Toward a Simple, Accurate Lagrangian Hydrocode

by

Tyler B. Lung

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2015

Doctoral Committee:

Professor Philip L. Roe, Chair
Assistant Professor Karthik Duraisamy
Associate Professor Krzysztof J. Fidkowski
Professor William R. Martin

"For my thoughts are not your thoughts, neither are your ways my ways," declares the Lord. "As the heavens are higher than the earth, so are my ways higher than your ways and my thoughts than your thoughts." - Isaiah 55:8-9

ACKNOWLEDGEMENTS

I have encountered many wonderful people during my graduate studies and they have made large contributions to my personal and professional development. To begin, I want to thank Professor Roe for agreeing to take me on as a student and for serving as a superb teacher, a patient mentor, a bright colleague, and a good friend over the past years. I have learned many things, been offered many opportunities, and traveled to many places that would not have been possible without him. I will always be grateful for his investment in me. I am also indebted to the other members of my dissertation committee, Karthik Duraisamy, Chris Fidkowski, and Bill Martin for all of the useful feedback and constructive criticism they have provided.

Many individuals at Los Alamos National Laboratory deserve my thanks. Don Burton originally recruited Professor Roe to work on Lagrangian hydrodynamics, who in turn recruited me, and he has provided much insight and support since. Scott Runnels and Nathaniel Morgan have each spent large amounts of their time and talent working with me through the Computational Physics Student Summer Workshop. Their mentoring has contributed immensely to this project and their hospitality contributed to two excellent summers at Los Alamos. Rob Lowrie also provided helpful critiques and encouragement, which were very welcome.

Andy Barlow from the Atomic Weapons Establishment in the United Kingdom has provided significant feedback on this work. His expertise and commentary was invaluable. He was also instrumental in securing travel funding for me to visit the Centre for Scientific Computing at the University of Cambridge and attend a con-

ference in Barcelona, Spain. Additionally, my graduate studies would not have been possible without the financial support I received at various times from Los Alamos National Laboratory, a Dobbins Fellowship, and the Department of Defense High Performance Computing Modernization Program via a National Defense Science and Engineering Graduate Fellowship.

My fellow students here at the University of Michigan have made graduate school much more enjoyable than it would have otherwise been. Kyle Ding, Brad Maeng, Rohan Morajkar, and Kevin Neitzel made the process of preparing for Preliminary Examinations bearable and I have learned much from each of them. I also want to thank Johann Dahm, Steve Kast, Loc Khieu, and Yimin Lou for many interesting discussions.

Nothing that I have achieved to this point in my life would have been possible without my parents, Darrin and Jill Lung. I am thankful beyond words for their teaching, love, and encouragement. The support I have received from the rest of my family has also been a blessing: Thank you Greg, Denise, Zack, Jessica, Caitlin, Kurt, Alyssa, Katelyn, Noelle, and Bethany. Finally, I want to thank my beautiful wife Miranda. I imagine that it is not easy being married to a Ph.D. student and I am grateful for her patience and love over the past four years. This thesis would not have been completed without her support.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

AF active flux

ALE arbitrary Lagrangian-Eulerian

CCH cell-centered hydrodynamics

CFD computational fluid dynamics

CFL Courant-Friedrichs-Lewy

DG discontinuous Galerkin

ENO essentially non-oscillatory

FCT flux-corrected transport

FEM finite element method

FUP First-order Upwind

FV finite volume

GCL geometric conservation law

LANL Los Alamos National Laboratory

LF Lax-Friedrichs

LHS left hand side

LPE low phase error

LW Lax-Wendroff

MOOD multi-dimensional optimal order detection

MUSCL monotone upstream-centered scheme for conservation laws

MUSCL-H MUSCL-Hancock

RHS right hand side

RR Rotated Richtmyer

SGH staggered-grid hydrodynamics

SLaM simple Lagrangian method

TVD total variation diminishing

VPFCT vorticity preserving flux-corrected transport

VPFV vorticity preserving finite volume

VPLW vorticity preserving Lax-Wendroff

WENO weighted essentially non-oscillatory

ABSTRACT

Toward a Simple, Accurate Lagrangian Hydrocode

by

Tyler B. Lung

Chair: Philip L. Roe

Lagrangian hydrocodes play an important role in the computation of transient, compressible, multi-material flows. This research was aimed at developing a simply constructed cell-centered Lagrangian method for the Euler equations that respects multidimensional physics while achieving second-order accuracy. Algorithms that can account for the multidimensional physics associated with acoustic wave propagation and vorticity transport are needed in order to increase accuracy and prevent mesh imprinting. Many of the building blocks of traditional finite volume schemes, such as Riemann solvers and spatial gradient limiters, have their foundations in one-dimensional ideas and so were not used here. Instead, multidimensional point estimates of the fluxes were computed with a Lax-Wendroff type procedure and then nonlinearly modified using a temporal flux limiting mechanism.

The linear acoustic equations were used as a simplified test environment for the Lagrangian Euler system. Here Lax-Wendroff methods that exactly preserve vorticity were investigated and found to resist mesh imprinting. However, the dispersion properties of the schemes were poor and so third-order accurate vorticity preserving methods were developed to remedy the problem. The third-order methods guided the

construction of a temporal limiting mechanism, which was then used in a vorticity preserving flux-corrected transport scheme. While the acoustic work was interesting in its own right, it also proved to be a useful stepping stone to Lagrangian hydrodynamics. The acoustics algorithms were extended to produce the Simple Lagrangian Method (SLaM). Standard test problems have shown that a first-order accurate version of the method is able to resist mesh imprinting and spurious vorticity despite its minimalistic structure. SLaM is capable of second-order accuracy with a simple parameter change and some preliminary work was done to extend the temporal flux limiting ideas from acoustics to the Lagrangian case. The limited SLaM method converges at second-order for smooth data and is able to capture shocks without producing large unphysical oscillations.

CHAPTER I

Introduction

The vast majority of computational fluid dynamics (CFD) methods in common use, especially in the world of aerospace engineering, make use of Eulerian meshes. In an Eulerian mesh, the computational points where the solution is approximated are fixed in space. The prevalence of Eulerian methods is easily explained given the nature of the fluid flows commonly encountered in aerospace applications and, more generally, most flows of engineering interest. In these flows, fluid elements are advected over large distances and undergo severe deformations. A Lagrangian mesh that moves with the fluid would, therefore, distort and tangle over time making computations difficult.

There is, however, a class of problems that lends itself to the use of Lagrangian meshes. In general, these problems involve transient compressible flows which are highly energetic, brief, and often involve multiple materials. Examples include the simulation of explosions and hyper-velocity deformations of solids (e.g. projectile-armor interactions). In the context of this specialized problem set, the flows occur over exceptionally short time scales. This means that a Lagrangian mesh must only be advected over relatively short distances, making the approach more viable. There are two primary advantages related to Lagrangian meshes that make them not just viable, but preferable for these problems. First, the governing equations for compressible flow

simplify a great deal in the Lagrangian frame since the nonlinear advective terms in the governing equations are hidden. Second, a Lagrangian mesh prevents the numerical mixing of different materials in multi-material problems, thereby avoiding equation of state modeling difficulties and smeared contact discontinuities.

Lagrangian numerical methods aimed at solving the types of flows mentioned above trace their origins to the Manhattan Project at Los Alamos National Laboratory (LANL) in the 1940s. Of course, it was the quest for nuclear weapons that drove their development. In particular, it was necessary to understand the implosion of a nuclear weapon in which high explosives generate strong shocks to compress nuclear material to its supercritical density. During such an implosion, the enormous temperatures and pressures generated by detonating the high explosives cause metals and other solids to deform like fluids. As a result, the experiments used to understand the implosion process were referred to as hydrotests [33]. During this period, the computational work of von Neumann and Richtmyer [88] at LANL allowed one-dimensional shocks representative of those encountered in an implosion to be computed numerically in Lagrangian coordinates. Their work birthed a class of methods that are referred to as Lagrangian hydrodynamic codes, or Lagrangian hydrocodes.

1.1 Foundations of Lagrangian Hydrocodes

Since the Manhattan Project, Lagrangian hydrocodes have been the focus of intense, though often isolated, research efforts due to the role they play in solving problems of importance to national security. Traditionally there are two main classes of hydrocodes: staggered-grid and cell-centered. staggered-grid hydrodynamics (SGH) methods are descendants of the one-dimensional method invented by von Neumann and Richtmyer and have historically been responsible for the bulk of practical hydrodynamics computations. These methods are characterized by a stencil that employs staggered storage, locating the thermodynamic variables (density, internal energy,

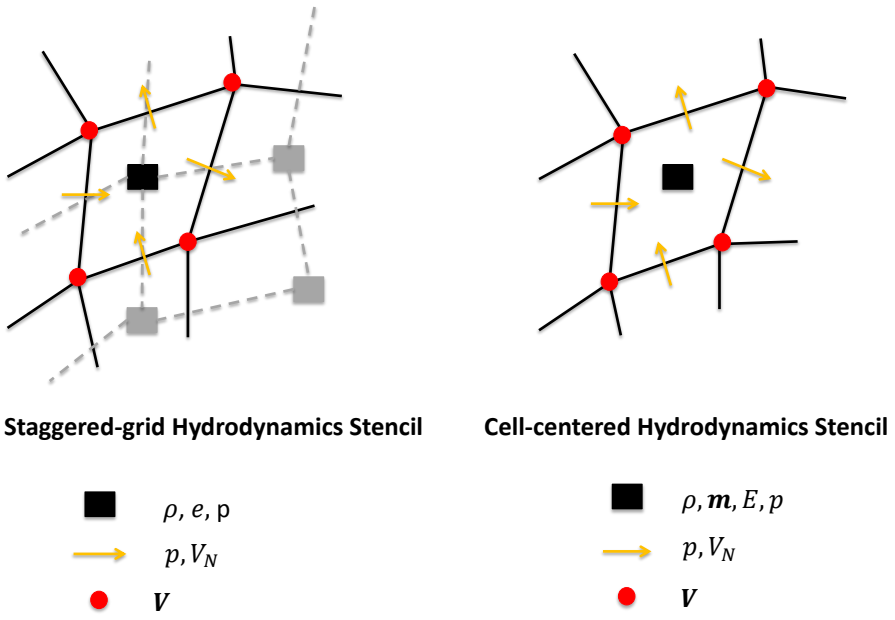

Figure 1.1: Stencils representative of those used for SGH methods (left) and CCH methods (right) are shown.

pressure) at the center of cells and the kinematic variables (momentum) at cell vertices on a dual mesh. In contrast, cell-centered hydrodynamics (CCH) methods store the conserved variables (density, momentum, total energy) at cell centers. While the genesis of cell-centered codes can be attributed to the work of Godunov during the late 1950s [36], this author also believes that Peter Lax and Burton Wendroff deserve mention. The original Lax-Wendroff paper [47] solved the one-dimensional hydrodynamics equations in terms of the conserved variables using Lagrangian coordinates. Figure 1.1 shows conceptual depictions of the computational stencils used by two-dimensional SGH and CCH methods.

A major reason that SGH methods have been heavily used in the past is that the staggered storage allows for the mesh motion to be determined in a straight forward manner. The momentum variables define the mesh motion and they are easily updated by applying Newton's second law on the staggered mesh. For example, in the

case of gas dynamics, the force on the staggered control volumes is obtained from the pressures stored in the adjacent cells. On the other hand, a major drawback of the staggered storage is that the resulting methods are not naturally conservative. This arises from fact that the mass and momentum variables are not co-located and the total energy is not updated directly. It is possible to derive conservative staggered schemes, which are referred to as compatible, however a careful discretization procedure is required. In contrast, CCH methods are attractive because they are naturally conservative in the same way as any other finite volume scheme. Time and space averaged fluxes are computed at cell boundaries to update the conserved variables. Traditionally, the fluxes are obtained from Riemann solvers that have been modified to account for moving geometry.

Riemann solvers compute, either exactly or approximately, the one-dimensional unsteady waves that would form at a cell boundary if the states defined by the cell-centered data were allowed to interact. This type of initial value problem is known as Riemann's problem in gas dynamics. In general, a Riemann solution consists of the four states shown in Figure 1.2: the initial left and right states and then two intermediate ones created by the waves. A Riemann solver determines which state the current face lies in and then computes the fluxes from it. Godunov solved the Riemann problems at cell faces exactly on Eulerian meshes, inventing what is now simply referred to as Godunov's method [35]. Its success led to the development of many approximate Riemann solvers that were designed to give acceptable solutions at reduced cost (see e.g. [72; 38; 68]). Schemes that use approximate Riemann solvers are called Godunov-type methods and they form the basis for most current Eulerian compressible flow codes.

The primary difficulty with the Godunov-type approach in multidimensional CCH is that there is no obvious way to determine the motion of mesh vertices from the one-dimensional Riemann velocities at faces. Early attempts at doing so led to al-

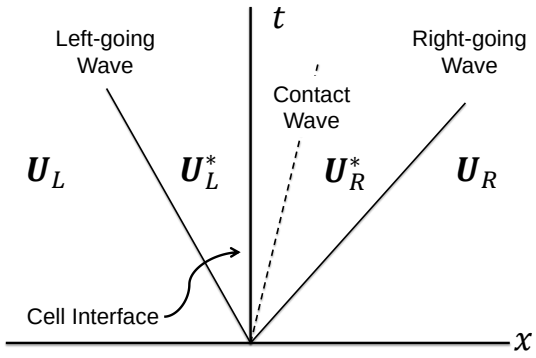

Figure 1.2: The Riemann problem from gas dynamics is an initial value problem in which the unsteady interaction of two uniform, discontinuous states is determined. Riemann solvers are used to compute face fluxes in Godunov-type methods.

gorithms that were prone to mesh instabilities and spurious vorticity production. Furthermore, whether one is interested in Lagrangian or Eulerian methods, the wisdom in constructing multidimensional methods with Riemann solvers may be called into question since they are intrinsically one-dimensional.

1.2 An Overview of Staggered-grid Hydrodynamics Methods

The method of von Neumann and Richtmyer has two important characteristics that are still present in SGH methods today. The first, of course, is the staggered storage described in Section 1.1. The second is the use of a nonlinear dissipation term which they termed artificial viscosity. This term was added to the pressure in the governing equations and was designed to add numerical dissipation near shocks, but vanish in smooth regions of the flow. This had the practical effect of smoothing out discontinuities and reducing spurious oscillations, which permitted successful shock-capturing. Another notable early work related to artificial viscosity is due to Landshoff [46], who added a linear term to von Neumann and Richtmyer's quadratic one. Often artificial viscosity is associated with enforcing physical waves (e.g. pro-

viding an entropy increase across shocks), providing numerical stability, and reducing spurious overshoots all at the same time. However, from a numerical perspective, these three issues are distinct. It is often confusing when they are not treated as such. Regardless, the concept and use of artificial viscosity is deeply embedded in the hydrodynamics literature, particularly in regard to SGH methods.

The seminal extension of the staggered-grid methodology to multiple space dimensions was done by Wilkins [90] in the early 1960s. Here, hydrodynamic flows in elastic-plastic solids were computed in two space dimensions on moving meshes. For the next few decades, SGH research was focused on building upon the existing staggered-grid framework instead of fundamentally changing it. For example, much emphasis was placed on devising improved artificial viscosities and extending the idea of artificial viscosities to multidimensional problems (see *e.g.* [44; 91; 7; 20]). Fundamental improvements to the staggered-grid framework did come about in the 1990s with the invention of the so-called compatible methods. These methods were fully conservative and constructed in terms of subzonal masses and pressures [12; 19; 21]. The compatible methods used the discrete conservation of energy equation and a corner-force-based sub-cell discretization to ensure that the discrete energy conversions between kinetic and internal energies (and vice versa) were done in a "compatible" manner. That is, any reduction/increase in kinetic energy was accompanied by the appropriate increase/decrease in internal energy. Compatible methods form the basis for modern SGH codes, which are still widely used to solve complex, multiphysics flows of engineering interest. Examples of more recent research related to SGH include [11], [54], and [64] which focus on the derivation of multidimensional artificial viscosities using Riemann-like solvers.

1.3 A Persistent Failing: Spurious Mesh Movement

While one of the primary benefits of utilizing Lagrangian meshes is that the nonlinear advective terms in the governing equations are hidden, the payment for this convenience is extracted during the task of moving the mesh. Since each mesh zone tracks a parcel of fluid, the mesh distorts and moves over time. In vortical flows or flows with with significant shear the mesh will undergo large distortions and tangle, eventually requiring the computation to be stopped and the mesh fixed. Of course, mesh tangling due to physical flow features is unavoidable. Unfortunately, this is not the sole, or even primary, concern. The mesh can tangle due to nonphysical motions, instabilities, or some combination of the two. Caramana and Shashkov describe these problems as the "bane of Lagrangian hydrodynamics calculations" [17]. Specifying the mesh motion in a simple, stable way that minimizes spurious motions is one of the central problems related to Lagrangian computations and a major focus of this work.

As previously noted, SGH codes have been heavily used because the motion of mesh nodes is easily obtained. However, simplicity does not imply accuracy. Indeed, SGH methods commonly suffer from spurious mesh tangling, mesh instabilities, and mesh imprinting when computing multidimensional problems. An example of spurious mesh motion is shown in Figure 1.3. Here, vertices in the initially square mesh should only experience radial motion; no vorticity should be present. Instead unphysical mesh motions have clearly occurred, especially near the origin, and badly formed high aspect ratio or nonconvex cells have resulted. The badly formed elements are indicative of past solution errors and a numerical solution that does possess the proper radial symmetry. Many techniques were developed to fix the spurious motions in SGH such as damping them with special artificial viscosities, removing them with filtering techniques, modifying stress-induced forces to match solution symmetries, or utilizing sub-zonal pressures and masses [34; 57; 10; 18; 17]. Despite these attempts,

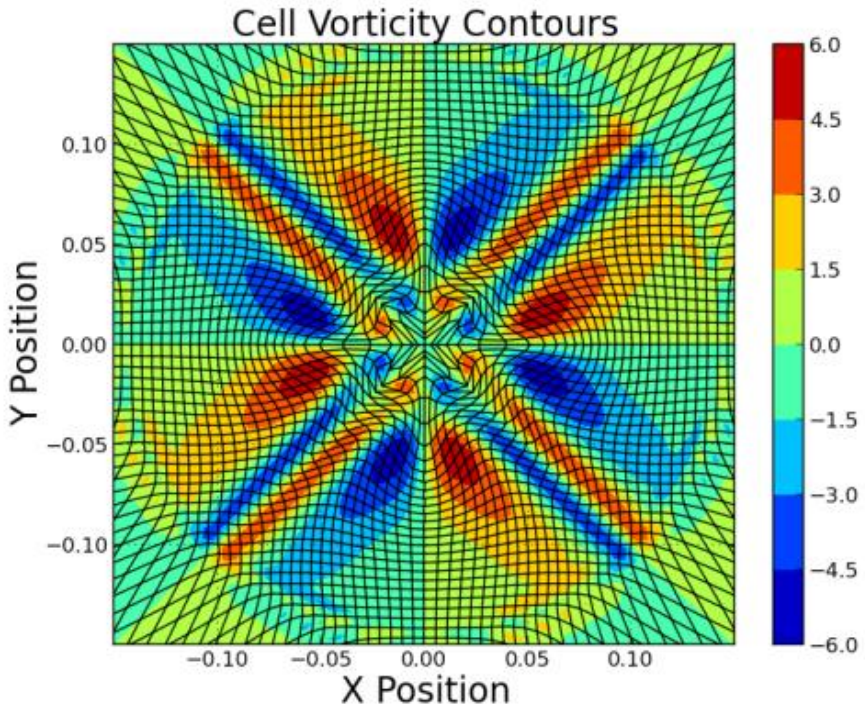

Figure 1.3: Mesh imprinting, which is accompanied by spurious vorticity, destroys a Lagrangian computation that should possess perfect radial symmetry.

a report from the Methods and Algorithms group at LANL in 2009 noted that there was no code available in the weapons complex that could correctly compute an idealized implosion without the use of "artificial symmetrizing constraints" [13]. Clearly, this represented a large concern. If the available hydrodynamic methods required *ad hoc* features that forced solutions to obey known symmetries, how could they be used to compute general problems with confidence?

In some instances, mesh imprinting problems can be reduced by using meshes that mimic known symmetries in the solution. For example, when computing a radially symmetric explosion in two spatial dimensions, one might employ a radial mesh. However, this approach fails in practice for two main reasons. First, there is no way to construct a discrete mesh in three dimensions that mimics spherical symmetry [13; 21]. Second, constructing a tailor-made mesh is only possible if a great deal is known about the solution *a priori* and this is obviously not the case for general

problems. The only way to truly solve these issues is to develop a hydrodynamics method that does not suffer from mesh imprinting and respects multidimensional physics without artificial fixes.

1.4 Recent Progress: Cell-centered Hydrodynamics Methods

The first CCH algorithms originated in Russia due to the work of Godunov and his co-workers during the late 1950s and they were used to solve multidimensional hydrodynamics problems [36]. These methods were, not surprisingly, constructed along the lines of what are now called Godunov-type finite volume schemes. To update the conserved variables, which were stored at cell centers, Riemann solvers were modified to give the fluxes across cell faces on a moving mesh. In the United States, multidimensional cell-centered algorithms were being worked on by the late 1970s and into the 1980s with the PISCES [7] and CAVEAT codes [1; 30]. An advantage of the CCH approach is that it naturally produces conservative methods, while one of the primary challenges is determining how to relate Riemann face velocities to the mesh nodes. The CAVEAT code used a least-squares fit of the face velocities around each node to move the mesh. However, serious problems with spurious mesh motions were encountered. The problems with the CAVEAT code are particularly well documented in [29] and [7]. The quandary of how to properly specify the mesh motion in traditional CCH methods inhibited their use and was another reason for the popularity of SGH methods.

During the last decade, a large shift toward CCH methods, particularly with respect to research efforts, has taken place. The impetus behind the shift was a desire to overcome the problems outlined in Section 1.3 and the success in doing so originally demonstrated for gas dynamics by Després and Mazeran [27] and then

Maire *et. al.* [60] with the cell-centered framework. The key idea proposed in these works was to shift the emphasis in cell-centered algorithms from computing face fluxes to computing corner fluxes. Node-based solvers were constructed that compute the fluxes through each cell corner and the adjacent node velocity in a single, consistent step. This approach was significant not only because it unambiguously specified the mesh motion, but also because it resulted in methods that automatically satisfy the geometric conservation law (GCL). Satisfying the GCL can be thought of as maintaining consistency between the volume change of a zone due to the mesh motion and the volume change of the same zone that is defined implicitly through the energy fluxes (which contain an estimate of the velocity divergence). Satisfying the GCL is important since violations can lead to mesh instabilities and methods that do not preserve a uniform flow field.

The works by Després and Mazeran and Maire *et. al.* have led to a fundamental shift in the way cell-centered methods are constructed and thought about. A brief description of Maire's original first-order accurate method will be given. Again, the key concept is the construction of a nodal solver which computes the corner fluxes and mesh motion simultaneously. To construct it, degrees of freedom were added to the standard cell-centered stencil by way of half-face pressures as shown in Figure 1.4. That is, each cell in the mesh was assumed to possess unique pressures along each half-face. For a quadrilateral element, this translates to the addition of eight new pressures. These pressures are independent of the cell-centered one and are not continuous at cell boundaries. Due to the lack of continuity, the conservative nature of the algorithm is initially lost (the fluxes for both momentum and total energy contain the pressure). To regain it, the conservation of momentum (and energy) was imposed in an integral sense around each vertex. An entropy constraint was then used to provide the additional information needed to write a fully determined system of equations at each node for the adjacent half face pressures and the node velocity.

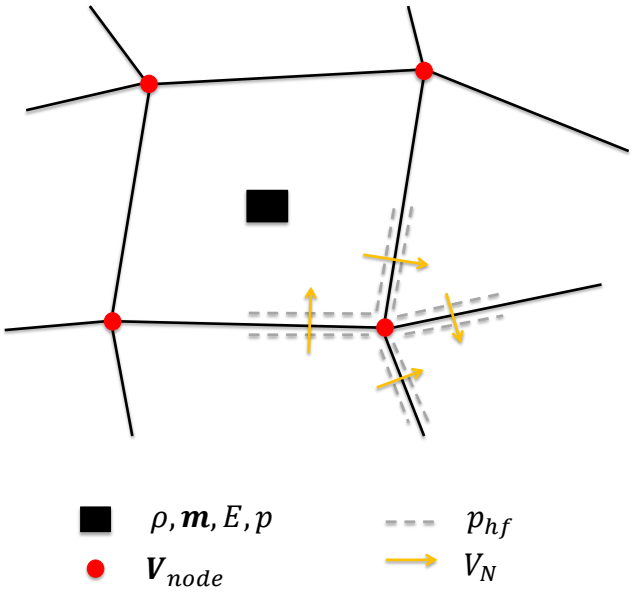

Figure 1.4: Shown here is a conceptual representation of Maire's CCH method. The nodal solver computes the half-face pressures (p_{hf}) and nodal velocity ($\mathbf{V}_{node}$) in a single step.

Then, by taking the velocity of each face to be the average of the adjacent node velocities, the mesh motion and the fluxes are known and the conserved variables can be updated by integrating the fluxes around the cell-centered control volumes. Interestingly, the resulting nodal velocity can be shown to be special fit of the adjacent one-dimensional half-face velocities predicted by an acoustic Riemann solver. For additional details, the reader is referred to the journal paper [60].

The corner-based cell-centered methods developed in France were much less prone to mesh imprinting, with no *ad hoc* fixes, than any previous methods. This success did not go unnoticed and triggered a flurry of research activity both in France and around the world. Després and Mazeran's scheme (GLACE) was extended to work on unstructured meshes with second-order accuracy [22] and solve hyperelasticity problems [41]. Furthermore, the weak consistency of the GLACE scheme was proved for unstructured meshes [25] and its mesh robustness was improved using sub-zonal

entropies in [26]. Similarly, the method of Maire *et. al.* (EUCCLHYD) was extended to achieve second-order accuracy, accommodate unstructured meshes, provide the Lagrangian capability in an arbitrary Lagrangian-Eulerian (ALE) framework, and solve problems with strength [58; 59; 61; 62]. Researchers at LANL developed a modified node-based solver and a method that could solve gas and solid dynamics problems on unstructured meshes with second-order accuracy [14; 16]. Barlow and Roe took a slightly different approach and proposed a first-order cell-centered algorithm that used a dual mesh to define the mesh motion [4] and then Barlow extended the method to be second-order accurate [3]. This approach relies heavily on Riemann solvers, though they are slightly modified to account for a vector velocity field.

1.5 Other Approaches

There are many other approaches being used to construct Lagrangian methods and some will be mentioned here briefly. A large effort, primarily at LANL, has focused on the development of point-centered Lagrangian methods (see e.g. [89; 65]). They are designed to work on unstructured tetrahedral meshes and are notable in that the mass flux between elements is not zero on a discrete mesh. The fluxes are computed via Riemann solvers at face midpoints. Since in an unstructured mesh there are usually many faces associated with each point, one-dimensional Riemann problems are solved in many directions, which helps reduce mesh imprinting. Other Lagrangian methods are derived from various Galerkin-type approaches. For example, Dobrev *et. al.* use the finite element method (FEM) [28]. Their approach allows for arbitrarily high orders of accuracy, can use curved elements, and recovers standard SGH methods in the low order limit. Others have utilized the discontinuous Galerkin (DG) framework to derive methods that compute the solution in Lagrangian coordinates. Li *et. al.* develop such a scheme, which they consider cell-centered. It uses a semi-Lagrangian form of the governing equations and weighted essentially non-oscillatory (WENO)

reconstructions to limit the spatial discretization [49]. Both essentially non-oscillatory (ENO) and WENO methodologies have been used within the finite volume frame work to produce Lagrangian and ALE methods with higher-order accuracy (see e.g. [23; 63]). The Lagrangian ENO method presented in [51] shares some commonality with the work presented in this thesis due to the use of Lax-Wendroff type time stepping.

1.6 A New Proposal

While significant progress has been made in regard to Lagrangian hydrocodes in the last decade, there is still plenty of room for further work. The project documented in this thesis was first proposed in 2011 by Philip Roe at the International Conference on Numerical Methods for Multi-material Fluid Flows [73]. Here Professor Roe proposed a new structure for a cell-centered Lagrangian algorithm that could solve the Euler equations. The algorithm structure was designed to produce a simple, accurate algorithm that respects multidimensional physics. There was a strong emphasis on simplicity of the method, as most of the applications where Lagrangian codes are used involve additional physics phenomena such as material strength or radiation transport. Therefore, it would be desirable if the hydrodynamics algorithm was as simple as possible to better facilitate the addition of other physics models. Of course, a simple algorithm would also be easier to implement and probably more efficient. In the view of Professor Roe and the author, simplicity is an attribute of current CCH methods that could be improved.

Perhaps more importantly, the new structure proposed by Professor Roe was designed to be intrinsically multidimensional. While the corner-based CCH methods are a step in the right direction, they still rely heavily on the use of Riemann solvers and, therefore one-dimensional physics. See [53] for a discussion. The motivation for a new method that properly accounts for multidimensional physics was to reduce

mesh imprinting. In particular, Roe sought a method that would obey the laws of vorticity transport, or require only a small correction procedure to eliminate non-physical vorticity production. The importance of vorticity control was illustrated in a paper by Dukowicz and Meltz that linked the presence of spurious vorticity with mesh imprinting and tangling [29]. They demonstrated that the solution to the Saltzman problem, a one-dimensional piston generated shock that is computed on a two-dimensional skewed mesh, was vastly improved by removing spurious vorticity. Roe noted that physical vorticity transport could be guaranteed at the numerical level for the simpler, but related, case of linear acoustics. There it was demonstrated that a Lax-Wendroff-type method, the Rotated Richtmyer scheme, could exactly preserve a discrete definition of the vorticity [66]. The key characteristic of the RR method that permitted this property was its formulation in terms of vertex fluxes. Now, the linear acoustic equations can be thought of as the linearized Lagrangian gas dynamic equations when written in system form. Due to the close relationship between acoustics and Lagrangian gas dynamics, Roe hypothesized that a similarly constructed hydrodynamic algorithm would tend to more closely, though not exactly, follow the laws of vorticity transport and suffer less from mesh imprinting.

The RR scheme was of interest not only due to the vorticity preservation property, but also because it has a Lagrangian friendly structure. When used to solve the acoustic equations, the RR scheme can be viewed as a linearized Lagrangian method. To better understand the connection, first consider the general update procedure used by the acoustic RR method in two space dimensions on a box mesh:

1. Using the four neighboring cell-centered values, $\mathbf{U} = (p, \mathbf{V})^T$, interpolate vertex fluxes, $\boldsymbol{f} = (a_0\mathbf{V}, a_0 p)^T$.

2. Evolve each vertex flux through one half time step

3. Update the cell-centered variables by integrating the vertex fluxes around a cell

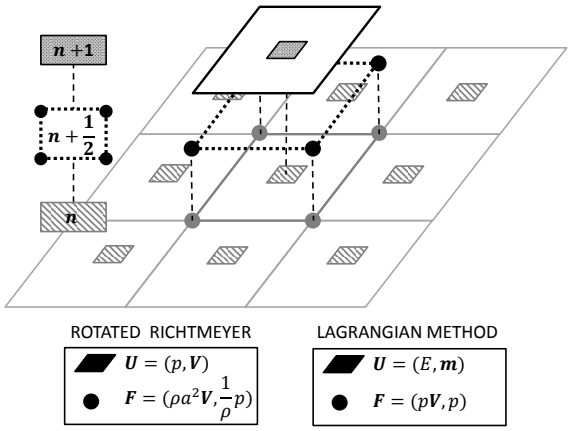

Figure 1.5: An illustration to reinforce the connection between the RR scheme and its Lagrangian analog. The primary difference in the case of a Lagrangian algorithm is the need to move the mesh. Note that unique nodal velocities can be defined from the nodal fluxes.

centered control volume with the Trapezium rule

A Lagrangian algorithm would look remarkably similar if we take the view that the Eulerian form of the equations will be solved on a mesh that moves with the local fluid velocity. This is to say that instead of working with the Lagrangian form of the gas dynamic equations, the advective terms will be subtracted from the Eulerian form. This view point leads to an algorithm that is nearly identical to RR. The acoustic variables are exchanged for momentum, $\mathbf{m}$, and total energy, E and the fluxes become $\boldsymbol{f} = (p, p\mathbf{V})^T$. A step can then be added between 2 and 3 in which the mesh is moved according to nodal velocities defined via the vertex fluxes. Figure 1.5 helps to illustrate the general structure of the acoustic RR method and the link between it and a Lagrangian algorithm for inviscid gas dynamics.

It should be stressed that the vertex fluxes under consideration here are pointwise, continuous estimations of the fluxes at vertices. The point estimates are eventually used in a quadrature rule to compute space-averaged face fluxes. Perhaps, then, it is preferable to think of the vertex fluxes as integration points. Note that they differ significantly from the corner fluxes discussed earlier which were composed of

half-face fluxes. Shifting from the traditional finite volume paradigm of computing fluxes through faces to computing point estimates of the fluxes has a number of consequences. First, the vertex fluxes cannot be computed by considering only the pairwise interaction of states. In other words, Riemann solvers, or related ideas, cannot be used. This abandonment of one-dimensional thinking also steers us away from standard limiting techniques which have their foundations in one-dimensional, scalar problems. Next, it is noted that since the vertex fluxes are shared between adjacent cells and used to compute a unique flux through each face, the resulting algorithm is naturally conservative in the same sense as any other finite volume scheme. The final advantage to the vertex fluxes is that they automatically define the mesh motion since the Lagrangian Euler fluxes contain estimates of the velocity, as will be shown in Chapter II. One of the central problems with CCH methods, determining how to move the mesh, never arises in the proposed framework.

It was mentioned in the last paragraph that the vertex flux formulation is not conducive to traditional limiters. Professor Roe proposed to use flux-corrected transport (FCT), originally developed by Boris and Book [8], as a general framework for limiting. It provides a multidimensional approach, provided that a multidimensional flux limiter can be devised. FCT methods can be broken down into three main steps: transport, diffusion, and antidiffusion. The transport and diffusion steps are often implemented simultaneously using a first-order method that guarantees no spurious features are admitted in the solution. Antidiffusive fluxes are then defined as the difference between the flux from some higher-order method and the first-order one. These are modified using a nonlinear limiter just enough to prevent new extrema from being created during the final antidiffusion step. In order for this procedure to work in the proposed context, a multidimensional vertex-centered flux limiter is needed.

In summary, Roe proposed to construct a simple, second-order accurate Lagrangian hydrocode with vertex fluxes, which automatically define the mesh motion,

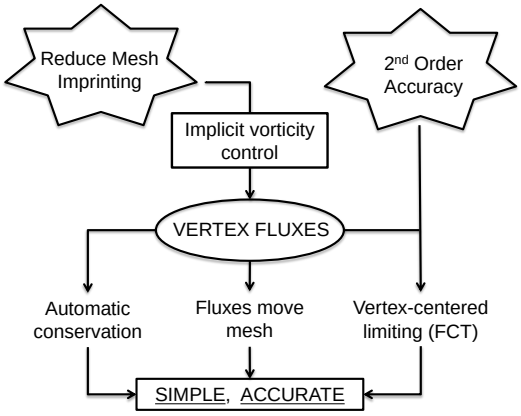

Figure 1.6: A graphical representation of Roe's proposed CCH method.

and FCT based limiting. The hydrocode stencil is cell-centered and results in a fully conservative algorithm. Great emphasis was placed on removing all dependence on one-dimensional physics from the method. The proposed algorithm structure is presented graphically in Figure 1.6. This thesis is devoted to exploring the merit and usefulness of this proposal.

1.7 Research Strategy

Development of a new and improved Lagrangian hydrodynamics algorithm as described in Section 1.6 from scratch is an ambitious goal. Instead of using the nonlinear Euler equations and moving grids as the starting point for this research project, a more gradual approach was adopted in which the two-dimensional acoustic equations and Lax-Wendroff (LW) type methods were used as a simplified test environment. Two central problems, or crises, were studied in the simpler environment that were sure to arise when solving the Euler equations on moving methods. The goal, of course, was to develop tools and knowledge that could be easily extended to the Lagrangian problem. Chapter III discusses vorticity preserving methods of the LW family that

are second-order accurate and are free of mesh imprinting when computing radially symmetric solutions on square meshes.

The first crisis was mesh imprinting, which is strongly related to problems with spurious mesh motion in Lagrangian computations. When computing acoustics problems on a square mesh, mesh imprinting is easily observed as a failure to preserve radial symmetry. Acoustic methods were sought which could properly preserve radially symmetric waves in this situation. The methods were derived by insisting on a discrete vorticity preservation property and by making the truncation errors in the numerical dispersion relations isotropic.

The second crisis considered for the acoustic equations was the limiting problem. As discussed, Professor Roe proposed to use the FCT framework to accomplish this and so work was focused on developing a multidimensional, vertex-centered flux limiter. Key areas of emphasis in the limiting work included the incorporation of physics into limiting decisions, multidimensional construction, and the avoidance of placing *a priori* bounds on the solution. This eventually led to a temporal approach. Chapter IV explores some vorticity preserving third-order methods that are cheaply obtained using a FCT update procedure. The increased accuracy is made possible by incorporating information from a first-order provisional solution in antidiffusive fluxes and eliminates most of the undesirable dispersion behavior commonly associated with second-order methods. In Chapter V, the temporal flux limiters are developed that mimic the behavior of the third-order methods and remove the any remaining spurious extrema.

While the acoustic work is interesting in its own right, it work proved to be very useful in the context of Lagrangian hydrodynamics. The first foray into the Euler equations and moving meshes involved a simple, first-order extension of the RR method. Even this simple extension was able to produce some promising numerical results and so effort was directed toward detail improvements to the base method.

No fundamental changes to the algorithm structure were needed to produce excellent results with little to no mesh imprinting. The first-order Lagrangian work is documented here in Chapters VI and VII. Finally, a the acoustic temporal limiting strategy was extended to construct a nonlinear Lagrangian method that is second-order accurate in regions of smooth flow and able to capture shock waves. The final method and numerical results are detailed in Chapter VIII. Some concluding remarks and discussion of future work are included in Chapter IX.

1.8 Broader Contributions

While the impetus for this thesis was Lagrangian hydrodynamics, there are significant contributions that have broader implications. The investigation of the acoustic RR method and other vorticity preserving Lax-Wendroff variants has led to a deeper understanding of the Lax-Wendroff family and shown that it has a worse reputation than deserved. If a Lax-Wendroff method is tailored to the governing equations at hand, they can be quite effective. Another area of contribution is limiting for multidimensional methods. The desire for truly multidimensional schemes is not unique to the Lagrangian community and there is strong interest for the same attribute in Eulerian algorithms. Major research efforts are currently being undertaken in an effort to rethink and reformulate the foundations of compressible flow algorithms. An example of one such effort is the active flux (AF) family of schemes [32]. These methods are designed from the ground up to solve the gas dynamics equations in a multidimensional framework that is third-order accurate, yet inexpensive enough for practical computations. No Riemann solvers are used. One of the major barriers to the widespread adoption of next-generation methods such as AF is a lack of appropriate limiting techniques. A temporal flux limiting approach is proposed in Chapter V for acoustics that shows promise. The limiter acts on pointwise estimates of the fluxes and uses temporal information to produce a compact, multidimensional mechanism.

The ideas were then extended to produce a second-order Lagrangian algorithm in Chapter VIII. The author hopes that some of these ideas, particularly in regard to limiting, will prove useful outside of the Lagrangian community.

CHAPTER II

Governing Equations

While many of the typical applications for Lagrangian methods involve simulating solid materials, the field of CFD is very relevant due to the violent nature of the problems. Events such as hyper-velocity impacts or explosions involve incredibly large material deformations and pressures. As a result, solids can flow like a fluid and shock physics play a central role in the problems. Furthermore, the governing equations for such flows are best cast in terms of stress, strain, and displacement *rates*. In some deformation regimes, the governing equations begin to take on a flavor that is reminiscent of compressible fluid mechanics. If one is primarily concerned with exploring numerical issues related to shock capturing and mesh motion, then the nonequilibrium effects due to viscosity may be neglected. Often the time scales are so short that there is insufficient time for the nonequilibrium effects to be important anyway. Due to these considerations, the Euler equations of gas dynamics have been used extensively to develop Lagrangian computational methods as evidenced by many of the works cited in Chapter I. In some sense, choosing to work with the Euler equations and ideal gases makes life very difficult despite the simple stress tensor (it only contains the pressure) and simple closure models (*e.g.* the ideal gas). The absence of any physical damping in the system and the nonlinear nature of the equations tends to be unforgiving. Numerical errors, particularly those related to

vorticity or entropy, tend to persist and accumulate once incurred.

There are two different approaches that can be taken when numerically solving the Euler equations in the Lagrangian frame. The first is to write the governing equations in terms of Lagrangian coordinates and variables. See the first chapter of Zel'dovich and Raizer [94] for more information. This approach was the first attempted (see e.g.[88; 47]) and is advantageous because the mesh, which is defined in terms of the Lagrangian coordinates, does not deform in time. However, to obtain the solution in Eulerian coordinates a mapping must be performed. Furthermore, it is often difficult to formulate multidimensional problems in terms of Lagrangian coordinates and singularities develop in the presence of strong rarefactions (i.e. near vacuum) which are difficult to treat numerically [36]. Due to these difficulties, most modern Lagrangian methods use Eulerian coordinates and solve the Eulerian equations on a mesh that moves with the fluid. Since each mesh element tracks a fixed parcel of fluid, the advective terms must be removed from the Eulerian fluxes. Solution changes due to dilatation are easily accounted for by tracking the volume change of each element. While no coordinate mapping difficulties are encountered, extreme mesh deformations must be dealt with. The moving mesh approach will be employed in this thesis to solve the unsteady Euler equations in two space dimensions.

The set of partial differential equations that describe the conservation of mass, momentum, and total energy for unsteady, inviscid, compressible flows are commonly referred to as the Euler equations and will be referred to as such here. Using vector notation, we have

$$\rho_t + \nabla \cdot (\rho \mathbf{V}) = 0 \tag{2.1a}$$

$$(\mathbf{M})_t + \nabla \cdot (\rho \mathbf{V} \otimes \mathbf{V}) + \nabla p = 0 \tag{2.1b}$$

$$E_t + \nabla \cdot [\mathbf{V}(E + p)] = 0, \tag{2.1c}$$

where $\mathbf{V} = (u, v, w)^T$ and p denote the fluid velocity and pressure, respectively. The equations are presented in the conservative differential form where the conserved variables ρ, $\mathbf{M}$, and E denote the mass, momentum, and total energy *densities*, respectively. The equations are written with respect to the three dimensional Eulerian spatial variables x, y and z. The subscript t denotes the partial derivative with respect to time. An equation of state is needed to close the system. In this thesis, only the ideal gas relation $p = (\gamma - 1)\rho e$, where γ is the ratio of specific heats and e is the specific internal energy, is considered, but no special difficulty would arise from a different choice. In this chapter some important properties and forms of the governing equations will be explored. Furthermore, vorticity transport will be reviewed and the relationship between the Lagrangian Euler equations and linear acoustics will be discussed to reinforce the point that the acoustic system provides a useful model problem.

2.1 The Euler Equations in the Lagrangian Frame

The equations (2.1) can be written down in a form familiar to aerospace engineers in terms of state and flux vectors. In two spatial dimensions, x and y, we have

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0, \text{ where} \tag{2.2a}$$

$$\mathbf{U} = (\rho, m, n, E)^T, \tag{2.2b}$$

$$\mathbf{F} = (m, um + p, un, u(E + p))^T \text{ and } \mathbf{G} = (n, vm, vn + p, v(E + p))^T. \tag{2.2c}$$

Here $m = \rho u$ and $n = \rho v$ for convenience. We need to write (2.2) in a frame that moves with the fluid. To illustrate one way to do this, consider the x-momentum equation

$$m_t + (um + p)_x + (vm)_y = 0. \tag{2.3}$$

Now differentiate the flux terms with respect to x and y to obtain

$$m_t + u_x m + u m_x + p_x + v_y m + v m_y = 0.$$

Recalling the substantial derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{V} \cdot \nabla, \tag{2.4}$$

which describes the time rate change of a quantity that moves with the fluid, (2.3) can be written as

$$\frac{Dm}{Dt} + m(\nabla \cdot \mathbf{V}) + p_x = 0. \tag{2.5}$$

The velocity divergence will be eliminated from this relation using the continuity equation, which is

$$\rho_t + m_x + n_y = 0.$$

For a Lagrangian fluid parcel, this becomes

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \mathbf{V}) = 0. \tag{2.6}$$

Now substitute $\rho = \mathcal{M}/\mathcal{V}$ into (2.6), where $\mathcal{M}$ is the mass of a differential Lagrangian fluid parcel and $\mathcal{V}$ is its volume. The mass, $\mathcal{M}$, is constant and after canceling we have

$$\frac{D1/\mathcal{V}}{Dt} + \frac{1}{\mathcal{V}}(\nabla \cdot \mathbf{V}) = 0. \tag{2.7}$$

Equation (2.7) can be further manipulated by noting that

$$\frac{D\mathcal{V}(1/\mathcal{V})}{Dt} = 0 = \frac{1}{\mathcal{V}}\frac{D\mathcal{V}}{Dt} + \mathcal{V}\frac{D1/\mathcal{V}}{Dt} \Rightarrow \frac{D1/\mathcal{V}}{Dt} = -\frac{1}{\mathcal{V}^2}\frac{D\mathcal{V}}{Dt},$$

and, after substitution into (2.7), the familiar expression

$$\frac{1}{\mathcal{V}}\frac{D\mathcal{V}}{Dt} = \nabla \cdot \mathbf{V} \tag{2.8}$$

is obtained. Plugging (2.8) into (2.5) gives

$$\frac{1}{\mathcal{V}}\frac{D\mathcal{V}m}{Dt} + p_x = 0. \tag{2.9}$$

The continuity, y-momentum, and energy equations can be treated in a similar manner, leading to the Euler system as it is often written when dealing with cell-centered algorithms:

$$\frac{1}{\mathcal{V}}\mathbf{U}_T + \mathbf{F}_x + \mathbf{G}_y = 0, \text{ where} \tag{2.10a}$$

$$\mathbf{U} = (\mathcal{V}\rho, \mathcal{V}m, \mathcal{V}n, \mathcal{V}E), \tag{2.10b}$$

$$\mathbf{F} = (0, p, 0, pu), \text{ and } \mathbf{G} = (0, 0, p, pv). \tag{2.10c}$$

The subscript T denotes the substantial derivative and describes the time rate change along a particle path. Note that, strictly speaking, the system is not in conservation form due to the volume term. Additionally, the state variables are the mass, momentum, and total energy as opposed to the densities of each.

2.2 Information Propagation

The unsteady, compressible Euler system is a hyperbolic set of partial differential equations. This means the solution at any point in the flow only depends on a finite part of the domain, which is usually referred to as the domain of dependence. It is imperative that the numerical methods used to solve hyperbolic equations have access to the information contained therein. Furthermore, a method should use as little

information outside of the domain of dependence as possible, as this information has no business affecting the numerical solution. In the CFD literature for compressible flows, the concept of using only relevant information in a numerical method, and ensuring that it is only transmitted in physical ways, is referred to as upwinding.

