the tsunami, as described in Section 10.1. The bathymetry in all of the simulations is from the GEBCO data set, a global one minute gridded data set. For higher grid refinements nautical chart interpolation was performed where possible. All of the simulations shown were computed on a latitude-longitude spherical grid as described in Chapter 8. Figure 10.1 shows a snapshot from a simulation on the Indian Ocean as the tsunami approaches Africa. The figure shows three levels of grid resolutions, with grid lines omitted from the finest level for clarity. The code for all of the simulations shown in this chapter is available at [28].

10.1 The 2004 Sumatra-Andaman Earthquake and Generation of the Tsunami

The 2004 Sumatra-Andaman Earthquake, one of the largest earthquakes in recorded history, initiated the Indian Ocean Tsunami—eventually killing hundreds of thousands of people throughout the Bay of Bengal and Indian Ocean region. The tsunami was the largest in modern history, and waves spanning the globe and were recorded on every continent (*e.g.* [95]). The fault rupture initiated on December 26th off the coast of Sumatra and proceeded north-northwest extending nearly 1300 kilometers along the Andaman islands trough at a speed of about 2.5 kilometers per second according to Ammon *et al* [3]. The peak vertical fault displacements are estimated to be approximately 15 meters by the same authors in [3]. Although these fault rupture speeds are fast compared to tsunami propagation (which can approach 0.2 kilometers per second), the long duration (about 500 seconds [43]) of the fault motion might significantly affect the tsunami wave patterns, as the tsunami might propagate upwards of 100 kilometers at the source before the rupture ends. This means that the inundation in the islands of Indonesia had begun before the generation of the tsunami 1300 kilometers to the north had ended.

10.1.1 Generation with a Spatial-Temporal Model of the Fault Motion

For the above reasons, the simulations shown in this chapter were generated using a dynamic fault model of the spatial-temporal movement of the seafloor. The model was provided by the Caltech Seismolab, and was generated using the spectral element code specfem3D, described in [54, 55]. The specific data used was Chen Ji's (Model III), described in [3]. The model divides the region surrounding the fault into a 15 minute grid cells and provides displacement of the seafloor at 0.1 second intervals for 790 seconds. The simulations shown

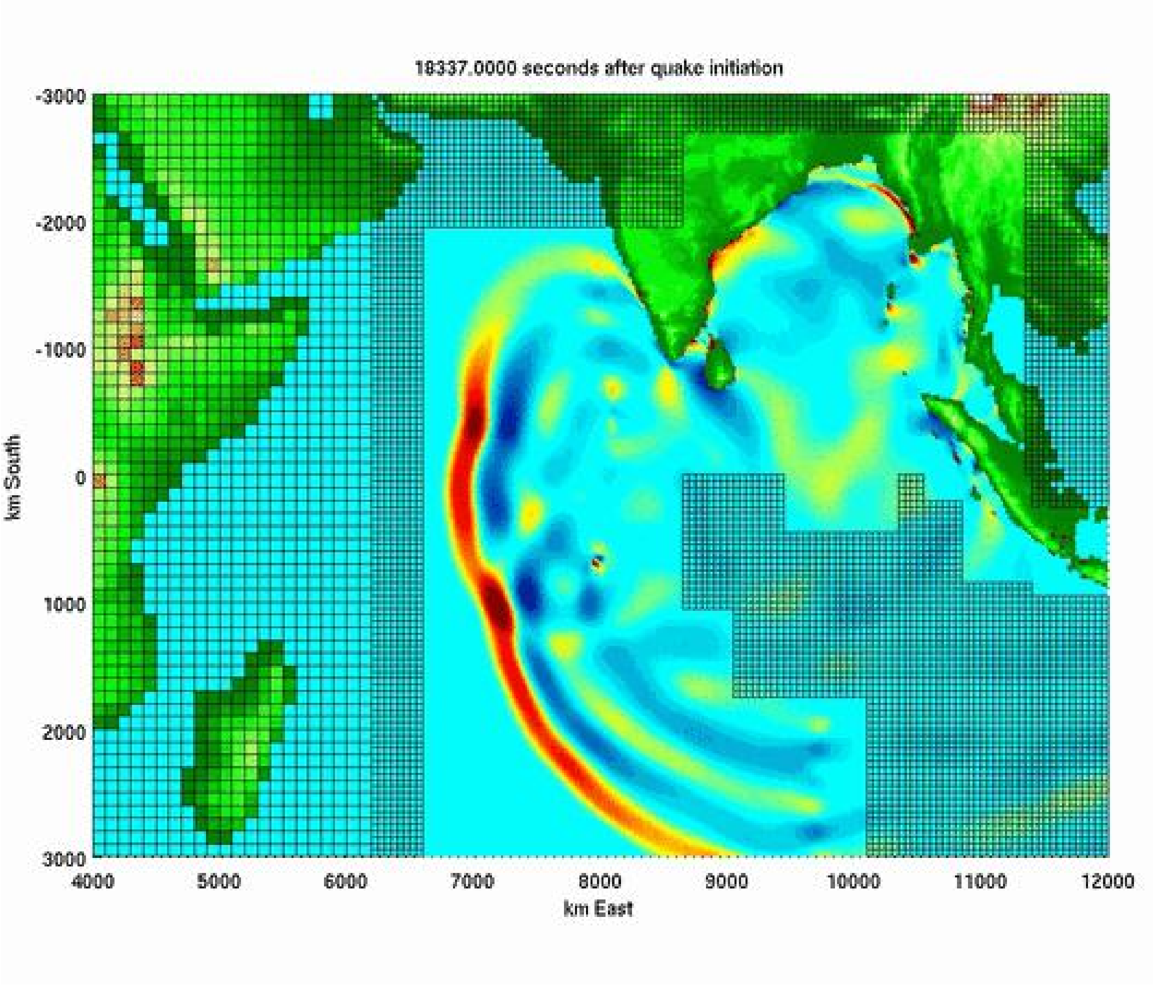


Figure 10.1: Simulation of the Indian Ocean Tsunami showing the entire Indian Ocean on three grid levels. Grid lines are omitted from the finest level.

in this chapter were all generated by numerically moving the bottom bathymetry according to the model, using time-steps dictated by stability in the TsunamiClaw code. The fault model was therefore interpolated spatially and temporally.

Figure 10.2 shows several snapshots around the fault region during the fault motion in a TsunamiClaw simulation. Clearly the tsunami begins to propagate away from the source as the fault continues to rupture northward.

10.1.2 Comparison to the Jason Satellite Data

The Jason-1 and Topex/Poseidon satellites, under a joint project between NASA and the French Space Agency, orbit the earth collecting satellite altimetry data for oceanographic research. The satellites happened to be passing over the Indian Ocean on December 26 2004. Researchers at the space agencies determined that the satellites apparently detected the tsunami. This was accomplished by comparing the satellite data on December 26th with that from the same orbital pass (pass 129) at different times (cycles). Figure 10.3 (a) is a plot of the Jason-1 sea surface height anomaly data (ssha) from cycle 108 in light blue, which was the previous time the satellites were on the same pass (about 10 days before the tsunami), and cycle 109 in violet which occurred during the tsunami. A crude estimate of the tsunami therefore can be obtained by subtracting the two data sets. Figure 10.3 (b) shows the subtracted data sets (in blue) and the computed solution (in red), using TsunamiClaw. The blue line is a polynomial fit to the subtracted data sets. Based on wave speed estimates and the time of the initial earthquake, it seems clear that the leading edge of the tsunami is at approximately $-5°$ latitude when it is encountered by the Jason-1 satellite. The apparent disturbance between $-15°$ and $-10°$ is most likely due to (unknown) factors other than the tsunami. This might give a rough indication of the inherent noise in the data and the process of subtracting the data at different times.

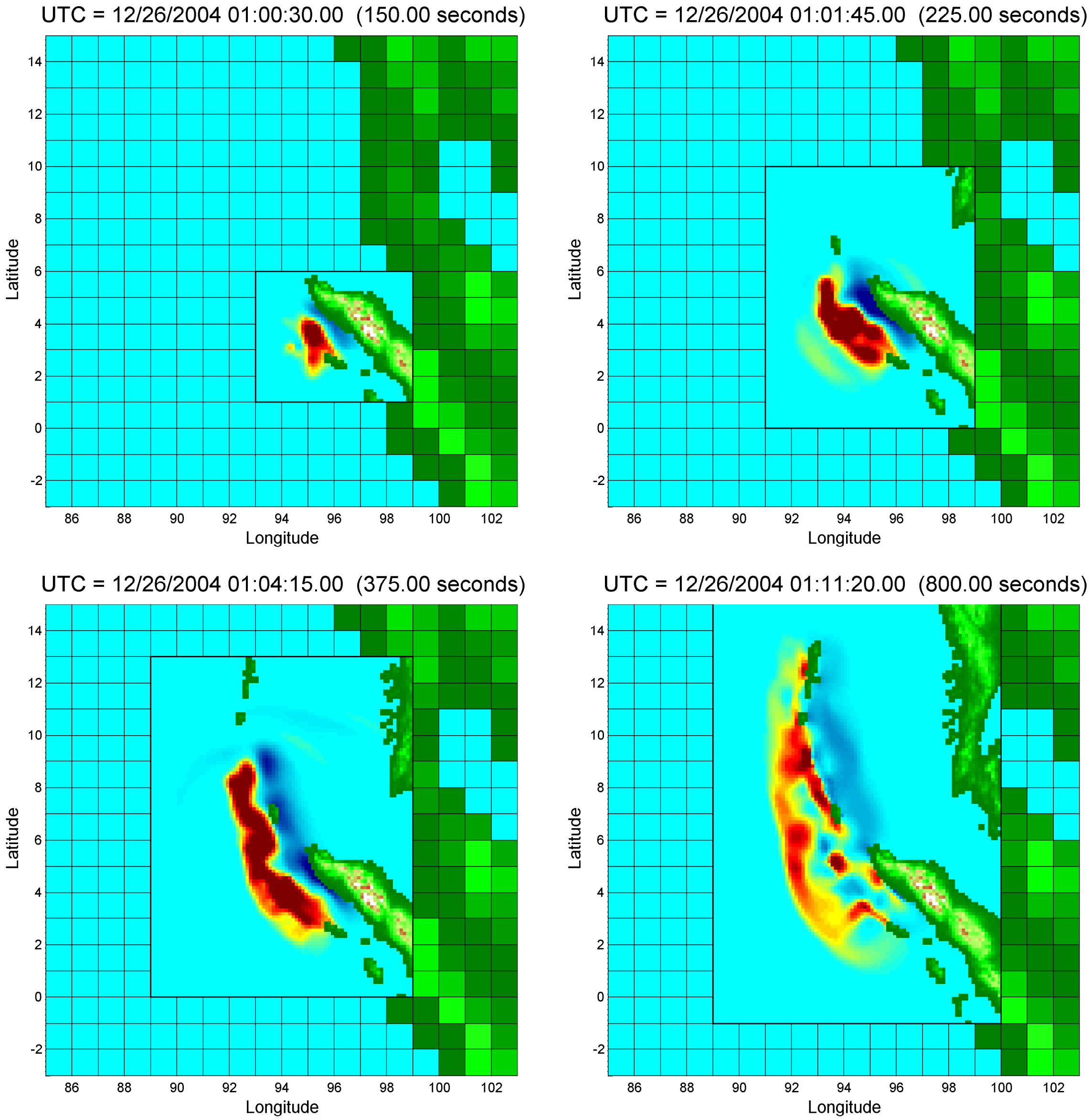

UTC = 12/26/2004 01:00:30.00 (150.00 seconds)
UTC = 12/26/2004 01:01:45.00 (225.00 seconds)
UTC = 12/26/2004 01:04:15.00 (375.00 seconds)
UTC = 12/26/2004 01:11:20.00 (800.00 seconds)
Latitude
Longitude
14
12
10
8
6
4
2
0
-2
86
88
90
92
94
96
98
100
102

