Wave Propagation Algorithms On Curved Manifolds With Applications To Relativistic Hydrodynamics

Derek S. Bale

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

Doctor of Philosophy

University of Washington

2002

Program Authorized to Offer Degree: Department of Applied Mathematics

University of Washington
Graduate School

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

Derek S. Bale

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 Supervisory Committee:

Randall J. LeVeque

Reading Committee:

James Bardeen

Jirair Kevorkian

Peter Schmid

Date:

University of Washington

Abstract

Wave Propagation Algorithms
On Curved Manifolds With
Applications To Relativistic Hydrodynamics

by Derek S. Bale

Chair of Supervisory Committee:

Professor Randall J. LeVeque
Applied Mathematics

Many interesting physical systems are successfully modeled with time-dependent hyperbolic conservation laws on smooth manifolds. Examples of such systems include classical hydrodynamic flows in non-trivial geometry that are important in aerodynamic modeling, or the structure of shallow water flow on the surface of the earth important in climate modeling and oceanography. Another example arises in the study of relativistic flow of matter in the presence of compact objects like neutron stars and black holes. Astronomers wish to study this in order to to understand the high-energy radiation emitted from violent relativistic events such as the formation of bi-polar jets in active galactic nuclei, the collapse of very massive stars, and γ-ray bursts.

In this dissertation, we present a new high-resolution finite volume algorithm, an extension of the wave propagation method of LeVeque, to solve hyperbolic systems of conservation laws on general manifolds, $\mathscr{M}$. The representation of such equations on $\mathscr{M}$ introduces geometric terms due to the fact that the underlying basis will, in general, be spatially varying. For scalar components of the conservation laws, these geometric terms appear only in the flux function so that an autonomous flux function gains an explicit dependence on spatial variables through metric functions on $\mathscr{M}$. We develop a general approach to handle such spatially varying flux functions in the context of wave propagation.

In addition to spatially varying flux functions, vector components of the conservation laws will have geometrically induced source terms. The algorithm presented in this thesis avoids numerical complications that can arise from such source terms by treating the divergence operator in a local way. Cell averages are updated by solving modified Riemann problems. The modifications to Riemann data take care of source terms locally, thereby avoiding the need for an operator split approach for the sources.

The new algorithm has been implemented in the software package CLAWMAN, a freely available set of Fortran 90 routines. The implementation is general and can be used for different systems of conservation laws on a variety of manifolds. We demonstrate that the method works well on both flat and curved manifolds by presenting examples of the

algorithm and its implementation for both classical and relativistic conservation laws on various manifolds.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express sincere gratitude for my research advisor, Randall J. LeVeque. Randy's tireless interest in new ideas, remarkable talent for teaching difficult concepts through simple examples, and his endless well of patience are only a few of his inspirational attributes that I have greatly benefited from during my career as a graduate student. His generous allocation of funding resources have allowed me the opportunity to focus on research, present my work at numerous conferences, and interact with scientists all over the world. I am honored to have studied under him.

I am also very grateful to Jim Bardeen for his invaluable insite into many of the numerical and physical issues that came up during my research. In particular, I would like to thank Jim for teaching me so much about the general theory of relativity.

The rest of my committee members have also been very helpful to me during my time at the University of Washington. I thank Peter Schmid for his courses in numerical analysis, integral equations, and fluid dynamics. I also thank Jerry Kevorkian for his courses in partial differential equations, perturbation theory, and independent problems that he suggested for me to study. I would like to express my appreciation for James Anderson, the Graduate School Representative on my committee. He graciously joined my committee just before the defense. I would also like to thank my friends and colleagues in the Department of Applied Mathematics. In particular, I have enjoyed working with James Rossmanith and Arnold Kim.

I would also like to express gratitude and deep appreciation for my family and friends who have supported me during my time at the University of Washington. My wife Nicole has been, and will always be, the inspiration for my life and work. For this inspiration, and her loving support, I am very grateful. I am also thankful for the unyielding encouragement throughout my life from my parents. I am inspired by the strength of my Mother Robin. Throughout my studies, I have enjoyed discussing my research with my brother Brandon. I would like to thank my wife's family for all the encouragement that they have given me over the years. Particularly, I am grateful of the support that Nicole's Mother and Father, Stuart and Denise Ames, have given Nicole and I. I would like to thank my friend Max for thirty years worth of good times. Finally, I thank esoli inc. for the time necessary to finish this dissertation.

Chapter 1

INTRODUCTION

Many interesting physical systems are successfully modeled with time-dependent conservation laws on smooth manifolds, $\mathscr{M}$. Examples of such systems include hydrodynamic flows in non-trivial geometry, important in aerodynamic modeling, for example. Though in many aerodynamic applications the manifold is simply Euclidean space, $\mathscr{M} = \mathbb{R}^3$, the curvilinear basis on which one computes is often non-orthogonal and quite complicated. Another example is geophysical flow, important for climate modeling and oceanography. In this case the manifold is the surface of the earth, which is approximated by a two-sphere, $\mathscr{M} = \mathcal{S}^2$. Researchers implement a wide range of choices for the bases that cover $\mathcal{S}^2$. One particular choice is shown in Figure 1.1, for which the sphere is covered by six non-orthogonal curvilinear sets of bases to avoid the polar singularity in standard coordinates. The algorithm presented in this thesis is being applied by Rossmanith and LeVeque [50] to various problems using such a coordinate setup.

The main example used in this thesis comes from modern numerical astrophysics. The hydrodynamic accretion of gas onto compact objects such as neutron stars and black holes is a current topic of research in astronomy and astrophysics. Figure 1.2 shows an example of gas spherically accreting onto a Schwarzschild black hole. The four dimensional curved manifold in this case is a solution to the Einstein equations of general relativity. Since we focus on relativistic applications in Chapter 8, a full description follows in Section 1.2.

Consider the flow of a substance whose state variables are ρ, $\vec{s}$, and e, the density, momentum, and energy, respectively. In the absence of any sources or sinks, the time rate of change of the integral of each state variable over the volume $\mathscr{V}$ is only dependent on the flux of that variable through the boundary, $\partial\mathscr{V}$. Mathematically, this is expressed with the following integral conservation law

$$\frac{d}{dt}\int_{\mathscr{V}} \boldsymbol{q}(t,\vec{x})\, d\mathscr{V} + \int_{\partial\mathscr{V}} \boldsymbol{f}(\boldsymbol{q}) \cdot \mathrm{d}\vec{\sigma} = 0, \tag{1.1}$$

where $\boldsymbol{q}$ is a vector of the state variables, $\mathrm{d}\vec{\sigma}$ is a surface normal, and $\boldsymbol{f}$ is the corresponding flux, possibly a nonlinear function of the components of $\boldsymbol{q}$. In the remainder of this thesis, bold-face quantities will denote vectors in state-space, for example, $\boldsymbol{q} = (\rho, s^i, e)^T$, where s^i represents the i^{th} component of the vector $\vec{s}$.

The differential form of (1.1) is written

$$\partial_t \boldsymbol{q}(t) + \vec{\nabla} \cdot \boldsymbol{f}(\boldsymbol{q}) = 0, \tag{1.2}$$

and is well known to a diverse community of computational fluid dynamicists. It is also well known that the integral conservation law may have discontinuous solutions that satisfy

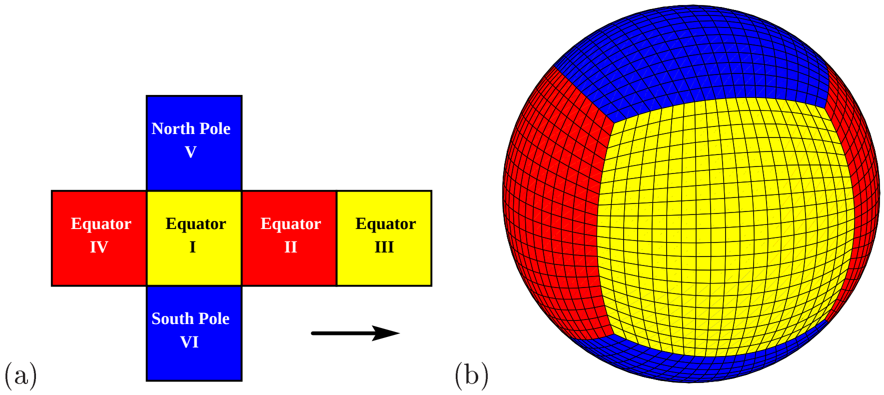


Figure 1.1: Six coordinate patches that cover the surface of a sphere, avoiding the polar singularity in standard coordinates.

a weak form of (1.2) together with an entropy condition imposed by the underlying physics of the system at hand. Finite difference numerical methods applied to such discontinuous solutions often produce non-physical oscillations in the solution. High-resolution finite volume algorithms, based on Godunov's original method [27], make use of local Riemann problems and correction fluxes that are limited to produce non oscillatory solutions with high accuracy in smooth regions. Since we are often interested in capturing discontinuous solutions of (1.1), we focus on the development of high-resolution finite volume methods.

In order to solve (1.2) numerically, it is necessary to express the conservation law in some basis. Typically, equations (1.2) are written, and solved, in a Cartesian basis $\{\vec{e}_{(1)}, \vec{e}_{(2)}, \vec{e}_{(3)}\}$. In such a basis these equations have the form

$$\frac{\partial}{\partial t}\boldsymbol{q}(t,\vec{x}) + \sum_i \frac{\partial}{\partial x^i}\boldsymbol{f}^{(i)}(\boldsymbol{q}) = 0, \tag{1.3}$$

with $i \in \{1,2,3\}$ for three dimensions, and we have introduced the notation that brackets around a subscript or superscript denote the vector along that particular coordinate direction (i.e. $\boldsymbol{f}^{(1)}$ is the flux in the x^1-direction).

Many systems of interest, however, involve complex geometry requiring that (1.2) be solved on a smooth manifold, $\mathscr{M}$, covered by a set of non-Cartesian basis vectors. In general, on a coordinate basis equation (1.2) takes the form of a balance law

$$\frac{\partial}{\partial t}\boldsymbol{q}(t,\vec{x}) + \sum_i \frac{\partial}{\partial x^i}\boldsymbol{f}^{(i)}(\vec{x},\boldsymbol{q}) = \boldsymbol{\psi}(\vec{x},\boldsymbol{q}), \tag{1.4}$$

where the flux has gained an explicit dependence on the spatial coordinates, and a geometrically induced source term has appeared. Details of the derivation of (1.4) and the forms of both the spatially varying flux and induced source terms can be found in Section 5.8.

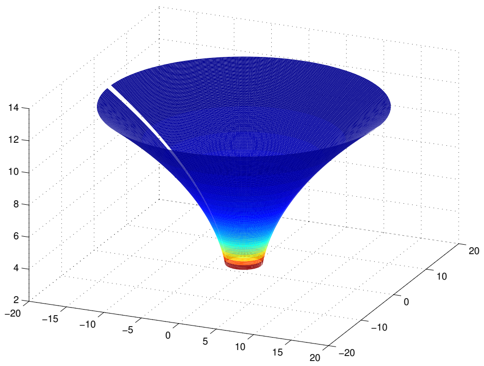


Figure 1.2: Steady-state accretion of a gas onto a Schwarzschild black hole. Plotted on an embedding diagram is the density at a late time of a typical simulation.

In this thesis I present a new unsplit high-resolution finite volume algorithm, an extension of the wave propagation method of LeVeque [42], to solve hyperbolic systems of conservation laws on general manifolds. The representation of such systems on $\mathscr{M}$, as discussed above, introduces a number of possible numerical complications due to geometric terms introduced by the representation of the divergence on a general basis. The algorithm presented here is based on both a *local* treatment of the divergence operator, and the solution of *Riemann problems* in locally orthonormal sets of bases, to avoid such numerical complications. We demonstrate that the method works well on both flat and curved manifolds by presenting examples of the algorithm for different systems of equations on various manifolds, focusing on results for a relativistic test fluid.

1.1 A class of modified wave propagation algorithms for conservation laws on curved manifolds

The representation of the divergence operator in (1.2) on a coordinate basis in $\mathscr{M}$ is responsible for both the explicit dependence of the flux on spatial coordinates, and the geometrically induced source term in (1.4). Such terms increase the computational complexity of numerical solutions and, in some cases, introduce numerical inaccuracies. The class of wave propagation algorithms developed in this thesis include the following features:

1. The algorithm Handles the spatially varying flux by defining f-waves (see Section 3.1), a set of waves that carry flux increments.

2. The method incorporates the effect of the geometrically induced source term into Riemann problems, thereby avoiding numerical complications with source terms.

3. The algorithm solves Riemann problems in sets of locally orthonormal basis vectors.

This has advantages over directly solving the Riemann problems in the spatially varying coordinate basis.

1.1.1 CLAWMAN *software package*

Successful algorithms have long been developed for hyperbolic systems in curvilinear coordinates [12], [78], and for specific systems of equations in more general manifolds [6], [20], but we are aware of no unsplit, high-resolution finite volume code able to handle different systems of equations on a variety of curved manifolds. The implementation of our modified algorithm is both easy to use, and versatile, making it useful to a broad class of users.

The original wave propagation algorithm [39], [44], [45] is implemented in the software package CLAWPACK (**C**onservation **LAW**s **PACK**age), a collection of freely available Fortran routines [41]. This software requires the user to provide, in addition to initial and boundary conditions, only the Riemann solver for the system of hyperbolic PDEs at hand. Consequently, CLAWPACK is easily applied to a variety of systems on standard Cartesian grids.

The algorithm presented in this thesis is implemented in the software package CLAWMAN (**C**onservation **LAW**s on **MAN**ifolds), a modification of CLAWPACK to handle the numerical techniques discussed throughout this dissertation. These routines are written in Fortran 90 and freely available [3]. In addition to the necessary routines for CLAWPACK, users need only provide a routine to define the metric tensor (discussed in Section 5.5), thereby defining the manifold's geometry. The package internally computes all necessary geometric information derived from the metric, such as derivatives of the metric and Christoffel symbols (i.e. see Section 5.6). Since Riemann problems are solved in sets of locally orthonormal basis vectors, approximate Riemann solvers need only be modified to handle spatially varying flux functions, as discussed in Section 3.3. That is, no geometric modifications are necessary within the Riemann solvers. We consider this to be an attractive feature of CLAWMAN. We note, however, that solving Riemann problems in locally orthonormal sets of basis vectors assumes that curvature in the manifold is well resolved on the grid being used.

1.2 Numerical relativistic astrophysics

The Einstein equations of General Relativity describe the geometric structure of spacetime. Solutions to these equations describe astrophysical phenomena such as black holes, neutron stars, supernovas, and the large-scale structure of the universe. Unfortunately, the Einstein equations are a set of very complex, highly coupled, nonlinear PDEs involving ten functions of one time-like variable and three space-like variables. Most of what is known about this set of equations comes from idealized solutions arrived at by linearizing the equations, or perturbing the equations about a solution with a high degree of symmetry. In order to investigate the full set of Einstein equations with no prescribed symmetry it is necessary to turn to numerical approximations. The field of study in which the numerical solutions to the Einstein equations are sought is called *numerical relativity*.

1.2.1 *Numerical study of relativistic problems*

In the mid 1960s, Hahn and Lindquist [29] made the first attempt to numerically evolve the collision of black holes, marking the birth of numerical relativity. Their pioneering work, however, was hindered by a poor theoretical understanding of black hole spacetimes and a bad choice of coordinates. At about the same time, May and White [59] successfully evolved the one dimensional collapse of a relativistic system with matter. Since these very early days, numerical relativity has been divided into two branches :

1. Numerical Relativistic Hydrodynamics

2. Numerical Relativistic Geometrodynamics.

The first branch, numerical relativistic hydrodynamics, is the numerical study of relativistic systems in the presence of matter (i.e. a hydrodynamic fluid). This branch will simply be called *relativistic hydrodynamics.* Numerical relativistic geometrodynamics is the numerical study of the dynamics of the gravitational field in vacuum. It is this branch that is typically referred to as *numerical relativity* and we follow this convention.

Test fluid approximation

Relativistic hydrodynamics continued to grow through the sixties with numerical calculations of supernova explosions by Wilson [81], and more investigations of relativistic collapse by May and White [60], and Matsuda and Sota [58]. Progress continued through the 1980s with calculations of nonspherical black hole accretion by Hawley and Smarr [33], [32], and relativistic stellar collapse by Nakamura and Sato [63], [64]. The nonspherical accretion problems of Hawley and Smarr were done using the *test fluid approximation.* All relativistic numerical calculations presented in this thesis have been done using this approximation.

The test fluid approximation refers to systems in which the flow of matter fields is significant, but the matter itself has little influence on the dynamics of the background gravitational fields. The gravitational field, therefore, is not changed by the flow of a test fluid. For example, a black hole accreting mass at the Eddington rate will take $\approx 10^8$ years to double its mass. The flow, in this case, has little influence on the gravitational dynamics on a relatively shorter time scale. Applications of the test fluid approximation include, but are by no means limited to, fluid accretion onto compact objects, the creation and subsequent propagation of extra-galactic jets, tidal disruption of stellar size objects in strong gravitational fields, shock acceleration of cosmic rays, and supernovas explosions. Though the test fluid approximation applies to any matter field with insufficient energy density to change the gravitational field, we concentrate on hydrodynamic matter fields in this thesis.

Today, increased computer speed and memory have allowed modern, high-resolution finite volume codes to accurately capture the flow of a test fluid for a number of astrophysical systems. The simplest test fluid is that of a *special relativistic* perfect fluid for which the background gravitational field is negligible. Riemann solvers have been developed for this set of equations by Martí and Müller [55], Balsara [5], Pons, Martí and Müller [68], and others

[17], [20], [76]. These Riemann solvers have been applied in finite volume methods for the special relativistic fluid equations by Font et al. [21], Martí et al. [54], [56], [57], Marquina et al. [52], and others. These numerical methods have been applied to extra-galactic radio jets [1], [53]. In Chapter 8, we present high-resolution numerical results for two standard test cases found in the literature. We demonstrate that the algorithm presented in this thesis performs well for a special relativistic test fluid.

In recent years, astronomers have been interested in the relativistic flow of matter in the presence of compact objects like neutron stars and black holes. In this case the background gravitational field is not negligible, but fixed. In such ultra-relativistic flows, gases emit high-energy radiation that provides astronomers with clues to the underlying physical processes taking place. A powerful way to improve the understanding of such complex flows is through accurate and stable numerical simulation. There are only a few numerical codes based on high-resolution finite volume updates that have been developed for a relativistic perfect fluid in a fixed background spacetime [6], [20], [38], [67]. The main goal of the research presented in this thesis is to develop a code for a relativistic perfect fluid in a fixed background. The novel feature of the algorithm presented is that it is unsplit and based on wave propagation. In Chapter 8, test cases from the literature are presented to demonstrate that this goal has been achieved.

1.2.2 Vacuum Spacetimes

It was not until the mid 1970s that the second branch, numerical relativity, became active again. In 1975 Eppley [19] successfully evolved the collision of two equal mass black holes in vacuum. The gravitational waves generated in systems like this should propagate to observers here on earth, but the large distance of propagation will attenuate the strength of the waves. Researchers are hard at work constructing ground based laser interferometers (*LIGO and GEO*)[77], to measure these gravitational waves. However, the very weak signal received will be noisy and it will be important to know what wave forms the Einstein equations predict to make the process of signal extraction easier. An important role of numerical relativity will be to catalog many such systems to aid in the data analysis from laser interferometry.

Progress in this branch has been slow due to the complexity of the vacuum Einstein equations. Significant challenges in numerical relativity that lead to open questions include:

- The Einstein equations can be cast in many different forms. It is not yet clear what analytical form of the equations is best suited for numerical implementation.

- There exists an intrinsic gauge freedom in general relativity resulting in a freedom in the choice of coordinates. A bad choice of coordinates will result in *coordinate shocks* that introduce numerical inaccuracies and perhaps instabilities.

- Once an analytic form of the Einstein equations has been chosen, a numerical scheme must be chosen to solve them. It is an open question as to what numerical properties are beneficial.

- Boundary conditions pose a numerical challenge both at interior boundaries near the event horizon of a black hole, and in the radiation field.

One approach in recent years has been to formulate the Einstein equations as a hyperbolic set of balance laws [9], [10], [11]. Once the equations have been cast as a hyperbolic set of balance laws, a large body of work for high-resolution finite volume methods is available. The hope is that the large gradients that develop near boundaries or due to a poor choice of coordinates will be handled in a stable way by such numerical schemes. Bardeen and Buchman are applying high-resolution wave propagation algorithms to the vacuum Einstein equations written as several different hyperbolic systems [7].

1.2.3 The full problem

We have discussed two active areas of research in relativistic astrophysics: numerical relativistic hydrodynamics (test fluids) and numerical relativity. As I mentioned earlier, these are approximations to the full problem in which gravitational fields evolve and push matter around, but the matter in motion affects the evolution of the gravitational fields. This is considerably more challenging than either problem alone.

One benefit to the hyperbolic formulation of the vacuum equations and their numerical solution is that this system can be combined with the hyperbolic equations for a test fluid. The entire system can then be evolved as one hyperbolic system of balance laws. The possible advantages of this include:

- The high-resolution capabilities of finite volume schemes will capture sharp hydrodynamic shock waves while providing additional robustness for coordinate shocks and ill-behaved boundary conditions.

- The gauge freedom in general relativity makes it difficult to impose physical boundary conditions on the correct fields. Since high-resolution finite volume schemes are based on local characteristic decompositions, there is a chance of handling such boundary conditions in a stable way.

The class of wave propagation algorithms developed in this thesis have been applied to problems with stationary metrics. These methods can be modified to handle a metric that evolves in time. Such modifications could then be combined with a hyperbolic code for the vacuum equations to solve the full problem.

1.3 Outline of this thesis

Since the methods developed here are based on the standard wave propagation algorithm, we present the standard algorithm in Chapter 2.

Chapter 3 is based on a recently completed paper [4], in which we develop high-resolution finite volume methods for hyperbolic conservation laws with spatially-varying flux functions. The calculation of f-waves is discussed, and examples are presented to demonstrate that the method works well.

Chapter 4 discusses various problems that can arise for conservation laws with source terms. We present two un-split ways to handle source terms. One, the quasi-steady method, assumes that flux gradients are being balanced by source functions. In such a situation, the solution should be near steady state. This method, however, is not very applicable to problems away from steady state. A second method that we discuss assumes nothing about such a balance between flux gradients and source terms.

We use concepts and notation from both differential geometry and general relativity in the remainder of the thesis. There exists no single standard for notation throughout these fields. Chapter 5 introduces some basic concepts we use throughout the thesis and sets the notational convention. We introduce the standard wave propagation method in Chapter 2.

In Chapter 6 we use ideas developed in Chapters 3 and 4 to develop the main algorithm presented in this dissertation. We present, in detail, the modifications to the standard wave propagation algorithm that are necessary to solve conservation laws on smooth manifolds. We present examples of systems of various manifolds to demonstrate that the method works well.

The algorithm developed in this dissertation is capable of handling many different systems of hyperbolic conservation laws on general manifolds. An implementation of the algorithm that takes advantage of this generality will be useful to a wide range of researchers. In Chapter 7 we present an implementation of the algorithm in the software package CLAWMAN.

Finally, in Chapter 8, we apply the new algorithm to the relativistic hydrodynamic equations. We develop the equations as a set of conservation laws, discuss some of the difficulties when trying to numerically solve them, and finally present examples to show that the algorithm performs well on relativistic problems.

Chapter 2

FINITE VOLUME METHODS AND THE WAVE PROPAGATION ALGORITHM

The goal of this thesis is to present an algorithm to solve hyperbolic conservation laws on smooth curved manifolds. It was stated in Chapter 1 that physical systems described by the integral conservation law (1.1) may have discontinuous solutions that satisfy a weak form of the PDE (1.2). A numerical method that will be useful for conservation laws with strong discontinuities on curved manifolds must have the following characteristics:

1. The algorithm must sharply resolve discontinuities without introducing oscillations while producing an accurate solution in smooth regions.

2. Since many of the systems of interest are nonlinear, shock speeds must be approximated accurately. It follows that physically conserved quantities must be numerically conserved.

3. Geometry will play a significant roll in the structure of the solution (i.e. in the vicinity of a black hole), so the method must handle a variety of geometries in a robust manner.

As discussed in the previous chapter, second order finite difference methods do not satisfy this set of characteristics due to the oscillations they produce near discontinuities. High resolution finite volume methods, however, have been used for decades now to produce non oscillatory solutions in the vicinity of strong discontinuities, while providing high accuracy in smooth regions of the solution [15], [27], [83]. The main contribution of this thesis is to incorporate the third characteristic, the ability to handle geometry in a robust way.

One particular class of finite volume algorithms developed by LeVeque [39], [42], [44], is based on the propagation of waves. In this chapter we introduce the wave propagation algorithm as developed in [42] and [44]. Chapters that follow extend this class of high resolution algorithms to solve hyperbolic conservation laws on general smooth manifolds.

2.1 Finite volume methods

Most finite volume methods are developed for grid volumes covered by a Cartesian basis. In order to keep consistent with notation for a general basis on manifolds discussed in Chapters 5, 6, and 8, we often denote a Cartesian basis by $\{\vec{e}_{(1)}, \vec{e}_{(2)}\}$ with $(x, y) = (x^1, x^2)$ in two spatial dimensions, or $\{\vec{e}_{(1)}, \vec{e}_{(2)}, \vec{e}_{(3)}\}$ with $(x, y, z) = (x^1, x^2, x^3)$ in three dimensions. We take $\vec{e}_{(1)} = \vec{e}_{(x)}$ and $x^1 = x$ when we discuss one-dimensional problems.

Finite volume methods are based on subdividing a computational volume into *finite volume* grid cells. At a time t_n, components of $\boldsymbol{q}$ are averaged onto the grid resulting in a

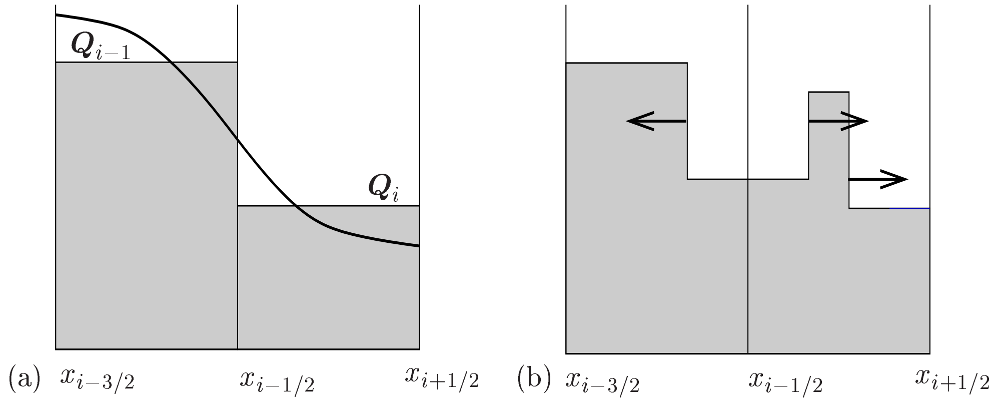


Figure 2.1: (a) The result of averaging $\boldsymbol{q}$ onto a one-dimensional finite volume grid. (b) Waves emanate from finite volume cell edges.

piece-wise constant representation of each state variable. Let $\boldsymbol{Q}_i^n$ represent an approximation to the cell average

$$\boldsymbol{Q}_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(t_n, x)\, dx \equiv \frac{1}{\Delta x} \int_{\mathcal{C}_i} q(t_n, x)\, dx, \tag{2.1}$$

where we denote the one-dimensional finite volume cell of length $\Delta x = x_{i+1/2} - x_{i-1/2}$ by

$$\mathcal{C}_i = (x_{i-1/2}, x_{i+1/2}). \tag{2.2}$$

The finite volume approximation, $\boldsymbol{Q}_i^n$, of a smooth function is shown in Figure 2.1(a).

At the beginning of each time step, the piece-wise constant representation $\boldsymbol{Q}_i^n$ sets up unsteady *Riemann* problems at each grid cell boundary. These Riemann problems decay, inducing flux through cell boundaries throughout the time step. Figure 2.1(b) shows the cell boundary at $x_{i-1/2}$ where the initial discontinuity $\boldsymbol{Q}_i^n - \boldsymbol{Q}_{i-1}^n$ has decayed producing flux through the cell boundary. Directly computing the averaged solution at the end of the time step may be complicated due to the fact that information traveling from the left edge of the cell may be interacting with the information traveling from the right edge. This interaction will in general be non-trivial for nonlinear systems.

Instead of directly averaging the solution at the end of the time step, we integrate the one-dimensional version of equation (1.1) in time from t_n to $t_{n+1} = t_n + \Delta t$ to get

$$\begin{aligned} \int_{\mathcal{C}_i} \boldsymbol{q}(t_{n+1}, x)\, dx = \int_{\mathcal{C}_i} \boldsymbol{q}(t_n, x)\, dx \\ - \left(\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{q}(t_n, x_{i+1/2}))\, dt - \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{q}(t_n, x_{i-1/2}))\, dt \right). \end{aligned} \tag{2.3}$$

This relates the integral of $\boldsymbol{q}$ over $\mathcal{C}_i$ at t_{n+1}, the end of the time step, to what it was at t_n modified by the time integral of the flux at each cell boundary. With a good approximation to the *numerical flux* defined by

$$\boldsymbol{F}^n_{i-1/2} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{f}\left(\boldsymbol{q}(t, x_{i-1/2})\right) dt, \tag{2.4}$$

an explicit numerical method follows from (2.3). Many different numerical flux functions can be found in the literature [17], [18], [27], [69]. We discuss approximations to numerical flux functions in Appendix A. Using the numerical flux (2.4), together with (2.1), we can write (2.3) as

$$\boldsymbol{Q}^{n+1}_i = \boldsymbol{Q}^n_i - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}^n_{i+1/2} - \boldsymbol{F}^n_{i-1/2}\right), \tag{2.5}$$

which is the typical form of a finite volume numerical method based on flux differences.

The numerical method (2.5) has the property that if

$$\boldsymbol{F}^n_{i+1/2} = \boldsymbol{F}^n_{i-1/2}, \tag{2.6}$$

meaning that what is flowing in one side of the cell is flowing out the other side, then the finite volume approximation $\boldsymbol{Q}^n_i$ remains unchanged over the time step. In fact, update (2.5) exactly conserves $\boldsymbol{Q}^n_i$ at each time step since on a grid with N cells we have

$$\begin{aligned}\sum_{i=1}^{N}\boldsymbol{Q}^{n+1}_i &= \sum_{i=1}^{N}\boldsymbol{Q}^n_i - \frac{\Delta t}{\Delta x}\left(-\boldsymbol{F}^n_{1-1/2} + \text{telescoping terms} + \boldsymbol{F}^n_{N+1/2}\right)\\ &= \sum_{i=1}^{N}\boldsymbol{Q}^n_i - \frac{\Delta t}{\Delta x}\left(-\boldsymbol{F}^n_{1-1/2} + \boldsymbol{F}^n_{N+1/2}\right),\end{aligned} \tag{2.7}$$

which states that the numerical solution over a time step is only affected by the flux at grid boundaries. This is a very desirable discrete representation of the conservative properties of the conservation law (1.1).

2.2 *The Riemann problem*

In one space dimension with $x^1 = x$, the conservation law (1.3) has the form

$$\boldsymbol{q}_{,t} + \boldsymbol{f}(\boldsymbol{q})_{,x} = 0. \tag{2.8}$$

A Riemann problem consists of this hyperbolic PDE together with piecewise constant initial data having a single jump discontinuity at $x = 0$,

$$\boldsymbol{q}(t=0, x) = \begin{cases} \boldsymbol{q}_l & \text{if } x < 0 \\ \boldsymbol{q}_r & \text{if } x > 0. \end{cases} \tag{2.9}$$

The jump discontinuity can be placed at any $x = X$, but without loss of generality we take $X = 0$. When f is a linear function of q the initial discontinuity breaks up and travels along characteristics of the equation. The solution involves piece-wise constant states separated by waves carrying part of the original jump discontinuity. The solution to a Riemann problem for a general constant-coefficient linear system is presented in Section 2.2.1. We are interested in the linear solution because nonlinear systems are often locally linearized, resulting in a linear Riemann problem.

Solutions to Riemann problems for equations with nonlinear flux functions are more complicated. The solution still consists of piece-wise constant states separated by waves, but the waves can be genuinely nonlinear discontinuities (shock waves), smooth rarefaction waves, or linearly degenerate discontinuities (contact waves). We briefly describe the nonlinear waves in Section 2.2.2. Many more details about both linear and nonlinear systems with examples can be found in [43], [46], [48].

2.2.1 Linear systems

Consider a linear flux function

$$\boldsymbol{f}(\boldsymbol{q}) = A\,\boldsymbol{q}, \tag{2.10}$$

where $A \in \mathbb{R}^{n_{eq}\times n_{eq}}$ is a diagonalizable constant coefficient matrix with real eigenvalues. Using this flux function in equation (2.8) we get

$$\boldsymbol{q}_{,t} + A\,\boldsymbol{q}_{,x} = 0. \tag{2.11}$$

This linear PDE is hyperbolic and since A is diagonalizable, we can write $\mathbf{A}_x = \mathbf{R}\Lambda\mathbf{R}^{-1}$, where

$$R = \left(\boldsymbol{r}^1\middle|\cdots\middle|\boldsymbol{r}^{n_{eq}}\right), \quad \Lambda = \mathrm{diag}(\lambda^1, \cdots, \lambda^{n_{eq}}), \quad R^{-1} = \begin{pmatrix} \boldsymbol{l}^1 \\ - \\ \vdots \\ - \\ \boldsymbol{l}^{n_{eq}} \end{pmatrix}. \tag{2.12}$$

The right eigenvectors, $\boldsymbol{r}^i$, and left eigenvectors, $\boldsymbol{l}^j$, satisfy the relationship $\boldsymbol{l}^j \cdot \boldsymbol{r}^i = \delta^{ij}$, and we order the eigenvalues such that $\lambda^1 \leq \lambda^2 \leq \cdots \leq \lambda^{n_{eq}}$.

Now consider a Riemann problem for the system (2.11) with the discontinuity placed at $X = 0$. Since the system is linear, the discontinuity will propagate along characteristics of (2.11). Transforming to *characteristic* variables defined by

$$\boldsymbol{w} = R^{-1}\,\boldsymbol{q}, \tag{2.13}$$

we can write equation (2.11) as

$$\boldsymbol{w}_{,t} + \Lambda\,\boldsymbol{w}_{,1} = 0, \tag{2.14}$$

which is a set of uncoupled advection equations for the components of $\boldsymbol{w}$. We also project both $\boldsymbol{q}_l$ and $\boldsymbol{q}_r$ onto the set of right eigenvectors so that

$$\begin{aligned} \boldsymbol{q}_l &= R\,\boldsymbol{w}_l = \sum_{p=1}^{n_{eq}} w_l^p \boldsymbol{r}^p \\ \boldsymbol{q}_r &= R\,\boldsymbol{w}_r = \sum_{p=1}^{n_{eq}} w_r^p \boldsymbol{r}^p, \end{aligned} \tag{2.15}$$

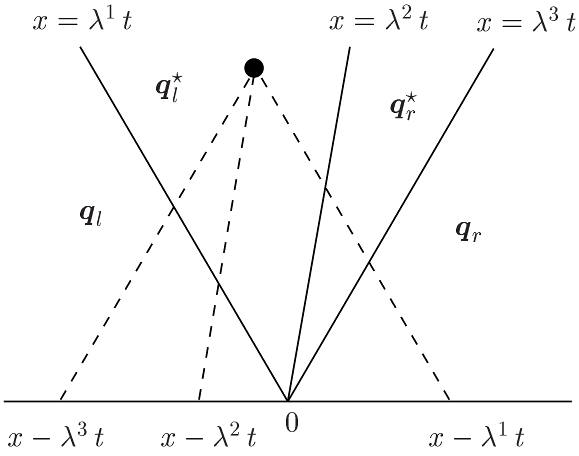


Figure 2.2: The solution for linear problems is a superposition of waves. Waves separate piece-wise constant regions of the solution.

and the initial condition for the p^{th} advection equation becomes

$$w^p(t=0,x) = \begin{cases} w_l^p & \text{if } x < 0 \\ w_r^p & \text{if } x > 0. \end{cases} \tag{2.16}$$

Each advection equation in (2.14) with initial data (2.16) has the solution

$$w^p(t,x) = w_l^p\,\mathcal{H}(\lambda^p t - x) + w_r^p\,\mathcal{H}(x - \lambda^p t), \tag{2.17}$$

where $\mathcal{H}(x)$ is the Heaviside function. If we define $\mathscr{P}^-$ to be the set of all p such that $\lambda^p < 0$, and similarly define $\mathscr{P}^+$ to be the set such that $\lambda^p > 0$, then we can transform (2.17) back using $\boldsymbol{q} = R\,\boldsymbol{w}$ to find

$$\boldsymbol{q}(t,x) = \sum_{p\in\mathscr{P}^-} w_r^p\,\boldsymbol{r}^p + \sum_{p\in\mathscr{P}^+} w_l^p\,\boldsymbol{r}^p, \tag{2.18}$$

which is the desired result. The solution is a superposition of waves traveling along characteristics of (2.11).

Figure 2.2 illustrates the construction of the solution (2.18) at a particular point in the spacetime domain. Denote the solution at this point by $\boldsymbol{q}_l^\star$ and trace back along characteristics to see what part of the initial condition affects the solution. In the figure we see that

$$\begin{aligned} \boldsymbol{q}_l^\star &= w_r^1\,\boldsymbol{r}^1 + w_l^2\,\boldsymbol{r}^2 + w_l^3\,\boldsymbol{r}^3 \\ &= \boldsymbol{q}_l + \left(w_r^1 - w_l^1\right)\,\boldsymbol{r}^1, \end{aligned} \tag{2.19}$$

so $\boldsymbol{q}_l^\star$ can be viewed as the left state modified by a left-propagating wave $\boldsymbol{\mathcal{W}}^1 = (w_r^1 - w_l^1)\,\boldsymbol{r}^1$.

It is worth noting that the Riemann solution is constant and equal to $\boldsymbol{q}_l^\star$ along the ray $x/t = 0$ in Figure 2.2. If this were the edge of a finite volume grid cell, then a very good numerical flux function $\boldsymbol{F}^n$ from (2.4) would be

$$\boldsymbol{F}^n = \boldsymbol{f}(\boldsymbol{q}_l^\star) = \boldsymbol{f}(\boldsymbol{q}_l + \boldsymbol{\mathcal{W}}^1). \tag{2.20}$$

In this special case $\boldsymbol{F}^n$ is not an approximation to the time average of the flux, but the exact value. Defining the numerical flux as the flux function evaluated at the exact solution to the Riemann problem leads to the Godunov method [27].

2.2.2 Nonlinear systems

As previously discussed, the solution to the Riemann problem for nonlinear flux functions is more complicated than the linear case. The waves generated by the decay of the initial discontinuity may be shock waves, rarefaction waves, or contact discontinuities. The type of wave generated depends on the local characteristics, which for nonlinear problems depend on the solution itself.

Consider a scalar version of equation (2.8) so that $n_{eq} = 1$. As long as $q(t, x)$ is smooth, characteristic curves $X(t)$ are defined by the quasi-linear form

$$q_{,t} + f^{'}(q)\, q_{,x} = 0. \tag{2.21}$$

These curves are defined so that $X^{'}(t) = f^{'}(q(t, X(t)))$ which leads to the fact that for (2.21) q is constant along such curves because

$$\begin{aligned} \frac{d}{dt} q(t, X(t)) &= q_{,t} + X^{'}(t)\, q_{,x} \\ &= q_{,t} + f^{'}(q)\, q_{,x} \\ &= 0. \end{aligned} \tag{2.22}$$

If q is constant, then $f^{'}(q)$ is also constant so that $X^{'}(t) =$ const. and the characteristics are straight lines.

Shock waves

Suppose $f^{'}(q)$ is a monotonically increasing function of q so that the initial conditions in Figure 2.3(a) would have $f^{'}(q_l) > f^{'}(q_r)$. In this case, the characteristics to the left and right of the initial discontinuity would be disappearing into the line $x = st$ shown in Figure 2.3(b), where s is the shock speed given by the *Rankine-Hugoniot* conditions

$$s = \frac{f(q_r) - f(q_l)}{q_r - q_l}. \tag{2.23}$$

Rarefaction waves

Initial conditions as depicted in Figure 2.4(a), however, have $f^{'}(q_l) < f^{'}(q_r)$ so that the characteristics spread as shown in Figure 2.4(b). In this case the solution remains smooth.

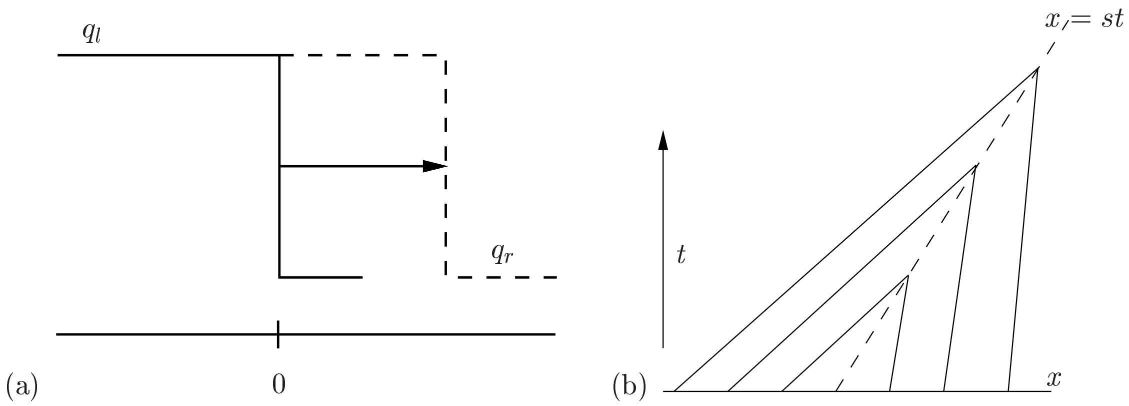


Figure 2.3: (a) The function q is constant along characteristics. (b) The initial discontinuity in q travels at speed s. Since $f'(q_l) > f'(q_r)$ the characteristics disappear into the line $x = st$.

For systems, the quasi-linear form of the conservation law becomes

$$\boldsymbol{q}_{,t} + A^k(\boldsymbol{q})\boldsymbol{q}_{,k} = 0, \tag{2.24}$$

where $A^k \to A^k(\boldsymbol{q}) \equiv \partial \boldsymbol{f}^{(k)}/\partial \boldsymbol{q}$ is now the Jacobian of the flux function $\boldsymbol{f}^{(k)}(\boldsymbol{q}) : \mathbb{R}^{n_{\text{eq}}} \to \mathbb{R}^{n_{\text{eq}}}$. Throughout this thesis, as for the linear case, we assume hyperbolicity so that each A^k has a complete set of linearly independent eigenvectors $\{\boldsymbol{r}^1, \cdots, \boldsymbol{r}^{n_{eq}}\}$ associated with which is an ordered set of real eigenvalues $\{\lambda^1, \cdots, \lambda^{n_{eq}}\}$ at each point of the solution domain.

For many nonlinear systems an exact solution to the Riemann problem like (2.18) is either not possible, or too computationally expensive. In that case an accurate and efficient approximate solution is desirable. These approximate solutions produce a wide range of numerical flux functions $\boldsymbol{F}^n$ [5], [17], [69]. We explore several approximate Riemann solvers in Appendix A used to compute numerical results that we show in most of the following chapters. For more detail about nonlinear systems, approximate Riemann solvers, and examples of nonlinear systems the reader is referred to [26], [46], [74].

2.3 Standard wave propagation

The wave propagation algorithm is an unsplit multidimensional finite volume method used to solve hyperbolic conservation laws on a Cartesian basis. So far, we have discussed finite volume methods in general, and how to use the solution of Riemann problems that arise at each cell edge to approximate the flux at these edges. We now describe the wave propagation algorithm as developed by LeVeque in [44] and [42] and how it fits into the class of finite volume methods (2.5) based on flux differences.

We begin by presenting in Section 2.3.1 the one-dimensional algorithm and show how the numerical update is constructed from waves like those discussed in Section 2.2. It is also

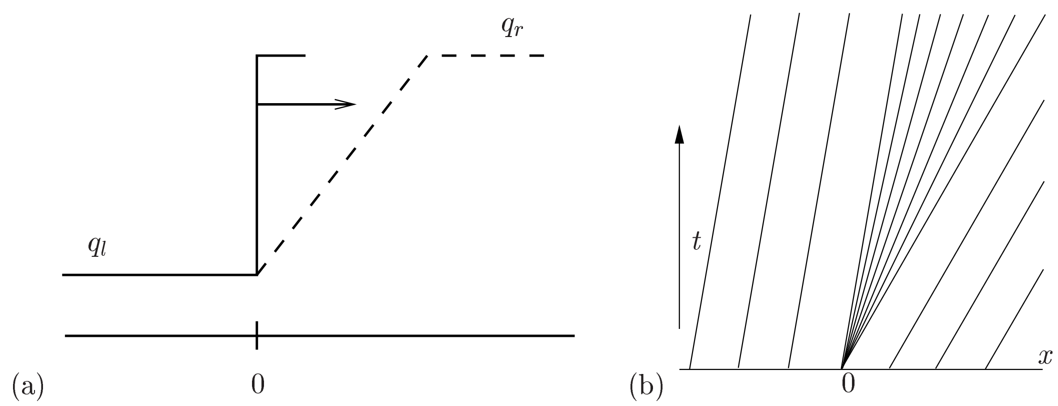


Figure 2.4: (a) The function q is still constant along characteristics. (b) The initial discontinuity in q now decomposes into a continuously varying rarefaction wave. The characteristics fan out from the location of the initial discontinuity.

shown that the update is numerically conservative. We then consider the multidimensional algorithm in Section 2.3.2.

2.3.1 One dimensional algorithm

Data at time t_n in each cell, $\mathcal{C}_i$, is represented by piece-wise constant functions $\boldsymbol{Q}_i^n$, the finite volume average (2.1). These averages define one-dimensional planar Riemann problems at each cell interface. A set of n_{eq} waves $\{\boldsymbol{\mathcal{W}}_{i-1/2}^1, \cdots \boldsymbol{\mathcal{W}}_{i-1/2}^{n_{eq}}\}$ is generated in each Riemann problem that carry the entire jump in $\boldsymbol{Q}_i^n$

$$\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1} = \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{W}}_{i-1/2}^{(p)}. \tag{2.25}$$

Each wave has an associated speed for which $s_{i-1/2}^1 < \cdots < s_{i-1/2}^{n_{eq}}$. As discussed in Section 2.2.2, the waves may be shock waves, rarefaction waves, or contact discontinuities, depending on the local characteristics.

Fluctuations defined by

$$\boldsymbol{\mathcal{A}}^{\pm}\boldsymbol{\Delta Q}_{i-1/2} = \sum_p \left[s_{i-1/2}^p\right]^{\pm} \boldsymbol{\mathcal{W}}_{i-1/2}^p, \tag{2.26}$$

are used in the cell update. In equation (2.26) we have followed the notation of [45] and used $s^+ = \max(0, s)$, $s^- = \min(s, 0)$. Figure 2.5 shows a finite volume cell being influenced by a fluctuation $\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2}$ traveling to the right from the left edge of the cell, and $\boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i+1/2}$ traveling to the left from the right edge.

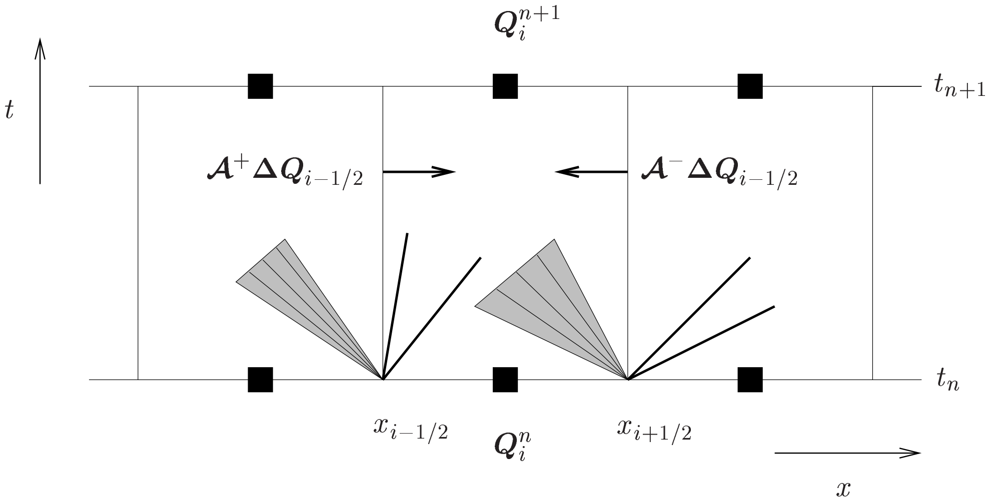


Figure 2.5: Wave generation of fluctuation differences

We motivate the use of fluctuations by considering the Taylor expansion of the constant coefficient advection equation

$$q_{,t} + u\, q_{,x} = 0, \tag{2.27}$$

which has the form

$$q(t+\Delta t) = q(t) - \Delta t \underbrace{u\, q_{,x}(t)}_{I} + \frac{1}{2}\Delta t^2 \underbrace{u^2\, q_{,xx}(t)}_{II(a)} + \mathcal{O}(\Delta t^3). \tag{2.28}$$

In order to have a second order method for smooth solutions we must numerically approximate two types of terms indicated in the expansion (2.28). Type I terms are first order and we refer to them as *upwind* terms. Type II(a) terms involve the second derivative of q and will be approximated by *correction* terms.

Type I terms: upwind

Upwind contributions approximate type I terms by using the influence of waves generated in Riemann problems normal to each cell edge. A wave that emanates from the edge at $x = x_{i-1/2}$ travels a distance $u\,\Delta t$ into the upwind cell over one time step. The influence of this wave on the upwind cell average is computed by multiplying the wave strength $\boldsymbol{\mathcal{W}}_{i-1/2}$ by the fraction ν, of the cell volume that the wave occupies at the end of the time step

$$\nu = \frac{u\,\Delta t}{\Delta x}. \tag{2.29}$$

The upwind cell average (for $u > 0$) is modified by

$$\begin{aligned} \boldsymbol{Q}_i^{n+1} &= \boldsymbol{Q}_i^n - \frac{u\,\Delta t}{\Delta x}\boldsymbol{\mathcal{W}}_{i-1/2} \\ &= \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x}\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2}. \end{aligned} \tag{2.30}$$

Using the definition (2.26) to incorporating the influence of multiple waves traveling in both directions as depicted in Figure 2.5, we write the standard first-order upwind method

$$\begin{aligned} \boldsymbol{Q}_i^{n+1} &= \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x}\left(\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2} + \boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i+1/2}\right) \\ &= \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x}\boldsymbol{\Delta}_i. \end{aligned} \tag{2.31}$$

We use $\boldsymbol{\Delta}_i$ to denotes the influence of first-order fluctuations in $\mathcal{C}_i$

$$\boldsymbol{\Delta}_{\mathcal{C}} = \boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2} + \boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i+1/2}. \tag{2.32}$$

Conservation of type I terms

It was shown in Section 2.1 that a property of high resolution finite volume methods in flux difference form (2.5) is that they numerically conserve the $\boldsymbol{Q}_i^n$ that approximate components of $\boldsymbol{q}$. We have discussed the importance of maintaining such a conservative update. It is, therefore, necessary to show that the wave propagation update (2.32) can be cast into the flux difference form (2.5).

So far we have described how the waves that are generated in Riemann problems affect cell averages, but we have said little about the generation of the waves and the speeds at which they travel. If, for example, we are working with a nonlinear system, then waves strengths may be found by intersecting Hugoniot loci and integral curves, while adhering to physical entropy conditions. An exact solution, $\boldsymbol{Q}^\star_{i-1/2}$ along $x = x_{i-1/2}$ can be constructed using this information. Wave speeds can then be computed based on shock speeds or characteristic speeds, depending on the type of wave. As discussed in Section 2.2.1, with $\boldsymbol{Q}^\star_{i-1/2}$ known a natural choice for a numerical flux function would be $\boldsymbol{F}^n_{i-1/2} = \boldsymbol{f}(\boldsymbol{Q}^\star_{i-1/2})$. Conservation can then be guaranteed by defining

$$\boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i-1/2} = \boldsymbol{F}^n_{i-1/2} - \boldsymbol{f}(\boldsymbol{Q}^n_{i-1}) \tag{2.33}$$

$$\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2} = \boldsymbol{f}(\boldsymbol{Q}^n_{i}) - \boldsymbol{F}^n_{i-1/2}, \tag{2.34}$$

so that across any interface the numerical flux exactly cancels and the exact flux difference remains.

If, however, the exact solution is too complicated, or too computationally expensive to compute, one may define a Jacobian matrix $A_{i-1/2}$ that linearizes the flux Jacobian at each cell interface. The nonlinear problem is then replaced by a linear one with a solution discussed in Section 2.2.1. Using a linearized Jacobian will yield a conservative method only if the condition

$$
\begin{aligned}
A_{i-1/2}\left(\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1}\right) = \sum_{p=1}^{n_{eq}} \alpha^p_{i-1/2}\, A_{i-1/2}\, \boldsymbol{r}^p_{i-1/2} \\
= \sum_{p=1}^{n_{eq}} \boldsymbol{s}^p_{i-1/2}\, \boldsymbol{\mathcal{W}}^p_{i-1/2} \\
= \boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}),
\end{aligned}
\tag{2.35}
$$

where $\boldsymbol{r}^p$ are the eigenvectors and $\boldsymbol{s}^p_{i-1/2}$ are the eigenvalues of $A_{i-1/2}$ and the waves $\boldsymbol{\mathcal{W}}^p_{i-1/2} = \alpha^p\, \boldsymbol{r}^p$ with speeds $\boldsymbol{s}^p_{i-1/2}$ have been defined. This condition is imposed by defining a Jacobian matrix based on a Roe average defined in [69]. In this case when waves for a Riemann solution consisting of a single shock for a nonlinear scalar equation satisfy both (2.25) and (2.35), the Rankine-Hugoniot condition (2.23) is satisfied exactly as we would expect.

Conservation of type I terms in this case can be realized by defining a numerical flux denoted by $\boldsymbol{F}^n$ from (2.4) that approximates the time integral of the true flux $\boldsymbol{f}$ along a cell interface during a single time step. The numerical flux for the wave propagation algorithm at the left edge of $\mathcal{C}_i$ is written

$$
\boldsymbol{F}^n_{i-1/2} = \frac{1}{2}\left(\boldsymbol{f}(\boldsymbol{Q}_i) + \boldsymbol{f}(\boldsymbol{Q}_{i-1}) - \sum_{p=1}^{n_{eq}} |s^p_{i-1/2}| \boldsymbol{\mathcal{W}}^p_{i-1/2}\right). \tag{2.36}
$$

Using this definition of the numerical flux and the fluctuations (2.26) we can write

$$
\begin{aligned}
\boldsymbol{\Delta}_i &= \sum_p \left[s^p_{i-1/2}\right]^+ \boldsymbol{\mathcal{W}}^p_{i-1/2} + \sum_p \left[s^p_{i+1/2}\right]^- \boldsymbol{\mathcal{W}}^p_{i+1/2} \\
&= \sum_{p=1}^{n_{eq}} \frac{1}{2}\left(s^p_{i-1/2} + |s^p_{i-1/2}|\right) \boldsymbol{\mathcal{W}}^p_{i-1/2} + \sum_{p=1}^{n_{eq}} \frac{1}{2}\left(s^p_{i+1/2} - |s^p_{i+1/2}|\right) \boldsymbol{\mathcal{W}}^p_{i+1/2} \\
&= \boldsymbol{F}^n_{i+1/2} - \boldsymbol{F}^n_{i-1/2}.
\end{aligned}
\tag{2.37}
$$

The upwind update (2.31) for the wave propagation algorithm can then be written

$$
\boldsymbol{Q}^{n+1}_i = \boldsymbol{Q}^n_i - \sum_{k=1}^{n_d} \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}^n_{i+1/2} - \boldsymbol{F}^n_{i-1/2}\right), \tag{2.38}
$$

which is in flux difference conservative form.

As long as waves are defined so that equation (2.35) holds, the wave propagation algorithm will generate conservative numerical solutions. We will also demonstrate that higher order corrections to these type I terms are also flux conservative, yielding a high-resolution flux conservative method when the definition of a flux function is appropriate.

Type II(a) terms: high resolution corrections

In the previous section, we demonstrated that the upwind contribution to the cell update can be viewed as computing the influence a wave has on a cell volume. Consider equation (2.27) with $u > 0$. The upwind contribution to the i^{th} cell average in this case was found to be

$$\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x} u \boldsymbol{\mathcal{W}}_{i-1/2}. \tag{2.39}$$

Taking into account equation (2.25) we can write

$$-\frac{\Delta t}{\Delta x} u \boldsymbol{\mathcal{W}}_{i-1/2} = u \frac{\boldsymbol{Q}_i^n - \boldsymbol{Q}_{i-1}^n}{\Delta x} \tag{2.40}$$

which is a one-sided and first order accurate approximation to $u_{i-1/2} q_{,x}$. In order to capture higher order terms and produce a more accurate algorithm in regions of smooth solutions, the upwind contribution $\boldsymbol{\Delta}_i$ must be modified. However, the modifications must not destroy the conservative properties demonstrated in Section 2.3.1.

