Development and Application of a Multi-Block High Order Finite Element Modeling Code as an Engineering Design Tool

Weston B. Lowrie

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

Doctor of Philosophy

University of Washington

2011

Program Authorized to Offer Degree: Aeronautics and Astronautics

University of Washington
Graduate School

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

Weston B. Lowrie

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

Chair of the Supervisory Committee:

_____
Uri Shumlak

Reading Committee:

_____
Uri Shumlak

_____
Thomas Jarboe

_____
Brian Nelson

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 Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, 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

Development and Application of a Multi-Block High Order Finite Element Modeling Code as an Engineering Design Tool

Weston B. Lowrie

Chair of the Supervisory Committee:
Professor Uri Shumlak
Aeronautics and Astronautics

An engineering design tool is developed to streamline the process of creating, verifying, and using complex computational meshes for use with numerical simulations. A fully three-dimensional high order finite element code is developed and verified with several different types of physics equations including anisotropic thermal conduction, and magnetohydrodynamcis (MHD). A multi-block framework and CAD/mesh generator interface is developed such that complex, non-axisymmetric, and non-simply connected topologies are possible with minimal complexity for the user. An *a priori* error estimation technique is developed using mesh quality metrics and is included as a step in the engineering design tool. One can assess a mesh's quality prior to numerical simulation and determine if it will yield acceptable results. It is found that the mesh quality analysis can predict the global error norms in the solution and therefore can be used as an *a priori* guide to improving computational meshes.

The multi-block framework is verified by solving a $m = 1$ kink mode in a Z-pinch and comparing to a linear stability analysis, yielding a positive agreement. Further studies of the Z-pinch include wall stabilization in a cylindrical geometry, and subsequently, a study of wall stabilization in a non-axisymmetric geometry made possible by the multi-block framework. The mesh deformation analysis is applied to the Z-pinch meshes and previous results are confirmed.

A non-axisymmetric and non-simply connected geometry representing the HIT-SI experiment is created using the CAD and mesh generator interface and multi-block framework. A mesh deformation analysis is applied to identify degenerate and poor mesh regions during mesh creation. Methods for repairing the mesh from degeneracies and further improvement for more accurate simulations is demonstrated. A spheromak MHD solution is computed on the HIT-SI mesh as a demonstration of the practicality of using the developments in this dissertation as an engineering design tool.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I feel exceptionally lucky to have been given the opportunity to pursue the education I have received. From my early days all the way up to now receiving a Ph.D, I am grateful for all that has enabled me to do this. I would especially like to thank my family for their encouragement and support in my pursuit of a career in science and engineering. I want to thank the Aeronautics and Astronautics Department and all the professors that have inspired and guided me through graduate school, especially the chair of my committee and advisor Professor Uri Shumlak. He has been an exceptional mentor and inspiration throughout my graduate school career and is always willing to help. I express my gratitude to Dr. Slava Lukin for his immense help with the HiFi code, his teaching me many of the ropes of computational plasma physics, and for always being a willing mentor. I am thankful for the opportunity he and Dr. Mark Linton gave me to visit and work at the Naval Research Laboratory in Washington D.C. in Autumn 2009. Dr. Alan Glasser has been a great resource with his knowledge of computational plasma physics and I would like to acknowledge all the invaluable discussions we have had over the years. I would like to thank the PI of the PSI-Center, Prof. Tom Jarboe for providing the opportunity to work in the center and also for discussions with him that helped keep the overarching goals of the project in mind. I am thankful for the valuable suggestions of my other committee members including Prof. Brian Nelson and Prof. Tom Mattick as well as Prof. Ulrich Hetmaniuk for his invaluable help with the grid metric analysis. I would like to thank the PSI-Center and all the associated computational resources that have made many of the simulations in this dissertation possible. Thanks to Susan Griffith for maintaining the PSI-Center computing resources and dealing well with the sometimes thankless nature of the job. Additionally I would like to thank the US Department of Energy for

funding the PSI-Center, and providing us access to the world class NERSC computing resources.

Lastly I would like to thank the amazing friends and fellow graduate students that have made graduate school such a memorable experience. Whether it was in the office or in the mountains it was great to have such an wonderful group of people to share the experience with. A special thanks goes to Aruna for bringing so much happiness into my life and giving me so much positive motivation.

Chapter 1

INTRODUCTION

One of the grand challenges of the 20th and continuing into the 21st century is human-engineered, controlled thermonuclear fusion on a scale large enough for commercial energy production. Controlled thermonuclear fusion has been demonstrated on a small scale, but it remains a challenge to have a commercially viable, economic, and environmentally clean means of obtaining fusion on earth. A major hurdle in this grand challenge is the understanding of plasma physics, which is the dominant field of study in the fusion community. Like many of the science fields, plasma science benefits greatly from the use of computer-aided design and numerical modeling. It is generally considered a necessary and essential aspect of plasma research and development to use such computational resources.

The focus and purpose of this dissertation is to contribute in development of a state of the art numerical tool for the advancement of computational plasma physics and apply the tool to problems of interest to the thermonuclear fusion community. The research is conducted through the Plasma Science and Innovation Center (PSI-Center) [1], and funded by U.S. Department of Energy, which aims to improve computational predictability of fusion devices.

The contributions made in this dissertation are all in light of creating a user-friendly engineering design tool with the ability to accurately model complex three-dimensional geometries with relative ease. There is a particular focus on problems involving fusion energy science and plasma physics. The main contributions are:

- ***A priori* mesh quality error analysis**

  The ability to analyze and quantify the errors associated with a mesh before running a simulation and predicting *a priori* potential solution errors is a useful design tool. This allows one to analyze a mesh prior to simulation with minimal computational effort, and identify areas of a mesh that need improvement. One can then improve problem mesh areas before simulation, which results in more accurate simulations and less wasted computational effort in poor simulations.

- **Multi-block framework**

  A multi-block framework is designed such that many structured element blocks

can be combined into a single computational domain in an unstructured arrangement. This enables significantly more complex geometries to be represented, while preserving the fundamental structured block elements. The ability to create complex geometries that are non-axisymmetric, and non-simply connected is essential to be a viable design tool. This is especially true in the fusion community where many of the experiments have complex geometries.

- **CAD and mesh generator interface**
  A useful design tool must have a user-friendly interface where one can quickly load or create the desired geometry, create a mesh, and run the simulation. Having a computer-aided design (CAD) and mesh generator interface accomplishes this requirement. The CUBIT [2] software is used as a CAD interface and mesh generator that can create the desired files for use in simulations.

1.1 Summary of Document

The use of high performance computing resources makes this development and research possible. A brief history of both the finite element method and high performance computing is thus first discussed in Chapter 2. The motivation for the development and the contributions made in this dissertation are discussed in Chapter 3. The main computational tool developed for this dissertation is the HiFi [3, 4] code and is discussed in more detail in Chapter 4, including the flux-source formulation, the high-order finite (spectral) element spatial discretization, implicit time advance, boundary condition, and initial condition formulation. A verification of HiFi with a visco-resistive MHD equation system is described in Chapter 5. A study of a mesh quality analysis is presented in Chapter 6, where an *a priori* error detection approach is discussed.

The multi-bock framework, and a CAD and mesh generator interface integration is discussed in Chapter 7. An application of the multi-block framework on Z-pinch simulations is discussed next in Chapter 8, and an application of the mesh quality metrics analysis to a Z-pinch problem is discussed in Chapter 9. A further exercise of the multi-block framework with a simulation using more complex HIT-SI geometry is

discussed in Chapter 10. Finally conclusions about the development and application work done in this dissertation are discussed in Chapter 11.

Chapter 2

HISTORY OF HIGH PERFORMANCE COMPUTING AND THE FINITE ELEMENT METHOD

Historically the finite element method and digital computer advancements have been intertwined. Dealing with large amounts of arithmetic is necessary in solving problems using the finite element method, and a digital computer is the only economical means of dealing with these large calculations. As computational power has increased over the decades, so has the ability to solve larger and larger problems. Both the origins of the finite element method and the use of high performance computing machines in the context of scientific computing is described below. This is meant to be a brief overview of each, and not an all inclusive description of their history. The goal is to give a better understanding of how the research in this dissertation fits into the current state of the art.

2.1 Finite Element Method

Despite the finite element method's name being coined by Clough [5] in 1960, the method's principles and use extend much further back in history. Leibniz in 1696 used the variational principle along with a piecewise polynomial approximation to derive the governing differential equation for the brachistochrone problem [6]. Although Leibniz was attempting to find a differential equation and not solve one, the variational principle and piecewise polynomial approximations are fundamental to the finite element method. Similarly, at the same time as Leibniz, Euler used these principles in the derivation of the Euler equations, named after himself. In 1851 Schellbach [7] used a finite element like method to solve for the minimum surface area enclosed by some arbitrary curve using a mesh of triangles and piecewise linear functions within the triangles.

Structural problems like plane elasticity was solved by Hrennikoff [8] in 1941, by breaking the domain into finite pieces and representing the connections between the pieces as bars, beams, and springs [9]. In 1943 McHenry also solved a plane stress problems, but used piecewise linear and cubic approximations over rectangular cells. Courant was also studying structural problems at this time and used variational methods for equilibrium and vibration solutions [10].

In the 1940s and 1950s a significant effort went into analyzing aircraft structures [11, 12, 13] and this lead to work at Boeing [14] of vibration and flutter analysis of a delta wing structure. The method used stiffness matrices of a collection triangular plates assembled into arbitrary shapes and was called the “direct stiffness method” [12]. Having the piecewise triangular elements allowed for strong agreements with the laboratory measurements of the physical models. It wasn’t until 1960 when Clough [5] used the term “finite element method (FEM)” in a paper written to demonstrate the procedure for the civil engineering profession [12]. This paper also demonstrated that the approximate solution would converge to the exact solution as the element size decreased [15]. This mathematical property is essential to the method and allowed it to be used successfully in solving many difficult engineering problems.

The early structural analysis of aircraft provided much of the foundation for the FEM and many different methods were proposed by private industry in the 1960s [9]. By 1962 the spectral finite element method had been proposed, and by the late 1960s bi-cubic splines were used for polynomial interpolation. It was also realized in the late 1960s that finite element methods could be used for unsymmetric, or non-self-adjoint operators found in fluid mechanics problems [15]. Oden in 1968 and 1970 [16, 17] and later Heinrich [18] published work on finite element used for fluid dynamics applications including the Navier-Stokes equations. Since the 1970s fluid problems have been solved extensively using the FEM. Algorithmic techniques like ‘upwinding’ further improved the computational efficiency, and led to more widespread use of finite element in fluid applications [15, 18]

As computing power increased over the decades, so did the size of problems attempted with FEM. In the early 1960s problems involving hundreds of variables were common, and by the late 1960s a three-dimensional FEM calculation of a pressure vessel involved approximately 18,000 variables [15]. By the 1980s fluid dynamics problems with millions of variables were possible. At this point a sweet spot of computational resources and algorithm efficiency made it possible to study with detail and precision complex fluid problems, including supersonic aircraft, automobile aerodynamics, as well as the MHD equations [19]. Solving systems with millions of variables

was routine throughout the 1990s and 2000s [20], and now it is possible to solve problems with billions of variables. The performance of computing resources is expected to increase further as *petascale* computing becomes more frequent, and *exascale* computing is on the horizon [21], and surely larger and larger problems will be solved on these machines.

2.2 High Performance Computing

High Performance Computing (HPC), also known as supercomputing, originated out of the early days of electronic computers. Early computers were designed to perform mathematical calculations, like the Complex Number Calculator (CNC) in 1940 [22]. It did not take long to realize the scientific possibilities of the early machines, and in 1945 the ENIAC [23, 24] computer was built at the University of Pennsylvania to compute firing tables for the U.S. Army, which involved repeatedly solving complex mathematical expressions. In 1951 the first commercial computer appeared: the Univac I [13] and were first sold to US Government agencies. Soon afterward in 1952 the Univac 1103 was sold, which had scientific computation in mind. IBM also launched a commercial machine in 1953: the 701 model [13].

In these early days of scientific computing, large aircraft companies purchased computers for analysis, but it was not until 1957 when a high level language called FORTRAN (short for "formula translation") was introduced. FORTRAN made it practical for engineers and scientists to use the computational machines of those days. This allowed scientists and engineers to solve problems of hundreds of variables in the early 1960s [13], and by the late 1960s problems with many thousands of variables were possible.

By the mid 1970s the first "supercomputer" was built by Seymour Cray, called the Cray-1 [25]. The Cray-1 was considered a vector computer, which could perform more than one operation at a time (as opposed to scalar processors). These vector operations can be thought of as the first type of parallel processing. This allowed for much faster calculations then the scalar processors when dealing with large arrays, but the

programmer must understand the process of "vectorization", which was a cumbersome programming technique. These machines were developed over the 1980s and 1990s, and did allow for large fluid dynamics problems to be solved, as well as many other scientific applications.

By the mid 1990s a shift from vector computing to parallel computing took place. The parallel machines used many off-the-shelf processors configured to compute a problem simultaneously. The term "massively parallel" refers to this type of computing, where thousands of processors could be used at once to solve a particular problem. These parallel machines represent the current state of the art, and continue to add more and faster processors. Much like the vector computers, programming for these machines requires writing software in parallel. Fortunately standards like MPI (message passing interface) [26, 27] were developed to manage processor-to-processor communication in these machines to ease the burden of programming. The current paradigm of increasing the total number of processors to increase performance seems to be a continuing trend as the field advances, but lately mixing specialized processors, and other types of computing hardware like graphical processing units (GPUs) to gain performance has become more common [21].

Chapter 3

MOTIVATION

The goal of the work in this dissertation is to be able to model physical systems of interest with predictive capabilities in a user-friendly manner for use as an effective engineering design tool. For complex geometries and complex equation systems, this is a challenging task, but having a state of the art numerical tool moves us closer to success in that direction. All of the features described are necessary for a useful numerical tool that has potential for predictive modeling.

The main numerical tool, HiFi, has been a work in progress throughout the work in this dissertation. Some features were present before the work began, and other were developed in parallel. A motivation for the code's features, and a brief description of each is described in this Chapter. Additionally, a motivation and description for the main contributions made to the advancement of the code are made.

3.1 Motivation and Description of Numerical Tool

3.1.1 Three Dimensional (3D) Spatial Representation:

Many physical phenomena are inherently 3D, and thus in order to understand the physics involved, simulations must also include all three dimensions. HiFi is a fully 3D high order finite (spectral) element code and has the ability to model these types of systems.

3.1.2 High Accuracy:

The finite element method is way of approximating a system of partial differential equations (PDEs) and in order for the approximation to be valid, the solution must be close to the actual PDE solution. Sufficient spatial accuracy is one of the necessary requirements to achieve an accurate approximation. HiFi has a spectral representation within each element, which means it can use an arbitrarily high polynomial degree. It can also have an arbitrary number of elements like linear finite element. Both the number of elements and spectral polynomial degree are used to achieve the desired spatial accuracy in HiFi.

3.1.3 Nonlinear Time Advance:

Another important aspect of the PDE approximation in the finite element method is its temporal accuracy. For a time dependent solution to be accurate, it must also have a sufficiently accurate time advance scheme. This becomes especially true when the system of equations is highly nonlinear. HiFi uses a second order nonlinear implicit Θ method to advance solutions in time (when $\theta = 0.5$ it is also known as the Crank-Nicholson method). The implicit time advance allows dynamically relevant time-steps rather than CFL imposed time-step limits. This is of great importance when dealing with systems of equations like MHD, that can have disparate spatial and temporal scales.

3.1.4 Scalable, Portable, and Extensible:

The state of the art computing resources use massively parallel architectures, and have a fairly short life cycle (3-5 years). This means the algorithms employed must take advantage of the large number of processors, and be able to keep up with the upgrade cycle of the computational resources. It is also desirable to have portability across a diverse range of machine architectures. HiFi uses the portable, extensible, toolkit for scientific computation (PETSc)[28, 29] libraries. These libraries include a large number efficient parallel linear and nonlinear solvers, as well as preconditioners, diagnostics, and a data structure framework for simple use of their libraries. They are available on many different systems, and employ state of the art linear algebra techniques, with a simple means of switching to different types of solvers, preconditioners, etc. HiFi uses these libraries to avoid "reinventing the wheel" while taking advantage of the latest mathematical solver techniques.

3.1.5 Simple PDE, Initial and Boundary Condition Specification:

A simple and straightforward means of specifying the governing equations, initial and boundary conditions allows a user to focus on the physics, rather than the numerics. This makes for a more efficient use of time spent formulating a model and time spent

using computational resources. HiFi uses the flux-source equation form, which allows simple specification of a large class of PDEs. Hundreds of different equation systems are possible, and small changes to previously formulated equation systems are trivial to make. This allows the user great flexibility in formulating a model for a physical system of interest. Initial and boundary conditions are equally simple to formulate, and follow the same flux-source framework. Additionally, the physics equation specification has been modularized and separated from the main solver library. This enables a user to focus on the physics without having to deal with the "guts" of the code.

3.2 Major Contributions

The major contributions made are described below. A brief motivation for each is described as well as some of the details of the development.

3.2.1 A priori Error Estimation

An important aspect of numerical simulations is the ability to quantify if a mesh is of acceptable quality and whether or not it needs refinement. Large deformations in a mesh can yield unacceptable errors in a simulation solution, and need careful consideration. An *a priori* mesh quality metric analysis is used to quantify the quality of a mesh. If the metrics detect unacceptable levels of distortion, the mesh can be refined before any simulation is performed in order to improve solution accuracy, and reduce wasted CPU usage on inaccurate simulations. The description of the mesh quality analysis is described in Chapter 6. A more practical application of the mesh quality analysis is described in Chapter 9.