Figure 10.2: Dynamic generation of the Indian Ocean Tsunami using a spatial-temporal pattern of the fault motion. The figures show the sea surface displacement at four times during the movement of the seafloor. Sea surface elevations above and below 0 are shown in red and blue respectively. The saturation is at ±1 meter. The simulation has 2 levels of grid refinement, with a refinement factor of 8. Second-level grids appear and track the rupturing fault. Grid lines are omitted from the 2^{nd} level for clarity.

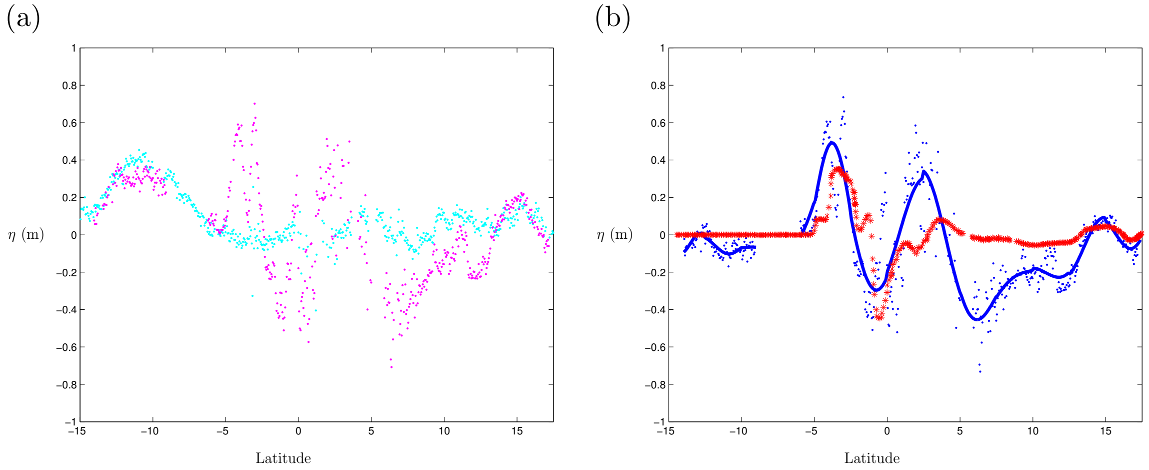


Figure 10.3: Comparison to Jason-1 Satellite Altimetry Data (a) Jason-1 satellite laser altimetry data, from pass 129—an orbit that passed over the Indian Ocean from south to north periodically in 2004. SSHA (sea surface height anomaly) data from cycle 108—which occurred about 10 days before the Tsunami is shown in light blue. Earlier and later cycles after the tsunami (not shown) match cycle 108 fairly consistently. SSHA data from cycle 109—which passed over the Indian Ocean during the tsunami is shown in violet. An abnormal sea surface height is apparent. (b) SSHA data from cycle 108 and 109 is differenced for a rough estimate of the detected sea surface height on December 26th. The blue points are the subtracted raw data at a given latitude, and the solid blue curve is a polynomial fit to the data. The computational solution using TsunamiClaw is shown in red.

10.2 A Three-Level Simulation For the Bay of Bengal

The simulation shown in this section employs three grid levels, with spatial refinement ratios of 8 and 8 for the second and third levels respectively. Temporal refinement to the second level was also 8, but the third level was temporally refined by a factor of 4 (4 time-steps for every time-step on the second-level grids.) This is possible since the third-level grids are reserved for shallow regions where the wave speeds are smaller, yielding smaller Courant numbers as described in Section 7.4. The coarsest grid had 40 by 40 grid cells, 1° by 1° each. Third level grids were restricted to Sri Lanka and eastern India for demonstration of computational times, given the more common scenario where only a given region is inundated by typical transoceanic tsunamis. The simulation was run on a single processor (3.2 GHz running Linux) and took 39 minutes to complete. The simulated time was 10,000 seconds.

Figure 10.4 shows four snapshots of the simulation at different times. As the tsunami waves spread throughout the Bay of Bengal, the propagation is resolved on second-level grids. The grid cell size, $(1/8)^\circ$ (roughly 8 miles), is sufficient to resolve the long wavelength tsunami. Computational efficiency is maximized by the very coarse resolution of the domain in regions where nothing is happening. This savings is realized even further for larger domains, such as that shown in Figure 10.1. As the waves approach Sri Lanka, note the appearance of third-level grids near the eastern coast. The individual third-level grids are outlined in Figure 10.4 (b). All three grid levels can be seen in Figure 10.4 (b) and (c), where the grid lines on the second level have been added for comparison to the coarsest grid at the far western border. The third-level grid resolution, $(1/64)^\circ$ (roughly 1 mile), provide more detailed resolution of impacted shorelines and allow a rough estimate of inundated regions. Note that by only resolving shorelines as waves approach them, computational efficiency is maximized. By maximizing computational efficiency, ultimately higher resolutions or finer grids can be used with the same computational cost as fixed telescoping grids.

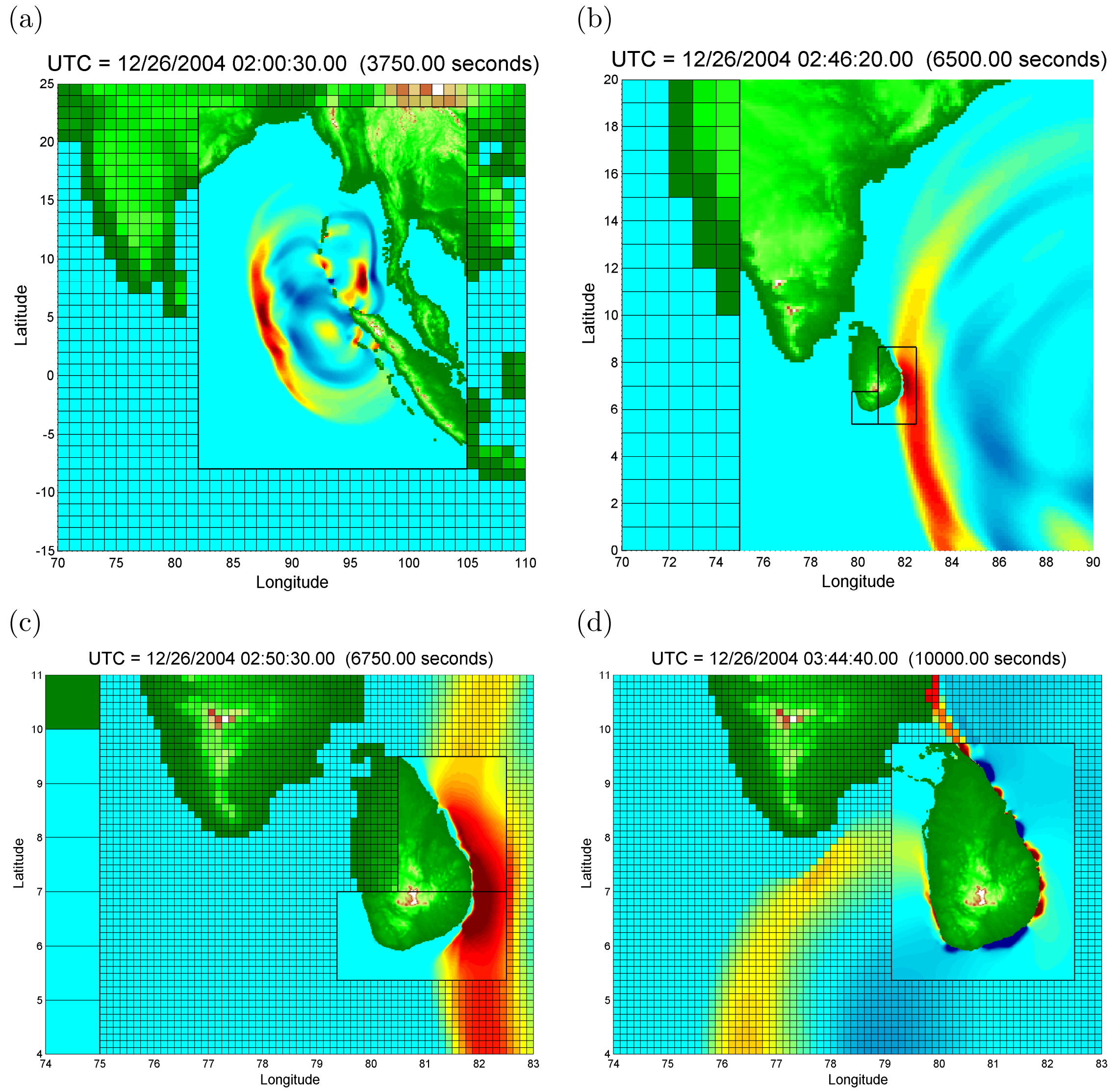


Figure 10.4: Simulation of the Indian Ocean Tsunami using three grid levels. The coarsest grid has 1° grid cells. Grid refinement ratios of 8 are used for the second and third levels. (a) Second-level grids track the deep ocean propagation at a refinement factor of 8—resulting in grid cells of approximately 8 nautical miles. Grid lines are omitted on the second level for clarity. (b) Third-level grids, reserved for shallow coastal areas, appear near the east coast of Sri Lanka as the waves approach. The third-level grids are outlined in black. (c) The shoreline of Sri Lanka is resolved on third-level grids. The second-level grid lines are shown. (d) The final time of the computation described in this section.