A correction flux is defined by

$$\tilde{\boldsymbol{F}}_{i-1/2} = \frac{1}{2} \sum_{p=1}^{n_{eq}} |s_{i-1/2}^p| \left(1 - \frac{\Delta t}{\Delta x} |s_{i-1/2}^p|\right) \tilde{\boldsymbol{\mathcal{W}}}_{i-1/2}^p \tag{2.41}$$

and implemented in a flux conservative way. The limited waves

$$\tilde{\boldsymbol{\mathcal{W}}}_{i-1/2}^p = \text{Limiter}(\boldsymbol{\mathcal{W}}_{i-1/2}^{(p)}, \boldsymbol{\mathcal{W}}_{I-1/2}^{(p)}), \tag{2.42}$$

have been introduced in the correction flux where

$$I = \begin{cases} i-1 & \text{if } s_{i-1/2}^p > 0 \\ i+1 & \text{if } s_{i-1/2}^p < 0 \end{cases}. \tag{2.43}$$

The wave-limiting function introduced here reduces the strength of the correction fluxes in regions of sharp gradients such as shock waves and contact discontinuities. Reducing the influence of higher order corrections limits well known oscillations produced by second order methods near such gradients. Details about the limiter functions used in calculations presented here can be found in [45], [46].

The modified update to the cell average can now be written

$$\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x} \left(\boldsymbol{\Delta}_i + \tilde{\boldsymbol{F}}_{i+1/2} - \tilde{\boldsymbol{F}}_{i-1/2}\right). \tag{2.44}$$

The first term of this correction flux improves accuracy on the upwind approximation to $u\, q_{,x}$ by centralizing the difference. The second term gives a second-order accurate representation of type II(a) terms that look like $u^2\, q_{,xx}$. If these terms were not captured by the method then the numerical solution would smear the results. Finally, note that the high resolution correction fluxes introduced here are implemented in a flux difference so that the conservative properties of the update are kept.

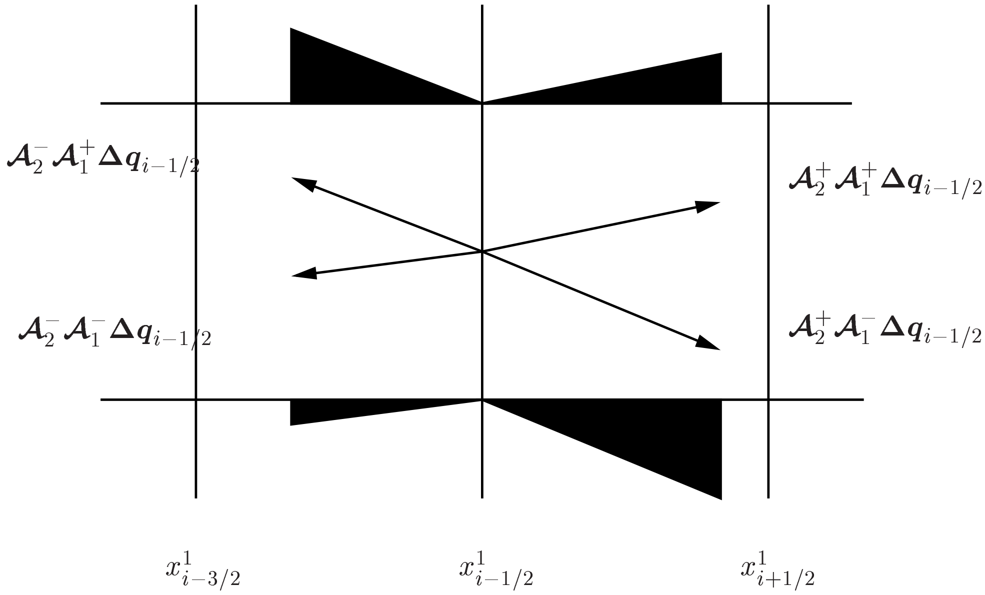


Figure 2.6: (a) A wave propagates in a multidimensional way that affects more cells than just the adjacent cells. (b) The correction triangles that result from the multidimensional propagation of waves.

2.3.2 Multidimensional algorithm

So far we have concentrated on one-dimensional contributions to the cell average. Many high resolution finite volume schemes base multidimensional cell updates on purely one-dimensional contributions and rely on dimensionally split time evolution to incorporate multidimensional effects. In reality the waves propagate in a multidimensional way and affect not only cell averages adjacent to the interface, but also any cell within the causal path of the waves over a time step. The influence of a wave traveling through a grid is depicted in figure 2.6 with all cells adjacent the the cell interface being affected. The standard wave propagation algorithm is an unsplit method that directly incorporates the effect of waves propagating in a multidimensional manner.

Consider the Taylor series of the multidimensional version of equation (2.27)

$$q_{,t} + u^k\, q_{,k} = 0, \tag{2.45}$$

which has the form

$$q(t+\Delta t) = q(t) - \Delta t\, \underbrace{u^k\, q_{,k}(t)}_{I} + \frac{1}{2}\Delta t^2\, \underbrace{u^k\, u^j\, q_{,kj}(t)}_{II(a)\text{ and }II(b)} + \mathcal{O}(\Delta t^3). \tag{2.46}$$

A new type of term, denoted II(b), appears due to cross derivatives at second order. When $k = j$ the second order term is purely one dimensional and of Type I(a) that was discussed in the previous section. In this section we concentrate on terms of Type II(b) with $k \neq j$.

Type II(b) terms: Transverse propagation

Suppose that the wave travels through a grid with velocity $\vec{v} = u^1\vec{e}_{(1)} + u^2\vec{e}_{(2)}$ and $u^1, u^2 > 0$. The triangular portion of the wave that is entering cell $\mathcal{C}_{i,j+1}$ rather than $\mathcal{C}_{i,j}$ modifies the correction flux between these cells by

$$\tilde{\boldsymbol{F}}^{(2)}_{i,j+1/2} \rightarrow \tilde{\boldsymbol{F}}^{(2)}_{i,j+1/2} - \frac{\Delta t}{2\,\Delta\, x^1}\, u^1\, u^2 \left(\boldsymbol{Q}^n_{ij} - \boldsymbol{Q}^n_{i-1j}\right). \tag{2.47}$$

Notice that we once again use the notation that an subscript index or a superscript index with braces around it indicates a spatial vector in that direction (i.e. $\tilde{\boldsymbol{F}}^{(2)}_{i,j}$ is the correction flux for cell $\mathcal{C}_{i,j}$ in the x^2 direction). This modification can be viewed as decomposing the right-going fluctuation into fluctuations traveling in the transverse direction.

A system of equations typically has both a left-going fluctuation $\boldsymbol{\mathcal{A}}^-_k \boldsymbol{\Delta Q}_{ij}$, and a right-going fluctuation $\boldsymbol{\mathcal{A}}^+_k \boldsymbol{\Delta Q}_{ij}$ for a sweep through the grid in the x^k direction. Each of these will be split into an up-going transverse piece $\boldsymbol{\mathcal{A}}^+_2 \boldsymbol{\mathcal{A}}^*_1 \boldsymbol{\Delta q}_{ij}$ and a down-going transverse piece $\boldsymbol{\mathcal{A}}^-_2 \boldsymbol{\mathcal{A}}^*_1 \boldsymbol{\Delta q}_{ij}$. These transverse fluctuations create four triangular corrections modifying the cell updates in the four neighboring cells as indicated in figure 2.6. In general, the transverse fluctuations are used to modify the four neighboring fluxes according to

$$\tilde{\boldsymbol{F}}^{(2)}_{i,j+1/2} \rightarrow \tilde{\boldsymbol{F}}^{(2)}_{i,j+1/2} - \frac{1}{2}\frac{\Delta t}{\Delta x^1} \boldsymbol{\mathcal{A}}^+_2 \boldsymbol{\mathcal{A}}^+_1 \boldsymbol{\Delta q}_{i-1/2} \tag{2.48}$$

$$\tilde{\boldsymbol{F}}^{(2)}_{i,j-1/2} \rightarrow \tilde{\boldsymbol{F}}^{(2)}_{i,j-1/2} - \frac{1}{2}\frac{\Delta t}{\Delta x^1} \boldsymbol{\mathcal{A}}^-_2 \boldsymbol{\mathcal{A}}^+_1 \boldsymbol{\Delta q}_{i-1/2} \tag{2.49}$$

$$\tilde{\boldsymbol{F}}^{(2)}_{i-1,j+1/2} \rightarrow \tilde{\boldsymbol{F}}^{(2)}_{i-1,j+1/2} - \frac{1}{2}\frac{\Delta t}{\Delta x^1} \boldsymbol{\mathcal{A}}^+_2 \boldsymbol{\mathcal{A}}^-_1 \boldsymbol{\Delta q}_{i-1/2} \tag{2.50}$$

$$\tilde{\boldsymbol{F}}^{(2)}_{i-1,j-1/2} \rightarrow \tilde{\boldsymbol{F}}^{(2)}_{i-1,j-1/2} - \frac{1}{2}\frac{\Delta t}{\Delta x^1} \boldsymbol{\mathcal{A}}^-_2 \boldsymbol{\mathcal{A}}^-_1 \boldsymbol{\Delta q}_{i-1/2}, \tag{2.51}$$

so that the form of the full high resolution multidimensional update is

$$\boldsymbol{Q}^{n+1}_{\mathcal{C}} = \boldsymbol{Q}^n_{\mathcal{C}} - \sum_{k=1}^{n_d} \frac{\Delta t}{\Delta x^k} \left(\boldsymbol{\Delta}^k_{\mathcal{C}} + \tilde{\boldsymbol{F}}^{(k)}_{i+1/2} - \tilde{\boldsymbol{F}}^{(k)}_{i-1/2}\right) \tag{2.52}$$

where we denote the finite volume cell by $\mathcal{C}$, the sum is over the n_d spatial dimensions and $\boldsymbol{\Delta}^k_{\mathcal{C}}$ is the first order update in the x^k direction.

Chapter 3

CONSERVATION LAWS WITH SPATIALLY VARYING FLUX FUNCTIONS

In the previous chapter we introduced a class of wave propagation algorithms for conservation laws with autonomous flux functions of the form $\boldsymbol{f}(\boldsymbol{q})$. In Chapter 1, however, we mentioned that conservation laws written in a general basis (i.e. a basis that covers a manifold $\mathscr{M}$) have flux functions that depend explicitly on the spatial coordinates x^k. The general form of such a conservation law is shown in equation (1.4), and a derivation of the functional form of the flux for this equation follows in Chapter 5.

Spatially-varying flux functions are not unique to conservation laws on manifolds, but arise in many different applications. An application to a problem in numerical relativity is discussed by Bardeen and Buchman [7]. Spatially-varying flux functions also arise in the modeling of traffic flow on roads with varying conditions, nonlinear elasticity in heterogeneous materials, or flow through a heterogeneous porous media.

In this chapter we present a general approach to solving conservation laws with spatially-varying flux functions. The method is based on solving modified Riemann problems that give rise to a new set of waves that are called f-waves. In Chapter 6 we will use this approach to handle the spatial variation that appears on manifolds. This chapter is based on a paper by Bale, LeVeque, Mitran, and Rossmanith [4].

3.1 The f-wave decomposition

In one space dimension the equation for a conservation law with a spatially-varying flux function has the form

$$\boldsymbol{q}_{,t} + \boldsymbol{f}(\boldsymbol{q}, x)_{,x} = 0, \tag{3.1}$$

where $\boldsymbol{q}$ is again a vector of state variables and $\boldsymbol{f}(\boldsymbol{q}, x) : \mathbb{R}^{n_{\text{eq}}} \times \mathbb{R} \to \mathbb{R}^{n_{\text{eq}}}$ is the flux. Recall that in equation (2.1) the grid function $\boldsymbol{Q}_i$ represents the finite volume approximation of $\boldsymbol{q}$ in each cell. Similarly, for a flux function that depends on a spatial variable we assume a cell-centered piecewise constant discretization such that $\boldsymbol{f}_i(\boldsymbol{q})$ is the value of the flux throughout the i^{th} grid cell. Any discretization consistent with a finite volume representation will work, for example $\boldsymbol{f}_i(\boldsymbol{q}) = \boldsymbol{f}(\boldsymbol{q}, x_i)$. Using this cell-centered approximation to the flux, the generalized Riemann problem at the cell interface $x = x_{i-1/2}$ consists of the equation

$$\boldsymbol{q}_{,t} + \mathcal{F}_{i-1/2}(\boldsymbol{q}, x)_{,x} = 0, \tag{3.2}$$

where

$$\mathcal{F}_{i-1/2}(\boldsymbol{q}, x) = \begin{cases} \boldsymbol{f}_{i-1}(\boldsymbol{q}) & \text{if } x < x_{i-1/2} \\ \boldsymbol{f}_i(\boldsymbol{q}) & \text{if } x > x_{i-1/2}, \end{cases} \tag{3.3}$$

together with initial data (2.9) with $\boldsymbol{q}_l = \boldsymbol{Q}_{i-1}$ and $\boldsymbol{q}_r = \boldsymbol{Q}_i$.

At each cell interface, the generalized Riemann problem (3.2) is solved using the eigenvectors of a matrix $A_{i-1/2}$ that approximates the Jacobian $\partial \boldsymbol{f}/\partial \boldsymbol{q}(\boldsymbol{q}, x)$. Typically, for the standard wave propagation algorithm based on splitting jumps in $\boldsymbol{Q}$, $A_{i-1/2}$ uses Roe averages (i.e. see [69]) to ensure that equation (2.35) is satisfied and the method remains conservative. One of the features of the approach presented here is that the update will remain conservative for any reasonable choice of $A_{i-1/2}$, making it no longer necessary to compute Roe averages. For simple systems this is not very beneficial, but for complicated systems like that of magneto-hydrodynamics, the Roe average is difficult to find and expensive to compute.

The approximate Jacobian can be left quite general so that

$$A_{i-1/2} = A(\boldsymbol{Q}_{i-1}, x_{i-1}, \boldsymbol{Q}_i, x_i), \tag{3.4}$$

where A is some matrix-valued function that approximates the Jacobian. It will be shown in Section 3.3 that, provided $\boldsymbol{q}(t,x)$ and $\boldsymbol{f}(\boldsymbol{q},x)$ remain smooth, the wave propagation algorithm presented here is second order if A approximates the Jacobian so that

$$A\left(\boldsymbol{q}(t, x-\Delta x), x-\Delta x, \boldsymbol{q}(t,x), x\right) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\left(\boldsymbol{q}\left(t, x-\frac{\Delta x}{2}\right), x-\frac{\Delta x}{2}\right) + E(x, \Delta x), \tag{3.5}$$

where the error $E(x, \Delta x)$ satisfies

$$E(x, \Delta x) = \mathcal{O}(\Delta x) \quad \text{as } \Delta x \to 0 \tag{3.6}$$

and

$$\frac{E(x,\Delta x) - E(x-\Delta x, \Delta x)}{\Delta x} = \mathcal{O}(\Delta x) \quad \text{as } \Delta x \to 0. \tag{3.7}$$

Another discussion of the accuracy of this approach applied to gravitational plane waves in general relativity can be found in [7]. For a particular choice of $A_{i-1/2}$, the nonlinear problem (3.2) is replaced by a linear one with

$$\boldsymbol{q}_{,t} + A_{i-1/2}\, \boldsymbol{q}_{,x} = 0. \tag{3.8}$$

We have seen how to solve the linear problem in Section 2.2.1.

The matrix $A_{i-1/2}$ is never constructed explicitly since only its set of eigenvalues $\{\lambda^p\}$ and eigenvectors $\{\boldsymbol{r}^p\}$ are needed in the solution of the Riemann problem. In practice there are various approaches to choose the eigenvectors. For example, one way is to choose the eigenvectors of $\partial \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1}, x_{i-1})/\partial \boldsymbol{q}$ that correspond to the negative eigenvalues and therefore left-propagating waves, and combine these with eigenvectors of $\partial \boldsymbol{f}_i(\boldsymbol{Q}_i, x_i)/\partial \boldsymbol{q}$ that correspond to positive eigenvalues. Another way is to choose the eigenvectors of the Jacobian $\partial \boldsymbol{f}/\partial \boldsymbol{q}(\overline{\boldsymbol{Q}}_{i-1/2}, x_{i-1/2})$ evaluated at some average $\overline{\boldsymbol{Q}}_{i-1/2}$ of the states to the left and right of the interface. This average may be a simple arithmetic average, or a more complicated average like a Roe average.

It was shown in Section 2.2.1 that the Riemann solution for the constant coefficient linear system (3.8) can be written as a superposition of waves $\boldsymbol{\mathcal{W}}_{i-1/2}$. The standard wave propagation algorithm defines waves by decomposing the jump in $\boldsymbol{Q}$ across the interface

$$\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1} = \sum_{p=1}^{n_{eq}} \alpha^p_{i-1/2}\, \boldsymbol{r}^p_{i-1/2} = \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{W}}^p_{i-1/2}, \tag{3.9}$$

where

$$\alpha^p = \boldsymbol{l}^p \cdot \left(\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1}\right). \tag{3.10}$$

These waves are then used to construct the fluctuations (2.26) that are used in the one-dimensional update (2.31) or the multidimensional update (2.52). Recall, however, that the updates (2.31) and (2.52) for the linear system (3.8) are only conservative if the condition (2.35) is satisfied implying that

$$\boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}) = \sum_{p=1}^{n_{eq}} \alpha^p_{i-1/2}\, s^p_{i-1/2}\, \boldsymbol{r}^p_{i-1/2} \tag{3.11}$$

is true across each interface $x = x_{i-1/2}$.

As was previously stated, a novel feature of the algorithm presented in this chapter is that any attempts to define a Jacobian matrix $A_{i-1/2}$ such that the condition (**??**), and equivalently equation (2.35), is satisfied are abandoned. A classical decomposition of the jump in $\boldsymbol{Q}$ across the interface at $x = x_{i-1/2}$ is not used to solve the Riemann problem. Instead, the flux-based decomposition

$$\boldsymbol{f}_i(\boldsymbol{Q}_i) - \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1}) = \sum_{p=1}^{n_{eq}} \beta^p_{i-1/2}\, \boldsymbol{r}^p_{i-1/2} \equiv \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}^p_{i-1/2}, \tag{3.12}$$

where

$$\beta^p_{i-1/2} = \boldsymbol{l}^p_{i-1/2} \cdot \left(\boldsymbol{f}_i(\boldsymbol{Q}_i) - \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1})\right) \tag{3.13}$$

is used to construct the solution. This decomposition defines a set of f-waves $\{\boldsymbol{\mathcal{Z}}_{i-1/2}\}$ at the interface. In the following section we demonstrate that the set of f-waves are in fact the natural set of waves to use for spatially-varying flux functions. In Section 3.3 we present a modified wave propagation algorithm that makes use of these f-waves.

3.2 The generalized Riemann problem

It will be shown in this section that it is the f-waves that yield the correct solution to the generalized Riemann problem (3.2) associated with a spatially-varying flux function. As in Section 2.3.1 the advection equation is considered, but the velocity will now depend on x so that the equation has the form

$$q_{,t} + \left(u(x)\, q(t,x)\right)_{,x} = 0. \tag{3.14}$$

The discretization for the flux is naturally taken to be $f_i(q) = u(x_i)\, q \equiv u_i\, q$, where u_i is the advection velocity in the i^{th} cell. Assuming that u_i does not change sign across the interface, the structure of the Riemann solution (for $u_i > 0$) is shown in Figure 3.1(a) and is comprised of a single propagating wave $\boldsymbol{\mathcal{W}}^1_{i-1/2}$ with speed u_i and strength $Q_i - Q^r_{i-1/2}$, together with a stationary discontinuity at $x = x_{i-1/2}$ with amplitude $Q^r_{i-1/2} - Q_{i-1}$. The stationary discontinuity exists due to the fact that for bounded solutions of equation (3.1), it is expected that the flux will be continuous across the interface so that

$$u_{i-1}\, Q_{i-1} = u_i\, Q^r_{i-1/2}, \tag{3.15}$$

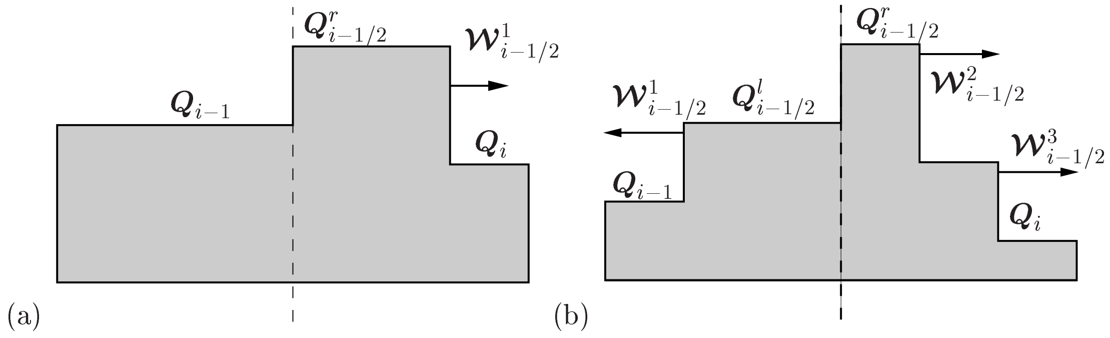


Figure 3.1: (a) The solution to the generalized Riemann problem for the advection equation consists of a traveling wave $\mathcal{W}^1_{i-1/2}$ and a stationary jump in $\boldsymbol{Q}$ across the interface $x = x_{i-1/2}$. (b) The solution for a system of equations also contains a stationary jump.

which sets the necessary amplitude for the value of $Q^r_{i-1/2}$ just to the right of the interface. The solution to the generalized Riemann problem can then be written as

$$q(t,x) = \begin{cases} Q_{i-1} & \text{for } x < x_{i-1/2} \\ Q^r_{i-1/2} & \text{for } x_{i-1/2} < x < x_{i-1/2} + u_i\,(t - t_n) \\ Q_i & \text{for } x > x_{i-1/2} + u_i\,(t - t_n), \end{cases} \tag{3.16}$$

where

$$Q^r_{i-1/2} = \frac{u_{i-1}}{u_i}\, Q_{i-1} \tag{3.17}$$

is the value of q in the spacetime wedge between the lines $x = x_{i-1/2}$ and $x = x_{i-1/2} + u_i\,(t - t_n)$.

Suppose that the classical decomposition (3.9) was used to to generate the waves used to solve the Riemann problem just discussed. Define the Jacobian $A_{i-1/2} = u_i$ (once again for $u_i > 0$) with $\boldsymbol{r}^1_{i-1/2} = 1$ and $s^1_{i-1/2} = u_i$. The wave is then defined by

$$\mathcal{W}^1_{i-1/2} = \alpha^1_{i-1/2} = Q_i - Q_{i-1}, \tag{3.18}$$

and will propagate at a speed $s^1_{i-1/2} = u_i$. In this case, the wave strength is the entire jump $Q_i - Q_{i-1}$ leaving no stationary discontinuity at $x = x_{i-1/2}$. This is not the correct Riemann solution since there is jump $Q_{i-1}(f_i(Q_i) - f_{i-1}(Q_{i-1})$ in the flux that remains at the interface.

If instead, the flux-based decomposition (3.12) is used with the same definition of the Jacobian, then

$$\mathcal{Z}^1_{i-1/2} = \beta^1_{i-1/2} = u_i\, Q_i - u_{i-1}\, Q_{i-1}. \tag{3.19}$$

The strength of the traveling wave from Figure 3.1(a) can be recovered by dividing the

f-wave by $s^1_{i-1/2}$

$$\begin{aligned} \mathcal{W}^1_{i-1/2} &= \mathcal{Z}^1_{i-1/2}/s^1_{i-1/2} \\ &= (u_i\, Q_i - u_{i-1}\, Q_{i-1})\,/u_i \\ &= Q_i - Q^r_{i-1/2}, \end{aligned} \tag{3.20}$$

just as should be expected. It is also true that the f-wave carries the entire jump in the flux so that no discontinuity in the flux remains at the interface.

For systems of equations the solution of the generalized Riemann problem has a structure depicted in Figure 3.1(b). There will be n_{eq} waves $\boldsymbol{\mathcal{W}}^p_{i-1/2}$ traveling at speeds $\boldsymbol{s}^p_{i-1/2}$, and a stationary jump in $\boldsymbol{q}$ at $x = x_{i-1/2}$ with strength $\boldsymbol{Q}^r_{i-1/2} - \boldsymbol{Q}^l_{i-1/2}$. Once again, if equation (3.1) is expected to have a bounded solution then the flux will be continuous across the interface $x = x_{i-1/2}$ so that

$$\boldsymbol{f}_{i-1}(\boldsymbol{Q}^l_{i-1/2}) = \boldsymbol{f}_i(\boldsymbol{Q}^r_{i-1/2}). \tag{3.21}$$

This condition can be used to calculate the jump across the stationary discontinuity at $x = x_{i-1/2}$.

3.3 Wave propagation algorithms using the f-waves

The formulas presented in Section 2.3 are easily modified to make use of the set of f-waves, $\{\boldsymbol{\mathcal{Z}}^p_{i-1/2}\}$, instead of the standard set of waves, $\{\boldsymbol{\mathcal{W}}^p_{i-1/2}\}$. Since $\boldsymbol{\mathcal{Z}}^p_{i-1/2} = s^p_{i-1/2}\,\boldsymbol{\mathcal{W}}^p_{i-1/2}$, the fluctuations (2.26) are replaced with

$$\boldsymbol{\mathcal{A}}^{\pm}\boldsymbol{\Delta Q}_{i-1/2} = \sum_{p \in \mathscr{P}^{\pm}} \boldsymbol{\mathcal{Z}}^p_{i-1/2}, \tag{3.22}$$

where as before, $\mathscr{P}^-$ is the set of p for which $s^p_{i-1/2} < 0$ and similarly, $\mathscr{P}^+$ is the set for which $s^p_{i-1/2} > 0$.

The correction flux (2.41) can be replaced with

$$\tilde{\boldsymbol{F}}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{n_{eq}} \operatorname{sgn}(s^p_{i-1/2}) \left(1 - \frac{\Delta t}{\Delta x}|s^p_{i-1/2}|\right) \tilde{\boldsymbol{\mathcal{Z}}}^p_{i-1/2}, \tag{3.23}$$

where the tilde implies that a limiter is used. The limiters are applied directly to the f-waves so that

$$\tilde{\boldsymbol{\mathcal{Z}}}^p_{i-1/2} = \text{Limiter}(\boldsymbol{\mathcal{Z}}^p_{i-1/2}, \boldsymbol{\mathcal{Z}}^p_{I-1/2}), \tag{3.24}$$

with

$$I = \begin{cases} i-1 & \text{if } s^p_{i-1/2} > 0 \\ i+1 & \text{if } s^p_{i+1/2} > 0, \end{cases} \tag{3.25}$$

just as for the standard waves. For linear problems it can be shown that the first order algorithm based on f-waves reduces to the standard algorithm. The high-resolution corrections, however, are different due to the fact that the limiters are applied directly to the f-waves.

Since standard waves must be defined in terms of f-waves as $\boldsymbol{\mathcal{W}}^p_{i-1/2} = \boldsymbol{\mathcal{Z}}^p_{i-1/2}/s^p_{i-1/2}$, the fact that the high-resolution correction can be implemented in terms of the f-waves directly avoids problems when $s^p_{i-1/2} = 0$. Using the fluctuations (3.22), and the correction fluxes (3.23), the high-resolution wave propagation update becomes

$$\begin{aligned} \boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n &- \frac{\Delta t}{\Delta x}\left(\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2} + \boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i+1/2}\right) \\ &- \frac{\Delta t}{\Delta x}\left(\tilde{\boldsymbol{F}}_{i+1/2} - \tilde{\boldsymbol{F}}_{i-1/2}\right). \end{aligned} \tag{3.26}$$

This update looks identical to the standard update, but fluctuations and correction fluxes are computed using f-waves.

It was previously stated that the algorithm presented here would be second order provided that the solution, $\boldsymbol{q}(t,x)$, and flux function, $f(\boldsymbol{q},x)$, remain smooth and the conditions (3.6) and (3.7) on the approximate Jacobian are satisfied. To show this, consider the update (3.26) without limiters so that

$$\begin{aligned} \boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n \quad &- \tfrac{\Delta t}{\Delta x} \left[\sum_{p\in\mathscr{P}^+} \boldsymbol{\mathcal{Z}}^p_{i-1/2} + \sum_{p\in\mathscr{P}^-} \boldsymbol{\mathcal{Z}}^p_{i+1/2}\right] \\ &- \tfrac{\Delta t}{2\Delta x} \left[\sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_{i+1/2})\left(1 - \frac{\Delta t}{\Delta x}\left|s^p_{i+1/2}\right|\right)\boldsymbol{\mathcal{Z}}^p_{i+1/2}\right. \\ &\qquad \left. - \sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_{i-1/2})\left(1 - \frac{\Delta t}{\Delta x}\left|s^p_{i-1/2}\right|\right)\boldsymbol{\mathcal{Z}}^p_{i-1/2}\right] \end{aligned} \tag{3.27}$$

$$\begin{aligned} = \boldsymbol{Q}_i^n \quad &- \tfrac{\Delta t}{\Delta x} \left[\sum_{p\in\mathscr{P}^+} \boldsymbol{\mathcal{Z}}^p_{i-1/2} - \frac{1}{2}\sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_{i-1/2})\boldsymbol{\mathcal{Z}}^p_{i-1/2}\right. \\ &\qquad \left. + \sum_{p\in\mathscr{P}^-} \boldsymbol{\mathcal{Z}}^p_{i+1/2} + \frac{1}{2}\sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_{i+1/2})\boldsymbol{\mathcal{Z}}^p_{i+1/2}\right] \\ &+ \tfrac{\Delta t^2}{2\Delta x^2} \left[\sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_{i+1/2})\left|s^p_{i+1/2}\right|\boldsymbol{\mathcal{Z}}^p_{i+1/2}\right. \\ &\qquad \left. - \sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_{i-1/2})\left|s^p_{i-1/2}\right|\boldsymbol{\mathcal{Z}}^p_{i-1/2}\right], \end{aligned} \tag{3.28}$$

where all that has been done is a gathering of the powers in $\Delta t/\Delta x$. In this reordering, the first order fluctuations have gained terms that will convert upwind differences to central differences. The previous expression can be simplified by eliminating the sign($\cdot$) functions. The following relationships:

$$\begin{aligned} \sum_{p=1}^{n_{eq}} \operatorname{sign}(s^p_I)\boldsymbol{\mathcal{Z}}^p_I &= \sum_{p\in\mathscr{P}^+} \boldsymbol{\mathcal{Z}}^p_I - \sum_{p\in\mathscr{P}^-} \boldsymbol{\mathcal{Z}}^p_I, \\ \operatorname{sign}(s^p_I)\left|s^p_I\right| &= s^p_I, \end{aligned} \tag{3.29}$$

for $I = i \pm 1/2$ can be used in equation (3.28) to get

$$\begin{aligned}\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n \quad & -\tfrac{\Delta t}{\Delta x} \quad \left[\sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_{i-1/2}^p + \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_{i+1/2}^p\right] \\ & -\tfrac{\Delta t}{2\Delta x} \quad \left[\sum_{p=1}^{n_{eq}} s_{i+1/2}^p \boldsymbol{\mathcal{Z}}_{i+1/2}^p - \sum_{p=1}^{n_{eq}} s_{i+1/2}^p \boldsymbol{\mathcal{Z}}_{i+1/2}^p\right].\end{aligned} \tag{3.30}$$

The last line in the previous update can be related to the approximate Jacobian using

$$\sum_{p=1}^{n_{eq}} s_I^p \boldsymbol{\mathcal{Z}}_I^p = A_I \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_I^p, \tag{3.31}$$

since s_I^p is an eigen value of the matrix A_I with the corresponding eigenvector $\boldsymbol{\mathcal{Z}}_I^p$. The resulting update has the form

$$\begin{aligned}\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n \quad & -\tfrac{\Delta t}{\Delta x} \quad \left[\sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_{i-1/2}^p + \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_{i+1/2}^p\right] & (3.32) \\ & -\tfrac{\Delta t}{2\Delta x} \quad \left[A_{i+1/2} \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_{i+1/2}^p - A_{i-1/2} \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}_{i+1/2}^p\right]. \\ = \boldsymbol{Q}_i^n \quad & -\tfrac{\Delta t}{2\Delta x} \quad \left(\boldsymbol{f}(\boldsymbol{Q}_{i+1}^n, x_{i+1}) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}^n, x_{i-1})\right) & (3.33) \\ & -\tfrac{\Delta t^2}{2\Delta x^2} \quad \Big[A_{i+1/2} \left(\boldsymbol{f}(\boldsymbol{Q}_{i+1}^n, x_{i+1}) - \boldsymbol{f}(\boldsymbol{Q}_i^n, x_i)\right) \\ & \qquad -A_{i-1/2} \left(\boldsymbol{f}(\boldsymbol{Q}_i^n, x_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}^n, x_{i-1})\right)\Big],\end{aligned}$$

where in the last equality we have used equation (3.12) that states the entire flux difference is carried by the f-waves at each cell interface.

The Taylor series expansion for the true solution has the form

$$\begin{aligned}\boldsymbol{q}(t_{n+1}, x_i) = \boldsymbol{q}(t, x)|_{t_n, x_i} & - \Delta t\, \boldsymbol{f}(\boldsymbol{q}, x)_{,x}|_{t_n, x_i} \\ & + \frac{1}{2}\Delta t^2 \left[\partial \boldsymbol{f}/\partial \boldsymbol{q}\, (q, x)\, \boldsymbol{f}(\boldsymbol{q}, x)_{,x}\right]_{,x}|_{t_n, x_i} + \mathcal{O}(\Delta t^3).\end{aligned} \tag{3.34}$$

The $\mathcal{O}(\Delta t)$ terms of the update (3.33) and the Taylor expansion agree to $\mathcal{O}(\Delta t\, \Delta x^2)$. The conditions that the approximate Jacobian satisfy, namely (3.6) and (3.7), allow us to write

$$\begin{aligned}A(\boldsymbol{q}(t, x - \Delta x), \boldsymbol{q}(t, x)) & \left(\frac{\boldsymbol{f}(\boldsymbol{q}(t, x), x) - \boldsymbol{f}(\boldsymbol{q}(t, x - \Delta x), x - \Delta x)}{\Delta x}\right) = \\ & \partial \boldsymbol{f}/\partial \boldsymbol{q}\, (\boldsymbol{q}(t, x - \Delta x/2), x - \Delta x)[\\ & \quad \boldsymbol{f}(\boldsymbol{q}(t, x - \Delta x/2), x - \Delta x/2)]_{,x} + E_2(x, \Delta x),\end{aligned} \tag{3.35}$$

and guarantee that $E_2(x, \Delta x)$ is at least $\mathcal{O}(\Delta x)$. This is sufficient to show that the $\mathcal{O}(\Delta t^2)$ terms in (3.33) and (3.34) agree to $\mathcal{O}(\Delta t^2\, \Delta x)$. This demonstrates the second order accuracy for the algorithm based on f-waves.

3.4 A modified Burgers' Equation

As an example of the algorithm presented in this chapter, we consider the nonlinear initial value problem

$$\begin{aligned} &\frac{\partial}{\partial t} q(t,x) + \frac{\partial}{\partial x}\left[(1+\sigma x)\frac{q^2}{2}\right] = 0 \\ &q(t=0,x) = f(x), \quad x \in (-\infty, +\infty), \end{aligned} \tag{3.36}$$

where σ is taken to be a non-negative constant. Notice that if $\sigma = 0$ this PDE is simply Burgers' equation. In this example, we assume $f(x) > 0$ for all x and that $f'(x) < 0$ on some part of our computational domain. This guarantees that at some time $T_{\mathcal{S}} > 0$, characteristics of equation (3.36) will cross and the solution will shock. This will allow us to test the accuracy of the new algorithm for some time $T_{\mathcal{A}} < T_{\mathcal{S}}$ while the solution is still smooth. For $t > T_{\mathcal{S}}$ we can verify that the shock is captured with no numerical oscillations.

In order to test the accuracy of the new algorithm, we need an *exact* solution for all time $t < T_{\mathcal{S}}$. The method of characteristics can be applied to find the implicit solution

$$x = \frac{(\sigma\xi + 1)(2 + \sigma t f(\xi))^2}{4\sigma} - \frac{1}{\sigma} \tag{3.37}$$

$$q(t,x) = \frac{2f(\xi)}{2 + \sigma t f(\xi)}, \tag{3.38}$$

so that given any x and $t < T_{\mathcal{S}}$, a value for ξ is found by solving (3.37), which is then used in (3.38) to find $q(t,x)$. This classical solution to the initial value problem (3.36) exists for times in the interval $[0, T_{\mathcal{S}}]$, where

$$T_{\mathcal{S}} = \min_{\xi}\left[\frac{-2}{\sigma f(\xi) + 2f'(\xi)(1+\sigma\xi)}\right]. \tag{3.39}$$

In the example presented here we use the initial condition

$$f(x) = \alpha \exp\left[\left(\frac{x - x_0}{\omega}\right)^2\right], \tag{3.40}$$

so that the above assumptions are satisfied. A plot of this initial condition is shown in Figure 3.2(a) as a dashed line. Notice that is is non-negative, and has a region of negative slope as desired. The exact solution can then be computed for all $t < T_{\mathcal{S}}$ by solving the nonlinear, implicit equation (3.37) for ξ, and using this in equation (3.38). The exact solution, therefore, is only as accurate as the solution of (3.37), which we solve to machine precision.

In Figures 3.2(a) through 3.3(b) we present a calculation with $\sigma = 3/5$, $\alpha = 1$, $x_0 = 0$, and $\omega = 4/5$. In Figure 3.2(a), the exact solution has been compared to the computed solution at $t = 0.6 \approx 0.78\,T_{\mathcal{S}}$. At this time, we have done an error analysis of the new algorithm. Figure 3.3(a) contains the errors that the method makes in both the max norm and the one norm. A log-log plot of these errors can be found in Figure 3.3(b), where we

(a) (b)

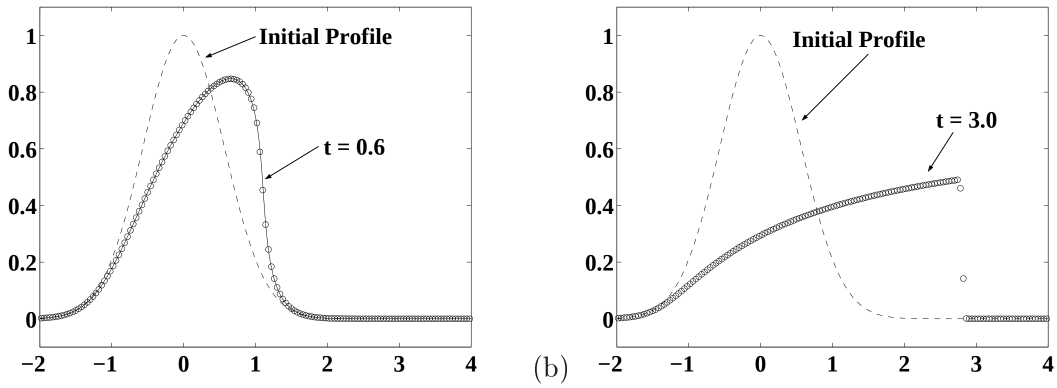


Figure 3.2: (a) A comparison between the exact solution (solid line) and a numerical solution (circles) at $t = 0.6 \approx 0.78T_b$. The solution is beginning to steepen as it propagates to the right. Also shown is the initial condition (dashed line). (b) The numerical solution at $t = 3.0$. The classical solution has shocked and we are computing the conservative weak solution. The dashed line, once again, represents the initial condition.

have plotted the one-norm errors with a circle and max-norm errors with an x. Also plotted is a linear fit to the errors from which an order of accuracy has been computed.

One thing to notice is that the one-norm errors are second order ($p = 1.98$), while the max-norm errors are slightly less than second order ($p = 1.76$). This is due to the fact that the solution is already quite steep at $t = 0.6$ (see Figure 3.2(a)), and errors in this steep region are dominant. The max norm is sensitive to these errors and will be slower to reach an asymptotic value for the order of accuracy.

In Figure 3.2(b) we show the numerical solution at $t = 3 > T_{\mathcal{S}}$. The classical solution no longer exists at this time. The circles represent the discontinuous weak solution to the conservation law (3.36). Notice the sharp resolution of the shock front, and the fact that there are no oscillations throughout the solution.

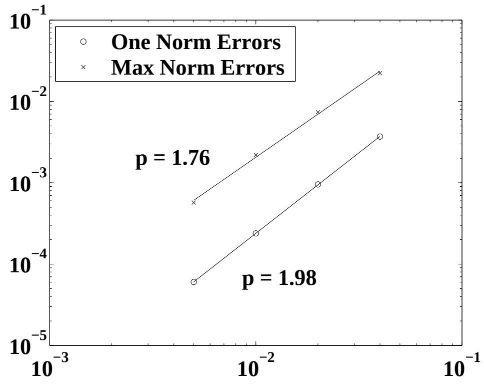
More examples of accuracy studies for the f-wave algorithm developed here can be found in [4], the paper on which this chapter is based. An accuracy study of the algorithm's performance for elastic waves in heterogeneous media is presented in Section 4 of [4], and for linear acoustics in Section 5.

3.5 Flow of an inviscid gas in a duct with varying cross-sectional area

In the following example, we investigate the performance of the algorithm for the flow of an inviscid gas in a duct with a cross-sectional area that is a function of the spatial variable. Other high-resolution schemes have been proposed for this system, for example in [16] a TVD scheme was proposed for this system including losses due to friction and heat transfer to the wall. In this section we neglect such losses and apply the algorithm to the flow of a

dx	$\|Q-q\|_\infty$	$\|Q-q\|_1$
0.04	2.226889e-02	3.696241e-03
0.02	7.376764e-03	9.555440e-04
0.01	2.199132e-03	2.386655e-04
0.005	5.712810e-04	6.031043e-05
order	1.76	1.98

(a)



(b)

Figure 3.3: (a) Errors in the computed solution for various cell widths at $t = 0.6$.(b) A Log-Log plot of the error as a function of the cell width dx. The One norm is plotted with a circle and the Max norm is plotted with an x. A linear fit is also plotted from which an order of accuracy is computed.

perfect, one-dimensional fluid in a duct as depicted in Figure 3.4. The cross-sectional area, $a(x)$, represents the distance between symmetric walls that follow the function $f(x)$.

The quasi-one-dimensional equations that govern the unsteady flow in such a duct are the one-dimensional Euler equations with a source term that results from pressure forces on the duct walls. They can be written

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ e\end{pmatrix} + \frac{1}{a(x)}\frac{\partial}{\partial x}\left[a(x)\begin{pmatrix}\rho u\\ \rho u^2+p\\ (e+p)\,u\end{pmatrix}\right] = \begin{pmatrix}0\\ \frac{a'(x)\,p}{a(x)}\\ 0\end{pmatrix}, \tag{3.41}$$

where the area $a(x)$ gives rise to a spatially-varying flux function. For steady flow in the duct, the conservation of mass (the first equation in (3.41)) suggests that the fluid must speed up as $a(x)$ gets smaller. It follows that there may exist a *critical* Mach number $\mathcal{M}^*$ at the inlet above which the flow will transition from a subsonic to supersonic flow as it passes through the constricted region of the duct. In order for the flow to return to a subsonic state, a stationary shock must be present with a location dependent on the pressure jump across the shock.

A good test for the algorithm presented in this chapter is to initialize an unsteady flow from left to right and apply boundary conditions that are consistent with a steady solution as mentioned above. After a period of unsteady flow, the numerical solution should settle into the correct steady solution. We present two such numerical calculations in this section. In the first calculation, we initialize a flow at the duct inlet with a velocity such that the Mach number $\mathcal{M}$ is below the critical value $\mathcal{M}^*$. The boundary conditions that are applied

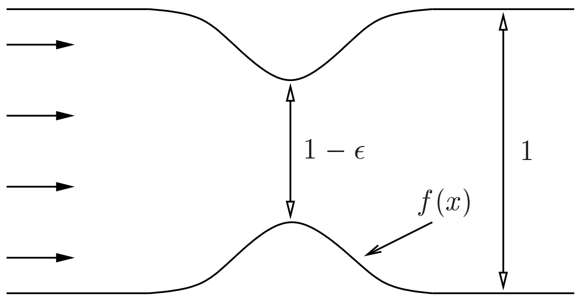


Figure 3.4: Flow in a duct with variable cross-sectional area. The wall is placed along the function $f(x)$. The minimum area is $a_{\min} = 1 - \epsilon$.

in this calculation are consistent with a subsonic flow so that the fluid remains subsonic everywhere, but with an increased velocity in the vicinity of the duct constriction shown in Figure 3.4. In the second calculation that we present the initial Mach number at the inlet is larger than the critical value, $\mathcal{M} > \mathcal{M}^*$. We also apply boundary conditions consistent with a stationary shock located at $x = 2.5$. In both cases we compare the numerical solution with an exact solution.

The exact solution can be derived using the fact that the fluid remains isentropic in smooth regions. If we define

$$\mathcal{G}(\mathcal{M}) = \mathcal{M} \left(\frac{\gamma + 1}{2 + (\gamma - 1)} \mathcal{M}^2 \right)^{\kappa}, \tag{3.42}$$

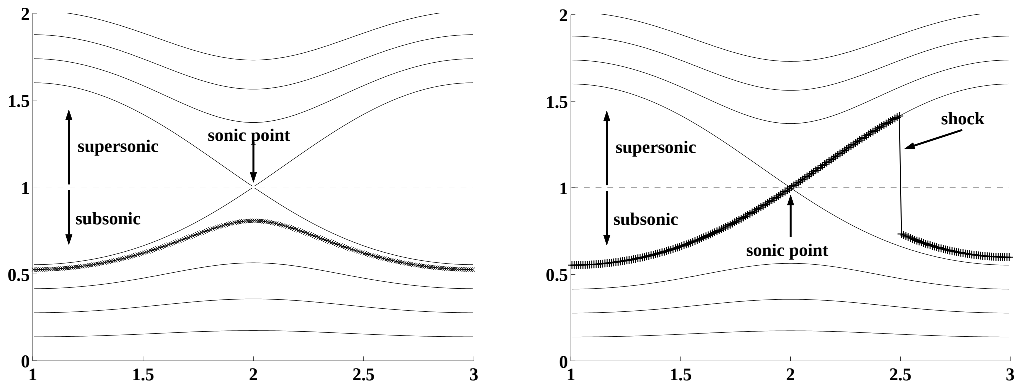
where γ is the polytropic index of the fluid and $\kappa = (\gamma + 1)/(2(\gamma - 1))$, then the isentropic nature of the flow in smooth regions is expressed by

$$a(x)\, \mathcal{G}(\mathcal{M}) = a(x_0)\, \mathcal{G}(\mathcal{M}_0), \tag{3.43}$$

where x_0 is the value of x at the left boundary and $\mathcal{M}_0 \equiv \mathcal{M}(x_0)$. To find $\mathcal{M}(x)$, we iterate to find $\mathcal{M}$ that satisfies the nonlinear equation

$$g(\mathcal{M}) \equiv \mathcal{F}(\mathcal{M}) - \frac{a(x_0)}{a(x)} \mathcal{F}(\mathcal{M}_0) = 0. \tag{3.44}$$

If $\mathcal{M}_0 > \mathcal{M}^*$ and there is a stationary shock in the duct as in Figure 3.5(b), then the Hugoniot jump conditions can be used to calculate the flow variables just to the right of the shock. After the shock the solution is once again smooth and will satisfy equation (3.44). Using the post-shock flow variables instead of the values at the left boundary, we can proceed with the iterative procedure to the right boundary (we assume only one shock since $f^{''}(x)$ will not change sign in this example). Figures 3.5(a) and 3.5(b) show two such exact solutions (marked by an x) for the Mach number $\mathcal{M}(x)$ of a steady flow in a duct on



(a) subsonic

(b) supersonic

Figure 3.5: The exact solution for flow in a duct with spatially-varying cross-sectional area. Plotted is the Mach number $\mathcal{M}(x)$ for (a) initial conditions so that the flow remains subsonic throughout the duct, and (b) initial conditions so that the flow becomes supersonic and a shock develops.

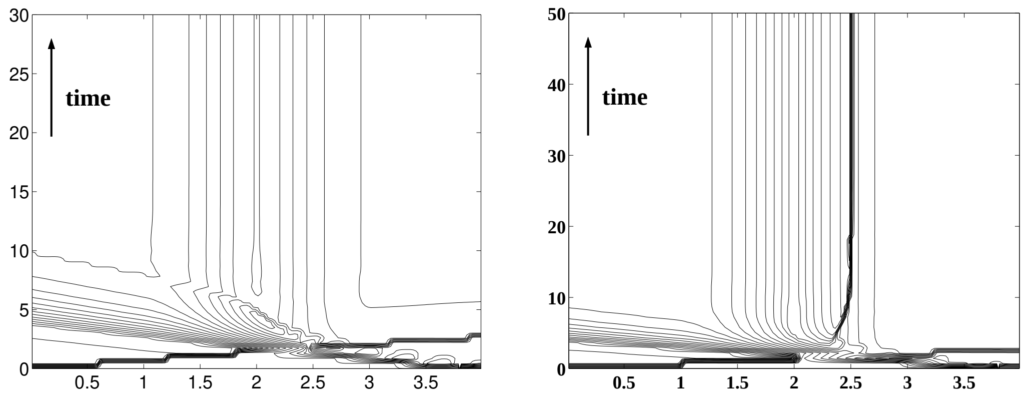
a domain $x \in [1, 3]$. The cross-sectional area of the duct has the form

$$a(x) = \begin{cases} 1 & \text{for } x < 1 \\ 1 + \frac{\epsilon}{2}\left[\cos \pi(x-1) - 1\right] & \text{for } 1 \leq x \leq 3 \\ 1 & \text{for } x > 3. \end{cases} \tag{3.45}$$

In both figures we have plotted several solutions with solid lines. Notice that as the Mach number at the inlet increases, the solution gets closer to the supersonic transition. In Figure 3.5(a) the marked solution remains subsonic throughout the domain. In Figure 3.5(b), however, there is a transition to supersonic flow and therefore stationary shock located at $x = 2.5$. These solutions can be used to compare with a numerical solution.

We now present the numerical solution generated by our algorithm and compare it with the exact solution for $a(x)$ defined by equation (3.45). The fluid is initialized to a uniform state in which the flow is from left to right. Boundary conditions are generated using the exact solutions discussed above and they are applied in each time step. Figures 3.6(a) and 3.6(b) show the time evolution of the numerical solution for the subsonic and supersonic problem, respectively. They depict the contours of the Mach number, $\mathcal{M}(x)$, in the x-t plane. Notice that both the subsonic solution (Figure 3.6(a)) and the supersonic solution (Figure 3.6(b)) are initially quite dynamic, but both eventually settle to a stationary solution. In the subsonic case, the Mach number remains smooth as expected. In the supersonic case, however, the shock develops around $t = 5$, moves around until about $t = 20$, and finally settles into the position at $x = 2.5$.

Figure 3.7 compares the density, velocity, and pressure of the exact (solid line) and



(a) subsonic

(b) supersonic

Figure 3.6: The time evolution of both the (a) subsonic solution and the (b) supersonic solution. Contours of the solution in the $x - t$ plane are shown. At early times both solutions are dynamic, but eventually settle to a steady-state.

numerical (circles) solution at the final time for the subsonic problem. In this calculation we have taken $\epsilon = 0.2$, and used 160 cells across the domain. As expected, the velocity of the fluid increases in the region where the area is small. The numerical solution captures the subsonic flow very well in all fields.

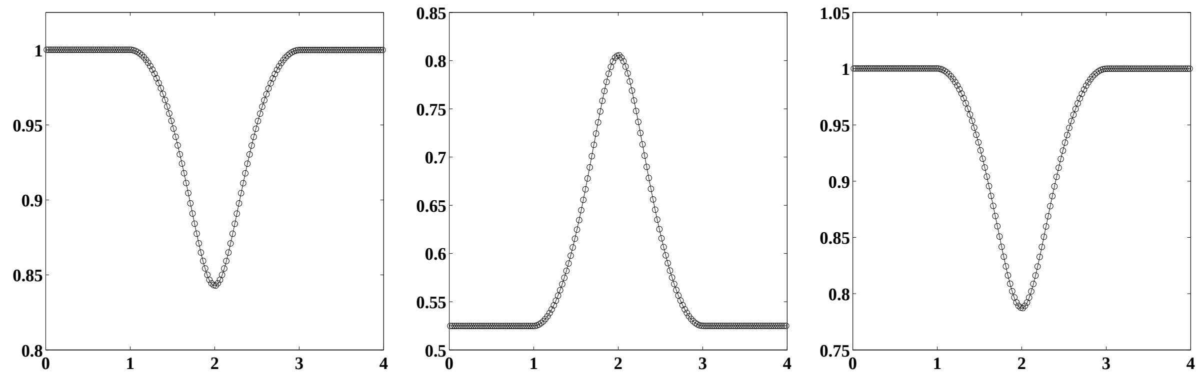
Finally, Figure 3.8 shows the same comparison for the supersonic problem. In this calculation the boundary conditions used were generated using the exact solution for a stationary shock located at $x = 2.5$ as shown in Figure 3.5(b). Once again, the numerical solution (circles) compares very well with the exact solution in all fields. It is important to note that the shock is not present in the initial data. It develops and then moves as time evolves as can be seen in Figure 3.6(b). Figure 3.8 demonstrates that the shock, in fact, moves to the correct location. The sharp resolution of both the shock strength and location can be seen in the density, velocity, and pressure.

3.6 Extension to higher dimensions

The algorithm based on f-waves presented in this chapter is easily extended to multidimensional conservation laws with spatially-varying flux functions of the form

$$\boldsymbol{q}_{,t} + \sum_{k=1}^{n_d} \boldsymbol{f}^{(k)}(\boldsymbol{q}, x^j)_{,k} = 0, \tag{3.46}$$

where, once again, braces around the k^{th} index indicate a vector along the x^k coordinate line and $j \in \{1, \cdots, n_d\}$. The procedure directly follows the extension of the standard wave propagation algorithm to multidimensional problems discussed in Section 2.3.2.

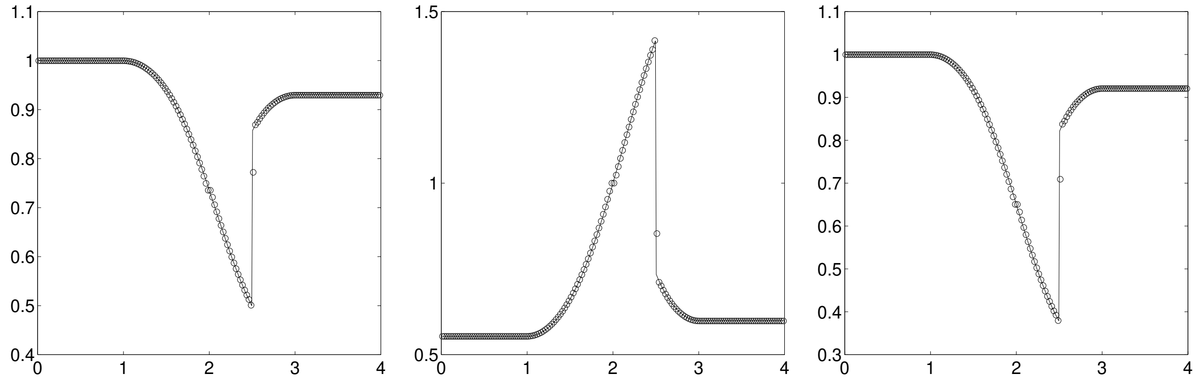


(a) density (b) velocity (c) pressure

Figure 3.7: A comparison of the numerical solution (circles) to the exact solution (solid line) for the entirely subsonic problem shown in Figure 3.5(a). Shown are the (a) density, (b) velocity, and (c) pressure for the flow.

If $n_d = 2$, for example, one-dimensional generalized Riemann problems are solved so that $\boldsymbol{f}_i(\boldsymbol{Q}_{i\,j}) - \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1\,j})$ is split into a sum of f-waves at each interface, all traveling along the x^i direction. These f-waves induce the left-going fluctuation $\boldsymbol{\mathcal{A}}_i^- \boldsymbol{\Delta Q}_{i-1/2\,j}$ and a right-going fluctuation $\boldsymbol{\mathcal{A}}_i^+ \boldsymbol{\Delta Q}_{i-1/2\,j}$ as before. These fluctuations are then split into eigenvectors of an approximation to the transverse Jacobian, in this case $\partial \boldsymbol{f}^{(j)} / \partial \boldsymbol{q}$.