3.2.2 Complex Geometries with Multi-Block Formulation:

In the magnetic confinement fusion community, many plasma experiments do not have trivial geometries. They can be non-axisymmetric, non-simply connected, and often have features that cannot be modeled as simple geometric structures (i.e. cylinder, cube, torus, etc.). To accurately model complex geometric systems, the computational

domain must be sufficiently complex to capture the intended geometry. A multi-block framework has been developed such that more complex geometries can be modeled. HiFi performs its calculations on unit logical cubes (blocks) and applies a logical-to-physical coordinate transformation in order to achieve domain shapes of interest. The computational domain geometry is specified by the user, either by external mesh file or manual geometric specification of the blocks. Each block must logically be a cube with a structured mesh, but the code allows for unstructured block connections. This multi-block development further increases the geometric flexibilities, and allows for more accurate representations of the modeled physical system. Section 7.1 describes the development of this multi-block framework.

In addition to development of the multi-block framework, application of the code improvements are included. Z-pinch simulations taking advantage of multi-block geometries are discussed in Chapter 8. Additionally a simulation with the HIT-SI [30, 31, 32, 33] geometry is discussed in Chapter 10.

3.2.3 Advanced Mesh Generation and CAD interface

With complex geometries comes complex mesh generation needs. Specifying the logical-to-physical coordinate transformation of a computational domain can be a tedious exercise, and sometimes practically unfeasible. It is therefore desirable to have a CAD interface for specifying the computational domain geometry. Coupled with the multi-block formulation is a CAD interface and advanced mesh generation abilities. It is important to be able to create high quality meshes of complex geometries and the CUBIT [2] mesh generator is employed to do this. CUBIT is an external mesh generator where a user can create a high quality mesh and export for use in HiFi simulations. Additionally, CUBIT has CAD file importing features, as well as its own minimal CAD interface, and is used as a CAD interface and mesh generator. It fits well into the multi-block framework and additionally a user can specify the boundary condition type graphically, greatly reducing the workload of a comparable manual specification. Section 7.3 describes the mesh generator interface.

Chapter 4

OVERVIEW OF THE HIFI HIGH-ORDER FINITE (SPECTRAL) ELEMENT MODELING CODE

The HiFi code was originally formulated by Glasser and Tang [3] at Los Alamos National Laboratory in 2002 as the two-dimensional SEL code. It was further developed by Glasser and Lukin [34] from 2003 to 2007. It was then brought to the University of Washington's PSI-Center [1], where Glasser, Lowrie, Lukin, and Shumlak [4] developed the three-dimensional version and called it HiFi. The development as of 2011 is continuing jointly at the University of Washington and the Naval Research Laboratory in Washington, DC.

An overview of the features of HiFi are described below. Specifically its partial differential equation (PDE) formulation, its means of spatial discretization: the high-order finite (spectral) element method, its implicit time advance scheme, as well as its boundary and initial condition formulation are described below.

4.1 Partial Differential Equation (PDE) Formulation - Flux-Source Form

The HiFi code uses the flux-source formulation, such that varying the PDEs is a simple task. This enables modeling of various PDE systems while using the same main solver formulation. The flux-source equation form is a generic equation class and has the form

$$\frac{\partial \vec{Q}}{\partial t} + \nabla \cdot \vec{F} = \vec{S}, \tag{4.1}$$

where $\vec{Q}$ is a vector of primary variables, and $\vec{F}$ and $\vec{S}$ are vectors of fluxes and sources associated with each of the primary variables. The fluxes and sources can be functions of time, space, the primary variables themselves, and the spatial derivatives of the primary variables,

$$\vec{F} = \vec{F}(t, \vec{x}, \vec{Q}, \nabla\vec{Q}), \tag{4.2}$$

$$\vec{S} = \vec{S}(t, \vec{x}, \vec{Q}, \nabla\vec{Q}). \tag{4.3}$$

This formulation allows for a wide range of applicable equation sets that is free form for the user to decide how to implement. This is useful for instance in testing different equation sets with the same solver, or simply making small modifications to an existing

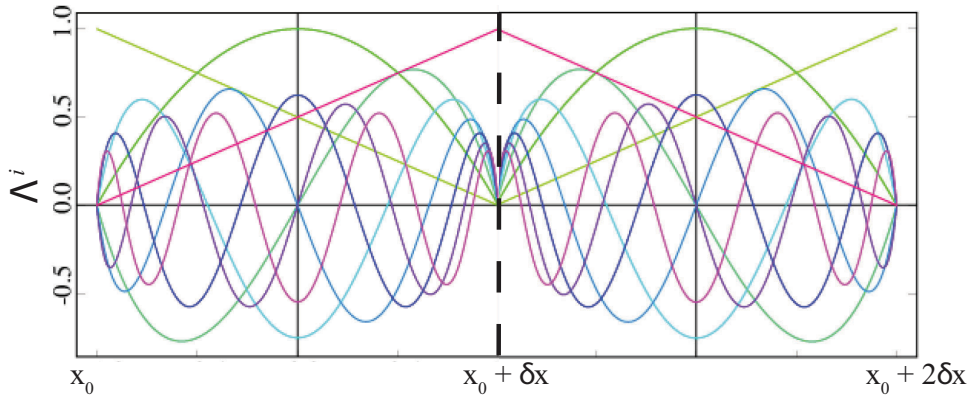


Figure 4.1: A one-dimensional representation of two adjacent spectral elements using the Jacobi polynomial basis functions Λ^i along with the linear functions used in the HiFi code spectral element spatial discretization. The domain range is $x \in [x_0 + n\delta x, x_0 + (n+1)\delta x]$. Each element has Jacobi polynomials ranging from 1^{st} order up to 8^{th} order ($np = 8$) with the linear functions providing the C^0 continuity between elements.[34].

equation set.

4.2 High-Order Finite (Spectral) Element Spatial Discretization

The method of spatial discretization of the PDEs in HiFi is a high-order finite (spectral) element method. Each element makes up a piecewise continuous solution of the whole solution domain. The elements are C^0 continuous across elements, which means continuity of the PDE functions is enforced between elements, but does not enforce continuous first or second derivatives. Within each element there is a spectral representation using basis functions of arbitrary polynomial degree. HiFi uses the Jacobi polynomials [35] as the basis functions for its spectral representation within elements, and linear functions to provide the C^0 continuity between elements. Figure 4.1 shows a one-dimensional picture of two adjacent elements. Notice the high-order basis functions go to zero at the element boundaries, while the linear basis functions are non-zero, and thus provide the continuity between elements.

4.2.1 Logical to Physical Grid Transformation

In order to allow the user to specify a physical domain shape of interest, a logical-to-physical transformation is used in the HiFi code. This transformation allows the code to make calculations on a logical cube, while the user can specify their equations, initial and boundary conditions on a physical domain of their choice. In 2D the logical coordinates are defined as

$$(\xi, \eta) \in \Omega_e : [0, 1] \times [0, 1]. \tag{4.4}$$

The logical dimensions are transformed to some physical coordinates x and y of arbitrary range, such that $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$.

To demonstrate how the transformation fits into the solver consider Poisson's equation

$$\nabla^2 \phi(x, y) = S(x, y), \tag{4.5}$$

where $\phi(x, y)$ is some scalar variable and $S(x, y)$ is some source forcing function. For this equation to fit into the flux-source form, it is rewritten as

$$\cancelto{0}{\frac{\partial \phi}{\partial t}} + \nabla \cdot \vec{F}(x, y) = S(x, y), \tag{4.6}$$

where $\vec{F}(x, y) = \nabla \phi(x, y)$. The equation is rewritten in terms of the logical coordinates ξ and η, and multiplied by the Jacobian J. This multiplication by the Jacobian is important, because J can potentially become singular for certain grid distortions. In logical metric space, the equation is written as

$$\begin{aligned} &\frac{\partial}{\partial \xi}\left[(\nabla\phi \cdot \nabla x) J \frac{\partial \xi}{\partial x} + (\nabla\phi \cdot \nabla y) J \frac{\partial \xi}{\partial y}\right] \\ +&\frac{\partial}{\partial \eta}\left[(\nabla\phi \cdot \nabla x) J \frac{\partial \eta}{\partial x} + (\nabla\phi \cdot \nabla y) J \frac{\partial \eta}{\partial y}\right] = JS, \end{aligned} \tag{4.7}$$

where

$$J = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}. \tag{4.8}$$

4.2.2 Weak Form of Equations

In the solver, PDEs are transformed into the weak form by projecting the solution onto a polynomial basis. With the Galerkin discretization, the 2D Equation 4.7 in weak form becomes

$$\int_{\Omega_e} \left[F_x \left(\frac{\partial \xi}{\partial x}\frac{\partial \alpha^i}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial \alpha^i}{\partial \eta} \right) + F_y \left(\frac{\partial \xi}{\partial y}\frac{\partial \alpha^i}{\partial \xi} + \frac{\partial \eta}{\partial y}\frac{\partial \alpha^i}{\partial \eta} \right) + \textit{boundary} - S\alpha^i \right] J d\xi d\eta = 0 \tag{4.9}$$

where $\alpha^i(\xi, \eta)$ is a 2D basis (trial) function polynomial defined in the domain Ω_e, and

$$F_x = \frac{\partial \phi}{\partial x} \quad \text{and} \quad F_y = \frac{\partial \phi}{\partial y}. \tag{4.10}$$

The weak form equation is solved for every element in the domain. Notice the fluxes F_x and F_y are multiplied by derivatives of the logical coordinates with respect to the physical coordinates, and the whole equation is integrated in the logical metric space.

The derivatives of the basis functions ($\frac{\partial \alpha^i}{\partial \xi}$ and $\frac{\partial \alpha^i}{\partial \eta}$) are known exactly, assuming the basis functions chosen have analytical derivatives. The derivatives $\frac{\partial \xi}{\partial x}$, $\frac{\partial \xi}{\partial y}$, $\frac{\partial \eta}{\partial x}$, and $\frac{\partial \eta}{\partial y}$ must also be known in order to integrate the equation and solve for $\phi(x, y)$. These derivatives can be represented in terms of the Jacobian and derivatives of the physical coordinates with respect to the logical coordinates.

$$\begin{aligned} \frac{\partial \xi}{\partial x} &= \frac{1}{J}\frac{\partial y}{\partial \eta} & \frac{\partial \xi}{\partial y} &= -\frac{1}{J}\frac{\partial x}{\partial \eta} \\ \frac{\partial \eta}{\partial x} &= -\frac{1}{J}\frac{\partial y}{\partial \xi} & \frac{\partial \eta}{\partial y} &= \frac{1}{J}\frac{\partial x}{\partial \xi}, \end{aligned} \tag{4.11}$$

where the Jacobian J is given by equation 4.8

4.3 Implicit Temporal Advance

A Newton-Krylov iterative method is used to advance the system of PDEs in time. After Equation 4.1 is cast in the weak form (as done for Poisson's equation in Section

4.2.2), it can be rewritten as a matrix-vector equation

$$\mathbb{M}\dot{\mathbf{q}} = \mathbf{r}(t, \mathbf{q}) \tag{4.12}$$

where $\mathbb{M}$ is a matrix representing the equation, $\mathbf{q}$ is a vector of the primary variables, and $\mathbf{r}$ is called the *right hand side*. Two well known algorithms for temporal advance are available in HiFi: the Θ scheme, and the second order backward differencing formulation (BDF2).

4.3.1 Θ Scheme

Equation 4.12 is temporal discretized as:

$$\mathbb{M}\left(\frac{\mathbf{q}^{n+1} - \mathbf{q}^{n}}{\Delta t}\right) = \theta\mathbf{r}(t^{n+1}, \mathbf{q}^{n+1}) + (1-\theta)\mathbf{r}(t^{n}, \mathbf{q}^{n}), \tag{4.13}$$

where θ is the method's time-centering parameter, and n is the time-step number. The Θ scheme is a second order accurate, non-dissipative method when $\theta = 0.5$, and also known as the Crank-Nicholson method. Throughout the work in this dissertation $\theta = 0.5$ is used.

Using equation 4.13, the goal is now to solve for $\mathbf{q}^{n+1}$. This is done by using a Newton-Krylov method. This method uses a combination of a nonlinear Newton iteration, along with a linear Krylov subspace method within each Newton iteration. Equation 4.13 is recast as

$$\mathbf{R}(\mathbf{q}_i^{n+1}) \equiv \mathbb{M}(\mathbf{q}_i^{n+1} - \mathbf{q}^{n}) - \Delta t[\theta\mathbf{r}(t^{n+1}, \mathbf{q}^{n+1}) + (1-\theta)\mathbf{r}(t^{n}, \mathbf{q}^{n})], \tag{4.14}$$

where i is the nonlinear iteration count, and $\mathbf{R}$ is called the residual. The goal of the Newton iteration is to drive $\mathbf{R} \to 0$ by successive linear Krylov iterations. Each Krylov iteration involves solving a large matrix-vector equation that takes most of the HiFi algorithm's computing time.

Both the nonlinear and linear solvers are performed in the PETSc framework. The scalable nonlinear equation solvers (SNES) as well as the scalable linear equation

solvers (KSP)[28, 29] are employed to handle the overhead in HiFi. Not only does PETSc manage much of the parallelization of the algorithms, it also allows for simple switching of linear or nonlinear solvers. For instance from run-time command line flags, the linear solver can be switched from the iterative GMRES method to the SuperLU direct solver method. Many more solvers are available through PETSc, and using their framework makes it simple switch to different types.

4.4 Boundary Condition Formulation

The boundary conditions (BC) in HiFi are formulated in the same flux-source form as the governing equations. This means any equation that can fit in the general flux-source form can be imposed. Additionally the fluxes and sources can be functions of the second spatial derivative of the primary variables and of an outward unit normal vector,

$$\vec{S} = \vec{S}(t, \hat{n}, \vec{x}, \vec{Q}, \nabla_{\vec{x}}\vec{Q}, \nabla_{\vec{x}\vec{x}}\vec{Q}). \tag{4.15}$$

This allows for a flexible BC specification, but also some special predetermined boundary conditions are included. A flux BC where the normal flux through a boundary can be specified, a “natural” BC where the interior PDEs determine the boundary flux, and “zeroflux” where the flux through a boundary is set to zero. Another special case is an integral BC, where a restriction on the integral of the solution along some face or edge is imposed. This allows for global coupling of the solution along this boundary. An example of an integral BC is the “polar” axis where a face is collapsed to a edge to allow a cylindrical domain from a logical cube. Additionally periodic boundary conditions are also included as a special case.

4.5 Initial Condition Formulation

The initial conditions in HiFi are specified either analytically or from a previous HiFi computation. An analytic specification is written with respect to the spatial coordinates

$$\vec{Q}_i = \vec{Q}_i(\vec{x}). \tag{4.16}$$

where $\vec{Q}_i$ is the initial condition for all equation system variables. The finite element spatial representation is projected onto basis functions, and the coefficients of the functions are computed and evolved in the code. The initial condition specified $\vec{Q}_i$ must be converted into the finite element space. To solve for the basis function coefficients, a linear system is solved prior to the time advance of the solution. A linear system is solved on each element for $\vec{Q}_\alpha$,

$$\mathbf{A}\vec{Q}_\alpha = \vec{Q}_i \tag{4.17}$$

where $\vec{Q}_\alpha$ are the basis function coefficients, and $\mathbf{A}$ is a matrix representing the particular basis functions.

Chapter 5

VERIFICATION OF THE HIFI CODE

In order to test the 3D HiFi code's functionality, both a linear and nonlinear MHD problem are chosen. A viscous, hyper-resistive MHD with isotropic heat conduction module written in 3D Cartesian coordinates provides the basis for performing the tests. The equations are written in flux-source form as

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\vec{v}\\ \vec{A}\\ \frac{1}{\gamma-1}p\\ 0\end{bmatrix}+\nabla\cdot\begin{bmatrix}\rho\vec{v}\\ \rho\vec{v}\vec{v}+p\vec{I}-\mu\nabla\vec{v}\\ -\nu\nabla\vec{j}\\ \frac{\gamma}{\gamma-1}p\vec{v}-\kappa\nabla T\\ (\nabla\cdot\vec{A})\vec{I}-\nabla\vec{A}\end{bmatrix}=\begin{bmatrix}0\\ \vec{j}\times\vec{B}\\ \vec{v}\times\vec{B}-\eta\vec{j}\\ (\vec{v}\cdot\nabla p+\eta\vec{j}\cdot\vec{j}+\mu(\nabla\vec{v}:\nabla\vec{v})\\ \vec{j}\end{bmatrix}, \tag{5.1}$$

where ρ is mass density, $\vec{v}$ is fluid velocity, $\vec{A}$ is magnetic vector potential, p is pressure, and γ is the ratio of specific heats. The symbol : is the tensor double dot product operator defined as

$$\nabla\vec{v}:\nabla\vec{v}=\sum_{i=1}^{3}\sum_{j=1}^{3}\nabla v_{ij}\nabla v_{ji}, \tag{5.2}$$

resulting in a scalar value. The dissipative parameters μ, η, ν, and κ are viscosity, resistivity, hyper-resistivity, and isotropic heat conduction respectively. These parameters represent the inverse of the dimensionless parameters such as the Reynolds number $\mu = 1/Re$ (See Appendix C for more details). The hyper-resistive parameter ν multiplies a fourth order term providing high-order dissipation. This is useful for damping high-frequency noise in the current $\vec{j}$ and providing numerical stability. Additionally the magnetic field intensity $\vec{B} = \nabla \times \vec{A}$, current density $\vec{j} = \nabla \times \nabla \times \vec{A}$ and temperature $T = p/\rho$ are defined. The linear test problem investigates MHD wave propagation, while the nonlinear test problem investigates the dynamics of a spheromak tilt instability using the system of equations in 5.1. These are described below.