10.3 A Fourth Level to Model Inundation at Madras, India

The simulation shown in this section was identical to the simulation shown in the previous section, except that an additional fourth level was allowed near Madras, India. The fourth-level grid was refined from the third level by a factor of 64. Note that this results in grid cells with roughly 25 meter sides, making the grid suitable for inundation modeling. Because of the restriction of the fourth-level grids to extremely shallow waters, a temporal refinement of 16 was used from the third level. Note that this implies that from the first to the fourth level, grids are refined spatially by a factor of 4096 and temporally by only 512. The simulation took approximately 10 hours to reach the simulated time of 10000 seconds on the same machine as the three level simulation described above, and took 13 hours to reach 11000 seconds of simulation time. The fourth grid level therefore represents a considerable added expense. However, as pointed out by LeVeque for a similar computation in [62], performing the same computation with the entire domain at the resolution of the fourth level would take roughly 4000 years of computational time.

Note that since the third-level grid cells are roughly the same resolution as the GEBCO gridded bathymetry, accurate inundation modeling in this region requires more refined bathymetry than is currently available in gridded digital form. For the simulation shown here, it was necessary to scan and interpolate, by hand, nautical navigational charts. This was done for the Madras India region because of some current studies in progress there.

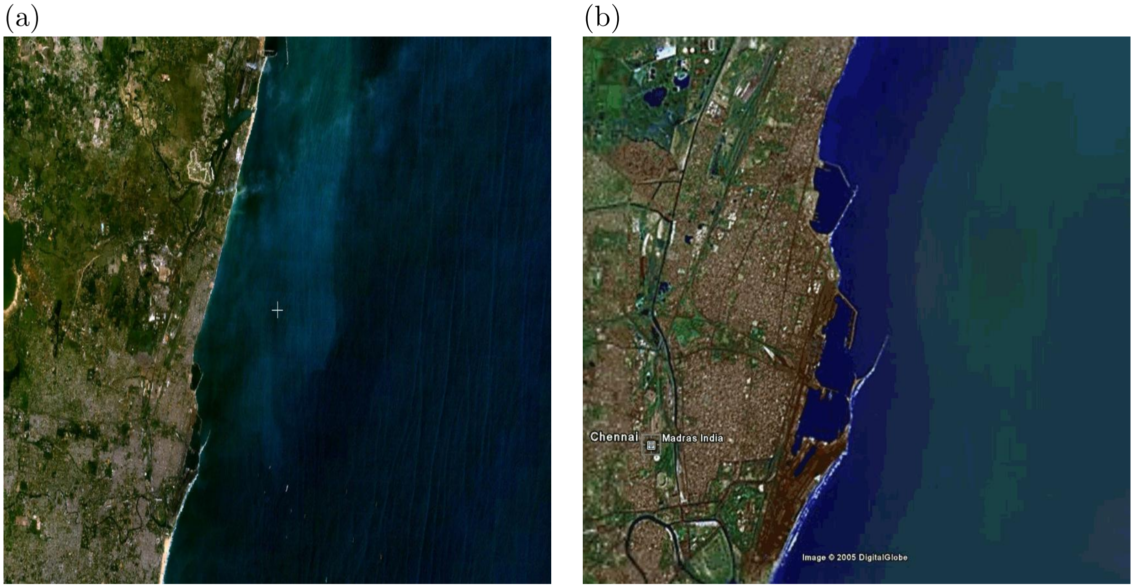


Figure 10.5: Satellite photographs of the coastal region around Madras, India. (a) The coast around Madras, India. (b). A closer view of the Madras harbors. The commercial shipping harbor is south of the smaller fishing harbor. Photographs courtesy of NASA World Wind and Digital Globe Data Set.

Figure 10.5 shows satellite photographs of the Madras region from the NASA World Wind program and the Digital Globe data set. In the photograph on the right, the Madras harbors and seawall are visible. The larger harbor to the south is a commercial port for large freight ships, and the smaller harbor to the north is a less substantial fishing harbor for small boats.

Figure 10.6 shows several snapshots from the simulation. In the first snapshot, the first three levels of grids can be seen. The grid lines are omitted from the third levels, which appear around Sri Lanka and the Madras region on the eastern coast of India. The fourth level appears in Figure 10.6 (b), which shows a close up of the region around Madras. The grid lines are shown for the second and third levels. The fourth-level grids appear near the middle of the third-level grid as the waves approach the coast. Figure 10.6 (c) shows a closer view of Madras and the fourth-level grids, of which there happen to be four of. Only the fourth level is shown in the last snapshot.

Inundation of the Madras coast is shown in four snapshots shown in Figure 10.7. Note that the commercial harbor is inundated as the tsunami approaches from the north, but the fishing harbor is spared by its seawall. The water in the commercial harbor continues to slosh for some time as the wave train inundates the region, while the fishing harbor is protected. Apparently, based on personal communication with several researchers who are working in this area [107, 76], this probably reflects what happened in the two harbors. However, this is still being investigated at this time.

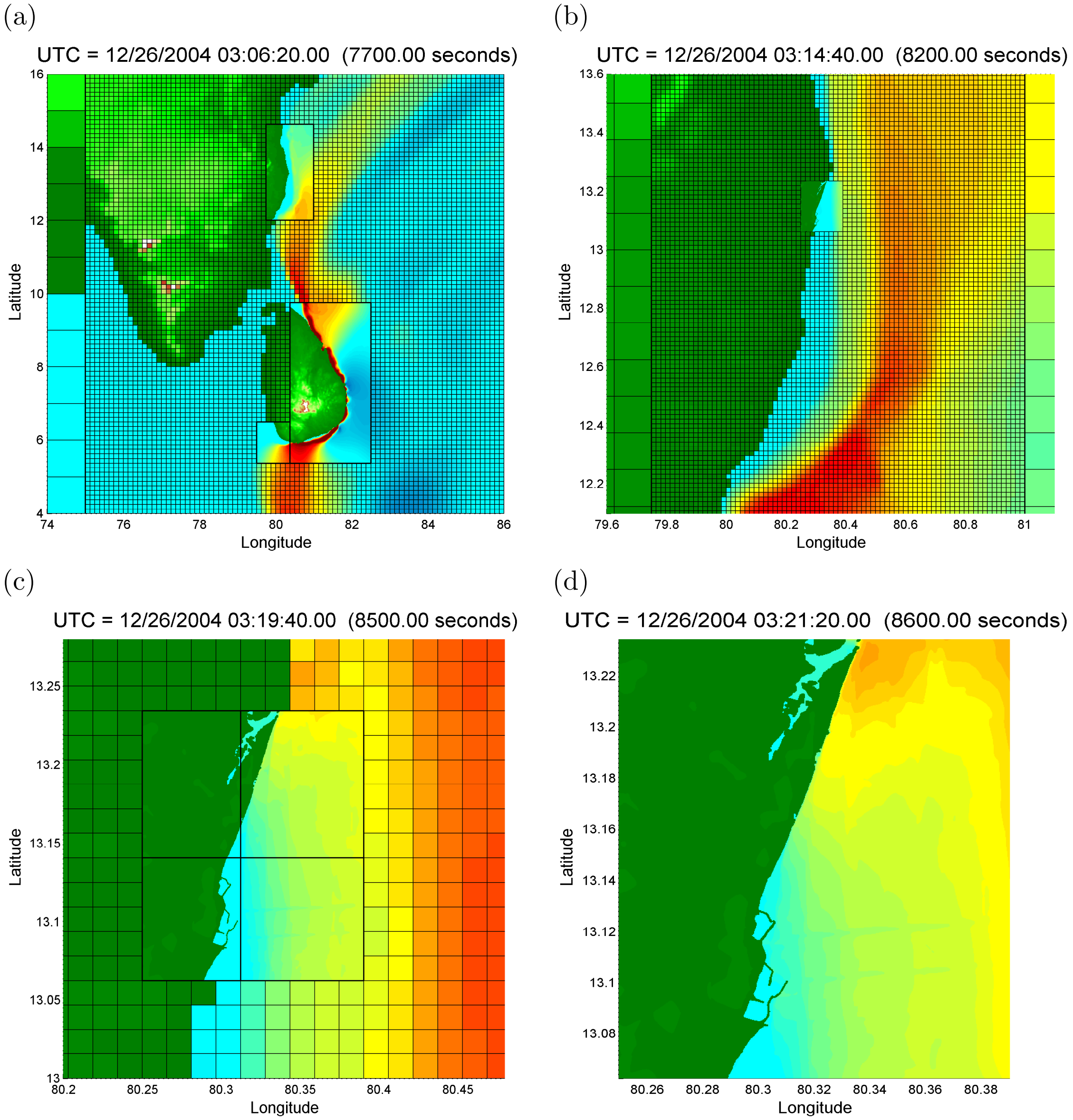


Figure 10.6: Simulation using a fourth-level grid for inundation modeling of the Madras Harbor. The coarsest grid, seen only on the far left of the top left figure, has 1° grid cells. The second and third grid levels are refined by a factor of 8, and the fourth level is refined by a factor of 64. (a) Third level grids around Sri Lanka, and Madras, India, are shown without grid lines. (b) A closer view of the Madras area. Grid lines are shown on the third-level grid, and fourth-level grids appear around the Madras area. (c) A closer view of the fourth-level grids around the Madras harbors as the tsunami approaches. (d) The fourth-level grids around Madras. The grid resolution is 1/4096 of 1°, or approximately 25 meters per grid cell.

