A NONLINEAR INVESTIGATION OF CORRUGATION INSTABILITIES IN MAGNETIC ACCRETION SHOCKS

by

SCOTT ERNST

A DISSERTATION

Presented to the Department of Physics
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 2011

DISSERTATION APPROVAL PAGE

Student: Scott Ernst

Title: A Nonlinear Investigation of Corrugation Instabilities in Magnetic Accretion Shocks

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Physics by:

Dr. James Schombert	Chair
Dr. James Imamura	Advisor
Dr. Alan Rempel	Member
Dr. John Toner	Member
Dr. Kent Stevens	Outside Member

and

Richard Linton	Vice President for Research and Graduate Studies/Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded March 2011

© 2011 Scott Ernst

DISSERTATION ABSTRACT

Scott Ernst

Doctor of Philosophy

Department of Physics

March 2011

Title: A Nonlinear Investigation of Corrugation Instabilities in Magnetic Accretion Shocks

Approved: ____
Dr. Jim Imamura

Accretion shock waves are present in many important astrophysical systems and have been a focus of research for decades. These investigations provide a large body of understanding as to the nature, characteristics, and evolutionary behaviors of accretion shock waves over a wide range of conditions. However, largely absent are investigations into the properties of accretion shock waves in the presence of strong magnetic fields. In such cases these strong magnetic fields can significantly alter the stability behaviors and evolution of the accretion shock wave through the production and propagation of magnetic waves as well as magnetically constrained advection. With strong magnetic fields likely found in a number of accretion shock systems, such as compact binary and protostellar systems, a better understanding of the behaviors of magnetic accretion shock waves is needed.

A new magnetohydrodynamics simulation tool, IMOGEN, was developed to carry out an investigation of instabilities in strong, slow magnetic accretion shocks by modelling their long-term, nonlinear evolution. IMOGEN implements a

relaxed, second-order, total variation diminishing, monotonic upwind scheme for conservation laws and incorporates a staggered-grid constrained transport scheme for magnetic advection.

Through the simulated evolution of magnetic accretion shocks over a wide range of initial conditions, it has been shown, for sufficiently high magnetic field strengths, that magnetic accretion shocks are generally susceptible to corrugation instabilities, which arise in the presence of perturbations of the initial shock front. As these corrugation instabilities grow, they manifest as magnetic wave propagation in the upstream region of the accretion column, which propagate away from the accretion shock front, and as density columns, or fingers, that grow into the higher density downstream flow, defined and constrained by current loops created during the early evolution of the instability.

CURRICULUM VITAE

NAME OF AUTHOR: Scott Ernst

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene
Gustavus Adolphus College, Saint Peter, Minnesota

DEGREES AWARDED:

Doctor of Philosophy, Physics, 2011, University of Oregon
Master of Science, Physics, 2005, University of Oregon
Bachelor of Arts, Physics, 2004, Gustavus Adolphus College

AREAS OF SPECIAL INTEREST:

Computational Astrophysics and Physical Simulation
Scientific Visualization

PROFESSIONAL EXPERIENCE:

Teaching assistant, Department of Physics, University of Oregon, Eugene, 2004-2010

GK-12 Fellow, University of Oregon, Eugene 2006-2007

Research Assistant, Department of Physics, Gustavus Adolphus College, Saint Peter, Minnesota, 2002-2004

Teaching Assistant, Department of Physics, Gustavus Adolphus College, Saint Peter, Minnesota, 2001-2004

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1

II. BACKGROUND AND THEORY 6
  2.1. Conservation Law Approach 6
  2.2. Ideal Magnetohydrodynamics 8
  2.3. Shocks 12
    2.3.1. Magnetic Shock Regimes 15
    2.3.2. Magnetic Shock Instabilities 16

III. NUMERICAL MAGNETOHYDRODYNAMICS 21
  3.1. Discrete Magnetohydrodynamics 22
    3.1.1. Lax-Wendroff Scheme 24
    3.1.2. Courant-Freidrichs-Lewy Condition 25
    3.1.3. Higher-Order Schemes 26
  3.2. Shock Capturing 27
    3.2.1. Upwind Schemes 28
    3.2.2. Second-Order Total Variation Diminishing Schemes 31
    3.2.3. Flux Limiters 33
  3.3. The Relaxed Second-Order TVD MUSCL Scheme 37
    3.3.1. The Relaxation Technique 37
    3.3.2. The Complete Scheme 41
    3.3.3. Extending to Multiple Dimensions 42
    3.3.4. Temporal Integration 43
  3.4. Magnetic Field Advection 44

IV. IMOGEN: ELEMENTS OF IMPLEMENTATION 49
  4.1. Cross-Component Magnetic Advection 49
  4.2. Non-Conservative Terms 51
    4.2.1. Radiation 54
    4.2.2. Artificial Viscosity 55
  4.3. External Boundary Conditions 58
    4.3.1. Common Boundary Types 64
    4.3.2. The Fade Shift Generator 64
  4.4. Active Grid Alignment 70

V. TESTING AND VERIFICATION 72
  5.1. Sod Shock Tube 73
  5.2. Brio-Wu Shock Tube 89
  5.3. Orszag-Tang Vortex 95

5.4. Other Tests 105
   5.4.1. Sedov-Taylor Blast Wave 106
   5.4.2. Kelvin-Helmholtz Instability 109

VI. MAGNETIC ACCRETION SHOCK WAVE RESULTS 111

6.1. Preliminary Results 117
6.2. Fade Shift Generator Verification 122
6.3. Simulation Comparisons 127
   6.3.1. Sonic Mach Variations 127
   6.3.2. Incident Angle Variations 131
   6.3.3. Alfvén Mach Variations 137
   6.3.4. Specific Heat Ratio Coefficient Variations 144
   6.3.5. Resolution and Diffusion Variations 146

VII. SUMMARY AND FUTURE DIRECTIONS 149

APPENDICES

A. IMOGEN SIMULATION SOFTWARE DESCRIPTION 151
   A.1. Interpreted Scientific Computing 152
      A.1.1. Productivity 152
      A.1.2. Accessibility 153
      A.1.3. Portability 153
      A.1.4. Optimization 154
   A.2. Computational Approach 155
      A.2.1. The Interface Layer 157
      A.2.2. The Research & Simulation Layer 158
      A.2.3. The Application & Control Layer 159
      A.2.4. The Performance Layer 160
   A.3. Design & Implementation 160
      A.3.1. Initializers 161
      A.3.2. Simulation Run Scripting 163
      A.3.3. Simulation Management 165

B. DYNAMICALLY-SCALED VISUALIZATION ALGORITHM 167

REFERENCES CITED 169

LIST OF FIGURES

Figure Page

1.1. Hydrodynamic and magnetohydrodynamic wave comparison for point disturbances 3

2.1. Cartesian component representation of the magnetohydrodynamics equations 12

2.2. Magnetic shock regime diagram 16

3.1. Example advection of a square wave 30

3.2. Comparison of the first-order MUSCL and second-order TVD MUSCL schemes 32

3.3. Possible behaviors for the slope ratio parameter 34

3.4. Staggered grid arrangement of the magnetic field 46

4.1. Assignment hierarchy for external boundary conditions 61

4.2. Illustration of the shifting process 63

4.3. Examples of shift generators 65

4.4. Influence profile for the fade shift NRABC boundary region 69

5.1. Diagram of the Sod shock tube test 74

5.2. Simulated discontinuity results of a Sod shock tube test 76

5.3. Profile of the mass density in a Sod shock tube simulation 78

5.4. Fractional error between the simulated and analytical 79

5.5. Waterfall evolution plot of the fractional error 80

5.6. Simulated and analytical velocity profiles 81

5.7. Simulated and analytical total specific energy density profiles 82

5.8. Results of a two-dimensional non-axis aligned Sod shock tube 84

5.9. Profiles of mass density 86

5.10. Fractional error plot between the simulated and analytical 87

5.11. Sod shock tube problem with fade shift edges 88

5.12. Mass density and total specific energy results 91

5.13. Shock plane parallel and perpendicular velocity results 92

5.14. Shock plane parallel and perpendicular magnetic field results ... 93
5.15. Initial conditions for the Orszag-Tang vortex 96
5.16. Sonic Mach values for the Orszag-Tang vortex simulation 97
5.17. Evolution of the mass density for an Orszag-Tang vortex 98
5.18. Evolution of the mass density in an Orszag-Tang vortex 99
5.19. Orszag-Tang vortex simulation at the critical time 100
5.20. Sonic Mach values for the Orszag-Tang vortex 101
5.21. Additional results of the Orszag-Tang vortex simulation 102
5.22. Comparison between Orszag-Tang vortex simulations 103
5.23. Mass density profiles of the Orszag-Tang vortex 105
5.24. Worst case scenario for a Sedov-Taylor blast wave 107
5.25. Mass density versus distance from the center 108
5.26. Kelvin-Helmholtz instability test results 110
6.1. Schematic illustration of the setup for the magnetic accretion shock simulation 111
6.2. Non-uniform cell spacing of the primary shock tube axis 112
6.3. Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial 121
6.4. Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial 123
6.5. Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial 124
6.6. Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial 125
6.7. Direct comparison of the growth rate plots 126
6.8. Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}5 \mid \alpha\text{-}1\}$ trial 128
6.9. Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}2 \mid \alpha\text{-}1\}$ trial 129
6.10. Comparison of corrugation instability growth parameters 131
6.11. Results of the $\{\theta\text{-}0° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial 133
6.12. Results of the $\{\theta\text{-}22.5° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial 134

Figure Page

6.13. Results of the {θ-45° | A-1/2 | S-10 | α-1} trial 135

6.14. Mass density contour plot 136

6.15. Results of the {θ-0° | A-1/8 | S-10 | α-1} trial 140

6.16. Results of the {θ-10° | A-1/8 | S-10 | α-1} trial 141

6.17. Results of the {θ-22.5° | A-1/8 | S-10 | α-1} trial 142

6.18. Results of the {θ-10° | A-10 | S-10 | α-1} trial 143

6.19. Comparison of instability growth evolution 144

6.20. Results of the {θ-10° | A-10 | S-10 | α-18/25} trial 145

6.21. Comparison of corrugation instability growth parameters 146

6.22. Results of the {θ-10° | A-1/2 | S-10 | α-1} trial 147

A.1. Design diagram for the complexity based layering 157

A.2. Example IMOGEN run simulation script 164

B.1. Pseudo-code example of how the nonlinear, dynamic-scale color profile algorithm works 168

CHAPTER I

INTRODUCTION

Accretion shock waves, which occur when matter is pulled by gravitational forces onto the surface of a large gravitational body, such as a star, from some other stellar partner or gas cloud, contribute to the observable and long-term evolutionary properties of many important astrophysical systems. The investigation of accretion shock waves has, therefore, been an active area of research for over fifty years in theoretical and computational astrophysics. That body of research has rendered a fairly robust understanding of the mechanisms that govern accretion shocks found in magnetic plasmas, and explain the observed properties of a wide range of accretion shock wave systems (Blondin, et al., 2003; Hartmann, 2007).

However, insights into variations among observed compact binary and protostellar systems, examples of which include X-Ray binaries, polars, intermediate polars, and AM Her and T Tauri type stars, could be due in part to the presence of magnetic accretion shocks (Burwitz,, et al., 2003; Donatil, et al., 2005; Girart, et al., 2006; King, 1988; Lamb, et al., 1973; Lamb, et al., 1975; Shu, et al., 1997). In many of these cases it is likely that the observed protostar or the primary star in a compact binary system have magnetic fields of sufficient magnitude that they influence, and in some cases dominate, the accretion column in which the shock wave resides by both channeling the flow of the accretion column and introducing new mechanisms for both transient and secular behaviors to occur (Lesson &

Desphande, 1967; Stone & Edelman, 1993).

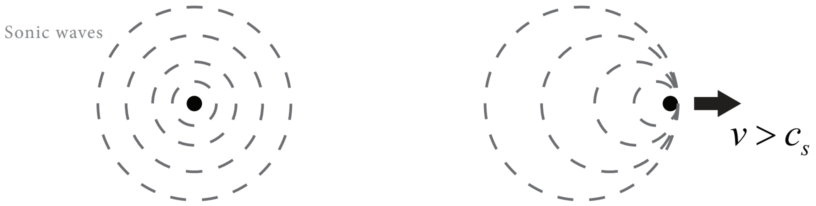
Of the large amount of research conducted on accretion shock waves, comparatively little work has been done to investigate them in the presence of strong magnetic fields even though magnetic fields can significantly alter the properties of the shock front and in turn influence the evolutionary behaviors of accretion systems. In the work that has been carried out, researchers have found that the magnetic field does in fact play an important role in the dynamics and stability of accretion shock systems, including systems with properties that potentially coincide with the previously mentioned accretion systems. In these systems the presence of the magnetic field leads to additional corrugation instabilities that ripple the shock front and can radically alter its long-term behavior (Lesson & Desphande, 1967; Stone & Edelman, 1993).

The presence of a magnetic field increases the number of degrees of freedom available to accretion shock waves and, in turn, provides a wider potential array of properties and behaviors. For example, consider a nonmagnetic shock wave, which has only thermal mechanisms for propagating energy away from the shock front into the upstream region, as the acoustic pressure waves travel with velocities less than the upstream flow speed (Boyd & Sanderson, 2003). Therefore, acoustic waves cannot propagate into the accretion column and neither transport energy away from the front nor influence the upstream accretion flow. Hence radiative transport has been a primary focus of accretion shock wave research for the purely hydrodynamic cases studied.

The addition of a magnetic field allows for three additional wave types for propagating energy away from the shock front. Two of these are the fast and slow magnetoacoustic waves, which are longitudinal waves like the purely acoustic wave but with propagation velocities that exceed that of the purely acoustic case. The other is Aflvén waves, transverse magnetic waves that can also propagate faster than a purely acoustic wave under the applicable conditions (Boyd & Sanderson, 2003). For cases where the shock speed is sub-Alfvénic, meaning at least one of these magnetic wave modes will propagate ahead of the shock front as is typically the case in strongly magnetic accretion systems. Figure 1.1 illustrates these wave modes as a comparison between hydrodynamic and magnetohydrodynamic cases.

Magnetohydrodynamic Versus Hydrodynamic Wave Propagation

Hydrodynamic disturbance:



Magnetohydrodynamic disturbance:

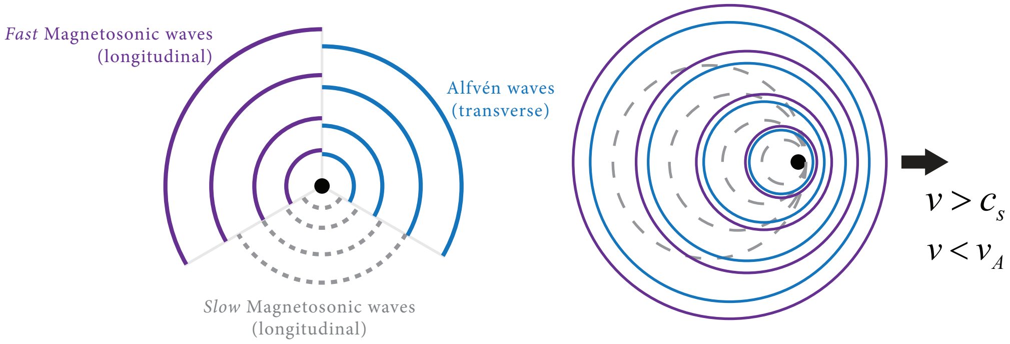

Fast Magnetosonic waves (longitudinal)
Alfvén waves (transverse)
Slow Magnetosonic waves (longitudinal)
$v > c_s$
$v < v_A$

Figure 1.1: Hydrodynamic and magnetohydrodynamic wave comparison for point disturbances. The left disturbances are stationary, showing isotropic propagation of the waves away from the disturbance. Disturbances on the right are moving rightward at supersonic speed, and only magnetic wave modes are able to propagate ahead of the disturbance.

In contrast to the single acoustic wave mode and propagation speed of any hydrodynamic disturbance, magnetohydrodynamic disturbances have additional wave modes with speeds in excess of the acoustic speed. Hence, in an accretion shock wave, the magnetic wave modes are still capable of impacting the upstream flow of the accretion column, and altering the evolution of the shock front in ways that could help explain observable properties of compact binary and protostellar systems (Stone & Edelman, 1993).

A thorough, nonlinear investigation into the stability properties and long-term behaviors of magnetic accretion shocks is needed and is the result of the research presented here. Previous work was conducted by Lessen and Desphande and Edeleman into the linear properties of magnetic accretion shocks under strong assumptions or limited dimensionality, and Stone and Edeleman have presented some nonlinear work that focuses mostly on two-dimensional systems (Lesson & Desphande, 1967; Stone & Edelman, 1993). The present research, by looking into the variabilities, or similarities, between shock behaviors over a wide range of parameters such as incident angle of the accretion column onto the shock front, the magnetic field strength, and propagation and acoustic velocities in the upstream flow, attempts to better understand how magnetic accretion shocks behave compared to their hydrodynamic counterparts and shed some light on the mechanisms responsible for the properties of observed accretion systems. It extends the previous work by conducting an investigation using a fully nonlinear, three-dimensional treatment of magnetic accretion shocks over a broad range of parameter space, and provides a deeper analysis into behavioral variations among corrugation instabilities in different areas of parameter space. In doing so, the results uncover reasons behind key changes in the instability growth for different initial conditions that were hitherto unaddressed. Additionally, this investigation utilizes recent developments in computational astrophysics, employs numerical techniques and algorithms, and takes advantage of the significantly greater availability of computational resources than the previous investigations that were conducted close to two decades ago.

In order to complete this investigation, a new simulation software, IMOGEN, was developed and tested, employing the latest computational astrophysical methods required for the investigation as described in chapters 3-5. Once complete, IMOGEN was put to use as the simulation tool for the magnetic accretion shock problem, the results of which are presented in chapter 6 and summarized in chapter 7. The goal of this research is the expanded understanding of the behaviors of shocks in the strong, slow magnetic shock regime to better define what drives these instabilities, resolve unanswered questions from previous work regarding their evolutionary variations and stability, and provide the foundational basis of understanding necessary to extend the investigation to include additional physics, specifically radiative cooling in the shock system.

CHAPTER II

BACKGROUND AND THEORY

Like many complex problems in physics, the most robust approach to solving astrophysical problems would be to consider the dynamics of their constituent elements, protons and electrons in hot regimes and atoms and molecules in cool regimes, within the system of interest and evolve each element in time from initial to final states. However, even with abundant computational resources, this type of approach is intractable. This forces astrophysicists to rely on continuum fluid mechanics, a reasonable, but nonetheless approximate, solution method based on fundamental conservation laws that are macroscopic and tractable.

2.1. Conservation Law Approach

For any quantity, Q, that is conserved within the temporal and spatial bounds of a system of interest, there exists a conservation equation that applies everywhere locally to the volume density, q, for that quantity and takes the general form,

$$\partial_t q + \nabla \cdot (q\boldsymbol{v}) = 0 \,, \tag{2.1}$$

where q is the density of the conserved quantity at some point within the system and $\boldsymbol{v}$ is the three-dimensional velocity associated with that density. For a conserved vector quantity, $\boldsymbol{Q}$, with its associated density, $\boldsymbol{q}$, a similar equation exists in the form,

$$\partial_t \boldsymbol{q} + \nabla \cdot (\boldsymbol{q}\boldsymbol{v}) = 0 \,. \tag{2.2}$$

Outer-product notation is used here in the term, ***qv***, which evaluates as a 3x3 square matrix, which when operated on by the divergence operator reduces to a vector with a component for each component of the density. Equation (2.2) can therefore be expanded component-wise as three equations of the form (2.1), one for each component.

Dimensional analysis of (2.1) and (2.2) reveals that the term inside the divergence operator represents an *amount per area per time*, also known as a flux. This gives rise to equations of the form (2.1) and (2.2) being classified as advection, or flow, equations where the time dependence in quantity, *Q* or ***Q***, is due to spatial transport as a result of variations in a corresponding flux. Furthermore, (2.1) and (2.2) are conservative advection equations because variations with respect to time are due only to the variations in the flux. Put another way the quantity at a point can change over time only because some or all of the quantity was transported to some other point in the system, no amount of the quantity can be created or destroyed during this process. This conservative nature holds regardless of the complexity of the flux term, as long as it reduces to the basic form of (2.1) or (2.2).

If, however, terms are included that cannot be written in terms of the divergence of a flux, the advection equation is no longer conservative, allowing for the change in the total quantity within the system over time. Such terms are called source terms to distinguish them from the conservative flux terms.

A final distinction of note regarding advection equations is the difference between a conservative advection equation and one written in conservative form. An advection equation in conservative form is any advection equation, either conservative or non-conservative, written with all of the conservative flux terms contained within a single divergence operator and all non-conservative source terms expressed on the right-hand side of the equation. A general advection equation in conservative form looks like

$$\partial_t q(\boldsymbol{r},t)+\nabla\cdot(\boldsymbol{F}(\boldsymbol{r},t))=S(\boldsymbol{r},t), \tag{2.3}$$

where $\boldsymbol{F}$ is the total flux vector, which could be made up of many conservative flux terms, and S is the collection of source terms. All advection equations contained within this work will be in conservative form, although not all are conservative advection equations.

2.2. Ideal Magnetohydrodynamics

Astrophysical systems are described using a grouping of interdependent advection equations, collectively called the magnetohydrodynamic equations. These equations, along with a number of auxiliary equations fully describe the evolution of an astrophysical system. The most commonly used form of the magnetohydrodynamic equations are,

$$\partial_t\rho+\nabla\cdot(\rho\boldsymbol{v})=0, \tag{2.4}$$

$$\partial_t(\rho\boldsymbol{v})+\nabla\cdot(\rho\boldsymbol{v}\boldsymbol{v}-\boldsymbol{B}\boldsymbol{B}+\mathbf{I}P^*)=0, \tag{2.5}$$

$$\partial_t E+\nabla\cdot((E+P^*)\boldsymbol{v}-\boldsymbol{B}(\boldsymbol{B}\cdot\boldsymbol{v}))=0, \tag{2.6}$$

$$\partial_t \boldsymbol{B} + \nabla \cdot \left(\boldsymbol{vB} - \boldsymbol{Bv} \right) = 0 \quad . \tag{2.7}$$

Here ρ is the mass density, $\boldsymbol{v}$ is the velocity vector, $\boldsymbol{B}$ is the magnetic field vector, P^* is the total pressure, and E the total energy. A number of assumptions were made to arrive at these equations, first of which is that the electrical conductivity of the system is infinite, as a proper description of resistance would require the addition of non-conservative diffusion and dispersal terms. The choice to assign an infinite value to the conductivity is a reasonable one in the investigation of hot plasmas, and the magnetohydrodynamic equations under this condition are called the ideal magnetohydrodynamics equations. In cooler plasmas, where resistivity is of greater significance, the more general resistive magnetohydrodynamic equations are required. However, due to the added complexity of the additional resistive source terms the resistive magnetohydrodynamic equations are used only when absolutely necessary (Boyd & Sanderson, 2003).

A natural unit system is used for these equations where the magnetic permeability, $\mu = 1$, and the magnetic field is re-normalized to include the factor of 4π that would normally appear in the denominator of the magnetic terms. The re-normalization is such that the original magnetic field $\bar{\boldsymbol{B}}$ is redefined as

$$\boldsymbol{B} = \frac{\bar{\boldsymbol{B}}}{\sqrt{4\pi}} \quad . \tag{2.8}$$

This leads to a system of equations without any extraneous irrational coefficients on the magnetic terms (Gardiner & Stone; 2005).

Equation (2.4) describes the conservative evolution of mass density through the system and is called the continuity equation. Equation (2.5) is the conservative evolution of the momentum, which is just an advection equation formulation of Newton's second law. In (2.5) the total pressure, P*, is introduced in the flux term as a scale coefficient to the identity matrix, which implies an isotropic pressure at each point. The total pressure for the system is defined as the sum of the hydrodynamic gas pressure, *P*, plus the magnetic pressure as,

$$P^* \equiv P + \tfrac{1}{2}\boldsymbol{B} \cdot \boldsymbol{B} \quad . \tag{2.9}$$

While the gas pressure is a primitive variable within a hydrodynamic system, no conservation law exists for it. Instead pressure is introduced through auxiliary equations of state, which are responsible for relating the pressure to the remaining primitive quantities. Formulating the magnetohydrodynamic equations in this way allows for the substitution of different equations of state as they pertain to specific problems. The most common of which is the ideal gas equation of state, which relates the gas pressure to the internal energy as,

$$P = (\gamma - 1)\varepsilon \quad , \tag{2.10}$$

where ε is the internal energy density and γ is the ratio of specific heats.

Equation (2.6) is the conservative form of the energy conservation equation, which is written in terms of total energy,

$$E \equiv \varepsilon + \tfrac{1}{2}\rho \boldsymbol{v} \cdot \boldsymbol{v} + \tfrac{1}{2}\boldsymbol{B} \cdot \boldsymbol{B} . \qquad (2.11)$$

While it is true that conservation of the internal energy would be the more fundamental choice, total energy turns out to be the more robust form to use for numerical purposes due to the additional interdependencies of the magnetic field and total pressure. This tends to produce more stable solutions and so has been widely adopted among the computational astrophysics community.

Finally (2.7) describes the conservative advection of the magnetic field, and is indeed Maxwell's induction equation reworked into conservative form by application of vector identities. Left out of the induction equation and the remainder of the magnetohydrodynamic equations is the divergence-free constraint on the magnetic field,

$$\nabla \cdot \boldsymbol{B} = 0 , \qquad (2.11)$$

which is implicitly conserved in analytic solutions to this system of equations but is not in a discrete, numerical one because the discretization of the magnetic field leads to quantization errors that tend to compound. Given the lack of this explicit constraint, computational magnetohydrodynamics has to treat the induction equation carefully and differently from the other conservation equations as will be discussed in more detail in following chapters (Balsara, 2004; Dai & Woodward, 1998).

From here on a Cartesian coordinate system will be used and all references to dimensionality will assume such a coordinate system. No care or attention will be given to presenting material in a general, coordinate-independent fashion. With a single coordinate system in mind it is useful to present the magnetohydrodynamic equations in a Cartesian component-wise fashion for later reference as shown in Figure 2.1.

Cartesian Component Form of the
Ideal Magnetohydrodynamic Equations

$$\partial_t \overbrace{\begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \\ B_x \\ B_y \\ B_z \\ \text{where:} \end{bmatrix}}^{t} + \sum_i \partial_i \overbrace{\begin{bmatrix} \rho v_i \\ \rho v_x v_i - B_x B_i + P^* I_{xi} \\ \rho v_y v_i - B_y B_i + P^* I_{yi} \\ \rho v_z v_i - B_z B_i + P^* I_{zi} \\ \left(E + P^*\right) v_i - \left(v_x B_x + v_y B_y + v_z B_z\right) B_i \\ v_i B_x - B_i v_x \\ v_i B_y - B_i v_y \\ v_i B_z - B_i v_z \\ B_i \end{bmatrix}}^{i} = 0$$

Figure 2.1: Cartesian component representation of the magnetohydrodynamics equations (2.4) - (2.7) along with the divergence-free constraint that must be imposed on magnetic advection equation.

2.3. Shocks

A shock, in the astrophysical sense, is defined as a discontinuous disturbance between two distinct regions of differing primitive states, ρ, $\rho\boldsymbol{v}$, E, and $\boldsymbol{B}$. On each side of the shock front, the surface of discontinuity that separates and defines the boundary between the two distinct regions, there exists a solution to the magnetohydrodynamic equations, (2.4) - (2.7), that is different from the corresponding solution in its adjoining region. For a shock to exist there must be an abrupt velocity transition between the regions across the shock front so that in one region the velocity is locally supersonic, with a velocity in excess of the local sound speed in that region, and in the other it is subsonic, with a velocity smaller than the local sound speed for that region. Hence as the shock wave propagates with respect to a laboratory reference frame, it moves into the supersonic, or upstream, region expanding the subsonic, or downstream, region.

The interdependencies of the primitive variables between the upstream and downstream regions of a shock are determined by setting the magnetohydrodynamic equations, (2.4) - (2.7) that define each of the regions equal to their corresponding equation in the adjoining region and then solving that two-region boundary system. The resulting group of relationships between the upstream and downstream variables is called the Rankine–Hugoniot jump conditions, or more succinctly the jump equations, that define how the variables abruptly change across the shock front (Boyd & Sanderson, 2003).

A common method for finding the jump equations is to consider the problem as a one-dimensional steady-state shock and solve the equations in the frame of the shock, in which case the front is stationary. It's important to realize that describing the system as one-dimensional only means that the spatial dependence along the directions perpendicular to the front are isotropic; it does not confine the velocity and other vector quantities to a single, shock-aligned direction.

For the scope of this work, the jump conditions will be considered in terms of a single angle that describes the relative angle between the upstream velocities and magnetic fields and the local, planar surface shock front that separates the two regions. The assumption is made in all cases that the incident, upstream magnetic field and velocity are parallel because the geometry of the magnetic shock wave accretion problem for strong magnetic fields produces magnetically constrained flows, which tend to align the velocity and magnetic field. Without significant resistivity or other, related diffusion terms there is no mechanism to counteract this tendency toward a parallelization of the magnetic field and velocity. For these cases the resulting jump conditions that describe the systems are,

$$\left[\rho v_{\parallel}\right]_1^2 = 0 , \tag{2.12}$$

$$\left[\rho v_{\parallel}^2 + P + \frac{B_{\perp}^2}{2}\right]_1^2 = 0 , \tag{2.13}$$

$$\left[\rho v_{\parallel} v_{\perp} - B_{\parallel} B_{\perp}\right]_1^2 = 0 , \tag{2.14}$$

$$\left[\left(\frac{5\alpha}{2} P + \frac{1}{2}\rho\left(v_{\parallel}^2 + v_{\perp}^2\right)\right) v_{\parallel} - \left(B_{\parallel} v_{\parallel} + B_{\perp} v_{\perp}\right) B_{\parallel}\right]_1^2 = 0 , \tag{2.15}$$

$$\left[v_{\parallel} B_{\perp} - B_{\parallel} v_{\perp}\right]_1^2 = 0 , \tag{2.16}$$

These jump conditions have been cast in a form that is predisposed to work well with numerical solvers instead of the most compact representation. The preference of form results from the need in the main body of work to generate a large set of initial conditions using the jump equations for a wide variety of cases (Boyd & Sanderson, 2003).

2.3.1. Magnetic Shock Regimes

Magnetic shocks can be broken into four regimes, combinations of slow or fast and weak or strong. The terms weak and strong are representative of the change in velocity between the upstream and downstream regions, where a weak shock is one where the change in velocity is relatively small between the upstream and downstream regions as opposed to a strong shock where the change in velocity between regions is large. Understanding that the barrier between weak and strong shocks is soft, and is often case specific, a fair assumption of the regimes can be defined in terms of the sonic Mach number,

$$M_s \equiv \frac{|\boldsymbol{v}|}{c_s}, \tag{2.17}$$

which is the ratio between the advection speed, $|\boldsymbol{v}|$, and the adiabatic speed of sound, c_s, in the surrounding upstream medium. Generally, shocks with upstream Mach numbers less than 3 are considered weak and values above 3 considered strong.

Similarly the terms slow and fast refer to the magnetic propagation speed,

$$v_A = \frac{|\boldsymbol{B}|}{\sqrt{\rho}}, \tag{2.18}$$

called the Alfvén speed, in the upstream medium. A slow magnetic shock, is one where the Alfvén Mach,

$$M_A \equiv \frac{|\boldsymbol{v}|}{v_A}, \tag{2.19}$$

is less than one, which signifies that magnetic waves are able to propagate faster than the advection speed. Fast magnetic shocks, where the advection speed exceeds the speed at which magnetic waves propagate within the medium, are therefore any shocks with an Alfvén Mach, M_A, greater than or equal to one.

Figure 2.2 illustrates the four possible shock regimes, and highlights the region of interest, the slow, strong regime, which is the regime investigated in this work.

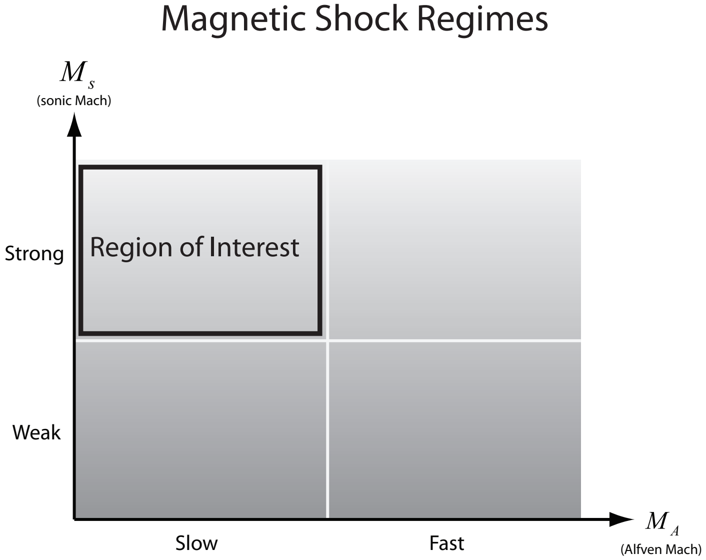


Figure 2.2: Magnetic shock regime diagram illustrating the four possible regime classifications for magnetic shocks. The strong slow regime has been highlighted as the region of interest for investigation.

2.3.2. Magnetic Shock Instabilities

A large amount of work has gone into the study of shock behaviors and instabilities for shocks classified by (2.12) - (2.16) in purely hydrodynamic cases, i.e. where the magnetic field is negligible everywhere, for both the weak and strong shock regimes. These investigations include, among other things, the effects of radiation and other thermal properties. Similarly, work has been done to understand the behaviors of fast magnetic shocks, both strong and weak, where the small magnetic field has at most an ancillary effect on the overall shock system. Much less work has gone into studying the slow magnetic shock regime, particularly in cases of strong shocks, due in part to the fact that the other regimes are much more common and because of the difficulties involved in investigating shocks with strong magnetic fields.

What makes the magnetic accretion shock system potentially interesting is that the presence of magnetic fields introduce three additional wave modes to the system. The modes are the fast and slow magnetoacoustic waves, compression waves of pressure coupled to the magnetic field, and the Alfvén wave mode, a transverse magnetic field wave. Unlike the acoustic compression waves found in a purely hydrodynamic system, a number of the magnetic waves can propagate faster than a shock wave under the right conditions, namely a sub-Alfvénic system where the Alfvén Mach, (2.19) is small. As previously mentioned, the result is that in a slow shock, these new magnetic wave modes are capable of propagating ahead of the shock front and, consequently, influencing the evolution of the front as it evolves in space and time. For more discussion and derivation of these wave modes see associated references (Boyd & Sanderson, 2003; Edelman, 1990; Edelman, 1993; Stone & Edelman 1995).

An investigation of strong, slow magnetic shocks can be carried out in either of two ways. The first, a linear investigation, involves linearizing the magnetohydrodynamic equations, (2.4) - (2.7), and finding solutions either analytically or through numerical means depending on the complexity of and simplifications applied to a particular shock system. The final results of this type of linear analysis are relationships that specify the stability of the system to perturbation as well as the relationships that give insight into how any instabilities form and behave during their early evolution. As the solution is linear the results hold only for the early development of the instability. That is until the nonlinear terms of the highly nonlinear magnetohydrodynamic equations become consequential in the evolution of the shock system. While these results are limited to the early development of instabilities, understanding if and how the instability forms, as well as its early evolution, is crucial to understanding the behaviors of the shocks more generally.

The second approach is to conduct a nonlinear investigation, which can only be carried out using numerical methods and with significant computational resources as the evolution of the magnetohydrodynamic equations are, for all but a few cases, extremely difficult to solve in a fully nonlinear context. The results of these nonlinear investigations are complete solutions for the evolution of the shock system, which can be analyzed to determine the stability and growth behaviors of the shock in both early and late developmental stages. These results can also be modelled to determine general behaviors as well as provide a way of comparison to the linear results to better understand how the formation of an instability influences its long-term development and eventual end.

Previous investigations of slow, strong magnetic shocks have focused primarily on linear approaches and have been carried out with significant simplifications to make finding solutions tractable. In these analyses, carried out in various forms by Lessen, Desphande, and Edeleman, the strong, slow shock regime was found to be unstable. Among these results differing regions were identified, typically for shocks with small incident angles between the shock front normal and the incident angle of the upstream flow, where stable was possible. However, in Edeleman's most recent work, an investigation of the fully three-dimensional shock system, he found these stability regions to be an artifact of the previous two-dimensional analysis and now supports the conclusion that oblique shocks are unconditionally unstable (Edelman, 1990; Edelman, 1993; Lesson & Desphande, 1967; Stone & Edelman 1995).

Some initial nonlinear numerical work was carried out by Stone to verify Edelman's linear results and these results, mostly conducted on two-dimensional cases but with some work in three-dimensions as well, support Edelman's conclusion that the regime is unconditionally unstable. In these unstable slow, strong magnetic shocks the growth of a corrugation instability occurs, which is a term used to describe a general class of shock instabilities that result in the deformation of shape of an initially well-defined shock front (Stone & Edelman 1995).

This research extends those earlier investigations with a fully three-dimensional, nonlinear examination that more thoroughly investigates the variations in these corrugation instabilities, or potentially lack thereof, under a much wider variety of initial conditions, relaxes the physical assumptions used to achieve the earlier results, and employs more modern numerical and computational methods and techniques. The goal of this research is the expanded understanding of the behaviors of shocks in the strong, slow magnetic shock regime to better define what drives these instabilities, resolve unanswered questions from previous work regarding their evolutionary variations and stability, and provide the foundational basis of understanding necessary to extend the investigation to include additional physics, specifically radiative cooling in the shock system.

To this end a new numerical and computational framework was developed to carry out the nonlinear investigation. The following chapters describe this framework, first the numerical methods and then their implementation into a new simulation system, IMOGEN, that was designed and developed to carry out the fully nonlinear, time-dependent, three-dimensional corrugation instability investigation for the strong, slow shock magnetic regime.

CHAPTER III

NUMERICAL MAGNETOHYDRODYNAMICS

While a number of different methods exist to solve the magnetohydryodynamics equations, (2.4) - (2.7), the most popular by far is to use Finite-Difference Time-Domain (FDTD), which in three dimensions is commonly referred to as Finite-Volume Time-Domain (FVTD). In FVTD a compact, bounded spatial domain is defined and that domain is discretized into an array of structured, regular finite volume elements. Each of these finite volume elements, or cells, represents a complete physical state of the simulated system within the volume represented by the cell with a set value, or values, for each primitive variable. The primitive variables are then collectively represented as a spatial field, i.e. a multi-dimensional array, with distinct values within each cell. With a single value per primitive variable per cell, the discretization is a piecewise constant approximation of the physical field where the value at a cell represents an average, uniform value for the entire cell. Obviously, by making this approximation, the larger the volume of the cell the greater the approximation and the less accurate the result (Liu & Liu, 2003; Tajima, 2004; Van Albada, et al., 1982).

As the advection equation is a differential equation in both temporal and spatial domains, time must be discretized as well. In FVTD temporal discretization is explicit, using a dynamic, iterative method of successive time-steps from initial time to final time. The details of which are described in the following sections.

3.1. Discrete Magnetohydrodynamics

Consider the advection equation, (2.1), in the one-dimensional case,

$$\partial_t q + \partial_x q v_x = 0 , \tag{3.1}$$

where q is a function in both space and time. The approach used to discretize this equation for use in a FDTD/FVTD method is to replace the derivatives with their most basic discrete representation,

$$\partial_x u \approx \frac{u(x+\Delta x) - u(x)}{\Delta x} , \tag{3.2}$$

which of course is a low-order approximation for a finite spacing, Δx. Using this formulation of the derivative (3.1) takes on the form,

$$q_i^{t+\Delta t} = q_i^t - \frac{\Delta t}{\Delta x}\left((q v_x)_{i_+}^t - (q v_x)_{i_-}^t \right) . \tag{3.3}$$

Here Δt is the discretized time-step, Δx is the length of the cell, and the superscripts and subscripts represent the value at which the functional quantity is evaluated in time and space respectively. For some time-step, Δt, the updated quantity, q, can be calculated for the cell, i, by evaluating the flux terms at the upper and lower boundaries of the cell, i_+ and i_-.

The main difficulty that arises from (3.3) is that the spatial discretization defines all of the quantities at the center of each discrete cell as the average value over that entire cell. Therefore, to carry out (3.3) the values of the advection terms need to be determined at the boundaries of the cell for each time-step in the solution. The simple approach would be to use a center-difference averaging of the values at the cell, i, and its neighboring cells, *i*-1 and *i*+1, which would look like,

$$\left(qv_x\right)_{i_+} = \frac{\left(qv_x\right)_i + \left(qv_x\right)_{i+1}}{2} \text{, and} \tag{3.4}$$

$$\left(qv_x\right)_{i-} = \frac{\left(qv_x\right)_i + \left(qv_x\right)_{i-1}}{2} . \tag{3.5}$$

Substituted back into (3.3), the equation takes on the form

$$q_i^{t+\Delta t} = q_i^t - \frac{\Delta t}{2\Delta x}\left(\left(qv_x\right)_{i+1}^t - \left(qv_x\right)_{i-1}^t\right) . \tag{3.6}$$

While successfully redefining the discretized advection equation in terms of values that exist in a FDTD/FVTD representation, this technique turns out to be unstable numerically. Numerical errors compounds rapidly in this formulation for the higher frequencies due to their truncation in the approximations. This can be shown by considering the plane wave solution to the advection equation, and using that to determine the dispersion relationship of the discrete advection equation (3.6). Inspection of the results shows a clear growth in the higher frequency Fourier modes, which rapidly dominate any numerical solution. For more detail on the analysis of the numerical instability in (3.6) associated references (Tajima, 2004; Trac & Pen; 2003).

It turns out that all discrete differencing schemes like (3.6), which are classified as spatially first-order because the discretized advection equation (3.3) includes only first order terms in space and time, are numerically unstable for the same reason. To do any sort of discrete computation, a higher-order differencing scheme is required (Van Leer, 1974; Trac & Pen, 2003).

3.1.1. Lax-Wendroff Scheme

The Lax-Wendroff scheme is the fundamental second-order differencing scheme. Instead of directly substituting the first order derivative approximation, (3.2), into the advection equation, a second-order Taylor series expansion in time is used to define the value of the quantity, q, at time *t+Δt*, and the advection equation is used to substitute the temporal derivatives with spatial ones. The resulting equation has the form,

$$q^{t+\Delta t} = q^{t} + \partial_x \left(q v_x \right)^t \cdot \Delta t + \partial_x \left(\partial_q \left(q v_x \right)^t \cdot \partial_q \left(q v_x \right)^t \partial_x q^t \right) \cdot \frac{\Delta t^2}{2} . \tag{3.7}$$

Using the same discretization methods as the first order technique described above, the discrete formulation of the Lax-Wendroff scheme has the form,

$$\begin{aligned} q_i^{t+\Delta t} &= q_i^t + \frac{\Delta t}{2\Delta x}\left(\left(q v_x \right)_{i+1}^t - \left(q v_x \right)_{i-1}^t \right) \\ &\quad + \frac{v_x}{2}\left(\frac{\Delta t}{\Delta x} \right)^2 \left(\left(q v_x \right)_{i+1}^t + \left(q v_x \right)_{i-1}^t - 2\left(q v_x \right)_i^t \right) . \end{aligned} \tag{3.8}$$

A comparison between (3.8) and the first-order scheme (3.6) reveals that the Lax-Wendroff scheme differs only by the addition of a second-order flux correction term. Unlike the first-order scheme of (3.6), which is unconditionally unstable, the Lax-Wendroff scheme turns out to be stable in well-defined, but potentially restrictive cases. The scheme is also only suitable for advecting fields consisting of low-frequency spatial modes. When higher frequency spatial modes, where high frequency is defined in terms of steep gradients between cells, the scheme exhibits numerical dispersion and diffusion that can be quite problematic (Lax & Wendroff, 1967).

A number of other second-order schemes exist, e.g. the Beam-Warming and Fromm schemes, that all use variations of the second order discretization of (3.1) to improve upon the reliability of higher frequency advection with varying degrees of success. However, no single second-order scheme can advect higher frequency spatial modes in a generally reliable way (Sweby, 1984).

