

Study of Magnetic Relaxation in Plasmas
Using a Parallel Implicit MHD Solver

by

Ogden S. Jones

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

Doctor of Philosophy

University of Washington

1997

Approved by _____

(Chairperson of Supervisory Committee)

Program Authorized

to Offer Degree _____

Date _____

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 1490 Eisenhower Place, P.O. Box 975, Ann Arbor, MI 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Study of Magnetic Relaxation in Plasmas
Using a Parallel Implicit MHD Solver

by Ogden S. Jones

Chairperson of Supervisory Committee: *Prof. D. Scott Eberhardt*
Aeronautics and Astronautics

A new implicit algorithm has been developed for solving the time-dependent, non-ideal magnetohydrodynamic (MHD) equations. The algorithm is a finite-volume scheme that uses an approximate Riemann solver for the hyperbolic fluxes and central differencing applied on nested control volumes for the parabolic fluxes that arise from the non-ideal terms (i.e. resistivity and viscosity). For sufficiently large Lundquist numbers, the implicit formulation is stable for any size time step, thus allowing efficient tracking of slower transients. The implicit operator is inverted using a lower-upper symmetric-Gauss-Seidel (LU-SGS) iteration. In addition to the implicit scheme, a related explicit scheme has also been developed. Both schemes were implemented in a computer code that has been optimized for parallel computers.

The code has been benchmarked against several analytical solutions. The code was then applied to two problems relevant to the magnetic relaxation of fusion plasmas. The first application was a planar sheet pinch. The linearized equations were first solved to obtain the most unstable eigenmode, which was then used as an initial perturbation. The nonlinear evolution of the resistive tearing mode was then followed using the implicit MHD solver. The second application was a simulation of the three-

dimensional relaxation of simple toroidal configurations that were initialized near equilibrium, but far from the force-free Taylor state. As the Lundquist number was increased, it was found that the fully three-dimensional plasma relaxes more rapidly than an axisymmetric plasma.

TABLE OF CONTENTS

List of Figures	iii
Chapter 1: Introduction	1
1.1 Background	2
1.2 Present Work	6
1.3 Organization of this Dissertation	7
Chapter 2: MHD Model	9
Chapter 3: Numerical Method	15
3.1 Background: Solving the MHD Equations	16
3.2 Description of the Algorithm	17
3.3 Implementation Details	31
Chapter 4: Code Validation Tests	38
4.1 1D Coplanar MHD Riemann Problem	38
4.2 Oblique Shock	41
4.3 Hartmann Flow	45
4.4 MPD Arc	48
Chapter 5: Planar Sheet Pinch	53
5.1 Background	54
5.2 Linear Analysis to Determine Eigenmode	55
5.3 Nonlinear Analysis Using EMI	61

Chapter 6: 3D Plasma Relaxation	65
6.1 Background	68
6.2 Approach	73
6.3 Numerical Details	75
6.4 Boundary Conditions	76
6.5 $S = 10^3$ Simulations	77
6.6 $S = 10^4$ Simulations	93
6.7 Low Beta Simulations	116
Chapter 7: Summary and Conclusions	120
7.1 Conclusions: Code Development	121
7.2 Conclusions: 3D Relaxation of Toroidal Plasmas	122
Bibliography	124
Appendix A: Eigenvectors for Approximate Riemann Solver	128
Appendix B: EMI Description and User's Guide	135

LIST OF FIGURES

1.1	<i>Poloidal cross-section of the Helicity Injected Tokamak showing the coaxial electrodes that surround the plasma and the toroidal field coils.</i>	4
3.1	<i>Grid stencil for parabolic fluxes. Dashed box is the offset mesh used to determine flux at $i+1/2$ face.</i>	28
3.2	<i>Internal ghost cells for case where four blocks meet. Ghost cells for block 1 are shaded.</i>	35
3.3	<i>An example of external ghost cells.</i>	36
4.1	<i>Numerical solution of coplanar Riemann problem. Density and transverse magnetic field are shown initially and after solution has evolved for 400 time steps.</i>	40
4.2	<i>Comparison of numerical and exact solution of coplanar Riemann problem for $B_x = 0$ case.</i>	42
4.3	<i>Geometry and boundary conditions for oblique shock test problem.</i>	43
4.4	<i>Density contours and field lines for an $M = 3$ flow impinging on a perfectly conducting plate at an angle of 25 degrees.</i>	43
4.5	<i>Logarithm of the two-norm of the energy equation residual per cell plotted as a function of iteration number for explicit and implicit solutions of oblique shock problem.</i>	44
4.6	<i>The Hartmann flow geometry showing the moving parallel plates and the cross magnetic field.</i>	45

4.7	<i>Hartmann flow simulation with $H = 10$. Flow velocity vectors and magnetic field lines are shown. The flow velocity only exists close to the plates. The magnetic field lines are linear around the midplane. .</i>	46
4.8	<i>Hartmann flow simulation with $H = 0.1$. Flow velocity vectors and magnetic field lines are shown. The velocity profile is linear and the magnetic field lines have an “S” shape caused by the bulk fluid flow. .</i>	47
4.9	<i>Hartmann flow simulation with $H = 10^4$. Flow velocity vectors and magnetic field lines are shown. The velocity of the flow is zero everywhere except at the plates. The magnetic field lines have a constant slope through the domain.</i>	47
4.10	<i>Geometry of the MPD arc problem.</i>	48
4.11	<i>Comparison of explicit and implicit solutions of plasma velocity versus x at several times for MPD arc problem.</i>	49
4.12	<i>Comparison of explicit and implicit solutions of B_z versus x at several times for MPD arc problem.</i>	51
5.1	<i>Planar sheet pinch geometry.</i>	54
5.2	<i>Equilibrium magnetic field for sheet pinch calculations.</i>	55
5.3	<i>Equilibrium B and η that satisfy zeroth order induction equation. . .</i>	58
5.4	<i>Eigenmodes for symmetric tearing mode for $\alpha = 0.5$ and $S = 10^3$. . .</i>	59
5.5	<i>Computed linear growth rates agree with theory at high S.</i>	60
5.6	<i>Reconnected flux (dimensionless) versus time during nonlinear evolution of symmetric tearing mode at $\alpha = 0.5$ and $S = 10^2$.</i>	61
5.7	<i>Magnetic flux contours for saturated tearing mode at $\alpha = 0.5$ and $S = 10^2$.</i>	64
6.1	<i>Torus with square cross-section and an aspect ratio (R_o/a) of 1.5. . .</i>	66

6.2	<i>Poloidal slice at $\phi = 0$ showing contours of toroidal field overlayed with poloidal field vectors at $t = 0$.</i>	66
6.3	<i>Contours of poloidal magnetic flux for Taylor state of torus with square cross-section and $R_0/a = 1.5$.</i>	70
6.4	<i>Contours of toroidal field magnitude for Taylor state of torus with square cross-section and $R_0/a = 1.5$.</i>	71
6.5	<i>Single block of the 16-block grid used for the $S = 10^3$ runs with $R_0/a = 1.5$.</i>	75
6.6	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 0$.</i>	78
6.7	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 2$.</i>	78
6.8	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 4$.</i>	79
6.9	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 6$.</i>	79
6.10	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 8$.</i>	80
6.11	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 10$.</i>	80
6.12	<i>Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.</i>	81
6.13	<i>Contours of toroidal velocity and poloidal velocity vectors in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.</i>	82
6.14	<i>Contours of density in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.</i>	82
6.15	<i>Contours of pressure in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.</i>	83
6.16	<i>Contours of poloidal flux in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.</i>	84
6.17	<i>Contours of poloidal flux at $t = 12$ for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	84

6.18	<i>Integrated magnetic, thermal, and kinetic energies as a function of time for 3D calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	85
6.19	<i>Integrated magnetic, thermal, and kinetic energies as a function of time for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	85
6.20	<i>Decay of magnetic helicity as a function of time for 3D calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	86
6.21	<i>Decay of magnetic helicity as a function of time for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	86
6.22	<i>Ratio of integrated magnetic energy to helicity as a function of time for 3D calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	87
6.23	<i>Ratio of integrated magnetic energy to helicity as a function of time for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.</i>	87
6.24	<i>Contours of toroidal current density overlaid with poloidal current vectors at $t = 0$ for $R_o/a = 1.5$ and $S = 10^3$.</i>	89
6.25	<i>Contours of toroidal current density overlaid with poloidal current vectors at $t = 12$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.</i>	90
6.26	<i>Contours of $\lambda = j/B$ at $t = 12$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.</i>	90
6.27	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 0$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.</i>	91
6.28	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 12$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.</i>	91
6.29	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 40$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.</i>	92

6.30	<i>Ratio of integrated magnetic energy to helicity for all of the $S = 10^3$ runs.</i>	93
6.31	<i>Contours of poloidal flux for $S = 10^4$ and $R_o/a = 1.5$ at $t = 5$.</i>	94
6.32	<i>Contours of poloidal flux for $S = 10^4$ and $R_o/a = 1.5$ at $t = 10$.</i>	95
6.33	<i>Contours of poloidal flux in $\phi = 0$ plane for $S = 10^4$ and $R_o/a = 1.5$ at $t = 10$.</i>	95
6.34	<i>Contours of poloidal flux at $t = 10$ for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	96
6.35	<i>Contours of pressure in $\phi = 0$ plane for $S = 10^4$ and $R_o/a = 1.5$ at $t = 10$.</i>	96
6.36	<i>Contours of pressure at $t = 10$ for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	97
6.37	<i>Contours of poloidal flux in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.</i>	97
6.38	<i>Contours of poloidal flux in $\phi = 90$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.</i>	98
6.39	<i>Contours of poloidal flux in $\phi = 180$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.</i>	98
6.40	<i>Contours of poloidal flux in $\phi = 270$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.</i>	99
6.41	<i>Integrated magnetic, thermal, and kinetic energies as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	100
6.42	<i>Integrated magnetic, thermal, and kinetic energies as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	101
6.43	<i>Decay of magnetic helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	101

6.44	<i>Decay of magnetic helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	102
6.45	<i>Ratio of integrated magnetic energy to helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	102
6.46	<i>Ratio of integrated magnetic energy to helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.</i>	103
6.47	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for 3D calculation with $R_o/a = 1.5$ and $S = 10^4$.</i>	104
6.48	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^4$.</i>	104
6.49	<i>Contours of poloidal flux for $S = 10^4$ and $R_o/a = 3.0$ at $t = 5$.</i>	105
6.50	<i>Contours of poloidal flux for $S = 10^4$ and $R_o/a = 3.0$ at $t = 10$.</i>	105
6.51	<i>Contours of poloidal flux in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	106
6.52	<i>Contours of poloidal flux at $t = 10$ for axisymmetric calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	107
6.53	<i>Contours of pressure in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	107
6.54	<i>Contours of pressure at $t = 10$ for axisymmetric calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	108
6.55	<i>Contours of poloidal flux in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	108
6.56	<i>Contours of poloidal flux in $\phi = 90$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	109
6.57	<i>Contours of poloidal flux in $\phi = 180$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	109

6.58	<i>Contours of poloidal flux in $\phi = 270$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.</i>	110
6.59	<i>Integrated magnetic, thermal, and kinetic energies as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 3.0$.</i>	110
6.60	<i>Integrated magnetic, thermal, and kinetic energies as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 3.0$.</i>	111
6.61	<i>Decay of magnetic helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 3.0$.</i>	111
6.62	<i>Decay of magnetic helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 3.0$.</i>	112
6.63	<i>Ratio of integrated magnetic energy to helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 3.0$.</i>	112
6.64	<i>Ratio of integrated magnetic energy to helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 3.0$.</i>	113
6.65	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for 3D calculation with $R_o/a = 3.0$ and $S = 10^4$.</i>	113
6.66	<i>$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for axisymmetric calculation with $R_o/a = 3.0$ and $S = 10^4$.</i>	114
6.67	<i>Ratio of integrated magnetic energy to helicity for all of the $S = 10^4$ runs.</i>	115
6.68	<i>Contours of toroidal field at intervals of 0.5 Alfvén time.</i>	116
6.69	<i>Contours of toroidal field at intervals of 0.5 Alfvén time.</i>	117
6.70	<i>Contours of toroidal field at intervals of 2.0 Alfvén times.</i>	117
6.71	<i>Contours of toroidal field at $t = 10$ for axisymmetric run at $S = 10^3$ and $R_o/a = 3.0$.</i>	118

ACKNOWLEDGMENTS

I would like to thank my committee for all their help, especially my advisor, Prof. Scott Eberhardt, who inspired me to enter this field and also provided the opportunity to pursue the research. Thanks also to Prof. Tom Jarboe for funding my first year and a half of research (on a Department of Energy grant). Thanks to the Air Force Office of Scientific Research for funding the remaining three and a half years of my research. Thanks to Prof. Uri Shumlak for administering the Air Force contract and providing very useful guidance in programming the code and in helping to understand the physics. Thanks to the Maui High Performance Computing Center for providing the parallel computing power and expertise I needed.

Special thanks to all my colleagues in the CFD Laboratory. In particular, I'd like to acknowledge the tremendous help I received from Bogdan Udrea, who, among other things, converted my serial code into a parallel code that enabled me to run the large simulations. And thanks to Steve Pluntze, who provided help and moral support during crunch time. You guys are the best.

And finally, I'd like to thank my wife, Jennifer, and my daughter, Emily, who make it all worthwhile.

Chapter 1

INTRODUCTION

Plasmas, like all natural systems, will, if allowed to, tend to evolve over time toward a minimum energy state. Generally speaking, if this evolution occurs gradually (that is, it is not due to a rapid global instability), then the plasma is said to "relax" toward the minimum energy state. In this research we are concerned only with fully ionized or "magnetized" plasmas. Thus, magnetic relaxation refers to any type of relaxation process occurring in a magnetized plasma. Although relaxation processes are basic to all fields of plasma physics, this research focuses on the role relaxation plays in magnetically confined fusion devices.

Relaxation processes are of considerable interest to magnetic fusion research because they are always present. That is, by design a magnetic fusion device squeezes a plasma together in order to get the nuclei to fuse and release energy, and this obviously puts the plasma far from its minimum energy state. Although one can generally stabilize global MHD modes that would otherwise cause the immediate disruption of the plasma, there are always various other localized micro-instabilities that are present and are available to drive the plasma back toward a lower energy state. In fact, it turns out that relaxation processes, when viewed in terms of the relative decay rates of various global invariants or integrals of the motion, can be analyzed in terms of variational principles. That is, one can minimize the energy with the constraint that, on the time scale of the changes in the energy, the global invariant is constant, and thus obtain quantitative information on the resulting relaxed state

of the plasma.

1.1 Background

A theory of plasma relaxation based on variational principles was first applied to fusion plasmas by Taylor [36]. He postulated that the magnetic helicity, which is given by

$$K = \int (\mathbf{A} \cdot \mathbf{B}) d\tau, \quad (1.1)$$

is a constant of motion for plasmas. Here, \mathbf{B} is the magnetic field and \mathbf{A} is the vector potential. When the magnetic energy is minimized under the constraint of constant helicity, the resulting plasma configuration is a force-free state ($\mathbf{j} \parallel \mathbf{B}$) with $\mathbf{j}/\mathbf{B} \equiv \lambda$ equal to a global constant. This theory has been successful in explaining the results of several different types of toroidal devices, including the Reverse Field Pinch (RFP) and the spheromak.

Recently, there has been an ongoing experiment at the University of Washington, the Helicity Injected Tokamak (HIT) [21], that relies on Taylor's relaxation theory in order to operate properly. HIT is based on the premise that a certain amount of relaxation takes place that drives the plasma toward the desired magnetic configuration (a "tokamak"). In order to explain how HIT works, it is necessary first to define what a tokamak is.

A tokamak is a particular type of device for magnetically confining and compressing plasmas in order to obtain controlled thermonuclear fusion power. Like many fusion concepts, the tokamak is a toroidal device. This has the advantage that the magnetic field lines close back on themselves, thus mitigating the end losses that dominate the confinement of open-ended configurations. A tokamak is distinguished from other toroidal fusion concepts in that its poloidal field is much smaller than the externally imposed toroidal field and is created by a toroidal plasma current. There are a number of ways in which to drive a toroidal current in a tokamak. Most toka-

makes use so-called ohmic drive, in which toroidal coils are used to create a changing magnetic flux through the center of the torus (the hole of the doughnut) that drives a toroidal current. In this case, the tokamak is behaving as a transformer, with the plasma acting as the secondary coil. This technique is called ohmic drive because one result of the current is that the plasma is ohmically heated due to its resistivity. This technique has the disadvantage of being pulsed, whereas there are obvious advantages to having a steady-state fusion reactor.

The primary objective of the HIT experiment is to explore the feasibility of using one particular technique, called coaxial helicity injection (CHI), for driving a steady-state plasma current in a tokamak (there are other possibilities, such as using particle beams). The idea behind CHI is to drive current on open magnetic field lines that connect two electrodes, and then to rely on Taylor relaxation to drive current throughout the plasma volume. Figure (1.1) is a schematic of the HIT experiment that illustrates how CHI works. A slice in the poloidal plane is shown. The plasma resides between a cylindrical center column and a outer shell (both copper) that are separated by toroidal insulators shown near the top and bottom of the figure. These are the electrodes. Contours of poloidal magnetic flux are shown in the region between the electrodes. A voltage, V_{inj} , is applied between the electrodes. This drives a current in the plasma and induces a flow of plasma from the injector (bottom) to the absorber (top). Initially the current flows along the open field lines that connect the center column and the shell. The forces push the open field against the walls, as shown. Toroidal flux is brought in with the plasma flow through the insulator. This toroidal flux links the poloidal flux between the electrodes, which is denoted as ψ_{inj} . This adds helicity at a rate given by

$$\dot{K}_{inj} = \psi_{inj} V_{inj}. \quad (1.2)$$

Once quasi-steady operation is achieved, this injected helicity makes up for the resistive decay of helicity. This drives "too much" current on the outer open field lines,

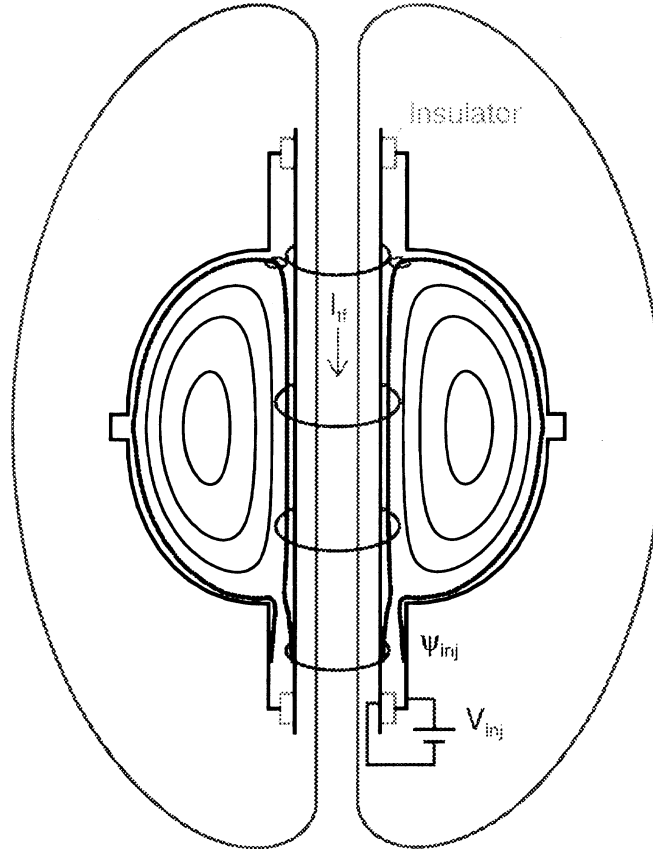


Figure 1.1: *Poloidal cross-section of the Helicity Injected Tokamak showing the coaxial electrodes that surround the plasma and the toroidal field coils.*

which then have a λ larger than the rest of the plasma. If Taylor's theory holds, then the plasma will tend to relax toward a uniform λ , thus redistributing the current so that it forms the toroidal current (closed loops of poloidal flux) of the tokamak.

The HIT experiment provided the motivation for this research, in that it was clear that very detailed knowledge of the details of the relaxation process would be required to understand the results of the HIT experiment. One way in which to gain such as understanding is to perform detailed numerical simulations of plasmas with

sufficient resolution to accurately capture the physics of Taylor relaxation.

1.1.1 Previous Numerical Studies of Relaxation

In the time since Taylor's theory was proposed in the mid-1970s, there has been a lot of work done on the numerical simulation of relaxation in fusion plasmas. In the following, we review a few representative papers from the fusion literature. This is by no means a complete review of the fusion literature on the subject, let alone the plasma literature as a whole.

Much of the previous work came out of modeling spheromak and RFP experiments. Caramana, et al. [4] studied current-driven resistive instabilities in an RFP using a quasi 3-D (single helicity) compressible MHD model. Although the model was able to predict many aspects of the plasma behavior observed in experiments, it was not able to predict the spontaneous reversal of the toroidal field and the maintenance of the reversed field that is observed in all RFP experiments and is predicted by Taylor's theory. However, Aydemir, et al. [1] were able to numerically predict field reversal using several different fully three-dimensional MHD codes. They found that compressibility was critical for getting the correct results. Sgro, et al. [31] studied the "relaxation oscillations" observed in the CTX spheromak experiment. In these oscillations the plasma was observed to undergo periods of slow decay away from the Taylor state, followed by periods of rapid movement back toward the Taylor state. They were able to reproduce these oscillations numerically using a 1-D transport code and a 3-D MHD code. Katayama and Katsurai [17] studied the relaxation of spheromaks with various q profiles using a three-dimensional MHD code and got good agreement with several experiments. All of the codes used in the work described here were finite difference codes.

Another approach has been to study the relaxation process more generally, and somewhat independently of a particular fusion device. This is the approach used in this work. Ting, et al. [37] studied two-dimensional MHD turbulence and "selective

decay” processes using high resolution spectral codes. They define selective decay as any process (such as Taylor relaxation) in which some quantity such as energy is dissipated more rapidly than another quantity, such as helicity. Riyopoulos et al. [26] and Horiuchi and Sata [16] studied relaxation in generic geometries. More recently, Sovinec and Prager [34] studied relaxation in a driven system with some similarities to the HIT experiment. They were able to simulate flux amplification similar to that observed in HIT. However, their geometry was a straight, periodic cylinder, rather than a low aspect ratio torus.

1.2 Present Work

Although there have been a number efforts at simulating Taylor relaxation, many of the efforts to date have been constrained by computer power to relatively low Lundquist numbers (the ratio of the resistive diffusion time to the Alfvén time) or to less than three dimensions. Thus, it is apparent that there is a need for more powerful computational tools to tackle this difficult problem, and others like it. This provided the motivation for the first, and ultimately the primary, objective of this research, which was to develop a new numerical method to enable higher resolution numerical studies of relaxation with parameters closer to those in experiments and with reasonable computer run times.

The simplest model that includes the effects of a resistive plasma, which is the crucial element in Taylor’s theory, is the resistive MHD model. Some of the techniques that have been used to solve these equations numerically are discussed in the next chapter. Ultimately, we chose to develop two new numerical schemes, one implicit and one explicit, based on an approximate Riemann solver. To our knowledge, these are the first Riemann-type schemes for MHD in three-dimensional generalized coordinates. It is also the first application of the LU-SGS implicit technique to the MHD equations. The development and testing of these new schemes is the primary

original contribution of this research.

Once the new code had been developed and checked out, we used it to perform numerical studies of relaxation along the lines of the previous studies described above. We decided to study simple geometries that were somewhat amenable to theoretical treatment in order that the results be as understandable and general as possible, rather than to immediately apply the code to the somewhat more complicated geometry of the HIT experiment.

The first real application was the study of the nonlinear behavior of a symmetric tearing mode in a planar sheet pinch geometry. This problem was chosen because resistive instabilities are thought to drive Taylor relaxation in many systems. While this problem provided an excellent test of the code's capabilities, it was a two-dimensional problem that had previously been solved by others.

On the other hand, the second and final application we studied - the three-dimensional relaxation of RFP-like toroidal configurations - is a system that has not been previously studied, especially at such high resolution. Although these simulations were done with the explicit scheme and thus were not run out as far in time as might be possible with an implicit scheme (which didn't converge for these problems), they did provide insight into the effects of Lundquist number and three-dimensionality in such a system. The results of the three-dimensional toroidal simulations constitute the other original contribution of this work.

1.3 Organization of this Dissertation

This dissertation is broken into two major sections. In the first section, which comprises chapters 2-4, a new multiblock, three-dimensional, parallel, implicit MHD solver for general geometries is developed and then benchmarked against analytical solutions. The resistive/viscous MHD equations are introduced in chapter 2. In chapter 3 the numerical algorithm is discussed. This includes a discussion of the

boundary conditions and other details of the implementation of the algorithm into a usable computer program. In chapter 4 we present the results of various benchmark runs that were used to validate the code.

The second section, chapters 5 and 6, describes the results of two relaxation problems that were studied. In the first problem we studied two-dimensional resistive tearing modes of a planar current sheet. In chapter 6 we use the code to study the three-dimensional relaxation of toroidal plasmas in a particularly simple geometry that is loosely analogous to the HIT experiment. Chapter 7 contains a summary and conclusions.

Chapter 2

MHD MODEL

Plasmas generally exhibit both collective (fluid) and individual (particle) behavior. In the MHD model, the plasma is treated like a conducting fluid having macroscopic parameters that accurately describe its particle-like interactions. The equations can be formally derived (see, for example, Freidberg [10]) by starting from the Boltzmann equation for a fully singly ionized plasma, averaging appropriately over the velocity distributions to get a two-fluid model (ions and electrons), and then simplifying further to obtain a one-fluid model for the ions (with the relative motion of the electrons and ions appearing in the equations as a current density). The main assumptions required for this model are that (1) the plasma is collision dominated, (2) the length scales are long compared to the Debye length, and (3) the frequency is low compared to the plasma oscillation frequency (and the cyclotron frequencies). The physical consequence of the second and third assumptions is that the electrons respond rapidly to shield out any local buildup of charge, thus keeping the plasma approximately neutral. This model is useful in many areas of plasma physics, including fusion plasmas, space and solar plasmas, and electric propulsion.

In the ideal MHD model, which is the simplest, the electrical resistivity and fluid viscosity are neglected. Although the ideal MHD model is useful for examining global stability properties of fusion plasmas, it is too restrictive for studying Taylor relaxation. The reason is that in the ideal MHD approximation the magnetic flux is "frozen" to the plasma motion. Thus, the field lines are precluded from tearing and reconnecting in order to achieve a more energetically favorable magnetic configuration.

When the effects of finite resistivity (due to electron-ion collisions) are included, the model is called resistive MHD. This is the simplest "nonideal" MHD model, and thus it is the simplest model for looking at Taylor relaxation. If finite viscosity (due to ion-ion collisions) is also included, the resulting model is called resistive/viscous MHD, and so on. For fusion plasmas, finite resistivity is typically the most important nonideal effect, although viscosity can become very important for sheared ion flows.

In this work, we use the three-dimensional, resistive/viscous MHD plasma model. The governing equations are the conservation of mass, momentum, and energy of the ions, and the low-frequency Maxwell's equations relating the magnetic and electric fields to the current density. The conservation of mass for the ions is expressed as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0, \quad (2.1)$$

where ρ is the ion density and \mathbf{v} is the ion velocity. The conservation of momentum is

$$\rho \frac{d\mathbf{v}}{dt} = -\nabla p + \nabla \cdot \bar{\bar{\tau}} + \mathbf{j} \times \mathbf{B}, \quad (2.2)$$

where p is the pressure, $\bar{\bar{\tau}}$ is the stress tensor, \mathbf{j} is the current density, and \mathbf{B} is the magnetic field. Note the analogy with the Navier-Stokes equations of fluid dynamics, except that there is an extra body force given by $\mathbf{j} \times \mathbf{B}$. The total energy density is the sum of the internal, kinetic, and magnetic energy densities, and is given by

$$e = \frac{p}{\gamma - 1} + \rho \frac{\mathbf{v} \cdot \mathbf{v}}{2} + \frac{\mathbf{B} \cdot \mathbf{B}}{2\mu_o}, \quad (2.3)$$

where γ is the ratio of the specific heats (i.e. $\gamma = c_p/c_v$). The conservation equation for this is

$$\frac{\partial e}{\partial t} + \nabla \cdot \left[(e + p + B^2/\mu_o) \mathbf{v} - (\mathbf{B} \cdot \mathbf{v}) \mathbf{B}/\mu_o \right] = \nabla \cdot [\mathbf{v} \cdot \bar{\bar{\tau}} - \bar{\bar{\eta}} \cdot \mathbf{j} \times \mathbf{B}], \quad (2.4)$$

where $\bar{\bar{\eta}}$ is the resistivity tensor. Generally, the resistivity is a strong function of temperature. The temperature is related to the density and pressure through an equation of state. In this work, we use the ideal gas law.

In the limit of low frequency ($\omega \ll c/L$), the displacement current in Ampere's law and the net charge in Gauss's law can both be neglected. In this limit, Maxwell's equations reduce to

$$\mu_0 \mathbf{j} = \nabla \times \mathbf{B}, \quad (2.5)$$

and

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}. \quad (2.6)$$

Here, μ_o is the permeability of free space. The first equation is Ampere's law. It is used to eliminate \mathbf{j} from the equations. The second equation is Faraday's law. \mathbf{E} is eliminated from Faraday's law by substituting for \mathbf{E} using the generalized Ohm's law, which is

$$\mathbf{E} + \mathbf{v} \times \mathbf{B} = \bar{\eta} \cdot \mathbf{j}. \quad (2.7)$$

It is useful to write the equations in non-dimensional form. As characteristic or reference variables, we choose length (L), field (B_o), density (ρ_o), resistivity (η_o), and kinematic viscosity (ν_o). Letting the primed variables denote the non-dimensional form, we have

$$\begin{aligned} x' &= \frac{x}{L} \\ y' &= \frac{y}{L} \\ t' &= \frac{t}{L/c_a} \\ \rho' &= \frac{\rho}{\rho_o} \\ v' &= \frac{v}{c_a} \\ B' &= \frac{B}{B_o} \\ e' &= \frac{e}{B_o^2/(2\mu_o)} = \frac{e}{\rho_o c_a^2} \\ p' &= \frac{p}{B_o^2/(2\mu_o)} \\ \nu' &= \frac{\nu}{\nu_o} \\ \eta' &= \frac{\eta}{\eta_o}, \end{aligned} \quad (2.8)$$

where $c_a = B_o/\sqrt{\mu_o\rho_o}$ is the Alfvén speed. When expressed in conservative, non-

dimensional form, the equation set is (dropping the primed notation)

$$\begin{aligned} \frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \mathbf{B} \\ e \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \mathbf{v} - \mathbf{B} \mathbf{B} + (p + \mathbf{B} \cdot \mathbf{B}/2) \mathbf{I} \\ \mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v} \\ (e + p + \mathbf{B} \cdot \mathbf{B}/2) \mathbf{v} - (\mathbf{B} \cdot \mathbf{v}) \mathbf{B} \end{bmatrix} = \\ \nabla \cdot \begin{bmatrix} 0 \\ (Re_A)^{-1} \bar{\bar{\tau}} \\ (S)^{-1} \bar{\bar{E}}_{res} \\ (Re_A)^{-1} \mathbf{v} \cdot \bar{\bar{\tau}} - (S)^{-1} \bar{\bar{\eta}} \cdot (\nabla \times \mathbf{B}) \times \mathbf{B} \end{bmatrix}. \end{aligned} \quad (2.9)$$

\mathbf{I} is the identity matrix. The non-dimensional numbers are defined as follows:

$$\begin{aligned} \text{Lundquist Number :} \quad S &\equiv \mu_o c_a L / \eta_o \\ \text{Modified Reynolds Number :} \quad Re_A &\equiv c_a L / \nu_o \end{aligned} \quad (2.10)$$

Note that the characteristic speed appearing in Re_A is the Alfvén speed, rather than the flow velocity.

The tensor $\bar{\bar{E}}_{res}$ is defined such that

$$\nabla \cdot \bar{\bar{E}}_{res} = -\nabla \times (\bar{\bar{\eta}} \cdot \nabla \times \mathbf{B}), \quad (2.11)$$

where the resistivity tensor, $\bar{\bar{\eta}}$, is

$$\bar{\bar{\eta}} = \text{diag}(\eta_x, \eta_y, \eta_z). \quad (2.12)$$

Typically, the resistivity is much less in the direction parallel to the magnetic field than in the direction perpendicular to the field. The divergence of the $\bar{\bar{E}}_{res}$ tensor accounts for the curl of the part of the \mathbf{E} field due to the electrical resistance. That is

$$\nabla \cdot \bar{\bar{E}}_{res} = \nabla \cdot \begin{bmatrix} 0 & \eta_z j_z & -\eta_y j_y \\ -\eta_z j_z & 0 & \eta_x j_x \\ \eta_y j_y & -\eta_x j_x & 0 \end{bmatrix} = -\nabla \times \begin{bmatrix} \eta_x j_x \\ \eta_y j_y \\ \eta_z j_z \end{bmatrix}. \quad (2.13)$$

In terms of \mathbf{B} , the non-zero components of the tensor are

$$E_{yx} = \eta_z j_z = \eta_z \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right) = -E_{xy} \quad (2.14)$$

$$E_{zx} = \eta_y j_y = \eta_y \left(\frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z} \right) = -E_{xz} \quad (2.15)$$

$$E_{yz} = \eta_x j_x = \eta_x \left(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z} \right) = -E_{zy} \quad (2.16)$$

Similarly, the components of the stress tensor, τ , are

$$\tau_{xx} = \frac{2}{3}\mu \left(2\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y} - \frac{\partial v_z}{\partial z} \right) \quad (2.17)$$

$$\tau_{yy} = \frac{2}{3}\mu \left(2\frac{\partial v_y}{\partial y} - \frac{\partial v_x}{\partial x} - \frac{\partial v_z}{\partial z} \right) \quad (2.18)$$

$$\tau_{zz} = \frac{2}{3}\mu \left(2\frac{\partial v_z}{\partial z} - \frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y} \right) \quad (2.19)$$

$$\tau_{xy} = \mu \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) = \tau_{yx} \quad (2.20)$$

$$\tau_{xz} = \mu \left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z} \right) = \tau_{zx} \quad (2.21)$$

$$\tau_{yz} = \mu \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right) = \tau_{zy} \quad (2.22)$$

Although the viscosity, μ , is generally anisotropic, and must be expressed as a tensor as the resistivity is, in this work we make the simplification that the viscosity is isotropic.

These equations are a mixed set of hyperbolic and parabolic equations. The left hand side of Eq. (2.9) contains the terms from ideal MHD. If the right hand side were zero, this would be a purely hyperbolic system of equations. The dissipative terms are on the right hand side of Eq. (2.9). We refer to these often as the parabolic terms of the equation. Physically, the parabolic terms in the momentum equation are due to the diffusion of momentum, while those in the magnetic induction equation (Faraday's law and Ohm's law) are due to the diffusion of magnetic field. Finally,

the two terms on the right hand side of the energy equation correspond to viscous heating and Joule heating.

For convenience, the MHD equation set (Eq. (2.9)) is rewritten in the following compact form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \bar{\bar{\mathbf{T}}}_h = \nabla \cdot \bar{\bar{\mathbf{T}}}_p, \quad (2.23)$$

where \mathbf{Q} is the vector of conservative variables, $\bar{\bar{\mathbf{T}}}_h$ is the tensor of hyperbolic fluxes, and $\bar{\bar{\mathbf{T}}}_p$ is the tensor of parabolic fluxes. The forms of these vectors and tensors can be seen from Eq. (2.9).