An example of the algorithm discussed here applied to the multidimensional linear acoustic equations can be found in [4]. In fact, a multidimensional implementation of this algorithm based on f-waves is a key component in the development of a numerical algorithm for conservation laws on manifolds which is the main goal of this thesis. In Chapter 1 we claimed that a conservation law on a manifold has the form of equation (1.4), involving a spatially-varying flux function. In Chapter 5 we derive the form of the spatial variance induced by the manifold $\mathscr{M}$, and present examples of the implementation of this multidimensional algorithm in Chapters 6 and 8.



(a) density (b) velocity (c) pressure

Figure 3.8: A comparison of the numerical solution (circles) to the exact solution (solid line) for the supersonic problem shown in Figure 3.5(b). Shown are the (a) density, (b) velocity, and (c) pressure for the flow.

Chapter 4

CONSERVATION LAWS WITH SOURCE TERMS

It was claimed in Chapter 1 that, in general, a conservation law that is expressed on a manifold has the form (1.4). In the following chapter the details for the derivation of this equation will be presented and the functional form of the source term will be derived. In this chapter, however, we consider the general balance law

$$\boldsymbol{q}_{,t} + \boldsymbol{f}^{(k)}(\boldsymbol{q})_{,k} = \boldsymbol{\psi}(t, \vec{x}, \boldsymbol{q}), \tag{4.1}$$

consisting of a conservation law with a source term that may depend on the components of time, space, and $\boldsymbol{q}$. Balance laws occur in many different applications, conservation laws expressed on manifolds being one example. Other examples also arise for shallow-water flow over varying bottom topography as discussed in [49] and [80]. A treatment of the source terms that appear in the approximation of detonation waves is discussed by Helzel, LeVeque, and Warnecke in [34].

Various methods have been developed for balance laws with an autonomous flux function $f(\boldsymbol{q})$ (e.g. [28], [37], [49], [70]). One common approach, called a split method, is to alternate between solving a homogeneous version of the balance law

$$\boldsymbol{q}_{,t} + \boldsymbol{f}^{(k)}(\boldsymbol{q})_{,k} = 0, \tag{4.2}$$

and the ordinary differential equation

$$\boldsymbol{q}_{,t} = \boldsymbol{\psi}(t, \vec{x}, \boldsymbol{q}). \tag{4.3}$$

Although this approach works well on many problems, there are disadvantages to split methods. For example, the boundary conditions imposed on the split problem should represent the true boundary conditions for the balance law. It may not be a simple task to generate the split boundary conditions. Another disadvantage of split methods is that they do not handle steady solutions very well.

Near steady state, the flux is nearly balanced by the source term so that $|\boldsymbol{q}_{,t}| \ll |\boldsymbol{f}_{,x}| \sim |\boldsymbol{\psi}|$. In order to capture this near-equilibrium flow, the split method must rely on the cancellation of artificially induced dynamics. As an example, consider a fluid in hydrostatic balance so that the flux produced by the pressure is nearly balanced by the gravitational source term. The first step of a split method neglects the flux term, resulting in a large inflow due to the gravitational field. The second step neglects gravity, producing a large outflow due to the pressure forces. Small perturbations on top of this near-steady flow may be lost due to the subtraction of these large and opposite flows. Rather than splitting off the source term in this case, it makes more sense to incorporate it into the Riemann problems that are solved in the homogeneous equation (4.2).

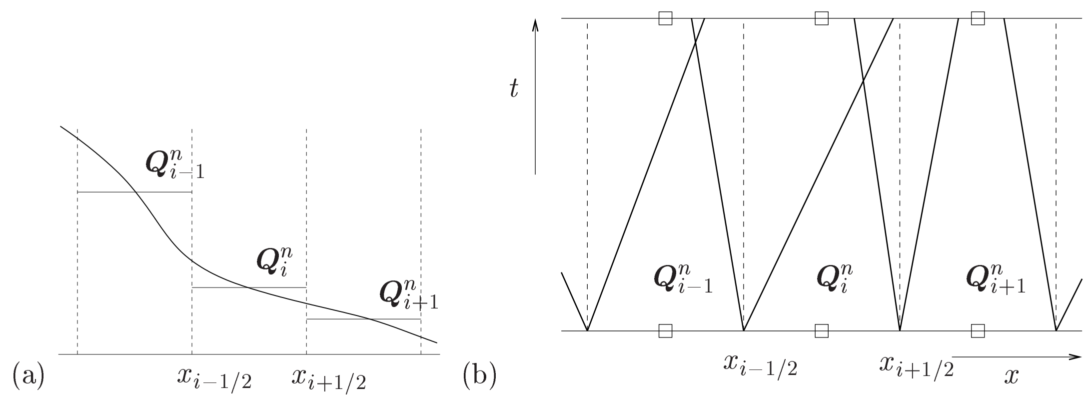


Figure 4.1: (a) Standard cell averaging (b) Wave propagation from cell interfaces resulting from solving the Riemann problems.

Like the previous example, it will be shown in Chapter 6 that it is also natural to incorporate the source terms generated by the geometry of a manifold into Riemann problems. In this chapter, therefore, we study two approaches based on wave propagation, both incorporating source terms into Riemann problems. The first approach, originally developed by LeVeque in [49], is called the quasi-steady wave propagation algorithm. This approach was applied to the Euler equations by LeVeque and Bale in [47]. The original quasi-steady method treats the source using an explicit, one-step method. The presentation here is an extension of this algorithm to multi-step methods. It will be shown that this approach handles near steady solutions very well. However, it will also be shown that this approach is impractical to use in an algorithm that solves general conservation laws on manifolds.

The second approach presented in this chapter incorporates the source term into the flux difference. This modified difference can then be used to generate f-waves as discussed in the previous chapter. This approach is shown to handle near-steady solutions very well. In addition, it does not share some of the disadvantages of the quasi-steady method, making it useful for the main goal of this dissertation; developing an algorithm for the solution of conservation laws on manifolds.

4.1 Multi-step Quasi-Steady Wave Propagation

Let us begin by studying the multi-step quasi-steady wave propagation method. As for the wave propagation methods developed in Chapter 2, one-dimensional data on each time slice is represented by piecewise constant functions having cell centered values $\boldsymbol{Q}_i^n$ at the n^{th} time level and in the i^{th} cell denoted by $\mathcal{C}_i \equiv [x_{i-1/2}, x_{i-1/2} + \Delta x]$. This finite volume representation of a function is depicted in Figure 4.1(a). The standard wave propagation algorithm, developed by LeVeque in [45], and reviewed in Section 2.3, is implemented in the software package CLAWPACK for the solution of homogeneous conservation laws like (4.2). If a source term exists, CLAWPACK provides routines to implement a split method

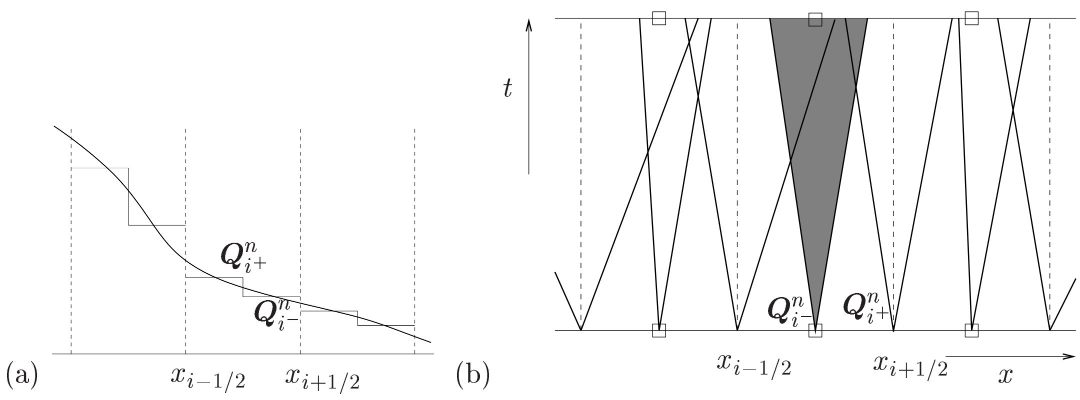


Figure 4.2: (a) Modified cell averages for the quasi-steady approach. (b) Wave propagation with cell centered Riemann problems included.

such as Strang or Godunov splitting. Suppose, instead, that it is beneficial to incorporate the source term into the Riemann problems. The quasi-steady wave propagation algorithm accomplishes this by introducing a jump in the center of each cell as depicted in Figure 4.2(a). The jump in the i^{th} cell has magnitude $|\boldsymbol{Q}_{i+}^n - \boldsymbol{Q}_{i-}^n|$ where the quantities $\boldsymbol{Q}_{i-}^n$ and $\boldsymbol{Q}_{i+}^n$ are defined in the left and right half of that cell, respectively. The jump at the center of the i^{th} cell is chosen such that the following conditions are satisfied:

$$\frac{1}{2}\left(\boldsymbol{Q}_{i-}^n + \boldsymbol{Q}_{i+}^n\right) = \boldsymbol{Q}_i^n \tag{4.4}$$

$$\boldsymbol{f}(\boldsymbol{Q}_{i+}^n) - \boldsymbol{f}(\boldsymbol{Q}_{i-}^n) = \Delta x \sum_{k=0}^{M-1} \alpha_k \boldsymbol{\psi}_i^{n-k}, \tag{4.5}$$

where we define $\boldsymbol{\psi}_i^n = \boldsymbol{\psi}(t^n, x_i, \boldsymbol{Q}_i^n)$. The first condition assures that the method will remain conservative by leaving cell averages unchanged. The second, as will be demonstrated below, assures that the cell is updated in such a way that the waves emanating from the cell-centered Riemann problems exactly cancel the influence of the source.

If the Courant number is chosen to be less than 1/2 (this restriction is only temporary) on each time slice, then waves generated by the jump located at the center of the i^{th} cell will remain entirely within the i^{th} cell. The domain of influence of the Riemann problem at the center of the i^{th} cell for such Courant numbers is depicted by the shaded region of Figure 4.2(b). Since these cell centered Riemann problems do not influence neighboring cells, it is unnecessary to resolve the resulting flux difference $\boldsymbol{f}(\boldsymbol{Q}_{i+}^n) - \boldsymbol{f}(\boldsymbol{Q}_{i-}^n)$ into right and left going pieces. Rather, this entire flux difference is added into the update for $\boldsymbol{Q}_i^{n+1}$. We must also consider the influence of the source. The influence of the source over a time step is included according to the explicit M-step method

$$\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n + \Delta t \left(\sum_{k=0}^{M-1} \alpha_k \boldsymbol{\psi}_i^{n-k} \right). \tag{4.6}$$

The set of coefficients $\{\alpha_0, \cdots, \alpha_{M-1}\}$ are the standard Adams-Bashforth multi-step coefficients. As for the standard algorithm, the influence of incoming waves at the right and left edge of the i^{th} cell is denoted by the fluctuations $\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2}$ and $\boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i+1/2}$, respectively. The unsplit modified update that incorporates the fluctuations and the influence of the source (4.6) can be written

$$\begin{aligned}\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x}\left(\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i-1/2} - \boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i+1/2}\right)\\ + \Delta t\left[\sum_{k=0}^{M-1}\alpha_k\boldsymbol{\psi}_i^{n-k} - \frac{1}{\Delta x}\left(\boldsymbol{f}(\boldsymbol{Q}_{i^+}^n) - \boldsymbol{f}(\boldsymbol{Q}_{i^-}^n)\right)\right].\end{aligned} \tag{4.7}$$

The tildes on $\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i-1/2}$ and $\boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i+1/2}$ serve as a reminder that the waves generated in Riemann problems are now based on modified data $\boldsymbol{Q}_{i-1^+}^n$ and $\boldsymbol{Q}_{i^-}^n$.

This update is a multi-step generalization of the quasi-steady updates used in [49] and [47]. The influence of the source in equation (4.7) is handled in a more accurate way for $M > 1$. It should be noted that this generalization includes the original quasi-steady update for $M = 1$.

If equation (4.5) is satisfied, then the terms in square brackets drop out completely and the first order update reduces to

$$\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n - \frac{\Delta t}{\Delta x}\left(\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i-1/2} + \boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i+1/2}\right), \tag{4.8}$$

which looks identical to the update (2.31), but with fluctuations $\boldsymbol{\mathcal{A}}^\pm\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{i-1/2}$ determined by solving the Riemann problems with data $\boldsymbol{Q}_{i-1^+}^n$ and $\boldsymbol{Q}_{i^-}^n$, as depicted in Figure 4.2(a), rather than the original flux-difference $\boldsymbol{\mathcal{A}}^\pm\boldsymbol{\Delta Q}_{i-1/2}$ based on $\boldsymbol{Q}_{i-1}^n$ and $\boldsymbol{Q}_i^n$ of Figure 4.1(a).

A natural choice for the shifted data in each cell is

$$\boldsymbol{Q}_{i\pm}^n = \boldsymbol{Q}_i^n \pm \boldsymbol{\delta}_i, \tag{4.9}$$

so that equation (4.4) is satisfied identically. Using $\boldsymbol{\delta}_i$, we can rewrite equation (4.5) as

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{Q}_i^n + \boldsymbol{\delta_i}) - \boldsymbol{f}(\boldsymbol{Q}_i^n - \boldsymbol{\delta_i}) &= \Delta x\sum_{k=0}^{M-1}\alpha_k\boldsymbol{\psi}_i^{n-k}\\ &= \Delta x\,\boldsymbol{\Theta}^M,\end{aligned} \tag{4.10}$$

where we have introduced the vector $\boldsymbol{\Theta}^M$ with a superscript denoting its dependence on the multi-step parameter M. The components of $\boldsymbol{\Theta}^M$ have absorbed the sum that appears on the right side of the first equality in the previous equation. If we denote the K^{th} component of the vectors $\boldsymbol{\psi}_i^n$ and $\boldsymbol{\Theta}^M$ by $\psi_{i[K]}^n$ and $\Theta_{[K]}^M$, respectively, then the components of $\boldsymbol{\Theta}^M$ are defined to be

$$\Theta_{[K]}^M = \begin{cases}\psi_{i[K]}^n\left(\alpha_0 + \sum_{k=1}^{M-1}\alpha_k\frac{\psi_{i[K]}^{n-k}}{\psi_{i[K]}^n}\right) & \text{if } \psi_{i[K]}^n \neq 0 \text{ for each } K\\ \psi_{i[K]}^n & \text{if } \psi_{i[K]}^n = 0 \text{ for any } K.\end{cases} \tag{4.11}$$

Note that if $M = 1$, then we are using the one-step Adams-Bashforth method (Euler's method) with $\alpha_0 = 1$. In this case equation (4.10) reduces to the original condition reported in [49] and [47].

Equation (4.10) is an algebraic system that must be solved for the components of the $\boldsymbol{\delta}_i$ that tell us how to modify the Riemann data. For $M > 1$ the components of the vector $\boldsymbol{\Theta}^M$ change the solution of this system to improve accuracy. Once equation (4.10) is solved, state variables are shifted according to equation (4.9) and Riemann problems based on $\boldsymbol{Q}^n_{i\pm}$ are solved. In fact, the shifted data is that which arises from the dynamical perturbations in the quasi-steady state. For this reason, the quasi-steady wave propagation method accurately captures quasi-steady solutions.

Although we derived this method assuming the Courant number is less than 1/2, the resulting method is stable for Courant numbers up to 1 due to the fact that the waves emanating from cell-centered Riemann problems are eliminated completely.

4.1.1 High resolution corrections and CLAWPACK*implementation*

The modified Riemann problem between states $\boldsymbol{Q}^n_{i-1^+}$ and $\boldsymbol{Q}^n_{i^-}$ gives rise to a set of waves $\mathcal{W}^p_{i-1/2}$ for $p = 1, \cdots, M_w$ with associated speeds $\lambda^p_{i-1/2}$ such that

$$\Delta \boldsymbol{Q}^n_i \equiv \boldsymbol{Q}^n_{i^-} - \boldsymbol{Q}^n_{i-1^+} = \sum_{p=1}^{M_w} \boldsymbol{\mathcal{W}}^{(p)}_{i-1/2}. \tag{4.12}$$

The first order method described above is extended to a high resolution method by adding corrections to the update (4.8). A correction flux similar to (2.41) is defined as

$$\tilde{\boldsymbol{F}}_{i-1/2} = \frac{1}{2} \sum_{p=1}^{M_w} |\lambda^p_i| \left(1 - \frac{\Delta t}{\Delta x}|\lambda^p_i|\right) \tilde{\boldsymbol{\mathcal{W}}}^{(p)}_{i-1/2}, \tag{4.13}$$

where the waves in this correction flux are based on the modified data. Once again, the tilde over the wave denotes that it has been limited as described in equation (2.42). Many more details about limiting can be found in [42], [43], and [46]. Incorporating these correction fluxes, the update (4.8) becomes

$$\begin{aligned} \boldsymbol{Q}^{n+1}_i = \boldsymbol{Q}^n_i &- \frac{\Delta t}{\Delta x}\left(\boldsymbol{\mathcal{A}}^+ \boldsymbol{\Delta} \tilde{\boldsymbol{Q}}_{i-1/2} + \boldsymbol{\mathcal{A}}^- \boldsymbol{\Delta} \tilde{\boldsymbol{Q}}_{i+1/2}\right) \\ &- \frac{\Delta t}{\Delta x}\left(\tilde{\boldsymbol{F}}_{i+1/2} - \tilde{\boldsymbol{F}}_{i-1/2}\right). \end{aligned} \tag{4.14}$$

This update looks very similar to the original wave propagation update (2.44), but is based on the modified data.

We implement the algorithm described here in the software package CLAWPACK. Only the Riemann solvers need to be modified. The process of preparing to solve Riemann problems based on the modified data is as follows:

1. Solve equation (4.10) for components of the vector $\boldsymbol{\delta}_i$ in each cell.

2. Modify the states according to equation (4.9).

3. Set up data for the i^{th} Riemann problem by setting the left state to $\boldsymbol{Q}^n_{i-1^+}$, and the right state to $\boldsymbol{Q}^n_{i^-}$.

Once this data has been set up, Riemann problems are solved generating the waves. The limiters are applied only to the waves representing disturbances from the steady state, not to the variations in the steady state itself. This leads to excellent resolution of small dynamic perturbations.

4.1.2 An implementation for the Euler equations

In Sections 4.3 and 4.4 of this chapter, the performance of both the quasi-steady algorithm and the approach presented in Section 4.2 is analyzed by solving the Euler equations for a fluid flowing under the influence of a gravitational force. In order to implement the quasi-steady algorithm on this set of equations, equation (4.10) must be solved. In this section the details of this solution are presented. These results were first presented in [47], but some of those formulas have errors. These errors have been corrected for the formulas presented in this chapter. In Sections 4.3 and 4.4, these results are used in the examples.

The Euler equations express the conservation of mass, momentum, and energy for a fluid. In Appendix B, they are derived for a perfect fluid under the influence of a general body force. The result, with gravity playing the role of the body force, can be written

$$\rho_{,t} + \vec{\nabla} \cdot \rho \vec{u} = 0, \tag{4.15}$$

$$(\rho \vec{u})_{,t} + \vec{\nabla} \cdot (\rho \vec{u} \otimes \vec{u} + \mathbb{I}\, p) = -\rho \vec{\nabla} \phi, \tag{4.16}$$

$$e_{,t} + \vec{\nabla} \cdot [(e + p)\, \vec{u}] = -\rho \vec{u} \cdot \vec{\nabla} \phi, \tag{4.17}$$

where ϕ is the gravitational potential, ρ the fluid density, $\vec{u}$ the velocity, e the fluid energy, and p the pressure. In equation (4.17), e represents the total fluid energy and does not include gravitational potential. This system is closed with the equation of state for a perfect fluid $p = (\gamma - 1)\rho\epsilon$, where ϵ is the specific internal energy. For a given potential ϕ, the gravitational acceleration is defined to be $\vec{a} = -\vec{\nabla}\phi$. It should be noted that the examples presented in this chapter involve a gravitational potential, but the methods presented here will work for any potential giving rise to a force on the fluid.

Wave propagation methods are based solving one-dimensional Riemann problems, so we will consider the Euler equations above along the x-axis. These one-dimensional equations take the form

$$\rho_{,t} + (\rho u)_{,x} = 0, \tag{4.18}$$

$$(\rho u)_{,t} + \left(\rho u^2 + p\right)_{,x} = \rho a^x, \tag{4.19}$$

$$e_{,t} + [(e + p)\, u]_{,x} = \rho u a^x, \tag{4.20}$$

where $\vec{u} = (u, 0, 0)^T$, and a^x is the component of the gravitational acceleration in the x direction. Note that we hesitate to write $a^x = -g$ because we would like to emphasize that we allow dynamic, non-uniform gravitational fields.

At each time step we would like to find components of $\boldsymbol{\delta}_i$ such that equation (4.5) is satisfied. The subscript i denoting quantities in the i^{th} cell will now be dropped and it will be understood that, for example, ρ^n represents the density in the grid cell under consideration at the n^{th} time step. Define $s \equiv \rho u$ and notice that the right hand side of equation (4.18) is zero. To satisfy equation (4.5), equations (4.18)-(4.20) require that

$$s_+^n = s_-^n \equiv s^n, \tag{4.21}$$

$$(\gamma-1)\left[e_+^n - e_-^n\right] + \frac{(3-\gamma)\,(s^n)^2}{2}\left[\frac{1}{\rho_+^n} - \frac{1}{\rho_-^n}\right] = \Delta x \rho^n a_x^n \Theta_{[s]}, \tag{4.22}$$

$$\gamma s^n\left[\frac{e_+^n}{\rho_+^n} - \frac{e_-^n}{\rho_-^n}\right] - \frac{(\gamma-1)\,(s^n)^3}{2}\left[\left(\frac{1}{\rho_+^n}\right)^2 - \left(\frac{1}{\rho_-^n}\right)^2\right] = \Delta x s^n a_x^n \Theta_{[e]}, \tag{4.23}$$

where we have used $p_\pm^n = \left(\gamma-1\right)\left(e_\pm^n - (s^n)^2/2\rho_\pm^n\right)$ to eliminate the pressure and used definition (4.11) to write the components of $\boldsymbol{\Theta}^M$ as

$$\boldsymbol{\Theta}_M = \begin{pmatrix} \Theta_{[\rho]} \\ \Theta_{[s]} \\ \Theta_{[e]} \end{pmatrix} = \alpha_0 + \sum_{k=1}^{M-1} \alpha_k \begin{pmatrix} 0 \\ \rho^{n-k}(a^x)^{n-k}/\rho^n(a^x)^n \\ s^{n-k}(a^x)^{n-k}/s^n(a^x)^n \end{pmatrix}, \tag{4.24}$$

when the momentum does not vanish. If it does vanish, then $\boldsymbol{\Theta}^M = (1,1,1)^T$.

The cell centered jump is introduced by defining

$$\rho_\pm^n = \rho^n \pm \tilde{\delta}_\rho \equiv \rho^n \ \left(1 \pm \delta_\rho\right) \tag{4.25}$$

$$e_\pm^n = e^n \pm \tilde{\delta}_e \equiv e^n \ \left(1 \pm \delta_e\right), \tag{4.26}$$

in which physical states must have $\delta_\rho, \delta_e \in (-1,1)$. In terms of δ_ρ and δ_e the non-trivial equations can be written

$$\frac{(s^n)^2(\gamma-3)}{\rho^n}\delta_\rho + \left[2(\gamma-1)e^n\delta_e - \Delta x \rho^n (a^x)^n \Theta_s\right]\left(1-\delta_\rho^2\right) = 0, \tag{4.27}$$

$$s^n\left\{2\left(\frac{s^n}{\rho^n}\right)(\gamma-1)\delta_\rho + \left[\frac{2\gamma e^n}{\rho^n}\left(\delta_e - \delta_\rho\right) - \Delta x \Theta_e\left(1-\delta_\rho^2\right)\right]\left(1-\delta_\rho^2\right)\right\} = 0. \tag{4.28}$$

Using the fact that equation (4.27) is linear in δ_e, this system can be rewritten in the form

$$\delta_e = \mathcal{K}\left(\frac{3-\gamma}{\gamma-1}\right)\frac{\delta_\rho}{1-\delta_\rho^2} + \frac{\mathcal{G}\Theta_s}{2(\gamma-1)}, \tag{4.29}$$

$$\left(\delta_\rho^2 - 1\right)\left[\left(\delta_\rho^2-1\right)\frac{\mathcal{G}\Theta_e}{2\gamma} - \delta_\rho + \frac{\mathcal{G}\Theta_s}{2(\gamma-1)}\right] - \mathcal{K}\frac{\Gamma}{\gamma}\delta_\rho = 0, \tag{4.30}$$

where we have defined the non-dimensional parameters

$$\Gamma \equiv \frac{\gamma^2-\gamma+2}{\gamma-1}, \quad \mathcal{G} \equiv \frac{\Delta x \rho^n (a^x)^n}{e^n}, \quad \mathcal{K} \equiv \frac{(s^n)^2}{2\rho^n e^n}. \tag{4.31}$$

The first parameter, Γ, is nondimensional since γ itself is nondimensional. The parameters $\mathcal{G}$ and $\mathcal{K}$ represent ratios of energies. The parameter $\mathcal{G}$ is the ratio of the difference of gravitational potential energy from cell edge to cell edge to the total fluid energy in that cell. The parameter $\mathcal{K}$ represents the ratio of kinetic to total fluid energy in the cell. The procedure for obtaining both density and energy shifts will be to solve equation (4.30) for δ_ρ, and use this in equation (4.29) to find δ_e.

For quasi-steady flows with length scales determined by L, the gravitational energy density should roughly balance the total fluid energy density. This means that

$$0 < |N \Delta x \rho a^x| \sim e \tag{4.32}$$

on a grid with N cells and $L \sim N\Delta x$. This requires that

$$0 < |\mathcal{G}| \ll 1, \tag{4.33}$$

since $1/N \ll 1$ for a sufficiently refined grid. The last equation implies that equation (4.30) is a fourth order polynomial for δ_ρ with the small parameter $\mathcal{G}$ in front of the highest order term. This singularly perturbed algebraic equation will have one root that behaves as $\mathcal{O}(1/\mathcal{G})$ and will, therefore, be outside of the physically acceptable interval $(-1, 1)$. We seek the physical root by expanding the solutions in an asymptotic power series of $\mathcal{G}$. This expansion can be made easier if we note that given a solution, $\boldsymbol{\delta}(\boldsymbol{q}; \mathcal{G})$, of equations 4.27-4.28 for an acceleration a^x, it must be true that

$$-\boldsymbol{\delta}(\boldsymbol{q}; -\mathcal{G}) = \boldsymbol{\delta}(\boldsymbol{q}; \mathcal{G}) \tag{4.34}$$

is also a solution for a gravitational acceleration of the same magnitude, but in the opposite direction ($a^x \to -a^x$). We can, therefore, seek asymptotic solutions in odd powers of $\mathcal{G}$ of the form

$$\delta_e = \delta_e^1 \mathcal{G} + \delta_e^3 \mathcal{G}^3 + \mathcal{O}(\mathcal{G}^5), \tag{4.35}$$

$$\delta_\rho = \delta_\rho^1 \mathcal{G} + \delta_\rho^3 \mathcal{G}^3 + \mathcal{O}(\mathcal{G}^5). \tag{4.36}$$

Any even powers of $\mathcal{G}$ would destroy the property (4.34).

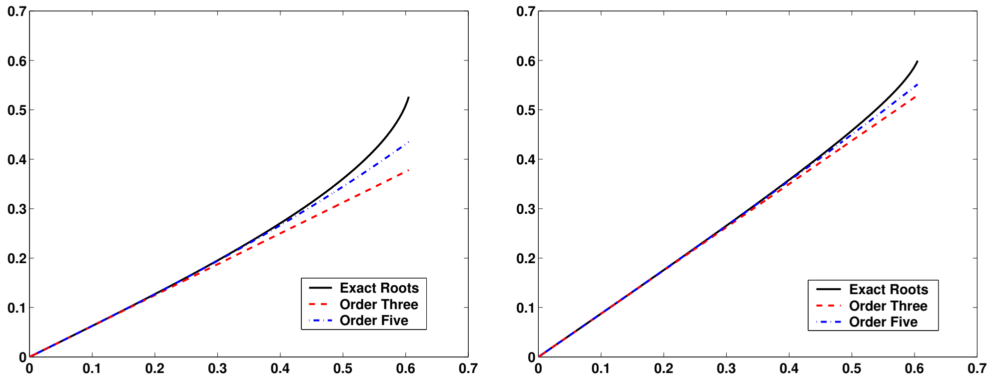
The expansion for δ_ρ has been computed through $\mathcal{O}(\mathcal{G}^3)$ and can be written as

$$\begin{aligned} \delta_\rho =& \frac{\gamma \Theta_s - (\gamma - 1)\,\Theta_e}{2\,(\gamma - 1)\,(\gamma - \mathcal{K}\Gamma)} \mathcal{G} + \\ & \frac{\gamma\,(\gamma\Theta_s - (\gamma-1)\,\Theta_e)^2}{8\,(\gamma-1)^2\,(\gamma - \mathcal{K}\Gamma)^4} \left[\left(1 - 2\frac{\mathcal{K}\Gamma}{\gamma}\right)\Theta_e + \frac{\mathcal{K}\Gamma}{\gamma - 1}\Theta_s \right] \mathcal{G}^3 + \mathcal{O}(\mathcal{G}^5), \end{aligned} \tag{4.37}$$

corresponding to the energy shift

$$\begin{aligned} \delta_e =& \left[\frac{\Theta_s}{2\,(\gamma - 1)} + \frac{\mathcal{K}\,(3-\gamma)\,(\gamma\Theta_s - (\gamma - 1)\Theta_e)}{2\,(\gamma - 1)^2\,(\gamma - \mathcal{K}\Gamma)} \right] \mathcal{G} + \\ & \frac{\gamma^2\,(\gamma\Theta_s - (\gamma - 1)\,\Theta_e)^2}{8\,(\gamma - 1)^2\,(\gamma - \mathcal{K})^4} \left[\frac{1}{\gamma - 1}\Theta_s - \frac{\mathcal{K}\Gamma}{\gamma^2}\Theta_e \right] \mathcal{G}^3 + \mathcal{O}(\mathcal{G}^5). \end{aligned} \tag{4.38}$$

Figures 4.3(a) and 4.3(b) show these solutions at various orders for a typical grid cell. The



(a) δ_ρ: the density shift (b) δ_e: the energy shift

Figure 4.3: A comparison of both third and fifth order asymptotics with the exact roots for (a) δ_ρ and (b) δ_e. The finite volume cell is taken to have $\mathcal{K} = 1/10$, $\gamma = 5/3$, $a^x = 1$, and $\mathcal{G} \in [0, 6/10]$. The solid line represents the exact roots as a function of $\mathcal{G}$, the dashed line is an $\mathcal{O}(\mathcal{G}^3)$ asymptotic approximation to these roots, and the dot-dashed line represents an $\mathcal{O}(\mathcal{G}^5)$ approximation.

solid line represents the numerical-exact solution, the dashed line shows that the $\mathcal{O}(\mathcal{G}^3)$ solution does very well for a while. Finally, the figures show that the $\mathcal{O}(\mathcal{G}^5)$ solution remains closer to the exact solution for larger $\mathcal{G}$.

The asymptotic shift solutions computed here can be used directly, or equation (4.37) can be used as an initial guess for an iteration procedure for equation (4.30). This iteration can be carried out to any given tolerance. A Newton iteration was used for the solution represented by solid lines in figures above. Once δ_ρ is known, equation (4.29) is used to find δ_e. Notice that an initial guess of zero fluid momentum is unnecessary for such an iterative process. If we use a one-step method ($M = 1$), then $\Theta_{[e]} = \Theta_{[s]} = 1$ and to first order these expressions for δ_ρ and δ_e reduce to equations (16) and (17) of [47], respectively. In the examples that follow, these shift solutions are used to modify the states.

4.2 A method based on modified flux differences

The quasi-steady method developed in the previous section is not a viable approach for the source terms generated by the geometry of manifolds. The disadvantages suffered by the method include the following:

1. The solution of the system (4.10) to find the amount by which states are modified is different for each system the method is applied to. Nonlinear fluxes have systems that, if a solution exists, are expensive to compute. We have seen this for the Euler equations.

2. If no solution to (4.10) exists, one is forced to return to a split method. This occurs for the Euler equations when momenta become only slightly larger than the steady value.

3. The quasi-steady waves do not naturally fit into the application of f-waves for spatially-varying flux functions.

Therefore, the quasi-steady method is appropriate for solving systems near a steady state, but is not a good choice for problems that are very dynamic. This is too restrictive for the goals of the algorithm developed in this dissertation.

However, the f-wave decomposition discussed in the previous chapter suggests a method that turns out to suffer none of the disadvantages described above. In this approach, the source term $\boldsymbol{\psi}$, is replaced by a function, $\Psi(\boldsymbol{q}, x)$, that consists of delta sources at each interface. This source function, therefore, has the form

$$\boldsymbol{\Psi}(\boldsymbol{q}, x) = \Delta x \sum_i \boldsymbol{\Psi}_{i-1/2}\, \delta(x - x_{i-1/2}), \tag{4.39}$$

so that $\boldsymbol{\Psi}$ represents the vector of source values at the interface where $x = x_{i-1/2}$. The source can be incorporated into Riemann problems by modifying equation (3.2) to include the influence of the source,

$$\boldsymbol{q}_{,t} + \mathcal{F}_{i-1/2}(\boldsymbol{q})_{,x} = \Delta x\, \boldsymbol{\Psi}_{i-1/2}\, \delta x - x_{i-1/2}, \tag{4.40}$$

with $\mathcal{F}_{i-1/2}$ given by (3.3) and solving it with the same initial data (2.9). The flux is no longer continuous across the interface due to the delta-function source located there,

$$\boldsymbol{f}_i(\boldsymbol{Q}_i) - \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1}) = \Delta x\, \boldsymbol{\Psi}_{i-1/2}. \tag{4.41}$$

This modified difference suggests that we define the f-waves based on the difference

$$\boldsymbol{f}_i(\boldsymbol{Q}_i) - \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1}) - \Delta x\, \boldsymbol{\Psi}_{i-1/2} = \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}^p_{i-1/2}, \tag{4.42}$$

instead of (3.12).

The approach can be summarized as:

step 1: Define the function $\boldsymbol{\Psi}(\boldsymbol{q}, x)$ so that the $\boldsymbol{\Psi}_{i-1/2}$ is known at each cell interface.

step 2: Decompose the modified flux difference (4.41) into f-waves as shown in (4.42).

step 3: Use the resulting f-waves in the update (3.26) as described in the previous chapter.

The implementation of this approach is much simpler than that for the quasi-steady method and is very natural to apply for systems with spatially-varying flux functions. Though f-waves are used to incorporate the source into Riemann problems, it is not necessary that the flux have spatial variation. An example of this method applied to an autonomous flux is presented in the following two sections.

Notice that there is no restriction on the application of this approach away from steady solutions as there is for the quasi-steady method. However, near steady solutions it is expected that flux gradients balance source terms so that

$$\frac{\boldsymbol{f}_i(\boldsymbol{Q}_i) - \boldsymbol{f}_{i-1}(\boldsymbol{Q}_{i-1})}{\Delta x} \approx \boldsymbol{\Psi}_{i-1/2}. \tag{4.43}$$

In this case, equation (4.41) is nearly satisfied leaving only a small difference on the left side of (4.42) to generate f-waves. Therefore, steady solutions and small perturbations of them are accurately captured by this modified flux difference approach. Examples of this will be seen in the following sections.

4.2.1 Accuracy of the modified flux method

It was demonstrated in Section 3.3 that the wave propagation method based on f-waves is second order provided that $\boldsymbol{q}$ and $\boldsymbol{f}$ remain smooth. This analysis performed there can be repeated taking into consideration the source $\boldsymbol{\psi}$. The Taylor series (3.34) becomes

$$\begin{aligned} \boldsymbol{q}(t_{n+1}, x_i) = \boldsymbol{q}(t,x)|_{t_n,x_i} &- \Delta t\left(\boldsymbol{f}(\boldsymbol{q},x)_{,x} - \boldsymbol{\psi}(\boldsymbol{q},x)\right)|_{t_n,x_i} \\ &+ \frac{1}{2}\Delta t^2\left[\partial \boldsymbol{f}/\partial \boldsymbol{q}\,(q,x)\left(\boldsymbol{f}(\boldsymbol{q},x)_{,x} - \boldsymbol{\psi}(\boldsymbol{q},x)\right)\right]_{,x}|_{t_n,x_i} \\ &- \partial \boldsymbol{\psi}/\partial \boldsymbol{q}(\boldsymbol{q},x)\left(\boldsymbol{f}(\boldsymbol{q},x)_{,x} - \boldsymbol{\psi}(\boldsymbol{q},x)\right)|_{t_n,x_i} + \mathcal{O}(\Delta t^3). \end{aligned} \tag{4.44}$$

The update (3.26) based on the f-waves defined in (4.42) captures all terms in the Taylor series to second order, except the last term. To make the method fully second order accurate, a term of the form

$$-\frac{\Delta t^2}{2\Delta x}\,\partial \boldsymbol{\psi}/\partial \boldsymbol{q}(\boldsymbol{Q}_i, x_i)\sum_{p=1}^{n_{eq}}\frac{1}{2}\left(\boldsymbol{\mathcal{Z}}^p_{i-1/2} + \boldsymbol{\mathcal{Z}}^p_{i+1/2}\right) \tag{4.45}$$

is added to the right side of the update (3.26).

4.3 An isothermal equilibrium

As an example of both methods described in this chapter, consider an isothermal equilibrium in one space dimension of the form

$$p_0(y) = \alpha\rho_0 = \exp\left[-g(y-y_0)\alpha\right], \tag{4.46}$$
$$u_0(y) = 0, \tag{4.47}$$
$$-\partial_y \phi_0(y) = g, \tag{4.48}$$

on the interval $y \in [0, 3/2]$ and with g constant and $\alpha = 1$. The grid is initialized to this equilibrium and the flow is then evolved.

Table 4.1: One-norm errors for the numerical solutions generated by three methods for the isothermal equilibrium. The method based on Strang splitting only computes the steady solution to first order. The methods based on the multi-step quasi-steady approximation and the modified flux difference both compute second order solutions.

	Strang Splitting	Quasi-Steady ($M = 2$)	Modified Flux Difference
dx	$\lVert p - p_0 \rVert_1$	$\lVert p - p_0 \rVert_1$	$\lVert p - p_0 \rVert_1$
1.500×10^{-2}	5.541×10^{-4}	9.561×10^{-7}	1.602×10^{-6}
7.500×10^{-3}	2.771×10^{-4}	2.365×10^{-7}	3.992×10^{-7}
3.750×10^{-3}	1.388×10^{-4}	5.880×10^{-8}	9.954×10^{-8}
1.875×10^{-3}	6.963×10^{-5}	1.466×10^{-8}	2.485×10^{-8}
order	1.00	2.01	2.00

Table 4.2: One-norm errors for the numerical solutions generated by the same three methods for the perturbed isothermal equilibrium. The method based on Strang splitting is, once again, only first order.

	Strang Splitting	Quasi-Steady ($M = 2$)	Modified Flux Difference
dx	$\lVert p - p_0 \rVert_1$	$\lVert p - p_0 \rVert_1$	$\lVert p - p_0 \rVert_1$
1.500×10^{-2}	5.553×10^{-4}	5.307×10^{-6}	5.881×10^{-6}
7.500×10^{-3}	2.775×10^{-4}	1.333×10^{-6}	1.483×10^{-6}
3.750×10^{-3}	1.390×10^{-4}	3.250×10^{-7}	3.775×10^{-7}
1.875×10^{-3}	6.970×10^{-5}	8.650×10^{-8}	1.386×10^{-7}
order	1.00	1.99	1.82

Four methods have been compared on this problem, a Strang-split method, the original quasi-steady method, a two-step quasi-steady method, and the method based on f-waves. Table 4.1 shows the one-norm errors generated by each method (the original quasi-steady errors are only included in Figure 4.5(a)). The table shows that both the quasi-steady method and the method based on f-waves generate second order solutions. The split method, however, is only first order on this problem. In addition, the split-method errors are a few orders of magnitude larger than those for the other two methods. Figure 4.5(a) shows a Log-Log plot of these errors. It is evident in this plot that both the quasi-steady methods, and the method based on f-waves do much better than the split method.

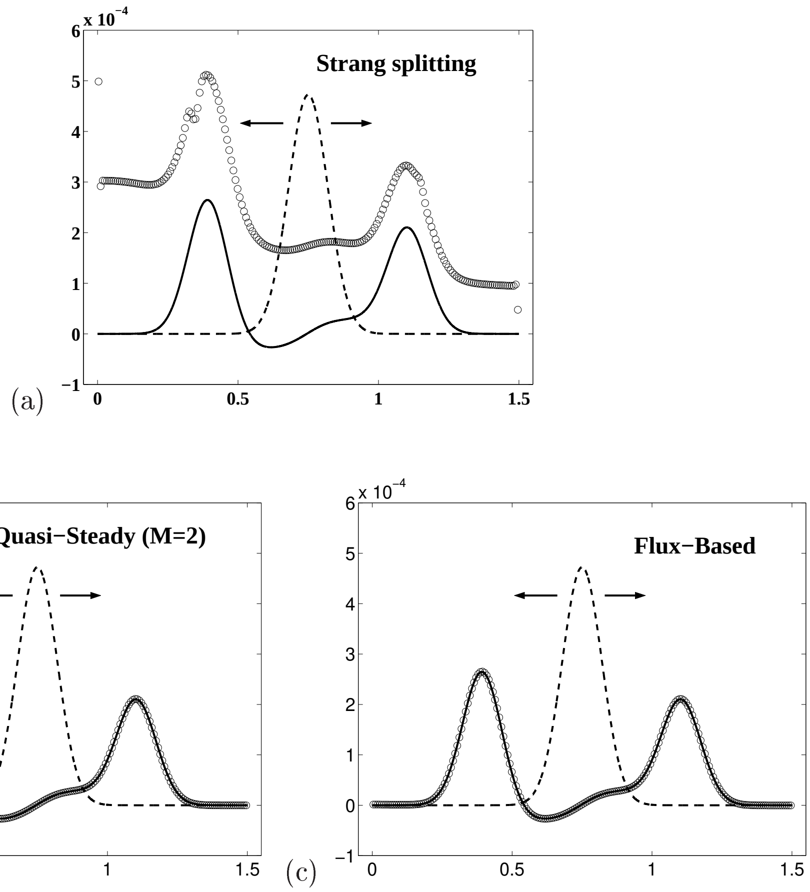


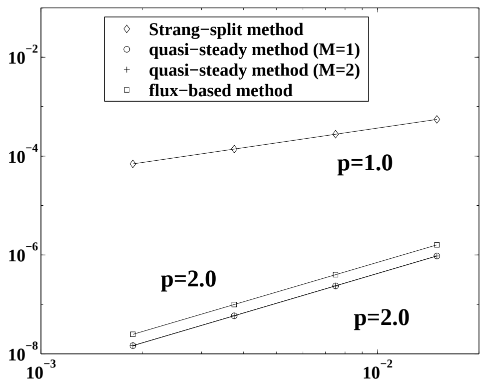
Figure 4.4: Numerical solutions for a smoothly perturbed isothermal Numerical results for the (a) Strang-split method, (b) quasi-steady method with $M = 2$, and (c) the method based on f-waves. Circles represent the computed values on a grid with 200 cells, the dashed line represents the initial condition, and the solid line was computed using the quasi-steady method on a grid with 8000 cells.

4.4 Perturbing the steady state

Now consider perturbing the equilibrium pressure field of the last section. We use a perturbed pressure defined by

$$p(t = 0, y) = p_0(y) + \epsilon \exp\left(-\frac{(y - y_0)^2}{\omega^2}\right), \tag{4.49}$$

where p_0 represents the equilibrium pressure and $0 < \epsilon \ll 1$. The resulting flow is essentially two acoustic pulses propagating away from this localized small disturbance. Figures 4.4(a) through 4.4(c) show the numerical results for pressure perturbation $p(t, y) - p_o(y)$ computed

(a)


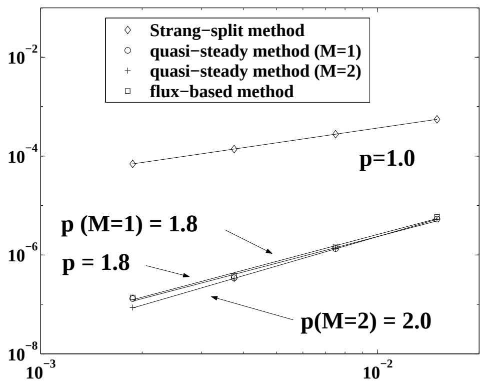
(b)


Figure 4.5: (a) Log-Log plot of the one-norm error in the perturbed pressure for the isothermal equilibrium. (b) Log-Log plot of the one-norm error for the perturbed equilibrium.

by the Strang-split method, a quasi-steady method with $M = 2$, and the method based on f-waves. Circles represent the computed values on a grid with 200 cells, the dashed line represents the initial condition, and the solid line was computed using the quasi-steady method on a grid with 8000 cells. The time was $t = 0.25$ when the computation was stopped. The pressure perturbation was set to $\epsilon = 4.76 \times 10^{-4}$.

Figure 4.4(a) demonstrates that when the perturbation is small enough, the split method fails to capture the correct value of the perturbed pressure. Notice that the split method has trouble maintaining the boundary values too. This is expected due to the fact that we are imposing boundary conditions on artificial problems equation (4.2) and equation (4.3). Both the quasi-steady method and the method based on f-waves, however, accurately capture the dynamics of the perturbation while maintaining the correct boundary values.

Table 4.2 shows the errors produced by each method. Once again, the two methods discussed in this chapter capture the dynamics to second order. The split method produces first order errors that are a few orders of magnitude larger than those produced by the other two methods. Figure 4.5(b) shows a Log-Log plot of these one-norm errors.

Typically, wave propagation methods are developed and applied to problems with discontinuities. It is important to know that the method is accurate in smooth regions of the solution, but it is equally important to know that discontinuities are sharply resolved. Figures 4.6(a) through 4.6(b) demonstrate that when the perturbation is discontinuous, it poses no problems for both methods discussed in this chapter. The split method fails to resolved the perturbation, or the boundary conditions.

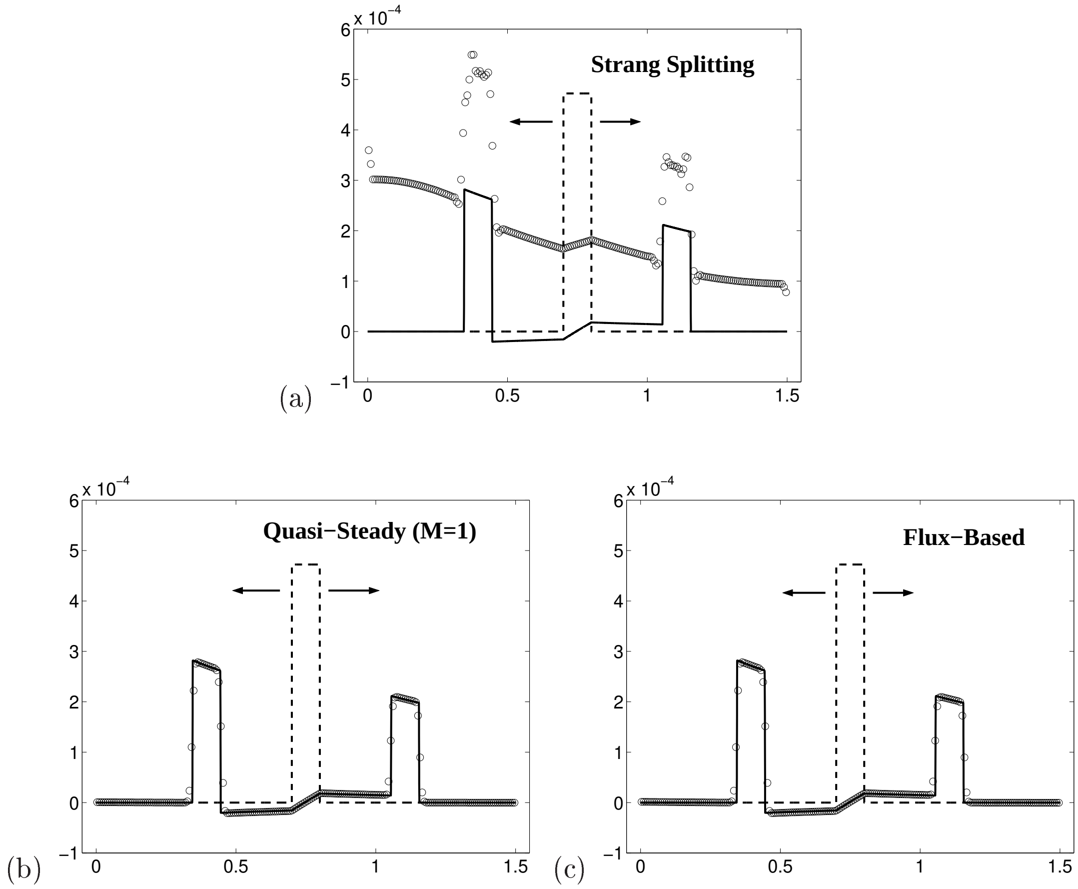


Figure 4.6: The same numerical methods are again applied to a perturbed isothermal equilibrium, but now the perturbation is not smooth. Both the quasi-steady method and the modified flux difference method handle the discontinuities very well.

Chapter 5

CONSERVATION LAWS ON MANIFOLDS

The natural mathematical arena to formulate conservation laws on curved manifolds is that of differential geometry. Ricci developed this branch of mathematics from 1887 to 1896. Subsequent work in differential geometry has made it an indespensible tool for solving problems in mathematical physics. In this chapter we develop the necessary ideas from differential geometry to formulate conservation laws on curved manifolds. We will derive the explicit form of both the spatially-varying flux and geometrically induced source term found in equation (1.4). The chapter also serves to set notational convention for subsequent use. We closely follow the notation of Misner, Thorne, and Wheeler [61] and Chandrasekhar [14].

Differential geometry is a vast field and we only attempt to cover necessary concepts in this chapter. For a complete study the reader is referred to the literature. Good sources for the methods of differential geometry from a physicist's point of view include Nakahara [62], Wald [79], Stewart [75], and Hawking and Ellis [30]. A more mathematical approach can be found in Bishop and Goldberg [8], Lovelock and Rund [51], or Lee [40]. Introductions to the ideas of differential geometry and general relativity can be found in Foster and Nightingale [25] or Schutz [71].

5.1 Manifolds

Differential geometry describes the geometric structure of a curved differentiable manifold, $\mathscr{M}$. A few examples of manifolds where briefly discussed in Chapter 1. For example, a manifold $\mathscr{M}$ may represent the nearly spherical surface of the earth, a four-dimensional curved spacetime in relativity theory, or the two-dimensional surface of an elastic membrane. We develop an intuitive notion of a manifold by the following heuristic definition :

Definition 5.1.1 *A manifold is a set of points that looks locally Euclidean in that this set can be entirely covered by a collection of local coordinate mappings.*

The main thing to notice about this definition of manifolds is that is suggests that it may take multiple coordinate maps to cover $\mathscr{M}$. We have encountered an example of this in Figure 1.1, where the surface of a sphere is covered by six coordinate patches. The use of these six coordinate patches eliminates the polar singularities in the standard Longitude-Latitude coordinate system in favor of more well behaved singularities at the points where three such grids meet.

The main reason that it may be necessary to use multiple coordinate maps to cover a manifold is that the global geometry of $\mathscr{M}$ may be very different from the local geometry near a point. It may be impossible to simply define a local coordinate system in the vicinity of a point and extend it to cover the entire manifold. For example, Figure 5.1 shows the

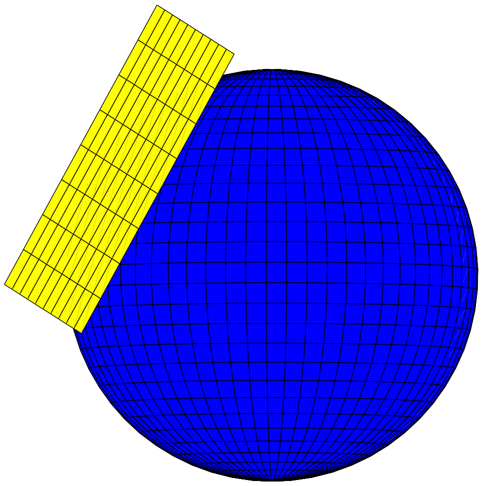


Figure 5.1: The extension of a locally flat coordinate patch tangent to $\mathcal{S}^2$ at a point.

tangent plane at an arbitrary point in $\mathcal{S}^2$. A Cartesian coordinate system on the tangent plane may describe the geometry in a very small region surrounding the point, but it fails to represent the geometry of $\mathcal{S}^2$ on a global scale.

It may be, however, that only a subset $\mathscr{U} \subset \mathscr{M}$ is of interest and that $\mathscr{U}$ can be covered by a single coordinate system. For example, suppose we are not interested in all of $\mathcal{S}^2$, but only the region shown in Figure 5.1 between the Tropic of Capricorn and the Tropic of Cancer. This region can be covered by a coordinate patch defined by

$$\begin{aligned} &\mathcal{X} : \mathscr{U} \subset \mathscr{M} \to \mathcal{U} \subset \mathbb{R}^2 \\ &\left(x^1, x^2\right) = (\theta, \phi)\,, \end{aligned} \tag{5.1}$$

with $\theta \in [\Theta_-, \Theta_+]$ and $\phi \in [0, 2\pi)$, where Θ_- and Θ_+ represent the degrees latitude of the Tropics of Capricorn and Cancer, respectively. If a geophysical flow were to be computed in the coordinate patch $\mathcal{U}$, then only the coordinate map $\mathcal{X}$ would be needed with boundary conditions at $x^1 = \Theta_-$ and $x^1 = \Theta+$, as well as periodic conditions at $x^2 = 0$. In Chapter 8 we calculate the in-fall of a pressure-less gas into a Schwarzschild black hole. In that case, much like the example here, we compute on a subset of the entire manifold for which the coordinates we use (Schwarzschild coordinates) have no singularities. Extending these coordinates to the full manifold, however, would introduce coordinate singularities similar to the polar singularities on $\mathcal{S}^2$ discussed above.

5.2 Tangent Vectors

In flat Euclidean space, a vector $\vec{v}$ is a geometric object that connects two distinct points $\{\mathcal{P}, \mathcal{Q}\}$ and possesses both a magnitude and direction. The magnitude of $\vec{v}$ is simply the

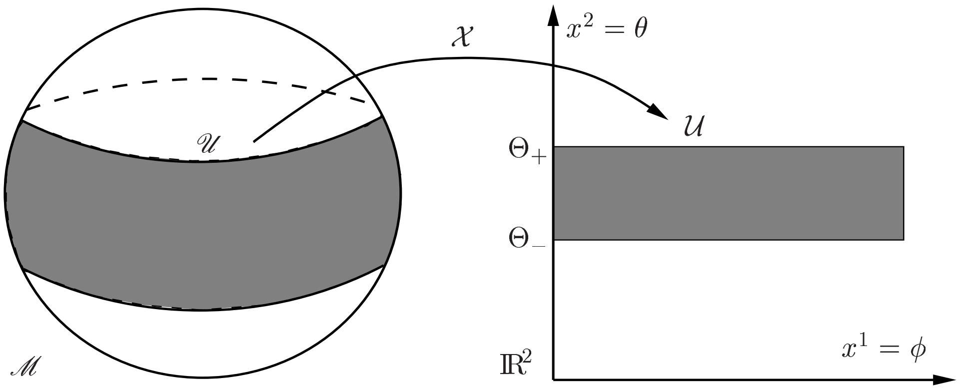


Figure 5.2: A local region $\mathscr{U} \subset \mathscr{M}$ of a 2-sphere is covered by the 2-dimensional coordinate map $\mathcal{X} = (\phi, \theta)$ in $\mathbb{R}^2$.

distance between the two points $|\mathcal{Q} - \mathcal{P}|$ and the direction is determined by connecting $\mathcal{P}$ and $\mathcal{Q}$ by a straight line $\mathcal{L}(\lambda) = \mathcal{P} + \lambda(\mathcal{Q} - \mathcal{P})$ as in Figure 5.3(a).

In a curved manifold, however, it is necessary to modify this flat-space notion of a vector. The concept of a straight line connecting $\mathcal{P}$ and $\mathcal{Q}$ does not even exist. The points can be connected with a curve, but in general there is no unique curve connecting distinct points in $\mathscr{M}$. Therefore, defining the direction and magnitude of a vector based on one such curve is ambiguous. Instead, a vector is defined in terms of the directional derivative tangent to a curve in $\mathscr{M}$ as shown in Figure 5.3(b). Before a precise definition for tangent vectors can be used, the representation of points, functions, and curves on $\mathscr{M}$ must be discussed.