3.1.2. Courant-Freidrichs-Lewy Condition

Despite the variations in second-order schemes, they all share the same numerical stability criteria,

$$\Delta t \leq \frac{\Delta x}{v_x} \quad . \tag{3.9}$$

This condition, (3.9), which is called the Courant-Freidrichs-Lewy (CFL) condition, sets the maximum allowed time-step for each iteration in a second-order advection scheme. The condition can be understood more intuitively in the form,

$$\frac{\Delta x}{\Delta t} \leq v_x \quad . \tag{3.10}$$

In this form it’s easier to see that the condition is setting a limit on the time-step based on the advection velocity of the cell, which prevents the quantity, *q*, at some cell *i*-1, from advecting completely through the cell *i* and into cell *i*+1 in a single step. The reason this is necessary is that the central differencing formulation relies only on the nearest neighbor cells. If the time-step were to allow the quantity, *q*, to advect from cell *i*-1 to cell *i*+1 without first advecting into cell *i*, the scheme would be unable to incorporate such transport and the advected quantity would no longer be conserved, resulting in an erroneous solution.

The impact of the CFL condition is best understood by considering an entire spatial domain. The solution at each cell is dependent upon the values of its neighboring cells, which requires the time-step to take on a global value for the entire spatial domain. The value that the CFL condition must take on is the most restrictive one, i.e. the value of the cell with the highest advection velocity and the smallest allowed time-step. Therefore, the highest advection velocity within the spatial domain controls the time-step and for problems with high dynamic advection velocity ranges is resource intensive and potentially prohibitively.

3.1.3. Higher-Order Schemes

A natural consequence to the limitations of a second-order scheme, the undesired numerical dispersion and diffusion in higher spatial mode advection and the limits imposed by the CFL condition, might lead one to the conclusion that if second-order schemes are better than first-order schemes, why not use an even higher order scheme instead.

The answer is that the higher order the scheme, the more problematic numerical dispersion and diffusion become. As the order of terms included in the advected cell increases, so does the spatial extent to which that cells depends. Referred to as the stencil, the relationship between the new quantity for a cell on its neighbors, grows with higher order. In the second order case, the stencil has a ±1 cell extent around the cell of interest. Increasing the order increases that extent, resulting in greater general diffusion for the scheme, something that is highly undesirable. The discretization of the spatial domain already imposes a minimum length scale, and in doing so an artificial viscosity. To introduce higher orders only exacerbates the problem. So while higher order schemes exist and are used when appropriate, their inherent properties are not generally used when exploring shock wave systems, where low diffusion and high-frequency advection are needed most (Liu & Liu, 2003; Sweby 1984; Tajima, 2004).

3.2. Shock Capturing

As already mentioned, the Lax-Wendroff and other second-order schemes are suitable for advecting fields consisting of low frequency spatial modes, but have difficulty advecting higher frequency spatial modes. The problem is that most numerical astrophysics simulations of interest involve shock waves, which, by their discontinuous nature, are high spatial frequency phenomena. With no alternative than to use a second-order scheme, the solution is to find a way to modify second-order schemes to handle shock discontinuities, an approach called shock capturing. There are a number of approaches to handling shock capturing. Early methods relied on front tracking, where shock discontinuities are explicitly identified and treated as boundaries between separate advection regions. This approach is successful at maintaining shock fronts but is very difficult to manage and use in general cases. More sophisticated forms of this technique remain in use today in cases where the physics happening on the two sides of the shock is different enough to warrant treating the two regions as separate fluids.

However, in astrophysics a shock-separated-fluid approach is rare, as the desire for greater flexibility in capturing general shocks is needed given the frequency in which they appear in problems of interest.

3.2.1. Upwind Schemes

A more general way to capture shocks is to make the differencing scheme adaptive by introducing directional, or upwind, biasing. The idea is to determine the direction of the advection within a cell and then preferentially pick the best differencing stencil for that direction. The result is a scheme that is much more stable than the static second-order schemes previously described and able to capture shocks. The scheme works by replacing the advection quantities defined at the boundaries in (3.3) with the following biasing,

$$\left(qv_x\right)_{i_+}^t = \begin{cases} \left(qv_x\right)_i^t & v > 0 \\ \left(qv_x\right)_{i+1}^t & v < 0 \end{cases}\text{, and} \tag{3.11}$$

$$\left(qv_x\right)_{i\_}^t = \begin{cases} \left(qv_x\right)_{i-1}^t & v > 0 \\ \left(qv_x\right)_i^t & v < 0 \end{cases} . \tag{3.12}$$

The resulting upwind scheme is, like (3.6), a first-order scheme. As such its first-order accuracy results in issues with high amounts of diffusion around shock waves, but, unlike the static second-order schemes, shock fronts do not introduce numerical dispersion problems (Trac & Pen, 2003; Van Leer, 1979).

To understand how different advection schemes compare in their handling shock capturing, and high frequency spatial mode advection in general, it is useful to test them with a square wave pulse with uniform advection velocity. By inspecting how the shape of the pulse deforms during the advection process the abilities and limitations of one scheme compared to another can be easily demonstrated. Figure 3.1 shows an example of both the Lax-Wendroff scheme and the upwind scheme advecting a square pulse.

The Lax-Wendroff scheme exhibits some diffusion around the edges of the pulse. It also suffers intense numerical dispersion, spurious oscillations, that ultimately overtakes and destroys any discontinuities. In comparison, the upwind scheme has no apparent numerical dispersion but suffers from intense diffusion around the pulse edges. Despite this intense diffusion, the upwind scheme is still considered better able to advect shock waves because the diffusion settles down once the shock front has been smoothed and it preserves monotonicity, and the preservation of monotonicity is

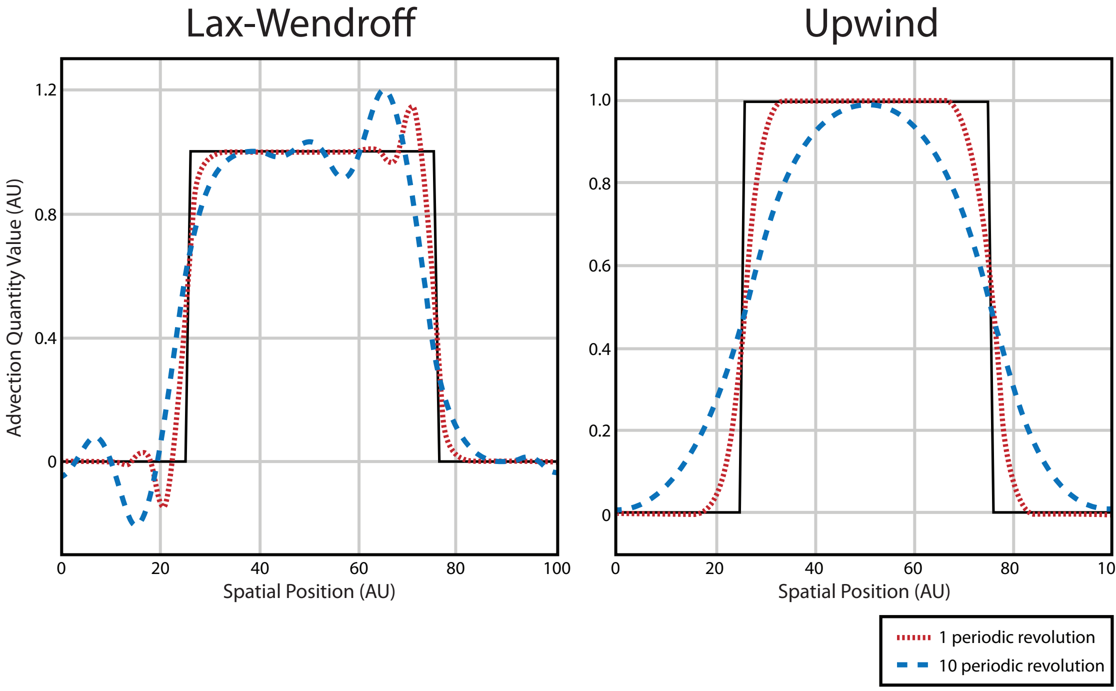


Figure 3.1: Example advection of a square wave with uniform advection velocity using the Lax-Wendroff second-order scheme (on left) and the first-order upwind scheme (on right).

extremely important in the stable numerical evolution of astrophysical problems.

Like the second-order static differencing schemes there are many variations of this upwind scheme, which are generally classified as Monotonic-Upwind Schemes for Conservation Laws, commonly referred to as MUSCL schemes.

There are also higher-order MUSCL schemes that reduce the intensity of diffusion around shock discontinuities. However, they rely on using higher-order differencing stencils that have extents greater than just nearest neighbor cells, which limit the diffusion correction, and are no longer really monotonic. Sergei Godunov was able to prove, in what is now called Godunov's Order Barrier Theorem, that monotonicity can only be preserved in a first-order differencing scheme (Wessling, 2001).

A second-order accurate scheme is certainly needed to reduce the diffusion of the MUSCL, but the correction cannot be to the differencing stencil itself because monotonicity must be preserved. The only way to do it is to add a correction term onto the scheme itself that also maintains monotonicty (Tajima, 2004; Toro, 1999).

3.2.2. Second-Order Total Variation Diminishing Schemes

A Total Variation Diminishing (TVD) scheme is any scheme in which monotonicity is preserved by obeying the properties that an updated quantity, $q_i^{t+\Delta t}$, cannot have more local extrema than its previous value, q_i^t, and that existing local extrema cannot grow, increase in amplitude, with each time-step. More specifically an advection calculation cannot introduce spurious oscillations from numerical dispersion and cannot exacerbate any existing spurious oscillations. All stable first-order schemes are TVD, including MUSCL, but as is noted the diffusion associated with the scheme is unsatisfactory and a second-order TVD scheme is needed.

The way such a scheme works is to modify the first-order MUSCL scheme with a second-order advection correction term that has been forced to obey the TVD properties by imposing a monotonic correction operator, called a flux limiter, to the calculated correction before applying it to the first-order advection term. This operation prevents new extrema from forming by removing the inherent dispersiveness of the correction term by limiting the advection at each cell boundary. A comparison of the square wave advection test between the first-order MUSCL and second-order TVD MUSCL schemes is shown in Figure 3.2.

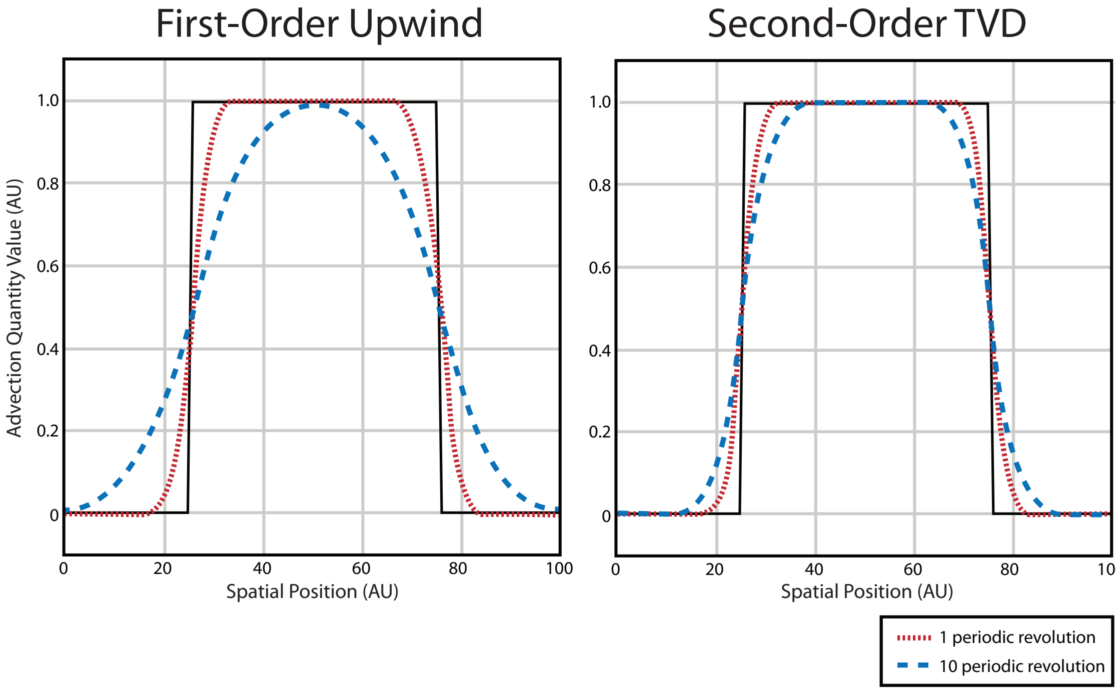


Figure 3.2: Comparison of the first-order MUSCL and second-order TVD MUSCL schemes for the square wave pulse advection test.

The second-order TVD MUSCL scheme, unlike other second-order schemes, has no apparent numerical dispersion. It also suffers much less diffusion around the pulse edges than the first-order MUSCL scheme, and the amount of diffusion is limited to a region of a few cells around the discontinuity; the diffusion does not compound continuously over time as it does in the first-order MUSCL scheme. In addition, the amount of diffusion can, to a great extent, be controlled by the choice of flux limiter used in the scheme (Trac & Pen, 2003).

3.2.3. Flux Limiters

Flux limiters are the application of a slope limiting operator to the conservative terms of the advection equations. To understand its application, one must first understand its purpose. For any discrete function, $f(x)$, a local parameter is defined, r_s, that represents the ratio of slopes between the function value at a cell, i, and its nearest neighbors, i-1 and i+1,

$$r_s \equiv \frac{S_-}{S_+} , \qquad (3.13)$$

where S_- is the slope between $f(x_{i-1})$ and $f(x_i)$ and S_+ is the slope between $f(x_i)$ and $f(x_{i+1})$. The slope ratio parameter, r_s, has four possible cases as shown in Figure 3.3.

The slope ratio cases 1, 2, and 4 are responsible for spurious oscillations, whereas case 3 is well behaved. Therefore, a slope limiter corrects differently for different slope ratios. For slope ratios less than or equal to zero, cases 1-2, no amount of correction will preserve monotonicity and the slope limiter correction must be zero. As the slope ratio tends toward unity, signifying smooth transitions between cells, the correction can be larger, with a maximum at a slope ratio of 1. As the slope ratio increases above 1, case 4, the correction is once again reduced to preserve monotonicity.

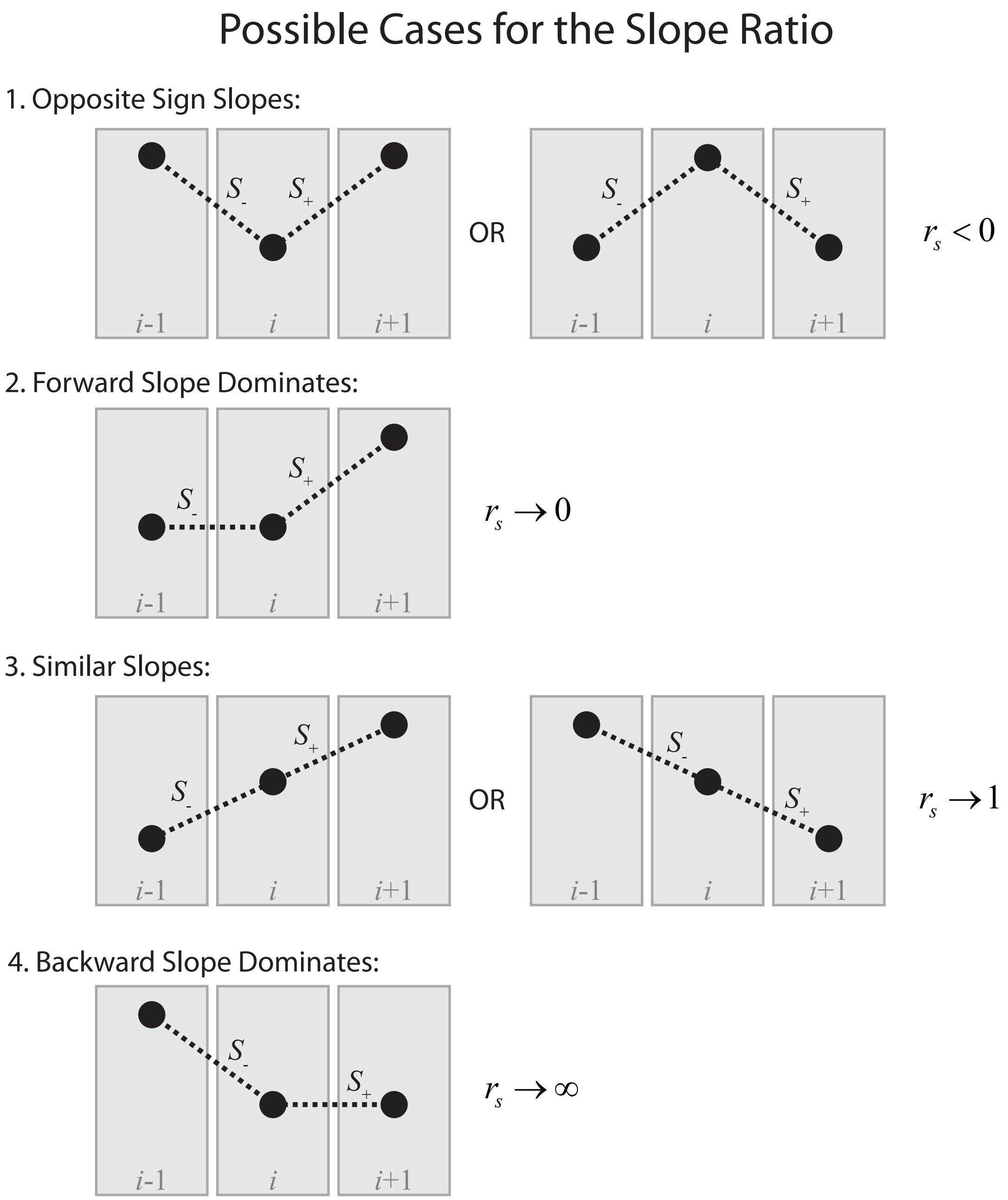


Figure 3.3: Possible behaviors for the slope ratio parameter, r_s, used in determining how a flux limiter should behave.

A more rigorous investigation of the intermediate slope ratios for the four slope ratio cases reveals a region, called the TVD limiting region, in which a slope limiting function must reside in order to preserve monotonicity. Therefore, any function that resides within this region for all possible slope ratio values is a valid slope limiter. For a detailed definition, discussion, and derivation of this region associated references (Sweby, 1984).

Applying the slope limiter to conservative advection, a flux limiter operates on the first order fluxes of a cell and its nearest neighbors, augmenting the first order flux with a second-order correction term to varying degree depending on the slope ratios around the cell of interest. A flux limiter works by taking two potentially competing flux correction terms as input and returns a correction term that maintains monotonicity in combination with the first-order MUSCL flux term for a given time-step. A limited flux has the general form,

$$\tilde{F} = F + \phi\left(\delta F_{-}, \delta F_{+}\right), \tag{3.14}$$

where $\tilde{F}$ is the corrected second-order TVD flux, F is the first-order MUSCL flux, ϕ is the flux limiter operator, and δF_{-} and δF_{+} are the two correction terms. The two flux correction terms are given as,

$$\delta F^{+} = \begin{cases} \frac{1}{2}\left(F_{i+1} - F_{i}\right) & v \geq 0 \\ \frac{1}{2}\left(F_{i+2} - F_{i+1}\right) & v < 0 \end{cases} \quad \text{and} \tag{3.15}$$

$$\delta F^{-} = \begin{cases} \frac{1}{2}\left(F_{i} - F_{i-1}\right) & v \geq 0 \\ \frac{1}{2}\left(F_{i+1} - F_{i}\right) & v < 0 \end{cases}. \tag{3.16}$$

Each of these corrections represents a second-order flux term that is upwind specific in accordance with the MUSCL scheme. The terms are also a measure of the gradient between first order flux values between adjacent upwind cells. Therefore, they serve the same function as the slopes did in the generic slope limiter; their ratio determines the appropriate correction.

Of the more than a dozen common slope limiting operators, three are popularly used as flux limiters in TVD MUSCL schemes. These three are MinMod,

$$\phi = \frac{1}{2}\left(\operatorname{sign}\left(\delta F_-\right) + \operatorname{sign}\left(\delta F_+\right)\right) \cdot \min\left(\left|\delta F_-\right|, \left|\delta F_+\right|\right), \tag{3.17}$$

where the *sign* operator returns ±1 depending on the sign of its input, Superbee,

$$\phi = \begin{cases} \operatorname{minmod}\left(2 \cdot \delta F_-, \delta F_+\right) & \text{if } \left|\delta F_-\right| < \left|\delta F_+\right| \\ \operatorname{minmod}\left(\delta F_-, 2 \cdot \delta F_+\right) & \text{otherwise} \end{cases}, \tag{3.18}$$

where the *minmod* operator is the MinMod slope limiter of (3.17), and Van Leer,

$$\phi = \frac{2 \cdot \delta F_- \cdot \delta F_+}{\delta F_- + \delta F_+}. \tag{3.19}$$

The Superbee flux limiter is the most aggressive, always choosing the steepest possible flux gradient that can be used to preserve monotonicity on a cell boundary. The MinMod flux limiter, on the other hand, is the least aggressive and chooses the smallest possible slope satisfying the monotonicity condition. The Van Leer flux limiter, is a moderation of these two extremes and one of only a few analytic flux limiters, which gives it a greater level of stability in high dynamic range problems. The second-order TVD MUSCL advection test in Figure 3.2 used a Van Leer flux limiter. Had a Superbee limiter been used the diffusion would have reduced to an even greater extent than the Van Leer case (Sweby, 1984; Van Leer, 1974).

3.3. The Relaxed Second-Order TVD MUSCL Scheme

This section outlines the complete numerical scheme used for the remainder of this work. The scheme is based on the second-order TVD MUSCL scheme, as described previously, and its particular formulation utilizes the relaxation technique, presented by Jin & Xin and popularized by Trac & Pen (Jin & Xin, 1995; Trac & Pen, 2003).

3.3.1. The Relaxation Technique

A significant omission in discussion of the MUSCL scheme above was how one applies it to the mangetohydrodynamic equations. While the mass density and magnetic field equations, (2.4) and (2.7), obey the simple formulation of the advection equation, the momentum and energy density equations, (2.5) and (2.6), do not. The more complicated flux terms in the energy and momentum equations lead to an upwind advection direction that depends not only on velocity but also on the other terms such as the pressure. As such, the asymmetry in the upwind scheme leads to a more involved method of determining the upwind direction for the equations at each time-step. Applying the relaxation technique to the advection equations results in a symmetric formulation of the second-order TVD MUSCL scheme where advection is inherently upwind and requires no additional calculation to determine the upwind direction as part of the advection process.

The relaxation technique is a reformulation of the advection equation that begins by separating it into a system of two advection equations with a shared advection velocity and a symmetry between the variables. Therefore, the general one-dimensional conservative advection equation,

$$\partial_t q + \partial_x F = 0, \tag{3.20}$$

formulated in the relaxation technique becomes,

$$\partial_t q + \partial_x (c\alpha) = 0 \quad \text{and} \tag{3.21}$$

$$\partial_t \alpha + \partial_x (cq) = 0. \tag{3.22}$$

Here *c* is the shared advection velocity, which should not be confused with the advection velocity expressed in the original advection equation, (3.20), and α is the modified flux needed to make the equations symmetric, which is given as,

$$\alpha = F/c. \tag{3.23}$$

These two equations can then be decoupled by a change of variables,

$$Q^{+} = \frac{q + \alpha}{2} \quad \text{and} \tag{3.24}$$

$$Q^{-} = \frac{q-\alpha}{2}, \tag{3.25}$$

which satisfy the relationship,

$$q = Q^{+} + Q^{-}. \tag{3.26}$$

While the change of variables may seem to be chosen arbitrarily at first glance they are the two characteristic variables of the relaxed system, (3.21) and (3.22). Their significance becomes clear by applying the change of variables that reduces (3.21) and (3.22) to

$$\partial_t q^{+} + \partial_x \left(c Q^{+} \right) = 0 \text{ and} \tag{3.27}$$

$$\partial_t q^{-} - \partial_x \left(c Q^{-} \right) = 0, \tag{3.28}$$

which represents two advected quantities, one moving in the forward (up) direction and the other moving in the backward (down). Forward and backward are used in this context to represent the two possible directions of advection in a one-dimensional system where forward advection moves in the direction, $i \Rightarrow i+1$, and backward advection movies in the direction, $i \Rightarrow i-1$. Given this form, the change of variables each represent a conserved quantity advected purely in either the forward direction, q^{+}, or in the backward direction, q^{-}.

Adding these two equations together, and simplifying the temporal derivative using (3.26), produces a single advection equation similar to the original, (3.20), but now with independent forward-moving and backward-moving flux advection terms,

$$\partial_t q + \partial_x c\left(Q^+ - Q^-\right) = 0 \quad . \tag{3.29}$$

With the forward and backward moving flux terms now separate and explicit, the combination represents the correct total flux for the system. For a simple advection equation of the form (3.1), with only a single directed flux, one of these flux terms will be zero at each time-step as the advection will either be forward moving or backward moving. However, for more complicated flux terms, like the momentum and energy equations, (2.5) and (2.6), which allow for both forward and backward advection simultaneously, the two flux terms can both be non-zero and contribute to provide the correct total flux in the system.

The artifact of this transformation process into the relaxed advection equation, (3.29), is the advection velocity of the separate quantities, c, that was introduced in (3.21) and (3.22). The quantity is called the freezing speed as it is the frozen speed at which the separate conserved quantities propagate. The parameter is free under the constraint,

$$c \geq |v| + c_s \quad , \tag{3.30}$$

which is the largest eigenvalue of the hydrodynamic equations. Here, c_s, is the sound speed. The reason for ignoring any magnetic contributions in the freezing speed will become clear shortly. In practice Jin & Xin set the freezing speed to a global constant for each time-step. However, the freezing speed is a function by its definition and Trac & Pen have found greater shock definition by allowing it to vary in space for each time-step (Chalabi & Qiu, 2000; Jin & Xin, 1995; Pen, et al., 2003; Schroll, 2002;

Schroll, 2004; Trac & Pen, 2003).

3.3.2. The Complete Scheme

The relaxed advection equation, (3.29), can then be discretized using the second-order TVD MUSCL scheme in the same fashion as before, with a few modifications. The most important modification is that instead of having to explicitly determine the upwind direction and apply the correct differencing stencil accordingly, the two possible differencing stencils are applied consistently to their correct forward or backward flux terms in (3.29). The discretized form of (3.29) is then,

$$q_i^{t+\Delta t} = q_i^t - \frac{c\Delta t}{\Delta x}\left(Q_{i_-}^{+,t} + Q_{i_+}^{-,t}\right), \tag{3.31}$$

where $Q_{i_-}^{+,t}$ is the forward moving flux at the lower cell boundary at time t, and $Q_{i_+}^{-,t}$ is the backward moving flux at the upper cell boundary at time t. Applying the appropriate upwind differencing stencil to each flux term and applying flux-limiting to the fluxes yields the final form of the discrete scheme,

$$q_i^{t+\Delta t} = q_i^t - \frac{c\Delta t}{\Delta x}\left(\tilde{Q}_i^{+,t} - \tilde{Q}_{i-1}^{+,t} + \tilde{Q}_i^{-,t} - \tilde{Q}_{i+1}^{-,t}\right), \tag{3.32}$$

where the fluxes are limited according to,

$$\tilde{Q} = Q + \phi\left(\delta Q_-, \delta Q_+\right). \tag{3.33}$$

The upwind determination is now handled implicitly within (3.32) by the summation of the oppositely oriented flux advection terms. The term with the greatest magnitude dominates, which correctly specifies the upwind direction for that particular cell for a given advection operation.

3.3.3. Extending to Multiple Dimensions

Until now the numerical schemes have been presented for a single dimension. The reason it has been possible to avoid discussing them in terms of multi-dimensions is that an advection operation can be broken down into multiple one-dimensional steps that combine into a single multi-dimensional advection operation. This approach is called dimensional splitting and relies on treating each dimension as a separate advection operation, *A*. Therefore, the operator A_x would advect the system forward one time-step in the *x* direction, corresponding to a one-dimensional advection operation describe above for each equation in the magnetodydrodynamic equations. The total multi-dimensional advection operation can be represented in terms of these dimensionally split operators, A_i, as,

$$q^{t+2\cdot\Delta t} = A_x A_y A_z A_z A_y A_x q^t \quad , \tag{3.34}$$

where A_x, A_y, and A_z are the split, single-dimension advection operations on the conserved quantity, q^t, that update it to $q^{t+\Delta t}$. To preserve second-order accuracy and prevent directional biasing in advection, a complete advection operation requires two full time-steps with the same Δt (Ryu, et al., 1998; Trac & Pen, 2003).

3.3.4. Temporal Integration

With a complete advection scheme, the final piece is to incorporate it into the temporal integration scheme. The most basic temporal integration scheme would be to update the conserved quantity, q, using (3.32) iteratively, in an Euler-like method, until reaching the desired final time. However, this scheme is only first-order accurate and would undo any second-order accuracy in the spatial scheme. To get around this, temporal integration is handled using a second-order Runge-Kutta scheme which breaks each time-step into two integration stages.

The first stage is used to calculate an intermediate value for the conserved quantity at time $t+\Delta t/2$. For the second-order relaxed TVD MUSCL advection equation, (3.32), this is,

$$q_i^{t+\Delta t/2} = q_i^t - \frac{c\Delta t}{2\Delta x}\left(\tilde{Q}_i^{+,t} - \tilde{Q}_{i-1}^{+,t} + \tilde{Q}_i^{-,t} - \tilde{Q}_{i+1}^{-,t}\right). \tag{3.35}$$

This is a prediction step with fluxes calculated using the previous value of the conserved quantity, q_i^t. This intermediate is then used to recalculate the flux terms for the second stage that steps forward the complete time-step to $t+\Delta t$. This correction step is,

$$q_i^{t+\Delta t} = q_i^t - \frac{c\Delta t}{2\Delta x}\left(\tilde{Q}_i^{+,t+\Delta t/2} - \tilde{Q}_{i-1}^{+,t+\Delta t/2} + \tilde{Q}_i^{-,t+\Delta t/2} - \tilde{Q}_{i+1}^{-,t+\Delta t/2}\right). \tag{3.36}$$

An optimization to the predictor step of this method is possible because the intermediate prediction value is used only to update the flux terms for the final correction step; the intermediate never appears explicitly in the correction step. Since the correction step fluxes are second-order flux limited, there is no need to limit the fluxes on the predictor step. The flux limiters, therefore, need only be applied to the correction fluxes, allowing (3.35) to be,

$$q_i^{t+\Delta t/2} = q_i^t - \frac{c\Delta t}{2\Delta x}\left(Q_i^{+,t} - Q_{i-1}^{+,t} + Q_i^{-,t} - Q_{i+1}^{-,t}\right), \tag{3.37}$$

which is just a relaxed form of the first-order MUSCL scheme. This optimization results in a significant reduction in computational needs to calculate a single advection time-step with no degradation in the accuracy or stability of the updated result (Trac & Pen, 2003).

3.4. Magnetic Field Advection

The second-order relaxed TVD MUSCL scheme described in the previous section is complete and sufficient for advection of the hydrodynamic evolution equations, (2.4) - (2.6), of the magnetohydrodynamic equations. However, without further modification the scheme will not correctly evolve the magnetic field advection equation, (2.7), because the magnetic field is under the additional divergence-free constraint,

$$\nabla \circ \boldsymbol{B} = 0, \tag{3.38}$$

which is not explicitly contained within magnetohydrodynamic equations. It is implicit in the magnetic field equation, meaning the constraint will hold for any analytical solutions whose initial conditions satisfy the constraint, but that does not hold true for the discretized form of the equation (Toth, 2000).

The reason it does not hold in the discretized case is that the constraint cannot be maintained everywhere at once because the magnetic field is represented by piece-wise constants at each cell. As such, the constraint cannot be simultaneously satisfied for both the center of the cells, where the primitive variables reside, and the cell boundaries through which advection occurs.

No way exists to formulate a discretization of the magnetic field equation that will be able to generally satisfy the divergence-free constraint at all critical locations within the cell at the same time for any possible magnetic field. Instead the techniques for advecting the magnetic field in a divergence-free manner focus on maintaining the constraint only when it really matters, during advection.

The constrained transport (CT) advection method focuses on maintaining the divergence-free constraint on the boundaries of the cells and ignores any divergences at the cell centers. This bias is chosen because the evolution of the magnetic field occurs on the cell boundaries and, as long as the evolution is divergence free, the advection process will tend toward preserving the constraint as it would in an analytical solution. For this approach to work, the magnetic field has to be redefined to exist on the component-specific edges of the cells instead of at their centers like the other primitive variables as shown in Figure 3.4.

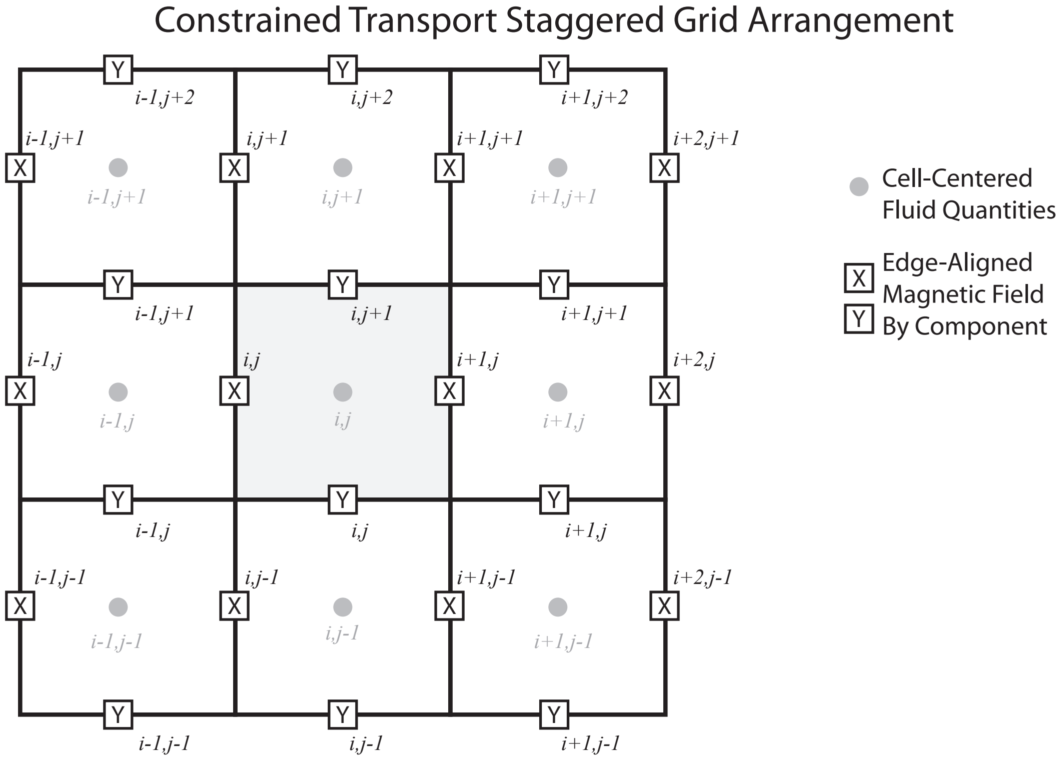


Figure 3.4: Staggered grid arrangement of the magnetic field for constrained transport. Cell indices in gray represent the grid values for the cell-centered quantities for mass, momentum, and energy densities, while cell indices in boxes represent the staggered indices for the magnetic field by component.

As Figure 3.4 illustrates the magnetic field for a given cell is broken up into its vector components, each of which reside on the component-specific lower boundary of the cell in which its corresponding fluid variables reside. With this staggered cell arrangement between the hydrodynamic primitive variables and the magnetic field it is then possible to maintain the divergence-free constraint on the edges of the cells and keep the constraint near zero during advection because the magnetic field now cohabits with its advection fluxes (Balsara & Kim, 2004; Toth, 2000).

This is, however, not a perfect solution. The magnetic field appears in the flux terms of more than just the magnetic field evolution equation. Given that these other equations are advected according to cell-centered quantities, the magnetic field in these equations must also be cell-centered. To produce a cell-centered magnetic field for use in advection of these equations, a second-order accurate interpolation is carried out on the magnetic field values that reside on the cell boundaries. Similarly, as the momentum is defined at the center of the cells, an advection velocity must be produced at the cell edges for the magnetic advection step, which is done using the same second-order accurate interpolation methods as for the cell-centered magnetic field case.

Both of these interpolations introduce some amount of error into magnetic advection, which in turn can produce some divergence of the magnetic field. The interpolation errors are small, unbiased, and do not compound, resulting in divergence constraint errors at least a few, but often many, orders of magnitude below the accuracy threshold of the general advection scheme. Consequently, the constrained transport technique is generally viable for all but a few cases, none of which pertain to this work.

With the component-wise staggered grid, the discretized advection equation for magnetic field evolution becomes,

$$\boldsymbol{B}_i^{t+\Delta t} = \boldsymbol{B}_i^t - \sum_{n \neq m} \frac{\Delta t}{\Delta x_n} \left(\tilde{F}_n^t - \tilde{F}_{n-1}^t \right). \tag{3.39}$$

Here n represents the summed dimensions over which to flux the m^{th} component of the magnetic field. There is, for example, no flux operation for B_x along the *x*-direction because the flux terms of the magnetic field evolution equation, (2.7), are anti-symmetric when the magnetic field and velocity vector components are the same dimension and hence the flux term is always zero. Equation (3.39) makes this anti-symmetric relationship explicit in the discretized advection scheme to avoid the computational resources associated with the zero-invariant advection component (Balsara & Kim, 2004; Dai & Woodward, 1998; Evans & Hawley, 1988; Gardiner & Stone, 2008; Londrillo, et al., 2004; Ryu, et al., 1995; Toth, 2000; Touma & Arminjon, 2006; Ziegler, 2004).

Notice also that (3.39) is not a form of the relaxed advection equation, for example, there is no freezing speed in the flux terms, but instead just the second-order TVD MUSCL scheme. The reason for this is the flux term in the magnetic advection equation is simple enough in form that velocity can be used to determine the upwind direction. For this advection equation no additional computation from either a direct upwind determination or the symmetric fluxing of the relaxed technique is required.

Combined with the Constrained Transport advection method for the magnetic field evolution equation, the second-order relaxed TVD MUSCL scheme is complete, fully capable of advecting the magnetohydrodynamic equations (2.4) - (2.7) for general astrophysical computation in the presence of shock waves when implemented as a magnetohydrodynamics simulation code.

CHAPTER IV

IMOGEN: ELEMENTS OF IMPLEMENTATION

IMOGEN is a new nonlinear magnetohydrodynamics simulation software created to implement the second-order, relaxed TVD MUSCL scheme presented in chapter 3 for the investigation of the corrugation shock instability discussed in chapters 1 and 2. While IMOGEN was developed as a general-purpose magnetohydrodynamics research tool, and has been used for investigations beyond the scope of this work, it implements a number of approaches and techniques specifically for the corrugation instability investigation. The following sections describe the key aspects of IMOGEN critical to the corrugation instability investigation. The complete IMOGEN software design and implementation description can be found in Appendix I.

4.1. Cross-Component Magnetic Advection

A complication in the implementation of the Constrained Transport advection method described in section 3.4 that isn't immediately apparent in the discrete magnetic field advection equation (3.39) is the shared, antisymmetric nature of the various advection terms in the magnetic field equation due to the cross product from which the advection term is formulated. To see how this behaves, recall the component-wise magnetic advection equations presented in Figure 2.1, expanded out for directionally-split component advection as,

$$\partial_t B_x = \partial_y \left(v_y B_x - B_y v_x \right) + \partial_z \left(v_z B_x - B_z v_x \right), \tag{4.1}$$

$$\partial_t B_y = \partial_x \left(v_x B_y - B_x v_y \right) + \partial_z \left(v_z B_y - B_z v_y \right) \text{, and} \tag{4.2}$$

$$\partial_t B_z = \partial_x \left(v_x B_z - B_x v_z \right) + \partial_y \left(v_y B_z - B_y v_z \right) . \tag{4.3}$$

In this form is readily apparent that the flux terms between components are shared, just with opposite signs. For example, the flux term for B_x in the y direction is just the negative of the flux term for B_y in the x direction. Similarly, the flux terms for B_x in the z direction is just the negative of the flux term of B_z in the x direction, and B_y in the z direction is just the negative of B_z in the y direction.

For magnetic advection that maintains the divergence-free constraint, these shared terms have to be calculated using the same magnetic field component values as their antisymmetric counterpart. This means using the same values for both flux terms even though the advection directions, as specified by the component of the derivative in which they reside, differ between the two antisymmetric instances. If the same values are not used in both instances, then all of the work done by implementing the Constrained Transport advection technique is lost as the variations between terms fail to adhere to the divergence-free constraint equation, (3.38). This would not be an issue, except that a complete conservative flux operation is carried out using multiple dimensionally split advection steps, see section 3.3.3 for details, and the values for the magnetic field components are updated intermediately with each of these steps.

The direct approach to solving this problem would be to calculate all of the flux terms at the beginning of the magnetic advection step and use those as a reference instead of the intermediate ones for each dimensionally split operation. While solving the problem this introduces another, which is a significant increase in memory usage to support a complete magnetic flux operation.

A more sophisticated approach that resolves the memory usage is to carry out the advection in a cross-component fashion (Pen, et al., 2003). In cross-component advection, each dimensionally-split advection step is responsible for calculating only the first, positive, portion of the flux term and using that for advection. This value is then immediately applied as the negative part of the flux term to its antisymmetric partner component. In doing this, remember that the components of the magnetic field exist at different edges of the cell, as shown in Figure 3.4, and that fluxing occurs on those edges, which requires that the application of the flux term to its partner component be interpolated from one edge to another. In this approach the advection process shares the necessary values with its counterpart, and the divergence-free constraint is maintained, while at the same time no reference data was needed and accordingly no additional memory used during the fluxing process.

4.2. Non-Conservative Terms

The magnetohydrodynamics equations, (2.4) - (2.7), presented in chapter 2 are absent any non-conservative terms. However, as part of the corrugation instability problem, as well as the general purpose use of IMOGEN, non-conservative terms must be handled as well as the conservative ones for the many processes involved in astrophysical simulation that cannot be resolved into a conservative formulation. The two non-conservative terms used in the corrugation investigation were radiation and artificial viscosity, the former being a physical property of the simulation and the latter being a computational technique to better resolve shocks under certain conditions.

Radiation was handled within a general sourcing routine that is paired with a complete flux action and handled within the second-order accurate temporal integration scheme described in detail in section 3.3.4. While the flux and sourcing happen separately from each other as distinct steps, they coexist within a single time-step and can be thought of as a multi-step process in which IMOGEN fluxes the magnetohydrodynamic equations from t to $t+\Delta t$ and then sources the magnetohydrodynamic equations at $t+\Delta t$ using the updated values from the flux step.

As sourcing operations are not a form of transport like the conservative fluxing operation, sourcing terms typically respond to the primary variable state and not process. This means that they will act non-conservatively to some extent with each time-step in response to the new state of the primary variables as determined by the flux operation. Instead of having to handle source terms over the range t to $t+\Delta t$, as fluxing is handled, they can simply be handled as a post-fluxing operation at $t+\Delta t$. It also means that source terms don’t usually have the well defined temporal integration restrictions that flux operations have in the form of the Courant-Freidrichs-Lewy (CFL) condition discussed in section 3.1.2.

Consequently, the global temporal integration for non-conservative forms of the magnetohydrodynamic equations remains governed entirely by the CFL condition. This can cause unphysical and undesirable behaviors in the evolution of a simulation if a source term renders one or more of the magnetohydrodynamic equations stiff, where the time-step for stable sourcing needs to be smaller than the CFL condition prescribes.

There are stiff integration techniques that can be implemented to try and circumvent this problem, but in many cases they are of limited use. Most such techniques rely on calculating the primary variables at a range of intermediate values over the offending time-step and using them progressively to stabilize and resolve the offending term. In a one-dimensional integration problem this may be viable, but for multi-dimensional magnetohydrodynamic simulation the memory requirements would be outrageous and intractable for all but the smallest spatial domain resolutions. Without *a priori* knowledge of the unphysical step, these unphysical source operations can plague simulations. As such, sourcing is often equal parts trial and error and a dark art, and rely heavily on successful techniques found previously by others. Often the best way of introducing a source term into a magnetohydrodynamics simulation is to include with it a coefficient that acts as a limiter to the magnitude of sourcing and carefully adjust that parameter to reach its stability limit. If the limit is unsatisfactory, then two options are available, either recast the source term into a form more conducive for stable integration or override the CFL condition to force smaller time-steps.