Chapter 3

NUMERICAL METHOD

In this chapter we describe the numerical schemes that were developed for solving the time-dependent, non-ideal MHD equations. Two schemes were developed: an explicit one and an implicit one. Both methods feature an approximate Riemann solver for evaluating the hyperbolic fluxes. This technique has been widely applied to the Euler equations in fluid dynamics, but has only recently been extended to the MHD equations, which have a significantly more complex wave structure. To our knowledge, this work presents the first approximate Riemann solver for MHD written for three-dimensional, generalized coordinates. The implicit scheme is based upon a flux-vector splitting of the hyperbolic fluxes, which also has never been tried before with the MHD equations. The formulation allows the time step to be chosen based on the time scales one wishes to resolve rather than on the stability of the numerical method. This can be important for problems where the time scales of interest are much longer than the transit time of the fast magnetosonic wave. As long as the Reynolds and Lundquist numbers are much larger than one, which is the case for fusion plasmas, this method is numerically stable for any CFL number. As with most implicit schemes, the implicit operator is solved iteratively at each time step. However, due to its simple form in this case, it can be solved using an approximate LU decomposition technique that is extremely efficient.

3.1 Background: Solving the MHD Equations

The MHD model comprises a coupled set of nonlinear partial differential equations that must be solved numerically. Finite-volume methods are one of several different techniques available to solve these equations. They are simple to implement, easily adaptable to complex geometries, and well-suited to handle non-linear phenomena such as mode coupling. The most difficult part of creating a finite-volume scheme for MHD is the determination of the hyperbolic fluxes at each cell interface. Approximate Riemann solvers are a class of methods for evaluating these fluxes that take into account the wave nature of hyperbolic equations. In the past, approximate Riemann solvers have been used extensively in the solution of the Euler and Navier-Stokes equations in fluid dynamics. Recently, a number of explicit schemes built around some type of approximate Riemann solver have been developed for the one-dimensional and multi-dimensional MHD equations [3, 42, 43, 25, 6, 7, 8].

Time-dependent MHD simulations are particularly challenging because of the wide range of time scales present in the model. While the MHD equations can be used to study plasma phenomena occurring on time scales as short as the transit time of a fast MHD wave, for many problems, the important physics occurs on time scales that are much longer. For example, it can be shown that resistive tearing modes, which are important in studying fusion plasmas, evolve on a time scale given by [13]

$$\tau_{tearing} \propto \tau_A^{2/5} \tau_\eta^{3/5} = (S)^{3/5} \tau_A, \quad (3.1)$$

where τ_A is the Alfvén time, τ_η is the resistive diffusion time, and S is the Lundquist number, which is defined as

$$S = \frac{\tau_\eta}{\tau_A}. \quad (3.2)$$

If S is 10^6 , which is typical for laboratory plasmas in fusion applications, the resistive tearing time is approximately 4000 times larger than the Alfvén time. In this case, an explicit scheme would limit the time step to a much smaller value than is

needed to accurately resolve the transient behavior. An implicit scheme removes the numerically-imposed time-step constraint, allowing much larger time steps.

There have been a number of applications of implicit finite-difference methods to nonlinear multidimensional MHD problems. Lindemuth and Killeen [20] and Schnack and Killeen [29] used fully implicit schemes that employed iterations at each time step to invert the implicit operator. However, for large, three-dimensional problems, inverting the full implicit operators became impractical, so various semi-implicit schemes were introduced. Weber *et al.*, [38] used a time-split method whereby the convective terms were solved explicitly and the diffusive terms were solved implicitly. The MACH2 and MACH3 codes [24, 11, 32] also use a time-splitting scheme, but the equations are split in such a way as to remove the Alfvén speed from the numerical stability considerations. Since the equations are decoupled in these schemes, the equations must be iterated at each time step until they converge. Schnack *et al.*, [30] introduced a class of semi-implicit schemes that used operator-splitting to remove the numerical time-step restrictions. These methods have the advantage of not requiring iterations, but inaccuracies are introduced by the operator splitting that limit the allowable time step.

3.2 Description of the Algorithm

The method presented here is based on an algorithm that has been applied to the time-dependent, incompressible Navier-Stokes equations [23]. In one dimension the algorithm is second order accurate in space and time. For multi-dimensions, the accuracy may be slightly less than second order because the one-dimensional flux formulas are applied without using operator splitting [5]. We will derive the algorithm for two dimensional Cartesian coordinates. The extension to three dimensions and general coordinates is straight forward. To begin, we express the MHD equations as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{F}_p}{\partial x} + \frac{\partial \mathbf{G}_p}{\partial y} = 0, \quad (3.3)$$

where \mathbf{F} is the hyperbolic flux vector in the x direction (i.e. $\bar{\bar{\mathbf{T}}}_h = (\mathbf{F}, \mathbf{G})$) and \mathbf{F}_p is the parabolic flux vector in the x direction (i.e. $\bar{\bar{\mathbf{T}}}_p = -(\mathbf{F}_p, \mathbf{G}_p)$). We then discretize Eq. (3.3) in space and time, evaluating the fluxes at the $n + 1$ time level to get

$$\frac{1}{2\Delta t} \left(3\mathbf{Q}_{ij}^{n+1} - 4\mathbf{Q}_{ij}^n + \mathbf{Q}_{ij}^{n-1} \right) = - \left[(\mathbf{R}_h)_{ij}^{n+1} + (\mathbf{R}_p)_{ij}^{n+1} \right], \quad (3.4)$$

where \mathbf{R}_h and \mathbf{R}_p are the discretizations of the hyperbolic and parabolic fluxes, respectively.

Eq. (3.4) is implicit and must be solved iteratively. Let $\mathbf{Q}^{n+1,m}$ denote the m th iteration of the solution at the $n + 1$ time level. To derive a recursive expression for the next iteration of \mathbf{Q}^{n+1} in terms of the previous iteration, we rewrite Eq. (3.4) as

$$\left(\frac{\partial \mathbf{Q}}{\partial t} \right)_{ij}^{m+1} = - \left[(\mathbf{R}_h)_{ij}^{n+1,m+1} + (\mathbf{R}_p)_{ij}^{n+1,m} \right], \quad (3.5)$$

where

$$\left(\frac{\partial \mathbf{Q}}{\partial t} \right)_{ij}^{m+1} \equiv \frac{1}{2\Delta t} \left(3\mathbf{Q}_{ij}^{n+1,m+1} - 4\mathbf{Q}_{ij}^n + \mathbf{Q}_{ij}^{n-1} \right) \quad (3.6)$$

This equation is still implicit, because the time derivative and hyperbolic flux terms are evaluated at the $m+1$ iteration. (Evaluating the parabolic flux at the old iteration level, m , results in a significantly simpler implicit operator, as will be explained later). These terms are related to the previous iteration by linearizing them using truncated Taylor series expansions. That is,

$$\left(\frac{\partial \mathbf{Q}}{\partial t} \right)_{ij}^{m+1} \approx \left(\frac{\partial \mathbf{Q}}{\partial t} \right)_{ij}^m + \frac{\partial (\partial \mathbf{Q} / \partial t)_{ij}^m}{\partial \mathbf{Q}} \left(\mathbf{Q}_{ij}^{n+1,m+1} - \mathbf{Q}_{ij}^{n+1,m} \right). \quad (3.7)$$

and

$$(\mathbf{R}_h)_{ij}^{m+1} \approx (\mathbf{R}_h)_{ij}^m + \frac{\partial (\mathbf{R}_h)_{ij}^m}{\partial \mathbf{Q}} \left(\mathbf{Q}_{ij}^{n+1,m+1} - \mathbf{Q}_{ij}^{n+1,m} \right). \quad (3.8)$$

It is important to note that the partial derivatives above are taken with respect to \mathbf{Q} at every cell, not just \mathbf{Q}_{ij} . The partial derivative of $\partial \mathbf{Q} / \partial t$ with respect to \mathbf{Q} is simply

$$\frac{\partial (\partial \mathbf{Q} / \partial t)}{\partial \mathbf{Q}} = \frac{3\mathbf{I}}{2\Delta t}, \quad (3.9)$$

since it only varies with $\mathbf{Q}_{ij}^{n+1,m}$. However, the partial derivative of \mathbf{R}_h is much more difficult to evaluate. In fact, in order to make it tractable, it is evaluated using a first order accurate approximation of the hyperbolic fluxes, which we will denote as $\hat{\mathbf{R}}_h$, rather than the full second order accurate discretization (\mathbf{R}_h) . This first order approximation can be written generally as

$$\left(\hat{\mathbf{R}}_h\right)_{ij} = f\left(\mathbf{Q}_{ij}, \mathbf{Q}_{i-1,j}, \mathbf{Q}_{i+1,j}, \mathbf{Q}_{i,j-1}, \mathbf{Q}_{i,j+1}\right), \quad (3.10)$$

so that the $m + 1$ iteration of $\left(\hat{\mathbf{R}}_h\right)_{ij}$ is coupled to five points from the previous iteration. Substituting these expressions back into Eq. (3.4) and rearranging, we get

$$\left[\frac{\partial\left(\hat{\mathbf{R}}_h\right)_{ij}^m}{\partial\mathbf{Q}} + \frac{3\mathbf{I}}{2\Delta t}\right]\Delta\mathbf{Q}_{ij}^m = -\left[\left(\mathbf{R}_h\right)_{ij}^m + \left(\mathbf{R}_p\right)_{ij}^m + \left(\frac{\partial\mathbf{Q}}{\partial t}\right)_{ij}^m\right], \quad (3.11)$$

where

$$\Delta\mathbf{Q}_{ij}^m \equiv \mathbf{Q}_{ij}^{n+1,m+1} - \mathbf{Q}_{ij}^{n+1,m}. \quad (3.12)$$

At each time step, Eq. (3.11) is iterated until $\Delta\mathbf{Q}^m$ is driven to approximately zero, at which point the original differential equation is approximately satisfied. The implementation of this iteration depends, of course, on the details of the discretization of the hyperbolic and parabolic fluxes $(\mathbf{R}_h$ and $\mathbf{R}_p)$, and on the linearization of the hyperbolic fluxes $(\partial\hat{\mathbf{R}}_h/\partial\mathbf{Q})$. The left hand side of Eq. (3.11) can be thought of as an implicit operator operating on $\Delta\mathbf{Q}^m$. The implicit operator is a large banded block matrix that is costly to invert directly. Instead, we make a number of simplifications to the operator, and then invert it using the Lower Upper - Symmetric Gauss Seidel (LU-SGS) technique [41].

The implicit scheme described here worked satisfactorily for the code validation tests described in chapter 4 and for the two-dimensional resistive tearing simulations in chapter 5. However, this technique proved unable to converge for the three-dimensional relaxation simulations in chapter 6. For those simulations, we used an explicit scheme given by

$$\frac{\mathbf{Q}_{ij}^{n+1} - \mathbf{Q}_{ij}^n}{\Delta t} = -\left[\left(\mathbf{R}_h\right)_{ij}^n + \left(\mathbf{R}_p\right)_{ij}^n\right]. \quad (3.13)$$

This scheme is first order accurate in time. It would be simple to increase the temporal accuracy by either using the $n - 1$ time level as in the implicit scheme, or by using a multi-step Runge-Kutta technique. However, the explicit scheme was originally intended to simply check the results of the implicit scheme and thus was not extended to higher order time accuracy.

Both the implicit and explicit schemes use the same techniques for differencing the right-hand-side fluxes. The hyperbolic fluxes are differenced by applying Harten's approximate Riemann solver [15] within the framework of the multidimensional technique developed by Powell [25]. The parabolic fluxes are discretized using central differencing on an offset finite volume mesh. In the following sections, each of these parts of the algorithm are described separately in more detail. In order to make the equations simpler, the algorithm is derived for a two-dimensional Cartesian grid. The extension to three dimensions is trivial, except in a few instances, which are noted. The extension to non-Cartesian grids is discussed separately in section 3.3.1.

3.2.1 Approximate Riemann Solver

The finite volume discretization of the hyperbolic fluxes can be written as

$$(\mathbf{R}_h)_{ij} = \mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j} + \mathbf{G}_{i,j+1/2} - \mathbf{G}_{i,j-1/2}. \quad (3.14)$$

Note that in this equation, and all that follow, the grid metric terms (cell areas and volumes) are omitted for clarity. The fluxes at the cell faces in each direction are evaluated by solving a one-dimensional linear Riemann problem defined by the discontinuous jump in \mathbf{Q} between each cell. That is, if we let \mathbf{Q}_l and \mathbf{Q}_r denote the states to the left and right of a cell interface, then the x direction flux is determined by solving

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}(\mathbf{Q}_l, \mathbf{Q}_r) \frac{\partial \mathbf{Q}}{\partial x} = 0, \quad (3.15)$$

where \mathbf{A} , the Jacobian of the \mathbf{F} with respect to \mathbf{Q} , is a function of the left and right states. Similarly, a linear Riemann problem in the y direction is solved to get the y

fluxes. There are a number of different types of approximate Riemann solvers that have been developed. In this work, we adapted a solver originally developed for the Euler equations by Harten [15].

To solve the linear Riemann problem defined above, we write \mathbf{A} as

$$\mathbf{A} = \mathbf{X}^{-1}\mathbf{\Lambda}\mathbf{X}, \quad (3.16)$$

where \mathbf{X} is a matrix whose columns are the right eigenvectors of \mathbf{A} , \mathbf{X}^{-1} is its inverse (its rows are the left eigenvectors of \mathbf{A}), and $\mathbf{\Lambda}$ is a matrix having the eigenvalues of \mathbf{A} along the diagonal. The eigenvalues of \mathbf{A} are

$$\lambda = (v_x, 0, v_x \pm c_f, v_x \pm c_s, v_x \pm v_a)^T, \quad (3.17)$$

where c_f and c_s are the fast and slow magnetosonic speeds in the x direction, and v_a is the Alfvén speed based on the x component of the magnetic field. These can be expressed as

$$c_f^2 = \frac{1}{2} \left\{ a^2 + c_a^2 + \left[(a^2 + c_a^2)^2 - 4a^2v_a^2 \right]^{1/2} \right\} \quad (3.18)$$

$$c_s^2 = \frac{1}{2} \left\{ a^2 + c_a^2 - \left[(a^2 + c_a^2)^2 - 4a^2v_a^2 \right]^{1/2} \right\} \quad (3.19)$$

$$v_a^2 = \frac{B_x^2}{\mu_0\rho}. \quad (3.20)$$

Here, a is the ion acoustic speed, which for a perfect gas is

$$a^2 = \frac{\gamma p}{\rho}. \quad (3.21)$$

The zero eigenvalue arises from the fact that the $\mathbf{j} \times \mathbf{B}$ force acts perpendicularly to the directions of \mathbf{j} and \mathbf{B} , so that the \mathbf{F} flux vector has a zero term corresponding to B_x . For approximate Riemann solvers, there are basically two approaches to solving this problem. The most common is to drop B_x from \mathbf{Q} when applying the Riemann solver to the x fluxes, drop B_y from \mathbf{Q} when applying it to the y fluxes, and so forth. That is, in each direction a different seven-variable Riemann problem is solved.

An alternate approach, which is used in this work, has been developed by Powell [25]. In this approach the Jacobians are modified in order to change the zero eigenvalue to the flow speed, and a source term is added that exactly cancels the terms introduced by the modification of the Jacobians. The eigenvalues of the modified Jacobian, $\tilde{\mathbf{A}}$, are

$$\lambda = (v_x, v_x, v_x \pm c_f, v_x \pm c_s, v_x \pm v_a)^T. \quad (3.22)$$

Thus the modification of \mathbf{A} has changed the zero eigenvalue to the flow speed, while keeping the others unchanged. In the same way, the zero eigenvalue for the y flux is changed to v_y . The source term, \mathbf{S}_{div} , is given by

$$\mathbf{S}_{div} = - \begin{bmatrix} 0 \\ \mathbf{B} \\ \mathbf{v} \\ \mathbf{v} \cdot \mathbf{B} \end{bmatrix} \nabla \cdot \mathbf{B} \quad (3.23)$$

It is proportional to $\nabla \cdot \mathbf{B}$, which is zero analytically, but not numerically. It can be shown [25] that the artificial eighth wave is associated with the convection of non-zero $\nabla \cdot \mathbf{B}$ produced by truncation errors.

The main reason we chose the eight-variable Riemann solver approach is that it was simpler to incorporate into our implicit iteration scheme. An additional benefit of this approach for the problems we have considered, is that the formulation automatically insures that $\nabla \cdot \mathbf{B}$ remains approximately zero and does not grow, because any finite $\nabla \cdot \mathbf{B}$ is convected out of the domain [25]. However, it should be noted that for more complicated flows having stagnation points and recirculation regions, this technique would not guarantee that $\nabla \cdot \mathbf{B}$ would remain zero everywhere in the domain. For those types of problems, we employ a "divergence fix" between time steps in which an auxiliary equation is solved in order to remove any non-solenoidal components of the field. This technique is commonly done in other MHD Riemann solvers (e.g. [43]) and is explained in section 3.2.4.

With the addition of the source term, \mathbf{R}_h becomes

$$(\mathbf{R}_h)_{ij} = \mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j} + \mathbf{G}_{i,j+1/2} - \mathbf{G}_{i,j-1/2} + (\mathbf{S}_{div})_{ij}. \quad (3.24)$$

The fluxes are calculated by using the modified Jacobians with Harten's scheme in the form used by Yee, et al. [40]. In this form, the x direction flux at the $i + 1/2$ face is

$$\mathbf{F}_{i+1/2} = \frac{1}{2} [\mathbf{F}_{i+1} + \mathbf{F}_i] + \frac{1}{2} \sum_k \phi^k \mathbf{r}_{i+1/2}^k, \quad (3.25)$$

where \mathbf{r}^k is the k^{th} right eigenvector of $\tilde{\mathbf{A}}$, and ϕ^k is a nonlinear dissipation term that is designed to provide just enough dissipation near sharp gradients to suppress numerical oscillations while still achieving higher order spatial accuracy throughout the rest of the domain. In smooth regions of the solution, ϕ is approximately zero, so that $\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}$ yields second order accurate central differencing. However, near steep gradients and discontinuities, ϕ makes the flux approximately equal to its first order accurate upwind form given by

$$\mathbf{F}_{i+1/2} = \frac{1}{2} [\mathbf{F}_{i+1} + \mathbf{F}_i] - \frac{1}{2} \sum_k \psi(\lambda_{i+1/2}^k) \alpha_{i+1/2}^k \mathbf{r}_{i+1/2}^k, \quad (3.26)$$

where

$$\alpha_{i+1/2}^k = \mathbf{l}_{i+1/2}^k \cdot (\mathbf{Q}_{i+1} - \mathbf{Q}_i), \quad (3.27)$$

λ^k is the k^{th} eigenvalue of $\tilde{\mathbf{A}}$, \mathbf{l}^k is the k^{th} left eigenvector of $\tilde{\mathbf{A}}$, and ψ is the absolute value function with a smoothing term that adds a small amount of dissipation in order to enforce the entropy condition. It is given by

$$\psi(z) = \begin{cases} \frac{1}{2} \left[\frac{z^2}{\epsilon} + \epsilon \right] & \text{if } |z| \leq \epsilon \\ |z| & \text{if } |z| > \epsilon \end{cases}, \quad (3.28)$$

where ϵ is typically set to 0.01. Note that the values at the cell interface ($i + 1/2$) are obtained by a simple average of the neighboring cells. It is possible to evaluate the average state at the cell interface using a more complicated and potentially more robust averaging such as the "Roe average". However, we have found in practice

that simple averaging is sufficient except for the most extreme cases, such as for very strong shocks.

This full expression for ϕ is

$$\phi_{i+1/2}^k = g_i^k + g_{i+1}^k - \psi(\lambda_{i+1/2}^k + \gamma_{i+1/2}^k)\alpha_{i+1/2}^k, \quad (3.29)$$

where

$$g_i = \frac{S}{2} \cdot \max \left[0, \min \left[\psi(\lambda_{i+1/2}^k)|\alpha_{i+1/2}^k|, S\psi(\lambda_{i-1/2}^k)\alpha_{i-1/2}^k \right] \right], \quad (3.30)$$

$$S = \text{sgn} \left(\alpha_{i+1/2}^k \right), \quad (3.31)$$

and

$$\gamma_{i+1/2}^k = \frac{g_{i+1}^k - g_i^k}{\alpha_{i+1/2}^k}. \quad (3.32)$$

It is essentially an upwind-weighted minmod function.

The eigenvectors were found analytically for generalized coordinates with the help of the Mathematica symbolic mathematics software package [39]. It is important that the eigenvectors of the modified Jacobians be properly normalized to insure that they remain well-defined and form a complete set for various degenerate cases. Here, we have followed the normalization developed by Balsara and Roe [28]. Their normalization was for Cartesian coordinates, so we had to extend this to generalized coordinates. The generalized normalization is significantly more complicated, although it reduces to the Cartesian form. The complete normalized eigenvectors and the procedure for handling all the possible degenerate cases are shown in appendix A.

3.2.2 LU-SGS Relaxation Scheme

As mentioned previously, the LU-SGS scheme is based on a simpler first order approximation to the hyperbolic fluxes, rather than on Harten's higher order discretization described above. The first order approximation, denoted by $\hat{\mathbf{R}}_h$, is actually a flux-vector splitting [2] of the hyperbolic fluxes. This splitting of the fluxes is closely

related to Harten's first order upwind form. If we set ϵ in Eq. (3.28) to zero, then Harten's first order flux becomes

$$\mathbf{F}_{i+1/2} = \frac{1}{2} \left[\mathbf{F}_{i+1} + \mathbf{F}_i - \sum_k |\lambda|_{i+1/2}^k \alpha_{i+1/2}^k \mathbf{r}_{i+1/2}^k \right]. \quad (3.33)$$

Equivalently, this can also be expressed as

$$\mathbf{F}_{i+1/2} = \frac{1}{2} \left[\mathbf{F}_{i+1} + \mathbf{F}_i - |\mathbf{A}|_{i+1/2} (\mathbf{Q}_{i+1} - \mathbf{Q}_i) \right]. \quad (3.34)$$

Here,

$$|\mathbf{A}| \equiv \mathbf{A}^+ - \mathbf{A}^-, \quad (3.35)$$

where

$$\mathbf{A}^\pm \equiv \mathbf{X}^{-1} \mathbf{\Lambda}^\pm \mathbf{X}, \quad (3.36)$$

and $\mathbf{\Lambda}^\pm$ is a matrix having either all positive or all negative eigenvalues of $\tilde{\mathbf{A}}$ along its diagonal.

Flux-vector splitting was developed for the Euler equations, which, for the ideal gas equation of state, have the property that the flux function is a homogeneous function of degree one in \mathbf{Q} [35] and thus can be written as

$$\mathbf{F} \equiv \mathbf{A}\mathbf{Q}. \quad (3.37)$$

For the MHD equations, this property no longer holds. However, in this work we have found that it is a good enough approximation to form the basis of a relaxation scheme. That is, we say that

$$\mathbf{F} \approx \mathbf{A}\mathbf{Q}, \quad (3.38)$$

so that we can define

$$\mathbf{F}^\pm \equiv \mathbf{A}^\pm \mathbf{Q}. \quad (3.39)$$

Using this type of splitting is equivalent to evaluating $|\mathbf{A}|$ in Eq. (3.34) at the cell centers, i and $i + 1$, rather than at the cell interface, $i + 1/2$. With this change, Eq. (3.34) becomes

$$\mathbf{F}_{i+1/2} = \frac{1}{2} \left[\mathbf{F}_{i+1} + \mathbf{F}_i - \left(|\mathbf{A}|_{i+1} \mathbf{Q}_{i+1} - |\mathbf{A}|_i \mathbf{Q}_i \right) \right]. \quad (3.40)$$

However, since $|\mathbf{A}| = \mathbf{A}^+ - \mathbf{A}^-$, the flux simplifies to

$$\mathbf{F}_{i+1/2} = \mathbf{A}_{i+1}^- \mathbf{Q}_{i+1} + \mathbf{A}_i^+ \mathbf{Q}_i = \mathbf{F}_{i+1}^- + \mathbf{F}_i^+. \quad (3.41)$$

In general, $\hat{\mathbf{R}}_h$ can be written as

$$\left(\hat{\mathbf{R}}_h\right)_{ij} = (\mathbf{F})_{i+1/2,j} - (\mathbf{F})_{i-1/2,j} + (\mathbf{G})_{i,j+1/2} - (\mathbf{G})_{i,j-1/2}. \quad (3.42)$$

Substituting Eq. (3.40) and similar expressions for $\mathbf{F}_{i-1/2}$, $\mathbf{G}_{i+1/2}$, and $\mathbf{G}_{i-1/2}$ into Eq. (3.42), we get

$$\left(\hat{\mathbf{R}}_h\right)_{ij} = \mathbf{F}_{ij}^+ - \mathbf{F}_{i-1,j}^+ + \mathbf{F}_{i+1,j}^- - \mathbf{F}_{ij}^- + \mathbf{G}_{ij}^+ - \mathbf{G}_{i,j-1}^+ + \mathbf{G}_{i,j+1}^- - \mathbf{G}_{ij}^-. \quad (3.43)$$

Next, we form $\partial\hat{\mathbf{R}}_h/\partial\mathbf{Q}$ and substitute it into the left hand side of Eq. (3.11) to obtain

$$\begin{aligned} & \left\{ \frac{3\mathbf{I}}{2\Delta t} + \mathbf{A}_{ij}^+ - \mathbf{A}_{i-1,j}^+ + \mathbf{A}_{i+1,j}^- - \mathbf{A}_{ij}^- + \mathbf{B}_{ij}^+ - \mathbf{B}_{i,j-1}^+ + \mathbf{B}_{i,j+1}^- - \mathbf{B}_{ij}^- \right\} \\ & \times \Delta\mathbf{Q}_{ij}^n = -\mathbf{R}_{ij}^n, \end{aligned} \quad (3.44)$$

where all of the terms on the right hand side of the equations have been lumped together in \mathbf{R} . Here, \mathbf{B} is the Jacobian of \mathbf{G} . This is a block pentadiagonal matrix, and the blocks themselves are fairly difficult to evaluate. To simplify the matrix, we approximate \mathbf{A}^\pm and \mathbf{B}^\pm as

$$\mathbf{A}^\pm \approx \frac{1}{2} \left(\tilde{\mathbf{A}} \pm \rho_A \mathbf{I} \right) \quad (3.45)$$

$$\mathbf{B}^\pm \approx \frac{1}{2} \left(\tilde{\mathbf{B}} \pm \rho_B \mathbf{I} \right), \quad (3.46)$$

where ρ_A is the maximum eigenvalue of $\tilde{\mathbf{A}}$, which is simply $v_x + c_f$, and ρ_B is defined similarly. The result of this approximation is to reduce the convergence rate of the relaxation because the operator is less closely coupled to the detailed wave structure of the right hand side fluxes. However, the reduction in work per iteration more than offsets the reduction in convergence rate. An important point here is that the approximation to the implicit operator does not reduce the accuracy of the solution

at each time step. As long as the scheme converges at each time step, the accuracy of the solution is determined by the time step size and the right hand side fluxes.

With this approximation,

$$\mathbf{A}^+ - \mathbf{A}^- = \rho_A \mathbf{I}, \quad (3.47)$$

so that Eq. (3.44) simplifies to

$$\left\{ \mathbf{D}_{ij} + \mathbf{A}_{i+1,j}^- + \mathbf{B}_{i,j+1}^- - \mathbf{A}_{i-1,j}^+ - \mathbf{B}_{i,j-1}^+ \right\} \Delta \mathbf{Q}_{ij}^n = -\mathbf{R}_{ij}^n, \quad (3.48)$$

where

$$\mathbf{D}_{ij} = \left(\frac{3}{2\Delta t} + \rho_A + \rho_B \right) \mathbf{I}_{ij}. \quad (3.49)$$

This block matrix equation can be solved in two steps using a forward Gauss-Seidel sweep followed by a backward sweep. The resulting algorithm can be written as

$$\begin{aligned} & \left\{ \mathbf{D}_{ij} - \mathbf{A}_{i-1,j}^+ - \mathbf{B}_{i,j-1}^+ \right\} \left\{ \mathbf{D}_{ij} + \mathbf{A}_{i+1,j}^- + \mathbf{B}_{i,j+1}^- \right\} \\ & \times \Delta \mathbf{Q}_{ij}^m = -(\mathbf{D}_{ij}) \mathbf{R}_{ij}^m, \end{aligned} \quad (3.50)$$

The forward sweep is equivalent to inverting a lower block diagonal matrix [the first braced term in Eq. (3.50)], and the backward sweep is equivalent to inverting an upper block diagonal matrix [second braced term in Eq. (3.50)]. That is, the operator has been split according to the sign of the eigenvalues. Note that this is not a directional splitting. The fluxes in the different directions are evaluated simultaneously. The appeal of this scheme is that it doesn't require any block matrix inversions, since the blocks along the diagonal, \mathbf{D} , contain only diagonal elements. Thus, a single LU-SGS iteration requires only slightly more (about 10% more) computations than a single explicit time step. We note here that the evaluation of the parabolic fluxes, \mathbf{R}_p , at the old iteration level (m) in Eq. (3.11) was motivated by the desire to achieve this diagonal form, since linearization of \mathbf{R}_p^{m+1} would have introduced some off-diagonal terms. The price for this simplification is that for low Re_A or S , the time step may be limited by the numerical stability of the parabolic terms. However, for Re_A and

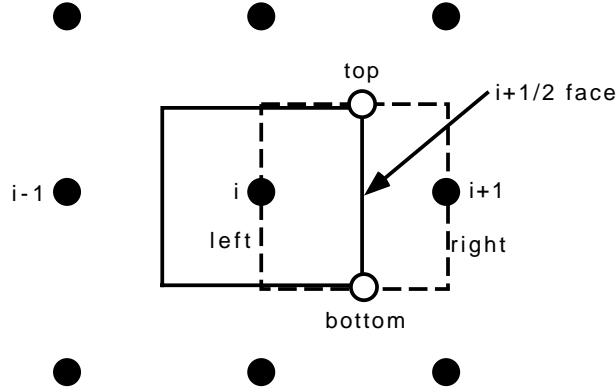


Figure 3.1: *Grid stencil for parabolic fluxes. Dashed box is the offset mesh used to determine flux at $i+1/2$ face.*

S much larger than one, the CFL number is not limited by numerical stability. This simple structure also lends itself well to parallelization using domain decomposition. In addition, the relaxation scheme is highly vectorizable if we sweep through the computational domain along lines of constant $i + j$ (in 2D), so that each term along these lines is independent of the others and depends only on data that has already been updated during the current sweep.

3.2.3 Parabolic Terms

The parabolic fluxes are differenced using a second order accurate finite-volume scheme. The parabolic terms are the terms due to resistivity and viscosity that appear on the right hand side of Eq. (2.9) in the momentum, induction, and energy equations. The discretization of the parabolic fluxes, \mathbf{R}_p is given by

$$(\mathbf{R}_p)_{ij} = (\mathbf{F}_p)_{i+1/2,j} - (\mathbf{F}_p)_{i-1/2,j} + (\mathbf{G}_p)_{i,j+1/2} - (\mathbf{G}_p)_{i,j-1/2}. \quad (3.51)$$

The face-centered fluxes that appear in the expression for \mathbf{R}_p are obtained by performing a flux balance on an offset, face-centered mesh.

Figure 3.1 shows the nine-point grid stencil for a uniformly-spaced Cartesian grid.

The solid dots show the locations of the cell centers. The solid lines indicate the boundaries of the cell surrounding point (i, j) . The dashed lines indicate the offset mesh used to calculate the parabolic flux at the $i + 1/2$ face. The faces of the offset mesh are labeled top, bottom, left, and right.

First we consider the differencing of the parabolic in the momentum equation, which is $\nabla \cdot \bar{\tau}$. As an example, consider τ_{xx} , which is part of the x-momentum equation (the second term of the \mathbf{F}_p vector). In two dimensions, τ_{xx} at the $i + 1/2$ face is given by

$$(\tau_{xx})_{i+1/2} = \frac{2}{3}\mu_{i+1/2} \left(2\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y} \right)_{i+1/2}, \quad (3.52)$$

where μ is the dynamic viscosity. The derivatives at the $i + 1/2$ face are given by one-dimensional applications of the divergence theorem as

$$\frac{\partial v_x}{\partial x} = \frac{[(v_x)_{i+1}(dS_x)_{right} - (v_x)_i(dS_x)_{left}]}{V_{fc}} \quad (3.53)$$

and

$$\frac{\partial v_y}{\partial y} = \frac{[(v_y)_{top}(dS_y)_{top} - (v_y)_{bottom}(dS_y)_{bottom}]}{V_{fc}}, \quad (3.54)$$

where V_{fc} is the volume of the face-centered cell, $(dS_x)_{right}$ is the area of its right face, and so forth. The face areas of the off-set mesh are obtained by appropriate averages of neighboring regular cell faces. For the cell in the figure, we have

$$(dS_x)_{left} = \frac{1}{2} [(dS_x)_{i-1,j} + (dS_x)_{i,j}], \quad (3.55)$$

$$(dS_x)_{right} = \frac{1}{2} [(dS_x)_{i,j} + (dS_x)_{i+1,j}], \quad (3.56)$$

$$(dS_y)_{bottom} = \frac{1}{2} [(dS_y)_{i-1,j} + (dS_y)_{i,j}], \quad (3.57)$$

and

$$(dS_y)_{top} = \frac{1}{2} [(dS_y)_{i-1,j+1} + (dS_y)_{i,j+1}]. \quad (3.58)$$

Finally, the viscosity at the $i + 1/2$ face is

$$\mu_{i+1/2} = \frac{2\mu_i\mu_{i+1}}{\mu_i + \mu_{i+1}}. \quad (3.59)$$

That is, the viscosities on either side of the interface are averaged like two resistances in parallel.

The variables at the left and right faces of the off-set mesh are given by \mathbf{Q}_i and \mathbf{Q}_{i+1} , respectively. The variables at the top and bottom faces, which are denoted by open circles in the figure, are located at the vertices of the original mesh. These vertex quantities are obtained by a volume-weighted average of the four cells surrounding each vertex (in two dimensions). In three dimensions, the situation is somewhat more complicated. In this case the vertex quantities are obtained by volume-weighted average the values of the eight neighboring cells. Then, to get the value on, for example, the top face of the three-dimensional offset mesh, two vertex quantities, one from the $k - 1/2$ plane and one from the $k + 1/2$ plane are averaged, yielding what amounts to an edge-centered quantity on the original mesh. Note that for nonuniform meshes, a slight increase in accuracy can be achieved by using a more accurate averaging scheme to obtain the parabolic fluxes at the cell faces. However, the simple averaging employed here is sufficiently accurate for most applications.

The parabolic term in the induction equation, $\nabla \cdot \bar{\bar{E}}$, is differenced in a similar manner. Consider one term in the $\bar{\bar{E}}$ tensor,

$$E_{yx} = \eta_z j_z = \eta_z \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right). \quad (3.60)$$

The resistivity is averaged in the same way as the viscosity, and the derivatives in \mathbf{B} are computed in the same way as the derivatives in \mathbf{v} .

The parabolic part of the energy equation has terms describing resistive heating and viscous heating. The viscous heating term is $\nabla \cdot (\mathbf{v} \cdot \bar{\bar{\tau}})$. Evaluation of this term requires \mathbf{v} at each face, which is obtained by averaging the values of the two cell centers on either side of the face. The stress tensor components on the cell faces are evaluated the same way as for the momentum equation. The resistive heating term is

$$-\nabla \cdot (\bar{\bar{\eta}} \cdot \mathbf{j} \times \mathbf{B}). \quad (3.61)$$

This is evaluated by using

$$-\bar{\eta} \cdot \mathbf{j} \times \mathbf{B} = \begin{bmatrix} B_y \eta_z j_z - B_z \eta_y j_y \\ B_z \eta_x j_x - B_x \eta_z j_z \\ B_x \eta_y j_y - B_y \eta_x j_x \end{bmatrix}. \quad (3.62)$$

The face-centered \mathbf{B} is obtained by averaging the values of the cell centers on either side of the face, while the remaining terms are the components of the \bar{E} tensor described above.

3.2.4 Divergence Cleaning

As discussed previously, the approximate Riemann solver used here only guarantees that the magnetic field remains divergence-free to within the truncation error of the method. In some calculations, it is possible that divergence errors could build up over time and contaminate the solution.