Points in the manifold can be expressed in terms of coordinate functions by using a coordinate map $\mathcal{X} : \mathscr{M} \to \mathbb{R}^{n_d}$. For example, a point, $\mathcal{P} \in \mathscr{M}$ has the coordinate representation

$$\mathcal{X}(\mathcal{P}) = (x^1(\mathcal{P}), \cdots, x^{n_d}(\mathcal{P})), \tag{5.2}$$

where each x^j represent coordinate functions. Functions defined on the manifold can be expressed in the coordinate system as

$$f(\mathcal{X}(\mathcal{P})) = f(x^1(\mathcal{P}), \cdots, x^{n_d}(\mathcal{P})). \tag{5.3}$$

A useful set of curves passing through a point $\mathcal{P} \in \mathscr{M}$ are called coordinate curves and denoted by $C_k(\lambda)$. In terms of the coordinates, these curves can be expressed as

$$\mathcal{X}(C_k(\lambda)) = (x^1(\mathcal{P}), \cdots, \underbrace{x^k(\mathcal{P}) + \lambda}_{k^{\text{th}}\text{ location}}, \cdots, x^{n_d}(\mathcal{P})), \tag{5.4}$$

where all coordinate functions are held constant at $\mathcal{P}$, except for the k^{th} coordinate which varies with λ. A single component of the coordinate representation of $C_k(\lambda)$ passing through $\mathcal{P}$ is expressed as

$$x^j(C_k(\lambda)) = x^j(\mathcal{P}) + \lambda\,\delta^j_k. \tag{5.5}$$

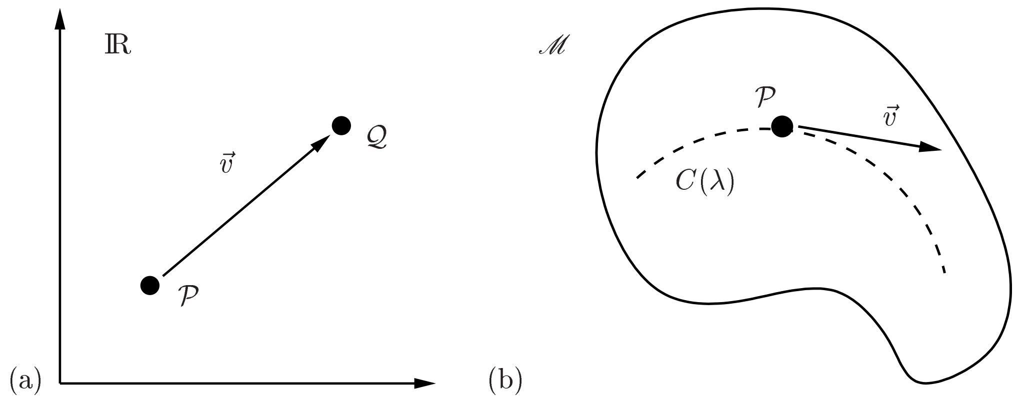


Figure 5.3: (a) A flat-space vector has both magnitude and direction defined by a parameterized line connecting two points $\mathcal{P}$ and $\mathcal{Q}$. (b) In curved space a vector is defined by a point-wise directional derivative operation on smooth functions.

The coordinate expressions described above are useful when defining a tangent vector on $\mathscr{M}$.

DEFINITION 5.2.1 *Let $\mathscr{F}$ be the set of all c^∞ functions defined on $\mathscr{M}$. A* **tangent vector** *at the point $\mathcal{P} = C(\lambda_0) \in \mathscr{M}$ along a curve $C(\lambda)$ is a map $\vec{v} : \mathscr{F} \to \mathbb{R}$ measuring the rate of change of a function $f \in \mathscr{F}$ along the curve $C(\lambda)$ at the point $\mathcal{P}$.*

Vectors that are tangent to coordinate curves are called canonical basis vectors and are denoted by $\vec{\partial}_{(k)}$. According to Definition 5.2.1, the action of $\vec{\partial}_{(k)}$ on a function $f(x^j(C_k(\lambda))) \in \mathscr{F}$ at $\mathcal{P}$ is the partial derivative

$$\begin{aligned} \vec{\partial}_{(k)} f &= \frac{d}{d\lambda}\Big(f(x^j(C_k(\lambda)))\Big)\Big|_{\mathcal{P}} \\ &= \sum_{j=1}^{n_d} \frac{\partial}{\partial x^j} f \,\frac{d}{d\lambda}\left(x^j(C_k(\lambda))\right)\Big|_{\mathcal{P}} \\ &= \frac{\partial}{\partial x^k} f, \end{aligned} \tag{5.6}$$

where we have used the fact that according to the coordinates (5.5), the tangent to the k^{th} coordinate curve (5.4) at the point $\mathcal{P}$ has a one in the k^{th} component and zeros in all the other components since $d(x^j(C_k(\lambda)))/d\lambda = \delta^j_k$.

Now consider the coordinate representation of the general curve shown in Figure 5.3(b)

$$\mathcal{X}(C(\lambda)) = \left(x^1(\lambda), \cdots, x^{n_d}(\lambda)\right), \tag{5.7}$$

where, as before, $x^k(\lambda)$ represents the k^{th}-parameterized coordinate function of the curve in the n_d-dimensional coordinate system $\mathcal{X}$. The direction tangent to C at the point $\mathcal{P}$ is now

$$\frac{d}{d\lambda}\Big(\mathcal{X}(C(\lambda))\Big)\Big|_{\mathcal{P}} = \left(\frac{d}{d\lambda}x^1(\lambda), \cdots, \frac{d}{d\lambda}x^{n_d}(\lambda)\right)\Big|_{\lambda=\lambda_0}$$
$$\equiv \left(\dot{x}^1, \cdots, \dot{x}^{n_d}\right)\Big|_{\lambda=\lambda_0}. \tag{5.8}$$

A vector $\vec{v}$ acts on a function restricted to this general curve, $f(C(\lambda)) = f(x^1(\lambda), \cdots, x^{n_d}(\lambda))$, to give the directional derivative

$$\begin{aligned}\vec{v}\, f &= \frac{d}{d\lambda}\Big(f(x^1(\lambda), \cdots, x^{n_d}(\lambda))\Big)\Big|_{\mathcal{P}=C(\lambda_0)} \\ &= \sum_{j=1}^{n_d}\left(\frac{dx^j}{d\lambda}\frac{\partial}{\partial x^j}\left(f(x^1, \cdots, x^{n_d})\right)\right)\Big|_{\mathcal{P}} \\ &\equiv \dot{x}^j\, \vec{\partial}_{(j)}\, f|_{\mathcal{P}},\end{aligned} \tag{5.9}$$

where, in the last step, we have used equation (5.6) and the convention that repeated indices are to be summed over.

Equation (5.9) serves to define the vector $\vec{v}$ as a linear combination of the set $\{\vec{\partial}_{(k)}\}$,

$$\vec{v} = v^k\, \vec{\partial}_{(k)}, \tag{5.10}$$

with components defined by $v^k \equiv \dot{x}^k$. In fact, the set $\{\vec{\partial}_{(k)}\}$ represents the basis for the n_d-dimensional vector space of all tangent vectors at the point $\mathcal{P} \in \mathscr{M}$. We denote this tangent space by $T_{\mathscr{M}}(\mathcal{P})$ and elements $\vec{v} \in T_{\mathscr{M}}(\mathcal{P})$ are called *tangent vectors* (or contravariant vectors). It is easy to show that the set $\{\vec{\partial}_{(k)}\}$ is linearly independent. Suppose $0 = v^k\, \vec{\partial}_{(k)}$, then it is also true that $0 = v^k\, \vec{\partial}_{(k)}[x^j] = v^k \delta^j_k = v^j$ demonstrating that they are linearly independent. Equation (5.9) demonstrates that any general vector in $T_{\mathscr{M}}(\mathcal{P})$ can be represented by this set of basis vectors so that $\mathrm{span}(\{\vec{\partial}_{(k)}\}) = T_{\mathscr{M}}(\mathcal{P})$. Figure 5.4 depicts the tangent space for a 2-dimensional manifold. The figure shows two coordinate curves C_1 and C_2 that go through the point $\mathcal{P}$ and generate the basis vectors $\{\vec{\partial}_{(1)}, \vec{\partial}_{(2)}\}$. It also shows a general curve C generating the vector $\vec{v}$, a linear combination of the basis vectors. One can imagine generating all other vectors in $T_{\mathscr{M}}(\mathcal{P})$ by choosing different curves $C(\lambda) \subset \mathscr{M}$ with different parameterizations that go through $\mathcal{P}$.

It is common to display the components of a vector as elements of an n_d-column vector, $\left[v^k\right] = (v^1, v^2, \cdots, v^{n_d})^T$, where the superscript T denotes a transpose. If we also define the $n_d \times n_d$ matrix E

$$E = \left(\left[\vec{\partial}_{(1)j}\right]\Big|\left[\vec{\partial}_{(2)j}\right]\Big|\cdots\Big|\left[\vec{\partial}_{(n_d)j}\right]\right) = \begin{pmatrix} \vec{\partial}_{(1)1} & \vec{\partial}_{(2)1} & \cdots & \vec{\partial}_{(n_d)1} \\ \vec{\partial}_{(1)2} & \vec{\partial}_{(2)2} & \cdots & \vec{\partial}_{(n_d)2} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{\partial}_{(1)n_d} & \vec{\partial}_{(2)n_d} & \cdots & \vec{\partial}_{(n_d)n_d} \end{pmatrix}, \tag{5.11}$$

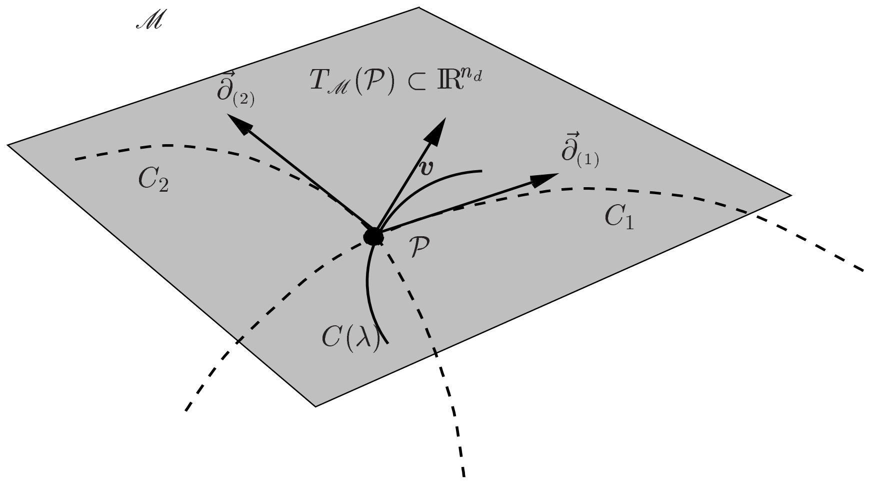


Figure 5.4: Tangent space $T_{\mathscr{M}}(\mathcal{P})$ at $\mathcal{P}$.

whose k^{th} column is the basis vector $\left[\vec{\partial}_{(k)j}\right]$ in matrix form with $j \in \{1, 2, \cdots, n_d\}$, then the vector (5.10) can be represented by

$$\vec{v} = E\left[v^k\right]. \tag{5.12}$$

We make use of this notation in much of what follows.

Since $T_{\mathscr{M}}(\mathcal{P})$ is an n_d-dimensional vector space, its basis is not unique. In fact, the basis can be any linearly independent set of vectors that span $T_{\mathscr{M}}(\mathcal{P})$. We can form a new basis by taking linear combinations of the canonical basis

$$\bar{E} = E\left[\Pi^j_{\bar{k}}\right], \tag{5.13}$$

so that the new set of basis vectors $\{\vec{e}_{(\bar{k})}\}$ make up the columns of $\bar{E}$. Each new basis vector can be written

$$\vec{e}_{(\bar{k})} = \Pi^j_{\bar{k}}\, \vec{\partial}_{(j)}, \tag{5.14}$$

where $\Pi^j_{\bar{k}}$ is the element in the j^{th} row and the $\bar{k}^{\text{th}}$ column of the transformation matrix $\left[\Pi^j_{\bar{k}}\right]$. Therefore, the $\bar{k}^{\text{th}}$ column of $\left[\Pi^j_{\bar{k}}\right]$ contains the coefficients for the linear combination (5.14).

A change of basis vectors will result in a different set of vector components $\left[v^{\bar{k}}\right]$. We discussed in Section 5.1 that it is often necessary to make use of multiple coordinate maps (and therefore multiple sets of basis vectors) when working on curved manifolds. Because of this, it is important to keep in mind that a tangent vector is a geometric object independent of any basis onto which it is projected. Using the notation from above, this can be expressed as

$$\vec{v} = E\left[v^k\right] = \bar{E}\left[v^{\bar{k}}\right]. \tag{5.15}$$

We can use this together with equation (5.13) to conclude that a transformation of basis vectors induces the following transformation of tangent vector components,

$$\left[v^j\right] = \left[\Pi^j_{\bar{k}}\right] \left[v^{\bar{k}}\right], \tag{5.16}$$

$$\left[v^{\bar{k}}\right] = \left[\Pi^j_{\bar{k}}\right]^{-1} \left[v^j\right] \equiv \left[\Pi^{\bar{k}}_j\right] \left[v^j\right]. \tag{5.17}$$

EXAMPLE 5.2.1 Consider an example where we scale the canonical basis in $\mathbb{R}^2$ to be the standard Cartesian basis

$$E = (\vec{e}_{(1)}|\vec{e}_{(2)}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{5.18}$$

We wish to express the components of the unit vector $\left[v^k\right] = \frac{1}{\sqrt{2}}(1,1)^T$ in a new basis that is a linear combination of the Cartesian basis vectors $\vec{e}_{(1)}$ and $\vec{e}_{(2)}$. The linear combination will be of the form

$$\begin{aligned} \vec{e}_{(\bar{1})} &= \alpha\, \vec{e}_{(1)} + \beta\, \vec{e}_{(2)} \\ \vec{e}_{(\bar{2})} &= -\alpha\, \vec{e}_{(1)} + \beta\, \vec{e}_{(2)}, \end{aligned} \tag{5.19}$$

where α, $\beta \neq 0$. Notice that the basis remains orthogonal if $|\alpha| = |\beta|$ since $\vec{e}_{(\bar{1})} \cdot \vec{e}_{(\bar{2})} = \beta^2 - \alpha^2$. In fact, the transformation represents a rotation (and possibly a scaling) if $\alpha = \beta$. In equations (5.13) and (5.14), the k^{th} column of $\left[\Pi^j_{\bar{k}}\right]$ represents the coefficients in the linear combination of $\vec{e}_{(j)}$ that are needed for the new basis vector $\vec{e}_{(\bar{k})}$. In this example the transformation matrix and its inverse take the simple form

$$\left[\Pi^j_{\bar{k}}\right] = \begin{pmatrix} \alpha & -\alpha \\ \beta & \beta \end{pmatrix} \qquad \text{and} \qquad \left[\Pi^j_{\bar{k}}\right]^{-1} = \frac{1}{2\alpha\beta} \begin{pmatrix} \beta & \alpha \\ -\beta & \alpha \end{pmatrix}. \tag{5.20}$$

The transformed matrix of basis vectors, according to equation (5.13), can be written

$$\bar{E} = \left(\vec{e}_{(\bar{1})}|\vec{e}_{(\bar{2})}\right) = \begin{pmatrix} \alpha & -\alpha \\ \beta & \beta \end{pmatrix}, \tag{5.21}$$

and the tangent vector components in this new basis $\{\vec{e}_{(\bar{k})}\}$ can be written using equation (5.17) as

$$\left[v^{\bar{k}}\right] = \frac{1}{2\sqrt{2}\alpha\beta} \begin{pmatrix} \alpha + \beta \\ \alpha - \beta \end{pmatrix}. \tag{5.22}$$

Notice that if $\alpha = \beta$ there is never a component in the direction of $\vec{e}_{(\bar{2})}$. In this case, as discussed above, the matrix $\left[\Pi^j_{\bar{k}}\right]$ always rotates the Cartesian basis through an angle of 45°. Since the vector $\left[v^k\right] = 1/\sqrt{2}(1,1)^T$ is along the 45° line, this vector will be entirely aligned with $\vec{e}_{(\bar{1})}$ in the transformed basis, leaving no component in the orthogonal direction of $\vec{e}_{(\bar{2})}$. The transformation discussed here with $\alpha = \beta$ is shown in Figure 5.5(b). The basis vectors are rotated by 45° so that the vector is along $\vec{e}_{(\bar{1})}$.

Since $\alpha = \beta$, but $\alpha \neq 1/\sqrt{2}$, the Cartesian basis is scaled by α as well as rotated by 45°. In this case, notice that the first component in $\left[v^{\bar{k}}\right]$ is inversely scaled with α so that $1/(\sqrt{2}\alpha)$ and the magnitude of the vector remains the same. The components of $\vec{v}$ in this new basis are shown in Figure 5.5(b).

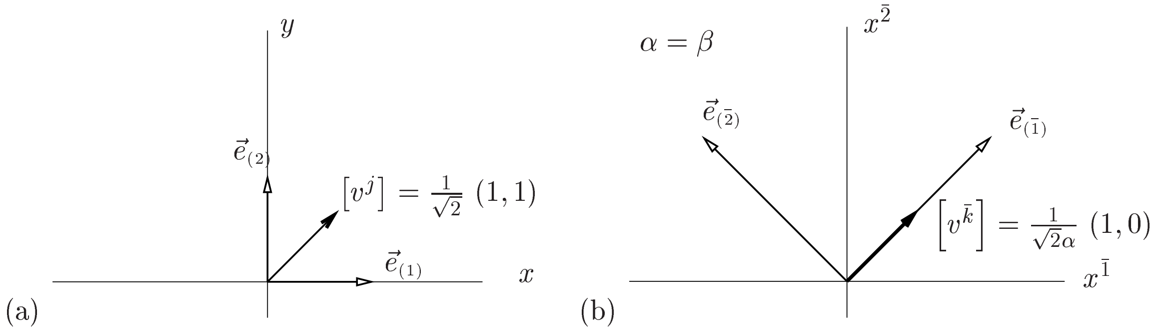


Figure 5.5: (a) The vector $\vec{v} = 1/\sqrt{2}(1,1)$ represented in a standard Cartesian basis. (b) The same vector represented in the new basis $\{\vec{e}_{(\bar{k})}\}$ that has been rotated through 45° and scaled by α. The component of $\vec{v}$ must be scaled by $1/\alpha$ to keep the magnitude the same.

5.2.1 Coordinate transformations

A special case of the transformation of basis vectors is that induced by a coordinate transformation of the form,

$$x^j \to x^j(x^{\bar{k}}). \tag{5.23}$$

The chain rule of partial derivatives requires that if $f \in \mathscr{F}$ then

$$\vec{\partial}_{(\bar{k})}[f] = \frac{\partial x^j}{\partial x^{\bar{k}}} \vec{\partial}_{(j)}[f], \tag{5.24}$$

from which is follows that the components of the transformation matrix $\left[\Pi^j_{\bar{k}}\right]$ have the form

$$\Pi^j_{\bar{k}} = \frac{\partial x^j}{\partial x^{\bar{k}}}. \tag{5.25}$$

A basis that is generated by transforming a canonical basis under a coordinate transformation (5.23) is called a *coordinate basis*. In a coordinate basis, the vector component v^k is a number that represents the projection of $\vec{v}$ along the coordinate curve C_k.

EXAMPLE 5.2.2 A coordinate transformation that gives rise to the transformed basis that we computed in Example 5.2.1 and show in Figure 5.5(b) is

$$x^1(x^{\bar{1}}, x^{\bar{2}}) = \alpha\,(x^{\bar{1}} - x^{\bar{2}}) \tag{5.26}$$

$$x^2(x^{\bar{1}}, x^{\bar{2}}) = \beta\,(x^{\bar{1}} + x^{\bar{2}}). \tag{5.27}$$

The transformation matrix has the form

$$\left[\Pi^j_{\bar{k}}\right] = \begin{pmatrix} \frac{\partial x^1}{\partial x^{\bar{1}}} & \frac{\partial x^1}{\partial x^{\bar{2}}} \\ \frac{\partial x^2}{\partial x^{\bar{1}}} & \frac{\partial x^2}{\partial x^{\bar{2}}} \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha \\ \beta & \beta \end{pmatrix}, \tag{5.28}$$

just as we found in (5.20).

5.3 Cotangent vectors

Associated with $T_{\mathscr{M}}(\mathcal{P})$ is a dual vector space $T^*_{\mathscr{M}}(\mathcal{P})$. Elements $\tilde{w} \in T^*_{\mathscr{M}}(\mathcal{P})$ are called **cotangent vectors** (also called covariant vectors, co-vectors, or one-forms) and are linear maps $\tilde{w} : T_{\mathscr{M}}(\mathcal{P}) \to \mathbb{R}$. A basis $\{\tilde{e}^{(k)}\}$ for $T^*_{\mathscr{M}}(\mathcal{P})$, associated with the basis $\{\vec{e}_{(k)}\}$ on $T_{\mathscr{M}}(\mathcal{P})$, maps any tangent vector $\vec{v} = v^k\,\vec{e}_{(k)}$ to its components,

$$\tilde{e}^{(j)}(\vec{v}) = \langle \tilde{e}^{(j)}, v^i\,\vec{e}_{(i)} \rangle = v^j \quad j \in \{1, 2, \cdots, n_d\}. \tag{5.29}$$

From this last equation (and the fact that the map is linear) it follows that

$$\tilde{e}^{(j)}(\vec{e}_{(i)}) = \langle \tilde{e}^{(j)}, \vec{e}_{(i)} \rangle = \delta^j_i. \tag{5.30}$$

The fact that the set $\{\tilde{e}^{(j)}\}$ constitutes a basis for the cotangent space follows if a general cotangent vector can be expressed as a linear combination of this set, and if they are linearly independent. To verify the former we define w_j to be the number to which the co-vector $\tilde{w}$ maps the basis vector $\vec{e}_{(j)}$ so that

$$\tilde{w}(\vec{e}_{(j)}) = \langle \tilde{w}, \vec{e}_{(j)} \rangle \equiv w_j, \tag{5.31}$$

and let $\vec{v} \in T_{\mathscr{M}}(\mathcal{P})$ be an arbitrary tangent vector. Using equation (5.30) and our definition of w_j we can write

$$\begin{aligned} \tilde{w}(\vec{v}) = \langle \tilde{w}, v^k\,\vec{e}_{(k)} \rangle &= w_k\,v^k \\ &= w_k \langle \tilde{e}^{(k)}, v^j\,\vec{e}_{(j)} \rangle \\ &= \langle w_k\,\tilde{e}^{(k)}, \vec{v} \rangle. \end{aligned} \tag{5.32}$$

Since this last equality is true for an arbitrary tangent vector, $\vec{v}$, it follows that

$$\tilde{w} = w_k\,\tilde{e}^{(k)}, \tag{5.33}$$

which demonstrates that $\mathrm{span}(\{\tilde{e}^{(k)}\}) = T^*_{\mathscr{M}}(\mathcal{P})$. Furthermore, if $w_k\,\tilde{e}^{(k)} = 0$, linear independence of the set $\{\tilde{e}^{(k)}\}$ follows because it is then also true that $w_k\langle \tilde{e}^{(k)}, \vec{e}_{(j)} \rangle = w_j = 0$. The set $\{\tilde{e}^{(k)}\}$, therefore, provides a basis for the cotangent space that will be referred to as the *co-basis*. In fact, the basis $\{\vec{e}_{(i)}\}$ and the co-basis $\{\tilde{e}^{(j)}\}$ are dual bases of the tangent and cotangent spaces, respectively.

The differential $\tilde{df}$ of any function $f \in \mathscr{F}$ is a one form whose action on the tangent vector $\vec{v}$ is defined by $\tilde{df}(\vec{v}) \equiv \vec{v}[f] \in \mathbb{R}$. Using (5.9) and the fact that $v^k = \dot{x}^k$ we can write

$$\tilde{df} = \frac{\partial f}{\partial x^k} \tilde{dx}^{(k)}, \tag{5.34}$$

so that the $\tilde{dx}^{(k)}$ form a the canonical basis for $T^*_{\mathscr{M}}(\mathcal{P})$. The cotangent basis $\{\tilde{dx}^{(k)}\}$ is naturally dual to the canonical tangent basis $\{\vec{\partial}_{(k)}\}$ since

$$\tilde{dx}^{(k)}(\vec{\partial}_{(j)}) = \frac{\partial}{\partial x^j} x^k = \delta^k_j. \tag{5.35}$$

A general co-basis $\{\tilde{e}^{(k)}\}$ may be the result of a transformation of a canonical co-basis.

It is often useful to write the components of a co-vector in matrix notation as we did for tangent vectors in Section 5.2. Cotangent vector components, however, are most naturally written as an n_d-row vector $[w_k] = (w_1, w_2, \cdots, w_{n_d})$. Using this matrix notation, a co-vector can be written

$$\tilde{w} = [w_k]\ E_*, \tag{5.36}$$

where we have defined

$$E_* = \begin{pmatrix} \left[\tilde{dx}^{(1)j}\right] \\ \hline \left[\tilde{dx}^{(2)j}\right] \\ \hline \vdots \\ \hline \left[\tilde{dx}^{(n_d)j}\right] \end{pmatrix} = \begin{pmatrix} \tilde{dx}^{(1)1} & \tilde{dx}^{(1)2} & \cdots & \tilde{dx}^{(1)n_d} \\ \tilde{dx}^{(2)1} & \tilde{dx}^{(2)2} & \cdots & \tilde{dx}^{(2)n_d} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{dx}^{(n_d)1} & \tilde{dx}^{(n_d)2} & \cdots & \tilde{dx}^{(n_d)n_d} \end{pmatrix}. \tag{5.37}$$

It is the k^{th} row that represents $\tilde{e}^{(k)}$ in E_*.

In Section 5.2 we derived the transformation of tangent vector components under the influence of the change of basis (5.13). Such a change of basis will induce a transformation of the co-basis since $E_* = E^{-1}$ and $\bar{E}_* = \bar{E}^{-1}$ are required by the duality of the basis and co-basis. Using this in the transformation (5.13) and rearranging we find

$$\bar{E}_* = \left[\Pi^j_{\bar{k}}\right]^{-1} E_*. \tag{5.38}$$

The co-vectors, like the tangent vectors, are geometric objects that are independent of the co-basis so that

$$\tilde{w} = [w_j]\ E_* = [w_{\bar{k}}]\ \bar{E}_*. \tag{5.39}$$

Using this last equation together with the co-basis transformation (5.38), we can write the transformation rule for the components of a cotangent vector

$$[w_j] = [w_{\bar{k}}] \left[\Pi^j_{\bar{k}}\right]^{-1} \equiv [w_{\bar{k}}] \left[\Pi^{\bar{k}}_j\right], \tag{5.40}$$

$$[w_{\bar{k}}] = [w_j] \left[\Pi^j_{\bar{k}}\right]. \tag{5.41}$$

Notice that the transformation to to barred basis involves $\left[\Pi^j_{\bar{k}}\right]$, and not the inverse as it did for tangent vector components in (5.17).

5.3.1 Coordinate transformations

Under a coordinate transformation of the form (5.23) discussed in Section 5.2.1, a canonical co-basis is transformed according to

$$\tilde{dx}^{(\bar{k})} = \frac{\partial x^{\bar{k}}}{\partial x^j}\, \tilde{dx}^{(j)}. \tag{5.42}$$

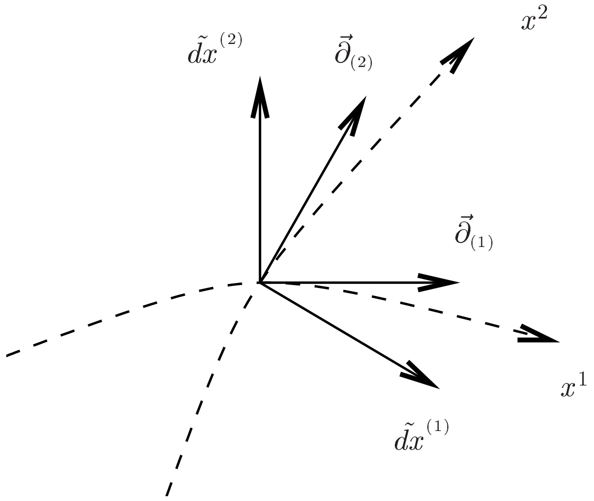


Figure 5.6: The basis $\{\vec{e}_{(i)}\}$ is tangent to the coordinate curves C_1 and C_2 at the point of intersection. The co-basis $\{\tilde{e}^{(i)}\}$, on the other hand, is perpendicular to the opposite coordinate curves.

The last equation makes use of the inverse transformation

$$x^{\bar{k}} \to x^{\bar{k}}(x^j), \tag{5.43}$$

instead of the transformation (5.23). This is the reason that the components of cotangent vectors transform using the inverse transformation matrix.

The duality of a coordinate basis and its co-basis has a geometric interpretation that is shown in Figure 5.6. The basis vectors were constructed based on the direction tangent to the coordinate curves C_k, and therefore point along coordinate lines. Figure 5.6 shows two such coordinate lines and the canonical tangent basis $\{\vec{\partial}_{(1)}, \vec{\partial}_{(2)}\}$ at the intersection point. Also shown are the co-basis vectors $\{\tilde{dx}^{(1)}, \tilde{dx}^{(2)}\}$ which, because of duality, point in a direction perpendicular to the opposite coordinate curve. That is, $\tilde{dx}^{(1)}$ is perpendicular to the tangent of C_2 and $\tilde{dx}^{(2)}$ is perpendicular to the tangent of C_1 at the point of intersection as shown in the figure.

5.4 Tensors

In the previous two sections we introduced tangent vectors and cotangent vectors. A tensor of type $\binom{r}{s}$ is a linear map $\overleftrightarrow{T} : \otimes^r T^*_{\mathscr{M}}(\mathcal{P}) \otimes^s T_{\mathscr{M}}(\mathcal{P}) \to \mathbb{R}$, where the symbols $\otimes^r$ and $\otimes^s$ indicate that the tensor takes r cotangent vectors, and s tangent vectors as arguments. Since tangent vectors map cotangent vectors to real numbers, they are type $\binom{1}{0}$ tensors. Similarly, cotangent vectors are type $\binom{0}{1}$ tensors since they map tangent vectors to $\mathbb{R}$.

The components of a type $\binom{r}{s}$ tensor may be displayed by projecting it onto a basis and its dual co-basis such that

$$\overset{\leftrightarrow}{T} = T^{j_1 \cdots j_r}{}_{k_1 \cdots k_s} \vec{e}_{(j_1)} \otimes \cdots \otimes \vec{e}_{(j_r)} \otimes \tilde{e}^{(k_1)} \otimes \cdots \otimes \tilde{e}^{(k_s)}. \tag{5.44}$$

Elements of the basis and its co-basis can be used (once again using duality) to map any tensor to its components. For example, the components of a type $\binom{2}{0}$ tensor $\overset{\leftrightarrow}{T}$ can be computed using two elements of the co-basis as

$$\overset{\leftrightarrow}{T}(\tilde{e}^{(i)}, \tilde{e}^{(j)}) = T^{ij}. \tag{5.45}$$

Similarly, the components of a type $\binom{0}{2}$ tensor are computed using two elements of the basis as

$$\overset{\leftrightarrow}{T}(\vec{e}_{(i)}, \vec{e}_{(j)}) = T_{ij}, \tag{5.46}$$

and components of a type $\binom{1}{1}$ tensor are computed using an element of both the basis and co-basis as

$$\overset{\leftrightarrow}{T}(\tilde{e}^{(i)}, \vec{e}_{(j)}) = T^i_j. \tag{5.47}$$

Multidimensional systems for which momentum is conserved have conservation laws with flux functions that are type $\binom{2}{0}$ symmetric tensors. We discuss such conservation laws in Section 5.8.

Components of the tensor defined in (5.44), like tangent vector components in Section 5.2 and cotangent vector components in Section 5.3, can be written in terms of different sets of basis and co-basis vectors. In this particular case there would be r components that transform as tangent vector components, and s components that transform as cotangent components.

In the next section the metric tensor, $\overset{\leftrightarrow}{h}$ is introduced. This is a second rank tensor often expressed in terms of its cotangent components h_{ij}. In matrix form this tensor can be expressed in terms of two bases

$$\overset{\leftrightarrow}{h} = [E_*]^T \; [h_{ij}] \; [E_*] = \left[\bar{E}_*\right]^T \; \left[h_{\bar{i}\bar{j}}\right] \; \left[\bar{E}_*\right], \tag{5.48}$$

as we saw for both tangent and cotangent vectors. Using the transformation properties in (5.38) for the co-basis, the cotangent components of this tensor transform according to the following:

$$[h_{ij}] = \left[\Pi^i_{\bar{k}}\right]^{-T} \; [h_{\bar{k}\bar{l}}] \; \left[\Pi^j_{\bar{l}}\right]^{-1} \tag{5.49}$$

$$[h_{\bar{k}\bar{l}}] = \left[\Pi^i_{\bar{k}}\right]^{T} \; [h_{ij}] \; \left[\Pi^j_{\bar{l}}\right]. \tag{5.50}$$

If we were interested in the transformation of the tangent components of a tensor, then the transformation (5.13) would be used.

5.5 The Metric Tensor

The metric tensor

$$\overset{\leftrightarrow}{h} = h_{ij}\,\tilde{e}^{(i)} \otimes \tilde{e}^{(j)} \tag{5.51}$$

is a symmetric type $\binom{0}{2}$ tensor that provides a measure of length on $\mathscr{M}$. The path length along a curve $C(\lambda)$ in $\mathscr{M}$ from $C(a)$ to $C(b)$ can be written

$$L = \int_a^b \left| h_{ij}\,\dot{x}^i(\lambda)\,\dot{x}^j(\lambda)\right|^{1/2} d\lambda, \tag{5.52}$$

where $x^k(\lambda)$ represents the coordinate parameterization of the curve. The integrand of the last integral suggests defining the infinitesimal interval dl between neighboring points of the manifold such that

$$dl^2 = h_{ij}\,dx^i\,dx^j, \tag{5.53}$$

which is consistent with equation (5.52). In the previous equation dx^i represents a displacement along the i^{th} coordinate curve, not the i^{th} co-basis $\tilde{dx}^{(i)}$. The metric components h_{ij} are computed using equation (5.45) as

$$h_{ij} = \overset{\leftrightarrow}{h}(\vec{e}_{(i)}, \vec{e}_{(j)}) \equiv \vec{e}_{(i)} \cdot \vec{e}_{(j)}. \tag{5.54}$$

The previous equation defines the *dot product* of two tangent vectors on $\mathscr{M}$ since

$$\begin{aligned} \overset{\leftrightarrow}{h}(\vec{v}, \vec{w}) = \overset{\leftrightarrow}{h}(v^i\,\vec{e}_{(i)}, w^j\,\vec{e}_{(j)}) &= v^i\,w^j\,\overset{\leftrightarrow}{h}(\vec{e}_{(i)}, \vec{e}_{(j)}) \\ &= v^i\,w^j\,h_{ij}, \end{aligned} \tag{5.55}$$

where, in the last equality, we have used equation (5.54) and linearity of $\overset{\leftrightarrow}{h}$.

On a well behaved manifold, the metric tensor must be non-singular. This requirement is equivalent to the requirement that the determinant h of the matrix $[h_{ij}]$ is non-singular at every point of the manifold. It follows that the matrix $[h_{ij}]$ has a unique inverse $[h_{ij}]^{-1} \equiv [h^{ij}]$ so that

$$h_{ij}\,h^{jk} = \delta_i^k. \tag{5.56}$$

The inverse metric is a type $\binom{2}{0}$ tensor with components

$$h^{ij} = \overset{\leftrightarrow}{h}(\tilde{e}^{(i)}, \tilde{e}^{(j)}) \equiv \tilde{e}^{(i)} \cdot \tilde{e}^{(j)}, \tag{5.57}$$

and defines the *dot product* of two cotangent vectors.

Since both the metric tensor and its inverse are non-degenerate, they define an isomorphism $T_{\mathscr{M}}(\mathcal{P}) \leftrightarrow T^*_{\mathscr{M}}(\mathcal{P})$, with

$$v^k = h^{kj}\,w_j \qquad \text{and} \qquad w_k = h_{kj}\,v^j. \tag{5.58}$$

It is, therefore, useful to terminate the notational distinction between the tangent vector and its co-vector. Instead, we distinguish co-vector components from tangent components by using subscripts for the former, and superscripts for the latter as shown in equation (5.58). This is true for tensors as well. Instead of calling a tensor type $\binom{2}{0}$ or type $\binom{0}{2}$, we simply refer to a *second rank* tensor with the isomorphism $T^{ij} \leftrightarrow T^i{}_j \leftrightarrow T_i^j \leftrightarrow T_{ij}$ understood.

5.6 Derivatives

The components of the basis, co-basis, and metric tensor will be functions of the spatial variables on $\mathscr{M}$. It is therefore necessary to modify the notion of a derivative on such a spatially dependent basis. Consider the gradient of the vector $\vec{v}$

$$\begin{aligned}\vec{\nabla}\vec{v} &= \tilde{e}^{(k)}\nabla_k \left(v^l\,\vec{e}_{(l)}\right) \\ &= \tilde{e}^{(k)}\left[v^l_{,k}\,\vec{e}_{(l)} + v^l\,\nabla_k\vec{e}_{(l)}\right], \end{aligned} \tag{5.59}$$

where we have, once again, made use of the notation that a comma followed by an index denotes the partial derivative, e.g., $\partial v^i/\partial x^j = v^i_{,j}$. Since $\nabla_k\vec{e}_{(l)}$ is a vector, we project it onto the basis

$$\nabla_k\vec{e}_{(l)} = \Gamma^j_{lk}\,\vec{e}_{(j)}, \tag{5.60}$$

where Γ^j_{lk} is the Christoffel symbol that represents the j^{th} component of the derivative of $\vec{e}_{(l)}$ with respect to x^k. The Christoffel symbols can also be written in terms of partial derivatives of metric components

$$\Gamma^j_{lk} = \frac{1}{2}\,h^{jm}\left(h_{mk,l} + h_{lm,k} - h_{lk,m}\right). \tag{5.61}$$

A contraction of the upper index with one of the lower indices can be written as

$$\Gamma^j_{mj} = \frac{1}{2}\,h^{jk}\,h_{jk,m} \tag{5.62}$$

where the fact that repeated indices are *dummy* indices and can be renamed has been used. Using cofactors of matrices, and the fact that $\left[h^{jk}\right]$ is the inverse of $[h_{jk}]$, it can be shown that the derivative of $h = \det(\overset{\leftrightarrow}{h})$ is

$$h_{,m} = h\,h^{jk}\,h_{jk,m}. \tag{5.63}$$

Using this in equation (5.62), the contracted Christoffel symbol can be expressed as

$$\Gamma^k_{ik} = \frac{\left(\sqrt{h}\right)_{,k}}{\sqrt{h}}. \tag{5.64}$$

This will be a useful expression throughout this thesis.

Using the Christoffel symbols in equation (5.59), one finds that

$$\vec{\nabla}\vec{v} = \left[v^l_{,k} + v^j\,\Gamma^l_{jk}\right]\tilde{e}^{(k)}\otimes\vec{e}_{(l)}, \tag{5.65}$$

where we have again used the fact that indices that have been summed on are dummy indices and can be renamed. This derivative is called the covariant derivative and is denoted by a semi-colon to distinguish it from the partial derivative. The covariant derivative of a vector is a type $\binom{1}{1}$ tensor whose components can be computed using equation (5.47). For example, the covariant derivative with respect to $\vec{e}_{(m)}$ of the i^{th} component of $\vec{v}$ is

$$\begin{aligned} \vec{\nabla}\vec{v}(\vec{e}_{(m)}, \tilde{e}^{(i)}) &= v^i_{,m} + v^j\,\Gamma^i_{jm} \\ &\equiv v^i_{;m} \end{aligned} \tag{5.66}$$

Since conservation laws depend on the divergence of flux vectors $\vec{\nabla}\cdot\vec{f}(\boldsymbol{q})$, Christoffel symbols will appear in the coordinate version of conservation laws on $\mathscr{M}$. Using equation (5.66) to write the divergence of a vector

$$\begin{aligned} \vec{\nabla}\cdot\vec{v} &= \vec{\nabla}\vec{v}(\vec{e}_{(k)}, \tilde{e}^{(k)}) \\ &= v^k_{,k} + v^j\,\Gamma^k_{ik} \\ &= \frac{1}{\sqrt{h}}\left(\sqrt{h}\,v^k\right)_{,k}, \end{aligned} \tag{5.67}$$

where h is the determinant of the metric tensor and equation (5.64) has been used.

It was mentioned in Section 5.4 that a multidimensional system in which momentum is conserved will have a flux represented by a second rank tensor. Therefore, we will also need to compute the divergence of such a tensor. The new term on the left hand side of equation (5.59) comes from the chain rule. The basis vector in the expression $\vec{v} = v^k\,\vec{e}_{(k)}$ depends on space and must be differentiated. For a second rank tensor $\overleftrightarrow{T} = T^{ij}\,\vec{e}_{(i)}\otimes\vec{e}_{(j)}$, there are two spatially-dependent basis vectors to differentiate using the chain rule. The covariant derivative of a second rank tensor has the form

$$\begin{aligned} \vec{\nabla}\overleftrightarrow{T} &= \tilde{e}^{(k)}\nabla_k\left(T^{lm}\,\vec{e}_{(l)}\,\vec{e}_{(m)}\right) \\ &= \tilde{e}^{(k)}\left[T^{lm}_{,k}\,\vec{e}_{(l)}\otimes\vec{e}_{(m)} + T^l m\,\Gamma^i_{lk}\,\vec{e}_{(i)}\otimes\vec{e}_{(m)} + T^{lm}\,\Gamma^i_{mk}\,\vec{e}_{(l)}\otimes\vec{e}_{(i)}\right] \\ &= \left(T^{lm}_{,k} + T^{im}\,\Gamma^l_{ik} + T^{li}\,\Gamma^m_{ik}\right)\,\tilde{e}^{(k)}\otimes\vec{e}_{(l)}\otimes\vec{e}_{(m)}. \end{aligned} \tag{5.68}$$

Notice that the covariant derivative raises the rank by one. The divergence of a second rank tensor can be computed from (5.68) by taking

$$\begin{aligned} \vec{\nabla}\cdot\overleftrightarrow{T} &= \vec{\nabla}\overleftrightarrow{T}(\vec{e}_{(k)}, \tilde{e}^{(k)}, \cdot) \\ &= \left(T^{km}_{,k} + T^{im}\,\Gamma^k_{ik} + T^{ki}\,\Gamma^m_{ik}\right)\,\vec{e}_{(m)} \\ &= \left(\frac{1}{\sqrt{h}}\left(\sqrt{h}\,T^{km}\right)_{,k} + T^{ki}\,\Gamma^m_{ik}\right)\,\vec{e}_{(m)}, \end{aligned} \tag{5.69}$$

where in this case the divergence is not a simple scalar, but a tangent vector. In fact, the divergence will always be of two ranks less than the gradient tensor because the divergence represents a contraction on two of the indices in the gradient tensor. The gradient of a vector, for example, was shown in equation (5.65) to be a second rank tensor. By contracting on two of the indices for the divergence (5.67), we are left with a scalar. The gradient of a second rank tensor, however, is of rank three (see (5.68)) so that the divergence is of rank one.

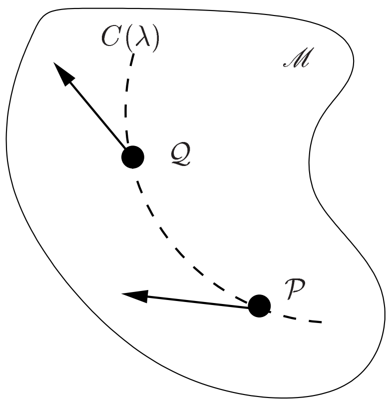


Figure 5.7: The parallel transport of a vector along a curve from a point $\mathcal{P}$ to $\mathcal{Q}$ in the manifold $\mathscr{M}$. Parallel transport with respect to the vector tangent to $C(\lambda)$ leaves the vector covariantly unchanged along the curve.

In the last equality of (5.69) we have again used equation (5.64), but could not eliminate all Christoffel symbols as we could for tangent vectors. This will play a major role in the development of our numerical algorithm in the next chapter.

5.7 Parallel Transport

In Euclidean space, vectors can be moved along arbitrary curves with no change to the components. This is no longer true when we are in a curved space. The best we can do is *parallel transport* a vector along a curve so that it *changes the least*. The *parallel transport* of a vector $\vec{v}$ along a curve with tangent vector $\vec{t} = (\dot{x}^1, \dot{x}^2, \cdots, \dot{x}^{n_d})^T$ is defined by

$$\vec{\nabla}_{\vec{t}}\vec{v} = \dot{x}^k\, v^m_{;k} = 0, \tag{5.70}$$

so that "*changes the least*" means that the vector $\vec{v}$ does not covariantly change along the curve. We will be interested in a discrete version of equation (5.70) in the next chapter so here we investigate the meaning of discrete parallel transport.

Consider two nearby points connected by a curve $C(\lambda)$ such that $\mathcal{P} = C(\lambda)$ and $\mathcal{Q} = C(\lambda + \Delta\lambda)$ as shown in Figure 5.7. The components v^i at $\mathcal{Q}$ can be written in terms of the components at $\mathcal{P}$ using a Taylor expansion

$$v^i(\mathcal{Q}) = v^i(\mathcal{P}) + \Delta\lambda\, \dot{v}^i(\mathcal{P}) + \frac{1}{2}\Delta\lambda^2\, \ddot{v}^i(\mathcal{P}) + \mathcal{O}\left(\Delta\lambda^3\right), \tag{5.71}$$

where we have dropped any terms of higher order than $\mathcal{O}(\Delta\lambda^2)$. Using the definition of parallel transport (5.70) together with the covariant derivative (5.66), we can write

$$\begin{aligned}\dot{x}^i\, v^j_{;i} &= \dot{x}^i \left(v^j_{,i} + v^m\, \Gamma^j_{mi}\right)\\ &= \dot{v}^j + v^m\, \dot{x}^i\, \Gamma^j_{mi}\\ &= 0.\end{aligned} \tag{5.72}$$

The fact that $\vec{v}$ has been parallel transported means that its components satisfy

$$\dot{v}^j = -v^m\, \dot{x}^i\, \Gamma^j_{mi}, \tag{5.73}$$

along the curve. Continuing in this manner we find

$$\ddot{v}^j = \dot{x}^k\, \dot{x}^l\, \Gamma^j_{mk}\, \Gamma^m_{nl}\, v^n - \dot{x}^k\, \dot{x}^l\, \mathcal{G}^j_{mkl}\, v^m - \ddot{x}^k\, \Gamma^j_{mk}\, v^m, \tag{5.74}$$

where we have defined $\mathcal{G}^j_{mkl} \equiv \Gamma^j_{mk,l}$. Using the above expressions for $\dot{v}^j$ and $\ddot{v}^j$ in the expansion (5.71) gives

$$\begin{aligned}v^j(\mathcal{Q}) = \big[\delta^j_m &- \left(\Delta\lambda\, \dot{x}^i + \frac{1}{2}\Delta\lambda^2\, \ddot{x}^i\right) \Gamma^j_{mi} + \frac{1}{2}\left(\Delta\lambda\, \dot{x}^k\right)\left(\Delta\lambda\, \dot{x}^l\right) \Gamma^j_{nk}\, \Gamma^n_{ml}\\ &- \frac{1}{2}\left(\Delta\lambda\, \dot{x}^k\right)\left(\Delta\lambda\, \dot{x}^l\right) \mathcal{G}^j_{mkl}\big]\, v^m(\mathcal{P}) + \mathcal{O}(\Delta\lambda^3).\end{aligned} \tag{5.75}$$

This last equation is written in terms of the components of the tangent vector $\vec{t}$, and the parameter λ. It is usually the case that we work directly with the coordinates. Therefore, we relate the change in coordinates between $\mathcal{P}$ and $\mathcal{Q}$ with the equation

$$\Delta x^i \equiv x^i(\mathcal{Q}) - x^i(\mathcal{P}) = \Delta\lambda\, \dot{x}^i + \frac{1}{2}\, \Delta\lambda^2\, \ddot{x}^i + \mathcal{O}(\Delta\lambda^3) \tag{5.76}$$

and the product of their difference can be written

$$\Delta x^i\, \Delta x^j = \left(\Delta\lambda\, \dot{x}^i\right)\left(\Delta\lambda\, \dot{x}^j\right) + \mathcal{O}(\Delta\lambda^3), \tag{5.77}$$

both expressions being correct through second order in $\Delta\lambda$. Finally, using these in the expression (5.75) for $v^j(\mathcal{Q})$, we find

$$v^j(\mathcal{Q}) = \left[\delta^j_m - \Delta x^i\, \Gamma^j_{mi} + \frac{1}{2}\, \Delta x^k\, \Delta x^l \left(\Gamma^j_{nk}\, \Gamma^n_{ml} - \mathcal{G}^j_{mkl}\right)\right] v^m. \tag{5.78}$$

This last equation prescribes a method for obtaining the components of the parallel propagated vector at the point $\mathcal{Q}$ based on the components of $\vec{v}$ at $\mathcal{P}$. The result is discrete in the sense that the components at $\mathcal{Q}$ are correct through second order in $\Delta\lambda$. We can think of multiplying the matrix representation of the components at $\mathcal{P}$ by a discrete operator of the form,

$$\left[v^j\right]_{\mathcal{Q}} = \left[P^j_m\right]\, \left[v^m\right]_{\mathcal{P}}, \tag{5.79}$$

with the matrix $\left[P^j_m\right]$ defined with components given by equation (5.78)

$$P^j_m = \delta^j_m - \Delta x^i\, \Gamma^j_{mi} + \frac{1}{2}\, \Delta x^k\, \Delta x^l \left(\Gamma^j_{nk}\, \Gamma^n_{ml} - \mathcal{G}^j_{mkl}\right). \tag{5.80}$$

5.8 Conservation Laws

We have mentioned throughout this thesis that a representation of the conservation law (1.2) on a manifold takes the form (1.4). The geometry of $\mathscr{M}$ is responsible for the spatially varying flux function $\boldsymbol{f}^{(i)}(\vec{x}, \boldsymbol{q})$, and the source term $\boldsymbol{\psi}(\vec{x}, \boldsymbol{q})$. In this section we derive the form of both the flux and source term as functions of the metric tensor and its derivatives.

The conservation law (1.2) is in *geometric* form so that no reference to a basis is needed. The divergence operation has geometric meaning independent of any coordinate system. In order to generate a numerical solution, however, it is necessary to write (1.2) in a specific coordinate system. We consider a coordinate system $\mathcal{X}$, with coordinate functions x^k for $k \in \{1, \cdots, n_d\}$, that covers a subset $\mathscr{U} \subset \mathscr{M}$ on which we are interested in computing the solution to the conservation law (1.2). Associated with $\mathcal{X}$ is a coordinate basis $\{\vec{e}_{(k)}\}$ and its co-basis $\{\tilde{e}^{(k)}\}$.

In order to write out the form of (1.2) in this basis, we need to know the rank of the flux vector (or tensor, see Section 5.6). Recall that the divergence of a vector is a scalar, and the divergence of a second rank tensor is a vector. We must, therefore, consider both scalar and vector conservation laws.

5.8.1 Scalar conservation laws

Scalar conservation laws describe the time-evolution of a scalar quantity q in a volume $\mathscr{V}$, whose value changes only as a result of the flux through the volume boundary $\partial\mathscr{V}$. The flux for a scalar conservation law is a vector $\vec{f}(q)$, a type $\binom{1}{0}$ tensor. For example, the conservation of mass density ρ of a fluid has a flux defined by the momentum $\rho\vec{u}$. In general, the scalar version of the conservation law (1.2) has the form

$$q_{,t} + \vec{\nabla} \cdot \vec{f}(q) = 0. \tag{5.81}$$

Using the second equality in the divergence of a vector (5.67), we write the conservation law on the coordinate basis as

$$q_{,t} + f^k(q)_{,k} = -\Gamma^k_{jk}\, f^j(q), \tag{5.82}$$

where the source term that has appeared involves the Christoffel symbols and therefore, by equation (5.61), the derivatives of metric coefficients. Notice that two of the indices in the Christoffel symbol are contracted on. This contraction allows us to use equation (5.64) to write (5.82) as

$$q_{,t} + \frac{1}{\sqrt{h}} \left(\sqrt{h}\, f^k(q)\right)_{,k} = 0, \tag{5.83}$$

where $h = det([h_{ij}])$ and we have absorbed the Christoffel symbol into a spatially-varying flux function

$$\vec{f}(\vec{x}, q) = \sqrt{h(\vec{x})}\, \vec{f}(q). \tag{5.84}$$

This last equation defines the specific functional form of the spatially-varying flux we have been referring to in equation (1.4).

5.8.2 Vector conservation laws

Many systems for which conservation laws are written conserve not just scalars, but vectors too. An example is a fluid that flows according to the conservation of mass, momentum, and energy. Mass and energy are both scalars, but the momentum is a vector. Vector conservation laws have fluxes that are rank two tensors. The vector form of the conservation law (1.2) is written

$$\vec{\boldsymbol{q}}_{,t} + \vec{\nabla} \cdot \overleftrightarrow{T}(\vec{\boldsymbol{q}}) = 0. \tag{5.85}$$

Using the second equality in the divergence for a second rank tensor (5.69), and taking the j^{th} component of the resulting vector we have

$$q^j_{,t} + T^{kj}_{,k} = -\Gamma^k_{mk}\, T^{jm} - \Gamma^j_{km}\, T^{km}. \tag{5.86}$$

In this last equation we have two sets of Christoffel symbols, one of which has a contraction of two indices as in the scalar equation case. The other set of Christoffel symbols, however, are new. We can, once again, use equation (5.64) to absorb the contracted Christoffel symbol into a spatially-varying flux function of the form (5.84). We are then left with one set of Christoffel symbols contracted with the components of the flux tensor,

$$q^j_{,t} + \frac{1}{\sqrt{h}}\left(\sqrt{h}\, T^{kj}\right)_{,k} = -\Gamma^j_{mk}\, T^{mk}. \tag{5.87}$$

This equation defines the specific functional form of the geometrically induced source term from equation (1.4),

$$\boldsymbol{\psi}^j(\vec{x}, \boldsymbol{q}) = -\Gamma^j_{mk}\, T^{mk}(\boldsymbol{q}), \tag{5.88}$$

where, in the last equation, we have written the tangent-vector components of the source term $\boldsymbol{\psi}$.

5.8.3 Systems of conservation laws

Most problems of interest, however, do not simply involve a single scalar or vector equation. Rather, they involve a system of both scalar and vector conservation laws. We have already seen such a system in Chapters 3 and 4 where examples involving the conservation laws for an inviscid fluid in n_d-dimensional motion were presented. This system consists of $n_d + 2$ equations that are presented in detail in Appendix B. Two of these equations are scalar equations representing the conservation of both mass, ρ, and energy, e, of the fluid. The other n_d equations represent conservation of each component, s^i, of the vector conservation law for the momentum, $\vec{s}$.

In order to apply high-resolution numerical methods to such a system, it is helpful to write the system in the standard form (1.2). To this end, the state vector

$$\boldsymbol{q} = (\rho, s^j, e)^T, \tag{5.89}$$

is defined with $n_d + 2$ components and $j \in \{1, \cdots, n_d\}$. The autonomous flux in the i^{th} coordinate direction is computed by gathering the vector components along $\vec{e}_{(i)}$ of the flux function for each individual equation. For example, the flux vector for the density equation is $\rho\,\vec{u}$ (see Appendix B) so that the first component of $\boldsymbol{f}^{(i)}$ is $\rho\,\vec{u}(\tilde{e}^{(i)})$. Similarly, the flux for the energy equation is $(p+e)\,\vec{u}$ so that the n_d+2 component of $\boldsymbol{f}^{(i)}$ is $(p+e)\,\vec{u}(\tilde{e}^{(i)})$, where p is the fluid pressure. Finally, the $1+j$ component of $\boldsymbol{f}^{(i)}$ is computed from the momentum flux tensor, $\overset{\leftrightarrow}{T}$ and has the form $\overset{\leftrightarrow}{T}(\tilde{e}^{(i)}, \tilde{e}^{(j)})$. Gathering each of these components, the flux along $\vec{e}_{(i)}$ can be written

$$\boldsymbol{f}^{(i)} = \left(\rho\,\vec{u}^{i}, T^{ij}, (p+e)\,u^{i}\right)^{T}, \tag{5.90}$$

where, as before, $j \in \{1, \cdots, n_d\}$, so that if $n_d = 2$ then T^{ij} represents the two components T^{i1} and T^{i2}.

It was shown in equation (5.87) that the representation of this set of equations on $\mathscr{M}$ will induce a source term for all vector components of $\boldsymbol{q}$. This source term is represented by $\boldsymbol{\psi}$ and will have the form

$$\boldsymbol{\psi} = \left(0, -\Gamma^{j}_{kl}\,T^{kl}, 0\right), \tag{5.91}$$

where, as demonstrated in (5.83), there is no source term for scalar equations.