4.2.1. Radiation

The radiation source term is applied to the energy equation, (2.6), in one of many possible forms depending on the type and sophistication desired. For the preliminary and limited investigation of radiating corrugation instabilities contained within this work only optically-thin radiative cooling was explored, one of the more basic forms. The parameterized form of the optically-thin radiative cooling term is,

$$\Pi_R = \alpha \rho^{\chi} T^{\mu} , \tag{4.4}$$

where α is a controlling coefficient, T is the temperature, and χ and μ are adjustable parameters that specify key properties of the radiation model. Under the ideal gas assumption the equation can be easily rewritten in terms of more primitive simulation variables as,

$$\Pi_R = \alpha P_g^{\mu} \rho^{\chi-\mu} , \tag{4.5}$$

where P_g is the gas pressure. The χ parameter specifies the type of radiation, with a value of 2 specifying purely particle-particle collisions, and values below that some combination of particle-particle collisions and particle-field collisions. In the presence of high magnetic fields, its reasonable to assume particle-field interactions play a role in the cooling process, and so the χ parameter would be less than 2 for the corrugation instability trials. The μ parameter, as shown more clearly in (4.4), represents strength of the dependence of the cooling function on temperature. A value of $\mu = ½$ is the well-known Bremsstrahlung particle-particle radiation.

The α parameter is included in the formulation of the optically-thin radiation term as a limiter to prevent the numerical instabilities of the type described above from occurring as a result of rapid cooling. The α parameter is also typically a function of density. Such density dependence is constructed to quench cooling efficiency as the density increases to keep internal energy from dropping too much anywhere in the simulated spatial domain.

All of these parameters are numerically sensitive, making radiatively cooled shock simulations a formidable challenge, and a primary reason why this area remains largely unexplored.

4.2.2. Artificial Viscosity

Artificial viscosity, as the name implies, is not a physically derived term in the magnetohydrodynamic equations. Instead, it is a computational technique designed to remove stiffness from a hydrodynamics simulation caused by undesirably high velocities within a simulation domain. This type of undesirable velocity typically occurs when a shock front converges enough that it falls below the resolution of the spatial domain, which can result in dramatic increases in velocity as the shock wave essentially feeds on itself.

In addition to the problem of under-resolution of the shock front causing localized unphysical velocity spikes, these velocity spikes in turn dictates the CFL condition and reduces the time-step of the temporal integration for the entire simulation. Very quickly, velocities can become high enough that the time-step drops by one or more orders of magnitude, bringing the simulation to a halt. For simulations that involve strong shocks, such as the corrugation instability, which already have fairly small time-steps as a result of the shock, this velocity growth makes completion of the simulation impractical. Even more frustrating is that these velocities spikes occur in only a few cells within the domain, and those few cells dictate the time-step for the entire simulation for the rest of the simulation.

To get around this problem the velocity needs to be kept from growing too large in such a way that it doesn't destroy the simulation or negatively impact its physical evolution. In order to do that the velocity cannot simply be clamped or bled from the offending cells, as the conservation of momentum, (2.5), reacts negatively to that type of action and does destroy the physicality of the solution.

The solution is to employ an artificial viscosity, which acts to convert the velocity, or more precisely the kinetic energy, into internal energy of the fluid. This is carried out in a systematic fashion that acts only on the velocity spikes by targeting gradients, i.e. shock fronts, that are converging. In such cases, the artificial viscosity term inflates the pressure of the converging front. The increase in pressure acts against the convergence and results in a smoothed shock front that is resolved over a few spatial domain cells, preventing the under-resolved degree of convergence that would produce the unphysical velocity spikes.

Used carefully, artificial viscosity can limit or prevent the velocity spikes without negatively impacting the evolution of the simulation.

IMOGEN has been designed to use many different formulations of artificial viscosity, each of which has benefits and drawbacks. For the corrugation instability, only the most commonly used type, Neumann-Richtmyer artificial viscosity was used, as it is designed for this type of simulation. In the Neumann-Richtmyer formulation, the artificial viscosity is represented by linear and quadratic term such that,

$$\tilde{P} = P + \begin{cases} \rho \Delta x \left(\nabla \circ \boldsymbol{v} \right) \left[\alpha c_s + \beta \Delta x \left(\nabla \circ \boldsymbol{v} \right) \right] & \nabla \circ \boldsymbol{v} < 0 \\ 0 & \nabla \circ \boldsymbol{v} \geq 0 \end{cases}, \qquad (4.X)$$

where P is the unmodified pressure, $\tilde{P}$ is the pressure with artificial viscosity, c_s is the sound speed, α is the linear strength coefficient, and β is the quadratic strength coefficient. The artificial viscosity term is only non-zero if the divergence of the velocity is negative, or the pressure is unmodified. The artificial viscosity term is also directly related to the magnitude of convergence, which acts to smoothly increase its effect as convergence grows.

The α and β parameters are controlling parameters that limit the strength of the artificial viscosity. They are set as part of the initialization for a simulation on the range [0, 1]. When applying artificial viscosity to a new simulation it is typical to start with control parameter values near zero and turn them up only as much as necessary to affect the simulation as little as possible (Liu & Liu, 2003).

4.3. External Boundary Conditions

One of the most challenging aspects in developing a magnetohydrodynamics code is creating a reliable method to handle the external boundary conditions created at the edges of the finite spatial domain. What makes it such a challenge is the need for astrophysical simulations to use the smallest possible spatial domain to reduce the computational load required for a given simulation. Instead of being able to set the boundaries far from the interesting simulation region, at an effective infinity, these edge boundaries must instead be setup as close to, or in many cases within, the domain of interest. With these boundary conditions *in media res* they couple strongly to the evolution of the system and must be extremely well behaved or they will negatively impact results by either incorrectly influencing the simulation evolution or, just as often, impede or destroy a simulation entirely.

The most commonly used approach to handling boundary conditions is to pad the edges of the domain with ghost cells. These ghost cells are set independently of the conservative fluxing and non-conservative sourcing routines according to prescribed behaviors set prior to simulation. The ghost cells exist on all spatial arrays, which allows for the control of both Dirichlet and Neumann boundary types. However, in most implementations only Dirichlet conditions are used. This type of ghost cell padding is easy to implement and fairly easy to stabilize but has two significant drawbacks.

The most obvious drawback is increased memory usage. For a second order accurate flux algorithm, which requires a second order stencil, two ghost cells must be applied to all array edges for all spatial arrays. The result is a dramatic increase in the number of cells in each spatial array with a similarly dramatic increase in the required memory. This is a huge drawback considering that available memory is often the primary limitation of the spatial simulation domain size despite recent advances in memory availability and allocation.

The other drawback of the ghost cell approach is that the ghost cells must be recalculated, affectionately called cleaned up, prior to each flux operation or the values will be incorrect enough to destabilize the boundary and potentially cause non-physical feedback for all but the most well behaved boundaries. Often this cleanup action occurs multiple times for many different arrays during a complete time-step, resulting in a lot of additional but undesirable maintenance in the fluxing and sourcing algorithms.

Another approach to handling external boundary conditions, used in many toy magnetohydrodynamic codes, is to implement the most basic boundary condition possible, a tiled or circular boundary in a dynamic fashion. In a circular boundary each domain edge uses its own values at the opposite edge in calculating its derivatives. Handled dynamically, instead of padding the arrays with ghost cells, these values are determined directly by shifting the cells off of one edge of the array and adding them back on to the other. In this method there are no ghost cells to increase memory usage and no maintenance required to correct the edge conditions as they remain correct by design. The obvious drawback of this approach is that it only allows for one specific type of boundary, which is of limited use in most astrophysical simulations.

IMOGEN uses a new model, called dynamic edge shifting, that extends the idea of the dynamic circular boundary approach to handle general boundary conditions to achieve the flexibility of the ghost cell approach with the memory footprint of the circular boundary approach. Dynamic edge shifting, as the name implies has no pad cells, instead new cells are generated as they are needed based on functional relationships to the current state of the array, and, like the circular boundary method, there is no overhead or maintenance associated with regular cleanup of the boundary conditions. As the new cell values are generated on demand, the process is quite efficient.

Yet another benefit of this new approach is that the cells created dynamically during the shifting process can be made dependent on the exact physical and computational state of the simulation, instead of during a more global cleanup phase. Hence, this on demand approach allows for the robust implementation and testing of non-trivial boundary conditions without having to specially integrate them into the flux routines on a per-case basis.

Dynamic edge shifting begins with the initialization of the simulation; see Appendix I for details on that process and how expressing the boundary conditions fits into that process. During initialization the user specifies what type, of many possible types, of dynamic edge shifting they would like to apply to a specific boundary and a specific array. Boundary types can be mixed and matched on a per-array basis to provide a flexible and extensible way of controlling the external boundaries for any type of simulation. Figure 4.1 shows the assignment hierarchy available in IMOGEN during simulation initialization.

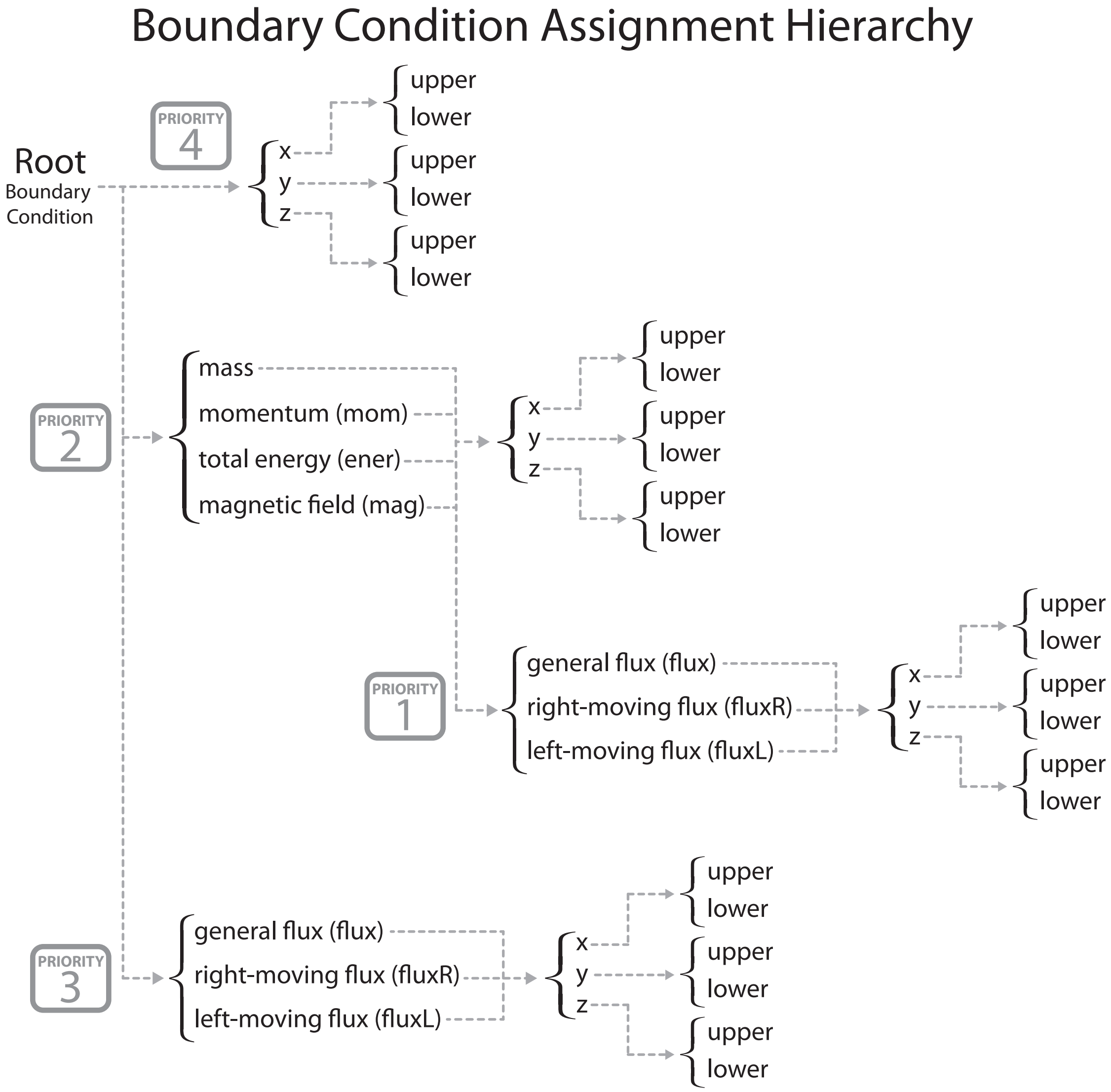
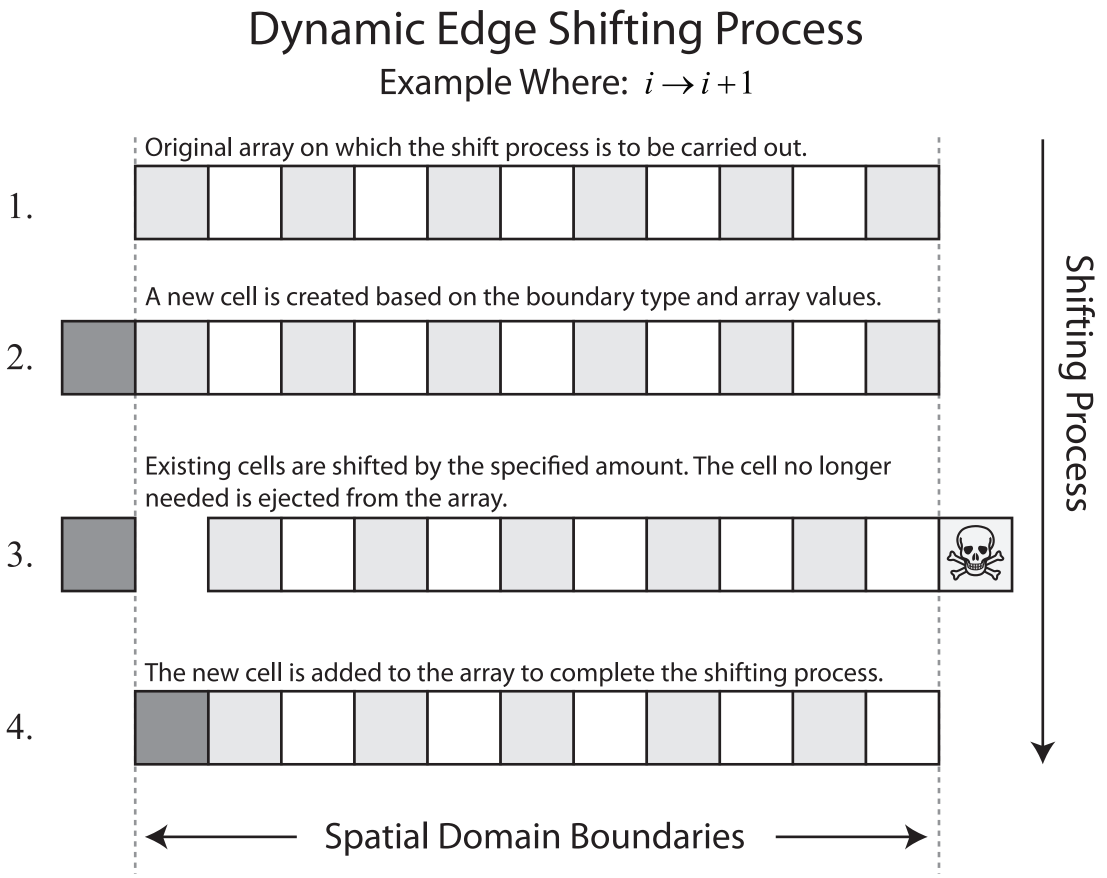


Figure 4.1: Assignment hierarchy for external boundary conditions during simulation initialization in terms of its inheritance tree.

The hierarchy uses priority-based inheritance to greatly reduce the complexity associated with the assignment requirements of the boundary conditions. If a specific value is not specified on a leaf entry, IMOGEN will navigate up the tree until it finds a non-null boundary value type specification and assign that value to the specific array. If, while traversing the tree, an assigned value is found on a different part of the tree with a higher priority the traversal process is aborted and that value is used. Note that the hierarchy includes both the primary arrays and their associated fluxes, which allows for the independent setting of both Dirichlet and Neumann boundary conditions.

When an IMOGEN simulation begins, and the spatial arrays are instantiated, this tree is parsed for each of the arrays and the enumerated values for each external boundary are assigned to the array object as a pointer to the correct shifting function. If initial parameters must be stored for a particular type, this is also done during construction of the array object. All of this occurs within the boundary conditions object instantiated and owned by its associated array.

Once complete, the array is ready to be dynamically shifted, a process that may occur many times during each flux and sourcing operation. During one of these calls the array object executes its assigned shifting function appropriate to the boundary being shifted and that function returns the array of dynamically create cells that are then appended onto the array boundary while the same number of cells on the opposite edge, which are no longer needed, are discarded as shown in Figure 4.2.



Dynamic Edge Shifting Process
Example Where: $i \rightarrow i+1$
Original array on which the shift process is to be carried out.
1.
A new cell is created based on the boundary type and array values.
2.
Existing cells are shifted by the specified amount. The cell no longer needed is ejected from the array.
3.
The new cell is added to the array to complete the shifting process.
4.
Spatial Domain Boundaries
Shifting Process

Figure 4.2: Illustration of the shifting process where a new cell is created dynamically and added onto one end of the array while the cell on the other end, which is no longer needed, is simultaneously discarded.

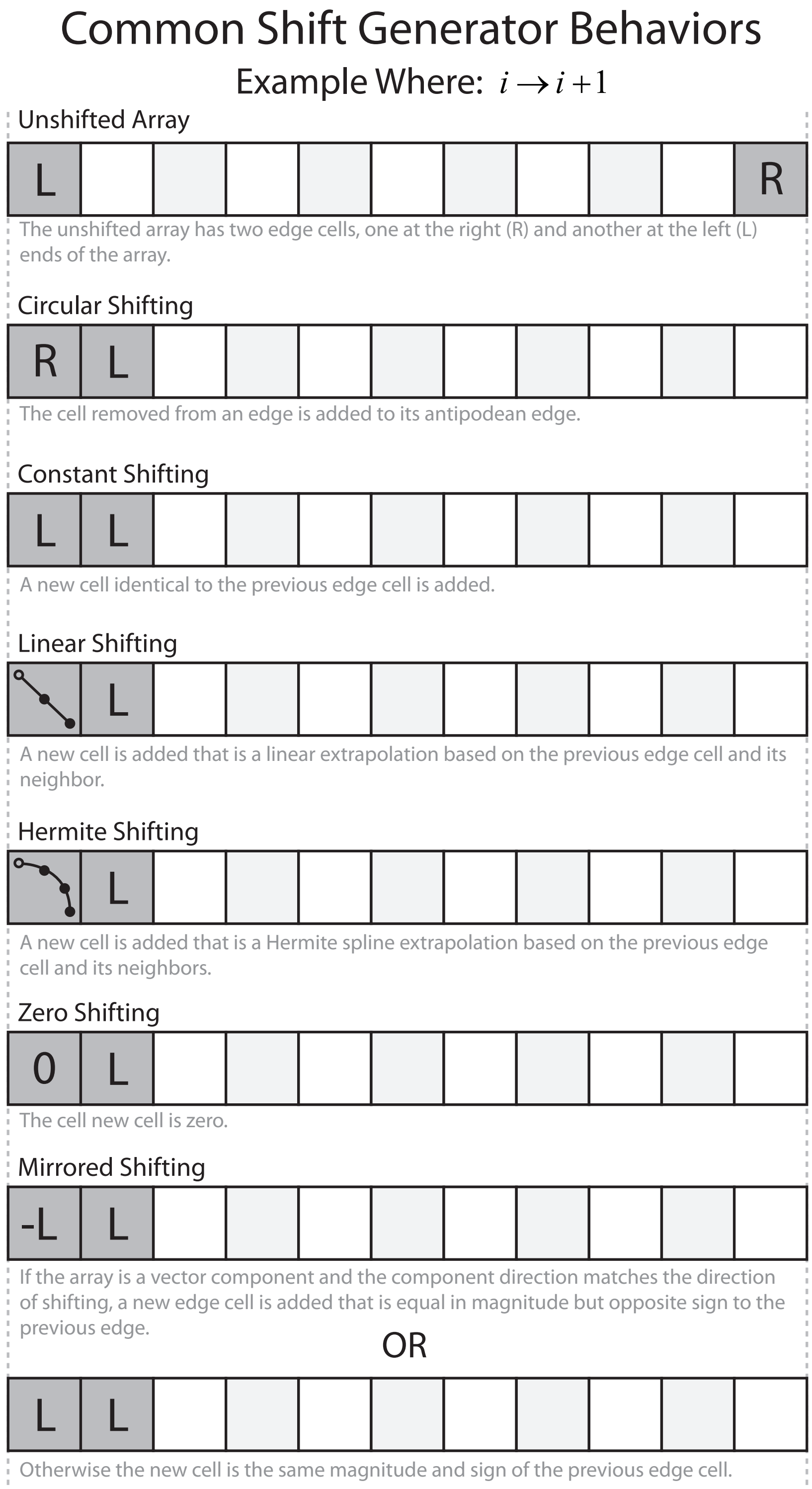
With the freedom to assign any value to the dynamically created cells during shifting, the functions and procedures of cell creation are responsible for defining the boundary type. These cell creation methods, henceforth called Shift Generators (SG), may be fully customized to fit a particular simulation, without modification to the flux routines. IMOGEN provides its shift generators with even more flexibility by providing them with the unique identifier for the array they will be shifting, the vector or scalar nature of that array, and the classification for that array, e.g. primary, flux, or secondary variable array types. This information makes it possible to adapt the cell generation process contextually, enabling even more refined and sophisticated boundary condition types.

4.3.1. Common Boundary Types

There are a number of boundary types used in commonly in astrophysics simulations, such as the circular, or tiled, boundary type previously describe where one spatial domain edge acts as the cell generator for its antipodean edge. IMOGEN includes shift generators for many of these common types, including the circular boundary case, which were used either directly in the corrugation instability investigation or as part of the testing process to verify that IMOGEN was functioning correctly (see Chapter 5 for the details of testing IMOGEN). These cases are enumerated in Figure 4.3 along with an example of shifting behavior.

4.3.2. The Fade Shift Generator

The investigation of the corrugation instability required a new type of shift generator to correctly handle the internal and external shock boundaries of the spatial domain. On both boundaries the accretion process is persistent for the life of the simulation, which requires stable, steady inflow and outflow conditions on the boundaries. At the same time magnetic waves, generated by the instability in the shock front, propagate out of both boundaries, and the deformation of the shock front creates density columns that grow into and beyond the internal boundary. Consequently, these boundary conditions had to simultaneously satisfy a constant inflow condition and a transparent outflow condition. This is a



The unshifted array has two edge cells, one at the right (R) and another at the left (L) ends of the array.

The cell removed from an edge is added to its antipodean edge.

A new cell identical to the previous edge cell is added.

A new cell is added that is a linear extrapolation based on the previous edge cell and its neighbor.

A new cell is added that is a Hermite spline extrapolation based on the previous edge cell and its neighbors.

The cell new cell is zero.

If the array is a vector component and the component direction matches the direction of shifting, a new edge cell is added that is equal in magnitude but opposite sign to the previous edge.

Otherwise the new cell is the same magnitude and sign of the previous edge cell.

Figure 4.3: Examples of shift generators for common boundary types.

difficult proposition given that outflow behaviors are unknown *a priori* and change over time.

For boundaries such as this Non-Reflective Artificial Boundary Conditions (NRABCs) are typically employed. In an NRABC the solution is extrapolated over the spatial domain boundary to attempt a rough approximation of the correct solution had the spatial domain continued beyond the boundary. These techniques are anisotropic, and as their namesake suggests, prevent non-physical reflections from occurring on the boundary. However, the non-reflective conditions are indiscriminate. Not only will they prevent non-physical reflections, they will prevent all reflections. This can cause problems if physical reflective behaviors happen on or near the boundary and must be carefully avoided as much as possible.

The two general approaches to implementing NRABCs are functional and physical. In the functional form, the extrapolation of the new boundary is generated by multi-dimensional interpolation of the existing bounding cells to produce a smooth approximation. The physical approach, which is more computational expensive, is to solve a simplified version of the advection equations, usually linearized, that are solved under conditions that prevent reflections. In either approach the results are approximate, which can lead to problems.

For the corrugation instability problem a number of NRABC methods were implemented and tested. In many cases these NRABC methods were able to prevent reflections from occurring in the internal, downstream boundary where the comparatively high densities and pressures supported a stable boundary. However, none of the techniques were successfully able to prevent reflections from occurring on the upstream, external boundary. The problem with the external boundary was that the combination of strong magnetic field strength of the inflow and the high frequency propagating magnetic wave outflow were never well resolved by the tested NRABC algorithms. In tests the NRABC solver would generate small, but significant, errors in the extrapolated solution of the magnetic field, resulting in errors in the divergence-free constraint along the boundary. In the subsequent magnetic advection step, the constrained transport scheme, which assumes a divergence-free magnetic field, would produce erroneous magnetic field strengths at the boundaries in response to these errors. This would in turn force the NRABC toward a larger correction during for the following step and in turn produce even greater deviations in the divergence-free constraint. The resulting feedback loop between the NRABC and the constrained transport advection scheme would quickly produce a large amplitude, high frequency inflow magnetic wave that would propagate into the spatial domain, destroying the simulation.

A number of attempts were made to try to stabilize the existing algorithms by either post-cleanup steps to the extrapolated conditions or to simultaneously satisfy the NRABC and divergence-free constraint. There were many improvements in these attempts, but nothing successful enough to be used in the actual simulations. The attempts finally culminated in an overly demanding NRABC solver that used too many computational resources to be worth pursuing further.

Instead a different approach was employed and a new NRABC algorithm developed that was much more aggressive in its NRABC condition. To create boundaries that would behave under these extreme conditions extrapolation was abandoned and the boundary converted from an edge condition into a multi-cell region a few cells thick along the border. By handling the boundary within the domain, the dynamic edge shifting is able to remove any outflows that would cause reflections while the still under the full advection scheme.

The technique, called Fade shifting, maintains a non-uniform, spatial varying superposition of the inflow condition with the simulated solution within the fade bounding region. The cells on the downstream edge of the boundary region are unaffected by the fade shifter. On the upstream boundary, however, the fade shifter imposes full the inflow condition and ignores any advection value assignments. Between these endpoints cubic Hermite interpolation is used to specify the mix between the full advection solution and the inflow condition. Figure 4.4 illustrates how the interpolation behaves in a fade boundary region.

As the namesake suggests, the effect of progressively mixing between the advection solution and the inflow condition is the fading away of any outflow behaviors before they hit the boundary to cause unwanted reflections. The fade NRABC takes maximum advantage of dynamic edge shifting, mixing the inflow conditions with the advection solutions for all spatial arrays, both primary and fluxes. The dynamic nature of

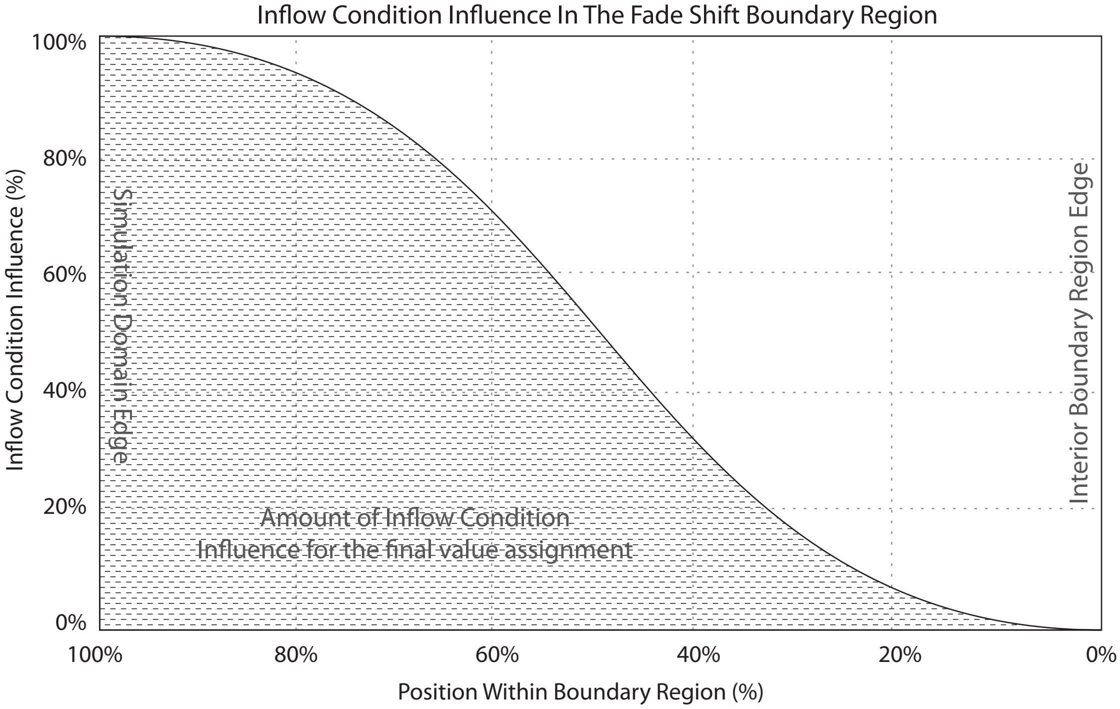


Figure 4.4: Influence profile for the fade shift NRABC boundary region. The inflow condition increases in influence for the superposition of the two value assignments closer to the edge of the spatial domain. With a maximum influence at the spatial domain boundary no outflow exists on the edge to be reflected.

the dynamic edge shifting technique is crucial to this approach because successive derivatives are calculated using the faded values of their constituent spatial arrays, e.g. the mass density flux is calculated from a faded mass density array. This effectively compounds the strength of fading for higher order derivatives, which is where the numerical instabilities in the NRABCs form.

The result is an extremely aggressive, and very stable NRABC for use in the corrugation instability simulation. It should be noted, however, that since the fade shifter was created it has been used as part of IMOGEN in support of many other simulation types. The inflow condition is not specified as part of the fade shifter, it is inferred from the initial conditions of the simulation, which makes it useful even in cases where the edge condition may not be an inflow at all. Hence, the fade shifter is a good general purpose NRABC boundary type for any case where outflow activity could be a problem for the evolution of the simulation. Demonstration and verification of the fade shifter can be found in the following chapter.

4.4. Active Grid Alignment

As will be discussed in chapter 6, all corrugation shock simulations place the shock front at the center of the spatial domain with initial conditions on the primitive variables such that the front remains stationary in the prescribed state. As the corrugation instability grows and the nonlinear behaviors take over, these conditions are broken and the shock front can propagate away from the center of the domain. Given the need to follow the evolution of corrugated shock fronts well into their nonlinear regimes, a mechanism was needed to keep the fronts from propagating out of the domain.

To do this, IMOGEN was extended to include an Active Grid Alignment (AGA) system. This AGA system works by locating a key position within the simulation domain and remapping the grid to keep that position at a specified place within the domain. For the corrugation instability, the key position was the plane of the shock front, which the AGA was set to find by recognizing the largest planar-averaged, shock-normal compression given as,

$$\Upsilon_i = \frac{1}{N_y \cdot N_z} \sum_j \sum_k \left| \partial_x v_{x,(i,j,k)} \right| , \qquad (4.X)$$

where N_y and N_z are the number of cells in the Y and Z dimensions of the grid and j and k are the indices over the cells in those dimensions respectively. When a corrugated shock front moved away from its initial position at the center of the grid the AGA would see the index of the maximum value of the Υ_i array shift by the same number of cells, which triggers the remapping process.

To remap the grid the AGA relies on shift generators. After a flux and source operation the AGA checks the Υ_i array and determines how many cells it needs to adjust the grid to re-center the corrugated shock front in the spatial domain. The AGA then shifts all of the primitive variable arrays by the specified number of cells. Since the shift generators act identically in this case to their use inside the flux routines, this operation is as stable as a flux action.

The only issue with the approach is for cases where the number of cells needed to remap is large compared to the total size of the grid. When this happens it means a large amount of information contained within the simulated domain was lost to the propagation. However, dramatic remapping such as this rarely occurred in the corrugation instability investigation, and for the few instances where it did happen the accretion shock fronts were already nearing the end of their evolution and aborting the simulation prematurely was not an issue.

CHAPTER V

TESTING AND VERIFICATION

With the development of any simulation software extensive testing is required before it can be put to research use. The software must be verified under a number of different test conditions to determine that is functioning properly as well as understand its capabilities and limitations.

Testing IMOGEN consisted of running through a suite of test problems, each one probing different aspects of its operation, and then comparing the results of each test with analytical solutions, in the few cases that such solutions exist, or, more commonly, numerical solutions from previous magnetohydrodynamics codes that have already been rigorously tested. The test process was broken down into two phases and corresponding tightly with the development process, first hydrodynamics and then magnetohydrodynamics. Before adding magnetic advection to the magnetohydrodynamic simulation software it is beneficial to have a fully functioning and tested hydrodynamic advection algorithm as a foundation. And since the algorithms used to advect hydrodynamic variables differ significantly from the magnetic field advection algorithm, magnetic flux is handled separately from hydrodynamic flux, hydrodynamic testing prior to any magnetohydrodynamics testing is a reasonable first step in the verification process.

The following are three of the many tests conducted on IMOGEN that highlight its capability as an effective simulation tool in the investigation of the corrugation shock wave problem. The first, the Sod shock tube, is a canonical test of hydrodynamic shocks and demonstrates the shock capturing capabilities of the second-order relaxed TVD MUSCL scheme. The second test, the Brio-Wu shock tube, is the magnetic extension of the Sod shock tube, demonstrating the addition of constrained transport magnetic advection in a shock wave system. The final test, the Orszag-Tang vortex, is the canonical multi-dimensional magnetohydrodynamics test problem as it taxes the fully magnetic advection scheme and is highly unstable to numerical errors. For each of the two shock tube tests a modified form of the test was conducted using fade edge shifting to demonstrates the effectiveness of the fade shifting described in section 4.3.2 in support of the corrugation instability investigation.

5.1. Sod Shock Tube

The Sod shock tube is an extension to a hydrodynamic system of equations of the more general Riemann problem, which describes the solution to the generic advection equation (2.1) given an initial condition of two regions of differing value separated by a step discontinuity. The Sod shock tube is traditionally a one-dimensional problem, hence the tube descriptor, with contact discontinuities in the mass and pressure densities between two otherwise homogenous regions. At time zero the imaginary barrier responsible for setting up the discontinuity between the two regions is removed, causing system evolution due to interaction across the discontinuity (Sod, 1978).

Given the right choices for the initial conditions, the discontinuity is responsible for creating a supersonic shock wave that propagates through the tube. The general arrangement of the initial conditions is shown in Figure 5.1(a), which according to the conditions set forth by Sod when he developed the problem are

$$\begin{aligned}&\rho_1 = 1 \quad \rho_2 = 0.125 \quad P_1 = 1 \quad P_2 = 0.1\\&v_{\|,1} = 0 \quad v_{\|,2} = 0 \quad v_{\perp,1} = 0 \quad v_{\perp,2} = 0\\&\gamma = 1.4\end{aligned} \quad . \tag{5.1}$$

Here the 1 and 2 subscripts denote the two separate initial regions on either side of the imaginary barrier. Under these conditions the propagation of the shock wave sets up five distinct regions within the tube, two representing the pre and post rarefaction wave propagation, one created by the initial contact discontinuity, two pre and post the shock front discontinuity as shown in the diagram of Figure 5.1(b).

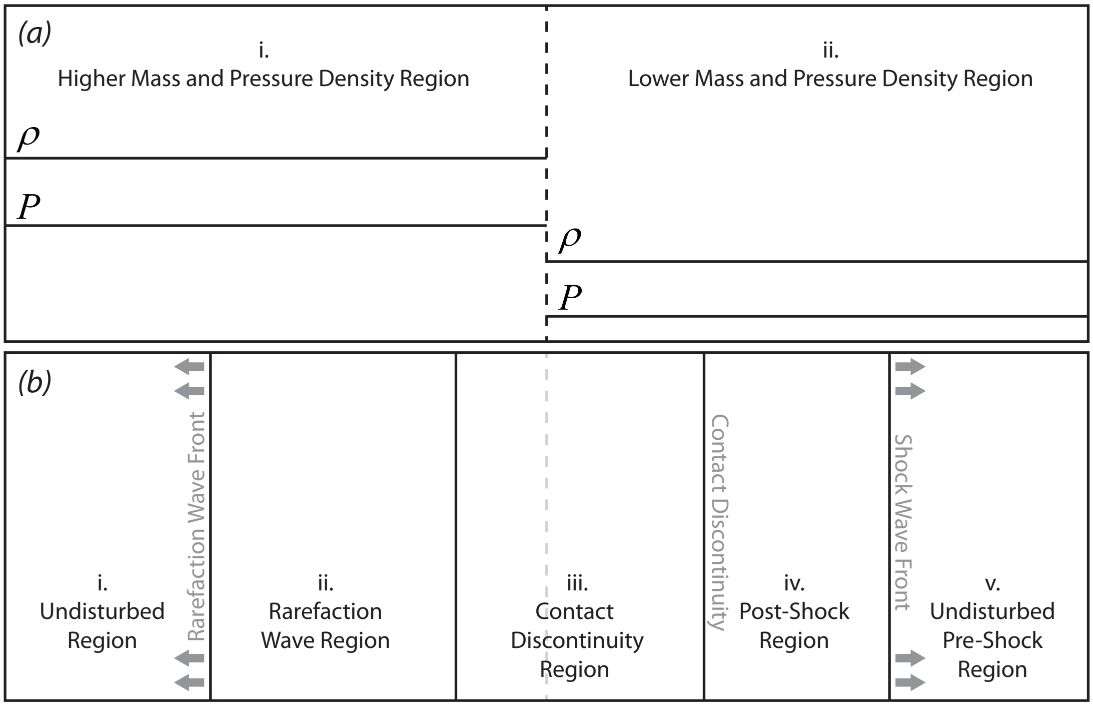


Figure 5.1: Diagram of the Sod shock tube test, (a) with initial conditions for two distinct regions, and (b) an finite time after the barrier has been removed and the system allowed to evolve into five distinct regions.

The Sod shock problem is widely used to verify shock and discontinuity handling in hydrodynamic advection algorithms. A large part of its value comes from the fact that it is also one of the only test problems in hydrodynamics shock simulations that is both rigorous and has an analytical solution for direct comparison (Fryxell, et al., 2000; Ryu & Jones, 1995; Sod, 1978).

The Sod shock tube test was carried on IMOGEN using a simulation specific Initializer class created to generate the appropriate initial conditions for a shock tube of fixed unit length but of arbitrary grid cell resolution, as well as a number of extensions to the problem discussed later. Initial testing was run on tubes with resolutions of 256, 512, 1024, and 2048 grid cells and a detailed comparison made of the different trials in an attempt to find any resolution dependent artifacts that would indicate scaling related problems with advection algorithm.

The only appreciable differences found during these comparisons were the absolute width of the shock discontinuities captured as shown in Figure 5.2. Higher resolution trials better defined the discontinuity in the mass density profile, a behavior that was expected as discontinuity capture in advection algorithms is limited by both a minimum number of grid cells and an absolute spatial domain width. As the resolution increased a similar number of grid cells were able to reproduce a better defined discontinuity, but with diminishing returns as the spatial width eventually begins to dominate. Comparing the decrease in discontinuity width between the 256 and 512 resolution runs and the 1024 and 2048 runs, it is apparent that the number of cells is the limiting factor for lower resolutions and decreases in importance as the resolution increases. The spatial width limitation is produced by the flux limiting done as part of the total variation diminishing advection scheme and can be somewhat controlled by the choice of the flux limiter, which in this case was Van Leer. Using a Min-Mod limiter, for example, would produce a much smaller spatial width at higher resolutions.

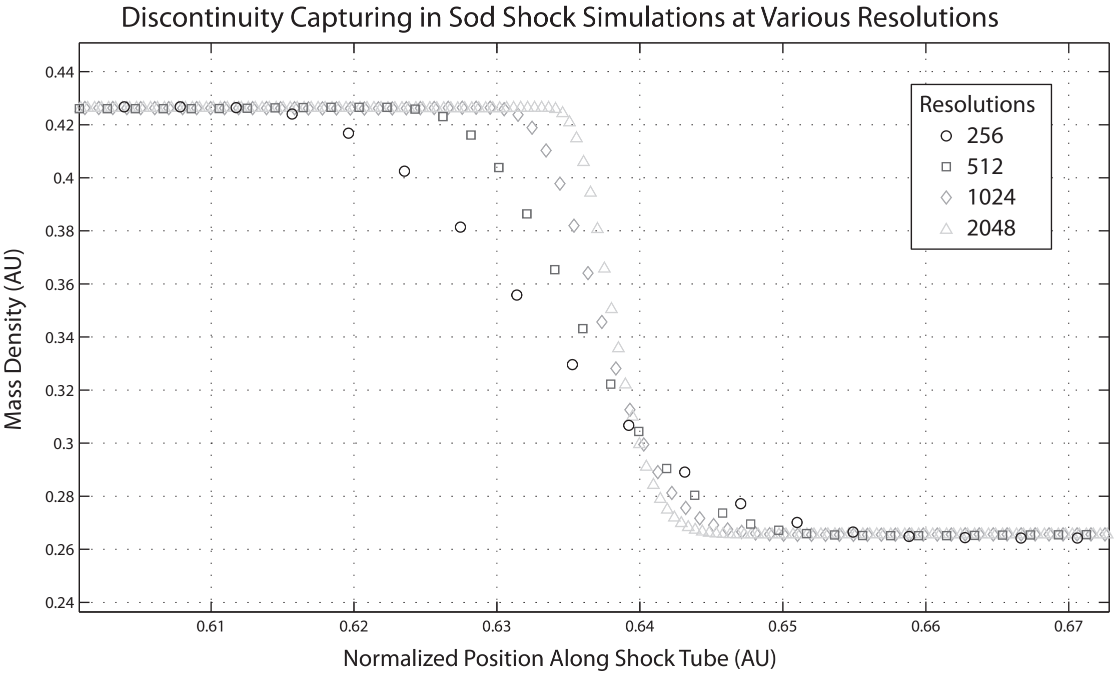


Figure 5.2: Simulated discontinuity results of a Sod shock tube test at various resolutions between 256 and 2048 grid cells. The higher the resolution the sharper the discontinuity in absolute units but the same number of grid cells are required to actually resolve the discontinuity.

Satisfied that IMOGEN was capable of resolving discontinuities on a range of commonly used resolutions, the 512 grid cell resolution was selected for the remainder of Sod shock tube testing because it represented an intermediate resolution likely to be used in research applications such as the corrugation instability problem.

The next step in the Sod shock tube test process was to examine simulated results against their corresponding analytical solutions. Figure 5.3 shows a typical comparison between the simulated and analytical mass density profiles for a 512 resolution run at a simulation time of 0.15. The mass density profile shows five distinct regions, as outlined in Figure 5.1, beginning with the unperturbed high density region on the left. To the right of that is the expanding rarefaction wave region, which propagates into the high density region. This is followed by the flat contact discontinuity region the expands both leftward and rightward over time, with the actual shock front region propagating a faster rightward velocity into the unperturbed low density region.

A visual inspection of the two profiles in Figure 5.3 reveals that the advection algorithm is functioning reliably for a hydrodynamic shock with little disagreement between the two solutions except for the grid cells located on and around the discontinuities. This is expected as advection algorithms require multiple cells to resolve transitions as previously investigated, and from the behavior in the discontinuity areas it is clear that the reasons for each of the discrepancies were finite transitions between the pre and post discontinuity values.