5.1 Dissipative MHD Wave Speeds and Decay Rates

Dissipative MHD wave propagation is tested in HiFi by initializing both the slow and fast magnetosonic waves in a doubly periodic box, and measuring the wave speed, as

well as the wave decay over some time. This is done by computing the complex eigensystem of a linearization of the dissipative MHD equations defined in Equation 5.1 and using the resulting eigenvectors to construct a perturbation. Each of the eigenvectors found from the linearized system of equations represent one of the waves (e.g. slow magnetosonic, fast magnetosonic, or shear Alfvén) present in the system. This enables "pure" waves to be perturbed, and when in the linear regime no other wave modes are present. The system of equations is linearized using the form

$$u(\vec{x}, t) = u_0 + \delta u_1(\vec{x}, t), \tag{5.3}$$

with u_1 assumed in the form

$$u_1(\vec{x}, t) \equiv e^{-i(\vec{k}\cdot\vec{x}-\omega t)}, \tag{5.4}$$

and δ is the magnitude of the perturbed quantity. Using this form of the perturbation and assuming all background quantities u_0 are constant or zero allows for sufficient linearization of the system of equations. Additionally the background magnetic field is assumed to be constant and aligned in one direction (in this case the $\hat{x}$ direction). The eigensystem is then found from the linearized system of equations. The full details of this linearization and eigensystem solve can be found in Appendix B.1. The assumed form of the perturbation from Equation B.5 can be rewritten in terms of sines and cosines and then applied to each of the system of equation's primary variables.

$$\vec{u}(\vec{x}) = \vec{u}_0 + \hat{\delta}\left[\mathbb{R}e(\vec{R})cos(\vec{k}\cdot\vec{x}) - \mathbb{I}m(\vec{R})sin(\vec{k}\cdot\vec{x})\right], \tag{5.5}$$

where $\vec{u}$ are the primary variables, $\vec{u}_0$ are the background quantities, $\hat{\delta}$ is the magnitude of the perturbation, $\vec{k}$ is the wave number, and $\vec{R}$ are the complex eigenvectors of the system. The real and imaginary parts of the eigenvectors $\mathbb{R}e(\vec{R})$ and $\mathbb{I}m(\vec{R})$ result from solving the linearized system of equations and are chosen for one of the particular characteristics in the system. Figure 5.1 is an example of magnitudes of v_y when one of the "pure" waves are initialized in a doubly periodic box. Other variables have similar forms, which are determined by their associated eigenvector magnitudes. In this case

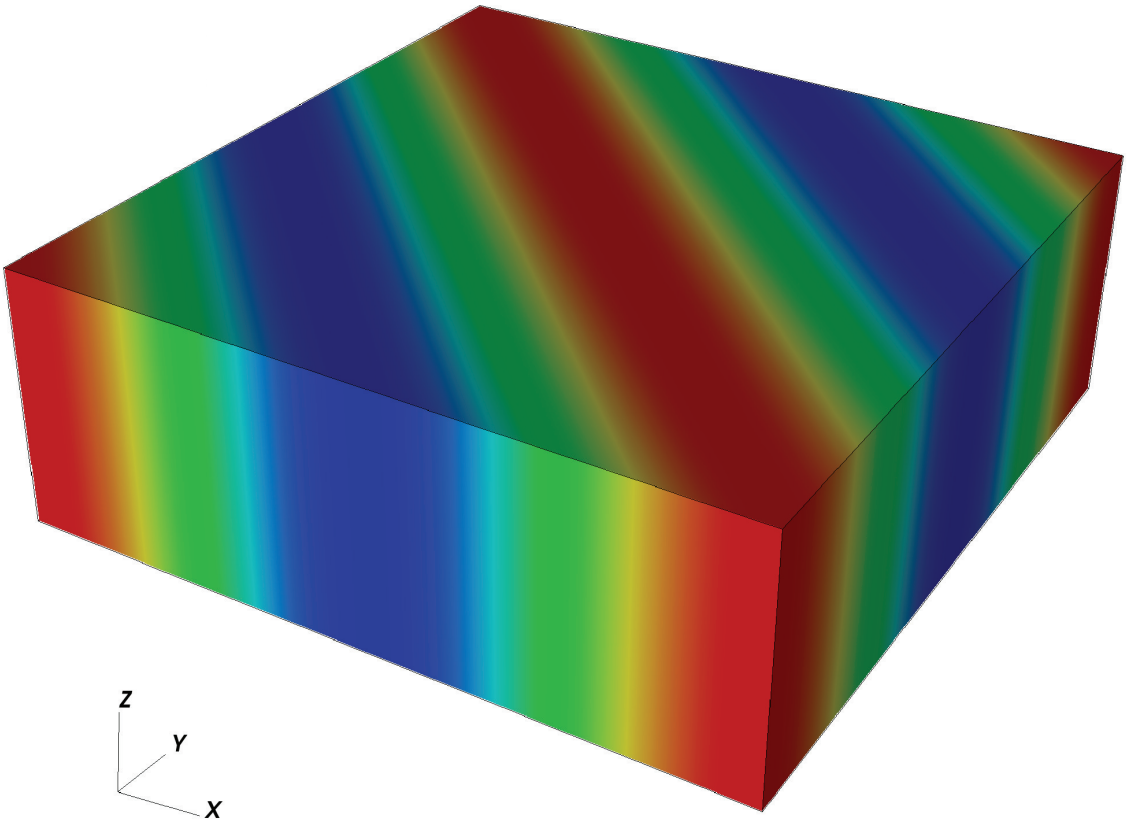


Figure 5.1: Pseudocolor plot of v_y magnitudes in a doubly periodic box ($x - y$ plane) with $l_x = l_y = 1/\sqrt{2}$, $k_x = k_y = 2\pi\sqrt{2}$, and $\hat{\delta} = 1 \times 10^{-3}$.

Table 5.1: Dissipative MHD wave percent error in decay rate after $t = 10T$ for three different dissipation parameter configurations. A grid resolution of $nx = 4$, $ny = 4$, and $nz = 1$ with a polynomial degree of $np = 5$ is used in all cases, and a time step $\Delta t = \omega_{Re}/100$.

	μ	η	κ	ω_{Re}	ω_{Im}	Analytic	Numeric	% Error
Fast	0.00e-0	0.00e-0	0.00e-0	19.08	0.00e+0	1.00e+0	9.954e-1	4.624e-1
Slow	0.00e-0	0.00e-0	0.00e-0	7.556	0.00e+0	1.00e+0	9.998e-1	2.020e-2
Fast	0.00e-0	1.00e-2	0.00e-0	19.06	-2.58e-1	4.27e-1	4.265e-1	4.582e-2
Slow	0.00e-0	1.00e-2	0.00e-0	7.544	-5.31e-1	1.20e-2	1.204e-2	5.203e-1
Fast	1.00e-2	1.00e-2	1.00e-2	19.06	-1.19e+0	1.97e-2	1.981e-2	4.213e-1
Slow	1.00e-2	1.00e-2	1.00e-2	7.551	-1.39e+0	1.00e-5	1.000e-5	2.926e+0

the dimensions are $l_x = l_y = 1/\sqrt{2}$ with a wavenumber $k_x = k_y = 2\pi\sqrt{2}$, which leads to a 45^o propagation of the wave across the $x - y$ plane.

When changing the domain size, wave number $\vec{k}$, or the magnitude of the dissipative parameters (μ, η, ν, and κ), the resulting eigensystem will change. Hyperresistivity is excluded from this study since the solutions are smooth and are numerically stable. For the domain size and wave propagation shown in Figure 5.1, the slow and fast magnetosonic waves are studied with varying dissipative parameters. For several cases, it is shown that the correct wave speed and the correct wave decay rates are calculated numerically in HiFi. Table 5.1 summarizes some of the results. The table shows the percent error in decay rate when compared to the analytically expected value, for both the slow and fast magnetosonic waves with three different dissipation parameter configurations. The first case is for ideal MHD with all the dissipation parameters set to zero. In the ideal case, the eigensystem is all real, and analytically there should be no wave decay. There is some numerical dissipation, and thus the percent error is not zero for the ideal MHD case. The second case has only nonzero resistivity, and the last has nonzero viscosity, resistivity, and heat conduction. For all cases a resolution of $nx = 4$, $ny = 4$, and $nz = 1$ with a polynomial degree of $np = 5$ is used.

The comparison made between the expected analytical values of decay and the com-

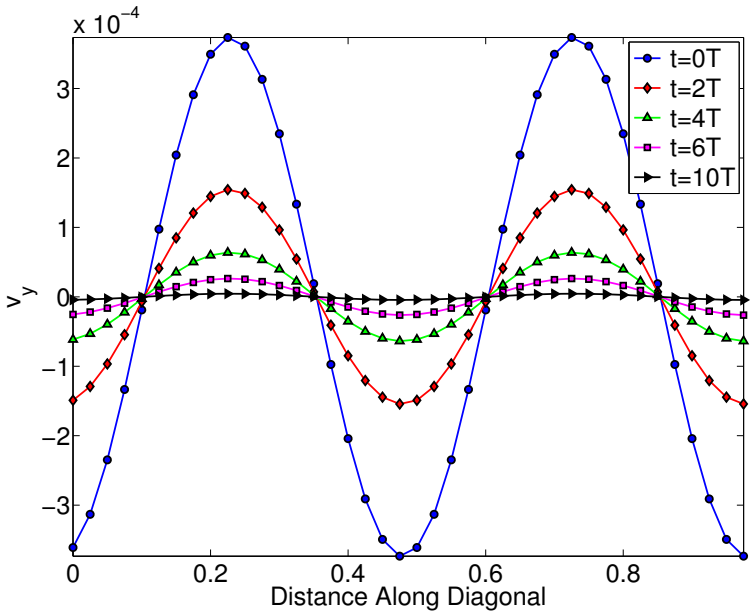


Figure 5.2: One-dimensional slice of v_y at $t = 0$ and after 2,4,6, and 10 wave periods. This case has $\eta = 1 \times 10^{-2}$ and all other dissipative parameters equal to zero.

puted numerical results were made by using the using the equation for amplitude

$$\delta_{analytic} = e^{-(t\frac{\omega_{Im}}{\omega_{Re}}k_x l_x)} \tag{5.6}$$

where ω_{Re} and ω_{Im} are the real and imaginary eigenvalues.

With dissipation, the eigensystem becomes complex, having both complex eigenvalues and eigenvectors. This means that the wave speeds are modified by the dissipative parameters. To demonstrate that the correct wave speeds are found, a slice of the solution is plotted in time. Figure 5.2 shows the initial wave form at $t = 0T$ and after 2,4,6, and 10 periods for the slow magnetosonic wave with resistivity $\eta = 1 \times 10^{-2}$ and other dissipative parameters equal to zero. It can be seen that the wave periods match as they have the same positions after whole periods. The wave decay can be seen with the different lines representing the solution after some number of periods. The peak amplitude values correspond to what is found in Table 5.1

5.2 Dynamics of a Nonlinear Spheromak Tilt Instability

The dynamics of a spheromak tilt instability is chosen due to its nonlinear nature. By studying the tilt mode instability it can verify the nonlinear aspects of the equation module and HiFi solver are working properly. The same dissipative MHD equation system (Equation 5.1) are used, with a spheromak initial condition.

5.2.1 Spheromak Initial Conditions

The spheromak initial condition in a cylindrical flux conserver for magnetic field intensity is the solution to a Bessel function, which in cylindrical coordinates looks like

$$\begin{aligned} B_r &= -\frac{\pi}{l_z} J_1(\alpha r)\cos(\pi z/l_z), \\ B_\phi &= -\lambda J_1(\alpha r)\sin(\pi z/l_z), \\ B_z &= \alpha J_0(\alpha r)\sin(\pi z/l_z), \end{aligned} \tag{5.7}$$

where J_0 and J_1 are the Bessel functions, and λ is an eigenvalue for a particular Taylor state. In this case $\lambda = \sqrt{\alpha^2 + (\pi/l_z)^2}$, and $\alpha = 3.8317$. Additionally the initial density and pressure are constant, ρv_x and ρv_y are zero, and ρv_z has a perturbation of the form

$$\rho v_z = \delta r \cos(\phi)\sin(\pi z/l_z) \tag{5.8}$$

where δ is the magnitude of the perturbation. This perturbation gives the initial spheromak an upward velocity parallel to the cylinder axis on one side, and a downward velocity on the other side. This velocity perturbation profile initiates the growth of the tilt mode instability.

Since the system of equations is written using a vector potential formulation, and the initial Taylor state Bessel function solution is in terms of magnetic field intensity, the spheromak initial condition in terms of vector potential first had to be solved. This is done by solving

$$\nabla \times \vec{B} = \lambda \vec{B} \tag{5.9}$$

with $\vec{B} = \nabla \times \vec{A}$. A Coulomb gauge $\nabla \cdot \vec{A} = 0$ is chosen and the equation is simplified to

$$\nabla^2 \vec{A} = -\lambda \vec{B} \tag{5.10}$$

where $\vec{B}$ is known from Equation 5.7. Equation 5.10 is solved for the initial vector potential $\vec{A}$ conditions.

5.2.2 Spheromak Boundary Conditions

The velocity boundary conditions are chosen such that velocity has perfect slip along the flux conserver walls. Zero normal velocity and zero normal gradient of the tangential velocity are

$$\begin{aligned} v_{\hat{n}} &= 0 \\ \nabla_{\hat{n}} v_t &= 0. \end{aligned} \tag{5.11}$$

Perfect conductor boundary conditions are applied to the vector potential, and thus the vector potential must be divergence free, and the tangential component of vector potential must be static in time:

$$\begin{aligned} \nabla \cdot \vec{A} &= 0, \\ \frac{\partial A_t}{\partial t} &= -E_t = 0, \end{aligned} \tag{5.12}$$

where E_t is the tangential component of electric field. Additionally mass density and current density have "natural" boundary conditions, where the flux at the boundary is determined by the interior equations, rather than explicitly specifying a separate boundary equation. Temperature has a zero normal gradient $\nabla_{\hat{n}} T = 0$.

5.2.3 Spheromak Relaxing to Taylor State Lambda

A first check for the nonlinear spheromak tilt mode dynamics is a qualitative check. When plotting the $\vec{B}$ field over time, it can be seen that the spheromak rotates from

its initial condition to a configuration rotated approximately 90^o counterclockwise (direction is determined by the perturbation) and remains stable in this position. Figure 5.3 shows snapshots of the tilt mode evolution. This is the expected behavior, since the Taylor minimum energy state predicts this configuration.

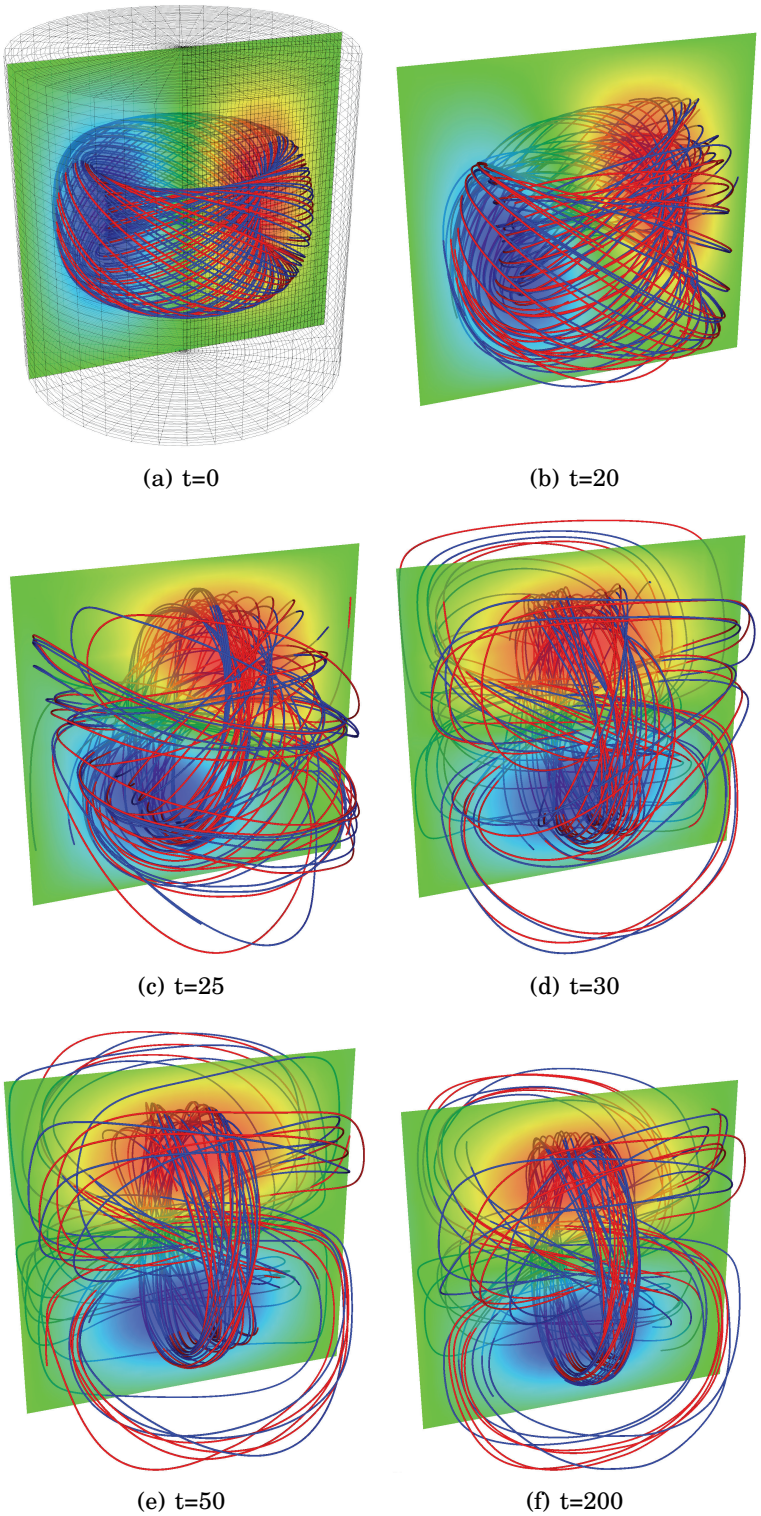
Another simple check for the spheromak tilt mode is tracking the kinetic energy. Figure 5.4 plots the kinetic energy over time for various values of hyper-resistivity. In the beginning of the trace, a small increase in kinetic energy can be seen due to the perturbation. This small amount of kinetic energy is about 1000 times smaller in magnitude then the initial magnetic energy, but is enough to initiate the growth of the instability. It can be seen that the kinetic energy quickly increases to a peak much larger than the perturbation magnitude, then falls at a similar rate as the increase, and finally settles to a near zero kinetic energy.