Using this state vector, flux vector, and source term, the system of conservation laws represented in a coordinate basis that covers a subset, $\mathscr{U}$ of a manifold $\mathscr{M}$ has the form

$$\boldsymbol{q}_{,t} + \frac{1}{\sqrt{h}}\left(\sqrt{h}\,\boldsymbol{f}^{(i)}\right)_{,i} = \boldsymbol{\psi}, \tag{5.92}$$

where there is an implicit sum on i over the spatial directions $\{1, \cdots, n_d\}$. This last equation is written in the standard form for most high-resolution finite volume methods. In the chapters that follow, details of the development and implementation of a new class of wave propagation algorithms that generate high-resolution numerical solutions to (5.92) are presented.

Chapter 6

WAVE PROPAGATION ALGORITHMS ON CURVED MANIFOLDS

In the remaining chapters of this thesis the details for a class of wave propagation algorithms that generate stable, accurate, and high-resolution numerical solutions to hyperbolic conservation laws on smooth manifolds are presented. Much of what has been discussed in Chapters 1 through 5 has been for the purpose of developing and implementing this algorithm. Details of the algorithm together with computational examples are presented in this chapter. An implementation of the algorithm in the software package CLAWMAN is presented in Chapter 7. Finally, in Chapter 8 the algorithm is applied to various standard test problems in relativistic hydrodynamics.

In the previous chapter, equations (5.83) and (5.87) were derived and represent, respectively, scalar and vector conservation laws in a coordinate system that covers a subset $\mathscr{U}$ of a manifold $\mathscr{M}$. We found that the flux functions in both equations gain an explicit spatial dependence through the determinant of the metric tensor. We also found that vector conservation laws are written in *weak* conservation form due to the source term generated by the representation of the divergence of a second rank tensor in the coordinate basis. Systems of conservation laws involving both scalar and vector equations were found to be expressed as a system (5.92). If the metric tensor is that of flat Euclidean space, $[h_{ij}] = \mathrm{diag}(1,1,1)$ in $\mathbb{R}^3$, however, then $\sqrt{h} = 1$ and $\boldsymbol{\psi} = 0$ in that equation. In that case, system (5.92) would be in a standard form handled by most high-resolution finite volume methods. It is really the spatially-varying flux and the geometrically induced source term that prevent conservation laws on manifolds from being standard problems. The main novel features of the algorithm developed in this thesis are the following:

1. Spatially-varying flux functions arising from the geometry are handled using f-waves.

2. The geometrically induced source terms for vector laws are handled using parallel transport of vector components.

In the following sections these features of the algorithm are described in detail.

The functional form of the components of equation (5.92) is a result of the choice of both state variables $\boldsymbol{q} = (q^1, \cdots, q^{n_{eq}})^T$ and the particular coordinate basis in which the conservation laws are expressed. For example, instead of choosing the tangent components of momentum ρu^i in (5.89), the cotangent components ρu_i could have been chosen and would have lead to a different functional form of the components for both the flux and source term. Similarly, a change of coordinates would lead to different components. One way to handle the terms in system (5.92) induced by the geometry of $\mathscr{M}$ is to change the basis in which the equations are expressed in order to cast them in a form that reduces, or entirely eliminates, the source terms. For example, in [78] the classical Euler equations are written in *strong conservation* form (i.e. with no source terms) by expanding a curvilinear set of basis vectors into a Cartesian basis, thereby eliminating undifferentiated terms such as Christoffel symbols. The resulting system consists of a hybrid set of equations containing both Cartesian and curvilinear elements. On a manifold, however, a single Cartesian basis does not exist. In [65], symmetry of the manifold is used to generate a basis that reduces the number of components in the source $\boldsymbol{\psi}$. The reduction of the number of components in the source is dependent on both the system of conservation laws and the geometry of the manifold on which this system is to be solved. The algorithm described here does not attempt to recast the equations or to eliminate the source terms that appear for conservation laws on manifolds. Rather, the algorithm handles them the way they appear in system (5.92).

Various methods have been developed to handle systems of balance laws in the form (5.92). One simple approach that is often used is based on fractional time steps. This method generates the solution by alternating between solving the homogeneous version of the system (5.92) and the ordinary differential equations

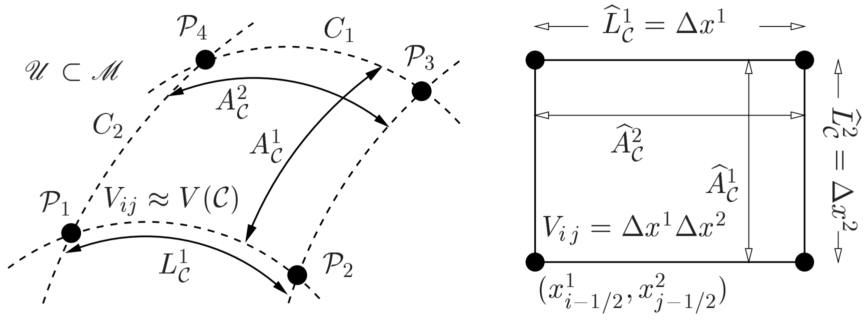
$$\boldsymbol{q}_{,t} = \boldsymbol{\psi}(\boldsymbol{q}, \vec{x}). \tag{6.1}$$

An example of such an *operator-split* method is developed in [20] and applied to the system of equations describing relativistic test fluids. Another approach is to use a *method of lines* discretization as in [21] and [67]. In this method the geometrically induced source term appears on the right hand side of the resulting system of ordinary differential equations (ODEs). A time-stepping algorithm that preserves the conservative properties of numerical flux functions computed at cell interfaces can then be applied to the system of ODEs.

The algorithms developed in this chapter share many features with the methods developed in [6] and [20]. Each of these methods, like the algorithms developed here, is based on the solution of Riemann problems and make use of limiters near discontinuities to generate non-oscillatory solutions. However, the basic cell updates, methods used to achieve spatial accuracy, and the way source terms are handled are different. Cell updates and spatial accuracy are generated using f-waves first discussed in Section 3.1, and applied to curved manifolds in Section 6.4. Source terms are handled *locally* by discrete parallel propagation operators that act on vector components of $\boldsymbol{q}$ and $\boldsymbol{f}$ as described in Section 6.5.

6.1 The coordinate grid

The algorithm presented in this chapter is implemented on a logically rectangular finite volume grid that covers a subset $\mathscr{U}$ of a smooth manifold $\mathscr{M}$. It is not necessary that a single coordinate system entirely cover $\mathscr{U}$. It may be convenient, as we saw in Figure 1.1, to cover $\mathscr{U}$ with multiple coordinate systems. In fact, the only requirement for the grid in the algorithm presented here is that the metric tensor remain smooth in each grid cell. This means that two adjacent cells can be described by two entirely different coordinate systems as long as within each cell the coordinate basis is smoothly varying. A jump in metric components across a cell interface can be handled by the way in which Riemann problems are solved and source terms are treated. This feature of the method is beneficial for manifolds like the sphere, $\mathcal{S}^2$, that are often covered by multiple coordinate systems.



(a) Physical cell

(b) Computational cell

Figure 6.1: (a) The physical cell consists of the four points $\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\}$ connected by the coordinate curves C_1 and C_2 and has a volume $V(\mathcal{C}) = \int_{\mathcal{C}} \sqrt{h}\, dx^1\, dx^2$. (b) The cell as it is represented on the computational grid. The volume is approximated by $V_{ij} = \sqrt{h_{ij}}\, \Delta x^1\, \Delta x^2 \approx V(\mathcal{C})$.

LeVeque and Rossmanith [50] have used this feature to compute geophysical flows on a sphere covered by the six coordinate systems shown in Figure 1.1.

Consider a single coordinate system, $\mathscr{X}$, that covers $\mathscr{U} \subset \mathscr{M}$, has coordinate functions x^k and basis vectors $\vec{\partial}_{(k)}$ for $k \in \{1, \cdots, n_d\}$. Both the examples and figures used here are two dimensional so that $n_d = 2$, but the geometric description is kept general for the extension of the algorithm to higher spatial dimensions. The coordinate region is discretized into a grid of logically rectangular finite volume cells

$$\begin{aligned} \mathcal{C}_{ij} &= \left(x^1_i - \Delta x^1/2, x^1_i + \Delta x^1/2\right) \times \left(x^2_j - \Delta x^2/2, x^2_j + \Delta x^2/2\right) \\ &\equiv \left(x^1_{i-1/2}, x^1_{i+1/2}\right) \times \left(x^2_{j-1/2}, x^2_{j+1/2}\right), \end{aligned} \tag{6.2}$$

with a natural extension for $\mathcal{C}_{ijk}$ in three dimensions. The numerical solution is evolved in the coordinate grid shown in Figure 6.1(b). It follows that coordinate lengths, cross-sectional areas, and volumes of this cell must be related to the corresponding physical quantities by the metric tensor. Useful coordinate quantities are depicted in Figure 6.1(b) and defined by the following:

$$\begin{aligned} \widehat{L}^k_{\mathcal{C}} &= \Delta x^k; && \text{the coordinate length of each edge in } \mathcal{C}, \\ \widehat{A}^k_{\mathcal{C}} &= \prod_{\substack{i=1 \\ i\neq k}}^{n_d} \Delta x^i; && \text{the coordinate cross-sectional area of } \mathcal{C} \text{ along } C_k, \\ \widehat{V}_{\mathcal{C}} &= \prod_{i=1}^{n_d} \Delta x^i; && \text{the coordinate volume of } \mathcal{C}. \end{aligned} \tag{6.3}$$

Often the coordinate quantities will refer to a particular cell in which case the subscript $\mathcal{C}$ is replaced by indices. For example, the coordinate lengths, cross-sectional areas, and volume in $\mathcal{C}_{ij}$ are denoted by $\widehat{L}^k_{|ij}$, $\widehat{A}^k_{|ij}$, and $\widehat{V}_{|ij}$, respectively, and the notation that subscripts following a $|$ denote cell indices. This notation will be used for quantities with which there may be confusion between subscripts that represent cell indices or cotangent components.

Each quantity listed above has a physical counterpart depicted in Figure 6.1(a). The physical and coordinate quantities are related to each other by the metric tensor. The true volume of the cell, $V(\mathcal{C})$, depicted in Figure 6.1(a), can be approximated by the physical volume $V_{\mathcal{C}}$ of the logically rectangular grid cell using the determinant of the metric tensor

$$\begin{aligned} V_{|ij} = \sqrt{h}_{|ij}\, \widehat{V}_{|ij} &= \sqrt{h}_{|ij}\, \Delta x^1\, \Delta x^2 \\ &\approx \int_{\mathcal{C}_{ij}} \sqrt{h}\, dx^1\, dx^2, \end{aligned} \tag{6.4}$$

where, once again, the notation that subscripts following a $|$ represent cell indices has been used. This notation is necessary since representing the determinant h in $\mathcal{C}_{ij}$ by h_{ij} may be confused with the cotangent components of the metric tensor $\overleftrightarrow{h}$ in the previous equation. Using subscripts that follow a $|$ to denote cell indices will eliminate any such confusion.

The physical distance traversed for a given coordinate displacement along each C_k can also be related to its coordinate counterpart, $\widehat{L}^k_{\mathcal{C}}$, using the metric. For example, the physical length of the lower edge of $\mathcal{C}_{ij}$, shown in Figure 6.1(a), is the distance along C_1 between points $\mathcal{P}_1$ and $\mathcal{P}_2$. This can be calculated using equation (5.53) together with the fact that $dx^2 = 0$ along this curve. The result is

$$\begin{aligned} L^1_{|i,j-1/2} = \sqrt{h}_{11|i,j-1/2}\, \widehat{L}^1_{|i,j-1/2} &= \sqrt{h}_{11|i,j-1/2}\, \Delta x^1 \\ &\approx |\mathcal{P}_2 - \mathcal{P}_1| \end{aligned} \tag{6.5}$$

where the notation defined above has been used so that the value of the metric component h_{ij} along the lower edge of $\mathcal{C}_{ij}$ has been denoted by $h_{ij|i,j-1/2}$. Note that the comma separating cell indices after the $|$ do not refer to partial differentiation. The physical distance along the left vertical edge between points $\mathcal{P}_1$ and $\mathcal{P}_4$ can be computed in a similar way using $\sqrt{h}_{22|i-1/2,j}$.

Finally, each physical cross section, $A^k_{\mathcal{C}}$, represents the area through which flux densities pass as material moves out of one cell and into another along x^k. They can be related to $\widehat{A}^k_{\mathcal{C}}$ by the *volume* element on the sub-manifold defined by the *coordinate plane* of the cross section. By keeping x^k fixed and allowing the other coordinates x^i, for $i \neq k$ to vary within the cell, a coordinate plane is established on which the restricted metric, $\overleftrightarrow{h}^{\perp}$, is defined. Since one of the coordinates is frozen in this sub-manifold, the volume element will have dimension $n_d - 1$. Therefore, in a three-dimensional grid, the volume elements of such sub-manifolds will be cross-sectional areas. However, in a two-dimensional grid, they represent one-dimensional lengths.

If, for example, $n_d = 2$, then there are two restricted metrics, one for each coordinate direction. The matrix representation of each restricted metric is 1×1:

$$\begin{aligned} \left[h_{ij}^{\perp^1}\right] &= h_{22}; \quad \text{along the } x^1 \text{ coordinate}, \\ \left[h_{ij}^{\perp^2}\right] &= h_{11}; \quad \text{along the } x^2 \text{ coordinate}. \end{aligned} \tag{6.6}$$

The fact that these restricted metrics only have one element is due to the fact that the coordinate planes in this case are represented by lines. We will use the name *coordinate plane* since it refers to the cross section of a cell with $n_d = 3$ which is, in fact, a plane. In three spatial dimensions the metric tensor restricted to the coordinate plane along x^1 will have the form

$$\left[h_{ij}^{\perp^1}\right] = \begin{pmatrix} h_{22} & h_{23} \\ h_{23} & h_{33} \end{pmatrix}, \tag{6.7}$$

with similar restricted metrics for the other two cross sections of the cell.

The volume element associated with $h_{ij}^{\perp^k}$ is

$$A_{\mathcal{C}}^k = \sqrt{h^{\perp^k}}\, \widehat{A}_{\mathcal{C}}^k = \sqrt{h^{\perp^k}} \prod_{\substack{i=1 \\ i \neq k}}^{n_d} \Delta x^i, \tag{6.8}$$

where $h^{\perp^k}$ is the determinant of the matrix $\left[h_{ij}^{\perp^k}\right]$. The last equation defines the physical area through which material passes when traveling in the x^k-direction. As mentioned above, if $n_d = 2$ this volume element represents a length, and if $n_d = 3$ it represents an area.

The preceding definitions of the physical quantities $V_{\mathcal{C}}$, $A_{\mathcal{C}}^k$, and $L_{\mathcal{C}}^k$ can be used to define the following important ratios for a coordinate grid cell:

$$\begin{aligned} \mathcal{V}_{\mathcal{C}} &= \frac{V_{\mathcal{C}}}{\widehat{V}_{\mathcal{C}}} = \sqrt{h_{\mathcal{C}}}; && \text{the ratio of physical to coordinate volume for the cell denoted by } \mathcal{C}, \\ \mathcal{A}_{\mathcal{C}}^k &= \frac{A_{\mathcal{C}}^k}{\widehat{A}_{\mathcal{C}}^k} = \sqrt{h^{\perp^k}}_{\mathcal{C}}; && \text{the ratio of physical to coordinate cross-sectional area along } x^k \text{ in } \mathcal{C}. \end{aligned} \tag{6.9}$$

The ratios $\mathcal{V}_{\mathcal{C}}$ and $\mathcal{A}_{\mathcal{C}}^k$ are defined as the square root of the determinant of the metric and restricted metric, respectively. Using properties of the determinant, these ratios can be related to the components of the inverse metric tensor by the relationship

$$\mathcal{A}^k = \mathcal{V} \sqrt{h^{kk}}, \tag{6.10}$$

which is true at each point in $\mathscr{U} \subset \mathscr{M}$. One can conclude from the previous equation that the physical volume on $\mathscr{M}$ is a product of A^k and a length perpendicular to the coordinate plane that defines A^k. For example, consider an orthogonal coordinate basis so that the off-diagonal elements of $[h_{ij}]$ vanish and $h^{kk} = 1/h_{kk}$. Use this in equation (6.10) and multiply through by $\widehat{V}$ to find

$$\begin{aligned} V &= A^k \sqrt{h_{kk}}\, \Delta x^k \\ &= A^k L^k. \end{aligned} \tag{6.11}$$

For a coordinate basis that is not orthogonal, the use of the inverse metric component means that the length is defined in terms of the cotangent basis which, as we saw in Figure 5.6, measures distance perpendicular to the coordinate plane that defines A^k.

6.2 Flux differences on a manifold

The definitions of the previous section can now be used to write down a numerical method on $\mathscr{M}$ that is based on flux differences and analogous to the multidimensional version of (2.5). The method is based on integrating the conservation law (1.1) from t to $t + \Delta t$ on the coordinate basis previously described. The result is

$$\begin{aligned}\int_{\mathcal{C}} \boldsymbol{q}(t+\Delta t, x^k)\sqrt{h}\, d^{n_d}x = \int_{\mathcal{C}} \boldsymbol{q}(t, x^k)\sqrt{h}\, d^{n_d}x \\ - \int_t^{t+\Delta t} \sum_{i=1}^{n_d} \Bigg(\int_{\widehat{A}^i_{\mathcal{C}+}} \boldsymbol{f}\left(\boldsymbol{q}(t, x_{i+1/2})\right) \cdot \sqrt{h}\, \vec{dx}^{(i)} \\ + \int_{\widehat{A}^i_{\mathcal{C}-}} \boldsymbol{f}\left(\boldsymbol{q}(t, x_{i-1/2})\right) \cdot \sqrt{h}\, \vec{dx}^{(i)} \Bigg) dt,\end{aligned} \tag{6.12}$$

where the subscript in $\widehat{A}^i_{\mathcal{C}\pm}$ denotes that the coordinate area is measured at the left or right edge of $\mathcal{C}$ along x^i. In the previous equation $d^{n_d}x = dx^1 \times \cdots \times dx^{n_d}$, and $\vec{dx}^{(i)}$ is the surface normal vector to the coordinate plane that defines the area $\widehat{A}^i_{\mathcal{C}}$.

Finite volume averages similar to (2.1) used in the standard algorithm are defined by

$$\boldsymbol{Q}^n_{\mathcal{C}} = \frac{1}{V_{\mathcal{C}}} \int_{\mathcal{C}} \boldsymbol{q}(t^n, x^k)\sqrt{h}\, d^{n_d}x \approx \frac{1}{V(\mathcal{C})} \int_{\mathcal{C}} \boldsymbol{q}(t^n, x^k)\sqrt{h}\, d^{n_d}x, \tag{6.13}$$

where the square root of the determinant is present to account for the volume of the cell. The numerical flux defined in (2.4) is generalized for a manifold as

$$\hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}} \approx \frac{1}{\Delta t} \int_t^{t+\Delta t} \left(\frac{1}{A^i_{\mathcal{C}}} \int_{\widehat{A}^i_{\mathcal{C}}} \boldsymbol{f}\left(\boldsymbol{q}(t, x^k)\right) \cdot \sqrt{h}\, \vec{dx}^{(i)} \right) dt, \tag{6.14}$$

and represents the time-averaged *flux per unit area* moving through the coordinate plane over a time step Δt. Using these definitions, equation (6.12) suggests the time-marching algorithm

$$\begin{aligned}\boldsymbol{Q}^{n+1}_{\mathcal{C}} &= \boldsymbol{Q}^n_{\mathcal{C}} - \frac{\Delta t}{V_{\mathcal{C}}} \sum_{i=1}^{n_d} \left(A^i_{\mathcal{C}+}\, \hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}+} - A^i_{\mathcal{C}-}\, \hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}-} \right) \\ &= \boldsymbol{Q}^n_{\mathcal{C}} - \frac{\Delta t\, \widehat{A}^i_{\mathcal{C}}}{\mathcal{V}_{\mathcal{C}}\, \widehat{V}_{\mathcal{C}}} \sum_{i=1}^{n_d} \left(\mathcal{A}^i_{\mathcal{C}+}\, \hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}+} - \mathcal{A}^i_{\mathcal{C}-}\, \hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}-} \right),\end{aligned} \tag{6.15}$$

where the definitions (6.9) have been used in the second equality. Using (6.3) we can write the numerical method as

$$\boldsymbol{Q}^{n+1}_{\mathcal{C}} = \boldsymbol{Q}^n_{\mathcal{C}} - \frac{\Delta t}{\mathcal{V}_{\mathcal{C}}\, \Delta x^i} \sum_{i=1}^{n_d} \left(\mathcal{A}^i_{\mathcal{C}+}\, \hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}+} - \mathcal{A}^i_{\mathcal{C}-}\, \hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}-} \right). \tag{6.16}$$

Notice that the previous numerical method is multidimensional since i ranges from 1 to n_d with $\mathcal{C}^-$ and $\mathcal{C}^-$ representing the "left" and "right" cell edges along x^i, respectively. This numerical method based on flux differences is the starting point for the derivation of a wave propagation algorithm.

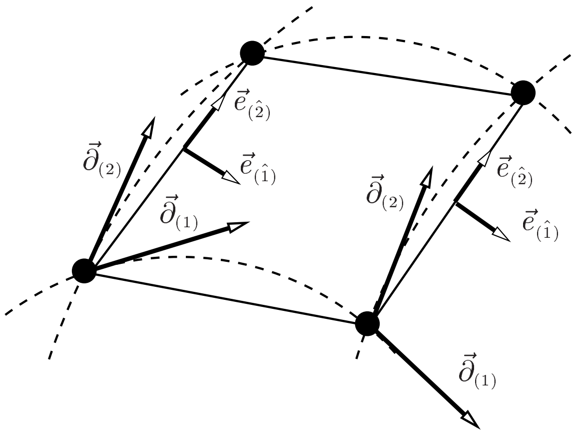


Figure 6.2: The orthonormal frames at each cell edge along x^1. Riemann problems are solved in these orthonormal frames to compute the flux per unit area at each cell interface.

6.3 *Orthonormal frames*

The numerical method (6.16) updates the solution in the coordinate grid. Therefore, the product $\mathcal{A}^i\,\hat{\boldsymbol{F}}^{(i)}$ should represent flux per coordinate area. Since $\mathcal{A}^i_{\mathcal{C}}$ represents the ratio of physical to coordinate area, $\hat{\boldsymbol{F}}^{(i)}$ represents the flux per unit of physical area through an interface along x^i. One way to be sure that flux per unit physical area is being computed in Riemann problems for a general metric is to locally remove the geometry of $\mathscr{M}$. This is always possible near a point on a manifold at which the metric is not singular.

Recall that in Figure 5.1 a local region of a point in $\mathcal{S}^2$ is represented by a tangent plane on which one can introduce Cartesian coordinates. Though this tangent plane does not represent the global geometry of the sphere, a small region surrounding the tangent point can be described by the Cartesian coordinates on the plane. In these coordinates the metric is the diagonal unit matrix with no spatial dependence. Therefore, the geometry has been locally removed. By solving Riemann problems in such locally Cartesian coordinates, we automatically compute the flux per unit physical area $\hat{\boldsymbol{F}}^{(i)}_{\mathcal{C}}$ needed for the numerical method (6.16). Another benefit to solving Riemann problems in locally Cartesian coordinates is that existing Cartesian solvers can be used. The implementation of such Cartesian solvers in the software package CLAWMAN is discussed in Chapter 7. For the first application of orthonormal coordinates in a general relativistic application see [67].

More will be said about this in Chapter 7.

Figure 6.2 shows a typical finite volume cell for $n_d = 2$. The edges of this cell are straight lines connecting the corner points, and the grid basis is tangent to the coordinate curves as shown in the figure. Flux per unit area, independent of the geometry in $\mathscr{M}$, can be computed in the orthonormal basis vectors $\{\vec{e}_{(\hat{k})}\}$ also depicted in Figure 6.2.

To compute the new basis, we use duality of the coordinate basis and co-basis that was discussed in Section 5.3.1. Figure 5.6 shows the orientation of the co-basis vectors relative to the tangent basis vectors. As the grid shown in Figure 6.2 is refined, the cell edges align with the coordinate curves shown as dashed lines in that figure. In the x^I direction, we would like to have $\vec{e}_{(\hat{I})}$ point in a direction orthogonal to the cell edge, and therefore to the I^{th} coordinate curve C_I. This is exactly the direction that the I^{th} co-basis vector $\tilde{dx}^{(I)}$ points. Recall from equation (5.57) that $|\tilde{dx}^{(I)}| = \sqrt{\tilde{dx}^{(I)} \cdot \tilde{dx}^{(I)}} = \sqrt{h^{II}}$. Using this together with the the fact that the metric provides an isomorphism between the basis and the co-basis (i.e., see Section 5.5), the first orthonormal basis vector can be written

$$\begin{aligned} \vec{e}_{(\hat{I})} = \frac{\tilde{dx}^{(I)}}{|\tilde{dx}^{(I)}|} &= \frac{\tilde{dx}^{(I)}}{\sqrt{h^{II}}} \\ &= \frac{h^{Ij}}{\sqrt{h^{II}}} \vec{\partial}_{(j)}, \end{aligned} \tag{6.17}$$

where x^I is to denote any of the n_d coordinate directions. A capital Latin index is used to signify a fixed index that is not to be summed on. We now have a single unit vector that is orthogonal to the cell interface along the direction x^I. The rest of the orthonormal basis vectors must now be computed.

The second vector in the basis can be chosen to be along $\vec{\partial}_{(J)}$, which is perpendicular to $\vec{e}_{(\hat{I})}$ by duality. This vector can be computed by normalizing $\vec{\partial}_{(J)}$ so that

$$\begin{aligned} \vec{e}_{(\hat{J})} &= \frac{\vec{\partial}_{(J)}}{\sqrt{\vec{\partial}_{(J)} \cdot \vec{\partial}_{(J)}}} \\ &= \frac{\vec{\partial}_{(J)}}{\sqrt{h_{JJ}}}, \end{aligned} \tag{6.18}$$

where the index J is fixed with $J \neq I$. If $n_d = 2$ that is the entire orthonormal set, but for $n_d = 3$ we need another basis vector $\vec{e}_{(\hat{K})}$. This vector must satisfy each of the following criteria:

1. $\vec{e}_{(\hat{K})} \cdot \vec{e}_{(\hat{I})} = 0$; the vector must be orthogonal to the first basis vector.

2. $\vec{e}_{(\hat{K})} \cdot \vec{e}_{(\hat{J})} = 0$; the vector must also be orthogonal to the second basis vector.

3. $\vec{e}_{(\hat{K})} \cdot \vec{e}_{(\hat{K})} = 1$; the vector must have unit length.

One choice that immediately guarantees that (1) is satisfied is

$$\vec{e}_{(\hat{K})} = \alpha\, \vec{\partial}_{(J)} + \beta\, \vec{\partial}_{(K)}. \tag{6.19}$$

The second two criteria are satisfied if α and β represent solutions to the following algebraic system

$$\alpha \sqrt{h_{JJ}} + \frac{\beta\, h_{JK}}{\sqrt{h_J J}} = 0 \tag{6.20}$$

$$\alpha^2\, h_{JJ} + \beta^2\, h_{KK} + 2\,\alpha\,\beta\, h_{JK} = 1. \tag{6.21}$$

Table 6.1: The role of the indices I, J, and K for each direction in which Riemann problems are solved.

	values		
directions	I	J	K
x^1	1	2	3
x^2	2	1	3
x^3	3	1	2

The first equation has solution

$$\alpha = -\frac{h_{JK}}{h_{JJ}}\beta, \tag{6.22}$$

and will be finite for a well behaved metric tensor since $h_{JJ} \neq 0$. This can be used in equation (6.21) to give

$$\beta^2 \left(h_{JJ}\, h_{KK} - (h_{JK})^2\right) = h_{JJ}, \tag{6.23}$$

which can immediately be solved for β. Notice, however, that the quantity inside the braces in the previous equation is just the determinant of the restricted metric tensor, $\overset{\leftrightarrow}{h}{}^{\perp^I}_{ij}$, that was discussed in the previous section. Using this we can write the solution for β as

$$\beta = \frac{\sqrt{h_{JJ}}}{\sqrt{h^{\perp^I}}}. \tag{6.24}$$

Therefore, the set of orthonormal basis vectors along the x^I direction can be written

$$\left\{\vec{e}_{(\hat{I})}, \vec{e}_{(\hat{J})}, \vec{e}_{(\hat{K})}\right\} = \left\{ \frac{h^{Ij}}{\sqrt{h^{II}}}\vec{\partial}_{(j)}\,,\ \frac{1}{\sqrt{h_{JJ}}}\vec{\partial}_{(J)}\,,\ -\frac{h_{JK}}{\sqrt{h_{JJ}}\sqrt{h^{\perp^I}}}\vec{\partial}_{(J)} + \frac{\sqrt{h_{JJ}}}{\sqrt{h^{\perp^I}}}\vec{\partial}_{(K)} \right\}, \tag{6.25}$$

where $I \neq J \neq K$.

The set of vectors (6.25) derived above represents an orthonormal basis for any choice of I, J, and K as long as they are different. Once I is set by the direction in which Riemann problems are being solved, however, the construction of $\vec{e}_{(\hat{J})}$ and $\vec{e}_{(\hat{K})}$ depends on the particular choice of J and K. In two space dimensions there is, of course, only one choice for J once I is set. In this case the basis defined in (6.25) is unique. In three dimensions, however, one must make a choice for J. For example, if $I = 1$, then J can be 2 or 3 with either choice resulting in a perfectly valid orthonormal basis. Table 6.1 shows the particular choice for the indices I, J, and K in the algorithm used here. This particular choice is made so that the transformation matrices in two dimensions look identical to the three dimensional transformations with one less row and column. Notice that in Table 6.1 I is always chosen to be the index of the direction in which Riemann problems are being solved.

In order to solve Riemann problems in the orthonormal basis in (6.25), it is necessary to transform both tangent and cotangent vector components into that frame. The matrix O^I is defined as the transformation matrix satisfying

$$\left[v^{\hat{k}}\right] = \left[O^{\hat{k}}_j\right]\left[v^j\right], \tag{6.26}$$

where $v^{\hat{k}}$ represents the components of $\vec{v}$ in the orthonormal frame. Since the basis (6.25) represents $\vec{e}_{(\hat{k})}$ as a linear combination of the canonical basis vectors, equation (5.14) can be used to write down the elements of the transformation matrix. Along the $\vec{\partial}_{(1)}$ direction we take $I = 1$, $J = 2$, and $K = 3$ so that equation (5.25) gives

$$\left[\Pi^{\hat{k}}_j\right] = \begin{pmatrix} \sqrt{h^{11}} & 0 & 0 \\ \frac{h^{12}}{\sqrt{h^{11}}} & \frac{1}{\sqrt{h_{22}}} & -\frac{h_{23}}{\sqrt{h_{22}}\sqrt{h^{\perp^1}}} \\ \frac{h^{13}}{\sqrt{h^{11}}} & 0 & \frac{h_{22}}{\sqrt{h^{\perp^1}}} \end{pmatrix}. \tag{6.27}$$

Using equation (5.17) together with the previous matrix, the transformation matrix that takes tangent vector components from the grid basis to the orthonormal basis along the x^1 direction can be written

$$\left[O^{\hat{k}}_j\right]^1 = \left[\Pi^{\hat{k}}_j\right]^{-1} = \begin{pmatrix} \frac{1}{\sqrt{h^{11}}} & 0 & 0 \\ \frac{h_{12}}{\sqrt{h_{22}}} & \sqrt{h_{22}} & \frac{h_{23}}{\sqrt{h_{22}}} \\ -\frac{h^{13}\sqrt{h^{\perp^1}}}{\sqrt{h^{11}}\sqrt{h_{22}}} & \frac{h_{23}}{\sqrt{h_2 2}} & \frac{\sqrt{h^{\perp^1}}}{\sqrt{h_{22}}} \end{pmatrix}. \tag{6.28}$$

The transformation matrices in the other directions can be computed in a similar way using Table 6.1 together with equation (6.25). The transformation matrix in the x^2 direction is

$$\left[O^{\hat{k}}_j\right]^2 = \begin{pmatrix} \sqrt{h_{11}} & \frac{h_{12}}{\sqrt{h_{11}}} & \frac{h_{13}}{\sqrt{h_{11}}} \\ 0 & \frac{1}{\sqrt{h^{22}}} & 0 \\ 0 & -\frac{h^{23}\sqrt{h^{\perp^2}}}{\sqrt{h^{22}}\sqrt{h_{11}}} & \frac{h^{\perp^2}}{\sqrt{h_1 1}} \end{pmatrix}, \tag{6.29}$$

and in the x^3 direction it has the form

$$\left[O^{\hat{k}}_j\right]^3 = \begin{pmatrix} \sqrt{h_{11}} & \frac{h_{12}}{\sqrt{h_{11}}} & \frac{h_{13}}{\sqrt{h_{11}}} \\ 0 & \frac{h^{\perp^3}}{\sqrt{h^{11}}} & -\frac{h^{32}\sqrt{h^{\perp^3}}}{\sqrt{h^{33}}\sqrt{h_{11}}} \\ 0 & 0 & \frac{1}{\sqrt{h^{33}}} \end{pmatrix}. \tag{6.30}$$

It was previously mentioned that all the examples in this thesis are done in manifolds with $n_d = 2$. The transformation matrices in that case are simply the first two rows and columns of the matrices above.

6.4 Scalar conservation laws

Let us first develop the algorithm for a scalar conservation law of the form (5.83). Consider the scalar advection equation

$$q(t, \vec{x})_{,t} + \vec{\nabla} \cdot q\, \vec{u} = 0, \tag{6.31}$$

in which the concentration q is advected with velocity $\vec{v}$. Using equation (5.83), this equation can be written on the coordinate basis of the finite volume grid as

$$q(t, x^k)_{,t} + \frac{1}{\sqrt{h}} \left(\sqrt{h}\, q\, u^i \right)_{,i} = 0, \tag{6.32}$$

where, once again, a subscript following a comma denotes a partial derivative and a sum is implied on repeated indices if one is up and the other down. Since this equation is scalar, there is no source term generated by the geometry. In this case, the algorithm only needs to handle the spatially-varying flux in equation (6.32). We have already discussed the details of a wave propagation algorithm based on f-waves in Chapter 3 and in the recent paper [4]. The f-waves were shown to handle spatially-varying flux functions both efficiently and accurately. The application of f-waves on manifolds is now developed.

The finite volume approximation, $\boldsymbol{Q}^n_{|ij}$, will be defined by the cell average (6.13) with $n_d = 2$. The discretized spatially-varying flux has the form

$$\boldsymbol{f}^{(k)}_{|ij}(\boldsymbol{Q}_{|ij}) = \mathcal{V}_{|ij}\, \boldsymbol{Q}_{|ij}\, u^k_{|ij}, \tag{6.33}$$

where the velocity components $u^k_{|ij}$ are discretized at cell centers so that

$$u^k_{|ij} = u^k(x^1_i, x^2_j), \tag{6.34}$$

and the point (x^1_i, x^2_j) is at the center of $\mathcal{C}_{ij}$. For the discussion here, we consider a velocity such that $u^k_{|ij} > 0$ throughout the grid.

The method developed here is based on solving one-dimensional Riemann problems along each coordinate direction in the orthonormal frames discussed in the previous section. Since equation (6.32) is scalar, the eigenvectors in each direction needed for a decomposition in the Riemann solution are simply $r^p = 1$ and have no geometric information associated with them. The eigenvalues suggested by (6.33), however, have the form

$$\lambda^k = \sqrt{h}\, u^k, \tag{6.35}$$

which contains information about the geometry through the determinant of the metric. Therefore, in order to compute $\hat{\boldsymbol{F}}^{(k)}$, we transform to an orthonormal frame.

Consider the Godunov numerical fluxes $\hat{\boldsymbol{F}}^{(1)}_{|i-1/2,j}$, and $\hat{\boldsymbol{F}}^{(1)}_{|i+1/2,j}$ that result from solving Riemann problems along x^1 in the orthonormal frames at the left and right edges of $\mathcal{C}_{ij}$, respectively. Since the velocity is everywhere positive, they have the form

$$\begin{aligned} \hat{\boldsymbol{F}}^{(\hat{1})}_{|i-1/2,j} &= \boldsymbol{Q}_{|i-1,j}\, u^{\hat{1}}_{|i-1,j}, \\ \hat{\boldsymbol{F}}^{(\hat{1})}_{|i+1/2,j} &= \boldsymbol{Q}_{|ij}\, u^{\hat{1}}_{|ij}. \end{aligned} \tag{6.36}$$

These fluxes represent the flux per unit area, but the are written in terms of the orthonormal tangent components. At the left edge of the cell, these components can be written in terms of the canonical components using the fact that

$$\begin{aligned} u^{\hat{1}}_{|i-1/2,j} = \vec{e}_{(\hat{1})} \cdot \vec{v}_{|i-1,j} &= \frac{\tilde{dx}^{(1)}}{\sqrt{h^{11}}_{|i-1,j}} \cdot \vec{v}_{|i-1,j} \\ &= \frac{v^1_{|i-1,j}}{\sqrt{h^{11}}_{|i-1,j}}. \end{aligned} \tag{6.37}$$

Similarly, $u^{\hat{1}}_{|i+1/2,j}$ at the right edge is defined so that

$$u^{\hat{1}}_{|i+1/2,j} = \frac{v^1_{ij}}{\sqrt{h^{11}}_{|ij}}. \tag{6.38}$$

The flux per unit area along the left and right edges of the cell can now be rewritten in terms of the coordinate velocity components as

$$\begin{aligned} \hat{\boldsymbol{F}}^{(1)}_{|i-1/2,j} &= \frac{\boldsymbol{Q}_{|i-1,j}\, u^1_{|i-1,j}}{\sqrt{h^{11}}_{|i-1,j}}, \\ \hat{\boldsymbol{F}}^{(1)}_{|i+1/2,j} &= \frac{\boldsymbol{Q}_{|ij}\, u^1_{|ij}}{\sqrt{h^{11}}_{|ij}}. \end{aligned} \tag{6.39}$$

Using these numerical fluxes together with the fluxes in the x^2 direction denoted by $\hat{\boldsymbol{F}}^{(2)}_{|i,j-1/2}$ and $\hat{\boldsymbol{F}}^{(2)}_{|i,j+1/2}$, the numerical method (6.16) can be written

$$\begin{aligned} \boldsymbol{Q}^{n+1}_{|ij} = \boldsymbol{Q}^n_{|ij} &- \frac{\Delta t}{\mathcal{V}_{|ij}\,\Delta x^1}\left(\frac{\mathcal{A}^1_{|ij}}{\sqrt{h^{11}}_{|ij}}\,\boldsymbol{Q}_{|ij}\,u^1_{|ij} - \frac{\mathcal{A}^1_{|i-1,j}}{\sqrt{h^{11}}_{|i-1,j}}\,\boldsymbol{Q}_{|i-1,j}\,u^1_{|i-1,j}\right) \\ &- \frac{\Delta t}{\mathcal{V}_{|ij}\,\Delta x^2}\left(\frac{\mathcal{A}^2_{|ij}}{\sqrt{h^{22}}_{|ij}}\,\boldsymbol{Q}_{|ij}\,u^2_{|ij} - \frac{\mathcal{A}^2_{|i,j-1}}{\sqrt{h^{22}}_{|i,j-1}}\,\boldsymbol{Q}_{|i,j-1}\,u^2_{|i,j-1}\right). \end{aligned} \tag{6.40}$$

Using equation (6.10) that relates the components of the inverse metric tensor to the ratios defined in equation (6.9), we can write the previous numerical update as

$$\begin{aligned} \boldsymbol{Q}^{n+1}_{|ij} &= \boldsymbol{Q}^n_{|ij} - \frac{\Delta t}{\mathcal{V}_{|ij}\,\Delta x^1}\left(\mathcal{V}_{|ij}\,\boldsymbol{Q}_{|ij}\,u^1_{|ij} - \mathcal{V}_{|i-1,j}\,\boldsymbol{Q}_{|i-1,j}\,u^1_{|i-1,j}\right) \\ &\quad - \frac{\Delta t}{\mathcal{V}_{|ij}\,\Delta x^2}\left(\mathcal{V}_{|ij}\,\boldsymbol{Q}_{|ij}\,u^2_{|ij} - \mathcal{V}_{|i,j-1}\,\boldsymbol{Q}_{|i,j-1}\,u^2_{|i,j-1}\right) \end{aligned} \tag{6.41}$$

$$\begin{aligned} &= \boldsymbol{Q}^n_{|ij} - \frac{\Delta t}{\mathcal{V}_{|ij}\,\Delta x^1}\left(\boldsymbol{f}^{(1)}_{|ij}(\boldsymbol{Q}_{|ij}) - \boldsymbol{f}^{(1)}_{|i-1,j}(\boldsymbol{Q}_{|i-1,j})\right) \\ &\quad - \frac{\Delta t}{\mathcal{V}_{|ij}\,\Delta x^2}\left(\boldsymbol{f}^{(2)}_{|ij}(\boldsymbol{Q}_{|ij}) - \boldsymbol{f}^{(2)}_{|i,j-1}(\boldsymbol{Q}_{|i,j-1})\right), \end{aligned} \tag{6.42}$$

where we have used the definition of the cell-centered flux (6.33) in the last equality. Recall that in equation (3.12) it was precisely those flux differences found in the previous update (6.42) that are decomposed into f-waves, $\{\boldsymbol{\mathcal{Z}}^p\}$. In the example used here, a single f-wave propagates in each direction, one from the left edge of the cell $\boldsymbol{\mathcal{Z}}^1_{|i-1/2,j}$, and the other from the lower edge of the cell $\boldsymbol{\mathcal{Z}}^1_{|i,j-1/2}$.

For a system of n_{eq} scalar equations, the flux difference is decomposed into a set of n_{eq} f-waves. In the x^1 direction, for example, this decomposition takes the form

$$\boldsymbol{f}^{(k)}_{|ij}(\boldsymbol{Q}_{|ij}) - \boldsymbol{f}^{(k)}_{|i-1,j}(\boldsymbol{Q}_{|i-1,j}) = \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{Z}}^p_{|i-1/2,j}, \tag{6.43}$$

and a similar decomposition exists in the x^2 direction. These decompositions allow us to replace the right hand side of the update (6.42) by fluctuations that are based on these f-waves. In this way, the numerical method (6.16) that is based on flux differences can be written as the wave propagation algorithm

$$\boldsymbol{Q}^{n+1}_{\mathcal{C}} = \boldsymbol{Q}^n_{\mathcal{C}} - \frac{\Delta t}{\mathcal{V}_{\mathcal{C}}\, \Delta x^i} \sum_{i=1}^{n_d} \left(\boldsymbol{\mathcal{A}}^+_i \boldsymbol{\Delta Q}_{\mathcal{C}^-} + \boldsymbol{\mathcal{A}}^-_i \boldsymbol{\Delta Q}_{\mathcal{C}^+} \right), \tag{6.44}$$

based on f-waves. In the last equation, as previously explained, the notation $\mathcal{C}^\pm$ refers to the left or right edge of the cell along the direction x^i. The fluctuations are defined in terms of f-waves as

$$\begin{aligned} \boldsymbol{\mathcal{A}}^- \boldsymbol{\Delta Q}_{\mathcal{C}^-} &= \sum_{p \in \mathscr{P}^-} \boldsymbol{\mathcal{Z}}^p_{\mathcal{C}^-} \\ \boldsymbol{\mathcal{A}}^+ \boldsymbol{\Delta Q}_{\mathcal{C}^-} &= \sum_{p \in \mathscr{P}^+} \boldsymbol{\mathcal{Z}}^p_{\mathcal{C}^-}, \end{aligned} \tag{6.45}$$

as they were in Chapter 3. High-resolution corrections and transverse propagation is handled in the same manner that it was in Chapter 3.

6.4.1 Wave speeds

The speed at which the wave travels in the x^k direction should approximate the eigenvalue $s^1 \approx \sqrt{h}\, u^k$. For the advection example that we are considering here, the speed is set based on the eigenvalue in the cell that the wave will propagate into so that

$$s^1_{|i-1/2,j} = \mathcal{V}_{|ij}\, u^1_{|ij}. \tag{6.46}$$

Recall from equation (3.20) that the original waves that carry jumps in q can be computed from the f-waves and speeds (as long as $s \neq 0$) using the fact that $\boldsymbol{\mathcal{W}}^1_{|i-1/2,j} = \boldsymbol{\mathcal{Z}}^1_{|i-1/2,j} / s^1_{|i-1/2,j}$. In this case $\boldsymbol{\mathcal{W}}^1_{|i-1/2,j}$ has a wave strength

$$\begin{aligned} \boldsymbol{\mathcal{W}}^1_{|i-1/2,j} = \frac{\boldsymbol{\mathcal{Z}}^1_{|i-1/2,j}}{s^1_{|i-1/2,j}} &= \frac{\mathcal{V}_{|ij}\, u^1_{|ij}\, \boldsymbol{Q}_{|ij} - \mathcal{V}_{|i-1,j}\, u^1_{|i-1,j}\, \boldsymbol{Q}_{|i-1,j}}{\mathcal{V}_{|ij}\, u^1_{|ij}} \\ &= \boldsymbol{Q}_{|ij} - \frac{\mathcal{V}_{|i-1,j}\, u^1_{|i-1,j}}{\mathcal{V}_{|ij}\, u^1_{|ij}} \boldsymbol{Q}_{|i-1,j}. \end{aligned} \tag{6.47}$$

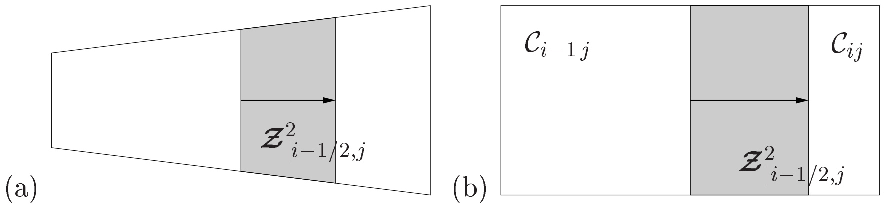


Figure 6.3: (a) Two f-waves traveling into cells $\mathcal{C}_{i-1\,j}$ and $\mathcal{C}_{ij}$ in the coordinate grid. (b) The same two f-waves traveling on the physical grid where it is clear the right-traveling wave is entering a cell with larger *volume*.

Let us suppose for a moment that the velocity is constant so that $u^1_{|i-1,j} = u^1_{|ij} > 0$, and material is flowing into the right cell so that $Q_{|i-1,j} > Q_{|ij}$. The wave propagates through the coordinate grid as depicted in Figure 6.3(b), and maps to the physical wave shown in Figure 6.3(a). A method that simply propagates a concentration wave with strength $Q_{|ij} - Q_{|i-1,j}$ into the right cell that has larger volume will update that cell with too much material. In reality, as the wave propagates into the right cell it has to fill more volume, hence the concentration decreases.

Equation (6.47) shows that for the method presented here based on f-waves, the wave propagating into the right cell will have strength

$$\mathcal{W}^1_{|i-1/2,j} = Q_{|ij} - \frac{\mathcal{V}_{|i-1,j}}{\mathcal{V}_{|ij}} Q_{|i-1,j}, \tag{6.48}$$

which is smaller than the full jump since $\mathcal{V}_{|i-1,j}/\mathcal{V}_{|ij} < 1$. Solving Riemann problems using f-waves, therefore, produces waves with geometry encoded into them and the geometry determines the strength of the waves that propagate.

6.4.2 Advection in a polar basis

In this section, we apply the numerical method (6.44) to the advection of a concentration subject to a radially symmetric velocity field. The manifold is taken to be a flat two dimensional plane, $\mathscr{M} = \mathbb{R}^2$, covered by polar coordinates $\{r, \theta\}$. These coordinates are related to the Cartesian coordinates using

$$\begin{aligned} x(r,\theta) &= r\,\cos\theta \\ y(r,\theta) &= r\,\sin\theta, \end{aligned} \tag{6.49}$$

in the usual way. The covariant components of the metric tensor in polar coordinates can be calculated using the transformation rule (5.50) with $x^i = \{x, y\}$ and $x^{\bar{k}} = \{r, \theta\}$. The elements (5.25) of the transformation form the matrix

$$\left[\Pi^j_{\bar{k}}\right] = \begin{pmatrix} \cos\theta & -r\,\sin\theta \\ \sin\theta & r\,\cos\theta \end{pmatrix}. \tag{6.50}$$

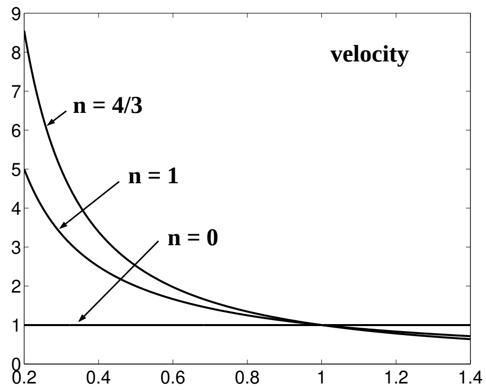


Figure 6.4: Radially symmetric velocity field used in advection example. The velocity becomes large and positive for small values of r with increasing n.

Using this matrix in the transformation defined by (5.50) together with the fact that the metric in Cartesian coordinates is $h_{ij} = \mathrm{diag}(1, 1)$, the metric in polar coordinates can be written

$$[h_{ij}] = \mathrm{diag}(1, r^2). \tag{6.51}$$

In the previous equation, the over-bar notation has been dropped and it is assumed that $x^i = \{r, \theta\}$. The determinant of the polar metric is $h = r^2$, so the volume element on the polar grid is given by

$$V_{|ij} = r_{|ij}\, \Delta r\, \Delta\theta, \tag{6.52}$$

which is independent of θ.

In this example an initial condition that is independent of θ is used together with a radially symmetric velocity field. Therefore, the solution should remain independent of θ. The radially symmetric velocity field has the form

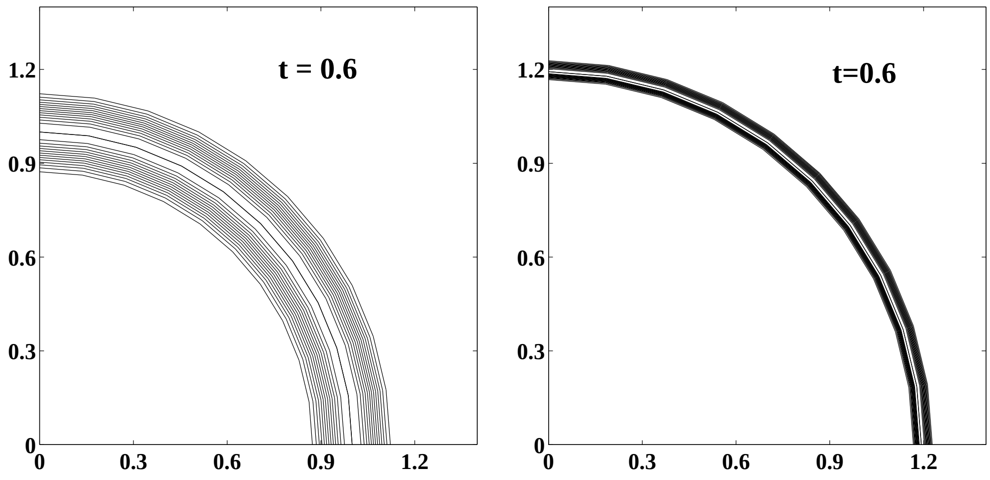
$$\vec{v} = \alpha\, r^{-n}\, \vec{\partial}_{(r)}, \tag{6.53}$$

where both α and n are constants. If $n = 0$ then α is the amplitude of a constant outflow. As n increases, the outflow becomes stronger for smaller r as shown in Figure 6.4. The particular values of n in the figure, $\{0, 1, 4/3\}$, are the values for which we compute solutions using this velocity field.

The initial condition used for $q(t, r, \theta)$ is independent of θ and has the form

$$q(t = 0, r) \equiv f(r) = \begin{cases} \frac{1}{2}\, a \left(\cos \frac{\pi(r - r_0)}{\omega} + 1\right) & \text{if } |r - r_0| < \omega \\ 0 & \text{if } |r - r_0| > \omega, \end{cases} \tag{6.54}$$

where a is the peak concentration, r_0 is the radius at which this peak occurs, and ω is a measure of the width of the concentration. Equation (6.31), together with this initial



(a) n=0

(b) n=4/3

Figure 6.5: (a) Contours of the concentration at $t = 0.6$ for $n = 0$. (b) Contours of the concentration at $t = 0.6$ for $n = 4/3$.

condition, can be exactly solved for a general n using the method of characteristics. The solution can be written in the explicit form

$$q(t,r,\theta) = \begin{cases} f\left(\left[r^2 - 2\,\alpha\,t\right]^{1/2}\right) & \text{if } n = 1 \\ r^{n-1}\,\left[r^{n+1} - \alpha\,(n+1)\,t\right]^{\frac{1-n}{1+n}}\, f\left(\left[r^{n+1} - \alpha\,(n+1)\,t\right]^{\frac{1}{1+n}}\right) & \text{if } n \neq 1 \end{cases}. \tag{6.55}$$

The reason this velocity field was chosen is that the geometric spreading of concentration can be eliminated, and in fact, overtaken by the fact that the velocity changes with r. For example, let $n = 0$ in the solution (6.55) to find $q = (1-\alpha\,t/r)f(r-\alpha\,t)$. As time progresses, the wave propagates to larger values of r and the solution decays. This is exactly what we should expect since $n = 0$ corresponds to a constant outflow. The same amount of material at larger r is filling more space in $\mathscr{M}$, hence the concentration decreases. However, if $n = 1$, then the amount by which the concentration is geometrically decreased is exactly balanced by the fact that the flow slows down as r increases. The fact that the flow slows means that material will pile up. For $n = 1$ these two effects exactly balance each other so that the amplitude remains constant. Finally, when $n > 1$ the rate at which material piles up as the wave travels to larger r is of greater influence than the geometric spreading of the wave. In this case, the concentration increases as the wave propagates outward.

Though the solutions for each n are functions only of r, the calculations have been done in two space dimensions so that we can test the radial symmetry of the method. Figures 6.16(a) and 6.16(b) show the two-dimensional contours of the concentration at $t = 0.6$ for $n = 0$ and $n = 4/3$. The squeezing of the material can be seen by the width of the contours

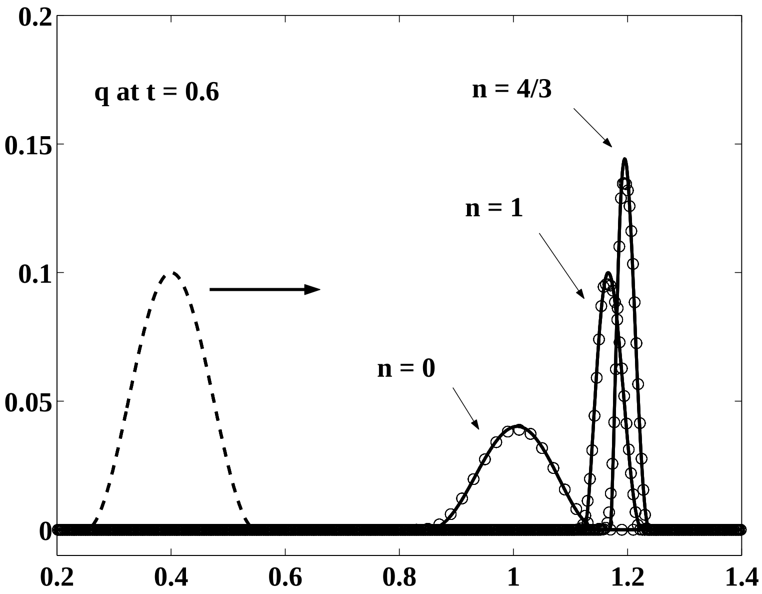


Figure 6.6: The exact solution at $t = 0.6$ for each $n \in \{0, 1, 4/3\}$ is represented by the solid lines. The initial condition is shown as a dashed line. A scatter plot of the computed solution is also shown for each n as circles.

in Figure 6.16(b). The radial profile of the initial condition (6.77) that has been used can be seen as the dashed line in Figure 6.6. The exact solutions for each $n \in \{0, 1, 4/3\}$ are represented by solid lines in this same figure. The circles represent a scatter plot of each data point in the two-dimensional calculations shown in Figures 6.16(a) and 6.16(b). The fact that there is no scatter in those points confirms that the data is radially symmetric. The computed values agree with the exact solution. Figure 6.7 is a plot of the one norm of the error for the solution with $n = 0$. It is clear that the method is second order on this problem.

6.5 Vector conservation laws

So far, we have discussed the use of f-waves that handle the spatially-varying flux that appears for conservation laws on manifolds. These f-waves induce fluctuations that are used in the update (6.44). In this section, the algorithm is generalized to handle vector conservation laws. In the next section, the systems consisting of both scalar and vector laws are discussed.