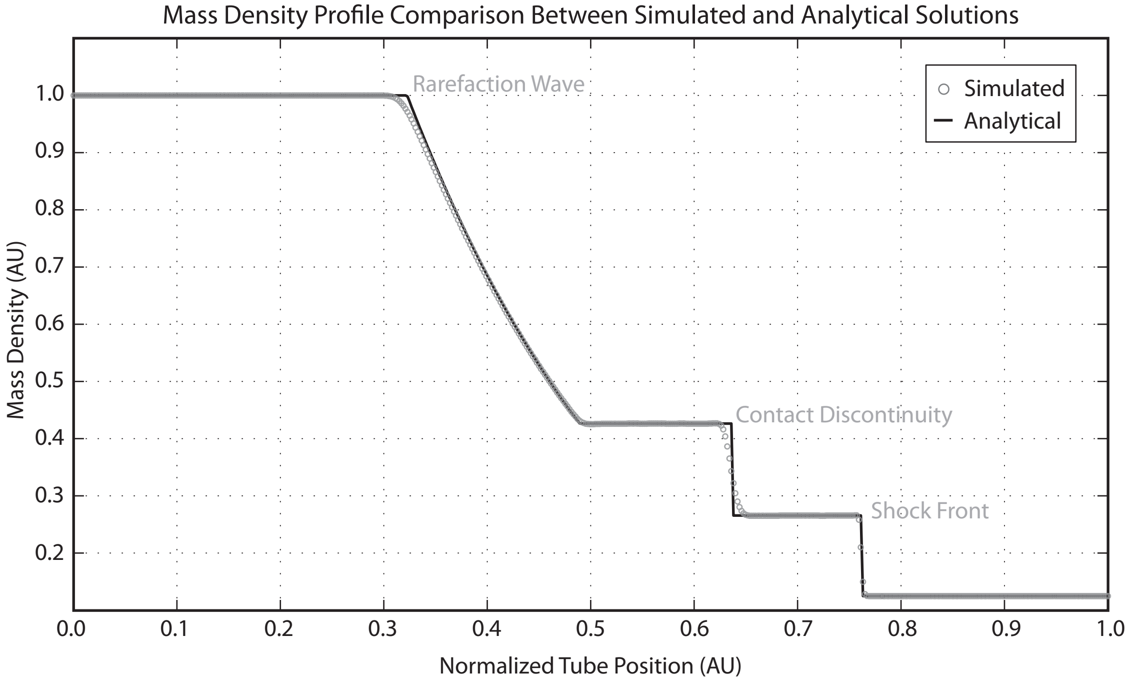


Figure 5.3: Profile of the mass density in a Sod shock tube simulation after appreciable simulation time as well as the analytical solution for comparison.

Plotting the fractional error between the simulated and analytical mass density profiles revealed the disagreement in more detail as shown in Figure 5.4. The fractional error peaks due to the shock front and the contact discontinuity are significant, but explained by the transition of the value across the respective discontinuities with the maximum fractional error for these peaks determined by half the height of the discontinuity. The width of these peaks is the more interesting and encouraging result revealed by this plot. The fractional error drops rapidly on either side of the peak, and in only a few cells on either side of these peaks the fractional error has dropped down to or below a few percent difference, which is well within the acceptable error tolerances for a second-order accurate advection algorithm around an area of rapidly changing value.

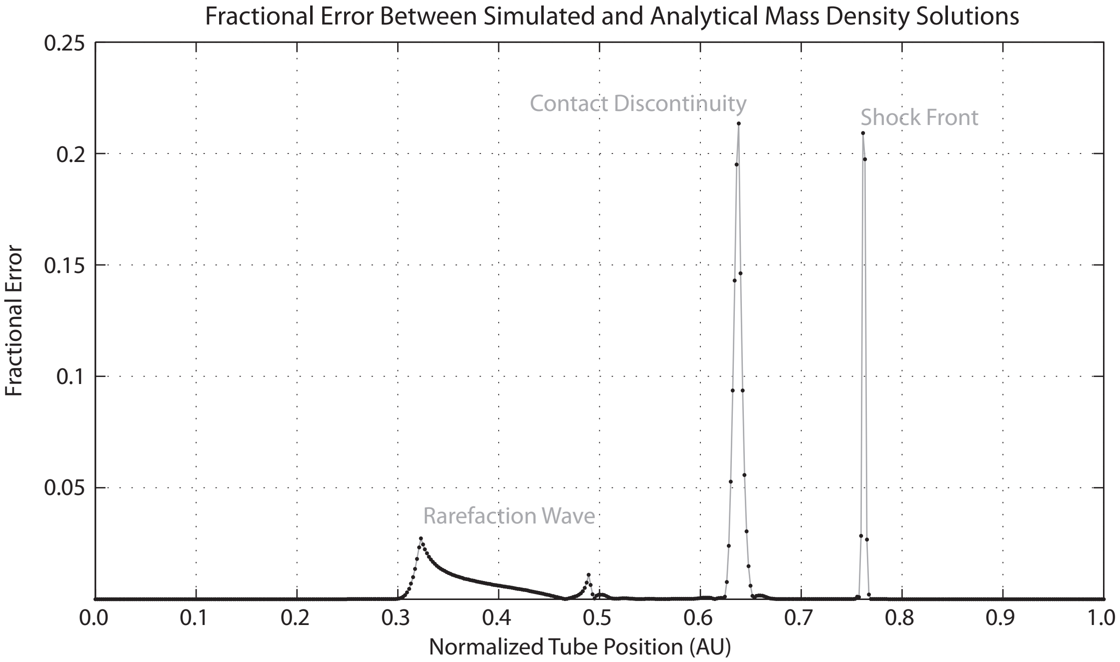


Figure 5.4: Fractional error between the simulated and analytical mass density profiles at a simulation time of 0.15.

Unlike the shock and contact discontinuities, the fractional error in the rarefaction region never peaks very high but has a much broader extent. This is a potential concern given the rarefaction wave discontinuity is small and the region smoothly transitions behind the rarefaction wave front, which should not produce large errors. However, given that the error never exceeds a few percent the discrepancy is also explained by the advection algorithm being only second-order accurate in areas of quick change. The larger question that Figure 5.4 suggests is does the IMOGEN advection algorithm recover from these errors over time? If the IMOGEN were functioning properly the errors would be transient and the, as the discontinuities propagate, areas through which discontinuities have passed should converge once again on the correct solution with in the insignificant error bounds limited by the advection algorithms order of accuracy. The wake behind the rarefaction wave in Figure 5.4 suggests that this is the case as the further from the wave front the smaller the error. However, to be absolutely sure a waterfall time plot is made of the fractional error in the simulated and analytical mass density profiles showing the evolution of the fractional error over time as shown in Figure 5.5.

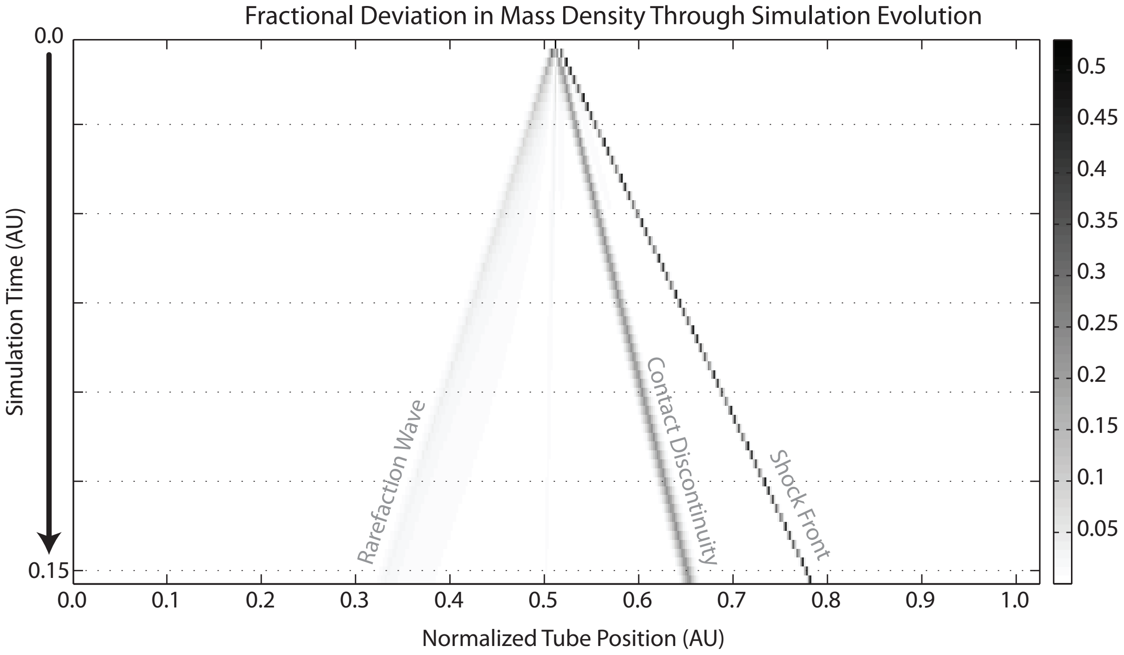


Figure 5.5: Waterfall evolution plot of the fractional error between the simulated and analytical solutions to the mass density profile in the Sod shock tube test.

Figure 5.5 displays the fractional error plot shown in Figure 5.4 as a horizontal line and colored darker where the fractional error is higher. Each horizontal slice represents a snapshot in simulation time of the mass density profiles. The slices are equally spaced in time beginning from the start of the simulation, time of 0, at the top until the end of the simulation, time of 0.15, at the bottom. Inspecting this plot it becomes obvious that the IMOGEN hydrodynamic advection algorithm recovers from errors created by discontinuities, as these errors clearly track the shock front and don't increase appreciably over time.

The same analysis shown here for the mass density plots was also made of the other conserved hydrodynamic variables in the system, all of which exhibited similar encouraging results. The details of each of these examinations is omitted, but examples of the velocity and total specific energy, total energy per unit mass density, profiles are included below in Figures 5.6 and 5.7 as a reference snapshot for illustration and documentation purposes.

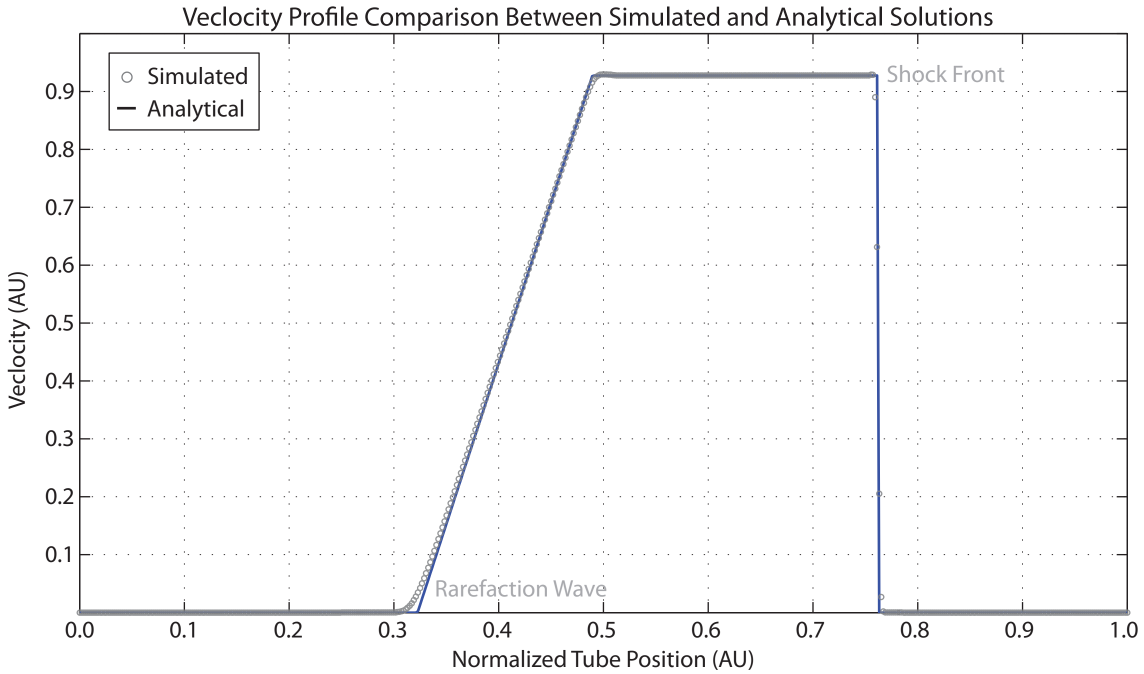


Figure 5.6: Simulated and analytical velocity profiles of the Sod shock tube problem at simulation time of 0.15.

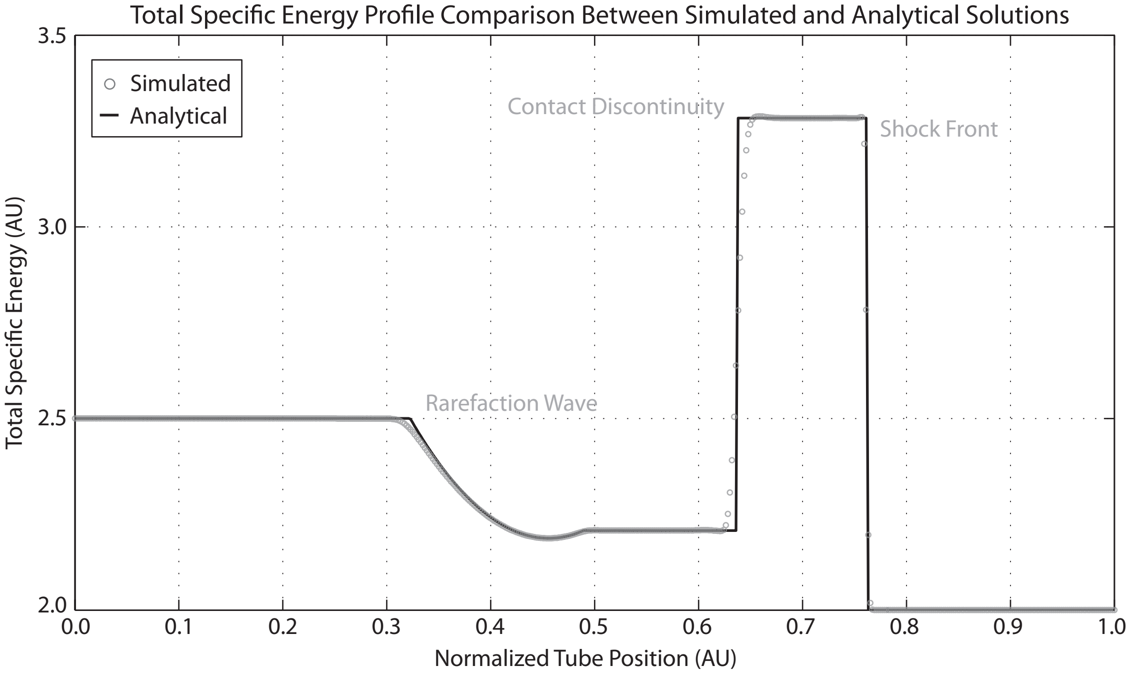


Figure 5.7: Simulated and analytical total specific energy density profiles for the Sod shock tube test at a time of 0.15. Specific energy is plotted instead of energy directly because this accentuates the energy variations in the contact and shock regions, providing a clearer understanding of the discontinuity capturing in the energy as a result.

In addition to the traditional Sod shock tube test, the IMOGEN Sod shock Initializer class was extended to allow for testing to take place on a multi-dimensional grid, along any choice of grid direction, and at arbitrary angles relative to any specified grid axis. The first set of these extended tests were reproductions of the one-dimensional tests discussed above on three-dimensional simulation domains to investigate the abilities of the advection algorithm to correctly maintain one-dimensional behavior in a three-dimensional environment. Aberrant behavior on the part of the advection algorithm during these tests would have resulted in a deviation from the analytical solution as a result of some incorrect spatial dependency created by the advection algorithm in dimensions parallel to the plane of the initial discontinuity. However, no such deviations were found and comparisons between the previous one-dimensional results and the multi-dimensional tests were identical. This supports the conclusion that the multi-dimensional advection algorithm capably and correctly handled the introduction of additional spatial degrees of freedom into the one-dimensional problem.

In the next phase of Sod shock tube tests, the one-dimensional and three-dimensional problems described above were repeated for tubes with propagation alignments along the remaining two available dimensions. In each case the results were compared to the respective analytical solution as well as the results of the original tests in search of any variation that would indicate errantly biased advection behavior along a specific axis. None of these tests revealed any deviation from the original results in either one or three dimensions, which leads one to the conclusion that the advection algorithm was, as desired, isotropic.

The final extension of the Sod shock tube is a two-dimensional test with the initial discontinuity existing at an angle to one of the axes for testing the ability of the advection algorithm to handle general non-axis aligned advection as well as discontinuities. Tests were conducted on a number of different alignment angles, but the one of most interest was 45° because it represents the most difficult advection direction given that two fluxing steps contribute equally for every complete advection step. Figure 5.8 shows the results of a 45° aligned two-dimensional test.

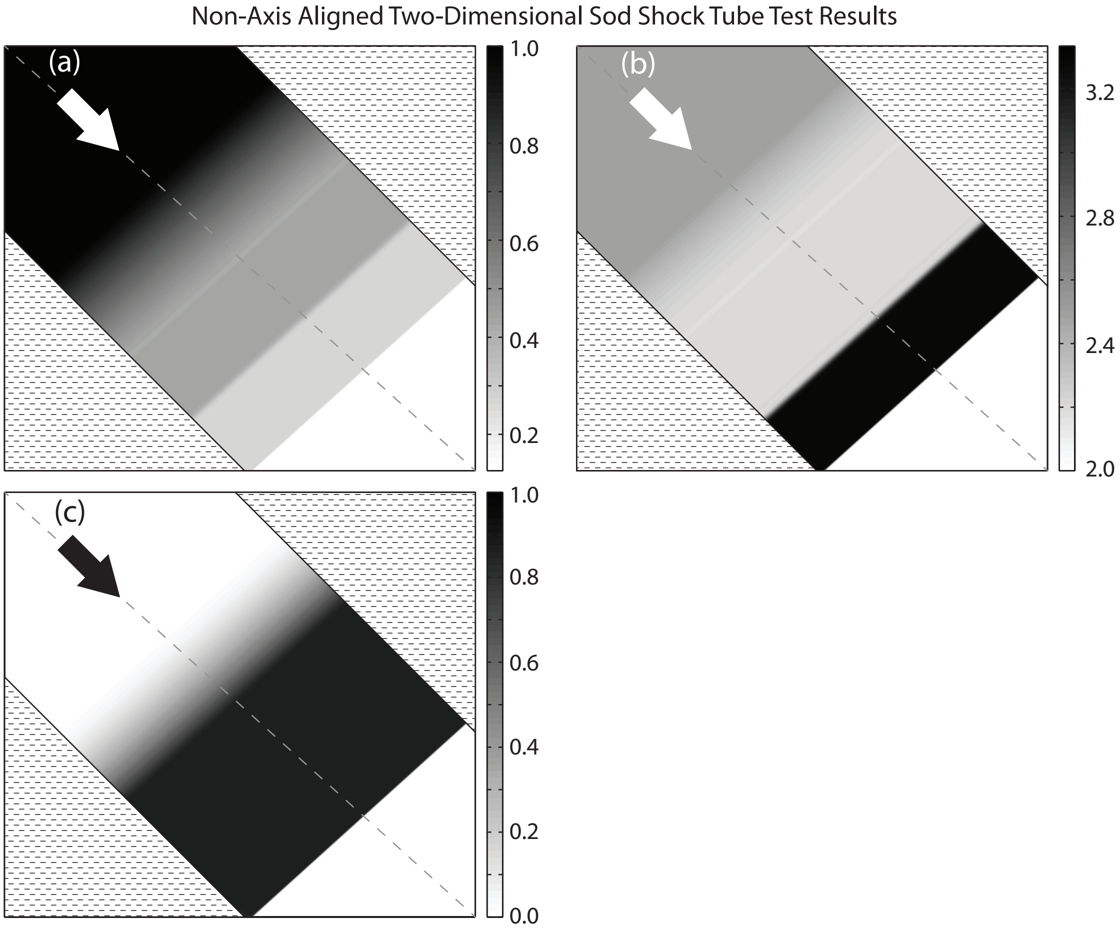


Figure 5.8: Results of a two-dimensional non-axis aligned Sod shock tube test with an angle of 45° from the primary axis. The panel (a) shows the mass density, (b) the total specific energy density, and (c) the velocity. The arrow in each panel indicates the direction along with the shock wave propagates during the simulation. The dotted line represents the center line of propagation with unit length.

The triangular areas blocked out in each panel of Figure 5.8 are areas where the shock wave had already struck the edge of the domain and reflected back and so were ignored because the result is no longer valid compared to previous tests and the analytical solution. The arrow in each panel indicates the direction of propagation for the shock wave, with the initial region i located in the upper left corner and the initial region ii in the lower right corner. The spatial domain was setup so that the largest length of a one-dimensional propagation profile, represented in Figure 5.8 by the dotted line, was unit length. This allowed for direct comparison with the previous axis aligned tests and the corresponding analytical solution. From the panels in Figure 5.8 it is obvious that the behavior is uniform perpendicular to the direction of propagation, meaning that even in the non-axis aligned cases the advection algorithm is capable of maintaining direction of propagation without erroneous behaviors caused by non-physical crosstalk between flux directions. To further investigate any potential errors caused by the off axis alignment the unit length profile, the dotted lines, of the mass density, velocity, and total specific energy were plotted against corresponding analytical solutions in the same fashion as the original one-dimensional test. The resulting plots are shown in Figure 5.9.

Inspection of Figure 5.9 shows no obvious errors introduced by the non-axis alignment. To be absolutely confident the fractional error plots were reproduced and compared to the one-dimensional trials. An example of the mass density fractional error profile is shown in Figure 5.10, which exhibits the same general form as the previous axis aligned trial shown in Figure 5.4 with only insignificant errors between the two plots.

In all cases the IMOGEN advection algorithm is able to reproduce the Sod shock tube result within reasonable error tolerances given the order of accuracy of the algorithm and under a wide range of important conditions.

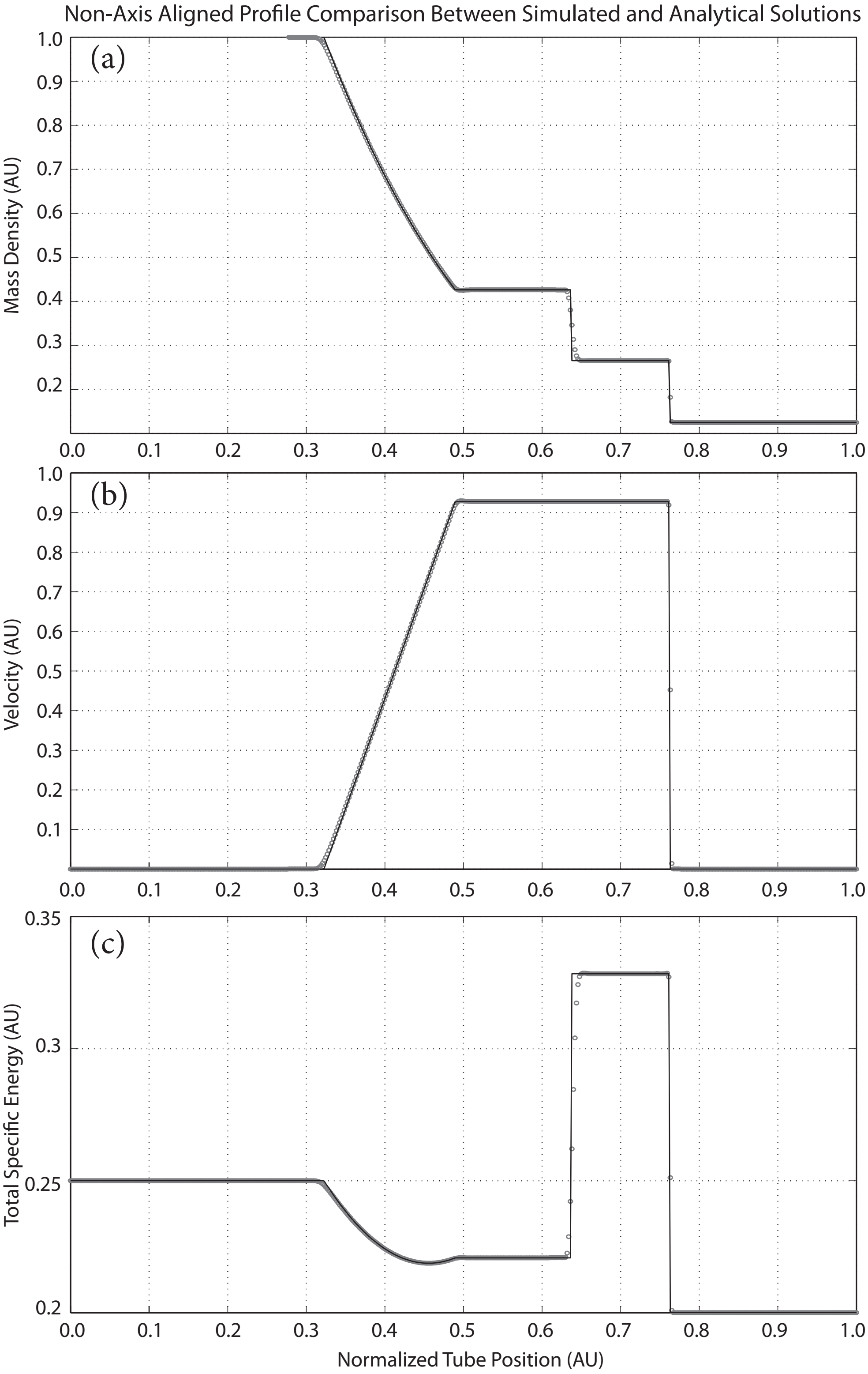


Figure 5.9: Profiles of mass density (a), velocity (b), and total specific energy (c), for the unit length profile of the non-axis aligned profiles illustrated by the dotted lines in Figure 5.9 and analytical solutions.

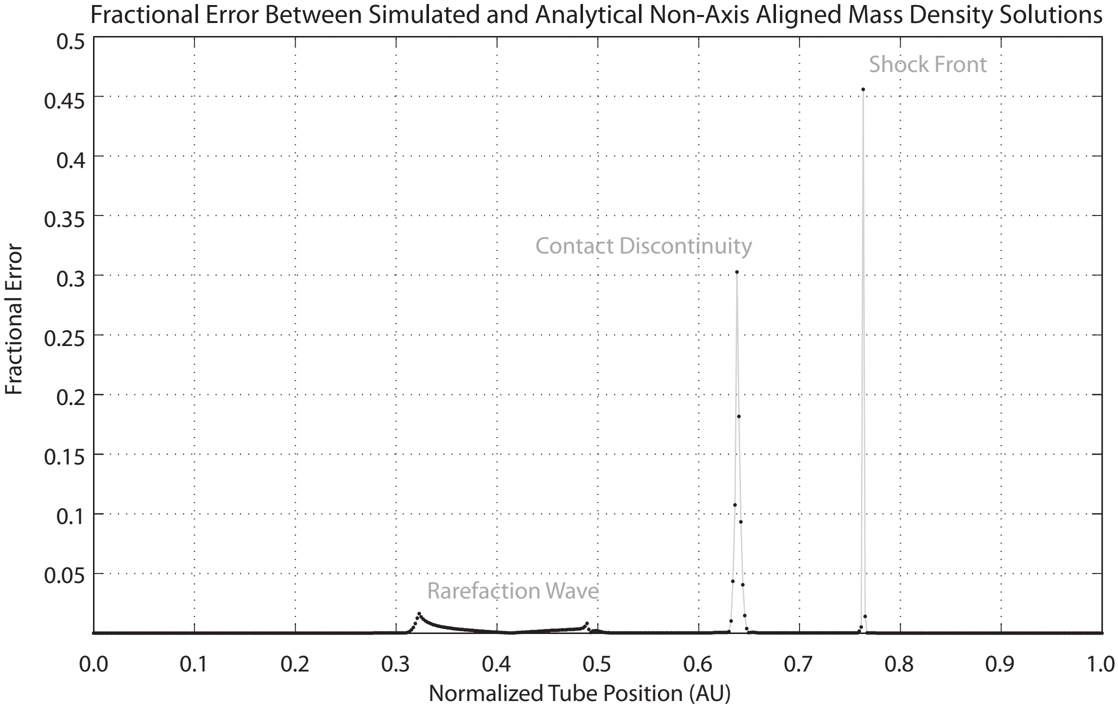


Figure 5.10: Fractional error plot between the simulated and analytical mass density profiles at simulation time of 0.15 for a non-axis aligned Sod shock tube test with a propagation angle of 45°.

As a final extension to the Sod shock tube test, the boundary conditions along the shock normal directions were changed from their default constant behaviors, as specified by the original problem, and replaced with fade shifting, see section 4.3.2 for details, to verify the effectiveness of the fade shifter to prevent reflections with a substantial shock wave acting as the outflow condition. In this case the Shock tube problem was run past the simulation time used above to give the shock front time to propagate off the edge of the grid, completely through the fade shift region. The results of this test are presented in figure 5.11. As the figure demonstrates, the fade shifter is able to successfully fade away the shock front without reflection.

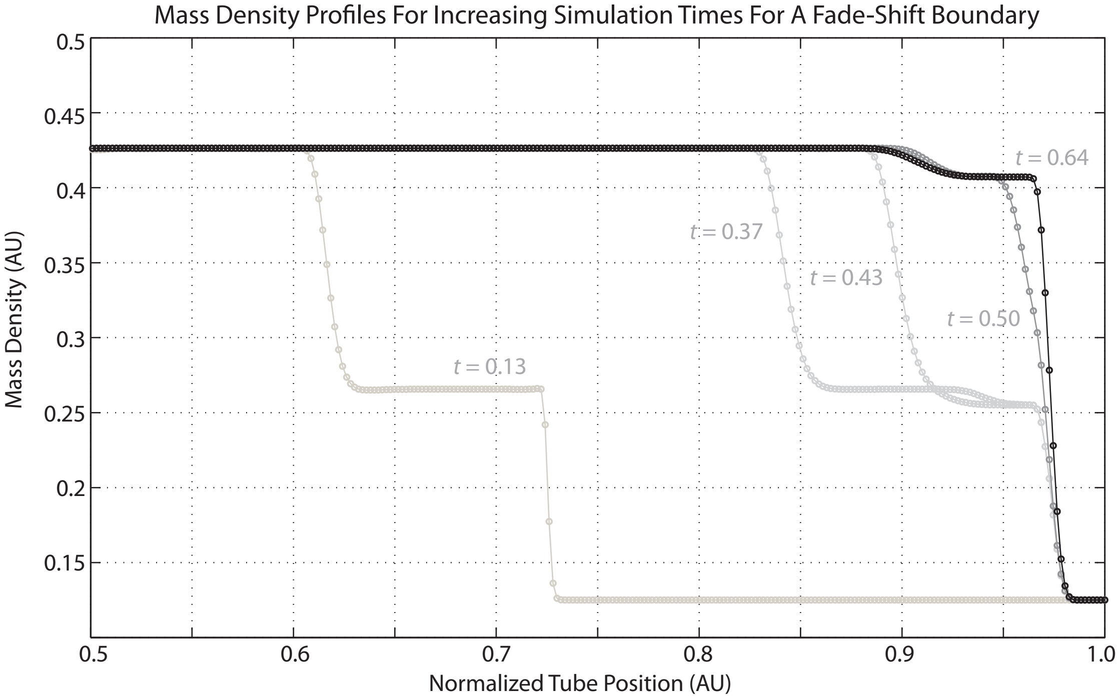


Figure 5.11: Sod shock tube problem with fade shift edges for the shock normal boundary conditions. The shock tube was evolved for a time long enough for the shock front to propagate off the edge of the domain.

None of the mass density profiles exhibit any reflective wave disrupting the solution outside of the fading region. By the *t* = 0.37 time units profile the shock front has entered the fade region and no reflection is observed by the discontinuity fading away on the boundary. The subsequent time profiles confirm that the process is without reflection. Obviously, the fade shifting has an effect on the boundary region but even at the t = 0.64 time units profile nothing has disturbed the smooth post-contact discontinuity profile at the center of the spatial domain. The fade shifter is, therefore, successfully able to transport the shock front out of the simulation domain without reflecting back into the domain for hydrodynamic shock waves. More verification of the fade shifting solver is reserved for section 6.1.1.

5.2. Brio-Wu Shock Tube

The Brio-Wu shock tube is a natural extension of the hydrodynamic Sod shock tube problem to magnetohydrodynamics. Similar in many ways to the Sod shock tube, the Brio-Wu shock tube begins with two distinct regions separated by an invisible barrier that is removed at the beginning of the simulation that allows the shock discontinuity to evolve over time. The magnetic extension for the Brio-Wu shock tube is the introduction of uniform magnetic fields parallel, perpendicular, or at an arbitrary angles to the front of the shock. The magnetic field changes the way in which the shock wave evolves as the magnetic field allows for additional propagation modes are greater in number and propagation speed diversity than the corresponding hydrodynamic system (Brio & Wu, 1985).

Unlike the Sod shock tube, the Brio-Wu shock tube has no analytical solution for a direct comparison. The solution is, however, simple enough that it can be compared with other published solutions of the problem as a consistency check. As each simulation software has slight variations in its solution profiles, there is little added value to a comparison as rigorous as the comparisons made previously for the Sod shock tube problem, especially with regard to the evolution of errors. Instead verification is achieved by comparing the locations and values of notable features between different results. The results of a Brio-Wu shock tube test using IMOGEN with the initial conditions,

$$\begin{aligned}&\rho_1 = 1 \quad \rho_2 = 1 \quad P_1 = 20 \quad P_2 = 1\\&v_{\|,1} = 10 \quad v_{\|,2} = -10 \quad \boldsymbol{v}_{\perp,1} = 0 \quad \boldsymbol{v}_{\perp,2} = 0\\&B_{\|,1\&2} = 5/\sqrt{4\pi} \quad \boldsymbol{B}_{\perp,1\&2} = \left(5/\sqrt{4\pi},0\right)\\&\gamma = 5/3\end{aligned} \quad , \tag{5.2}$$

where subscripts 1 and 2 represent the two distinct initial regions, are shown in Figures 5.12-5.14. This trial was run on a three-dimensional grid with the shock normal along the *x* direction. The resolution of the spatial domain was 1024 cells along the *x* direction and 32 cells along each of the *y* and *z* directions. Like the Sod shock tube problem, the Brio-Wu magnetic shock tube is a one-dimensional advection problem, but running it on a three-dimensional grid allows for an inspection of the magnetic advection algorithm to correctly maintain the uniform, one-dimensional nature of the problem.

A qualitative, feature-by-feature comparison between IMOGEN and the results of three other simulation tools exhibit no significant disagreement from the results presented by others under the same initial conditions (Gardiner, et al., 2008; Pen, et al., 2003; Ryu et al. 2008; Toth 2000).

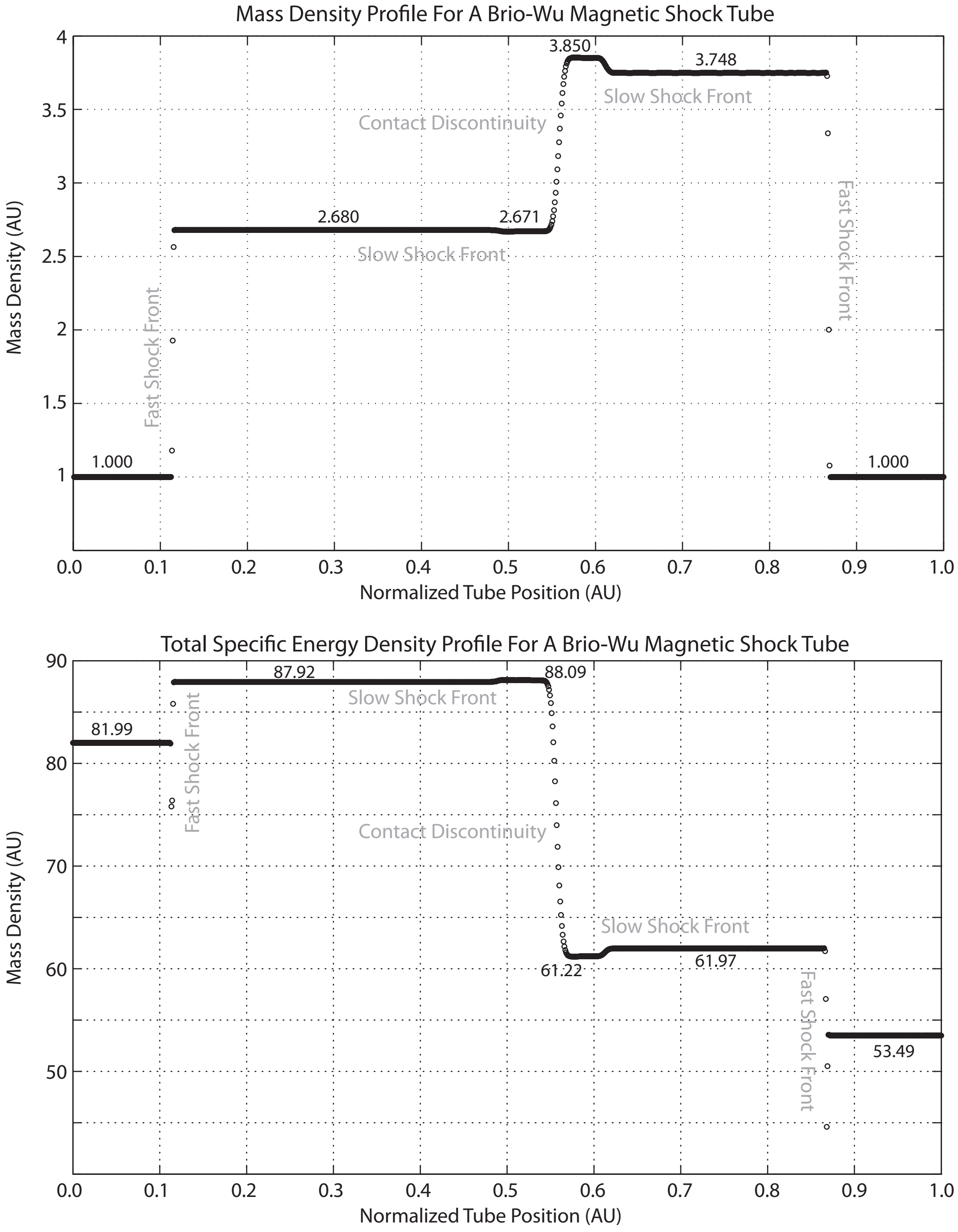


Figure 5.12: Mass density and total specific energy results of the Brio-Wu magnetic shock tube.

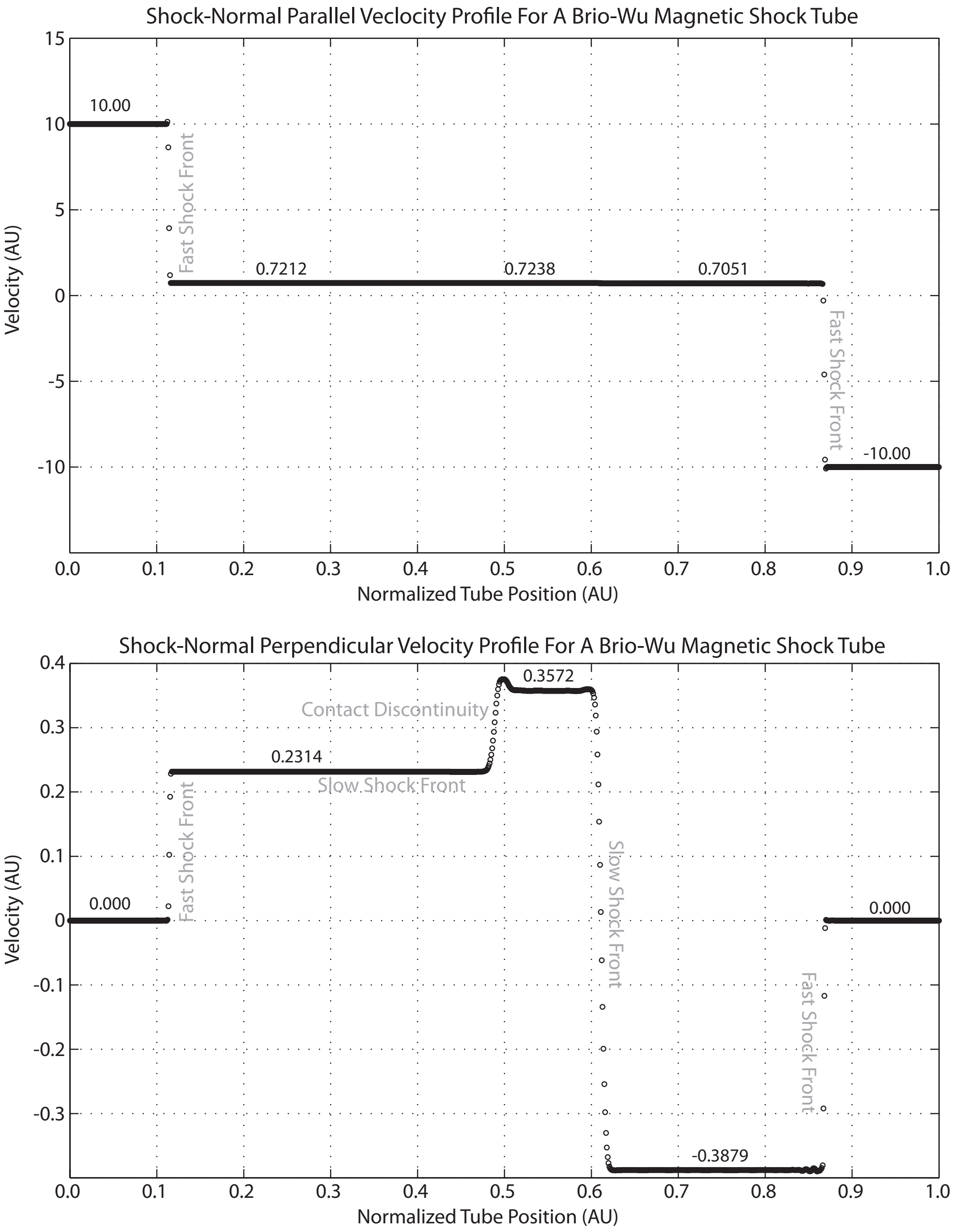


Figure 5.13: Shock plane parallel and perpendicular velocity results of the Brio-Wu magnetic shock tube.

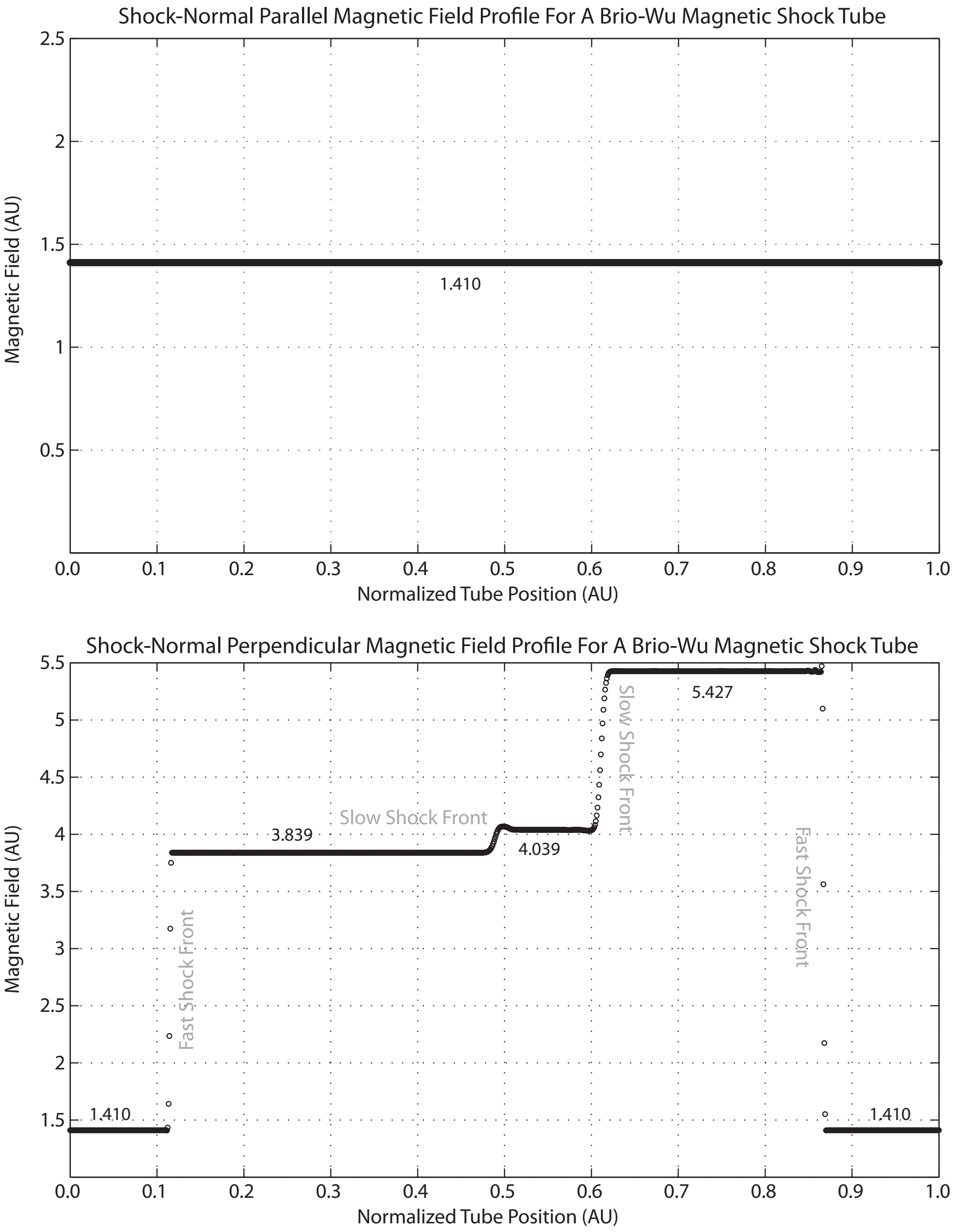


Figure 5.14: Shock plane parallel and perpendicular magnetic field results of the Brio-Wu magnetic shock tube.