To protect against this possibility, a "divergence cleaning" procedure was added to the algorithm. The procedure works as follows. Suppose that the calculated field, \mathbf{B} , has a non-zero divergence. Then, we define a modified field, \mathbf{B}' , as

$$\mathbf{B}' = \mathbf{B} + \nabla b, \quad (3.63)$$

where b is chosen to make $\nabla \cdot \mathbf{B}' = 0$. That is, we have

$$\nabla \cdot \mathbf{B}' = \nabla \cdot \mathbf{B} + \nabla^2 b = 0, \quad (3.64)$$

which implies

$$\nabla^2 b = -\nabla \cdot \mathbf{B}. \quad (3.65)$$

The Poisson equation for b is solved using a Jacobi iteration.

3.3 Implementation Details

The algorithm described in the previous sections has been implemented in a code we have named Enhanced MACH3 Implicit (EMI), because it will eventually become

integrated with the MACH3 code [32]. The code was originally written in Fortran 77, but was later changed to Fortran 90 in order to allow dynamic allocation of memory, which was essential for running large jobs on a parallel computer. In this section we describe some important aspects of the implementation of the algorithm, including generalized coordinates, axisymmetric source terms, multiblock grids, and parallelization. Appendix B is a "user's guide" for version v9701 of the code, which is the version that was used for the three-dimensional simulations in chapter 6. The user's guide includes a description of the program flow and input parameters, the format of the grid files, and other details that are needed to run this version of the code.

3.3.1 Finite Volume Formulation for General Hexahedral Cells

To derive the finite volume formulation, we begin with the equations in tensor form,

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \bar{\bar{\mathbf{T}}}_h = \nabla \cdot \bar{\bar{\mathbf{T}}}_p. \quad (3.66)$$

The integral form of these equations, which is obtained by applying the divergence theorem, is

$$\int_V \left(\frac{\partial \mathbf{Q}}{\partial t} \right) dV + \int_S \bar{\bar{\mathbf{T}}}_h \cdot d\mathbf{S} = \int_S \bar{\bar{\mathbf{T}}}_p \cdot d\mathbf{S}. \quad (3.67)$$

The discretized equations are obtained by applying the integral form of the equations to each computational cell. Each cell is an arbitrary hexahedron and has associated with it a cell volume, V , and six outward-pointing cell face area vectors. The areas and volumes are computed using the formulas in Kordulla and Vinokur [19]. The resulting equation is

$$\begin{aligned} V_{ijk} \left(\frac{\partial \mathbf{Q}}{\partial t} \right)_{ijk} + [(\bar{\bar{\mathbf{T}}}_h - \bar{\bar{\mathbf{T}}}_p) \cdot d\mathbf{S}]_{i+1/2} - [(\bar{\bar{\mathbf{T}}}_h - \bar{\bar{\mathbf{T}}}_p) \cdot d\mathbf{S}]_{i-1/2} \\ + [(\bar{\bar{\mathbf{T}}}_h - \bar{\bar{\mathbf{T}}}_p) \cdot d\mathbf{S}]_{j+1/2} - [(\bar{\bar{\mathbf{T}}}_h - \bar{\bar{\mathbf{T}}}_p) \cdot d\mathbf{S}]_{j-1/2} = 0. \end{aligned} \quad (3.68)$$

This can be written in more compact form by dividing by the cell volume and defining

$$\bar{\mathbf{F}}_{i+1/2} = \frac{\left(\bar{\mathbf{T}}_h\right)_{i+1/2} \cdot d\mathbf{S}_{i+1/2}}{V_{ijk}}, \quad (3.69)$$

with similar expressions for the other fluxes. Substituting these definitions into the equation, we get

$$\begin{aligned} & \left(\frac{\partial \mathbf{Q}}{\partial t}\right)_{ijk} + \bar{\mathbf{F}}_{i+1/2} - \bar{\mathbf{F}}_{i-1/2} + \bar{\mathbf{G}}_{j+1/2} - \bar{\mathbf{G}}_{j-1/2} \\ & + \left(\bar{\mathbf{F}}_p\right)_{i+1/2} - \left(\bar{\mathbf{F}}_p\right)_{i-1/2} + \left(\bar{\mathbf{G}}_p\right)_{j+1/2} - \left(\bar{\mathbf{G}}_p\right)_{j-1/2} = 0. \end{aligned} \quad (3.70)$$

In order to get the discretized equations for general geometries, one simply substitutes these generalized fluxes into the equations defined in the previous sections.

It is often convenient to think of the cell areas and volumes as defining a transformation from computational space to physical space. In this notation the direction of increasing i index is denoted as ξ , the direction of increasing j as η , and the direction of increasing k as ζ . In general, $\xi = \xi(x, y, z)$, and so forth. The coordinates (ξ, η, ζ) are known as generalized coordinates. The generalized fluxes are related to the Cartesian fluxes by

$$\bar{\mathbf{F}} = \xi_x \mathbf{F} + \xi_y \mathbf{G} + \xi_z \mathbf{H}. \quad (3.71)$$

By analogy with Eq. (3.69), we see that

$$(\xi_x)_{i+1/2} = \frac{(dS_x)_{i+1/2}}{V_{ijk}} \quad (3.72)$$

Terms like ξ_x are known as the grid metrics, since they define a transformation from physical space to computational space.

3.3.2 Coordinate System and Axisymmetric Terms

Although the toroidal simulations described in chapter 6 are most naturally described by a cylindrical coordinate system, EMI is actually a Cartesian-based code. That is,

although the equations are solved on an arbitrary, non-Cartesian mesh, the momentum and magnetic field components in \mathbf{Q} are Cartesian rather than cylindrical. This is one area where EMI and MACH3 differ. One advantage of this is that the code is applicable to a wide variety of geometries. Another reason for this choice was to simplify the development of the three-dimensional version of the code, since it is much easier to add an extra dimension to a Cartesian code than to a cylindrical code.

The only time cylindrical coordinates are used is for axisymmetric calculations. For these calculations, a source term is added to the right hand side of the equations. We use the convention that the x Cartesian coordinate becomes the R cylindrical coordinate, and the y Cartesian coordinate becomes the Z cylindrical coordinate. Note that with this pairing of coordinates, the out-of-plane coordinate, z , corresponds to the $-\phi$ direction, when considering the standard right-handed (R, ϕ, Z) cylindrical coordinate system. To obtain the source term, we follow the procedure used by Nietubicz, et al. [22] and Frese [11]. If we rewrite the momentum equation as

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \bar{\bar{M}} = 0, \quad (3.73)$$

and the induction equation as

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot \bar{\bar{N}} = 0, \quad (3.74)$$

then the axisymmetric source term, \mathbf{S}_{axi} is

$$\mathbf{S}_{axi} = - \begin{bmatrix} 0 \\ M_{\phi\phi}/R \\ 0 \\ -M_{\phi R}/R \\ 0 \\ 0 \\ -N_{\phi R}/R \\ 0 \end{bmatrix}, \quad (3.75)$$

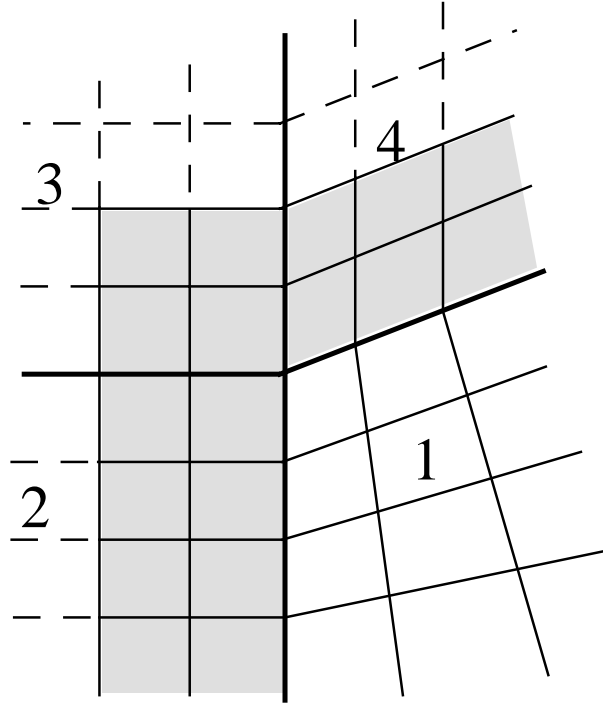


Figure 3.2: *Internal ghost cells for case where four blocks meet. Ghost cells for block 1 are shaded.*

where $M_{\phi\phi}$ is the contribution to the ϕ -momentum equation from the ϕ -face flux, $M_{\phi R}$ is the contribution to the ϕ -momentum equation from the R -face flux, and $N_{\phi R}$ is the contribution to the B_ϕ equation from the R -face flux. These three terms are given by

$$M_{\phi\phi} = \rho v_\phi^2 + p + \frac{B^2}{2} - B_\phi^2 + \frac{2}{3} \frac{\mu}{Re_A} \left(\frac{\partial v_R}{\partial R} + \frac{\partial v_Z}{\partial Z} \right), \quad (3.76)$$

$$M_{\phi R} = \rho v_R v_\phi - B_R B_\phi - \frac{\mu}{Re_A} \frac{\partial v_\phi}{\partial R}, \quad (3.77)$$

and

$$N_{\phi R} = v_\phi B_R - v_R B_\phi + \frac{\eta_Z}{S} \frac{\partial B_\phi}{\partial R}. \quad (3.78)$$

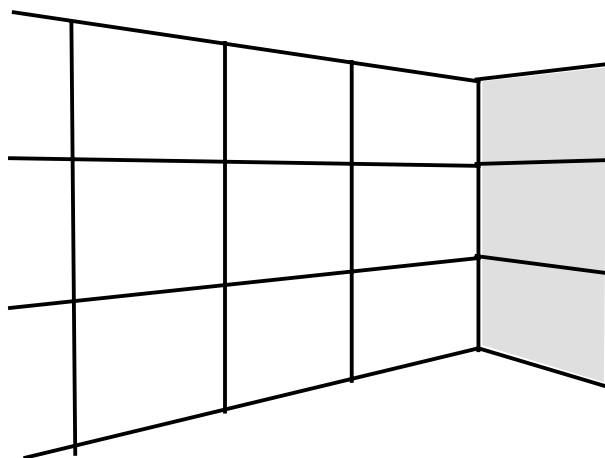


Figure 3.3: *An example of external ghost cells.*

3.3.3 Multiblock Capability and Ghost Cells

Further flexibility in modeling arbitrary geometries is provided by allowing for multiple computational blocks, which communicate to each other through internal boundaries. Internal and external boundaries are handled in EMI through the use of ghost cells. That is, there are two extra layers of cells surrounding each block. Figure 3.2 illustrates how the ghost cells work at an internal boundary. The figure shows a grid where four blocks meet. The block boundaries are shown in bold. The cells for block 1 are highlighted, with the ghost cells shaded. The ghost cells for block 1 map directly onto the last two rows of "real" cells of the surrounding blocks. The reason two rows are needed is to maintain second order accuracy in space in the hyperbolic fluxes at internal boundaries, because the flux at the $i + 1/2$ face depends on Q at i , $i + 1$, and $i + 2$. For the parabolic fluxes, only one layer of the ghost cells is used.

The corner ghost cells are only used by the parabolic fluxes. They are used when averaging to get the fluxes on the face-centered mesh at a corner. Since the parabolic fluxes only use one row, only one of the four ghost cells at each corner is used (the one diagonally adjacent to the real domain).

Ghost cells are also used to implement external boundary conditions. In the earlier versions of EMI, first order accurate (in space) boundary conditions were used, which required only a single row of ghost cells. The ghost cells on boundaries are created by reflecting the last real cell through the tangent to the boundary, as shown in Figure 3.3. When external ghost cells are created this way, the location of the corner ghost vertices can become somewhat difficult to determine for situations where two or three blocks meet. The grids used in this research were simple enough that this was not an issue. Later versions of EMI used second order boundary conditions, and thus required two rows of external ghost cells, just as for internal boundaries.

A number of different types of boundary conditions were used in the simulations. These included inflow, outflow, periodic, and perfectly conducting walls. The details of each boundary condition are explained separately for each application.

3.3.4 Parallel Implementation

The code was originally developed on a serial computer. It was parallelized by another graduate student, Bogdan Udrea. The parallelization was accomplished by means of domain decomposition, which fit naturally with the multiblock design of the serial code. The idea is that different parts of the domain are calculated on different processors. Boundary information is passed among the processors by means of message passing subroutines that pass the ghost cell values back and forth. The user must specify which blocks reside on which processor, and some post-processing is required to gather the output files from the different processors and create a single output file. Other than that, the fact that the code is parallel is transparent to the user.

The Message Passing Interface (MPI) [9, 33] standard was used for the message passing calls. The code was run primarily on the Maui High Performance Computing Center IBM SP2 using IBM's implementation of MPI (called MPL). It was also run locally using a public domain version of MPI called MPICH. The details of how all this was accomplished will appear in Bogdan Udrea's thesis.

Chapter 4

CODE VALIDATION TESTS

This chapter describes the tests that were done to benchmark or validate the code. The first test was the coplanar MHD Riemann problem. This problem was solved separately in the x , y , and z directions in order to benchmark the approximate Riemann solver. The oblique shock problem validated the approximate Riemann solver and the LU-SGS relaxation scheme in two-dimensions. This problem was solved in the $x - y$, $y - z$, and $x - z$ planes to check out the third dimension. The Hartmann flow test problem validated the parabolic terms. It also was exercised in all three coordinate directions. The computed results from each of these three test problems were compared against analytical solutions.

The fourth test problem, the magnetoplasdynamic (MPD) arc problem, was different in that it was not compared to an exact analytical solution. Instead, the unsteady solution was computed with the explicit and implicit schemes. This problem was used to check that all the parts of the implicit algorithm worked together correctly for time-dependent problems. The unsteady solution computed from the explicit scheme served as the "truth" data in this case.

4.1 1D Coplanar MHD Riemann Problem

This test problem was solved to verify that Harten's approximate Riemann solver worked for the MHD equations. One-dimensional ideal MHD (variations in x only) is described by a system of seven equations, since B_x is constant (in order to satisfy $\nabla \cdot \mathbf{B} = 0$). The coplanar MHD equations are obtained from the one-dimensional

ideal MHD equations by setting B_z and v_z to zero, thus allowing only planar flow and fields. This eliminates the $v_x \pm v_a$ eigenvalues, leaving a system of five equations with five eigenvalues. Mathematically, the Riemann problem is an initial boundary value problem in which there is initially a discontinuous jump in the data such that the left half of the domain is at one state and the right half of the domain is at another state. As the solution evolves in time, shock waves and rarefaction waves form that travel at speeds related to the wave speeds of the system. Although not physically realizable in plasmas, this problem is analogous to a shock tube in hydrodynamics.

For the full five-wave case, there is not a closed form analytical solution. Instead, the solution must be checked by calculating generalized Riemann invariants across the rarefaction waves and Rankine-Hugoniot jump conditions across the shock waves. Since this has already been done by Brio and Wu [3] for a specific set of conditions, we used those same initial conditions in order to allow direct comparison with their published solution. The initial left state was $p = 1$, $\rho = 1$, and $B_y = 1$. The initial right state was $p = 0.1$, $\rho = 0.125$, and $B_y = -1$. The velocities were zero and B_x was 0.75. Figure 4.1 shows the initial density and transverse magnetic field distributions and their numerical solution after 400 time steps on an 800 point grid with a CFL number of 0.8. Since there was no advantage to using the implicit scheme for this problem (due to the small time steps required for accuracy), the solution was computed using a simple first order (in time) explicit scheme given by

$$\frac{\mathbf{Q}_{ij}^{n+1} - \mathbf{Q}_{ij}^n}{\Delta t} = -(\mathbf{R}_h)_{ij}^n. \quad (4.1)$$

The solution clearly shows five waves formed corresponding to the five eigenvalues. They are a fast rarefaction wave, a slow shock, a contact surface moving to the right, a slow compound wave (rarefaction and shock), and a fast rarefaction wave moving to the left. Note that the numerical method is able to resolve the shocks over a few grid points without introducing numerical oscillations. The computed solution overlaid exactly on Brio and Wu's published solution.

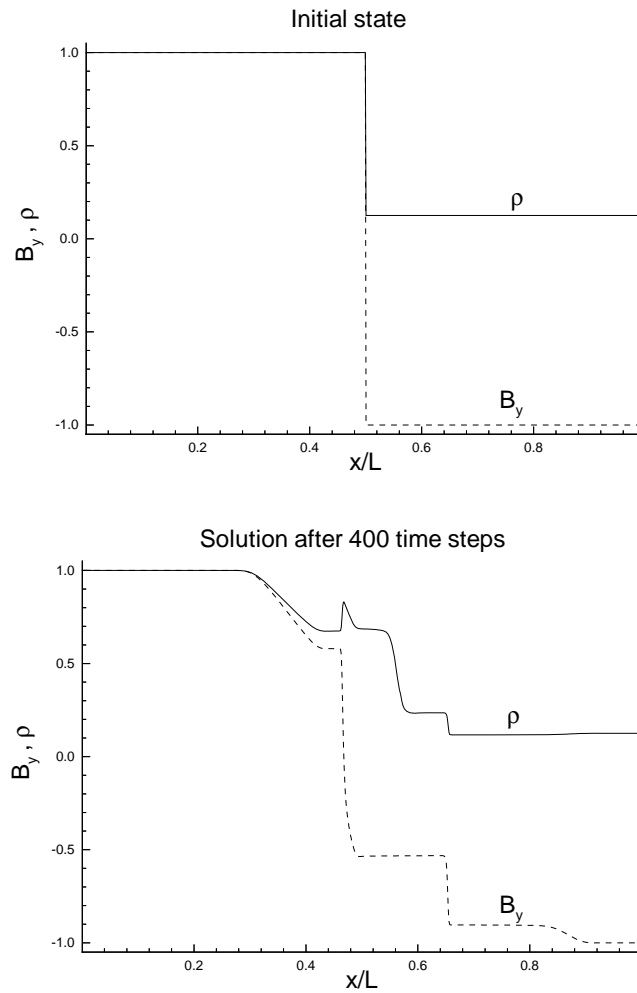


Figure 4.1: Numerical solution of coplanar Riemann problem. Density and transverse magnetic field are shown initially and after solution has evolved for 400 time steps.

If we set $B_x = 0$, then the problem reduces to a hydrodynamic shock tube problem if one replaces the thermodynamic pressure by the sum of the thermodynamic and magnetic pressures. For this case one can find a closed form exact solution to compare to the calculated solution. Figure 4.2 shows both the calculated and the exact solution for $p + B^2/2$ after 80 time steps on a 100 point grid. There is very good agreement with the plateau values and the shock is resolved in a few cells without numerical oscillations.

4.2 Oblique Shock

Figure 4.3 shows the geometry and boundary conditions for the oblique shock test problem. This steady-state problem served primarily as a test of our implementation of the approximate Riemann solver in two dimensions. In this problem \mathbf{v} and \mathbf{B} are initially zero in the interior of the domain. A super-Alfvénic ($M = 3$), inviscid, perfectly conducting plasma flows in from the left and upper boundaries, and impinges at an angle of 25 degrees on a perfectly conducting plate that forms the bottom boundary. The incoming flow carries with it a vertical field of $B_y = 0.2$. The boundary conditions for the perfectly conducting plate are $v_y = 0$ and $B_y = 0$. In general, boundary conditions were implemented by using a single layer of ghost cells surrounding the computational domain. The boundaries were treated explicitly. That is, the ghost cells were evaluated using values from the previous iteration. So, for example, $B_y = 0$ at the conducting plate was achieved by setting

$$(B_y)_{\text{ghost cell}} = -(B_y)_{\text{first cell}} \quad (4.2)$$

Figure 4.4 shows the steady-state solution of this problem. Contours of density and magnetic field lines are plotted. The density contours show that an oblique shock forms, as expected. Outside of the shock, the field is convected in from the boundary. At the shock, the field lines bend due to the change in direction of the flow at the shock. We verified that the divergence was less than 10^{-14} throughout the domain.

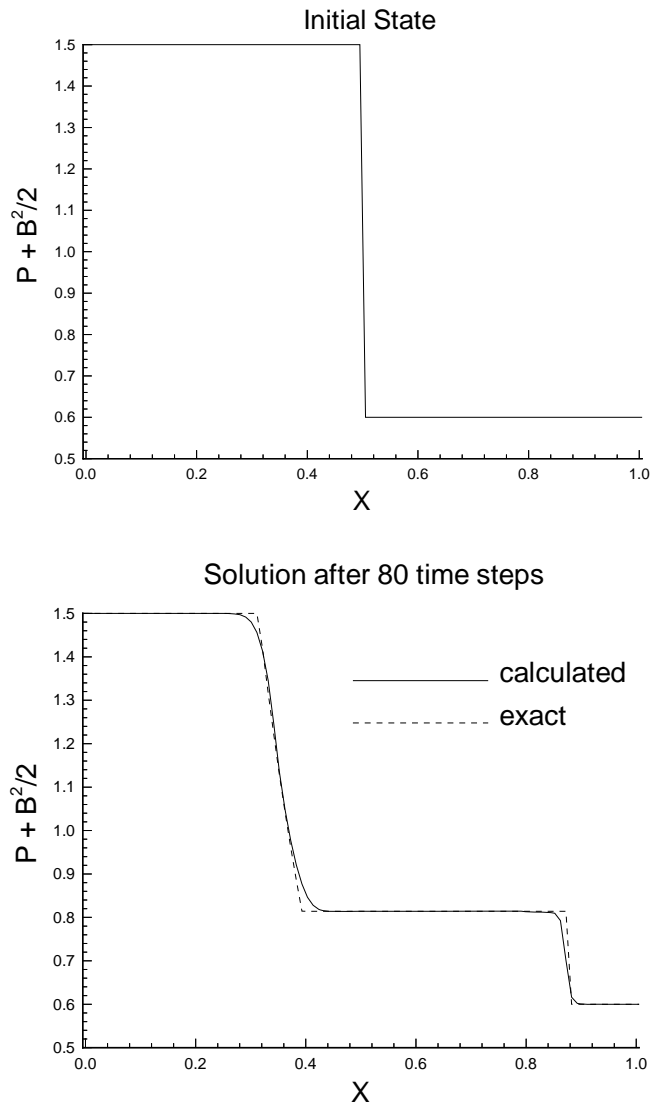


Figure 4.2: Comparison of numerical and exact solution of coplanar Riemann problem for $B_x = 0$ case.

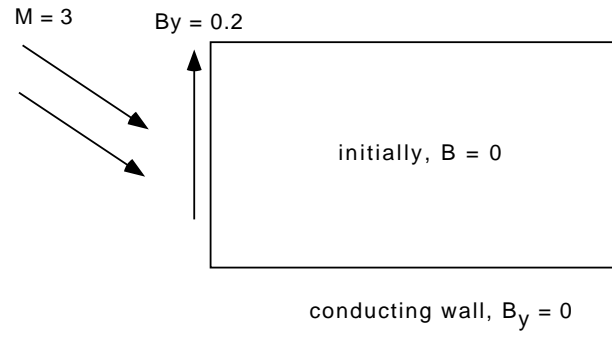


Figure 4.3: *Geometry and boundary conditions for oblique shock test problem.*

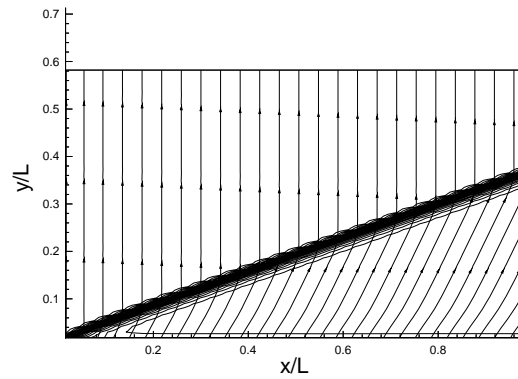


Figure 4.4: *Density contours and field lines for an $M = 3$ flow impinging on a perfectly conducting plate at an angle of 25 degrees.*

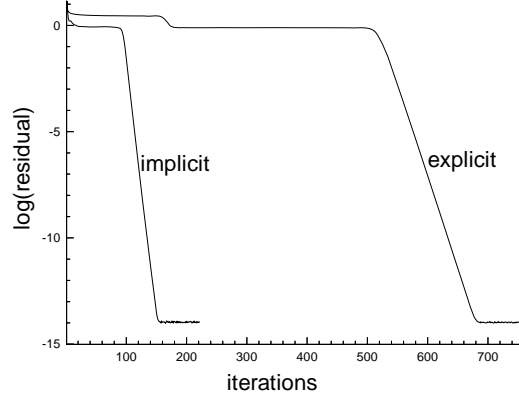


Figure 4.5: *Logarithm of the two-norm of the energy equation residual per cell plotted as a function of iteration number for explicit and implicit solutions of oblique shock problem.*

This solution was obtained from the implicit scheme, Eq. (3.11), with the parabolic terms and the time derivative terms all set to zero, so that the scheme becomes

$$\frac{\partial (\hat{\mathbf{R}}_h)_{ij}^m}{\partial \mathbf{Q}} [\mathbf{Q}_{ij}^{m+1} - \mathbf{Q}_{ij}^m] = -(\mathbf{R}_h)_{ij}^m. \quad (4.3)$$

To determine the efficiency of the implicit scheme as a steady-state solver, the oblique shock was also solved with the explicit scheme (Eq. (4.1)) at a CFL number of 0.8. It is recognized that Eq. (4.1) is not an optimized explicit scheme. It is used here simply to provide a rough benchmark for evaluating the implicit scheme. Figure 4.5 is a plot of the logarithm of the two-norm of the average residual of the energy equation as a function of the number of iterations (or time steps, in the case of the explicit scheme). In this case, the two-norm of the average residual is

$$\|R_{h,en}\|_2 = \frac{\sqrt{\sum_i \sum_j (R_{h,en})_{ij}^2}}{N_{cells}}. \quad (4.4)$$

The implicit scheme converged to 10^{-14} in about 150 iterations, whereas the explicit scheme required about 700 time steps (iterations). Since one implicit iteration takes

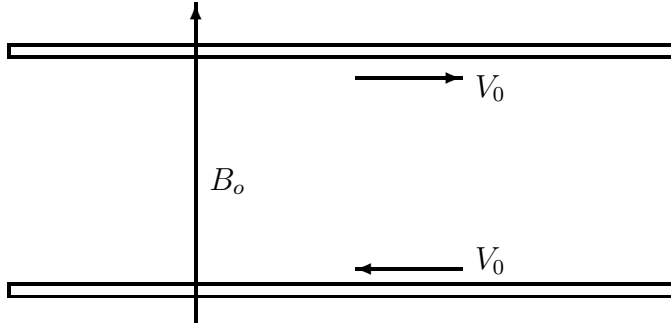


Figure 4.6: *The Hartmann flow geometry showing the moving parallel plates and the cross magnetic field.*

only about 10% more CPU time than one explicit time step, the implicit scheme required roughly four times less CPU time than the explicit scheme to converge to the steady-state solution.

4.3 Hartmann Flow

The validation of the parabolic terms consisted of applying the code to the Hartmann flow problem, whose steady-state solution can be solved analytically. The problem geometry is shown in Figure 4.6. It consists of two infinite parallel plates, each moving in opposite directions at velocity V_0 , and with an in-plane magnetic field B_o between them that is normal to the plates in the y direction.

For the case with equal and opposite plate velocities, the Hartmann flow is described by the differential equations

$$\frac{\partial^2 v_x}{\partial y^2} - \left(\frac{H^2}{L^2} \right) v_x = 0 \quad (4.5)$$

$$\frac{\partial B_x}{\partial y} = -(Rm) v_x, \quad (4.6)$$

where H , the Hartmann number, is

$$H \equiv \frac{B_o L}{\sqrt{\rho \nu \eta}} = \sqrt{(Re_A)(S)}, \quad (4.7)$$

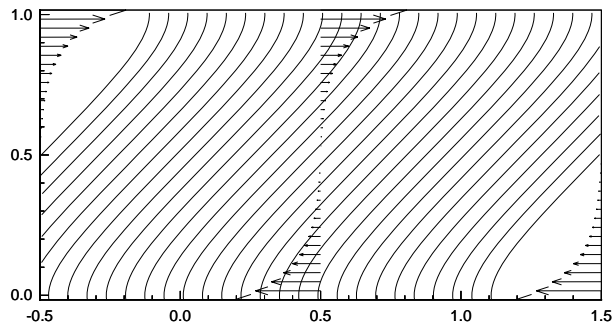


Figure 4.7: *Hartmann flow simulation with $H = 10$. Flow velocity vectors and magnetic field lines are shown. The flow velocity only exists close to the plates. The magnetic field lines are linear around the midplane.*

and Rm , the magnetic Reynolds number, is

$$Rm \equiv \frac{\mu_o LV}{\eta}. \quad (4.8)$$

Figure 4.7 shows the results from a simulation with $H = 10$. Near the plates there is a boundary layer with a scale length of L/H in which the velocity falls off rapidly to zero and the field develops a swayed shape as it is dragged by the fluid. In the center region of the channel the velocity is zero and the field has a uniform slope.

We also ran simulations at extremely large and small Hartmann numbers. In the limit of small Hartmann number the boundary layer extends to the opposite wall and a linear velocity profile develops, as shown in Figure 4.8. For large Hartmann numbers, the boundary layer shrinks to zero and the field has a uniform slope throughout the channel, as shown in Figure 4.9. In the cases shown here we found that the error between the computed solution and the analytical solution (as measured by the two-norm of the energy error vector normalized by the number of cells) was less than 10^{-6} .

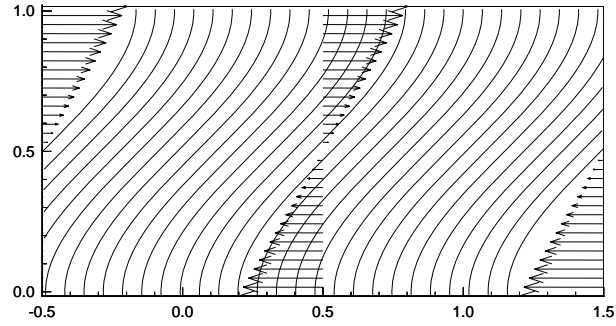


Figure 4.8: *Hartmann flow simulation with $H = 0.1$. Flow velocity vectors and magnetic field lines are shown. The velocity profile is linear and the magnetic field lines have an “S” shape caused by the bulk fluid flow.*

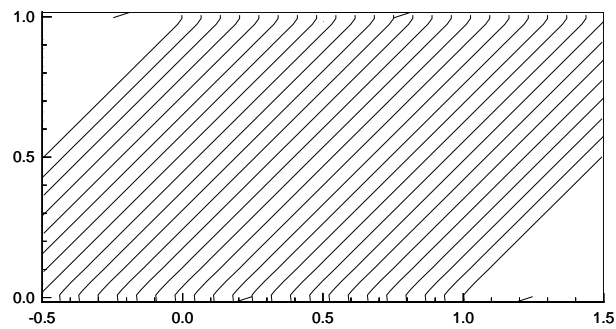


Figure 4.9: *Hartmann flow simulation with $H = 10^4$. Flow velocity vectors and magnetic field lines are shown. The velocity of the flow is zero everywhere except at the plates. The magnetic field lines have a constant slope through the domain.*

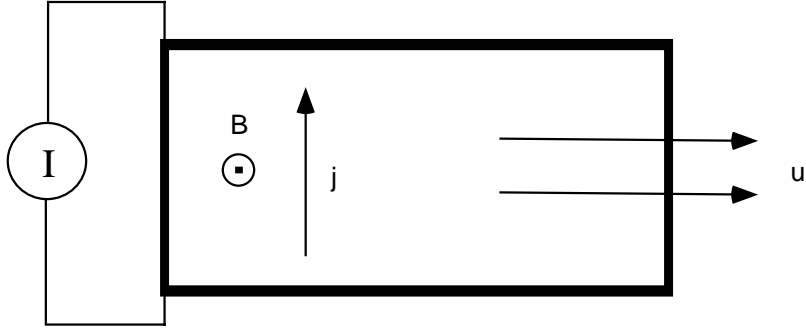


Figure 4.10: *Geometry of the MPD arc problem.*

4.4 MPD Arc

The three problems described to this point show that the various pieces of the algorithm work correctly, and that the implicit solver is an efficient relaxation scheme for steady problems. This next example shows the utility of the scheme for unsteady problems, which is the primary purpose of the algorithm. The problem geometry is shown in Figure 4.10. In this figure, positive x is to the right, positive y is up, and positive z is out of the page. Initially, \mathbf{v} and \mathbf{B} are zero throughout the computational domain, which is indicated by the thick-lined rectangle. A current sheet, with a total current per unit z of I , is applied to the left boundary ($x = 0$). The current density is

$$j_y = -\frac{\partial B_z}{\partial x}. \quad (4.9)$$

The current sheet at the left boundary is applied by setting $B_z = \mu_o I$ in the ghost cells, so that the initial current sheet is represented by the jump in B_z between the ghost cell and the first interior cell. The resulting $\mathbf{j} \times \mathbf{B}$ force accelerates the plasma in the positive x direction, so that it flows out at the right boundary. The upper and lower boundaries are perfectly conducting walls, whose boundary conditions were discussed previously. There is no viscosity, so the problem is one-dimensional in x .