Similarly a plot of the magnetic energy is shown in Figure 5.5 for different hyper-resistivity values, although here the magnetic energy decays throughout the whole tilt mode dynamics. Since magnetic energy is converted to kinetic energy during the growth of this instability it makes sense it's decaying throughout the time. Higher values of hyper-resistivity also increase the rate of magnetic energy decay, and this can be seen in this plot.

A more quantitative analysis is also performed to show the correct Taylor minimum energy state is achieved with the spheromak tilt mode simulation. The ratio of magnetic energy to magnetic helicity λ, is tracked throughout the instability. This ratio is defined by

$$< \lambda >= \frac{\int \vec{B} \cdot \vec{B}\, dV}{\int \vec{A} \cdot \vec{B}\, dV} = \frac{\text{Magnetic Energy}}{\text{Magnetic Helicity}}, \tag{5.13}$$

where $\vec{B}$ is the magnetic field intensity, and $\vec{A}$ is the magnetic vector potential. The time trace of this ratio for different hyper-resistivity values is shown in Figure 5.6. It can be seen that the initial $\lambda = \sqrt{\alpha^2 + (\pi/l_z)^2} = 4.141$ given by the initial condition described above, and relaxes to the approximate theoretical Taylor state of $\lambda_{taylor} = 3.978$. The exception is when $\nu = 5 \times 10^{-5}$ and the value of λ relaxes more rapidly. This is most likely due to an excessive amount of dissipation of the magnetic energy.



(a) t=0 (b) t=20

(c) t=25 (d) t=30

(e) t=50 (f) t=200

Figure 5.3: Spheromak during a nonlinear tilt instability at $t = 0, 20, 25, 30, 50$ and 200 seconds (units are not normalized). Length to radius ratio is, $L/R = 2.0$, pressure is held fixed to zero $p = 0$, and density is fixed at unity $\rho = 1.0$. Dissipative parameters are: $\mu = 5 \times 10^{-2}$, $\eta = 0$, $\nu = 1 \times 10^{-5}$, and $\kappa = 1 \times 10^{-1}$.

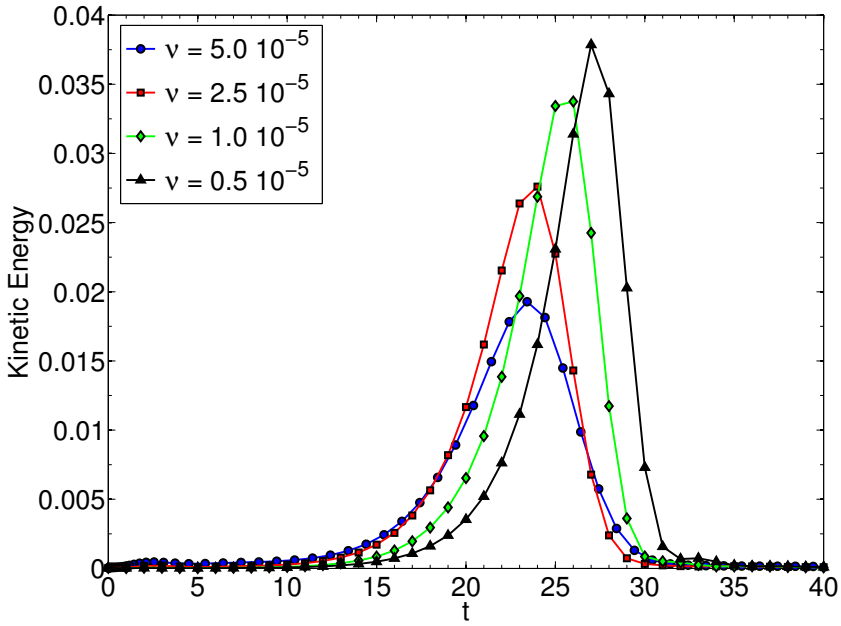


Figure 5.4: Kinetic energy over time during a spheromak tilt mode, with different hyper-resistivity values $\nu = 0.5 \times 10^{-5}, 1.0 \times 10^{-5}, 2.5 \times 10^{-5}$, and 5.0×10^{-5}.

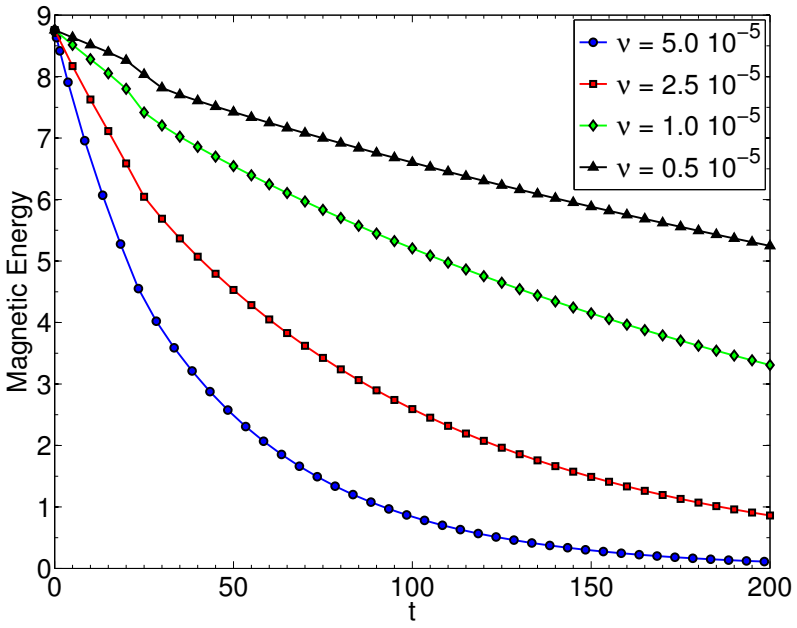


Figure 5.5: Magnetic energy over time during a spheromak tilt mode, with different hyper-resistivity values $\nu = 0.5 \times 10^{-5}, 1.0 \times 10^{-5}, 2.5 \times 10^{-5}$, and 5.0×10^{-5}.

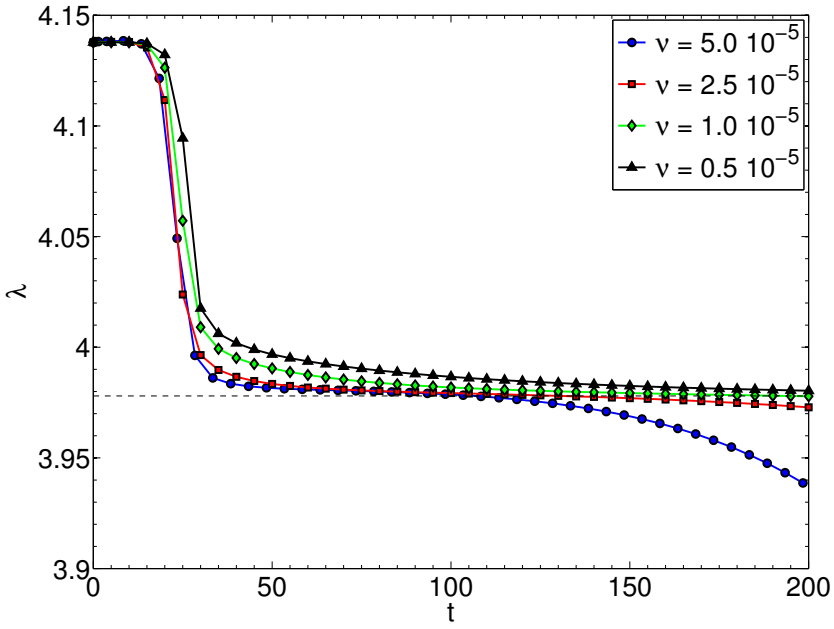


Figure 5.6: Ratio of magnetic energy to magnetic helicity (λ) integrated over time during a spheromak tilt mode, with different hyper-resistivity values $\nu = 0.5 \times 10^{-5}, 1.0 \times 10^{-5}, 2.5 \times 10^{-5}$, and 5.0×10^{-5}. It can be seen that value of lambda relaxes to the Taylor state minimum $\lambda = 3.978$ (dashed line).

Chapter 6

A PRIORI MESH QUALITY ERROR ANALYSIS FOR DEFORMED MESHES

6.1 Mesh Deformation Study Introduction

Mesh quality metrics have been used widely to quantify the amount and character of distortion of computational meshes [36, 37, 38]. Many metrics do not show a strong global correlation to solution accuracy, mainly because of the variety of problems that have to be considered and complexity of correlating them to solution error [37, 38, 39, 40]. Nevertheless, they can be used to find approximate relationships between mesh quality and solution error or to identify regions of a mesh that need repair. Here, several different mesh quality metrics are explored with the goal of finding a "universal metric" that can serve as a guide in predicting solution error magnitudes for different systems of equations. This capability enables a quantifiable measure of whether or not a mesh is of acceptable quality for a particular set of equations.

Little work has been performed on quantifying mesh quality for use in high-order finite element simulations. Most of the research to date has either been with finite volume methods [41, 42, 43] or linear finite element methods [37, 38, 39, 44, 40], though more recently quadratic finite elements have also been investigated [45]. In the finite volume case, a truncation error estimate based on the finite volume spatial discretization is commonly used. For the finite element method, the stiffness matrix condition number has been analyzed [46, 47, 48], where the stiffness matrix is formed by considering gradients of the linear finite element basis functions. This method naturally extends to high-order finite elements. More recent work [49] has examined a nodal spectral stiffness matrix for finite elements. In the research presented here, the spectral element stiffness matrix is calculated and used as one of the mesh quality metrics.

The approach chosen in this work is to study elemental grid deformations that impact the shape of a single high order finite element or the interaction between two neighboring high order finite elements. In doing so, the effects of such deformations on the differential operators are isolated. Deformations are chosen that are commonly encountered when numerically solving partial differential equations (PDEs). In the following sections, the global effects of the individual mesh distortions on solution quality are explored for known analytic problems by independently varying both the degree of

the distortions and the order of the finite elements. Several previously published mesh quality metrics are compared to the measured global solution error generated from high order finite element spatial discretizations.

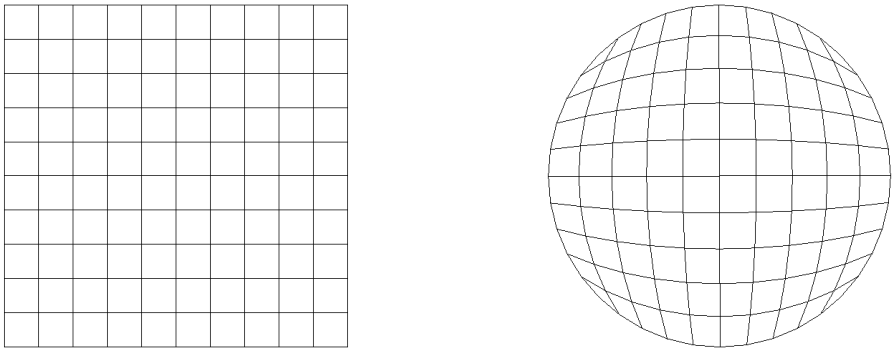
In the following analysis, the HiFi/SEL [3, 34, 4] high-order finite element code is used as described in Chapter 4. The code uses a flux-source formulation, which makes specifying nearly any particular set of partial differential equations straightforward. This enables analysis of various PDE systems while using the same formulation and meshes. HiFi is three-dimensional (3D) and SEL is two-dimensional (2D); their formulation is identical. The first part of the analysis is limited to the 2D code, and subsequently extended to the 3D code.

Section 6.2 outlines several distortion classifications: stretch, shear and skew. For each distortion type, a Jacobian is calculated to understand when a particular deformed mesh becomes degenerate. A degenerate element is defined as when the Jacobian is less then or equal to zero. A degenerate element is problematic for finding solutions on a mesh because the Jacobian multiplies the weak form integral (see Equation 4.9) and creates singularities.

Candidate mesh quality metrics are described in Section 6.3. These metrics are used to analyze each of the different distortion types. A representative mesh is analyzed with the different metrics and then compared to solutions of three equation sets (linear advection, Poisson's equation, and linearized MHD). These test problems have known analytic solutions. Section 6.4 outlines the three test problems, and Section 6.5 presents the analysis of the mesh metrics and the test problem solutions. The analysis is extended to a 3D formulation and described in Section 6.6, and results described in Section 6.7

6.2 Grid Distortions from a Logical Square

The logical to physical transformation introduces distortions, and often in order to achieve the desired physical shapes, the distortions can be severe. Mapping the logical square to a physical circle is an extreme example of this, where "the corners" of the



(a) Square Mesh (b) Circular Mesh

Figure 6.1: An example 10×10 square mesh 6.1(a) and a circular mesh 6.1(b), where the circular mesh has a square to circle mesh transformation.

circle have large and singular angle distortions. Figure 6.1 shows an example square mesh distorted into a circular shape. Here, in order to capture most of the basic distortion types, the distortions are classified into *stretch*, *shear*, *skew*, *large angle*, and *small edge*. This is not intended to be a complete set of distortions but provides a set of typical elemental grid deformations that are commonly encountered in real meshes. The goal of this study is to evaluate solution error due to such elemental distortions of single high order elements (e.g. shear) and interactions between two neighboring differently distorted high order elements (e.g. skew).

Definitions of these different distortions, the logical to physical mapping, and the grid Jacobians for each type are shown in the following sections.

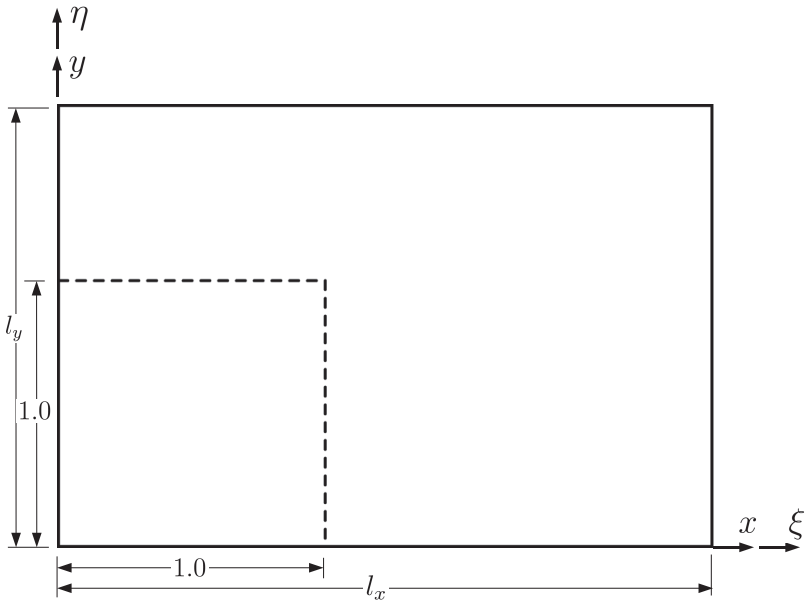


Figure 6.2: A single element with stretch deformation magnitudes of l_x and l_y. The dashed line represents the undeformed logical element, and the solid line represents the deformed element.

6.2.1 Stretch

Stretching is the simplest mapping with

$$x(\xi, \eta) = l_x \xi \tag{6.1}$$

$$y(\xi, \eta) = l_y \eta \tag{6.2}$$

where l_x and l_y are the respective stretching factors for the x and y dimensions (see Figure 6.2). The values of l_x and l_y do not have to be equal, and situations could arise where either l_x or l_y is much larger then the other. The resulting Jacobian is

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = l_x l_y. \tag{6.3}$$

It can be seen that the element approaches degeneracy as l_x or l_y approaches 0.