The best way to understand the flow of information in a hyperbolic system is to perform a characteristic analysis. When the one-dimensional Euler equations are subjected to such an analysis in a fixed frame, three characteristic speeds emerge. The acoustic signals propagate with the speeds $u \pm a$, where a is the isentropic sound speed, and entropy changes move with the fluid velocity, u. The acoustic propagation signals are asymmetric for nonzero flow velocities, which means that the domain of dependence is also asymmetric. This fact led to the popularity of Riemann solvers in Eulerian numerical methods, as they naturally incorporate upwinding and facilitate the proper flow of information.

If the Lagrangian frame is used to analyze the Euler equations then the acoustic signals propagate in a symmetric manner and travel outward from any point at the speed of sound, $\pm a$. This result stands in stark contrast to the Eulerian result and would seem to imply a purely Lagrangian method should be structured very differently than an Eulerian one. Indeed, it does seem strange that the original CCH methods used modified Riemann solvers in their formulations since the feature Riemann solvers were designed to add (biased information flow) is not required. In the Lagrangian case, a finite difference scheme with a completely symmetric appearance can be properly upwinded. Or, viewed from the other direction, upwind schemes for systems with symmetric wave speeds have a symmetrical appearance[1]. This opens the door for simple and multidimensional treatment of Lagrangian flows based on central difference methods, which is the approach pursued in this thesis.

[1]Consider, for example, the First-order Upwind (FUP) method for acoustics.

2.3 Vorticity Transport

The mechanisms by which vorticity evolves in a compressible, inviscid flow should be reviewed since vorticity evolution is a topic that will arise frequently in the chapters to follow. For such a flow, the vorticity transport equation can be written in vector form as

$$\frac{D\boldsymbol{\omega}}{Dt} = -\boldsymbol{\omega}\nabla\cdot\mathbf{V} + \boldsymbol{\omega}\cdot\nabla\mathbf{V} + \frac{1}{\rho^2}\nabla\rho\times\nabla p. \tag{2.11}$$

See Appendix A for a derivation. The first term on the right hand side (RHS) of the equation accounts for vortex intensification due to compressibility effects. The second, usually referred to as the vortex stretching term, accounts for the stretching of vortices due to gradients in the velocity field. The last term, which is referred to as the baroclinic term, describes the only means by which vorticity can be produced. It tells us that the vorticity will change if the density gradient and the pressure gradient are not parallel. Essentially, the pressure field is able to induce a net torque on a fluid element in this case. In a single material compressible flow, vorticity production can arise due to shocks with curved fronts or shocks of varying strength.

In the case of two dimensional flows, the vortex stretching term disappears since only velocity gradients which act parallel to a vortex tube can change its length. Then the vorticity transport equation reduces to

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega}\nabla\cdot\mathbf{V} + \frac{1}{\rho}\nabla\rho\times\nabla p. \tag{2.12}$$

This equation is not particularly frightening in its appearance. However, as was discussed in Chapter I, Lagrangian hydrocodes often cannot get the vorticity correct because its evolution is not strictly enforced. Even small vorticity errors will tend to accumulate and cause problems over time unless they are purposely damped out. Designing a Lagrangian algorithm that implicitly obeys this equation to the greatest extent possible is a primary goal of this work.

2.4 The Acoustic Equations

The linear acoustic equations describe the propagation of small, isentropic waves through a gas. They can be derived by linearizing the Euler equations about a stationary, uniform background state with the pressure, density, and sound speed denoted by p_0, ρ_0, and a_0, respectively. If small disturbances $p_0 + p'$, $\rho_0 + \rho'$, u', and v' are substituted into the Euler equations and the small terms are neglected, the acoustic system remains. Additional details will be omitted for the sake of brevity, but the interested reader can find more information in [50] or [94]. Here, we skip to the end and write down the equations:

$$p'_t + \rho_0 a_0^2 \nabla \cdot \mathbf{V}' = 0 \text{ and} \tag{2.13a}$$

$$\mathbf{V}'_t + \frac{1}{\rho_0} \nabla p' = 0. \tag{2.13b}$$

This system is easily written in conservation form. In two spatial dimensions it is

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0, \text{ where} \tag{2.14a}$$

$$\mathbf{U} = (p', u', v')^T, \tag{2.14b}$$

$$\mathbf{F} = (\rho_0 a_0^2 u', \rho_0^{-1} p', 0)^T, \text{ and } \mathbf{G} = (\rho_0 a_0^2 v', 0, \rho_0^{-1} p')^T. \tag{2.14c}$$

The vorticity transport law for acoustics can be quickly derived by taking the curl of the velocity evolution equation, which gives

$$(\nabla \times \mathbf{V})_t + \nabla \times \left(\frac{1}{\rho_0} \nabla p \right) = 0. \tag{2.15}$$

If the initial density is uniform everywhere and the vorticity is defined as $\boldsymbol{\omega} = \nabla \times \mathbf{V}$,

then

$$\boldsymbol{\omega}_t = -\frac{1}{\rho_0}\nabla \times \nabla p. \tag{2.16}$$

Since for any scalar function, ϕ, the identity $\nabla \times \nabla \phi = 0$ holds, the acoustic vorticity transport law is

$$\boldsymbol{\omega}_t = 0. \tag{2.17}$$

While this result is not very interesting from the point of view of the exact equations, most numerical methods fail to satisfy it at the discrete level. As such, numerical vorticity transport can be studied in this simple context, which is one of the main focuses of Chapter III.

2.4.1 Relationship with the Lagrangian Euler Equations

It is easy to show that the acoustics equations are hyperbolic partial differential equations and that the system has symmetric wave speeds ($\pm a_0$) like the Lagrangian Euler system. Therefore, both equation sets share key properties that affect the design of numerical solution techniques. Furthermore, vorticity transport, which is of keen interest for Lagrangian methods, can be studied in the simpler context of acoustic methods.

To further illustrate the relationship between the Lagrangian Euler equations and linear acoustics the primitive form of the Euler equations in the Lagrangian frame will be derived. The time evolution of the velocity variables are easily obtained from the momentum equations. Considering first the x-momentum equation (2.3), we expand it to give

$$\rho_t u + \rho u_t + \rho u u_x + u(\rho u)_x + p_x + \rho v u_y + u(\rho v)_y = 0. \tag{2.18}$$

Some slight manipulation yields

$$u_t + u u_x + v v_y + \frac{1}{\rho} p_x + \frac{u}{\rho}(\rho_t + (\rho u)_x + (\rho v)_y) = 0. \tag{2.19}$$

Right away the substantial derivative can be recognized. Also, note that the left side of the continuity equation has appeared, which is equal to zero. Therefore, (2.19) reduces to

$$u_T + \frac{1}{\rho} p_x = 0. \tag{2.20}$$

Similarly, the time evolution of v is given by

$$v_T + \frac{1}{\rho} p_y = 0. \tag{2.21}$$

Obtaining the pressure equation requires a bit more work. First, the total energy equation must be split in order to get the evolution equation for specific internal energy, which is

$$\rho \frac{De}{Dt} + p \nabla \cdot \mathbf{V} = 0. \tag{2.22}$$

The interested reader is referred to [50] for additional details on this step. This is then rewritten as

$$\frac{D\rho e}{Dt} - e \frac{D\rho}{Dt} + p \nabla \cdot \mathbf{V} = 0$$

and the continuity equation is used to give

$$\frac{D\rho e}{Dt} + (\rho e + p) \nabla \cdot \mathbf{V} = 0. \tag{2.23}$$

Now recall that the ideal gas equation of state is $\rho e = p/(\gamma - 1)$. After substituting this into (2.23) we have

$$\frac{D}{Dt} \frac{p}{\gamma - 1} + \frac{p\gamma}{\gamma - 1} \nabla \cdot \mathbf{V} = 0. \tag{2.24}$$

Simplification then yields

$$\frac{Dp}{Dt} + p\gamma \nabla \cdot \mathbf{V} = 0, \tag{2.25}$$

which can be recast by recalling that, for an ideal gas, the isentropic sound speed is $a^2 = \gamma p/\rho$. Finally, γ is eliminated leaving

$$p_T + \rho a^2 \nabla \cdot \mathbf{V} = 0. \tag{2.26}$$

Taken together, the governing equations in primitive form using vector notation are

$$p_T + \rho a^2 \nabla \cdot \mathbf{V} = 0 \text{ and} \tag{2.27a}$$

$$\mathbf{V}_T + \rho^{-1} \nabla p = 0. \tag{2.27b}$$

A system of equations that looks like linear acoustics has been recovered, although no assumption of linearity has been made.

It is important to note, however, that the nonlinearity is not the only difference. The system (2.27) is written in terms of substantial derivatives, which are hiding nonlinear advection terms. It would be a mistake to forget that these terms exist. On the other hand, there are no advection terms in the world of linear acoustics. Fortunately, this distinction is less important in the context of Lagrangian hydrodynamics since it is assumed that any necessary computations will take place at points moving with the local flow velocity. In this case, the advection terms do locally disappear in a sense and a Lagrangian method could be viewed in terms of solving many nonlinear acoustics problems on the mesh. The major caveat is that this viewpoint assumes that it is known how to properly move the mesh, which is not trivial. Nevertheless, it is clear that the linear acoustic equations provide a useful set of model equations if one desires to investigate a Lagrangian method that will solve the Euler equations. By considering linear acoustics, the difficulties associated with nonlinear equations and mesh movement are avoided. Furthermore, much more numerical analysis is possible. However, the linear equations are still intrinsically multidimensional since the changes in pressure are driven by the velocity divergence and a simple vorticity trans-

port law exists. These attributes permit the study of numerical problems related to mesh imprinting, vorticity transport, and limiting in a simpler context. Others have proposed to use the linear acoustic equations as a model problem for Lagrangian hydrodynamics; one such example is due to Bauer *et. al.* [6].

CHAPTER III

Second-order Methods for Acoustics

Chapter II provided justification for the use of linear acoustics as a model problem for Lagrangian hydrodynamics. This chapter begins the investigation of acoustic algorithms and focuses on reducing mesh imprinting. Mesh imprinting is the tendency of the mesh geometry to imprint itself on the solution, which often results in a failure to preserve symmetry. It is a primary concern for Lagrangian methods as it is related generally to spurious mesh motions. The goal here is to understand how to eliminate the problem in the simplified acoustic environment so that the knowledge can be later incorporated into a Lagrangian method. The family of two-dimensional, acoustic LW methods will be investigated in hopes of finding members that can accurately maintain circular solution symmetry on a square mesh. The full LW family, which uses a nine point stencil, has four free parameters and includes variants in which the fluxes are evaluated at either edges or vertices or both. The methods are constructed with central differences, and, therefore have a completely symmetric appearance. However, they can still be considered upwind schemes. This is possible because the acoustic system has equal wave speeds in all directions; a trait shared with the Lagrangian Euler equations. Therefore, despite appearances, the upwinding concept is not abandoned in this work. While only two spatial dimensions will be considered here, the extension of these methods to three dimensions should not cause great

difficulty.

The RR method [92] was the member of the LW family which was initially preferred because it exactly preserves vorticity and has a Lagrangian friendly structure, as was discussed in Chapter I. Recall that the method is formulated in terms of vertex fluxes and it has a simple three step update procedure in which initial point estimates of the fluxes are interpolated at vertices, evolved through one half time step, and then integrated around the cell boundaries using the trapezium rule. It may be preferable to think of the fluxes, then, as integration points. The pointwise nature of the fluxes eliminates the possibility of computing them from pairwise interactions. One-dimensional Riemann solutions will never make an appearance. This is a crucial attribute that prevents the incorporation of strictly one-dimensional physics into the method. However, the vertex flux formulation is conservative in the same sense as any other finite volume scheme since the point fluxes are shared between adjacent cells and eventually averaged over each face via the integration rule.

In his investigation of Lax-Wendroff variants, Turkel recommended against using the RR variant "because of large phase errors"[82]. However, his analysis related only to the case of a scalar problem, or, equivalently, a problem with commuting matrices, and in that case his concerns were warranted. The optimum parameters for a LW scheme do depend on the problem being solved, which mean his results are not universally applicable. That being said, the standard dispersion problem common to second-order accurate methods where undamped modes are propagated with the incorrect phase speed will be encountered and eventually dealt with in Chapter IV. This chapter is solely focused on making the errors isotropic.

Some notation and test problems are introduced in Section 3.1. Next, Section 3.2 demonstrates that enforcing physical vorticity evolution successfully reduces mesh imprinting by comparing numerical results from the RR method to results from the original two-dimensional LW scheme. In Section 3.3, additional improvements to RR

are sought via a dispersion analysis. A concrete link between vorticity preservation and isotropic numerical behavior is established and methods with less mesh imprinting than the RR method are derived. Some discussion will conclude the chapter in Section 3.4.

3.1 Notation and Test Problems

Here it will be convenient to work with the acoustic system

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0, \tag{3.1}$$

in the dimensionless form where $\mathbf{U} = (\frac{p'}{\rho_0 a_0^2}, \frac{u'}{a_0}, \frac{v'}{a_0})^T \equiv (p, u, v)^T$, $\mathbf{F} = (a_0 u, a_0 p, 0)^T$, and $\mathbf{G} = (a_0 v, 0, a_0 p)^T$. As in Chapter II, the variables p', u', and v' represent perturbations of the fluid properties from a uniform, stationary background state given by ρ_0, p_0, and a_0. The primary test problem utilized in the acoustic test environment was a discontinuous, radially symmetric pressure perturbation applied to a fluid at rest. The initial conditions were defined on the domain $x \in [-6, 6]$ and $y \in [-6, 6]$ and set as

$$\begin{aligned} &p = 2 \text{ if } r \leq 1, && p = 0 \text{ if } r > 1, \\ &u = 0, && v = 0, \text{ and} \\ &a_0 = 0. \end{aligned}$$

A smooth problem was also considered in which the discontinuous pressure distribution was replaced by the Gaussian profile

$$p(x, y, 0) = 2 \exp[-(x^2 + y^2)]. \tag{3.2}$$

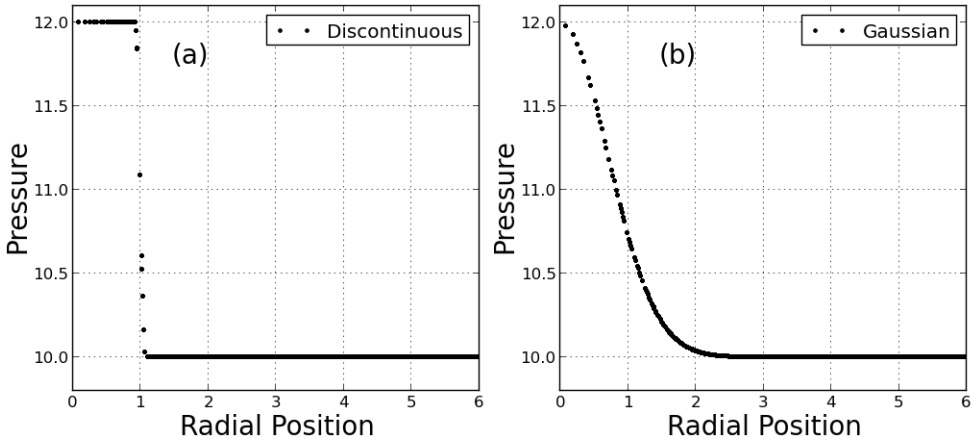

Figure 3.1: The initial pressure distributions for the discontinuous test problem (a) and smooth test problem (b) are shown.

The problems were solved on a 100×100 mesh unless otherwise noted. See Figure 3.1 for plots of the initial pressure distributions. These test problems were useful for a number of reasons. First, the solutions should have perfect radial symmetry and any deviation from this is easily observed by plotting the data as a function of radius[1]. Second, the resulting system of waves includes a pressure expansion that implodes at the origin and an outgoing compression wave. The shape of the waves changes over time. Third, the vorticity is zero initially, and therefore should remain zero for all time. This makes detecting spurious vorticity especially straightforward. Finally, in the case of the discontinuous problem, the ability of numerical methods to handle high frequency data is tested.

Some explanation of notation is needed before proceeding. Standard central differencing and averaging operators will be used throughout. They are denoted by δ and μ, respectively, and the result of an operator is located half way between the

[1]Judging the scatter in radial plots is a more stringent way to assess radial symmetry preservation than judging the circular appearance of contour lines.

input values. The definition for each operator is

$$\mu_x()_{j,k} = 0.5[()_{j+1/2,k} + ()_{j-1/2,k}], \qquad \delta_x()_{j,k} = [()_{j+1/2,k} - ()_{j-1/2,k}],$$
$$\mu_y()_{j,k} = 0.5[()_{j,k+1/2} + ()_{j,k-1/2}], \text{ and} \qquad \delta_y()_{j,k} = [()_{j,k+1/2} - ()_{j,k-1/2}].$$

Vector quantities are denoted by bold face font, while matrices are underlined. Superscripts appearing in discretizations denote the time level, while subscripts refer to spatial indices. Since all meshes are square, $\Delta x = \Delta y \equiv h$, and the Courant-Friedrichs-Lewy (CFL) number is defined as $\nu = a_0 \Delta t / h$.

3.2 Vorticity Control

Dukowicz and Meltz [29] explored the link between mesh tangling and spurious vorticity in the context of Lagrangian hydrocodes. They implemented a correction procedure to remove unwanted vorticity from the final computed solution, and though expensive and first order, it was successful at reducing mesh imprinting and spurious vorticity when solving the Saltzman problem. It would, therefore, be desirable to find a scheme that has built in vorticity control. This makes the RR scheme a good place to start since Morton and Roe showed that it exactly preserves compact vorticity, $\boldsymbol{\zeta} = \mu_y \delta_x v - \mu_x \delta_y u$, in [66]. Three questions arise here that will be addressed. First, is spurious vorticity generation a genuine concern for algorithms that are derived in the absence of any vorticity considerations? Second, can the link between vorticity control and mesh imprinting be established in the context of the acoustic equations and the LW family of schemes? Third, is vorticity control alone an adequate method to preserve symmetry?

Numerical results from the RR scheme and the original LW method were compared using the discontinuous test problem. The computations were run with $\nu = 0.6$. Contour plots of compact vorticity, rendered at $t = 3$, make a strong case that problems

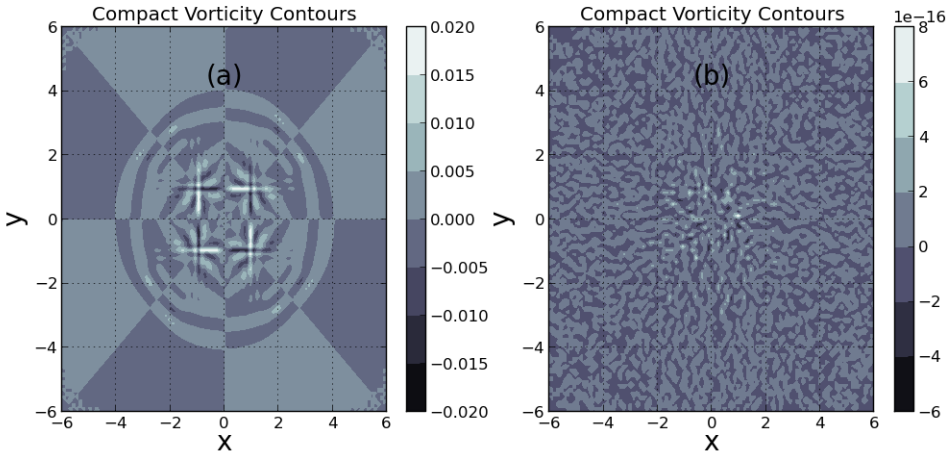

Figure 3.2: Compact vorticity contours predicted by LW (a) and RR (b) for the discontinuous test problem are plotted at $t = 3$. The computations were run with $\nu = 0.6$. The LW method generates spurious vorticity, but the RR method maintains zero vorticity to double precision.

with vorticity will exist if no special care is taken. Figure 3.2 shows that significant unphysical vorticity is present in the LW solution. In comparison, the RR scheme has properly maintained zero vorticity to double precision. Morton and Roe showed that spurious vorticity production will occur on the order of the truncation error of a scheme, but that the small errors will accumulate over time. While the errors would accumulate slowly for a smooth problem, here the LW method generates vorticity around the initial discontinuity early in the solution. In this region the higher-order terms in the truncation error are not well behaved. While the improvement demonstrated by RR with respect to vorticity is desirable, changes in mesh imprinting and isotropy are of keen interest. To gauge the effect of vorticity control on these issues, the pressure and velocity magnitude profiles were plotted as a function of radius. See Figure 3.3, which also includes a reference solution computed using standard methodology (an unsplit MUSCL-Hancock scheme[80] on on a 600 × 600 grid). Two things become evident. First, the RR scheme produces a solution that is much more isotropic than the LW method. In fact, the scatter has been reduced by upwards of 50 percent

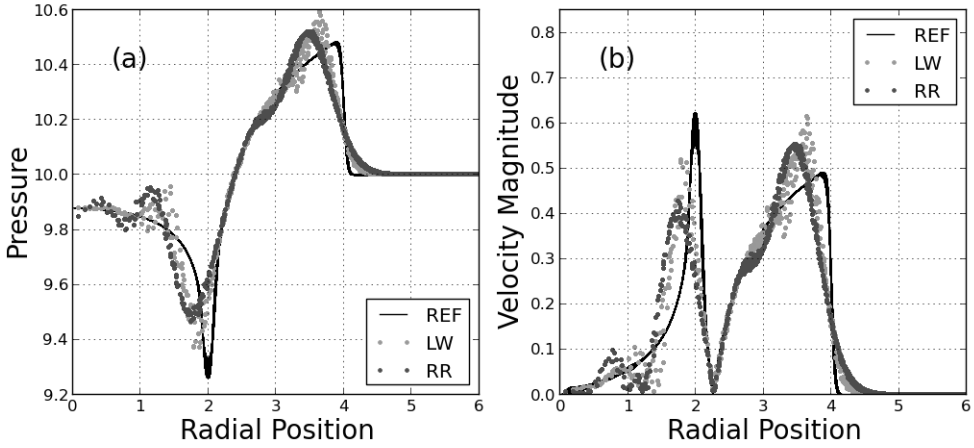

Figure 3.3: LW and RR comparison: Pressure (a) and velocity magnitude (b) profiles for the discontinuous test problem are plotted at $t = 3$. The computations were run with $\nu = 0.6$.

as shown in Appendix B. Second, when compared to the reference solution in the background, both the RR and LW schemes produce a number of spurious features. This is to be expected for a discontinuous problem, however, since no limiting was employed. Figure 3.4 shows results from the smooth, low frequency test problem. In this case the LW, RR, and the MUSCL-H reference solutions are all nearly the same. The high frequency dispersion relationships for each scheme and the resulting performance on the discontinuous test problem are of primary interest. While it has been demonstrated that the RR scheme is superior to the original LW scheme with respect to vorticity control and mesh imprinting, further improvements to the method may still be possible. The entire family of Lax-Wendroff type schemes was parameterized and investigated to explore this possibility.

3.2.1 Parameterization of the Lax-Wendroff Family

When using a nine point stencil in two spatial dimensions, there is considerable freedom in how a LW type scheme can be formulated. Here this family of schemes is parameterized in the most general way possible. The LW scheme can be derived by

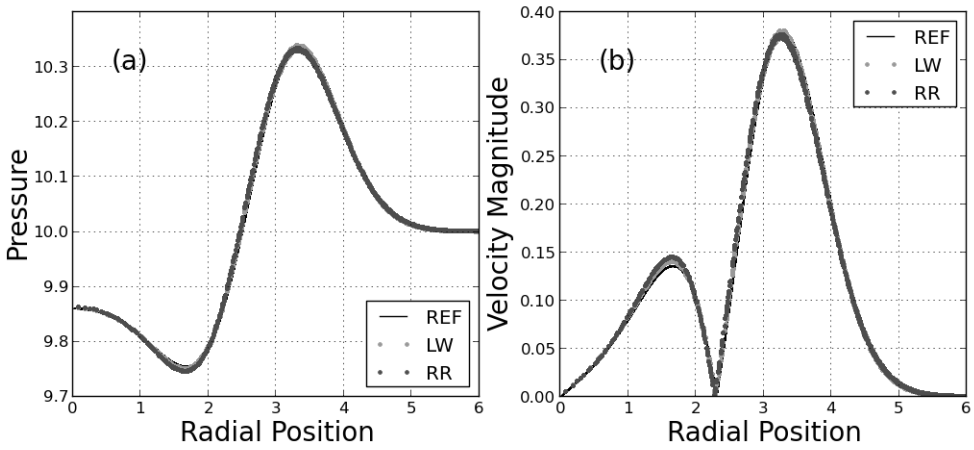

Figure 3.4: LW and RR comparison: Pressure (a) and velocity magnitude (b) profiles for the Gaussian test problem are plotted at $t = 3$. The computations were run with $\nu = 0.6$.

writing the Taylor expansion of $\mathbf{U}$ in time

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \mathbf{U}_t^n + \frac{\Delta t^2}{2!}\mathbf{U}_{tt}^n + O(\Delta t^3)$$

and then using (3.1) to replace the first and second order time derivatives with spatial derivatives. Denoting the flux Jacobians by $\underline{A} = \partial\mathbf{F}/\partial\mathbf{U}$ and $\underline{B} = \partial\mathbf{G}/\partial\mathbf{U}$ we have

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = -\mathbf{F}_x - \mathbf{G}_y + \frac{\Delta t}{2}\left[\underline{A}(\mathbf{F}_{xx} + \mathbf{G}_{yx}) + \underline{B}(\mathbf{F}_{xy} + \mathbf{G}_{yy})\right] + O(\Delta t^2). \quad (3.3)$$

The exact spatial derivatives in (3.3) must now be replaced with appropriate central difference approximations. These discretizations are not uniquely defined on the nine point stencil except in the case of the second order mixed partial derivative. The final approximations for $()_x, ()_{xx}, ()_y$, and $()_{yy}$ must be constrained so that x is treated equitably with y, and u with v. However, the weights assigned to the pressure derivatives need not be the same as those given to the velocities. Taken together, these guidelines result in four free parameters, $\alpha_1,\ \alpha_2,\ \phi_1$ and ϕ_2. The α weights were used for pressure derivatives, while the ϕ weights were used for velocity derivatives. The

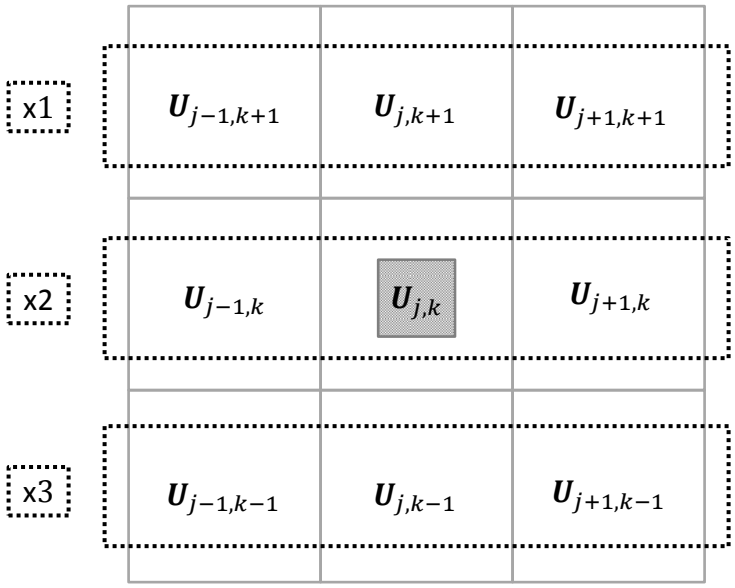

Figure 3.5: An example parameterization for the approximation of the second derivative of a generic state variable U with respect to x is shown.

subscripts one and two represent whether the parameter applies to first- or second-order derivative approximations, respectively. An example parameterization for a second derivative is shown in Figure 3.5. Writing the general form of the scheme as

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \underline{T}\mathbf{U}^n \tag{3.4}$$

the parameterized evolution operator, $\underline{T}$, is given by[2]

$$\underline{T} = \begin{bmatrix} \frac{\nu^2}{2}(\delta_x^2(1+\alpha_2\delta_y^2) + \delta_y^2(1+\alpha_2\delta_x^2)) & -\nu\mu_x\delta_x(1+\phi_1\delta_y^2) & -\nu\mu_y\delta_y(1+\phi_1\delta_x^2) \\ -\nu\mu_x\delta_x(1+\alpha_1\delta_y^2) & \frac{\nu^2}{2}\delta_x^2(1+\phi_2\delta_y^2) & \frac{\nu^2}{2}\mu_x\mu_y\delta_x\delta_y \\ -\nu\mu_y\delta_y(1+\alpha_1\delta_x^2) & \frac{\nu^2}{2}\mu_x\mu_y\delta_x\delta_y & \frac{\nu^2}{2}\delta_y^2(1+\phi_2\delta_x^2) \end{bmatrix}. \tag{3.5}$$

The challenge is to determine the best choices for the four free parameters so as to

[2]The identity $4(\mu_x^2 - 1)\phi = \delta_x^2\phi$ is needed to obtain the form of $\underline{\mathrm{T}}$ shown.

find an optimal LW type method in two spatial dimensions using a nine point stencil. Of course, this has been attempted many times in the past. A few examples will be highlighted here. Lax and Wendroff extended their original one-dimensional method [47] to two spatial dimensions in [48]. The simplest scheme they proposed, and the one taken here to be the "original" two-dimensional LW method, did not use the corner points except in the case of the second order mixed partial derivative. They also suggested a method that included a dissipation term which is proportional to the fourth order mixed partial derivative ∂_{xxyy}. Strang constructed a multidimensional LW type method from the one-dimensional LW difference operators in [76]. Like the schemes proposed by Lax and Wendroff, the corner points were not used in the approximations for u_x, u_{xx}, u_y, or u_{yy}. However, Strang's scheme included terms that approximated both third and fourth order derivatives. This and other early work was summarized by Turkel [82] about a decade later.

Gottlieb and Turkel [37] derived a modified LW method by considering the phase error. Another class of schemes, which are interesting due to the manner of derivation, have been developed by Lukáčová-Medvid'ová *et. al.* [55] using their Evolution-Galerkin approach and the bicharacteristic form of the equations. In this work a different path was taken in analyzing the parameterized Lax-Wendroff family of schemes. A two-dimensional dispersion analysis was performed to show how the isotropy of the numerical dispersion relations depended upon the free parameters. This dependence was to be minimized. While considerable gains were realized earlier by enforcing vorticity preservation, only the isotropy of the dispersion relations will be considered initially. Later, a connection between the two properties is made. All of the schemes discussed are described by special cases of (3.5).

3.3 Dispersion Analysis

A two-dimensional von Neumann analysis was performed on the parameterized scheme. This exposed anisotropic terms in the numerical dispersion relationships and highlighted choices of parameters that would eliminate them. To begin the analysis, a two-dimensional von Neumann substitution of the form $\mathbf{U}^n_{j,k} = g^n \exp[ir\theta]\mathbf{r} = g^n \exp[i(\theta_x j + \theta_y k)]\mathbf{r}$ was carried out with (3.4). Here $i = \sqrt{-1}$. This substitution assumes plane (or line) wave solutions that have frequency θ_r in the direction ψ, which is measured from the positive x-axis. Note that the operators in matrix $\underline{T}$ are replaced by exponentials that appropriately shift the assumed solution in space. For example,

$$\delta_x \rightarrow \exp\left[\frac{i\theta_x}{2}\right] - \exp\left[-\frac{i\theta_x}{2}\right] \quad \text{and} \quad \mu_y \rightarrow \frac{1}{2}\left(\exp\left[\frac{i\theta_y}{2}\right] + \exp\left[-\frac{i\theta_y}{2}\right]\right). \quad (3.6)$$

With some simplification the equation

$$g\mathbf{r} = \mathbf{r} + \underline{T}\mathbf{r}$$

is obtained. After letting $\underline{I} + \underline{T} = \widehat{\underline{T}}$ and more manipulation, the standard eigenvalue problem can be recovered

$$\widehat{\underline{T}}\mathbf{r} = g\mathbf{r}.$$

Here g represents the eigenvalues of the matrix $\widehat{\underline{T}}$ and $\mathbf{r}$ represents the eigenvectors. For the problem at hand, $\widehat{\underline{T}}$ is a 3x3 matrix. The three resulting eigenvalues are interpreted as follows. One is real and relates to the stationary vorticity mode; it should be exactly unity. The other two eigenvalues, which are a complex conjugate pair, represent right and left going acoustic waves. The magnitude of the eigenvalues gives the amplification factors, while their arguments give the phase changes per time step. After computing the eigenvalues symbolically, they can be expanded with

respect to the signal frequency θ_r. The resulting expressions are

$$\begin{aligned} g_1 =& 1 - \frac{\nu^2\theta_r^4}{32}(1-4\phi_2)\sin^2 2\psi \\ &+ \frac{\nu^2\theta_r^6}{768}\left[3(1-4\alpha_1)(1-4\phi_1)\sin^2 2\psi - 2(1-4\phi_2)\right]\sin^2 2\psi + O(\theta_r^8) \quad (3.7) \\ g_{2,3} =& 1 \pm i\nu\theta_r - \frac{\nu^2\theta_r^2}{2} \mp i\frac{\nu\theta_r^3}{24}\left[4 - 2(1-3(\alpha_1+\phi_1))(1-2\sin^2 2\psi)\right] \\ &+ \frac{\nu^2\theta_r^4}{192}\left[8 + (1-12\phi_2-24\alpha_2)\sin^2 2\psi\right] + O(\theta_r^5). \quad (3.8) \end{aligned}$$

It is immediately apparent that something special will happen to g_1 when any parameter is set equal to $1/4$, which corresponds to evaluating the relevant term from vertex fluxes. For the RR scheme, all of the parameters take this value. To maximize isotropy, terms that depend on the wave orientation ψ should be eliminated. The potentially anisotropic fourth-order term in (3.7) can be eliminated if $\phi_2 = 1/4$, so that g_1 becomes

$$g_1 = 1 - \frac{\nu^2\theta_r^6}{256}(1-4\alpha_1)(1-4\phi_1)\sin^2 2\psi + O(\theta_r^8). \quad (3.9)$$

The sixth-order term can also be removed by choosing either $\alpha_1 = 1/4$ or $\phi_1 = 1/4$[3]. By comparing the RR evolution operator with our parameterized evolution operator $\underline{T}$, one can deduce that setting $\alpha_1 = 1/4$ and $\phi_2 = 1/4$ will recover the vorticity preservation property. The choice of ϕ_2 is the same as was specified from the dispersion analysis. In addition, the quandary about whether to set $\phi_1 = 1/4$ or $\alpha_1 = 1/4$ has been resolved. It is interesting, however, that either choice results in a real, isotropic eigenvalue provided $\phi_2 = 1/4$. The choice of $\alpha_1 = 1/4$ corresponds to a family of schemes that exactly preserve compact vorticity. The choice of $\phi_1 = 1/4$ corresponds to a family of schemes that can exactly preserve physically correct steady solutions.

[3]In fact with these choices g_1 becomes exactly unity, since $\det(\underline{T})$ can be shown to vanish.

A concrete link between vorticity control and isotropy has been demonstrated. The two unique choices of α_1 and ϕ_2 that guarantee vorticity preservation also make the g_1 eigenvalue isotropic. However, there is still complete freedom in choosing α_2 and ϕ_1. This family of schemes will be referred to as vorticity preserving Lax-Wendroff (VPLW). The eigenvalue expansions now read

$$g_1 = 1 \text{ and} \tag{3.10}$$

$$\begin{aligned} g_{2,3} =& 1 \pm i\nu\theta_r - \frac{\nu^2\theta_r^2}{2} \mp i\frac{\nu\theta_r^3}{48}\left[8 - (1-12\phi_1)(1-2\sin^2 2\psi)\right] \\ &+ \frac{\nu^2\theta_r^4}{96}\left[4 - (1+12\alpha_2)\sin^2 2\psi + O(\theta_r^5)\right]. \end{aligned} \tag{3.11}$$

Inspection of (3.11) shows that the next two anisotropic terms could be eliminated by choosing $\phi_1 = 1/12$ and $\alpha_2 = -1/12$. The resulting scheme, denoted as VPLW1, is maximally isotropic. Results from this scheme are presented in Figure 3.6. The isotropy of the solution is improved to the point that it exceeds that of the reference solution obtained on a much finer grid. While this is impressive, numerical experiments showed that the stability of the scheme was reduced, perhaps because of the negative weight. The RR scheme is maximally stable up to a CFL number of one, while the VPLW1 method was observed to go unstable around $\nu = 0.7$. It was determined that stability could be improved by increasing α_2 and, since maximal stability was desired, the negative weight was discarded and a suitable positive replacement was sought. The α_2 parameter controls the discretization of the second-order terms in the pressure update equation, which approximate the Laplacian of the pressure. An isotropic spatial discretization for the scalar wave equation was proposed by Vichnevetsky and Bowles in [87]. It is obtained here by taking $\alpha_2 = 1/8$, leading to

$$\nabla^2() \approx (\delta_x^2 + \delta_y^2 + \frac{1}{4}\delta_x^2\delta_y^2)()/h^2 = \nabla^2() + \frac{h^2}{12}\nabla^4() + O(h^4).$$

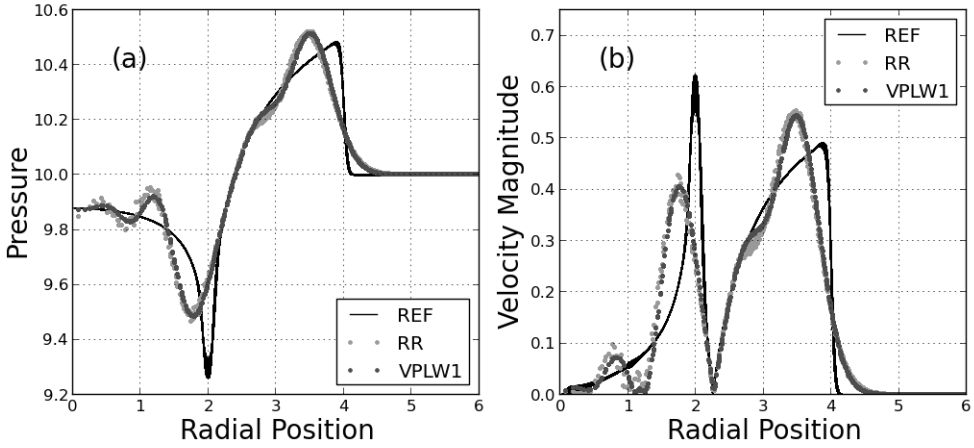

Figure 3.6: RR and VPLW1 comparison: Pressure (a) and velocity magnitude (b) profiles for the discontinuous test problem are plotted at $t = 3$. The computations were run with $\nu = 0.6$.

This was shown to be the most isotropic way to represent the Laplacian on nine points in [43][4]. The scheme that is defined by $\phi_1 = 1/12$ and $\alpha_2 = 1/8$ is denoted as VPLW2. Inspection of the results shown in Figure 3.7 shows that the isotropy of the VPLW2 scheme is still improved over RR. A quantitative assessment showed that deviations from radial symmetry were reduced by over 80 percent when compared to LW. In addition, numerical experiments have indicated that the VPLW2 scheme is stable up to a CFL number of one. Due to the favorable combination of isotropy and stability, the VPLW2 method is selected as the preferred choice. In anticipation of a future FCT implementation, the method is written in finite volume (FV) form. Denoting vertex fluxes by $\hat{()}$ and face fluxes by $\tilde{()}$, the vorticity preserving finite volume (VPFV)2 scheme is written as

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \frac{\Delta t}{h}(\mu_y \delta_x \hat{\mathbf{F}} + \mu_x \delta_y \hat{\mathbf{G}} + \delta_x \tilde{\mathbf{F}} + \delta_y \tilde{\mathbf{G}})$$

[4]However, taking $\alpha_1 = \phi_2 = 1/4$ is not the most isotropic way to evaluate the first derivatives. Instead it is the choice that gives overall the most isotropic behavior. That is why it was asserted earlier that the Lax-Wendroff family can only be optimized with a specific problem in mind.

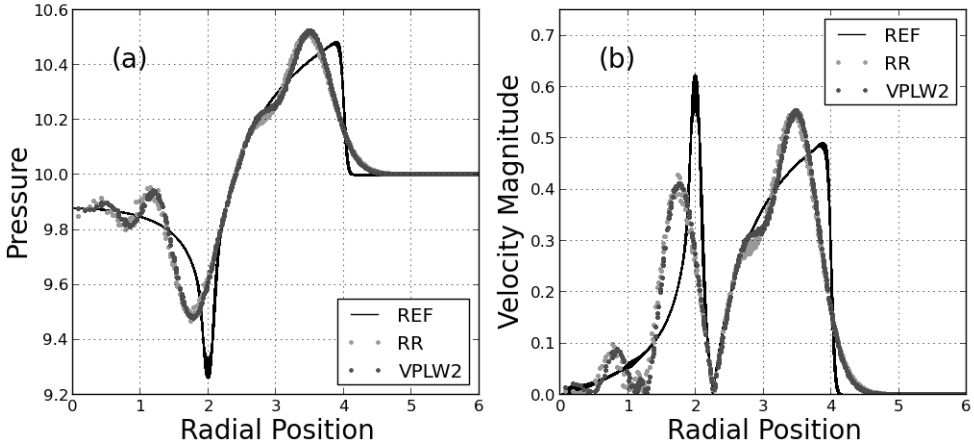

Figure 3.7: RR and VPLW2 comparison: Pressure (a) and velocity magnitude (b) profiles for the discontinuous test problem are plotted at $t = 3$. The computations were run with $\nu = 0.6$.

where, taking $q = \nu^2$, the fluxes are defined by

$$\hat{\mathbf{F}} = \begin{pmatrix} \frac{1}{3}\mu_x\mu_y au - \frac{qh}{4\Delta t}\hat{\beta}_u \\ \mu_x\mu_y ap - \frac{qh}{2\Delta t}\hat{\beta}_p \\ 0 \end{pmatrix}, \qquad \hat{\mathbf{G}} = \begin{pmatrix} \frac{1}{3}\mu_x\mu_y av - \frac{qh}{4\Delta t}\hat{\beta}_v \\ 0 \\ \mu_x\mu_y ap - \frac{qh}{2\Delta t}\hat{\beta}_p \end{pmatrix},$$

$$\tilde{\mathbf{F}} = \begin{pmatrix} \frac{2}{3}\mu_x au - \frac{qh}{4\Delta t}\tilde{\beta}_u \\ 0 \\ 0 \end{pmatrix}, \text{ and} \qquad \tilde{\mathbf{G}} = \begin{pmatrix} \frac{2}{3}\mu_y av - \frac{qh}{4\Delta t}\tilde{\beta}_v \\ 0 \\ 0 \end{pmatrix}.$$

The β quantities that appear in the fluxes will be referred to as the "driver quantities" since they drive temporal changes in the fluxes. The definition for each driver is contained in Table 3.1. Note that altering the pressure update equation has led to pressure fluxes being stored at both vertices and faces. Although this adds some expense, the cost increases appear to be justified by the increase in isotropy.

Table 3.1: Discrete Driver Definitions

Quantity	Vertex	Face
Pressure	$\hat{\beta}_p = \mu_y \delta_x u + \mu_x \delta_y v$	N/A
u-Velocity	$\hat{\beta}_u = \mu_y \delta_x p$	$\tilde{\beta}_u = \delta_x p$
v-Velocity	$\hat{\beta}_v = \mu_x \delta_y p$	$\tilde{\beta}_v = \delta_y p$

3.4 Discussion

An acoustic method that suffers from very little mesh imprinting was obtained by insisting on vorticity preservation and then making the leading truncation errors in the dispersion relations isotropic. In fact, it was shown that the two exercises are directly related. In order to obtain physical vorticity transport in the context of acoustics, the fluxes for the velocity updates must be stored at vertices. However, this constraint does not apply to the pressure update and further reductions in mesh imprinting resulted from using both face and vertex fluxes. Remember, however, that the "face fluxes" here are still point estimates and really just provide an additional integration point to use when determining the face-averaged values. Looking ahead to Lagrangian hydrodynamics, it would appear that constructing the method in terms of multidimensional point fluxes will be important in hopes that implicit vorticity control will be retained to the greatest extent possible.

The numerical results shown in this chapter contained many spurious features since no limiters were used. These were due to the numerical dispersion relationships, which propagate some modes with significant phase errors, but little damping. This problem is common to second-order accurate methods and easily recognized if the phase and damping relationships for the VPLW2 method are plotted. See Figures 3.8 and 3.9. It is disconcerting to note that, according to the figures, the problem gets much worse at lower CFL numbers. Figure 3.10 shows results from the discontinuous test problem when $\nu = 0.15$ and the spurious features are so prevalent

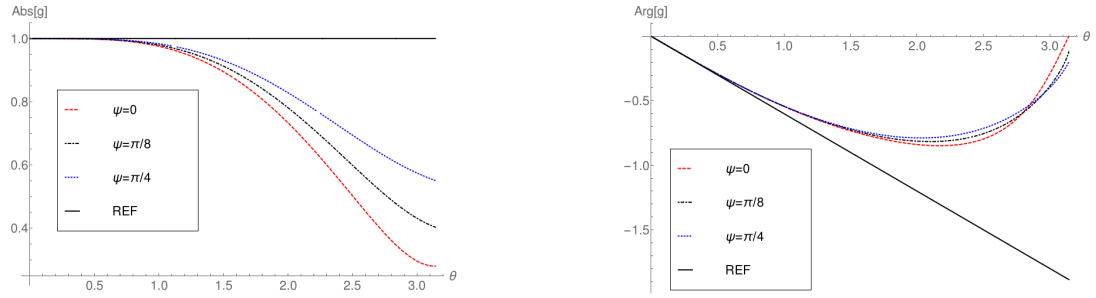

Figure 3.8: The phase and damping relationships for the VPLW2 method are plotted for the propagation directions $\psi = 0$, $\psi = \pi/4$, and $\psi = \pi/8$ when $\nu = 0.6$.

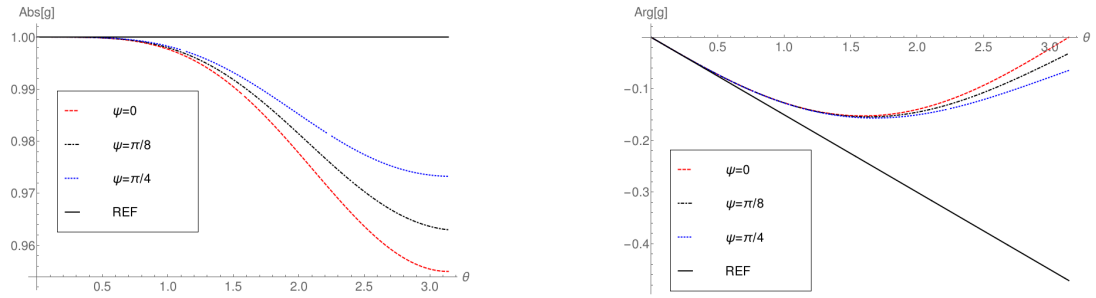

Figure 3.9: The phase and damping relationships for the VPLW2 method are plotted for the propagation directions $\psi = 0$, $\psi = \pi/4$, and $\psi = \pi/8$ when $\nu = 0.15$.