For a Lundquist number much larger than one and a relatively low current that

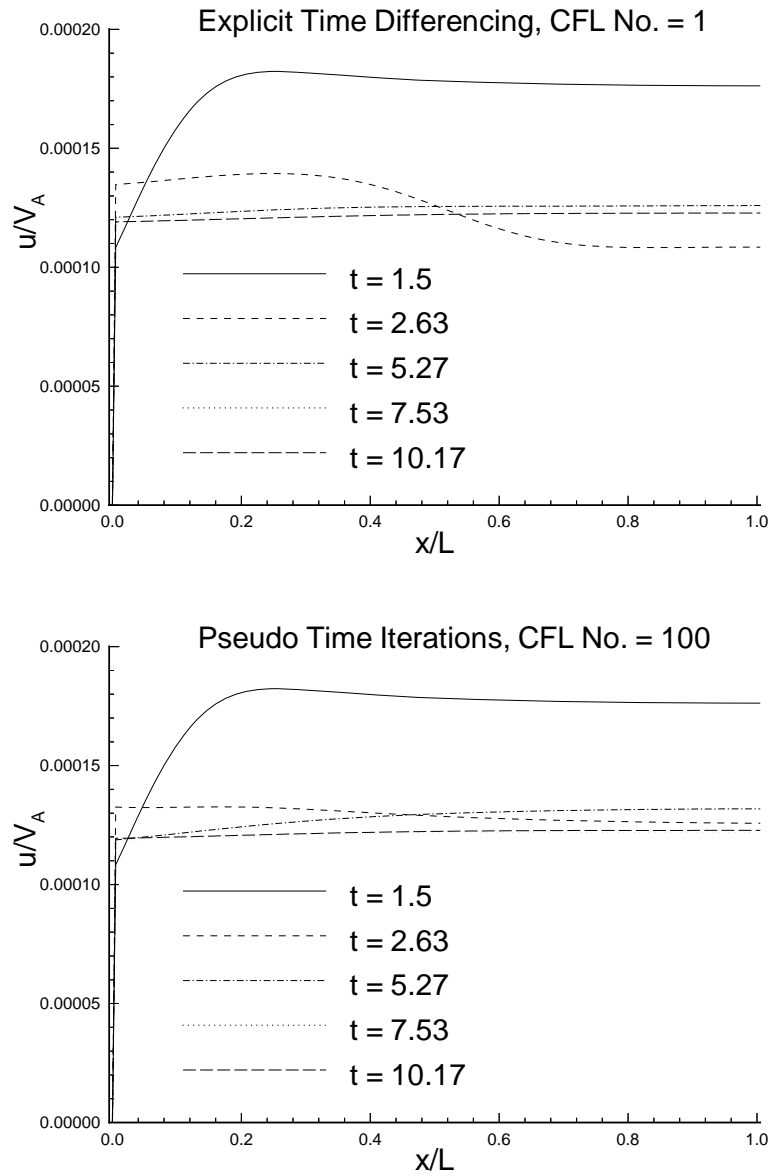


Figure 4.11: Comparison of explicit and implicit solutions of plasma velocity versus x at several times for MPD arc problem.

results in sub-Alfvénic flow, the plasma will be accelerated up to some exit velocity on a fast hydromagnetic time scale, while the magnetic field and current will be convected more slowly into the domain. The behavior of the plasma will be affected by convection and diffusion. In the limit of zero resistivity (ideal MHD), the current sheet will remain unmoving at the left boundary because no field will diffuse into the domain, and the boundary condition of zero velocity at the left boundary will keep the current sheet from being convected in. However, with a finite resistivity, some of the current sheet will diffuse into the domain, and then it will be convected by the accelerated plasma to the right. It is assumed for the purposes of this demonstration problem that we are only interested in the long-time behavior of the solution, and are willing to smear over some of the details of the faster dynamics by advancing the solution at a large CFL number. This problem was solved with the time-accurate implicit scheme at a CFL number of 100. Recall that the time-accurate implicit scheme is

$$\left[\frac{\partial (\mathbf{R}_h)_{ij}^m}{\partial \mathbf{Q}} + \frac{3\mathbf{I}}{2\Delta t} \right] \Delta \mathbf{Q}_{ij}^m = - \left[(\mathbf{R}_h)_{ij}^m + (\mathbf{R}_p)_{ij}^m + \left(\frac{\partial \mathbf{Q}}{\partial t} \right)_{ij}^m \right]. \quad (4.10)$$

For comparison purposes, the problem was also solved at a CFL number of 1 using the first order explicit scheme given by

$$\frac{\mathbf{Q}_{ij}^{n+1} - \mathbf{Q}_{ij}^n}{\Delta t} = - \left[(\mathbf{R}_h)_{ij}^n + (\mathbf{R}_p)_{ij}^n \right]. \quad (4.11)$$

Figure 4.11 shows the evolution of the plasma velocity profile as computed by explicit and implicit schemes. The upper plot shows the implicit computation at a CFL number of 100, while the lower plot is the result of the explicit calculation at a CFL number of 1. Both schemes show that the entire domain has been accelerated up to near the steady plasma velocity after about one Alfvén time, based on the length of the accelerator channel. This is just the time for the fast magnetosonic wave to propagate the length of the channel. After that, the velocity profile oscillates somewhat as the initially uniform density and pressure profiles adjust themselves as

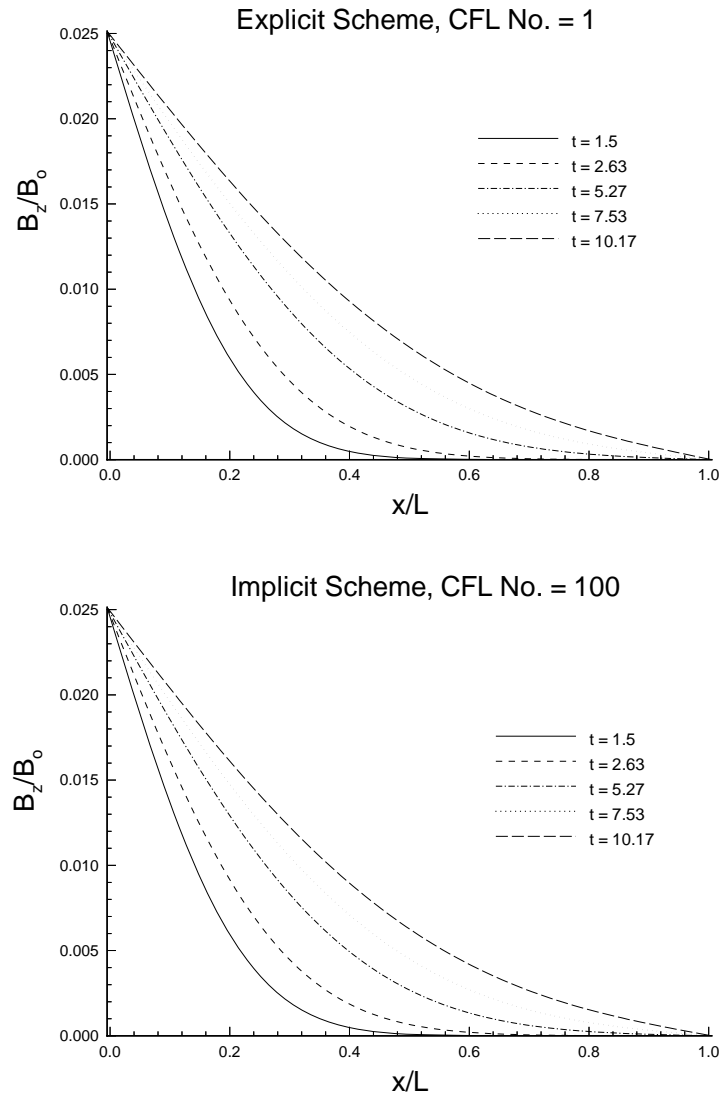


Figure 4.12: Comparison of explicit and implicit solutions of B_z versus x at several times for MPD arc problem.

a quasi-steadystate is approached. During this time, the implicit scheme loses some accuracy as compared to the explicit scheme, so the velocity profiles do not exactly match. However, by ten Alfvén times, both simulations find the same quasi-steady profile.

Figure 4.12 shows the evolution of the magnetic field for the same simulations. Again, the upper plot is the implicit result, while the lower one is the explicit result. The long-time behavior of the convection of the magnetic field across the domain is captured equally well at the larger CFL number, as expected. The explicit simulation took 2600 time steps to advance the solution to $t = 10.17$, while the implicit simulation required 26 time steps. However, the implicit scheme required approximately 30 iterations at each time step, and each iteration required about 10% more CPU time than an explicit time step. So, in terms of CPU time, the implicit scheme ran roughly three times faster than the explicit scheme for this problem.

Chapter 5

PLANAR SHEET PINCH

This chapter describes the first of the two applications in which we used the code to explore physics issues related to relaxation. The planar sheet pinch is a simple geometry for studying one of the basic instabilities in a resistive plasma: resistive tearing modes. There are several reasons we chose to study this system. First, it serves as another validation of the code, because the linear growth rates of the tearing mode can be computed analytically. However, it is more than just a validation, because when the modes grow into the nonlinear regime, there is no analytical solution to compare with. The second reason is that the tearing mode is the prototype of the type of resistive instability that could drive relaxation in configurations such as the HIT experiment that have highly non-uniform current distributions.

The geometry for the planar sheet pinch is shown in Figure 5.1. Initially, there is a planar current sheet of infinite extent, and with a finite thickness in the out-of-plane coordinate. Over time, an instability develops in which the current sheet breaks into filaments, as indicated by the thick arrows in the figure. The breakup into current filaments involves the tearing and reconnection of magnetic field lines, which is why it is called a tearing mode.

Assume that the current sheet lies in the x - z plane. In the most general case, the sheet current and the equilibrium magnetic field on either side of the current sheet are at some arbitrary angle with respect to each other, as shown in the figure. Without loss of generality we let the current and fields be perpendicular, with $B_0 = B_{0x}(y)$ and the current in the z direction. Figure 5.2 shows the equilibrium magnetic field, B_{0x} , and geometry that was used for the simulations presented in this section. Here,

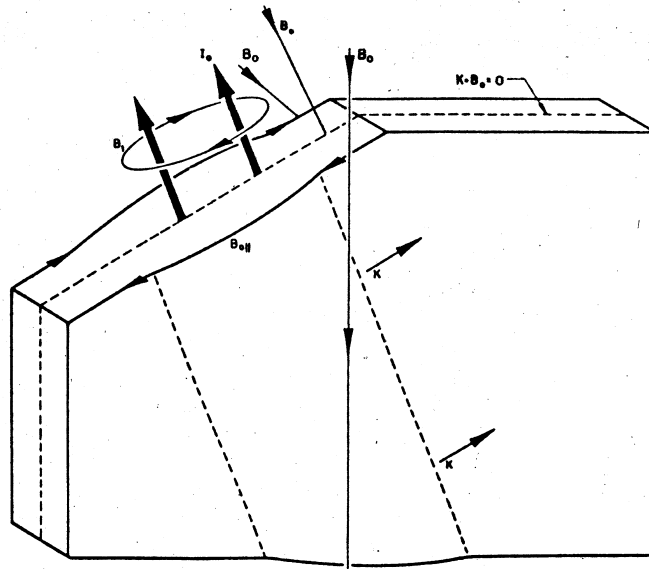


Figure 5.1: *Planar sheet pinch geometry.*

a is the half-width of the current sheet.

5.1 Background

The planar sheet pinch resistive tearing instability was first treated analytically by Furth, Killeen, and Rosenbluth [13]. Interest in resistive tearing modes dates back to the early years of controlled fusion, when it became apparent that ideal MHD theory could not explain the gross instabilities that were observed in some experiments [12]. The basic mechanism in a sheet pinch instability is that a thin sheet of current breaks into individual current filaments. This instability is found to grow on a time scale longer than the Alfvén time, but shorter than the resistive diffusion time.

The basic idea behind the theory of Furth, Killeen, and Rosenbluth is that as the Lundquist number is increased, the resistivity acts over an increasingly thin layer at the center of the current sheet. They solve for the linear growth rate of the tearing

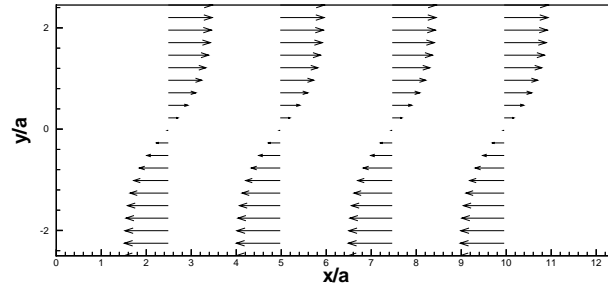


Figure 5.2: *Equilibrium magnetic field for sheet pinch calculations.*

mode by matching asymptotic solutions from an outer layer governed by ideal MHD, and a thin inner layer where the resistivity becomes important.

An alternative way to find the linear growth rates is to solve the incompressible, resistive MHD equations as an initial value problem. This has the added advantage of also yielding the eigenmodes as well as the growth rates. This technique is described in Killeen and Shestakov [18] and is shown below.

In order to find the nonlinear behavior of the tearing mode, the full, nonlinear MHD equations must be solved. This is the approach taken in our work. We follow the approach taken by Schnack and Killeen [29]. In their approach, the initial equilibrium was perturbed with the eigenmode solution of the linear equations, and then the solution was advanced in time until the mode saturated. The details of this technique will be explained in more detail in the following sections.

5.2 Linear Analysis to Determine Eigenmode

For the linear analysis, we assume that the plasma is incompressible and that the ion pressure is negligible. Then the plasma is governed by the following equations:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) - \nabla \times (\eta \nabla \times \mathbf{B}), \quad (5.1)$$

$$\nabla \times \left[\rho \frac{d\mathbf{v}}{dt} \right] = \nabla \times [\nabla \times (\mathbf{B} \times \mathbf{B})], \quad (5.2)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (5.3)$$

$$\nabla \cdot \mathbf{v} = 0. \quad (5.4)$$

For the linear approximation, we write the magnetic field as the sum of a background field, \mathbf{B}_0 , and a perturbed field, \mathbf{B}_1 . Similarly, the velocity and resistivity are also written as the sum of a background value and a perturbation. However, for the tearing mode, the perturbation in η is neglected. The zeroth order plasma equilibrium is specified by $\mathbf{B}_{0x}(y)$, $\eta_0(y)$, $p(y)$, and $v_0(y)$. Here we take $v_0 = 0$ and $d\mathbf{B}/dt = 0$. Ohm's law is

$$\mathbf{E}_0 = -\mathbf{v}_0 \times \mathbf{B}_0 + \eta_0 \mathbf{j}_0, \quad (5.5)$$

where $\mathbf{j}_0 = (\nabla \times \mathbf{B}_0) / \mu_0$. Taking the curl of both sides and noting that $\partial \mathbf{B}_0 / \partial t = 0$, the result is

$$\nabla \times (\eta_0 \nabla \times \mathbf{B}_0) = 0 \quad (5.6)$$

The other zeroth order equilibrium equations are

$$\nabla \cdot \mathbf{B}_0 = 0, \quad (5.7)$$

$$\mathbf{j}_0 \times \mathbf{B}_0 = 0. \quad (5.8)$$

Similarly, the first order equations are

$$\frac{\partial \mathbf{B}_1}{\partial t} = \nabla \times (\mathbf{v}_0 \times \mathbf{B}_1 + \mathbf{v}_1 \times \mathbf{B}_0) - \nabla \times (\eta_0 \nabla \times \mathbf{B}_1), \quad (5.9)$$

$$\rho_0 \left[\frac{\partial \mathbf{v}_1}{\partial t} + (\mathbf{v}_0 \cdot \nabla) \mathbf{v}_1 \right] = \nabla \times [(\mathbf{B}_0 \cdot \nabla) \mathbf{B}_1 + (\mathbf{B}_1 \cdot \nabla) \mathbf{B}_0], \quad (5.10)$$

$$\nabla \cdot \mathbf{B}_1 = 0, \quad (5.11)$$

$$\nabla \cdot \mathbf{v}_1 = 0. \quad (5.12)$$

We assume perturbations of the form $f_1(y, t) \exp(ikx)$. Substituting this form into the above equations, they can be separated into a pair of equations that involve only B_{y1}

and v_{y1} . These equations are put into dimensionless form by defining the following dimensionless variables:

$$\bar{y} = y/a, \quad (5.13)$$

$$\bar{t} = t/\tau_\eta, \quad (5.14)$$

$$\psi = B_{y1}/B, \quad (5.15)$$

$$w = -iv_{y1}k\tau_\eta, \quad (5.16)$$

$$B_{x0}/B, \quad (5.17)$$

$$F = B_{x0}/B, \quad (5.18)$$

$$\eta = \eta_0/\eta_{ref}, \quad (5.19)$$

and

$$\rho_0/\rho_{ref} = 1. \quad (5.20)$$

The resulting differential equations are

$$\frac{\partial \psi}{\partial \bar{t}} = \eta \left(\frac{\partial^2 \psi}{\partial \bar{y}^2} - \alpha^2 \psi \right) - Fw, \quad (5.21)$$

and

$$\frac{1}{\alpha^2 S^2} \frac{\partial}{\partial \bar{t}} \left(\frac{\partial^2 w}{\partial \bar{y}^2} - \alpha^2 w \right) = F \left(\frac{\partial^2 \psi}{\partial \bar{y}^2} - \alpha^2 \psi \right) - F''\psi, \quad (5.22)$$

where

$$\alpha = ka. \quad (5.23)$$

There are two free parameters for this equation set: the normalized wave number, α , and the Lundquist number, S . Thus, these equations govern the behavior of a particular mode, specified by α , for a particular Lundquist number.

The equilibrium field and resistivity must satisfy the zeroth order induction equation. We choose the functional form suggested by Furth that leads to simpler forms for the analytical solution. The field is

$$F(\bar{y}) = \tanh \bar{y}, \quad (5.24)$$

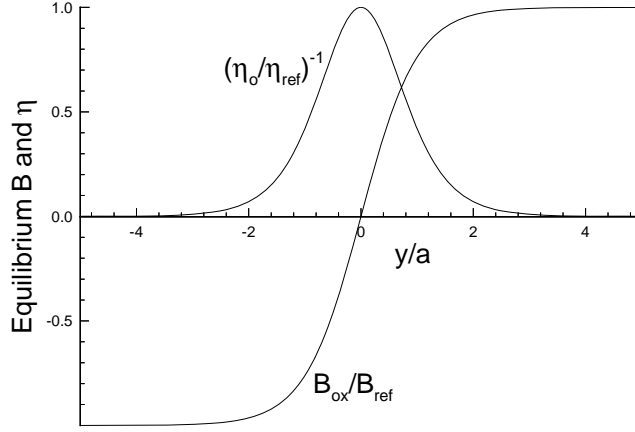


Figure 5.3: *Equilibrium B and η that satisfy zeroth order induction equation.*

and the resistivity is

$$\eta(\bar{y}) = \cosh^2 \bar{y}. \quad (5.25)$$

These are plotted in Figure 5.3. We must also satisfy the zeroth order momentum equation. When $F(\bar{y})$ is substituted into the momentum equation, the result is that we must have

$$p(\bar{y}) = \frac{1}{2} + \frac{1}{2} [1 - \tanh^2(\bar{y})]. \quad (5.26)$$

Note that p here is normalized by $B^2/2$, so that this profile gives $\beta = 1$ at $\bar{y} = 0$.

5.2.1 Solutions of Linearized Equations

The procedure for finding the linear eigenmodes is as follows. First, a wave number and Lundquist number are selected. Then, starting from an arbitrary initial guess, the equations are integrated forward in time until the solution at each point is growing at the same exponential growth rate (to within some epsilon, usually 10^{-6}). The equations are discretized using a finite difference formulation. The second derivatives

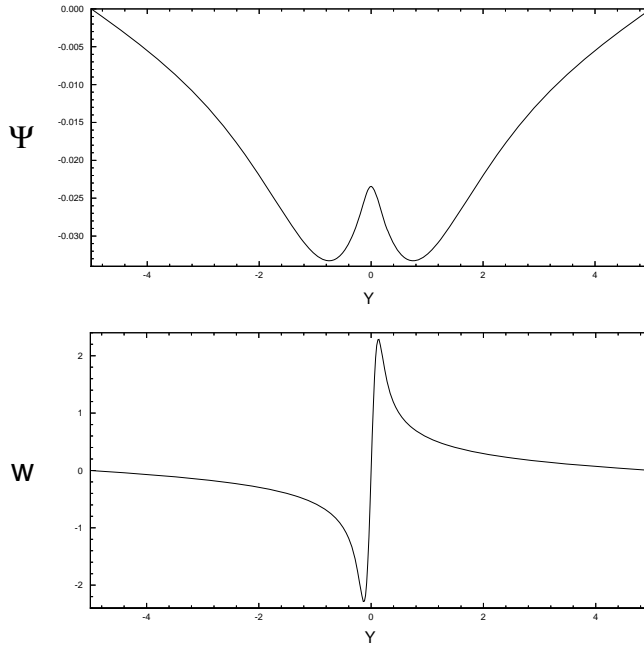


Figure 5.4: *Eigenmodes for symmetric tearing mode for $\alpha = 0.5$ and $S = 10^3$.*

in y are discretized using central differencing. For example

$$\frac{\partial^2 \psi}{\partial y^2} \approx \frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta y^2}. \quad (5.27)$$

Similarly, the time derivatives are evaluated using forward differencing. The equations are cast in implicit form to allow rapid integration regardless of S . Note that for modest peak values of S at $y = 0$, it is easy to have $S \ll 1$ far away from the current sheet, which can lead to very small time steps for an explicit scheme. When the equations are cast in implicit form, they can be written as

$$[A] \begin{bmatrix} \phi \\ w \end{bmatrix}_{i-1} + [B] \begin{bmatrix} \phi \\ w \end{bmatrix}_i + [C] \begin{bmatrix} \phi \\ w \end{bmatrix}_{i+1} = [D]. \quad (5.28)$$

This is a block tridiagonal system and can be solved using standard techniques. At each point the growth rate was calculated as

$$p = \frac{\psi^{n+1} - \psi^n}{(\psi^n + \psi^{n+1})/2}. \quad (5.29)$$

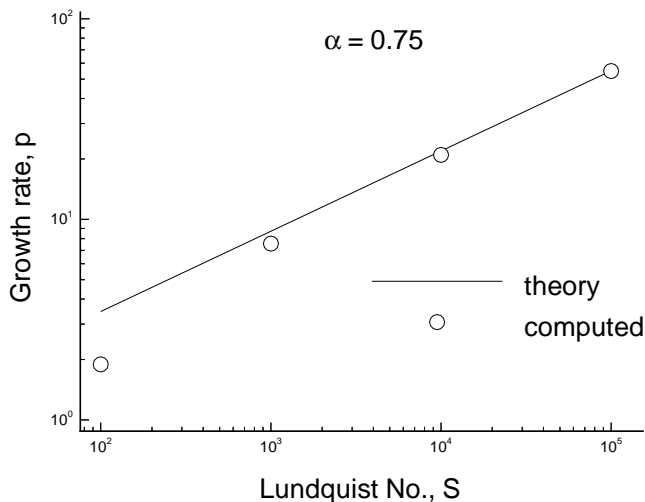


Figure 5.5: *Computed linear growth rates agree with theory at high S .*

When the difference between the minimum and maximum growth rates was less than 10^{-6} , then the solution was considered converged. A typical solution is shown in Figure 5.4. This is for $S = 10^3$, $\alpha = 0.5$, and with conducting walls at $\bar{y} = \pm 5$. Roughly speaking, the layer near the $\bar{y} = 0$ plane, where the velocity perturbation changes sign, corresponds to the resistive layer in the analytical theory. This layer becomes thinner and more difficult to resolve as S is increased.

In order to confirm that the eigenmodes were correct, the growth rates computed as above are compared with the analytical values from the theory of Furth, et al. As expected, the agreement between the two improves as S is increased, as is shown in Figure 5.5. This is the expected result since the theory is strictly true only in the limit as S approaches infinity.

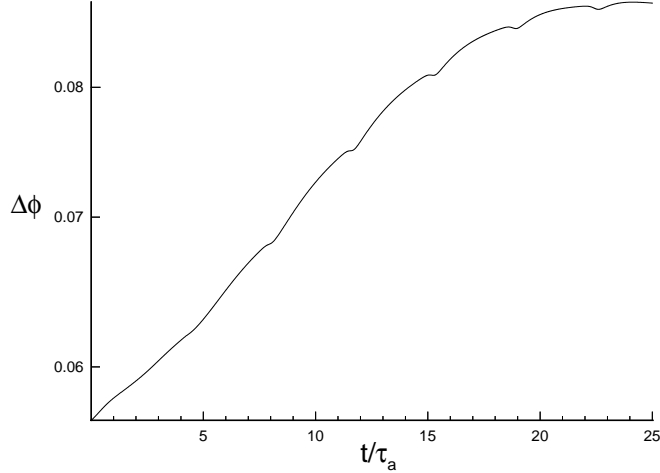


Figure 5.6: *Reconnected flux (dimensionless) versus time during nonlinear evolution of symmetric tearing mode at $\alpha = 0.5$ and $S = 10^2$.*

5.3 Nonlinear Analysis Using EMI

In order to apply the linear eigenmodes as an initial condition for EMI, two additional steps are required. First, the other two components of the perturbed vectors, B_{x1} and v_{x1} , must be computed. These are obtained by applying $\nabla \cdot \mathbf{B}_1 = \mathbf{0}$ and $\nabla \cdot \mathbf{v}_1 = \mathbf{0}$. Using these relations one can show that

$$B_{x1} = -\frac{\psi}{\alpha} \sin(2\pi\bar{x}). \quad (5.30)$$

and

$$v_{x1} = -\frac{w'}{S\alpha^2} \cos(2\pi\bar{x}). \quad (5.31)$$

Second, the amplitudes of the perturbation are rescaled so that the maximum perturbation is about 0.01 of the equilibrium value.

For the following nonlinear calculation, we used $S = 10^2$, $\alpha = 0.5$, and placed conducting walls at $\bar{y} = \pm 2.5$. The viscosity was zero. This was solved on a two-dimensional grid with 100 cells in the \bar{x} direction, and 200 cells in the \bar{y} direction. In

the \bar{x} dimension, the grid ran from 0 to $2\pi/\alpha = 12.57$, which corresponds to one full period of the mode.

5.3.1 Boundary Conditions

For this calculation, first order boundary conditions were used. At the left and right boundaries, periodic boundary conditions were applied. If the indices of the real cells in the x direction run from $i = 2$ to $i = icels - 1$, and the left and right ghost cells are labeled $i = 1$ and $i = icels$, respectively, then the periodic boundary conditions are

$$\mathbf{Q}_1 = \mathbf{Q}_{icels-1} \quad (5.32)$$

and

$$\mathbf{Q}_{icels} = \mathbf{Q}_2. \quad (5.33)$$

The upper and lower boundaries are perfectly conducting walls. The magnetic field boundary conditions at the lower boundary, for example, are

$$(B_y)_1 = -(B_y)_2 \quad (5.34)$$

and

$$(B_x)_1 = (B_x)_2. \quad (5.35)$$

For the velocity, the boundary conditions are

$$(v_y)_1 = -(v_y)_2 \quad (5.36)$$

and

$$(v_x)_1 = (v_x)_2. \quad (5.37)$$

Finally, the density and pressure are copied from the last real cell into the ghost cell. That is,

$$\rho_1 = \rho_2 \quad (5.38)$$

and

$$p_1 = p_2. \quad (5.39)$$

5.3.2 Results

There are several possible ways in which to measure the growth rate of the tearing mode for these calculations. We chose to follow Schnack and Killeen [29] by computing the growth of the "reconnected flux." As the tearing mode grows, lines of flux that originally spanned the entire domain tear and reconnect, forming magnetic islands in the vicinity of the developing current filaments. Thus, a convenient global measure of the development of the tearing instability is growth rate of the reconnected flux, $\Delta\phi$, which is defined as

$$\Delta\phi = \phi_{max} - \phi_{min}, \quad (5.40)$$

where

$$\phi(x) = \int_{-y_w}^0 B_x(0, y') dy' - \int_0^x B_y(x', 0) dx'. \quad (5.41)$$

Figure 5.6 is a plot of the growth rate as a function of time. It starts out very close to the rate predicted from linear theory, but then oscillates about the linear rate during the linear phase of the growth. This unsteadiness in the growth rate is due to slight plasma oscillations caused by the plasma being slightly out of equilibrium at the beginning of the calculation, presumably due to roundoff errors. The amplitude of the oscillations in the growth rate were reduced as the grid was refined. After the mode had been growing for about 15 Alfvén times, the mode begins to saturate as the amplitudes become large enough that the magnetic pressure in the islands balances the attraction of the current filaments. By 25 Alfvén times, the mode has stopped growing. Figure 5.7 is a plot of flux contours at saturation. The formation of magnetic islands is evident. This solution was calculated using the implicit scheme at a CFL number of 10.

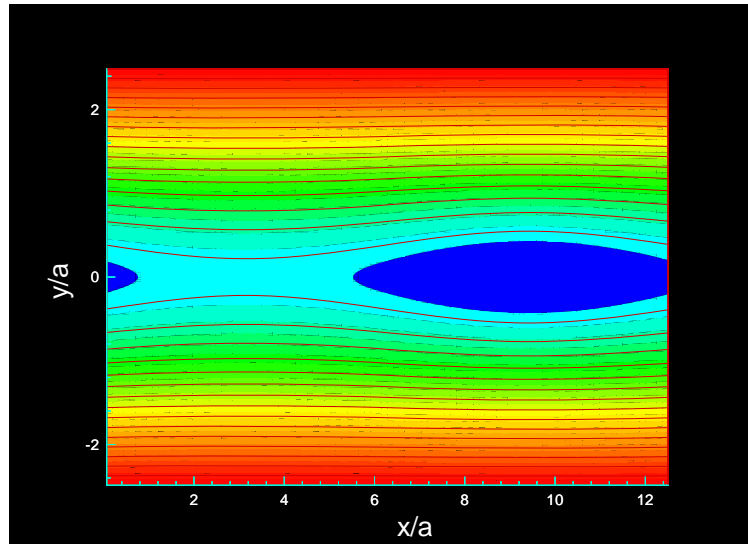


Figure 5.7: *Magnetic flux contours for saturated tearing mode at $\alpha = 0.5$ and $S = 10^2$.*

Chapter 6

3D PLASMA RELAXATION

In this chapter the relaxation of simple toroidal plasmas is studied. This is the first three-dimensional plasma physics problem that was studied in detail using the new code. The plasma configuration does not correspond to any particular fusion device. Rather, it is a generic configuration for studying general principles of magnetic relaxation that should be relevant to a variety of fusion and other applications. It is hoped that this problem could serve as a pathfinder for eventual simulations of actual fusion experiments, such as the HIT experiment.

The geometry is a torus with a square cross-section. Figure 6.1 shows one of the two toroidal configurations that were studied. The characteristic length used in all of the nondimensional numbers in this section is the length, L , of the sides of the square. Another important parameter is the minor "radius" of the torus, which we define as $a = L/2$. The major radius is $R_o = \frac{1}{2}(R_{min} + R_{max})$. The aspect ratio of the torus is defined as the ratio R_o/a . It is important to note that all the quantities plotted in this chapter are normalized in the manner described in chapter 2. For example, the poloidal flux is normalized by $B_o L^2$, the helicity by $B_o^2 L^4$, the energy by $B_o^2/(2\mu_o)$, and so forth.

Initially, the plasma consists of a core of purely toroidal field surrounded by a purely poloidal field. Figure 6.2 is a poloidal cross-section with contours of the toroidal field overlayed with poloidal field vectors. The toroidal and poloidal fields are separated by a current sheet. The density and pressure are uniform. The plasma is roughly in equilibrium, because the magnetic pressure is balanced across the current

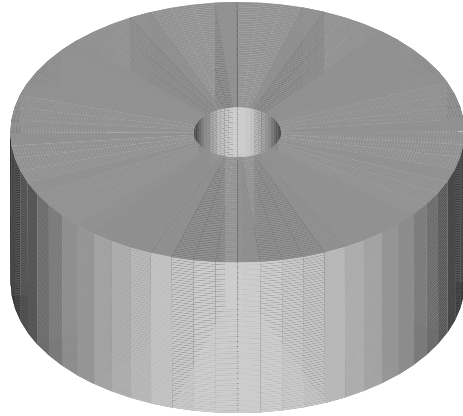


Figure 6.1: *Torus with square cross-section and an aspect ratio (R_o/a) of 1.5.*

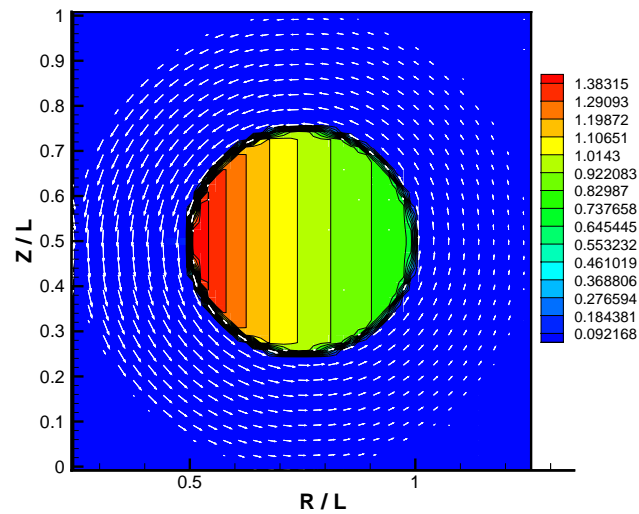


Figure 6.2: *Poloidal slice at $\phi = 0$ showing contours of toroidal field overlaid with poloidal field vectors at $t = 0$.*

sheet. The toroidal field in the core is

$$B_\phi = \frac{B_o R_o}{R}. \quad (6.1)$$

The formula for the poloidal field is simplest in "tokamak" coordinates r and θ , where r is the distance from the magnetic axis (in a poloidal plane), and θ is the counter-clockwise angle about the magnetic axis. Then, the poloidal field is

$$B_R = - \left[\frac{B_o R_o}{R} \right] \left[\frac{r_o}{r} \right] \sin \theta, \quad (6.2)$$

where $r_o = a/2$. In the corners of the square ($r > a$) \mathbf{B} is zero, so the plasma is obviously not in equilibrium near the corners.

The plasma configuration we have chosen to study is somewhat unusual. It is similar to the configuration modeled by Shumlak, et al. [32] It is essentially an equilibrium for a torus of circular cross-section that is placed into a torus of square cross-section. The reason for this choice of mixed geometries is simplicity. That is, the equilibrium for a torus of circular cross-section is much simpler to calculate than that for a square cross-section, whereas the force-free Taylor state that the configuration evolves toward is much simpler in the square geometry. Another advantage is that the helicity, which is an important quantity to track, is very easy to picture and calculate initially since there is one large toroidal flux tube linking the poloidal flux.

Physically, the configuration corresponds most closely to a "reverse field pinch" (RFP) experiment. Imagine that initially there is a purely toroidal field created by some set of external coils. Then, a toroidal current is driven along the outer part of the plasma, creating the poloidal field that pinches and compresses the trapped toroidal field, which moves inward. At this point all the coils are turned off, and the plasma is allowed to decay. This is roughly the situation described by the initial plasma condition (supposing it were bounded by a circular, rather than a square, conducting shell). Another important aspect of this configuration is the fact that the current lies initially in a sheet between the toroidal and poloidal fields, which

roughly approximates the situation in HIT when the current is driven on the outer (open) flux surfaces. As seen in the previous chapter, a current sheet is subject to tearing instabilities that certainly provide a mechanism for a plasma to relax in the Taylor sense toward a force free state. However, it must be noted that the simulations described here were not long enough to allow resistive modes time to grow very much.

Another very important feature of this configuration is that the "q" profile. In toroidal devices, the q parameter measures the amount of helical twist in the field lines. It is defined as the number of toroidal circuits required for a field line to make one poloidal circuit. Initially q is infinite in the core of purely toroidal field, whereas it is zero in the shell of purely poloidal field. Clearly, q must go through 1 in the current sheet separating the two layers. It turns out that the $q = 1$ surface is unstable to ideal MHD kink modes. This instability turns out to be important in the high Lundquist number runs shown below.

6.1 Background

Before we plunge into the details of the calculations and the results, we will revisit some of the theoretical issues that were introduced back in the first chapter. Recall that Taylor's theory [36] states that a plasma will tend to evolve or "relax" over time toward a state resulting from the minimization of the magnetic energy,

$$W_B = \frac{1}{2} \int B^2 d\tau, \quad (6.3)$$

subject to the constraint that the total magnetic helicity,

$$K = \int \mathbf{A} \cdot \mathbf{B} d\tau, \quad (6.4)$$

is an invariant. Physically, this means that the magnetic energy is reduced through tearing and reconnection of field lines much more rapidly than the helicity resistively decays, so that on the time scale of the changes in magnetic energy, the helicity is approximately invariant.

Because of the success of the theory in predicting the magnetic field configuration for spheromaks and RFPs, it is generally agreed that the theory is valid. However, the question of exactly why and how the relaxation occurs is still largely an open one. That is, why should certain quantities like the magnetic energy be reduced more rapidly than other quantities, such as the helicity?

One possible explanation is contained in a paper by Ting, et al. [37] They postulate that the dissipation of the magnetic energy occurs primarily at higher wave number than the dissipation of helicity, resulting in more rapid dissipation of the magnetic energy. The numerical evidence they have to support this consists of very high resolution simulations using spectral codes. They are essentially doing direct simulation of plasma turbulence, which limits the simulations to two dimensions.

Riyopoulos, et al. [26] used an approach similar to ours. They model a spheromak, and initialize the domain with a superposition of the Taylor state and a smaller amplitude, higher wave number force-free eigenmode. The resulting state is not in equilibrium, but since the Taylor state eigenmode contribution is dominant, it is not far from equilibrium or far from the Taylor state. Their calculations were done with a finite difference scheme on fine grids, but the calculations were axisymmetric. Most of their calculations were done 44 cells in the R direction and 30 cells in the Z direction. They observed that the plasma did approach the Taylor state in a non-smooth manner, with the bursts of relaxation activity coinciding with "filamentation" and reconnection events.