A calculation of the divergence of the magnetic field for the magnetic field results shown in Figure 5.14 was negligible, with values no higher than the unavoidable machine precision noise that arises from floating point arithmetic. The qualitative comparison included a number of features shown in Figures 5.12-5.14. The most important are summarized as follows:

- Fast shock fronts: $x \approx 0.1$ and $x \approx 0.85$.
- Slow shock fronts: $x \approx 0.5$ and $x \approx 0.6$.
- Contact discontinuity: $x \approx 0.55$.
- Approximate values within distinct profile regions (in order from left to right):
    - *Mass density:* 1.000, 2.680, 2.671, 3.850, 3.748, and 1.00.
    - *Total specific energy density:* 81.99, 87.92, 88.09, 61.22, 61.97, and 53.49.
    - *Velocity (shock normal parallel):* 10.00, 0.7212, 0.7238, 0.7051, and -10.00.
    - *Velocity (shock normal perpendicular):* 0.000, 0.2314, 0.3572, -0.3879, and 0.000.
    - *Magnetic field (shock normal perpendicular):* 1.410, 3.839, 4.039, 5.427, and 1.410.

An inspection of the shock-plane variance also revealed that the simulation correctly maintained its one-dimensional, axis-aligned flow. The success of the qualitative comparison between IMOGEN and three other published solutions, along with verification of the constrained transport magnetic advection scheme maintaining the divergence-free constraint, clearly demonstrate the capability of IMOGEN to handle magnetohydrodynamic advection in the presence of shock waves.

5.3. Orszag-Tang Vortex

The Orszag-Tang vortex is considered one of the more critical tests for a magnetohydrodynamics simulation software. From somewhat simple initial conditions, the Orszag-Tang vortex evolves into a very complex end state due to complex interactions between the many supersonic magnetic waves that propagate within the simulation domain (Orszag & Tang 1979). The initial conditions for the simulation are,

$$\begin{aligned}
&\rho = 25/(36\pi) \quad P = 5/(12\pi) \\
&\boldsymbol{v} = \left(-\sin(2\pi y),\ \sin(2\pi x),\ 0\right) \\
&\boldsymbol{B} = \left(-B_0 \sin(2\pi y),\ B_0 \sin(4\pi x),\ 0\right) \\
&B_0 = 1/\sqrt{4\pi} \quad .
\end{aligned} \tag{5.3}$$

For this arrangement a value of $\gamma=5/3$ is used as the ratio of specific heats in the equation of state. The domain is two-dimensional and uses circular (periodic) boundary conditions for all edges. The domain is normalized to unit length for both x and y dimensions so that the problem can be run and compared at different spatial resolutions. The initial configuration for the test problem is shown in Figure 5.15 for a 256×256 cell spatial domain.

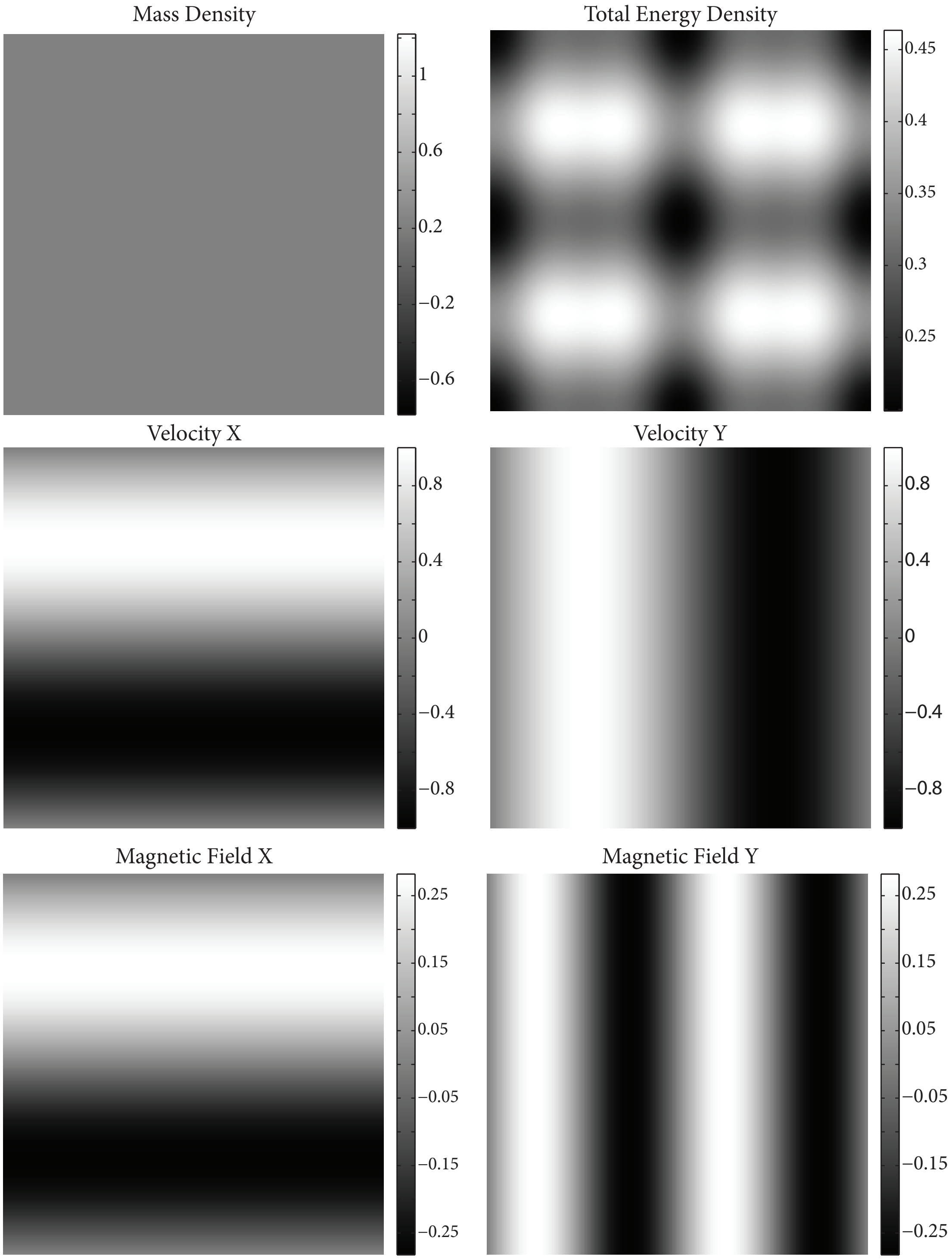


Figure 5.15: Initial conditions for the Orszag-Tang vortex test as specified by (5.3). The periodic behaviors of the velocity and magnetic field components are responsible for driving the evolution of the simulation.

For these initial conditions the simulation begins in a mixed state of subsonic, transonic, and supersonic Mach velocities as shown in Figure 5.16. During the simulation the velocities will increase as the magnetic effects drive the evolution. Comparing Figure 5.16 with the velocity conditions shown in Figure 5.15 it is apparent the y velocity is largely responsible for the initially supersonic regions.

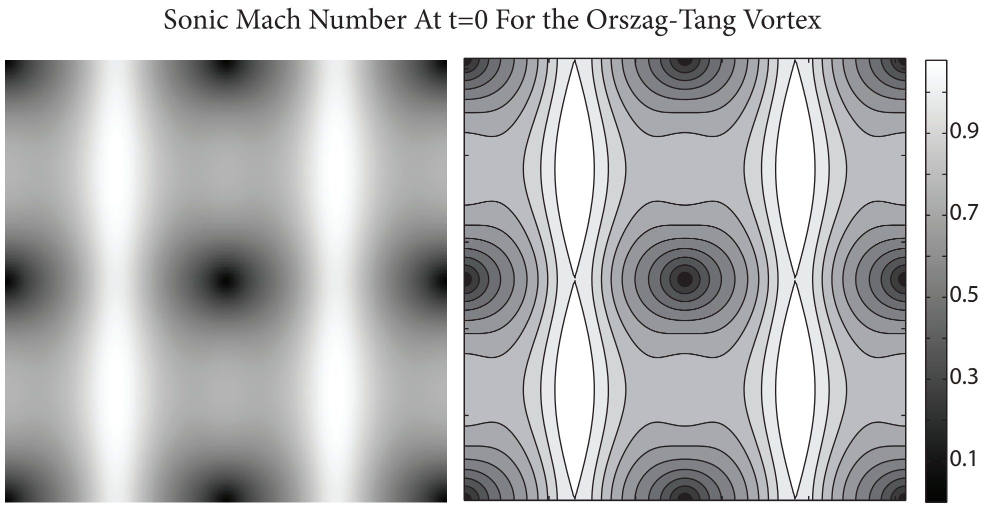
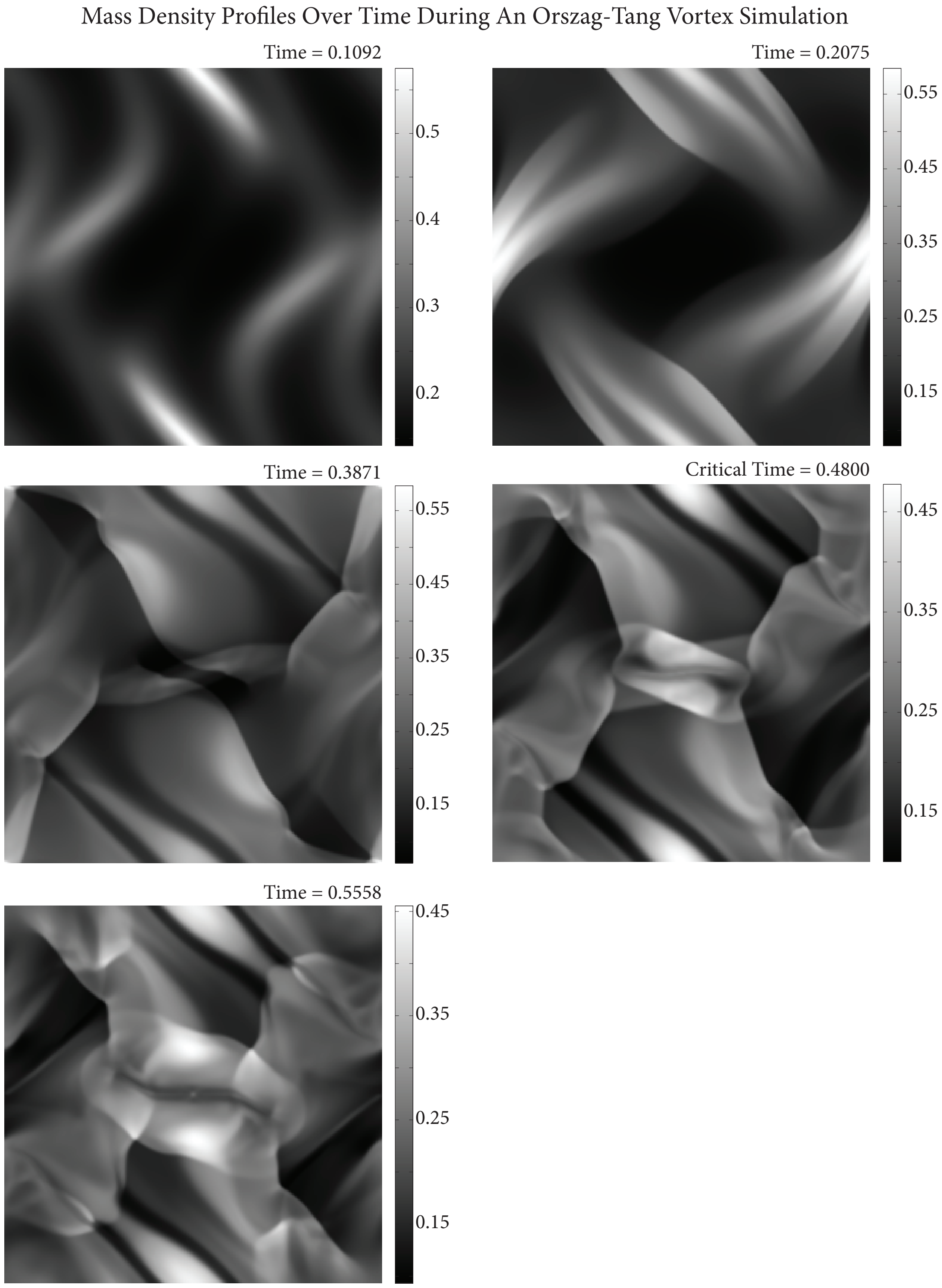

Sonic Mach Number At t=0 For the Orszag-Tang Vortex

Figure 5.16: Sonic Mach values for the Orszag-Tang vortex simulation at *t*=0.

Throughout the simulation the two-fold symmetry about the plane-normal axis is maintained. However, the evolution quickly loses its axially aligned symmetries as the non-zero velocities and magnetic fields contribute to an overriding radially Lorentz force that drives vortex behavior. The simulation is run until a critical time, *t*=0.48, which is the key comparison time. Figure 5.17 shows the evolution of the mass density over time until it reaches the critical time and shortly thereafter when turbulence begins to appear.



Mass Density Profiles Over Time During An Orszag-Tang Vortex Simulation

Time = 0.1092

Time = 0.2075

Time = 0.3871

Critical Time = 0.4800

Time = 0.5558

Figure 5.17: Evolution of the mass density for an Orszag-Tang vortex test up to the critical evolution time of t=0.4800 time units and just beyond. In the final time the turbulence starts to become apparent.

After the critical time the vortex fan structure that exists from the early evolution breaks down and magnetic turbulence dominates the evolution as shown in Figure 5.18 with snapshots of the mass density long after the critical time. A correct handling of the magnetic turbulence should appear quite chaotic while maintaining the two-fold symmetry the problem began with as it does in the various snapshots shown in Figure 5.18.

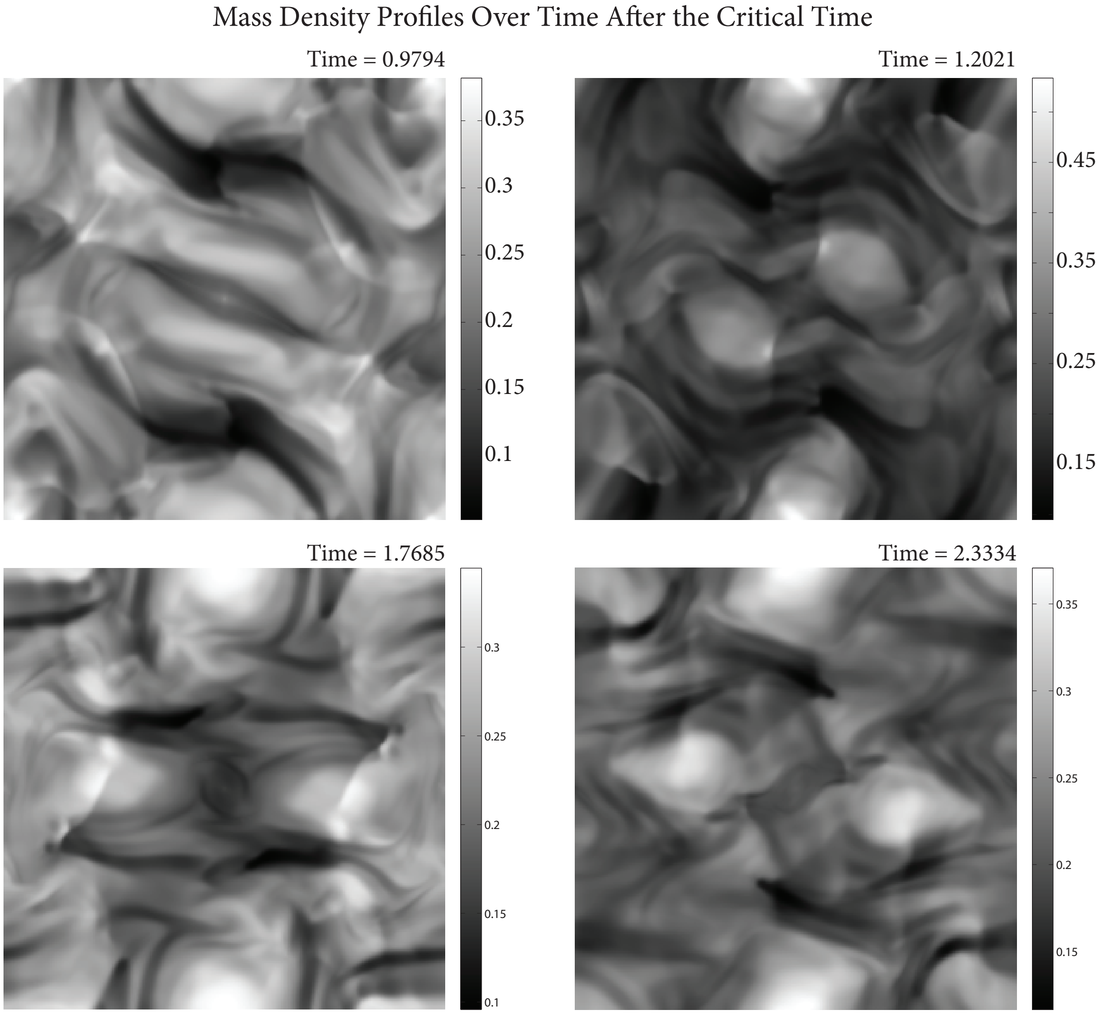


Figure 5.18: Evolution of the mass density in an Orszag-Tang vortex well after the critical time of t=0.48 time units, where magnetic turbulence has largely overtaken the evolutionary behaviors of the vortex.

As previously mentioned, the important point of comparison for the Orszag-Tang vortex is at the critical time, t=0.48 time units. Figure 5.19

shows the evolution of the vortex at this critical time in the same format as the initial conditions were presented in Figure 5.15.

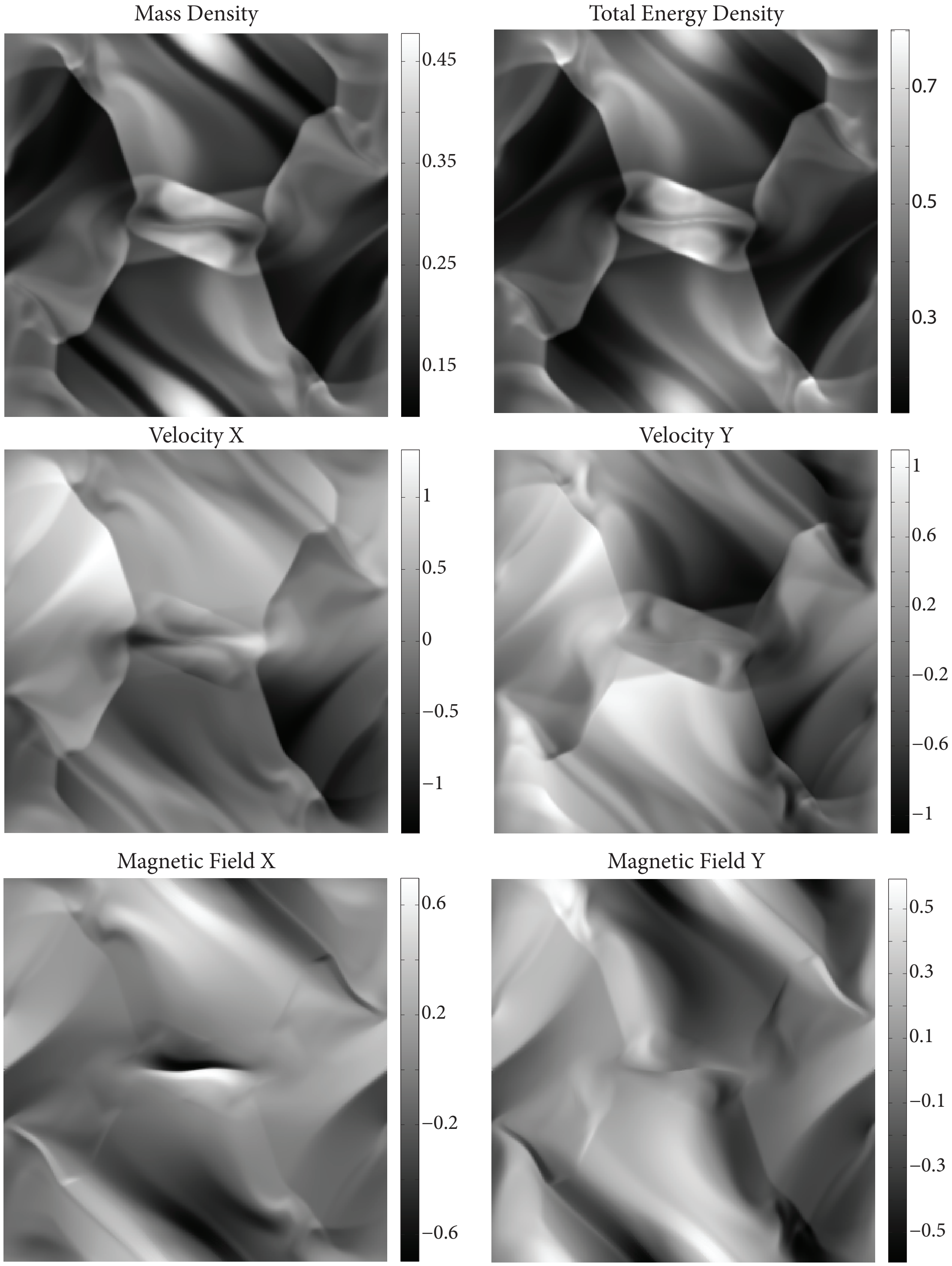


Figure 5.19: Orszag-Tang vortex simulation at the critical time, *t*=0.48.

As is demonstrated by inspecting the evolution of the mass density in Figures 5.17 and 5.18, the state of the vortex evolution in 5.19 has reached a point where the waves generated by the initial conditions have had a significant period of time in which to interact but not yet reached a point where they have become turbulently incoherent from the driving symmetries of the initial velocity and magnetic field conditions. It is also important to notice that the distribution of speeds within the system have changed significantly, including a growth in the supersonic nature of the waves as shown in the Mach results of Figure 5.20.

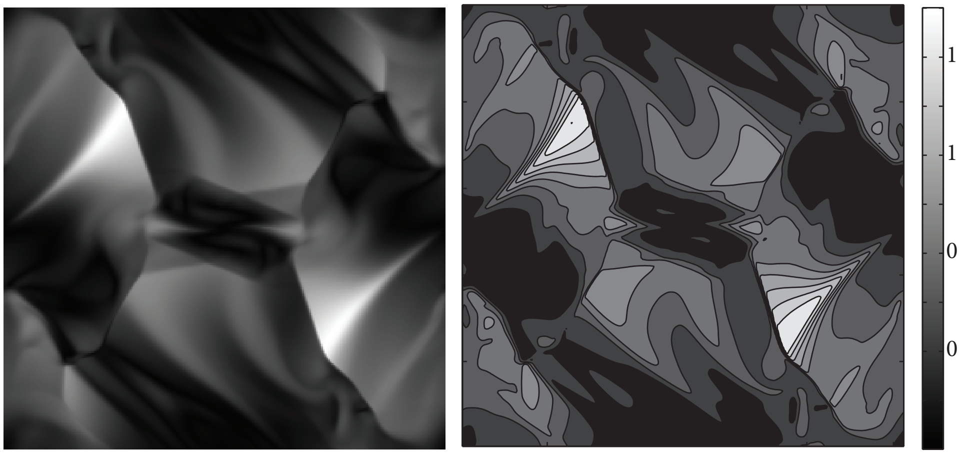

Sonic Mach Number At Critical Time t=0.48 For the Orszag-Tang Vortex

Figure 5.20: Sonic Mach values for the Orszag-Tang vortex simulation at the critical time, *t*=0.48 time units.

Figure 5.20 illustrates a key aspect of the Orszag-Tang vortex as a test problem for a new magnetohydrodynamic simulation software. The transition, growth, and evolution of complex, non-axis aligned supersonic magnetic shock waves within the domain are extremely sensitive to numerical and computational errors. Had the IMOGEN advection algorithms been found lacking in this regard the simulation would have quickly fallen apart and would look nothing like the results shown in the previous figures. To compare these results with others, the results shown in Figure 5.19 are augmented to inspect a few other properties of the system, specifically the pressures, compression, and vorticity of the vortex at the critical time. Figure 5.21 shows the results for these values, which are used for direct, qualitative comparison between the results of others.

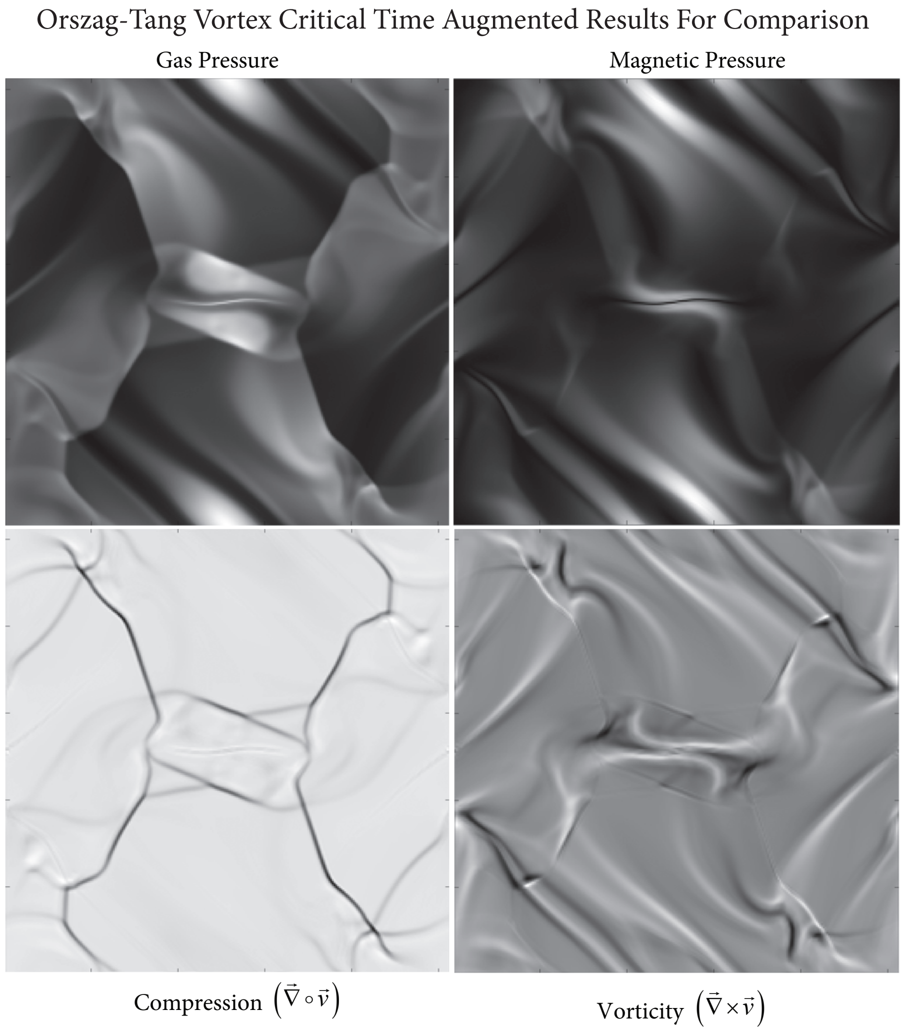


Figure 5.21: Additional results of the Orszag-Tang vortex simulation for comparison with other published simulation results.

The results in Figures 5.19 and 5.21 coincide favorably with the published results of many other Orszag-Tang vortex tests through a graphical, qualitative comparison (Arminjon & Touma, 2005; Dai & Woodward, 1998; Gardiner, et al., 2008; Londrillo & Del Zanna, 2004; Ryu, et al., 1998; Toth, 2000). In Ryu, et al. (1998), one-dimensional profiles of the gas and magnetic pressures were included at a position of *y*=0.428 at the critical simulation time. An extraction of those profiles, with overlays showing comparable values for the IMOGEN Orszag-Tang vortex simulation are shown in Figure 5.22.

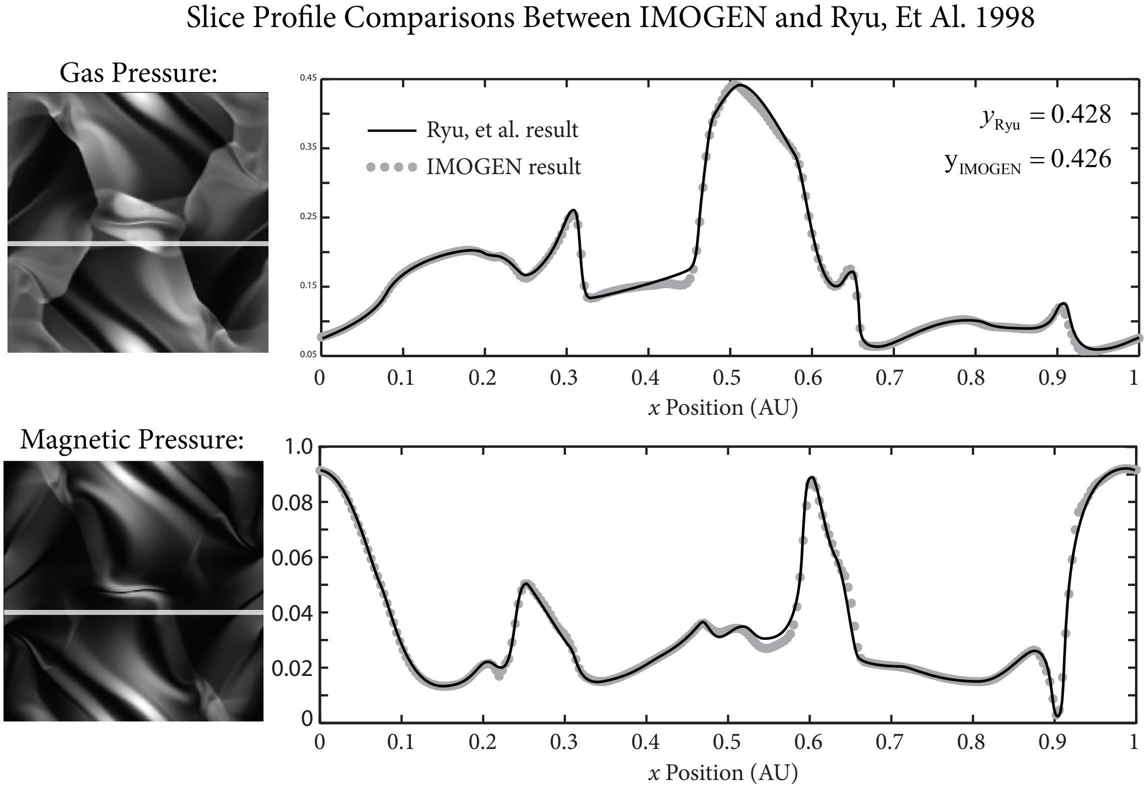


Figure 5.22: Comparison between Orszag-Tang vortex simulations conducted using IMOGEN and results published in Ryu, et al. 1998. Different handling of the boundary conditions and initial conditions introduces a slight offset in the absolute positions of the cells between the two results. Therefore, the results compared are at slightly different positions in the grids, *y*=0.428 for Ryu, et al. and *y*=0.426 for IMOGEN.

A slightly different handling of the spatial domain between the two simulations introduced a slight offset in the absolute positions of the two profiles compared in Figure 5.22. The results of Ryu, et al. (1998) are shown for a value of y=0.428, whereas the IMOGEN results are for a value of y=0.426. Even with the small discrepancy the correlation between the two results is quite high, all within reasonable tolerances given the difference in advection algorithms used, further demonstrating IMOGEN as a capable magnetohydrodynamic simulation tool.

A useful extension to the Orszag-Tang vortex test is to demonstrate the ability to reproduce the same result at different resolutions. Until after the critical time the vortex evolution should be stable enough for, within reason, a resolution independent result. After the critical time, once the magnetic turbulence takes over after the critical time the spatial frequencies allowed by the spatial domain resolution largely dictate the final structure. The results presented above were all for the standard 256×256 cell spatial resolution. Figure 5.23 shows the results of the 256 resolution along with resolutions of 128, 384, and 512. In each case the problem is normalized so that the absolute dimensions of the problem remain the same. The cell spacing in each case is, therefore, the reciprocal of the resolution so that the total length of an edge is one in all cases. As the boundary conditions are periodic, the results tile nicely against each other, a helpful aid in the resolution comparison.

From inspection of the various mass density profiles displayed in Figure 5.23, it is clear that IMOGEN is capable of correctly evolving the Orszag-Tang vortex for a number of different resolutions over a range of resolutions likely to be found in research applications. This and the success of the previous comparisons, demonstrates IMOGEN's readiness to handle simulations with complex magnetohydrodynamic advection.

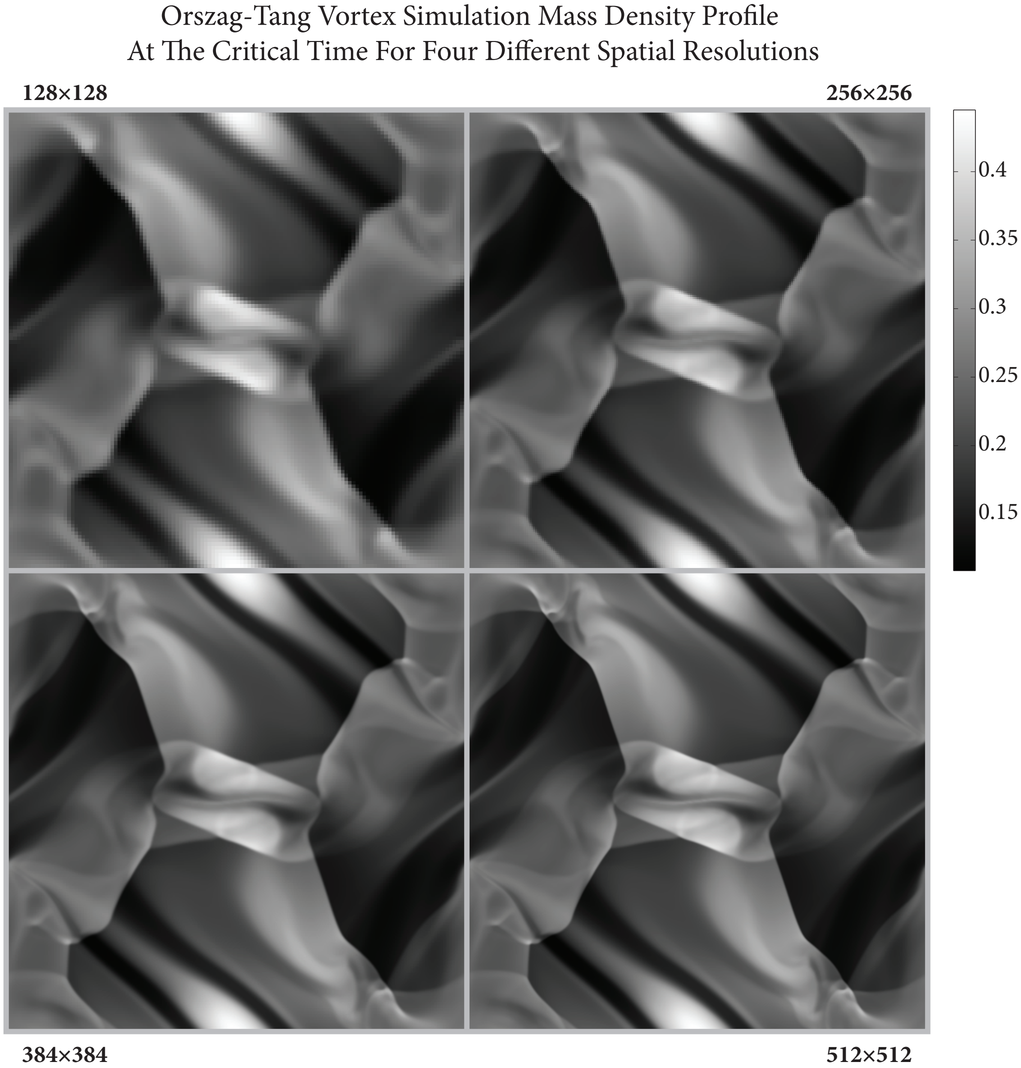


Figure 5.23: Mass density profiles of the Orszag-Tang vortex run at four different resolutions, 128×128, 256×256, 384×384, and 512×512 cells. In each case the large-scale structure of the vortex matches the other resolutions.

5.4. Other Tests

The three tests, the Sod shock tube, the Brio-Wu magnetic shock tube, and the Orszag-Tang vortex are sufficient to demonstrate IMOGEN's readiness for the magnetic accretion shock investigation. However, these represent only a fraction of the tests run during development and testing of IMOGEN as others were used to help refine the advection schemes, test the auxiliary algorithms, e.g. artificial viscosity, and demonstrate IMOGEN as a good general purpose magnetohydrodynamics simulation software for computational astrophysical research. It is worth briefly mentioning the results of two other tests for the sake of fortifying the previous test results.

5.4.1. Sedov-Taylor Blast Wave

The Sedov-Taylor blast wave problem is a good hydrodynamic test for verifying, among other things, the isotropy of the advection scheme. It begins with a uniform spatial domain of some low background density and pressure and at the center of the grid a the pressure is dramatically increased. In most cases this high pressure region is a sphere, in a three-dimensional domain, or a circle, in a two-dimensional domain, with a diameter of around 6-12 cells (Fryxell, et al. 2000).

When the simulation begins the high pressure region expands uniformly outward as a shock wave at supersonic speeds set by the ratio of the background to center pressures. After some evolution time the blast wave has expanded within the domain to a much larger sphere, or circle, with most of the density and pressure existing within the expanding shock; the region contained within the shock wave is evacuated as the shock wave expands outward.

It is then useful to look at how the simulation software handles uniform advection in all directions to understand the degree of anisotropy introduced by a spherical, or cylindrical, shock wave existing on a Cartesian grid.

For the cases of this test, the high density region was chosen to be only a single cell. This choice puts the IMOGEN hydroydnamic advection scheme at the greatest disadvantage because there is non way to recover the spherical nature of the blast wave from what was originally a single, rectangular, source cell. As such, this exposes the worst anisotropy possible for a spherical structure on a Cartesian grid. It should not be construed as how IMOGEN handles all spherical structures. Even with just a few cells of definition in the high pressure region the anisotropy caused by the rectangular structure of the grid is satisfactorily recovered.

Figure 5.24 shows the mass density profiles for a worst case scenario test. The anisotropy is readily apparent in the two-dimensional profile where the advection scheme conserves the advected quantities but does not handle the spherical propagation uniformly.

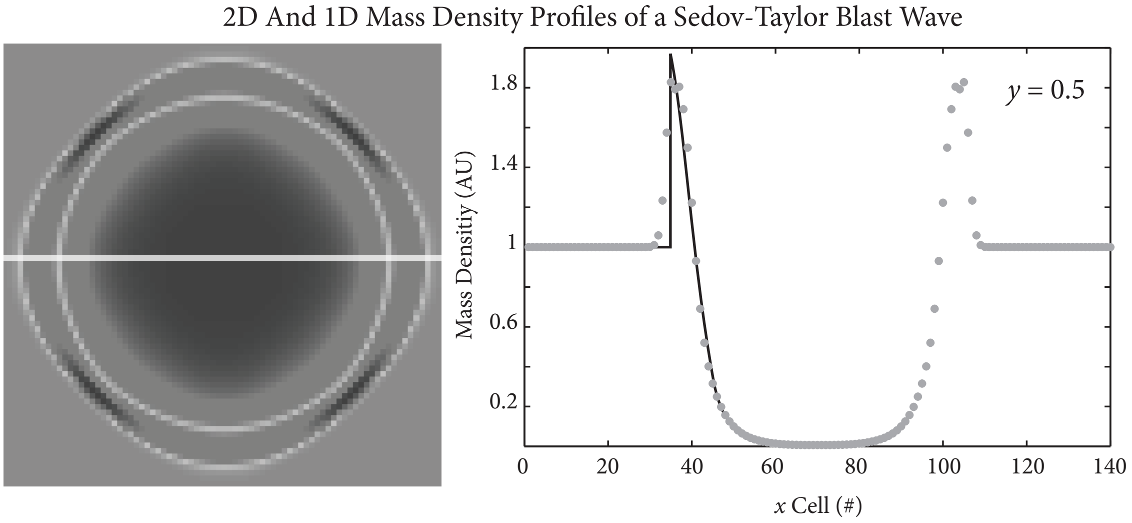


Figure 5.24: Worst case scenario for a Sedov-Taylor blast wave where the initial high pressure region was created using only a single cell to expose

the maximum anisotropy caused by the rectangular grid.

The simulation in Figure 5.24 was a three-dimensional run conducted on a grid 141×141×141 cells. To better analyze the advection anisotropy a plot, show in Figure 5.25, was generated showing the mass density at each cell within the grid versus its distance from the initial high density region at the center of the grid.

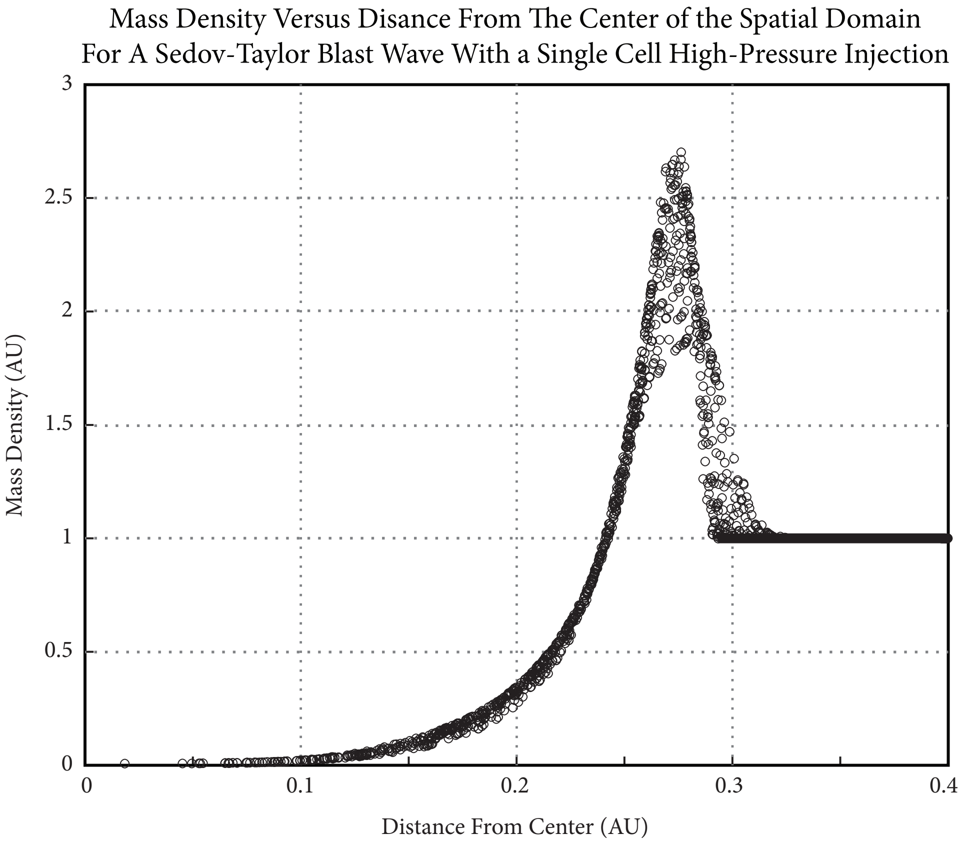


Figure 5.25: Mass density versus distance from the center of the spatial domain for all cells within the grid for the Sedov-Taylor blast wave results shown in Figure 5.25.