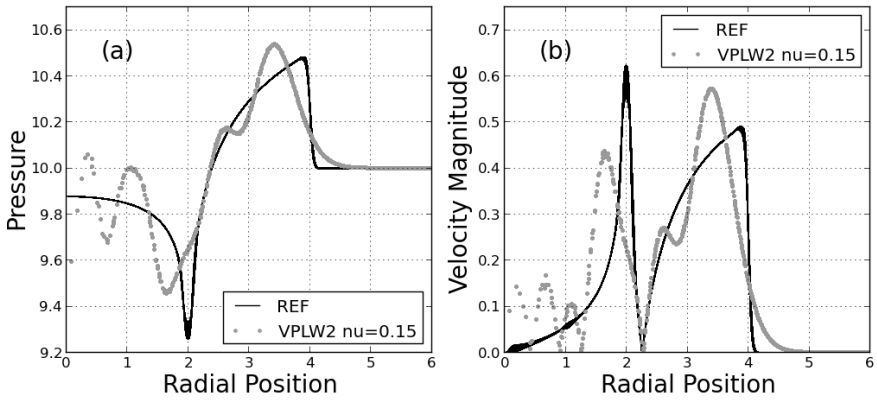

Figure 3.10: Results from the VPLW2 method for the discontinuous test problem show that the spurious features in the solution become much more severe when the CFL number is lowered to $\nu = 0.15$.

that the physical solution is almost unrecognizable. Nevertheless, this situation can be remedied. Some third-order accurate schemes that correct this deficiency will be explored in the next chapter.

CHAPTER IV

Third-order Methods for Acoustics

In Chapter III second-order accurate acoustic methods were obtained that suffered from very little mesh imprinting. However, substantial phase errors were present for mid- to high-frequency modes that also received little damping, especially at low CFL numbers. As a result, numerous spurious features persisted in the solutions to the discontinuous test problem. The unwanted combination of low damping and large phase errors is typical of schemes with even orders of accuracy and it has been observed that spurious oscillations can be reduced by using odd-ordered schemes [9]. Eymann and Roe argue that third-order methods are the most practical as they strike a good balance between performance and cost [32]. To try and improve the poor dispersion characteristics of the VPLW2 method, this chapter will investigate some third-order vorticity preserving acoustic methods.

It cannot be forgotten that the problem of spurious extrema in the solution will eventually need to be addressed by a limiter: Godunov proved that no linear scheme with better than first-order accuracy can fully prevent them [35]. However, it is a difficult task to design a limiter that can remove all the spurious features from the current VPLW2 solutions without damaging the physical ones in the process. To successfully accomplish this, some guidance was needed. The goal of this chapter is to find third-order methods that improve the numerical dispersion relations of the

VPLW2 method in way that maintains, or even enhances the prediction of physical features in the solution. The focus will remain on linear schemes for now, but nonlinear acoustic methods will be developed in Chapter V to mimic the third-order methods developed here.

Recall that FCT was proposed in Chapter I as a multidimensional limiting framework. The third-order methods considered in this chapter will be constructed using the VPLW2 method from Chapter III and written in terms of a FCT type update procedure. The increased accuracy will be made possible by incorporating information from the first-order provisional solution into antidiffusive corrections. Section 4.1 will use the linear advection equation to present a concise exposition on how a provisional first-order solution can be used to obtain third-order accuracy. Then, in Section 4.2 the technique will be carried over to two-dimensional acoustics and the phase and damping characteristics of some specific third-order methods will be assessed. Numerical results are presented in Section 4.3 to compare the performance of the new methods to the second-order VPLW2 method. Finally, Section 4.4 concludes the chapter with some discussion.

4.1 Third-order Accuracy

The technique that will be used to obtain third-order accuracy can be illustrated by considering the one-dimensional linear advection equation[1]

$$u_t + a u_x = 0,$$

where the quantity u is advected with constant speed a. The family of q-schemes discussed in [83] will be chosen to solve the problem. The usual single step update is

[1]The linear advection analysis included here not general, it holds only for the case of single, scalar CFL number, but it is straightforward and instructive.

written

$$u^{n+1} = u^n - \nu\mu\delta u^n + \frac{q}{2}\delta^2 u^n \tag{4.1}$$

and many well known schemes may be recovered with specific choices of the parameter q. Stability requires that $\nu^2 \leq q \leq 1$ and the choice $q = \nu^2$ is required for second-order accuracy. This is the original Lax-Wendroff method. Any other stable choice for q gives a first-order method and $q = |\nu|$, $q = 1$, or $q = 1/3 + 2\nu^2/3$ result in the FUP, Lax-Friedrichs (LF), and low phase error (LPE) methods, respectively.

The update in 4.1 will be decomposed into a two step procedure. In the first step, a first-order provisional solution is obtained according to

$$u^* = u^n - \nu\mu\delta u + \frac{q_C}{2}\delta^2 u, \tag{4.2}$$

where the choice of first-order method, which is determined by q_C, has not been specified. Then, an antidiffusive correction step is defined according to the difference between the cautious method and the accurate LW one. Defining $q^+ = q_C - \nu^2$, it is

$$u^{n+1} = u^* + \frac{q^+}{2}\delta^2 u^n.$$

As it stands, the proposed method is only second-order accurate and more information is required to achieve third-order accuracy. In this case, the information already exists in u^*; it just needs to be incorporated into the antidiffusion step. This is easily done by introducing a parameter κ and modifying the antidiffusive update to be

$$u^{n+1} = u^* + \frac{q^+}{2}[(1-\kappa)\delta^2 u^n + \kappa\delta^2 u^*]. \tag{4.3}$$

A limiter could be introduced to modify the antidiffusive correction, but this will be neglected for now.

To determine the constraints on κ and q_C necessary for third-order accuracy, a

Table 4.1: Third-order Constraints

Embedded First-Order Method	q_C	κ
LF	1	$\frac{1}{3}$
FUP	ν	$\frac{1}{3}\frac{\nu^2-1}{\nu^2-\nu}$
LPE	$\frac{1+2\nu^2}{3}$	1

von Neumann substitution was performed on the method described by 4.2 and 4.3. Then the amplification factor, g, was expanded in terms of the signal frequency θ giving

$$g = 1 - i\nu\theta - \frac{\nu^2\theta^2}{2} + \frac{i\nu\theta^3}{6}\left[1 - 3\kappa q^+\right] + O(\theta^4). \tag{4.4}$$

From the coefficient on the third-order term, it is evident that the method will be third-order accurate if

$$\kappa = \frac{1-\nu^2}{3q^+}. \tag{4.5}$$

Some specific values of κ are shown in Table 4.1 for choices of q_C that correspond to the FUP, LPE, and LF methods. Note that if the LPE method is chosen, then $\kappa = 1$ and the antidiffusive update only depends on the provisional solution. This would be advantageous from an implementation standpoint. While not explicitly written as such, the methods proposed here are essentially FCT methods with no flux limiter[2]. It should be noted that using the provisional solution in the antidiffusion step is not a new idea. In fact, when Boris and Book invented the FCT methodology, they applied the antidiffusion operator to the provisional first-order solution and noted that this formulation could improve the solution, even in the absence of a flux limiting mechanism [8]. The accuracy constraint used here is simply another way to make use of the provisional solution in a FCT method.

[2]If desired, these methods are easily written in FV form as $u^{n+1} = u^n - \frac{k}{h}\delta f$, where $f = \mu a u - \frac{qh}{2k}\delta u$. Then it follows that a FCT method can be defined by the fluxes $f_C = \mu a u - \frac{q_C h}{2\Delta t}\delta u$ and $f_{AD} = \frac{q^+ h}{2\Delta t}\delta u$.

4.2 Third-order Vorticity Preserving Methods for Acoustics

The update procedure and third-order accuracy constraint developed in the previous section carry over to acoustics and can be used to construct third-order methods based on the VPLW2 evolution operator from Chapter III. The methods will be written in FV form since it will be conducive to limiting in the next chapter. The cautious (C) first-order provisional solution is obtained from

$$\mathbf{U}^* = \mathbf{U}^n - \frac{\Delta t}{h}(\mu_y \delta_x \hat{\mathbf{F}}_C + \mu_x \delta_y \hat{\mathbf{G}}_C + \delta_x \tilde{\mathbf{F}}_C + \delta_y \tilde{\mathbf{G}}_C), \tag{4.6}$$

where the fluxes are defined by

$$\hat{\mathbf{F}}_C = \begin{pmatrix} \frac{1}{3}\mu_x\mu_y a_0 u - \frac{q_C h}{4\Delta t}\hat{\beta}_u^n \\ \mu_x\mu_y a_0 p - \frac{q_C h}{2\Delta t}\hat{\beta}_p^n \\ 0 \end{pmatrix}, \qquad \hat{\mathbf{G}}_C = \begin{pmatrix} \frac{1}{3}\mu_x\mu_y a_0 v - \frac{q_C h}{4\Delta t}\hat{\beta}_v^n \\ 0 \\ \mu_x\mu_y a_0 p - \frac{q_C h}{2\Delta t}\hat{\beta}_p^n \end{pmatrix},$$

$$\tilde{\mathbf{F}}_C = \begin{pmatrix} \frac{2}{3}\mu_x a_0 u^n - \frac{q_C h}{4\Delta t}\tilde{\beta}_u^n \\ 0 \\ 0 \end{pmatrix}, \text{ and } \qquad \tilde{\mathbf{G}}_C = \begin{pmatrix} \frac{2}{3}\mu_y a_0 v^n - \frac{q_C h}{4\Delta t}\tilde{\beta}_v^n \\ 0 \\ 0 \end{pmatrix}.$$

The antidiffusive step is then

$$\mathbf{U}^{n+1} = \mathbf{U}_C^* - \frac{\Delta t}{h}[\mu_y \delta_x \hat{\mathbf{F}}_{AD} + \mu_x \delta_y \hat{\mathbf{G}}_{AD} + \delta_x \tilde{\mathbf{F}}_{AD} + \delta_y \tilde{\mathbf{G}}_{AD}].$$

After defining enhanced (E) drivers of the form

$$\beta^E = (1-\kappa)\beta^n + \kappa\beta^* = \beta^n + \kappa(\beta^* - \beta^n) \tag{4.7}$$

the antidiffusive fluxes are

$$\hat{\mathbf{F}}_{AD} = \begin{pmatrix} \frac{q^+h}{4\Delta t}\hat{\beta}_u^E \\ \frac{q^+h}{2\Delta t}\hat{\beta}_p^E \\ 0 \end{pmatrix}, \qquad \hat{\mathbf{G}}_{AD} = \begin{pmatrix} \frac{q^+h}{4\Delta t}\hat{\beta}_v^E \\ 0 \\ \frac{q^+h}{2\Delta t}\hat{\beta}_p^E \end{pmatrix},$$

$$\tilde{\mathbf{F}}_{AD} = \begin{pmatrix} \frac{q^+h}{4\Delta t}\tilde{\beta}_u^E \\ 0 \\ 0 \end{pmatrix}, \text{ and} \qquad \tilde{\mathbf{G}}_{AD} = \begin{pmatrix} \frac{q^+h}{4\Delta t}\tilde{\beta}_v^E \\ 0 \\ 0 \end{pmatrix}.$$

The definitions for the driver quantities carry over from Chapter III, but two different estimates exist: one computed from $\mathbf{U}^n$ and one computed from $\mathbf{U}^*$. While it would be possible to specify unique κ weights for the pressure and velocity update equations in this case, this extra degree of freedom will not be considered. Note that the third-order methods are obtained by repeated application of the vorticity preserving VPLW2 evolution operator and, therefore, preserve vorticity themselves. This family of linear schemes will be referred to as VPFCTO3.

4.2.1 Dispersion Analysis

A two-dimensional dispersion analysis was performed to investigate the stability properties of the third-order methods and to see if their phase and damping characteristics were improved as compared to the VPLW2 method. The resulting amplification factor expansions, written in terms of the signal frequency θ, are

$$g_1 = 1 \text{ and}$$
$$g_{2,3} = 1 \pm i\nu\theta \mp \frac{\nu^2\theta^2}{2} \mp \frac{i\nu\theta^3}{6}[1 - 3\kappa q^+] + O(\theta^4).$$

The first eigenvalue is one, which confirms that the new family of schemes is vorticity preserving. As asserted earlier, the same conditions for third-order accuracy derived

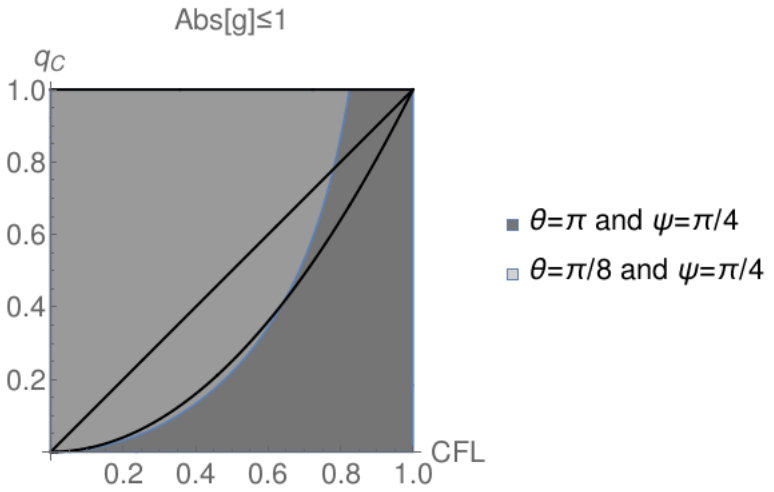

Figure 4.1: It is somewhat disappointing that none of the VPFCTO3 methods are optimally stable. It is evident here that the $\theta = \pi/8$ wave traveling with direction $\psi = \pi/4$ will be unstable by $\nu \approx 0.8$ regardless of the choice for q_C. The functions $q_C = \nu$, $q_C = \nu^2$, and $q_C = 1$ are plotted for reference.

in the case of linear advection can be obtained from the θ^3 term here.

To better understand the stability properties of the new third-order methods, various combinations of signal frequency and propagation direction were chosen and the corresponding stability regions were plotted as functions of q_C and the CFL number. In Figure 4.1, the stability regions for $\theta = \pi$ and $\theta = \pi/8$ waves traveling diagonally across the grid are shown. It is clear from the figure that the optimal stability limit is not obtainable since the $\theta = \pi/8$ mode is unstable above $\nu \approx 0.8$, regardless of the choice of first-order scheme. While somewhat disappointing, the decrease in the stability limit implied in the plot is not catastrophic and the methods still have a larger stability region than other unsplit methods such as MUSCL-Hancock (MUSCL-H), which is unstable above $\nu = 0.5$. Perhaps some of the methods will still prove useful.

Three third-order acoustic methods were further investigated by specifying specific values for q_C. The same choices were made as in Section 4.1 so that multidimensional analogs of the FUP, LPE and LF first-order methods were recovered. Some experimentation revealed that the stability limits for the VPFCTO3-FUP, VPFCTO3-LPE, and VPFCTO3-LF methods were approximately $\nu = 0.75$, $\nu = 0.75$, and $\nu = 0.8$, re-

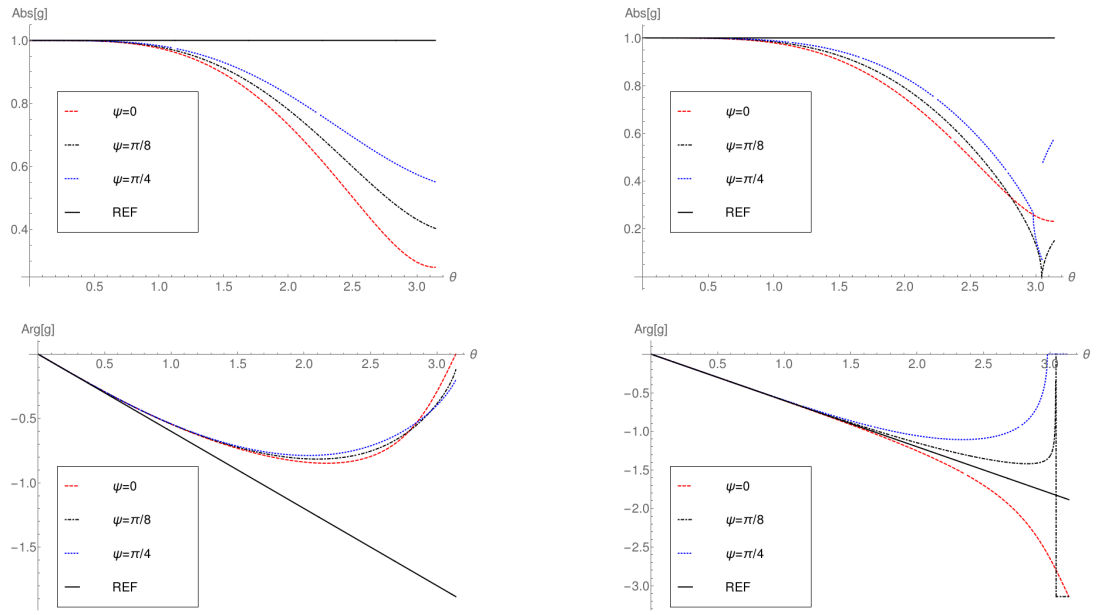

Figure 4.2: The amplification (top) and phase (bottom) relationships for the propagation directions $\psi = 0$, $\psi = \pi/8$, and $\psi = \pi/4$ are plotted for the second-order VPLW2 method and the VPFCTO3-FUP method when $\nu = 0.6$.

spectively. It was expected that the LF based method would have the largest stability region given Figure 4.1.

The damping and phase relationships for the VPFCTO3-FUP method are compared to the those of the second-order VPLW2 method in Figure 4.2. Here $\nu = 0.6$ and the relations are plotted for the directions $\psi = 0$, $\psi = \pi/8$, and $\psi = \pi/4$. It is clear that the jump to third-order accuracy has reduce the troublesome combination of high phase error and low damping. Not only are the phase errors smaller in general, but greater damping is present in the higher frequency ranges where significant phase errors occur. At higher CFL numbers the third-order method all exhibit similar behavior, so only results from the FUP based method are shown for that case.

More variation is observed in the dispersion characteristics of the third-order schemes at lower CFL numbers, so results from each will be shown for $\nu = 0.15$.

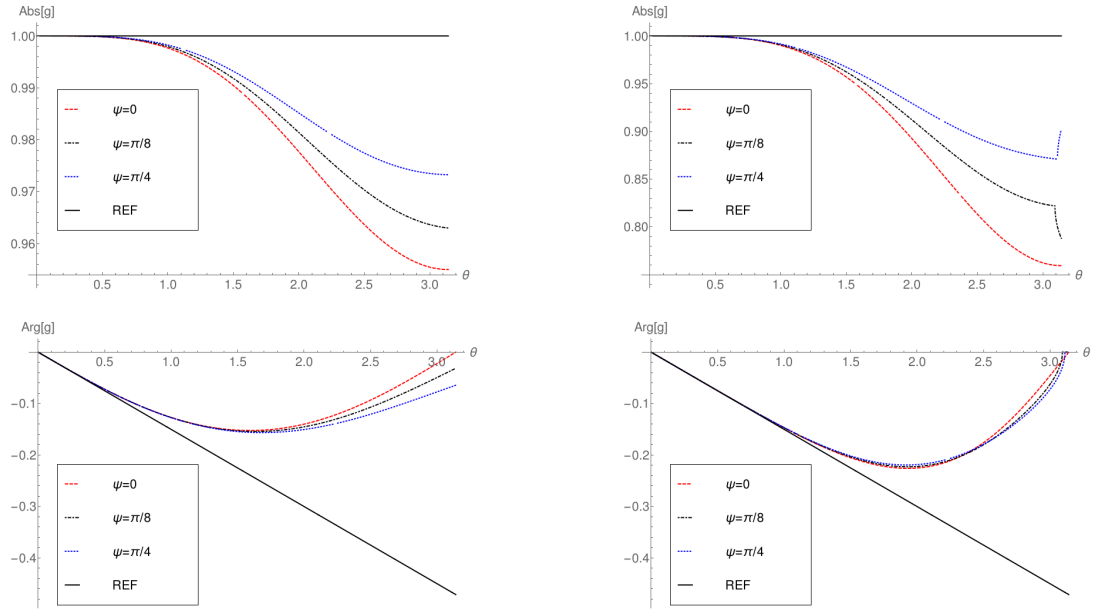

Figure 4.3: The amplification (top) and phase (bottom) relationships for the propagation directions $\psi = 0$, $\psi = \pi/8$, and $\psi = \pi/4$ are plotted for the second-order VPLW2 method and the VPFCTO3-FUP method when $\nu = 0.15$.

See Figures 4.3, 4.4, and 4.5. In all cases, the improvements in the dispersion properties brought about by the third-order methods are notable when compared to those belonging to the VPLW2 method. This is very encouraging. Interestingly, the FUP based method has more isotropic phase behavior than the LPE based one. The LF based method has enormous phase error near $\theta = \pi$, but more damping than the other methods to help mitigate its effects. However, numerical experiments did show that this method was inferior to the others for small CFL numbers and so it will not be discussed further.

4.3 Numerical Results

The VPFCTO3-FUP and VPFCTO3-LPE methods were implemented and a convergence analysis was performed using the smooth test problem found in [55]. In this

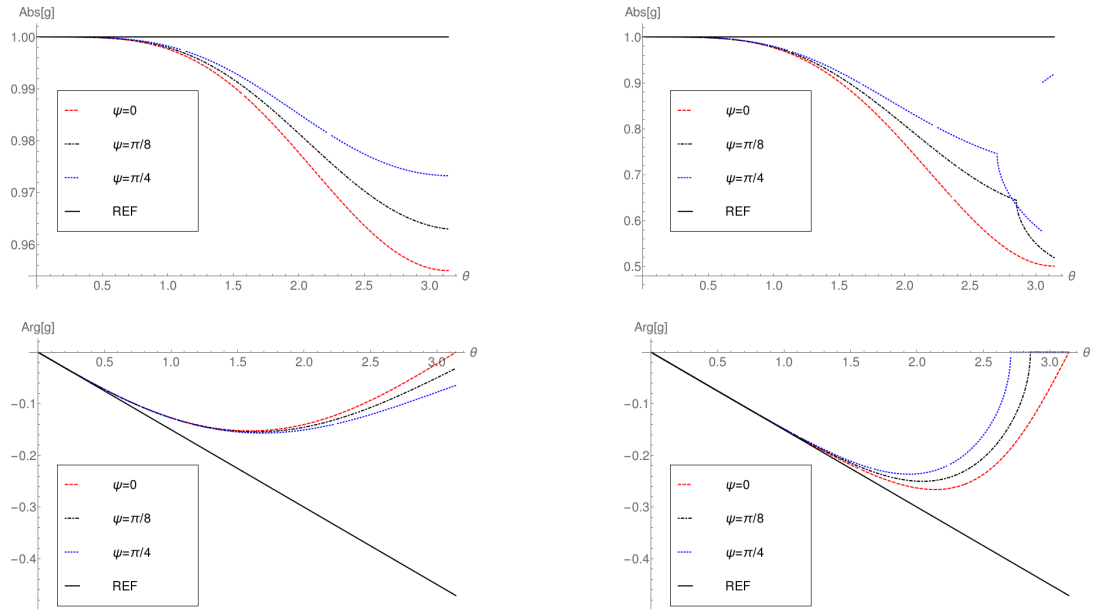

Figure 4.4: The amplification (top) and phase (bottom) relationships for the propagation directions $\psi = 0$, $\psi = \pi/8$, and $\psi = \pi/4$ are plotted for the second-order VPLW2 method and the VPFCTO3-LPE method when $\nu = 0.15$.

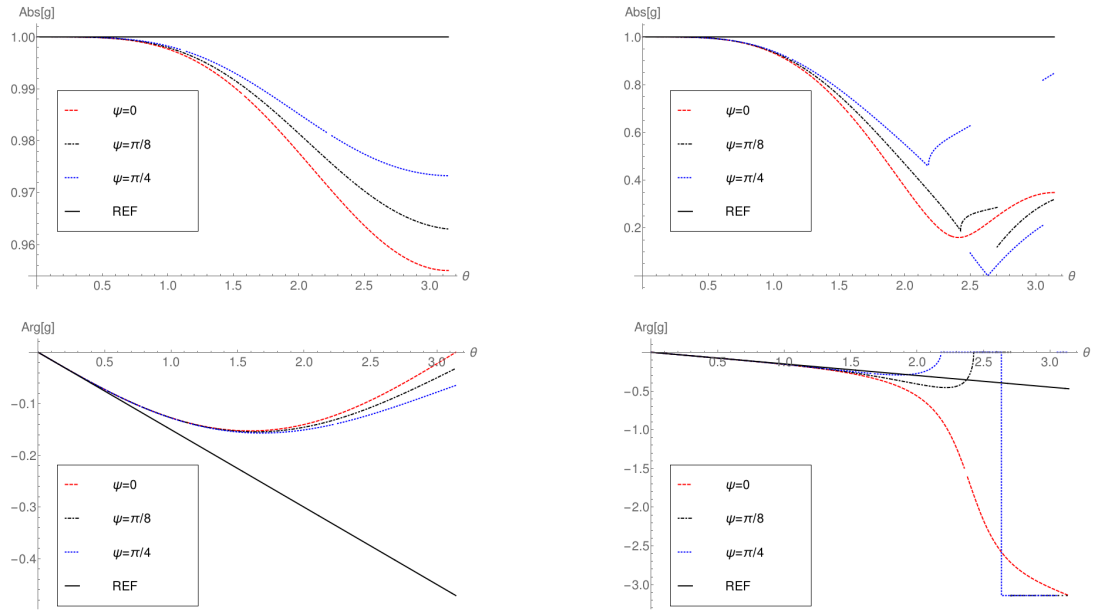

Figure 4.5: The amplification (top) and phase (bottom) relationships for the propagation directions $\psi = 0$, $\psi = \pi/8$, and $\psi = \pi/4$ are plotted for the second-order VPLW2 method and the VPFCTO3-LF method when $\nu = 0.15$.

problem a smooth periodic pressure distribution is applied to a fluid at rest. The initial conditions were defined on the domain $x \in [-1, 1]$ and $y \in [-1, 1]$ and set to

$$p(x, y, 0) = -\frac{1}{a_0}(\sin 2\pi x + \cos 2\pi y), \quad u(x, y, 0) = v(x, y, 0) = 0, \text{ and } \quad a_0 = 1.$$

The exact solution is then

$$p(x, y, t) = -\frac{1}{a_0} \cos 2\pi a_0 t (\sin 2\pi x + \cos 2\pi y),$$
$$u(x, y, t) = -\frac{1}{a_0} \sin 2\pi a_0 t \cos 2\pi x \quad \text{and} \quad v(x, y, t) = -\frac{1}{a_0} \sin 2\pi a_0 t \cos 2\pi y.$$

Computations were run until $t = 0.375$ with $\nu = 0.6$. The average error in each cell was measured according to the $L2$ error norm. Figure 4.6 shows that both of the methods converged at the design rate. The slight differences observed in the convergence of the methods are due to fourth-order effects, where the different choices of q_C produce different behavior.

Solutions for the discontinuous test problem from the previous chapter were computed to assess the performance of the third-order methods on high frequency data. Figures 4.7 and 4.8 show solutions computed with $\nu = 0.6$ and $\nu = 0.15$ with the VPFCTO3-FUP and VPFCTO3-LPE methods, respectively. Overall, the solutions are in good agreement with the reference solution and do not contain excessive spurious features, even though no limiters were used. The scheme derived using the LPE method is very comparable to the one derived using the FUP method at the higher CFL number, but the LPE based scheme would require less limiting at $\nu = 0.15$.

The goal of the chapter was improve the poor performance of the VPLW2 method for high frequency data, especially at low CFL numbers. Figure 4.9 shows velocity magnitude solutions from the discontinuous test problem for the VPFCT03-FUP method and the VPLW2 method at $\nu = 0.6$ and $\nu = 0.15$. The results clearly show this goal has been achieved. At both CFL numbers, the third-order solution is much

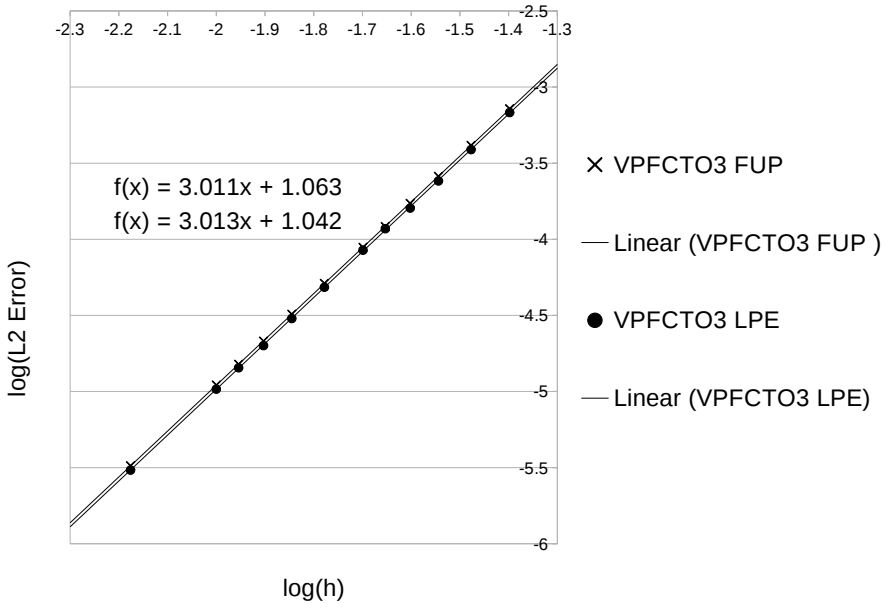

Figure 4.6: Both of the VPFCTO3 variants converge at third-order as expected. The experiments were run on meshes from 50×50 to 300×300 with a CFL number of 0.6 and the plots show the average L2 error norm for the u-velocity.

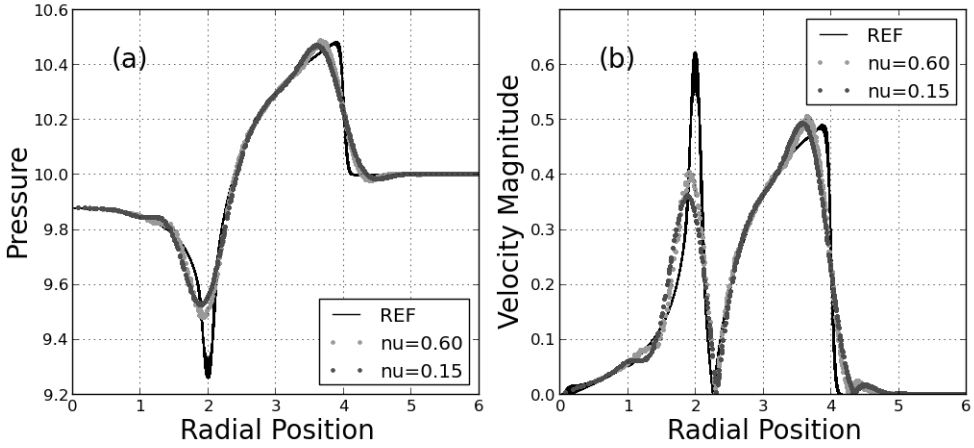

Figure 4.7: Results produced by the third-order VPFCTO3-FUP method with no limiter for the discontinuous problem are promising as few spurious features are present. (a) Pressure; (b) Velocity Magnitude

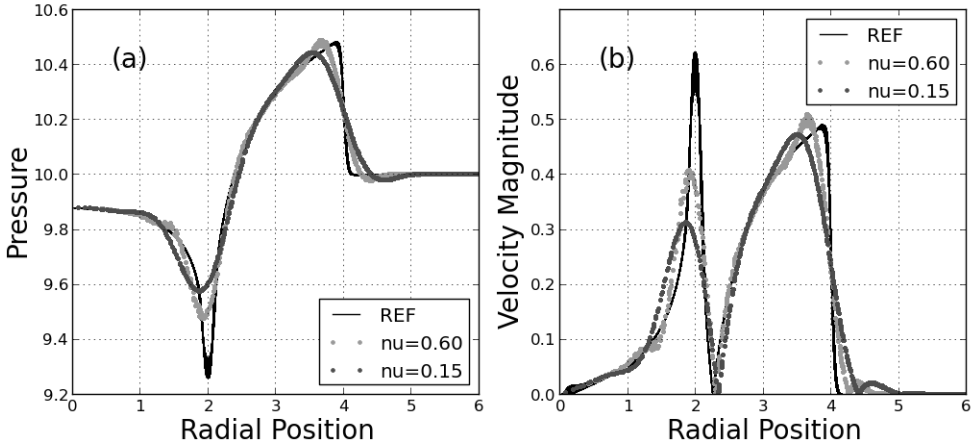

Figure 4.8: The performance of the VPFCTO3-LPE method with no limiter for the discontinuous problem was very similar to that observed with the FUP based method. (a) Pressure; (b) Velocity Magnitude

better than the second-order one. Fewer spurious features exist and the resolution of the physical waves is improved. The improvements in the solution when $\nu = 0.15$ are particularly noteworthy. Figure 4.10 displays the same information for the LPE based scheme, and the same observations apply. Again, the small differences in the solutions are due to fourth-order effects.

This section is concluded with results from the Gaussian pulse problem introduced in Chapter III. The solutions here were computed on a very coarse 50×50 mesh. Results from the FUP based method are shown in Figure 4.11 and results from the LPE based method are shown in Figure 4.12. Once again, the third-order methods produced solutions that were far superior to those obtained with the VPLW2 method. It is paradoxical that the LPE based method exhibited larger phase errors than the FUP based method at $\nu = 0.15$. However, this shows that one-dimensional analysis does not always translate well to higher dimensions. The FUP variant clearly outperformed the LPE one on this problem.

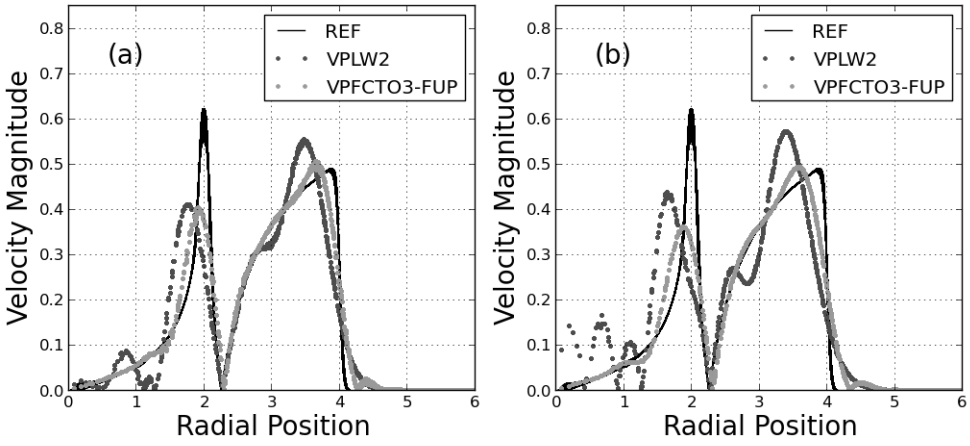

Figure 4.9: Results produced by the third-order VPFCTO3-FUP method for the discontinuous problem are clearly improved over those obtained with the second-order VPLW2 method. (a) $\nu = 0.6$; (b) $\nu = 0.15$

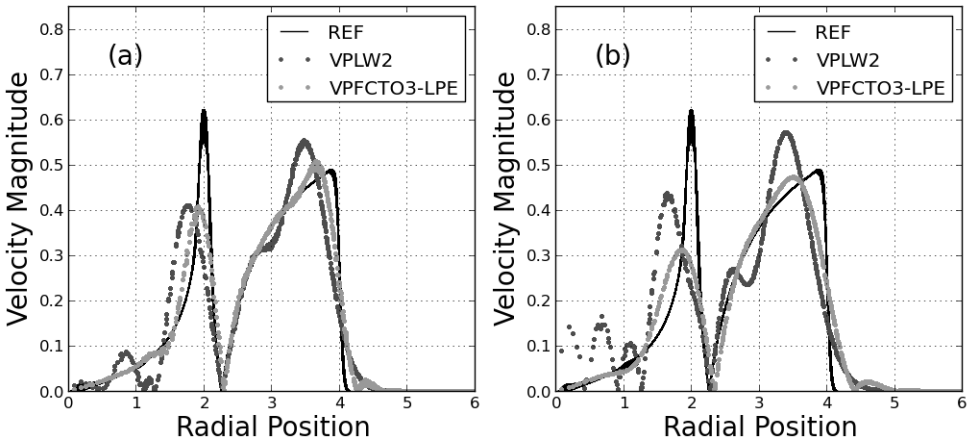

Figure 4.10: Results produced by the third-order VPFCTO3-LPE method for the discontinuous problem are clearly improved over those obtained with the second-order VPLW2 method. (a) $\nu = 0.6$; (b) $\nu = 0.15$

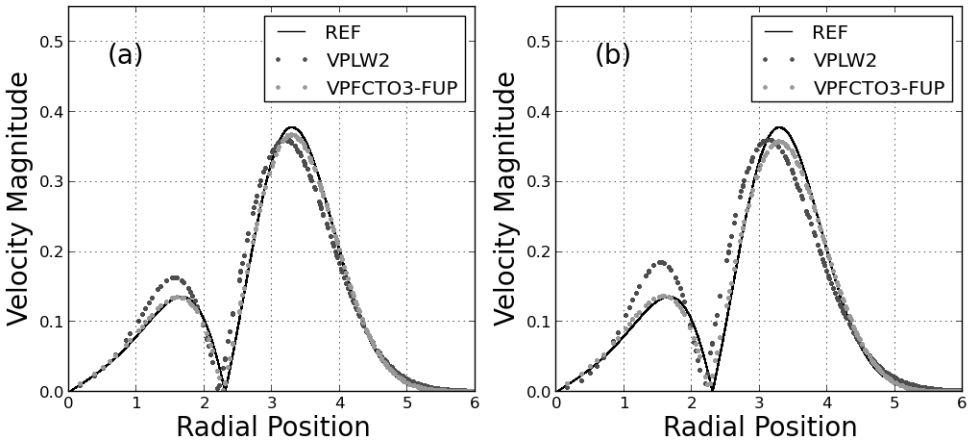

Figure 4.11: Results produced by the third-order VPFCTO3-FUP method for the smooth test problem on a very coarse 50×50 mesh show the improvements in resolution and phase errors due to the increased accuracy. (a) $\nu = 0.6$; (b) $\nu = 0.15$

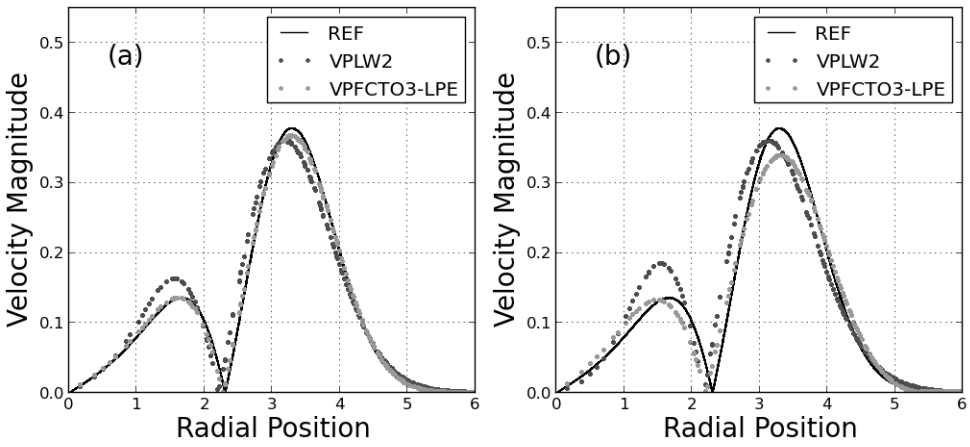

Figure 4.12: Results produced by the third-order VPFCTO3-LPE method for the smooth test problem on a very coarse 50×50 mesh clearly show the improvements in resolution and phase errors over the second-order VPLW2 method. However, at $\nu = 0.15$, the solution is noticeably inferior to the one produced by the VPFCTO3-FUP method. (a) $\nu = 0.6$; (b) $\nu = 0.15$

4.4 Discussion

Vorticity preserving methods have been developed that take advantage of the FCT update procedure to obtain third-order accuracy. By using the provisional solution that is already computed during a FCT update, the increased accuracy was obtained in an inexpensive and compact manner. The costs associated with the increased accuracy are the computation of the provisional drivers and a decrease in the stability limit of $20-25\%$. However, the dispersion characteristics of the third-order methods were far superior to the VPLW2 method, offsetting the extra expense. The increased performance was particularly impressive at low CFL numbers.

One way to view the antidiffusive fluxes for the third-order methods is that they are composed of one part that is responsible for second-order accuracy and then a correction term that increases the accuracy to third-order. Of course, the third-order corrections contain the provisional drivers, β^*. In the next chapter, flux limiters will be proposed that mimic the third-order corrections developed here.

CHAPTER V

Temporal Flux Limiting

Chapter IV introduced two third-order accurate vorticity preserving methods that had much better phase and damping characteristics than the second-order VPLW2 method from Chapter III. In this chapter, nonlinearity will be added to the antidiffusive fluxes from the third-order methods to produce a vorticity preserving flux-corrected transport (VPFCT) scheme that is free from spurious features. The desire is for the VPFCT method to converge at second-order in smooth regions of the flow, but still possess the enhanced dispersion properties of the third-order methods. Second-order convergence will be considered acceptable since some of the limiting tools developed here will be applied to the more complex Lagrangian hydrodynamics problem. When nonlinear physics and moving meshes must be dealt with, producing formal third-order accuracy becomes much more complicated. In this case it is probably more practical to impart the desired behavior to an algorithm that is formally second-order accurate. The chapter begins with a general review of limiting in Section 5.1. In Section 5.3, nonlinear functions are developed that modify the antidiffusive drivers from Chapter IV. Numerical results are presented in Section 5.4 and the chapter concludes with some discussion in Section 5.5.

5.1 Limiting Review

Limiting refers to the process of incorporating nonlinearity into better than first-order numerical methods to prevent spurious features from entering the solution. The need for limiters in numerical methods for hyperbolic equations was recognized early on by von Neumann and Richtmyer, though not fully understood. They devised a nonlinear artificial viscosity [88] that added numerical dissipation to the solution in regions with steep gradients, which allowed them to successfully capture shocks. Their rationale for doing so was that discontinuities in the solution must be smeared out so that the solution derivatives and, therefore, finite differences were defined everywhere. In 1959, Godunov proved that linear schemes which do not admit spurious overshoots must be first-order accurate [35]. Researchers were then able to understand that nonlinear methods were the key to constructing useful higher-order methods. In general, three different approaches emerged: artificial viscosity, slope limiting, and flux limiting.

Artificial viscosity methods add nonlinear viscosity like terms to the finite difference equations to increase dissipation in regions of flow with steep gradients. The use of artificial viscosity is still very prevalent in the shock hydrodynamics community, though it is often discussed in the context of removing mesh instabilities or achieving numerical stability. Slope limiting methods were first developed by van Leer in [85] and [86]. He used Godunov's first-order method as a starting point and increased its accuracy by reconstructing the solution inside cells. The solution profile in each cell was obtained from nonlinear combinations of neighboring reconstructions which were carefully chosen to ensure that new extrema were not introduced. This method became known as the monotone upstream-centered scheme for conservation laws (MUSCL). The basic idea of reconstructing the solution data locally and nonlinearly modifying the reconstructions to ensure monotonicity has formed the basis for many families of schemes which include the ENO [39], and WENO [52] methods.

Flux limiting methods can be subdivided into two categories, those that are derived from the multiple step FCT method pioneered by Boris and Book [8] and single step procedures that trace their origins to the works of van Leer [84], Harten [38], Roe (see e.g. [79]), Sweby [79] and others. In either case, flux correction methods use a flux decomposition

$$F_{AD} = F_A - F_C,$$

where an antidiffusive (AD) flux is defined as the difference between accurate (A) and cautious (C) ones. The cautious flux should produce a first-order scheme that will not admit spurious extrema. Then the antidiffusive fluxes are reduced using a nonlinear operation just enough to prevent unphysical features from appearing. The success of these methods hinges on the limiting step. In FCT, a multistep update procedure is used that advances the solution in three steps: transport, diffusion, and antidiffusion. Often the transport and diffusion operators are applied simultaneously in the form of a first-order method. Then limiting is applied to the antidiffusive fluxes and the final antidiffusive update is performed. Note that FCT methods generate a provisional solution, which may be incorporated into the antidiffusion step to yield a final method that differs from the one described by the accurate flux alone. In contrast, single step flux correction methods do not compute an intermediate provisional solution and will always recover the scheme described by the accurate flux if no limiting is applied.

Originally, limiting mechanisms were devised for solving one-dimensional, scalar conservation laws. In that context, researchers were afforded the luxuries of simple geometry and simple equations and they made good use of both. Of particular utility was the concept of the total variation (TV) of a solution

$$TV = \int \left| \frac{\partial u}{\partial x} \right| dx$$

and the design of schemes which were guaranteed to be total variation diminishing

(TVD) [38]. Such schemes could be derived by enforcing appropriate local bounds on the solution, which were known *a priori*, via a limiter. However, problems were immediately encountered when trying to extend these ideas to more difficult problems. There is no rigorous TVD or monotonicity principle for coupled systems or nonlinear problems. In addition, geometric considerations in higher dimensions are non-trivial and simple directional splitting schemes did not always work well, particularly in regard to FCT. Zalesak was the first to propose a multidimensional flux limiter for FCT [93], but it still enforced a local maximum principle which, strictly speaking, is not physical for many problems of interest. Many multidimensional MUSCL type techniques have subsequently been developed, usually along the lines of the limiter proposed by Barth and Jespersen [5; 69].

Despite the success of the flux limiting and slope limiting methods, there is more work to be done. Inadequate limiting techniques are holding back the widespread use of next generation methods such as DG [45]. These methods require compact and multidimensional limiters that can affect subtle changes in order to prevent unphysical extrema while preserving high-order accuracy in smooth regions. Much of the difficulty is related to the fact that no concrete answers exist to three fundamental questions about the general limiting problem: What quantities should a limiter act on, how should the limiting mechanism be designed, and how much limiting should be applied? This chapter proposes some answers to these questions in the form of a temporal flux limiting approach that will be used to construct a FCT implementation of the VPFV2 method from Chapter III.