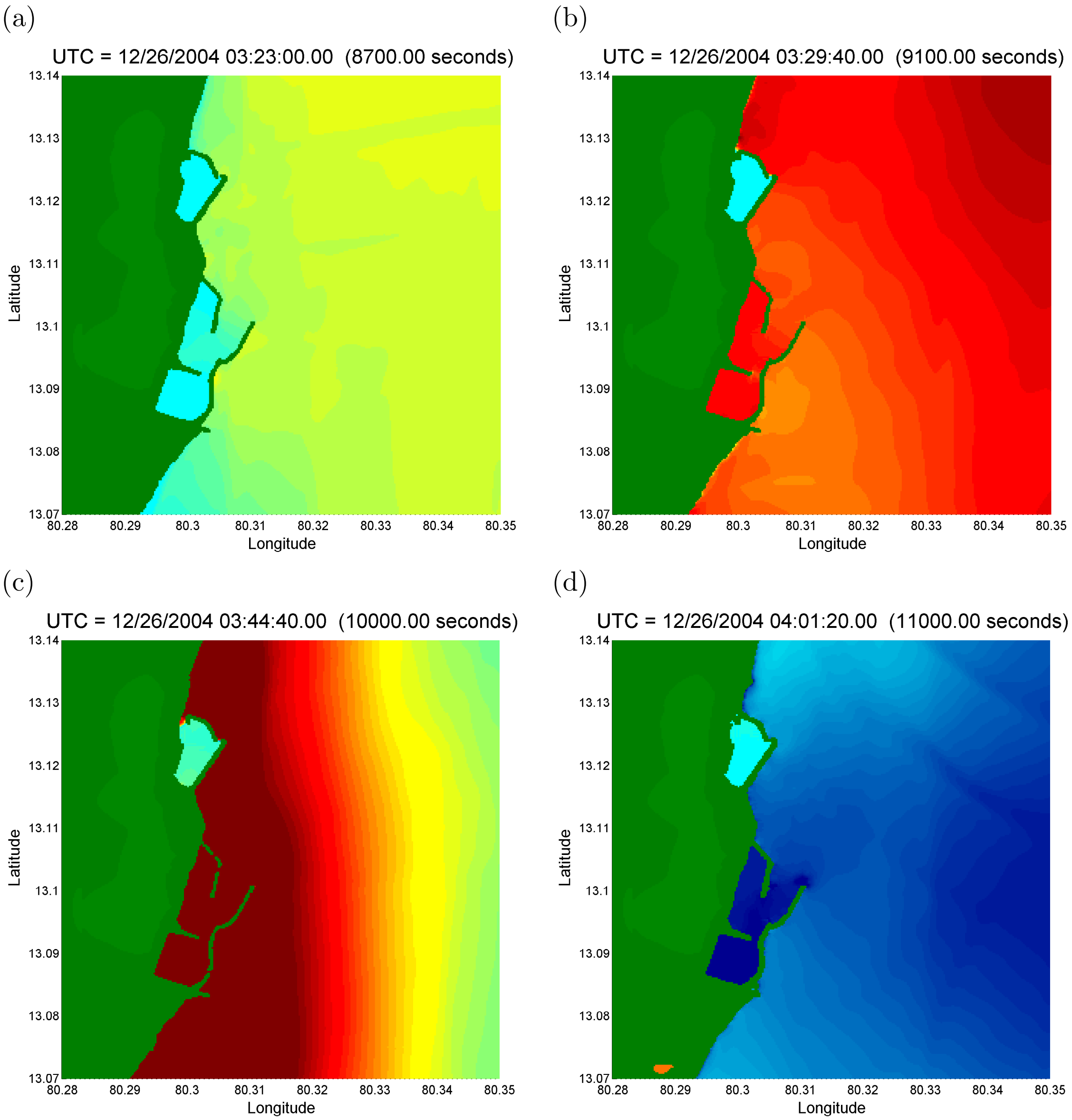

(a)
UTC = 12/26/2004 03:23:00.00 (8700.00 seconds)
(b)
UTC = 12/26/2004 03:29:40.00 (9100.00 seconds)
(c)
UTC = 12/26/2004 03:44:40.00 (10000.00 seconds)
(d)
UTC = 12/26/2004 04:01:20.00 (11000.00 seconds)
Latitude
13.14
13.13
13.12
13.11
13.1
13.09
13.08
13.07
Longitude
80.28
80.29
80.3
80.31
80.32
80.33
80.34
80.35

Figure 10.7: Inundation of the Madras Harbor. (a) The initial tsunami waves approach the harbor. (b) The commercial harbor to the south is inundated as the initial tsunami crest reaches the shore. (c) At the height of inundation the fishing harbor to the north is largely shielded. (d) The commercial harbor experiences sloshing as the trough of the first tsunami waves reach the coast.

Chapter 11

CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter a brief summary will be given of the results of this work, and possible future directions will be mentioned.

11.1 Conclusions

In this thesis, various modern Godunov-type numerical methods for hyperbolic PDEs were developed and extended to the application of tsunami modeling. A novel approximate Riemann solver, described in Chapter 6, was developed in order to deal with the diverse flow regimes exhibited by tsunamis. The Riemann solver is based on an overdetermined or augmented homogeneous system equivalent to the shallow water equations for smooth solutions. Moreover, the Riemann solver maintained properties of methods that converge to discontinuous solutions of the shallow water equations for which the overdetermined homogeneous system fails to be valid. For instance, the solver was shown to be equivalent to the Roe solver in the event of a single shock-wave solution to a Riemann problem. The solver is well-balanced in terms of integrating the source term, and is particularly adept at preserving non-stationary steady states, which was shown to be a necessity for modeling transoceanic tsunamis when using Godunov methods for the conservative form of the shallow water equations. The solver is depth-positive-definite in the sense of maintaining positivity of the solution to Riemann problems and is robust and accurate in the presence of moving shorelines and evolving dry regions. Although this solver is novel, the goals sought very much parallel other recent developments for hyperbolic systems and the shallow water equations from the hyperbolic numerical community, such as well-balanced solvers that preserve positivity.

The wave propagation methods of LeVeque [65, 64], applicable to hyperbolic conservation laws and related nonconservative hyperbolic systems, has been used to implement the methods described in this thesis. The wave propagation methods are particularly suitable for use with the approximate Riemann solver developed, since the solver is related to a nonconservative system. These methods are second-order accurate for smooth solutions to hyperbolic systems, and maintain the properties necessary to converge to discontinuous weak solutions.

Adaptive refinement algorithms, developed by Berger, Collela, Oliger and LeVeque [7, 9, 8] for general hyperbolic systems, were extended and modified to the application of tsunami modeling with the shallow water equations. This development is believed to be unique in the field of tsunami modeling, and is thought to be particularly useful due to the diverse scales of tsunamis. It is believed that adaptive refinement will be an integral piece of tsunami modeling in the future, particularly if real time tsunami forecasting becomes a tool of the hazard mitigation communities.

The goal of this thesis was to lay the groundwork for applying modern Godunov-type finite volume methods and adaptive refinement to tsunami modeling in a unified and practically useful manner. By dealing with realistic data and scales of tsunamis, many unforseen complications have been overcome.

11.2 Future Work

11.2.1 Progress in Riemann Solvers for Godunov Methods

Recently there has been rapid progress in the field of Riemann solvers for the shallow water equations. It is believed that the Riemann solver presented here provides a new simplified framework for well-balanced approximate solvers that preserve positivity. It is hoped that the resonant problem, where transonic Riemann data occurs in conjunction with variable bathymetry, may be dealt with in a more simplified and coherent manner in the future. It may also be that some of the Riemann solver ideas presented in this thesis may be extendable to other applications, such as gas dynamics, particularly for problems with near vacuum states.

11.2.2 Adaptive Refinement

Because of the unusual relationship between wave speeds and flow regimes for tsunamis (largest wave speeds occur in the deep ocean near a steady state), the standard time-stepping guidelines for hyperbolic equations, built in to adaptive routines, may be less than ideal for tsunami modeling. It would be useful to have a completely decoupled method, where time-stepping on every grid is determined entirely by stability on that grid, given a proper buffer zone for that grid.

11.2.3 Parallelization

Ideally, the algorithms presented here could be implemented on parallel platforms, allowing more rapid calculations. For instance, in Section 10.3, it was observed how a fourth-level grid slows the computation considerably. Given numerous Cartesian grids with the adaptive refinement algorithms presented, parallel processing, where different grids are allocated to different processors, would be ideal.

11.2.4 Scientific Modeling and Hazard Mitigation

In this thesis, emphasis was placed on the theoretical development of algorithms that are robust and accurate for solving the shallow water equations when those solutions exhibit diverse flow features indicative of tsunamis. It has been stated that this thesis is not a scientific investigation of tsunamis. However, the end goal of developing algorithms for tsunami modeling is not to use those algorithms for scientific studies. It is hoped that the algorithms presented here may form the basis of robust, accurate and user-friendly codes used by tsunami researchers in different arenas.

BIBLIOGRAPHY

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

[2] I. Aida. Numerical experiments for inundation of tsunamis, Susaki and Usa, in the Kochi Prefecture. *Bull. Earthq. Res. Inst.*, 52:441–460, 1977.

[3] C. J. Ammon et al. Rupture process of the 2004 Sumatra-Andaman earthquake. *Science*, 308:1133–1139, 2005.

[4] M. Arora and P. L. Roe. Postshock oscillations due to shock capturing schemes in unsteady flows. *J. Comput. Phys.*, 130:25–40, 1996.

[5] E. Audeusse, F. Bouchut, M. O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. Sci. Comp.*, 25(6):2050–2065, 2004.

[6] D. Bale, R. J. LeVeque, S. Mitran, and J. A. Rossmanith. A wave-propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM J. Sci. Comput.*, 24:955–978, 2002.

[7] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[8] M. J. Berger and R. J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35:2298–2316, 1998.

[9] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[10] F. Bouchut. *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources.* Birkhäuser Verlag, 2004.

[11] M. Briggs, C. Synolakis, and G. Harkins. Tsunami runup on a conical island. *Proc. of Waves—Physical and Numerical Modeling*, 1994.

[12] T. Buffard, T. Gallouët, and J. M. Hérard. A naive Godunov scheme to solve the shallow water equations. *CR Acad. Sci. Paris.*, 1(326):885–890, 1998.

[13] G. Carrier and H. Greenspan. Water waves of finite amplitude on a sloping beach. *Journal of Fluid Mech.*, 62:97–109, 1957.

[14] G. Carrier, T. Wu, and H. Yeh. Tsunami run-up and draw-down on a plane beach. *Journal of Fluid Mech.*, 475:79–99, 2003.

[15] A. Chinnayya and A. Y. LeRoux. A new general Riemann solver for the shallow water equations with friction and topography. *preprint*, 2003.

[16] A. Chinnayya, A. Y. LeRoux, and N. Seguin. A well-balanced numerical scheme for the aproximation of the shallow water equations with topgraphy: the resonance phenomenon. *to appear in Int. J. Finite Volume*, 2004.

[17] V. T. Chow. *Open Channel Hydraulics.* McGraw-Hill, 1959.

[18] J. F. Colombeau. *Multiplication of Distributions: A tool in mathematics, numerical engineering and theoretical physics.* Springer Verlag, Berlin, 1992.

[19] G. Dubois and G. Mehlman. A non-parameterized entropy correction for Roe's approximate riemann solver. *Numer. Math.*, 73:169–208, 1996.

[20] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 25:294–318, 1988.

[21] B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjogreen. On Godunov-type methods for near low densities. *J. Comput. Phys.*, 92:273–295, 1991.

[22] J. F. Friedlander and M. Joshi. *Introduction to the theory of distributions.* Cambridge University Press, 1998.