Figure 5.25 clearly shows the maximum possible advection anisotropy for a non-rectangular structure on the rectangular grid to better understand the limitations of the IMOGEN software given its Cartesian coordinate system and choice of advection schemes. Even in the worst case scenario there are a number of encouraging aspects to this result.

First the deviation in distance of the shock peak values is quite small, meaning that the advection schemes are doing very well at maintaining the structure despite the ill-conditioned initial conditions. Also, the shape of the shock front is reasonably well preserved, all things considered, as there are no spurious shapes or strange deformations caused by the mismatched structure and grid. Finally, the evacuation region behind the shock wave converges nicely to a nearly isotropic distribution. This demonstrates that the issue is largely contained to the shock wave, which is only first-order accurate anyway given the nature of the advection scheme.

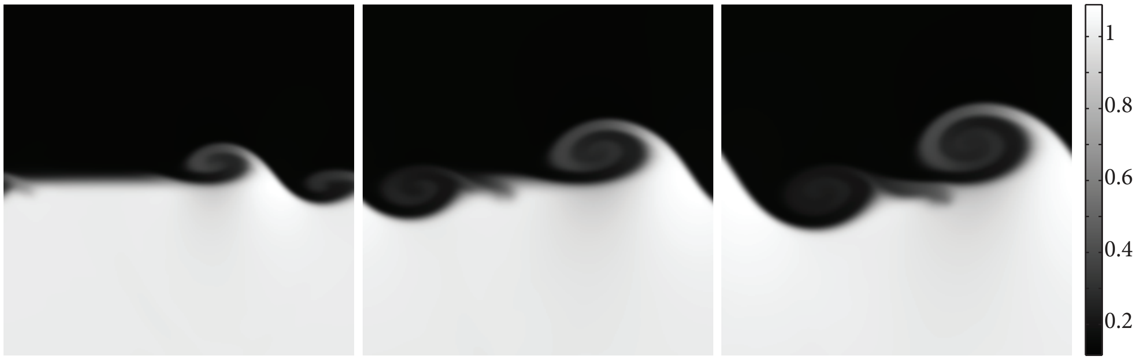
Hence, given a worst case scenario with ill-conditioned initial conditions IMOGEN is able to handle the advection of non-rectangular structures to a reasonable effect and result.

5.4.2. Kelvin-Helmholtz Instability

The other test worth a brief mention is the Kelvin-Helmholtz instability. The Kelvin-Helmholtz instability is a hydrodynamic instability caused when two regions of different mass densities and pressures shear against one another. This instability is extremely common in all kinds of hydrodynamic systems and has a distinctive behavior, the growth of a swirling pattern at the site of instability formation.

It is included here only as a qualitative example showing that IMOGEN correctly represents the behavior of the instability as the Kelvin-Helmholtz instability is likely present in some of the magnetic accretion shock wave simulations and so is worth presenting for reference.

Figure 5.26 shows the results of a Kelvin-Helmholtz simulation with the initial conditions of two regions of differing mass density and pressures stacked vertically. A uniform relative velocity was introduced between the two regions, in this case with a supersonic Mach number of one. To seed the instability the mass density of a single cell was perturbed along the border between the two regions. This test was conducted on a two-dimensional, 256×256 spatial domain resolution.


Mass Density Profiles Of The Kelvin-Helmholtz Instability At Various Simulation Times

Time = 157.2 Time = 240.2 Time = 289.9

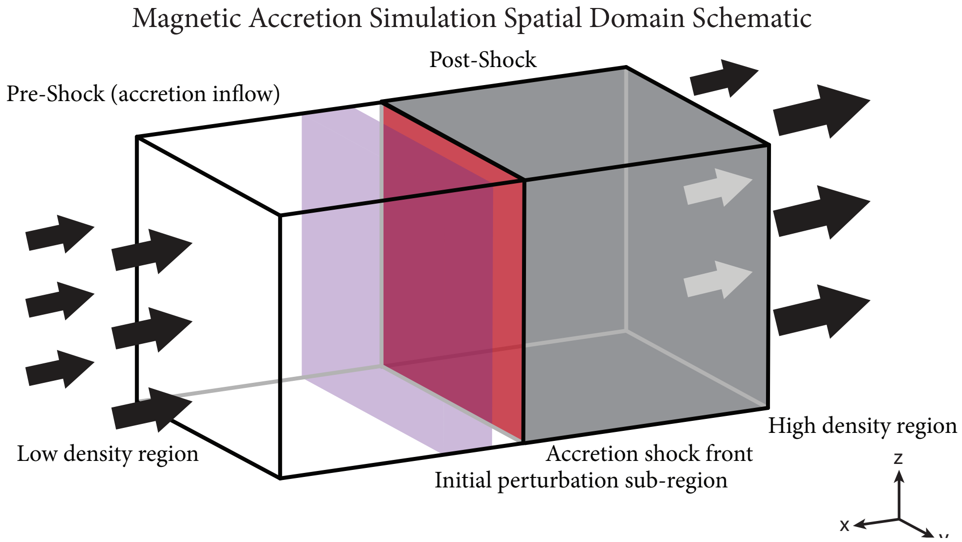
Figure 5.26: Kelvin-Helmholtz instability test results for a case with a supersonic Mach number of 1.0.

The results shown in Figure 5.26 exhibit the distinctive swirling pattern of a Kelvin-Helmholtz instability, demonstrating that IMOGEN correctly produces this common instability under the correct conditions. It is also interesting to note that in the absence of the perturbation IMOGEN is able to run the test indefinitely without generating the instability. In the absence of any explicit perturbation the numerical error isn’t sufficient to seed the instability itself in any simulation times tested.

CHAPTER VI

MAGNETIC ACCRETION SHOCK WAVE RESULTS

With IMOGEN fully developed and tested, it was finally time to employ it in the investigation of magnetic accretion shock waves. The simulation was constructed as a three-dimensional shock tube, similar to the Sod and Brio-Wu shock tube tests. The shock front was centered about the primary, *x*, axis with two distinct regions, the pre-shock accretion region and the post-shock region, as shown in Figure 6.1.



Magnetic Accretion Simulation Spatial Domain Schematic
Post-Shock
Pre-Shock (accretion inflow)
High density region
Low density region
Accretion shock front
Initial perturbation sub-region
z
x
y

Figure 6.1: Schematic illustration of the setup for the magnetic accretion shock simulation.

For the sake of easy comparison with the previous work presented in Stone & Edelman (1995), the spatial domain was scaled such that $-0.5L \leq y \leq 0.5L$ and $-0.5L \leq z \leq 0.5L$ where L=0.01. Unlike Stone & Edelman (!995), the *x* dimension was not held to a fixed, resolution-independent length but instead allowed to grow or shrink in absolute size depending on the resolution of the other dimensions. Additionally, the *x* dimension was handled in a non-uniform fashion with cell spacings that gradually grew larger toward the edge of the boundaries as shown in Figure 6.2. At the center 1/3 of the tube length the *x*-spacing of the cells were set to the smaller of the uniform *y* or *z* cell spacing. The gradual increase in the *x*-cell spacing occurred entirely within the first and last 1/3 of the tube up to a spacing five times larger at the *x* spatial domain boundaries than at the center of the tube.

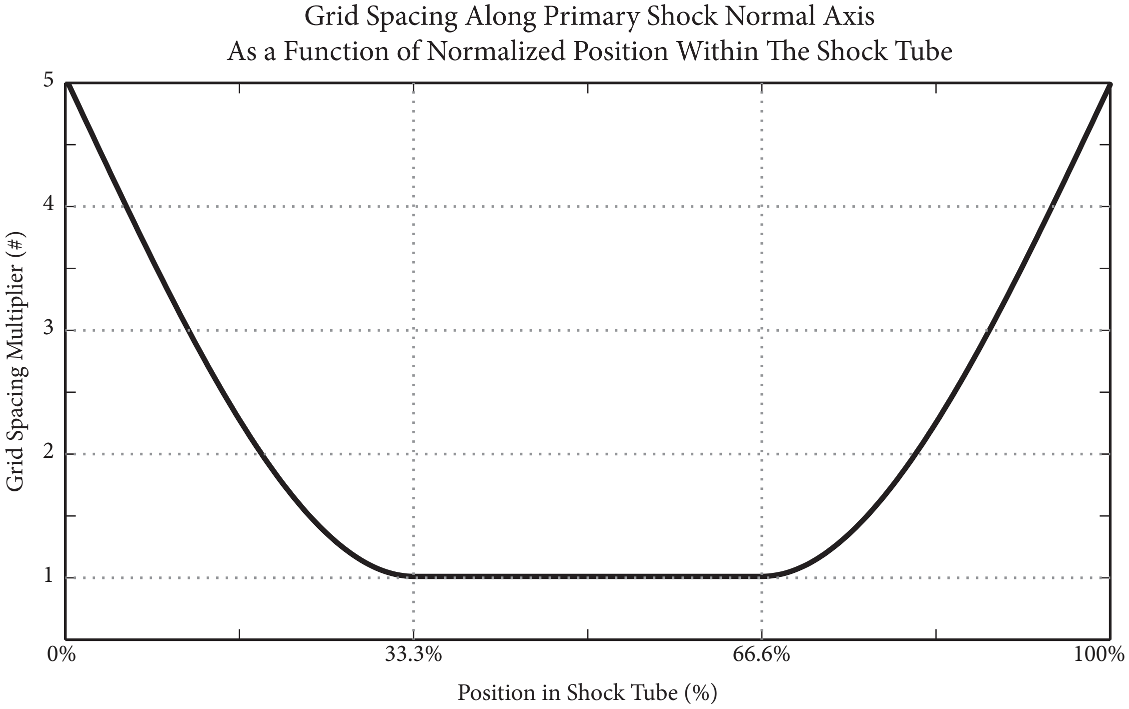


Figure 6.2: Non-uniform cell spacing of the primary shock tube axis to provide both a large shock tube and high resolution around the shock front. Shown in terms of a grid spacing multiplier coefficient applied to the default spacing determined by the smaller spacing of the shock-plane axes.

This non-uniform spacing allowed for both a large shock tube in which the with system could evolve as well as high resolution around the center region where the shock front resided.

A fade shifter, described in 4.3.2, was used for the edge boundary conditions on both ends of the primary axis, and a periodic, circular shifter used for all other edge boundaries along the other two, y and z, directions. The non-uniform spacing also helped stabilize the fade shifter as the gradual lengthening of the tube preferentially quenched the higher spatial frequency wave modes near the boundaries, which are the most difficult to handle for a non-reflective artificial boundary condition solver.

The initial conditions for the two regions were determined by solving the jump conditions, (2.12) - (2.16), using the normalizations,

$$\begin{aligned} &\rho_1 = 1 \quad P_1 = 1 \\ &v_{z,1\&2} = 0 \quad B_{z,1\&2} = 0 \end{aligned} . \tag{6.1}$$

Here the subscript 1 denotes the pre-shock accretion region and subscript 2 denotes the post-shock region. The jump conditions, (2.12) - (2.16), used to generate the initial conditions for the simulations were parameterized into four variables M_s, M_A, θ, and α. The sonic Mach number, M_s, specified how strong the shock front was in the pre-shock accretion region. The Magnetic or Alfvén Mach number, M_A, specified how slow or fast the shock was compared to the magnetic wave propagation speeds. The incident angle between the accretion shock column and the plane of the shock front, θ, which was constrained to lie within the x-y plane. The ratio of the specific heats coefficient, α, represented the deviation from the default γ_0=5/3 such that $\gamma = \alpha\gamma_0$, which roughly parameterized the thermodynamic state of the constituent plasma. The resulting parameterized jump conditions, including the normalizations of (6.1) were then,

$$v_{x,1} = \rho_2 v_{x,2}\,, \tag{6.2}$$

$$\rho_1 v_{x,1}^2 + 1 + \frac{B_{y,1}^2}{2} = \rho_2 v_{x,2}^2 + P_2 + \frac{B_{y,2}^2}{2}\,, \tag{6.3}$$

$$v_{x,1} v_{y,1} - B_x B_{y,1} = \rho_2 v_{x,2} v_{y,2} - B_x B_{y,2}\,, \tag{6.4}$$

$$\begin{aligned} &\frac{v_{x,1}}{2}\left(5\alpha + \boldsymbol{v}_1^2 + 2\boldsymbol{B}_1^2\right) - B_x\left(\boldsymbol{v}_1 \circ \boldsymbol{B}_1\right) \\ &= \frac{v_{x,2}}{2}\left(5\alpha P_2 + \boldsymbol{v}_2^2 + 2\boldsymbol{B}_2^2\right) - B_x\left(\boldsymbol{v}_2 \circ \boldsymbol{B}_2\right)\text{, and} \end{aligned} \tag{6.5}$$

$$v_{x,1} B_{y,1} - B_x v_{y,1} = v_{x,2} B_{y,2} - B_x v_{y,2}\,. \tag{6.6}$$

Note that there are no subscripts on the B_x values denoting the region as the shock normal magnetic jump condition specifies that $B_{x,1}=B_{x,2}=B_x$. Here $v_{x,1}$, $v_{y,1}$, B_x, and $B_{y,1}$ are defined in terms of the specified parameterization as,

$$v_{x,1} = \frac{M_s}{\sqrt{3/(5\alpha)}}\,, \tag{6.7}$$

$$B_x = v_x / M_A\,, \tag{6.8}$$

$$v_{y,1} = v_{x,1}\tan\theta\text{, and} \tag{6.9}$$

$$B_{y,1} = B_x \tan\theta\,. \tag{6.10}$$

The initial conditions for each simulation were determined by numerically solving the equations (6.2) - (6.10) for a given set parameters M_s, M_A, θ, and α, which were then used to populate the spatial domain.

After generating the primary spatial arrays from the calculated equilibrium initial conditions, a perturbation was applied to the mass density array in a section of the pre-shock accretion region just before the shock front to seed the growth of any instabilities. The perturbation was applied equally to the entire three-dimensional section that was 36 cells thick in the x-direction along the entire width, y, and height, z, of the tube as illustrated in Figure 6.1.

Simulations were generally run at a three different spatial resolutions $[x_{cells}, y_{cells}, z_{cells}]$:

- Low resolution: [300, 48, 48]
- Standard resolution: [300, 96, 96]
- High resolution: [300, 144, 144]

When necessary, tests were also run using the above y and z resolutions with different shock tube lengths of 600, 900, or 1200 cells, instead of the standard 300 cells.

Simulations were run over a volume of parameter space within the bounds,

- $2 \leq M_s \leq 10$
- $1/8 \leq M_A \leq 1/2$
- $0° \leq \theta \leq 75°$
- $18/25 \leq \alpha \leq 1$

The sonic Mach values were chosen to include both weak shocks, Mach

2, and strong shocks, Mach 10. While trials were run at intermediate values during early testing it quickly became clear that there was little nuance in the behaviors for intermediate ranges; the shocks were either weak or strong. This makes sense given the dependence in the magnetohydrodynamic equations to the square of the Mach number, which means that the strong shock regime quickly dominates even in the lower part of the strong regime.

The Alfvén Mach values were chosen because they provide a range of magnetic field strengths reasonable, by ratio to the normalization, to those assumed present in the astrophysical systems of interest described in the early chapters. Some additional testing was done with very large Alfvén Mach values, e.g. 10, 100, and 1000, which have very weak fields that will also be discussed. However, those correspond to the fast shock regime and are of interest mostly for comparison.

The range of incident angles was chosen to be as large as possible. Parallel shocks provide an interesting trial because the analytical solutions are much less difficult, which means more work has been done to understand those cases. However, in most systems oblique accretion angles are highly likely and so they were fully explored. Depending on the strength of the magnetic field there is a maximum angle beyond which no equilibrium solution exists. For the range of Alfvén Mach values specified, these limits lie between 55°-80°. For values near the equilibrium solution limit all simulations were violently unstable. The magnetic turbulence was so strong that it became difficult for IMOGEN to evolve the system for any significant length of time before numerical errors would destroy the shock. There are certainly additional steps, such as increasing artificial viscosity, to prevent these from happening, but that would require a substantial amount of fine tuning and wasn't necessary for this first investigation. Therefore, for most simulations the incident angle was limited to values of 45° or less, as that angle was successfully evolved for almost all cases.

The range of specific heat ratio coefficients was chosen so that the ratio of specific heats, γ, would lie between 6/5 and 5/3. The nominal value used for most simulations was 5/3, which represents a mono-atomic gas as would be expected in a hot plasma. The other values tested were 7/5, representative of a generic diatomic gas, and 6/5 representative of an even more complex gas at lower temperatures.

The results of each simulation are displayed in a separate figure with a key at the top to easily read the values for each of these parameters to aide in comparison. Each simulation will also be referred to by these four parameters. For example, a simulation with M_s=10, M_A=1/8, θ=10°, and α=1 would be identified as: {θ-10° | A-1/8 | S-10 | α-1}.

6.1. Preliminary Results

The {θ-10° | A-1/2 | S-10 | α-1} simulation was chosen as the first test trial as it corresponds to a slightly oblique shock with a magnetic field strength on the lower end of the parameter space and the most commonly used specific heat ratio coefficient, that of a mono-atomic fluid. It was also a set of parameters investigated by Stone & Edelman (1995), serving as a source of comparison to confirm IMOGEN's correct handling of the magnetic accretion shock simulations. The results of the {θ-10° | A-1/2 | S-10 | α-1} trial are shown in Figure 6.3.

Each simulation trial figure, like Figure 6.3, follows the same format beginning with a parameter identification key at the upper right corner. To the left of that is a three-dimensional contour image of the mass density at the simulation termination time, a time that differs depending on the simulation, and is colored by the strength of the z component of the magnetic field. Given that the initial conditions specify no z component for the magnetic field, this coloring illustrates the magnetic waves generated by the growth and evolution of the corrugation instability of the shock front. Below the parameter identification key is a smaller contour image that includes a wireframe that displays how the full contour image is arranged within the spatial domain for the simulation.

Beneath the contour image is the instability growth plot that shows more quantitatively the evolution of the corrugation instability for the magnetic accretion shock wave simulation. This instability growth parameter, originally developed by Stone & Edeleman (1995) is defined as,

$$\xi \equiv \log\left(\frac{\left\langle B_z^2 \right\rangle_{\nabla\circ\boldsymbol{v}}}{\boldsymbol{B}^2}\right). \tag{6.11}$$

Hence the growth of the corrugation instability is measured as the log of the compression-weighted average of the z component of the magnetic field squared over the square of the magnitude of the total magnetic field and tracks the growth in the z component of the magnetic field, which is initially zero. The compression-weighted average is used to preferentially select growth that occurs locally around the shock front as a direct result of the corrugation of the front.

For instability growth parameter plots that exhibit an early, distinct linear growth phase, which corresponds to exponential growth as the ξ parameter is logarithmic, the growth plot also includes a normalized early growth rate value. As will be shown in section 6.3.5, the absolute growth rates are resolution dependent, and therefore, all growth rates have been normalized by the growth rate of the initial trial at the standard resolution for direct comparison. This also provides an easier means of comparison between the growth rate results of ongoing linear research.

Below the corrugation growth plot are two *x*-*y* plane profiles of the magnetic field in the area around the shock front, arranged with the pre-shock to the left and the post-shock to the right. The left of these profiles displays the ratio of the *z*-component of the magnetic field over the total magnetic field strength,

$$\Psi = \frac{B_z}{|\boldsymbol{B}|} . \tag{6.12}$$

The right of the two profiles is a nonlinear, dynamically-scaled remapping of the plot on the left to preferentially amplify the magnetic wave phenomena even though it is often many orders of magnitude smaller than the *z* magnetic field strength at the surfaces of the shock front. For details on the processing algorithm used to generate this profile see appendix 2. This profile is useful for visualizing the characteristic magnetic wave phenomena generated by the growth and evolution of the instability that is often hidden by the strong growth in the field around the surface of the front. In both profiles the initial position of the shock front is shown for reference as a dotted line. Also, both profiles have been truncated to show only the center 1/3 of the simulation domain where the grid cell spacing is uniform along the primary, shock normal direction.

All of the plots and profiles, except for the growth plot, are snapshots taken at the conclusion of the simulation and correspond to the terminal point on the growth plot.

In Figure 6.3 the corrugation of the shock front is clearly visible in the mass density contour plots. What was an initially smooth front first rippled and then grew over the course of the simulation. This growth was cumulative, not periodic despite a contour profile that might suggest periodic behavior. In the post-shock region the corrugation instability was also responsible for the formation of high and low density columns, or fingers, which penetrated and grew downstream into the stellar medium. These fingers, which are surprisingly stable throughout the simulation, were able to remain cohesive in large part due to a three-dimensional structure of current loops, generated as part of the initial growth of the instability, that define, constrain, and prevent them from being destroyed by other types of instabilities, such as the Kelvin-Helmholtz instability.

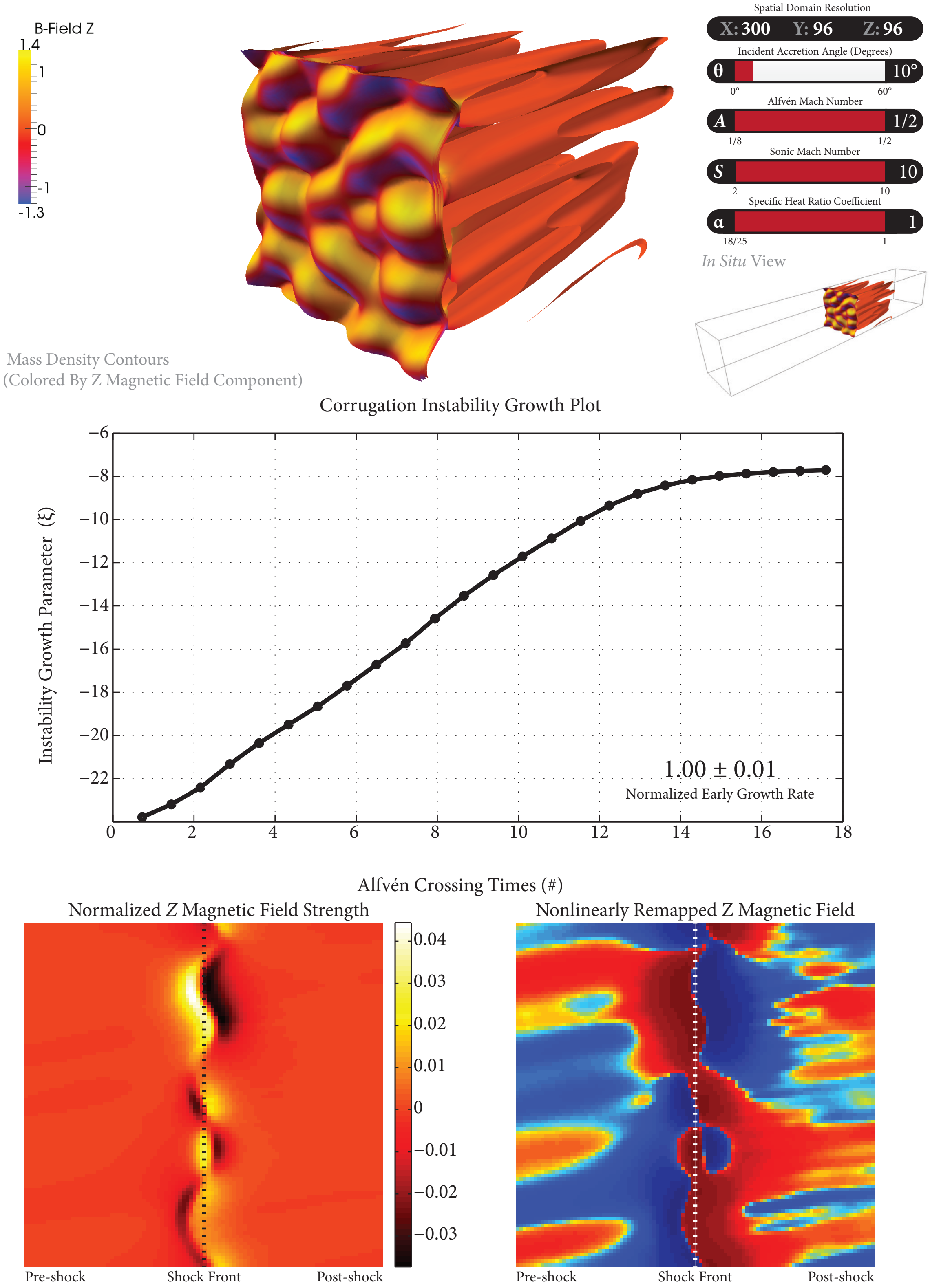


Figure 6.3: Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial at standard resolution as the archetype for testing and later comparison.

The corrugation instability growth plot for the {θ-10° | A-1/2 | S-10 | α-1} trial correlates well with the results published by Stone & Edelman (1995), exhibiting a distinct exponential growth that eventually saturates out well into the nonlinear regime.

6.2. Fade Shift Generator Verification

After the successful first trial, the effectiveness of the fade shifting boundary condition solver, see 4.3.2 for details, had to be addressed before expanded exploration. That the results of the {θ-10° | A-1/2 | S-10 | α-1} trial compared well with other published results was encouraging, but more direct testing was necessary to be certain.

Without a known solution the only option was to make a self-consistent comparison of the evolution of the trial simulation conditions using different shock tube lengths. Had the boundary conditions somehow negatively impacted the evolution of the simulation then comparison of trials run at different tube lengths would have exposed the problem. This would have been observable as significant deviations between the evolutionary behavior of the different trials given that the separation between the boundary region and the shock front would change the length and time scales of any interference from the boundary solver.

Three trials were run using the same initial conditions as {θ-10° | A-1/2 | S-10 | α-1}, but this time at low resolution so that they could be run for much longer simulation times for more thorough comparison. In each of the three trials the tube length was set to a different length by changing the resolution of the primary axis of the spatial domain. The resolutions

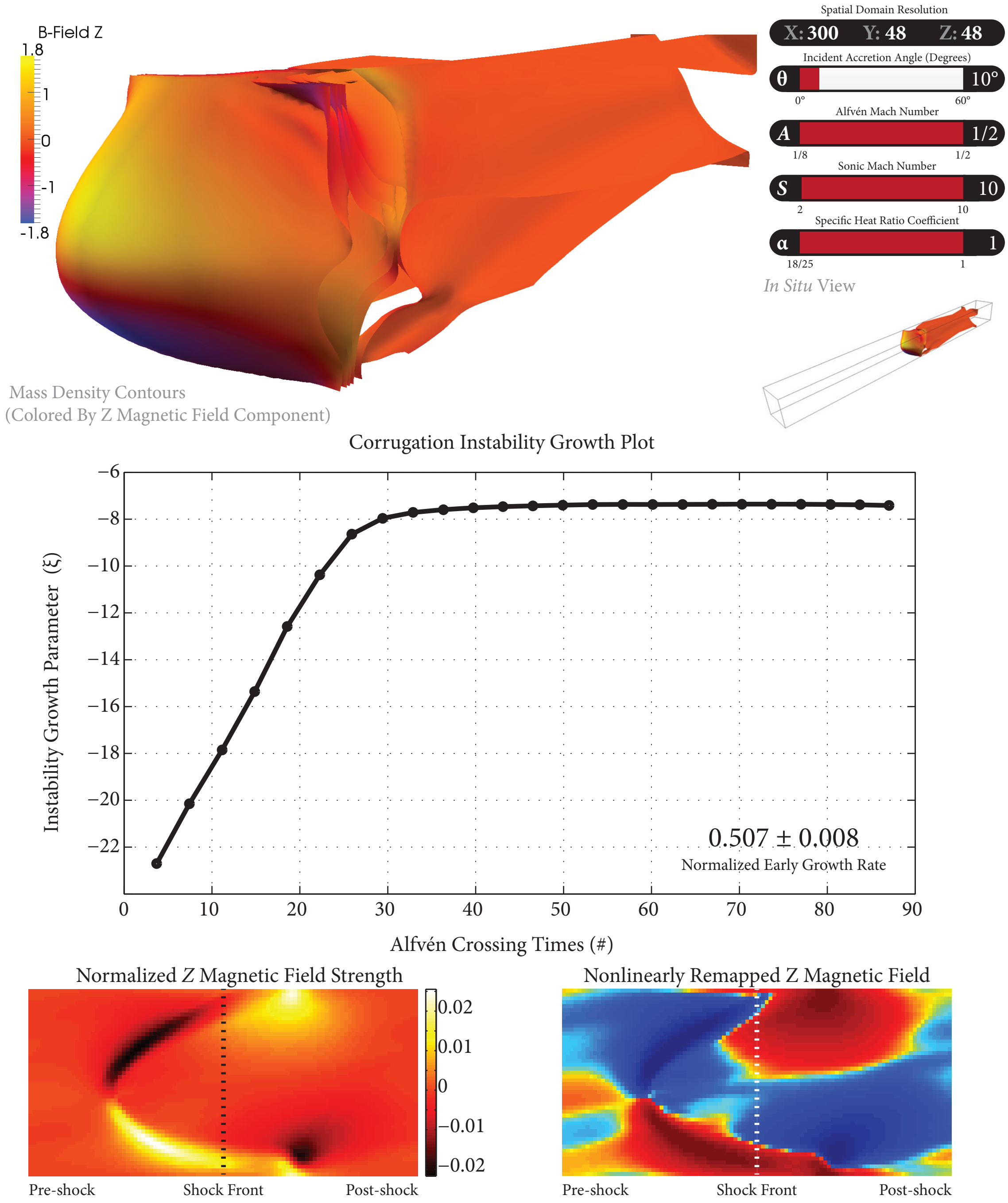


Figure 6.4: Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial at low resolution with a standard 300 cell primary axis resolution for testing of the fade shifting boundary conditions.

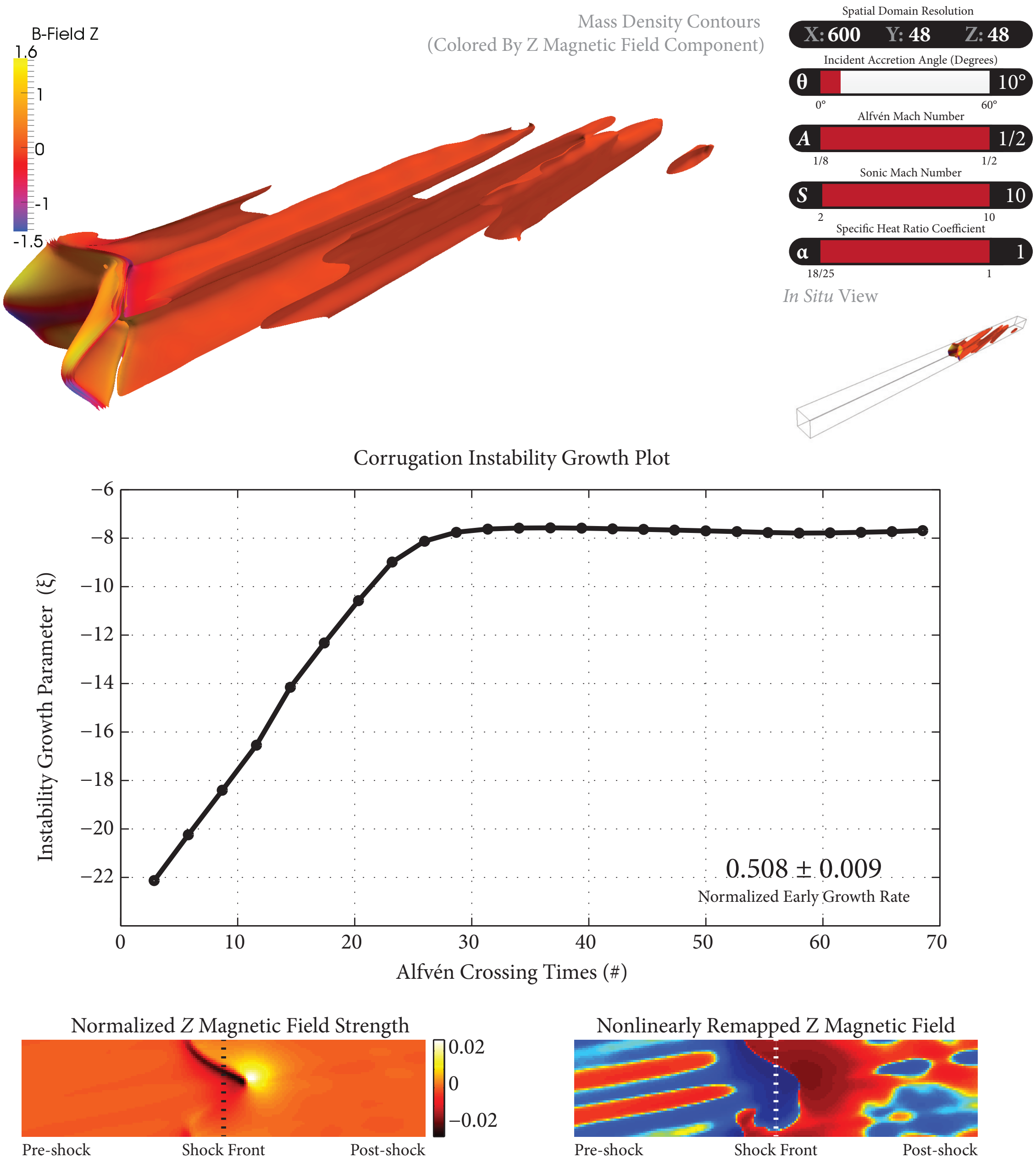


Figure 6.5: Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial at low resolution with a double-length, 600 cell primary axis resolution for testing of the fade shifting boundary conditions.

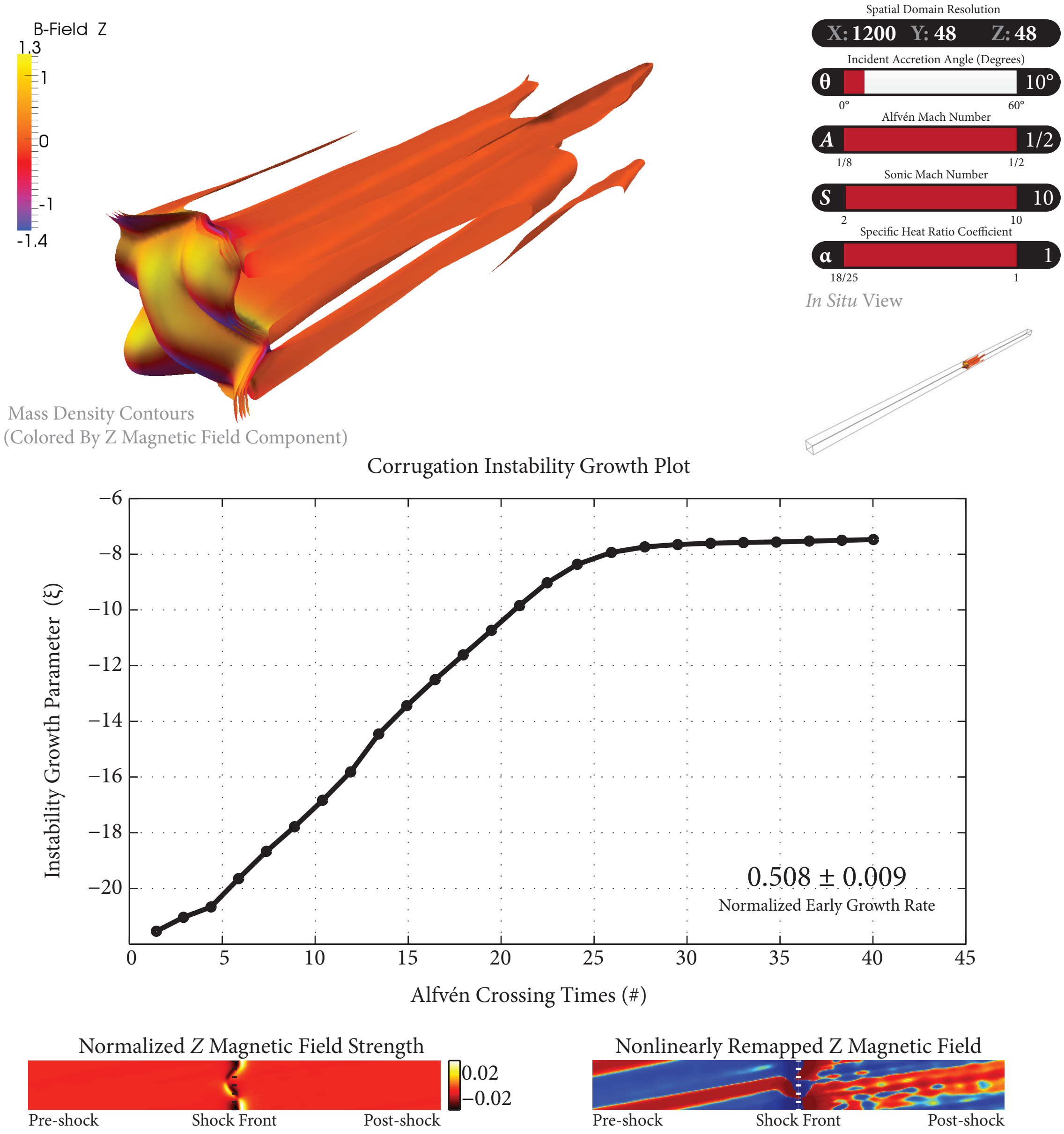


Figure 6.6: Results of the {θ-10° | A-1/2 | S-10 | α-1} trial at low resolution with a quadruple-length, 1200 cell primary axis resolution for testing of the fade shifting boundary conditions.

chosen were the default 300 cells, a double-length tube of 600 cells, and a quadruple-length tube of 1200 cells. The results of each of these trials is shown in Figures 6.4-6.6.

A direct comparison of the growth parameter curves for these three trials is shown in Figure 6.7. The plot demonstrates no significant deviation between the three trials and indicates strongly that the fade shifter correctly handles the boundaries during magnetic accretion shock simulations.

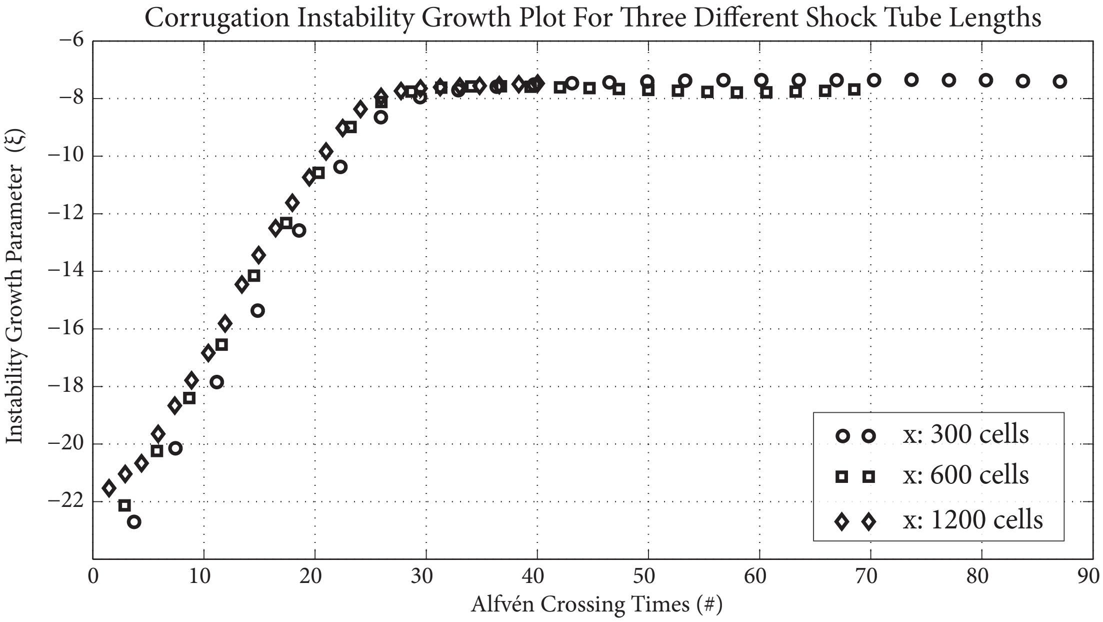


Figure 6.7: Direct comparison of the growth rate plots for three different shock tube lengths, 300, 600, and 1200 cell resolutions of the {θ-10° | A-1/2 | S-10 | α-1} trial. The high correlation between the three growth plots suggests that the fade shift boundary condition functions as expected.

Some additional observational testing was conducted on the boundary solver by monitoring handling of the initial waves generated by the shock front in response to the seed noise. During this early simulation time the reflections of the noise off of the shock front are many orders of magnitude larger than the waves generated afterward from the early growth of the instability. Consequently, the noise reflection acts as a solitary outflow wave, and were any non-physical reflections produced by its propagation into the boundary regions, there would have been clearly observable counter-reflections against but none were ever detected.

6.3. Simulation Comparisons

Satisfied with the initial trials and performance of the fade shifter, the investigation was expanded to cover the range of parameter space previously discussed. The following sections present some key results of this investigation.

6.3.1. Sonic Mach Variations

Figure 6.8 show a trial for {θ-10° | A-1/2 | S-5 | α-1}, similar to the previous {θ-10° | A-1/2 | S-10 | α-1}, except for a sonic Mach number of 5 instead of 10. A sonic Mach number of 5 is still in considered to be in the strong regime and the evolution of the instability behaves similarly to the sonic Mach 10 trial, Figure 6.3, with regard to the instability growth and the z magnetic field behaviors. Interestingly, the fingers that extend into the post-shock region are less pronounced in the sonic Mach 5 trial even though the instability growth is nearly identical. The contour image can be a bit deceptive here as the fingers aren't actually smaller in the sonic Mach 5 trial, they are just softer, with smaller density deviations from the background that alter their contour display.