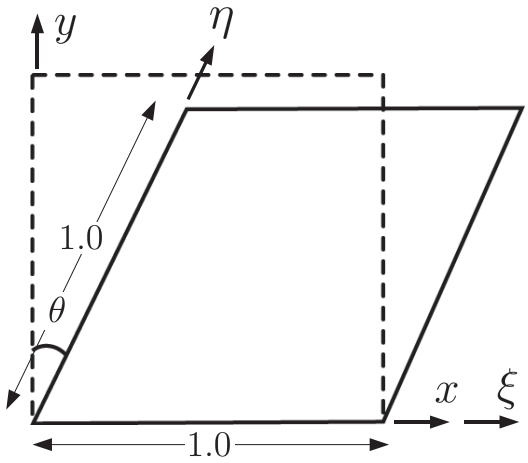


Figure 6.3: A single element with a shear angle of θ. The dashed line represents the undeformed logical element, and the solid line represents the deformed element.

6.2.2 Shear

The shear mapping can be defined as

$$x(\xi,\eta) = \xi + \eta \sin\theta \tag{6.4}$$

$$y(\xi,\eta) = \eta \cos\theta \tag{6.5}$$

where θ is the shear angle for the element (see Figure 6.3). The Jacobian is

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = \cos\theta. \tag{6.6}$$

In this case when $\theta \to \pi/2$, $J \to 0$ and the grid is considered degenerate. When the shear angle approaches $\pi/2$, the quadrilateral element has both small angles as well as obtuse angles, which can be problematic.

6.2.3 Skew

Skew involves an interaction between elements and cannot be classified in the same way as the stretch, and shear transformations. The skew can be seen as two adjacent

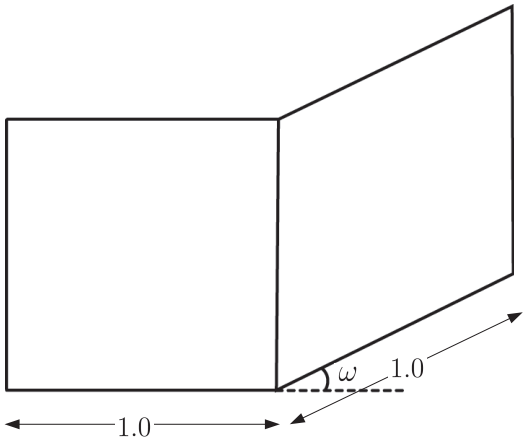


Figure 6.4: Adjacent elements with a skew angle ω.

elements with an angle ω between them (see Figure 6.4). In the case shown in Figure 6.4, the skewed mesh has a rectangular element adjacent to an element with a shear deformation. Each element has a separate Jacobian defined. The rectangular element has no deformation and the Jacobian is $J = 1$. The adjacent element is a sheared element with a transformation:

$$x(\xi, \eta) = \xi \cos\omega \tag{6.7}$$

$$y(\xi, \eta) = \eta + \xi \sin\omega, \tag{6.8}$$

and the Jacobian is,

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = \cos\omega. \tag{6.9}$$

The important distinguishing feature of the skew case is that an undeformed element is adjacent to a sheared element, and there is some skew interaction between the two elements.

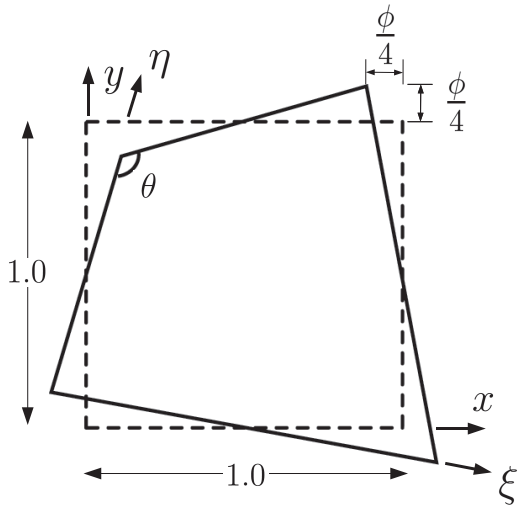


Figure 6.5: A single element with a large angle deformation θ.

6.2.4 Large Angle

The large angle deformation mapping can be defined as

$$x(\xi, \eta) = \xi + m_x \eta + b_x \tag{6.10}$$

$$y(\xi, \eta) = \eta + m_y \xi + b_y \tag{6.11}$$

where

$$m_x = (1 - 2\xi)\frac{\phi}{2} \tag{6.12}$$

$$m_y = -(1 - 2\eta)\frac{\phi}{2} \tag{6.13}$$

$$b_x = -(1 - 2\xi)\frac{\phi}{4} \tag{6.14}$$

$$b_y = (1 - 2\eta)\frac{\phi}{4}. \tag{6.15}$$

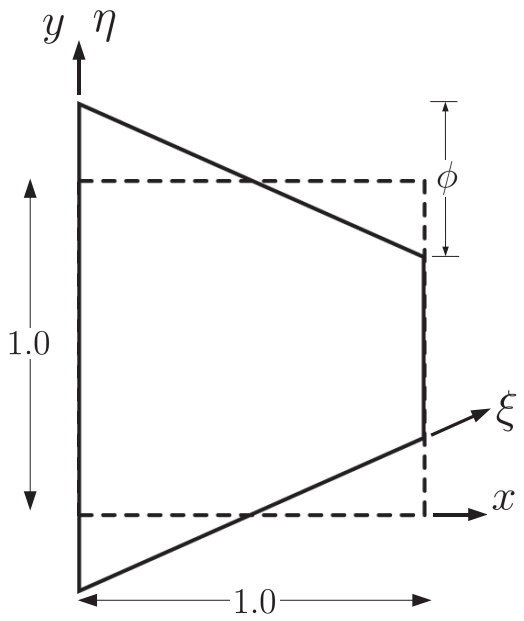


Figure 6.6: A single element with a small edge deformation.

The deformation parameter ϕ is in the range $[0, 1]$, where 0 is no deformation, and 1 is deformed such that the angle θ in Figure 6.5 is $180°$. The Jacobian is

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = 1 + \phi(\xi - \eta). \tag{6.16}$$

In this case when $\phi = 0$, $J = 1$, which is the undeformed case. When $\phi = 1$ ($\theta = 180°$), then $J = 1 + \xi - \eta$. This Jacobian does not necessary imply that the element will be degenerate when the angle has opened up to $180°$.

6.2.5 Small Edge

The small edge deformation mapping is defined as

$$x(\xi, \eta) = \xi \tag{6.17}$$

$$y(\xi, \eta) = \eta + m_y\xi + b_y \tag{6.18}$$

where

$$m_y = (1 - 2\eta)\phi \tag{6.19}$$

$$b_y = -(1 - 2\eta)\frac{\phi}{2}. \tag{6.20}$$

The deformation parameter ϕ is in the range $[0, 1]$, where 0 is no deformation, and 1 is deformed such that one vertical edge has collapsed to a length of zero (Figure 6.6). The opposite edge increases its length by the same amount the shrinking edge loses length. The Jacobian is defined as

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = 1 - \phi(2\xi - 1), \tag{6.21}$$

and it can be seen that when $\phi = 1$, the Jacobian will be zero in the element, and thus it is degenerate.

6.3 Mesh Quality Metrics

In this study new mesh quality metrics are not defined, but instead previously published metrics are explored with the intent of correlating them to solution error in an *a priori* fashion. This does not mean that the metrics or the results of this study can only be used for *a priori* analysis, but that is the focus of the analysis presented here. (For instance, the same metrics could also be applied to *a posteriori* analysis of PDE solutions.)

A unique aspect of *a priori* analysis is that the mesh quality metrics themselves depend solely on the mesh geometry and are independent of the equations being solved and of the solution itself. Here, the correlation between the solution error and the grid metrics is estimated by using equations that are representative of a physical problem. The fact that the metrics are independent of the representative equations makes correlation to solution error a challenge because there are no assumptions about the physical problem at hand. At the same time, they can potentially be useful because the analysis can be applied before any numerical simulations are performed.

Three types of mesh quality metrics are explored. A set of metrics that deals with element to element interaction based on Kallinderis and Kontzialis [42] is the first type described in Section 6.3.1. A set of metrics based on an algebraic framework using the mesh Jacobian from Knupp [39] is explored in Section 6.3.2. The third type is based on the spectral element stiffness matrix condition number and described in Section 6.3.3.

6.3.1 Kallinderis / Kontzialis Metrics

The mesh metrics described in Kallinderis and Kontzialis [42] are based on analytic forms of the truncation error for the gradient operator. The error E is defined in terms of some variable u on the mesh,

$$E = \nabla^h u - \nabla^a u \tag{6.22}$$

with the superscript h and a denoting the numerical and analytic gradients, respectively. The error is expanded in a Taylor series and can be grouped into different order terms. The coefficients for each order are used to calculate the metrics. For instance in 2D the first order terms are defined as

$$e^x_{xy} = \frac{1}{2S} \sum_{e=1}^{n} \left(\Delta x_{e,1} \Delta y_{e,1} + \Delta x_{e,2} \Delta y_{e,2} \right) \Delta y_e, \tag{6.23}$$

$$e^x_{xx} = \frac{1}{2S2!} \sum_{e=1}^{n} \left(\left(\Delta x_{e,1} \right)^2 + \left(\Delta x_{e,2} \right)^2 \right) \Delta y_e, \tag{6.24}$$

$$e^x_{yy} = \frac{1}{2S2!} \sum_{e=1}^{n} \left(\left(\Delta y_{e,1} \right)^2 + \left(\Delta y_{e,2} \right)^2 \right) \Delta y_e, \tag{6.25}$$

where

$$\left(\Delta x_{e,k} \right)^j \left(\Delta y_{e,k} \right)^l = \frac{1}{N_{e,k}} \sum_{m=1}^{N_{e,k}} \left(x_m|_{e,k} - x_0 \right)^j \left(y_m|_{e,k} - y_0 \right)^l, \tag{6.26}$$

and $x_m|_{e,k} - x_0$, $y_m|_{e,k} - y_0$ are the coordinates offset from centroid, $N_{e,k}$ is total number of nodes associated with a dual vertex averaging, and S represents the area of the dual mesh. The subscript e represents the edge number and is defined on a dual mesh, which spans four elements in the structured quadrilateral mesh case. The spanning of

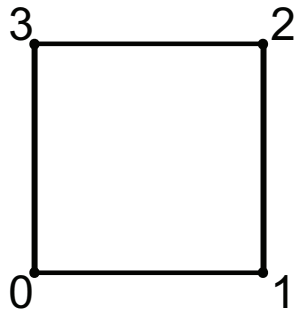


Figure 6.7: An example Knupp reference quadrilateral.

four elements is the basis for the metric quantifying element to element interactions. Both first and second order metrics are defined (in 2D) and are normalized using a length scale of the elements. Equations 6.27 and 6.28 show the first order metrics, where the error coefficients defined in equations 6.23-6.25 are normalized by a characteristic length scale. The characteristic length scales for a uniform structured mesh are simply the element edge lengths. The normalized metrics are denoted with a over bar.

$$Q_x = |\bar{e^x}_{xx}| + |\bar{e^x}_{yy}| + |\bar{e^x}_{xy}| \tag{6.27}$$

$$Q_y = |\bar{e^y}_{xx}| + |\bar{e^y}_{yy}| + |\bar{e^y}_{xy}| \tag{6.28}$$

More details about the the calculation of the metrics are in Kallinderis and Kontzialis [42].

6.3.2 Knupp Metrics

Knupp's metrics are defined by an algebraic framework described in [39], and in more detail in [37]. They are based upon the mesh Jacobian A_k for linear finite elements. The metric is a local (element wise) metric compared against some reference quadrilateral (see Figure 6.7). The indexing k is counterclockwise around the element, and is modulo three, such that if $k = 2$, then $k + 1$ becomes 0, and $k + 2$ becomes 1 [39]. The

Jacobian is defined as

$$A_k = \begin{pmatrix} x_{k+1} - x_k & x_{k+3} - x_k \\ y_{k+1} - y_k & y_{k+3} - y_k \end{pmatrix}. \tag{6.29}$$

From this Jacobian A_k, four 'metric' tensor matrices are defined

$$\lambda_{ij}^k = (A_k^T A_k)_{ij} \tag{6.30}$$

where the superscript T denotes transpose, and the λ_{ij}^k denotes the ijth element of kth metric tensor. Additionally, the parameter α_k is defined as the determinant of the Jacobian matrix,

$$\alpha_k = det(A_k) \tag{6.31}$$

and τ_k as

$$\tau_k = (\alpha_k + \alpha_{k+2})/2w \tag{6.32}$$

where w is the total area of a reference quadrilateral, and the undeformed element area is used as the reference quadrilateral in the subsequent analysis. From these four parameters one can define three different mesh quality metrics.

A size metric is defined as:

$$f_{size} = \min(\tau_k, 1/\tau_k) \tag{6.33}$$

which compares the element of interest's area to the area w of a reference quadrilateral element. This metric has the property of being 1 if and only if the quadrilateral has the same total area as the reference quadrilateral, and the metric is 0 if and only if the quadrilateral has a total area of zero [39].

The second metric is a shape metric

$$f_{shape} = \frac{8}{\sum_{k=0}^{3} \left(\lambda_{11}^k + \lambda_{22}^k\right) / (\alpha_k)} \tag{6.34}$$

which compares the shape of an element to a square reference element. It has the properties of being 1 if and only if the quadrilateral is a square, and 0 if the quadrilateral

is degenerate. It also has the property of being scale invariant [39].

The third metric is called the skew metric and is defined as

$$f_{skew} = \frac{4}{\sum_{k=0}^{3} \left(\sqrt{\lambda_{11}^{k} + \lambda_{22}^{k}} \right) / \left(\alpha_k \right)} \tag{6.35}$$

which detects distortions based on angles within the element. It has a rectangular reference element and has the properties of being 1 if and only if the quadrilateral is a rectangle, and 0 if and only if the quadrilateral is degenerate. The metric is also scale invariant and independent of element edge length ratios [39]. The name of the skew metric should not be confused with the skew deformation as defined in section 6.2.3. The metric measures angles in an element which could arise from either a *shear* or *skew* deformation.

A unique feature of the Knupp metrics is their range from 1 (undeformed) to 0 (degenerate), which allows for a combined metric by forming their product. The range of 1 to 0 remains and only the metrics that detect deformation will contribute to the product. The products of the size and shape metrics, as well as the the size and skew metrics, are useful because they do not require a choice of which metric to use for different types of distortions. In the subsequent analysis these products are used as metrics. The shape and skew metrics are not combined together because they both account for the skew deformation.

6.3.3 Spectral Stiffness Matrix

The stiffness matrix condition number has previously been used to estimate the quality of a mesh. Zgainski et al. [48], as well as Tsukerman [47] explore properties of the finite element stiffness matrix in order to estimate solution accuracy. Zgainski specifically looks at the stiffness matrix condition number, while Tsukerman looks at the eigenvalues of the stiffness matrix. As noted by Tsukerman, this is also related to the "maximum angle" condition studied by Babuška and Aziz [44].

Here, the spectral element stiffness matrix is used as a metric. The matrix is de-

fined globally (for the whole domain) and the eigenvalues then correspond to the problem area in the mesh. The method for creating the stiffness matrix and calculating its condition number is described.

Matrix Construction

The 2D gradient operator is defined in terms of the logical coordinates as:

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} \\ \frac{\partial \xi}{\partial y}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta} \end{bmatrix}. \tag{6.36}$$

Using the mesh metrics from equation 4.11, the gradient operator becomes

$$\nabla = \begin{bmatrix} \frac{1}{J}\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{1}{J}\frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta} \\ -\frac{1}{J}\frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi} + \frac{1}{J}\frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta} \end{bmatrix}. \tag{6.37}$$

With the definition of the gradient operator from Equation 6.37, the elements of the 2D stiffness matrix K are,

$$K^{ij} = \int_\Omega \nabla\alpha^i(\xi,\eta)\cdot\nabla\alpha^j(\xi,\eta)dxdy = \int_\Omega \nabla\alpha^i(\xi,\eta)\cdot\nabla\alpha^j(\xi,\eta)Jd\xi d\eta, \tag{6.38}$$

where α^i and α^j are the 2D basis functions, and the integrand can be expanded as

$$\begin{aligned} &\nabla\alpha^i(\xi,\eta)\cdot\nabla\alpha^j(\xi,\eta) = \\ &\frac{\partial\xi}{\partial x}\frac{\partial\xi}{\partial x}\frac{\partial\alpha^i}{\partial\xi}\frac{\partial\alpha^j}{\partial\xi} + \frac{\partial\eta}{\partial x}\frac{\partial\eta}{\partial x}\frac{\partial\alpha^i}{\partial\eta}\frac{\partial\alpha^j}{\partial\eta} + \frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial x}\frac{\partial\alpha^i}{\partial\xi}\frac{\partial\alpha^j}{\partial\eta} + \frac{\partial\eta}{\partial x}\frac{\partial\xi}{\partial x}\frac{\partial\alpha^i}{\partial\eta}\frac{\partial\alpha^j}{\partial\xi} \\ +&\frac{\partial\xi}{\partial y}\frac{\partial\xi}{\partial y}\frac{\partial\alpha^i}{\partial\xi}\frac{\partial\alpha^j}{\partial\xi} + \frac{\partial\eta}{\partial y}\frac{\partial\eta}{\partial y}\frac{\partial\alpha^i}{\partial\eta}\frac{\partial\alpha^j}{\partial\eta} + \frac{\partial\eta}{\partial y}\frac{\partial\xi}{\partial y}\frac{\partial\alpha^i}{\partial\eta}\frac{\partial\alpha^j}{\partial\xi} + \frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial y}\frac{\partial\alpha^i}{\partial\xi}\frac{\partial\alpha^j}{\partial\eta}. \end{aligned} \tag{6.39}$$

The integral is numerically approximated using Gaussian quadrature,

$$\begin{aligned} K^{ij} &= \int_\Omega \left[\nabla\alpha^i(\xi,\eta)\cdot\nabla\alpha^j(\xi,\eta)\right] Jd\xi d\eta = \\ &\sum_{k=1}^{nq}\sum_{l=1}^{nq} J_{kl}W_{kl}\left[\nabla\alpha^i(\xi_k,\eta_l)\cdot\nabla\alpha^j(\xi_k,\eta_l)\right], \end{aligned} \tag{6.40}$$

where nq is the the number of quadrature points along each dimension, and W_{kl} is an array of weights at the quadrature points.