[23] K. Fujima, M. J. Briggs, and D. Yuliadi. Runup of tsunamis with transient wave profiles incident on a conical island. *Coastal Engineering Journal*, 42(2):175–195, 2000.

[24] K. Fujima, R. Dozono, and T. Shigemura. Generation and propagation of tsunami accompanying edge waves on a uniform sloping shelf. *Coastal Engineering Journal*, 42(2):211–236, 2000.

[25] K. Fujima, K. Masamura, and C. Goto. Generation and propagation of tsunami accompanying edge waves on a uniform sloping shelf. *Coastal Engineering Journal*, 44(4):373–397, 2002.

[26] T. Gallouët, J. M. Hérard, and N. Seguin. Some approximate Godunov schemes to compute shallow water equations with topography. *Computers and Fluids*, 32:479–513, 2003.

[27] P. García-Navarro and M. E. Vázques-Cendón. On numerical treatment of the source terms in the shallow water equations. *Comput. Fluids*, 29:17–45, 2000.

[28] D. L. George. Website url. `http://www.amath.washington.edu/~dgeorge`.

[29] D. L. George. Numerical approximation of the nonlinear shallow water equations: a Godunov-type scheme. Master's thesis, University of Washington, 2004.

[30] G. Gisler, R. Weaver, C. Mader, and M. Gittings. Two- and three-dimensional simulations of asteroid ocean impacts. *Science of Tsunami Hazards*, 21:119.

[31] M. Gobbi, J. Kirby, and G. Wei. A fully nonlinear Boussinesq model for surface waves. part ii. extension to $o(kh)^4$. *Journal of Fluid Mechanics*, 405:182–210, 2000.

[32] E. Godlewski and P. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer, New York, 1996.

[33] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb.*, 47:271–306, 1959.

[34] F. Gonzalez and Y. Kulikov. Tsunami dispersion observed in the deep ocean. *Tsunamis in the World*, pages 7–16, 1993.

[35] L. Gosse. Localization effects and measure source terms in numerical schemes for balance laws. *Math. Comp.*, 71:553–582, 2002.

[36] C. Goto and K. Sato. Development of tsunami numerical simulation system for Sanriku coast in Japan. *Rep. Port Harbor Res. Inst.*, 32(2):3–44, 1993. In Japanese.

[37] N. Goutal and F. Maurel. Proceedings of the second workshop on dam-break wave simulation. Technical report, Chatou, France, 1997. EDF/DER HE-43/97/016/B.

[38] J. M. Greenberg and A. Y. LeRoux. A well-balanced scheme for numerical processing of source terms in hyperbolic equations. *SIAM Journal of Numerical Analysis*, 33:1–16, 1996.

[39] J. M. Greenberg, A. Y. LeRoux, R. Barraille, and A. Noussair. Analysis and approximantion of conservation laws with source terms. *SIAM Journal of Numerical Analysis*, 34:1980–2007, 1997.

[40] A. Harten. High resolution shcemes for hyperblic conservation laws. *J. Comput. Phys.*, 49:357–393, 1983.

[41] A. Harten and J. M. Hyman. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 50:235–269, 1983.

[42] A. Harten, P. D. Lax, and B. van Leer. On upstream differencing and Godunov-tpye schemes for hyperbolic conservation laws. *SIAM Review*, 25:235–61, 1983.

[43] D. Helmberger, S. Ni, and H. Kanamori. Energy radiation form the Sumatra earthquake. *Nature*, 434:582–583, 2005.

[44] S. Hibberd and D. H. Peregrine. Surf and run-up on a beach: a uniform bore. *Journal of Fluid Mech.*, 95:323–345, 1979.

[45] 1993 HOKKAIDO TSUNAMI SURVEY GROUP. Tsunami devastates Japanese coastal region. *EOS Trans. AGU*, 74(37), 1993.

[46] J. Horillo, Z. Kowalik, and S. Yoshinori. Wave dispersion study in the Indian Ocean Tsunami of december 26, 2004. *Science of Tsunami Hazards*, 25(1):42–63, 2006.

[47] T. Hou and P. LeFloch. Why non-conservative schemes converge to the wrong solutions: Error analysis. *Math. of Comput.*, 62:497–530, 1994.

[48] F. Imamura. Review of tsunami simulation with a finite difference method. *Long-Wave Runup Models*, pages 25–42, 1996.

[49] S. Jin and X. P. Xin. The relaxation schemes for systems of conservation laws in arbitrary space dimensions. *Comm. Pure Appl. Math.*, 48(235), 1995.

[50] R. Johnson. *A Modern Introduction to the Mathematical Theory of Water Waves.* Cambridge University Press, 1997.

[51] S. Karni and S. Canic. Computations of slowly moving shocks. *J. Comput. Phys.*, 136:132–139, 1997.

[52] T. Katsaounis, B. Perthame, and C. Simeoni. Upwinding sources at interfaces in conservation laws. *to appear in Appl. Math. Lett.*

[53] A. Kennedy, J. Kirby, Q. Chen, and R. Dalrymple. Boussinesq type equations with improved nonlinear behavior. *Wave Motion*, 33:225–243, 2001.

[54] D. Komatitsch and J. Tromp. Spectral-element simulations of global seismic wave propagation I. *Geophysical Journal International*, 149:390–412, 2002.

[55] D. Komatitsch and J. Tromp. Spectral-element simulations of global seismic wave propagation II. three-dimensional models, oecans, rotation and self-gravitation. *Geophysical Journal International*, 150:303–318, 2002.

[56] Z. Kowalik, W. Knight, T. Logan, and P. Whitmore. Modeling of the global tsunami: Indonesian Tsunami of 26 December 2004. *Science of Tsunami Hazards*, 23(1):40–56, 2005.

[57] E. Kulikov, A. B. Rabinovich, R. Thomson, and B. Bornhold. The landslide tsunami of November 3, 1994, Skagway Harbor, Alaska. *Journal of Geophysical Research*, 101(C3):6609–6616, 1996.

[58] Y. Kulikov. The highly dispersive waves of the Indian Ocean Tsunami. *Russian Academy of Sciences*, 2005.

[59] A. Kurganov and D. Levy. Central upwind schemes for the St. Venant system. *Mathematical Modelling and Numerical Analysis*, 36:397–425, 2002.

[60] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math*, 13:217–237, 1960.

[61] R. J. LeVeque. CLAWPACK software. `http://www.amath.washington.edu/~claw`.

[62] R. J. LeVeque. Wave propagation software, computational science, and reproducible research. *Proceedings of the International Congress of Mathematicians*, page to appear.

[63] R. J. LeVeque. High resolution finite volume methods on arbitrary grids via wave propagaton. *Journal of Computational Physics*, 78:36–63, 1988.

[64] R. J. LeVeque. *Numerical Methods For Conservation Laws.* Birkhäuser-Verlag, 1990.

[65] R. J. LeVeque. Wave propagation algorithms for multi-dimensional hyperbolic systems. *Journal of Computational Physics*, 131:327–335, 1997.

[66] R. J. LeVeque. Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave propagation algorithm. *Journal of Computational Physics*, 146:346–365, 1998.

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

[68] R. J. LeVeque and D. L. George. High-resolution finite volume methods for the shallow water equations with bathymetry and dry states. In P. Liu, editor, *Proceedings of Long-Wave Workshop, Catalina*, page to appear, 2004.

[69] R. J. LeVeque and M. Pelanti. A class of approximate Riemann solvers and their relation to relaxation schemes. *Journal of Computational Physics*, 172:572–591, 2001.

[70] P. Liu, editor. *Proceedings of the 3rd International Long-Wave Workshop, Catalina*, Catalina Island, California, 2004.

[71] P. L. Liu and Y. S. Cho. An integral equation model for for wave propagation with bottom frictions. *J. Waterway, Port, Coastal, Ocean Engng., ASCE*, 120:594–608, 1994.

[72] P. L. Liu, Y. S. Cho, M. Briggs, U. Kanoglu, and C. Synolakis. Runup of solitary waves on a circular island. *Journal of Fluid Mech.*, 302:259–285, 1995.

[73] C. L. Mader. *Numerical Modeling of Water Waves*. CRC Press, 2004.

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

[75] P. Madsen and O. Sorensen. A new form of the Boussinesq equations with improved linear dispersion. *Coastal Engineering*, pages 183–204, 1992.

[76] T. S. Murty, 2006. Personal Communication.

[77] S. Noelle, N. Pankratz, G. Puppo, and J. R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *J. Comput. Phys.*, 213:474–499, 2006.

[78] O. Nwogu. Alternative form of the Boussinesq equations for nearshore wave propagation. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 119(6):618–638, 1993.

[79] National Oceanographic and Atmospheric Administration. NOAA Center for Tsunami Research. `http://www.http://nctr.pmel.noaa.gov/`.

[80] D. H. Peregrine. Long waves on a beach. *Journal of Fluid Mechanics*, 27(4):815–827, 1967.

[81] P. Rivera. Modeling the Asian tsunami evolution and propagation with a new generation mechanism and a nonlinear dispersive wave model. *Science of Tsunami Hazards*, 25(1):18–33, 2006.

[82] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[83] P. L. Roe. *Upwind differencing schemes for hyperbolic conservation laws with source terms.*, volume 1270 of *Lecture Notes in Math*, pages 41–51. Springer-Verlag, 1987.

[84] J. A. Rossmanith. *A Wave Propagation Method with Constrained Transport for ideal and shallow water magnetohydrodynamics.* PhD thesis, University of Washington, 2002.

[85] J. A. Rossmanith, D. S. Bale, and R. J. LeVeque. A wave propagation algorithm for hyperbolic systems on curved manifolds. *J. Comput. Phys.*, 199:631–662, 2004.

[86] K. Sitanggang and P. Lynett. Parallel computation of a highly nonlinear boussinesq equation model through domain decomposition. *International Journal for Numerical Methods in Fluids*, In Press, 2005.

[87] J. J. Stoker. *Water Waves:The Mathematical Theory with Applications.* Interscience Publishers, New York, 1957.

[88] G. Strang. On construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, pages 506–517, 1968.

[89] P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal of Numerical Analysis*, 21:995–1011, 1984.

[90] C. E. Synolakis. The runup of solitary waves. *J. Fluid Mech.*, 185:523–545, 1987.

[91] C. E. Synolakis. Generation of long waves in the laboratory. *J. Waterways, Port, Coastal, Ocean Engng. ASCE*, 116:252–266, 1990.

[92] C. E. Synolakis, P. Liu, G. Carrier, and H. Yeh. Tsunamigenic sea-floor deformations. *Science*, 278:598–600, 1997.

[93] V. V. Titov. *Numerical Modeling of Long Wave Runup.* PhD thesis, University of Southern California, 1997.