Figure 6.9 shows a trial, {θ-10° | A-1/2 | S-2 | α-1}, for an even smaller

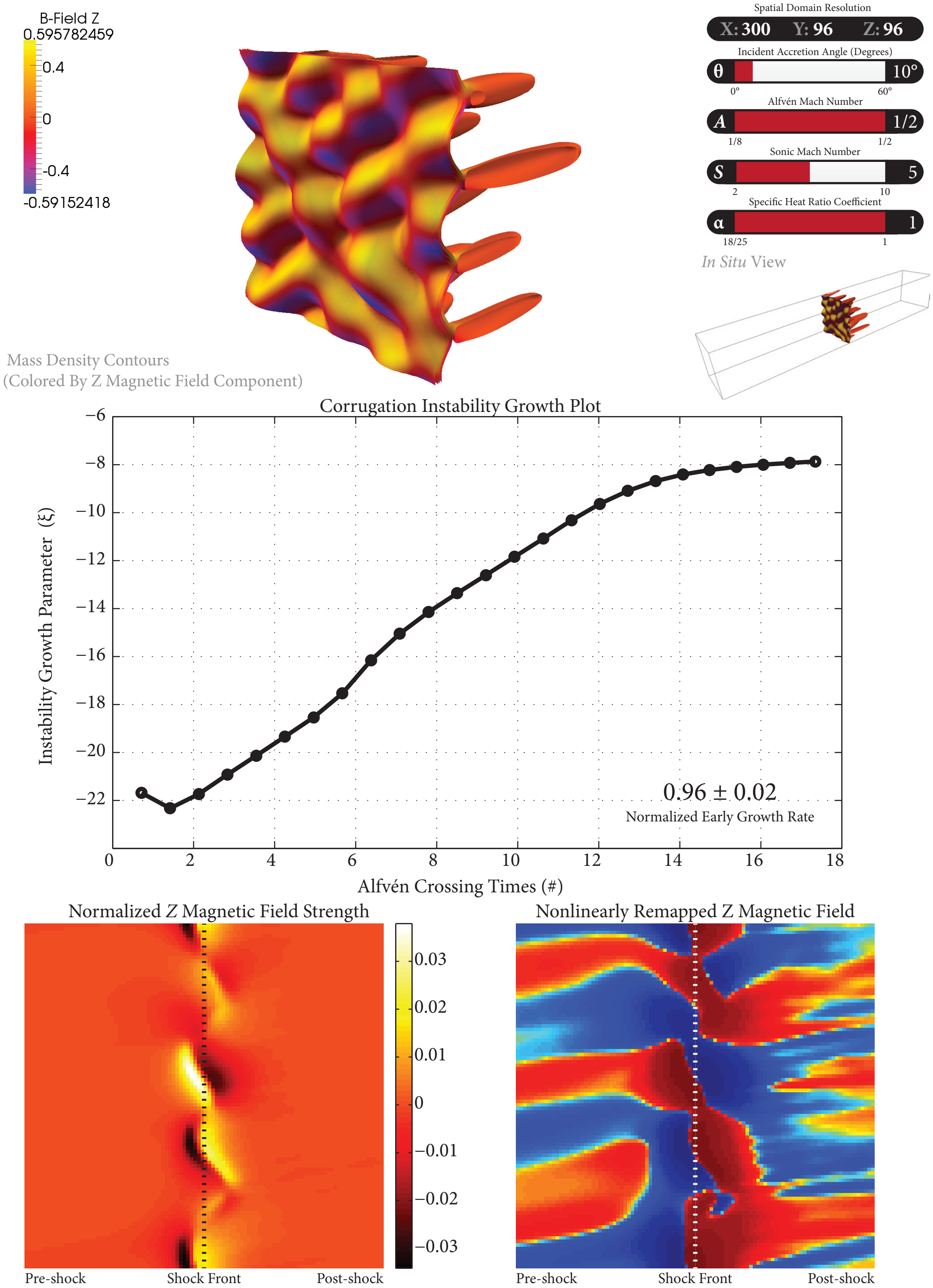


Figure 6.8: Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}5 \mid \alpha\text{-}1\}$ trial for comparison with the previous sonic Mach 10 trial, $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$.

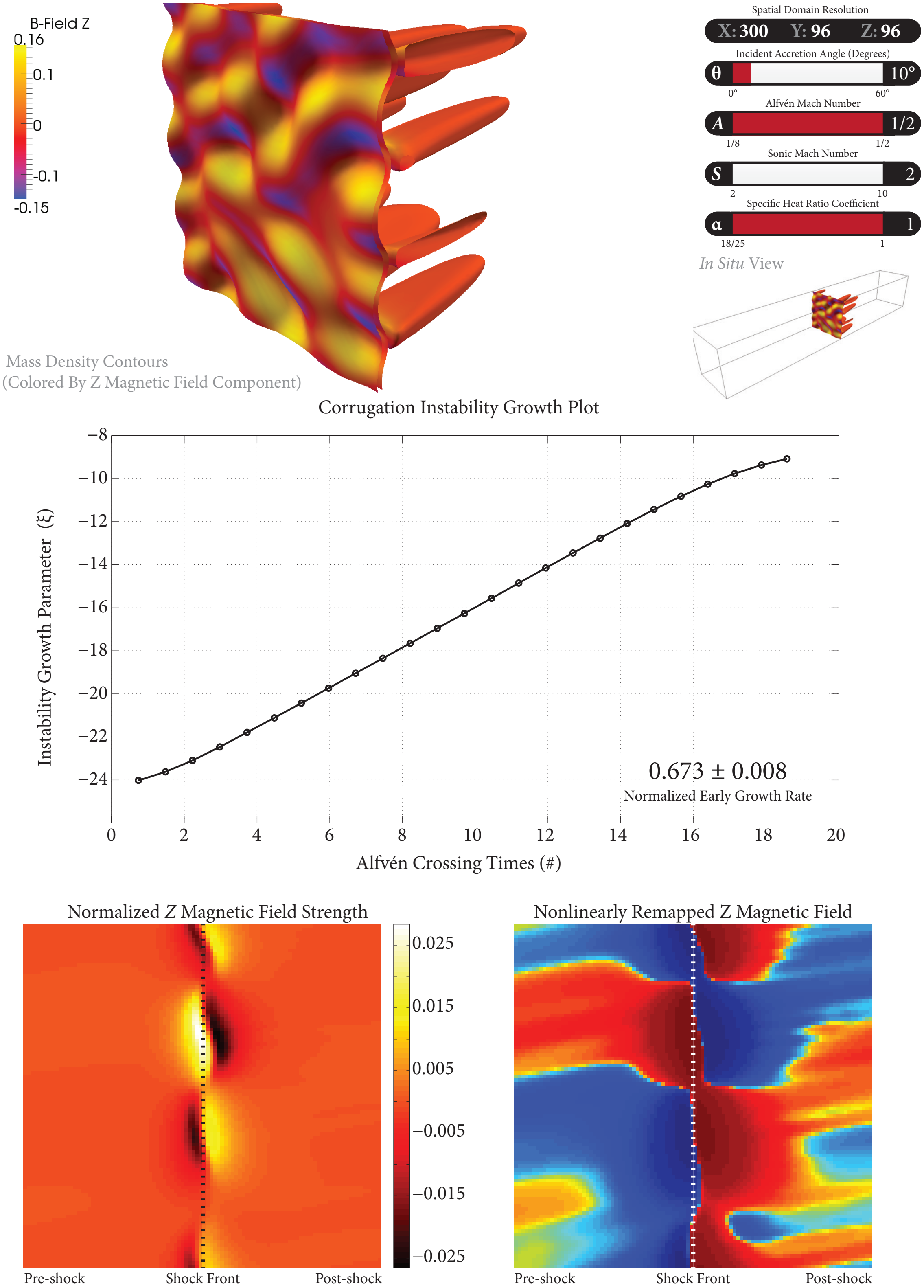


Figure 6.9: Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}2 \mid \alpha\text{-}1\}$ trial for comparison with the previous sonic Mach 10 and 5 trials.

sonic Mach number of 2, which is in the weak shock regime. In this case the instability growth is significantly slower than the strong shock regime trials.

A direct comparison of the growth parameters for the three trials, Figures 6.3, 6.8, and 6.9 is shown in Figure 6.10. There is a distinct difference between the growth rates of the corrugation instability in strong versus weak shock regimes, but very little difference between growths of different sonic Mach numbers within the strong shock regime. Both regimes, however, appear to tend toward the same final unstable state. This disagrees with earlier work that suggested that weaker shocks were more unstable than strong ones. There is a sizable difference in the growth rates of the instability between the weak and strong shock regimes, and weak shocks always tend to grow more slowly than their strong brethren.

While Figure 6.10 only illustrates the weak versus strong shock regime for one set of initial conditions, the weak versus strong test was carried out for a number of differing initial conditions within the specified parameter space and Figure 6.10 is representative of their commonly shared behavior.

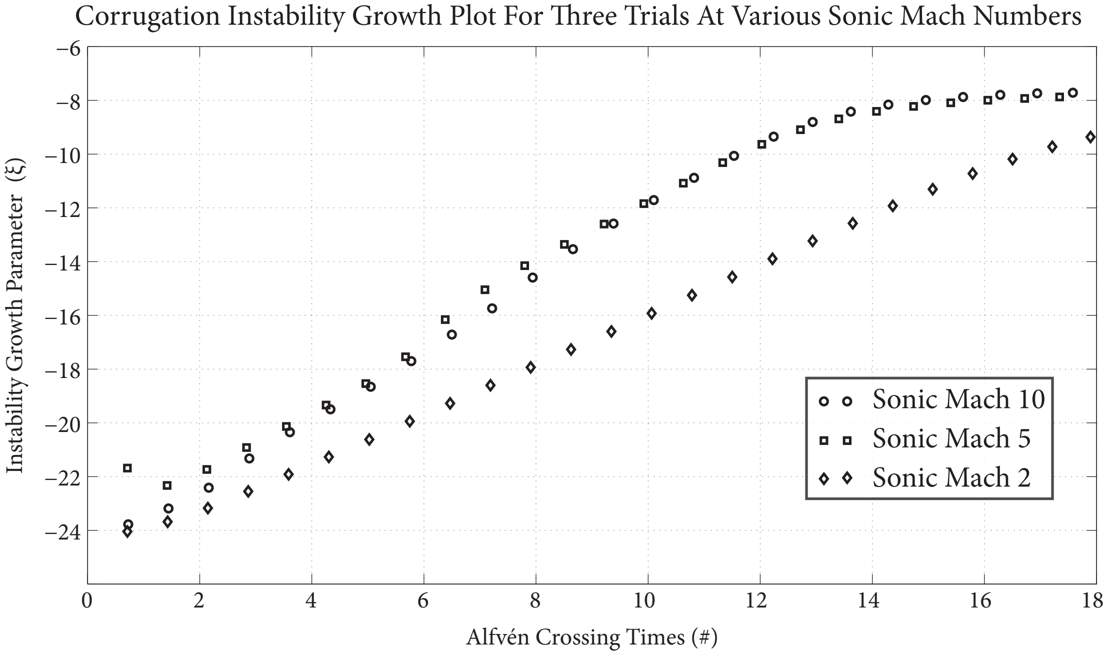


Figure 6.10: Comparison of corrugation instability growth parameters of the {θ-10° | A-1/2 | S-10 | α-1} , {θ-10° | A-1/2 | S-5 | α-1} , and {θ-10° | A-1/2 | S-2 | α-1} trials. A clear distinction exists between growth of the instability in strong and weak shock waves.

6.3.2. Incident Angle Variations

Next, trials were conducted to explore instability variations based on the initial incident angle of the accretion column with respect to the normal direction of the shock front. Results for these trials are shown in Figures 6.11-6.13.

In each of these trials the initial conditions were identical to the original test trial, {θ-10° | A-1/2 | S-10 | α-1}, except for variation in the incident angle parameter. Additional trials at other angles were also conducted, but these three, along with the initial trial, are representative of the larger set of runs. For small incident angles, including the parallel shock case, the corrugation instability follows a well-behaved growth and saturation as seen in Figure 6.3 and Figure 6.11.

For the parallel, 0° degree incident angle, shock case, as shown in Figure 6.11, the nonlinear results correspond well to previous linear predictions. Those linear results found that parallel shocks, and likely small incident angle shocks as well, exhibit aperiodic instability growth along with fast dissipation of the instability in the pre-shock region and slow dissipation of the instability in the post-shock region due to the large and small amplitudes of the dissipation portion of the linear wave solution in the respective regimes. The nonlinear simulations support this with much larger finger growth into the post-shock region than into the pre-shock region.

As the incident angle increases, the growth properties change to be slower and noisier as it appears that the shearing instabilities become more important. For the higher angles, above some critical angle that is dependent on multiple parameters, particularly the strength of the magnetic field, the shearing instabilities and corrugation instabilities form a feedback loop that prevents either from dominating early. The $\{\theta\text{-}45° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial, shown in Figure 6.13 is such an example.

From additional testing of the $\{\theta\text{-}45° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial, run for long periods at low resolution, the corrugation instability eventually forms, but behaves differently because of the shearing behavior at the surface of the front as shown in Figure 6.14. The behavior is less random in terms of spatial frequencies then the lower resolutions. There seems to be a natural frequency that develops between the shear instability and corrugation

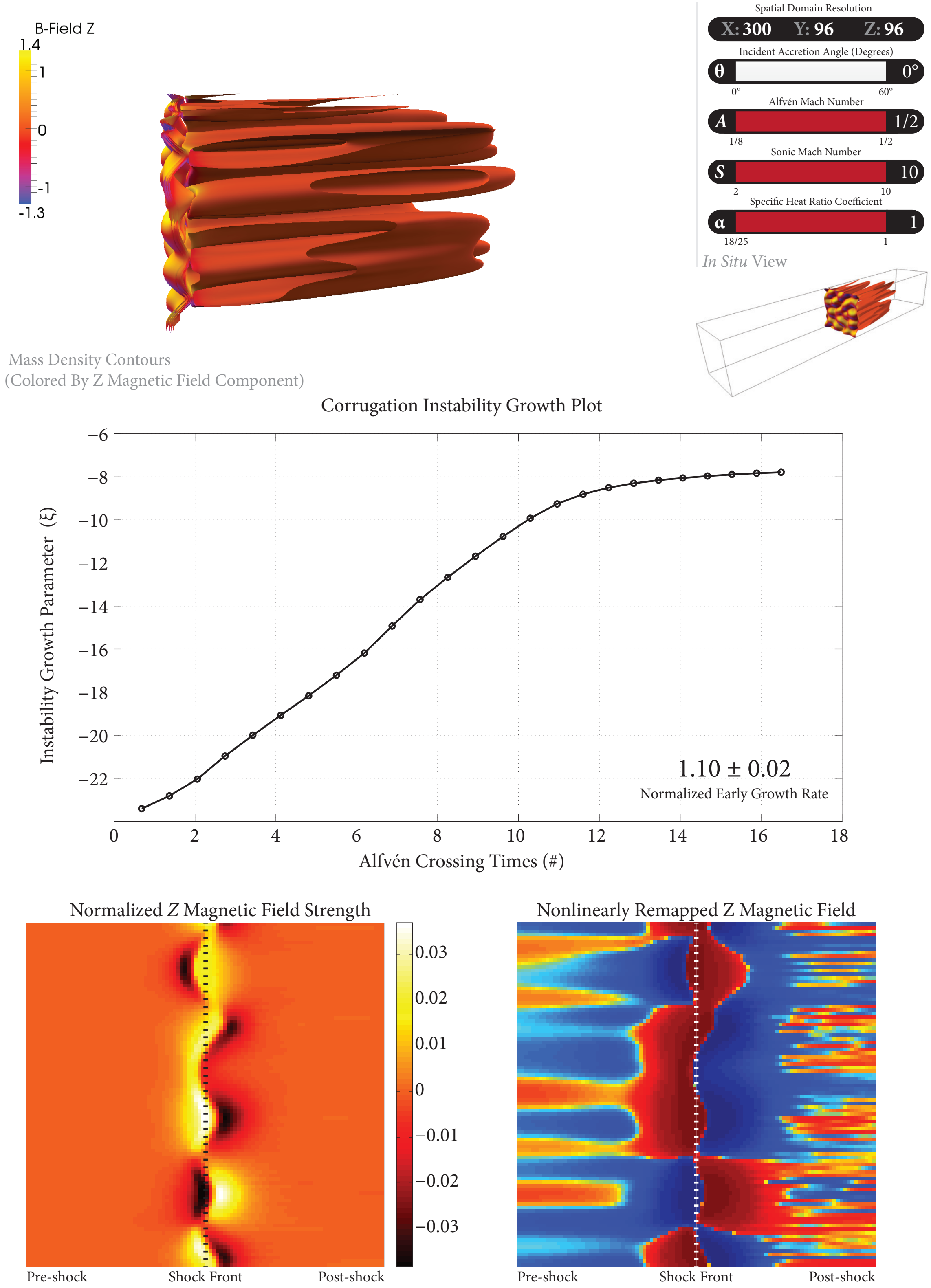


Figure 6.11: Results of the $\{\theta\text{-}0° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial for incident angle variation comparison.

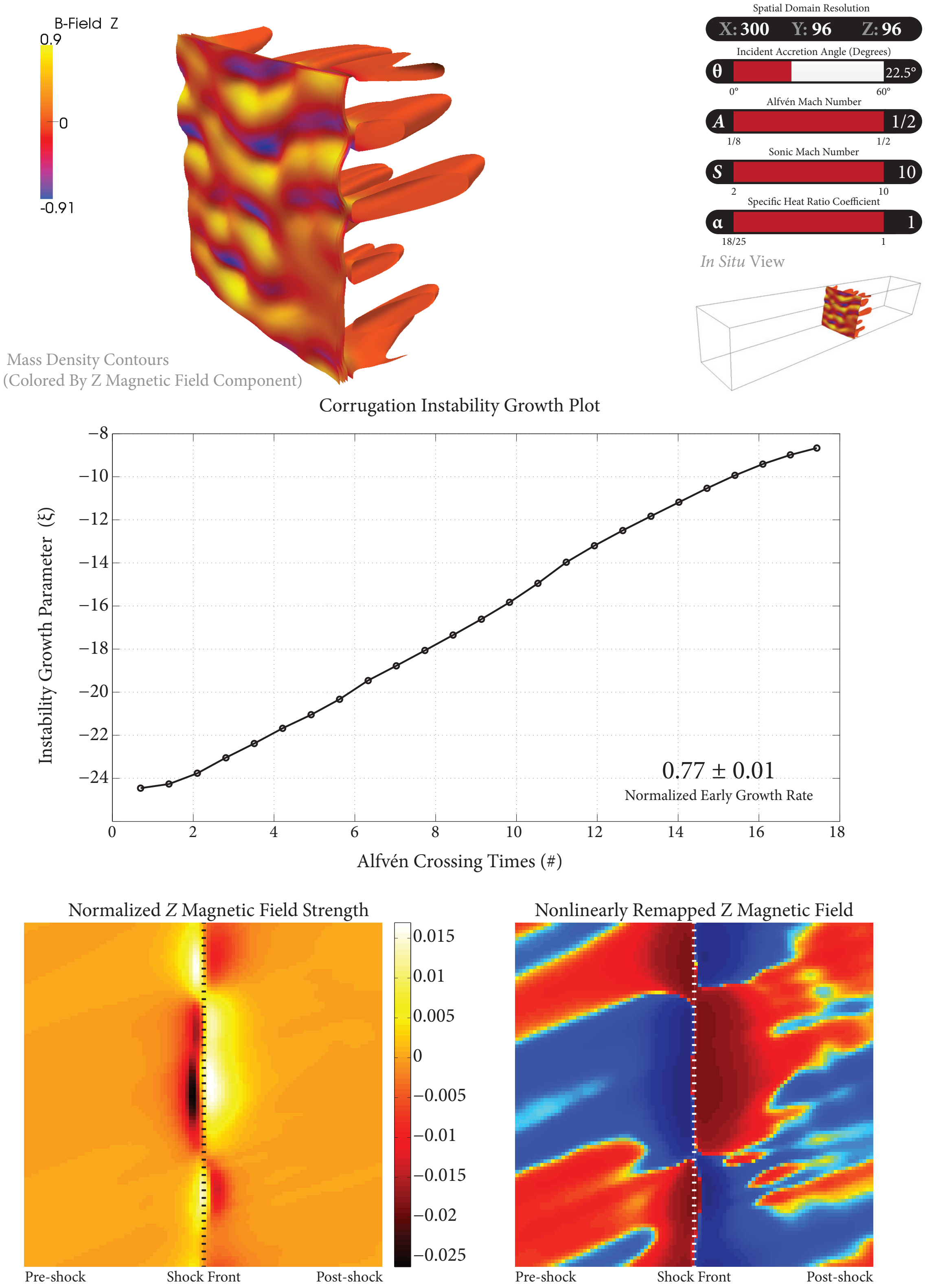


Figure 6.12: Results of the $\{\theta\text{-}22.5° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial for incident angle variation comparison.

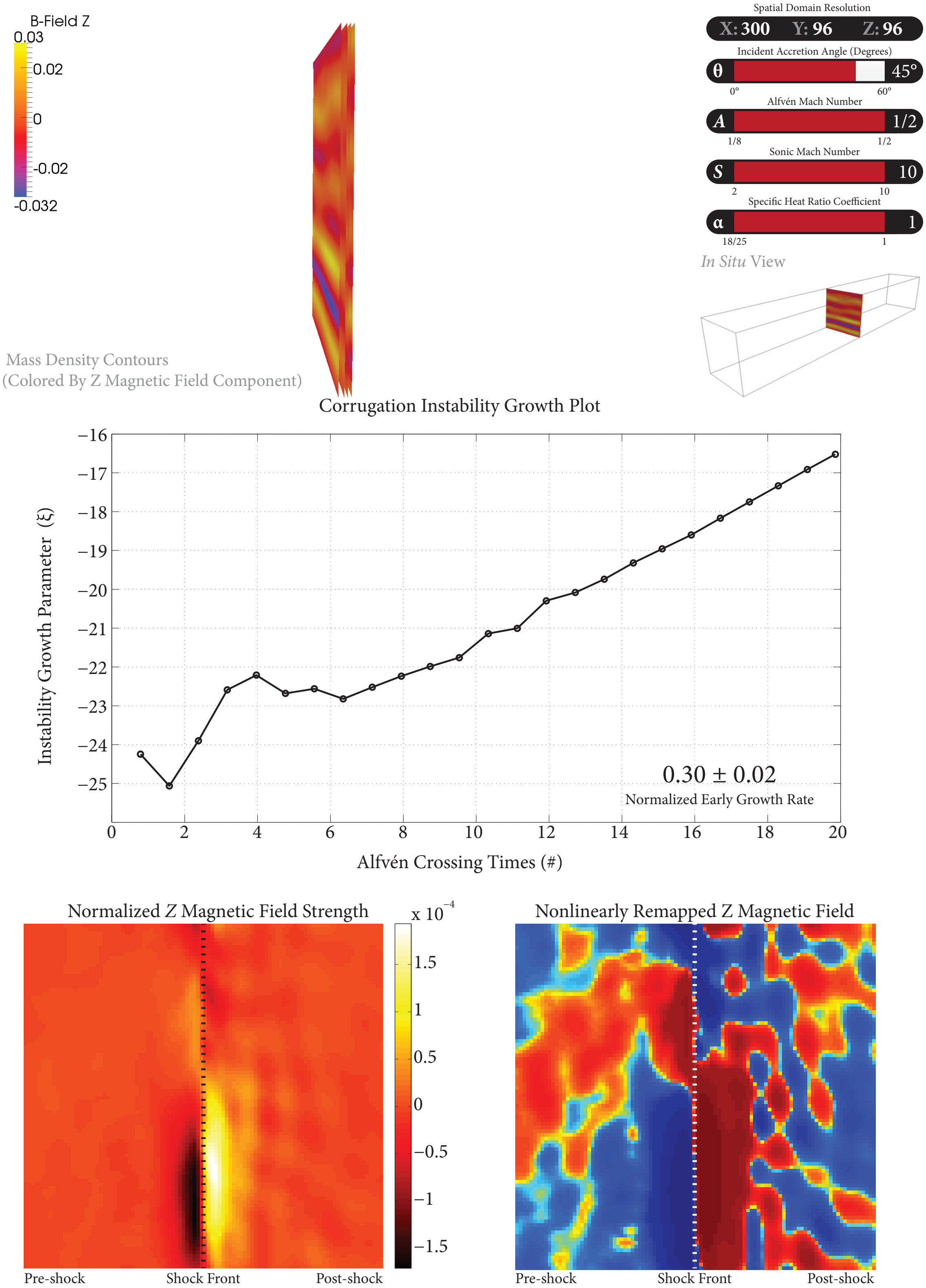


Figure 6.13: Results of the $\{\theta\text{-}45° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial for incident angle variation comparison.

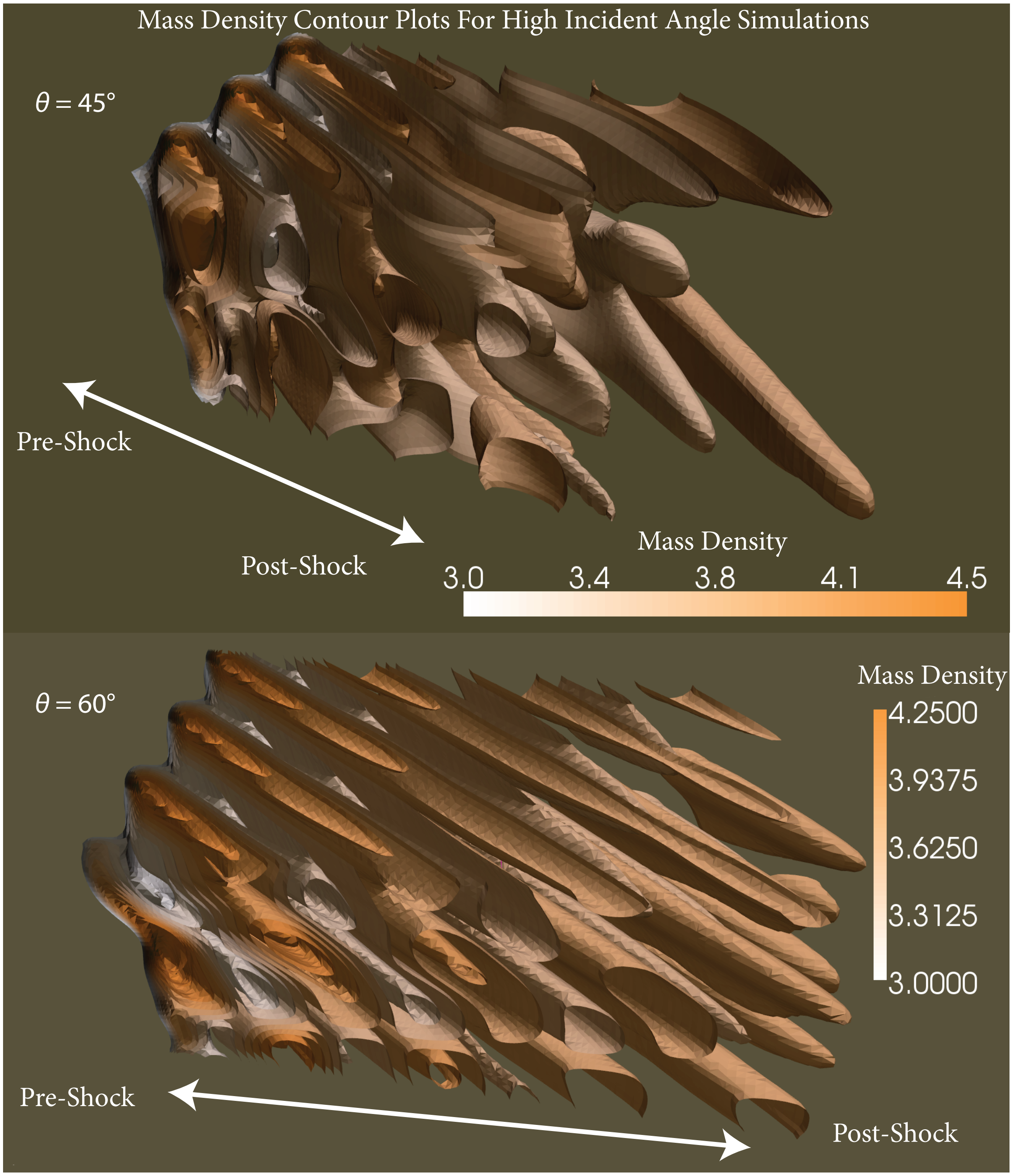


Figure 6.14: Mass density contour plot for a $\{\theta\text{-}45° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ and a $\{\theta\text{-}60° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ at simulation times of between 75-78 Alfvén crossing times run at low resolution. A single spatial frequency clearly defines the corrugation instability unlike lower angles, which are more random in their spatial frequency distribution.

instability that dominates in the nonlinear regime. Until that spatial mode grows, the instability grows more slowly than the lower angles. However, once that frequency reaches its nonlinear phase the growth is surprisingly rapid.

6.3.3. Alfvén Mach Variations

To explore variations in the Alfvén Mach parameter, trials were conducted that mirrored all of the previous incident angle trials replacing the Alfvén Mach number of 1/2 with 1/8, a much stronger magnetic field as shown in Figures 6.5-6.7. Similar testing was done with an Alfvén Mach numbers 1/4 but these have been omitted because they do not exhibit any unique behaviors not found in either the 1/2 or 1/8 trials.

In these trials the growth of the instability was slower than in the 1/2 trials and tended to accelerate once in the nonlinear regime. The parallel shock {θ-0° | A-1/8 | S-10 | α-1} trial, Figure 6.15, appears to be stable unlike all of the other trials. Additional parallel trials run at low resolution for up to 100 Alfvén crossing confirm this stability. However, during these longer tests the instability, which trends negatively for up to 30 Alfvén crossing times after the initial seeding of the noise eventually begins to trend upward. In the 100 Alfvén crossing time tests the upward trend was very small but could suggest instability on very long time scales.

All of the 1/8 Alfvén Mach trials share the behavior of the growth parameter originally trending downward, followed by a resurgence and subsequent growth that is slower, but not unlike their Alfvén Mach of 1/2 counterparts. Another difference between the Alfvén Mach 1/2 and 1/8 trials is that the shearing instability seems to become important at smaller angles. In the Alfvén Mach 1/2 trials the shearing instability did not significantly alter the corrugation instability behavior until incident angles higher than 22.5°, whereas in the 1/8 case the $\{\theta\text{-}22.5° \mid A\text{-}1/8 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial clearly shows a mature combined corrugation and shear behavior. That stronger magnetic fields enhance this behavior suggests that there is something about the transverse magnetic field component that incites the shearing; it is obviously not just a hydrodynamic, Kelvin-Helmholtz type shear instability.

Higher angles in the Alfvén Mach 1/8 trials were also much less stable. They all began with the initial stability exhibited by the low angle trials, but the shearing was so strong once the nonlinear regime was reached that the shock fronts were quickly destroyed by the instability resulting in large amounts of matter being ejected into the pre-shock region, which destabilized and so ended the simulation.

While investigating changes in the Alfvén Mach parameter a trial was also conducted with a very large Alfvén Mach number, corresponding to a small magnetic field. An Alfvén Mach number of 10 was used to place the trial well into the fast shock regime. Figure 6.18 shows the results of that $\{\theta\text{-}10° \mid A\text{-}10 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial.

As expected, in the fast shock regime the accretion shocks are stable as no magnetic waves are able to propagate into the pre-shock region and alter the flow and the magnetic field strength is so small that the initial perturbations to the front are not enough to cause significant rippling in the front and grow an instability. Compared to all of the other trials the amplified z magnetic field plot in Figure 6.18 shows that no magnetic waves were propagating into the pre-shock region and very little magnetic activity occurred at all.

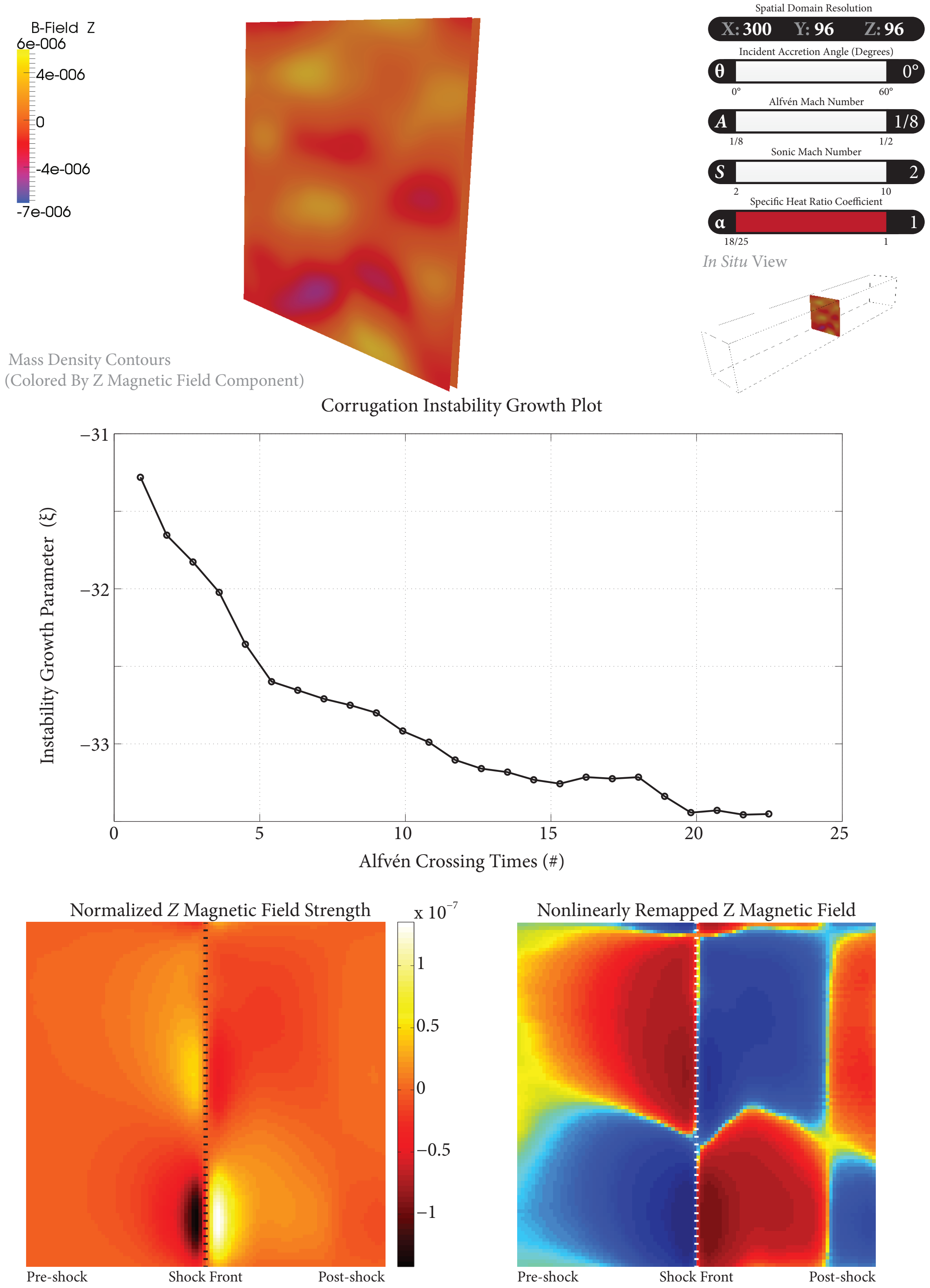


Figure 6.15: Results of the {θ-0° | A-1/8 | S-10 | α-1} trial for Alfvén Mach comparison.

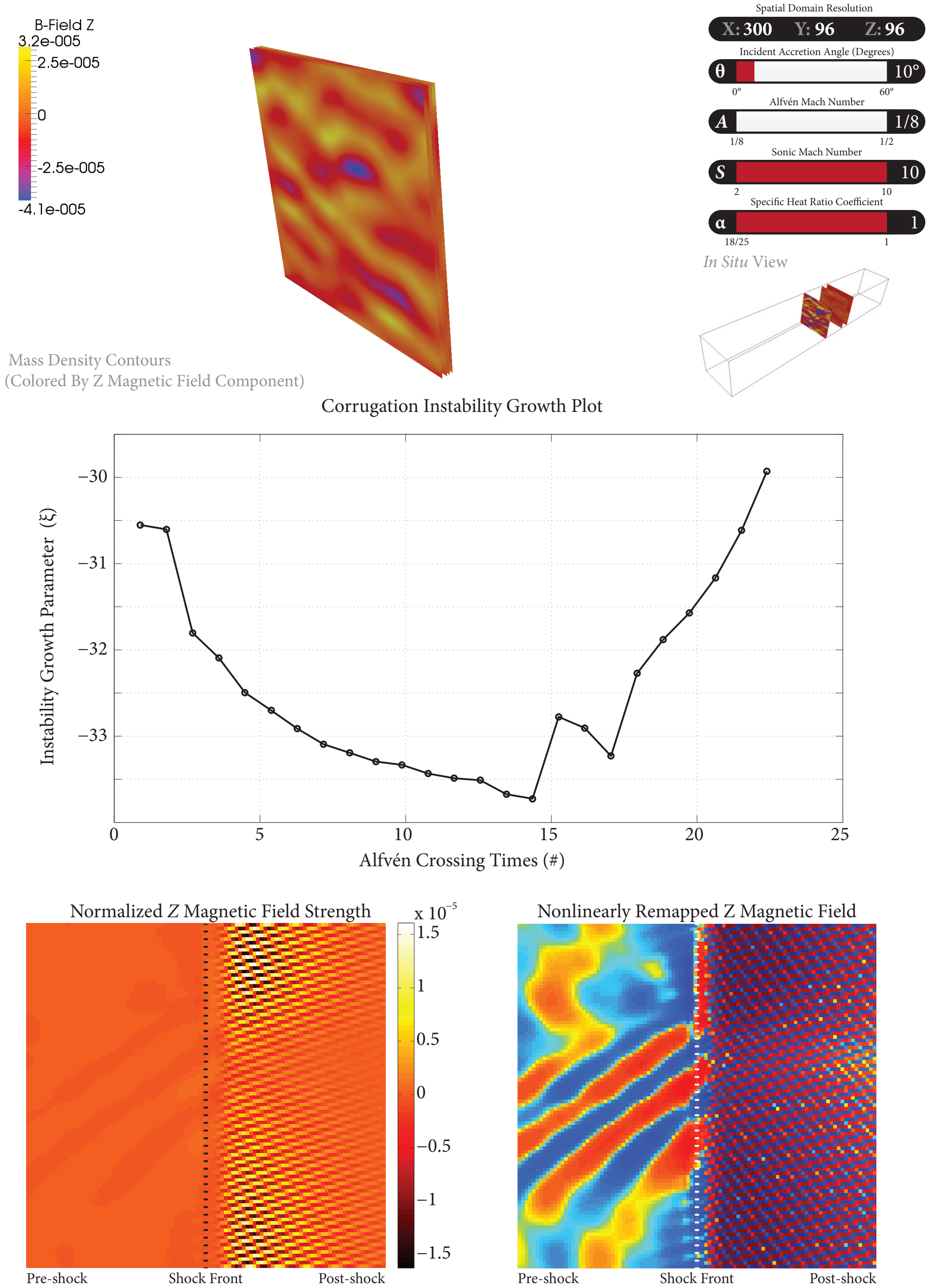


Figure 6.16: Results of the {θ-10° | A-1/8 | S-10 | α-1} trial for Alfvén Mach comparison.

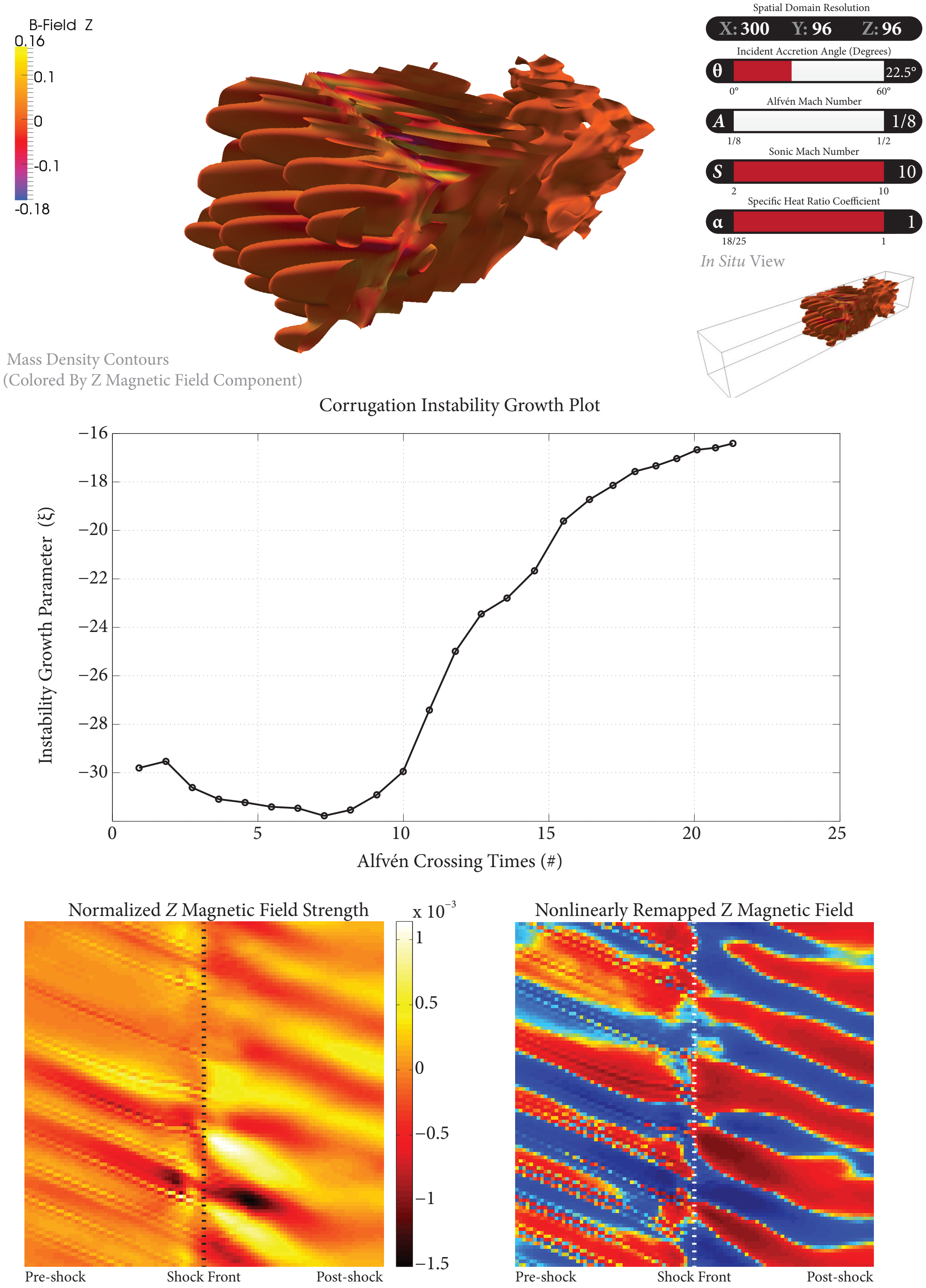


Figure 6.17: Results of the $\{\theta\text{-}22.5° \mid A\text{-}1/8 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial for Alfvén Mach comparison.

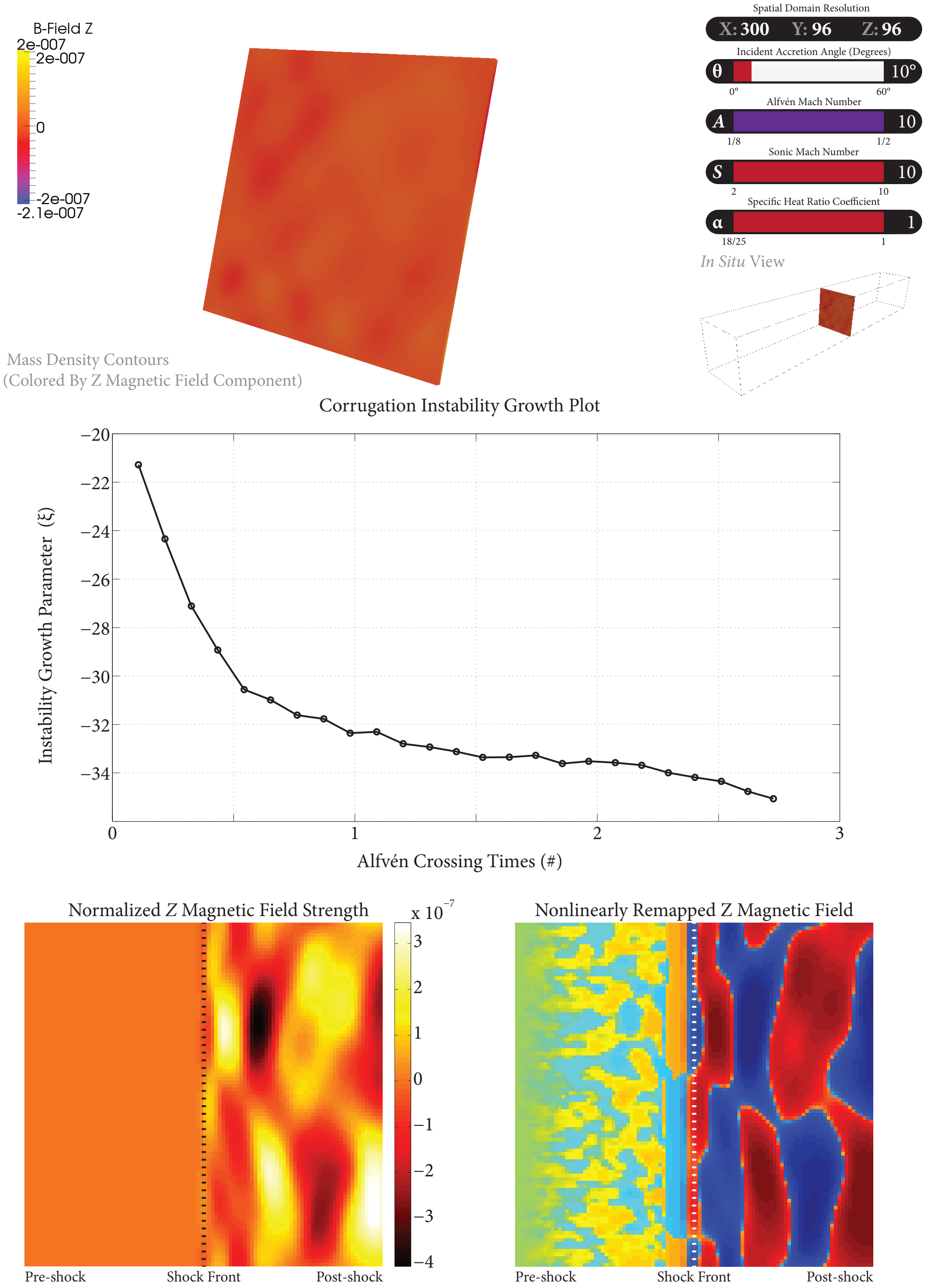


Figure 6.18: Results of the $\{\theta\text{-}10° \mid A\text{-}10 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial for Alfvén Mach comparison.