The full stiffness matrix can be constructed within the HiFi/SEL framework since all the derivatives of the logical variables and derivatives of the basis functions are known in the code.

Condition Number Calculation

The condition number of a matrix is defined as the ratio of the magnitude of its largest eigenvalue to its smallest eigenvalue, $|e_{max}|/|e_{min}|$. In the case of the global stiffness matrix, there is a null eigenvalue and therefore the smallest nonzero eigenvalue is used. These eigenvalues are calculated in parallel using the SLEPc (Scalable Library for Eigenvalue Problem Computations)[50] solver package.

The condition number generally increases with resolution and with increasing spectral polynomial order, and must be normalized. A simple normalization by the total number of degrees of freedom ($nx \times ny \times np^2$) is sufficient for meaningful results, where nx and ny are the number of elements in the x and y directions respectively and np is the polynomial order of the basis function.

6.4 2D Test Problems

In order to correlate the mesh quality metrics with solution error, three test problems are chosen. The advection equation is chosen to quantify mesh distortions on the ∇ operator, and Poisson's equation is chosen for the ∇^2 operator. These operators are some of the most common differential operators to be encountered in PDEs in fluid and plasma physics, and are therefore chosen to give the best indication of whether the mesh quality metrics could possibly predict solution error for practical problems of interest. Linearized MHD is also chosen to quantify solution error due to mesh distortions for a more complex system of equations. All three test problems have simple analytical solutions, and therefore error norms could easily be calculated and compared to the different mesh quality metrics.

6.4.1 Meshes for Testing Deformation

Several model meshes (Figure 6.8) are chosen for capturing the effects of different mesh distortions. Each mesh is intended to capture the effects of one of the grid distortions types described in Section 6.2. The meshes are chosen such that the distortion is repeated over the whole domain in a tiled or tessellated pattern. This allows the solution error due to each particular distortion to be studied separately, while the solution is evenly impacted by the distortion. It also allows for an arbitrarily large mesh at any resolution without impacting the distortion shape, and thus the interesting solution can be isolated from any solution boundary effects.

In order to eliminate boundary effects for both the advection equation and the linearized MHD equations, a doubly periodic domain is chosen. Additionally, in the advection case, the density advected is near zero at the boundary using a Gaussian concentration in the domain centroid. Likewise, in the case of the Poisson equation, the source is centered in the domain and falls off to a near zero value at the boundary. Dirichlet boundary conditions are used to solve the Poisson equation.

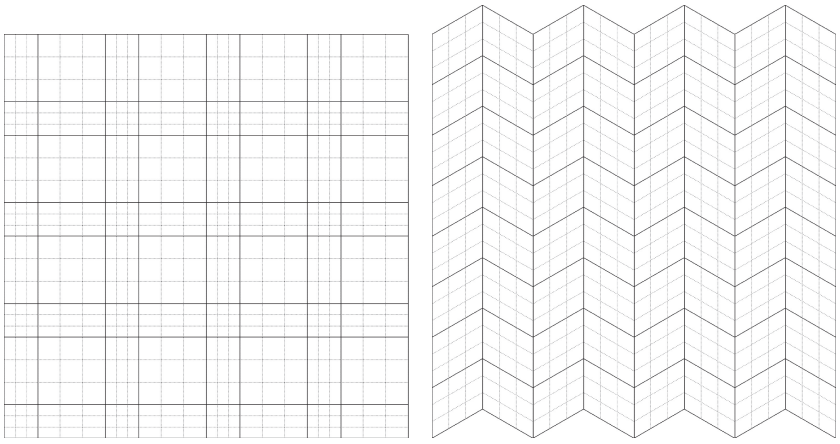
6.4.2 Advection Equation

A 2D advection equation is solved:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{c}\phi) = 0 \tag{6.41}$$

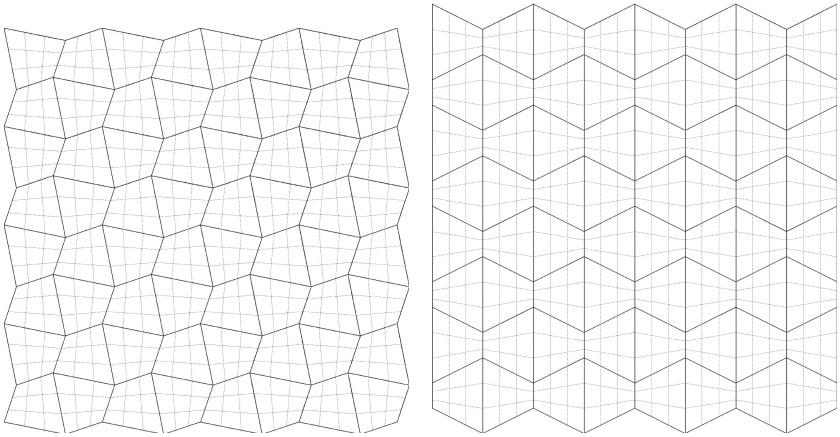
where ϕ is the Gaussian "density concentration" scalar, and $\vec{c} = [c_x, c_y]$ is the advection "wave" speed vector. The solution is the same Gaussian shape translated across the domain at speed and direction $\vec{c}$.

The Gaussian density concentration is translated the same total distance for each mesh size. Additionally the time step is chosen to be small enough such that the temporal accuracy has a negligible effect when compared to the spatial accuracy. In the cases presented here a time step $\Delta t = 10^{-5}$ with 1000 steps and $c = 1$ is chosen such that the Gaussian is translated a distance $x = 0.01$ in nominal domain size of



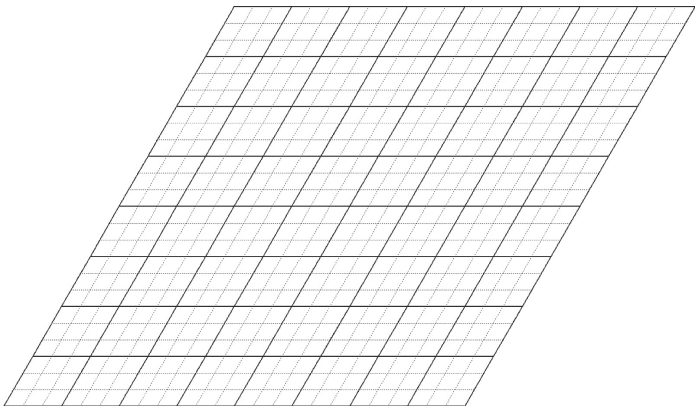
(a) Stretch Deformation

(b) Skew Deformation



(c) Large Angle Deformation

(d) Small Edge Deformation



(e) Shear Deformation

Figure 6.8: Sample meshes for testing different distortion types. Grid resolution is $nx = ny = 8$ with $np = 3$. The solid lines represent the element boundaries and the dashed lines represent the interior distribution within the elements.

1.0 by 1.0. The choice of the direction $\vec{c}$ depended on the mesh type. For all the test meshes the propagation is chosen to be in the x-direction, except for the sheared mesh, which is chosen in the y-direction. These propagation directions are chosen such that the deformation is "experienced" by the Gaussian. For the sheared mesh case the y-direction is chosen based on how the shear deformation is applied. In the stretched mesh case either direction is acceptable.

6.4.3 Poisson's Equation

A 2D Poisson's equation is solved:

$$\nabla^2 \phi = S \tag{6.42}$$

where ϕ is a scalar field, and S is a source forcing function. In this case S is chosen such that the solution is a Gaussian:

$$S = -4A\left[((x-x_0)^2 + (y-y_0)^2)/r_c^4\right] e^{-((x-x_0)^2+(y-y_0)^2)/r_c^2} \tag{6.43}$$

where A is the amplitude of the Gaussian, x_0 and y_0 are the center coordinates, and r_c characterizes the width of the Gaussian. The source is initialized in the center (centroid) of the mesh, such that the resulting solution is a Gaussian also at the centroid.

6.4.4 Linearized MHD Equations

A linearized MHD system of equations is solved:

$$\begin{aligned} \rho\frac{\partial \mathbf{v}}{\partial t} &= \mathbf{j}\times\mathbf{B} - \nabla p, \quad \mathbf{j} = \nabla\times\mathbf{b} \\ \frac{\partial \mathbf{b}}{\partial t} &= \nabla\times(\mathbf{v}\times\mathbf{B}), \quad \nabla\cdot\mathbf{b} = 0 \\ \frac{\partial p}{\partial t} &+ \gamma P\nabla\cdot\mathbf{v} = 0, \end{aligned} \tag{6.44}$$

where $\mathbf{B} \equiv B\hat{n}$, $\mathbf{b}$ is the magnetic field perturbation, and p is the total pressure. The form of a linear perturbation is: $\mathbf{v}(\mathbf{x},t) = \mathbf{v}_0 \exp[i(\mathbf{k}\cdot\mathbf{x} - \omega t)]$. The eigensystem of these equations is solved and used to initialize the simulation for a particular MHD

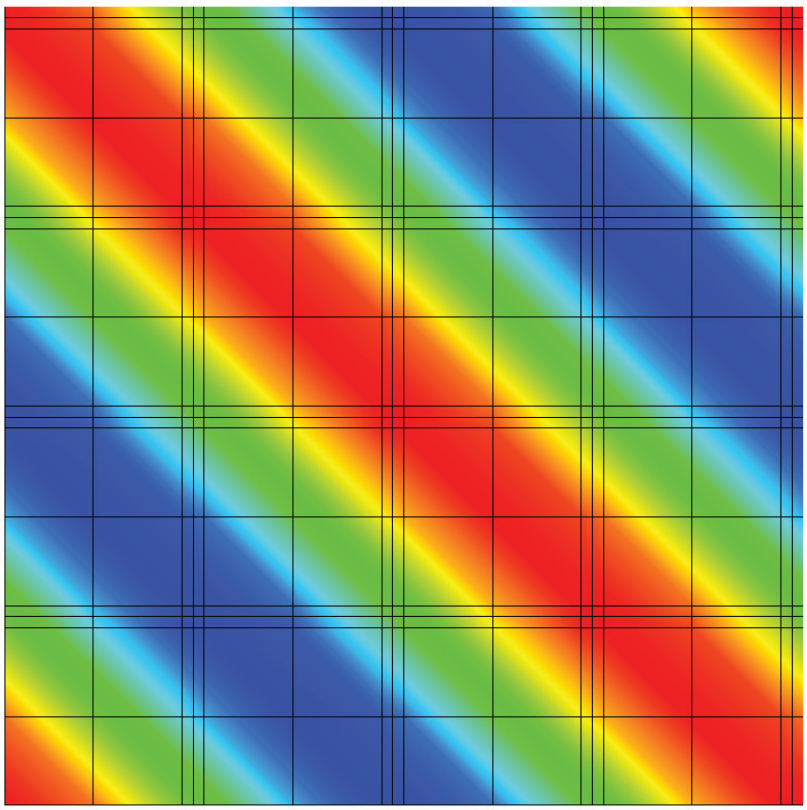
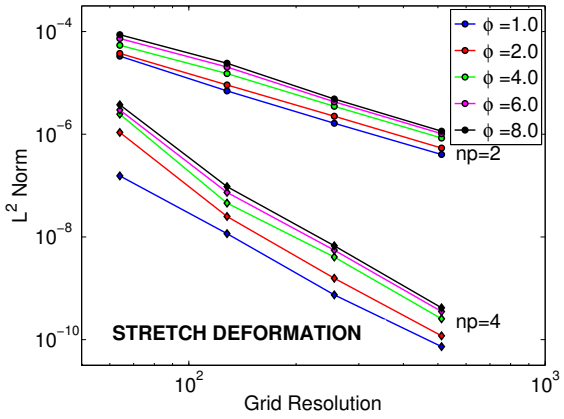
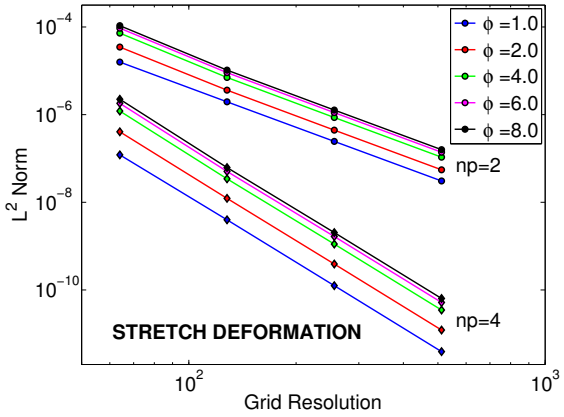


Figure 6.9: Example linearized MHD initialization in a doubly periodic domain with a stretch deformation.

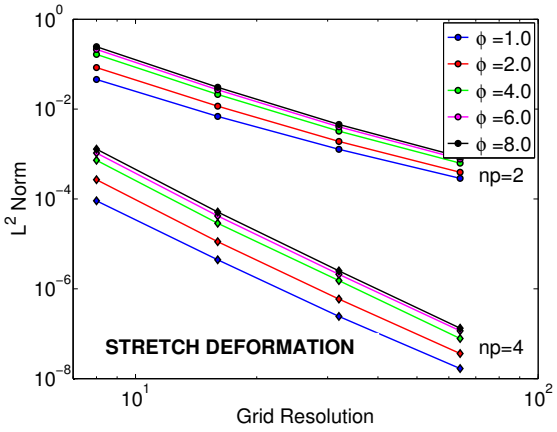
wave. Using this technique a pure wave is initialized with a known frequency and amplitude. In this study the slow magnetosonic wave is initialized and propagated in a doubly periodic domain at a 45 degree angle with respect to a uniform magnetic field aligned in the $\hat{x}$-direction. The solution is known because it is simply the translation of this wave in the domain propagated in the $\hat{k}$-direction. Figure 6.9 shows an example wave in a mesh with a stretch deformation.



(a) Advection Equation

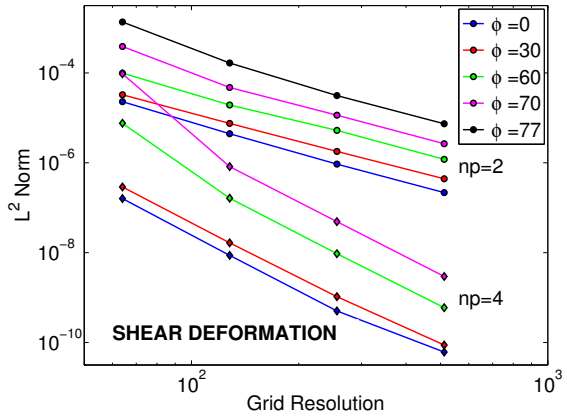


(b) Poisson's Equation

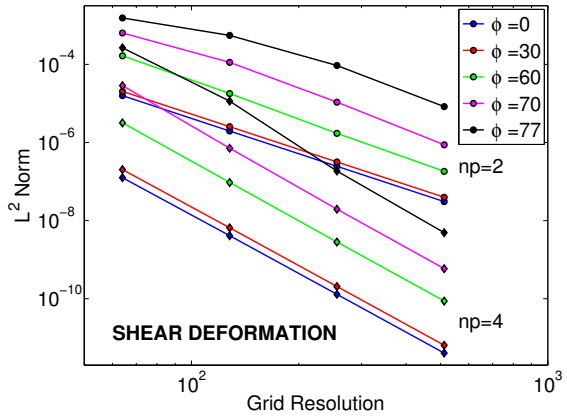


(c) Linearized MHD Equations

Figure 6.10: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) L^2 norm versus grid resolution for varying degrees of stretch and for both $np = 2$ and $np = 4$.

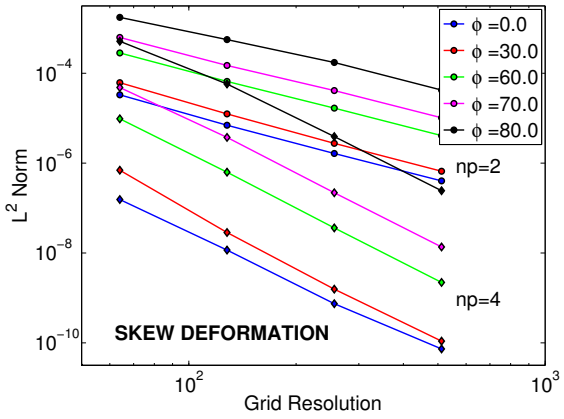


(a) Advection Equation

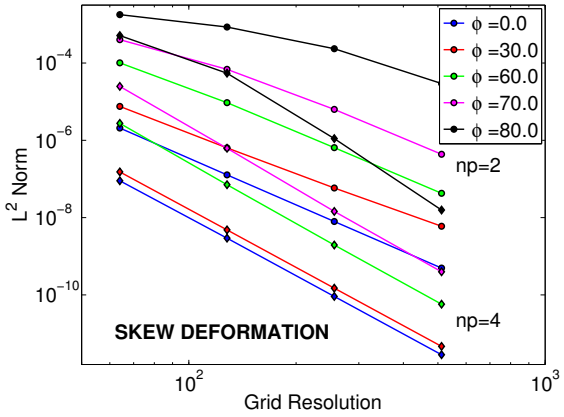


(b) Poisson's Equation

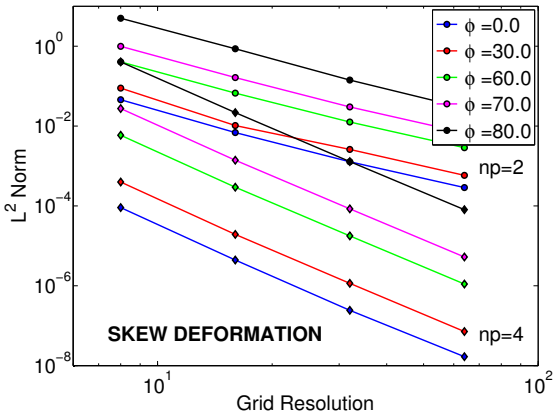
Figure 6.11: Advection equation (a) and Poisson's equation (b) L^2 norm versus grid resolution for varying shear angles and for both $np = 2$ and $np = 4$.