[94] V. V. Titov and F. I. Gonzalez. Implementation and testing of the method of splitting tsunami (MOST) model. *NOAA Technical Memorandum ERL PMEL-112.*

[95] V. V. Titov, A. B. Rabinovich, H. O. Mofjeld, R. E. Thomson, and F. I. Gonzalez. The global reach of the 26 December 2004 Sumatra Tsunami. *Science*, 309(5743):2045–2048.

[96] V. V. Titov and C. E. Synolakis. Extreme inundation flows during the Hokkaido-Nansei-Oki Tsunami. *Geophys. Res. Lett.*, 24(11):1315–1318.

[97] V. V. Titov and C. E. Synolakis. Modeling of breaking and nonbreaking long wave evolution and runup using VTCS-2. *Jounal of Waterways, Ports, Coastal and Ocean Engineering*, 121(6):308–316.

[98] V. V. Titov and C. E. Synolakis. Numerical modeling of tidal wave runup. *Jounal of Waterways, Ports, Coastal and Ocean Engineering*, 124(4):157–171.

[99] V. V. Titov and C. E. Synolakis. A numerical study of wave runup of the September 2, 1992 Nicaraugan tsunami. *Tsunami 93 Proc. IUGG/IOC Intl Tsunami Symp.*, pages 627–636, 1993.

[100] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer–Verlag, Berlin, 1997.

[101] E. F. Toro. *Shock Capturing Methods for Free Surface Shallow Flows.* John Wiley and Sons, Chichester,United Kingdom, 2001.

[102] R. Walters. A semi-implicit finite element model for non-hydrostatic (dispersive) surface waves. *International Journal for Numerical Methods in Fluids*, 7:721–737, 2005.

[103] G. Whitham. *Linear and Nonlinear Waves.* John Wiley and Sons, New York, 1974.

[104] H. Yeh. Nonlinear progressive edge waves: Their instability and evolution. *Journal of Fluid Mechanics*, 152:479–499, 1985.

[105] H. Yeh. Experimental study of standing edge waves. *Journal of Fluid Mechanics*, 168:291–304, 1986.

[106] H. Yeh. Free-surface dynamics. In P. L. F. Liu, editor, *Advances in Coastal and Ocean Engineering*, volume 1, pages 1–75, 1994.

[107] H. Yeh, 2006. Personal Communication.

[108] H. Yeh, Ghazali A., and I. Marton. Experimental study of bore runup. *Journal of Fluid Mechanics*, 206:563–578, 1989.

[109] H. Yeh, F. Imamura, C. E. Synolakis, Y. Tsuji, P. Liu, and S. Shi. The Flores Island tsunamis. *EOS, Trans. AGU*, 74(33)(369):371–373, 1993.

[110] H. Yeh, P. Liu, M. Briggs, and C. Synolakis. Propagation and amplification of tsunamis at coastal boundaries. *Nature*, 372:353–355, 1994.

[111] H. Yeh and K. M. Mok. On turbulence in bores. *Physics of Fluids*, 2:821–828, 1990.

[112] H. Yeh, V. Titov, V. Gusiakov, E. Pelinovsky, V. Khramushin, and V. Kaistrenko. The Shikotan earthquake tsunamis. *Pure. Appl. Geophys.*, 1995.

Appendix A

TSUNAMICLAW USER'S GUIDE

A.1 Introduction

TsunamiClaw is an extension of the Clawpack software, a set of fortran routines for solving hyperbolic systems of PDEs. TsunamiClaw is available for download at the Clawpack website [61]. TsunamiClaw contains many of the standard routines from Clawpack, and the adaptive mesh routines Amrclaw, as well as many specialized routines for solving the nonlinear shallow water equations, particularly for tsunami modeling. The standard TsunamiClaw library contains all of the necessary library routines, and is available at the Clawpack website. This document contains information specific to TsunamiClaw. More general documentation for Clawpack and Amrclaw is available at [61] and should be consulted before reading this document.

A.2 General Features of TsunamiClaw

TsunamiClaw allows one to model global-scale tsunamis and inundation on a latitude-longitude grid (or smaller local tsunamis on a Cartesian grid) with a diverse range of spatial and temporal scales. This is accomplished by using a single coarse *level* 1 grid for the entire domain, and evolving rectangular Cartesian sub-grids of higher refinement, *level* 2 $,\cdots,$*level* n, that track moving waves and inundation at the shoreline. Up to six levels may be used (though typically four or less is recommended). At any given time in the calculation, a particular level of refinement may have numerous disjoint grids associated with it. User specified integers determine the refinement ratios between particular levels, which can lead to a large degree of refinement even for a small number of levels.

The following example illustrates a typical usage of grid-level refinement for calculating the Indian Ocean Tsunami. Consider four levels with refinement ratios of 10. The entire Bay of Bengal can be covered with a *level* 1 grid with 50×50 grid cells, each 100km × 100km ($1^\circ \approx 100$km). A second level might be dedicated to tracking the deep ocean tsunami with grid cells that are $(10\text{km})^2$ (this resolution is sufficient given the long wavelength of deep ocean tsunamis). Third-level grids near the shore could resolve the compressing tsunami onto $(1\text{km})^2$ grid cells, resolution sufficient to identify vulnerable coastlines. Fourth-level grids would then allow resolution onto $(100\text{m})^2$ grid cells. Alternatively, if a specific region was of interest, one could easily resolve a specific community or shoreline onto $(10\text{m})^2$ grid cells using four levels and the right refinement ratios, allowing detailed inundation studies. Calculations of this sort can easily be done on personal desktop or laptop computers using TsunamiClaw. Computing the same domain at this single highest resolution would require 10^{10} grid cells and prohibitively small time-steps, and would not be feasible given current computing technology.

TsunamiClaw is based on a finite volume numerical method, which means that the solution is represented as a piecewise constant, with numerical values approximating the average solution value in each discrete computational grid cell. There is no specific reference to the shoreline—grid cells may simply be wet or dry depending on their location, and may fill-up or drain-out of water as waves inundate or draw-down at the shoreline. This means that dry land is part of the computational domain, and a single grid is a simple rectangle that may overlap the shoreline. TsunamiClaw solves the shallow water equations in their physically relevant conservative form, therefore, the solution is represented as water depth and momentum. This is one feature that allows convergence to discontinuous bores. See [67] and [8] for a mathematical descriptions of these methods.

A.3 Running TsunamiClaw

This document assumes a basic working knowledge of Unix or Linux. TsunamiClaw is written in fortran 77, and can be run on any system with a proper fortran compiler. However, using the provided Makefiles requires that you are using a Unix or Linux operating system. TsunamiClaw consists of a directory of library routines, as well as an application directory that will be used to specify problem specific parameters. Routines in the library should typically not require any modification, and should only be modified at your own risk. (One possible exception is the `call.i` file. See the standard Clawpack documentation).

A.4 The TsunamiClaw *library*

The standard TsunamiClaw library,
`tsunamiclaw/lib`,
is available at the Clawpack website as a tar-zipped directory,
`tsunamiclaw.tar.gz`.
On your system, the unzipped directory should be in a path specified by an environment variable that you should name `TSUNAMICLAW`. The path to the TsunamiClaw library will therefore be referred to as,
`$TSUNAMICLAW/lib`.
It is recommended to set this environment variable in one of your login scripts (such as .cshrc or .bashrc depending on your shell). Before using TsunamiClaw, the `.f` fortran files in the library must be made into `.o` object files. This can be done by typing "`make`" from the library directory
`$TSUNAMICLAW/lib`.
(Note: the variable `TSUNAMICLAW` must be set before using `make`.) If the library is modified (not recommended), or moved from one system to another, `make` should be run again.

The TsunamiClaw library contains a mix of routines, some that are unique and some that are standard routines of Clawpack and Amrclaw. Routines that are unique to the TsunamiClaw library are named by the convention `filename_tc.f`. Other versions from the standard Clawpack library may or may not be compatible with the `filename_tc.f` routines. Use caution if updating to newer versions of the Clawpack or Amrclaw libraries.

A.5 Application Directory

TsunamiClaw will be run from an application directory that may reside anywhere in your directory structure. An application directory will contain input data files where parameters will be set for each specific problem. TsunamiClaw uses two input data files, `amr2ez.data` and `setprob.data`. Standard Clawpack computational parameters are set in `amr2ez.data`, such as the physical domain and refinement parameters, and the other file, `setprob.data`, is used to set computational parameters specific to TsunamiClaw. The application directory will also contain several required fortran routines (`.f` files), explained in section section A.8, that can optionally be modified to your needs. However, for standard applications it should only be necessary to set parameters in the input data file. To make the executable that runs TsunamiClaw, type "`make xtsunami`" from the application directory. The executable file `xtsunami` can then be run from this directory. An example application directory comes with TsunamiClaw,

`$TSUNAMICLAW/tc_example`,

that contains the required fortran routines (`.f` files) and the input files (`.data` files), as well as some matlab plotting scripts (`.m` files). It is recommended that you copy this directory to another location outside of `$TSUNAMICLAW` before making modifications, so that if you update TsunamiClaw to newer versions, your personal application directories will not be overwritten.

A.6 Data Input (`setprob.data`)

The computational parameters set in `setprob.data` are described below. Note that integer variables start with the letters `m` or `l`, so values specified in `setprob.data` for these variables should not contain any decimal points.

`gravity:` The gravitational constant should typically be set to 9.8, unless units other than meters and seconds are desired. When using the default latitude-longitude units, internal calculations rely on meters as the physical units, so the default value of 9.8 should be used.

`depthtolerance:` Water depths below this value (in meters) will be effectively treated as zero. This makes the computation more efficient, since it prevents computing the solution in areas with vanishingly small depths. It is recommended to use a reasonable value such as 10^{-3} (1 millimeter) or larger. Since the shallow water equations have no viscosity, water velocities can be unrealistically large in regions with very shallow depths, and this can lead to inefficient time-stepping or even instabilities. (Note that friction can help alleviate this problem.) Using a value in the range of 10^{-3} to 10^{-2} is recommended.

`shallowdepth:` This parameter is used for refinement purposes. Typically more refinement is desired, and needed due to decreasing wavelengths, in shallow coastal waters. This parameter indicates areas that are considered "*shallow*" in meters. Larger values will lead to larger areas of refinement, while smaller values will lead to limited refinement areas closer to shore. A typical value might be 100 (meters). See `maxdeepoceanlevel` below.

`wavetolerance:` This parameter is used for refinement purposes near waves, and controls how sensitive refinement is to deviations from level sea-surface. Smaller values will therefore lead to larger areas of refinement and larger values will lead to smaller areas of refinement surrounding only the biggest waves. Since deep ocean tsunamis do not necessarily have large amplitudes, a value such as 10^{-1} (10 cm) might be a typical value for a global scale tsunami.