Horiuchi and Sato [16] used a similar approach, except that they used a three-dimensional finite difference code on a rectangular domain with periodic boundary conditions in the third coordinate direction to simulate a "straight torus". They initialized the plasma as a force-free state that was not the Taylor state, and then gave random perturbations to the magnetic field. They did their calculations on a $45 \times 45 \times 50$ grid with a Lundquist number of 10^4 . Based on the calculations we have done, these calculations were probably under-resolved. However, they did achieve

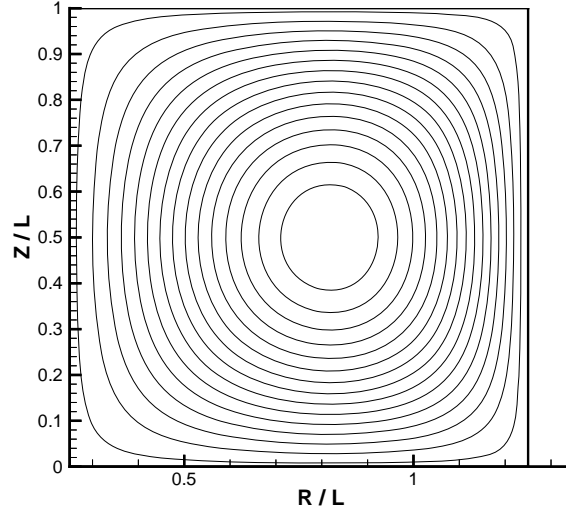


Figure 6.3: *Contours of poloidal magnetic flux for Taylor state of torus with square cross-section and $R_0/a = 1.5$.*

very interesting results in that depending on ratio of the length of the rectangular domain to the width, the plasma would evolve to a helical flux tube with a "toroidal" mode number that corresponded to the minimum energy helical state for that geometry.

6.1.1 Taylor State for 3D Toroid

For a plasma enclosed by a perfectly conducting toroidal shell, the resulting equilibrium satisfies

$$\nabla \times \mathbf{B} = \lambda \mathbf{B}, \quad (6.5)$$

where λ is a constant. That is, the Taylor state is force-free ($\mathbf{j} \parallel \mathbf{B}$) with $\lambda = j/B$ a constant.

This equation is an eigenvalue problem. It can be rewritten in terms of a stream

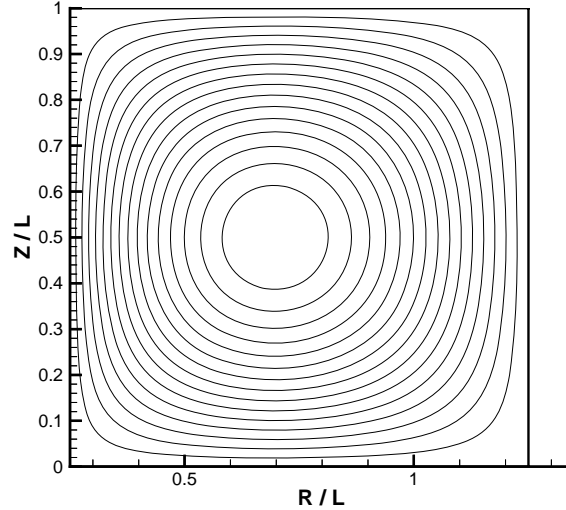


Figure 6.4: *Contours of toroidal field magnitude for Taylor state of torus with square cross-section and $R_0/a = 1.5$.*

function, Ψ , as

$$\left(\nabla^2 + \lambda^2\right) \Psi = 0. \quad (6.6)$$

The components of \mathbf{B} are related to Ψ by

$$B_\phi = \frac{\partial \Psi}{\partial R}, \quad (6.7)$$

$$B_R = -\frac{\partial^2 \Psi}{\partial Z \partial R}, \quad (6.8)$$

and

$$B_Z = \frac{1}{R} \frac{\partial}{\partial R} \left(R \frac{\partial \Psi}{\partial R} \right). \quad (6.9)$$

The boundary conditions we use are that $\mathbf{B} = 0$ at the walls. Note that this boundary condition, while making the calculation of the Taylor state simple, is not consistent with a perfectly conducting boundary (i.e. $B_\phi = 0$ is not consistent with this). This will be touched on again when the simulation boundary conditions are discussed.

This equation can be solved using the technique of separation of variables. Here we are interested in axisymmetric solutions, since they have the lowest energy, so we assume $\Psi = \Psi(R, Z)$. The solution for the largest eigenvalue (the minimum energy state) is

$$\Psi = C \left[J_0 \left(\lambda_R \frac{R}{R_{min}} \right) - \frac{J_1(\lambda_R)}{Y_1(\lambda_R)} Y_0 \left(\lambda_R \frac{R}{R_{min}} \right) \right] \sin \left(\pi \frac{Z}{L} \right), \quad (6.10)$$

where λ_R satisfies

$$J_1(\lambda_R R_{max}) Y_1(\lambda_R R_{min}) - J_1(\lambda_R R_{min}) Y_1(\lambda_R R_{max}) = 0, \quad (6.11)$$

and $\lambda^2 = \lambda_R^2 + \pi^2$. Figure 6.3 is a contour plot of the poloidal magnetic flux for the Taylor state with an aspect ratio (R_0/a) of 1.5. Figure 6.4 shows contours of B_ϕ . Note that in this case the minor radius (a) is half the length of the side of the square cross-section (L).

For the Taylor state, it can be shown that

$$\lambda = \frac{W_B}{K}. \quad (6.12)$$

We verified this numerically, although it could be shown analytically as well. This quantity is easy to calculate, and is a convenient way to measure the deviation of a plasma from the Taylor state.

6.1.2 Calculating Helicity

As stated previously, the definition of helicity is given by

$$K = \int \mathbf{A} \cdot \mathbf{B} d\tau. \quad (6.13)$$

There are some subtleties involved with the actual calculation of this quantity. The vector potential, \mathbf{A} , is defined as

$$\nabla \times \mathbf{A} = \mathbf{B}. \quad (6.14)$$

The vector potential is not gauge invariant since some function, ∇u , can be added to \mathbf{A} without changing \mathbf{B} . Thus, the helicity as defined above is not generally gauge invariant. However, for the case where $\mathbf{B} \cdot \mathbf{n} = 0$, which is the case here, the helicity is gauge invariant.

To actually calculate K , it is more convenient to rewrite the helicity in terms of flux as

$$K = \oint^{\Phi_t} \psi' d\Phi, \quad (6.15)$$

where ψ' is the amount of poloidal flux linking the incremental toroidal closed flux tube $d\Phi$, and Φ_t indicates that the integral is performed on all flux tubes within the closed volume. If we let $\bar{\psi}(R, \phi, Z)$ denote the poloidal flux per unit toroidal angle and note that $d\Phi = B_\phi dRdZ$, then the formula for the helicity becomes

$$K = \int \bar{\psi} B_\phi dRdZd\phi. \quad (6.16)$$

6.2 Approach

Since the plasma is initially far from a force-free state, the idea is to study how the plasma relaxes back toward the force-free state. Some of the questions to be answered are:

- Can relaxation be modeled with a resistive MHD code?
- When is a calculation fully resolved?
- How does one know when relaxation is occurring?
- Is 3D modeling required?
- What differences in behavior are expected for low aspect ratio toroids?

We attempt to answer these questions by examining the data in two ways. The first way is to track the evolution of global quantities such as the magnetic energy and helicity. The second way is to look at the structure of the solution at various times during the evolution. In particular, it is instructive to plot the λ profile at various slices to see how it approaches the constant λ profile of the Taylor state.

The parameters we decided to vary are the Lundquist number, the aspect ratio of the torus, and whether or not the calculation was axisymmetric or not. Varying the Lundquist number helps address the first three questions. We know that Taylor relaxation relies on a separation of time scales between resistive diffusion and resistive tearing. This separation increases with increasing Lundquist number (S), as previously discussed. However, as S is increased, the calculation becomes more costly since the grid must be refined in order to maintain the same resolution. Comparing axisymmetric and three-dimensional calculations addresses the second to last point. Comparing different aspect ratios addresses the last question. The run matrix consists of two Lundquist numbers ($S = 10^3$ and $S = 10^4$), two aspect ratios ($R_o/a = 1.5$ and $R_o/a = 3.0$), and each computed with and without invoking axisymmetry. This gives a total of eight different test cases.

Another important parameter is $\beta = 2p/B^2$. The original idea was to run with as low a value of β as possible, because the Taylor theory is actually a zero β theory. However, we found that it was difficult to run the three-dimensional simulations with β less than about .05 without getting a negative pressure during the plasma oscillations that occur as a result of the initial force imbalance in the corners. Thus, the basic test matrix described above was run with $\beta \geq .05$. However, we did run some axisymmetric cases at lower β to allow comparison with the higher β runs.

All of the runs were done at a modified Reynolds number of 10^5 . This amount of viscosity was chosen to provide some numerical stabilization while also ensuring that viscous effects are secondary to resistive effects.

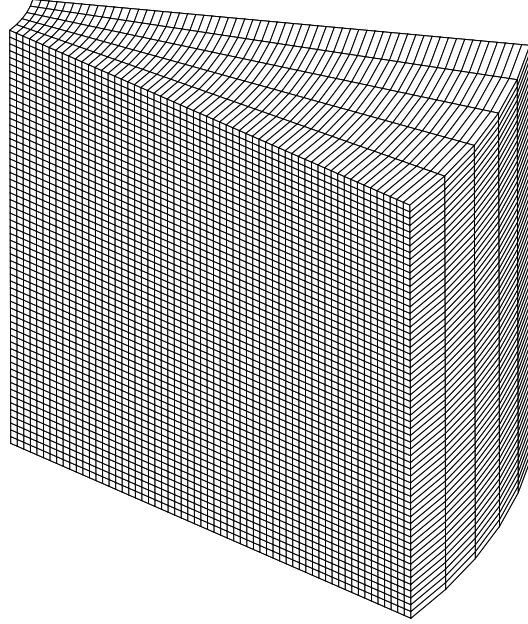


Figure 6.5: *Single block of the 16-block grid used for the $S = 10^3$ runs with $R_0/a = 1.5$.*

6.3 Numerical Details

The three-dimensional runs required very large grids and thus were done on a parallel computer. For the $S = 10^3$ cases, the torus was divided into 16 toroidal blocks, and each block was sent to a separate processor. Each block was solved with a $60 \times 60 \times 4$ grid ($64 \times 64 \times 8$ when including the ghost cells). Figure 6.5 shows the grid for one of the toroidal blocks. The cells were uniformly spaced in each direction.

The grid spacing for the $S = 10^3$ runs was determined by performing a grid resolution study on the axisymmetric problem with poloidal grids of 30×30 , 60×60 , and 120×120 . By comparing the solutions, we determined that the 60×60 solution was fully resolved. Actually, this was very conservative, as the 30×30 solution was very nearly converged at this S . Less resolution was required in the toroidal direction since the toroidal gradients were much much less than the poloidal ones. We determined

that 64 cells in the toroidal direction were sufficient.

For the $S = 10^4$ runs, the domain was divided into 32 blocks and sent to 32 processors. Each block was solved with a $100x100x2$ grid. In this case, the grid size was determined by computer and runtime constraints. That is, a formal grid resolution study was not done at the higher Lundquist number. However, a simple scaling argument shows that if the Lundquist number is increased by a factor of 10, then the truncation error of the hyperbolic terms should be decreased by a factor of 10 to ensure that the numerical diffusion from the hyperbolic truncation errors remains safely below the physical diffusion in the parabolic terms. Since the method is second order accurate in space, this argument suggests that if the $30x30$ grid is very nearly converged for $S = 10^3$, then going from $30x30$ to $100x100$ in the poloidal plane should allow one to increase S by a factor of $(100/30)^2 = 11.1$ and get comparable accuracy.

Dividing the domain toroidally as done here is not the most efficient from a parallel processing point of view. To minimize the amount of information passed between the processors, it would be better to have the i , j , and k dimensions of the blocks more nearly equal. However, dividing it toroidally had the advantage of avoiding the need to pass data between diagonally adjacent blocks, which was not implemented in the code at the time of these runs.

6.4 *Boundary Conditions*

The boundary conditions for these simulations were

$$\mathbf{v} = 0, \tag{6.17}$$

$$\mathbf{B} = 0, \tag{6.18}$$

$$\frac{\partial p}{\partial n} = 0, \tag{6.19}$$

and

$$\frac{\partial \rho}{\partial n} = 0. \quad (6.20)$$

These were implemented using two layers of ghost cells, and thus were second order accurate boundary conditions. The velocity boundary condition is just the no-slip condition. The conditions on the magnetic field require some explanation.

The most physical boundary conditions for the magnetic field would be to have perfectly conducting walls, since they would form a helicity barrier, which is necessary for the Taylor theory to hold. The boundary conditions for a perfect conductor are

$$\mathbf{B} \cdot \mathbf{n} = 0, \quad (6.21)$$

and

$$\mathbf{n} \times \mathbf{j} = 0. \quad (6.22)$$

The condition that $B_\phi = 0$ clearly does not satisfy the parallel current condition for a perfectly conducting wall. However, $B_\phi = 0$ is consistent with a Taylor state throughout the domain, whereas the perfectly conducting wall is not [26]. The simulations described in this chapter were not long enough to get appreciable toroidal field from the inner core out to the boundaries, so for these simulations no appreciable helicity was lost or gained through the boundaries. However, for longer simulations it would be necessary to implement the more physically correct conducting boundary conditions.

6.5 $S = 10^3$ Simulations

In this section we mostly consider the low aspect ratio torus ($R_o/a = 1.5$). Figure 6.6 is a plot of the initial poloidal flux contours. Note that initially there is no poloidal field in the corners (where $B = 0$) or in the central core of purely toroidal field. Figures 6.7 through 6.12 show contours of poloidal flux at intervals of 2 Alfvén times. After two Alfvén times, the flux has already filled in the corners of the square, since

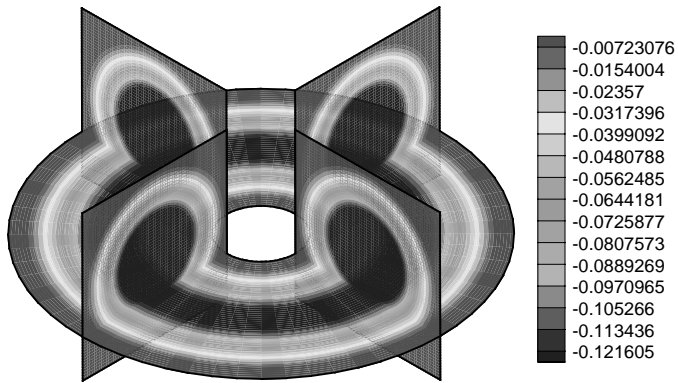


Figure 6.6: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 0$.*

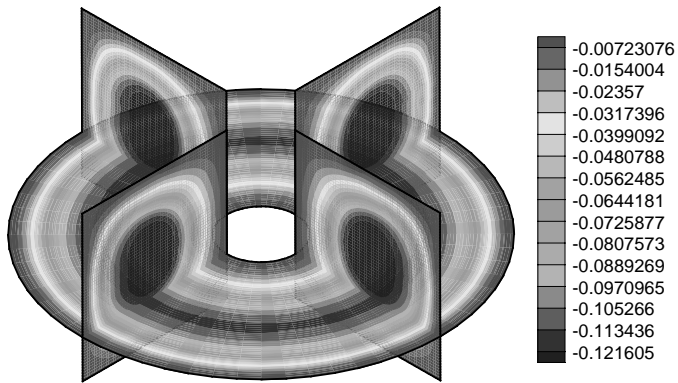


Figure 6.7: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 2$.*

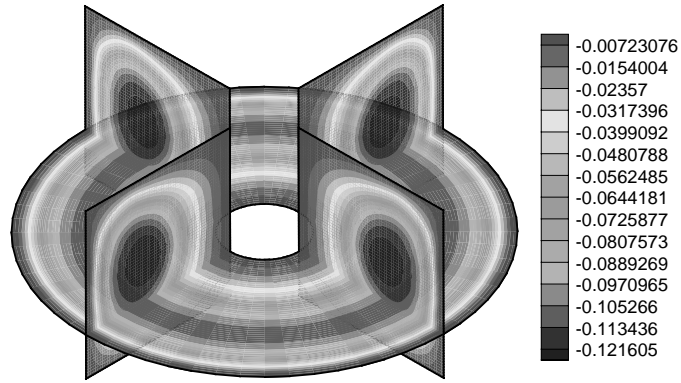


Figure 6.8: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 4$.*

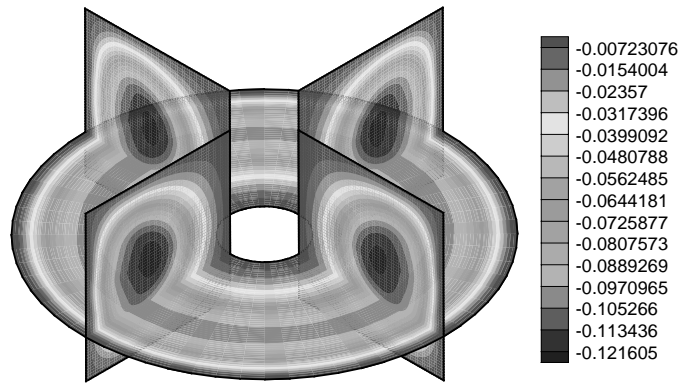


Figure 6.9: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 6$.*

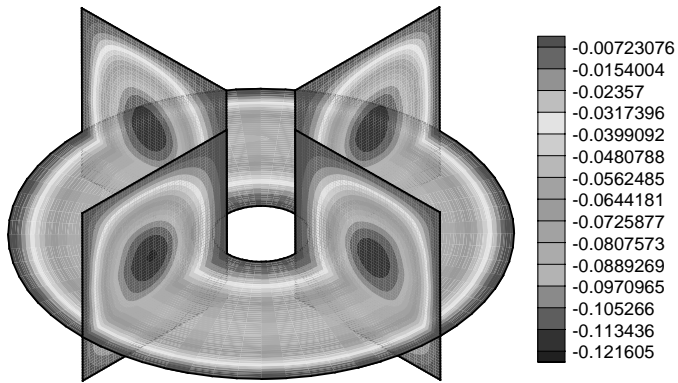


Figure 6.10: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 8$.*

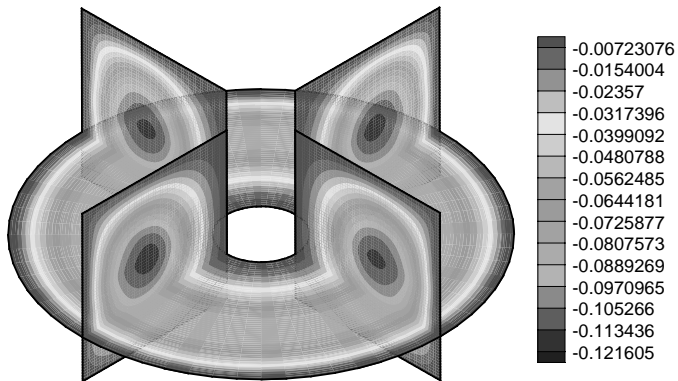


Figure 6.11: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 10$.*

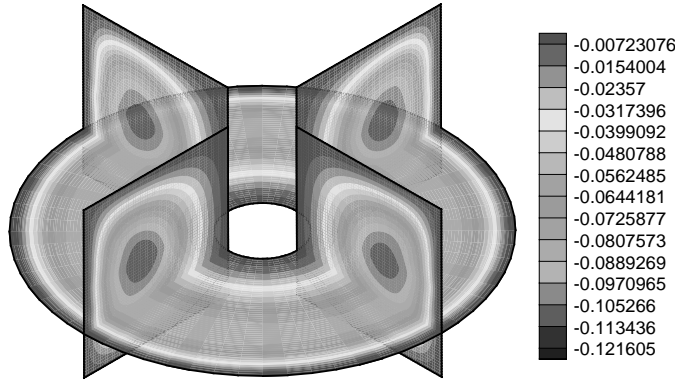


Figure 6.12: *Contours of poloidal flux for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.*

there is initially no magnetic pressure to hold the plasma in its original round shape. In fact, the plasma flows into the corners and then bounces back as an oscillation develops, and the motion in the corners leads to a radial oscillation of the toroidal plasma column. This type of motion can be seen most easily seen by examining "movies" of flux contours at shorter intervals than shown here. After 12 Alfvén times, the plasma is still very far from a Taylor state, although there has been some interpenetrating of the toroidal and poloidal fields. This is not surprising, since less than one resistive tearing time has elapsed ($0.19\tau_{tearing}$), and relaxation will certainly require several tearing times in order to evolve. Also, the evolution of the fields is very axisymmetric, even though this is a three-dimensional calculation.

Figures 6.13 through 6.15 show the velocity, density, and pressure at $t = 12$. These plots are of a single poloidal slice at the $\phi = 0$ plane (which is the half-plane corresponding to $z = 0$ and $x > 0$ in the Cartesian system). The velocity vectors show the plasma is still sloshing into the corners and back out at $t = 12$ as it tries to establish an equilibrium. This movement to the corners leads to a smaller side-

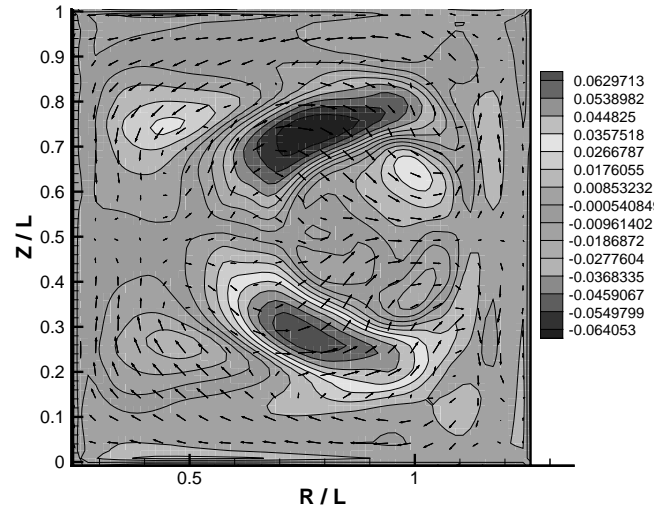


Figure 6.13: *Contours of toroidal velocity and poloidal velocity vectors in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.*

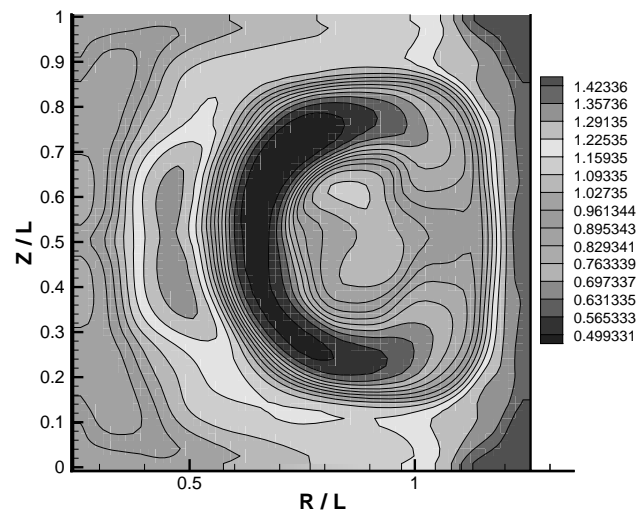


Figure 6.14: *Contours of density in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.*

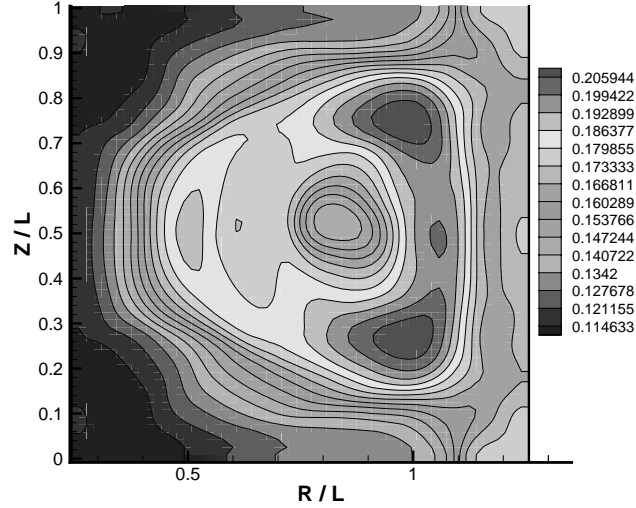


Figure 6.15: *Contours of pressure in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.*

to-side motion of the entire plasma column. The density is highest on the outside as the plasma shifts outward and compresses the plasma against the outer wall. The pressure plot shows that the pressure is being compressed on the outer portion of the initially pure toroidal field portion of the plasma. An $m = 3$ structure is developing as well, where m refers to the poloidal mode number. A mode developing over such a short time scale must be an ideal MHD mode.

Figures 6.16 and 6.17 are plots of poloidal flux at $t = 12$ for the three-dimensional and axisymmetric runs, respectively. They show the slight asymmetry that develops in the magnetic field over time. This amount of asymmetry is not enough to make the plasma relax at a different rate than the axisymmetric plasma, and is in marked contrast to the higher Lundquist number runs shown in the next section.

The three-dimensional case was only run out to $t = 12$ due to runtime considerations. The axisymmetric runs were not limited in this way. To look at some longer time behavior, the axisymmetric case was run out to 40 Alfvén times (corresponds

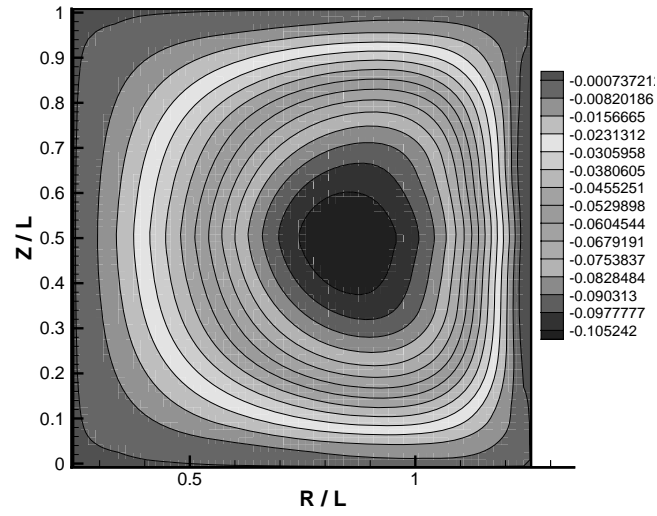


Figure 6.16: *Contours of poloidal flux in $\phi = 0$ plane for $S = 10^3$ and $R_o/a = 1.5$ at $t = 12$.*

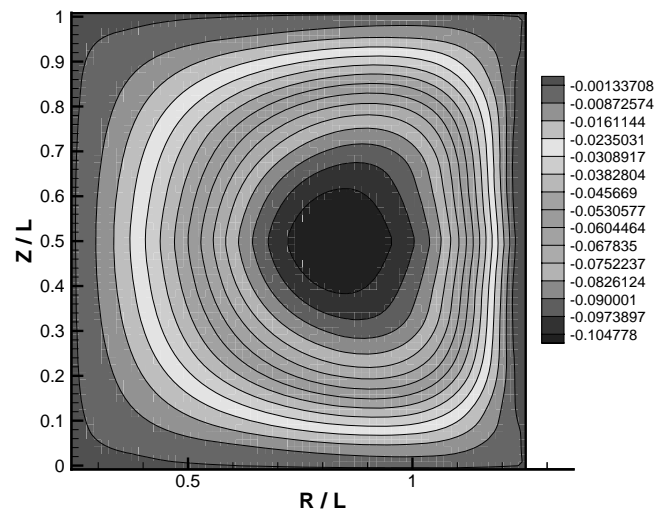


Figure 6.17: *Contours of poloidal flux at $t = 12$ for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.*

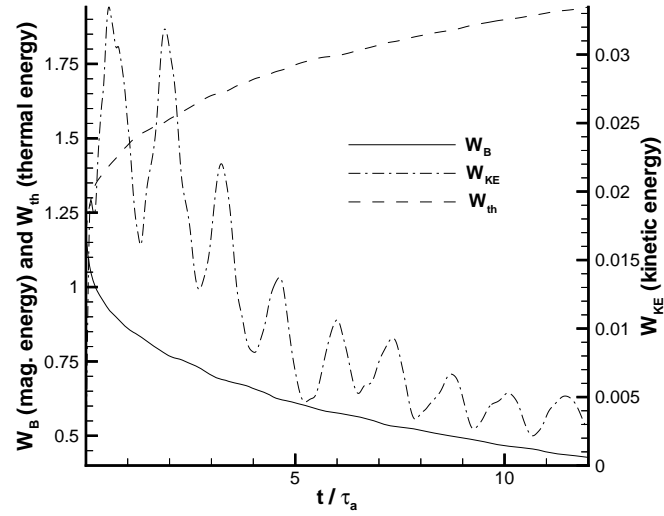


Figure 6.18: *Integrated magnetic, thermal, and kinetic energies as a function of time for 3D calculation with $S = 10^3$ and $R_o/a = 1.5$.*

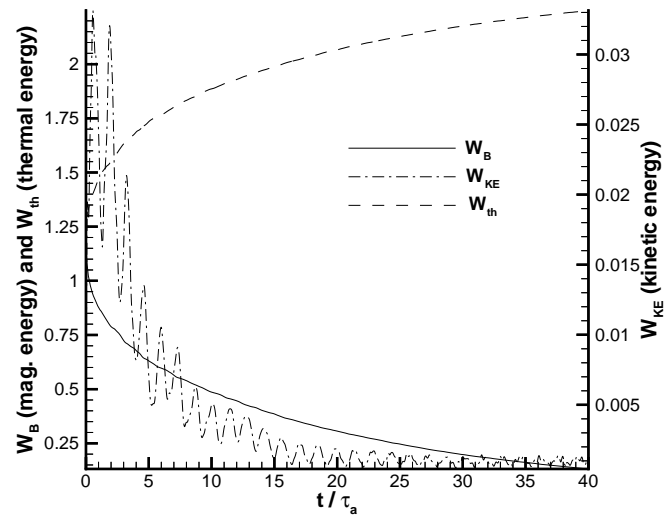


Figure 6.19: *Integrated magnetic, thermal, and kinetic energies as a function of time for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.*

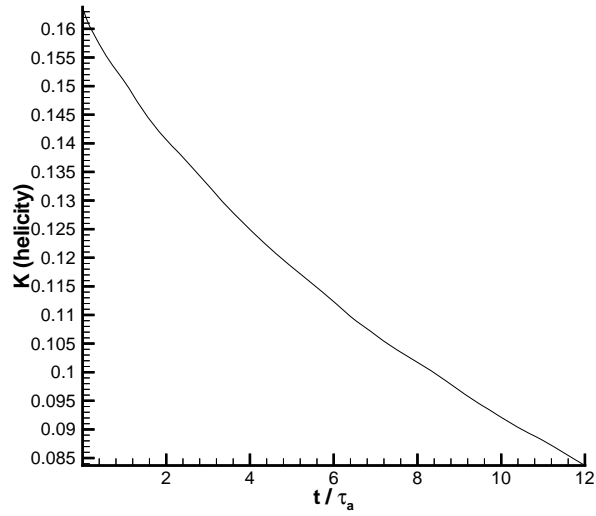


Figure 6.20: *Decay of magnetic helicity as a function of time for 3D calculation with $S = 10^3$ and $R_o/a = 1.5$.*

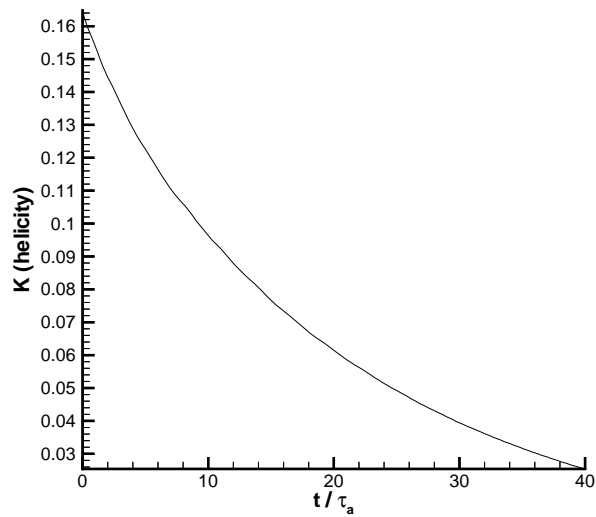


Figure 6.21: *Decay of magnetic helicity as a function of time for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.*

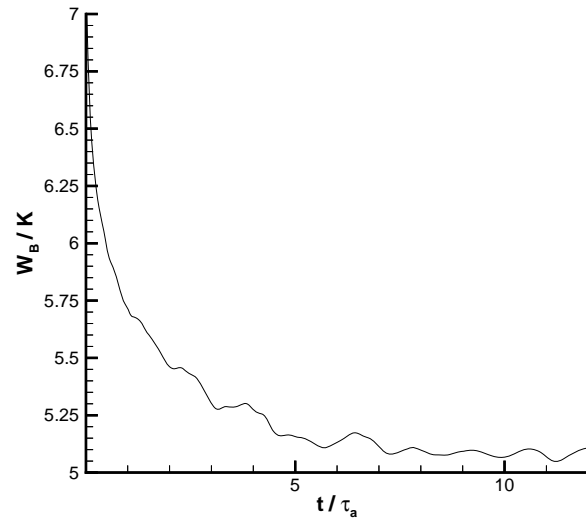


Figure 6.22: *Ratio of integrated magnetic energy to helicity as a function of time for 3D calculation with $S = 10^3$ and $R_o/a = 1.5$.*

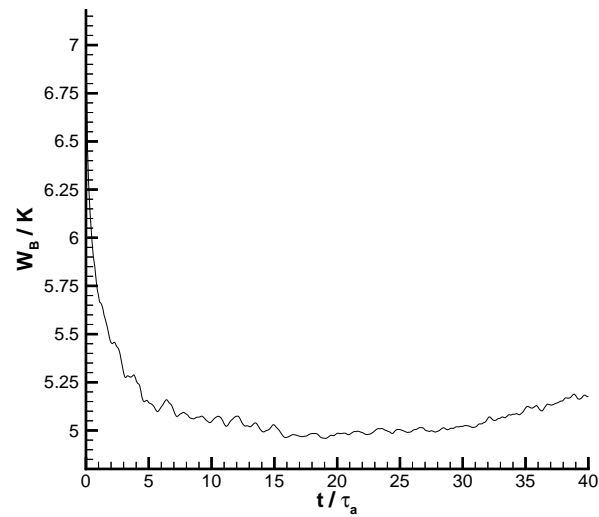


Figure 6.23: *Ratio of integrated magnetic energy to helicity as a function of time for axisymmetric calculation with $S = 10^3$ and $R_o/a = 1.5$.*

to $0.63\tau_{tearing}$). The next series of plots shows the time evolution of the helicity, the energy, and the ratio of the two. These global quantities provide a way of quantifying how near the plasma is to the Taylor state, and also for comparing different runs.

Figures 6.18 and 6.19 show the magnetic, thermal, and kinetic energy integrals as a function of time. The magnetic and internal energies are much larger than the kinetic energy. The kinetic energy is largest early in the run as the plasma moves toward an equilibrium. The side to side oscillations are apparent from the kinetic energy plot. Figure 6.19 shows that the plasma is close to an equilibrium state by 20 Alfvén times, since the kinetic energy is close to zero. Note that as the magnetic energy is reduced it is converted to thermal energy primarily through resistive heating (and also some viscous heating). Since there are no thermal transport equations in this version of the code, there was no way to dump the heat to the walls, as would happen in a real plasma. The result is that the overall β of the plasma increases over time.

Figures 6.20 and 6.21 show that the helicity decays over time due to resistive diffusion. At this Lundquist number the decay is very monotonic. If the two plots were shown on the same time scale they nearly overlay. The ratio of the magnetic energy to the magnetic helicity as a function of time is shown in Figures 6.22 and 6.23. Recall that for a Taylor state, this ratio is equal to the eigenvalue, λ . For this aspect ratio, the Taylor state has $W_B/K = 4.62$. After 12 Alfvén times, W_B/K has decreased to just over 5 for both cases. Over this time span, there is very little difference between the three-dimensional run and the axisymmetric run. The axisymmetric plot shows that at later times, the relaxation stalls at $W_B/K = 5$ and begins to increase. It is not surprising that the axisymmetric runs has difficulty in relaxing all the way to the Taylor state, since the axisymmetric assumption prevents helical distortions that enable much more rapid reconfigurations of the plasma, as will become evident when looking at the higher S runs. Another possible explanation for this long time behavior is that the boundary condition is allowing helicity to cross the boundary as

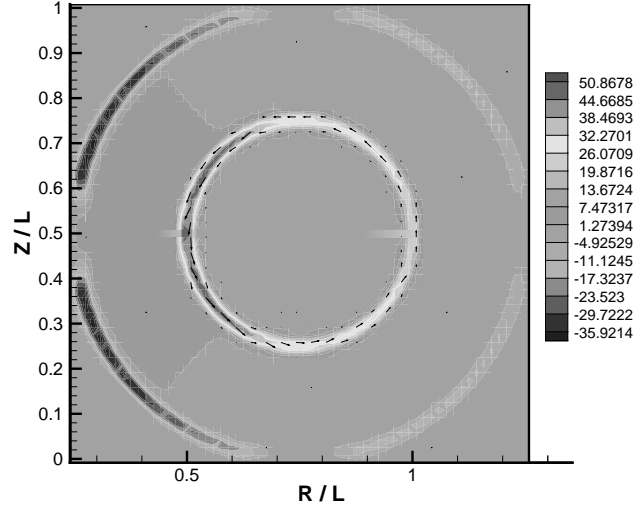


Figure 6.24: *Contours of toroidal current density overlaid with poloidal current vectors at $t = 0$ for $R_o/a = 1.5$ and $S = 10^3$.*

the toroidal field diffuses enough so that it is non-zero at the boundary.