In Section 5.8.2 we saw that a vector conservation law on a manifold can be written in the form

$$v^j_{,t} + \frac{1}{\sqrt{h}} \left(\sqrt{h}\, T^{kj} \right)_{,k} = -\Gamma^j_{mk} T^{mk}, \tag{6.56}$$

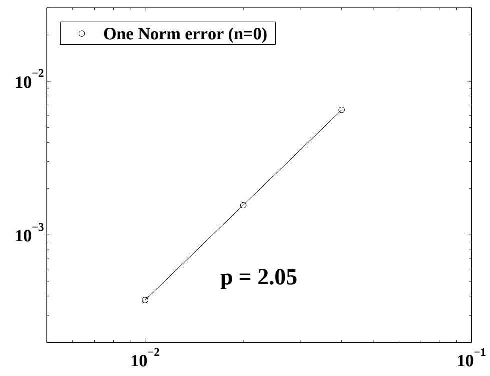


Figure 6.7: A plot of the one norm for the error in the computed solution with $n = 0$. The order has been computed to be 2.05.

where v^j are the tangent components of the conserved vector $\vec{v}$, and T^{ij} are the tangent components of $\overleftrightarrow{T}$, a second rank symmetric (i.e. $\overleftrightarrow{T}(\tilde{e}^{(i)}, \tilde{e}^{(j)}) = \overleftrightarrow{T}(\tilde{e}^{(j)}, \tilde{e}^{(i)})$) flux tensor. Recall that the Christoffel symbols in equation (6.56), denoted by Γ^i_{jk}, represent components of the derivatives of the canonical basis $\{\vec{\partial}_{(k)}\}$.

The new feature of a vector conservation law is that the quantity being conserved depends on the basis. This fact is not a problem when the basis vectors are independent of the coordinates as they are in a Cartesian basis. However, when the basis is spatially varying as it is in a general manifold, a meaningful comparison of the same component v^j at two distinct locations in $\mathscr{M}$ must be made with reference to only a single basis. This was not true for scalar equations because a scalar does not need to be measured against any basis vector. The fact that there is a different set of basis vectors at each point of $\mathscr{M}$ did not affect the scalar fields.

In Section 5.7, the parallel transport of vector components along a general curve was introduced. If the curve along which we parallel transport the components of a vector $\vec{v}$ is a coordinate curve C_k, then the fact that $\dot{x}^i = \delta^i_k$ along this curve allows us to write equation (5.73) as

$$\begin{aligned} v^j_{;k} &= v^j_{,k} + v^m\, \Gamma^j_{mk} \\ &= 0. \end{aligned} \tag{6.57}$$

If we start with a vector component, v^j, at some point $\mathcal{P} \in \mathscr{M}$ with components $(x_0^1, \cdots, x_0^{n_d})$, and parallel transport that component to a nearby point separated from $\mathcal{P}$ by Δx^k along C_k, then the new components according to the ODE (6.57) will be

$$v^j = v^j(\mathcal{P}) - \int_{x_0^k}^{x_0^k + \Delta x^k} v^m\, \Gamma^j_{mk}\, dx^k. \tag{6.58}$$

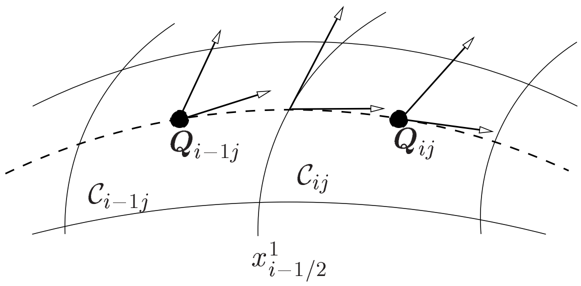


Figure 6.8: The basis vectors at the centers of cells $\mathcal{C}_{i-1j}$ and $\mathcal{C}_{ij}$ are different than the basis at the cell edge. Vector and tensor components must be parallel propagated to the basis at the cell edge.

The components of $\vec{v}$ are only changing to compensate for the fact that the basis has changed at the new point. In effect, the component is simply being projected into the basis at the new point without changing itself.

The tangent components of a tensor suffer the same problem as those discussed for vector components. The source term, $-T^{mn}\,\Gamma^j_{mn}$, that appears in equation (6.56) is due to the fact that the spatially separated components of the flux tensor that are needed for the derivatives in the divergence must be projected into the same coordinate basis before a meaningful difference can be taken. One of the main novel features of the algorithm presented here is that the source term in equation (6.56) is handled using parallel transport of vector components.

6.5.1 Discrete parallel transport

Figure 6.8 shows a typical coordinate grid for $n_d = 2$ with different bases at each of the cell centers and the cell interface. Suppose that we are solving the momentum equation for a fluid (see Appendix B for the equations) so that components of the state vector $\boldsymbol{Q}$ and flux $\boldsymbol{f}^{(k)}$ involve the momentum components $\rho\, v^k$. Typically, Riemann problems at each cell interface are based on decomposing the difference between such components located at the centers of the contiguous cells shown in Figure 6.8. Because this difference involves two different sets of bases, the source term in equation (6.56) must also be incorporated.

Instead, the method proposed here parallel propagates vector components of $\boldsymbol{Q}$ to the cell interface prior to the solution of Riemann problems. In this way, as we will show, the source term is locally incorporated into each Riemann problem. By doing so, we hope to avoid problems discussed in Chapter 4.

In Section 5.7, the discrete parallel transport of vector components along general curves was derived and written in the form

$$\left[s^i\right]_{\mathcal{Q}} = \left[P^i_m\right]\,\left[s^m\right]_{\mathcal{P}}\,, \tag{6.59}$$

where the two points $\mathcal{P}$ and $\mathcal{Q}$ are on the curve. Since only one-dimensional Riemann problems are solved in the algorithm developed here, the curves along which parallel transport must take place are the coordinate curves C_k. This being the case, the elements of the discrete parallel propagation matrix, $\left[P^i_m\right]$ along the K^{th} coordinate curve will have the form

$$P^i_m = \delta^i_m - \Delta x^K\, \Gamma^i_{mK} + \frac{1}{2}\left(\Delta x^K\right)^2 \left(\Gamma^i_{nK}\,\Gamma^n_{Km} - \mathcal{G}^i_{mKK}\right) + \mathcal{O}\left((\Delta x^K)^3\right), \tag{6.60}$$

where, once again, the use of a capital Latin index signifies that it is a single number and never to be summed, and $\mathcal{G}^i_{mKK}$ represents the partial derivative of Γ^i_{mK} along C_K. The direction along C_K that the components are propagated depends on the sign of Δx^K. For $\Delta x^K < 0$, components are parallel propagated to the left, but for $\Delta x^K > 0$, they are propagated to the right.

Generalizing the discrete flux defined in (6.33), the cell-centered discretization of the flux is defined to be

$$\boldsymbol{f}^{(k)}_{|ij}(\boldsymbol{Q}_{|ij}) = \mathcal{V}_{|ij}\,\boldsymbol{f}^{(k)}(\boldsymbol{Q}_{|ij}). \tag{6.61}$$

As mentioned above, simply decomposing the difference of this flux at each cell interface into a set of f-waves would yield a difference of vector components at spatially separated points. Doing so would also require that the source on the right side of (6.56) be incorporated in some way. Instead, the source can be absorbed into the Riemann problem by taking a flux difference between the modified fluxes defined by

$$\boldsymbol{f}^{(1)-}_{|ij} = [P]^-_{ij}\;\boldsymbol{f}^{(1)}_{|ij}; \quad \text{modified right flux} \tag{6.62}$$

$$\boldsymbol{f}^{(1)+}_{|i-1,j} = [P]^+_{i-1,j}\;\boldsymbol{f}^{(1)}_{|i-1,j}; \quad \text{modified left flux,} \tag{6.63}$$

where $[P]^-_{ij}$ parallel propagates vector components of $\boldsymbol{f}^{(1)}_{|ij}$ to the left edge of $\mathcal{C}_{ij}$, and $[P]^+_{i-1,j}$ propagates $\boldsymbol{f}^{(1)}_{|i-1,j}$ to the right edge of $\mathcal{C}_{i-1,j}$. Similar modifications take place in the other spatial directions. The result is a flux difference at the cell edge for which both left and right fluxes are expressed in the same basis. The Riemann problem will involve the modified data.

A calculation computing the influence of parallel propagation on the flux difference shows that differencing the modified fluxes incorporates the source term in present equation (6.56). If terms higher than first order are neglected, one finds that

$$\begin{aligned}
\boldsymbol{f}^{(1)-}_{|ij} - \boldsymbol{f}^{(1)+}_{|i-1,j} &= [P]^-_{ij}\;\boldsymbol{f}^{(1)}_{|ij} - [P]^+_{i-1,j}\;\boldsymbol{f}^{(1)}_{|i-1,j} \\
&= \left(\mathcal{I} + \frac{\Delta x^1}{2}\Gamma^i_{m1|ij}\right)\boldsymbol{f}^{(1)m}_{|ij} - \left(\mathcal{I} - \frac{\Delta x^1}{2}\Gamma^i_{m1|i-1,j}\right)\boldsymbol{f}^{(1)m}_{|i-1,j} \\
&= \boldsymbol{f}^{(1)m}_{|ij} - \boldsymbol{f}^{(1)m}_{|i-1,j} - \Delta x^1\,\frac{1}{2}\left(\Gamma^i_{n1|i-1,j}\boldsymbol{f}^{(1)n}_{|i-1,j} + \Gamma^i_{n1|ij}\boldsymbol{f}^{(1)n}_{|ij}\right),
\end{aligned} \tag{6.64}$$

along the x^1 direction, and that

$$\begin{aligned}
\boldsymbol{f}^{(2)-}_{|ij} - \boldsymbol{f}^{(2)+}_{|i,j-1} &= [P]^-_{ij}\;\boldsymbol{f}^{(2)}_{|ij} - [P]^+_{i,j-1}\;\boldsymbol{f}^{(2)}_{|i,j-1} \\
&= \left(\mathcal{I} + \frac{\Delta x^2}{2}\Gamma^i_{m2|ij}\right)\boldsymbol{f}^{(2)m}_{|ij} - \left(\mathcal{I} - \frac{\Delta x^2}{2}\Gamma^i_{m2|i,j-1}\right)\boldsymbol{f}^{(2)m}_{|i,j-1} \\
&= \boldsymbol{f}^{(2)m}_{|ij} - \boldsymbol{f}^{(2)m}_{|i,j-1} - \Delta x^2\,\frac{1}{2}\left(\Gamma^i_{n2|i,j-1}\boldsymbol{f}^{(2)n}_{|i,j-1} + \Gamma^i_{n2|ij}\boldsymbol{f}^{(2)n}_{|ij}\right),
\end{aligned} \tag{6.65}$$

along the x^2 coordinate. The average that appears in both modified flux differences suggests that we define

$$\begin{aligned}\boldsymbol{\psi}^{(1)}_{|i-1/2,j} &\equiv \frac{1}{2}\left(\Gamma^i_{n1|i-1,j}\boldsymbol{f}^{(1)n}_{|i-1,j} + \Gamma^i_{n1|ij}\boldsymbol{f}^{(1)n}_{|ij}\right)\\ \boldsymbol{\psi}^{(2)}_{|i,j-1/2} &\equiv \frac{1}{2}\left(\Gamma^i_{n2|i,j-1}\boldsymbol{f}^{(2)n}_{|i,j-1} + \Gamma^i_{n2|ij}\boldsymbol{f}^{(2)n}_{|ij}\right)\end{aligned} \tag{6.66}$$

at the cell edges. If these definitions are used in the flux differences (6.64) and (6.65), then they look just like the modified flux difference (4.42) for the source term method discussed in Section 4.2.

Notice that through both x^1 and x^2 sweeps through the grid, a source term of the form

$$\boldsymbol{\psi} = \Gamma^i_{11}\boldsymbol{f}^{(1)1} + \Gamma^i_{21}\boldsymbol{f}^{(1)2} + \Gamma^i_{12}\boldsymbol{f}^{(2)1} + \Gamma^i_{22}\boldsymbol{f}^{(2)2} \tag{6.67}$$

is being incorporated into the multidimensional numerical update. The source term on the right hand side of equation (6.56) can be written

$$\boldsymbol{\psi} = \underbrace{\Gamma^i_{11}T^{11} + \Gamma^i_{21}T^{21}}_{x^1\text{ sweeps}} + \underbrace{\Gamma^i_{12}T^{12} + \Gamma^i_{22}T^{22}}_{x^2\text{ sweeps}}, \tag{6.68}$$

which is equivalent to (6.67) since $\boldsymbol{f}^{(i)j} = T(\tilde{dx}^{(i)}, \tilde{dx}^{(j)}) = T^{ij}$. This shows that the source term has been absorbed into the flux difference, and therefore into the Riemann problems that are solved at each grid interface.

6.6 Accuracy

The algorithm developed here is intended for applications with discontinuous solutions. It is not expected to achieve a formal order of accuracy greater that one on such solutions. However, second order accuracy in smooth regions of the solution is often important to sharply resolve features away from the discontinuities.

It was demonstrated in Section 3.3 that an algorithm based on f-waves will generate second order numerical solutions if both the spatially-varying flux function, and the solution remain smooth. In the previous section, we found that solving Riemann problems based on fluxes that have been parallel transported is equivalent to incorporating the geometric source term using the modified flux method discussed in Section 4.2. It was also shown that in order to maintain second order accuracy in the presence of a spatially-varying flux, the numerical update for the modified flux method must include the correction (4.45).

The correction (4.45) was written in terms of $\partial\boldsymbol{\psi}/\partial\boldsymbol{q}$ for a general source term $\boldsymbol{\psi}$. For conservation laws on manifolds, the m^{th} component of the source is

$$\psi^{[m]} = \begin{cases} 0 & \text{for scalar components of } \boldsymbol{q} \\ -\Gamma^m_{jk}T^{jk} & \text{for the } m^{\text{th}} \text{ vector component of } \boldsymbol{q}. \end{cases} \tag{6.69}$$

In order to implement the correction (4.45), a vector $\boldsymbol{\mathcal{S}}$ is introduced at each cell edge. Along a slice in the x^1 direction, the m^{th} vector component of $\boldsymbol{\mathcal{S}}^{(1)}_{|i-1/2}$ is defined to be

$$
\begin{aligned}
\mathcal{S}^{(1)[m]}_{|i-1/2} &= -\Gamma^m_{jk}\frac{\partial \boldsymbol{f}^{(j)k}}{\partial \boldsymbol{q}}\sum_{p=1}^{n_{eq}}\boldsymbol{\mathcal{Z}}^p_{|i-1/2} \\
&= -\Gamma^m_{1k}\left[A^1_{|i-1/2}\sum_{p=1}^{n_{eq}}\boldsymbol{\mathcal{Z}}^p_{|i-1/2}\right]^{[k]} - \Gamma^m_{2k}\left[A^2_{|i-1/2}\sum_{p=1}^{n_{eq}}\boldsymbol{\mathcal{Z}}^p_{|i-1/2}\right]^{[k]} \\
&= -\Gamma^m_{1k}\left[\sum_{p=1}^{n_{eq}} s^p_{|i-1/2}\boldsymbol{\mathcal{Z}}^p_{|i-1/2}\right]^{[k]} - \Gamma^m_{2k}\left[A^2_{|i-1/2}\left(\boldsymbol{\mathcal{A}}_1^-\boldsymbol{\Delta Q}_{i-1/2} + \boldsymbol{\mathcal{A}}_1^+\boldsymbol{\Delta Q}_{i-1/2}\right)\right]^{[k]},
\end{aligned}
\tag{6.70}
$$

where the fact that $\boldsymbol{\mathcal{Z}}^p_{|i-1/2}$ is the p^{th} eigenvector of the approximate Jacobian $A^1_{|i-1/2}$ has been used. The terms in square brackets in the previous equation are vectors with n_{eq} components, but the sum on k only ranges over the spatial dimensions 1 to n_d. To perform the sum, scalar components of these vectors are neglected and only the vector components are included.

The first term on the right side of the last equality in equation (6.70) can be computed using the f-waves and speeds returned by the normal Riemann solvers. The second term involves decomposing the right and left-going fluctuations into eigenvectors of the transverse Jacobian matrix. That is exactly what is done in the transverse Riemann problems described in Section 2.3.2. Therefore, this term can be computed using the transverse fluctuations $\boldsymbol{\mathcal{A}}_2^-\boldsymbol{\mathcal{A}}_1^*\boldsymbol{\Delta q}_{i-1/2}$ and $\boldsymbol{\mathcal{A}}_2^+\boldsymbol{\mathcal{A}}_1^*\boldsymbol{\Delta q}_{i-1/2}$ returned from the transverse Riemann solvers.

These edge vectors can be used to approximate the correction (4.45) as

$$
-\frac{\Delta t^2}{2\Delta x^1}\,\frac{1}{2}\left(\boldsymbol{\mathcal{S}}^{(1)}_{|i-1/2} + \boldsymbol{\mathcal{S}}^{(1)}_{|i+1/2}\right). \tag{6.71}
$$

The average is used in the previous correction to approximate the value of $\boldsymbol{\mathcal{S}}^{(1)}$ at the cell center. There is also a correction of this form along the x^2 direction denoted by $\boldsymbol{\mathcal{S}}^{(2)}$.

6.7 Features of the new algorithm

We have now described the details of a new wave propagation algorithm to solve hyperbolic conservation laws of both scalar and vector equations on manifolds. The main features of the algorithm are summarized as follows:

- The spatially-varying flux functions that result from expressing the conservation law on a coordinate basis that covers $\mathscr{M}$ are handled using f-waves.

- The geometrically induced source terms that appear for vector components of $\boldsymbol{q}$ are handled by parallel transporting the components from the cell center to the cell edge prior to solving Riemann problems.

- Corrections needed for a second order method have been shown to require only the data returned by normal and transverse Riemann solvers.

- Riemann problems can be solved in orthonormal frames at the edge of each cell. This allows the use of Riemann solvers based on Cartesian coordinates, as long as the solver returns f-waves.

The unsplit, high-resolution update for conservation laws on manifolds can be written in the form

$$\begin{aligned}\boldsymbol{Q}_{\mathcal{C}}^{n+1} = \boldsymbol{Q}_{\mathcal{C}}^{n} - \frac{\Delta t}{\mathcal{V}_{\mathcal{C}}\,\Delta x^i}\sum_{i=1}^{n_d}\Big[&\boldsymbol{\mathcal{A}}_i^{+}\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{\mathcal{C}-} + \boldsymbol{\mathcal{A}}_i^{-}\boldsymbol{\Delta}\tilde{\boldsymbol{Q}}_{\mathcal{C}+} + \tilde{\boldsymbol{F}}_{\mathcal{C}+}^{(i)} - \tilde{\boldsymbol{F}}_{\mathcal{C}-}^{(i)} \\ &+ \frac{\Delta t}{4}\left(\boldsymbol{\mathcal{S}}_{\mathcal{C}-}^{(i)} + \boldsymbol{\mathcal{S}}_{\mathcal{C}+}^{(i)}\right)\Big],\end{aligned} \tag{6.72}$$

where the tildes over the fluctuations signify that they are computed based on the modified data $\boldsymbol{f}_{|i-1/2,j}^{(k)+}$ and $\boldsymbol{f}_{|ij}^{(k)-}$. The correction fluxes, $\tilde{\boldsymbol{F}}^{(i)}$, are based on limited f-waves as defined for spatially-varying fluxes in (3.23).

6.8 The scalar field equation

The scalar field equation is useful in describing a number of physical systems for which it is appropriate to test the new algorithm. The equation has the form

$$-\varphi_{,tt} + \vec{\nabla}\cdot\left(\vec{\nabla}\,\varphi\right) = 0, \tag{6.73}$$

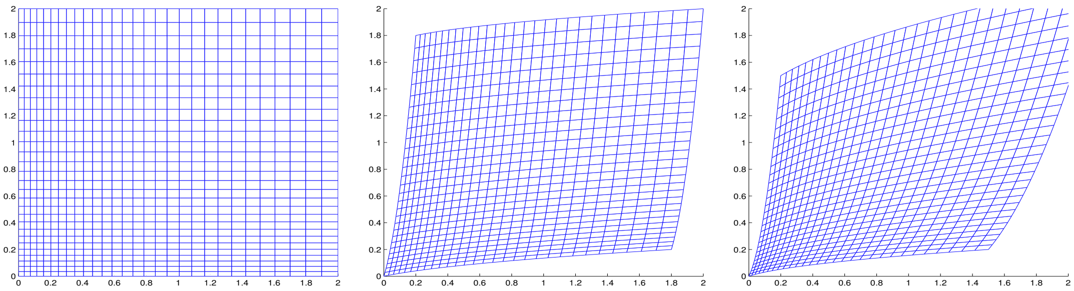
where φ is the scalar field. The velocity field for a relativistic perfect fluid with no vorticity and an equation of state $P = \rho$ is the gradient of such a scalar field. An application with a hard relativistic sphere immersed in such a perfect fluid is discussed in Chapter 8. In this chapter the application used is nonrelativistic acoustics. For this application, the second order equation is reduced to a first order system by defining

$$\begin{aligned} p &= \varphi_{,t} \\ \vec{v} &= \vec{\nabla}\,\varphi. \end{aligned} \tag{6.74}$$

In terms of these new variables, equation (6.73) takes the form

$$\begin{aligned} p_{,t} - \vec{\nabla}\cdot\vec{v} &= 0, \\ \vec{v}_{,t} - \vec{\nabla}\cdot\left(p\,\overset{\leftrightarrow}{h}\right) &= 0. \end{aligned} \tag{6.75}$$

If $n_d = 2$, the state vector has the form $\boldsymbol{q} = (p, v^1, v^2)^T$ with one scalar component and two vector components. Since the algorithm developed in this thesis treats all geometric terms within local Riemann problems, it solves equation (6.73) in the form (6.75).



(a) grid-1 (b) grid-2 (c) grid-3

Figure 6.9: Three curvilinear grids with a variety of geometric complexity.

It will be useful to compare solutions generated by the methods discussed here with well established algorithms. To this end, use the covarient invariance of the metric, together with the fact that only stationary metrics are considered to write equations (6.75) in the form

$$
\begin{aligned}
p_{,t} - v^m_{,m} &= v^m\,\Gamma^j_{mj} \\
v^k_{,t} - h^{km}\,p_{,m} &= 0.
\end{aligned}
\tag{6.76}
$$

In the previous set of equations the definition of the covariant derivative has been used. The standard wave propagation algorithm discussed in Chapter 2 can be applied to the equations in this form. Notice that a generalized Riemann solver must be written to handle the spatially-varying coefficients, h^{km}, that appear in the flux. A split approach can be used to handle the source term in the equation for p.

6.9 A Euclidean manifold: $\mathscr{M} = \mathbb{R}^2$ covered by curvilinear coordinates

The first example considered investigates the behavior of the algorithm on three different curvilinear grids that cover $\mathbb{R}^2$. The grids used in this example are shown in Figure 6.9. These grids are defined by the coordinate transformation

$$
\begin{aligned}
x \to x(\xi,\eta) &= \alpha\,\xi + \beta\,\eta + \theta\,\xi^2 + \delta\,\xi\,\eta \\
y \to y(\xi,\eta) &= \alpha\,\eta + \beta\,\xi + \theta\,\eta^2 + \delta\,\xi\,\eta,
\end{aligned}
$$

with the parameter values shown in Table 6.2. These grids have been chosen to activate more Christoffel symbols as the grid number increases. By selectively turning Christoffel symbols on, we can test the performance of the method for a wide range of geometric complexity. Figure 6.9 shows the grids defined by parameters in Table 6.2.

The third grid has all nonzero Christoffel symbols. Though the following computation was carried out on all grids, only the results for the third grid are presented. A radially

(a) (b)

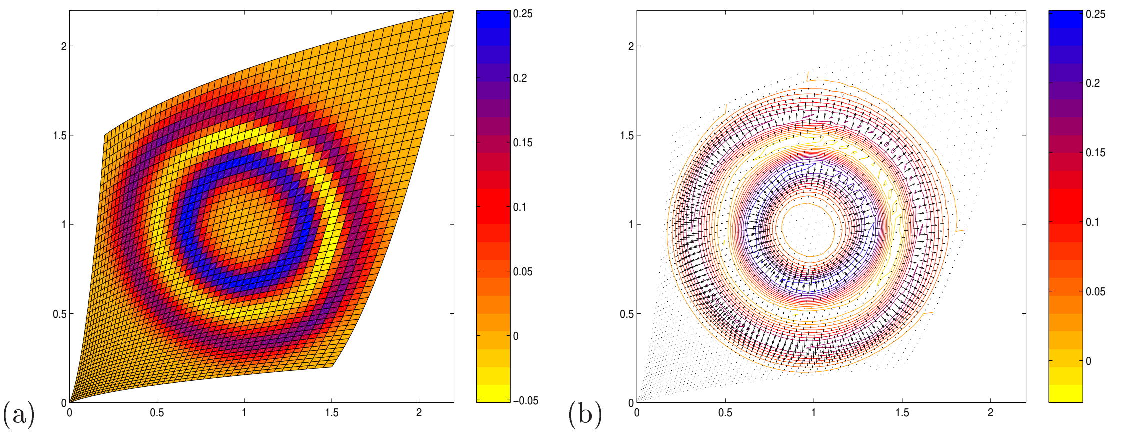


Figure 6.10: (a) A plot of the solution with the physical grid lines. (b) contours of the same solution. Also plotted are vectors indicating the direction of the velocity.

symmetric pressure pulse is initialized on the grid and centered around the point ($x = 1, y = 1$). The functional form of the pulse is taken to be

$$p(t=0,r) = \begin{cases} a \cos \frac{\pi\,|r-r_0|}{\delta} + 1.0) & \text{if } |r-r_0| < \delta \\ 0 & \text{if } |r-r_0| > \delta, \end{cases} \tag{6.77}$$

with no initial velocity. The pressure pulse should split into an in-going and an out-going wave that both remain radially symmetric as they evolve. Using the fact that the solution remains radially symmetric, a highly-resolved one-dimensional computation can be done in the radial direction. This solution can be used to calculate the errors generated by the new algorithm. Figure 6.10(a) shows the solution as it evolves through the physical grid. Notice that the grid lines are skewed, and the volume of the grid cells changes, but the solution remains radially symmetric. Figure 6.10(b) shows the contours of the solution, together with arrows that show the direction of the velocity.

Table 6.2: Parameters used in the curvilinear transformation. The particular parameters are chosen to increase the complexity of the grids. The last grid has all nonzero Christoffel symbols.

	α	β	θ	δ
grid-1	1.0	0.0	1.0	0.0
grid-2	0.5	0.3	1.0	0.0
grid-3	0.5	0.3	1.0	0.5

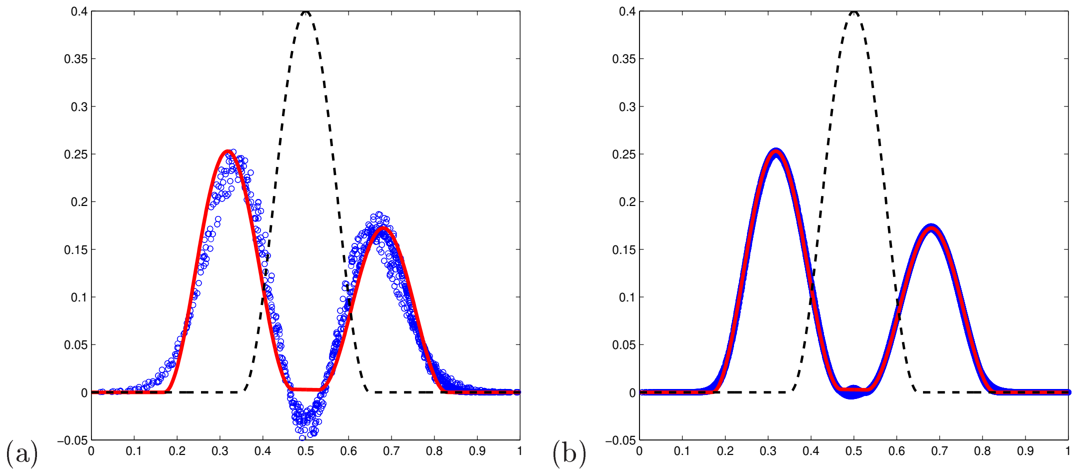


Figure 6.11: (a) A scatter plot of the computed solution on a 50×50 grid. Computed values are displayed as circles, and the solid line represents the highly-refined one-dimensional solution (b) The same as (a) for a 200×200 grid.

The radial symmetry of the solution can be seen in Figures 6.11(a) and 6.11(b). In these figures, the initial condition as a function of r is shown as a dashed line. The solid line represents the highly-refined one-dimensional solution, and the circles represent all the computed points in the grid shown in Figure 6.10(a). Figure 6.11(a) shows the computed solution on a 50×50 grid. Virtually all the scatter is gone on the 200×200 grid shown in Figure 6.11(b). The one-norm errors of the pressure, and both components of velocity for this example are listed in Table 6.3.

6.10 A Riemannian manifold: $\mathscr{M}$ defined by a surface $z = b(x^1, x^2)$

The previous example demonstrates that the algorithm presented in this dissertation can handle nonorthogonal curvilinear coordinates that cover a flat manifold defined by $\mathscr{M} = \mathbb{R}^2$. In this example we compute the solution to equation (6.73) on the Riemannian manifold defined by a surface $z = b(x^1, x^2)$. The examples presented in this section will use the Cartesian coordinates

$$\begin{aligned} x^1 &\to x^1(x, y) = x \\ x^2 &\to x^2(x, y) = y. \end{aligned} \tag{6.78}$$

The surface that defines $\mathscr{M}$ for all the examples in this section is a Gaussian dip with the functional form

$$b(x, y) = \alpha \exp - \frac{\left((x^1 - x_0)^2 + (x^2 - y_0)^2\right)^{1/2}}{\omega^2}. \tag{6.79}$$

Table 6.3: The one norm of the error in the computed pressure, p, first component of the velocity, v^1, and the second component of velocity, v^2.

	Pressure	1-Velocity	2-Velocity
grid	$\|p_{\text{exact}} - p_{\text{CLAWMAN}}\|_1$	$\|p_{\text{exact}} - p_{\text{CLAWMAN}}\|_1$	$\|p_{\text{exact}} - p_{\text{CLAWMAN}}\|_1$
50x50	$.4328 \times 10^{-1}$	$.1328 \times 10^{-1}$	$.1328 \times 10^{-1}$
100x100	$.1364 \times 10^{-1}$	$.4158 \times 10^{-2}$	$.4158 \times 10^{-2}$
200x200	$.3695 \times 10^{-2}$	$.1345 \times 10^{-2}$	$.1345 \times 10^{-2}$
order	1.9	1.6	1.6

The point (x_0, y_0) defines the location in the coordinate plane of the minimum value in the surface. Figures 6.12(a) and 6.12(b) show this surface embedded in a three-dimensional Euclidean space with two different values of the amplitude α.

The Euclidean space in which the surface is embedded has the Cartesian metric

$$[h_{IJ}] = \text{diag}(1, 1, 1), \tag{6.80}$$

where capital Latin indices range over the dimension of the flat Euclidean space, e.g., $I, J \in \{1, 2, 3\}$. Points on the surface are described in this space as $(x, y, b(x, y))$. The metric tensor on the surface can be computed by projecting $[h_{IJ}]$ onto the sub-manifold defined by the surface $b(x, y)$. To do this, define a projection matrix with components $M_I^k = \frac{\partial x^k}{\partial x^I}$ so that the matrix has the form

$$[M_k^I] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ b_{,1} & b_{,2} \end{pmatrix}. \tag{6.81}$$

This projection matrix is used to define the metric tensor restricted to the surface by computing

$$\begin{aligned} [h_{kj}] &= \left[M_k^I\right]^T [h_{IJ}] \left[M_j^J\right] \\ &= \begin{pmatrix} 1 & 0 & b_{,1} \\ 0 & 1 & b_{,2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ b_{,1} & b_{,2} \end{pmatrix} \\ &= \begin{pmatrix} 1 + (b_{,1})^2 & b_{,1}\, b_{,2} \\ b_{,1}\, b_{,2} & 1 + (b_{,2})^2 \end{pmatrix}. \end{aligned} \tag{6.82}$$

The fact that the off-diagonal terms of this metric are nonzero implies that the coordinate lines on the surface are skewed. Figure 6.12(b) shows the skewing of coordinate lines that occurs in the region near the dip.

The metric h_{kj} derived above defines the geometry of the two-dimensional manifold on which we solve equation (6.73). One way to get a feel for the geometry defined by h_{kj} is to compute geodesics on the surface. Geodesics are curves that represent the "straightest

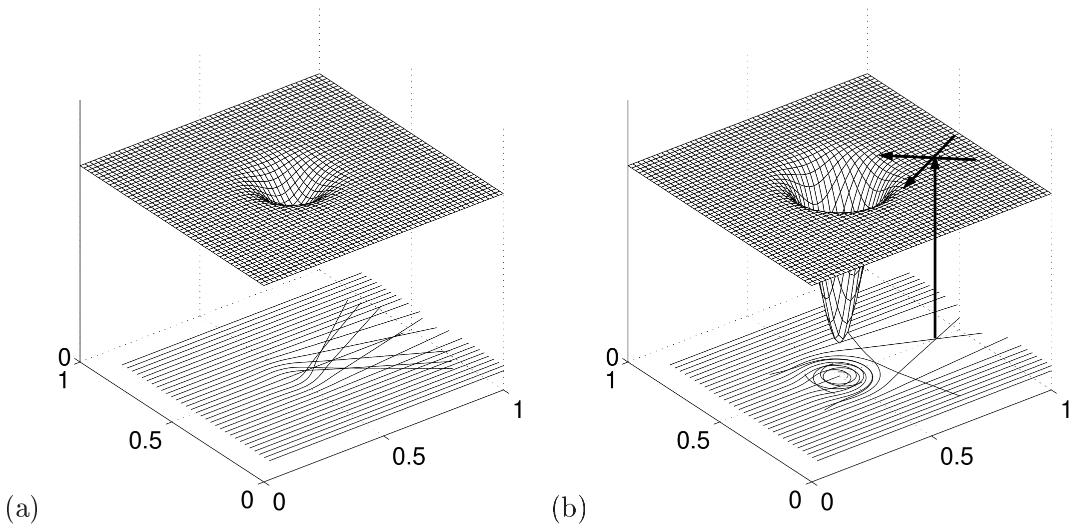


Figure 6.12: The surface that defines $\mathscr{M}$ for the examples presented in Section 6.10. (a) Geodesics initialized as right-going curves along the left boundary are projected down onto a coordinate plane. The depression in the surface causes the geodesics to converge. (b) As the dip gets larger in amplitude, the geodesics converge more and eventually wrap around the center of the surface as shown here.

possible lines" in $\mathscr{M}$. For example, Figure 6.12(a) shows geodesics that are initialized along the left boundary to be right-going. These geodesics are projected down to the coordinate plane for visualization purposes. The figure shows that the geodesics that avoid the dip and travel through the flat space remain straight, but the curves that encounter the dip begin to converge and their paths are altered. If the amplitude of the surface is increased as in Figure 6.12(b), the direction of geodesic curves in $\mathscr{M}$ can be changed entirely.

The geodesics shown in Figures 6.12(a) and 6.12(b) resemble the paths of photons traveling past large clusters of galaxies. Due to Einstein's general theory of relativity, a cluster of galaxies will warp the fabric of spacetime similar to the surface shown in the figures. Clusters with more mass will have larger dips. Light from distant objects will arrive at the cluster traveling in straight and parallel paths much like the geodesics shown in the figures. However, when the light encounters the curvature of the dip produced by the cluster, light rays converge as they do in the figures. Figure 6.12(b) shows the intersection of two light rays that have passed by the dip. An observer at this intersection would see the same object coming from two different directions. This phenomena is called gravitational lensing, and is observed by astronomers.

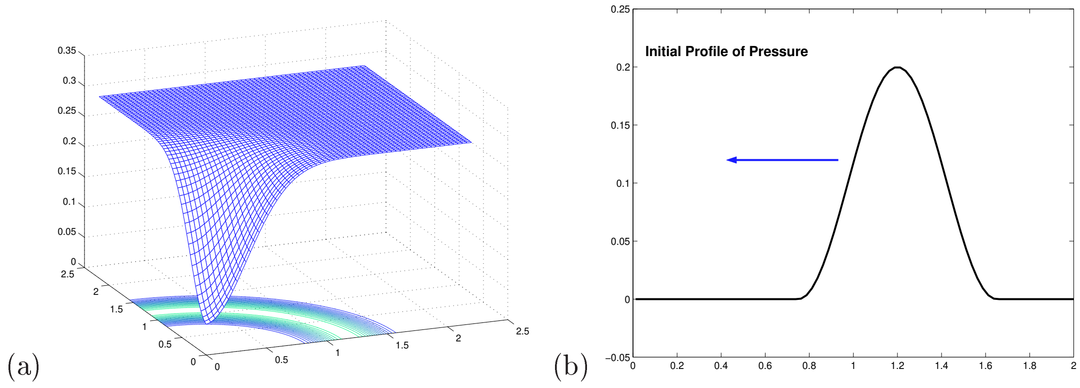


Figure 6.13: (a) The manifold on which the radially symmetric solution is computed. Contours of the initial pressure pulse are shown. (b) The initial profile of the pressure as a function of radius.

6.10.1 Radially symmetric initial conditions

Consider the radially symmetric initial condition (6.77) that was used in Section 6.9. The axis of symmetry is taken to be the location where the surface has its minimum value, (x_0, y_0). Figure 6.13(a) shows the upper quadrant of the surface together with the contours of the initial pressure pulse. Figure 6.13(b) shows the initial condition for the pressure as a function of r. The pulse is initially set up in the flat region outside the surface dip. Once again, the pulse will decay into an out-going and an in-going piece. The out-going pulse will simply radiate away since it only travels in the flat part of $\mathscr{M}$. However, the in-going pulse will interact with the dip in the surface.

Since the initial condition and the surface share the same axis of radial symmetry, the solution will remain radially symmetric. Therefore, a highly-resolved one-dimensional computation can be done by applying the standard wave propagation method to the one-dimensional version of equations 6.76. A split method has been used to handle the source terms. These results can be compared with the results of the new algorithm.

The two-dimensional calculation is done on the upper quadrant shown in Figure 6.13(a). Simple extrapolation boundary conditions are applied to the right and upper boundaries where the manifold is flat. Reflecting boundary conditions are used on the left and lower boundaries. Notice that along boundaries that intersect the dip, grid lines are skewed and the curvature of $\mathscr{M}$ is exponential. These are difficult boundaries to capture.

Figures (6.14(a))-(6.14(d)) show a time evolution of computed data for this radially symmetric pulse. The solid line represents the one-dimensional solution, and the circles represent the solution computed by the new method. All two dimensional data points are shown in the figures. The fact that there is no scatter in the data means that the algorithm maintains the expected symmetry. In Figure 6.14(a), the out-going pulse can be seen leaving the domain at the right edge of the plot. These figures show that the agreement between

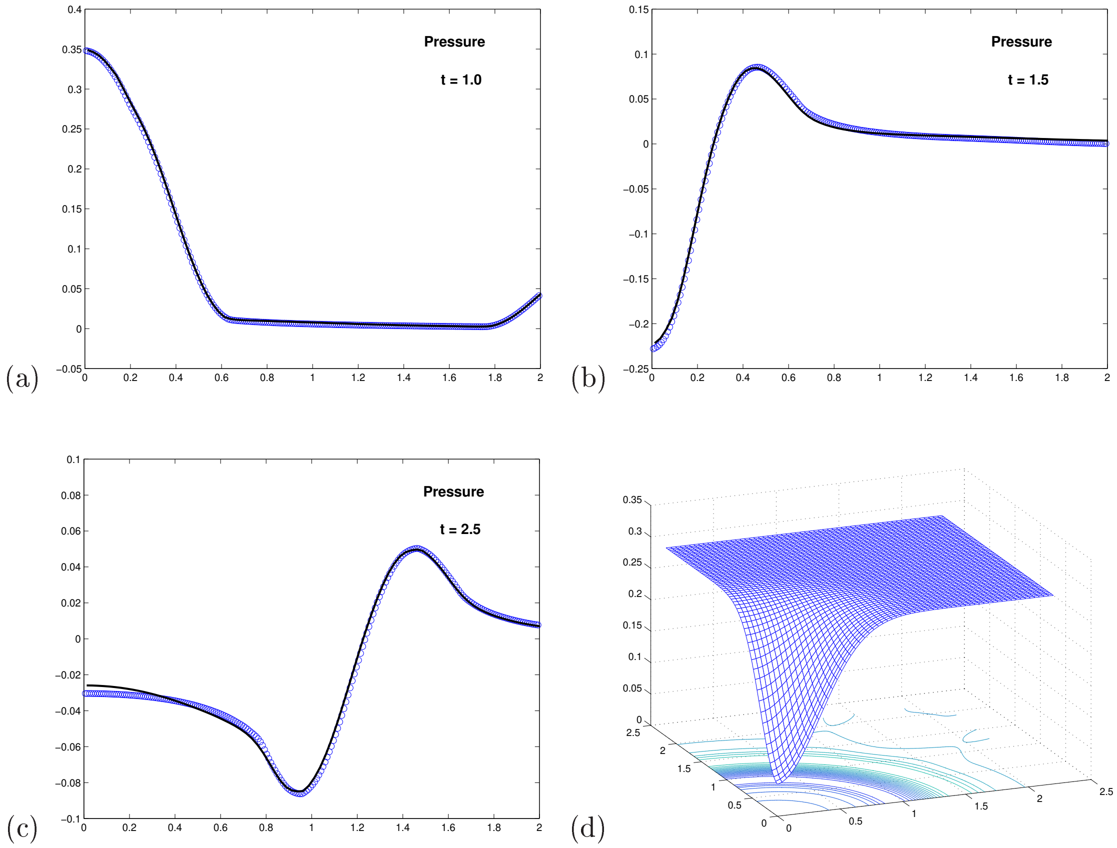


Figure 6.14: The time evolution of the radially symmetric problem. Each image is a scatter plot of all computed data. (a) $t = 1.0$ (b) $t = 1.5$ (c) $t = 2.5$ (d) Contours of the final solution plotted together with the manifold.

the one-dimensional calculation and the new method is good.

6.10.2 Initial conditions with no symmetry

In order to test the multidimensional properties of the new algorithm, initial conditions that share no symmetric axes with the surface dip are used. Specifically, the initial pressure pulse has the functional form

$$p(t=0,x,y) = \begin{cases} a \cos \frac{\pi\,(x^1 - x_l)}{\delta} + 1.0) & \text{if } |x^1 - x_l| < \delta \\ 0 & \text{if } |x^1 - x_l| > \delta \end{cases}, \tag{6.83}$$

where x_l is taken to be a value in the flat region near the left edge of the grid. This pulse is initialized to be right-propagating. As the wave travels to the right, part of it interacts with the dip in the surface as shown in Figures (6.15(a))-(6.15(f)). In these figures, the

Table 6.4: The one-norm values of the difference between pressures and velocities computed by CLAWMANand CLAWPACKas the grid is refined.

	Pressure	1-Velocity	2-Velocity
Cells	$\Vert p_{\text{CLAWPACK}} - p_{\text{CLAWMAN}}\Vert_1$	$\Vert v^1_{\text{CLAWPACK}} - v^1_{\text{CLAWMAN}}\Vert_1$	$\Vert v^2_{\text{CLAWPACK}} - v^2_{\text{CLAWMAN}}\Vert_1$
50	$.361 \times 10^{-4}$	$.236 \times 10^{-4}$	$.173 \times 10^{-4}$
100	$.192 \times 10^{-4}$	$.103 \times 10^{-4}$	$.110 \times 10^{-4}$
200	$.511 \times 10^{-5}$	$.313 \times 10^{-5}$	$.332 \times 10^{-5}$
order	1.9	1.7	1.7

surface is plotted together with the projection of the pressure onto the coordinate plane. They show the time evolution of the pulse as it interacts with the geometry of the dip. The solution that is shown in these figures can be understood by looking at the geodesics of Figure 6.12(b). Two effects that the geometry have on geodesic curves are demonstrated in that figure. First, curves that interact with the edge of the dip converge and cross on the down-wind side of the dip. Second, curves that interact with the central region of the dip can end up heading in the same direction that they came from. Both of these effects are seen in Figures (6.15(a))-(6.15(f)). In Figure 6.15(c) it appears that the pressure wave is following the geodesics of the surface. In the figures that follow, a small pressure wave begins to travel in the direction from which it came. Notice that the portion of the wave that does not interact with the geometry continues on as it would in a Cartesian space.

This solution can be compared to a numerical solution computed by the standard wave propagation algorithm. In order to apply the standard methods to this problem, equation (6.73) needs to be cast in the form (6.76). Specifically, on this surface these equations take the form

$$\begin{aligned}
&p_{,t} - (v^1)_{,1} - (v^2)_{,2} = \left(\frac{b_{,1}b_{,11} + b_{,2}b_{,12}}{1 + (b_{,1})^2 + (b_{,2})^2}\right) v^1 + \left(\frac{b_{,2}b_{,22} + b_{,1}b_{,12}}{1 + (b_{,1})^2 + (b_{,2})^2}\right) v^2 \\
&v^1_{,t} - \left(\frac{1 + (b_{,2})^2}{1 + (b_{,1})^2 + (b_{,2})^2}\right) p_{,1} + \left(\frac{b_{,1}b_{,2}}{1 + (b_{,1})^2 + (b_{,2})^2}\right) p_{,2} = 0 \\
&v^2_{,t} + \left(\frac{b_{,1}b_{,2}}{1 + (b_{,1})^2 + (b_{,2})^2}\right) p_{,1} - \left(\frac{1 + (b_{,1})^2}{1 + (b_{,1})^2 + (b_{,2})^2}\right) p_{,2} = 0.
\end{aligned} \tag{6.84}$$

A Riemann solver that can handle the spatially-varying functions that appear in the flux for the velocity must be used together with a split method to handle the source term in the pressure equation. Figures (6.16(a))-(6.16(f)) show a comparison between the pressures and velocities computed by the standard wave propagation method, and the new algorithm developed in this thesis. Small oscillations that appear in the contours of the solution generated by CLAWMAN are an artifact of setting the contour levels by the maximum and minimum values of the solutions generated by CLAWPACK. These figures show that the two methods generate the same solution.

This is a significant test of the new algorithm for the following reasons:

1. The solutions being compared in Figures (6.16(a))-(6.16(f)) were generated by solving different forms of equation (6.73). The solution generated by CLAWMAN came from the system in the form (6.75), and the solution generated by CLAWPACK came from the system in the form (6.76).

2. Geometric terms that appear in both forms of the equations are handled by each method using entirely different techniques.

3. The CLAWPACK software is extensively used for many different applications. It has been shown to generate accurate and robust solutions.

The agreement of these numerical solutions is demonstrated in Table 6.4. The one-norm difference of the computed solutions on various grids is shown in this table. As the grid is refined, the solutions generated by the two methods converge.

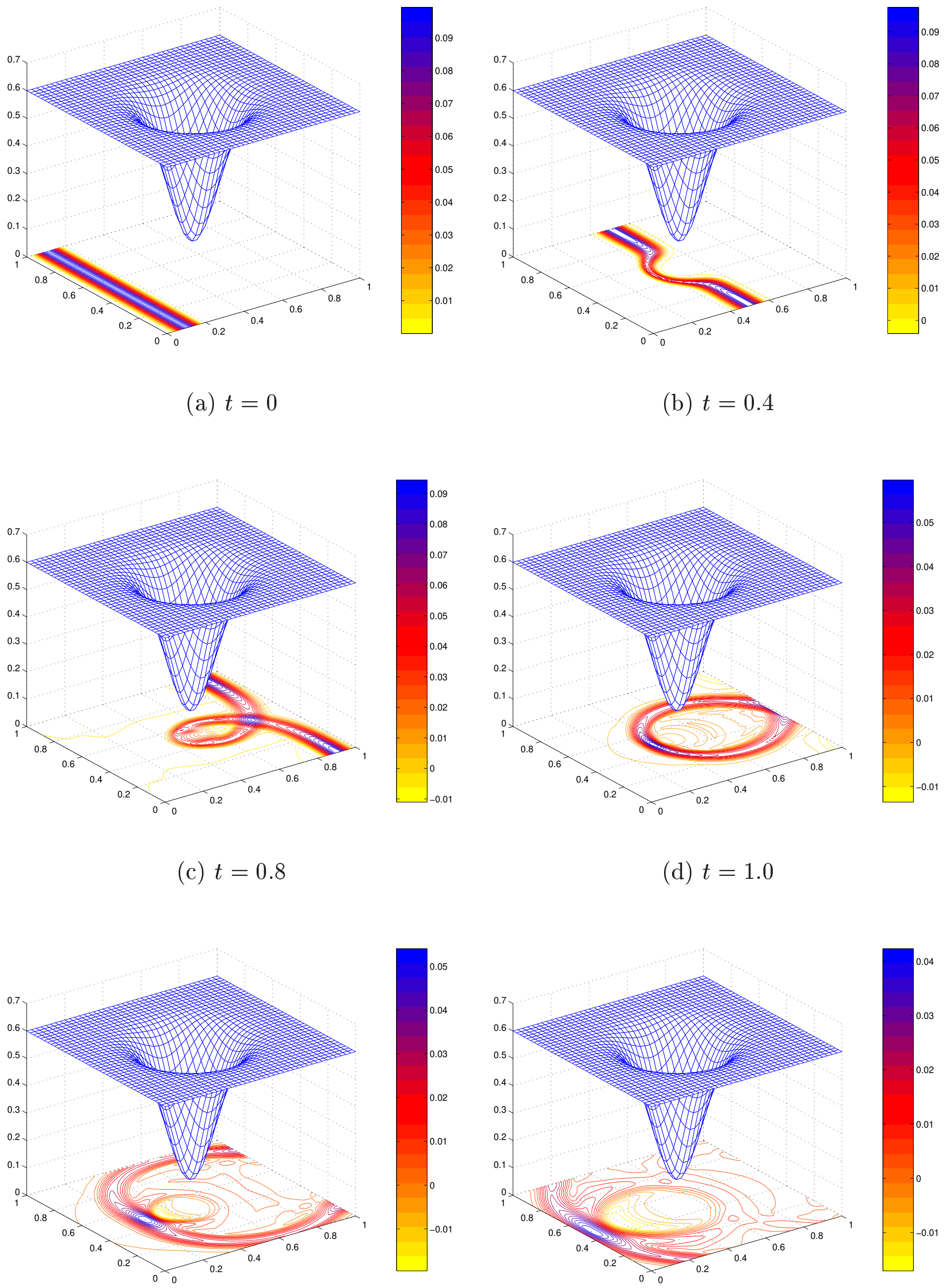
The advantages of using the new algorithm include:

1. Using f-waves locally incorporates the changing volume of grid cells into Riemann problems. The standard algorithm is forced to treat the changing volume with a source term.

2. Using parallel transport locally incorporates the geometrically induced source that appears in the equations for the velocity components.

3. The Cartesian version of the Riemann solver can be used. For complicated systems like relativistic magneto-hydrodynamics, this will likely be a necessary feature.

In the next chapter we discuss an implementation of the new algorithm in the software package CLAWMAN that was used to generate the examples in this chapter.

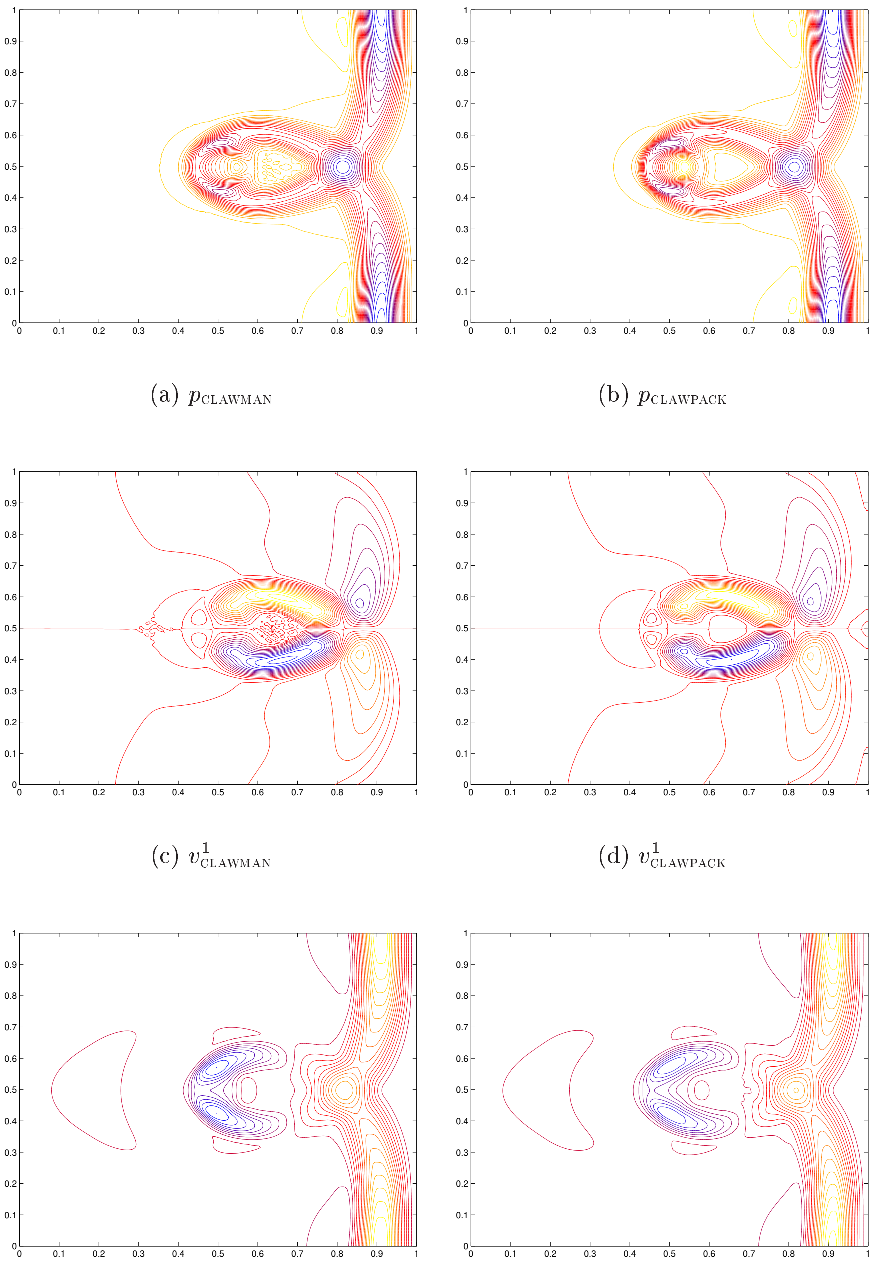


(a) $t = 0$ (b) $t = 0.4$

(c) $t = 0.8$ (d) $t = 1.0$

(e) $t = 1.2$ (f) $t = 1.4$

Figure 6.15: A time sequence of the multidimensional solution. The contours of the solution are plotted on a plane projected down from the manifold.



(a) p_{CLAWMAN} (b) p_{CLAWPACK}

(c) v^1_{CLAWMAN} (d) v^1_{CLAWPACK}

(e) v^2_{CLAWMAN} (f) v^2_{CLAWPACK}

Figure 6.16: A comparison of the pressure and velocity computed by CLAWMAN and CLAWPACK for the multidimensional example.

Chapter 7

THE SOFTWARE PACKAGE CLAWMAN

In the previous chapter, the details for a new class of high-resolution wave propagation algorithms for conservation laws on curved manifolds were presented. These methods have been shown to accurately capture the solution of systems consisting of both scalar and vector equations on various manifolds. One feature of the methods developed in the preceding chapters is that they are quite general. Specifically, the methods were not developed for a particular conservation law on a particular manifold. In fact, they can be applied to a wide variety of hyperbolic systems on general manifolds.

In this chapter, we present an implementation of these algorithms in the software package CLAWMAN (**C**onservation **LAW**s on **MAN**ifolds), a freely available collection of Fortran 90 routines. The implementation in CLAWMAN incorporates the generality of the methods developed in the previous chapters of this dissertation, making it useful to a broad class of users. The goal of this chapter is to give first-time users of the package enough information to be able to begin running, and modifying the examples provided with the code.

The CLAWMAN package is an extension of the CLAWPACK (**C**onservation **LAW**s **PACK**age) software discussed in [41], [42], and [46]. The CLAWPACK software is also freely available and can be found on the web at,

`http://www.amath.washington.edu/~claw`.

The documentation found on the website for CLAWPACK will be helpful for users unfamiliar with the basic implementation of high-resolution wave propagation algorithms.

7.1 Obtaining CLAWMAN

The software package CLAWMAN is freely available on the web at,

`http://www.amath.washington.edu/~claw/clawman`.

First-time users can go to the link that says, "download software", and choose the root directory called

`/clawman/2d`.

This link will transfer the entire contents of the two-dimensional version of CLAWMAN, including some examples described below. The file received is a directory archive that is gzipped for compression. The `.tar.gz` file should be placed in any directory that we will denote by `/install`. A good choice for `/install` is a user's home directory, e.g., `/install` = `/home/dbale`. The file can be unzipped, and the directory structure restored by executing the following commands:

```
>gunzip clawman.tar,

>tar -xvf clawman.tar,
```

where the UNIX or LINUX prompt has been denoted by `>`. Once these commands have been executed, `/install` will contain a directory called `/clawman`. In `/clawman`, there are directories for implementations in various numbers of spatial dimensions. The relevant directory for the methods developed in this thesis is `/2d`. The directory `/2d` contains the entire two-dimensional implementation discussed in this thesis.