`frictioncoeff:` This parameter sets the Manning coefficient used for optional bottom friction. Bottom friction may be important for realistic run-up heights and can speed calculation times as well. The friction formula is a standard empirically derived formula. Higher values of the Manning coefficient correspond to greater friction. A typical value used in a tsunami calculation might be 0.025. For friction, the source-term splitting option must be set in `amr2ez.data`. If you do not want to use friction, you can set `frictioncoeff` to 0.0, or turn-off source term integration in `amr2ez.data`.

`maxdeepoceanlevel:` This parameter determines what level of refinement is allowed in the deep ocean. Typically deep ocean tsunamis have very long wavelengths, and so high levels of refinement are not needed and would be computationally wasteful in the deep ocean. A typical computation might only allow the second level of refinement in the deep ocean, reserving higher level grids for shallow regions near the shore. For classification of deep and shallow TsunamiClaw uses the `shallowdepth` parameter.

`levelallowdomain:` This parameter determines what level of refinement is allowed on the entire domain. For instance, it might be desired to restrict high levels of refinement to certain regions of the domain. In this case `levelallowdomain` should be set to some lower value that is allowed on the entire domain. See below for parameters that allow higher refinement in certain regions.

`methodquake:` This parameter determines the type of method that is to be used to generate the tsunami. It can have integer values of 0,1, or 2. If `methodquake`= 2, then a dynamic fault model is used, as specified in a user provided data file. If `methodquake` = 1, then a static initial condition is used, as specified in a user provided data file. The formats for these files are described in the section “Fault Models”. If `methodquake`= 0 then the user must provide some other means of generating the tsunami, such as specifying an initial condition directly in `qinit.f`, or perhaps a user specified boundary condition. To use these latter options, see the standard Clawpack documentation.

`minfaultrefinelevel`: This parameter is used for determining refinement in the region described by the fault model. It can have any integer value up to the number of refinement levels specified in `amr2ez.data`. Setting `mfaultrefinelevel`= n will enforce that level n grids or finer will cover the fault region during the fault motion. If you do not want to enforce a particular level, but wish for TsunamiClaw to treat the fault region like any other region, refining based on wave heights and water depths, then set `mfaultrefinelevel` to a low value such as 0 or 1.

`faultfname`: On the line following `mfaultrefinelevel`, you should place the relative or full path to the file of the fault model in single quotes. If `methodquake`= 0, then the line should be blank.

`mbathfiles`: Set `mbathfiles` to the integer number of bathymetry files that you would like to read in. If you are specifying bathymetry directly by some other means, set `mathfiles` = 0. See the description of `setaux.f`.

`levelallowbathfile(mbathfiles)`: This line should contain a list of `mbathfiles` integers, specifying the level of refinement allowed over each bathymetry file to be used. This variable is useful if you would like to use a certain level of refinement over most of the domain, yet reserve high levels of refinement for regions where you have more detailed bathymetry. For more about how bathymetry files can be used, see the section Bathymetry.

`bathfname(mbathfiles)`: The next `mbathfiles` lines should contain the paths to the bathymetry files in single quotes. There should be at least one, unless `mbathfiles` = 0, in which case the line should be empty. The order that they are listed in should correspond to values specified in the list of integers.

`mregions`: For additional control and restriction of refinement to particular regions, set `mregions` to an integer number of regions that you would like to specify.

`levelallowregion(mregions)`: This line should contain a list of `mregions` integers, specifying the level of refinement allowed over each region to be specified. This option is useful if you would like use high levels of refinement for regions of interest, even if these regions do not correspond specifically to bathymetry files.

`regioncoords(4,mregions)`: The next `mregions` lines should each contain a list of four coordinates specifying the particular region: `xlower xupper ylower yupper`.

A.7 Data Input (`amr2ez.data`*)*

The file `amr2ez.data` is used to set standard computational parameters for Clawpack and Amrclaw. Consult the standard Clawpack documentation, particularly the section on Amrclaw, for complete information about this file. In this section, a few of these parameters will be described that should have specific values for TsunamiClaw.

`method(5)`: This variable specifies whether to integrate a source-term. For TsunamiClaw, setting this to 1, will incorporate friction. Otherwise set this variable to 0. Using friction will improve stability, and is recommended when trying to speed the calculation or use anisotropic time refinement explained in section A.12.

`meqn`: This indicates the number of equations to be solved. For TsunamiClaw this should be 3, since the shallow water equations are solved.

`mwaves`: This indicates the number of waves in the solution. For TsunamiClaw this should be 3 also.

`mcapa`: This should be set to 2, when using a latitude-longitude grid, or 0 otherwise. See section A.9 for more information.

`maux`: This should be set to 3 (or greater if you add extra auxiliary variables), when using a latitude-longitude grid, or 0 otherwise. See section A.9 for more information. When solving on a latitude-longitude grid, `maux` should be followed by three lines with words in quotes: “center,” “capacity,” and “yleft” respectively. Consult the standard Clawpack documentation for more information about auxiliary variables.

`t0`: This specifies the initial starting time. It is recommended to use 0, or perhaps a positive number. If you are performing a restart from an old run, then reset this value to the new starting time when using a dynamic fault model. See section A.11 for an explanation, and see the standard Clawpack for information about how to perform a restart.

A.8 User Accessible Fortran Routines

This section gives a brief description of the fortran routines (`.f` files) in the application directory. These files can optionally be modified for your particular application if the default behavior of TsunamiClaw is not sufficient.

`setprob.f` This routine performs all of the data input at the beginning of the computation from `setprob.data`. It should typically not need to be modified, but could be used to input additional data.

qinit.f This routine initializes the solution at the start of the computation (except in the case of a restart). If you use methodquake $= 1, 2$ then you will not need to modify this routine. If you wish to initialize the solution in some other manner, then this is the relevant routine to use. More information can be found in the standard CLAWPACK documentation.

setaux.f This routine initializes auxiliary arrays that can be used to define some other quantity in computational grid cells. TSUNAMICLAW uses aux(i,j,1), aux(i,j,2) and aux(i,j,3) for internal purposes, such as bathymetry and capacity functions associated with a latitude-longitude grid. If you would like to use additional auxiliary arrays for some other quantity see the standard CLAWPACK documentation.

b4step2.f This routine is called before each new time step on each grid. If you need to perform a check on the solution or output information, this is a good place to do it. For instance, this is a convenient place to make calls to user-written routines, or to output the solution if the standard output is not sufficient.

allowflag.f This routine is used for adaptive refinement purposes. Refinement parameters, such as the number of refinement levels and refinement ratios are specified in amr2ez.data. AMRCLAW refines by flagging cells on a particular grid at a particular level that meet certain criteria (such as wave-height in TSUNAMICLAW). A clustering algorithm then groups flagged cells into rectangular regions which will then be refined to the next higher level. allowflag.f controls whether a cell at location (x, y) and time t is allowed to be flagged. In the default version of TSUNAMICLAW, allowflag.f is given information (described in section A.6) from setprob.data to determine which areas are allowed to be refined. If the standard options are not sufficient, allowflag.f can be modified to meet special needs.

flag2refine.f This is the routine that actually flags the cells if they are allowed to be flagged by allowflag.f. The standard flag2refine.f in TSUNAMICLAW uses parameters form setprob.data. Modify this routine if you would like to refine based on some other criteria.

A.9 The Computational Domain

TSUNAMICLAW can compute on either a latitude-longitude grid or a Cartesian grid. In the former case, the latitude-longitude computational units are mapped internally to spherical physical space using a capacity function that takes into account the true physical geometry of each computational grid cell. Use of this capacity function requires that mcapa $= 2$ in amr2ez.data, and maux ≥ 3. When computing on a latitude-longitude domain the bounds of the domain are specified by longitude (x), and latitude (y) in amr2ez.data. Latitude should be specified in the standard $-90°$ to $90°$ convention, where the southern latitudes are negative. Although it is theoretically possible to compute all the way to the poles, this code has not been tested at extreme latitudes (because of vanishing grid cells at the poles, some numerical issues may arise if computing at extreme latitudes). Typically it is possible to use either the −180° to 180° convention for longitude, or the 0° to 360° convention, where in both cases, the prime meridian corresponds to 0° and longitude increases toward the east. If the computational domain crosses the dateline however, and does not wrap around the entire globe, then the 0° to 360° convention must be used, so that the longitude of the computational domain increases logically. If the computational domain wraps around the entire globe in longitude, then periodic boundary conditions in x should be specified in `amr2ez.data`. In this case, −180°to 180°, degree units could be used for longitude as well, with the computational domain being −180°to 180°, however, it is preferable to place the computational boundary at 0° at the prime meridian rather than at 180° in the middle of the Pacific Ocean which is much more likely to be of interest. When computing on a latitude-longitude grid, output units are in meters and seconds. This cannot be adjusted because of internal calculations.

If you wish to compute on a Cartesian domain, then the physical space and computational space are the same, and domain bounds are set in meters or some other desired unit. It is important that `mcapa` $= 0$ in this case. When computing on a Cartesian domain, units are selected by adjusting the gravitational constant in `setprob.data` appropriately. (See the standard Clawpack documentation for more details about the `mcapa` and `naux` variables).

A.10 Bathymetry

The user must provide a file or files that contain the bathymetric and topographic data. The names of these numeric ascii text files are specified in `setprob.data`. When computing on a latitude-longitude domain, the files should contain three columns: longitude, latitude and elevation (in meters above sea level so that seafloor values are negative). The files must conform to the standard GIS format, where data begins in the northwest corner and advances in longitude first toward the northeast corner before moving to the next latitude south. This code assumes that there are no missing values, either above or below sea level, and that data points are uniformly spaced on a rectangular domain (in latitude-longitude space). If your bathymetry data does not conform to this standard, you must preprocess your data appropriately by interpolation of some sort, before using the files in TsunamiClaw.

When computing on a Cartesian domain, bathymetry should be supplied in the same logical format; 3 columns: x, y and elevation, advancing in the x direction first, from the largest y value to the smallest. Units should be compatible with the gravitational constant.

At the start of the computation, TsunamiClaw will read in all of the bathymetry files indicated in `setprob.data` and put the data into arrays. Separate bathymetry files can be nested or overlap in any manner desired, as long as the union of all of the files is at least as large as the computational domain specified in `amr2ez.data`. Typically there will be at least one file that has bathymetry covering the entire computational region, this isn't strictly necessary however. TsunamiClaw will determine the bathymetry in computational cells by interpolating or averaging the bathymetry arrays, depending on the relative resolution of the bathymetry and the computational cell. When two or more bathymetry arrays overlap the same area, TsunamiClaw will use the array with the highest resolution.