5.2 A Vorticity Preserving Flux-corrected Transport Scheme

The goal of this chapter is to assemble a VPFCT method that resists mesh imprinting like the second-order VPFV2 method presented in Chapter III and has the improved dispersion properties of the third-order methods from Chapter IV, but does

not admit any spurious features into the solution. The update procedure will be implemented in three steps. First, a provisional first-order solution will be computed in the same fashion as shown in the last chapter. The same two first-order methods will be considered: FUP ($q_C = \nu$) and LPE ($q_C = 1/3 + 2\nu^2/3$). Second, the enhanced drivers β^E will be modified via a limiting mechanism to become β^{lim}. Third, the final antidiffusive update will be performed according to

$$\boldsymbol{U^{n+1}} = \boldsymbol{U^*} + \frac{\Delta t}{h}[\mu_y \delta_x \hat{\boldsymbol{F}}_{\boldsymbol{AD}} + \mu_x \delta_y \hat{\boldsymbol{G}}_{\boldsymbol{AD}} + \delta_x \tilde{\boldsymbol{F}}_{\boldsymbol{AD}} + \delta_y \tilde{\boldsymbol{G}}_{\boldsymbol{AD}}], \tag{5.1}$$

where the antidiffusive fluxes are now

$$\hat{\mathbf{F}}_{AD} = \begin{pmatrix} \frac{q^+ h}{4\Delta t}\hat{\beta}_u^{lim} \\ \frac{q^+ h}{2\Delta t}\hat{\beta}_p^{lim} \\ 0 \end{pmatrix}, \qquad \hat{\mathbf{G}}_{AD} = \begin{pmatrix} \frac{q^+ h}{4\Delta t}\hat{\beta}_v^{lim} \\ 0 \\ \frac{q^+ h}{2\Delta t}\hat{\beta}_p^{lim} \end{pmatrix},$$

$$\tilde{\mathbf{F}}_{AD} = \begin{pmatrix} \frac{q^+ h}{4\Delta t}\tilde{\beta}_u^{lim} \\ 0 \\ 0 \end{pmatrix}, \text{ and} \qquad \tilde{\mathbf{G}}_{AD} = \begin{pmatrix} \frac{q^+ h}{4\Delta t}\tilde{\beta}_v^{lim} \\ 0 \\ 0 \end{pmatrix}.$$

The critical step in the process is the computation of the limited drivers β^{lim} and the development of a limiting mechanism is the focus of the rest of the chapter. Note that, as shown by Morton and Roe [66], the fluxes for the velocity updates must be limited at vertices to retain the vorticity preservation property. This will be done here and the limiting for the pressure update will also be performed on the point estimates. As a reminder, the relevant driver quantity definitions are contained in Table 5.1.

Table 5.1: Driver definitions for the VPFCT Scheme

Flux Component	Driver Quantity
$\hat{p}$	$\mu_y\delta_x u + \mu_x\delta_y v$
$\hat{u}$	$\mu_y\delta_x p$
$\hat{v}$	$\mu_x\delta_y p$
$\tilde{u}$	$\delta_x p$
$\tilde{v}$	$\delta_y p$

5.3 Flux Limiting

Recall from the last chapter that the enhanced drivers from the third-order methods take the form

$$\beta^E = \beta^n + \kappa(\beta^* - \beta^n). \tag{5.2}$$

The first-term is the responsible for second-order accuracy and the second term, which contains the provisional driver, is a correction that increases the accuracy to third-order. This means that the second term is responsible for the vastly improved dispersion properties associated with the third-order methods. The improved dispersion relationships take care of most of the work needed to successfully limit the second-order VPFV2 method and, therefore should be disturbed as little as possible by any modifications that are introduced. On the other hand, the second-order antidiffusive flux causes undesirable behavior in the presence of steep gradients and should be aggressively modified in those regions. To accomplish these tasks a smoothness indicator ϕ is introduced and used to write a flux correction procedure of the form

$$\beta^E \rightarrow \beta^{lim} = F_0(\phi,\nu)\beta^n + F_1(\kappa,\phi,\nu)(\beta^* - \beta^n). \tag{5.3}$$

The job of the smoothness monitor is to highlight regions of the flow that need limiting. Then the functions F_0 and F_1 control how much limiting is applied. In smooth regions of the flow, the functions should approach $F_0 = 1$ and $F_1 = \kappa$ in order to return fluxes as close as possible to 5.2.

Before the limiting function can be determined, a suitable definition for ϕ must be specified. Traditionally, smoothness monitors are constructed from the ratio of neighboring spatial gradients. For example, van Leer's original definition [84], for a generic quantity α, was

$$\phi_{VL} = \frac{\Delta_{j-k/2}\alpha}{\Delta_{j+k/2}\alpha},$$

where $\Delta_{j+1/2}() = ()_{j+1} - ()_j$. Here a different strategy is adopted. The smoothness of the solution will be assessed by looking at temporal expansions of the fluxes. Take, for example, the pressure which can be expanded through second-order as

$$p^{n+1} = p^n + \Delta t p_t^n + \frac{\Delta t^2}{2} p_{tt}^n.$$

For the acoustic system, this can be written in terms of spatial derivatives as

$$p^{n+1} = p^n - \Delta t a_0 \frac{\beta_p^n}{h} - \frac{\Delta t^2}{2}(a_0 \frac{\beta_p^n}{h})_t.$$

To proceed, it was reasoned that in regions of the flow where limiting is needed the higher-order terms in the temporal expansion will become large when compared with the lower-order ones. Therefore, a smoothness monitor was constructed by looking at the ratio of the second and third terms in the series. The resulting definition for ϕ_p is

$$\phi_p = \left| \frac{\frac{\Delta t^2}{2}(a_0 \beta_p^n)_t}{a_0 \Delta t \beta_p^n} \right| = \left| \frac{\Delta t (\beta_p^n)_t}{2\beta_p^n} \right| \approx \left| \frac{\beta_p^* - \beta_p^n}{2\beta_p^n} \right|. \tag{5.4}$$

Indicator quantities are defined for each individual point flux by plugging the proper driver definitions into 5.4.

The smoothness monitor will highlight regions where limiting is needed, but the amount of limiting applied will depend on the form of the functions F_0 and F_1 in 5.3. Since, a precise answer to the question of how much limiting should be applied is not available for general problems, some empiricism will be relied upon. Starting out it

was hoped that the function F_1 could simply return κ and only the second-order flux would need to be limited. Consider first, then, F_0. In general, the function should approach zero as ϕ approaches one. Experiments have shown that a function of the form

$$F_0(\phi, \nu) = \max\left[0, 1 - f(\nu)\phi\right] \tag{5.5}$$

produces the desired behavior. The part of the antidiffusive flux responsible for second-order accuracy is reduced as the third term in the temporal expansion approaches the same magnitude as the second one. In practice, something must be done when the initial driver is zero. In this case, F_0 should evaluate to zero. Either a local extremum is present in the flux quantity and the first-order method should be relied upon or nothing is happening in the solution and there is no difference between the first-order flux and the high-order one. The CFL number dependence introduced by $f(\nu)$ should vary depending on the first-order scheme and this will be determined later.

Numerical experiments were run using the the definition of F_0 described above in conjunction with various forms of $f(\nu)$ while leaving the third-order corrections unmodified ($F_1 = \kappa$). Unfortunately, F_0 was unable to remove a small spurious wave from the discontinuous test problem that was located in front of the discontinuity. Thinking back to Chapter IV, this wave appeared when the antidiffusive fluxes were modified for third-order accuracy, and so it was inferred that some modification to the third-order corrections was necessary. The spurious feature was only produced at the discontinuity. Therefore, it was hypothesized that F_1 should only deviate from κ in the most demanding instances. Some additional experimentation revealed that the function

$$F_1(\kappa, \phi, \nu) = \kappa \min\left(\frac{f(\nu)}{\phi}, 1\right) \tag{5.6}$$

successfully removed the spurious wave. The use of $1/\phi$ ensured that F_1 reacts only

to the most challenging data. In the numerical implementation, if $\beta^* - \beta^n = 0$, then $F_1 = 1$, but the value is of no consequence as the third-order corrections are zero anyway. In both F_0 and F_1, the functions $f(\nu)$ depend on the choice of first-order scheme. Here the functions are taken to be

$$f(\nu)|^{FUP} = \frac{3(1-\nu)}{2} \quad \text{and} \quad f(\nu)|^{LPE} = \frac{3\nu}{2}.$$

As a result of basing the structure of the limiter off the third-order fluxes, point estimates of u and v at a given vertex or face are limited by different amounts. In contrast, only one limited pressure is computed at each vertex. Note that the velocity divergence is not split into individual velocity gradients when limiting the pressure. However, the velocity vector components may be treated individually by considering the pressure gradient components. As a result, no difficulty arises when limiting vector quantities or multidimensional scalars. More complicated mechanisms for vector limiting, such as the one in [56], are avoided.

To summarize, a temporal flux limiting mechanism has been constructed by mimicking the fluxes from the VPFCTO3 methods. The temporal structure of the limiter is due to the use of the provisional first-order solution and smoothness monitors obtained from temporal expansions of the fluxes. The resulting limiter mechanism is genuinely multidimensional and can be thought of as zero-dimensional with respect to space. Furthermore, no *a priori* bounds are placed on the solution.

It should be mentioned briefly that others have used future information when designing limiters before. Two examples will be highlighted. First, the multi-dimensional optimal order detection (MOOD) method proposed in [24] uses a provisional solution. There an unlimited high-order step is taken and the order of the method is then locally reduced in a iterative process until some smoothness criterion has been satisfied. Second, Duraisamy and Baeder constructed temporal limiters for use with

implicit schemes in the method-of-lines framework [31]. While their schemes are very different than the ones considered here, they used MUSCL type reconstructions of the solution in time in conjunction with standard spatial reconstructions to prevent spurious oscillations, even when large time steps were taken. Finally, the idea of building limiters that mimic third-order methods and have CFL number dependence is also not new. For example, see Arora and Roe [2].

5.4 Numerical Results

Numerical results from the discontinuous test problem computed with the VPFCT methods were compared with unlimited VPFV2 solutions to judge the effectiveness of the limited drivers at removing spurious features from the solution. Figures 5.1 and 5.2 show results for the VPFCT-FUP and VPFCT-LPE methods, respectively. The CFL number was $\nu = 0.6$. In both cases, the limited fluxes remove the spurious features from the solution and produce a result that is free of mesh imprinting. As shown in Appendix B, the deviations from radial symmetry produced by the original LW method have been reduced by over 85 percent.

The VPFCT-FUP and VPFCT-LPE methods were compared to an unsplit MUSCL-H method that used the MinMod and Superbee slope limiting functions. In all cases, the CFL numbers were chosen to correspond to the same fraction of the theoretical maximum for each scheme. For example, 0.8 of the theoretical limit corresponds to $\nu = 0.6$ for the VPFCT methods and $\nu = 0.4$ for the MUSCL-H methods. Figure 5.3 shows convergence results for all four schemes at these CFL numbers. The results were obtained using the smooth exact solution described in Chapter IV. The VPFCT methods have smaller absolute errors than the MUSCL-H methods by nearly one order of magnitude and they converge at second-order. Figure 5.4 shows the same analysis, except that the computations were run at $\nu = 0.15$ for the VPFCT methods and $\nu = 0.1$ for the MUSCL-H methods. The convergence rates of the VPFCT

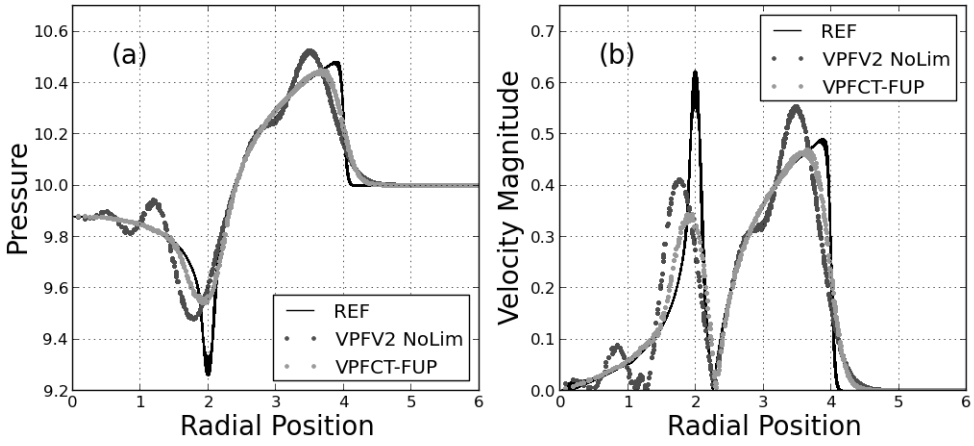

Figure 5.1: The limited drivers in the VPFCT-FUP method remove all of the spurious features from the unlimited VPFV2 method and improve the phase accuracy of the physical waves.

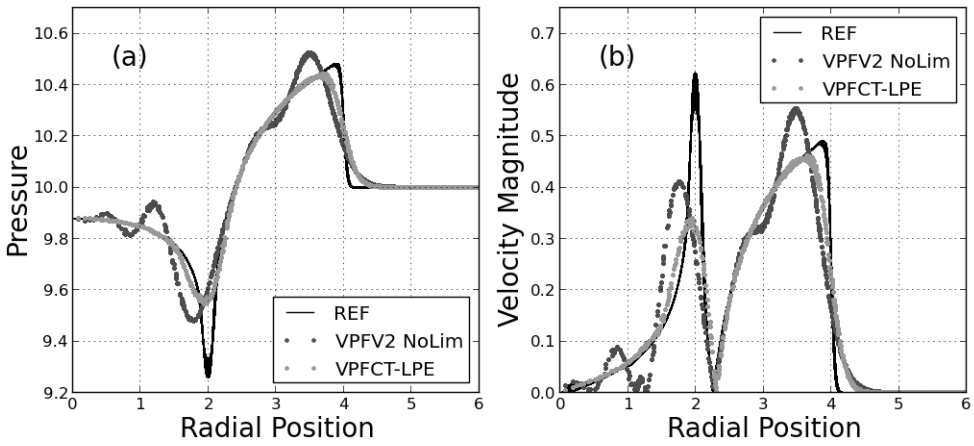

Figure 5.2: The limited drivers in the VPFCT-LPE method remove all of the spurious features from the unlimited VPFV2 method and improve the phase accuracy of the physical waves.

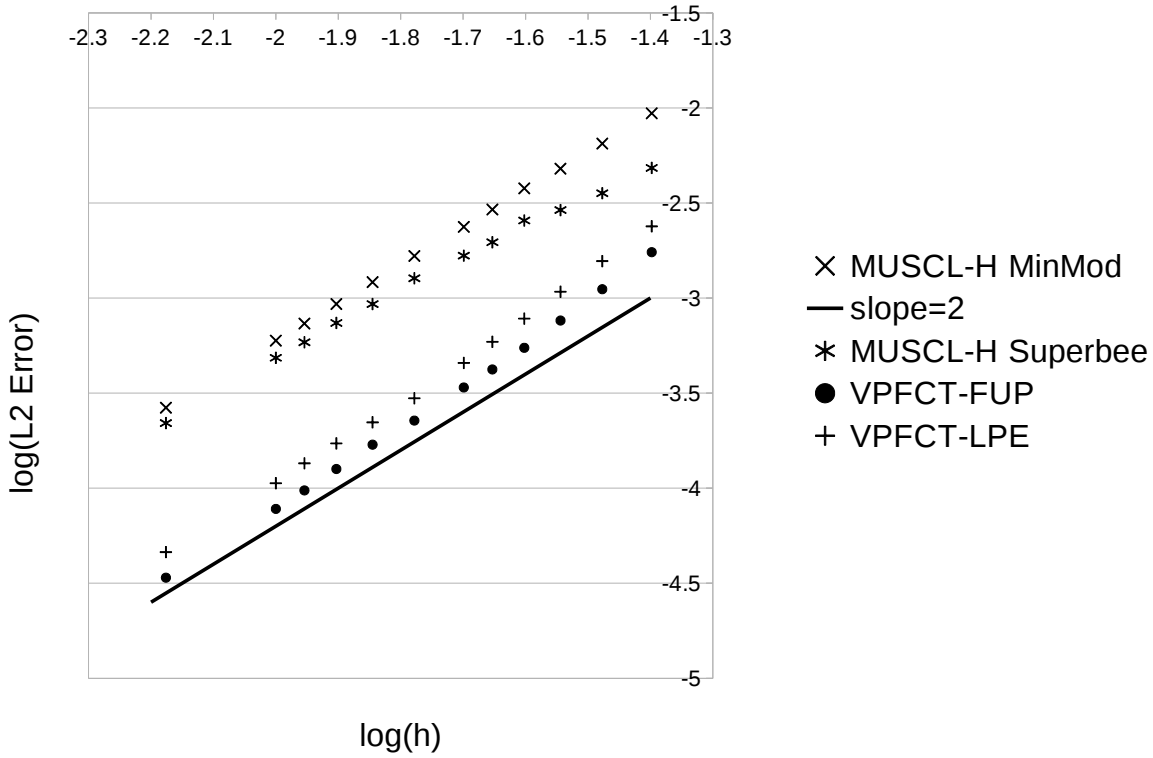

Figure 5.3: The VPFCT methods ($\nu = 0.6$) have small absolute errors as compared to the MUSCL-H method ($\nu = 0.4$) when either the Superbee or MinMod slope limiters are used and converge at second-order

methods improved in this case. In fact, the VPFCT-LPE method converged at nearly third-order as $f(\nu)|^{LPE}$ was able to relax.

Figures 5.5 and 5.6 compare solutions from the discontinuous problem obtained with the VPFCT-FUP and VPFCT-LPE methods to the Superbee and MinMod limited MUSCL-H algorithms. The VPFCT methods clearly best the MinMod limited MUSCL-H method and approach the resolution of the Superbee limiter while producing solution that are free from mesh imprinting. Figures 5.7 and 5.8 compare the schemes at the lower CFL numbers. The same comments apply, except that the

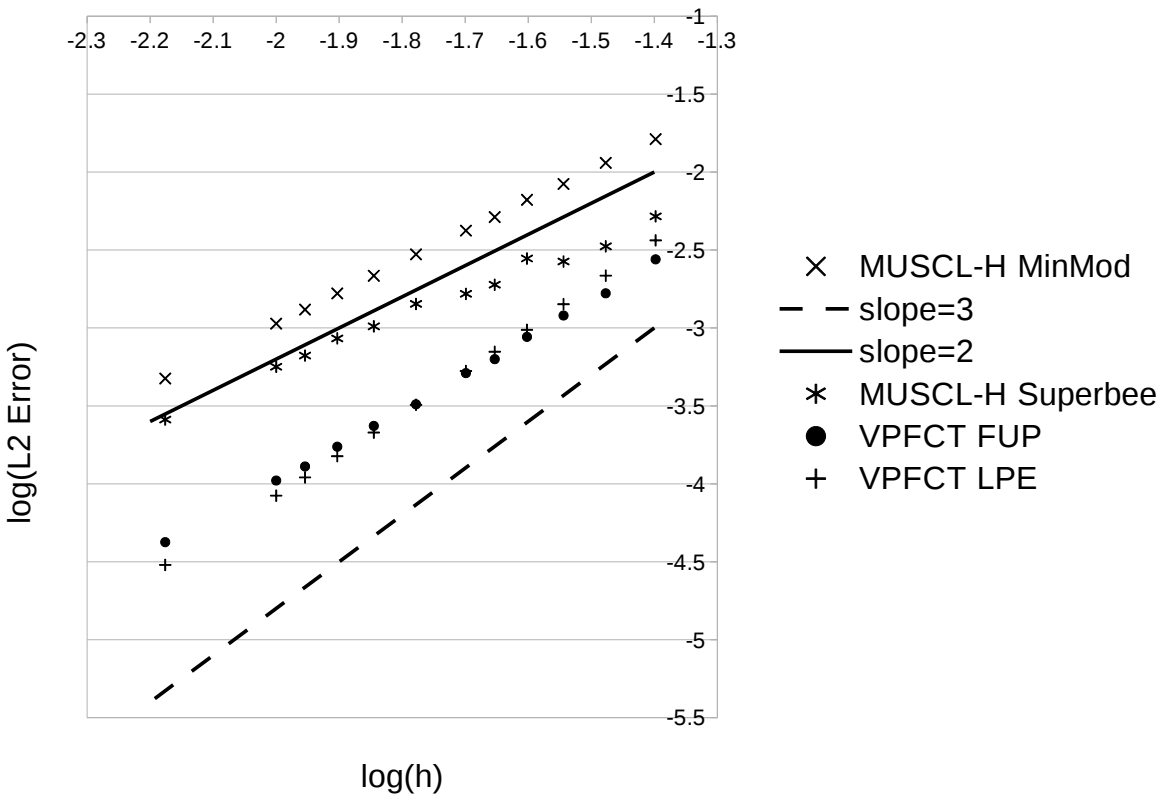

Figure 5.4: The convergence rates of the VPFCT methods ($\nu = 0.15$) improved when the CFL number was lowered. The VPFCT-LPE scheme converged near third-order as the function $f(\nu)|^{LPE}$ relaxed.

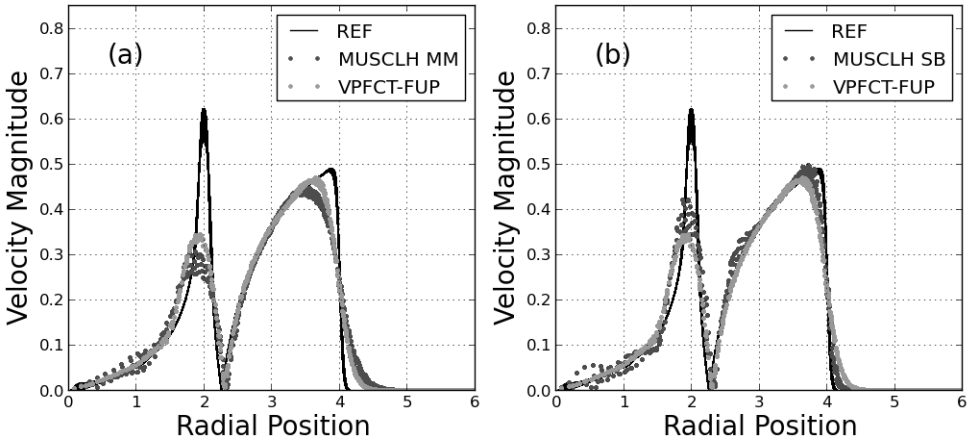

Figure 5.5: Results from the discontinuous test problem: (a) VPFCT-FUP method ($\nu = 0.6$) and the MUSCL-H method ($\nu = 0.4$) with the MinMod limiter (b) VPFCT-FUP method ($\nu = 0.6$) and the MUSCL-H method ($\nu = 0.4$) with Superbee limiter

Superbee solutions are notably sharper when compared with the LPE based method.

Solutions to the discontinuous test problem were computed with the first-order VPFV2-FUP method, unlimited second-order VPFV2 method, and limited VPFCT-LPE method and plotted on the same set of axes. See Figure 5.9. Results are shown for two different meshes: 100×100 and 300×300. The CFL number was $\nu = 0.6$. Comparing the solutions obtained with the ingredients that make up the VPFCT-FUP method to the final solution helps illustrate how the limiter functions. No features are present in the final solution that are no found in the cautious scheme, but the accuracy is dramatically improved. The limiter is able to identify problem areas in the second-order solution and correct them while incurring minimal damage to the physical waves. The limited drivers steepen the discontinuous fronts and remove some phase errors due to the third-order like behavior.

The ability of the VPFCT scheme to preserve physical vorticity will be demonstrated with a new test problem that combines a steady vortical flow with the two-dimensional unsteady interaction of four planar pressure waves. The initial velocity

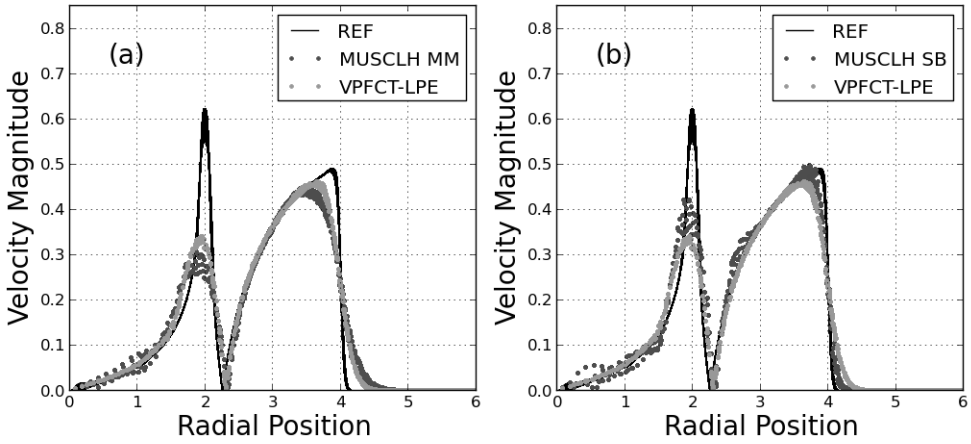

Figure 5.6: Results from the discontinuous test problem: (a) VPFCT-LPE method ($\nu = 0.6$) and the MUSCL-H method ($\nu = 0.4$) with the MinMod limiter (b) VPFCT-LPE method ($\nu = 0.6$) and the MUSCL-H method ($\nu = 0.4$) with Superbee limiter

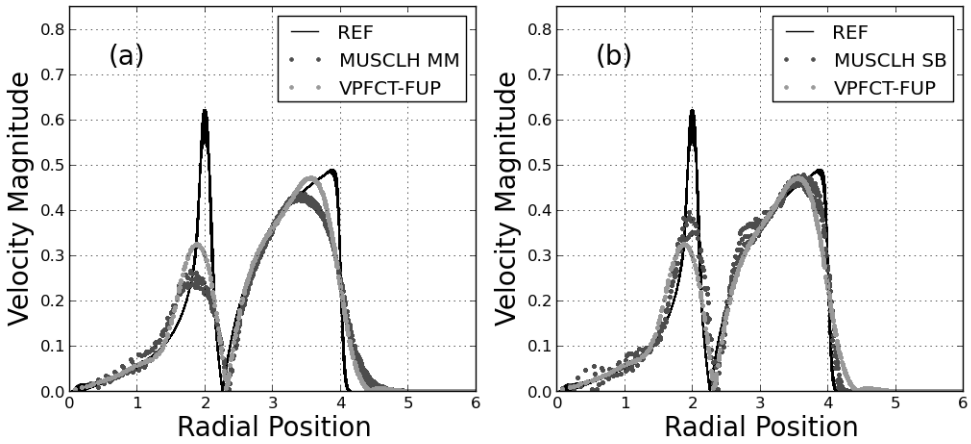

Figure 5.7: Results from the discontinuous test problem:(a) VPFCT-FUP method ($\nu = 0.15$) and the MUSCL-H method with MinMod limiter ($\nu = 0.1$) (b) VPFCT-FUP method ($\nu = 0.15$) and the MUSCL-H method ($\nu = 0.1$) with Superbee limiter

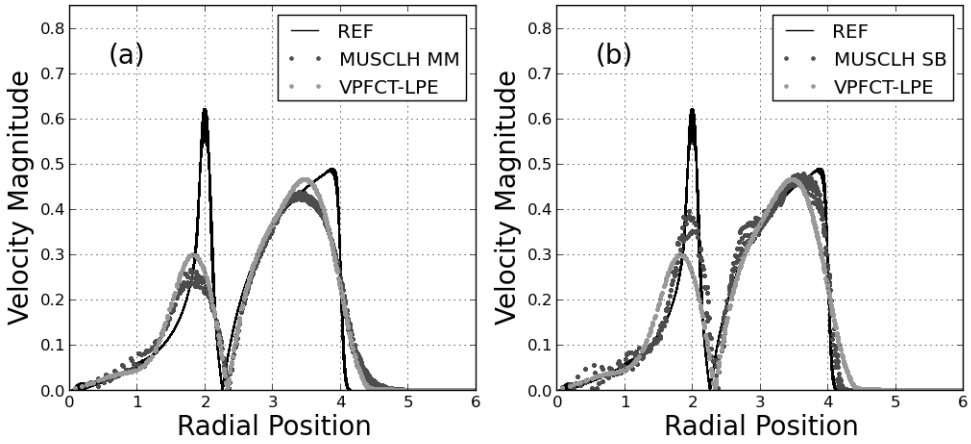

Figure 5.8: (a) The limited VPFCT-LPE method ($\nu = 0.15$) is superior to the MUSCL-H method ($\nu = 0.1$) when using the MinMod slope limiter by any measure for the discontinuous test problem. (b) The limited VPFCT-LPE method ($\nu = 0.15$) diffuses the waves more than the MUSCL-H method ($\nu = 0.1$) with the Superbee limiter on the discontinuous test problem, but is free of spurious features and mesh imprinting.

data form a modified combination vortex in which the core is prescribed as the usual solid body rotation, but the potential vortex region is replaced with a tangential velocity field that decays with the square of the radial position, r. This makes the problem more interesting by introducing additional vorticity. The divergence of the velocity field is zero and, therefore, the data represent a steady solution if a uniform pressure field is specified. However, four plane waves centered at $x = -10$, $x = 10$, $y = -10$, and $y = 10$ with magnitude two were introduced. Specifically, the initial conditions were defined on the domain $x \in [-20, 20]$ and $y \in [-20, 20]$ as

- $u(x, y, 0) = -\frac{y}{5}$ and $v(x, y, 0) = \frac{x}{5}$ if $r \leq 2$,
- $u(x, y, 0) = -\frac{16y}{10(x^2+y^2)^{3/2}}$ and $v(x, y, 0) = \frac{16x}{10(x^2+y^2)^{3/2}}$ if $r > 2$,
- $p(x, y, 0) = 4$ if $x, y \in (-10.5, -9.5)$ and $x, y \in (9.5, 10.5)$,
- $p(x, y, 0) = 2$ if $x, y \in (-10.5, -9.5)$ or $x, y \in (9.5, 10.5)$

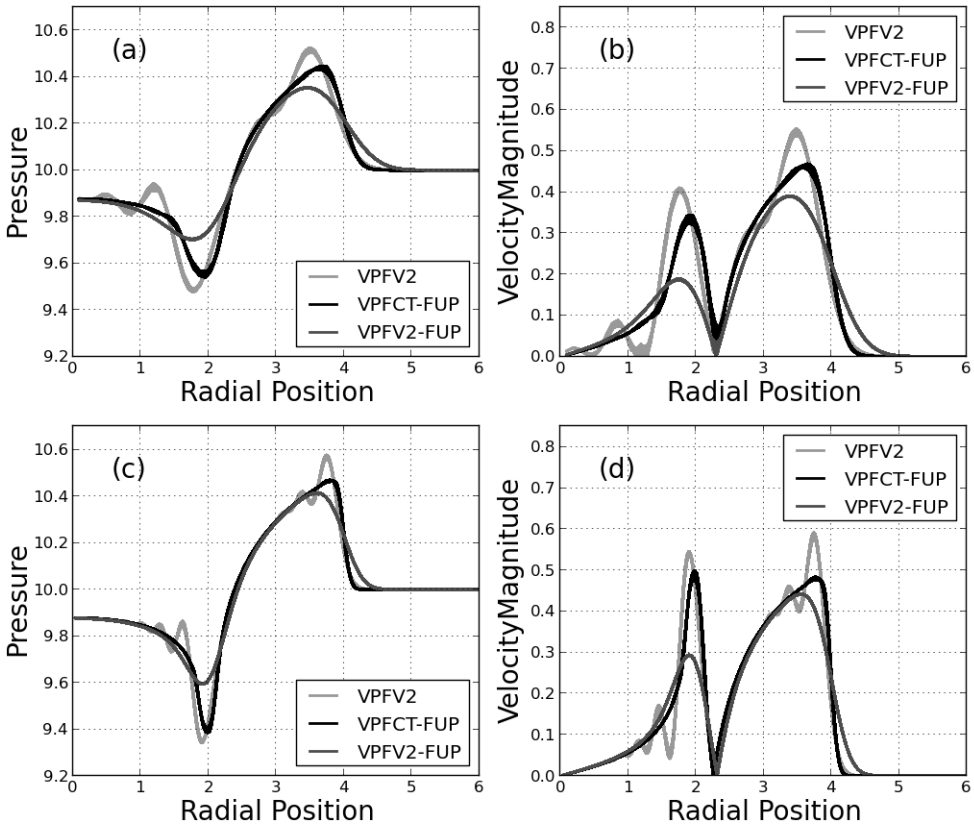

Figure 5.9: A comparison of the VPFV2-O1, unlimited VPFV2, and VPFCT schemes for the discontinuous pressure problem on two different meshes ($\nu = 0.6$, $t = 3$): (a) Pressure, 100×100 mesh; (b) Pressure, 300×300 mesh; (c) Velocity magnitude, 100×100 mesh; (d) Velocity magnitude, 300×300 mesh

- $p(x, y, 0) = 1$ otherwise and
- $a_0 = 10$ everywhere.

The plane waves split and part of each travels toward the origin. These waves influence the velocity field as they move. Figure 5.10 shows contours of the pressure and velocity magnitude at $t = 1.2$ to demonstrate the complexity of the transient flow. At $t = 2$, the left and right going wave pairs in each direction have passed through each other and arrived at the locations of the initial disturbances. The velocity field in the inner region should have returned to its initial state. Figure 5.11 compares the final velocity magnitude and vorticity fields obtained from the VPFCT-FUP algorithm and the Superbee limited MUSCL-H algorithm in a 6×6 square centered on the origin. The VPFCT-FUP method is able to maintain the amplitude of the steady velocity solution with good accuracy and the radial symmetry of the solution is left intact. As expected, the vorticity field is exactly preserved. In contrast, the MUSCL-H scheme is not able to maintain the steady velocity field or the vorticity solution. The radial symmetry of the problem is severely damaged and very large overshoots are present in the vorticity.

This section is concluded with more results from the plane wave vortex problem. Figure 5.12 compares pressure solutions obtained at $t = 1.2$ from the VPFCT-LPE method with those computed by the MinMod and Superbee limited MUSCL-H methods. The extrema that occur where the plane waves intersect makes for a challenging test for limiters. The same trends in the solutions that have been previously observed hold here, but these results demonstrate that the VPFCT method is able to properly limit a complex multidimensional flow. Finally, Figure 5.13 compares the pressure solutions at $t = 1.2$ from the unlimited VPFV2 and the VPFCT-FUP methods. The VPFCT scheme removed both the high frequency ripples and the large spurious waves from the second-order solution.

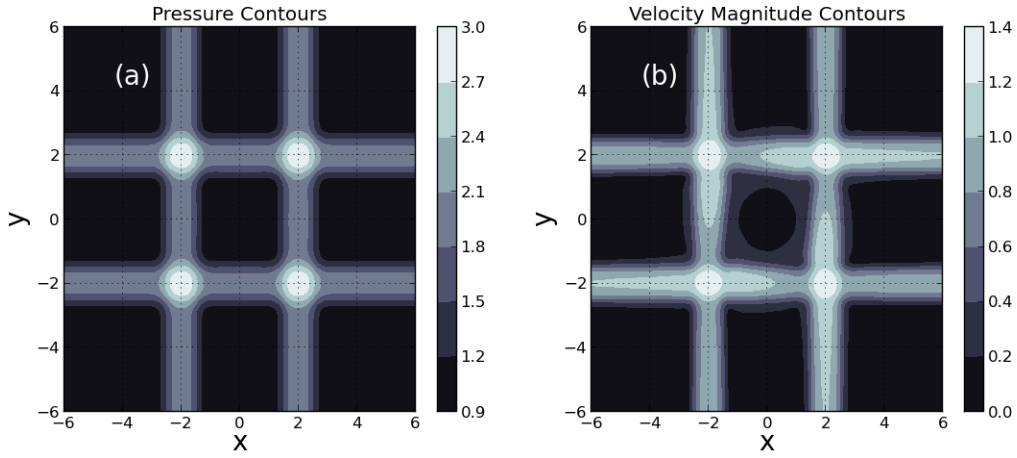

Figure 5.10: Intermediate results from a computation in which four plane waves interact with a steady, rotational velocity field (400×400 mesh, $t = 1.2$) demonstrate the complexity of the transient flow. (a) Pressure contours computed with the VPFCT method; (b) Velocity magnitude contours computed with the VPFCT method

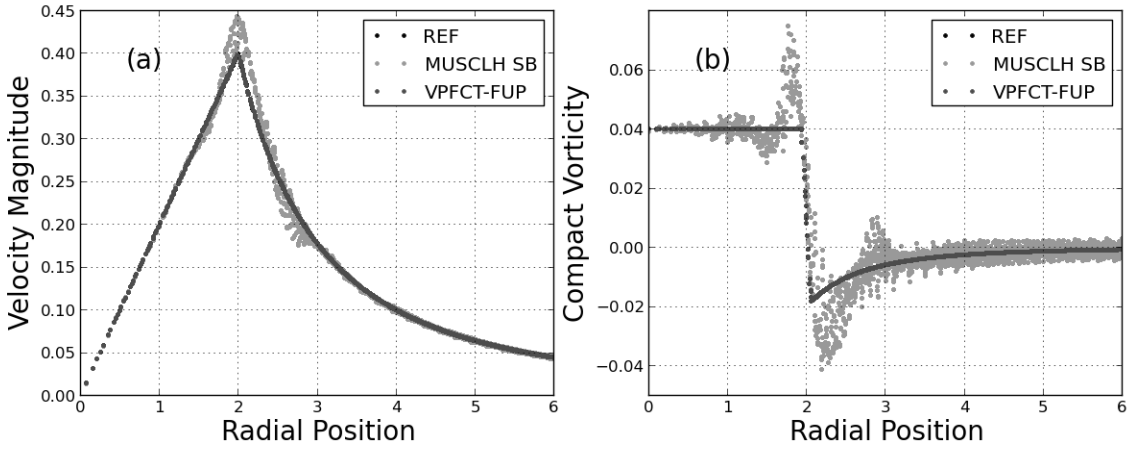

Figure 5.11: Final results ($t = 2$) from a computation in which four plane waves interact with a steady, rotational velocity field (400×400 mesh) from the VPFCT-FUP method ($\nu = 0.6$) and the Superbee limited MUSCL-H method ($\nu = 0.4$). (a) Radial velocity magnitude profiles; (b) Radial vorticity profiles

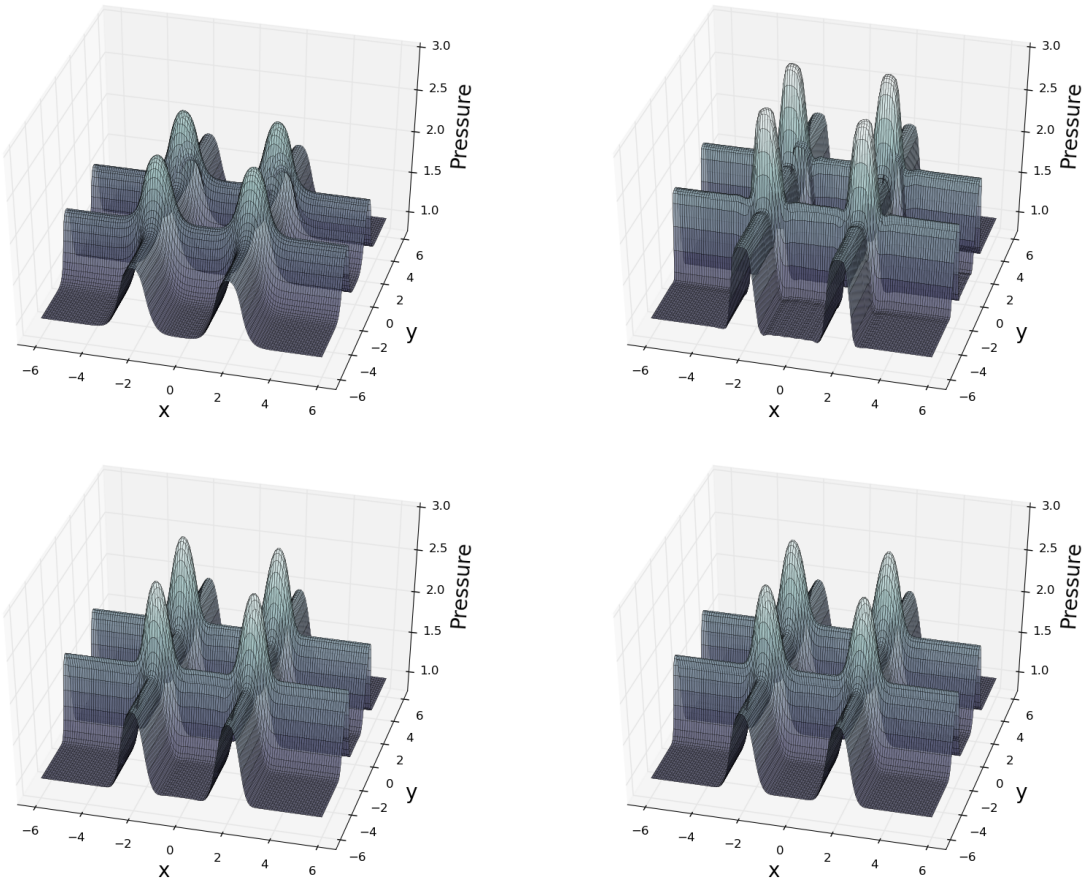

Figure 5.12: Pressure solutions are shown for the vortex-plane wave problem at $t = 1.2$ on a 400×400 mesh. Top Left: MUSCL-H MM $\nu = 0.4$; Top Right: MUSCL-H SB $\nu = 0.4$; Bottom Left: VPFCT-FUP $\nu = 0.6$; Top Right: VPFCT-LPE $\nu = 0.6$

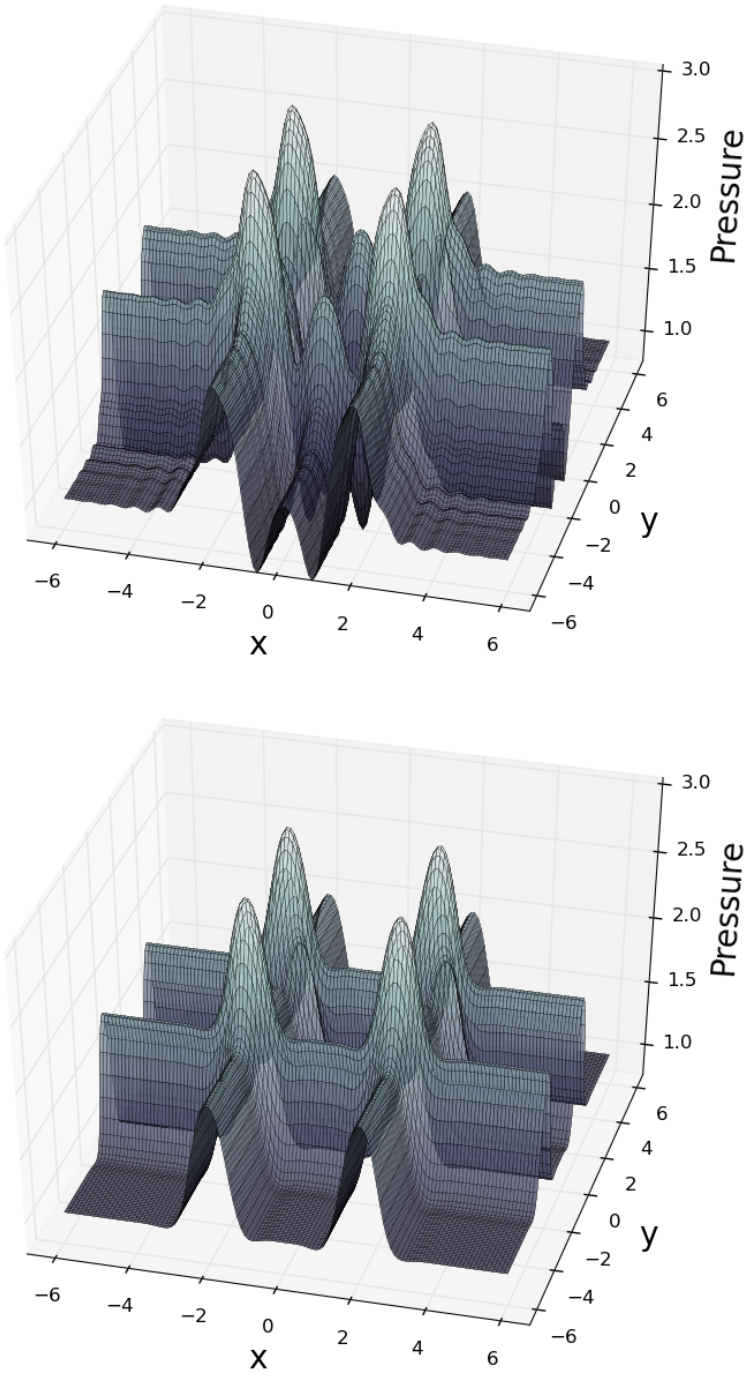

Figure 5.13: The temporal flux limiter is able to remove both the high frequency ripples and large spurious features from the pressure waves in the vortex problem, even at low CFL numbers. Left: VPFV2 Method with no limiter, 400×400 mesh, $\nu = 0.15$, t=1.2; Right: VPFCT-FUP 400×400 mesh, $\nu = 0.15$, $t = 1.2$

5.5 Discussion

Numerical results have shown that the VPFCT scheme developed with temporal flux limiting performs well on both smooth and discontinuous problems. For smooth data, the temporal flux limiters allow at least second-order convergence and produce absolute errors significantly smaller than an unsplit MUSCL-H type method with either the MinMod or Superbee slope limiters. The general form of the limited fluxes was obtained from third-order methods. This proved very useful as it greatly improved the dispersion properties of the method. Nonlinear behavior was produced by empirically determined functions. While some degree of empiricism is necessary for general problems, it may be beneficial to study the limiting framework presented here in a context where TVD type constraints are valid. Perhaps such a study could better motivate the functions F_0 and F_1 and produce a limiter that obeys a TVD type constraint when appropriate. At any rate, the present functions are not claimed to be optimal and may be improved upon in the future.

The incorporation of the provisional driver estimates into the antidiffusive fluxes was particularly convenient in the FCT framework. A provisional solution is already computed and so they can be obtained for little additional cost. However, a temporal approach could still be applied in one-step flux limited schemes. Essentially, if no provisional data is available, then $F_1 = 0$ and only F_0 must be determined. However, in this case, a new smoothness monitor may need to be developed if second-order terms are not available in the fluxes.

One of the attractive features of the temporal approach presented here is that it incorporates physics into the limiting process via the driver quantities and the provisional solution. This is not the case with methods that modify reconstructions of the local solution. The reconstruction and limiting steps are essentially separate from the physics of the problem. While a physics-based limiting procedure may seem desirable, the practicality of this approach may be called into question for problems

with complex physics. For the acoustic system, the fluxes were governed by simple equations. However, this will obviously not always be the case. Perhaps one way to proceed will be to individually limit the components of fluxes. For example, the total energy flux for the Euler equations in the Lagrangian frame is $p\mathbf{V}$. Perhaps this flux could be limited by restricting the values of p and $\mathbf{V}$ separately. While there is much work still to be done, it appears that the ideas presented in this chapter may be useful in developing genuinely multidimensional, compact flux limiters. To conclude, we return to the three questions posed at the beginning of the chapter and summarize the answers that have been proposed here:

Q What quantities should a limiter act on?

A Driver quantities

Q How should the limiting mechanism be designed?