6.3.4. Specific Heat Ratio Coefficient Variations

The final parameter, the ratio of specific heats coefficient, was tested by conducting trials at values of 1, 21/25, and 18/25, which correspond to specific heat ratios of 5/3, 7/5, and 6/5 respectively. The results of the {θ-10° | A-10 | S-10 | α-18/25} trial are shown in Figure 6.20. Comparatively, higher ratio of specific heats have much faster instability growth, but the lower values do not appear to otherwise alter the terminal behavior of the instability. Figure 16.19 shows the instability growth parameters of the α=1, from Figure 6.3, and α=18/25 trials for comparison.

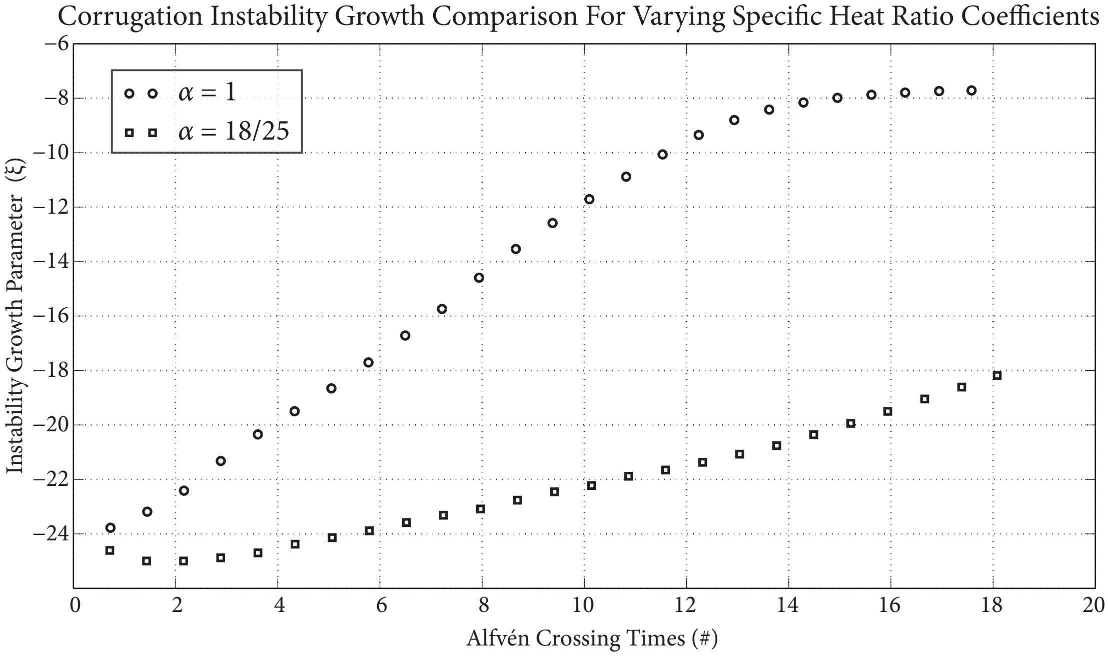


Figure 6.19: Comparison of instability growth evolution for different specific heat ratio coefficients.

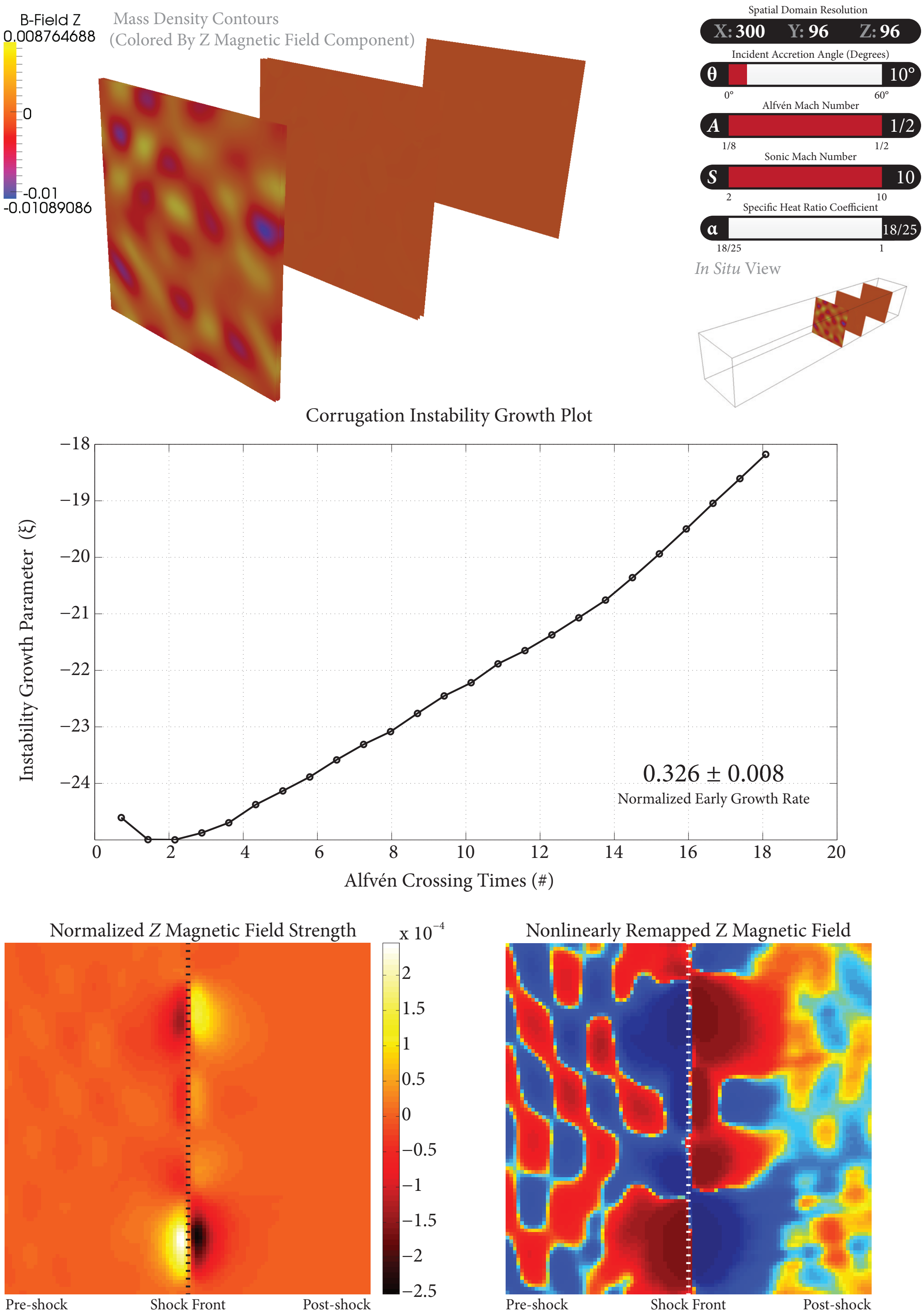


Figure 6.20: Results of the $\{\theta\text{-}10° \mid A\text{-}10 \mid S\text{-}10 \mid \alpha\text{-}18/25\}$ trial for specific heat coefficient ratio variation.

6.3.5. Resolution and Diffusion Variations

Previous and ongoing linear work suggests that higher spatial frequencies dominate the early growth of corrugation instabilities. So as a final investigation identical trials were run at different resolutions, low, medium, and high to see how the growth rates for the instability changed depending on resolution. The standard resolution trial is shown in Figure 6.3, the low resolution trial in Figure 6.4, and the high resolution trial in Figure 6.22. The combined growth parameter plot for these three trials is shown in Figure 6.21 and clearly the higher the resolution, which permits higher spatial frequencies, the faster the growth of the instability.

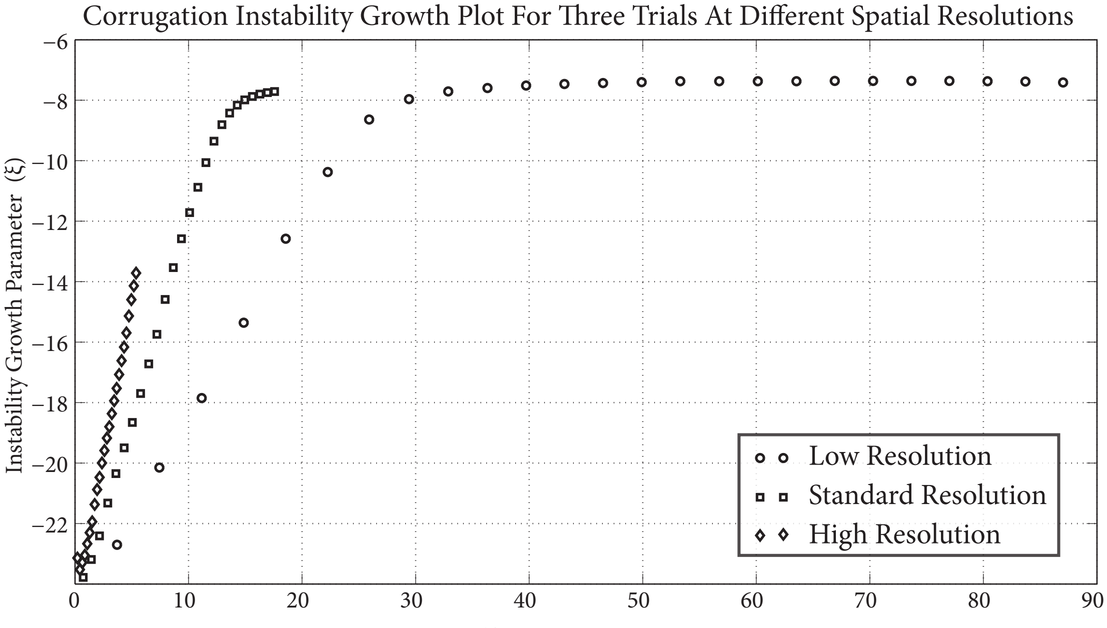


Figure 6.21: Comparison of corrugation instability growth parameters for the same {θ-10° | A-10 | S-10 | α-1} trials at low, standard, and high resolutions. The higher the resolution the faster the instability grows.

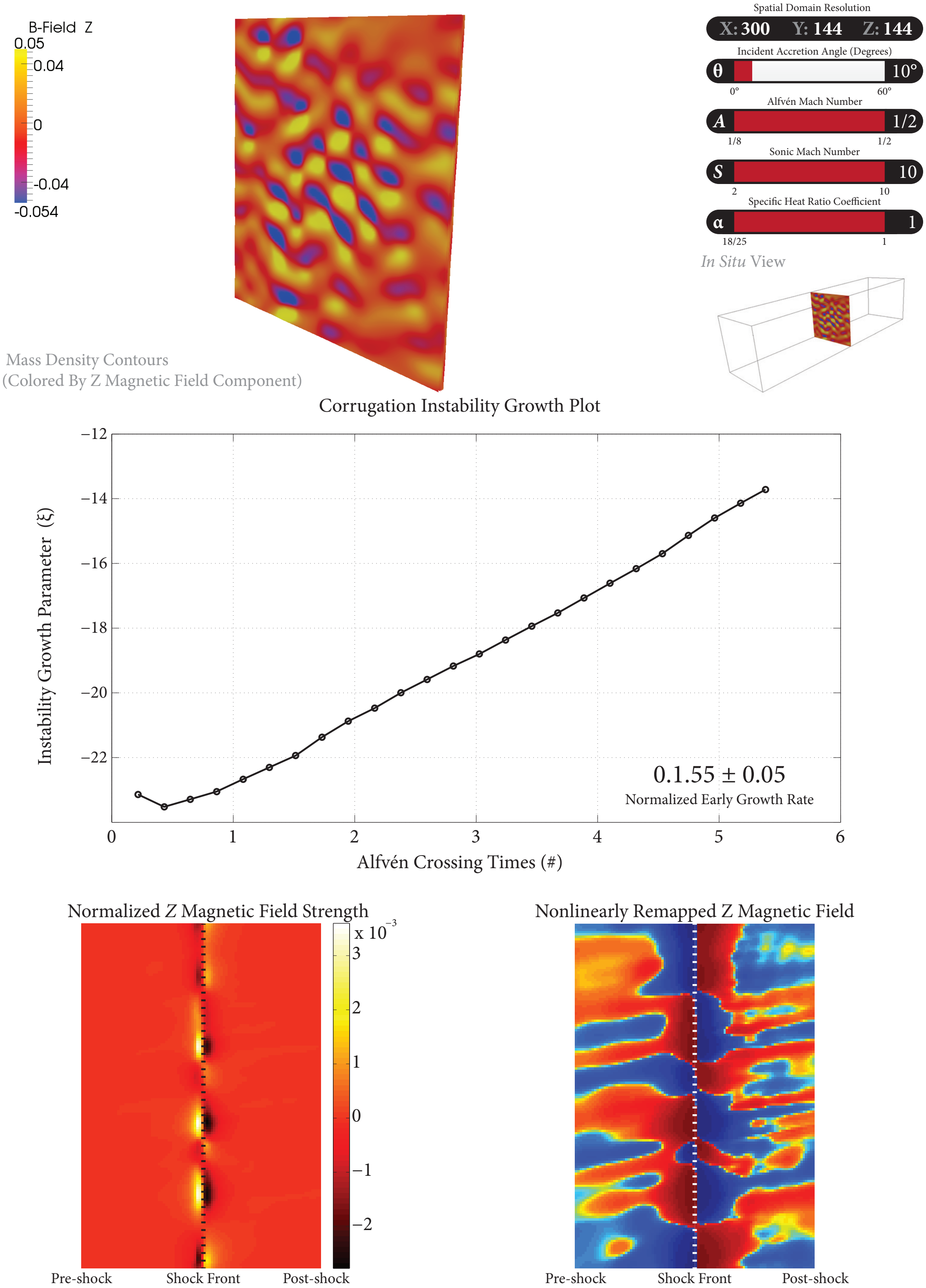


Figure 6.22: Results of the $\{\theta\text{-}10° \mid A\text{-}1/2 \mid S\text{-}10 \mid \alpha\text{-}1\}$ trial at high resolution for the resolution dependent instability growth comparison.

Also, all of the previous trials were conducted using the Van Leer flux limiter, which while it is the most useful in general circumstances has a significant amount of diffusion compared to the Superbee limiter. Therefore, a trial was also run to repeat the original {θ-10° | A-1/2 | S-10 | α-1} trial using the Superbee limiter. The results of that trial were as expected, the same instability behaviors as the original trial but with much faster instability growth than with the Van Leer limiter. The Superbee limiter was not used for all trials because, while it is much less diffusive, it tends to generate the kinds of high density regions that often halt or destroy a simulation of this kind where compression around the shock front is common and unavoidable. Even in the single trial the system could not be evolved for very long before the simulation was brought to a halt by the compression artifacts. There is a possibility that with carefully applied artificial viscosity these issues could be avoided while running with less general diffusion than the Van Leer limiter, but there was never a clear need to go to the extra work to achieve this; the Van Leer limiter was sufficient for the general investigation and the Superbee limiter served its purpose to verify the accelerated growth with higher spatial frequency contributions.

CHAPTER VII

SUMMARY AND FUTURE DIRECTIONS

The IMOGEN magnetohydrodynamics simulation software was successfully implemented as a new computational tool in support of the investigation of instabilities in magnetic accretion shock waves. It passed numerous tests to verify its capabilities and performance, passing them all and demonstrating its viability for use in computational astrophysics research. It should also be noted that IMOGEN has since been turned into a general purpose magnetohydrodynamics simulation software and is currently being used in support of investigations of bow shocks, accretion disks, planetary migration, and other domains.

The magnetic accretion shock wave investigation revealed that strong, slow magnetic accretion shock waves are generally unstable in the nonlinear regime except for the possible case of very high magnetic field strength, i.e. low Alfvén Mach number, and very small incident angle. The evolutionary behavior of the instabilities in the front was also found to be dependent upon all of the parameters investigated.

Variations in the incident angle parameter revealed two distinct behavioral regimes, corrugation and corrugation plus shear instabilities, which lead to different nonlinear terminal states for the shock front. Sonic Mach number variations confirmed that strong shocks have faster instability growth than shocks in the slow regime, but that differences within the strong shock regime have negligible impact on the instability growth. The Alfvén Mach parameter investigation found the strength of the magnetic field plays a crucial role in defining the growth behavior as well as contributing to the incident angle at which the shearing corrugation behavior takes effect. Finally, the ratio of specific heats was found to impact the growth rates, with the monotonic, γ=5/3, case growing faster than cooler fluids with greater internal degrees of freedom.

Based on these results it’s possible to speculate on the behavioral variability in corrugation instabilities effecting the evolution of the many astrophysical systems in which magnetic accretion shocks are likely found. The fast aperiodic growth of corrugation instabilities at small incident angles is notably different than the more turbulent but slower growth above the critical angle. Hence, there are likely significant differences between otherwise similar systems in the presence of the necessary physical drivers, such as rapid rotation or larger scale magnetic fields that force specific alignment and containment geometries, that would tend to increase or decrease the incident angle of the accretion column.

The next logical extension to this work is to add radiative cooling, a non-conservative term that could potentially mediate the growth of corrugation instabilities in magnetic accretion shock waves and might stabilize parts of the explored parameter space. Initial work has already begun in this respect. Parameterized, optically thin cooling has been added to IMOGEN and a few trials already completed with weak radiation coefficients. However, the results are too preliminary to be included here.

APPENDIX A

IMOGEN SIMULATION SOFTWARE DESCRIPTION

In the last few years, interpreted computer languages, i.e. ones that are not compiled for a specific computer architecture prior to execution, have become fairly popular among computational scientists, including numerous uses in high-performance computing applications. The maturity of these interpreted languages, along with general improvements in their run-time performance and just-in-time compiling, have significantly narrowed the performance gap between interpreted and compiled applications (Cai, et al., 2005; Choy & Edelman, 2005; Luszczek, 2009; Mignone, et al., 2007; Ousterhout, 1998; Perez & Granger, 2007; Sharma & Martin, 2009).

Despite these improvements, interpreted languages still lag behind compiled ones in raw performance for general cases, but the lag is reduced, becoming reasonable, for the kinds of data-centric, memory intensive operations that apply to a large class of high-performance computing applications. In large, data intensive operations where computation density is high, i.e. the ratio of computational activity to lines of code is large, performance losses due to the interpretation of code are greatly diminished. In combination with scalable parallel deployment, and the significantly larger and more complex memory usages that accompany it, the more advanced communication and memory management functionality of interpreted languages leads to performances that are competitive with their compiled brethren. With an increasing availability of multi-core, parallel hardware, along with serious development trends in interpreted language based, massively parallel server-side software for both scientific and general computing, it is reasonable to conclude that interpreted applications now have a place in high-performance scientific computing.

A.1. Interpreted Scientific Computing

The argument for using interpreted languages in scientific applications is strengthened when considering more than just performance in the evaluation of interpreted versus compiled languages. Interpreted languages offer a number of benefits over their compiled brethren that are directly applicable to computation in the sciences (Boehm, 2006; Heroux & Willenbring, 2009; Ousterhout, 1998).

A.1.1. Productivity

Applications developed with interpreted languages typically require 10-25x fewer lines of code than an equivalent compiled application due to a high-level structure and language constructs that offer greater degrees of implicit computational activity. For efficient and scalable parallel development this value jumps by an order of magnitude given the added complexities involved in handling distributed computation, communication, and memory.

Have a more concise code base provides important development benefits such as rapid prototyping, flexible extension and modification, and a reduction in the maintenance tasks required to keep a code up to date as the software evolves and new algorithms and hardware become available.

A.1.2. Accessibility

An important disadvantage of using compiled computer languages is that their low-level framework is an impediment to accessible reading and writing. Compiled languages build on primitive data types, language constructs, and semantics that, while are largely responsible for greater performance over interpreted languages, result in opaque syntaxes that require greater expertise to author and greater effort to read and comprehend. Clarity becomes an even greater issue during optimization, where utilizing less intuitive semantics is often an effective means of improving performance.

Interpreted languages, on the other hand, utilize a higher-level framework with more intuitive, less primitive data types and different, less opaque optimization techniques. The result is a cleaner code base that requires less effort and expertise in development and use. All of which lead to software with greater accessibility, which is useful in scientific computing where the lifetime of an application is often many developer-generations long, turnover and training of new students is an involved and lengthy process, and efficient communication fosters effective, successful collaborations.

A.1.3. Portability

The key difference between interpreted and compiled languages, and the reason for their respective names, is how they are deployed. Compiled languages must be, as their name implies, compiled to a specific platform prior to use. This means that complex coding often requires platform specific design and optimization, making ports between platforms arduous and time consuming. Given the recent growth and changes in advanced parallel platforms, e.g. stream processing accelerated architectures and shifts toward resource fat computational nodes, as well as the increasing diversity the available supercomputing platforms, ease in deployment portability is of growing concern for the scientific computing community.

Interpreted language based applications are, in contrast, largely platform agnostic because the language interpreter, also referred to as the virtual machine, contains the platform specific components. Therefore, interpreted software will deploy to any platform that its interpreter supports. For widely adopted interpreted languages these interpreters are rigorously maintained for the majority of new and important platforms. In scientific computing, where access to large hardware resources varies, collaborators deploy to different environments, and the need to keep up with new and changing platforms is necessary to advancing research goals, interpreted languages offer clear advantages over compiled ones.

A.1.4. Optimization

Compared to industrial computing, which has a much larger pool of development resources and expertise, scientific computing applications are often poorly optimized. The small usage base for scientific applications, their research specific employment, and the difficulty of maintaining adequate funding for their ongoing development make any language-inherent optimizations important. Here the value of interpreted languages is that a significant amount of optimization comes from the interpreter and shared libraries.

As such, the performance of an interpreted application can grow over time without any optimization work from the application developer. These shared libraries and interpreters are typically maintained and optimized by collaborations between robust open-source communities and large industrial entities with the resources and expertise that most scientific developers lack. In contrast, compiled-language software in the sciences tends to stagnate as much more of the implementation and optimization are left to the application developer.

A.2. Computational Approach

Mindful of the rapid improvements in, and benefits of, interpreted languages, the astrophysical simulation software tool, IMOGEN, was developed in an interpreted language framework, and built using software engineering techniques more commonly employed in successful, scalable industrial applications with the intention of producing a more useful and valuable tool for the astrophysics research community. In the fall of 2010, IMOGEN reached its first fully stable and tested build point, at which time it was opened to wide development and usage as a publicly available and managed open-source project. The details and further information on the IMOGEN open-source project is available at http://www.imogenproject.org.

While the short goal for developing IMOGEN has been to support the research contained within this work, the larger goal has been to create a generally useful tool for the computational astrophysics community while broadening the historical focus in computational magnetohydrodynamics on performance to include greater attention to rapid extensibility and modular design. IMOGEN, by design, eases the modification and development of new physics and numerical approaches, maintains a concise code base with clear documentation and accessibility tools, facilitates improved collaboration in both development and usage, and integrates interactive analysis and visualization packages to make data analysis more efficient and effective and reduce the need for endless amounts of data storage.

To facilitate the achievement of the broader goals, IMOGEN has been created with a layered complexity design with four distinct layers, as shown in Figure A.1, that abstracts the opaque aspects of the software away from regular research usage and development in order to streamline common research applications. A research developer describes a researcher with only a limited background in computer science that develops primarily for the purpose of extending the physics or basic functionality of IMOGEN in direct support of specific research goals. This is in contrast to the more computationally adept application developers that are responsible for the advancement, expansion, and improvement in functionality of IMOGEN to support more broadly applicable research goals and trends.

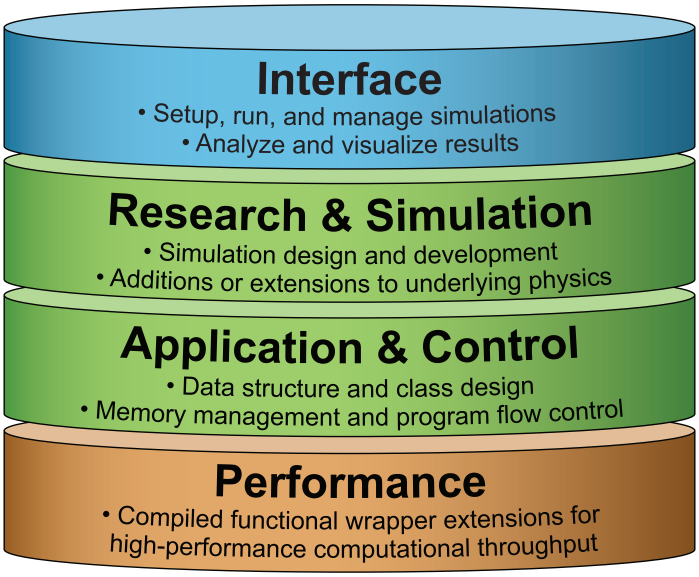

Interface
• Setup, run, and manage simulations
• Analyze and visualize results
Research & Simulation
• Simulation design and development
• Additions or extensions to underlying physics
Application & Control
• Data structure and class design
• Memory management and program flow control
Performance
• Compiled functional wrapper extensions for high-performance computational throughput

Figure A.1: Design diagram for the complexity based layering of IMOGEN.

Each layer, from bottom up, is structured to encapsulate certain functionality and reduce the exposed complexity in the layers above it to provide a development and usage environment conducive to research within groups of diverse skill levels.

A.2.1. The Interface Layer

The interface layer exposes all of the operational functionality for IMOGEN, including simulation design and execution, data input, output and format translation, and supporting analysis and visualization features. It is implemented as a dynamic, object-based library that presents to users as a simple scripting interface. The simple scripting approach allows researchers, recognizing the large number of astrophysicists that need access to simulation software, to use IMOGEN without much familiarity with its language or the computational background and time needed to learn its data structures and program design. The scripting approach also makes it possible for IMOGEN to function either interactively or by invoking predefined execution scripts. In the interactive mode, users may visualize any simulation data, including intermediate data that would be too expensive to store. This is a useful feature when first exploring new models as well as when trying to ascertain the cause of unexpected physical or numerical behaviors of a simulation.

To keep the interface scripting as simple as possible, a translation routine was developed that effectively decouples the interface from the core classes and data structures used during execution. By translating interface objects instead of passing them directly, interface scripting can be dynamic, include simulation specific defaults, use advanced error checking and handling, and make use of heterogeneous initialization software and techniques all completely abstracted from the IMOGEN core.

The visualization aspects are handled by format translation routines that take IMOGEN simulation data, regularly stored in compact, HDF5 (Hierarchical Data Format version 5) format, to common visualization formats, notably VTK (Visualization Tool Kit) and MA/MB (Autodesk Maya ASCII & Binary), for use with well-established visualization tools.

A.2.2. The Research & Simulation Layer

The research and simulation layer contains all of the astrophysical functionality within IMOGEN and was designed with flexibility for modification and extension of the underlying simulation physics in mind.

Unlike the interface layer, the application layer elements interact directly with the IMOGEN core, a necessity for performance considerations, but the layer implementation is such that exposure to complex data structures are limited as much as possible both by the interpreted framework and by explicit suppression of unneeded elements within the data structures.

Within the application layer also resides the actual simulation initialization framework, which is responsible for converting the scripts of the interface layer into the initial conditions and corresponding settings used to drive a simulation. As such, researchers can easily design, create, and extend simulations without the need for developing separate algorithms and data format translations in order to prepare a simulation.

A.2.3. The Application & Control Layer

The application and control layer is the structural core of IMOGEN, where program flow, numerical algorithms, data structures, and classes reside. Like its preceding layers, this layer is implemented entirely in an interpreted framework, utilizing the benefits of interpreted languages to achieve efficient, scalable performance in a wide range of parallel deployment environments with minimum effort.

It is within this layer that the numerical techniques described in chapter 3 are implemented, and as they are fully encapsulated within this layer, a research can modify or extend the physical model and design new simulations without the knowledge of how these numerical methods carry out the simulation.

A.2.4. The Performance Layer

Yet another benefit of most interpreted languages is the ability to use compiled-language elements to extend them, wrapping compiled functions, algorithms, and data structures into the interpreted framework. Hence, the performance layer is a collection of low-level compiled extensions to IMOGEN that boost performance in the cases where the performance of an interpreted language would be poor. Performance layer elements are chosen carefully using regular performance benchmarks because they introduce undesirable complexity into the code base. However, a benefit of developing these elements beneath the application and control layer is that they do not require the more complex data structures and memory management elements that would be necessary in a purely compiled implementation.

A.3. Design & Implementation

At the highest level IMOGEN is broken down into two largely independent systems. The first is the simulation definition, modification, and specification (SDMS) system and the second is the simulation execution and operation (SEO) system. IMOGEN is invoked using simulation run scripts that are created using the SDMS system as the preparatory step for a particular simulation run. The last line of any simulation run script is the call to the main IMOGEN method that invokes the SEO system, translating and passing the simulation definition to the SEO for execution.

A.3.1. Initializers

Every IMOGEN simulation is built around a simulation-specific Initializer object, which is responsible for taking the basic input from a user and converting that into the much more complex data necessary needed to drive the simulation. Simulation Initializers all subclass an abstract Initializer base class that encapsulates universal simulation settings and exposes the general functionality for converting those settings into the IMOGEN core data structures. Once an Initializer has been created, a user may adjust any number of its settings to suit the needs of a particular simulation. When the user is satisfied with the various settings, the Initializer is passed into the main entry method for IMOGEN, which requests the core data structures from the Initializer and then discards it before beginning the simulation.

One of the benefits of the Initializer is that they are designed to auto-fill settings on a per simulation basis. Many magnetohydrodynamic simulation tools require one or more configuration files for the array of settings used to drive a simulation. The problem with this approach is that as the complexity of the software increases, so do the number of settings available to be modified. The result are large configuration files with endless settings, only a few of which are regularly modified when updating a run. This makes it difficult to manage and compare collections of runs and exposes users of the software immediately to the complexity of the system and a need to understand all of its initialization settings. Initializers, bury these settings by intelligently auto-generating all values that are not specified explicitly by the user in the simulation run script.

This approach significantly reduces the size and complexity of simulation run scripts and allows the developer of a simulation to prescribe ideal defaults not only as static values, but dynamically based on related settings that may be interdependent. As there are many groupings of interdependent settings, this is highly beneficial allowing users to control simulations from only a few settings instead of many.

There are, however, groups of settings that are so tightly integrated and numerous that they warrant special consideration. Examples include gravity, radiation, and artificial viscosity. For these more sophisticated groupings, the Initializer concept was extended with a new class of SubInitializers. SubInitializers are Initializers within Initializers and are specific to one of the aforementioned interdependent grouping of settings. When an Initializer is instantiated, the necessary SubInitializers are also instantiated and exposed for setting modification through the Initializer directly. SubInitializers behave in a similar fashion to Initializers, except that they are only invoked by the Initializer that owns them instead of the user or the IMOGEN core.

Developing a new type of simulation requires the creation of a new Inititalizer class. This is achieved simply by extending the Initializer class or one of its existing subclasses and modifying it in the following ways. The new Initializer class must define all settings that will be unique to that simulation type. The constructor for the new Initializer class must auto-populate reasonable defaults to these new properties, or the properties must be defined dynamically through getters and setters. Finally, the private method that generates the initial conditions based on the simulation settings must be overridden and modified to suit the particular Initializer. All other core data structures are generated by the Initializer class and do not need to be modified for a new simulation type.

Once the new Initializer class has been setup in this way, it is immediately ready for use in simulation run scripts. No other development is required unless previously unsupported physics is being introduced for the new simulation type.

A.3.2. Simulation Run Scripting

As previously mentioned Initializers are used in simulation run scripts, which are modified by the user on a per simulation trial basis and represent an identifiable fingerprint for any simulation. For this reason IMOGEN saves a copy of any simulation run script in its associated results folder for users for their future reference. The scripting system is simple and exposes all of the desired physical attributes associated with a particular simulation type through the Initializer object created within the script. An example script is shown in Figure A.2, for the specification of a corrugation instability simulation the results which will be discussed later.

```
01  % Create the Initializer object:
02  run             = CorrugationShockInitializer([1200 48 48]);
03
04  % Set max number of iterations:
05  run.iterMax     = 25000;
06
07  % Set max run time in hours:
08  run.wallMax     = 320;
09
10  % Set Percentages between snapshot saves for 2D and 3D data:
11  run.ppSave.dim2 = 0.5;
12  run.ppSave.dim3 = 4;
13
14  % Corrugation instability problem settings:
15  run.theta       = 10;
16  run.sonicMach   = 10;
17  run.alfvenMach  = 0.5;
18
19  % Information about the run:
20  run.alias       = 'Athena';
21  run.info        = sprintf(['Corrugation instability test [Th=10, Ms=10, Ma=0.5]'
22                            'with grid [1200, 48, 48]']);
23  run.notes       = '4x length rerun of Prometheus.';
24
25  % Create initial conditions and pass them into the main IMOGEN method:
26  [mass, mom, ener, magnet, statics, ini] = run.getInitialConditions();
27  imogen(mass, mom, ener, magnet, ini, statics);
```

Figure A.2: Example IMOGEN run simulation script that defines a corrugation instability simulation run.

The example simulation run script begins on line 2 with the creation of a corrugation instability Initializer object with a three-dimensional grid specified by 1200 cells in the X dimension and 48 cells in each of the Y and Z dimensions. The actual length of this grid is determined within the Initializer transparently to the user to maintain the desired normalization for the simulation type as needed for running simulations at many different resolutions.

The next section of the simulation script is where all of the parameters are set. Notice that for this simulation very few parameters were needed because the Initializer object takes care of setting sensible defaults for a corrugation instability simulation. In this particular case, the only physical properties set for this run were the theta value, incident angle of the accreting matter and the sonic (hyrdodynamic) and Alfven (magnetic) Mach numbers that are used to determine the initial velocity and strength of the magnetic field in the pre-shock regions of the instability run.

Once all of the properties are set the Initializer converts the input into initial conditions for the primary arrays and those are passed into the main IMOGEN method to begin the execution of the simulation.

A.3.3. Simulation Management

During simulation the associated states and properties are managed and maintained by a collection of manager classes. Each manager class is responsible for a specific aspect of the simulation operation and interacts with the other manager classes as necessary to provide the complete operational environment for a simulation. Manager classes are built using a dynamic decorator pattern such that functionality is completely and transparently extensible during simulation.

The idea is that a specific manager is constructed as a base class with the common properties associated with any simulation and with a set of method hooks that can be reassociated during initialization with any method that fits the appropriate signature. Combined with the flexibility of interpreted languages, where methods are also handled objects, this allows for the easy dynamic construction of a manager to fit any simulation need. Of course, this approach would be unnecessary in an environment where all developers had a clear understanding of the architecture behind IMOGEN, but that would be unrealistic given the goals of its development.

Instead, this approach has been utilized to allow simulation developers to modify a piece of IMOGEN easily without having any understanding of its larger structure. This allows for rapid development and testing of new and different physics without having to change any aspect of the IMOGEN core. The IMOGEN package structure is setup in such a way that each manager has their own package. Within that package are separate functions, not connected directly to a class, that when compiled through the interpreted compiler become method objects that can be assigned to the method hooks of their manager.

Therefore, a simulation developer that, for example, wants to try a different radiation model, or a special kind of artificial viscosity for a particular simulation need only create an independent function in that package and then associate that function with an enumeration in the Initializer, so that during the initialization process that function gets compiled into the manger class and becomes part of the SEO core.

APPENDIX B

DYNAMICALLY-SCALED VISUALIZATION ALGORITHM

During the investigation of the magnetic wave properties of the corrugation instability it was difficult to visualize the waves directly because of the multi-scale nature of the problem. In a typical unstable simulation waves would be generated with z magnetic field component values that varied by up to seven orders of magnitude. In this case it was important to look at all scales because the wave length and nature were of interest, not just the amplitude.

To visualize the waves using a generally applicable visualization technique, a new remapping algorithm was developed to treat the data in a logarithmic fashion without the sign and discontinuity problems associated with using logarithmic function directly. The algorithm developed works as follows.

The spatial array, in this case the z magnetic field component (z velocity was also used but never included in the final results), was first normalized. Frequency data was then generated by binning the spatial array according to the order of magnitude for each value, followed by normalizing this frequency data so that it summed to 1. The frequency bins were set to a logarithmic scale in the range $[10^{-10},1]$, which is why the initial normalization of the spatial array was necessary.

Hermite cubic interpolation was then applied to remap the data to a single order of magnitude color space in the range [-1, 1]. The remapping was done such that the higher the frequency of a bin the more space it received in the output color space. This would accentuate orders of magnitude where there was lots of activity and negate orders of magnitude where little or nothing was going on. An example of the process is shown as Matlab-styled pseudo-code in Figure B.1.

```
% For a source array, a.
% Normalize the input array
a      = a./max(abs(a));

% Generate bins for the frequency function.
levels = logspace(-10,0,11);

% Generates frequency data for the spatial array for the given
% levels bin parameter.
freq    = frequencyCount(a, levels);

% Apply the levels and frequencies symmetrically over the range [-1,1].
freq    = [flipdim(freq,2), 0, freq];
levels  = [-flipdim(levels,2), 0, levels];

% Normalize the frequencies for application to the color space.
freq    = freq/sum(freq);

% Generate the color values based on the frequency data.
colorValues(1) = -1;
delta     = freq(1);
for i=2:length(freq)
    colorValues(i) = colorValues(i-1) + delta + freq(i);
    delta = freq(i);
end

% Remap the original array, a, to the new color space.
a = interpolate(levels, colorValues, a);
```

Figure B.1: Pseudo-code example of how the nonlinear, dynamic-scale color profile algorithm works to visualize multi-scale, high-dynamic range data for inspecting magnetic wave properties in the corrugation shock wave investigation.

REFERENCES CITED

Arminjon, P. & Touma, R. 2005, J Comput. Phys., 204, 737

Balsara, D. S. & Kim, J. 2004, ApJ, 602, 1079

Blondin, J. M., Mezzacappa, A., & DeMarino, C. 2003, ApJ, 584, 971

Boehm, B. 2006, P. Int. Con. Software Eng., 12

Burwitz, V., Haberl, F., Neuhauser, R., Predehl, P., Trumper, J., & Zavlin, V. E. 2003, Astron. Astrophys., 399, 1109

Boyd, T. J. M. & Sanderson, J. J. 2003, The Physics of Plasmas (Cambridge, UK: Cambridge University Press)

Brio, M. & Wu, C. C. 1985, J Comput. Phys., 75, 400

Cai, X., Langtangen, H.P. & Moe, H. 2005, Sci. Prog., 13, 31

Chalabi, A. & Qiu, Y. 2000, J. Sci. Comput., 15, 395

Choy, R. & Edelman, A. 2005, P. IEEE, 93, 331

Dai, W. & Woodward, P.R. 1998, ApJ, 494, 317

Donati1, J., Paletou1, F., Bouvier, J., & Ferreira, J. 2005, Nature, 438, 466

Edelman, M. A. 1990, Astrophysics, 31, 758

Evans, C.R. & Hawley, J.F. 1988, ApJ, 332, 659

Fryxell, B., et al. 2000, ApJ Suppl. S., 131, 273

Gardiner, T. A. & Stone, J. M. 2005, J. Comput. Phys., 205, 509

Gardiner, T. A. & Stone, J. M. 2008, J. Comput. Phys., 227, 4123

Gardiner, T. A., Stone, J. M., Teuben, P., Hawley, J., & Simon, J. B. 2008, ApJ Suppl. S., 178, 137

Girart, J. M., Ramprasad, R., & Marrone, D. P. 2006, Nature, 313, 812

Hartmann, L. 2007, Accretion Processes in Star Formation (Cambridge, UK: Cambridge University Press)

Heroux, M., & Willenbring, J. 2009, ICSE Workshop Software Eng. Comput. Sci. Eng., 15

Jin, S. & Xin, Z. 1995, Commun. Pur. Appl. Math, 48, 235

King, A. R. 1988, Q. J. R. Astron. Soc., 29, 1

Lamb, F. K., Pethick, C. J., & Pines, D. 1973, ApJ, 184, 271

Lamb, D. Q. & Masters, A. R. 1975, ApJ, 234, 117

Lax, P. D. & Wendroff, B. 1964, Commun. Pure Appl. Math., 17, 381

Lesson, M. & Deshpande, N. V. 1967, Plasma Phys., 1, 463

Liu, G. R. & Liu, M. B. 2003, Smoothed Particle Hydrodynamics (New Jersey: World Scientific)

Londrillo, P. & Del Zanna, L. 2004, J. Comput. Phys., 195, 17

Luszczek, P. 2009, Int. J. High. Perform. C., 23, 277

Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., & Zanni, C. 2007, ApJ Suppl. S., 170, 228

Orszag, S. A. & Tang, C. 1979, J Fluid Mech., 90, 129

Ousterhout, J. K. 1998, Computer, 31, 23

Pen, U. 1998, ApJ Suppl. S., 115, 19

Pen, U., Arras, P., & Wong, S. K. 2003, ApJ Suppl. S., 149, 447

Pérez, G. & Granger, B. E. 2007, Comput. Sci. Eng., 9, 21

Ryu, D., Miniati F., Jones, T. W., & Frank, A. 1998, ApJ, 509, 244

Ryu, D. & Jones, T. W. 1995, ApJ, 442, 228

Schroll, H.J. 2002, J. Sci. Comput., 17, 599

Schroll, H.J. 2004, J. Sci. Comput., 21, 251

Sharma, G. & Martin, J. 2009, Int. J. Parallel Prog., 37, 3

Shu, F. H., Shang, H., Glassgold. A. E., & Lee, T. 1997, Nature, 277, 1475

Sod, G. A. 1978, J. Comput. Phys., 27, 1

Stone, J. M., Hawley, J. F., Evans, C. R., & Norman, M. L. 1992, ApJ, 388, 415

Stone, J. M. & Edelman, M. 1995, ApJ, 454, 182

Sweby, P. K. 1984, SIAM Num. Analysis, 995

Tajima, T. 2004, Computational Plasma Physics (Reading, MA: Addison-Wesley)

Tóth, G. 2000, J. Comput. Phys., 161, 605

Touma, R. & Arminjon, P. 2006, J. Comput. Phys., 212, 617

Toro, E. F. 1999, Riemann Solvers and Numerical Methods for Fluid Dynamics (Berlin: Springer)

Trac, H. & Pen, U. 2003, Publ. Astron. Soc. Pac., 115, 303

Trac, H. & Pen, U. 2004, New Astron., 9, 443

Van Albada, G. D., Van Leer, B., & Roberts, W. W. 1982, Astron. Astrophys., 108, 76

Van Leer, B. 1974, J. Comput. Phys., 14, 361

Van Leer, B. 1979, J. Comput. Phys., 32, 101

Wesseling, P. 2001, Principles of Computational Fluid Dynamics (Berlin: Springer)

Ziegler, U. 2004, J. Comput. Phys., 196, 393