(a) Advection Equation

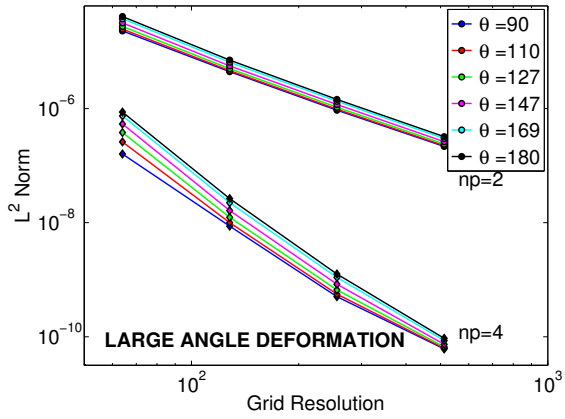


(b) Poisson's Equation

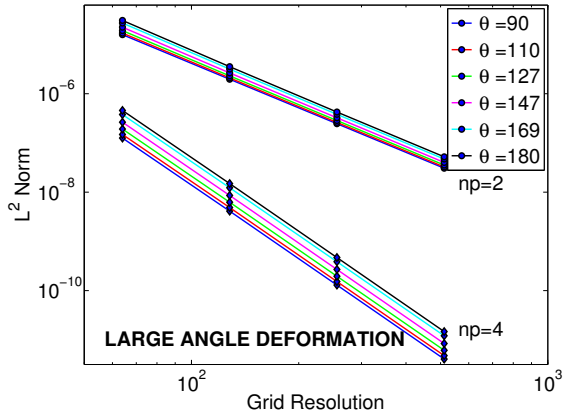


(c) Linearized MHD Equations

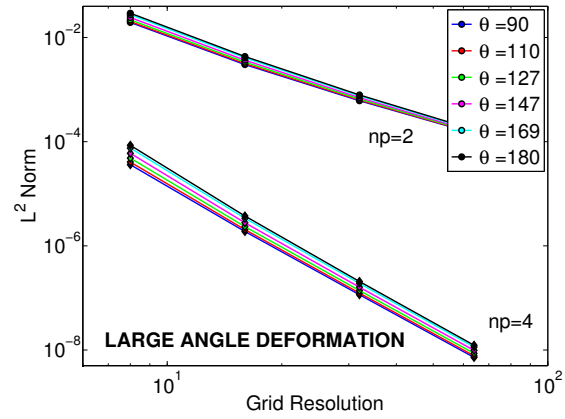
Figure 6.12: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) L^2 norm versus grid resolution for varying skew angles and for both $np = 2$ and $np = 4$.



(a) Advection Equation

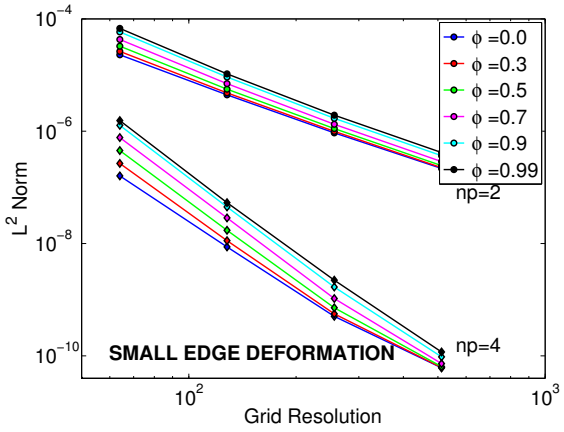


(b) Poisson's Equation

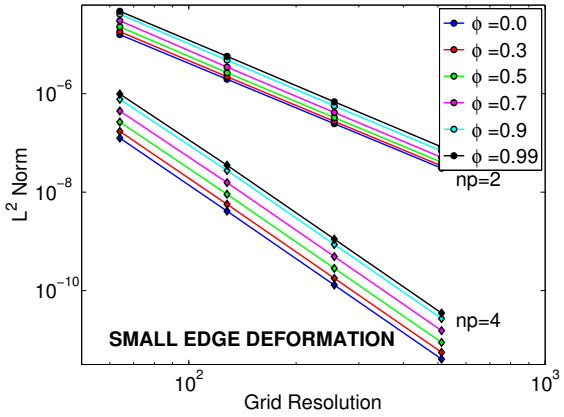


(c) Linearized MHD Equations

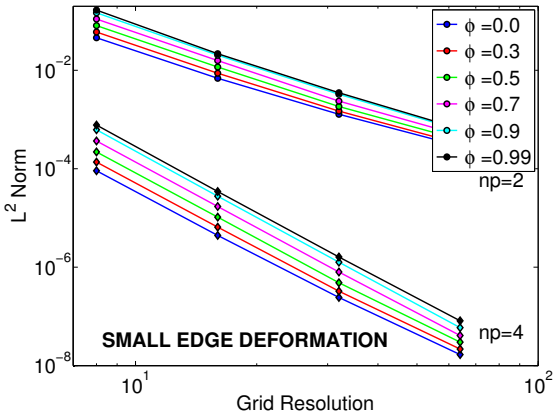
Figure 6.13: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) L^2 norm versus grid resolution for varying deformation degrees (large angle from 90 to 180 degrees) and for both $np = 2$ and $np = 4$.



(a) Advection Equation



(b) Poisson's Equation



(c) Linearized MHD Equations

Figure 6.14: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) L^2 norm versus grid resolution for varying deformation degrees (small edge from length 1 to length 0) and for both $np = 2$ and $np = 4$.

6.5 Results from 2D Analysis

6.5.1 Order Accuracy with Varying Deformations

The three test problems are solved for each of the test meshes (Figures 6.10-6.14) with varying degree of deformation, varying spatial resolution, as well as for different polynomial orders of the basis function np of 2 (quadratic) and 4 (quartic). Boundary effects are minimized by using periodic domains and limiting the solution to near the domain centroid (advection and Poisson's equations). Tiling the domain with uniformly distorted meshes ensures the distortion affects all points of the solution and eliminates any boundary effects.

A normalized time-step $\Delta t = 10^{-5}$, which is much smaller than the characteristic time-scales of the test problems, is used to prevent the temporal order of the solver (second order Crank-Nicholson) impacting the solution accuracy. Additionally, only smooth solutions are chosen, since C^0 finite elements inherently produce numerical oscillations when attempting to resolve non-smooth solutions.

The resulting solutions are compared to the analytical solution and an L^2 error norm is calculated. The error norm is computed for varying spatial resolution, order of the basis functions, and degree of grid deformation. Figures 6.10-6.14 show the L^2 error norm versus spatial resolution on a logarithmic scale for the different deformation types solving the advection equation, Poisson's equation, and the linearized MHD equations.

Looking at the L^2 error norm for each of the cases allows a baseline (no deformation) result to be compared to the formal accuracy of the method, as well as to determine how the different grid deformations deviate from the baseline calculation. For each case it is observed that increasing the level of mesh deformation does not reduce the order of the spatial accuracy of the method, but it does increase the magnitude of the error. Notice the lines plotted on a logarithmic scale in Figures 6.10-6.14 are generally parallel with the slope given by the polynomial order np, demonstrating the order does not change. The distortions do however scale the solution error L^2 error norm by some

factor d

$$\|E\|_2 = d\|E_0\|_2 = \mathscr{O}(\Delta x^n), \tag{6.45}$$

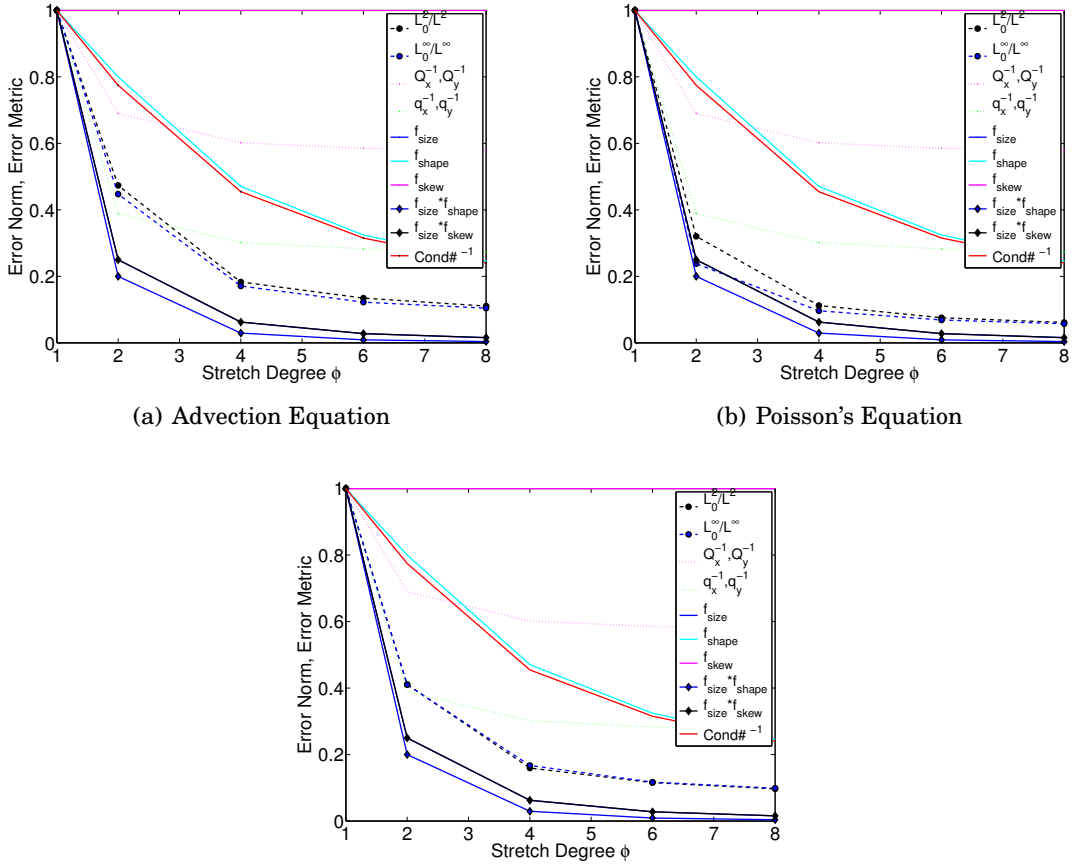
where $\|E_0\|_2$ is the L^2 error norm of the undistorted mesh. This result is consistent with theory and demonstrates that the convergence rates for different degrees of deformation preserve the order of spatial discretization, although the total error magnitude increases with higher degree of mesh deformation.

6.5.2 Solution Error and Mesh Quality Metrics

The mesh quality metrics depend solely on the mesh geometry and know nothing about the equations being solved or the solution. Here a correlation between the metrics and the solution error is investigated. The desired outcome is the determination of a metric that can be used as the indicator of a potential solution error, or at least to identify areas of the mesh that need improvement.

The error metrics defined in sections 6.3.1 and 6.3.2, as well as stiffness matrix condition number defined in section 6.3.3 are plotted along with the solution error norms calculated for test problems in Figures 6.15-6.19. The metrics as well as the inverse of the L^2 norm are normalized such that 1 is the undeformed case, and 0 is the degenerate case. Knupp's metrics are defined on the scale from 0 to 1, and thus do not require normalization. Taking advantage of this characteristic, the products of Knupp metrics is also plotted. Figures 6.15(a)-(b), 6.16(a)-(b), 6.17(a)-(b), 6.18(a)- (b), and 6.19(a)-(b) are for the advection equation and Poisson's equation on the stretched, sheared, skewed, large angle, and small edge meshes respectively, and figures 6.15(c), 6.17(c), 6.18(c) and 6.19(c) are for the linearized MHD equations. Each of these cases has a fixed resolution of 256×256 with $np = 4$. For each of the mesh types and for the three test problems it can be seen that the error norms follow the same trend as some of the error metrics.

In order to quantify the correlation between the solution error norms and the mesh deformation metrics, a χ^2 goodness of fit test statistic is calculated for the error norms compared to the expected mesh metric value. In this case the solution norm is the

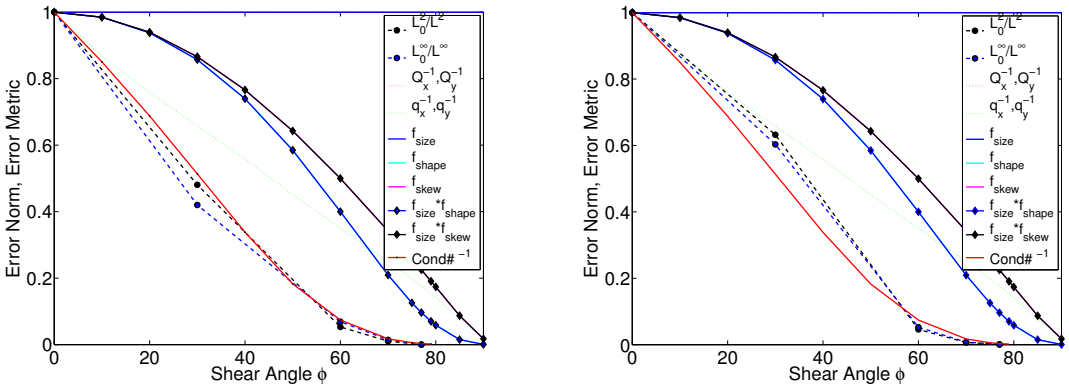


(a) Advection Equation

(b) Poisson's Equation

(c) Linearized MHD Equations

Figure 6.15: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) inverse error norms and mesh quality metrics (Section 6.3) for a stretch deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. Both the inverse L^2 norm and the inverse L^∞ norm are plotted. Q_x^{-1} and q_x^{-1} are in the 1st and 2nd order inverse Kallinderis/Kontzialis metrics, f_{size}, f_{shape}, and f_{skew} are the Knupp metrics. Products of the Knupp metrics are also included. Cond^{-1} is the inverse stiffness matrix condition number.



(a) Advection Equation (b) Poisson's Equation

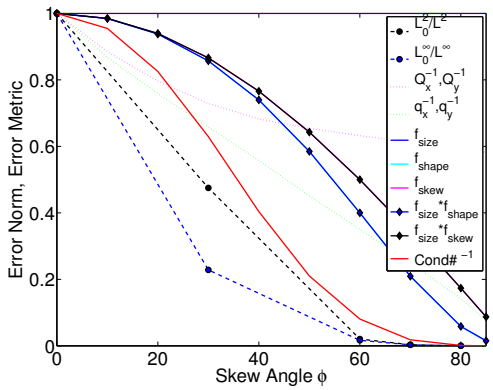
Figure 6.16: Advection equation (a) and Poisson's equation (b) inverse error norms and error metrics for a shear deformation. Error norms and mesh quality metrics (Section 6.3) are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. Both the inverse L^2 norm and the inverse L^∞ norm are plotted. Q_x^{-1} and q_x^{-1} are in the 1st and 2nd order inverse Kallinderis/Kontzialis metrics, f_{size}, f_{shape}, and f_{skew} are the Knupp metrics. Products of the Knupp metrics are also included. Cond^{-1} is the inverse stiffness matrix condition number.

observed quantity and the mesh deformation metric is the expected quantity. The goodness of fit test statistic is defined as

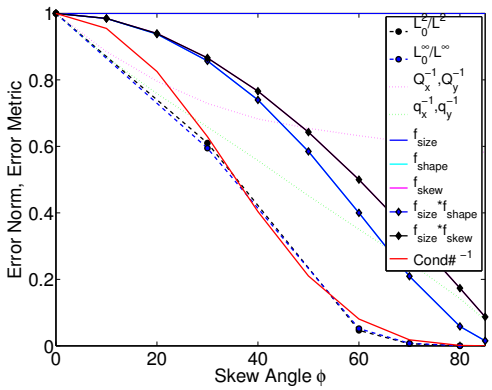
$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{6.46}$$

where O_i are the observed solution error norms, and E_i are the expected mesh deformation metric values. Table 6.1 has the test statistic values for each of the mesh metrics and different mesh deformation types for each of the test problems.