A Temporal construction

Q How much limiting should be applied?

A Guided by a third-order scheme, a smoothness monitor, and empirically determined functions

CHAPTER VI

First-order Methods for Lagrangian Hydrodynamics: Part I

Chapter I began by introducing Lagrangian hydrocodes and discussing the desire to design a simple cell-centered method that resists mesh imprinting. The past three chapters have given an account of an extensive investigation in to acoustic algorithms. In this chapter we finally return to Lagrangian hydrodynamics and extend the acoustic methods to construct a first-order solver for the Euler equations on a moving mesh. Recall that the Euler fluxes in the Lagrangian frame are p and $p\mathbf{V}$ for the momentum and total energy, respectively. If it is assumed that each mesh vertex is moving with the local fluid speed than each component of the flux, the pressure and velocity, obey the nonlinear acoustic equations. As such, the acoustic flux evolution formulas from the previous chapters should be useful. This chapter begins in Section 6.1 by constructing a Lagrangian analog to the RR method that computes estimates of p and $\mathbf{V}$ at cell vertices. All vertex quantities will be denoted by $\hat{()}$. This algorithm will be constructed along the lines originally proposed by Roe [73] and will use the following update procedure:

1. Interpolate initial estimates of the flux components, $\hat{p}^n$ and $\hat{\mathbf{V}}^n$, from cell centers to the cell vertices.

2. Evolve the vertex flux components through one-half time step using a Lax-Wendroff-type (LW) procedure.

3. Move the mesh vertices according to the flux velocities $\hat{\mathbf{V}}^{n+1/2}$.

4. Update the conserved variables in cells by integrating the vertex fluxes, $\hat{p}^{n+1/2}$ and $(\hat{p}\hat{\mathbf{V}})^{n+1/2}$, around each cell-centered control volume using the trapezium rule.

The performance of the resulting algorithm will be assessed using some common test problems in Section 6.2. Some deficiencies will be identified that will be addressed in the next chapter.

6.1 From Acoustics to a Simple Lagrangian Method (SLaM)

A basic extension of the two-dimensional, acoustic RR method to the two-dimensional Euler equations on a Lagrangian grid will now be described step-by-step. The resulting method will be referred to as the simple Lagrangian method (SLaM)-A. The mesh is made up of quadrilateral elements and all of the methods presented here will use the nine cell stencil shown in Figure 6.1. The cell centers are assumed to be located at the geometric centroid of each quadrilateral, which is consistent with the center of mass for a uniform density distribution. Extending the RR method to Lagrangian hydrodynamics is primarily an exercise in extending the acoustic flux evolution formulas to deformed meshes. In dimensional form, we have

$$\hat{p}^{n+1/2} = \hat{p}^n - \frac{\hat{Q}_p \hat{h}^2}{2\hat{a}^2 \Delta t} \hat{\beta}_p, \tag{6.1a}$$

$$\hat{u}^{n+1/2} = \hat{u}^n - \frac{\hat{Q}_V \hat{h}^2}{2\hat{a}^2 \Delta t} \hat{\beta}_u, \text{ and} \tag{6.1b}$$

$$\hat{v}^{n+1/2} = \hat{v}^n - \frac{\hat{Q}_V \hat{h}^2}{2\hat{a}^2 \Delta t} \hat{\beta}_v. \tag{6.1c}$$

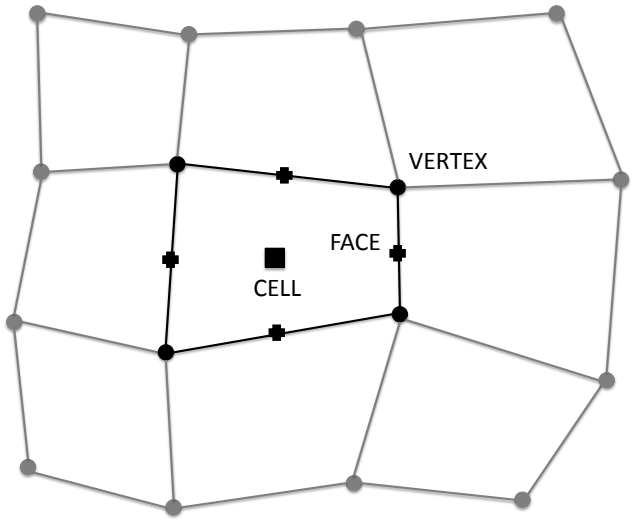

Figure 6.1: Lagrangian stencil illustration and nomenclature

As written, the vertex drivers are

$$\hat{\beta}_p = \hat{\rho}\hat{a}^2 \left(\Delta_x u + \Delta_y v\right), \qquad \hat{\beta}_u = \frac{1}{\hat{\rho}}\Delta_x p, \text{ and} \qquad \hat{\beta}_v = \frac{1}{\hat{\rho}}\Delta_y p. \tag{6.2}$$

To construct the method, an interpolation procedure is needed to obtain initial estimates of the solution at vertices, the Q-parameters must be specified, a definition for the characteristic cell size h must be established, and the first derivative approximations Δ_x and Δ_y must be defined. To aid in these tasks, the vertex centered control volume shown in Figure 6.2 was used.

6.1.1 Initial Flux Interpolation

Several different approaches could be adopted to provide initial vertex estimates of the pressure, velocity, density, and sound speed. Here a linear interpolation was performed by constructing planes from the cell-centered data around each vertex. Each vertex shares four cells in a quadrilateral mesh and, therefore, a plane cannot be uniquely determined. Instead, a least-squares best fit was performed by solving the

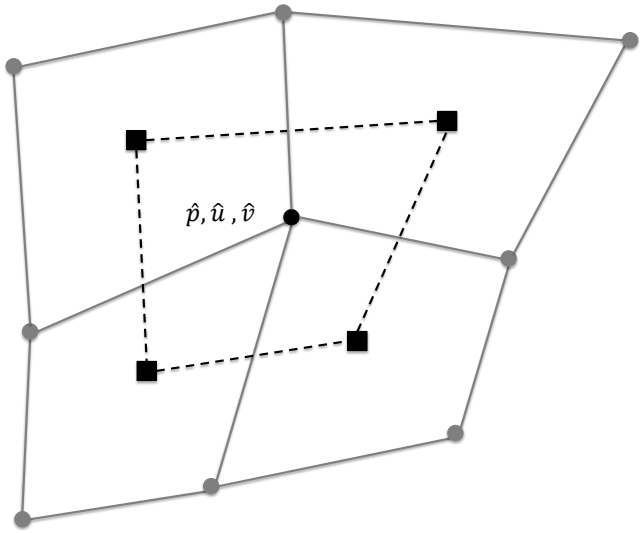

Figure 6.2: A staggered control volume is used to compute the vertex quantities $\hat{h}$, $\hat{Q}_p$, $\hat{Q}_V$, $\hat{h}$, Δ_x and Δ_y.

normal equations. This procedure has the advantage of being simple and robust in the sense that the geometry of the surrounding cell centers does not affect the computation of the plane. However, on highly deformed grids a vertex can fall outside of the control volume shown in Figure 6.2 and the procedure will turn into an extrapolation. In this case, the arithmetic mean of the cell-centered quantities was used. Appendix C contains more details on interpolation.

6.1.2 Characteristic Cell Size

The flux evolution equations require a characteristic length scale, $\hat{h}$, to be associated with each vertex. Once again, there are many possible ways to do this. Currently, $\hat{h}$ is taken to be the ratio of the volume (area), $\hat{\mathcal{V}}$, of the vertex centered control volume to its surface area (perimeter), $\hat{\mathcal{A}}$,

$$\hat{h} \equiv \hat{\mathcal{V}}/\hat{\mathcal{A}}. \tag{6.3}$$

Other choices are certainly viable and could be further explored.

6.1.3 Q-parameter Selection

The Q-parameters $\hat{Q}_p$ and $\hat{Q}_V$, which control the dissipation and accuracy of the method, must be defined. For first-order accuracy, the choice of the Q-parameters is arbitrary and only constrained by stability considerations. To give a first-order method, the Q-parameters could be set to $\hat{Q}_p = \hat{Q}_V = \hat{\nu}$ in analogy with the FUP method. Here $\hat{nu}$ is a local estimate of the CFL number. We will proceed with this choice, except that the q-parameter for the pressures will be modified to

$$\hat{Q}_p = \hat{\nu} + \min\left(0, \frac{\Delta t \hat{\beta}_p}{\hat{\rho}\hat{a}^2}\right). \tag{6.4}$$

This adds an $O(h^2)$ nonlinear artificial viscosity like term to the pressure evolution formulas. The extra dissipation is only active during compression and was useful for preventing overshoots near strong shocks and ensuring that there is numerical dissipation in the method in cases were the sound speed is very small.

6.1.4 Differentiation Operators

Second-order accurate approximations for ∂_x and ∂_y on a nonuniform mesh are needed. Here we follow the suggestion of Roe [74] and look to Gauss's theorem to derive the discrete operators. For a scalar quantity, ξ, defined over a control volume $\mathcal{V}$, the integral relations

$$\int \partial_x \xi d\mathcal{V} = \oint \xi dS_x \Rightarrow \overline{\partial_x \xi} = \frac{\int \partial_x \xi d\mathcal{V}}{\int d\mathcal{V}} = \frac{\oint \xi dS_x}{\int d\mathcal{V}} \text{ and} \tag{6.5a}$$

$$\int \partial_y \xi d\mathcal{V} = \oint \xi dS_y \Rightarrow \overline{\partial_y \xi} = \frac{\int \partial_y \xi d\mathcal{V}}{\int d\mathcal{V}} = \frac{\oint \xi dS_y}{\int d\mathcal{V}} \tag{6.5b}$$

hold. The quantities dS_x and dS_y denote the area-weighted face normal dotted into the x and y unit vectors, respectively. The discrete operators, ∇_x and ∇_y, can be obtained by replacing the exact contour integrals with the trapezium rule and applying

the formulas over the vertex centered control volumes. The resulting discretizations will be exact for linear data, which was a principle that was followed when selecting the discretizations for deformed meshes. See Appendix D for more information. Writing approximations for the first derivatives in terms of surface and volume integrals is convenient since they can be applied on a discrete volume of arbitrary shape and, therefore, they are directly extendable to deformed meshes with arbitrary connectivity.

6.1.5 Mesh Movement and Flux Integration

The mesh motion has already been defined by $\hat{u}^{n+1/2}$ and $\hat{v}^{n+1/2}$, which are assumed to be constant over each time step. To make second-order accuracy possible, which will be sought later, the fluxes must be integrated over the mesh geometry at the half time step. The mesh vertices are first moved over a half time step interval, the fluxes are integrated over each cell, and then the mesh vertices are moved to the final locations. The numerical integration is performed with the trapezoidal quadrature rule as in the RR method over bilinear faces in space-time. While the faces are functions of time, the trapezoidal quadrature naturally handles this by using face end points at $n + 1/2$. The momentum fluxes are integrated over a face of length $l^{n+1/2}$ according to

$$\int p \mathrm{d}l = \frac{1}{2}(\hat{p}_0^{n+1/2} + \hat{p}_1^{n+1/2})l^{n+1/2}, \tag{6.6}$$

where $\hat{p}_0^{n+1/2}$ and $\hat{p}_1^{n+1/2}$ are the vertex pressures on each end of the face. In the case of the total energy fluxes, it was assumed that $\hat{p}^{n+1/2}, \hat{u}^{n+1/2}$, and $\hat{v}^{n+1/2}$ each vary linearly over the faces. Linear functions for each variable can be reconstructed and

then integrated exactly. This leads to the quadrature rules

$$\int pu\mathrm{d}l \approx \frac{1}{6}(2(\hat{p}_0\hat{u}_0)^{n+1/2} + (\hat{p}_0\hat{u}_1)^{n+1/2} + (\hat{p}_1\hat{u}_0)^{n+1/2} + 2(\hat{p}_1\hat{u}_1)^{n+1/2})l^{n+1/2}, \tag{6.7a}$$

$$\text{and } \int pv\mathrm{d}l \approx \frac{1}{6}(2(\hat{p}_0\hat{v}_0)^{n+1/2} + (\hat{p}_0\hat{v}_1)^{n+1/2} + (\hat{p}_1\hat{v}_0)^{n+1/2} + 2(\hat{p}_1\hat{v}_1)^{n+1/2})l^{n+1/2}. \tag{6.7b}$$

6.1.6 Time Step Selection

A local CFL constraint must be used to restrict the time step size for each iteration of the computation. The local Courant number in each cell was defined by the cell geometry and the state it contains: $\nu = a\Delta t/h$. Here the characteristic cell size was defined according to

$$h \equiv \min(h_x, h_y), \tag{6.8}$$

where

$$h_x = \frac{\oint |dS_x|}{2\mathcal{V}} \quad \text{and} \quad \frac{\oint |dS_y|}{2\mathcal{V}}.$$

This definition will return the shortest side of a rectangle, which is more appropriate than the definition of h previously used for the flux evolution formulas as it helps to ensure that the numerical domain of dependence contains the physical one. At the beginning of each time step, an allowable step size is determined in each cell, i, according to

$$\Delta t_i = h_i \nu_m / a_i, \tag{6.9}$$

where ν_m is the maximum allowable CFL number specified by the user, a_i is the local sound speed, and h_i is computed as described above. The time step is then selected as $\Delta t = \min(\Delta t_i)$. Experience has shown that cautious choices for ν_m (< 0.5) help to prevent overshoots near shock waves on some difficult problems.

While the CFL constraint would provide a sufficient restriction on the time step

for an Eulerian method, the same cannot be said in the Lagrangian case. Equation (6.9) does not consider the motion of the mesh and, therefore, could admit a time step large enough to invert a cell in regions of compression where the relative velocity of the nodes in a single cell are large. To avoid this issue, a secondary constraint on the time step was used that limits the relative volume change of each cell. It was formulated by taking advantage of the relation

$$\nabla \cdot \mathbf{V} = \frac{1}{\mathcal{V}} \frac{D\mathcal{V}}{Dt}, \tag{6.10}$$

which implies that an estimate of the relative volume change could be obtained as

$$\frac{\delta \mathcal{V}}{\mathcal{V}} \approx \Delta t \nabla \cdot \mathbf{V}. \tag{6.11}$$

An arbitrary restriction between zero and one on the relative volume change can then be specified

$$\Delta t \nabla \cdot \mathbf{V} \leq K_{\Delta t} \tag{6.12}$$

to reduce the time step from the CFL condition if needed.

The time step selection procedure is implemented as follows. An initial time step is chosen according to the CFL condition. Then, the vertex fluxes are computed. The velocity divergence is computed in each cell using the evolved vertex velocities and then the time step is reduced if the volume change constraint is violated. Since the flux evolutions depend on the time step, the divergence constraint must be checked in an iterative manner. While this adds some expense, iterations are usually only necessary during the initial phases of the solution. Currently, $K_{\Delta t}$ is taken to be $1/2$.

6.1.7 SLaM-A Update Procedure

All the tools necessary to perform a Lagrangian update with first-order accuracy have been established. The final update procedure can be summarized as:

1. Compute initial estimates of $\hat{p}, \hat{u}, \hat{v}, \hat{\rho}$ and $\hat{a}$ at each vertex.
2. Evolve $\hat{p}, \hat{u}$, and $\hat{v}$ to $n + 1/2$.
3. Move the mesh to $n + 1/2$.
4. Integrate the fluxes over each face.
5. Move the mesh to $n + 1$.
6. Update the cell centered variables:

$$\mathbf{U}^{n+1} = \frac{1}{\mathcal{V}^{n+1}} \left[\mathcal{V}^n \mathbf{U}^n - \Delta t \left(\overline{\mathbf{F}} + \overline{\mathbf{G}} \right) \right] \tag{6.13}$$

where,

$$\mathbf{U} = (\rho, \rho u, \rho v, E), \tag{6.14a}$$

$$\overline{\mathbf{F}} = \left(0, \oint \hat{p}^{n+1/2} dS_x, 0, \oint \hat{p}^{n+1/2} \hat{u}^{n+1/2} dS_x \right)^T, \text{ and} \tag{6.14b}$$

$$\overline{\mathbf{G}} = \left(0, 0, \oint \hat{p}^{n+1/2} dS_y, \oint \hat{p}^{n+1/2} \hat{v}^{n+1/2} dS_y \right)^T. \tag{6.14c}$$

6.2 SLaM-A Numerical Results

Some test problem commonly found in the Lagrangian literature will be used to assess the performance of SLaM-A. A quick comment about units is in order. The test problems that will be considered are based on exact solutions to the Euler equations. The exact solutions are derived under very specific (and somewhat contrived)

conditions. The units of the quantities do not have much meaning since any consistent system could be specified. However, strictly speaking, dimensional quantities will be plotted and, therefore, they will be labeled as such. As is commonly done in the Lagrangian literature, the units will be defined on a centimeter(cm)-gram(g)-microsecond(μs) basis and the pressure and energy densities will be expressed in terms of megabars (Mbar). The volume in cubic centimeters will be abbreviated as "cc". The fluid will be an ideal gas in all cases.

6.2.1 Convergence Analysis

A convergence analysis was performed using a smooth two-dimensional exact solution to the Euler equations. The test problem is a projection of Kidder's three-dimensional isentropic compression problem [40] onto a two-dimensional Cartesian space that was devised by S. Ramsey at Los Alamos National Laboratory [70]. A linear velocity field, which is zero at the origin, and a Gaussian density field are prescribed in the initial data. While the exact solution is generally given in terms of a time variable that goes from $t \in [-1, 1]$, here the time variable was shifted such that $t^* \in [0, 2]$. The gas is undergoing compression when $t \in [0, 1)$ and expansion when $t \in (1, 2]$. The exact solution is given by

$$\rho(x, y, t^*) = \frac{2}{1 + (t^* - 1)^2} \exp\left[-\frac{x^2 + y^2}{1 + (t^* - 1)^2}\right] \text{ g/cc}, \tag{6.15a}$$

$$e(x, y, t^*) = \frac{1}{2[1 + (t^* - 1)^2]} \text{ Mbar*cc/g}, \tag{6.15b}$$

$$u(x, y, t^*) = \frac{x(t^* - 1)}{1 + (t^* - 1)^2} \text{ cm}/\mu\text{s and} \tag{6.15c}$$

$$v(x, y, t^*) = \frac{y(t^* - 1)}{1 + (t^* - 1)^2} \text{ cm}/\mu\text{s}. \tag{6.15d}$$

The solution was initialized on the box $x, y \in [-3, 3]$ cm and computations were run until $t = 0.6\mu$s with the maximum CFL number specified as 0.4. The ratio of specific

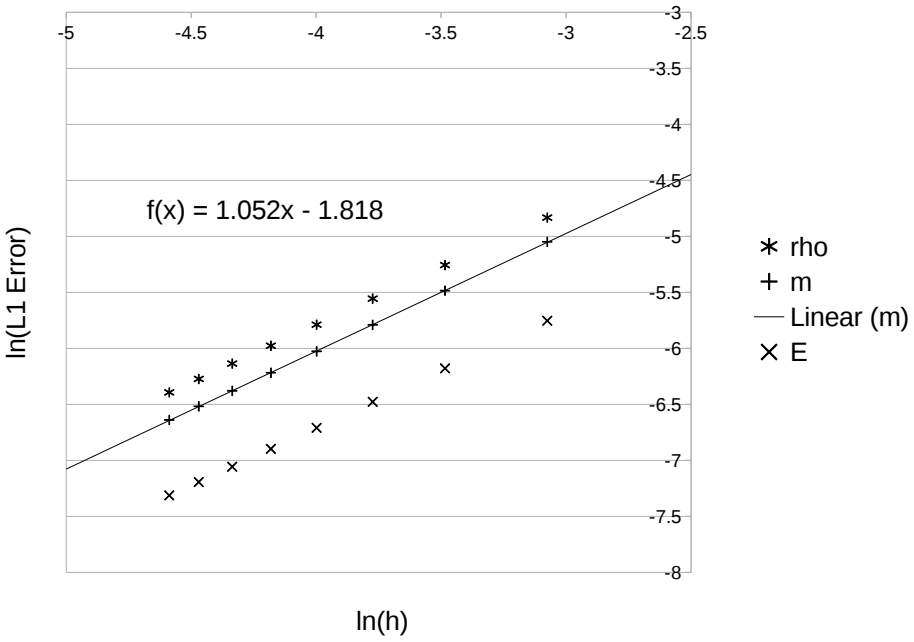

Figure 6.3: The SLaM-A method converges at first-order on the smooth Cartesian Kidder test problem.

heats, γ, is 2. The volume averaged $L1$ norm given by

$$L1_{\mathcal{V}} = \frac{1}{\mathcal{V}_{TOTAL}} \sum_{i=1}^{N} \mathcal{V}_i \epsilon_i \tag{6.16}$$

was used to measure the numerical errors on meshes ranging from 100×100 cells to 450×450 cells. A circular sample region was defined in the domain that ranged from $r = 0$ cm to $r = 1.5$ cm to avoid any boundary disturbances. The characteristic cell size was taken to be the average value of h for all of the cells on the last time step in the sample region using (6.3). Figure 6.3 shows that the Lagrangian method is converging at first-order as expected.

6.2.2 Sedov

The two-dimensional Sedov problem [75] is an exact solution to the Euler equations that models an idealized blast. The blast is generated by a large energy input at the origin. Here the problem was solved on a square mesh with 50×50 cells per quadrant. The ratio of specific heats was $\gamma = 5/3$ and the problem was solved on the domain $x \in [-1.25, 1.25]$ cm and $y \in [-1.25, 1.25]$ cm. In the four cells surrounding the origin, the total energy density was prescribed to be $E = 0.56114/4$ Mbar. The initial density was $\rho = 1$ g/cc everywhere and the initial pressure was set to $\epsilon = 10^{-12}$ Mbar outside the high energy region[1]. The problem was run until the $t = 1$ μs with $\nu_m = 0.4$. Results that include the final mesh and radial plots[2] of the mass density, total energy density, and pressure are shown in Figure 6.4. The results were encouraging as the SLaM-A method not only successfully computed the solution, but also preserved the radial symmetry of the problem to a reasonable degree. While the mesh is not badly deformed, some scatter is present in the radial plots near the origin. This is particularly evident in the pressure profile.

6.2.3 Noh

The Noh problem [67] is a challenging test of Lagrangian hydrocodes that can provoke many pathologies. The flow implodes at the origin and the challenge is to accurately compute the radial, outgoing shock that brings the flow to rest. The initial velocity field has a unit radial velocity, $v_r = -1$ cm/μs. The initial pressure and density are $\epsilon = 10^{-12}$ Mbar and 1 g/cc, respectively[3]. The ideal gas was assumed to have $\gamma = 5/3$ and the problem was computed on an initially square mesh with 50×50 cells per quadrant. The domain was $x \in [-1, 1]$ cm and $y \in [-1, 1]$ cm. Each

[1] The exact value of the pre-shock pressure is zero, which enables the exact solution.

[2] As was done for the acoustic results, these plots contain every point in the computational domain.

[3] Again, the pre-shock flow is assumed to have zero internal energy and the exact solution specifies the pressure as zero.

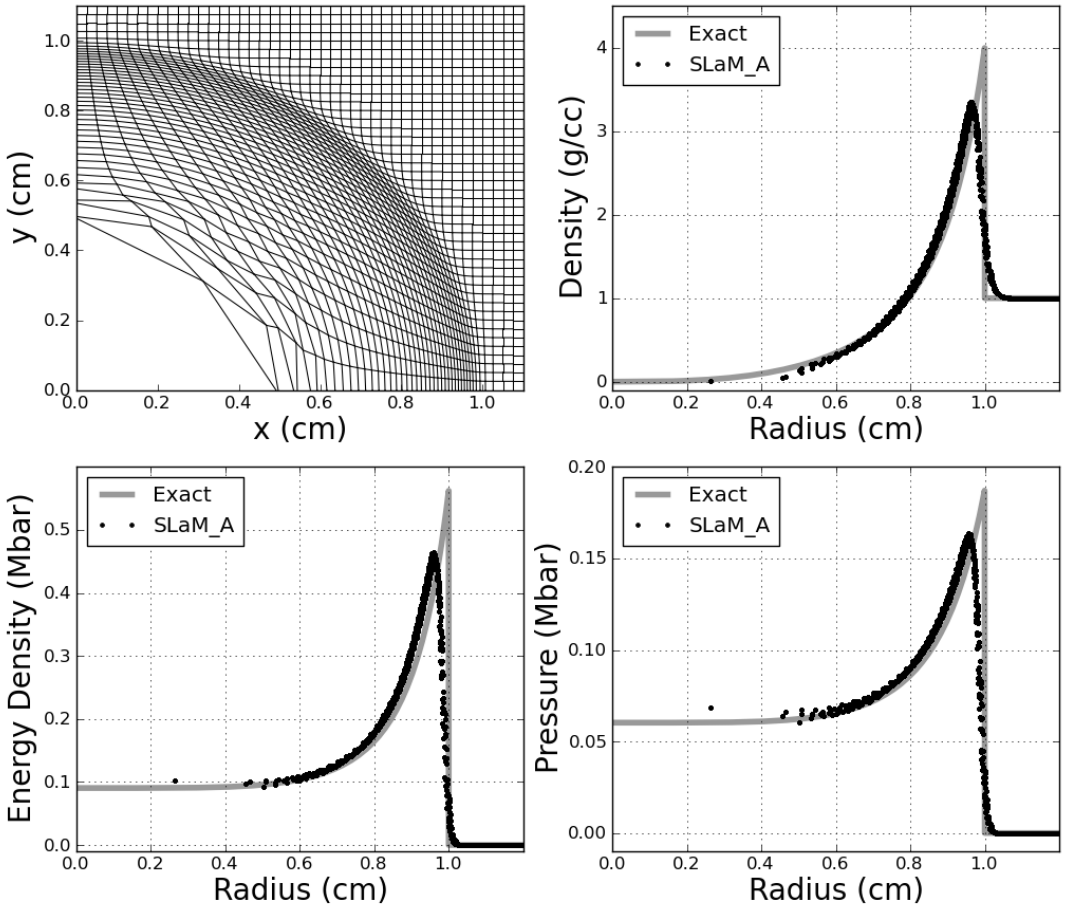

Figure 6.4: The SLaM-A method was able to successfully compute the two-dimensional Sedov problem on a Cartesian mesh.

computation was run until $t = 0.6$ μs and the maximum CFL number was specified as $\nu_m = 0.4$.

As expected, the Noh problem proved significantly more difficult than the Sedov problem. Figure 6.5 shows that while the solution was successfully computed, significant symmetry loss and mesh imprinting occurred. The symmetry loss was accompanied by spurious vorticity, as shown in Figure 6.6. This illustrates some of the new difficulty associated with Lagrangian hydrodynamics that was circumvented with the acoustics algorithms: Physical vorticity evolution can no longer be guaranteed. The large defect in the density near the origin is known as wall heating. It is a chronic issue that occurs in Lagrangian computations and it is related to excessive numerical entropy production (or heating). For more discussion see Noh's paper [67] or a more recent work by [71]. While this error is universally present in Lagrangian numerical solutions to problems like Noh, the magnitude observed here is substantial. Also note that the exact plateau values are not predicted correctly by the code, even though it is conservative, since the correct post-shock state never properly develops early in the solution. These discrepancies are related to errors in the mesh motion and are universally observed in Lagrangian computations of the Noh problem.

6.2.4 Saltzman

The Saltzman problem is another difficult test of the ability of Lagrangian hydrocodes to resist spurious mesh motions (see *e.g.* [29]). A one-dimensional piston generated shock is computed on a two-dimensional domain in which the mesh elements have been purposely skewed so they are not aligned with the shock. The mesh geometry is defined according to

$$x_{ij} = i\Delta x + (10 - j)\Delta y \sin\left[\frac{\pi i}{100}\right] + x_0 \text{ cm} \qquad y_{ij} = j\Delta y + y_0 \text{ cm}, \tag{6.17}$$

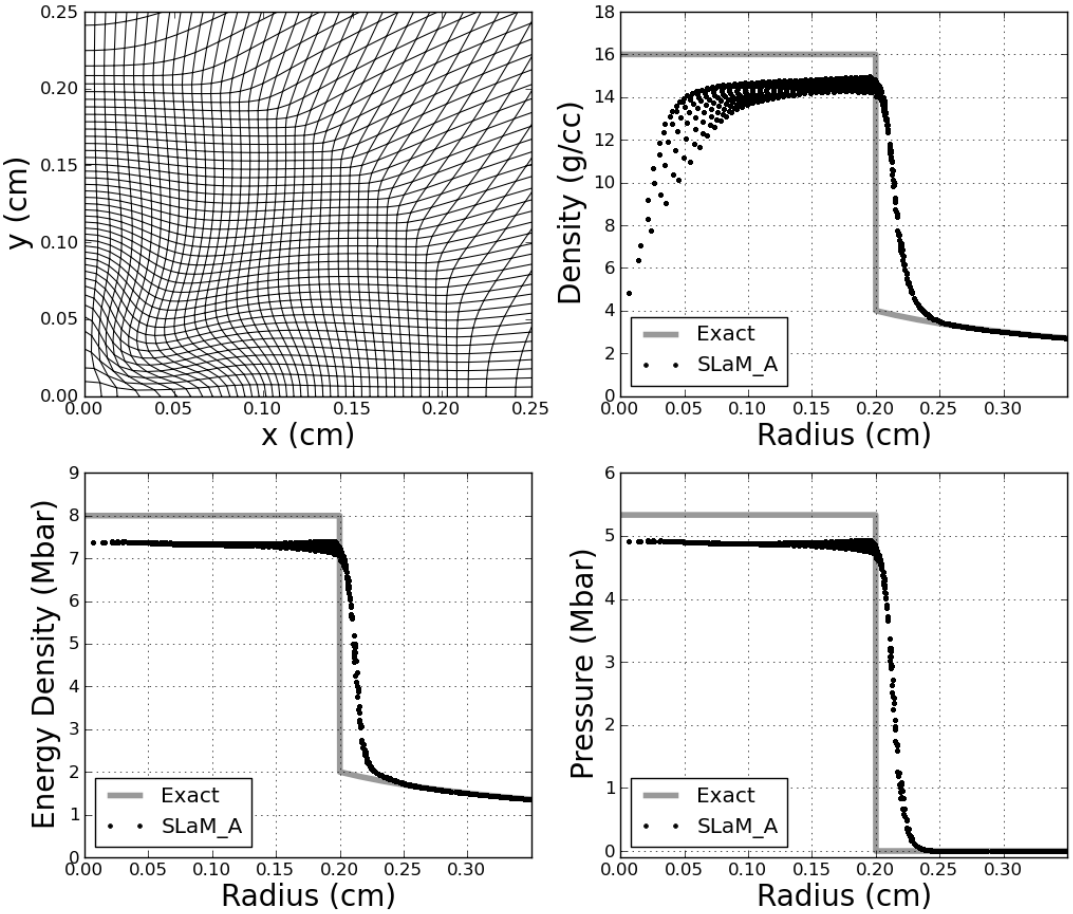

Figure 6.5: The SLaM-A method was able to compute the solution to the Noh problem, but noticeable symmetry losses and mesh imprinting occurred. In addition, the wall heating is substantial.

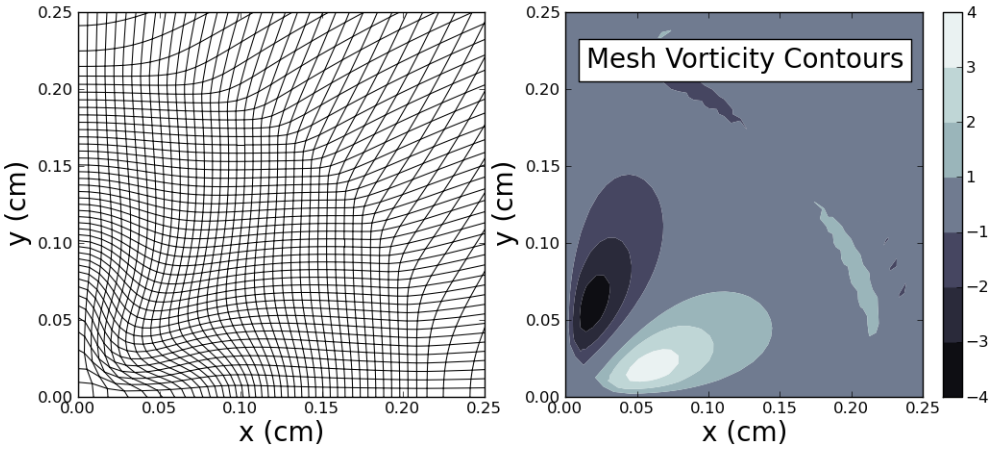

Figure 6.6: The symmetry loss observed on the Noh problem was accompanied by spurious vorticity.

where $\Delta x = \Delta y = 0.01$ and $i, j \in [0, 100]$. Traditionally, $x_0 = y_0 = 0$ cm and the piston is defined on the domain $x \in [0, 1]$ cm and $y \in [0, 0.1]$ cm, however, here we take $x_0 = -1$ cm, $y_0 = -0.01$ cm and use the domain $x \in [-1, 0]$ cm and $y \in [-0.01, 0]$ cm for reasons that will be explained shortly. Figure 6.7 shows the initial mesh geometry. The initial conditions involve a stationary ideal gas with $\gamma = 5/3$. The initial density and specific internal energy are 1 g/cc and 10^{-4} MbarCC/g, respectively. The piston, which is on the left side of the domain, moves with the speed 1 cm/μs. The other three boundaries are reflective walls. The exact solution gives a shock speed of 1.333 cm/μs and a post-shock state with $\rho = 4$ g/cc and $e = 1/2$ Mbar*cc/g. While this problem appears benign, it is famous for breaking Lagrangian methods. The object of the test problem is to compute the solution for as long as possible while preserving the one-dimensional shock wave. All codes will eventually crash as the piston will collide with the end wall when $t = 1\mu$s, but failure often occurs much earlier due to spurious mesh motions.

The SLaM-A method failed prematurely due to mesh tangling when computing the Saltzman problem. Figure 6.8 shows the mesh just prior the crash at $t = 0.34$.

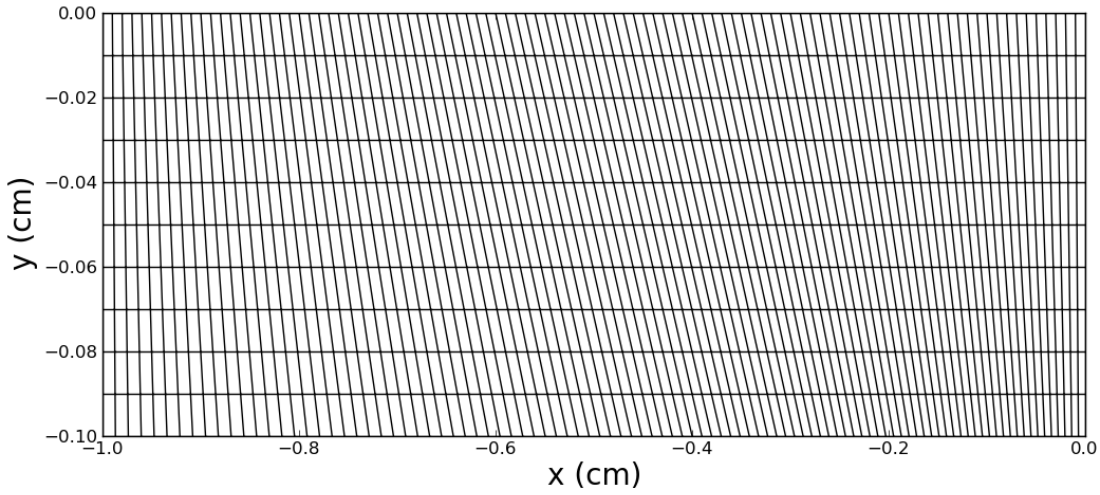

Figure 6.7: The initial Saltzman mesh has skewed elements so that it is not aligned with the one-dimensional shock generated by the piston on the left boundary.

All of the tangling in the mesh is concentrated near the top wall behind the shock wave. Initially, it was suspected that the problem was due to boundary conditions. To further investigate, the original Saltzman mesh was reflected over the x- and y-axes. This makes the top and bottom wall boundaries periodic and places the end wall in the interior of the domain. The only boundary conditions that must be specified in this case are the left and right piston. Figure 6.9 shows the initial mesh geometry in the left-hand plane. The original Saltzman mesh was placed in quadrant three earlier to better facilitate this mesh.

The problem was run again on the reflected mesh. Unfortunately, as Figure 6.10 shows, the problem was not due to the boundary conditions. The code crashed at the same time as before. Inspection of the figure shows that the same difficulty has occurred in the same location. This illustrates a peculiar feature of the Saltzman problem: The most difficult part of the problem occurs at the boundaries where the mesh geometry is discontinuous[4].

[4]Note that the reflected geometry can only be a problem for methods and boundary conditions that make use of it. A traditional Riemann solver would not care about the geometry; it only is

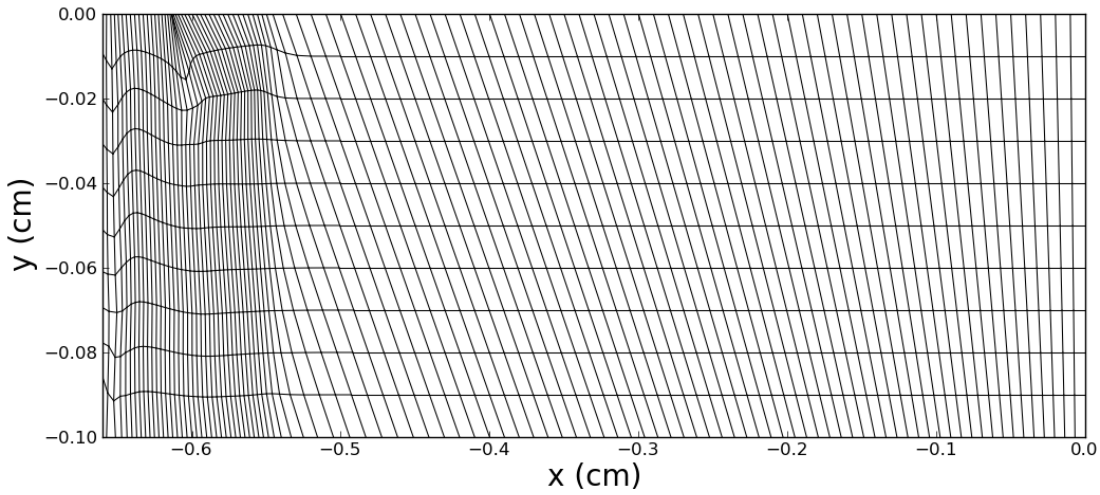

Figure 6.8: The SLaM-A method failed prematurely due to mesh tangling near the top boundary just after $t = 0.34\mu$s.

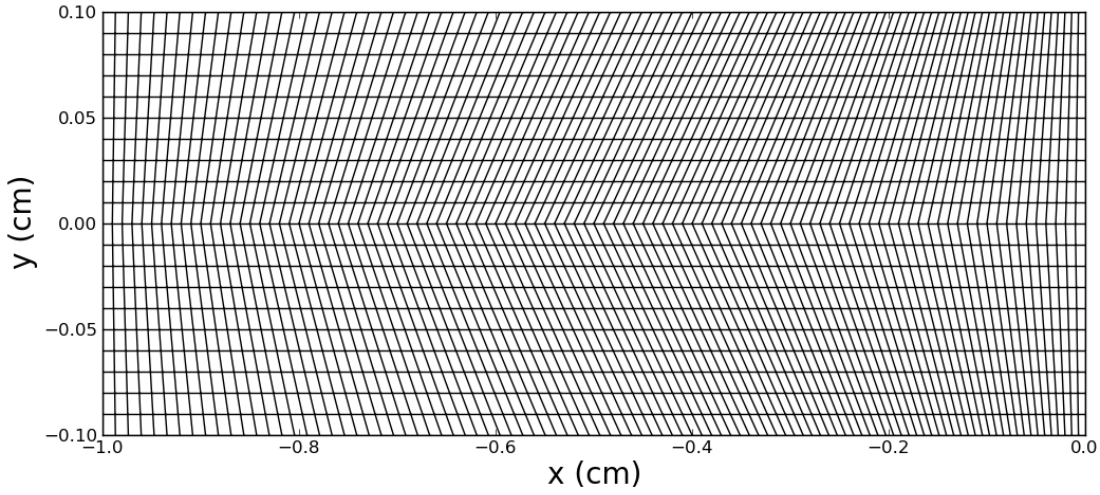

Figure 6.9: The Saltzman mesh was reflected over the x and y axes to study what happens near the boundaries. Only the left hand plane is shown here.

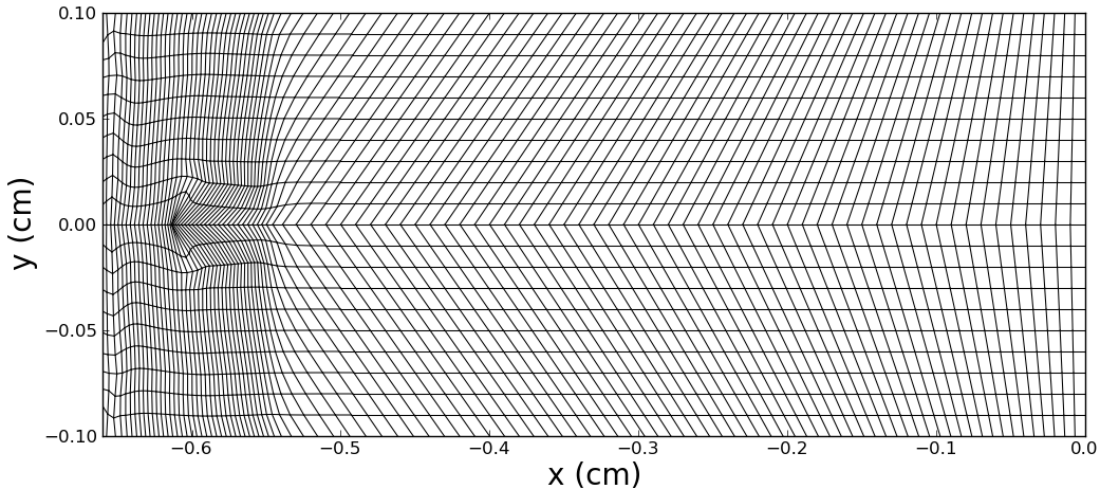

Figure 6.10: Reflecting the original Saltzman mesh over the x and y axes allowed periodic boundaries to be used everywhere except at the pistons. This mesh configuration shows that the most challenging part of the Saltzman problem occurs at the top and bottom wall boundaries where the mesh geometry is discontinuous.

To avoid the distraction of the discontinuous geometry, a new mesh was devised with smooth Gaussian grading. The mesh was constructed according to

$$y_j = j\Delta y + y_0 \text{ and}$$

$$x_j = K_x \exp\left[-K_y y_j^2\right] \sin\left[\pi(x_i^U - x_0)\right] \sin\left[2\pi(|y_j| + y_0)\right] + x_i^U,$$

$$\text{where } K_x = \frac{\Delta x(2500\Delta x - |x_i^U|)}{2}, \quad K_y = 20000\Delta y, \quad \text{and } x_i^U = i\Delta x + x_0.$$

The constants were taken to be $x_0 = -1$, $y_0 = -0.01$, and $\Delta x = \Delta y = 0.01$ cm. The left half of the mesh is shown in Figure 6.11. The problem was run once again and the code failed just as before. Figure 6.12 shows the mesh just before failure and vorticity contours computed from the nodal velocities. The tangling is accompanied by spurious vorticity as expected. This failure was taken as confirmation of a deficiency in the SLaM-A method.

concerned with the states on either side of the boundary.

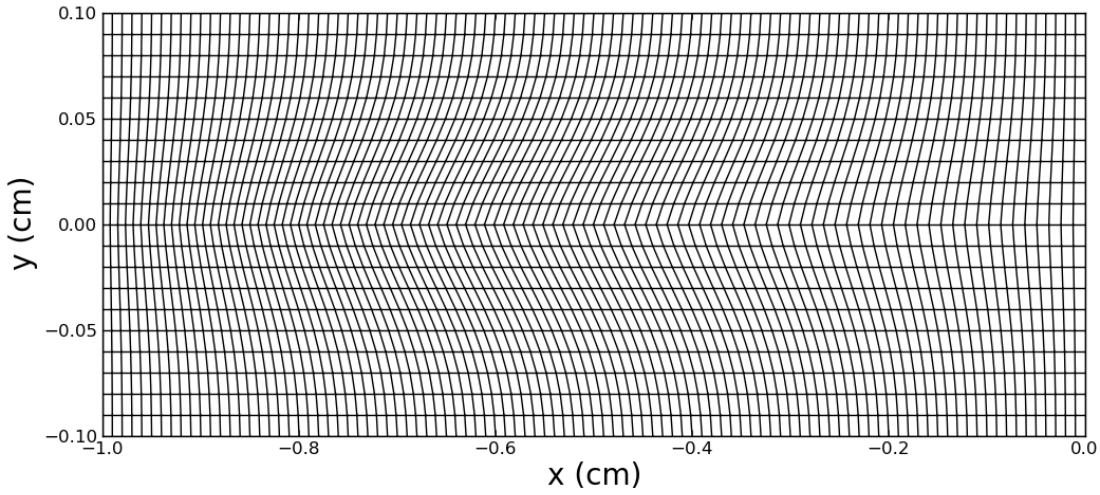

Figure 6.11: A mesh with smooth Gaussian grading was designed so the Saltzman problem could be run without the distraction of discontinuous geometry at boundaries.

It was hypothesized that the problem was related to a break down of the vertex centered control volumes, which are used to compute the fluxes. Figure 6.13 illustrates an example where the vertex lies outside of the control volume drawn using the adjacent cell-centers as specified by the mesh connectivity. In this situation, the geometrical interpretation of the CFL condition is violated and the flux estimation procedure breaks down. One situation where this may occur is in regions of the mesh where high aspect ratio elements are arranged in a chevron pattern. This is exactly the situation encountered in the Saltzman mesh at the upper and lower walls after the shock has passed. Some investigation into a way to overcome this issue will be carried out in the next chapter.