Another way to look at the data is in terms of the current. Figure 6.24 shows contours of toroidal current density overlaid with poloidal current vectors at $t = 0$. The current sheets are clearly visible. Figure 6.24 shows the currents after 12 Alfvén times. The current sheets have spread considerably as the plasma evolves toward uniform $\lambda = j/B$. One feature of the Taylor state that is present is that the magnetic axis is offset from the point of maximum toroidal field, as was seen in the plots of the Taylor state in Figures 6.3 and 6.4. Figure 6.26 shows contours of λ . There are large regions of the plasma that are far above and far below the Taylor state λ of 4.62.

Perhaps an easier way to visualize the data is to examine the λ profile at the $Z/L = 0.5$ midplane as it evolves in time. Figures 6.27 through 6.29 show the λ profile at $t = 0$, $t = 12$, and $t = 40$, respectively. These are all for the axisymmetric run. The λ profile for the three-dimensional run at $t = 12$ is very similar to the

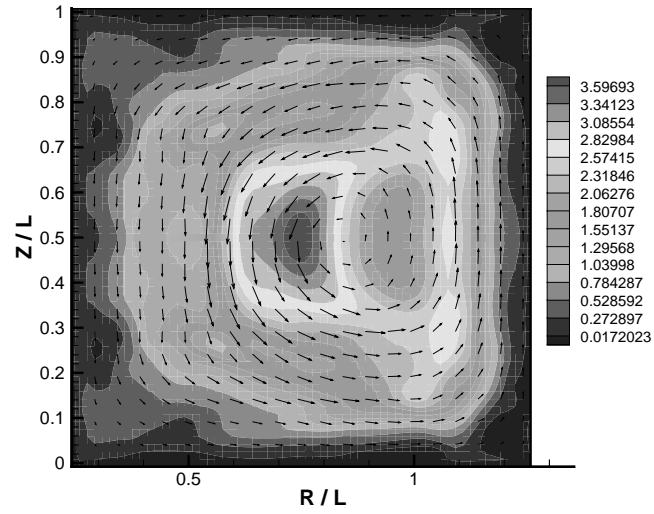


Figure 6.25: *Contours of toroidal current density overlayed with poloidal current vectors at $t = 12$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.*

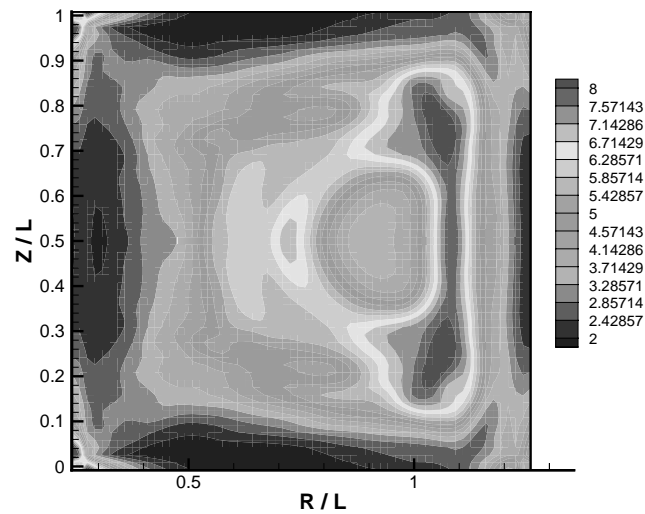


Figure 6.26: *Contours of $\lambda = j/B$ at $t = 12$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.*

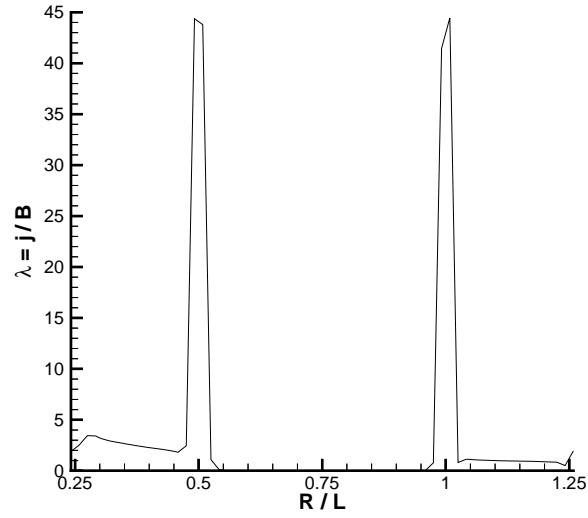


Figure 6.27: $\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 0$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.

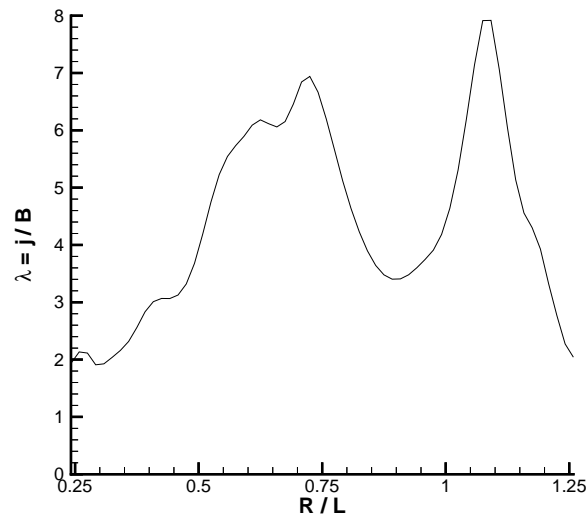


Figure 6.28: $\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 12$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.

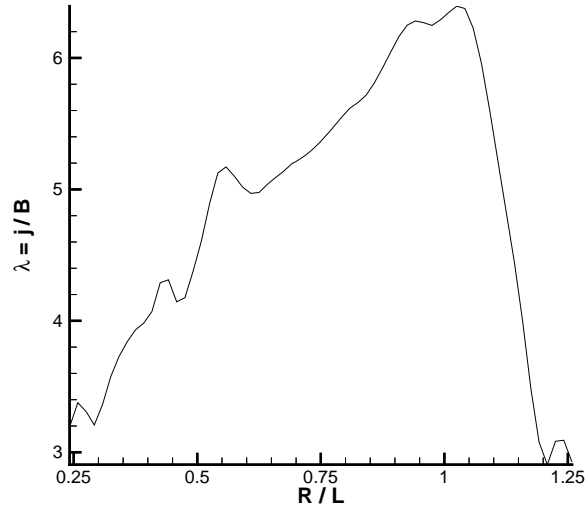


Figure 6.29: $\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 40$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^3$.

axisymmetric one, and thus is not shown. After 40 Alfvén times, the extreme humps in the profile that correspond to the initial current sheets have smoothed out, resulting in a fairly monotonic profile having too much current at large R and too small at small R .

Figure 6.30 compares the ratio of the magnetic energy to the helicity as a function of time for all the $S = 10^3$ cases, including the axisymmetric case at an aspect ratio of 3. (Note that the three-dimensional run at $R_o/a = 3.0$ was not done.) The most important feature of the data is that the axisymmetric and three-dimensional runs are virtually identical (for the aspect ratio of 1.5). In fact, the axisymmetric case is approaching the asymptotic value of $W_b/K = 4.62$ slightly more rapidly than the three-dimensional case, which is surprising and difficult to explain. The large aspect ratio case falls more rapidly than the smaller aspect ratio case because it is evolving toward a lower value of λ of 4.48.

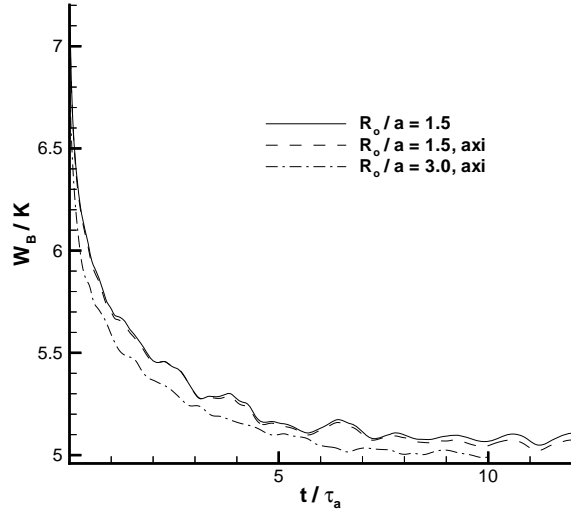


Figure 6.30: *Ratio of integrated magnetic energy to helicity for all of the $S = 10^3$ runs.*

6.6 $S = 10^4$ Simulations

These runs were done at a Lundquist number of 10^4 , which is still several orders of magnitude lower than most fusion experiments, but high enough that some high S effects begin to be seen. As discussed previously, the three-dimensional runs required large grids and significant amounts of CPU time. In particular, each three-dimensional simulation required approximately 24 hours on 32 processors of an IBM SP2 to run out to 10 Alfvén times. Since the Lundquist number is larger for these runs, 10 Alfvén times corresponds to an even smaller fraction of a tearing time (0.04) than the $S = 10^3$ runs. Ideally, of course, one would like to have longer simulations, but that was not possible in this case. The axisymmetric runs were also run to 10 Alfvén times for ease of comparison with the three-dimensional runs.

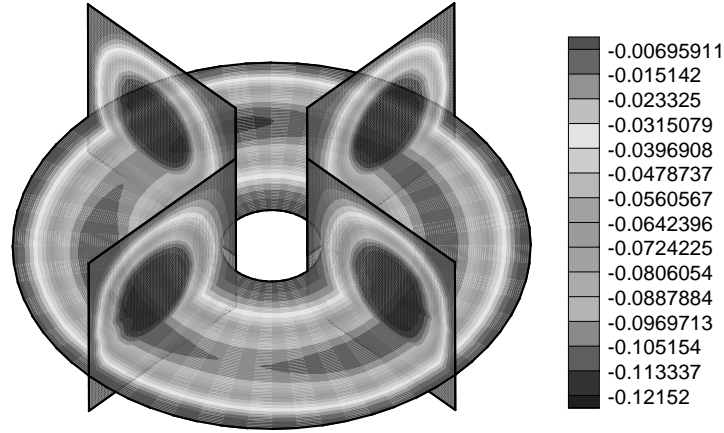


Figure 6.31: *Contours of poloidal flux for $S = 10^4$ and $R_o/a = 1.5$ at $t = 5$.*

6.6.1 Low Aspect Ratio Simulations

Poloidal flux contours at $t = 5$ and $t = 10$ are shown in Figures 6.31 and 6.32. Unlike the lower S runs, there is an obvious toroidal mode structure that develops after just a few Alfvén times and is well developed by $t = 10$. Since this instability evolves on an Alfvénic time scale, this must be an ideal MHD instability. Apparently these ideal modes were heavily damped at $S = 10^3$. The dominant toroidal mode appears to be $n = 3$.

Figures 6.33 and 6.34 show the large differences in the magnetic field configurations between the axisymmetric and three-dimensional cases. Figures 6.35 and 6.36 show similar differences in the pressure field. The pressure plots make it especially clear that the instability seems to occur at the $q = 1$ surface and mostly disturbs the plasma inside the $q = 1$ surface. This strongly suggests that the toroidal mode structure is due to an $n = 3$ kink mode.

The toroidal structure and motion is quite complicated and difficult to visualize. It appears to have some of the characteristics of a kink mode. Figures 6.37 through

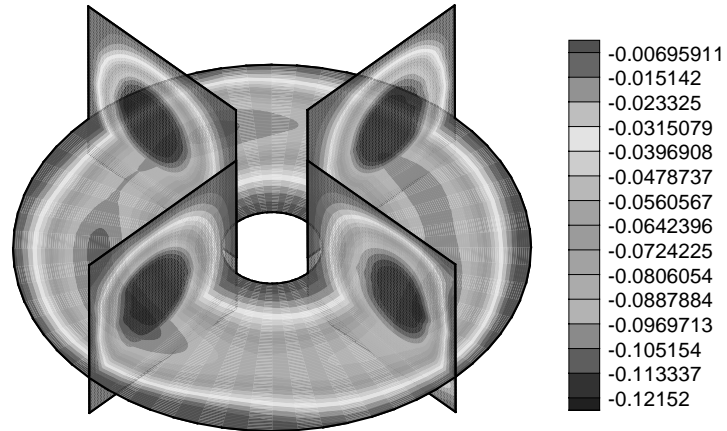


Figure 6.32: *Contours of poloidal flux for $S = 10^4$ and $R_o/a = 1.5$ at $t = 10$.*

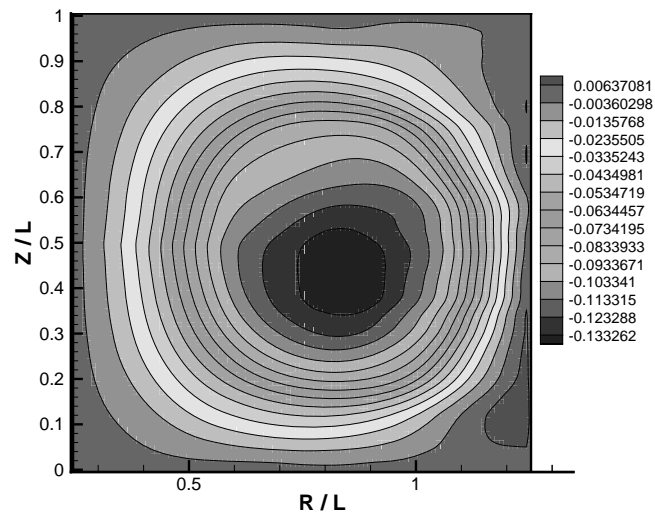


Figure 6.33: *Contours of poloidal flux in $\phi = 0$ plane for $S = 10^4$ and $R_o/a = 1.5$ at $t = 10$.*

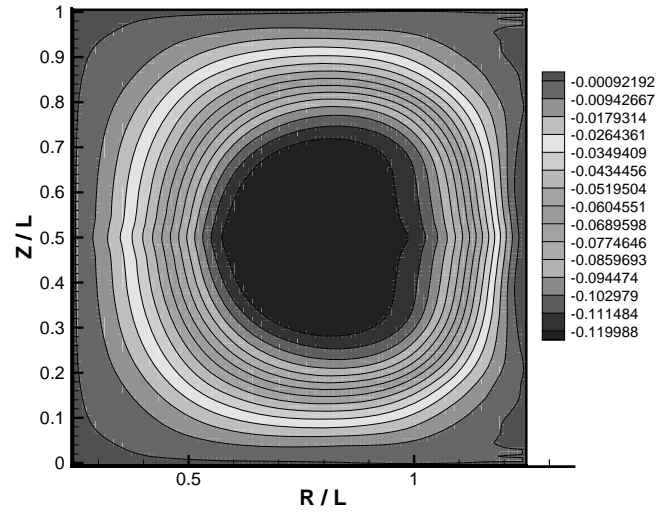


Figure 6.34: *Contours of poloidal flux at $t = 10$ for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.*

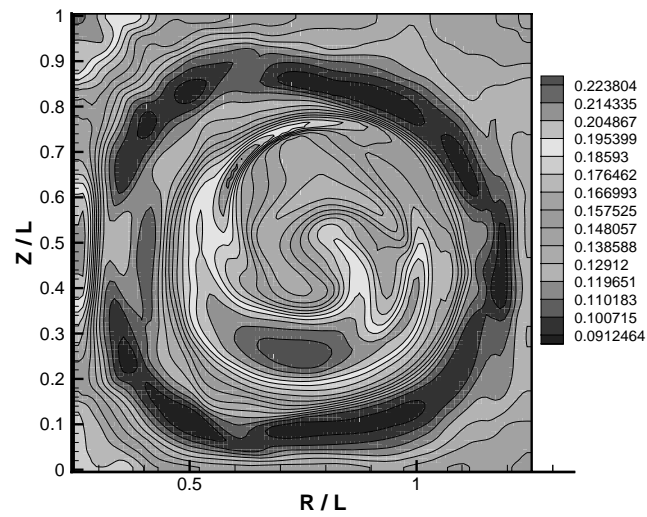


Figure 6.35: *Contours of pressure in $\phi = 0$ plane for $S = 10^4$ and $R_o/a = 1.5$ at $t = 10$.*

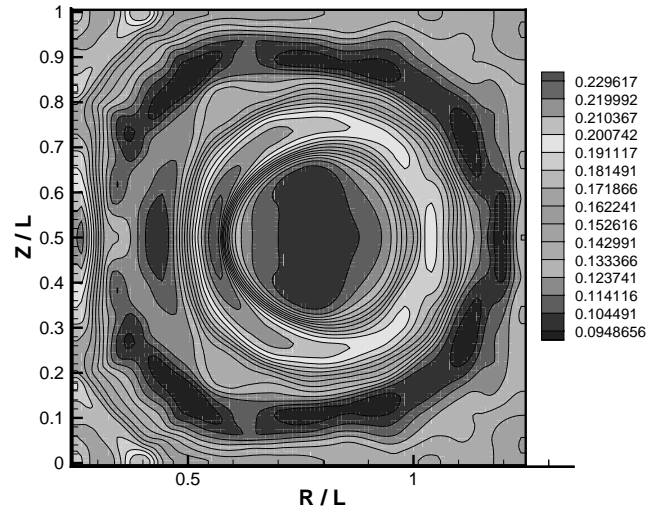


Figure 6.36: *Contours of pressure at $t = 10$ for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.*

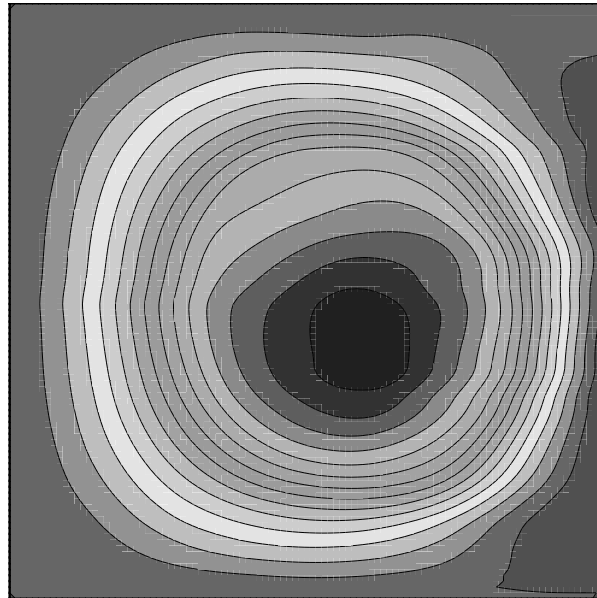


Figure 6.37: *Contours of poloidal flux in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.*

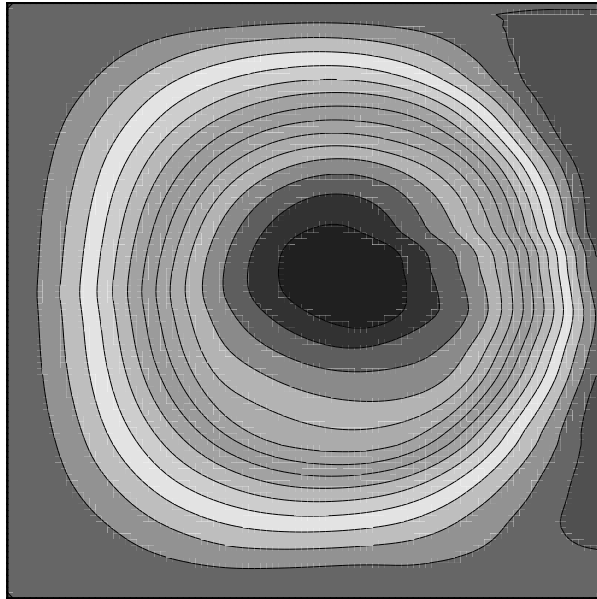


Figure 6.38: *Contours of poloidal flux in $\phi = 90$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.*

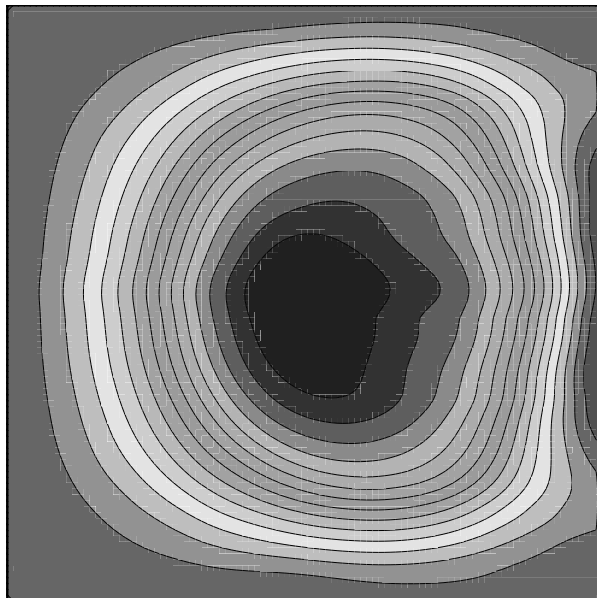


Figure 6.39: *Contours of poloidal flux in $\phi = 180$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.*

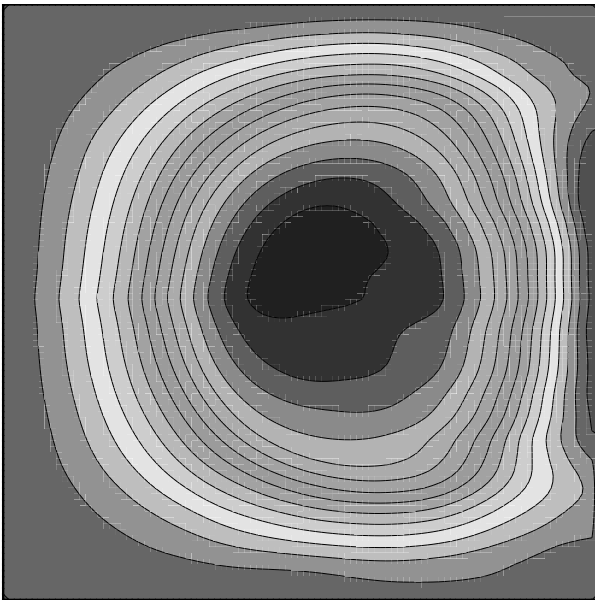


Figure 6.40: *Contours of poloidal flux in $\phi = 270$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 1.5$.*

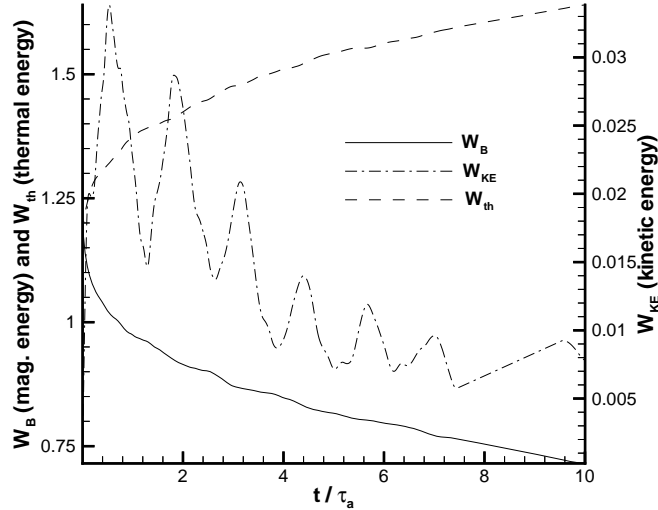


Figure 6.41: *Integrated magnetic, thermal, and kinetic energies as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 1.5$.*

6.40 show the poloidal flux at the $\phi = 0$, $\phi = 90$, $\phi = 180$, and $\phi = 270$ degree planes.

Plots of the evolution of energy, helicity, and the ratio of magnetic energy to helicity reveal a finer structure than was seen in the $S = 10^3$ runs. That is, the magnetic energy, the helicity, and their ratio all decay in a less smooth fashion than for the $S = 10^3$ runs. Some of this may be due to the smaller time steps that were required with the finer grid. However, it is more likely due to the ideal MHD activity that is driving some relaxation in this case. Note that the plots for the three-dimensional runs are missing data from about $t = 7.5$ to $t = 9.5$ due to a file transfer problem. This accounts for the abrupt change in slope seen in the data at those times.

The plots of W_B/K indicate that the ratio has decayed to about 5.5 after 10 Alfvén times. This is not as low as the $S = 10^3$ case got after the same number of Alfvén times, but that is not surprising. Since the Lundquist number is ten times as

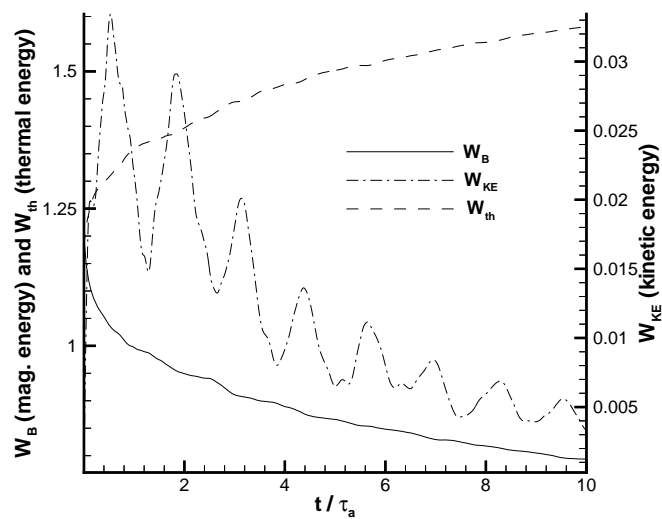


Figure 6.42: *Integrated magnetic, thermal, and kinetic energies as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.*

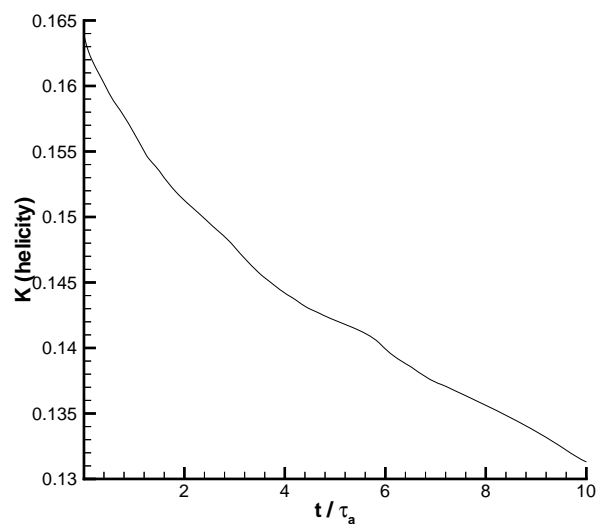


Figure 6.43: *Decay of magnetic helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 1.5$.*

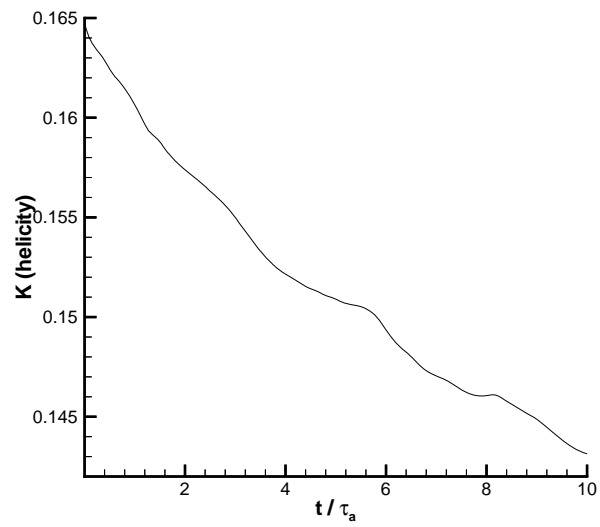


Figure 6.44: *Decay of magnetic helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.*

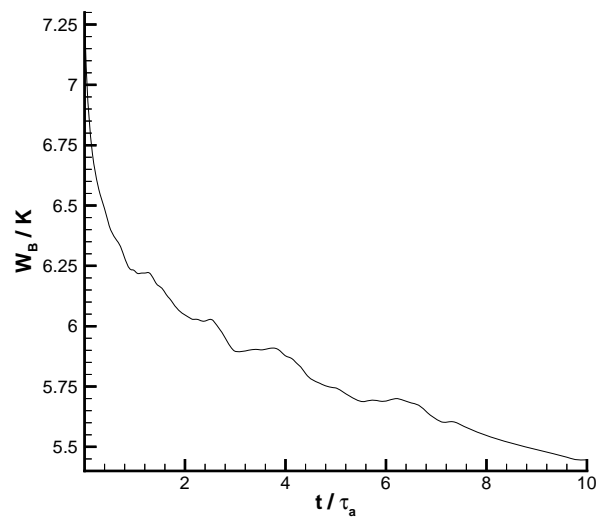


Figure 6.45: *Ratio of integrated magnetic energy to helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 1.5$.*

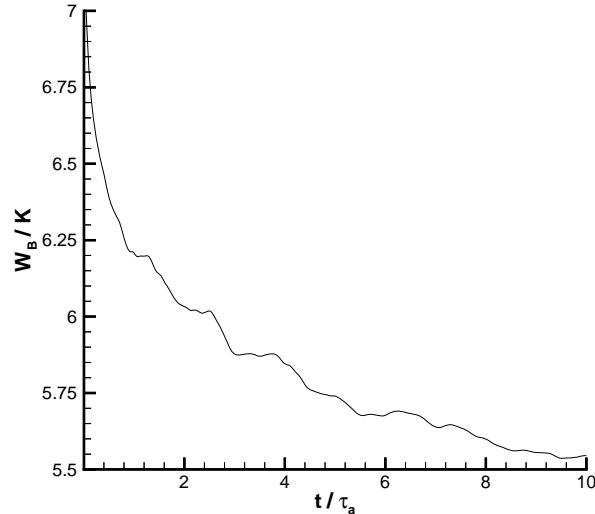


Figure 6.46: *Ratio of integrated magnetic energy to helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 1.5$.*

large, this solution has only evolved one tenth as far in terms of resistive diffusion time, and thus should not be expected to have evolved as far.

Plots of the λ profiles are quite revealing. Figures 6.47 and 6.48 show λ versus radius at the midplane for the three-dimensional and axisymmetric case, respectively. The striking feature is that the kink instability has resulted in significantly enhanced diffusion of the poloidal and toroidal fields in the core region of the three-dimensional simulation. This enhanced diffusion is presumably due to the large gradients caused by the thrashing of the toroidal flux tube. This can be seen by comparing the values of λ at $R/L = 0.8$.

6.6.2 High Aspect Ratio Simulations

Poloidal flux contours at $t = 5$ and $t = 10$ are shown in Figures 6.49 and 6.50 for $R_o/a = 3.0$. As in the lower aspect ratio case, the ideal MHD kink instability clearly

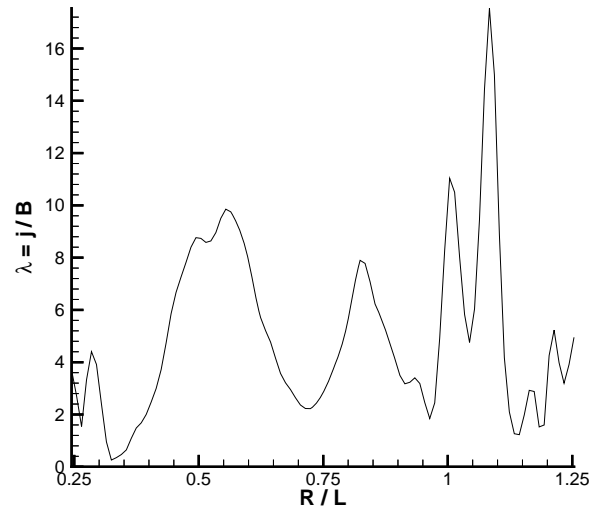


Figure 6.47: $\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for 3D calculation with $R_o/a = 1.5$ and $S = 10^4$.

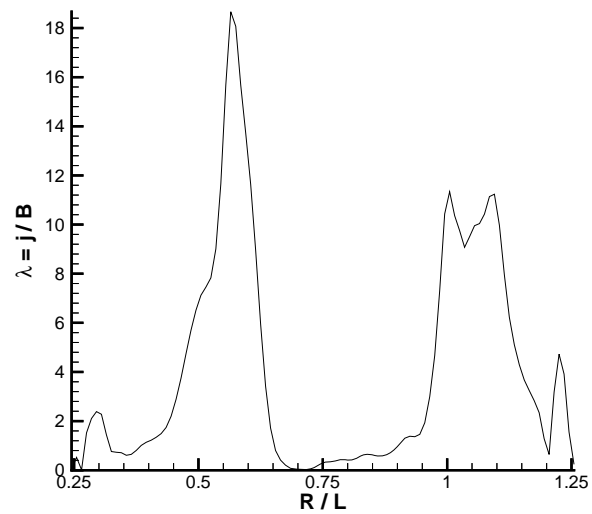


Figure 6.48: $\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for axisymmetric calculation with $R_o/a = 1.5$ and $S = 10^4$.