It is possible to specify idealized bathymetry (such as a simple function) rather than using a data file. (See `mbathfiles` in `setprob.data`)

A.11 Fault Models

TsunamiClaw is capable of using dynamic fault models to initialize the tsunami. The dynamic model should describe the time-dependent seafloor displacement in some region. If a dynamic model of this form is used, set `methodquake` = 2 in `setprob.data`. This model should be specified in a numeric ascii text file with four columns: time, longitude, latitude, and displacement (from undisturbed bathymetry in meters). The time column should advance the most slowly, and longitude and latitude should advance in the standard GIS way as described in the Bathymetry section. The spatial domain of the fault model (the last 3 columns at each given time) should conform to the same standards as bathymetry described in the previous section. That is, all values on a rectangular domain should be specified, and should be uniformly spaced in latitude and longitude. The time values in the first column, should be uniformly spaced as well. If your fault model does not conform to these standards, then you should preprocess you data before using TsunamiClaw.

It is also possible to use a static initialization. In this case the bathymetry will not be altered dynamically, and the tsunami will be initiated by an initial condition that describes the sea surface at the start of the computation. This model should be specified in a numeric ascii text file with three columns corresponding to longitude, latitude and displacement of the sea surface (in meters). The longitude and latitude columns should advance in the standard GIS way, and should conform to the same standards described above. If this option is used set `methodquake` = 1 in `setprob.data`.

It is possible to instantaneously displace the sea-floor at some time after the start of the computation, in order to generate the tsunami. If this option is desired, use the dynamic fault model option. The dynamic fault file would then just have a single time in the first column corresponding to the time of displacement.

If you would like to generate the tsunami in some other way, modify `qinit.f` as needed, or enforce an appropriate boundary condition. Boundary conditions can be easily and stably enforced in the Clawpack software packages by simply specifying ghost cell values that correspond to the edge of the domain. See the standard Clawpack documentation for more details.

A NOTE ABOUT RESTARTS: Clawpack has a restart feature that allows a computation to be resumed from a previous computation. This is very useful in TsunamiClaw since, often, a single real tsunami may be computed many times for studying different regions. When a restart is used in conjunction with a dynamic fault model, it is important to change the initial time in `amr2ez.data` to the computational time at the beginning of the restart. This is so that the state of the bathymetry at the end of the previous computation can be resumed correctly.

A.12 Time-Stepping and Anisotropic Refinement

The number of grid levels and refinement ratios between grid levels are set in amr2ez.data. The variable mxnest sets the maximum number of grid levels, and the variable inrat is a list of (mxnest -1) integers specifying the refinement ratios between these levels. The default behavior of AMRCLAW is to refine isotropically in space and time. For general hyperbolic conservation laws, maintaining stability (and accuracy) typically requires this equal refinement in time. That is, a grid at *level* n must typically take inrat(n-1) time-steps for each time-step of *level* $n-1$ grids. This restriction is due to the CFL condition which, in the case of hyperbolic problems, depends on the speed at which characteristic information propagates. For the shallow water equations, characteristic wave speeds are related to the depth of the water. In *shallow* water, wave speeds are slower, which means that grids in water that is much more shallow than the rest of the domain could take much larger time-steps while still maintaining stability and accuracy. Therefore, if grids at a particular level are restricted to shoreline regions, or regions with relatively shallow water compared to grids at the next lower level, those grids would not typically require refinement in time as large as refinement in space. For this reason, it is desirable to select anisotropic refinement in TsunamiClaw. To select this option, mxnest should be preceded by a negative sign in amr2ez.data. In this case, three lists of integers, inratx, inraty and inratt are required in amr2ez.data. The ratios for inratx and inraty would typically be identical, but inratt may have some integers that are smaller for levels that are restricted to shallow regions. Note that the variables maxleveldeepocean and shallowdepth in setprob.data allow control of grid level restrictions to shallow regions. By carefully tuning shallowdepth, maxleveldeepocean, inratx, inraty and inratt, one can greatly speed calculations and still maintain stability. An example is shown below.

Suppose that the following values are set in setprob.data:

```
⋮                    ⋮
1.0d2                shallowdepth
⋮                    ⋮
2                    maxleveldeepocean
⋮                    ⋮
```

Then it might be feasible to reduce time-steps taken on *level* 3 grids since they are restricted to shallow regions, by setting the following variables in amr2ez.data:

```
⋮                      ⋮
-3                     mxnest
6 6                    inratx
6 6                    inraty
6 2                    inratt
⋮                      ⋮
```

In this case, *level* 3 grids will be refined spatially in x and y by a factor of 6, but will only take 2 time-steps for every time-step taken on *level* 2 grids. Since the ocean may have depths of up to 4000 meters, it might be possible to get away with even fewer time-steps, possibly 1. Since wave speeds are related to $\sqrt{depth}$, a rough calculation suggests that wave speeds will be faster by a factor of $\sqrt{4000/100} \approx 6$ in the deep ocean, implying that time-steps on *level* 3 might approach that of time-steps on *level* 2. However, wave speeds are also related to water velocity, which can be quite large near the shoreline, so generally one should be more conservative. When reducing time-step refinement, careful attention should be paid to the Courant numbers reported for each level (which can be output by selecting the `verbose` options in `amr2ez.data`) .

A.13 Memory Allocation (`bathcommon.i`*)*

Memory allocation for bathymetry and other arrays in TsunamiClaw is specified in an include file that resides in the application directory, named `bathcommon.i`. If the bathymetry is larger than the space allocated in `bathcommon.i`, then the calculation will abort and you will be instructed to increase memory. You must recompile the executable afterward, even if `make` doesn't recognize that the code has been modified. At the beginning of each computation, TsunamiClaw will output some information to the standard output about how much memory is allocated for bathymetry, and how much memory is actually used. If more memory is allocated than needed, you can abort the calculation and modify `bathcommon.i` if you wish to allocate less memory.

A.14 Data Output

TsunamiClaw uses the same output format as the standard Clawpack software. At user specified times (set in `amr2ez.data`) the entire solution on every grid is output into an ascii text file `fort.qnnnn`, where the `nnnn` ranges from 0000 for the initial time to the number of output times. Information about this file, such as the time and the number of grids, is output into `fort.tnnnn` with a corresponding number. These files conform to a standard format that can be read by Matlab graphics routines for visualization (see section A.15). The solution in `fort.qnnnn` is grouped by grids—the solution on a particular grid is written in several columns following some header information about that grid's size and location. Each column of the solution data represents a variable, and each row represents a particular grid cell. The solution data for a grid starts at the lower left corner and advances in the x-direction first (indexed by `i`). In TsunamiClaw, the solution is output as four variables in four columns: `q(i,j,1)` is the water depth, `q(i,j,2)` and `q(i,j,3)` are the momenta (depth times water velocity) in the x and y directions respectively, and `q(i,j,4)` is the surface elevation, where 0 is mean sea level. Note that having the surface elevation and the water depth allows determination of the bathymetry in a grid cell, since depth subtracted from surface elevation gives the elevation of the bathymetry or topography. Also, since TsunamiClaw solves the shallow water equations in their physically relevant conservative form in order to allow convergence to bores, momenta are output rather than the more common variable of water velocity. Velocity can be recovered by dividing the momentum by depth. Some care should be exercised when doing this, since velocities at vanishing depths may not be accurate, and division by zero for dry cells should obviously be avoided.

When using the latitude-longitude option to specify the computational domain, all output quantities are in units of meters and seconds. Therefore, in a particular grid cell, the momenta output in the solution are in the locally orthogonal longitude and latitude directions, and given in meters squared per second.

A.15 Matlab Graphics and Output Visualization

TsunamiClaw comes with a set of Matlab plotting routines,`.m` files, in the directory

`$TSUNAMICLAW/matlab`,

as well as a few routines in the example application directory. These routines are a mix of the standard Clawpack matlab plotting routines as well as several specialized routines for plotting with TsunamiClaw. If you use Clawpack software for other applications, then the `$TSUNAMICLAW/matlab` directory must be higher on your Matlab path than the standard Clawpack Matlab routines, since some modified routines share the same name. (Much like Unix itself, which uses the `PATH` variable to specify its directory search path, Matlab uses the variable, `MATLABPATH` when run under Unix, which can also be set in your login scripts.)

TsunamiClaw uses the Clawpack Matlab script `plotclaw2.m` as the main plotting routine. Run `plotclaw2` from the application directory (the file for the routine resides in the TsunamiClaw Matlab library). A number of plotting options are specified in `setplot2.m`, which resides in the application directory (see Clawpack documentation for descriptions and examples). `plotclaw2` calls `setplot2` if you select `yes` when queried from the Matlab prompt. For some specialized TsunamiClaw plotting features, set `PlotType=5` in `setplot2.m`. This creates a colored surface plot of the water surface and land, using some colormaps provided in `$TSUNAMICLAW/matlab.` Waves are shaded from red at the peaks to dark blue at the troughs. The default view is directly overhead, but note that these surface plots can be rotated and zoomed, either manually or by adjusting Matlab's `view` properties. Specific user issued plotting features, such as axis labeling, axis limits, color axis properties for the wave shading, or other options, can be specified in `afterframe.m`. The standard Clawpack documentation provides a full explanation of how to use the Matlab plotting features.

A.16 Acknowledgements

TsunamiClaw is an extension of the existing Clawpack software, including Amrclaw adaptive mesh refinement routines and Matlab plotting routines. The two-dimensional Clawpack routines were written by

Randall J. LeVeque
University of Washington.

The adaptive mesh refinement algorithms were written by

Marsha Berger
Courant Institute, NYU.

Many of the Matlab plotting routines were written by

Donna Calhoun
CEA, Saclay France.

Numerous other people have contributed to Clawpack and Amrclaw in a variety of ways. Development of this software has been supported by grants from the NSF and the DOE.

A.17 Usage Restrictions

This software is made available for research and instructional use only. You may copy and use this software for these non-commercial purposes at no charge. For all other uses (including distribution of modified versions), please contact the authors. Since TsunamiClaw is an extension of Clawpack, all of the usage restrictions and copyright notices for Clawpack apply to TsunamiClaw.

This software is made available "as is," without any assurance that it will work for your purposes. The software may have defects and is to be used at your own risk.

VITA

David L. George was born in San Francisco on February 20th, 1973. He earned Bachelor's degrees in Anthropology, Biology and Physics from the University of California, Santa Barbara. He received an M.S. in Applied Mathematics in 2004 and a Ph.D. in Applied Mathematics in 2006 from the University of Washington, Seattle.