6.3 Discussion

All things considered, the extension of the RR method to Lagrangian hydrodynamics was encouraging. The Sedov and Noh problems were successfully computed on Cartesian meshes and the SLaM-A method was able to preserve the radial symme-

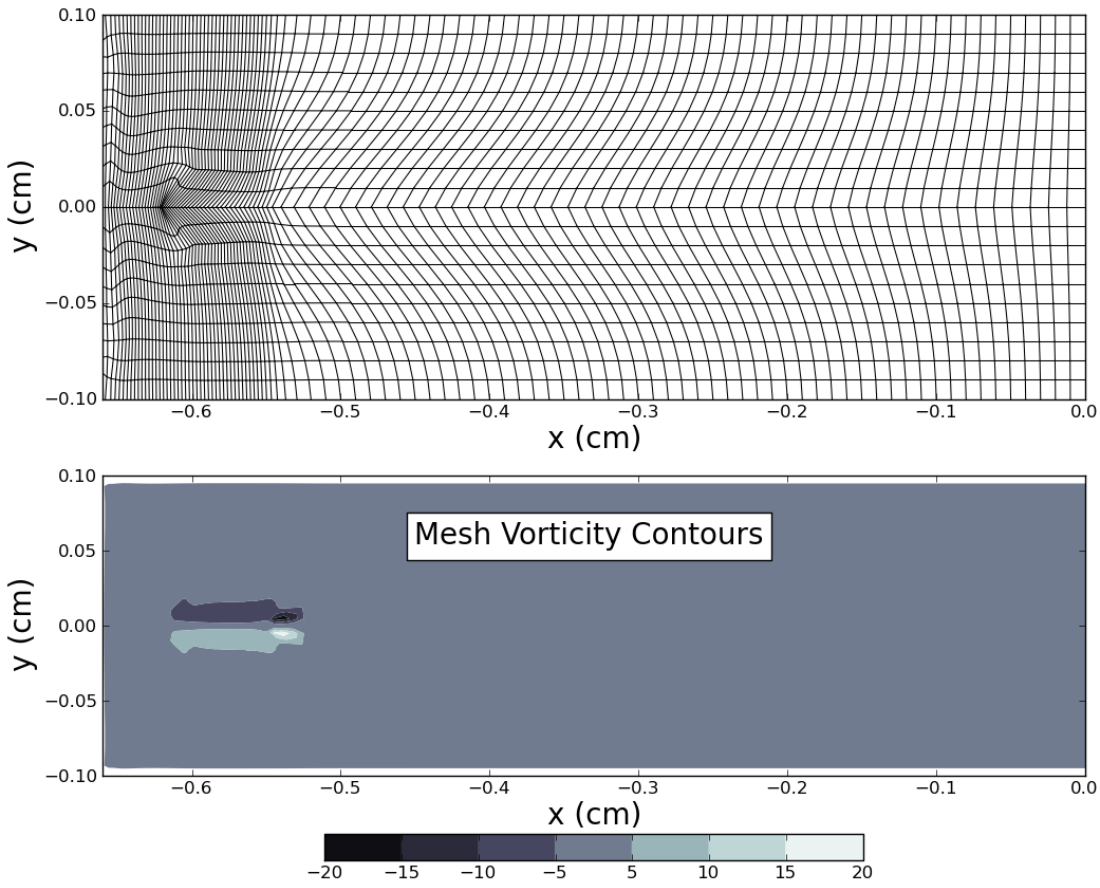

Figure 6.12: The SLaM-A method crashes around $t = 0.34\mu$s due to tangling in the same region as the original Saltzman mesh even on the mesh with smooth geometry. Spurious vorticity accompanies the mesh tangling.

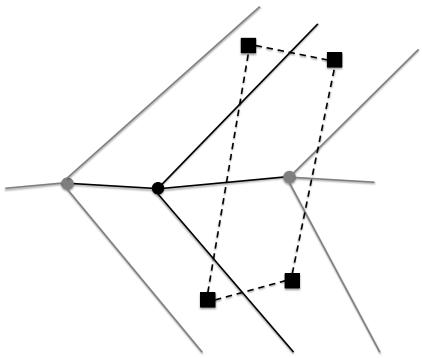

Figure 6.13: The vertex centered control volume that is specified according the mesh connectivity may not contain the vertex on highly deformed meshes, which means it is no longer useful for interpolation or computing the driver quantities.

try of the problems to a reasonable degree despite its simple construction. However, work remains. Compared with other solutions in the literature, excessive mesh imprinting was still evident, particularly in the Noh problem. Furthermore, the SLaM-A method failed prematurely on the Saltzman problem. A more robust flux estimation procedure is needed for badly deformed grids. The next chapter will aim to address these issues.

CHAPTER VII

First-order Methods for Lagrangian Hydrodynamics: Part II

The SLaM-A method presented in Chapter VI was constructed along the lines originally proposed by Roe. While results show that the proposed algorithm structure has potential, some improvements are needed. When computing the Noh problem, excessive mesh distortion and spurious vorticity damaged the solution behind the shock wave, suggesting that some mechanism is needed to resist badly formed mesh elements. Furthermore, the Saltzman problem failed prematurely, which was thought to be caused by a breakdown of the vertex-centered control volumes used to compute the fluxes. To make the mesh more robust and reduce mesh imprinting, face pressures will be added to the SLaM algorithm in Section 7.1. Then, in Section 7.2, the robustness of the flux evolution formulas will be improved by modifying the interpolation algorithms and incorporating face midpoints into the driver control volumes. The resulting algorithm will be referred to as SLaM-B. The usefulness of the changes will be assessed using the test problems from the previous chapter in Section 7.4 and some closing comments are included in Section 7.5.

7.1 Face Pressures

Mesh imprinting was reduced in the acoustic algorithms from Chapter III by adding point estimates of the fluxes for the pressure update at faces. This meant that unique estimates of the velocity components were assigned to faces. A similar improvement is needed for the SLaM method to reduce mesh imprinting, however, unique face velocities may not be specified in the Lagrangian algorithm. Doing so would create an inconsistency between the mesh motion and the fluxes, resulting in violations of the GCL. However, there is no such restriction on the pressure.

Point estimates of the pressure were added to the SLaM algorithm using a procedure analogous to the one used at vertices. Denoting all face quantities by $\tilde{()}$, the evolution equation for the face pressures is

$$\tilde{p}^{n+1/2} = \tilde{p}^{n} - \frac{\tilde{Q}_p \tilde{h}^2}{2\tilde{a}^2 \Delta t} \tilde{\beta}_p, \tag{7.1}$$

where $\tilde{\beta}_p = \tilde{\rho}\tilde{a}^2 \left(\Delta_x u + \Delta_y v\right)$. Initial estimates of the pressures at each face, $\tilde{p}^n$, were obtained from linear interpolation. Again, see Appendix C for details. To define the face-centered velocity divergence, the discrete operators from Chapter VI were applied on the control volume shown in Figure 7.1. The vertex velocities were taken to be the initial interpolations $\hat{u}^n$ and $\hat{v}^n$ and the cell-centered estimates were obtained from the momentum and density variables. Using this control volume will ensure that the face pressures respond to large distortions in the mesh and, therefore, it was expected that they would improve its robustness.

7.1.1 Q-parameter and Time Step Selection

With the addition of the face pressures, three Q-parameters must now be specified. Since the face pressures help stabilize the mesh, no artificial viscosity like terms were

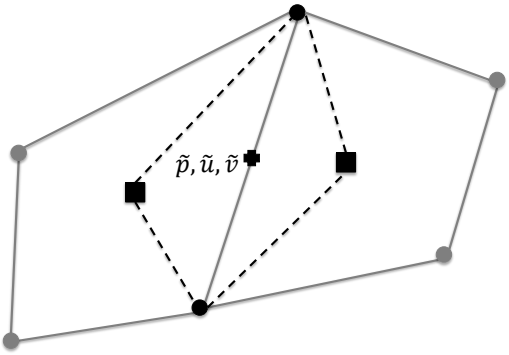

Figure 7.1: The control volume used to compute point estimates of the pressure at faces is shown.

found necessary. The q-parameters are taken to be

$$\hat{Q}_p = \hat{Q}_V = \hat{\nu} \quad \text{and} \quad \tilde{Q}_p = \tilde{\nu}.$$

The parameter $K_{\Delta t}$ used in the velocity divergence constraint was reduced to 0.2 to help prevent excessive time steps early on in some problems.

7.1.2 Flux Integration

The numerical quadrature rule used to average the fluxes over faces was modified to account for the face pressures. The quadrature rule was derived by assuming a quadratic variation of the pressure and linear variations of the velocity components over each face. After integrating the resulting polynomials exactly, Simpson's rule emerges. In the case of the energy fluxes, the velocity estimates at the faces are taken as the average of the adjacent vertices. This ensures that the GCL is not violated. For example, the time and space averaged x component of the energy flux at a face

is given by

$$\overline{pu}^{n+1/2} = \frac{1}{6}[(\hat{p}^{n+1/2}\hat{u}^{n+1/2})_0 + 4\tilde{p}^{n+1/2}\tilde{u}^{n+1/2} + (\hat{p}^{n+1/2}\hat{u}^{n+1/2})_1]l^{n+1/2}. \tag{7.2}$$

7.2 Flux Formula Robustness

In the previous chapter, the SLaM-A method crashed prematurely due to mesh tangling when computing the Saltzman problem. The cause of the mesh tangling was thought to be the break down of the vertex-centered control volumes used to compute the flux components on highly distorted grids. One way to fix this problem would to be to stop the computation and fix the distorted cells in the mesh. This is usually called rezoning. In order to implement a rezoning capability, some indicator of grid quality would be required to determine when and where mesh modifications were necessary. Then a new mesh must be specified, at least locally, and the solution variables mapped to the new geometry.

Two particular cell configurations are known to be problematic when dealing with quadrilateral meshes. The first is the formation of nonconvex cells in which the centroid of the zone does not lie in its interior. The other, which is observed in the Saltzman problem, are high aspect ratio zones that form a chevron pattern. To prevent nonconvex cells, the mesh could be locally rezoned if intersection point of the diagonals in a zone approaches the zone boundary. Another indicator that could be used to trigger a rezoning phase is the distance of the current vertex from the centroid of its control volume. If the cells are square, the two points will coincide, but they will generally become distinct on a deformed mesh. Cells could be flagged if the distance from the centroid to the vertex become too large. The allowable distance would be a tunable parameter, but could reasonably be specified as a fraction of $\hat{h}$, say 1/4. While more expensive, it may be better to rezone the mesh after every Lagrangian step. With this strategy it is possible to avoid heuristic measures of mesh quality.

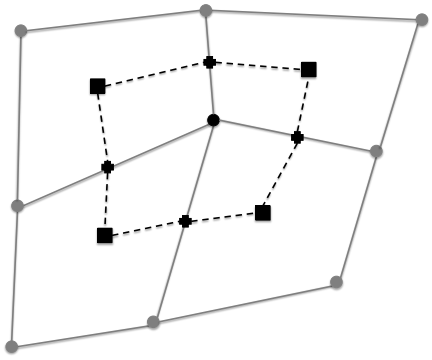

Figure 7.2: The vertex centered control volume was modified to include the faces.

A smoothing operation can be applied to the mesh after each step as is done using reference Jacobian matrices and an optimization procedure in [42].

The topic of mesh rezoning techniques falls outside of this thesis and so no additional discussion on the topic will be included. However, the SLaM method should be able to compute the Saltzman problem on a purely Lagrangian mesh well past the initial reflection of the shock from the end wall. Some ways to increase the robustness of the flux formulas on highly deformed meshes will now be considered.

7.2.1 Control Volume for Driver Estimation

The control volumes used to estimate the vertex driver quantities were modified to include the face mid-points. This incorporated more data into the driver estimates and ensured that the each vertex would fall inside its respective control volume, even on highly distorted grids. Since computing the driver estimate is an intermediate step in the flux prediction process, it was possible to use unique velocity estimates at the faces when determining the vertex-centered velocity divergence without violating the GCL. Figure 7.2 shows the modified vertex-centered control volume.

7.2.2 Interpolation

On highly deformed grids it is possible that the vertex- and face-centered control volumes used to interpolate initial estimates of the solution do not contain the vertex of interest. This forces the values to be extrapolated, which is obviously not desirable and violates the CFL condition. To help remedy the situation each vertex was checked to see if it fell inside its respective control volume. It it did not, then a control volume was sought from the neighboring data that did contain the point and the interpolation was performed with the new data. More information can be found in Appendix C.

7.3 SLaM-B Update Procedure

The SLaM-B method uses the following procedure to update the cell-centered conserved variables $(\rho,\ \rho\mathbf{V},\ E)$ from n to $n+1$:

1. Interpolate initial estimates of $\hat{p}^n, \hat{u}^n, \hat{v}^n, \hat{\rho}^n$ and the sound speed, $\hat{a}^n$, at each vertex.

2. Interpolate initial estimates of $\tilde{p}^n$, $\tilde{u}^n$, $\tilde{v}^n$, $\tilde{\rho}^n$, and $\tilde{a}^n$ at each face.

3. Evolve $\hat{p}^n,\ \hat{u}^n,\ \hat{v}^n,$ and $\tilde{p}^n$ to the $n+1/2$ time level.

4. Update the mesh position to $n+1/2$ using $\hat{u}^{n+1/2}$ and $\hat{v}^{n+1/2}$.

5. Integrate the fluxes over each face.

6. Update the mesh position to $n+1$.

7. Update the cell centered variables according to

$$\mathbf{U}^{n+1} = \frac{1}{\mathcal{V}^{n+1}}\left[\mathcal{V}^n\mathbf{U}^n - \Delta t\left(\overline{\mathbf{F}} + \overline{\mathbf{G}}\right)\right] \tag{7.3}$$

where,

$$\mathbf{U} = (\rho, \rho u, \rho v, E), \tag{7.4a}$$

$$\overline{\mathbf{F}} = \left(0, \oint p^{n+1/2} dS_x, 0, \oint p^{n+1/2} u^{n+1/2} dS_x\right)^T, \text{ and} \tag{7.4b}$$

$$\overline{\mathbf{G}} = \left(0, 0, \oint p^{n+1/2} dS_y, \oint p^{n+1/2} v^{n+1/2} dS_y\right)^T. \tag{7.4c}$$

Note that the conceptual progression listed here does not represent an efficient implementation strategy.

7.4 SLaM-B Numerical Results

7.4.1 Convergence

The convergence rate of the SLaM-B method was measured using the smooth Kidder problem and the volume-weighted $L1$ norm from the previous chapter. Figure 7.3 shows that the method converged at first-order as expected.

7.4.2 Sedov

Sedov solutions computed with the SLaM-B method have near perfect radial symmetry as shown in Figure 7.4. The changes discussed in this chapter have removed nearly all of the mesh imprinting from the solution. Furthermore, the solutions are in better agreement with the exact answers as compared with those presented for the SLaM-A method previously. Figure 7.5 shows density contours and the radial density profile when the solution in computed on a finer mesh with 100×100 cells per quadrant. The solution is converging to the exact answer and no deviations from radial symmetry are observed.

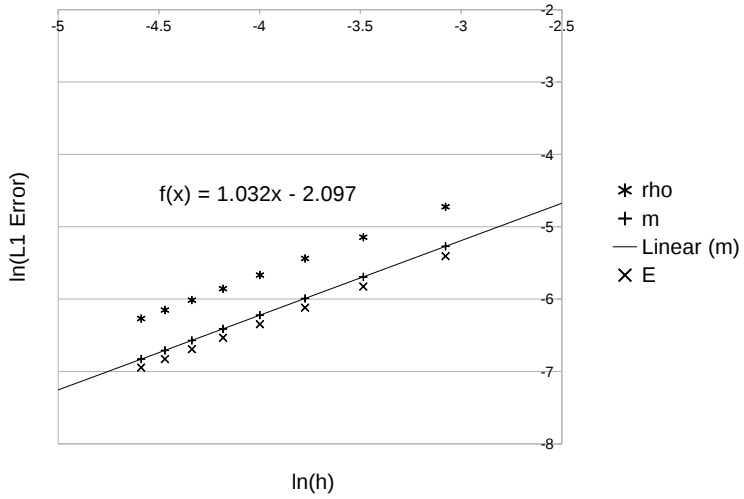

Figure 7.3: The SLaM-B method converges at first-order on the smooth Cartesian Kidder test problem.

7.4.3 The Riemann Solver Pitfall

One might be tempted to to obtain the face pressures in the SLaM-B algorithm from a Riemann solver of some type. While this is certainly possible and would more closely parallel traditional finite volume practice, it is not advisable. Figure 7.6 shows Sedov results from the SLaM-B method if the face pressures are obtained from an exact Riemann solver. The Riemann solver caused a large degradation in the solution isotropy by allowing the mesh to imprint itself on the solution. Figure 7.7 shows the density contours, which have been flattened where the shock is not aligned with the grid. This example illustrates the importance of incorporating multidimensional physics into multidimensional algorithms. The Riemann solver has no concept of the multidimensional velocity divergence and, therefore, is not able to accurately compute the pressure everywhere in the domain.

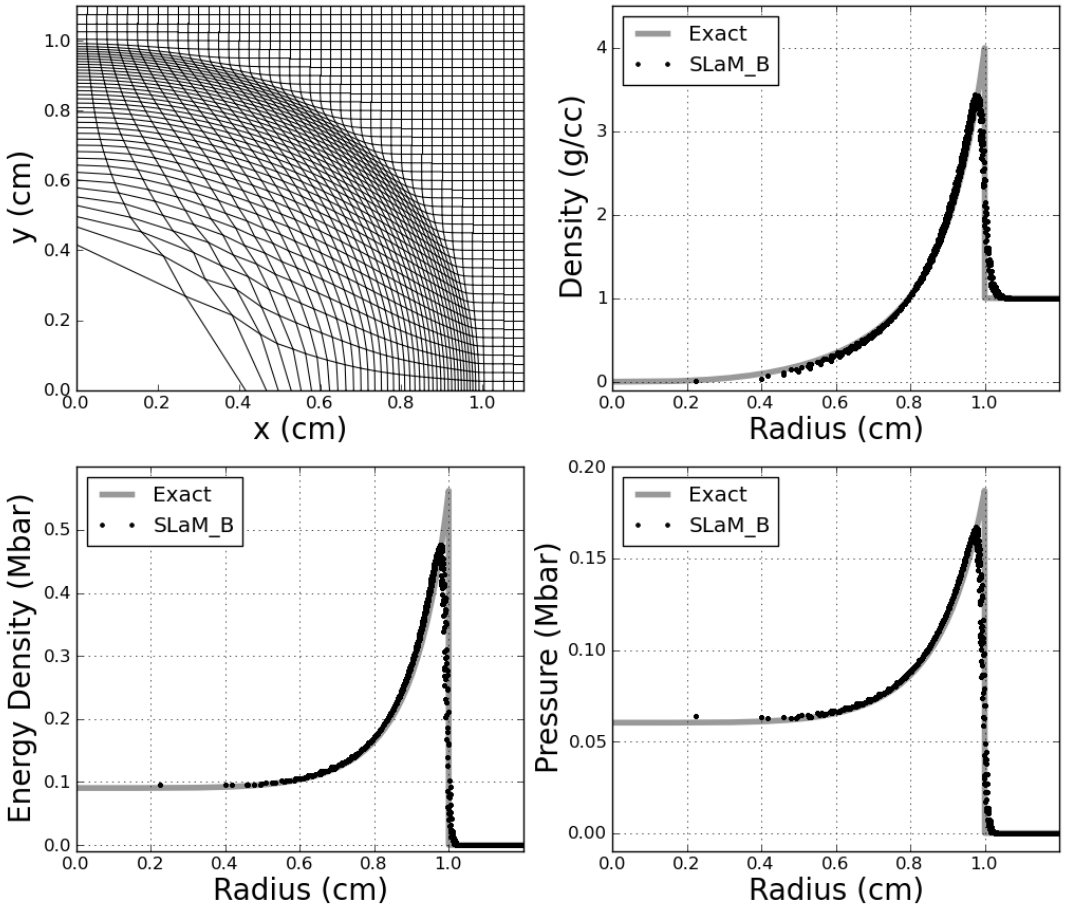

Figure 7.4: The SLaM-B method suffers from almost no mesh imprinting when computing the Sedov problem on a Cartesian mesh with 50×50 cells per quadrant.

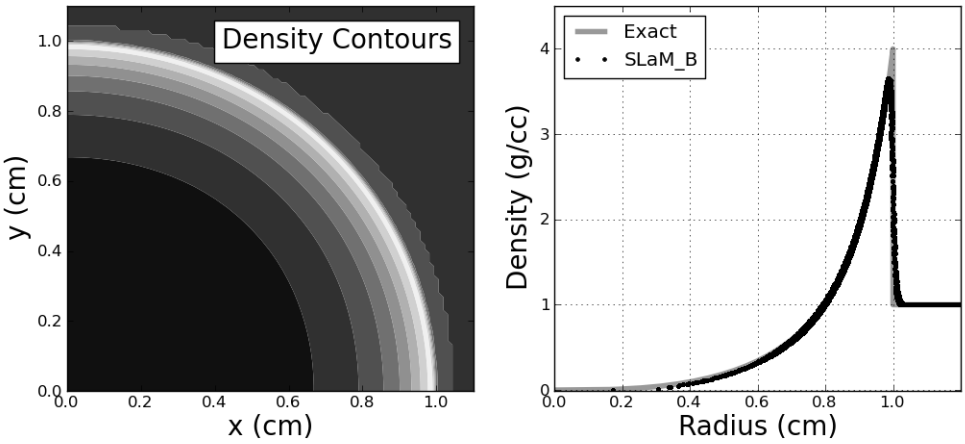

Figure 7.5: No deviations from radial symmetry are present in the Sedov solution obtained with the SLaM-B method on a mesh with 100×100 cells per quadrant.

7.4.4 Noh

The Noh problem was run with the SLaM-B method to better gauge the effectiveness of the modifications introduced in this chapter. Figure 7.8 shows the solution computed on a mesh with 50×50 cells per quadrant. The results are extremely encouraging since nearly all mesh imprinting has been removed from the solution. The improvement in the mesh quality, and therefore the density profile, in the post-shock region is impressive when compared to the corresponding SLaM-A results presented before. In this region, the scatter in the density data was reduced by over 75 percent. See Appendix B. The severity of the wall heating has also been substantially reduced by removing the artificial viscosity like term from the q-parameters[1]. The solution quality is further improved if the maximum allowable CFL number is lowered. Figure 7.9 shows the density contours and radial profile produced when $\nu_m = 0.1$. The shock front is free from overshoots in all directions. Also, Figure 7.10 shows that very little spurious vorticity is present in the post-shock state. Finally, Figure 7.11 shows a

[1]Since the most difficult part of the Noh problem occurs near the origin during start-up, an experiment was done in which the problem was initialized at $t = 0.3$ μs with the exact data. See Appendix E for more information

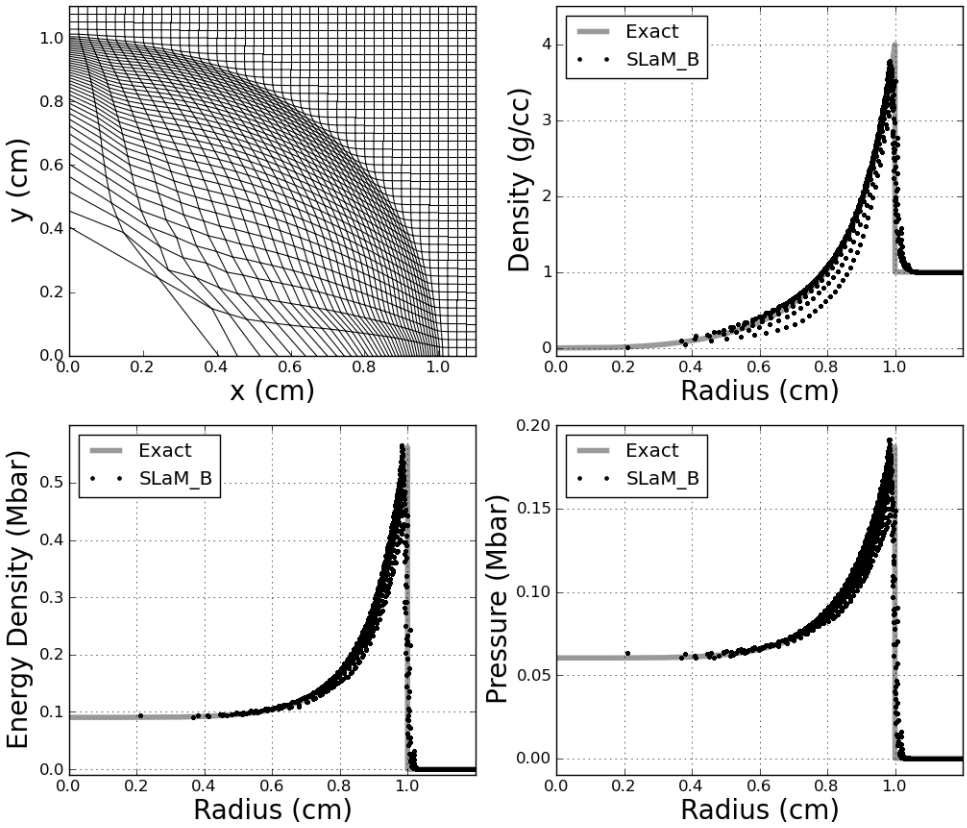

Figure 7.6: The multidimensional face pressures in the SLaM-B method were replaced with pressures obtained from an exact Riemann solver. The Riemann solver caused mesh imprinting and destroyed the radial symmetry of the problem.

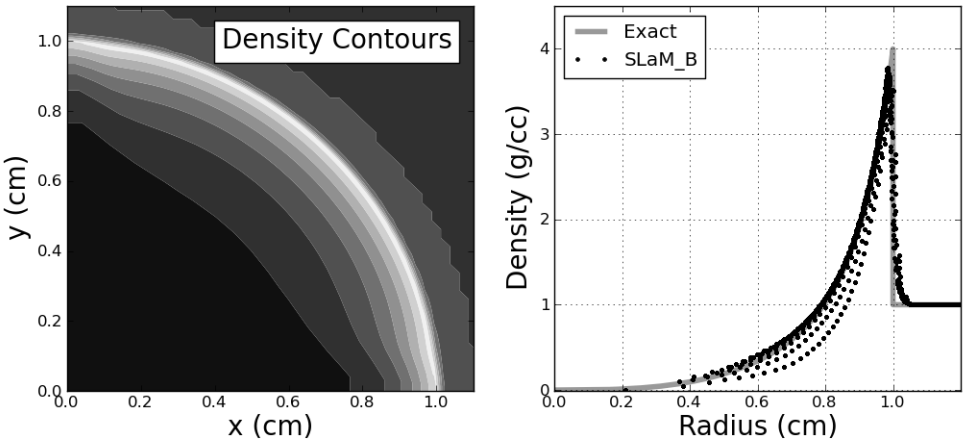

Figure 7.7: The exact Riemann solver has flattened the density contours in regions where the shock is not aligned with the grid.

solution computed on a finer mesh with 100×100 cells per quadrant. The solution is converging and maintains all of the positive attributes discussed.

7.4.5 Triple Point

The modifications made when constructing the SLaM-B method were effective in preventing spurious vorticity from entering the solution when computing the Noh problem. A valid question then, is whether or not the method prohibits physical vorticity production. To investigate, the triple point problem was run (see *e.g.* [15]). In the version implemented here, three different gas states were initialized at rest. The states interact at the "triple point" and vorticity should develop due to baroclinic production. The initial data was specified as shown in Figure 7.12. Results very early in the solution at $t = 0.25$ μs show that vortex roll-up is well underway and the mesh has tangled as a result. See Figure 7.13. Clearly the SLaM-B method still allows physical vorticity production. The problem cannot be continued unless the mesh is fixed[2].

[2]Some first-order Lagrangian methods are able to compute the solution up until $t \approx 3$ μs. However, the only way this is possible is if the method strongly damps the physical vorticity.

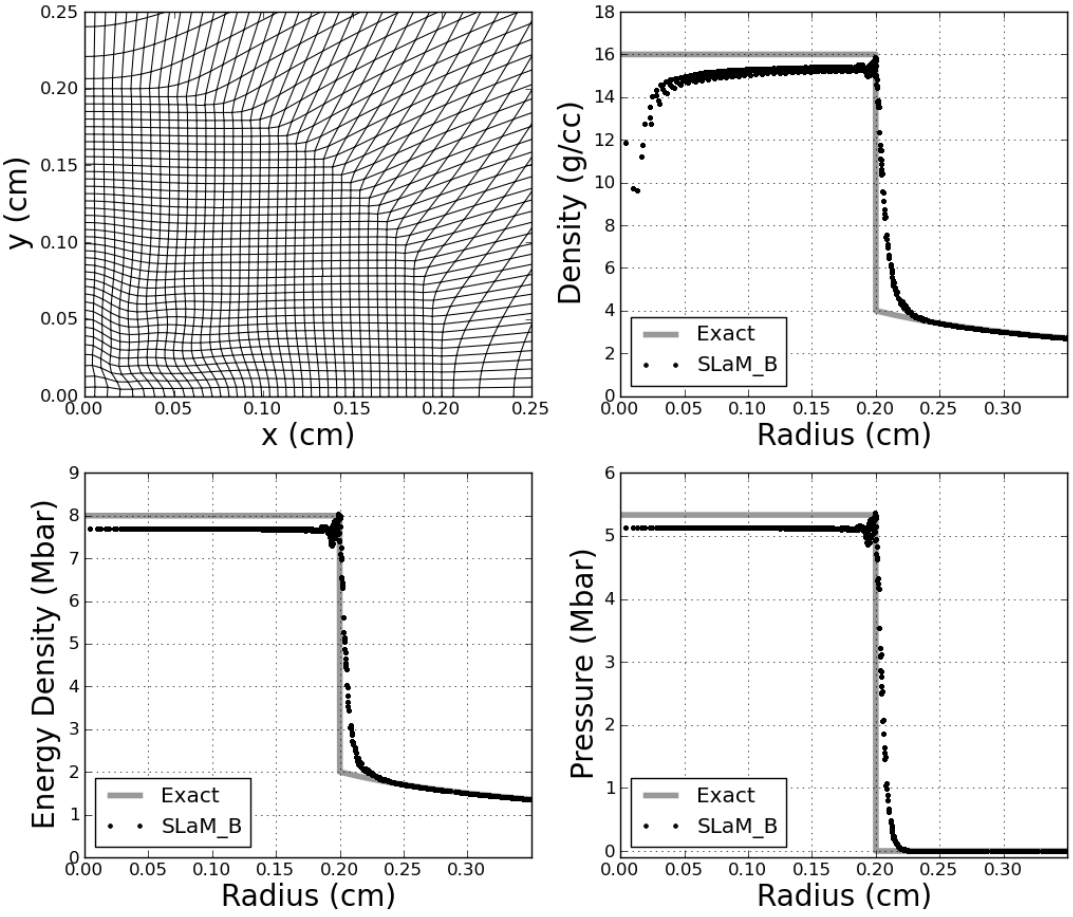

Figure 7.8: The SLaM-B method is able to compute the Noh problem with very little mesh imprinting.

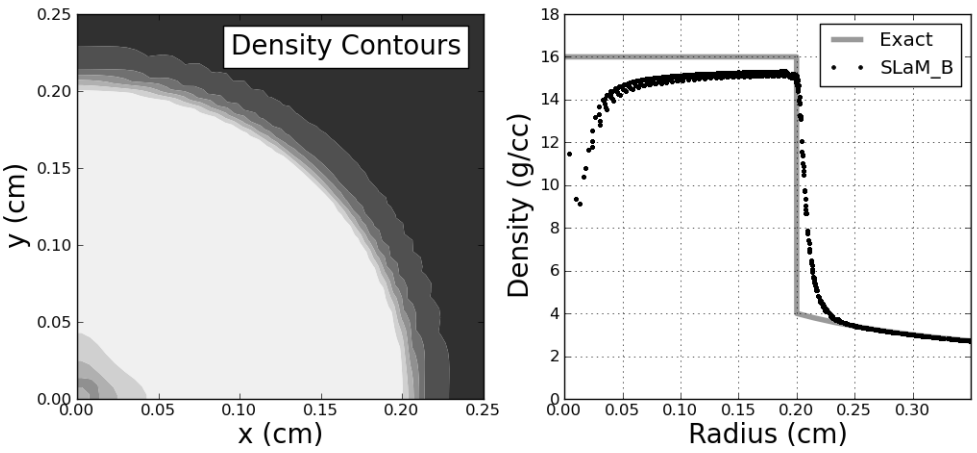

Figure 7.9: A perfectly radial shock front with no overshoots can be computed with the SLaM-B method by reducing the maximum allowable CFL number. Here $\nu_m = 0.1$.

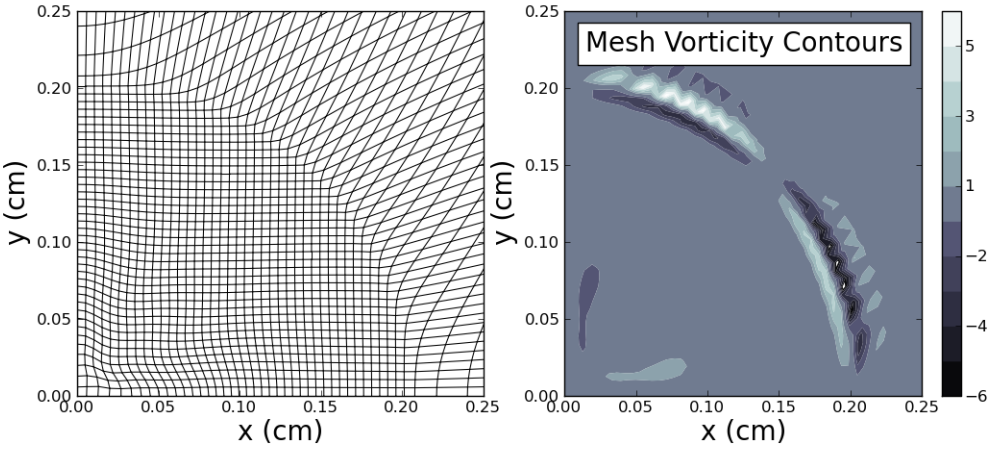

Figure 7.10: The SLaM-B method generates very little spurious vorticity in the post-shock region of the flow. Some vorticity is imparted to the mesh as elements pass through the shock.

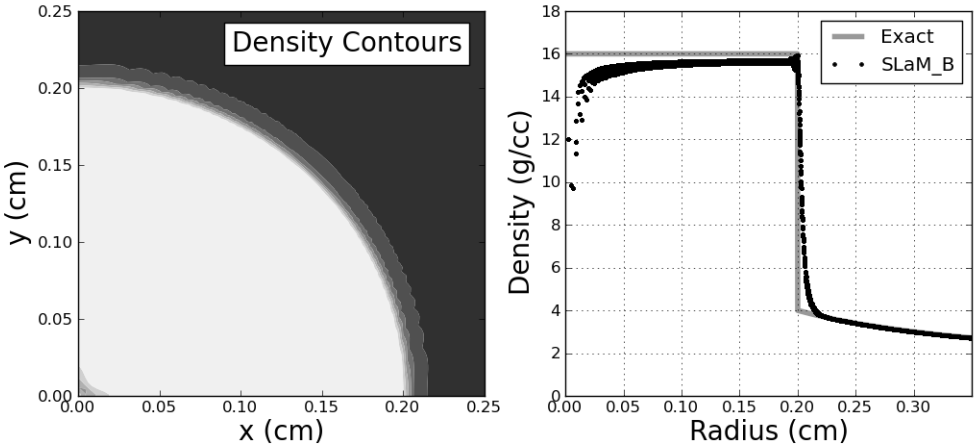

Figure 7.11: A Noh solution obtained with the SLaM-B method on a finer mesh with 100×100 cells per quadrant is shown.

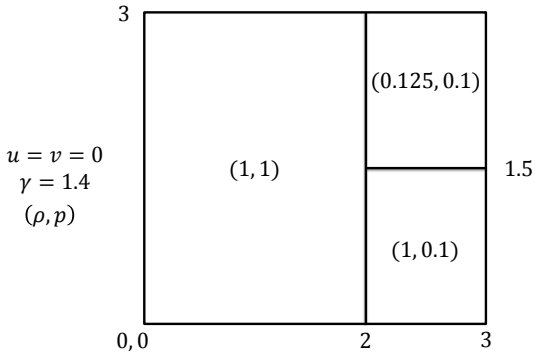

Figure 7.12: The graphic shows the initial conditions for the triple point problem.

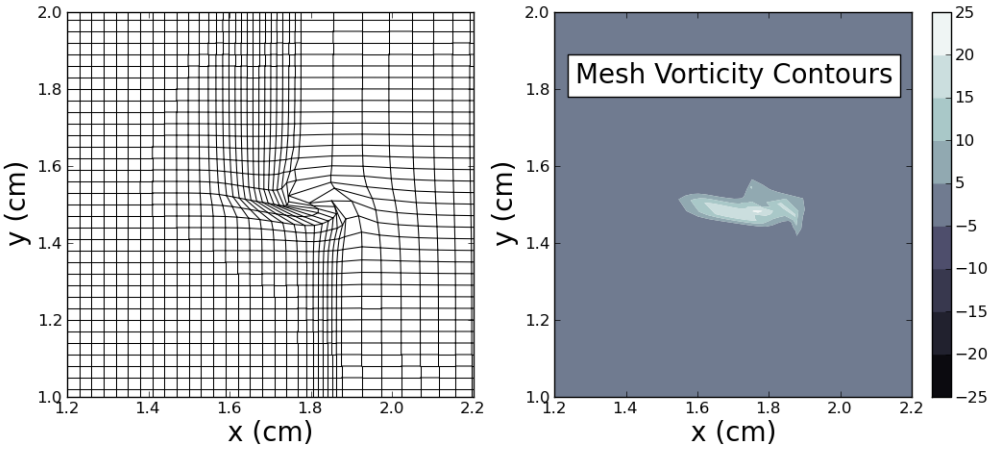

Figure 7.13: The SLaM-B method allowed physical vorticity production to occur right away in the triple point problem. ($t = 0.25$ μs, $\nu_m = 0.4$)

7.4.6 Saltzman

The Saltzman problem was computed until $t = 0.34$ μs with the SLaM-B method, which is just prior to failure for the SLaM-A method. The resulting meshes from each method were then compared. Figure 7.14 shows the original Saltzman mesh, while Figure 7.15 shows the Gaussian mesh. The improvements aimed at increasing the robustness of the interpolation and flux evolution algorithms in the SLaM-B method have certainly had a positive effect. The mesh tangling that crashed the SLaM-A method is not yet evident in either of the SLaM-B results.

The SLaM-B method was allowed to compute the Saltzman problem to failure on both meshes. In each case, the code crashed around $t \approx 0.81$ μs, which is just after the shock reflects off the end wall. Both meshes were plotted at $t = 0.8$ μs and included in Figure 7.16. While the capability of the SLaM-B method is certainly improved over the SLaM-A method, the results are still somewhat disappointing since a modern CCH code should be able to compute the solution well past $t = 0.9$ μs. However, all is not lost. The remaining problem areas in the solution occur in regions where no suitable control volume can be found for the vertices using the methodology described

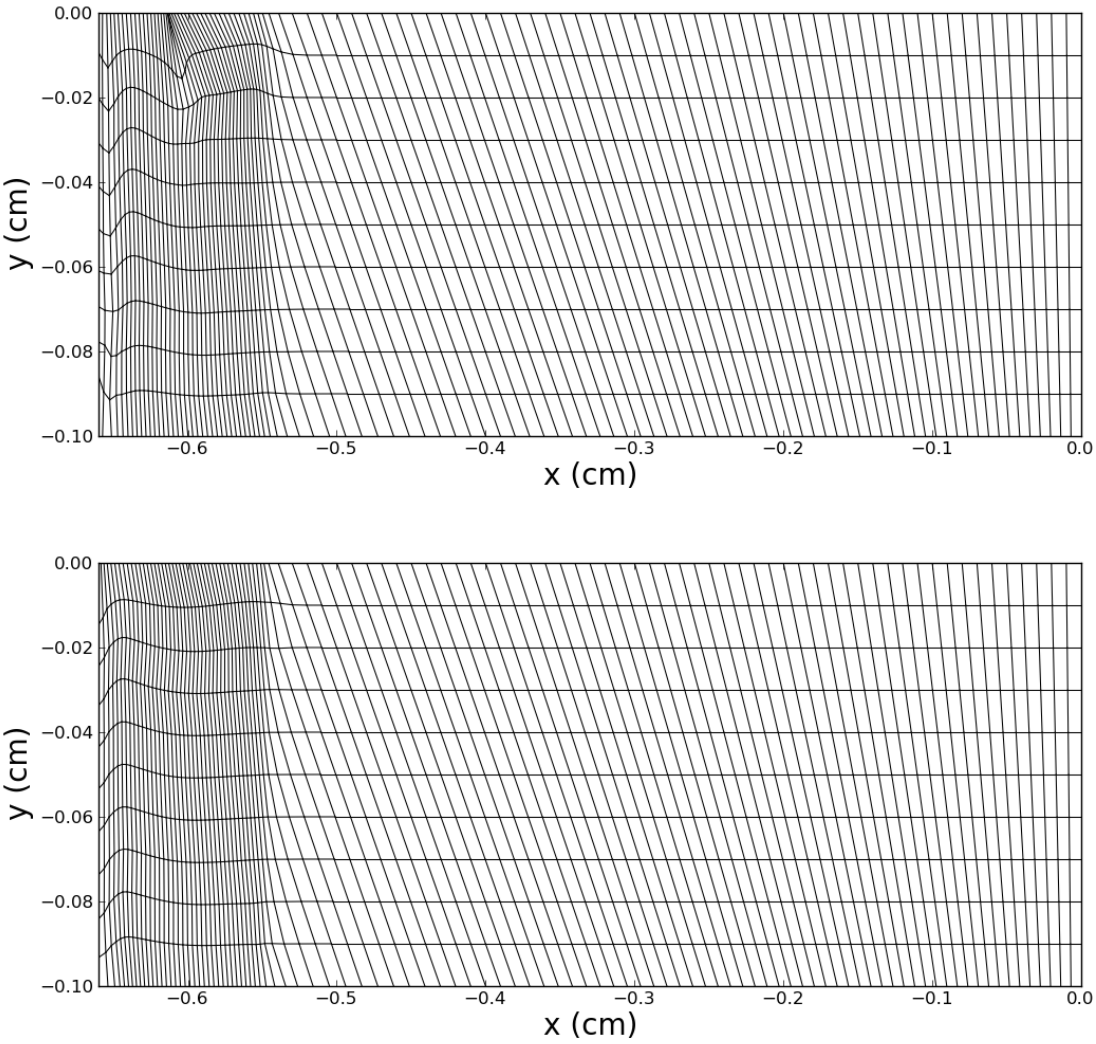

Figure 7.14: The SLaM-B method shows no indications of mesh tangling at $t = 0.34$ μs on the original Saltzman mesh, whereas the SLaM-A method was about to crash. Top: SLaM-A; Bottom: SlaM-B

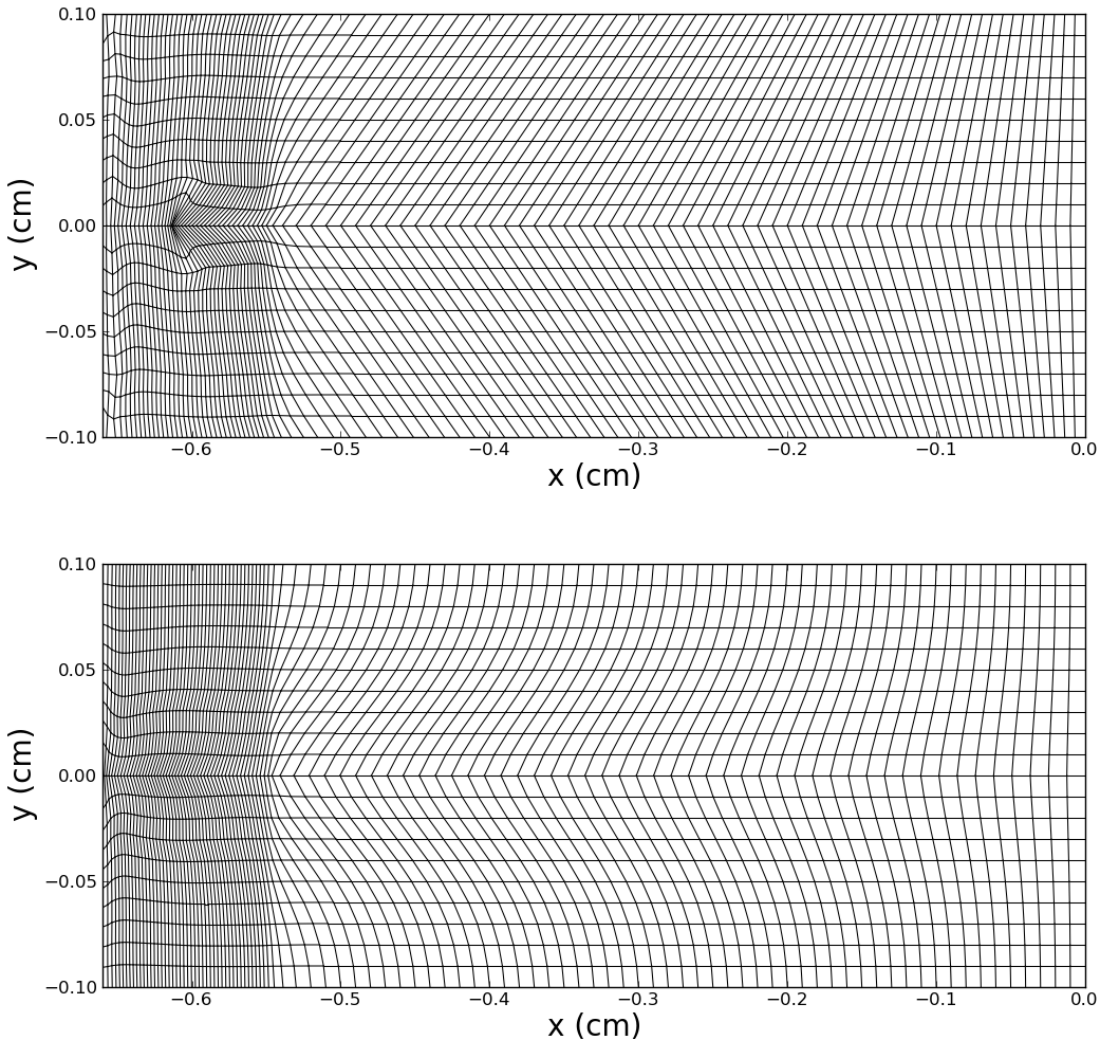

Figure 7.15: The SLaM-B method shows no indications of mesh tangling at $t = 0.34$ μs on the Gaussian mesh. Top: SLaM-A; Bottom: SLaM-B

in Appendix C. A more robust method for finding appropriate data for interpolation is needed. Furthermore, even when the procedure works, it is still quite crude and switches the data used for interpolation instantly when a vertex crosses into another control volume. It would be preferable to implement a procedure that facilitates a smooth transition. Instead of devoting resources to resolving this problem now, it will be dropped in favor of extending some of the limiting ideas developed in Chapter V to Lagrangian hydrodynamics. Additional improvements to the interpolation procedure are left to future work.

7.5 Discussion

The improvements incorporated into the SLaM-B method in this chapter led to an algorithm that was far superior to the SLaM-A method from Chapter VI. Nearly all mesh imprinting and spurious vorticity were removed from the Sedov and Noh solutions and, while the Saltzman problem still presents some challenges, much improvement was demonstrated there, too. What is, perhaps, most encouraging is that the large improvements did not require a large increase in the complexity of the scheme or any fundamental deviations from the framework proposed in Chapter I. It is interesting to note that the most troublesome problem with the current methods, the break down of vertex-centered control volumes, is a direct consequence of using quadrilateral meshes. These problems would be completely avoided on an unstructured triangular mesh and there is nothing about the methods described here that would prevent their extension to such. Now, some new difficulties may arise in that case, but in the author's opinion, exploration of the idea would be worthwhile. Given the superior performance of the SLaM-B method, it will taken as the preferred scheme and referred to in the next chapter simply as SLaM, where a second-order extension of the method will be done.