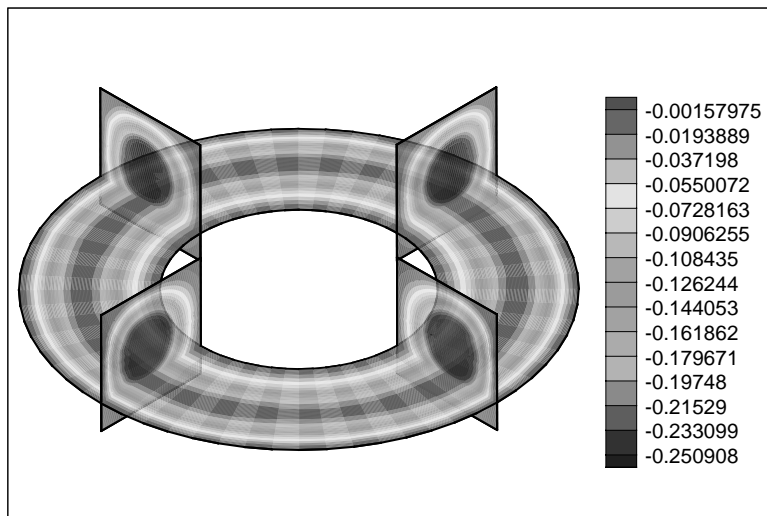


Figure 6.49: *Contours of poloidal flux for $S = 10^4$ and $R_o/a = 3.0$ at $t = 5$.*

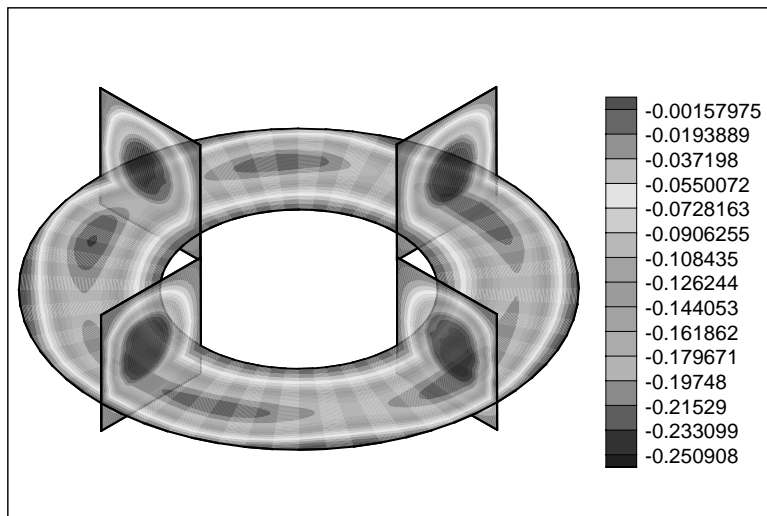


Figure 6.50: *Contours of poloidal flux for $S = 10^4$ and $R_o/a = 3.0$ at $t = 10$.*

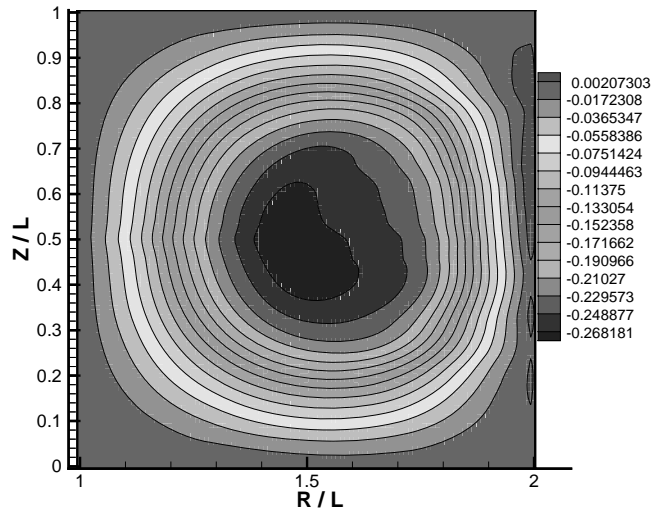


Figure 6.51: *Contours of poloidal flux in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.*

is affecting the magnetic field structure. In this case, the dominant toroidal mode is $n = 5$ or $n = 6$ (recall that the $R_o/a = 1.5$ case had an $n = 3$ structure).

In order to highlight the differences between axisymmetric and three dimensional results, plots at the $\phi = 0$ poloidal plane are compared to the axisymmetric results. Figures 6.51 and 6.52 compare the poloidal flux. Figures 6.53 and 6.54 compare the pressure. The results are similar to the low aspect ratio results.

As before, we look at poloidal slices at four toroidal angles spaced 90 degrees apart. Figures 6.55 through 6.58 show the poloidal flux at the $\phi = 0$, $\phi = 90$, $\phi = 180$, and $\phi = 270$ degree planes. While the outer flux surfaces remain mostly unperturbed, the inner flux tube clearly has the helical structure of a kink mode.

The global plots in Figures 6.59 through 6.64 are similar to ones for the $R_o/a = 1.5$ aspect ratio. They are included here to give a complete record of all the high resolution three-dimensional runs.

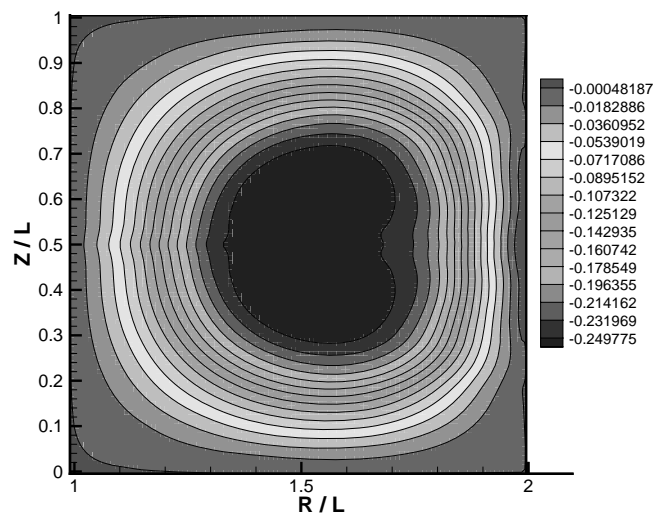


Figure 6.52: *Contours of poloidal flux at $t = 10$ for axisymmetric calculation at $S = 10^4$ and $R_o/a = 3.0$.*

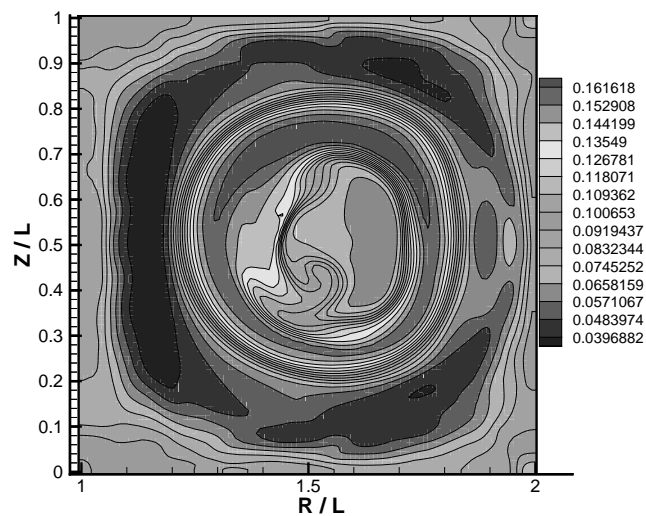


Figure 6.53: *Contours of pressure in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.*

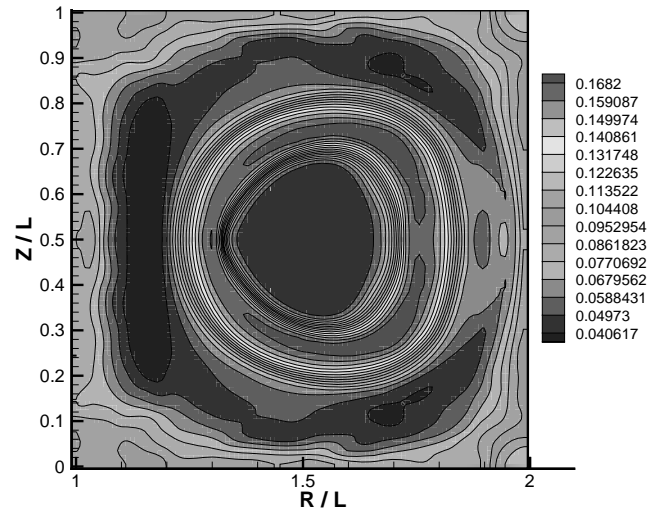


Figure 6.54: *Contours of pressure at $t = 10$ for axisymmetric calculation at $S = 10^4$ and $R_o/a = 3.0$.*

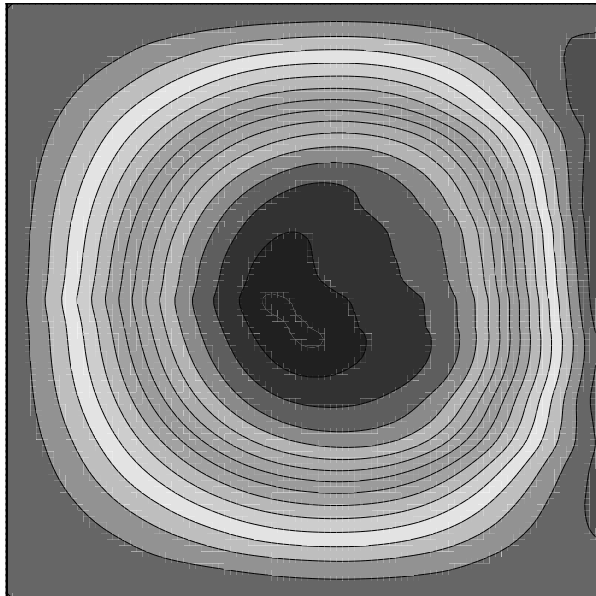


Figure 6.55: *Contours of poloidal flux in $\phi = 0$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.*

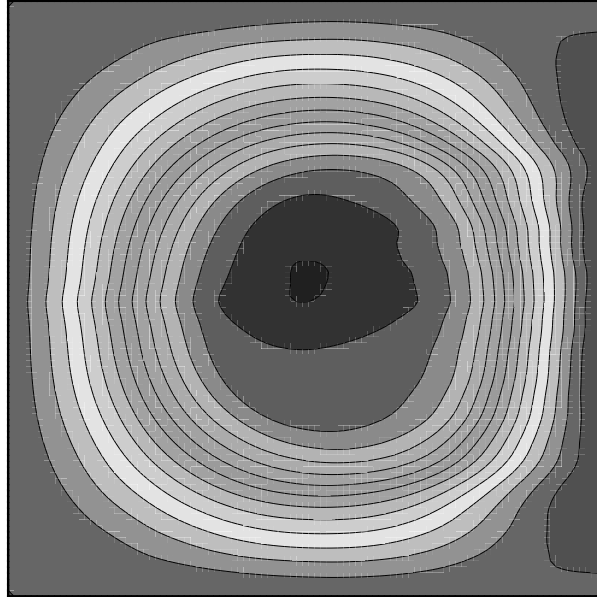


Figure 6.56: *Contours of poloidal flux in $\phi = 90$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.*

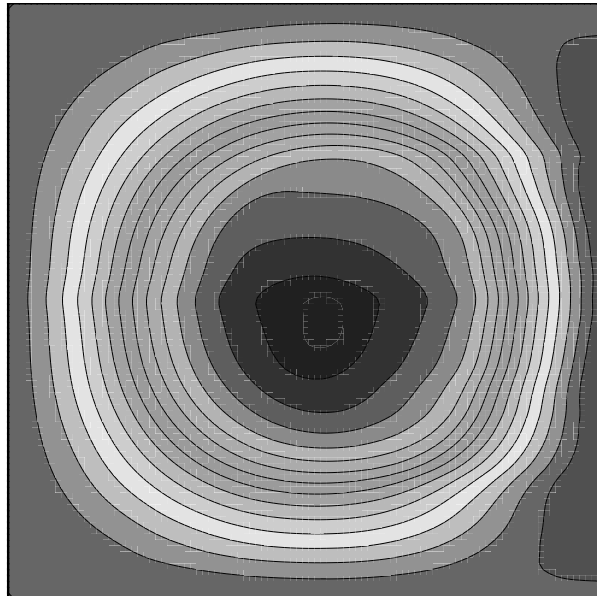


Figure 6.57: *Contours of poloidal flux in $\phi = 180$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.*

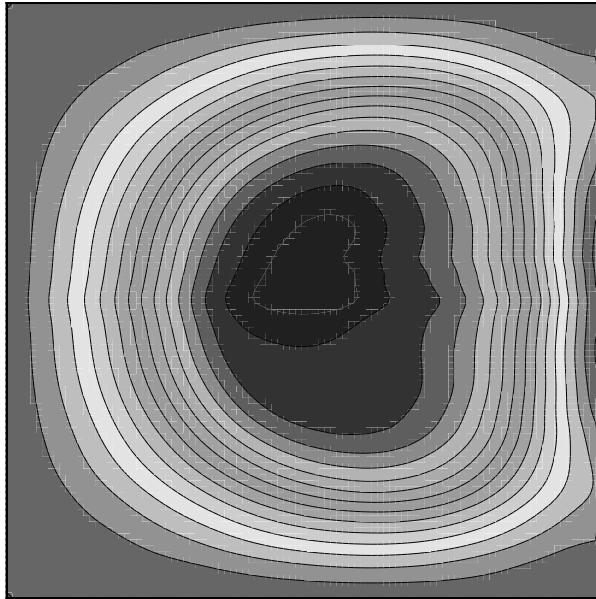


Figure 6.58: *Contours of poloidal flux in $\phi = 270$ plane at $t = 10$ for 3D calculation at $S = 10^4$ and $R_o/a = 3.0$.*

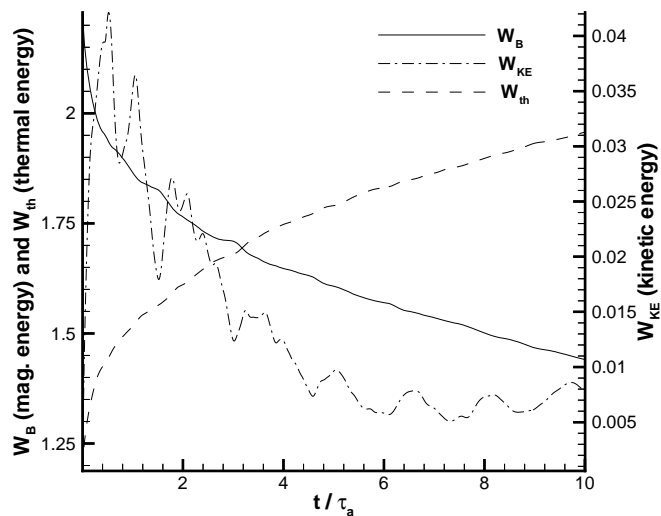


Figure 6.59: *Integrated magnetic, thermal, and kinetic energies as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 3.0$.*

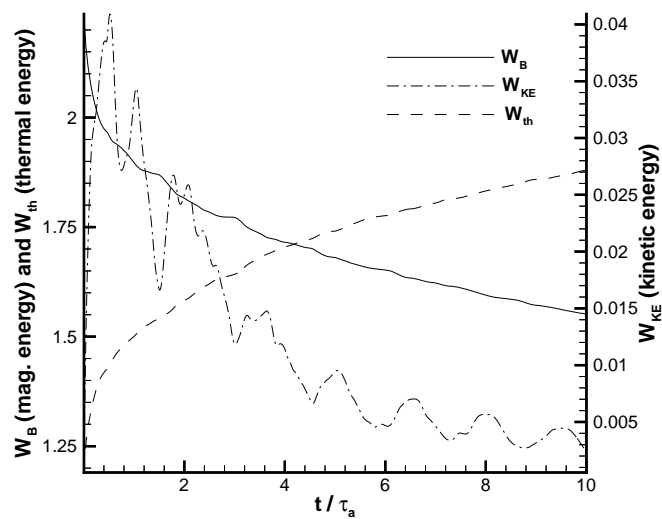


Figure 6.60: *Integrated magnetic, thermal, and kinetic energies as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 3.0$.*

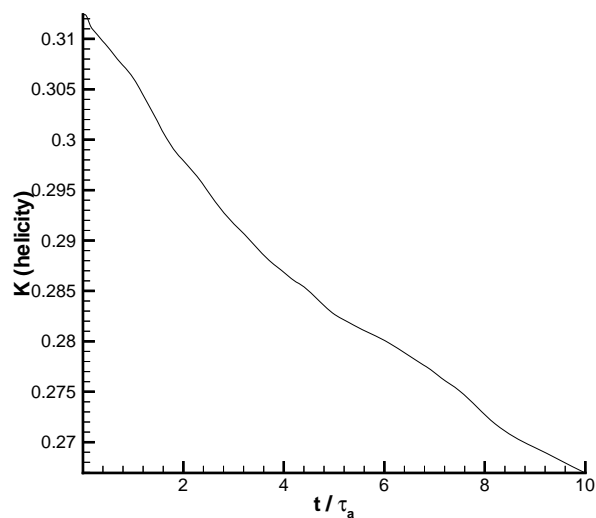


Figure 6.61: *Decay of magnetic helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 3.0$.*

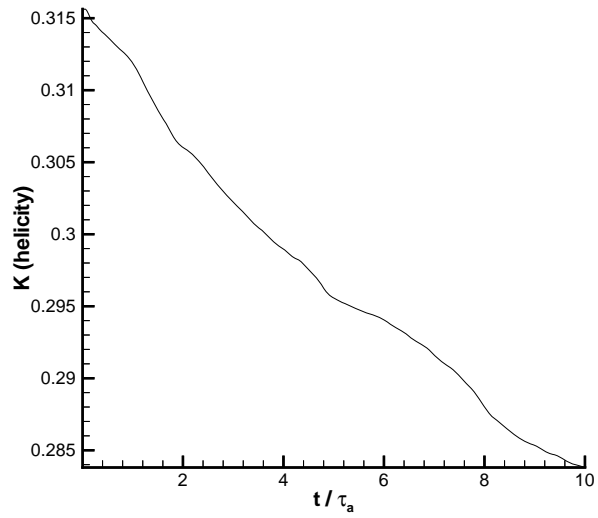


Figure 6.62: *Decay of magnetic helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 3.0$.*

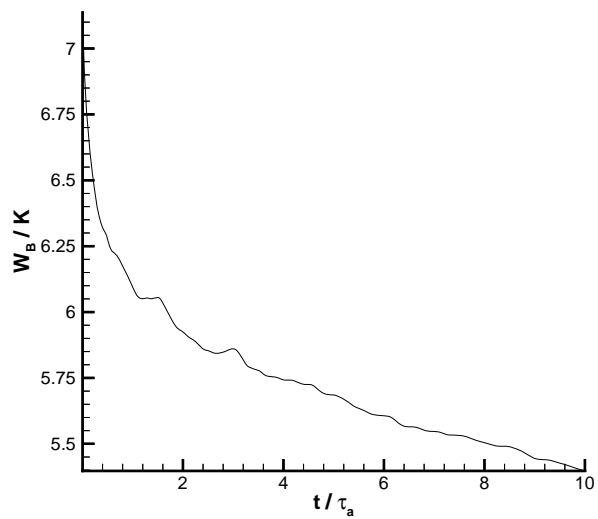


Figure 6.63: *Ratio of integrated magnetic energy to helicity as a function of time for 3D calculation with $S = 10^4$ and $R_o/a = 3.0$.*

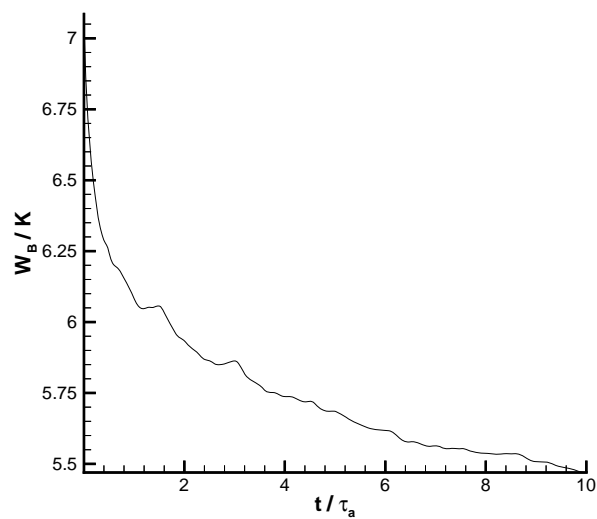


Figure 6.64: *Ratio of integrated magnetic energy to helicity as a function of time for axisymmetric calculation with $S = 10^4$ and $R_o/a = 3.0$.*

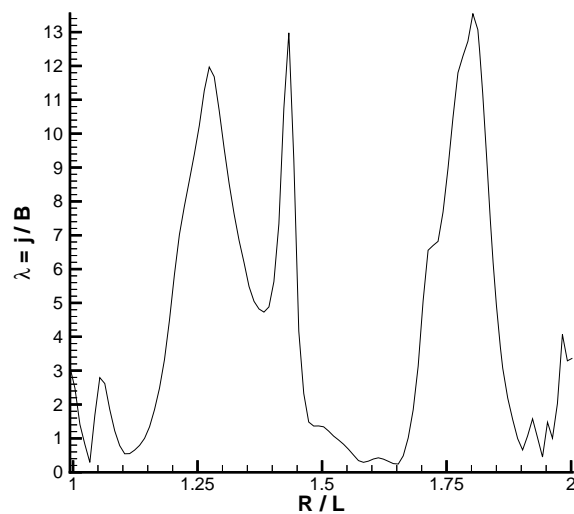


Figure 6.65: *$\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for 3D calculation with $R_o/a = 3.0$ and $S = 10^4$.*

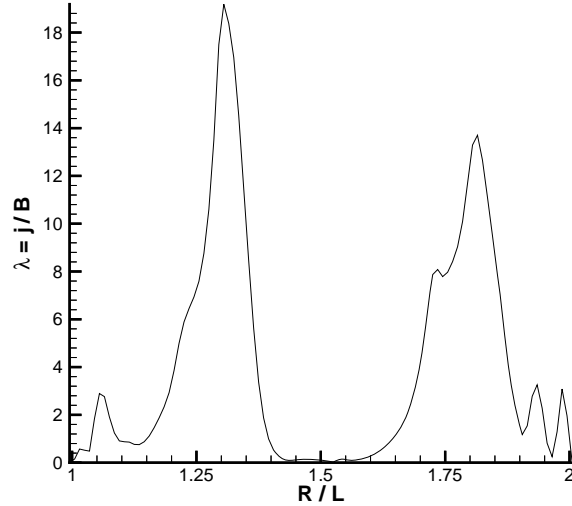


Figure 6.66: $\lambda = j/B$ versus R/L at midplane ($Z/L = 0.5$) at $t = 10$ for axisymmetric calculation with $R_o/a = 3.0$ and $S = 10^4$.

Figures 6.65 and 6.66 show λ versus radius at the midplane for both cases. Although less dramatic than the low aspect ratio runs, these plots also indicate the contrast in the evolution of the three-dimensional and axisymmetric runs. The kink instability again drives enhanced diffusion that allows the central core of the plasma to approach the Taylor state more rapidly than possible with purely axisymmetric diffusion. As a result, the λ profile is somewhat flatter for the three-dimensional simulation, although still obviously far from the Taylor state.

Figure 6.67 compares the ratio of the magnetic energy to the helicity as a function of time for all the $S = 10^4$ cases. The most important feature of the data is that the three-dimensional runs show a more rapid approach toward the Taylor state than the axisymmetric runs. This is due to an ideal kink instability originating from the $q = 1$ surface that compresses the poloidal and toroidal fields together, thus greatly enhancing the diffusion of the poloidal field into the central toroidal core as compared

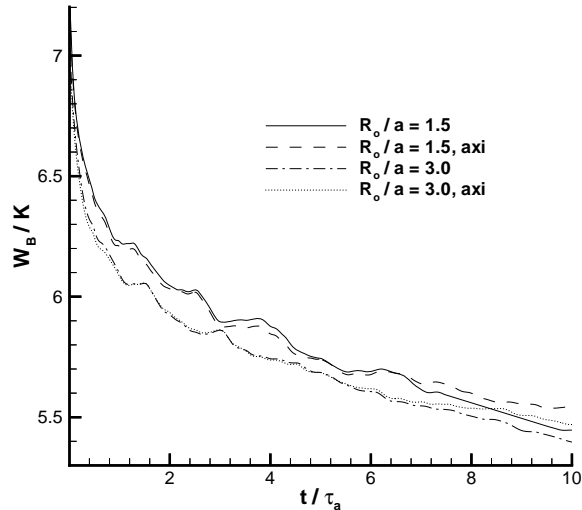


Figure 6.67: *Ratio of integrated magnetic energy to helicity for all of the $S = 10^4$ runs.*

to the axisymmetric case, for which the non-symmetric kink mode is not allowed.

This is not Taylor relaxation in the usual sense because the finite resistivity does not play a significant role. However, it is very like Taylor relaxation in that an instability allows the plasma to access lower energy states more rapidly than the helicity is dissipated. It is analogous to the situation where an ideally stable plasma undergoes resistive instabilities that drive the plasma toward the Taylor state.

The $R_o/a = 3.0$ plasma is at a lower W_B/K than the $R_o/a = 1.5$ plasma at $t = 10$ because it is evolving toward a lower eigenvalue of 4.48 (versus 4.62). The difference between the axisymmetric and the three-dimensional case is less for the higher aspect ratio runs. This is probably because the ideal MHD instabilities are more violent and disruptive at the lower aspect ratio.

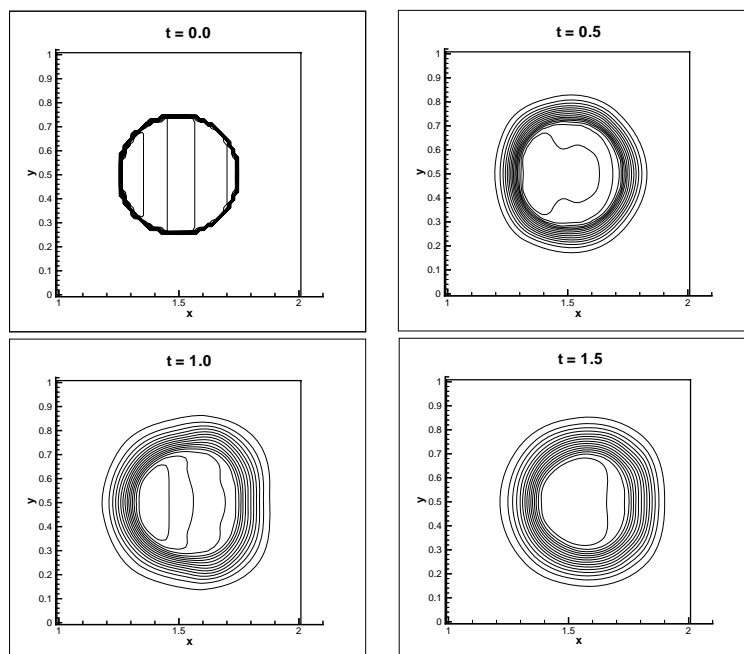


Figure 6.68: *Contours of toroidal field at intervals of 0.5 Alfvén time.*

6.7 Low Beta Simulations

This axisymmetric solution was done on a 62×62 poloidal grid. The torus has an aspect ratio of 3. The Lundquist number was 10^3 . Initially there is a uniform pressure such that $\beta = 0.01$. This is the value of β that we would have liked to have used for all of the simulations, because the Taylor theory is a zero pressure theory, and should be strictly true only in the limit as β goes to zero. In practice, experiments at a few percent β seem to be adequately explained by the Taylor theory. Unfortunately, at lower aspect ratios and higher Lundquist numbers, the code could not be reliably run at β this low.

The purpose of this simulation was simply to see how different the time evolution of the magnetic field is when comparing $\beta = 0.01$ with $\beta = 0.05$. Figures 6.68, 6.69, and 6.70 show a time sequence of contours of the toroidal field. For comparison, we

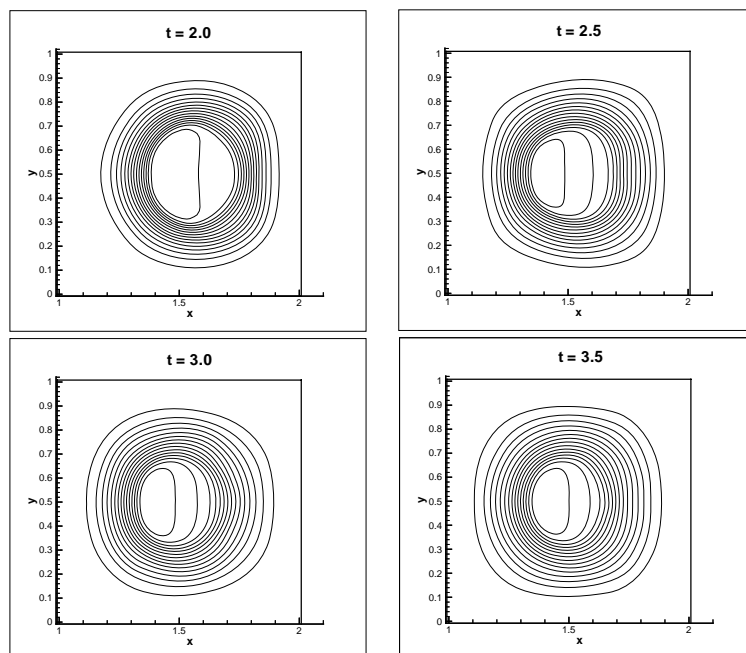


Figure 6.69: *Contours of toroidal field at intervals of 0.5 Alfvén time.*

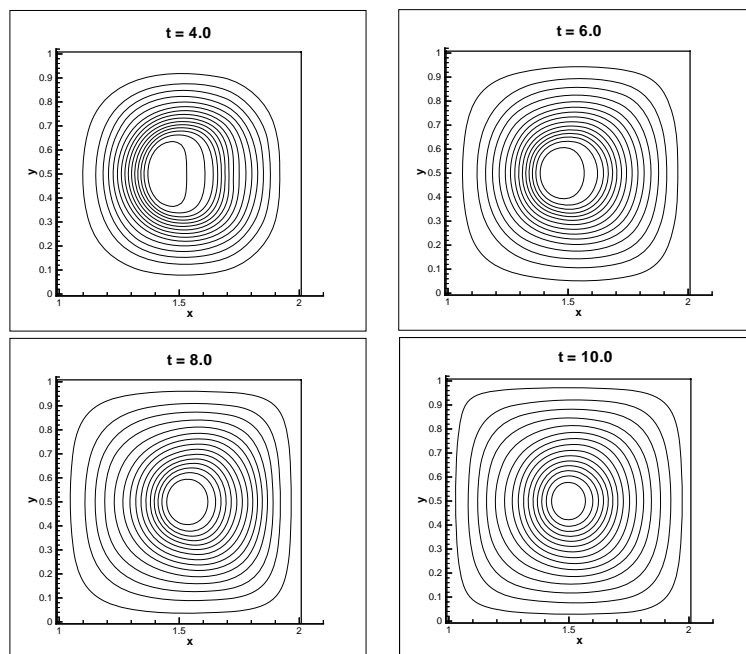


Figure 6.70: *Contours of toroidal field at intervals of 2.0 Alfvén times.*

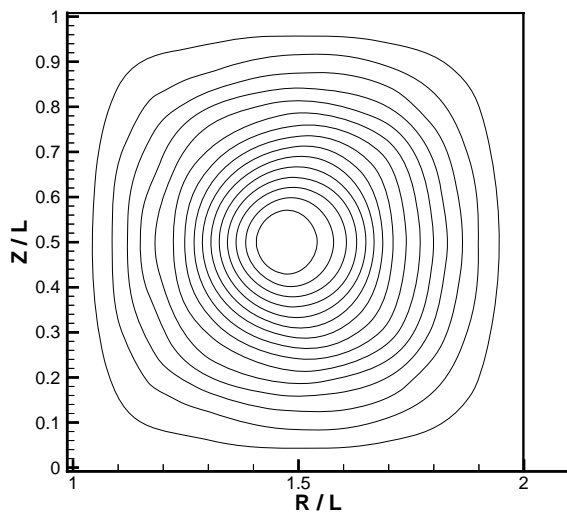


Figure 6.71: *Contours of toroidal field at $t = 10$ for axisymmetric run at $S = 10^3$ and $R_o/a = 3.0$.*

also show a contour plot of B_ϕ at $t = 10$ for an identical run at $\beta = 0.5$. This plot is shown in Figure 6.71. The larger pressure tends to damp out the plasma oscillations, and also leads to larger pressure gradients that appear to cause some pressure-induced instabilities on the outer flux surfaces. This leads to a slightly greater distortion of the flux surfaces at $t = 10$ for the higher β run.

Chapter 7

SUMMARY AND CONCLUSIONS

We presented new implicit and explicit algorithms for solving the non-ideal MHD equations. The algorithms are designed for solving problems at high Lundquist and Reynolds numbers. For this class of problems, the implicit algorithm is stable for any CFL number. The algorithms feature an approximate Riemann solver for the hyperbolic terms. The approximate Riemann solver combines Powell's multidimensional technique with Harten's discretization of the hyperbolic fluxes. The parabolic terms are discretized with a finite volume technique that uses an offset, face-centered mesh to calculate the interface fluxes. The implicit operator is inverted by using the LU-SGS iteration.

We then showed the results of several code validation test cases. The first was a one-dimensional MHD Riemann problem that verified that Harten's fluxes were correctly implemented. The second was an oblique shock problem that demonstrated that the approximate Riemann solver worked in two dimensions. It also showed that the implicit scheme was roughly four times more efficient at relaxing to the steady-state oblique shock solution than the explicit scheme. A Hartmann flow problem was solved to validate the resistive and viscous terms. Finally, a transient problem involving the convection of a magnetic field into an MPD thruster channel demonstrated that the implicit technique could maintain accuracy while taking time steps 100 times larger than allowed by the explicit scheme, which translated into a factor of three in CPU savings when the work required for the implicit iterations was accounted for.

We then applied the new code to the planar sheet pinch. This is a two-dimensional problem in which a current sheet of finite thickness and infinite extent out of the

plane is subject to resistive instabilities that cause the sheet to break into current filaments. The first step in our approach to solving this problem was to solve the linear, incompressible MHD equations to determine the most unstable eigenmode for a given wavelength. Then, this eigenmode was used to perturb the equilibrium current sheet. The code was then used to follow the linear behavior of the mode and the transition into nonlinear evolution and saturation. We were able to use the implicit code to solve this problem at a time step ten times larger than the possible with the explicit scheme.

Finally, the code was used to study the three-dimensional evolution and relaxation of a generic, RFP-like toroidal configuration. Simulations were done at Lundquist numbers of 10^3 and 10^4 and at aspect ratios of 1.5 and 3.0. These simulations were done using the explicit version of the code, since the implicit version could not be made to converge for these cases. The key aspect of these simulations was a comparison of the three-dimensional results with axisymmetric simulations at the same conditions. At the higher Lundquist number, the three-dimensional simulations displayed a more rapid relaxation toward the Taylor state than equivalent axisymmetric simulations. Specific conclusions from these runs are presented below.

7.1 Conclusions: Code Development

The chief advantage of this algorithm over an explicit approximate Riemann solver is the flexibility in choosing the time step. That is, if one is following dynamics that are occurring on time scales much longer than the shortest MHD time scales, then one can select the time step based on the desired accuracy, rather than on the numerical stability.

However, the major conclusion of the code development portion of this project was that the LU-SGS implicit scheme, while adequate for two-dimensional simulations such as the planar sheet pinch, was not able to converge for the three-dimensional

toroidal simulations. Apparently, the implicit operator was too approximate to get convergence for complicated three-dimensional simulations.

A suggestion for future work in this area would be to use a more accurate implicit operator that could achieve convergence for a wider range of problems. A leading candidate is a point-implicit scheme in which the hyperbolic terms are linearized exactly, and then the implicit matrix is solved using symmetric Gauss-Seidel iteration. Another advantage of this scheme would be that the parabolic terms could also be handled implicitly, which could allow simulations with highly resistive areas that are not practical with the current method.

7.2 Conclusions: 3D Relaxation of Toroidal Plasmas

The major conclusion of this portion of the work was that at $S = 10^4$, there was a significant difference between axisymmetric and three-dimensional simulations of toroidal relaxation. It was concluded that at this Lundquist number, the plasma was subject to an ideal kink instability in the vicinity of the $q = 1$ surface that caused enhanced diffusion that in turn led to rapid intermingling of the central toroidal flux core with the surrounding poloidal field. This mode was not available in the axisymmetric calculations, so for those simulations the fields intermingled more slowly due to axisymmetric diffusion. At an aspect ratio of 1.5, the dominant toroidal mode was $n = 3$, while at an aspect ratio of 3.0, the dominant mode was $n = 5$.

At $S = 10^3$, the ideal modes were damped out. In this case, there was virtually no difference in the three-dimensional and axisymmetric results.

The major limitation in this work was the relatively short run times. The highly resolved $S = 10^4$ runs were only run to 10 Alfvén times, which was only a small fraction of a resistive tearing time. Thus, it was not long enough to observe relaxation driven by resistive modes. In addition, the boundary conditions did not prevent helicity from entering or leaving the domain, which would be necessary for longer time

calculations. To continue this work in the future, it is recommended that perfectly conducting boundary conditions be applied and that the simulations be run out for several tearing times in order to observe resistively driven relaxation.