Once the file has been downloaded, unzipped, and the directory structure restored, the subdirectory structure underneath `/2d` is organized as follows:

dir: `/modules` – Define and store elements of the code. These modules take the place of common blocks in the Fortran 77 version of CLAWPACK.

- module: `init_module.f90` – Defines and stores initialization parameters, vectors and arrays relevant to the code. The elements of this module are made available to the routines in CLAWMAN using the command `use init_module`. The elements of this module are discussed in Section 7.2.1.
- module: `grid_module.f90` – Defines and stores grid related parameters, vectors, and arrays. Metric information is defined here. The elements of this module are made available to routines by including the statement `use grid_module`.
- module: `step_module.f90` – Defines and stores parameters for each time step. For example, this module stores the current time step if the evolution is using a variable time step.
- module: `slice_module.f90` – Defines parameters and vectors relevant for each one-dimensional slice used to solve Riemann problems. Elements of this module are made available to the routines by including the statement `use slice_module`.

dir: `/cm_claw` – The directory containing the main time-stepping routines.

- file: `cm_claw2.f90` – The main stepping routine.
- file: `cm_step2.f90` Evolves the conservation law over one time step. This routine calls `cm_flux2.f90`.
- file: `cm_flux2.f90` – Parallel transports and orthonormalizes data before calling the Riemann solver.
- file: `init_params.f90` – Initializes code parameters discussed below.
- file: `init_slice.f90` – Allocates the memory for one-dimensional slices used during computation.
- file: `set_slice.f90` – Sets the solution and metric data along the one-dimensional slices on which Riemann problems are solved.

dir: /grid Contains the routines that set the coordinate grid, set the coordinate flux, and perform other operations related to the coordinates. The routines that perform parallel propagation and orthonormalization are in this directory.

- file: set_grid.f90 – Sets the grid according to the defined coordinates. The process by which the coordinates and geometry is set is outlined in Section 7.3.
- file: set_flux.f90 – Computes the spatially varying flux by calling fluxfunct.f90 to compute $\boldsymbol{f}(\boldsymbol{q})$, and then multiplies that result by $\sqrt{h}$.
- file: parallel_trans.f90 – Performs parallel transport on vector components of $\boldsymbol{q}$ from the cell center to the cell edge. This process is described in Section 7.4.
- file: parallel_trans_inv.f90 – Performs the inverse parallel transport from the cell edge to the cell center.
- file: ortho.f90 – Performs the orthonormalization of vector components of $\boldsymbol{q}$ at the cell edge.
- file: plot_grid.f90 – Outputs the location of each grid cell for plotting purposes. This file can be modified to print either cell-centered values, or cell-edge values.
- file: plot_aux.f90 – Outputs the components of the auxiliary array in each grid cell.

dir: /manifolds – Contains three subdirectories /euclidean, /riemannian, and /numerical described below.

- dir: /euclidean – Contains a directory for each coordinate system that covers $\mathscr{M} = \mathbb{R}^2$. For example, one subdirectory found here is /polar which contains the necessary files to describe the polar basis. A general directory, like /polar, will be denoted by /geometry.
  - dir: /geometry – A general directory containing the routines that set the geometry for a given basis.
    - func: hfunct.f90 – Computes the components of the metric tensor $[h_{ij}]$.
    - func: chr.f90 – (optional routine) Computes the Christoffel symbols Γ^i_{jk}.
    - func: dchr.f90 – (optional routine) computes the derivatives of the Christoffel symbols $\mathcal{G}^i_{jkl} = \Gamma^i_{jk,l}$.
- dir: /riemannian – Contains a directory for each Riemannian manifold. Once again, these directories contain the functions that define h_{ij}, Γ^i_{jk} and $\mathcal{G}^i_{jkl}$.
  - dir: /geometry – A general directory containing the routines that set the geometry for a given basis.
    - func: hfunct.f90 – Computes the components of the metric tensor $[h_{ij}]$.
    - func: chr.f90 – (optional routine) Computes the Christoffel symbols Γ^i_{jk}.
    - func: dchr.f90 – (optional routine) computes the derivatives of the Christoffel symbols $\mathcal{G}^i_{jkl} = \Gamma^i_{jk,l}$.

dir: `/numerical` – The reason `chr.f90` and `dchr.f90` are optional in the user-provided directories is that they also provided here. In this directory Γ^i_{jk} and $\mathcal{G}^i_{jkl}$ are approximated with finite differences.

func: `chr.f90` – Computes a finite difference approximation to the Christoffel symbols Γ^i_{jk}.

func: `dchr.f90` – Computes a finite difference approximation to the derivatives of the Christoffel symbols $\mathcal{G}^i_{jkl} = \Gamma^i_{jk,l}$.

dir: `/systems` – Contains a directory for each system of equations that have been implemented. For example, in the distribution version there are two directories, `/advection` and `/scalarfield`. The typical system directory will be referred to as `/system`

dir: `/system` –

dir: `/fluxfunct` – Contains files pertaining to the coordinate flux.

file: `flux2.f90` – A subroutine that returns the coordinate flux $\boldsymbol{f}(\boldsymbol{q})$.

dir: `/rp` – Contains Riemann solvers.

file: `rpn2.f90` – A subroutine that solves the normal Riemann problems and returns $\boldsymbol{\mathcal{A}}^{\pm}\boldsymbol{\Delta Q}$, and each f-wave $\boldsymbol{\mathcal{Z}}^p$. If orthonormalization is being used, this should be a Cartesian-based solver.

file: `rpt2.f90` – A subroutine that solves the transverse Riemann problems and returns the transverse fluctuations.

dir: `/execution` – Contains routines defined by the user.

file: `main.f90` – The main subroutine of the code. This subroutine calls all initialization and time-stepping routines.

file: `user_module.f90` – Defines and stores all user-defined parameters. Any elements needed for a specific application should be defined in this module. These elements are made available to other routines by the statement `use user_module`.

file: `init_user_params.f90` – A user-defined routine that reads the file `init.data` to set the elements defined and stored in the `user_module`.

file: `ic.f90` – The subroutine that sets initial conditions.

file: `param.data` – Input file for the initialization parameters defined in Section 7.2.1. An example file is shown in Table 7.1.

file: `init.data` – Input file for the user-defined parameters discussed in Section 7.2.2. An example of this file is shown in Table 7.2.

dir: `/bc` – Contains routines that set boundary conditions. Boundary conditions specific to a particular application can be kept in the `/execution` directory.

file: `bc2ex0.f90` – A subroutine that fills the ghost cells with zero-order extrapolated data. This sets outflow boundary conditions.

Table 7.1: A typical `param.data` file used to define initialization parameters described in Section 7.2.1.

```
4, 4, 1                      ...  meqn, mwaves, mvec
3                            ...  mvec_v(1:mvec)
1, 2, 2, 0, 0, 0, 2, 2       ...  method(1:8)
4, 4, 4, 4                   ...  mthlim(1:mwaves)
2, 2, 2, 2                   ...  bnd(1:4)
1.0, 0.9                     ...  cfl_v(1:2)
0.0, 1.0, 0.0, 1.0           ...  (x10,x1f),(x20,x2f)
2, 50, 50                    ...  mbc, mx1, mx2
0                            ...  mrstrt
1.0, 10                      ...  tend, mout
999999                       ...  nstep_v(1)
0.005                        ...  dt_v(1)
1,1,2                        ...  mmetric, trans, morder
```

dir: `/limiters` – The limiting process involves the comparison of f-waves at different locations in $\mathscr{M}$. Theoretically, parallel transport should be used to make this comparison for vector components of q. However, extensive tests with exponential curvature demonstrated that this was not necessary.

file: `cm_limiter.f90` – A subroutine that performs the limiting process on f-waves.

func: `cm_philim.f90` – A function that defines the particular limiter being used.

dir: `/misc` – Contains the subroutines that perform a restart in the case that a computation is not starting from an initial condition. The routines in this directory are simply Fortran 90 versions of those found in CLAWPACK.

dir: `/matlab` – Contains Matlab routines (`.m` files) that enable plotting of the results. For more about the visualization of the output see [41].

There are also examples available in the current directory structure. For example, the two dimensional advection equation in polar coordinates, and the Riemannian surface example, both discussed in Chapter 6. With the directory structure in mind, we now discuss the necessary parameters for the code.

7.2 Code parameters

Parameters used in CLAWMAN are separated into two sets. The first set are *initialization* parameters and are necessary to initialize the constants, vectors, and arrays used by the code during the computation. For example, these parameters set the number of equations being solved, whether correction fluxes and transverse waves will be included in the update, the final time for the computation, and other parameters described in Section 7.2.1. The initialization parameters are defined in a file called `param.data` that resides in each `/execution` directory described above. The routine `init_params.f90` is the first routine called from `main.f90` and reads these parameters into the memory module set up by `init_module.f90`. Each routine that has the statement `use init_module` will have access to the elements of the initialization parameters.

Elements of the other set are called *user* parameters and are used to set application specific data. For example, any parameters needed for initial conditions or boundary conditions would be considered user parameters since they would change for each application. These parameters are defined in a file called `init.data` that also resides in each `/ execution` directory. They are read by the routine `init_user_params.f90` that is called directly after initialization parameters are read. The user parameters, and data structures that they help to define are stored in a memory module set up by `user_module.f90`. Just like the initialization parameters, these data structures are available to any routine that includes the statement `use user_module`.

7.2.1 Initialization parameters

Table 7.1 shows a typical `param.data` file used by `init_params.f90` to define the parameters and data structures needed by the code. The parameters shown in the figure are similar to those needed in the CLAWPACK software, but there are some differences that need to be discussed. The parameters needed in the CLAWMAN software are the following:

`meqn`: The number of equations in the system being solved. For the classical or relativistic Euler equations this parameter is set to `meqn` $= n_{eq} + 2$.

`mwaves`: The number of f-waves that that are used to solve Riemann problems. This is almost always set to `mwaves` $=$ `meqn`.

`mvec`: The number of vectors in $\boldsymbol{q}$. For example, the wave equation and the Euler equations have `mvec` $= 1$.

`mvec_v`: A vector with `mvec` components. Each component of `mvec_v` defines the element in $\boldsymbol{q}$ that holds the first component of a vector. For example, the Euler equations have `mvec` $= 1$ and that vector is the momentum $\vec{s}$ with tangent components s^i. The state vector is defined to be $\boldsymbol{q} = (\rho, s^i, e)$, so that `mvec_v(1)` $= 2$. This information is used to apply parallel transport and orthonormalization operations on vector components of $\boldsymbol{q}$.

`method_v`: There are a few differences between these parameters and those in `method` for CLAWPACK. These differences are described in the following:

`method v(2)` – A parameter that determines whether the correction fluxes (**??**) and the source corrections (**??**) are included in the update.

method_v(2) $= 1$ – No correction fluxes or source corrections are included in the update.

method_v(2) $= 2$ – Correction fluxes, but no source corrections are included in the update.

method_v(2) $= 3$ – Both correction fluxes and source corrections are included in the update.

method v(3) – A parameter that determines whether transverse propagation is included in the update.

method_v(3) $= 0$ – No transverse information is used in the update.

method_v(3) $= 1$ – Only first order transverse information is used in the update.

method_v(3) $= 2$ – Correction fluxes are included in the transverse decomposition.

method_v(3) $= 3$ – Both correction fluxes and source corrections are included in transverse decomposition.

method_v(5) – This parameter is defined the same as it is in CLAWPACK. It determines whether a split method will be used for a source term. In CLAWMAN, geometric sources are handled by parallel transport so that method_v(5) $= 0$, unless there is another source present.

method_v(8) – This is a parameter that determines how many components of the auxiliary array will be output. If method_v(8) $\neq 0$, then the components are output to files with names like aux.001.

mthlim: The particular limiter that will be applied to each f-wave.

bnd These parameters can be used to set different boundary conditions along each grid boundary.

cfl_v: Maximum allowed CFL, and desired CFL.

x10: The coordinates grid is the rectangle $[\mathtt{x10}, \mathtt{x1f}] \times [\mathtt{x20}, \mathtt{x2f}]$.

mbc: The number of ghost cells along each boundary.

mx1,mx2: The number of grid cells in each coordinate direction.

mrstrt: Set to zero to begin the computation from the initial conditions. Otherwise, the run is a restart from previously evolved data.

tend: Final time of the computation

mout: Number of times at which the code will output results.

nstep_v: Holds the maximum number of time steps allowed.

Table 7.2: A typical `init.data` file used to define user parameters.

```
-0.5,0.15,2.0,2.0      ...  alf,omega,x0,y0
0.2,0.15,0.11          ...  a,delta,xloc
```

`dt_v`: Holds the initial time step.

`mmetric`: If set to zero, there will be no transformation to an orthonormal frame at each cell interface. Otherwise, the transformation occurs and it is assumed the Riemann solver has no geometry associated with it.

`trans`: If set to zero, there will be no parallel propagation of vector components. In this case, the parameter `method(5)` $\neq 0$ so that the source is handled by a split method. Otherwise, parallel transport handles the geometric source.

`morder`: Determines the order at which parallel transport is carried out. Since the matrices that carry out parallel transport are only computed once at the beginning of the computation, efficiency is not significantly diminished by setting `morder` $= 2$.

7.2.2 User parameters

In Section 6.10 the method developed in this thesis was applied to the scalar field equation on a manifold defined by the surface $b\,(x^1, x^2)$. The calculations presented in that section were carried out using CLAWMAN, and in Section 7.5, the implementation of this particular system will be discussed. Recall from Section 6.10 that the surface had the form

$$b(x,y) = \alpha \; \exp - \frac{\left((x-x_0)^2 + (y-y_0)^2\right)^{1/2}}{\omega^2}, \tag{7.1}$$

and the initial condition was a right-propagating pressure pulse that had the functional form

$$\boldsymbol{q}(t=0,x,y) = \begin{cases} a \, \cos \frac{\pi\,(x-x_l)}{\delta} + 1.0) & \text{if } |x-x_l| < \delta \\ 0 & \text{if } |x-x_l| > \delta \end{cases}. \tag{7.2}$$

Initializing the user parameters for these functions is done in the file `init.data`. Table 7.2 shows the particular file that initializes the necessary parameters for this example. The parameters used are as follows:

- Surface:

  `alf`: Magnitude of the Gaussian dip in the surface that defines the manifold. Denoted by α in the surface definition (7.1).

  `omeg`: Denoted by ω in (7.1).

Table 7.3: A typical function `hfunct.f90` that is used to set the metric coefficients h_{11}, h_{12}, and h_{22}.

```
DOUBLE PRECISION FUNCTION hfunct(mcomp,x1,x2)
 USE user_module
 IMPLICIT NONE
 DOUBLE PRECISION x1,x2,b,bx,by
 INTEGER mcomp
 b = alf*dEXP(-((x1-x0)**2 + (x2-y0)**2)/(omega*omega))
 bx = -b*( (2.d0*(x1-x0))/(omega*omega) )
 by = -b*( (2.d0*(x2-y0))/(omega*omega) )
 go to (11,12,13) mcomp
11 CONTINUE
 hfunct = 1.d0 + bx**2
 RETURN
12 CONTINUE
 hfunct = bx*by
 RETURN
13 CONTINUE
 hfunct = 1.d0 + by**2
 RETURN
END FUNCTION hfunct
```

`x0`: x-location of the center of the Gaussian surface.

`y0`: y-location of the center of the Gaussian surface.

– Initial Conditions:

`a`: Height of the initial pressure pulse denoted by a in the initial condition (7.2).

`delta`: Denoted by δ in (7.2).

`xloc`: center of the initial pressure pulse along the x-direction. This parameter is denoted by x_l in (7.2).

Any other user-defined parameters or data structures that are needed would be added to `init.data`, stored in the module created by `user_module.f90`, and read in by `init_user_params.f90`.

7.3 Setting the geometry on $\mathscr{M}$

The geometry for a specific manifold is set in CLAWMAN by defining the metric tensor, $[h_{ij}]$, the Christoffel symbols, Γ^i_{jk}, and their derivatives $\mathcal{G}^i_{jkl}$. These quantities are used in `set_grid.f90` to define geometric quantities including the volume element $\sqrt{h}$, the matrices $\left[P^i_k\right]$ that represent discrete parallel propagation operators, and the transformation matrices $\left[O^i_k\right]$ that perform the orthonormalization at each cell edge. The components h_{ij}, Γ^i_{jk}, and $\mathcal{G}^i_{jkl}$ are set in the functions `hfunct.f90`, `chr.f90`, and `dchr.f90`, respectively. These functions should be placed in the directory denoted by `/geometry` in Section 7.1. Note that the definition of the geometry for a particular manifold is kept separate from application directories so that one manifold can be used for multiple systems without having to redefine the geometry.

A typical file defining the function `hfunct.f90` is shown in Table 7.3. In this file, components are computed for the two-dimensional metric that describes the surface defined by equation (7.1). This file was used for the implementation of the examples in Section 6.10. The function in Table 7.3 only specifies three of the four components for $\overset{\leftrightarrow}{h}$. This is due to the fact that symmetry of the metric tensor requires that $h_{21} = h_{12}$.

The file `chr.f90` defines the functional form of the Christoffel symbols Γ^i_{jk}. The indices that label each symbol can take on values from 1 to n_d, resulting in a total of n_d^3 Christoffel symbols. However, not all n_d^3 symbols are independent. Using equation (5.61) which defines Christoffel symbols in terms of the metric components, together with the symmetry of the metric, we see that $\Gamma^i_{jk} = \Gamma^i_{kj}$. This symmetry reduces the number of Christoffel symbols that must be set in `chr.f90` to $n_d^2(n_d+1)/2$. In two dimensions, this is a reduction from eight to six, but for three dimensions, the reduction is from twenty seven symbols to sixteen.

The derivatives of the Christoffel symbols are denoted by $\mathcal{G}^i_{jkl}$ and are set in the function defined by `dchr.f90`. Notice that there are n_d^4 possible components, but the symmetry of the Christoffel symbols reduces the number of independent components to $n_d^3(n_d+1)/2$. A further reduction in the number of needed components comes from the fact that parallel transport only occurs along coordinate lines. The necessary components are shown in equation (6.60), where the last two indices of $\mathcal{G}^i_{jkl}$ are always the same. The resulting number of independent components that need to be set in `dchr.f90` is $n_d^2(n_d^2 - n_d + 2)/2$. In two dimensions this reduction is from sixteen to eight components, but in three dimensions it is a reduction from eighty one to thirty six components.

A feature of CLAWMAN that was mentioned in Section 7.1, but should be mentioned again is the fact that `chr.f90` and `dchr.f90` are optional routines. Since Γ^i_{jk} and $\mathcal{G}^i_{jkl}$ can be defined in terms of the derivatives of metric components, the user-defined function in `hfunct.f90` can be used to generate finite difference approximations for Christoffel symbols and their derivatives. Such finite differences have been implemented in `chr.f90` and `dchr.f90` that reside in the `/numerical` directory described in Section 7.1. These routines can be used in place of user-defined functions discussed above. This is a useful feature for metrics with complicated components.

7.4 Solving the Riemann problems

As we saw in Chapter 6, the algorithm solves Riemann problems by incorporating geometric source terms using parallel transport. The algorithm also orthonormalizes tangent components at each cell interface where the Riemann problems are solved. Parallel transport

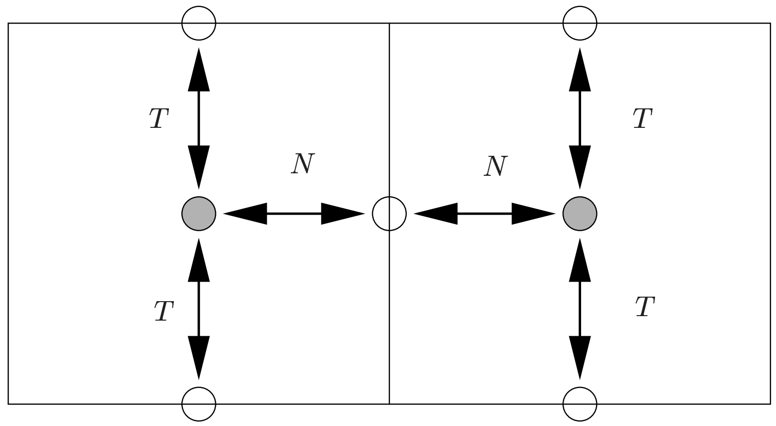


Figure 7.1: Parallel transport to both the edge at which the normal Riemann problem is solved, and the four edges at which transverse Riemann problems are solved.

and orthonormalization are carried out by the matrices `pmat` and `omat`, respectively. These matrices are initialized in `setgrid.f90`, and stored in the `grid_module`. The routines that utilize these matrices to perform the parallel transport, and orthonormalization are called `parallel_trans.f90` and `ortho.f90`, respectively. Figure 7.1 shows the process by which both normal and transverse Riemann problems are solved. The transport paths taken by vector components in a normal Riemann problem are denoted with an "N", and the paths taken for a transverse Riemann problem are denoted with a "T".

7.4.1 Normal Riemann problems

The paths denoted with an "N" in Figure 7.1 depict the process of solving a normal Riemann problem. These steps are listed in the following:

1. Use `parallel_trans.f90` to transport both the left and right cell-centered fluxes to the cell edge. This sets up the modified fluxes (6.62) and (6.63) from Chapter 6.

2. Use `ortho.f90` to orthonormalize tangent components in the modified fluxes.

3. Solve an orthonormal (e.g. Cartesian) Riemann problem that generates fluctuations based on f-waves.

4. Use `ortho.f90` to transform the vector components of fluctuations and f-waves to the coordinate basis at the cell edge.

5. Use `parallel_trans_inv.f90` to transport both the fluctuations and the f-waves to cell centers where they are used in the update (6.72).

Once the normal Riemann problems have been solved, the cell centered fluctuations are used in transverse Riemann problems to propagate information in a multidimensional way.

7.4.2 Transverse Riemann problems

Figure 7.1 also has paths denoted with a "T". These paths depict how vector components of fluctuations are transported to solve transverse Riemann problems. The figure shows that both the left-going and right-going fluctuations are used to compute transverse information at the four contiguous interfaces in the transverse direction. The process by which transverse Riemann problems are solved at one of these interfaces is the following:

1. Use `parallel_trans.f90` to transport the cell-centered fluctuation returned from the normal Riemann problem to the interface.

2. Use `ortho.f90` to orthonormalize tangent components in the modified fluctuations.

3. Solve an orthonormal (e.g. Cartesian) transverse Riemann problem based on this modified data.

4. Use `ortho.f90` to transform tangent components of the transverse fluctuations back to the coordinate basis.

5. Finally, Use `parallel_trans_inv.f90` to transport the transverse fluctuations back to the cell center where they are used to modify the correction fluxes.

The parallel transport of fluctuations returned by normal Riemann solvers are expensive. The process described above occurs four times for each normal Riemann problem. Tests have shown that, though it makes sense to implement parallel propagation for these fluctuations, it is not necessary. Numerical results are only slightly improved by the use of parallel transport of the transverse terms.

7.5 Environment variables

In order to compile and run the examples provided with CLAWMAN, environment variables must be set to make sure that the appropriate files can be found. The necessary environment variables include:

`$INSTALL`: Points to the installation directory denoted by `/install`.

`$CLAWMAN2d`: Points to `$INSTALL/clawman/2d`.

`$EUCLIDEAN`: Points to `$CLAWMAN2d/manifolds/euclidean`.

`$RIEMANNIAN`: Points to `$CLAWMAN2d/manifolds/riemannian`.

`$NUMERICAL`: Points to `$CLAWMAN2d/manifolds/numerical`.

`$SYSTEMS`: Points to `$CLAWMAN2d/systems`.

These environment variables are used in `compile` files in the application's main directory denoted above by `/execution`. Once the proper environment variables are set, the code will compile to the executable file `a.out`. The code can be run with the command

`>a.out`.

Input parameters are read from both `param.data` and `init.data` in the same directory.

Chapter 8

RELATIVISTIC HYDRODYNAMICS

Classical fluid dynamics is unable to capture interesting phenomena produced by the coupling of space and time in the framework of the special and general theory of relativity. In the last decade, high-resolution finite volume schemes based on the solution of Riemann problems have played an important role in the advancement of numerical solutions to the partial differential equations governing inviscid, non-heat conducting, relativistic fluid dynamics (e.g., see [6], [21], [22], [23], [38], [52], [54], [65], [67]). These schemes have been successfully applied in ultra-relativistic regimes where the relativistic effects are important because of the following:

1. The fluid pressure is comparable to the mass density, causing the internal fluid energy to become relativistic.

2. The fluid has a bulk velocity that approaches the speed of light.

3. The fluid dynamics occur in the vicinity of a compact object like a black hole or neutron star.

Old schemes based on artificial viscosity used by Wilson [82] and others [13], [31] were unable to cope with such ultra-relativistic flows. Most high energy astrophysical flows require a relativistic treatment, making applications of relativistic fluid dynamics numerous.

In previous chapters of this thesis we developed a new high-resolution wave propagation algorithm to solve conservation laws on manifolds. The new algorithm has been applied to various classical equations including the Euler equations, and the scalar field equation. In this chapter, applications for relativistic hydrodynamics are presented. The examples are chosen from the literature and represent a preliminary set of tests for the algorithm. Future work, as described in Chapter 9, will involve a more detailed analysis of the performance of the algorithm for relativistic systems.

8.1 Einstein's relativity theory

Albert Einstein introduced the special theory of relativity in 1905. This theory revolutionized the way in which scientists think about space and time. The understanding of time before relativity theory was clearly stated by Newton in the *Principia*, "Absolute, true, and mathematical time, of itself, and from its own nature, flows equably without relation to anything external". This Newtonian notion of absolute time is replaced with a notion of the relative passage of time. Two different observers in spacetime need not agree on the passage of time between the same events. Einstein based the special theory of relativity on two simple principles :

- *No experiment can measure the absolute speed of a spacetime observer.*

- *The speed of light is a universal constant.*

In this thesis we consider units in which the speed of light, c, has the value $c = 1$.

Special relativity is a great success in many areas of physics, however, it is inconsistent with a Newtonian theory of gravity in which the field propagates with infinite speed (*violating the second principle*). In 1915, Einstein made another fundamental contribution to science by introducing the general theory of relativity. The general theory postulates that gravitational forces are not forces in the Newtonian sense, but are generated by a geometric tidal field tensor related to the source density of gravitational field. The theory predicts that free particles travel on geodesics of the spacetime geometry. If there is a local concentration of energy, geodesics in this region will converge (*since the energy density of normal matter is positive*) and observers traveling past this region would precieve a force as the geodesic path deviates from a straight line. An important principal of the general theory insures local consistency with the special theory of relativity. This principle in called the strong equivalence principle and says the following :

- *In a neighborhood of each event in spacetime, physical laws take on their special relativistic form.*

The strong equivalence principle will enable us to solve general relativistic Riemann problems using locally special relativistic frames. This is essentially the same process used to solve Riemann problems in orthonormal frames as described in Chapter 6. The first application of this can be found in [67].

The equations governing the dynamics of the gravitational field in the presence of matter can be written in the simple and elegant form

$$\overleftrightarrow{G} = 8\pi\overleftrightarrow{T}. \tag{8.1}$$

The Einstein tensor, $\overleftrightarrow{G}$, is a second rank symmetric tensor in four dimensions (one time-like and three space-like) that contains information about the spacetime metric and its derivatives. The energy momentum tensor $\overleftrightarrow{T}$ is also of second rank and symmetric, but contains information about matter fields, e.g., electrodynamic or hydrodynamic. The Einstein equations (8.1) describe the way in which matter generates spacetime curvature, and in turn, the curvature of spacetime modifies the flow of matter.

It was mentioned in Chapter 1 that the simple appearance of equation (8.1) is quite deceiving, and simple solutions exist only when one considers systems with physically restrictive symmetries. In fact, the Einstein equations (8.1) are a system of ten nonlinear partial differential equations in one time-like variable, and three spatial variables. In order to solve these equations with little or no symmetry, one must turn to numerical approximations. The numerical evolution of the Einstein equations, however, is also a very difficult task. Shapiro and Teukolsky [72], [73] suggest four requirements for a successful numerical code in relativity. They claim the code should (1) avoid singularities, (2) handle black hole

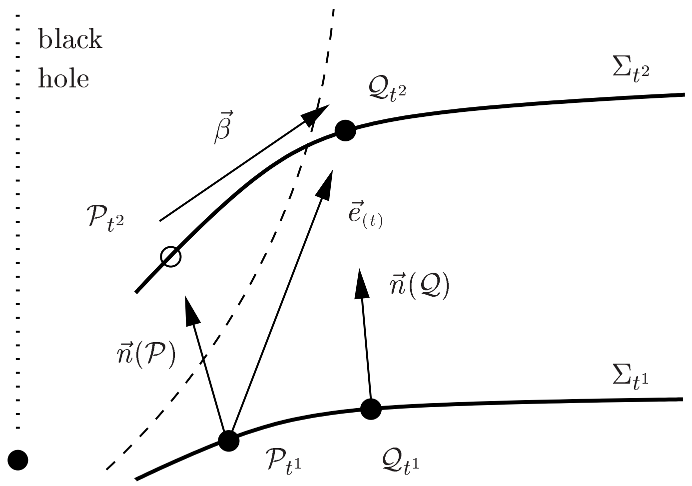


Figure 8.1: $\{3+1\}$ foliation of spacetime showing the role of the shift vector $\vec{\beta}$.

spacetimes, (3) maintain high accuracy, and finally, (4) run forever. These were suggested in 1986, yet there exists no code today (*to the author's knowledge*) that satisfies all of these requirements given moderately general initial and boundary conditions for the fields. For this reason, it is the opinion of the author that, at this time, it is important to develop accurate and stable algorithms not for the full problem, but for systems governed by the simplifications discussed in Chapter 1. Therefore, in the rest of this chapter we focus on the test fluid approximation in which the metric is not evolved in time. Applications of wave propagation algorithms in numerical relativity can be found in Bardeen and Buchman [7].

8.2 {3+1} Foliation of Spacetime

The numerical solution of equations (8.1) usually takes place in the framework of the $\{3+1\}$ formalism [2],[35], [36]. This formalism views the four dimensional spacetime $\{\mathscr{M}, \overleftrightarrow{g}\}$ as a time-foliation of three dimensional space-like hypersurfaces $\{\Sigma_t\}$. Such hypersurfaces are constructed by laying down a coordinate system and holding the time coordinate constant. Each Σ_t has a three-dimensional Riemannian metric $[h_{ij}]$, where here and in what follows Latin indices range over spatial coordinates (i.e. $i, j, k, etc. \in \{1, 2, 3\}$), and Greek indices range over spacetime (i.e. $\alpha, \beta, \mu, etc. \in \{0, 1, 2, 3\}$).

We define *Eulerian* observers, ϑ_E, as those frames at rest in the spatial hypersurface Σ_{t^1}. These Eulerian observers have a four velocity equal to the time-like unit normal at each point of the hypersurface. Two such frames are shown in figure 8.1 at points $\mathcal{P}_{t^1}$ and $\mathcal{Q}_{t^1}$ with the associated time-like unit normals $\vec{n}(\mathcal{P})$ and $\vec{n}(\mathcal{Q})$, respectively. If the observer at $\mathcal{P}$ adopted the non-coordinate basis $\{\vec{n}, \vec{e}_{(j)}\}$, then the spacetime metric in that frame has the form

$$ds^2 = -d\tau^2 + h_{ij}\, dx^i\, dx^j, \tag{8.2}$$

$$[g_{\mu\nu}] = \begin{pmatrix} -1 & 0 \\ 0 & h_{ij} \end{pmatrix}, \tag{8.3}$$

$$[g^{\mu\nu}] = \begin{pmatrix} -1 & 0 \\ 0 & h^{ij} \end{pmatrix}, \tag{8.4}$$

where τ is the proper time for the Eulerian observer. Figure 8.1 shows an Eulerian observer leaving $\mathcal{P}_{t^1} \in \Sigma_{t^1}$, and arriving at $\mathcal{P}_{t^2} \in \Sigma_{t^2}$ after a time $\Delta\tau$. A *spacetime* observer, ϑ_S, with coordinates $x^\mu = \{t, x^i\}$ and basis $\{\vec{e}_{(t)}, \vec{e}_{(i)}\}$ would follow the same path to find an elapsed time $\Delta\, t = N(x^\mu)\, \Delta\, \tau$, and spatial coordinates $x^i(\mathcal{P}_{t^2}) = x^i(\mathcal{P}_{t^1}) - \vec{\beta}\, \Delta\, t$. In these coordinates, the spacetime metric has the form

$$ds^2 = -N^2\, dt^2 + h_{ij}(dx^i + \beta^i\, dt)(dx^j + \beta^j\, dt), \tag{8.5}$$

$$[g_{\mu\nu}] = \begin{pmatrix} -N^2 + \beta_i\, \beta^i & \beta_j \\ \beta_i & h_{ij} \end{pmatrix}, \tag{8.6}$$

$$[g^{\mu\nu}] = \begin{pmatrix} -1/N^2 & \beta^j/N^2 \\ \beta^i/N^2 & h^{ij} - \beta^i\, \beta^j/N^2 \end{pmatrix}, \tag{8.7}$$

where the lapse function, $N(x^\mu)$, and a shift vector, $\vec{\beta}$ have been introduced. Using equations (8.6) and (8.7), it is easy to show that

$$\sqrt{-g} = N\, \sqrt{h}, \tag{8.8}$$

where we have used the notation $g = \det(\overleftrightarrow{g})$ and $h = \det(\overleftrightarrow{h})$. We will make extensive use of this fact in the development of relativistic hydrodynamics.

8.3 Relativistic Hydrodynamics

The flow of a relativistic gas is governed by the local conservation of both matter and energy-momentum

$$\check{\nabla} \cdot \check{J} = 0 \tag{8.9}$$

$$\check{\nabla} \cdot \overleftrightarrow{T} = 0, \tag{8.10}$$

where $\check{\nabla}$ is the spacetime covariant derivative, $\check{J}$ is the current of rest mass, and $\overleftrightarrow{T}$ is the stress-energy tensor from equation (8.1). In terms of the primitive variables of the flow these have the form

$$\check{J} = \rho\, \check{u} \tag{8.11}$$

$$\overleftrightarrow{T} = \rho\, \varepsilon\, \check{u} \otimes \check{u} + p\, \overleftrightarrow{g}, \tag{8.12}$$

where ρ is the proper gas density, $\breve{u}$ is the four velocity of the flow, and ε is the relativistic specific enthalpy

$$\varepsilon = 1 + \epsilon + p/\rho, \tag{8.13}$$

ϵ being the specific internal energy of the fluid. These equations are closed by the normalization of the four velocity, $\breve{u} \cdot \breve{u} = -1$, and an equation of state $p \to p(\rho, \epsilon)$. In the examples presented here we assume an equation of state for a perfect gas

$$p(\rho, \epsilon) = (\gamma - 1)\, \rho\, \epsilon, \tag{8.14}$$

where γ is a constant.

We write down the conservation laws (8.9) and (8.10) on a coordinate basis adopted by spacetime observers ϑ_S discussed in Section 8.2 with basis $\{\vec{e}_{(\alpha)}\}_{\vartheta_S} = \{\vec{\partial}_{(t)}, \vec{\partial}_{(i)}\}$. Flow quantities, however, are most naturally measured by Eulerian observers at rest in the hypersurface with basis $\{\vec{e}_{(\bar{\alpha})}\}_{\vartheta_E} = \{\hat{n}, \vec{\partial}_{(i)}\}$. These bases are related by $\vec{e}_{(\alpha)} = \Pi_\alpha^{\bar{\beta}}\, \vec{e}_{(\bar{\beta})}$, with

$$[\Pi_\alpha^{\bar{\beta}}] = \begin{pmatrix} N & 0 \\ \beta^i & \mathcal{I}^{ij} \end{pmatrix}. \tag{8.15}$$

Using (5.16) and (5.17), it is easy to show that the components of four velocities transform between the two bases as

$$u^{\bar{0}} = N\, u^t \tag{8.16}$$

$$u^{\bar{i}} = u^i + u^t\, \beta^i, \tag{8.17}$$

and similarly

$$u_{\bar{0}} = u^t/N - \left(\beta^k\, u_k\right)/N \tag{8.18}$$

$$u_{\bar{i}} = u^i. \tag{8.19}$$

We can use these relationships to compute properties of the flow in the frame for which the calculation is simple. For example, observers ϑ_E experience a flow Lorentz factor

$$\begin{aligned} W &= -\hat{n} \cdot \breve{u} \\ &= u_{\bar{0}} \\ &= N\, u^t, \end{aligned} \tag{8.20}$$

where we have used the fact that $u_{\bar{0}} = u^{\bar{0}}$ in the frame of ϑ_E. It follows that observers ϑ_E measure a flow velocity of

$$v_i = \frac{\breve{u} \cdot \vec{e}_{(\bar{i})}}{W} \tag{8.21}$$

with contravariant components

$$v^i = u^i/W + \beta^i/N. \tag{8.22}$$

The normalization of the four velocity $\breve{u} \cdot \breve{u} = -1$ allows us to represent the Lorentz factor in the familiar way,

$$W = \frac{1}{\sqrt{1 - \vec{v} \cdot \vec{v}}}, \tag{8.23}$$

where $\vec{v}\cdot\vec{v} = h_{ij}\, v^i\, v^j$.

Consider the set of four vectors

$$\mathscr{Q} = \{\check{J}, \check{s}_{(i)}, \check{e}\}, \tag{8.24}$$

where, once again, $\check{J}$ is the rest-mass current density, $\check{s}_{(i)}$ is the momentum four vector in the $\vec{e}_{(i)}$ direction, and $\check{e}$ is the energy four vector defined as

$$\check{s}_{(i)} = \overset{\leftrightarrow}{T}(\vec{e}_{(i)}, \cdot) = \rho\,\varepsilon\, u_i\, \check{u} + p\, \overset{\leftrightarrow}{g}(\vec{e}_{(i)}, \cdot) \tag{8.25}$$

$$\check{e} = -\overset{\leftrightarrow}{T}(\hat{n}, \cdot) = \rho\,\varepsilon\, W\, \check{u} + p\, \overset{\leftrightarrow}{g}(\hat{n}, \cdot). \tag{8.26}$$

We can write the conservation laws (8.9) and (8.10) in terms of the set $\mathscr{Q}$ of four vectors

$$\check{\nabla}\cdot\mathscr{Q} = \mathscr{S}, \tag{8.27}$$

where, due to the chain rule, the source term has the form

$$\mathscr{S} = \{0, \overset{\leftrightarrow}{T} : \check{\nabla}\vec{e}_{(i)}, \overset{\leftrightarrow}{T} : \check{\nabla}\hat{n}\}. \tag{8.28}$$

Since we are writing the conservation laws onto the coordinate basis of ϑ_S, but we have covariant derivatives of the basis for ϑ_E, we will have the introduction of Christoffel symbols in the source term. We can write

$$\begin{aligned} \overset{\leftrightarrow}{T} : \vec{e}_{(i)} &= T^{\alpha\beta}\, \nabla_\alpha \vec{e}_{(i)\beta} \\ &= T^{\alpha\beta}\left(g_{i\beta,\alpha} - \Gamma^\delta_{\beta\alpha}\, g_{i\delta}\right). \end{aligned} \tag{8.29}$$

Similarly, for the energy equation we can write

$$\begin{aligned} \overset{\leftrightarrow}{T} : \hat{n} &= T^{\alpha\beta}\nabla_\alpha \hat{n}_\beta \\ &= N\left(T^{\alpha t}\,(\log N)_\alpha - T^{\alpha\beta}\,\Gamma^t_{\beta\alpha}\right). \end{aligned} \tag{8.30}$$

We now define our state variables in terms of the time components of the four vectors above

$$\boldsymbol{q} = \begin{pmatrix} -\check{J}(\hat{n}) \\ -\check{s}_{(i)}(\hat{n}) \\ -\check{e}(\hat{n}) \end{pmatrix} \equiv \begin{pmatrix} D \\ S_i \\ E \end{pmatrix}. \tag{8.31}$$

The conservation laws can now be written in standard form

$$\frac{\partial}{\partial t}\boldsymbol{q} + \frac{1}{\sqrt{h}}\left(N\sqrt{h}\,\boldsymbol{f}^{(k)}\right)_{,k} = N\boldsymbol{\psi}, \tag{8.32}$$

where we have made use of the fact that, for a four-vector $\check{X}$, $-\check{X}(\hat{n}) = N\,X^t$, so that, for example, $D = N\,J^t$ and the fluxes have the form

$$\boldsymbol{f}^{(k)} = \begin{pmatrix} D\left(v^k - \beta^k/N\right) \\ S_j\left(v^k - \beta^k/N\right) + p\delta^k_j \\ E\left(v^k - \beta^k/N\right) + p\,v^k \end{pmatrix}. \tag{8.33}$$

8.3.1 Computational difficulties

Though equations (8.50) with flux (8.33) look similar to the classical Euler equations, there are differences that pose computational difficulties worth mentioning. These difficulties can be summarized as follows:

1. One consequence of relativity theory is that mass and energy are proportional (*Einstein's famous* $E = mc^2$). The constant of proportionality $c^2 \sim 10^{17} m^2/s^2$ is very large and so the energy density $E = T^{tt}$ is dominated by the rest mass contribution. Capturing dynamics of the fluid's thermodynamic energy is difficult because of the relative size of the mass contribution. Numerically it makes sense to define an energy perturbation denoted here by $\tau = E - D$, where we note that we are working in units with $c = 1$. Subtracting D from E effectively removes the rest mass energy as measured by observers ϑ_S. It is τ that is used in the conserved vector $\boldsymbol{q}$
$$\boldsymbol{q} = \begin{pmatrix} -\check{J}(\hat{n}) \\ -\check{s}_{(i)}(\hat{n}) \\ -\check{e}(\hat{n}) + \check{J}(\hat{n}) \end{pmatrix} \equiv \begin{pmatrix} D \\ S_i \\ \tau \end{pmatrix}, \tag{8.34}$$
with the following modifications to the flux:
$$\boldsymbol{f}^{(k)} = \begin{pmatrix} D\left(v^k - \beta^k/N\right) \\ S_j\left(v^k - \beta^k/N\right) + p\delta_j^k \\ \tau\left(v^k - \beta^k/N\right) + p\,v^k \end{pmatrix}. \tag{8.35}$$

2. The solution of Riemann problems in most hydrodynamic codes involves using primitive variables of the system, for example, $\{\rho, u_i, p\}$, the density, components of velocity, and pressure, respectively. The algebraic relationship between primitive and conserved variables must be inverted at each time step. This algebraic relationship for a classical inviscid perfect gas has the form
$$\begin{aligned} \rho &= \rho \\ s_i &= \rho\, u_i \\ E &= \frac{p}{\gamma - 1} + \frac{1}{2}\rho\vec{u}\cdot\vec{u}, \end{aligned} \tag{8.36}$$
which is trivially inverted. However, this is not the case for the equations of relativistic hydrodynamics. The relationship between the conserved and primitive variables for this system is
$$\begin{aligned} D &= \rho\, W, \\ S_i &= \rho\,\varepsilon\, W^2\, v_i, \\ \tau &= \rho\,\varepsilon\, W^2 - p - \rho\, W. \end{aligned} \tag{8.37}$$
The difficulty arises due to the presence of the Lorentz factor $W = (1 - \vec{v}\cdot\vec{v})^{-1/2}$, which nonlinearly couples the sets of variables in relativistic regimes for which $W > 1$. An iterative procedure is needed to carry the inversion out in each time step. We follow the procedure discussed in [56] for an equation of state for a perfect gas, $p = (\gamma - 1)\epsilon\rho$.

(a) Define a function

$$f(\tilde{p}) = \tilde{p} - (\gamma - 1)\,\epsilon_\star(\tilde{p})\,\rho_\star(\tilde{p}). \tag{8.38}$$

(b) Compute a $W_\star$ based on the velocity

$$v_i = \frac{S_i}{p + D + \tau}. \tag{8.39}$$

(c) Use $W_\star$ to compute

$$\rho_\star = \frac{D}{W_\star} \tag{8.40}$$

$$\epsilon_\star = \frac{\tau + D(1 - W_\star) + \tilde{p}(1 - W_\star^2)}{D\,W_\star}, \tag{8.41}$$

which gives a value of f to test for convergence.

The procedure is carried out using a bisection method as a predictor, and then a Newton iteration for accuracy.

3. Yet another complication arises in multidimensional relativistic hydrodynamics. Multidimensional finite volume Godunov-type algorithms rely on the solution of one-dimensional Riemann problems normal to each grid interface. The classical Euler equations fit very nicely into this framework due to the fact that the Riemann solution is completely determined by the density, pressure, and normal velocity. The transverse velocity is constant through all normal shock and rarefaction waves. It is the shear waves that carry the jumps in transverse velocities.

   Equations (8.50), however, do not retain this property. The coupling between the normal and transverse directions is expected due to the presence of W. It is not true that transverse velocities are constant through the shock and rarefaction waves of the one-dimensional Riemann problems. Balsara presents details as to how one handles this in [5]. In his paper, a 2-shock approximate Riemann solver treats the transverse velocity exactly for shock waves. In practice, one finds good results by ignoring this coupling. This is most likely due to the cell averaging procedure used in the solution of Riemann problems.

8.3.2 Special relativistic test fluids

The simplest relativistic test fluid is one that flows in Minkowski spacetime $(\mathscr{M}, \overset{\leftrightarrow}{\eta})$, where $\overset{\leftrightarrow}{\eta}$ is a stationary metric defined by

$$[\eta_{\alpha\beta}] = \mathrm{diag}(-1, 1, 1, 1), \tag{8.42}$$

in three spatial dimensions. The Minkowski metric is the relativistic analog of the Cartesian metric. Points in Minkowski spacetime are called events, and are denoted by the vector $\check{x} = (t, x^1, x^2, x^3)^T$ in three spatial dimensions.

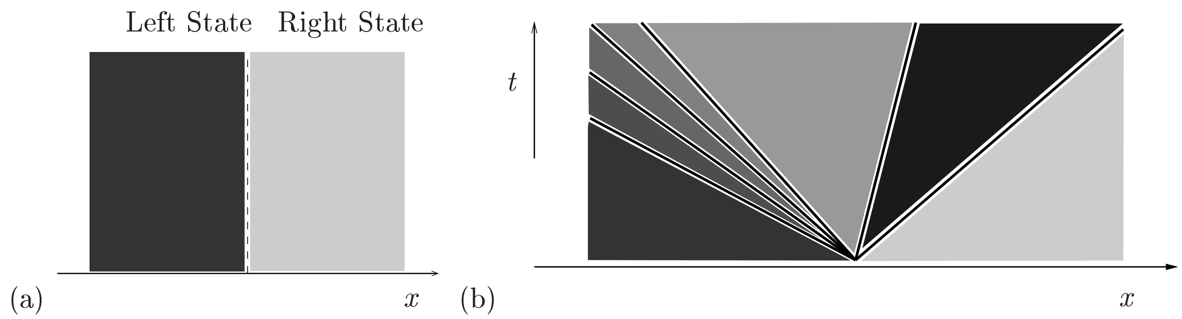


Figure 8.2: (a) Piecewise constant initial data with a jump across an interface. (b) Time evolution of the decay of this initial data.

A fluid element has four-velocity $\check{u} \equiv (d/d\tau)\check{x} = W(1, v^1, v^2, v^3)^T$, where τ is the proper time and $\vec{v}$ is the element's three-velocity. As previously discussed, the governing equations are the laws of local conservation of mass, momentum, and energy density. The energy momentum tensor $\overset{\leftrightarrow}{T}$, and the rest mass current density $\check{J}$ have components

$$\begin{aligned} T^{\alpha\beta} &= \rho h\, u^\alpha\, u^\beta + p\, \eta^{\alpha\beta}, \\ J^\alpha &= \rho\, u^\alpha, \end{aligned} \tag{8.43}$$

where ρ is rest mass density, $h = c^2 + \epsilon + p/\rho$ the relativistic specific enthalpy, ϵ the specific internal energy, and p is the fluid pressure. In an inertial (lab) frame with basis $\{\vec{\partial}_{(t)}, \vec{\partial}_{(i)}\}$, so that $[\nabla_\nu] = (\partial_t, \partial_{x^1}, \partial_{x^2}, \partial_{x^3})$, the conservation equations (8.50) can be written

$$\begin{aligned} &D_{,t} + \left(D\, v^k\right)_{,k} = 0 \\ &{S^i}_{,t} + \left(S^i\, v^k + p\, \delta^{ik}\right)_{,k} = 0 \\ &\tau_{,t} + \left(S^k - D\, v^k\right)_{,k} = 0, \end{aligned} \tag{8.44}$$

where i ranges over the spatial dimensions $\{1, \cdots, n_d\}$ and the definitions (8.37) have been used. An equation of state for a perfect gas

$$p = (\gamma - 1)\ \rho\, \epsilon, \tag{8.45}$$

closes equations (8.44), where γ is the ratio of specific heats for the fluid.

There are a number of well tested one dimensional examples in the literature. We present two such examples here, both will be Riemann problems with piecewise constant initial data. The initial setup is shown in Figure 8.2(a), and consists of two piecewise constant fluid states on each side of an imaginary membrane. The membrane is removed at $t = 0$, and the initial data decays into waves that propagate away from the interface as shown in Figure 8.2(b).

The first test problem consists of a high-pressure, dense gas on the left that expands into a less dense gas at low pressure. As this Riemann data decays, a left-going transonic

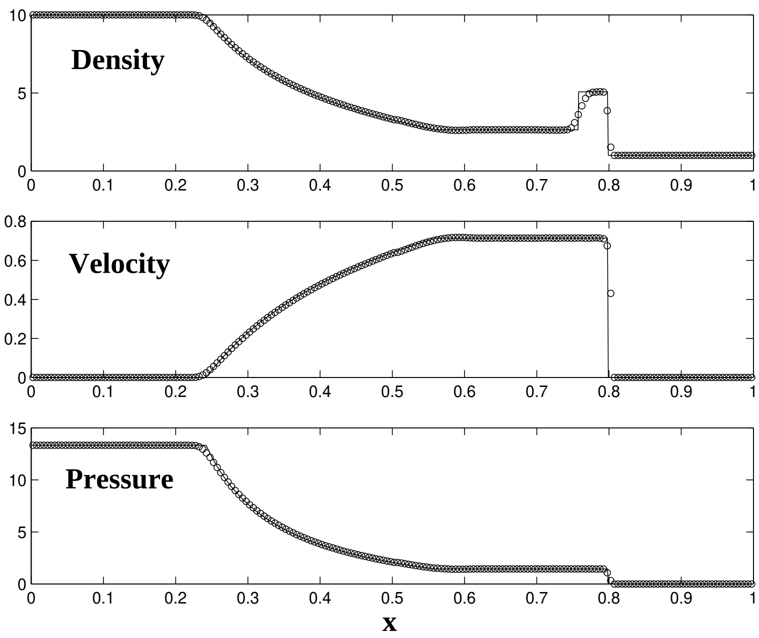


Figure 8.3: Comparison of computed solution and an exact solution for the first special relativistic example. The computed solution is shown as circles and the exact solution as a solid line.

rarefaction wave is produced. Numerically, the left variables at $t = 0$ are initialized to $\boldsymbol{q}_l = (\rho, u^x, p)^T = (10.0, 0.0., 13.3)^T$ and the right to $\boldsymbol{q}_r = (1.0, 0.0, 0.00000067)^T$. These numbers have been used in the literature, e.g., [55] and [57].

Figure 8.3 compares the solution generated by the wave propagation algorithm developed in this dissertation with an exact solution. In this figure, circles represent the computed solution and the solid line represents the exact solution. There are 100 cells across the domain in the computation and the gas constant was set to $\gamma = 5/3$. The new method compares well with those found in the literature, e.g., see [55] and [57]. The strong transonic rarefaction wave has been captured accurately, and the shock wave in the velocity is captured within two grid cells.

The second special relativistic Riemann problem presented in this section, also found in [55] and [57], involves a large jump in the pressure across the membrane. Once the initial data decays, the solution contains a thin dense shell created by a strong right-going shock. The left state at $t = 0$ is defined by $\boldsymbol{q}_l = (1.0, 0.0, 1000.0)^T$, and the right state by $\boldsymbol{q}_r = (1.0, 0.0, 0.01)^T$. This problem is more severe than the first example due to the fact that the pressure jumps by five orders of magnitude across the interface in the initial conditions.

Figure 8.4 shows the comparison of the computed solution with an exact solution. Once again, circles represent the computed solution and the solid line represents the exact solution.

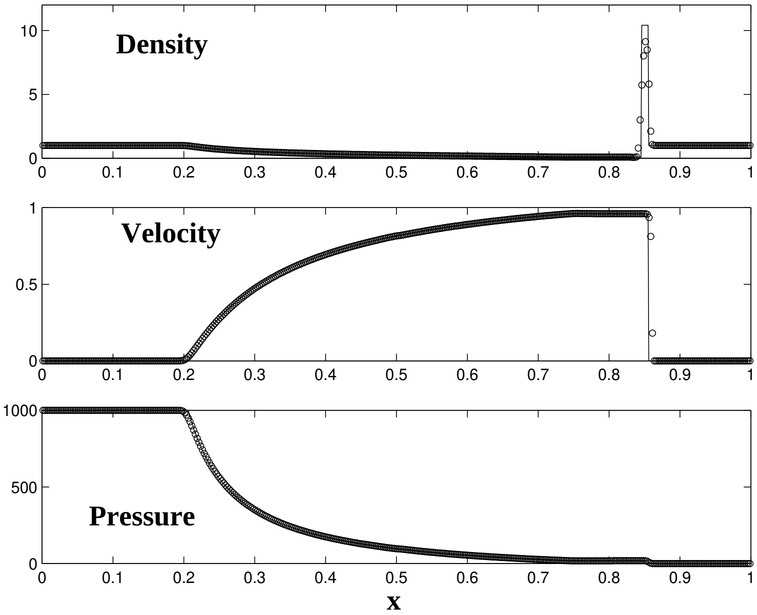


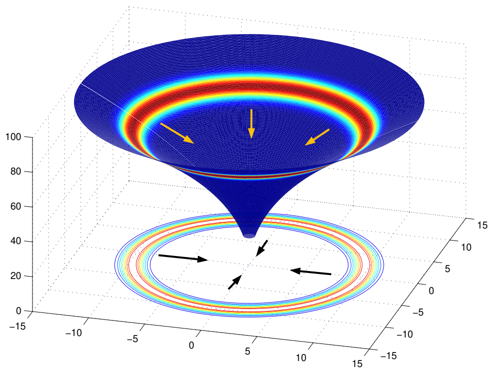
Figure 8.4: Comparison of computed solution and an exact solution for the second special relativistic example. The computed solution is shown as circles and the exact solution as a solid line.

The computations shown in this figure have been done with 400 cells across the domain. Once again, the gas constant was taken to be $\gamma = 5/3$. Notice that, once again, the strong shock in the velocity has been captured in two grid cells, and the rarefaction wave is accurately represented by the numerical results. The thin dense shell is also captured by the method.

8.3.3 Spherical accretion of dust onto a Schwarzschild black hole

In the next example, the relativistic accretion of dust onto a spherically symmetric black hole is computed. This test for the algorithm has been used by other authors, e.g., [6] and [31]. The assumption for this calculation is that the pressure remains negligible as the fluid particles flow inward along geodesics of the spacetime manifold. The geometry of the manifold is described by a metric that represents the Schwarzschild solution to the Einstein equations (8.1) with $\overleftrightarrow{T} = 0$. Using spherical coordinates, $\{t, r, \theta, \phi\}$, the Schwarzschild metric has the form

$$[g_{\alpha\beta}] = \begin{pmatrix} 1 - 2M/r & 0 & 0 & 0 \\ 0 & \frac{1}{1-2M/r} & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2 \sin^2(\theta) \end{pmatrix}, \tag{8.46}$$



(a) Schwarzschild Black Hole

Figure 8.5: An embedding diagram of the Schwarzschild spacetime manifold with ϕ held fixed, and r and θ allowed to vary. Also depicted is a wave traveling towards the black hole singularity.

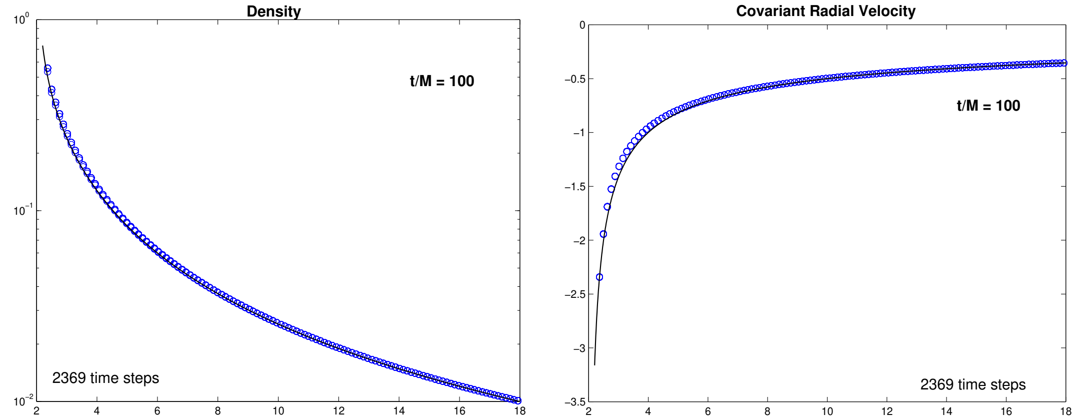
where M is the mass of the black hole. Notice that there is a singularity in the metric at $r = 2M$, the Schwarzschild radius. The metric singularity at this radius is only due to the coordinates, and can be removed by transforming to a new coordinate system. The only true spacetime singularity, e.g., where the curvature grows without bound, occurs at the center of the black hole.