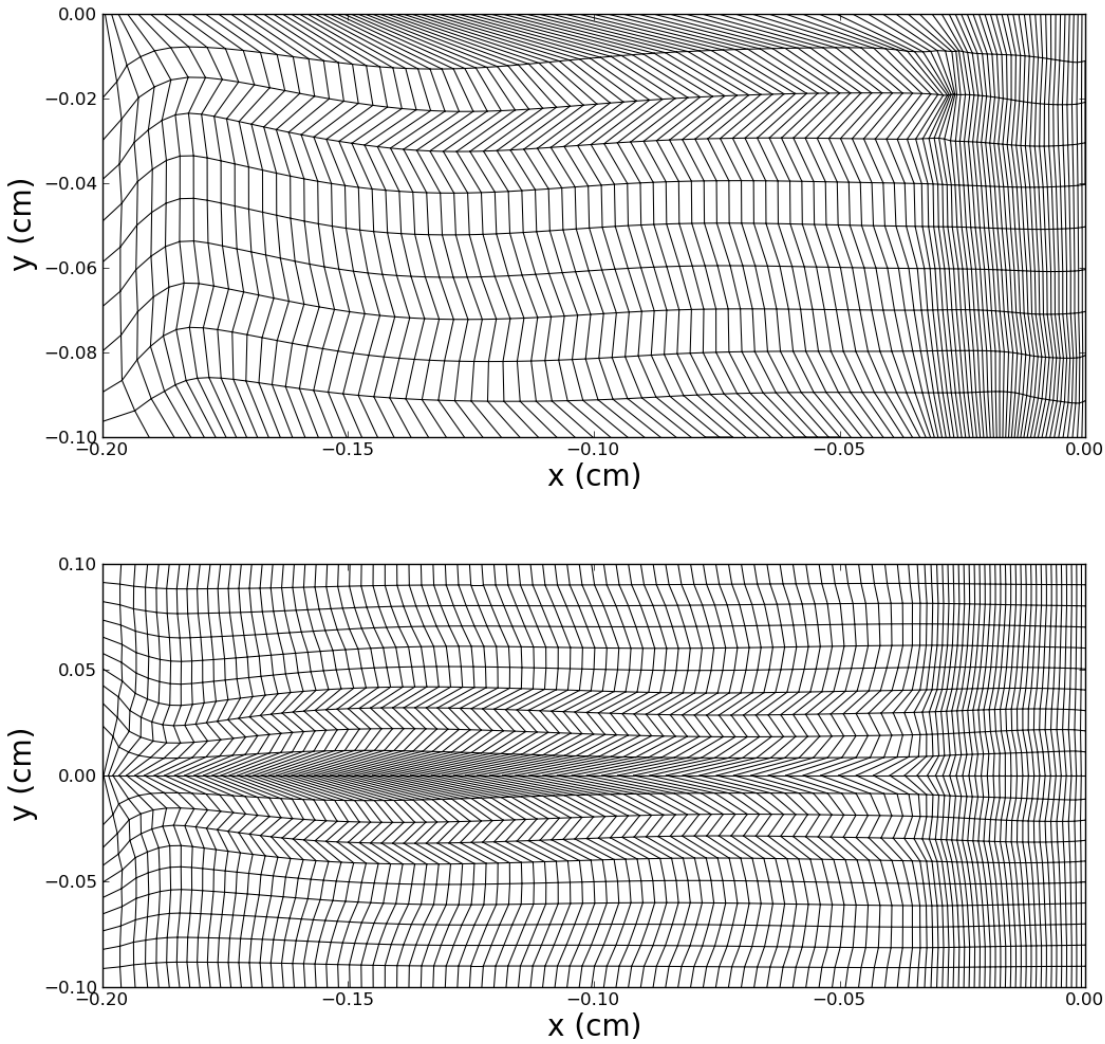

Figure 7.16: The SLaM-B method was able to compute the Saltzman problem past $t = 0.8$ μs before failing. The regions with the largest mesh distortions cause failure as the current algorithm cannot find a suitable vertex control volume among the nearest neighbors. Top: Original Saltzman mesh; Bottom: Gaussian mesh

CHAPTER VIII

A Second-order Method for Lagrangian Hydrodynamics

The SLaM method developed in the last chapter resists mesh imprinting, but it is still only first-order accurate. As discussed in Chapter I, the goal of this research was the development of a second-order accurate Lagrangian scheme. When constructing the first-order SLaM methods in Chapters VI and VII, care was taken to select discretization techniques that would produce the exact answer for linear data. The payoff for this prior work is realized in this chapter and it will be shown that SLaM is capable of second-order accuracy if the q-parameters are properly chosen. Then, some of the temporal flux limiting ideas from Chapter V will be used to produce a non-linear method. The SLaM method with temporal flux limiting will be referred to as SLaM-TFL. The ability of SLaM to achieve second-order convergence on the smooth Kidder like problem will be demonstrated in Section 8.1. In Section 8.2, a temporal limiting approach will be described. Numerical results from the Kidder, Sedov and Noh problems are presented in Section 8.4. Finally, some concluding comments are made in Section 8.5.

8.1 Second-order Accuracy

The q-parameters in the first-order implementation of the SLaM method, which will now be denoted as the cautious (C) choices, were specified as

$$\hat{Q}_p^C = \hat{Q}_V^C = \hat{\nu}, \text{ and } \quad \tilde{Q}_p^C = \tilde{\nu}. \tag{8.1}$$

Since the interpolation and differentiation operators used in Chapters VI and VII are exact for linear data, the only thing that must be done to achieve second-order accuracy is to compute the time averaged fluxes with to the trapezoidal rule. This is accomplished with the accurate (A) q-parameters

$$\hat{Q}_p^A = \hat{Q}_V^A = \hat{\nu}^2, \text{ and } \quad \tilde{Q}_p^A = \tilde{\nu}^2. \tag{8.2}$$

To verify that the implementation was carried out correctly, the accurate q-parameters were specified and a convergence analysis was performed with no limiting. Figure 8.1 shows that the accuracy has increased as expected. It should be stressed that the increased accuracy comes at no extra expense as compared to the first-order SLaM method and does not introduce any new complexity in unlimited form.

8.2 General Limiting Approach

The temporal approach used to limit point estimates of the acoustic fluxes in Chapter V was very successful. Similar ideas will be applied here to construct a SLaM method with temporal flux limiting, which will be referred to as SLaM-TFL. The Euler fluxes were obtained by individually computing the acoustic variables p, u and v and the practice of treating each separately will be carried over to the limiting. To make this possible, enhanced drivers were defined that incorporate a provisional solution just as was done for acoustics. They take the form $\beta^E = \beta^n + \kappa(\beta^* - \beta^n)$,

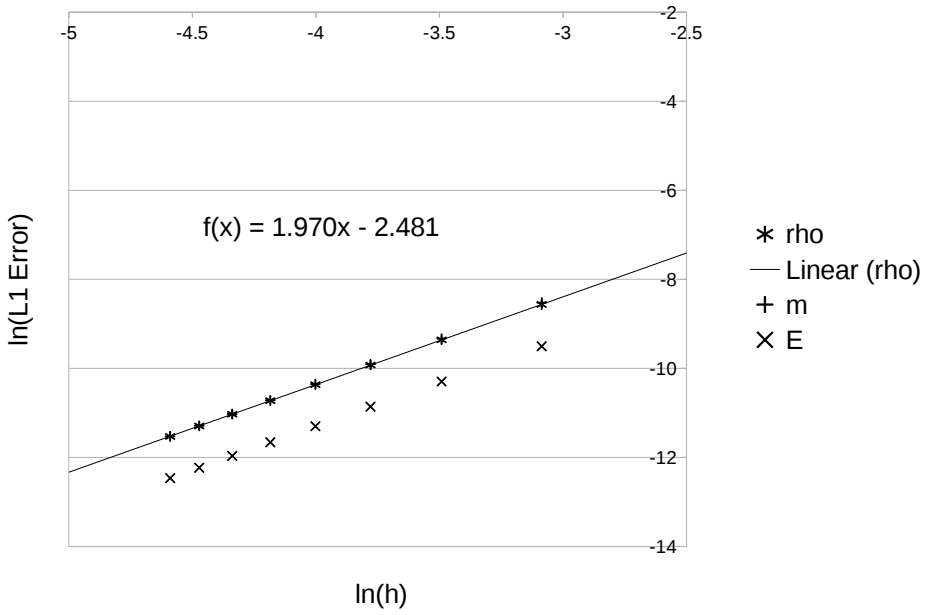

Figure 8.1: The SLaM method converges at second-order when the accurate q-parameters are used.

where the Lagrangian drivers are

$$\beta_p = \rho a^2(\Delta_x u + \Delta_y v), \quad \beta_u = \frac{1}{\rho}\Delta_x p, \quad \text{and} \quad \beta_v = \frac{1}{\rho}\Delta_y p.$$

Then the limited drivers are written

$$\beta^{lim} = F_0(\nu, \phi)\beta^n + F_1(\kappa, \phi)(\beta^* - \beta^n), \tag{8.3}$$

and used to defined antidiffusive flux components. For example, the limited vertex pressure will be

$$\hat{p}^A \to \hat{p}' = \hat{p}^C + \frac{(\hat{Q}_p^C - \hat{Q}_p^A)\hat{h}^2}{2\hat{a}^2\Delta t}\hat{\beta}_p^{lim}. \tag{8.4}$$

Initially, the indicator quantity ϕ and the general form of the functions F_0 and F_1 were carried over from acoustics. However, it was observed that the indicator quantity did not adequately highlight difficult regions of the flow, resulting in mesh imprinting and large overshoots when computing the Noh problem. See Figure 8.2.

One way to proceed is to add an additional indicator quantity, which can be taken as the ratio of the first two terms in the temporal expansions of the acoustic variables. This means two indicators are available, which, in terms of a generic quantity α, are

$$\phi_\alpha^A = \left|\frac{\alpha^n}{\Delta t \beta_\alpha^n}\right| \quad \text{and} \quad \phi_\alpha^B = \left|\frac{2\beta_\alpha^n}{(\beta_\alpha^* - \beta_\alpha^n)}\right|. \tag{8.5}$$

The limited drivers now become

$$\beta_\alpha^{lim} = F_0(\phi_\alpha^A, \nu)\beta_\alpha^n + F_1(\kappa, \phi_\alpha^B, \nu)(\beta_\alpha^* - \beta_\alpha^n).$$

and the empirical functions were taken to be

$$F_0(\phi_\alpha^A, \nu) = \min\left[1, \frac{\nu\phi_\alpha^A}{16}\right] \quad \text{and} \quad F_1(\kappa, \phi_\alpha^B, \nu) = \kappa \min\left[1, \frac{\nu\phi_\alpha^B}{8}\right]. \tag{8.6}$$

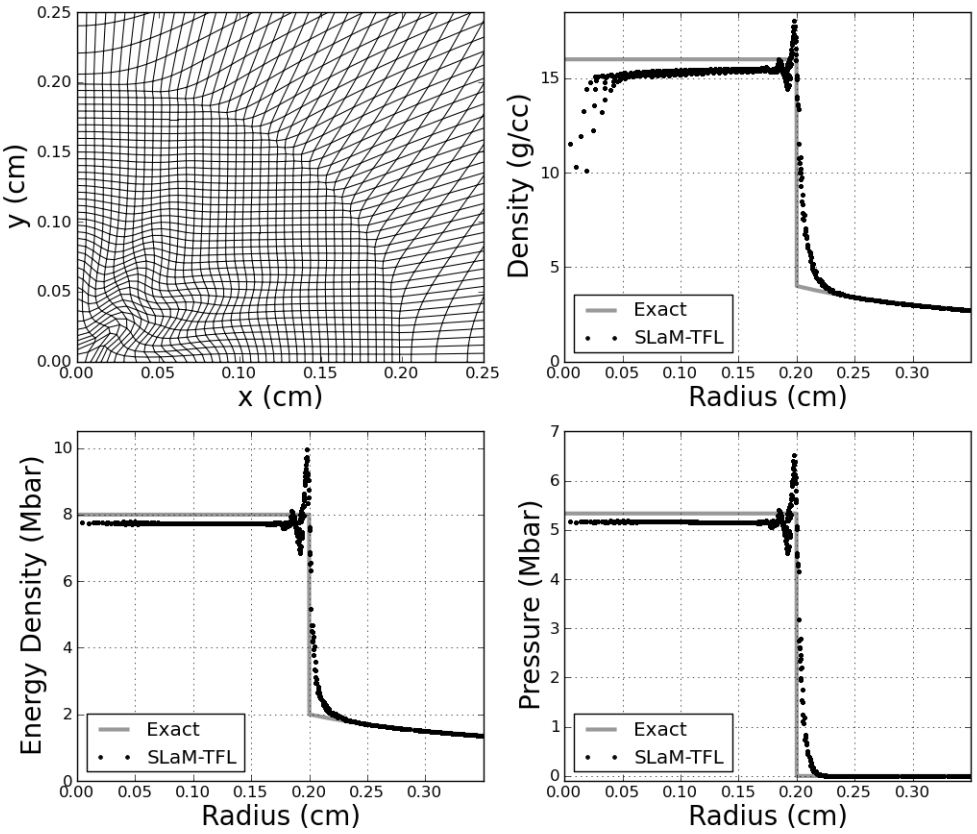

Figure 8.2: The indicator quantity used in the acoustic limiting was found to be lacking when implemented in the SLaM-TFL method.

In the implementation, each of these formulas evaluates to zero if the denominator in the respective indicator quantity is zero. There is one other special case for the functions F_0 when limiting the velocities. Here it does not make sense to automatically throw away the antidiffusive terms when the initial velocity component and, therefore, ϕ^A is zero. If this situation is detected, the higher-order terms are relied upon and F_0 is set equal to F_1.

8.3 SLaM-TFL Update Procedure

The SLaM-TFL method updates the solution according to the following procedure:

1. Interpolate estimates of $\hat{p}^n, \hat{u}^n, \hat{v}^n, \hat{\rho}^n$ and the sound speed, $\hat{a}^n$, at each vertex.

2. Interpolate initial estimates of $\tilde{p}^n$, $\tilde{u}^n$, $\tilde{v}^n$, $\tilde{\rho}^n$, and $\tilde{a}^n$ at each face.

3. Evolve cautious and accurate estimates of $\hat{p}^n$, $\hat{u}^n$, $\hat{v}^n$, and $\tilde{p}^n$ to the $n + 1/2$ time level.

4. Compute a provisional solution from the cautious flux components as in the previous chapter. The initial solution data is stored.

5. Compute the provisional driver quantities at faces and vertices.

6. Compute the limited driver quantities according to the methodology described in the previous section.

7. Perform a final update using the limited flux components according to

$$\mathbf{U}^{n+1} = \frac{1}{\mathcal{V}^{n+1}} \left[\mathcal{V}^n \mathbf{U}^n - \Delta t \left(\overline{\mathbf{F}} + \overline{\mathbf{G}} \right) \right] \tag{8.7}$$

where,

$$\mathbf{U} = (\rho, \rho u, \rho v, E), \tag{8.8a}$$

$$\overline{\mathbf{F}} = \left(0, \oint p' dS_x, 0, \oint p' u' dS_x\right)^T, \text{ and} \tag{8.8b}$$

$$\overline{\mathbf{G}} = \left(0, 0, \oint p' dS_y, \oint p' v' dS_y\right)^T. \tag{8.8c}$$

Again, the conceptual progression listed here does not represent an efficient implementation strategy. Note that the final update starts with the data at n, not the provisional solution as in FCT. This was done to simplify the initial implementation, but a true FCT update procedure could be used instead. The velocity divergence time step constraint was reduced in the second-order method ($K_{\Delta t} = 0.1$) to prevent excessive mesh distortions over a single time step.

8.4 Numerical Results

8.4.1 Sedov

Results from the Sedov problem show that the SLaM-TFL method computes solutions with very little mesh imprinting and the resolution is improved as compared to the first-order implementation. Figure 8.3 summarizes the solution obtained on a mesh with 50×50 cells per quadrant. The maximum CFL number was $\nu_m = 0.4$. In Figure 8.4, solutions are compared from the first-order SLaM method and the SLaM-TFL method. The second-order method has better resolution as expected and no unphysical features are found in the solution. However, the limiter appears to be overly aggressive when limiting the velocities, which is evidenced by the relatively small improvement in the SLaM-TFL density solution. Additional improvements to the limiter are left for future work. Finally, density solutions for three different mesh resolutions are shown in Figure 8.5, which demonstrate that the solution is converging

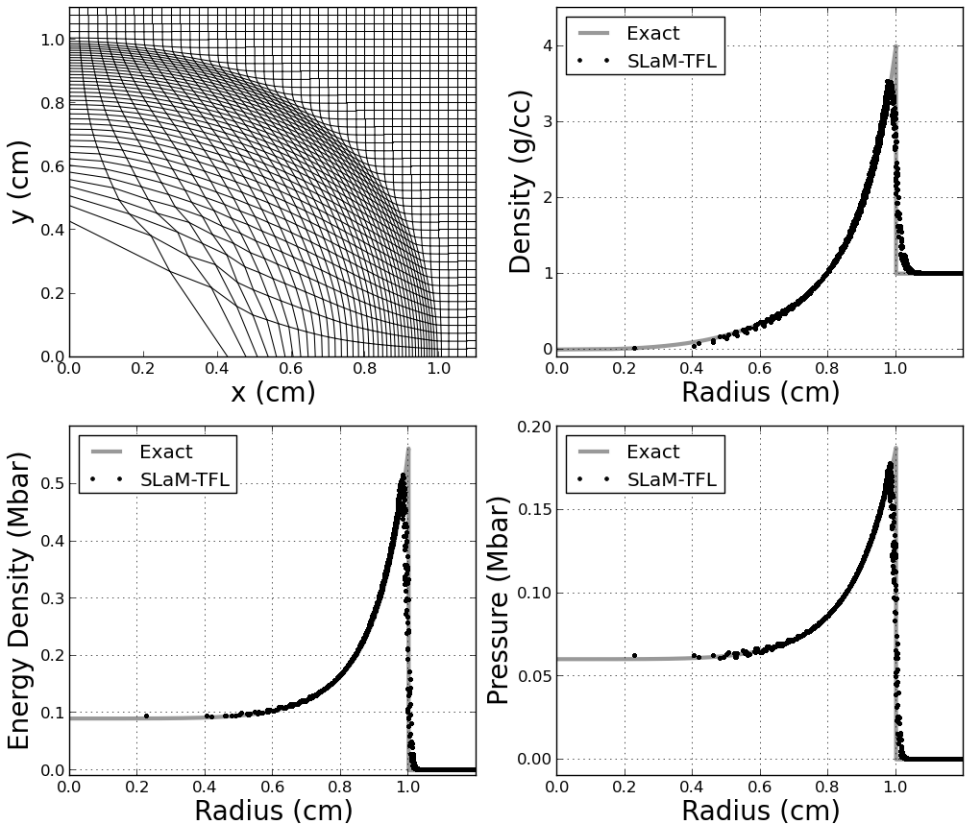

Figure 8.3: Sedov results generated by the SLaM-TFL method. ($\nu_m = 0.4$, 50×50 cells per quadrant)

toward the exact answer.

8.4.2 Noh

Figure 8.6 shows the Noh solution as computed by SLaM-TFL on a mesh with 50×50 cells per quadrant. The addition of the second indicator quantity has removed the mesh imprinting previously observed near the origin and substantially reduced the overshoots behind the shock wave. Overall, the solution possesses excellent radial symmetry and the wall heating is not excessive. As shown in Appendix B, the scatter in the post-shock density has been reduced by over 80 percent as compared to the

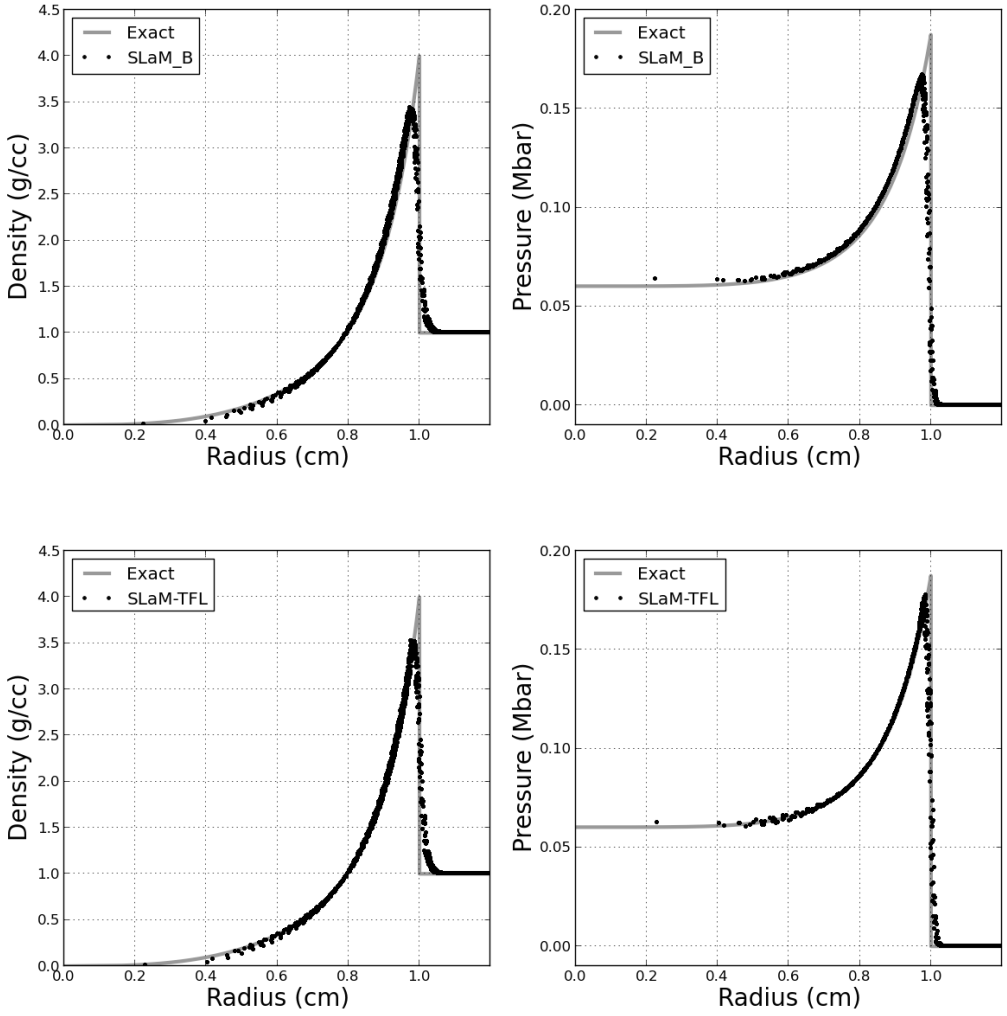

Figure 8.4: Coarse grid Sedov solutions demonstrate the improved accuracy of the SLaM-TFL method (25×25 cells per quadrant). Top: First-order SLaM; Bottom: SLaM-TFL

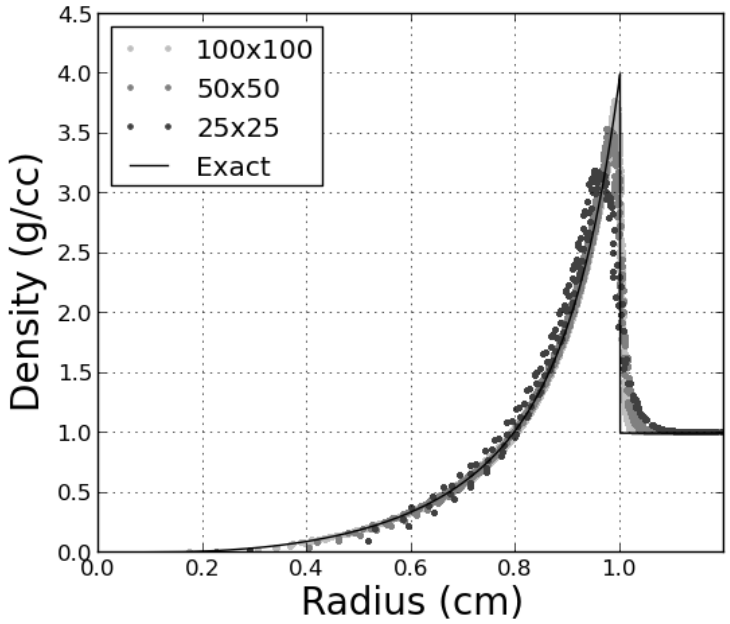

Figure 8.5: Sedov results generated by the SLaM-TFL method on three different meshes: 25×25, 50×50, and 100×100 cells per quadrant. ($\nu_m = 0.4$)

SLaM-A solution. Once again, solutions were computed on meshes with three different resolutions to demonstrate that the method is converging to the exact solution. See Figure 8.7. Finally, Figure 8.8 shows a solution computed on a finer mesh with 100×100 cells per quadrant and the maximum CFL number lowered to 0.1.

8.4.3 Limited Convergence

A convergence analysis was run with the limiter turned on using the two-dimensional Kidder like problem. Figure 8.9 shows that the method converges at second-order despite the active limiting mechanism.

8.5 Comments

The numerical results obtained with SLaM-TFL on the Sedov, Noh, and Kidder problems display very little mesh imprinting while achieving second-order accuracy in smooth regions of flow. In short, multidimensional design principles have led to

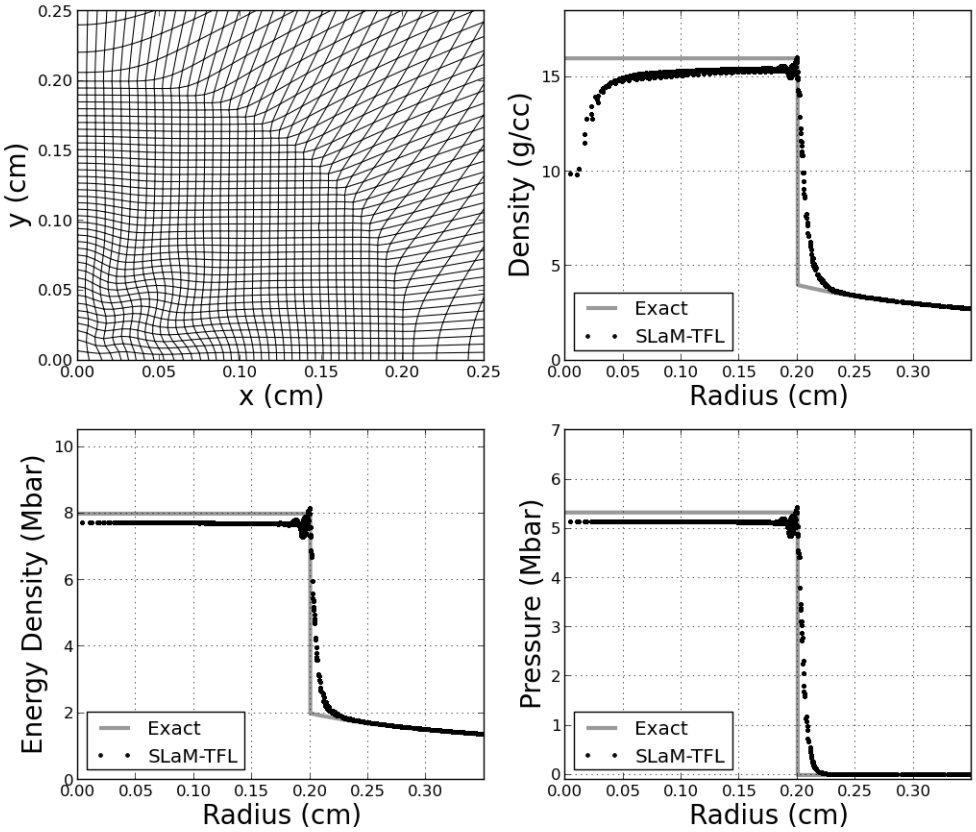

Figure 8.6: Noh results generated by the SLaM-TFL method show problems. ($\nu_m = 0.4$, 50×50 cells per quadrant)

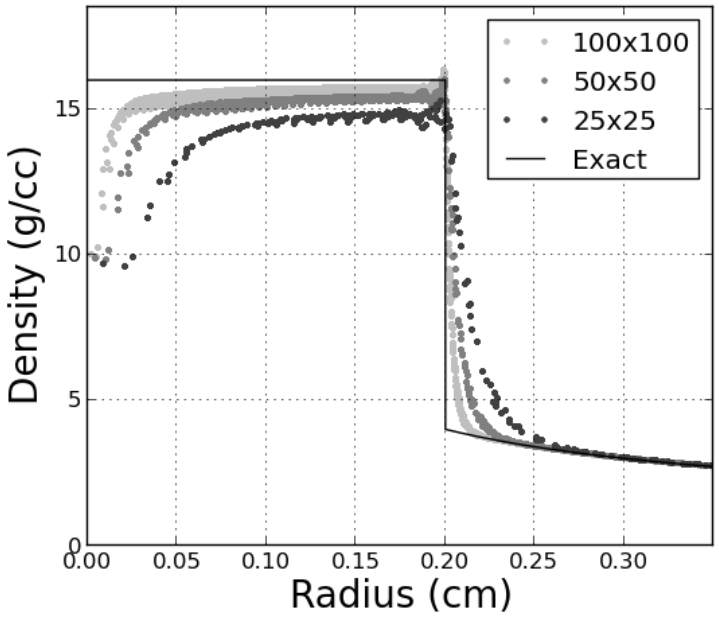

Figure 8.7: Noh results generated by the SLaM-TFL method on three different meshes: 25×25, 50×50, and 100×100 cells per quadrant. ($\nu_m = 0.4$)

a method that is simple and accurate. While the current results are very satisfying, more work remains. The method still needs to be made more robust on highly deformed meshes to overcome the problems documented in Chapter VII when computing the Saltzman problem. In addition, refinements to the limiting would be desirable, particularly with regard to the mechanism that modifies the velocities. The challenge is to design a limiter that is able to turn on when computing a problem like Noh where there is mesh movement in a fluid with zero pressure, but is not overly aggressive in less demanding situations.

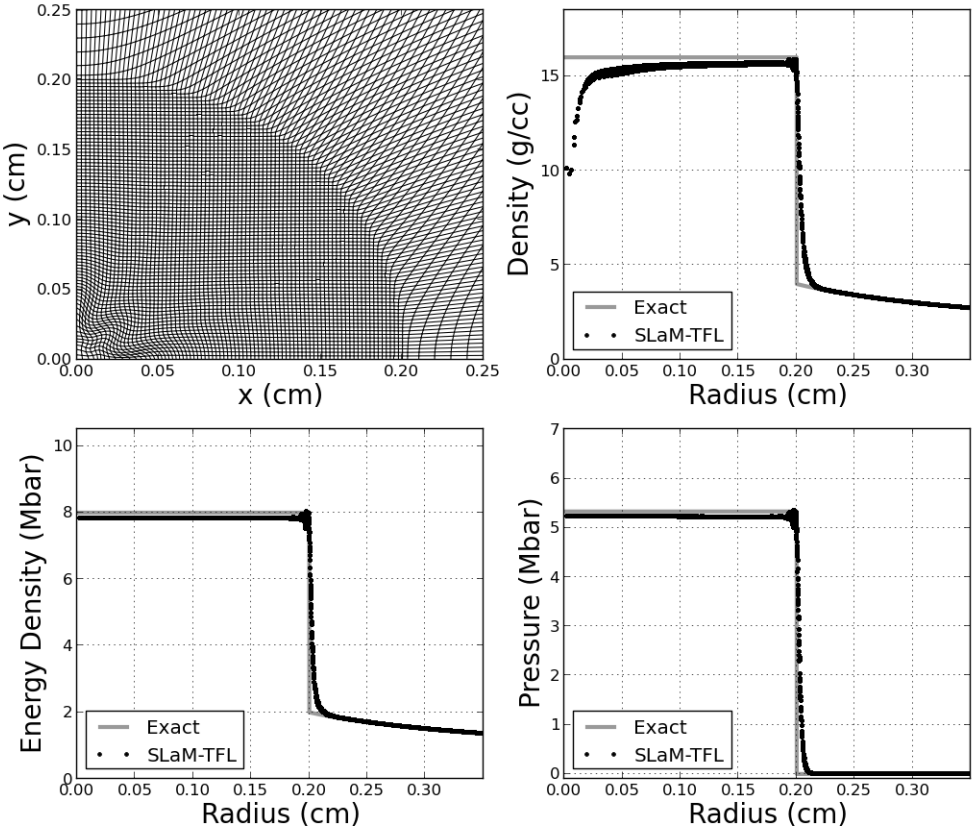

Figure 8.8: Noh results generated by the SLaM-TFL method on a mesh with 100×100 cells per quadrant and $\nu_m = 0.1$.

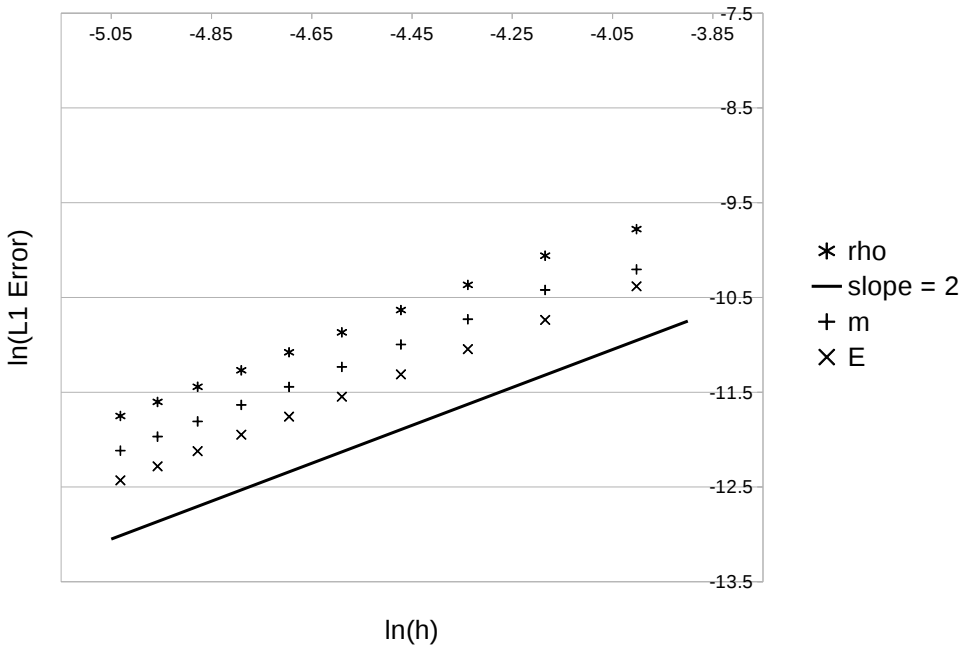

Figure 8.9: The SLaM-TFL method converges at second-order on the smooth Kidder type problem.

CHAPTER IX

Concluding Remarks and Future Work

The goal of the this thesis was to construct a simple, accurate Lagrangian hydrocode and much progress has been made. The preliminary investigation into acoustic algorithms proved to be much more interesting than expected and broadened the scope of the work done considerably. Chapter III led to a greater understanding of the multidimensional family of Lax-Wendroff schemes and showed that they have a poorer reputation than they deserve. If they are optimized with respect to a specific set of governing equations, they can be very effective. For example, the second-order accurate VPLW methods were able to resist mesh imprinting when computing radially symmetric problems on square meshes after the leading truncation errors had been made isotropic.

Chapter IV presented a straightforward and inexpensive method by which the accuracy of the VPLW schemes could be increased to third-order. The third-order methods used the optimized evolution operators developed in Chapter III, and, therefore, preserved vorticity and resisted mesh imprinting. However, the dispersion properties of the third-order methods were far superior to the those of the second-order ones, especially at low CFL numbers. The large improvement in the solution to problems with high frequency waves, even with no limiting, provides (more) strong evidence in favor of the utility of third-order methods.

In Chapter V, two VPFCT methods were developed that converged with at least second-order accuracy in smooth regions of the flow. A flux limiting approach was presented that used the third-order methods for inspiration. This limiter modified point estimates of the fluxes by considering temporal information in the form of a provisional first-order solution and a smoothness monitor. The limiter can be thought of as zero-dimensional with respect to space as a consequence of this construction; it has a compact structure that is not influenced by the surrounding mesh geometry. Since *a priori* bounds on the solution were not available, empirically determined functions were relied upon. The VPFCT method was able to retain the resistance to mesh imprinting and desirable dispersion properties of the third-order methods from Chapter IV, but spurious features were prevented.

The focus returned to Lagrangian hydrodynamics in Chapter VI and the acoustic RR method was extended to solve the Euler equations on a moving mesh with first-order accuracy. Point estimates of the fluxes were computed at cell vertices using a Lax-Wendroff type procedure. The fluxes automatically defined the mesh motion, and were integrated according to the trapezoid rule. Despite the simple construction of the method, solutions were successfully computed to the Noh and Sedov problems on Cartesian meshes. However, substantial mesh imprinting occurred on the Noh problem. The results showed that while the the approach to cell-centered Lagrangian hydrodynamics originally proposed by Roe held promise, additional work was needed.

Refinements to the Lagrangian method were presented in Chapter VII, resulting in the first-order SLaM method. Since undesirable mesh imprinting, accompanied by spurious vorticity, was encountered in Chapter VI, multidimensional point estimates of the pressure were introduced at faces. Furthermore, the interpolation and flux evolution procedures were made more robust for cases where the mesh is highly deformed. These additions greatly improved the performance of the algorithm without requiring any fundamental changes to the SLaM method.

Finally, it was verified in Chapter VIII that SLaM is capable of second-order accuracy with only a simple change of the q-parameters. Some limiting work was presented that followed the acoustic strategy, but introduced an additional indicator quantity. The resulting SLaM-TFL method produced excellent results on the Sedov and Noh problems. In addition, the method converged at second-order when computing the smooth Kidder like problem.

The beauty of the SLaM methods presented here are truly in their simplicity. One-dimensional ideas were totally and painstakingly avoided and the payoff was substantial: The SLaM method has everything it needs to respect multidimensional physics and resist mesh imprinting and nothing that it does not. It may seem somewhat ironic that so much work was required to produce simple, accurate algorithms. This fact is well summed up by a quote from Truesdell [81] who noted that, "Simplicity does not come of itself but must be created."

9.1 Suggestions for Future Work

9.1.1 Lax-Wendroff Methods

Much more investigation into Lax-Wendroff methods could be done. In particular, the LW family could be optimized with respect other sets of governing equations to better understand the full scope of its usefulness. Furthermore, a more comprehensive analysis of the work done in this thesis in comparison with other efforts in the literature could be completed to uncover similarities and differences. These efforts could produce a comprehensive and authoritative work on Lax-Wendroff methods.

9.1.2 Limiting

The potential for more investigation into the limiting ideas presented here is nearly endless. More test problems could be run to gain a better understanding of the ca-

pabilities of the current procedure. Additional work with the the acoustic system could be aimed at motivating better forms of the limiting functions F_0 and F_1. Or, perhaps more insight could be obtained by performing traditional analyses on a problem where the solution is bounded (e.g. linear advection). Investigation into ways to extend the temporal limiting ideas to more complicated sets of equations and other methods could also be done.

9.1.3 SLaM Method

The primary remaining concern with the current SLaM method is the breakdown in the robustness of the flux prediction formulas on highly deformed grids. One way to address the robustness of the interpolation and flux evolution procedures could be to design a more intricate mechanism that is able to find appropriate neighboring information for interpolation and to identify a valid control volume. A procedure that smoothly adjusts the data used by the flux formulas as vertices cross control volume boundaries should be sought. Another way to proceed would be to extend the method to work on unstructured triangular meshes as the control volumes specified by the mesh connectivity would always be valid in this case. This would primarily be a coding exercise; the operators used to construct the SLaM method do not care what the mesh looks like. More investigation into limiting techniques is warranted. Mechanisms that can deal with challenging flows like the Noh problem, but adequately relax in less challenging flows should be sought. Finally, the method could be extended to include a mesh remapping capability, to compute problems in $r - z$ coordinates, or to solve problems involving multiple materials or strength.

APPENDICES

APPENDIX A

The Vorticity Transport Equation for Inviscid Gas Dynamics

Often the vorticity transport equation is written down with no explanation for its origins beyond that it is the obtained by "taking the curl of the momentum equation". Here we will take the time to derive it for an unsteady, compressible, inviscid fluid. Perhaps someone will find the derivation useful. The starting point is the momentum equation, which in conservative form is

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_l}{\partial x_l} + \frac{\partial p}{\partial x_i} = 0 \tag{A.1}$$

using index notation. Before attempting to take the curl, it will be useful to rewrite (A.1) in primitive form. To do this, the first two terms can be expanded with the product rule to give

$$\rho \left(\frac{\partial u_i}{\partial t} + u_l \frac{\partial u_i}{\partial x_l} \right) + \rho u_i \frac{\partial u_l}{\partial x_l} + u_i \frac{\partial \rho}{\partial t} + u_i u_l \frac{\partial \rho}{\partial x_l} + \frac{\partial p}{\partial x_i} = 0. \tag{A.2}$$

Now recall that the continuity equation is

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_l}{\partial x_l} = \frac{\partial \rho}{\partial t} + u_l \frac{\partial \rho}{\partial x_l} + \rho \frac{\partial u_l}{\partial x_l} = 0,$$

and use it to substitute for the time derivative of the density in (A.2). After some cancellation we have the primitive momentum equation

$$\frac{\partial u_i}{\partial t} + u_l \frac{\partial u_i}{\partial x_l} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} = 0. \tag{A.3}$$

Now, the curl of a vector a_i can be represented as

$$\nabla \times \mathbf{a} = \frac{\partial}{\partial x_j} a_i \epsilon_{jik},$$

where ϵ_{jik} is the Levi-Civita symbol. Using this notation, the curl of (A.3) is

$$\frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial t} \epsilon_{jik} + \frac{\partial}{\partial x_j} \left(u_l \frac{\partial u_i}{\partial x_l} \right) \epsilon_{jik} + \frac{\partial}{\partial x_j} \left(\frac{1}{\rho} \frac{\partial p}{\partial x_i} \right) \epsilon_{jik} = 0. \tag{A.4}$$

The first term on the left hand side (LHS) is easily rewritten as

$$\boxed{\frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial t} \epsilon_{jik} = \frac{\partial}{\partial t} \frac{\partial u_i}{\partial x_j} \epsilon_{jik} = \frac{\partial \omega_k}{\partial t}.} \tag{A.5}$$

The third term is also straight forward. Using the product rule and the identity $\nabla \times \nabla \phi = 0$, we have

$$\boxed{\frac{\partial}{\partial x_j} \left(\frac{1}{\rho} \frac{\partial p}{\partial x_i} \right) \epsilon_{jik} = \left(\frac{\partial}{\partial x_j} \frac{1}{\rho} \right) \left(\frac{\partial p}{\partial x_i} \right) \epsilon_{jik}.} \tag{A.6}$$

The second term in (A.4) requires more work. Here, we follow the procedure shown in [78]. Recalling the vector identity

$$\mathbf{A}\cdot\nabla\mathbf{A} = \frac{1}{2}\nabla(\mathbf{A}\cdot\mathbf{A}) + (\nabla\times\mathbf{A})\times\mathbf{A},$$

we can rewrite the term as

$$\frac{\partial}{\partial x_j}\left(u_l\frac{\partial u_i}{\partial x_l}\right)\epsilon_{jik} = \frac{\partial}{\partial x_j}\left[\frac{1}{2}\frac{\partial u_l u_l}{\partial x_i} + \left(\left(\frac{\partial u_m}{\partial x_l}u_m\epsilon_{lmk}\right)u_n\epsilon_{kni}\right)\right]\epsilon_{jio}. \tag{A.7}$$

Again using the identity $\nabla\times\nabla\phi = 0$, (A.7) is

$$\frac{\partial}{\partial x_j}\left(u_l\frac{\partial u_i}{\partial x_l}\right)\epsilon_{jik} = \frac{\partial}{\partial x_j}\left[(\omega_k u_n\epsilon_{kni})\right]\epsilon_{jio}.$$

One final vector identity is necessary to further modify the previous expression: $\nabla\times(\mathbf{A}\times\mathbf{B}) = \mathbf{A}(\nabla\cdot\mathbf{B}) - \mathbf{B}(\nabla\cdot\mathbf{A}) + (\mathbf{B}\cdot\nabla)\mathbf{A} - (\mathbf{A}\cdot\nabla)\mathbf{B}$. We now have

$$\begin{aligned}\frac{\partial}{\partial x_j}\left(u_l\frac{\partial u_i}{\partial x_l}\right)\epsilon_{jik} &= \frac{\partial}{\partial x_j}\left[(\omega_k u_n\epsilon_{kni})\right]\epsilon_{jio}\\ &= \omega_k\frac{\partial u_l}{\partial x_l} - u_k\frac{\partial\omega_l}{\partial x_l} + u_l\frac{\partial}{\partial x_l}\omega_k - \omega_l\frac{\partial}{\partial x_l}u_k.\end{aligned}$$

Since $\nabla\cdot\nabla\times\mathbf{a} = 0$, we finally have

$$\boxed{\frac{\partial}{\partial x_j}\left(u_l\frac{\partial u_i}{\partial x_l}\right)\epsilon_{jik} = \omega_k\frac{\partial u_l}{\partial x_l} + u_l\frac{\partial}{\partial x_l}\omega_k - \omega_l\frac{\partial}{\partial x_l}u_k.} \tag{A.8}$$

Taking (A.4), (A.5), (A.6), and (A.8) together the vorticity transport equation is

$$\frac{\partial\omega_k}{\partial t} + u_l\frac{\partial\omega_k}{\partial x_l} = -\omega_k\frac{\partial u_l}{\partial x_l} + \left(\omega_l\frac{\partial}{\partial x_l}\right)u_k - \left(\frac{\partial}{\partial x_j}\frac{1}{\rho}\right)\left(\frac{\partial p}{\partial x_i}\right)\epsilon_{jik}, \tag{A.9}$$

which can be written in vector form as

$$\frac{D\boldsymbol{\omega}}{Dt} = -\boldsymbol{\omega}(\nabla \cdot \mathbf{V}) + \boldsymbol{\omega}(\nabla \cdot \mathbf{V}) + \frac{1}{\rho^2}\nabla\rho \times \nabla p. \tag{A.10}$$

APPENDIX B

Quantifying Anisotropy

Radial plots were used throughout the thesis to help assess the ability of numerical methods to preserve radial symmetry and resist mesh imprinting. Here a simple procedure is presented that allows the amount of scatter in a radial plot to be quantified. The procedure is then used to compare the performance of the acoustic and Lagrangian methods.