BIBLIOGRAPHY

- [1] A. Y. Aydemir, D. C. Barnes, E. J. Caramana, A. A. Mirin, R. A. Nebel, D. D. Schnack, and A. G. Sgro. *Phys. Fluids*, 28:898, 1985.
- [2] T. J. Barth. Analysis of implicit local linearization techniques for upwind and tvd algorithms. Technical Report 87-0595, AIAA Aerospace Sciences Meeting, 1987.
- [3] M. Brio and C. C. Wu. *J. Comput. Phys.*, 75:400, 1988.
- [4] E. J. Caramana, R. A. Nebel, and D. D. Schnack. *Phys. Fluids*, 26:1305, 1983.
- [5] P. Colella. *J. Comput. Phys.*, 87:171, 1990.
- [6] W. Dai and P. R. Woodward. *J. Comput. Phys.*, 111:354, 1994.
- [7] W. Dai and P. R. Woodward. *J. Comput. Phys.*, 115:485, 1994.
- [8] W. Dai and P. R. Woodward. *J. Comput. Phys.*, 121:51, 1995.
- [9] Message Passing Interface Forum. *Mpi: A message passing interface standard*. 1994.
- [10] J. P. Freidberg. *Ideal Magnetohydrodynamics*. Plenum Press, 1987.
- [11] M. H. Frese. Mach2: A two-dimensional magnetohydrodynamic simulation code for complex experimental configurations. Prepared by NumerEx for Phillips Laboratory, 1990.

- [12] H. P. Furth. *Phys. Fluids*, 28:1595, 1985.
- [13] H. P. Furth, J. Killeen, and M. N. Rosenbluth. *Phys. Fluids*, 6:479, 1963.
- [14] T. I. Gambosi, K. G. Powell, and D. L. De Zeeuw. *J. Geophys. Res.*, 99:21525, 1994.
- [15] A. Harten. *J. Comput. Phys.*, 49:357, 1983.
- [16] R. Horiuchi and T. Sato. *Phys. Fluids*, 29:1161, 1986.
- [17] K. Katayama and M. Katsurai. *Phys. Fluids*, 29:1939, 1986.
- [18] J. Killeen and A. I. Shestakov. *Phys. Fluids*, 21:1746, 1978.
- [19] W. Kordulla and M. Vinokur. *AIAA J.*, 21:917, 1983.
- [20] I. Lindemuth and J. Killeen. *J. Comput. Phys.*, 13:181, 1973.
- [21] Brian A. Nelson, Tom R. Jarboe, David J. Orvis, Leslie A. McCullough, J. Xie, C. Zhang, and L. Zhou. *Phys. Rev. Lett.*, 72(23):3666, 1994.
- [22] C. J. Nietubicz, T. H. Pulliam, and J. L. Steger. Numerical solution of the azimuthal-invariant thin-layer navier-stokes equations. Technical Report 79-0010, AIAA Aerospace Sciences Meeting, 1979.
- [23] H. Ok and D. S. Eberhardt. Solution of unsteady incompressible navier-stokes equations using an lu decomposition scheme. Technical Report 91-1611, AIAA Fluid Dynamics, Plasma Dynamics, and Lasers Conference, 1991.
- [24] R. E. Peterkin, A. J. Giancola, and J. E. Sturtevant. Mach2: A reference manual - fifth edition. Technical Report MRC/ABQ-R-1490, Mission Research Corporation, 1992.

- [25] K. G. Powell. Technical Report 94-24, ICASE, 1994.
- [26] S. Riyopoulos, A. Bondeson, and D. Montgomery. *Phys. Fluids*, 25:107, 1982.
- [27] P. L. Roe. *J. Comput. Phys.*, 43:357, 1981.
- [28] P. L. Roe and D. S. Balsara. *SIAM J. of App. Math.*, 56:57, 1996.
- [29] D. Schnack and J. Killeen. *J. Comput. Phys.*, 35:110, 1980.
- [30] D. D. Schnack, D. C. Barnes, Z. Mikic, D. S. Harned, and E. J. Caramana. *J. Comput. Phys.*, 70:330, 1987.
- [31] A. G. Sgro, A. A. Mirin, and G. Marklin. *Phys. Fluids*, 30:3219, 1987.
- [32] U. Shumlak, T. W. Hussey, G. J. Marklin, and R. E. Peterkin. *Bull. Am. Phys. Soc.*, 38:1894, 1993.
- [33] M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra. *MPI: The Complete Reference*. The MIT Press, 1996.
- [34] C. R. Sovinec and S. C. Prager. *Phys. Plasmas*, 3:1038, 1996.
- [35] J. L. Steger and R. F. Warming. *J. Comput. Phys.*, 40:263, 1981.
- [36] J. B. Taylor. *Phys. Rev. Lett.*, 33:1139, 1974.
- [37] A. C. Ting, W. H. Matthaeus, and D. Montgomery. *Phys. Fluids*, 29:3261, 1986.
- [38] W. J. Weber, J. P. Boris, and J. H. Gardner. *Comput. Phys. Comm.*, 16:243, 1979.

- [39] S. Wolfram. *Mathematica - A System for Doing Mathematics by Computer*. Addison-Wesley Publishing Company, Inc., 1991.
- [40] H. C. Yee, R. F. Warming, and A. Harten. *J. Comput. Phys.*, 57:327, 1985.
- [41] S. Yoon and A. Jameson. *AIAA J.*, 26:1025, 1988.
- [42] A. L. Zachery and P. Colella. *J. Comput. Phys.*, 99:341, 1992.
- [43] A. L. Zachery, A. Malagoli, and P. Colella. *SIAM J. of Sci. Comp.*, 15(2):263, 1994.

Appendix A

EIGENVECTORS FOR APPROXIMATE RIEMANN SOLVER

In this appendix the eigenvalues and eigenvectors of the modified Jacobian are presented for three-dimensional generalized coordinates. The discretized hyperbolic fluxes are

$$\begin{aligned} (\bar{\mathbf{R}}_h)_{ijk} &= \bar{\mathbf{F}}_{i+1/2} - \bar{\mathbf{F}}_{i-1/2} \\ &+ \bar{\mathbf{G}}_{j+1/2} - \bar{\mathbf{G}}_{j-1/2} + \bar{\mathbf{H}}_{k+1/2} - \bar{\mathbf{H}}_{k-1/2}. \end{aligned} \quad (\text{A.1})$$

Recall that

$$\bar{\mathbf{F}} = \xi_x \mathbf{F} + \xi_y \mathbf{G} + \xi_z \mathbf{H}, \quad (\text{A.2})$$

$$\bar{\mathbf{G}} = \eta_x \mathbf{F} + \eta_y \mathbf{G} + \eta_z \mathbf{H}, \quad (\text{A.3})$$

and

$$\bar{\mathbf{H}} = \zeta_x \mathbf{F} + \zeta_y \mathbf{G} + \zeta_z \mathbf{H}. \quad (\text{A.4})$$

The flux at each of the six cell faces is computed using the approximate Riemann solver formulas. So, for the ξ flux at the $i + 1/2$ face, for example, we have

$$\bar{\mathbf{F}}_{i+1/2} = \frac{1}{2} [\bar{\mathbf{F}}_{i+1} + \bar{\mathbf{F}}_i] + \frac{1}{2} \sum_k \bar{\phi}_{i+1/2}^k \bar{\mathbf{r}}_{i+1/2}^k. \quad (\text{A.5})$$

The summation term depends on the eigenvalues and eigenvectors of the generalized modified Jacobian, which is

$$\bar{\mathbf{A}} = \xi_x \tilde{\mathbf{A}} + \xi_y \tilde{\mathbf{B}} + \xi_z \tilde{\mathbf{C}}. \quad (\text{A.6})$$

The eigenvalues of $\bar{\mathbf{A}}$ are

$$\lambda = (U, U, U \pm c_f, U \pm c_s, U \pm v_a)^T, \quad (\text{A.7})$$

where

$$U = \xi_x v_x + \xi_y v_y + \xi_z v_z, \quad (\text{A.8})$$

$$c_f^2 = \frac{\xi_x^2 + \xi_y^2 + \xi_z^2}{2} \left\{ a^2 + c_a^2 + \left[(a^2 + c_a^2)^2 - 4a^2 \frac{v_a^2}{(\xi_x^2 + \xi_y^2 + \xi_z^2)} \right]^{1/2} \right\}, \quad (\text{A.9})$$

$$c_s^2 = \frac{\xi_x^2 + \xi_y^2 + \xi_z^2}{2} \left\{ a^2 + c_a^2 - \left[(a^2 + c_a^2)^2 - 4a^2 \frac{v_a^2}{(\xi_x^2 + \xi_y^2 + \xi_z^2)} \right]^{1/2} \right\}, \quad (\text{A.10})$$

and

$$v_a = \xi_x \frac{B_x}{\sqrt{\mu_o \rho}} + \xi_y \frac{B_y}{\sqrt{\mu_o \rho}} + \xi_z \frac{B_z}{\sqrt{\mu_o \rho}}. \quad (\text{A.11})$$

U is the contravariant velocity, c_f is the fast magnetosonic speed, c_s is the slow magnetosonic speed, and v_a is the Alfvén speed based on the ξ component of the magnetic field. The eigenvalues of $\bar{\mathbf{B}}$ (here, the Jacobian of \mathbf{G}) and $\bar{\mathbf{C}}$ (the Jacobian of \mathbf{H}) are obtained from the expression for $\bar{\mathbf{A}}$ by substituting the corresponding metric terms.

It turns out that it is much easier to determine the eigenvectors in terms of a "parameter vector" [27]. Following Roe and Balsara [28], we chose the parameter vector

$$\mathbf{W} = (\rho, v_x, v_y, v_z, B_x, B_y, B_z, p)^T. \quad (\text{A.12})$$

The Jacobian in terms of the parameter vector, denoted by $\bar{\mathbf{A}}_p$, is related to the conservative variable form through

$$\bar{\mathbf{A}}_p = \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{W}} \right]^{-1} \bar{\mathbf{A}} \frac{\partial \mathbf{Q}}{\partial \mathbf{W}}. \quad (\text{A.13})$$

Let $\bar{\mathbf{r}}_p$ denote a right eigenvector of $\bar{\mathbf{A}}_p$ and let $\bar{\mathbf{r}}$ denote the corresponding right eigenvector of the conservative Jacobian, $\bar{\mathbf{A}}$. These are related by

$$\bar{\mathbf{r}} = \frac{\partial \mathbf{Q}}{\partial \mathbf{W}} \bar{\mathbf{r}}_p. \quad (\text{A.14})$$

Similarly, the left eigenvectors are related by

$$\bar{\mathbf{I}} = \bar{\mathbf{I}}_p \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{W}} \right]^{-1}. \quad (\text{A.15})$$

It is relatively straight forward to find analytical expressions for the eigenvectors of $\bar{\mathbf{A}}_p$ using something like Mathematica. However, the difficulty lies in putting the eigenvectors into a form that remains well defined for various degeneracies that can occur. [3, 42] That is, for certain combinations of the magnetic field and ion sound speed, some of the wave speeds coincide. If the eigenvectors are kept in their non-normalized form, some of the terms go to infinity or to 0/0 for the degenerate cases. By factoring the eigenvectors appropriately, it is possible to normalize them in such a way that the eigenvectors will remain finite for all the degeneracies. Such normalizations have been done before for the Cartesian case. Here, we follow Roe and Balsara's [28] Cartesian normalization, and extend it to generalized coordinates.

First, we introduce a number of new variables. The normalized grid metrics are

$$\hat{\xi}_x = \sqrt{\frac{\xi_x^2}{\xi_x^2 + \xi_y^2 + \xi_z^2}} \quad (\text{A.16})$$

and so on for the others. The component of the magnetic field in the coordinate direction (the contravariant field) is

$$B_{con} = \hat{\xi}_x B_x + \hat{\xi}_y B_y + \hat{\xi}_z B_z. \quad (\text{A.17})$$

The component perpendicular to the coordinate direction is

$$B_{perp} = \sqrt{B^2 - B_{con}^2}. \quad (\text{A.18})$$

Then we introduce six variables that normalize the combinations of \mathbf{B} components that appear in the eigenvector expressions. These parameters are

$$\beta_1 = \frac{\hat{\xi}_y B_z - \hat{\xi}_z B_y}{B_{perp}} \quad (\text{A.19})$$

$$\beta_2 = \frac{\hat{\xi}_z B_x - \hat{\xi}_x B_z}{B_{perp}} \quad (\text{A.20})$$

$$\beta_3 = \frac{\hat{\xi}_x B_y - \hat{\xi}_y B_x}{B_{perp}} \quad (\text{A.21})$$

$$\beta_4 = \frac{\hat{\xi}_x \hat{\xi}_y B_y + \hat{\xi}_x \hat{\xi}_z B_z - (\hat{\xi}_y^2 + \hat{\xi}_z^2) B_x}{B_{perp}} \quad (\text{A.22})$$

$$\beta_5 = \frac{\hat{\xi}_y \hat{\xi}_x B_x + \hat{\xi}_y \hat{\xi}_z B_z - (\hat{\xi}_x^2 + \hat{\xi}_z^2) B_y}{B_{perp}} \quad (\text{A.23})$$

$$\beta_6 = \frac{\hat{\xi}_z \hat{\xi}_x B_x + \hat{\xi}_z \hat{\xi}_y B_y - (\hat{\xi}_x^2 + \hat{\xi}_y^2) B_z}{B_{perp}}. \quad (\text{A.24})$$

Note that

$$\beta_1^2 + \beta_2^2 + \beta_3^2 = 1, \quad (\text{A.25})$$

and

$$\beta_4^2 + \beta_5^2 + \beta_6^2 = 1. \quad (\text{A.26})$$

Finally, we also define

$$\alpha_f^2 = \frac{a (\xi_x^2 + \xi_y^2 + \xi_z^2) - c_s^2}{c_f^2 - c_s^2}, \quad (\text{A.27})$$

and

$$\alpha_s^2 = \frac{c_f^2 - a^2 (\xi_x^2 + \xi_y^2 + \xi_z^2)}{c_f^2 - c_s^2}. \quad (\text{A.28})$$

Not coincidentally, these factors remain finite for all possible magnitudes and directions of the magnetic field, as long as they are properly defined in the limit of B_{perp} approaching zero. In that case, we set β_1 through β_6 equal to $1/\sqrt{3}$.

Next, the normalized eigenvectors are presented. There are two eigenvectors corresponding to the two U eigenvalues. One corresponds to the entropy wave, and its left and right eigenvectors are

$$\mathbf{l}_{ent} = (1, 0, 0, 0, 0, 0, 0, -1/a^2), \quad (\text{A.29})$$

$$\mathbf{r}_{ent} = (1, 0, 0, 0, 0, 0, 0, 0)^T. \quad (\text{A.30})$$

The second U eigenvalue is a direct result of the modification to the Jacobian. Since this wave acts in such a way as to maintain $\nabla \cdot \mathbf{B} = 0$, it has been called a divergence wave. Its eigenvectors are

$$\mathbf{l}_{div} = (0, 0, 0, 0, \hat{\xi}_x, \hat{\xi}_y, \hat{\xi}_z, 0), \quad (\text{A.31})$$

$$\mathbf{r}_{div} = (0, 0, 0, 0, \hat{\xi}_x, \hat{\xi}_y, \hat{\xi}_z, 0)^T. \quad (\text{A.32})$$

The fast magnetosonic eigenvectors ($U \pm c_f$) are

$$\mathbf{l}_f^T = \frac{1}{2a^2} \begin{bmatrix} 0 \\ \pm [\hat{\xi}_x \alpha_f c_f + \alpha_s c_s \text{sgn}(v_a) \beta_4] \\ \pm [\hat{\xi}_y \alpha_f c_f + \alpha_s c_s \text{sgn}(v_a) \beta_5] \\ \pm [\hat{\xi}_z \alpha_f c_f + \alpha_s c_s \text{sgn}(v_a) \beta_6] \\ -\beta_4 \alpha_s a / \sqrt{\rho} \\ -\beta_5 \alpha_s a / \sqrt{\rho} \\ -\beta_6 \alpha_s a / \sqrt{\rho} \\ \alpha_f / \rho \end{bmatrix} \quad (\text{A.33})$$

and

$$\mathbf{r}_f = \begin{bmatrix} \alpha_f \rho \\ \pm [\hat{\xi}_x \alpha_f c_f + \alpha_s c_s \text{sgn}(v_a) \beta_4] \\ \pm [\hat{\xi}_y \alpha_f c_f + \alpha_s c_s \text{sgn}(v_a) \beta_5] \\ \pm [\hat{\xi}_z \alpha_f c_f + \alpha_s c_s \text{sgn}(v_a) \beta_6] \\ -\beta_4 \alpha_s a \sqrt{\rho} \\ -\beta_5 \alpha_s a \sqrt{\rho} \\ -\beta_6 \alpha_s a \sqrt{\rho} \\ \alpha_f \rho a^2 \end{bmatrix} \quad (\text{A.34})$$

Similarly, the slow magnetosonic eigenvectors are

$$\mathbf{I}_s^T = \frac{1}{2a^2} \begin{bmatrix} 0 \\ \pm [\hat{\xi}_x \alpha_s c_s - \alpha_f c_f \text{sgn}(v_a) \beta_4] \\ \pm [\hat{\xi}_y \alpha_s c_s - \alpha_f c_f \text{sgn}(v_a) \beta_5] \\ \pm [\hat{\xi}_z \alpha_s c_s - \alpha_f c_f \text{sgn}(v_a) \beta_6] \\ \beta_4 \alpha_f a / \sqrt{\rho} \\ \beta_5 \alpha_f a / \sqrt{\rho} \\ \beta_6 \alpha_f a / \sqrt{\rho} \\ \alpha_s / \rho \end{bmatrix} \quad (\text{A.35})$$

and

$$\mathbf{r}_s = \begin{bmatrix} \alpha_s \rho \\ \pm [\hat{\xi}_x \alpha_s c_s - \alpha_f c_f \text{sgn}(v_a) \beta_4] \\ \pm [\hat{\xi}_y \alpha_s c_s - \alpha_f c_f \text{sgn}(v_a) \beta_5] \\ \pm [\hat{\xi}_z \alpha_s c_s - \alpha_f c_f \text{sgn}(v_a) \beta_6] \\ \beta_4 \alpha_f a \sqrt{\rho} \\ \beta_5 \alpha_f a \sqrt{\rho} \\ \beta_6 \alpha_f a \sqrt{\rho} \\ \alpha_s \rho a^2 \end{bmatrix} \quad (\text{A.36})$$

The slow magnetosonic eigenvectors are obtained from the expressions for the fast magnetosonic eigenvectors by replacing α_f with α_s , α_s with $-\alpha_f$, and interchanging c_f with c_s .

Lastly, the Alfvén wave eigenvectors ($U \pm v_a$) are

$$\mathbf{l}_a = \left(0, \pm \frac{\beta_1}{\sqrt{2}}, \pm \frac{\beta_2}{\sqrt{2}}, \pm \frac{\beta_3}{\sqrt{2}}, -\frac{\beta_1}{\sqrt{2\rho}} - \frac{\beta_2}{\sqrt{2\rho}}, -\frac{\beta_3}{\sqrt{2\rho}}, 0 \right) \quad (\text{A.37})$$

and

$$\mathbf{r}_a = \left(0, \pm \frac{\beta_1}{\sqrt{2}}, \pm \frac{\beta_2}{\sqrt{2}}, \pm \frac{\beta_3}{\sqrt{2}}, -\beta_1 \sqrt{\frac{\rho}{2}}, -\beta_2 \sqrt{\frac{\rho}{2}}, -\beta_3 \sqrt{\frac{\rho}{2}}, 0 \right)^T \quad (\text{A.38})$$

Appendix B

EMI DESCRIPTION AND USER'S GUIDE

This appendix provides a description of version v9701 of the EMI MHD solver. The code is written in Fortran (mostly F77, but some F90). It is a parallel code, and uses the Message Passing Interface (MPI) standard subroutine calls to transfer data between parallel processors. The same code can also be run on a single processor.

Program Flow

The code consists of a main program, in `pmain.f`, that calls the various subroutines in which the actual calculations are performed. The major subroutines are "flux", "resistivity", "viscosity", "lower", and "upper". "Flux" finds the hyperbolic fluxes. It is the implementation of the approximate Riemann solver. It calls "average", which averages the data to the cell faces, and "limiter", which invokes the minmod limiter function. "Flux" is called separately for each coordinate direction. "Resistivity" and "viscosity" find the parabolic fluxes due to the resistive and viscous terms, respectively. "Lower" and "upper" invert the lower and upper triangular matrices that comprise the LU-SGS scheme. The following is a "pseudocode" summary of `pmain.f` that explains the program flow and the functions of the major subroutines.

```
read input file
initialize MPI
allocate arrays
call readgrid - reads in x,y,z arrays from grid file
call geometry - calculates volumes, areas, cell centers
```

```

call init - sets up initial conditions
if restart=true, call rstream - read restart file
call bc - sets up external boundary conditions
pass data between blocks
    call Bdrysend - sends data to buffers
    call Bdryrecv - requests data from buffers into ghost cells
    call BdryWait - waits until data received
call timestep - sets timestep according to input CFL number
call BdryWait - makes sure send buffers cleared
TOP OF OUTER TIME LOOP (for implicit, time-dependent calculations)
    time = time + dt_phys
    icount_phys = icount_phys + 1
    store q from previous time n and n-1 in qn and qn1
TOP OF INNER TIME LOOP
    tau = tau + dt
    icount = icount + 1
    call flux - puts hyperbolic fluxes in f,g, and h arrays
    call src_divb - puts divergence source term into rhs array
    if pseudo=true, call src_pseudo - adds time derivative source to rhs
    call resistivity - adds resistive fluxes to rhs
    call viscosity - adds viscous fluxes to rhs
    if axi=true, call src_axi - adds axisymmetric source to rhs
    rhs = f(i,..)-f(i-1,..) + g(..,j,..)-g(..,j-1,..) + ... - rhs
    if imp_flag=false, dq = -dt*rhs - finds new q for explicit case
    if imp_flag=true
        call lower
        call upper - finds dq for implicit case
    endif
endif

```



```

if mod(icount,ncycdiv)=0
    call cleandivb - clean up div B
    call bc
    call bdrySend, etc. - passes updated B between blocks
endif
call bc
call polflux - integrates B to get poloidal flux
call helicity - calculates magnetic helicity
call BdrySend, etc. - pass updated q between blocks
call timestep - find new time step
call converge - test for convergence of inner loop subiterations
dump output and restart files (for explicit case)
if tau > tfin, stop
BOTTOM OF INNER TIME LOOP
dump output files (for implicit case)
if time > tfin, stop
BOTTOM OF OUTER TIME LOOP

```

Indexing Conventions

The "i" index for real cells runs from 2 to $icels - 1$. The ghost cells are labeled 0, 1, $icels$, and $icels + 1$. The i direction is also known as the ξ direction, the j direction as the η direction, and the k direction as the ζ direction. The (i, j, k) triplet labels the cell center. The cell faces are labeled by 1/2 indices. The "left" face is $(i - 1/2, j, k)$, the "right" face is $(i + 1/2, j, k)$, the "bottom" face is $(i, j - 1/2, k)$, the "top" face is $(i, j + 1/2, k)$, the "back" face is $(i, j, k - 1/2)$, and the "front" face is $(i, j, k + 1/2)$. Similarly, the cell vertices have 1/2 indices. For example, the left, bottom, back vertex of cell (i, j, k) is labeled $(i - 1/2, j - 1/2, k - 1/2)$.

In the program, of course, all the indices must be integers, so we use the convention

that the $i - 1/2$ face is labeled as i , the $j - 1/2$ face as j , and so on. The only exception to this rule are the f,g, and h hyperbolic fluxes. For those quantities, i refers to the $i + 1/2$ face, and so on. The cell vertices use the convention that the $(i - 1/2, j - 1/2, k - 1/2)$ vertex is labeled (i, j, k) .

For some of the arrays, the ξ face is given the index of 1, the η face is 2, and the ζ face is 3. The blocks must also have labels. Using the same naming convention as for a cell, the "top" of the block is 1, the "right" is 2, the "bottom" is 3, the "left" is 4, the "back" is 5, and the "front" is 6.

Output Files

The output consists of terminal output, plot files, and restart files. The terminal output consists of the cycle number, time, and helicity at each iteration. The terminal output is currently set up for explicit runs only.

There are two types of plot files. One is the global output file, which writes the globally integrated magnetic energy, kinetic energy, internal energy, and magnetic helicity at each time to the file `pemi-global.tec`. The `.tec` suffix indicates that the file is in Tecplot format. For parallel runs, the integrals from each processor are accumulated and summed into a single file.

In addition, there is a 3D plot file that is dumped every `ncycdump` iterations or every multiple of `tdump` in terms of time. Each processor dumps output to a separate file called `p_ppp_ccccc.tec`, where `ppp` is the three-digit processor number (e.g. 000, 001, etc.), and `cccccc` is the six-digit cycle number. A typical file header looks like:

```
VARIABLES = x,y,z,rho,u,v,w,bx,by,bz,p,vi,vj,vk,bi,bj,bk,psi
ZONE T="zone 1",I=65,J=65,K=6,F = POINT
xc(1,1,1,1) yc(1,1,1,1) zc(1,1,1,1) ro(1,1,1,1) ...
xc(2,1,1,1) yc(2,1,1,1) zc(2,1,1,1) ro(2,1,1,1) ...
...
```

```
ZONE T="zone 2",I=65,J=65,K=6,F = POINT
```

```
...
```

The first line of the file lists the variables. Here, x , y , and z are the coordinates of the cell centers, ρ is the density, u , v , and w are the Cartesian velocity components, b_x , b_y , and b_z are the Cartesian components of the magnetic field, p is the pressure, v_i , v_j , and v_k are the velocity components in the ξ , η , and ζ directions, b_i , b_j , and b_k are the magnetic field components in those directions, and ψ is the helicity. For each block or "zone", there is a zone header that contains the zone number (e.g. zone 1), the number of points in the i , j , and k directions, and the specification of the Tecplot format ($F = \text{POINT}$). The POINT format means that at each point in the mesh, all of the variable values are listed in order. The ordering of the points is such that the i index varies fastest, followed by the j index, followed by the k index.

There is also a restart file that is dumped every `ncycrst` cycles or every multiple of `trst` in terms of time. Each processor dumps a restart file to `rst_ppp_#####.tec`. This file is similar to the `p_ppp_#####.tec` file, except that only the cell center locations and \mathbf{Q} are dumped at each point. In addition, there is a global restart file, `rst_#####.dat`, that saves the values of `icount`, `icount_phys`, `tau`, `time`, `dt`, `dt_phys`, and `tnorm`. Here, `icount` is the number of explicit time steps, `icount_phys` is the number of implicit time steps, `tau` is the time for an explicit calculation, `time` is the time for an implicit calculation, `dt` is the explicit time step, `dt_phys` is the implicit time step, and `tnorm` is the two-norm of the energy residual (meaningful for a steady-state iteration).

Input Files

For restart runs, the "`r_#####.tec`" and "`rst_#####.dat`" files are read from the "out-Dir" directory, where `#####` is the six-digit cycle number. For example, if the restart file was output at 100 iterations, then the restart files are `r_000100.tec` and `rst_000100.dat`. The `r_#####.tec` file is created by concatenating the `rst_ppp_#####.tec`

files created by each processor, where ppp is the three-digit processor number. Note that the processor numbers start with 000, rather than 001. For a single processor run, the rst_000_cccccc.tec file is simply renamed as r_cccccc.tec. A restart run is indicated by setting the restart flag to "true" and setting nycrst_select to the cycle number corresponding to the desired restart file.

The grid file is a Tecplot format file that contains the (x, y, z) location of each real and ghost cell vertex. The Tecplot format is discussed above.

The input file is called "input". It needs to be located in the directory that you run EMI from. The file consists of two NAMELIST sets. The complete list of input variables and their definitions is compiled in the following table.

Name	Type	Definition
icels(ndim)	INTEGER	number of cells in i direction
jcels(ndim)	INTEGER	number of cells in j direction
kcels(ndim)	INTEGER	number of cells in k direction
imp_flag	LOGICAL	if true, then implicit solve
alf	REAL*8	1 for pure implicit (other choices obsolete)
kap	REAL*8	weighting factor for LU-SGS (≤ 1 for stability)
newton	LOGICAL	if true, then Newton iteration for LU-SGS
mfs	REAL*8	free stream "Mach" number for oblique shock
flowangle	REAL*8	angle between flow and plate for oblique shock
bang	REAL*8	angle between B and plate for oblique shock
bmag	REAL*8	magnitude of B for oblique shock
shockx	LOGICAL	if true, then x direction shock tube
shocky	LOGICAL	if true, then y direction shock tube
rol	REAL*8	density of "left" state for shock tube
ul	REAL*8	u of "left" state for shock tube
vl	REAL*8	v of "left" state for shock tube
wl	REAL*8	w of "left" state for shock tube
<i>continued on next page</i>		

<i>continued from previous page</i>		
Name	Type	Definition
bxl	REAL*8	Bx of "left" state for shock tube
byl	REAL*8	By of "left" state for shock tube
bzl	REAL*8	Bz of "left" state for shock tube
pl	REAL*8	pressure of "left" state for shock tube
ror	REAL*8	density of "right" state for shock tube
ur	REAL*8	u of "right" state for shock tube
vr	REAL*8	v of "right" state for shock tube
wr	REAL*8	w of "right" state for shock tube
bxr	REAL*8	Bx of "right" state for shock tube
byr	REAL*8	By of "right" state for shock tube
bzr	REAL*8	Bz of "right" state for shock tube
pr	REAL*8	pressure of "right" state for shock tube
gam	REAL*8	ratio of specific heats
etag	REAL*8	1. if resistivity on, else 0.
lund1	REAL*8	inverse of Lundquist number
viscg	REAL*8	1. if viscosity on, else 0.
reyn1	REAL*8	inverse of Reynolds number
first_order_space	LOGICAL	if true, drop to first order in space
first_order_time	LOGICAL	if true, first order in time (implicit)
cn_phys	REAL*8	CFL number for implicit time steps
cn_pseudo	REAL*8	CFL number for explicit time steps
eps_subiter	REAL*8	convergence criterion for implicit subiterations
pseudo	LOGICAL	if true, then implicit, time-dependent
x0	REAL*8	reference scale length
b0	REAL*8	reference field strength
ro0	REAL*8	reference density
beta	REAL*8	initialization pressure
roig	REAL*8	initialization density
uig	REAL*8	initialization u velocity
<i>continued on next page</i>		

<i>continued from previous page</i>		
Name	Type	Definition
vig	REAL*8	initialization v velocity
wig	REAL*8	initialization w velocity
bxg	REAL*8	initialization Bx
byg	REAL*8	initialization By
bzg	REAL*8	initialization Bz
xmin	REAL*8	min radius ("R") for torus
xmax	REAL*8	max radius ("R") for torus
ymin	REAL*8	min "Z" coordinate for torus
ymax	REAL*8	max "Z" coordinate for torus
axi	LOGICAL	if true, then axisymmetric about y axis
epseig	REAL*8	eigenvalue smoothing for limiter
epseig	INTEGER	total number of blocks
cy11	LOGICAL	must be true if axi true
initflag	CHARACTER	string that specifies initialization
bctype(6,ndim)	CHARACTER	specifies BCs on each boundary of each block
blkData(7,ndim)	CHARACTER	specifies neighbors of each block
eps_div	REAL*8	convergence criterion for divergence clean
ncycdiv	INTEGER	cycles between divergence cleans
ncycdump	INTEGER	cycles between plot dumps
ncycfin	INTEGER	stop after this many cycles
tdump	REAL*8	time between plot dumps
tfin	REAL*8	stop at this time
trst	REAL*8	time between restart file dumps
ncycrst	INTEGER	cycles between restart file dumps
restart	LOGICAL	if true, read restart file
ncycrst_select	INTEGER	cycle number of restart file to be read
thetmin(ndim)	REAL*8	min "theta" (phi) per block for toroidal grids
thetmax(ndim)	REAL*8	max "theta" (phi) per block for toroidal grids

Other Variables

Most of the major variables are arrays that are dynamically allocated in pmain. There are also a small number of key variables that are passed through common blocks. The following table describes these variables. The header file or program file in which they are created is shown in parentheses. The index "n" always refers to block "n".

Name	Type	Definition
$x(i, j, k, n)$	REAL*8	x coordinate of vertex (i,j,k) (alloc.h)
$y(i, j, k, n)$	REAL*8	y coordinate of vertex (i,j,k) (alloc.h)
$z(i, j, k, n)$	REAL*8	z coordinate of vertex (i,j,k) (alloc.h)
$xc(i, j, k, n)$	REAL*8	x coordinate of cell center (i,j,k) (alloc.h)
$yc(i, j, k, n)$	REAL*8	y coordinate of cell center (i,j,k) (alloc.h)
$zc(i, j, k, n)$	REAL*8	z coordinate of cell center (i,j,k) (alloc.h)
$dsx(i, j, k, 1-3, n)$	REAL*8	x components of face 1-3 of cell (i,j,k) (alloc.h)
$dsy(i, j, k, 1-3, n)$	REAL*8	y components of face 1-3 of cell (i,j,k) (alloc.h)
$dsz(i, j, k, 1-3, n)$	REAL*8	z components of face 1-3 of cell (i,j,k) (alloc.h)
$vol(i, j, k, n)$	REAL*8	volume of cell (i,j,k) (alloc.h)
$f(i, j, k, 8, n)$	REAL*8	xi-direction hyperbolic flux vector (alloc.h)
$g(i, j, k, 8, n)$	REAL*8	eta-direction hyperbolic flux vector (alloc.h)
$h(i, j, k, 8, n)$	REAL*8	zeta-direction hyperbolic flux vector (alloc.h)
$q(i, j, k, 8, n)$	REAL*8	Q vector at time level n+1 (alloc.h)
$qn(i, j, k, 8, n)$	REAL*8	Q vector at time level n (alloc.h)
$qn1(i, j, k, 8, n)$	REAL*8	Q vector at time level n-1 (alloc.h)
$dq(i, j, k, 8, n)$	REAL*8	$Q(n+1) - Q(n)$ (alloc.h)
$rhs(i, j, k, 8, n)$	REAL*8	vector containing parabolic and source terms (alloc.h)
$etax(i, j, k, n)$	REAL*8	resistivity in x direction (alloc.h)
$etay(i, j, k, n)$	REAL*8	resistivity in y direction (alloc.h)
$etaz(i, j, k, n)$	REAL*8	resistivity in z direction (alloc.h)
$visc(i, j, k, n)$	REAL*8	coefficient of dynamic viscosity (alloc.h)

continued on next page

<i>continued from previous page</i>		
Name	Type	Definition
<code>phib(i,j,k,n)</code>	REAL*8	non-solenoidal magnetic field (alloc.h)
<code>psip(i,j,k,n)</code>	REAL*8	poloidal magnetic flux (alloc.h)
<code>ld</code>	INTEGER	dimension of Q vector (8) (param.h)
<code>ndim</code>	INTEGER	max number of blocks (param.h)
<code>TAG_FAK</code>	INTEGER	used to uniquely determine MPI tags (param.h)
<code>gm1</code>	REAL*8	ratio of specific heats - 1 (common.h)
<code>pie</code>	REAL*8	pi (common.h)
<code>mu0</code>	REAL*8	permeability of free space (common.h)
<code>time</code>	REAL*8	time for implicit runs (common.h)
<code>tau</code>	REAL*8	time for explicit runs (pmain.f)
<code>icount</code>	INTEGER	cycle counter for explicit runs (common.h)
<code>icount_phys</code>	INTEGER	cycle counter for implicit runs (pmain.f)
<code>nblock</code>	INTEGER	number of blocks on this processor (common.h)
<code>MAXBLOCKS</code>	INTEGER	max blocks per processor (parall.h)
<code>nprocs</code>	INTEGER	number of processors (parall.h)
<code>taskID</code>	INTEGER	processor ID number (0 to nprocs-1) (parall.h)
<code>blockID</code>	INTEGER	relates local block ID to global ID (parall.h)
<code>flipTable</code>	LOGICAL	orders indices for complex grids (parall.h)
<code>griddir</code>	CHARACTER	name of directory where grid file is (parall.h)
<code>gridfile</code>	CHARACTER	name of grid file (parall.h)
<code>outdir</code>	CHARACTER	name of output directory (parall.h)
<code>mydt</code>	REAL*8	local dt (parall.h)
<code>mydt_phys</code>	REAL*8	local dt_phys (parall.h)
<code>sendReq</code>	INTEGER	array of MPI send request IDs (parall.h)
<code>recvReq</code>	INTEGER	array of MPI receive request IDs (parall.h)