Figure 8.5 shows an embedding diagram for a Schwarzschild black hole. The two-dimensional surface represents a function of the spacetime curvature along a slice with ϕ held fixed. Therefore, the surface is plotted in the coordinates $\{r, \theta\}$. This two-dimensional coordinate plane is discretized into the numerical domain $\{r_i, \theta_j\}$ with $r \in [r_0, r_f]$ and $\theta \in [\theta_0, \theta_f]$.

The initial conditions are those of vacuum, e.g., all variables are set to negligible values in each grid cell. The boundary conditions in the radial direction are set so that at $r = r_0$, zero-order outflow conditions are imposed, and at $r = r_f$ the exact inflow solution is imposed. In the θ-direction, there is no need to compute on the entire domain that has $\theta_f = 2\pi$. Instead, the computational domain is defined such that $\theta_f = \pi$, and symmetry preserving reflecting boundary conditions are used at both $\theta_0 = 0$ and $\theta_f = \pi$.

The initial data is allowed to evolve to a steady state, at which time the computed solution is compared with an exact solution. Figures 8.6(a) and 8.6(b) show the comparison of a computed solution with the exact solution on a grid with 160×60 cells after 2369 time steps. The solid lines represent the exact solution, while the circles represent computed values at all locations in the grid as a function of r. In the calculation shown, $r_0 = 2.2M$ and $r_f = 18M$.

The computed steady state compares well with the exact solution. However, notice that there is a small scatter in the data as r approaches $2M$. This is due to the fact



(a) Density

(b) Velocity

Figure 8.6: Spherically symmetric accretion of dust onto a Schwarzschild black hole. An exact solution is compared to the two-dimensional computed data as a function of r. The solid line represents the exact solution, and circles represent the computed data. (a) The steady state density profile. (b) The steady state radial coordinate velocity.

that the metric is singular near three of the four boundaries of the numerical domain. We have already discussed the singularity at the inner boundary where r is approaching the Schwarzschild radius. In addition to this, the metric (8.46) is also singular at both $\theta = 0$ and $\theta = \pi$.

This application is a good test of the algorithm for the following reasons first noted by Hawley, Smarr, and Wilson [31]:

- The problem tests the radial transport terms in the code. These terms show up in the spatially-varying flux functions since

$$\det(\overset{\leftrightarrow}{h}) = \frac{r^4 \sin^2(\theta)}{1 - 2M/r}.$$

  It is the f-waves that handle these terms in the new algorithm.

- The problem also tests the ability of the method to handle general relativistic red-shift, and the effects of the lapse N.

- Finally, the problem tests that the code properly normalizes the four velocity of the fluid as needed in (8.23).

However, since the flow is only radial, the multidimensional components of the algorithm are not tested. For this, we turn to the next example.

8.3.4 Relativistic flow of an irrotational, isentropic perfect fluid

The relativistic Euler equations for an irrotational, isentropic perfect fluid have been formulated as a potential problem in several papers [24], [66]. This is done by making two assumptions:

1. The fluid obeys the equation of state $p = \rho \propto n^2$, where p is the pressure, ρ is the mass-energy density, and n is the baryonic number density.

2. The fluid is a perfect fluid with no vorticity.

Since the fluid has no vorticity, the velocity can be written as the gradient of a potential function

$$h\, u_\alpha = \varphi_{,\alpha}, \tag{8.47}$$

where $h = (p + \rho)/n$ is the specific enthalpy, and $\breve{u}$ is the four velocity of the fluid. The equation of continuity, together with assumption (2) above leads to an equation for φ that has the form

$$(\varphi^{,\alpha})_{;\alpha} = 0, \tag{8.48}$$

where, once again, the semi-colon represents the covariant derivative with respect to the spacetime metric, and a sum is implied on α. This equation is the relativistic extension of the scalar field equation studied in Chapter 6.

In this example, equation (8.48) is solved in a Schwarzschild spacetime generated by a massive hard sphere. The metric that describes the geometry is once again the Schwarzschild metric (8.46). The grid on which the calculation was done is the same grid that was used in the preceding example. Since ϕ is fixed, no derivatives with respect to that variable must be considered.

As we did for the scalar field equation of Chapter 6, we write equation (8.48) as a system of first-order conservation laws by defining the auxiliary variables:

$$p = \varphi_{,t}, \qquad \psi_r = \varphi_{,r}, \qquad \psi_\theta = \varphi_{,\theta}. \tag{8.49}$$

Using these variables in equation (8.48), the system that needs to be solved can be written as

$$\boldsymbol{q}_{,t} + \boldsymbol{f}^{(r)}_{,r} + \boldsymbol{f}^{(\theta)}_{,\theta} = \boldsymbol{\psi}, \tag{8.50}$$

where

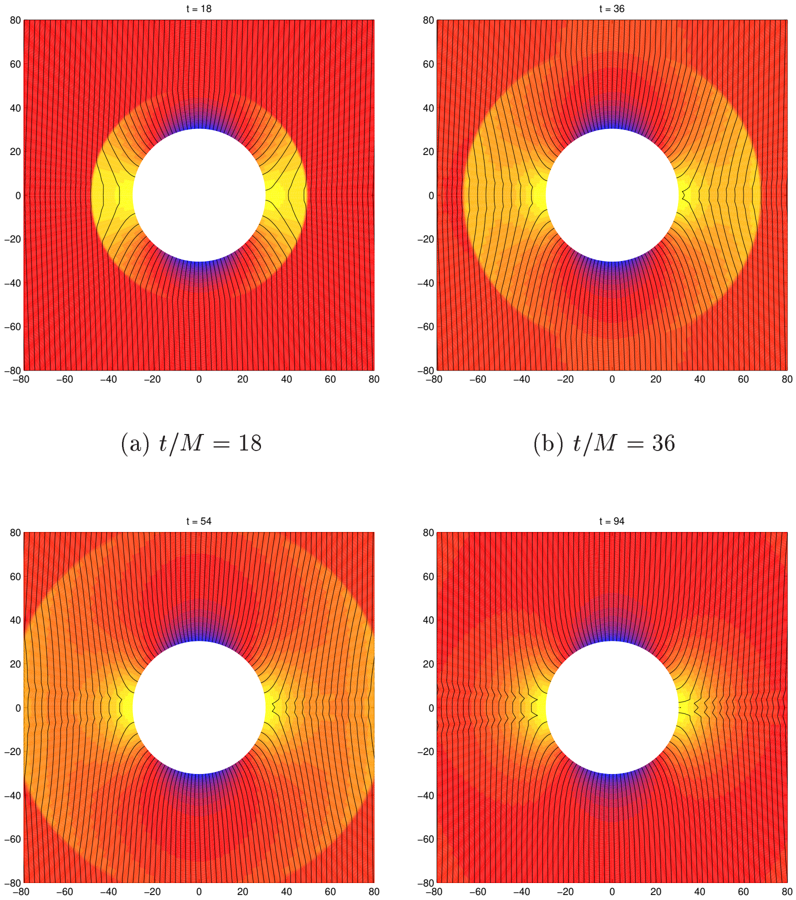
$$\boldsymbol{q} = \left(p, \psi^r, \psi^\theta\right)^T \tag{8.51}$$

is the state vector of unknowns, and

$$\boldsymbol{f}^{(r)} = \left(-(1 - 2M/r)^2\, \psi^r, -p, 0\right)^T \tag{8.52}$$

is the flux along $\vec{\partial}_{(r)}$, and

$$\boldsymbol{f}^{(\theta)} = \left(1 - (1 - 2M/r)\, \frac{1}{r^2}\, \psi^\theta, 0, -p\right)^T \tag{8.53}$$



(a) $t/M = 18$

(b) $t/M = 36$

(c) $t/M = 54$

(d) $t/M = 94$

Figure 8.7: The time evolution of flow past a relativistic hard sphere. Plotted are the equi-potential lines of the flow. Particles travel along lines that are perpendicular to these contours.

is the flux along $\vec{\partial}_{(\theta)}$. Finally, the source terms that appear on the right hand side of equation (8.50) have the form

$$\boldsymbol{\psi} = \left((1 - 2M/r)\ (1 - 3M/r)\ \frac{2}{r}\,\psi^r + (1 - 2M/r)\,\frac{\cos(\theta)}{r^2\,\sin(\theta)}\,\psi^\theta, 0, 0 \right)^T . \tag{8.54}$$

Figures (8.7(a))-(8.7(d)) show the time evolution of a flow past the hard sphere. These figures show contours of equi-potential lines. Particles flow along paths that are perpendicular to these contours. The grid, $\{r_i, \theta_j\}$, in this calculation covers $r \in [30.4M, 700M]$ and $\theta \in [0, \pi]$ with 200 radial cells, and 60 cells in the θ-direction. The initial data has been set so that the flow is from left to right with

$$p(t=0) = -\left((1-2M/r)\left(1+(1-2M/r)\,\psi_{,r}^2 + \frac{1}{r^2}\psi_{,\theta}^2\right)\right)^{1/2} \tag{8.55}$$

$$\psi_r(t=0) = u_\infty\,\cos(\theta) \tag{8.56}$$

$$\psi_\theta(t=0) = -u_\infty\, r\,\sin(\theta), \tag{8.57}$$

where u_∞ is the right-going velocity at a large distance away from the sphere. The boundary conditions are such that at the outer radius of the grid, the conditions at infinity are imposed. At the surface of the sphere, the radial component of the velocity is reflected to ensure the behavior of a hard surface.

Notice that in Figures (8.7(a))-(8.7(d)) there is a sharp wave traveling outward from the surface of the sphere at the speed of light. The theory of relativity does not allow anything to travel faster than this wave. This wave represents the flow adjusting to the presence of the hard sphere. The out-going wave leaves behind it a steady-state flow around the hard sphere.

Chapter 9
CONCLUSIONS AND FUTURE WORK

The work presented in this thesis is naturally separated into three categories: new algorithms, software, and applications. The contributions of this work in each category are discussed in the following sections. The final section of this chapter discusses future directions of the work presented in this dissertation.

9.1 A new class of wave propagation methods

The main contribution of the work presented in this dissertation is a new class of wave propagation methods to solve hyperbolic conservation laws on curved manifolds. Two novel features of these algorithms include:

1. Riemann problems are solved by decomposing modified flux differences into f-waves. These f-waves where shown to handle spatially-varying flux functions by appropriately modifying wave strengths to generate the correct solution to the generalized Riemann problem. For conservation laws on a manifold expressed in a coordinate basis, the spatially-varying flux was shown to take the form
$$\boldsymbol{f}(\boldsymbol{q}, x^k) = \sqrt{h}\, \boldsymbol{f}(\boldsymbol{q}).$$
In this case, the f-wave decomposition was shown to to naturally incorporate changing cell volumes into the strength of propagating waves.

2. Source terms of the form
$$\boldsymbol{\psi} = -\Gamma^i_{jk} T^{jk}$$
that appear for the components of vector conservation laws were incorporated into Riemann problems by using parallel transport. Each vector component of the flux is transported from cell center to cell edge prior to the solution of Riemann problems. This process sets up the modified initial data for each Riemann problem.

These two features of the algorithm incorporate all geometric influence of the manifold into the local Riemann problems. The motivation for this is that the conservation law (1.2) is a local statement of conservation. The algorithms developed in this thesis respect this and treat the conservation locally.

In addition to these features, the algorithm utilizes the solution of Riemann problems in orthonormal sets of basis vectors. The advantage of using such bases is that standard Riemann solvers based on Cartesian bases can be used directly, as long as they are modified to use f-waves. This is an advantage for systems, e.g., general relativistic magnetohydrodynamics, that have a complicated wave structure in the solution of Riemann problems. It will be useful to keep the Riemann solvers for such systems as simple as possible.

Though the main focus of the work in this thesis was on developing an algorithm for conservation laws on manifolds, this led to the development of wave propagation algorithms for general spatially-varying flux functions, and algorithms for conservation laws with general source terms. In Chapter 3, we developed a new wave propagation algorithm based on f-waves to solve conservation laws with spatially-varying flux functions. Such fluxes appear in many different physical applications. The method based on f-waves was shown to perform well on both a linear scalar equation and a nonlinear system.

In Chapter 4, we developed two different wave propagation methods to handle source terms that appear in various applications. The first method was based on quasi-steady assumptions, and was shown to perform well near a steady state. However, this method fails in some cases when the solution is dynamic. The second method, based on modified flux differences, performed well for both steady-state and dynamic solutions. The new algorithm for conservation laws on manifolds was shown to implement the second method based on modified flux differences.

9.2 Software

The algorithms developed in this thesis can be applied to many different hyperbolic conservation laws on general manifolds, making them useful to a broad class of researchers. To make use of this generality, the algorithms have been implemented in an equally general way. The software package CLAWMAN, described in Chapter 7, implements these algorithms so that the user only has to provide:

1. A Cartesian-based Riemann solver for the particular system of conservation laws needed for the application. The Riemann solver need only be modified to decompose the flux difference into f-waves as described in Chapter 3.

2. The routine `hfunct.f90`, shown in Table 7.3, which sets the components of the metric tensor $[h_{jk}]$. The metric describes the geometry on $\mathscr{M}$.

3. The routines `user_module.f90` and `init_user_params.f90` described in Section 7.2.2 that set any user-defined data structures and parameters.

4. The routines that set initial and boundary conditions.

This code is freely available and can be downloaded at

`http://www.amath.washington.edu/~claw/clawman`

with applications included. The number of available applications will continue to grow as the code is applied to more conservation laws and different manifolds.

9.3 Applications

We have discussed many different applications throughout this thesis to verify that the proposed methods work well on practical problems. In Chapter 3, we presented results generated by the method based on f-waves for two applications. The first was a modified Burgers' equation with a flux that depended linearly on the spatial variable. In the second application, we presented results for the quasi-one-dimensional flow of a gas in a duct with variable cross section. In both examples the method based on f-waves was shown to perform well.

In Chapter 4, we presented results generated by two wave propagation methods for conservation laws with source terms. The examples in this chapter involved the Euler equations of a gas under the influence of a stationary gravitational field. An isothermal equilibrium was set up and pressure perturbations were set in motion on top of that equilibrium. The solutions of the two new methods were compared with a solution generated by a split method. Both of these new methods were shown to perform better than the split method for this example.

The examples that test the main algorithm of this thesis were presented in Chapter 6. The purpose of these examples was to show that the new algorithm for conservation laws on manifolds performs well on different systems of equations on various manifolds. The examples shown in Chapter 6 where done using CLAWMAN. In the first example, the Euler equations were solved in a curvilinear set of coordinates that cover $\mathbb{R}^2$. This example shows that the method can be used for conservation laws in curvilinear coordinates. The next set of examples in Chapter 6 involve a linear system of equations on a Riemannian manifold. These examples demonstrate that the new algorithm performs well on curved manifolds.

The motivation for developing wave propagation algorithms for conservation laws on curved manifolds has been the solution of the equations governing relativistic test fluids. The geometry of the spacetime manifold affects the flow of such fluids. Chapter 8 includes several applications of the methods discussed in this thesis to standard problems for relativistic fluids. First, two special relativistic Riemann problems are presented. The first Riemann problem has a strong transonic rarefaction wave, and the second involves a thin dense shell created by a strong shock wave. The method performs well on both problems.

The second example in Chapter 8 presents the spherical accretion of dust into a Schwarzschild black hole. In this case, the manifold is curved by the mass of the black hole. Finally, a multidimensional application is presented for an isentropic, irrotational perfect fluid flowing past a relativistic hard sphere. In this example, the speed of sound is equal to the speed of light.

9.4 Future directions of this work

The algorithms developed in this dissertation, and the implementation of them in CLAWMAN is general so that they can be applied to different systems of conservation laws on a variety of manifolds. This generality allows for many directions of future research.

Most of the work presented in this thesis concerned developing general algorithms for solving conservation laws on Riemannian manifolds. There are many classical systems that can be investigated with the methods developed in the previous chapters. For example, the propagation of elastic waves on the curved surface of a membrane or shell would be an interesting application of the method.

The main motivation for developing these algorithms was to solve the equations of general relativistic hydrodynamics. Since there are only a few high-resolution finite volume methods based on the solution of Riemann problems that have been applied to relativistic hydrodynamics, there are many directions for further application of the new methods in this field. The applications presented in Chapter 8 are only a preliminary set of relativistic systems to which the new methods can be applied.

Numerical results for the accretion of relativistic gases onto moving black holes have been reported in several papers, e.g., see [6] and [23]. The methods presented in this thesis will be applied to such problems and the results will be compared with those found in the literature.

In the relativistic accretion problems, the curvature of the manifold is centered on the black hole. The boundaries of the computation are easy to implement if they are out in the flat regions away from the hole. Unfortunately, the grid must be fairly large to accomplish this while maintaining good grid refinement. Eventually, the methods presented in this thesis will be combined with the adaptive refinement capabilities of the software package BEARCLAW. This will enable the placement of boundaries in flat regions of black hole spacetimes. It will also allow the investigation of the breakup of small objects like single stars as they plunge into the central regions of black holes.

Finally, the implementation in CLAWMAN is based on a metric that is independent of time. However, there is no reason that the metric cannot be dependent on time. This would allow the methods developed in this thesis to be applied to moving surfaces, or the full Einstein equations in which the metric tensor is evolving in time with the fluid variables. Bardeen and Buchman [7] are applying wave propagation algorithms to the vacuum Einstein equations. Eventually, the goal is to combine the methods for relativistic test fluids developed in this thesis with wave propagation methods of the form they are using to produce an algorithm that is capable of solving the full Einstein equations.

BIBLIOGRAPHY

[1] Miguel A. Aloy, José Ma. Ibáñez, José Ma. Martí, José L. Gómez, and Ewald Müller. High-resolution 3d simulations of relativistic jets. *Astrophys. J.*, 523:125, 1999.

[2] R. Arnowitt, S. Deser, and C. W. Misner. *Gravitation: An Introduction to Current Research.* John Wiley, New York, 1962.

[3] D. S. Bale. CLAWMAN software. available on the Web at the URL `http://www.amath.washington.edu/~claw/clawman`.

[4] D. S. 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. Submitted, 2001.

[5] Dinshaw S. Balsara. Riemann solver for relativistic hydrodynamics. *J. Comput. Phys.*, 114:284–297, 1994.

[6] Francesc Banyuls, José A. Font, José Ma. Ibáñez, José Ma. Martí, and Juan A. Miralles. Numerical $\{3+1\}$ general relativistic hydrodynamics: A local characteristic approach. *Astrophys. J.*, 476:221–231, 1997.

[7] J. Bardeen and L. Buchman. Numerical tests of evolution systems, gauge conditions, and boundary conditions for 1d colliding gravitational plane waves. Submitted, 2001.

[8] Richard L. Bishop and Samuel I. Goldberg. *Tensor Analysis on Manifolds.* Dover, 1980.

[9] C. Bona and J. Massó. A hyperbolic evolution system for numerical relativity. *Phys. Rev. Letters*, 68:1097, 1992.

[10] C. Bona and J. Massó. Numerical relativity: Evolving spacetime. *International Journal of Modern Physics C:Phys. and Computers*, 4:883, 1993.

[11] Carles Bona, Joan Massó, Edward Seidel, and Joan Stela. New formalism for numerical relativity. *Phys. Rev. Letters*, 74:600, 1995.

[12] D. L. Brown. An unsplit Godunov method for systems of conservation laws on curvilinear overlapping grids. *Mathl. Comput. Modelling*, 20(10/11):29–48, 1994.

[13] J. Centrella and J. R. Wilson. Planar numerical cosmology ii. the difference equations and numerical tests. *Astrophys. J. Suppl. Ser.*, 54:229, 1984.

[14] S. Chandrasekhar. *The Mathematical Theory of Black Holes.* Oxford University Press, 1983.

[15] P. Colella. Glimm's method for gas dynamics. *SIAM J. Sci. Stat. Comput.*, 3:76–110, 1982.

[16] J.M. Corberán and M.L. Gascón. Tvd schemes for the calculation of flow in pipes of variable cross-section. *Mathl. Comput. Modelling*, 21:85–92, 1995.

[17] R. Donat and A. Marquina. Capturing shock reflections: an improved flux formula. *J. Comput. Phys.*, 125:42–58, 1996.

[18] B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjogreen. On Godunov type methods near low densities. *J. Comput. Phys.*, 92:273–295, 1991.

[19] K. Eppley. Evolution of time-symmetric gravitational waves: Initial data and apparent horizons. *Phys. Rev.*, D16:1609, 1977.

[20] R. Eulderink and G. Mellema. General relativistic hydrodynamics with a Roe solver. *Astron. Astrophys. Suppl. Ser.*, 110:587–623, 1995.

[21] J. A. Font, José Ma. Ibáñez, A. Marquina, and J. Ma. Martí. Multidimensional relativistic hydrodynamics: characteristic fields and modern high-resolution shock-capturing schemes. *Astron. Astrophys*, 282:304–314, 1994.

[22] José A. Font. Numerical hydrodynamics in general relativity. Living Reviews in Relativity, Vol. 3, 2000.

[23] José A. Font and José Ma. Ibáñez. A numerical study of relativistic bondi-hoyle accretion onto a moving black hole: Axisymmetric computations in a schwarzschild background. *Astrophys. J.*, 494:297–316, 1998.

[24] José A. Font, José Ma. Martí, José Ma. Ibáñez, and Juan A. Miralles. Numerical simulations of 2d potential flows in general relativity. *Computer Physics Communications*, 75:31–46, 1993.

[25] J. Foster and J. D. Nightingale. *A Short Course in General Relativity.* Springer-Verlag, 1995.

[26] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Springer-Verlag, 1996.

[27] S. K. Godunov. Finite difference methods for numerical computations of discontinuous solutions of equations of fluid dynamics. *Mat. Sb.*, 47:271, 1959.

[28] J. M. Greenberg, A. Y. LeRoux, R. Baraille, and A. Noussair. Analysis and approximation of conservation laws with source terms. *SIAM J. Numer. Anal.*, 34:1980–2007, 1997.

[29] S. G. Hahn and R. W. Lindquist. The two-body problem in geometrodynamics. *Ann. Phys.*, 29:304–331, 1964.

[30] S.W. Hawking and G.F.R. Ellis. *The large scale structure of space-time*. Cambridge University Press, 1973.

[31] J. F. Hawley, L. L. Smarr, and J. R. Wilson. A numerical study of nonspherical black hole accretion. ii. finite differencing and code calibration. *Astrophys. J. Suppl. Ser.*, 55:211, 1984.

[32] J. F. Hawley and J. R. Wilson. A numerical study of nonspherical black-hole accretion: Equations and test problems. *Astrophys. J. Supp.*, 55:211–246, 1984.

[33] J. F. Hawley and J. R. Wilson. A numerical study of nonspherical black-hole accretion: Finite differencing and code calibration. *Astrophys. J.*, 277:296–311, 1984.

[34] C. Helzel, R. J. LeVeque, and G. Warnecke. A modified fractional step method for the accurate approximation of detonation waves. *SIAM J. Sci. Comput.*, 22:1489–1510, 2000.

[35] Jr. J. W. York. Role of conformal three-geometry in the dynamics of gravitation. *Phys. Rev. Lett.*, 28:1082–1085, 1972.

[36] Jr. J. W. York. *Sources of Gravitational Radiation, edited by L. Smarr*. Cambridge, Cambridge, 1979.

[37] P. Jenny and B. Müller. Rankine-hugoniot-riemann solver considering source terms and multidimensional effects. *J. Comp. Phys.*, 145:575–610, 1998.

[38] S. S. Komissarov. A godunov-type scheme for relativistic magnetohydrodynamics. *Mon. Not. R. Astron. Soc.*, 303:343–366, 1999.

[39] Jan Olav Langseth and Randall J. LeVeque. A wave propagation method for three-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 165:126–166, 2000.

[40] John M. Lee. *Riemannian Manifolds*. Springer-Verlag, 1997.

[41] R. J. LeVeque. CLAWPACK software. available on the Web at the URL `http://www.amath.washington.edu/~claw`.

[42] R. J. LeVeque. High resolution finite volume methods on arbitrary grids via wave propagation. *J. Comput. Phys.*, 78:36–63, 1988.

[43] R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser-Verlag, 1990.

[44] R. J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33:627–665, 1996.

[45] R. J. LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. *J. Comput. Phys.*, 131:327–353, 1997.

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

[47] R. J. LeVeque and D. S. Bale. Wave-propagation methods for conservation laws with source terms. *International series of numerical mathematics*, 130:609–618, 1999.

[48] R. J. LeVeque, D. Mihalas, E. A. Dorfi, and E. Müller. *Computational Methods for Astrophysical Fluid Flow.* Springer-Verlag, 1998.

[49] Randall J. LeVeque. Balancing source terms and flux gradients in high-resolution godunov methods: The quasi-steady wave-propagation algorithm. *J. Comput. Phys.*, 146:346–365, 1998.

[50] R.J. LeVeque and J.A. Rossmanith. A wave propagation algorithm for the solution of pdes on the surface of a sphere. In *to appear in the Proceedings of Eighth International Conference on Hyperbolic Problems*, 2001.

[51] David Lovelock and Hanno Rund. *Tensors, Differential Forms, and Variational Principles.* Dover, 1989.

[52] A. Marquina, J. Ma. Martí, J. Ma. Ibáñez, J. A. Miralles, and R. Donat. Ultrarelativistic hydrodynamics: high-resolution shock-capturing methods. *Astron. Astrophys.*, 258:566–571, 1992.

[53] J. M. Martí, E. Müller, J. A. Font, J. M. Ibáñez, and A. Marquina. Morphology and dynamics of relativistic jets. *ApJ*, 479:151, 1997.

[54] José Ma. Martí, José Ma. Ibáñez, and Juan A. Miralles. Numerical relativistic hydrodynamics: Local characteristic approach. *Phys. Rev. D*, 43:3794–3801, 1991.

[55] José Ma. Martí and Ewald Müller. The analytical solution of the riemann problem in relativistic hydrodynamics. *J. Fluid Mech.*, 258:317–333, 1994.

[56] José Ma. Martí and Ewald Müller. Extension of the piecewise parabolic method to one-dimensional relativistic hydrodynamics. *J. Comput. Phys.*, 126:1–14, 1996.

[57] José Ma. Martí and Ewald Müller. Numerical hydrodynamics in special relativity. Living Reviews in Relativity, Vol. 2 (1999), 1999.

[58] T. Matsuda and H. Sato. Hydrodynamical behavior of gas spheres with masses of $10^4\, m_\odot$ to $10^{20}\, m_\odot$. *Prog. Theor. Phys.*, 41:1021, 1969.

[59] M. M. May and R. H. White. Hydrodynamic calculations of general-relativistic collapse. *Phys. Rev.*, 141:1232–1241, 1966.

[60] M. M. May and R. H. White. Stellar dynamics and gravitational collapse. *Methods. Comput. Phys.*, 7:219–258, 1967.

[61] C.W. Misner, K.S. Thorne, and J.A. Wheeler. *Gravitation.* W.H. Freeman and Company, 1973.

[62] M. Nakahara. *Geometry, Topology, and Physics.* IOP Publishing Ltd, 1990.

[63] T. Nakamura and H. Sato. General relativistic collapse of rotating supermassive stars. *Prog. Theor. Phys.*, 66:2038–2051, 1981.

[64] T. Nakamura and H. Sato. General relativistic collapse of non-rotating, axisymmetric stars. *Prog. Theor. Phys.*, 67:1396–1405, 1982.

[65] Philippos Papadopoulos and José A. Font. Analysis of relativistic hydrodynamics in conservation form. submitted, 2000.

[66] L. I. Petrich, S. L. Shapiro, and S. A. Teukolsky. Accretion onto a moving black hole: An exact solution. *Phys. Rev. Lett.*, 60:1781–1784, 1988.

[67] José A. Pons, José A. Font, José Ma. Ibáñez, José Ma. Martí, and Juan A. Miralles. General relativistic hydrodynamics with special relativistic riemann solvers. *Astron. Astrophys.*, 339:638–642, 1998.

[68] José A. Pons, José Ma. Martí, and Ewald Müller. The exact solution of the riemann problem with non-zero tangential velocities in relativistic hydrodynamics. *J. Fluid Mech.*, 422:125–139, 2000.

[69] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[70] P. L. Roe. Upwind differencing schemes for hyperbolic conservation laws with source terms. Nonlinear Hyperbolic Problems, C. Carraso, P. A. Raviart, and D. Serre, eds., Springer-Verlag, Lecture Notes in Mathematics 1270, 1986.

[71] Bernard F. Schutz. *A first Course in General Relativity.* Cambridge University Press, Cambridge, 1990.

[72] S. L. Shapiro and S. A. Teukolsky. Relativistic stellar dynamics on the computer. i. motivation and numerical method. *Astrophys. J.*, 298:34–57, 1985.

[73] S. L. Shapiro and S. A. Teukolsky. *Dynamical Spacetimes and Numerical Relativity.* Cambridge University Press, Cambridge, 1986.

[74] Joel Smoller. *Shock Waves and Reaction-Diffusion Equations.* Springer-Verlag, 1983.

[75] John Stewart. *Advanced general relativity.* Cambridge University Press, 1991.

[76] Kevin W. Thompson. The special relativistic shock tube. *J. Fluid Mech.*, 171:365–375, 1986.

[77] K. Thorne. Gravitational waves. Proceedings of the Snowmass 95 Summer Study on Particle and Nuclear Astrophysics and Cosmology, 1996.

[78] M. Vinokur. Conservation equations of gasdynamics in curvilinear coordinate systems. *J. Comput. Phys.*, 14:105–125, 1974.

[79] Robert M. Wald. *General Relativity.* The University of Chicago Press, 1984.

[80] G. Watson, D. H. Peregrine, and E. F. Toro. Numerical solution of the shallow-water equations on a beach using the weighted average flux method. Computational Fluid Dynamics '92, C. Hirsch et al., eds., Elsevier Science Publishers, 1992.

[81] J. R. Wilson. A numerical study of gravitational stellar collapse. *Astrophys. J.*, 163:209, 1971.

[82] J. R. Wilson. Numerical study of fluid flow in a kerr space. *Astrophys. J.*, 173:431, 1972.

[83] P. R. Woodward and P. Colella. The piecewise parabolic method (ppm) for gas-dynamical simulations. *J. Comput. Phys.*, 54:174, 1984.

Appendix A

APPROXIMATE RIEMANN SOLVERS

There are many different ways to approximate the flux across the cell interface over a time step Δt. The literature is rich with successful numerical flux functions, e.g., see [5], [17], [69]. In this appendix, we briefly discuss the implementation of three numerical flux functions in the wave propagation context. The first flux function that we discuss is an exact Godunov flux function. Second, Marquina's flux function is described. Though this flux function is more naturally applied in a flux vector splitting, we describe its wave propagation implementation. Finally, a flux function based on an arithmetic averaging of the states on either side of the interface is described. This flux function dates to the first use of f-waves.

A.1 A Riemann solver based on Godunov's flux function

Most of the Riemann problems that were discussed in this thesis were based on the use of f-waves. It was shown in Chapter 3 that the algorithms we developed were second order for smooth solutions and flux functions if approximate Jacobian matrices used to solve the linearized Riemann problems satisfied the properties (3.6) and (3.7). However, in general, it is not necessary to linearize the Riemann problems for the standard wave propagation algorithm. Instead, one can seek an exact solution to the Riemann problem, and use this information to generate the fluctuations, wave strengths, and wave speeds necessary for the standard algorithm.

Figure A.1 shows the Hugoniot loci, and integral curves for two states, q_l, and q_r, of a relativistic fluid. Notice that the left state has a higher initial pressure than the right state. For the initial data shown in the figure, the right state will be connected to the intermediate state, q^*_{sr}, by a shock, and the left state will be connected to the intermediate state by a rarefaction wave. The intermediate state, or solution to the Riemann problem, can be seen in the figure to have a pressure of about $p \approx 2.7$, and a velocity of $v = 0.4\,c$. The dashed lines in the figure represent the classical solution to the Riemann problem. Since the pressure jump is so large, the classical solution has a velocity larger than the speed of light, which violates Einstein's second principal. For more information about exact solutions to Riemann problems, the reader is referred to [46].

Once the exact solution to the Riemann problem is known, the numerical flux is simply the true flux along the interface,

$$\boldsymbol{F}_{i-1/2} = \boldsymbol{f}(q^*_{\mathrm{sr}}). \tag{A.1}$$

The fluctuations can then be computed according to equations (2.33) and (2.34) as discussed in Section 2.3.1. Since waves in the standard algorithm carry jumps in $\boldsymbol{q}$, wave strengths can be computed directly using the left and right states together with the intermediate state. Wave speeds are determined by jump conditions for shock waves, and characteristic speeds

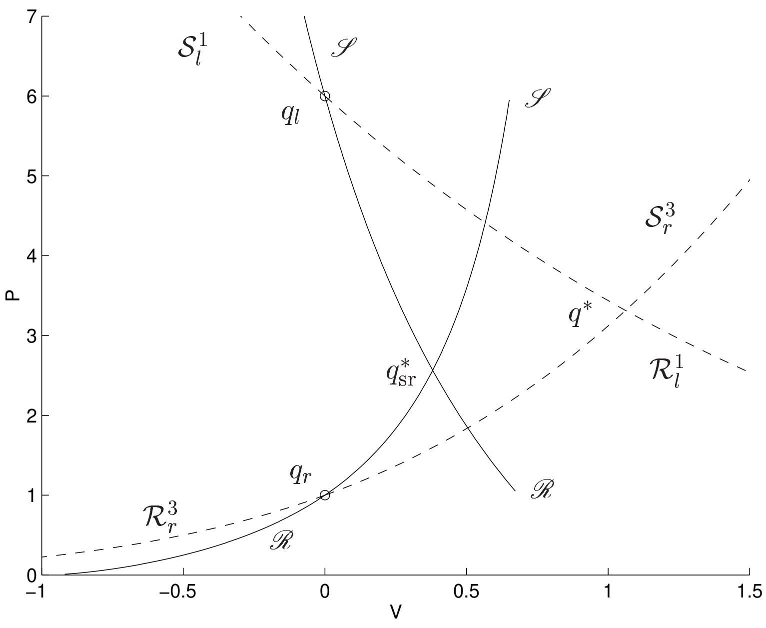


Figure A.1: Hugoniot curves intersecting for left and right states that produce super-luminal speeds for the classical equations.

for rarefaction waves. Details for the exact solution of Riemann problems for the equations of special relativistic hydrodynamics can be found in [55] and [68].

A.2 A Riemann solver based on the Marquina flux function

The Marquina flux function is based on a flux vector splitting within each cell. The Jacobian matrix, $\partial \boldsymbol{f}(\boldsymbol{q})/\partial \boldsymbol{q} \in \mathbb{R}^{n_{\mathrm{eq}} \times n_{\mathrm{eq}}}$, is decomposed into its eigenvalues $\{\lambda^p(\boldsymbol{q})\}$ and normalized right eigenvectors $\{r^p(\boldsymbol{q})\}$, $p = 1, 2, ..., n_{eq}$. On each time slice, a left state $\boldsymbol{Q}_{i-1}$, and right state, $\boldsymbol{Q}_i$ is define at the i^{th} cell interface. These states and the flux function evaluated at these states are decomposed into characteristic variables

$$\boldsymbol{Q}_{i-1} = \sum_p \omega_{i-1}^p \, r^p(\boldsymbol{Q}_{i-1}) \qquad \boldsymbol{Q}_i = \sum_p \omega_i^p \, r^p(\boldsymbol{Q}_i) \tag{A.2}$$

$$\boldsymbol{f}(\boldsymbol{Q}_{i-1}) = \sum_p \phi_{i-1}^p \, r^p(\boldsymbol{Q}_{i-1}) \qquad \boldsymbol{f}(\boldsymbol{Q}_i) = \sum_p \phi_i^p \, r^p(\boldsymbol{Q}_i). \tag{A.3}$$

The numerical flux function is then computed in the following way: For each $k = 1, 2, ..., n_{eq}$, set

$$\delta_k = \mathrm{sign}(\lambda^k(\boldsymbol{Q}_{i-1})), \tag{A.4}$$

$$\sigma_k = \mathrm{sign}(\lambda^k(\boldsymbol{Q}_i)) \cdot \delta_k. \tag{A.5}$$

Notice that $\sigma_k = \pm 1$ depending on whether $\lambda^k(\boldsymbol{q})$ changes sign for any $\boldsymbol{q} \in [\boldsymbol{Q}_{i-1}, \boldsymbol{Q}_i]$. Similarly $\delta_k = \pm 1$ depending on the sign of $\lambda^k(\boldsymbol{Q}_{i-1})$. Utilizing these definitions, the characteristic numerical fluxes, $\phi^k_\pm$, are written

$$\phi^k_+ = \phi^k_{i-1}\delta^+_k\sigma^+_k - \frac{1}{2}\left(\phi^k_{i-1} + \alpha_k\omega^k_{i-1}\right)\sigma^-_k\,, \tag{A.6}$$

$$\phi^k_- = -\phi^k_i\delta^-_k\sigma^+_k - \frac{1}{2}\left(\phi^k_i - \alpha_k\omega^k_i\right)\sigma^-_k\,, \tag{A.7}$$

where $\alpha_k \equiv \underset{\boldsymbol{q}\in\Gamma}{\mathrm{Max}}|\lambda^k(\boldsymbol{q})|$ and $\Gamma \to \Gamma(\boldsymbol{Q}_{i-1}, \boldsymbol{Q}_i)$ is a curve in phase space connecting $\boldsymbol{Q}_{i-1}$ and $\boldsymbol{Q}_i$. The familiar notation $a^+ = \mathrm{Max}(a,0)$ and $a^- = \mathrm{Min}(a,0)$ has been used for both $\sigma^\pm_k$ and $\delta^\pm_k$. Finally, the Marquina flux function is written

$$\boldsymbol{F}^n_{i-1/2} = \sum_{p=1}^{n_{eq}}\left(\phi^p_+\, r^p(\boldsymbol{Q}_{i-1}) + \phi^p_-\, r^p(\boldsymbol{Q}_i)\right). \tag{A.8}$$

A Riemann solver based on Marquina's flux function will not allow entropy violating solutions. This is due to the fact that if λ^k changes sign across an interface, the numerical flux function is similar to a diffusive Lax-Friedrichs numerical flux.

Fluctuation splitting of Marquina's flux

Using the numerical flux function $(A.8)$, the fluctuation differences are defined as

$$\begin{aligned}\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2} &= \boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{F}^n_{i-1/2}\\ &= \sum_{p=1}^{n_{eq}}\left(-\phi^p_+\, r^p(\boldsymbol{Q}_{i-1}) + \left(\phi^p_i - \phi^p_-\right) r^p(\boldsymbol{Q}_i)\right)\end{aligned} \tag{A.9}$$

$$\begin{aligned}\boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i-1/2} &= \boldsymbol{F}^n_{i-1/2} - \boldsymbol{f}(\boldsymbol{Q}_{i-1})\\ &= \sum_{p=1}^{n_{eq}}\left(\left(\phi^p_+ - \phi^p_{i-1}\right) r^p(\boldsymbol{Q}_{i-1}) + \phi^p_-\, r(\boldsymbol{Q}_i)\right).\end{aligned} \tag{A.10}$$

The decomposition of $\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1}$ into the set $\{\boldsymbol{\mathcal{W}}^p_{i-1/2}\}$ is still needed for the high-resolution wave propagation method. The most natural definition of these waves comes from the characteristic variables already found in each cell. That is,

$$\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1} = \sum_p\left(\omega^p_i\, r^p(\boldsymbol{Q}_i) - \omega^p_{i-1}\, r^p(\boldsymbol{Q}_{i-1})\right), \tag{A.11}$$

so a natural choice for $\boldsymbol{\mathcal{W}}^p_{i-1/2}$ is

$$\boldsymbol{\mathcal{W}}^p_{i-1/2} = \omega^p_i\, r^p(\boldsymbol{Q}_i) - \omega^p_{i-1}\, r^p(\boldsymbol{Q}_{i-1}). \tag{A.12}$$

One problem that arises from this definition when using an algorithm based on wave propagation, is that there is no clear speed associated with these waves. The arithmetic average

$$\bar{\lambda}^p = \frac{1}{2}\left(\lambda^p(\boldsymbol{Q}_{i-1}) + \lambda^p(\boldsymbol{Q}_i)\right), \tag{A.13}$$

is taken and seems to work for most problems.

A.3 A Riemann solver based on arithmetic averaging

The flux function introduced here utilizes desirable properties from both Roe's flux function and the HLLE flux function developed in [18]. A Riemann solver was developed in [18] that increases the wave speeds generated by a Roe flux function. This has the effect of increasing the numerical diffusion of the scheme.

Associated with the Jacobian matrix $\bar{A}(\boldsymbol{q}) \equiv \partial \boldsymbol{f}(\boldsymbol{q})/\partial \boldsymbol{q}$, are the set of eigenvalues $\{\lambda^p(\boldsymbol{q})\}$, and eigenvectors $\{r^p(\boldsymbol{q})\}$, $p = 1, 2, ..., n_{eq}$. On each time slice the average state, $\bar{\boldsymbol{Q}} = \frac{1}{2}(\boldsymbol{Q}_{i-1} + \boldsymbol{Q}_i)$,is defined at the i^{th} cell interface. Both the jump in conserved variables and the flux difference are decomposed into characteristic variables based on this arithmetic average so that

$$\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1} = \sum_{p=1}^{n_{eq}} \alpha_p \, r^p(\bar{\boldsymbol{Q}}) \tag{A.14}$$

$$\boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}) = \sum_{p=1}^{n_{eq}} \beta_p \, r^p(\bar{\boldsymbol{Q}}). \tag{A.15}$$

Let $\alpha_p \, r^p(\bar{\boldsymbol{Q}}) = \boldsymbol{\mathcal{W}}^p$ so that $\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1} = \sum_{p=1}^{n_{eq}} \boldsymbol{\mathcal{W}}^p$ since this is needed for the extension of the first order scheme to a high resolution method.

One immediate problem with using $\bar{\boldsymbol{Q}}$, rather than the Roe average $\hat{\mathbf{Q}}$, is that it is not true, in general, that $\bar{A}(\bar{\boldsymbol{Q}})\left(\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1}\right) = \boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1})$. To see this note that

$$\begin{aligned}
\bar{A}(\bar{\boldsymbol{Q}})\left(\boldsymbol{Q}_i - \boldsymbol{Q}_{i-1}\right) &= \sum_p \alpha_p \lambda^p(\bar{\boldsymbol{Q}}) \, r^p(\bar{\boldsymbol{Q}}) && \text{(A.16)}\\
&= \sum_p \lambda^p(\bar{\boldsymbol{Q}}) \boldsymbol{\mathcal{W}}^p && \text{(A.17)}\\
&\neq \sum_p \beta_p \, r^p(\bar{\boldsymbol{Q}}) && \text{(A.18)}\\
&= \boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}). && \text{(A.19)}
\end{aligned}$$

It is clear that the solution to the linearized Riemann problem based on $\bar{\boldsymbol{Q}}$ will result in a scheme for which the fluctuation splitting will not in general be consistent with the true flux difference. This was discussed in Section 2.3.1.

One way to assure that the fluctuation splitting will be consistent is to think of decomposing the flux difference into the set of f-waves $\{\boldsymbol{\mathcal{Z}}^p\}$. That is,

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}) &= \sum_p \beta_p \, r^p(\bar{\boldsymbol{Q}}) && \text{(A.20)}\\
&= \sum_p \nu_p \boldsymbol{\mathcal{W}}^p && \text{(A.21)}\\
&= \sum_p \boldsymbol{\mathcal{Z}}^p, && \text{(A.22)}
\end{aligned}$$

where we have defined $\nu_p \equiv \beta_p/\alpha_p$.

Before writing down the flux function, once again we define

$$\sigma_k = \text{sign}(\lambda^k(\boldsymbol{Q}_i) \cdot \lambda^k(\boldsymbol{Q}_{i-1})). \tag{A.23}$$

With the previous definition, a Roe-type numerical flux function can be defined as

$$\boldsymbol{F}^n_{i-1/2} = \frac{1}{2}\left(\boldsymbol{f}(\boldsymbol{Q}_{i-1}) + \boldsymbol{f}(\boldsymbol{Q}_i) - \sum_p \xi_p \boldsymbol{\mathcal{W}}^p_{i-1/2}\right), \tag{A.24}$$

where $\xi_p = |\nu_p|\sigma_p^+ - \max\left(|\lambda^p(\mathbf{Q}_{i-1})|, |\lambda^p(\boldsymbol{Q}_i)|\right)\sigma_p^-$. The effect of ξ_p is to increase the wave speed of any characteristic family whose eigenvalues change sign across the interface. As pointed out in [18], this will add numerical diffusion to the scheme.

A.3.1 Fluctuation splitting

The fluctuation splitting for the numerical flux function based on the arithmetic mean is quite straight forward. Using equations (2.33) and (2.34), the fluctuation differences can be written

$$\begin{aligned}
\boldsymbol{\mathcal{A}}^+\boldsymbol{\Delta Q}_{i-1/2} &= \frac{1}{2}\left(\boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}) + \sum_p \xi_p \boldsymbol{\mathcal{W}}^p_{i-1/2}\right) \\
&= \sum_p \frac{1}{2}\left(\nu_p + \xi_p\right)\boldsymbol{\mathcal{W}}^p_{i-1/2} \\
&= \sum_p \frac{1}{2}\left(\nu_p + |\nu_p|\sigma_p^+ - \max\left(|\lambda^p(\boldsymbol{Q}_{i-1})|, |\lambda^p(\boldsymbol{Q}_i)|\right)\sigma_p^-\right)\boldsymbol{\mathcal{W}}^p_{i-1/2}
\end{aligned} \tag{A.25}$$

$$\begin{aligned}
\boldsymbol{\mathcal{A}}^-\boldsymbol{\Delta Q}_{i-1/2} &= \frac{1}{2}\left(\boldsymbol{f}(\boldsymbol{Q}_i) - \boldsymbol{f}(\boldsymbol{Q}_{i-1}) - \sum_p \xi_p \boldsymbol{\mathcal{W}}^p_{i-1/2}\right) \\
&= \sum_p \frac{1}{2}\left(\nu_p - \xi_p\right)\boldsymbol{\mathcal{W}}^p_{i-1/2} \\
&= \sum_p \frac{1}{2}\left(\nu_p - |\nu_p|\sigma_p^+ + \max\left(|\lambda^p(\boldsymbol{Q}_{i-1})|, |\lambda^p(\boldsymbol{Q}_i)|\right)\sigma_p^-\right)\boldsymbol{\mathcal{W}}^p_{i-1/2}
\end{aligned} \tag{A.26}$$

Notice that if λ^p does not change sign across the interface, then the wave speed is simply $\nu_p^{\pm}$, and it is true that

$$\boldsymbol{\mathcal{A}}^{\pm}\boldsymbol{\Delta Q}_{i-1/2} = \sum_p \nu_p^{\pm}\boldsymbol{\mathcal{W}}^p_{i-1/2},$$

as discussed in [45]. That is, the p^{th} wave has a single speed associated with it. However, if λ^p does change sign, then the wave speeds are increased in absolute value as one would hope. This means that each wave no longer has a single speed associated with it and equation (A.27) no longer holds. In this case the wave speed is taken to be the average of the speeds and this is ν_p.

Appendix B

EULER EQUATIONS FOR A CLASSICAL FLUID

An inviscid, non-heat conducting fluid in n_d-dimensional motion is governed by the local conservation of mass density ρ, momentum density $\vec{s} \equiv \rho\,\vec{u}$, and energy density e. The geometric form of these equations, independent of any coordinate basis can be written

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \vec{s}\\ e\end{pmatrix} + \vec{\nabla}\cdot\begin{pmatrix}\rho\,\vec{u}\\ \overset{\leftrightarrow}{T}\\ (e+p)\,\vec{u}\end{pmatrix} = \boldsymbol{\psi}(\vec{x},\boldsymbol{q}), \tag{B.1}$$

where the momentum tensor is

$$\overset{\leftrightarrow}{T} = \rho\,\vec{u}\otimes\vec{u} + \overset{\leftrightarrow}{h}\,p, \tag{B.2}$$

and $\boldsymbol{\psi}$ is a generic source term that may represent gravity or geometry, for example. If the source is generated by a Newtonian gravitational force, then the source term in the system $(B.1)$ has the form

$$\boldsymbol{\psi} = \left(0, -\rho\,\vec{\nabla}\phi, -\vec{s}\cdot\vec{\nabla}\phi\right)^T, \tag{B.3}$$

where the potential, ϕ, satisfies the Poisson equation $\boldsymbol{\Delta}\phi = 4\pi G\rho$, and G is the universal gravitational constant. If, on the other hand, the geometry of a manifold generates the source, then $\boldsymbol{\psi}$ takes the form found in equation (5.88) so that

$$\boldsymbol{\psi} = \left(0, \overset{\leftrightarrow}{T} : \vec{\nabla}\vec{e}_{(k)}, 0\right)^T, \tag{B.4}$$

where the double contraction of indices of two tensors $\overset{\leftrightarrow}{T}$ and $\overset{\leftrightarrow}{G}$ has been denoted by $\overset{\leftrightarrow}{T} : \overset{\leftrightarrow}{G} = T^{ij}\,G_{ij}$. In cartesian coordinates, these balance laws take the form

$$\rho_{,t} + \left(\rho\,u^j\right)_{,j} = \psi^1 \tag{B.5}$$

$$s^i_{,t} + \left(\rho u^i\,u^j + p\,\delta^{ij}\right)_{,j} = \psi^{1+i} \tag{B.6}$$

$$e_{,t} + \left[(e+p)\,u^j\right]_{,j} = \psi^{n_d+2}, \tag{B.7}$$

where, as before, a subscripted comma denotes a partial derivative (i.e. $f_{,j} \equiv \partial f/\partial x^j$), and we have made use of the Einstein summation convention in which repeated indices are to be summed over the set $\{1,\cdots,n_d\}$. We will work under the assumption that the fluid is polytropic so that the internal fluid energy is a function of temperature alone, $\epsilon = C_v\,T$. For smooth flows, the entropy will remain constant and the pressure is a power law of the density, $p \propto \rho^\gamma$, where γ is a constant called the polytropic index. The equation of state for a perfect fluid then closes this system of balance laws and takes the form

$$p = (\gamma - 1)\epsilon. \tag{B.8}$$

Using this functional form of the pressure of a perfect fluid, the energy density can written

$$
\begin{aligned}
e &= \epsilon + \frac{1}{2}\rho\,\vec{u}\cdot\vec{u} \\
&= \frac{p}{\gamma-1} + \frac{1}{2}\rho\,\vec{u}\cdot\vec{u}.
\end{aligned} \tag{B.9}
$$

We define a vector of *conserved* quantities $\boldsymbol{q}$, with components $\boldsymbol{q} = (\rho, s^i, e)^T \in \mathbb{R}^{n_{eq}}$, where $n_{eq} \equiv n_d + 2$. The system $(B.1)$ can be formulated as a system of conservation laws

$$
\boldsymbol{q}_{,t} + \boldsymbol{f}^{(i)}(\boldsymbol{q})_{,i} = \boldsymbol{\psi}, \tag{B.10}
$$

where the flux $\boldsymbol{f}^{(i)} : \mathbb{R}^{n_{eq}} \to \mathbb{R}^{n_{eq}}$ is defined by

$$
\boldsymbol{f}^{(i)} = \overset{\leftrightarrow}{T}(\tilde{e}^{(i)}, \cdot) = \left(s^i, u^i\, s^j + p\,\delta^{ij}, (e+p)\,u^i\right)^T. \tag{B.11}
$$

Using the flux $\boldsymbol{f}^{(i)}$ and the vector $\boldsymbol{q}$, the flux Jacobian $A_i \equiv \partial \boldsymbol{f}^{(i)}/\partial \boldsymbol{q}$ is formed. It is the eigen-decomposition of the matrices A_i that are used in upwind, high resolution Godunov-type schemes.

We present the decomposition in the x-direction along the $\vec{e}_{(1)}$ basis vector. The decomposition in other coordinate directions can be found by simple permutations. The flux Jacobian in the x-direction has the form

$$
A_1 = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
\frac{1}{2}\tilde{\gamma}\theta - (u^1)^2 & (3-\gamma)u^1 & u^1 - \tilde{\gamma}u^2 & u^1 - \tilde{\gamma}u^3 & \tilde{\gamma} \\
-u^1\,u^2 & 0 & 0 & 0 & 0 \\
-u^1\,u^3 & 0 & 0 & 0 & 0 \\
\frac{1}{2}\tilde{\gamma}\theta u^1 - \frac{e+p}{\rho}\,u^1 & \frac{e+p}{\rho}\,u^1 + \frac{1}{2}\tilde{\gamma}(u^1)^2 & \tilde{\gamma}\,u^1\,u^2 & \tilde{\gamma}\,u^1\,u^3 & \gamma\,u^1
\end{pmatrix}, \tag{B.12}
$$

where we have defined $\theta \equiv \vec{u}\cdot\vec{u}$ and $\tilde{\gamma} \equiv \gamma - 1$. We decomponse A_1 into eigenvectors and eigenvalues such that $A_1 = R\Lambda R^{-1}$, and

$$
R = \left(r^1 \middle| \cdots \middle| r^{n_{eq}}\right), \quad \Lambda = \mathrm{diag}(\lambda^1, \cdots, \lambda^{n_{eq}}), \quad R^{-1} = \begin{pmatrix} l^1 \\ - \\ \vdots \\ - \\ l^{n_{eq}} \end{pmatrix}. \tag{B.13}
$$

The right eigenvectors, r^p, and left eigenvectors, l^p, satisfy the relationship $l^p \cdot r^p = \delta^{pq}$, and we order the eigenvalues such that $\lambda^1 \leq \lambda^2 \leq \cdots \leq \lambda^{n_{eq}}$. In three spatial dimensions, the set of eigenvalues for A_1 is

$$
\{\lambda^1, \cdots, \lambda^5\} = \{u^1 - c, u^1, u^1, u^1, u^1 + c\}, \tag{B.14}
$$

where the speed of sound is $c = \sqrt{\gamma\, p/\rho} > 0$. The right eigenvectors can be written

$$\begin{aligned}
r^1 &= \left(1, u^1 - c, u^2, u^3, \frac{e+p}{\rho} - u^1\, c\right)^T \\
r^2 &= \left(0, 0, 1, 0, u^2\right)^T \\
r^3 &= \left(1, u^1, u^2, u^3, \frac{\theta}{2}\right)^T \\
r^4 &= \left(0, 0, 0, 1, u^3\right)^T \\
r^5 &= \left(1, u^1 + c, u^2, u^3, \frac{e+p}{\rho} + u^1\, c\right)^T,
\end{aligned} \tag{B.15}$$

and the left eigenvectors have the form

$$\begin{aligned}
l^1 &= \left(\frac{\tilde{\gamma}\theta/2 + c\, u^1}{2c^2}, -\frac{\tilde{\gamma}\, u^1 + c}{2c^2}, -\frac{\tilde{\gamma}\, u^2}{2c^2}, -\frac{\tilde{\gamma}\, u^3}{2c^2}, \frac{\tilde{\gamma}}{2c^2}\right) \\
l^2 &= \left(-u^2, 0, 1, 0, 0\right) \\
l^3 &= \left(1 - \frac{\tilde{\gamma}\,\theta}{2c^2}, \frac{\tilde{\gamma}\, u^1 + c}{c^2}, \frac{\tilde{\gamma}\, u^2}{c^2}, \frac{\tilde{\gamma}\, u^3}{c^2}, -\frac{\tilde{\gamma}}{2c^2}\right) \\
l^4 &= \left(-u^3, 0, 0, 1, 0\right) \\
l^5 &= \left(\frac{\tilde{\gamma}\theta/2 - cu^1}{2c^2}, -\frac{\tilde{\gamma}\, u^1 - c}{2c^2}, -\frac{\tilde{\gamma}\, u^2}{2c^2}, -\frac{\tilde{\gamma}\, u^3}{2c^2}, \frac{\tilde{\gamma}}{2c^2}\right).
\end{aligned} \tag{B.16}$$

It is easy to verify that $l^p \cdot r^q = \delta^{pq}$ for the previous sets of left and right eigenvectors of A_1.

VITA

Name:	Derek S. Bale
Date of Birth:	March 23, 1971
Place of Birth:	Spokane, Washington, USA
Parents:	Dr. Richard H. Bale and Robin M. Bale

Education

Ph.D., Applied Mathematics, Winter 2002
University of Washington, Seattle, WA, USA

B.S., Engineering Physics, Spring 1994
University of Colorado, Boulder, CO, USA

B.S., Applied Mathematics, Spring 1994
University of Colorado, Boulder, CO, USA