Methodology

A simple measure of the amount of anisotropy in a numerical solution is sought for the case when the exact solution should possess radial symmetry. To facilitate this, a sample region was defined in the computational domain according to $r \in [r_{min}, r_{max}]$ and $\psi \in [0, \pi/4]$, where ψ is the angle measured counterclockwise from the positive x-axis. Next, the sample space was divided into bins with constant spacing along the radial direction. In statistics the standard deviation is commonly used as a measure of the amount of dispersion in a data set. However, here some amount of dispersion in the data is correct (the exact solution in each bin is not constant) and so comparing each solution point with the mean value in a bin would be inappropriate.

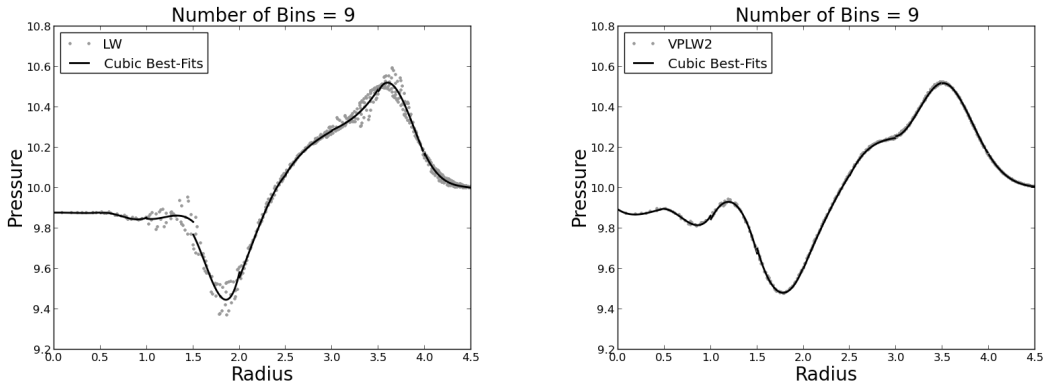

Figure B.1: Local cubic fits of the solution data were used as a reference from which to quantify deviations from radial symmetry.

Instead, a least-squares procedure[1] was used to fit a cubic function, f, to the data in each bin. Then a standard deviation-like quantity η_i can be computed in the ith bin by comparing each data point from the numerical solution, s_j, with the locally reconstructed function. The definition for η_i in a bin with N data points is

$$\eta_i = \left\{ \frac{1}{N} \sum_{j=1}^{N} [s_j - f(r_j)]^2 \right\}^{1/2} . \tag{B.1}$$

Acoustic Methods

Figure B.1 shows acoustic solutions from the LW and VPLW2 methods that have been separated into nine bins over the range $r_{min} = 0$ to $r_{max} = 4.5$. The minimum number of points in a single bin in the case of the discontinuous test problem and a 100×100 mesh was eight.

A summary of the η_i values for the acoustic data on the discontinuous test problem is shown in Figure B.2. Inspection of the figure shows that the RR solution had over 50 percent less scatter in the solution than the original LW method. Furthermore,

[1]The SciPy function polyfit() was used to calculate the polynomial coefficients.

the VPLW2 method and the resulting third-order and limited methods all had over 80 percent less scatter than the original LW method.

Lagrangian Methods

The analysis was carried out with the SLaM-A, SLaM-B, and SLaM-TFL methods on the Noh problem. The post-shock solutions for each method were analyzed by specifying 5 bins from $r_{min} = 0.2$ to $r_{max} = 0.2$. Figure B.3 shows the the polynomial fits for the SLaM-A and SLaM-TFL methods in the sample region.

A summary of the η_i values for the SLaM methods is shown in Figure B.4. The first-order SLaM-B method had over 75 percent less scatter in the post-shock solution than the SLaM-A method. Similarly, the limited second-order SLaM-TFL method removed over 80 percent of the scatter from the SLaM-A solution.

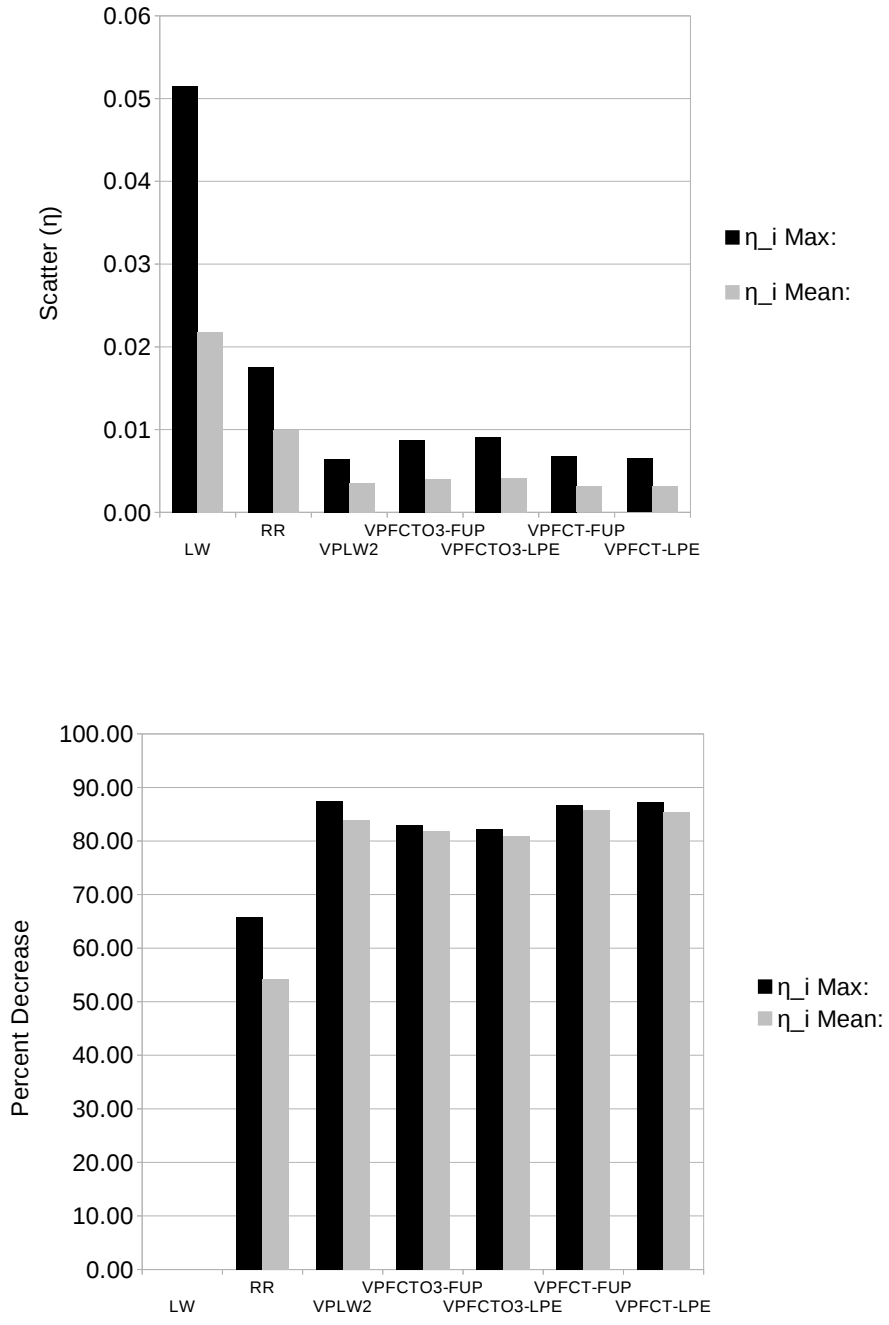

Figure B.2: The average and maximum η_i values for the acoustic methods on the discontinuous test problem are shown along with the percentage decrease in each parameter relative to the original LW method.

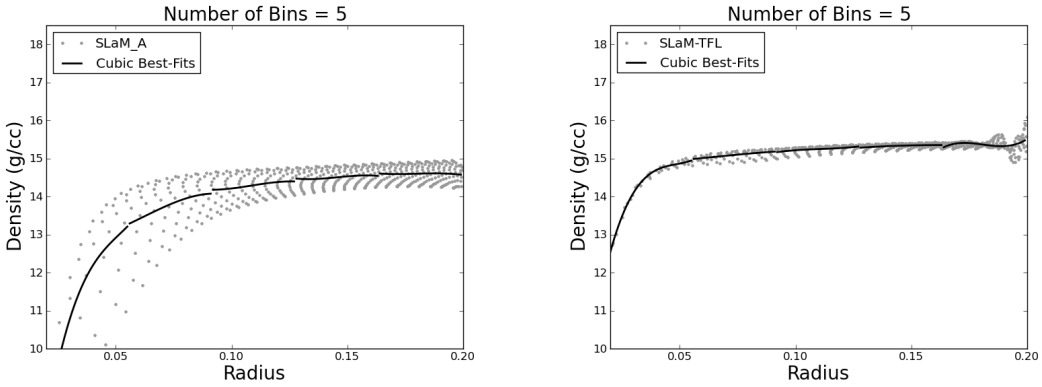

Figure B.3: Five bins were used to quantify the deviations from radial symmetry in the post-shock Noh solution.

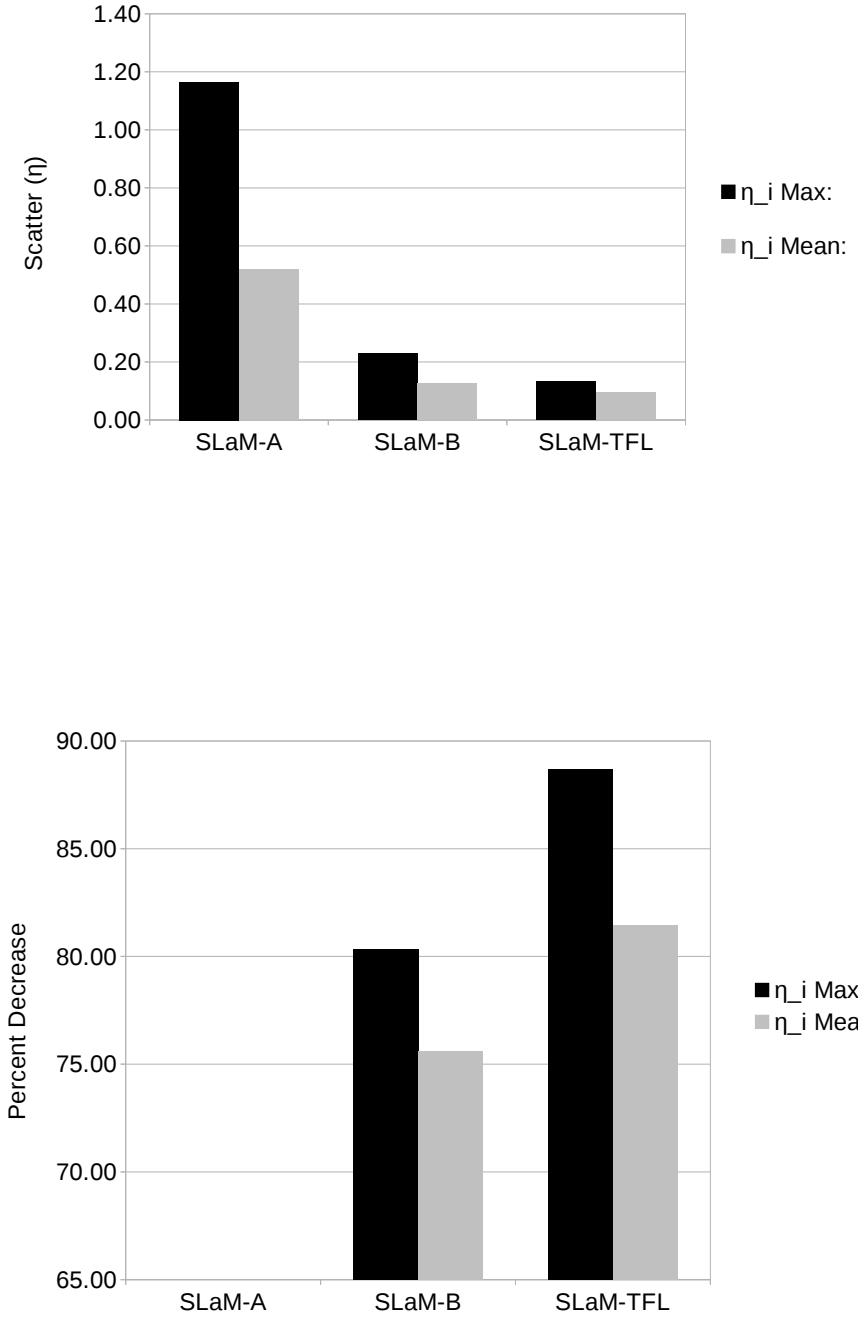

Figure B.4: The average and maximum η_i values for the SLaM methods on the Noh problem are shown along with the percentage decrease in each parameter relative to SLaM-A.

APPENDIX C

Interpolation

Interpolation at Vertices

Best-fit planes were used to interpolate estimates of p, u, v, ρ, and a at vertices from the cell-centered data in both the SLaM-A and SLaM-B methods. On a quadrilateral mesh and for a quantity of interest α, four data points (x_i, y_i, α_i) are available. We seek a best-fit plane of the form

$$\alpha(x, y) = Ax + By + C, \tag{C.1}$$

where the coefficients are chosen to minimize the error in a least-squares sense. The coefficients can be computed from the normal equations which, in the present case, corresponds to solving the system

$$\underline{A}\mathbf{x} = \mathbf{b}, \tag{C.2}$$

where

$$\underline{A} = \begin{pmatrix} \sum_i^N x_i^2 & \sum_i^N x_i y_i & \sum_i^N x_i \\ \sum_i^N y_i x_i & \sum_i^N y_i^2 & \sum_i^N y_i \\ \sum_i^N x_i & \sum_i^N y_i & \sum_i^N 1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} A \\ B \\ C \end{pmatrix}, \text{ and } \qquad \mathbf{b} = \begin{pmatrix} x_i \alpha_i \\ y_i \alpha_i \\ \alpha_i \end{pmatrix}.$$

Since the system of of equations is only 3×3, the solution is easy to write down explicitly and hard code. More details on the mathematical basis for this procedure can be found in [77].

Interpolation at Faces

A linear interpolation was used to obtain estimates of the required face quantities in the SLaM-B method. A local one-dimensional η coordinate system was established between the cells opposite of each face as shown in Figure C.1. The vectors $\mathbf{f}$ and $\mathbf{r}$ were defined according to the figure in the local coordinate system and then the projection $\mathbf{f}' = \mathbf{f} \cdot \mathbf{r}$ was computed. The length of the vector $\mathbf{f}'$ is was taken to be the local coordinate of the face and a standard linear interpolation was carried out with the cell-centered data. If the the mesh was so deformed that the projection $\mathbf{f}'$ was negative, than the interpolated values were taken to be the arithmetic mean of the adjacent cells specified by the mesh connectivity.

Interpolation Robustness at Vertices

In the SLaM-A method, the only check on the interpolation procedure was to determine if the vertex was outside its control volume. If this was the case, the interpolated values were taken to be the arithmetic mean of the cell-centered values. In the SLaM-B method, a switching routine was implemented to use a different control volume if a vertex did not lie in the control volume specified by the mesh connectivity.

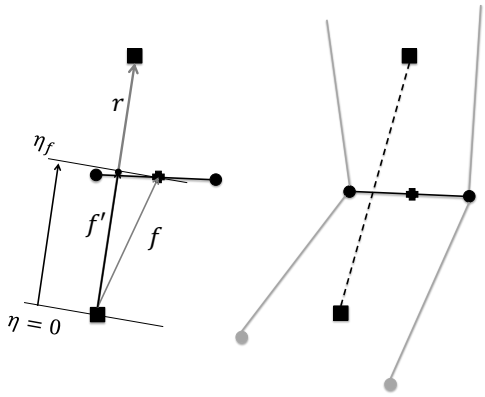

Figure C.1: A linear interpolation was used to infer values at faces from the cell-centered data in the SLaM-B method.

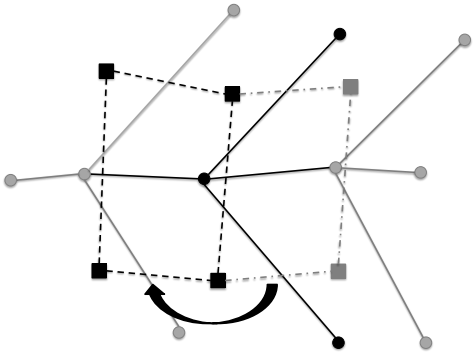

Figure C.2: A different control volume was sought from the neighboring data if a vertex did not fall in the original one specified by the mesh connectivity.

The control volumes belonging to the neighboring eight vertices were searched to find an acceptable one as shown in Figure C.2. Interpolation was then performed on the new control volume. If still no suitable control volume was found among the neighboring data, then the interpolated values were again taken to be the arithmetic mean.

APPENDIX D

Approximating First Derivatives with Second-order Accuracy on Deformed Meshes

Second-order accurate approximations for first-derivatives were needed to construct the Lagrangian algorithms presented in this thesis. Consider the quadrilateral shown in Figure D.1 and form the vectors $\mathbf{q}_0 = \mathbf{v}_2 - \mathbf{v}_0$ and $\mathbf{q}_1 = \mathbf{v}_3 - \mathbf{v}_1$. The required approximations can be easily obtained by taking advantage of the well known formula for the area of a quadrilateral

$$A = \frac{1}{2}(\mathbf{q}_0 \times \mathbf{q}_1). \tag{D.1}$$

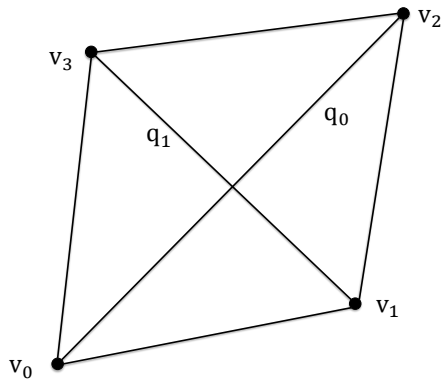

Figure D.1: Notation for the vertices and diagonals of a quadrilateral.

A function could be defined, call it $S(x, y)$, that outputs A given the vertex coordinates. Now consider $S(x, \phi)$, where ϕ is some scalar quantity that is defined at the vertices of the quadrilateral. Some simple experimentation will shown that if

$$\phi = \text{const.}, \text{ then } S(x, \phi) = 0$$
$$\text{and if } \phi = x, \text{ then } S(x, \phi) = 0.$$

Of course, it is also true that

$$\begin{aligned} &\text{if } \phi = y, \text{ then } S(x, \phi) = A, \text{ and} \\ &\text{if } \phi = f(x, y), \text{ then } S(x, \phi) = \text{const.}; \end{aligned} \tag{D.2}$$

by inspection it is clear that for any *linear* function

$$S(x, \phi) = A\frac{\partial \phi}{\partial y}. \tag{D.3}$$

It follows that

$$S(\phi, y) = A\frac{\partial \phi}{\partial x},$$

and the necessary second-order accurate approximations are

$$\frac{S(\phi, y)}{S(x, y)} = \frac{\partial \phi}{\partial x} + O(h^2) \text{ and } \frac{S(x, \phi)}{S(x, y)} = \frac{\partial \phi}{\partial y} + O(h^2). \tag{D.4}$$

APPENDIX E

Initializing the Noh Problem After Shock Formation

The most challenging part of the Noh problem occurs during start-up when the initial velocity discontinuity at the origin gives rise to a shock wave. The shock must bring the flow and, therefore, the mesh to rest. A pressure rise is required to stop the flow, but the initial fluid has zero internal energy. As a result, it is difficult to prevent spurious mesh motions from occurring while the fluid "heats up" and develops a non-zero pressure. It might be interesting, then, to initialize the Noh problem at a later time with the exact post-shock data and observe the results.

Such an experiment was run with the SLaM-B method. The exact solution was initialized at $t = 0.3$ μs on a square mesh and then computed until $t = 1$ μs. In cells that contained the shock wave, averaged data was initialized according to the fraction of the cell area in the pre- and post-shock states. Numerical results show that start-up errors still occur in this scenario. The initial data does not represent the shock exactly as the method would prefer and so waves are shed back into the exact post-shock state while the method resolves the assumed discrepancy. During this process wall heating like entropy errors are made at the initial shock location. See Figures E.1 and E.2. Additionally, small amounts of vorticity are introduced into

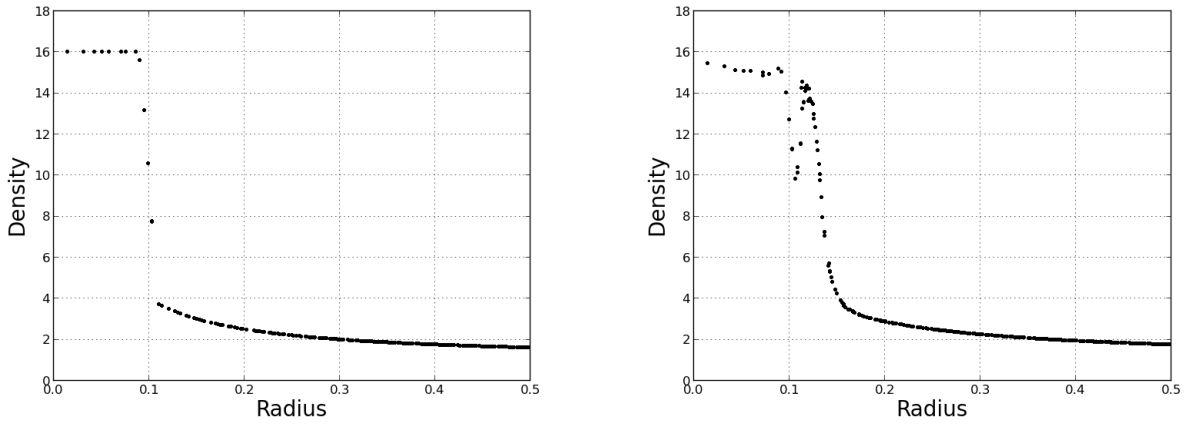

Figure E.1: Density profiles early in the solution: Left - The Noh problem is initialized at $t = 0.3$ μs on a square mesh. Right - Start up errors still occur in this scenario that include transient acoustic waves and a wall heating like entropy error.

the data by the averaging process used for cells that initially contain the shock. This vorticity persists and slowly damages radial symmetry over time.

Errors in the pressure are self-healing as they are eliminated over time by transient acoustic waves. The density also tends toward the exact solution over time with the exception of the entropy and vorticity errors. Both the entropy and circulation are transported along streamlines and so the Lagrangian method treats the errors as if they were physical and propagates them with no damping. Figure E.3 shows the density and pressure profiles late in the solution. The vorticity errors are observed here in the density solution by scatter. This example illustrates one of the difficulties associated with solving the gas dynamics equations in the Lagrangian frame: entropy and vorticity errors are unforgiving.

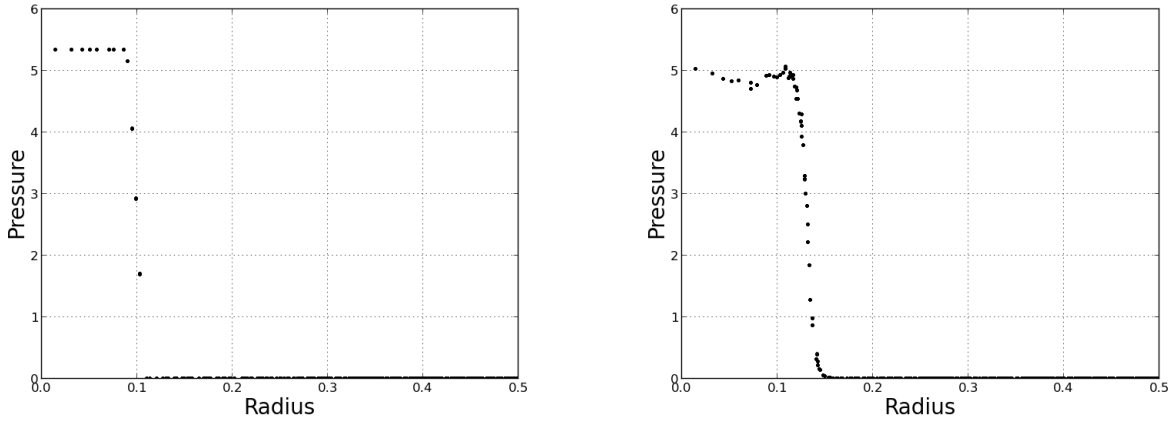

Figure E.2: Pressure profiles early in the solution: Left - The Noh problem is initialized at $t = 0.3$ μs on a square mesh. Right - The transient acoustic waves that are produced early in the solution are evident in the pressure profile.

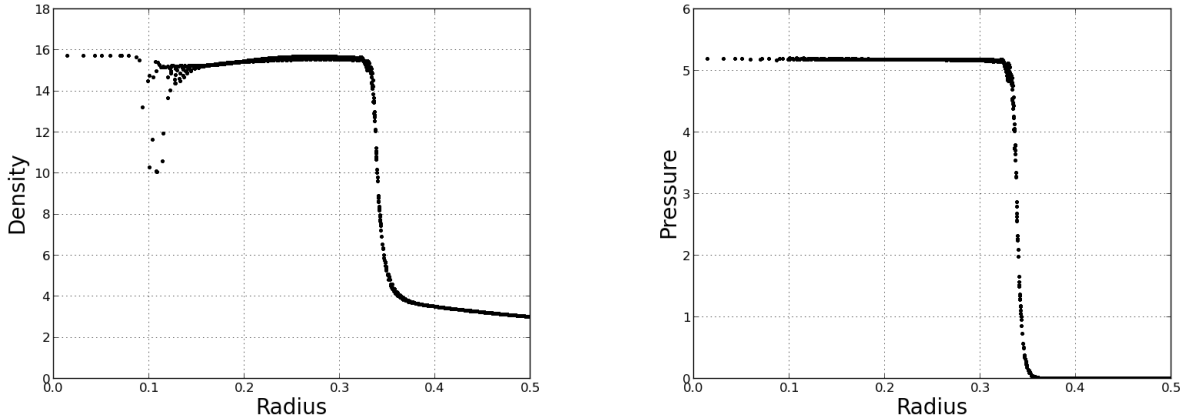

Figure E.3: Density and pressure profiles late in the solution: Left - While the overall density profile tends toward the exact solution over time, the entropy and vorticity errors persist. Right - Acoustic waves are able to remove errors in the pressure over time.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] F.L. Addessio, J.R. Baumgardner, J.K. Dukowicz, N.L. Johnson, B.A. Kashiwa, R.M. Rauenzahn, and C. Zemach. CAVEAT: A computer code for fluid dynamics problems with large distortion and internal slip. Technical report, Los Alamos National Laboratory, LA-10613-MS, Rev. 1, 1992.

[2] M. Arora and P. L. Roe. A well-behaved TVD limiter for high-resolution calculations of unsteady flow. *Journal of Computational Physics*, 132(1):3 – 11, 1997. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.1996.5514. URL `http://www.sciencedirect.com/science/article/pii/S002199919695514X`.

[3] A.J. Barlow. A high order cell centred dual grid Lagrangian Godunov scheme. *Computers & Fluids*, 83(0):15 – 24, 2013. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2013.02.009. URL `http://www.sciencedirect.com/science/article/pii/S0045793013000649`. Numerical methods for highly compressible multi-material flow problems.

[4] A.J. Barlow and P.L. Roe. A cell centred Lagrangian Godunov scheme for shock hydrodynamics. *Computers & Fluids*, 46(1):133 – 136, 2011. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2010.07.017. URL `http://www.sciencedirect.com/science/article/pii/S0045793010001921`. 10th {ICFD} Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010).

[5] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA paper*, pages 89–0366, 1989.

[6] A.L. Bauer, R. Loubère, and B. Wendroff. On stability of staggered schemes. *SIAM Journal on Numerical Analysis*, 46(2):996–1011, 2008. doi: 10.1137/060660151. URL `http://dx.doi.org/10.1137/060660151`.

[7] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(23):235–394, 1992. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/0045-7825(92)90042-I.

[8] J. P. Boris and D. L. Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11(1):38 – 69, 1973. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(73)90147-2. URL `http://www.sciencedirect.com/science/article/pii/0021999173901472`.

[9] D. Bouche, G. Bonnaud, and D. Ramos. Comparison of numerical schemes for solving the advection equation. *Applied Mathematics Letters*, 16(2):147 – 154, 2003. ISSN 0893-9659. doi: http://dx.doi.org/10.1016/S0893-9659(03)80024-1. URL http://www.sciencedirect.com/science/article/pii/S0893965903800241.

[10] P.L. Browne and K.B. Wallick. The reduction of mesh tangling in two-dimensional Lagrangian hydrodynamics codes by the use of viscosity, artificial viscosity, and temporary triangular subzoning (TTS) for long-thin zones. Technical report, Los Alamos National Laboratory (LA-4740-MS), 1971. URL http://www.osti.gov/scitech/servlets/purl/4704503.

[11] A. Burbeau-Augoula. A node-centered artificial viscosity method for two-dimensional Lagrangian hydrodynamics calculations on a staggered grid. *Communications in Computational Physics*, 8(4):877, 2010.

[12] D.E. Burton. Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity. In *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*, pages 7–19. Springer, 1991.

[13] D.E. Burton, M.J. Shashkov, S.R. Runnels, R.B. Lowrie, M.A. Kenamond, and P.J. Henning. New generation hydrodynamics methods: from art to science. Technical report, Los Alamos National Laboratory (LA-UR-09-03132), 2009.

[14] D.E. Burton, T.C. Carney, N.R. Morgan, S.R. Runnels, and M.J. Shashkov. Exploration of a cell-centered Lagrangian hydrodynamics method. In *Conference proceedings, Los Alamos National Laboratory, presented at the SIAM Conference on Computational Science and Engineering Reno, Nevada (February 28-March 4, 2011)*, 2011.

[15] D.E. Burton, T.C. Carney, N.R. Morgan, and M.A. Kenamond. Verification studies of CCH and xALE. Technical report, Los Alamos National Laboratory (LANL), 2013.

[16] D.E. Burton, T.C. Carney, N.R. Morgan, S.K. Sambasivan, and M.J. Shashkov. A cell-centered Lagrangian Godunov-like method for solid dynamics. *Computers & Fluids*, 83(0):33 – 47, 2013. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2012.09.008. URL http://www.sciencedirect.com/science/article/pii/S0045793012003647.

[17] E.J. Caramana and M.J. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *Journal of Computational Physics*, 142(2):521–561, 1998. doi: 10.1006/jcph.1998.5952. URL http://dx.doi.org/10.1006/jcph.1998.5952.

[18] E.J. Caramana and P.P. Whalen. Numerical preservation of symmetry properties of continuum problems. *Journal of Computational Physics*, 141(2):174 –

198, 1998. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.1998.5912. URL http://www.sciencedirect.com/science/article/pii/S0021999198959125.

[19] E.J. Caramana, D.E. Burton, M.J. Shashkov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics*, 146(1):227–262, 1998. doi: 10.1006/jcph.1998.6029. URL http://dx.doi.org/10.1006/jcph.1998.6029.

[20] E.J. Caramana, M.J. Shashkov, and P.P. Whalen. Formulations of artificial viscosity for multi-dimensional shock wave computations. *Journal of Computational Physics*, 144(1):70 – 97, 1998. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.1998.5989. URL http://www.sciencedirect.com/science/article/pii/S0021999198959897.

[21] E.J. Caramana, C.L. Rousculp, and D.E. Burton. A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *Journal of Computational Physics*, 157(1):89 – 119, 2000. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.1999.6368. URL http://www.sciencedirect.com/science/article/pii/S0021999199963684.

[22] G. Carré, S. Del Pino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228(14):5160 – 5183, 2009. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2009.04.015. URL http://www.sciencedirect.com/science/article/pii/S002199910900196X.

[23] J. Cheng and C.W. Shu. A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics*, 227(2):1567 – 1596, 2007. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2007.09.017. URL http://www.sciencedirect.com/science/article/pii/S002199910700424X.

[24] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation lawsmulti-dimensional optimal order detection (MOOD). *Journal of Computational Physics*, 230(10):4028 – 4050, 2011. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2011.02.026. URL http://www.sciencedirect.com/science/article/pii/S002199911100115X.

[25] B. Després. Weak consistency of the cell-centered Lagrangian GLACE scheme on general meshes in any dimension. *Computer Methods in Applied Mechanics and Engineering*, 199(4144):2669 – 2679, 2010. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/j.cma.2010.05.010. URL http://www.sciencedirect.com/science/article/pii/S0045782510001593.

[26] B. Després and E. Labourasse. Stabilization of cell-centered compressible Lagrangian methods using subzonal entropy. *Journal of*

Computational Physics, 231(20):6559 – 6595, 2012. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2012.04.018. URL http://www.sciencedirect.com/science/article/pii/S0021999112001994.

[27] B. Després and C. Mazeran. Lagrangian gas dynamics in two dimensions and Lagrangian systems. *Archive for Rational Mechanics and Analysis*, 178 (3):327–372, 2005. ISSN 0003-9527. doi: 10.1007/s00205-005-0375-4. URL http://dx.doi.org/10.1007/s00205-005-0375-4.

[28] V.A. Dobrev, T.V. Kolev, and R.N. Rieben. High-order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM Journal on Scientific Computing*, 34(5):B606–B641, 2012. doi: 10.1137/120864672. URL http://dx.doi.org/10.1137/120864672.

[29] J.K. Dukowicz and B.J.A. Meltz. Vorticity errors in multidimensional Lagrangian codes. *Journal of Computational Physics*, 99(1):115 – 134, 1992. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(92)90280-C. URL http://www.sciencedirect.com/science/article/pii/002199919290280C.

[30] J.K. Dukowicz, M.C. Cline, and F.L. Addessio. A general topology Godunov method. *Journal of Computational Physics*, 82(1):29 – 63, 1989. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(89)90034-X. URL http://www.sciencedirect.com/science/article/pii/002199918990034X.

[31] K. Duraisamy and J.D. Baeder. Implicit scheme for hyperbolic conservation laws using nonoscillatory reconstruction in space and time. *SIAM Journal on Scientific Computing*, 29(6):2607–2620, 2007. doi: 10.1137/070683271. URL http://dx.doi.org/10.1137/070683271.

[32] T.A. Eymann and P.L. Roe. Multidimensional active flux schemes. In *21st aiaa computational fluid dynamics conference*, 2013.

[33] B. Fishbine. Code validation experiments. *Los Alamos Research Quarterly*, pages 6–14, 2002.

[34] D.P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17(5):679–706, 1981. ISSN 1097-0207. doi: 10.1002/nme.1620170504. URL http://dx.doi.org/10.1002/nme.1620170504.

[35] S.K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3): 271–306, 1959.

[36] S.K. Godunov. Reminiscences about difference schemes. *Journal of Computational Physics*, 153(1):6–25, 1999. doi: 10.1006/jcph.1999.6271. URL http://dx.doi.org/10.1006/jcph.1999.6271.

[37] D. Gottlieb and E. Turkel. Phase error and stability of second order methods for hyperbolic problems. II. *Journal of Computational Physics*, 15(2):251 – 265, 1974. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(74)90087-4. URL http://www.sciencedirect.com/science/article/pii/0021999174900874.

[38] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49(3):357 – 393, 1983. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(83)90136-5. URL http://www.sciencedirect.com/science/article/pii/0021999183901365.

[39] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, {III}. *Journal of Computational Physics*, 71(2):231 – 303, 1987. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(87)90031-3. URL http://www.sciencedirect.com/science/article/pii/0021999187900313.

[40] R.E. Kidder. The theory of homogeneous isentropic compression and its application to laser fusion. In Helmut J. Schwarz and Heinrich Hora, editors, *Laser Interaction and Related Plasma Phenomena*, pages 449–464. Springer US, 1974. URL http://dx.doi.org/10.1007/978-1-4684-8416-8_3.

[41] G. Kluth and B. Després. Discretization of hyperelasticity on unstructured mesh with a cell-centered Lagrangian scheme. *Journal of Computational Physics*, 229(24):9092 – 9118, 2010. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2010.08.024. URL http://www.sciencedirect.com/science/article/pii/S0021999110004717.

[42] P. Knupp, L.G. Margolin, and M. Shashkov. Reference jacobian optimization-based rezone strategies for arbitrary lagrangian eulerian methods. *Journal of Computational Physics*, 176(1):93 – 128, 2002. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.2001.6969. URL http://www.sciencedirect.com/science/article/pii/S0021999101969694.

[43] A. Kumar. Isotropic finite-differences. *Journal of Computational Physics*, 201(1):109 – 118, 2004. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2004.05.005. URL http://www.sciencedirect.com/science/article/pii/S0021999104002037.

[44] VF Kuropatenko. On difference methods for the equations of hydrodynamics. *Difference methods for solutions of problems of mathematical physics, I. American Mathematical Society, Providence, RI*, 206, 1967.

[45] D. Kuzmin. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *Journal of Computational and Applied Mathematics*, 233(12):3077 – 3085, 2010. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2009.05.028. URL

http://www.sciencedirect.com/science/article/pii/S0377042709003318. Finite Element Methods in Engineering and Science (FEMTEC 2009).

[46] R. Landshoff. A numerical method for treating fluid flow in the presence of shocks. Technical report, Los Alamos National Laboratory, 1955.

[47] P. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13(2):217–237, may 1960. doi: 10.1002/cpa.3160130205. URL http://dx.doi.org/10.1002/cpa.3160130205.

[48] P. D. Lax and B. Wendroff. Difference schemes for hyperbolic equations with high order of accuracy. *Communications on Pure and Applied Mathematics*, 17(3):381–398, 1964. ISSN 1097-0312. doi: 10.1002/cpa.3160170311. URL http://dx.doi.org/10.1002/cpa.3160170311.

[49] Z. Li, X. Yu, and Z. Jia. The cell-centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two-dimensions. *Computers & Fluids*, 96(0):152 – 164, 2014. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2014.03.018. URL http://www.sciencedirect.com/science/article/pii/S0045793014001170.

[50] H.W Liepmann and A. Roshko. *Elements of gasdynamics*. Dover Publications, Inc., Mineola, New York, 1957.

[51] W. Liu, J. Cheng, and C.W. Shu. High order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations. *Journal of Computational Physics*, 228(23):8872 – 8891, 2009. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2009.09.001. URL http://www.sciencedirect.com/science/article/pii/S0021999109004860.

[52] X.D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.1994.1187. URL http://www.sciencedirect.com/science/article/pii/S0021999184711879.

[53] Y. Liu, W. Shen, B. Tian, and D. Mao. A two dimensional nodal Riemann solver based on one dimensional Riemann solver for a cell-centered Lagrangian scheme. *Journal of Computational Physics*, 284(0):566 – 594, 2015. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2014.12.031. URL http://www.sciencedirect.com/science/article/pii/S0021999114008481.

[54] R. Loubère, P.H. Maire, and P. Váchal. A second-order compatible staggered Lagrangian hydrodynamics scheme using a cell-centered multidimensional approximate Riemann solver. *Procedia Computer Science*, 1(1):1931 – 1939, 2010. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2010.04.216. URL http://www.sciencedirect.com/science/article/pii/S1877050910002176. {ICCS} 2010.

[55] M. Lukáčová-Medvidová, K. Morton, and G. Warnecke. Evolution Galerkin methods for hyperbolic systems in two space dimensions. *Mathematics of Computation of the American Mathematical Society*, 69(232):1355–1384, 2000.

[56] G. Luttwak and J. Falcovitz. Slope limiting for vectors: A novel vector limiting algorithm. *International Journal for Numerical Methods in Fluids*, 65(11-12):1365–1375, 2011. ISSN 1097-0363. doi: 10.1002/fld.2367. URL http://dx.doi.org/10.1002/fld.2367.

[57] G. Maenchen and S. Sack. The tensor code. Technical report, California University Lawrence Radiation Laboratory, 1963. URL http://www.osti.gov/scitech/servlets/purl/4694346.

[58] P.H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228(7):2391 – 2425, 2009. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2008.12.007. URL http://www.sciencedirect.com/science/article/pii/S0021999108006359.

[59] P.H. Maire and J. Breil. A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *International Journal for Numerical Methods in Fluids*, 56(8):1417–1423, 2008. ISSN 1097-0363. doi: 10.1002/fld.1564. URL http://dx.doi.org/10.1002/fld.1564.

[60] P.H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29(4):1781–1824, 2007. doi: 10.1137/050633019. URL http://dx.doi.org/10.1137/050633019.

[61] P.H. Maire, J. Breil, and S. Galera. A cell-centred arbitrary Lagrangian–Eulerian (ALE) method. *International Journal for Numerical Methods in Fluids*, 56(8):1161–1166, 2008. ISSN 1097-0363. doi: 10.1002/fld.1557. URL http://dx.doi.org/10.1002/fld.1557.

[62] P.H. Maire, R. Abgrall, J. Breil, R. Loubère, and B. Rebourcet. A nominally second-order cell-centered Lagrangian scheme for simulating elasticplastic flows on two-dimensional unstructured grids. *Journal of Computational Physics*, 235(0):626 – 665, 2013. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2012.10.017. URL http://www.sciencedirect.com/science/article/pii/S0021999112006183.

[63] M.Dumbser and W. Boscheri. High-order unstructured Lagrangian one-step WENO finite volume schemes for non-conservative hyperbolic systems: Applications to compressible multi-phase flows. *Computers & Fluids*, 86(0):405 – 432, 2013. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2013.07.024. URL http://www.sciencedirect.com/science/article/pii/S0045793013003034.

[64] N.R. Morgan, K.N. Lipnikov, D.E. Burton, and M.A. Kenamond. A Lagrangian staggered grid Godunov-like approach for hydrodynamics. *Journal of Computational Physics*, 259(0):568 – 597, 2014. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2013.12.013. URL http://www.sciencedirect.com/science/article/pii/S0021999113008127.

[65] N.R. Morgan, J.I. Waltz, D.E. Burton, M.R. Charest, T.R. Canfield, and J.G. Wohlbier. A Godunov-like point-centered essentially Lagrangian hydrodynamic approach. *Journal of Computational Physics*, 281(0):614 – 652, 2015. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2014.10.048. URL http://www.sciencedirect.com/science/article/pii/S002199911400730X.

[66] K.W. Morton and P.L. Roe. Vorticity-preserving Lax–Wendroff-type schemes for the system wave equation. *SIAM Journal on Scientific Computing*, 23(1):170–192, 2001. doi: 10.1137/S106482759935914X. URL http://dx.doi.org/10.1137/S106482759935914X.

[67] W.F Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics*, 72(1):78 – 120, 1987. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(87)90074-X. URL http://www.sciencedirect.com/science/article/pii/002199918790074X.

[68] S. Osher. Riemann solvers, the entropy condition, and difference. *SIAM Journal on Numerical Analysis*, 21(2):217–235, 1984. doi: 10.1137/0721016. URL http://dx.doi.org/10.1137/0721016.

[69] J.S. Park, S.H. Yoon, and C. Kim. Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids. *Journal of Computational Physics*, 229(3):788 – 812, 2010. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2009.10.011. URL http://www.sciencedirect.com/science/article/pii/S0021999109005567.

[70] S. Ramsey. *Private communication.* XCP-8, Los Alamos National Laboratory, 2014.

[71] W.J. Rider. Revisiting wall heating. *Journal of Computational Physics*, 162(2):395 – 410, 2000. ISSN 0021-9991. doi: http://dx.doi.org/10.1006/jcph.2000.6544. URL http://www.sciencedirect.com/science/article/pii/S0021999100965446.

[72] P.L Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357 – 372, 1981. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(81)90128-5. URL http://www.sciencedirect.com/science/article/pii/0021999181901285.

[73] P.L. Roe. A new structure for Lagrangian hydrocodes. Presented at the International Conference on Numerical Methods for Multi-Material Fluid Flows, 2011.

[74] P.L. Roe. *Considerations on moving meshes. (private communication)*. Department of Aerospace Engineering, University of Michigan, 2012.

[75] L.I. Sedov. *Similarity and dimensional methods in mechanics (similarity and dimensional methods in mechanics*. Academic Press, New York, 1959.

[76] G. Strang. Accurate partial difference methods. *Numerische Mathematik*, 6(1):37–46, 1964. ISSN 0029-599X. doi: 10.1007/BF01386051. URL http://dx.doi.org/10.1007/BF01386051.

[77] G. Strang. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/Cole, 2006.

[78] K.L. Su. *The small scales mixing: Governing equations*. 2005. URL http://www.imaging.me.jhu.edu/PDF/635lec11-smallscales2.pdf.

[79] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 21(5):995–1011, 1984. doi: 10.1137/0721062. URL http://dx.doi.org/10.1137/0721062.

[80] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2009.

[81] C. Truesdell. *Forward to The Collected Works of Leonhardt Euler, Series II, Part 2*. Fussli, Switzerland, 1960.

[82] E. Turkel. Phase error and stability of second order methods for hyperbolic problems. I. *Journal of Computational Physics*, 15(2):226 – 250, 1974. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(74)90086-2. URL http://www.sciencedirect.com/science/article/pii/0021999174900862.

[83] B. van Leer. Towards the ultimate conservative difference scheme I. The quest of monotonicity. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pages 163–168. Springer, 1973.

[84] B. van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics*, 14(4):361 – 370, 1974. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(74)90019-9. URL http://www.sciencedirect.com/science/article/pii/0021999174900199.

[85] B. van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23(3):276 – 299, 1977. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(77)90095-X. URL http://www.sciencedirect.com/science/article/pii/002199917790095X.

[86] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101 – 136, 1979. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(79)90145-1. URL http://www.sciencedirect.com/science/article/pii/0021999179901451.

[87] R. Vichnevetsky and J.B. Bowles. *Fourier analysis of numerical approximations of hyperbolic equations*, volume 5. SIAM, 1982.

[88] J. VonNeumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):232–237, 1950. doi: http://dx.doi.org/10.1063/1.1699639. URL http://scitation.aip.org/content/aip/journal/jap/21/3/10.1063/1.1699639.

[89] J. Waltz, N.R. Morgan, T.R. Canfield, M.R.J. Charest, L.D. Risinger, and J.G. Wohlbier. A three-dimensional finite element arbitrary Lagrangian–Eulerian method for shock hydrodynamics on unstructured grids. *Computers & Fluids*, 92(0):172 – 187, 2014. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2013.12.021. URL http://www.sciencedirect.com/science/article/pii/S0045793013005136.

[90] M.L. Wilkins. Calculation of elastic-plastic flow. Technical report, DTIC Document, 1963. URL http://hdl.handle.net/2027/mdp.39015077597253.

[91] M.L. Wilkins. Use of artificial viscosity in multidimensional fluid dynamic calculations. *Journal of Computational Physics*, 36(3):281 – 303, 1980. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/0021-9991(80)90161-8. URL http://www.sciencedirect.com/science/article/pii/0021999180901618.

[92] J.C. Wilson. Stability of Richtmyer type difference schemes in any finite number of space variables and their comparison with multistep strang schemes. *IMA Journal of Applied Mathematics*, 10(2):238–257, 1972. doi: 10.1093/imamat/10.2.238. URL http://imamat.oxfordjournals.org/content/10/2/238.abstract.

[93] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335 – 362, 1979. ISSN 0021-9991.

[94] Y.B. ZelDovich and Y.P. Raizer. *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena*. Dover Publications, Inc., Mineola, New York, 2002.