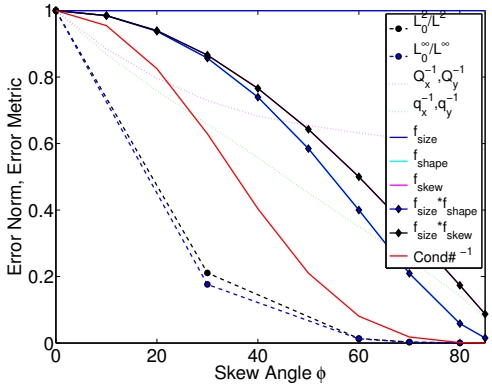
The closer the χ^2 values are to 0 the higher the probability that the expected value will be met. It is observed that for all but the skew deformation type the solution error norm has the highest probability of following the Knupp metric for all test problems. However, in the skew case the stiffness matrix condition number has the lowest χ^2 test statistic. In order to provide some context, when comparing the metrics to a problem whose error does not increase with increasing deformation (e.g. metric is constant at



(a) Advection Equation

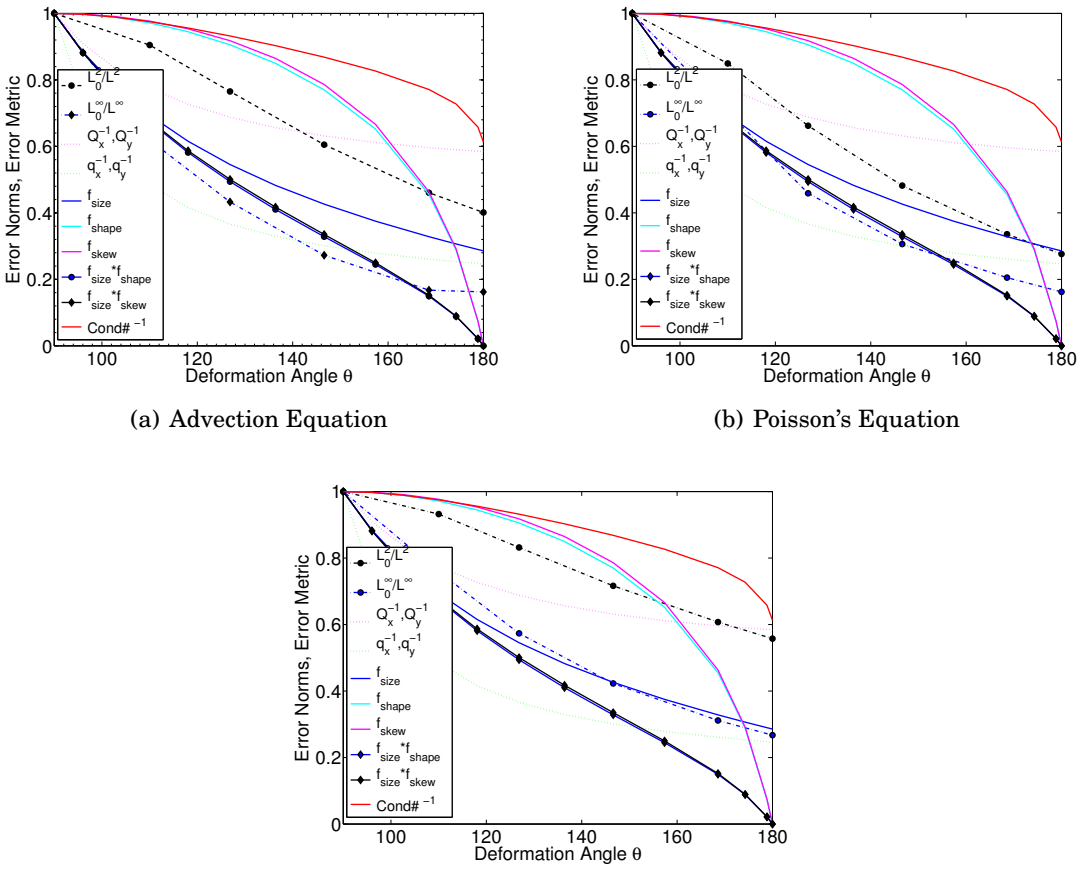


(b) Poisson's Equation



(c) Linearized MHD Equations

Figure 6.17: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) inverse error norms and mesh quality metrics (Section 6.3) for a skew deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. Both the inverse L^2 norm and the inverse L^∞ norm are plotted. Q_x^{-1} and q_x^{-1} are in the 1st and 2nd order inverse Kallinderis/Kontzialis metrics, f_{size}, f_{shape}, and f_{skew} are the Knupp metrics. Products of the Knupp metrics are also included. Cond^{-1} is the inverse stiffness matrix condition number.

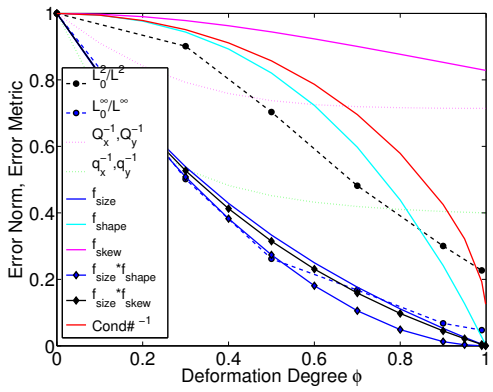


(a) Advection Equation

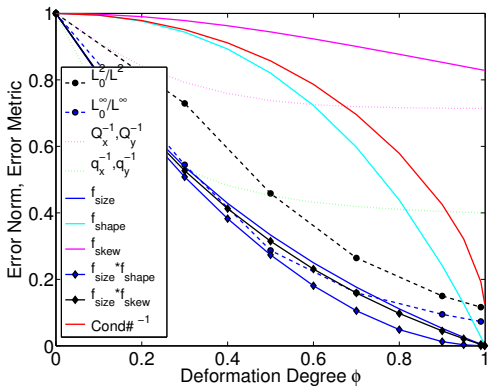
(b) Poisson's Equation

(c) Linearized MHD Equations

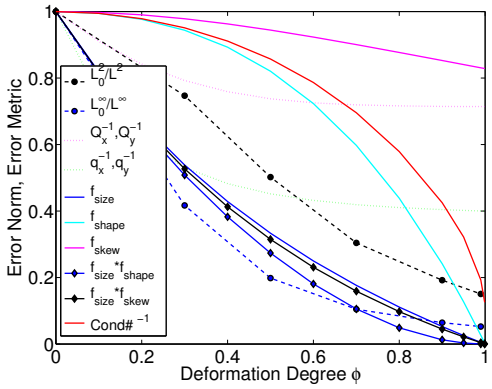
Figure 6.18: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) inverse error norms and mesh quality metrics (Section 6.3) for a large angle deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. Both the inverse L^2 norm and the inverse L^∞ norm are plotted. Q_x^{-1} and q_x^{-1} are in the 1st and 2nd order inverse Kallinderis/Kontzialis metrics, f_{size}, f_{shape}, and f_{skew} are the Knupp metrics. Products of the Knupp metrics are also included. Cond^{-1} is the inverse stiffness matrix condition number.



(a) Advection Equation



(b) Poisson's Equation



(c) Linearized MHD Equations

Figure 6.19: Advection equation (a), Poisson's equation (b), and Linearized MHD (c) inverse error norms and mesh quality metrics (Section 6.3) for a small edge deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. Both the inverse L^2 norm and the inverse L^∞ norm are plotted. Q_x^{-1} and q_x^{-1} are in the 1st and 2nd order inverse Kallinderis/Kontzialis metrics, f_{size}, f_{shape}, and f_{skew} are the Knupp metrics. Products of the Knupp metrics are also included. Cond^{-1} is the inverse stiffness matrix condition number.

Table 6.1: χ^2 goodness of fit test values for Poisson's equation, the advection equation, and linearized MHD equation's L^∞ and L^2 error norms compared to expected mesh metrics values, where q_x and Q_x are the Kallinderis/Kontzialis 2nd and 1st order metrics respectively, Knupp is the $f = f_{size} \cdot f_{skew}$ metric, and Cond No. is the stiffness matrix condition number.

Poisson		Stretch	Skew	Large Angle	Small Edge
q_x	L^∞	$3.390 \cdot 10^{-2}$	$3.150 \cdot 10^{-2}$	$1.810 \cdot 10^{-2}$	$6.160 \cdot 10^{-2}$
	L^2	$2.661 \cdot 10^{-2}$	$3.161 \cdot 10^{-2}$	$6.795 \cdot 10^{-2}$	$4.505 \cdot 10^{-2}$
Q_x	L^∞	$2.417 \cdot 10^{-1}$	$2.472 \cdot 10^{-1}$	$9.530 \cdot 10^{-2}$	$2.674 \cdot 10^{-1}$
	L^2	$2.168 \cdot 10^{-1}$	$2.480 \cdot 10^{-1}$	$3.618 \cdot 10^{-2}$	$1.868 \cdot 10^{-1}$
Knupp	L^∞	$4.453 \cdot 10^{-4}$	$5.990 \cdot 10^{-2}$	$9.916 \cdot 10^{-4}$	$8.319 \cdot 10^{-4}$
	L^2	$1.395 \cdot 10^{-3}$	$9.467 \cdot 10^{-2}$	$2.458 \cdot 10^{-2}$	$2.165 \cdot 10^{-2}$
Cond No.	L^∞	$1.572 \cdot 10^{-1}$	$5.875 \cdot 10^{-4}$	$2.038 \cdot 10^{-1}$	$2.050 \cdot 10^{-1}$
	L^2	$1.217 \cdot 10^{-1}$	$2.326 \cdot 10^{-4}$	$9.430 \cdot 10^{-2}$	$1.035 \cdot 10^{-1}$
Advection		Stretch	Skew	Large Angle	Small Edge
q_x	L^∞	$1.550 \cdot 10^{-2}$	$1.092 \cdot 10^{-1}$	$9.200 \cdot 10^{-3}$	$7.000 \cdot 10^{-2}$
	L^2	$1.472 \cdot 10^{-2}$	$4.790 \cdot 10^{-2}$	$1.182 \cdot 10^{-1}$	$6.018 \cdot 10^{-2}$
Q_x	L^∞	$1.621 \cdot 10^{-1}$	$3.224 \cdot 10^{-1}$	$1.132 \cdot 10^{-1}$	$2.884 \cdot 10^{-1}$
	L^2	$1.524 \cdot 10^{-1}$	$2.701 \cdot 10^{-1}$	$1.518 \cdot 10^{-2}$	$9.253 \cdot 10^{-2}$
Knupp	L^∞	$1.250 \cdot 10^{-2}$	$2.095 \cdot 10^{-1}$	$2.700 \cdot 10^{-3}$	$3.774 \cdot 10^{-4}$
	L^2	$1.300 \cdot 10^{-2}$	$1.359 \cdot 10^{-1}$	$5.899 \cdot 10^{-2}$	$1.005 \cdot 10^{-1}$
Cond No.	L^∞	$6.910 \cdot 10^{-2}$	$7.230 \cdot 10^{-2}$	$2.354 \cdot 10^{-1}$	$2.240 \cdot 10^{-1}$
	L^2	$6.010 \cdot 10^{-2}$	$1.087 \cdot 10^{-2}$	$4.242 \cdot 10^{-2}$	$1.837 \cdot 10^{-2}$
MHD		Stretch	Skew	Large Angle	Small Edge
q_x	L^∞	$1.560 \cdot 10^{-2}$	$1.289 \cdot 10^{-1}$	$3.610 \cdot 10^{-2}$	$8.690 \cdot 10^{-2}$
	L^2	$1.632 \cdot 10^{-2}$	$1.159 \cdot 10^{-1}$	$1.784 \cdot 10^{-1}$	$3.762 \cdot 10^{-2}$
Q_x	L^∞	$1.706 \cdot 10^{-1}$	$3.388 \cdot 10^{-1}$	$4.530 \cdot 10^{-2}$	$3.270 \cdot 10^{-1}$
	L^2	$1.729 \cdot 10^{-1}$	$3.283 \cdot 10^{-1}$	$1.210 \cdot 10^{-2}$	$1.587 \cdot 10^{-1}$
Knupp	L^∞	$9.200 \cdot 10^{-3}$	$2.402 \cdot 10^{-1}$	$8.100 \cdot 10^{-3}$	$4.300 \cdot 10^{-3}$
	L^2	$6.903 \cdot 10^{-3}$	$2.490 \cdot 10^{-1}$	$9.704 \cdot 10^{-2}$	$2.918 \cdot 10^{-2}$
Cond No.	L^∞	$8.030 \cdot 10^{-2}$	$9.230 \cdot 10^{-2}$	$1.263 \cdot 10^{-1}$	$2.831 \cdot 10^{-1}$
	L^2	$8.131 \cdot 10^{-2}$	$7.881 \cdot 10^{-2}$	$1.182 \cdot 10^{-2}$	$8.315 \cdot 10^{-2}$

1.0) the χ^2 value is approximately 0.5. Thus, relative to $\chi^2 = 0.5$, it is reasonable to assume that the metrics with $\chi^2 < 0.25$ provide statistically significant correlation to solution error.

6.6 Mesh Deformation Analysis Extension to 3D

The mesh deformation work described in Sections 6.2-6.5 is only a 2D analysis. It is necessary to extend the ability to calculate the different mesh metrics in 3D to be of practical use in HiFi. The two metrics that provided the best predictive capabilities in the 2D analysis, the Knupp metric and the stiffness matrix condition number, have fairly straightforward extensions to 3D. These two metrics are used in a similar 3D analysis.

The basic deformations of shear, stretch, skew translate easily to 3D meshes and are used in the analysis. Additionally meshes with a controllable amount of 'randomness' are also included. These 'random' meshes include some amount of all the basic deformations studied (stretch, shear, skew, and large and small angle). The amount of deformation is controlled by a parameter that when large enough yields degenerate elements. This gives meshes that are more complete in terms of the types of distortions encountered, and are more representative of meshes that may be encountered with complex geometries. Figure 6.20 shows an example mesh with a 'random' deformation.

6.6.1 Knupp Mesh Metrics in 3D

The 2D quadrilateral Knupp metrics described in Section 6.3.2, were based on Knupp [39]. Similarly the paper also describes the same metrics, size, shape, and skew, for hexahedrons. It is therefore a straightforward extension to include the metric calculations in 3D.

The three Knupp metrics in 3D are defined as:

$$f_{size} = \min(\tau, 1/\tau), \tag{6.47}$$

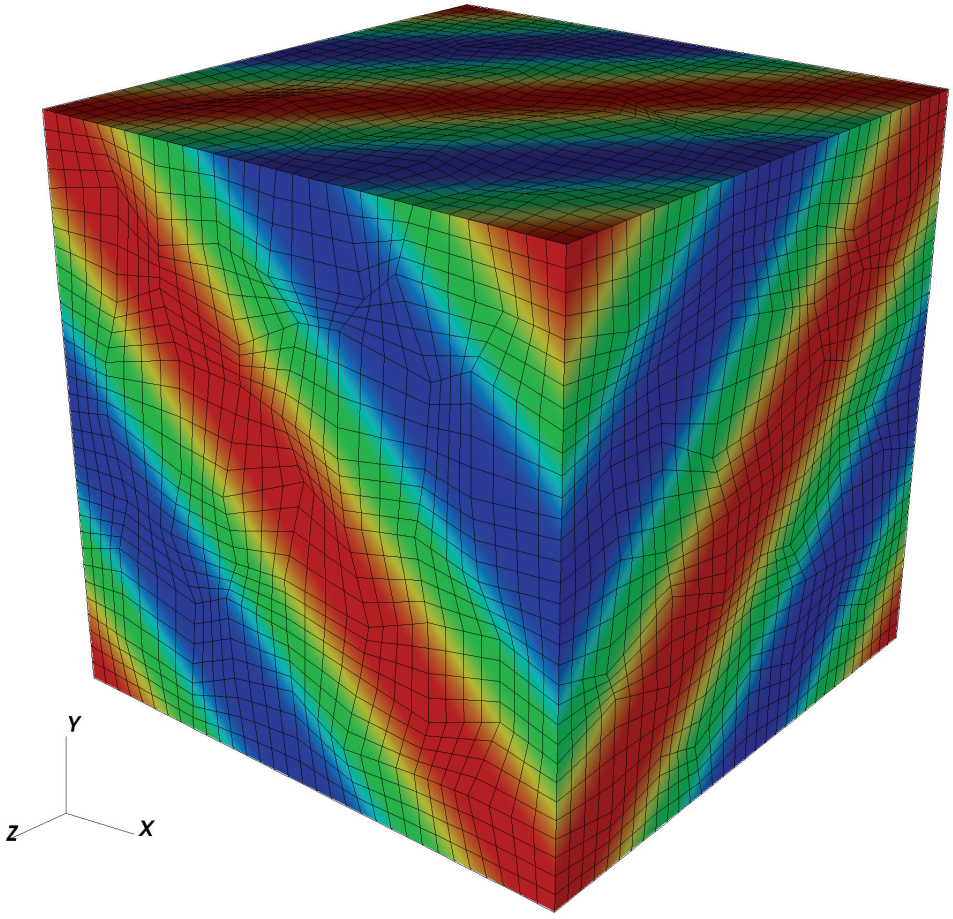


Figure 6.20: Example linearized MHD initialization in a triply periodic domain with a ‘random’ deformation.

$$f_{shape} = \frac{24}{\sum_{k=0}^{7} \left(\lambda_{11}^k + \lambda_{22}^k + \lambda_{33}^k\right) / \left(\alpha_k^{2/3}\right)}, \tag{6.48}$$

and

$$f_{skew} = \frac{8}{\left(\left(\sqrt{\lambda_{11}^k \lambda_{22}^k \lambda_{33}^k}\right) / \left(\alpha_k\right)\right)^{2/3}} \tag{6.49}$$

where τ, λ, and α are defined in a similar way as their 2D counterparts. Based on the 2D analysis, the product of these metrics was the best error indicator, so it makes sense in 3D to use the same products

$$f_{size} f_{shape}, \tag{6.50}$$

$$f_{size} f_{skew}. \tag{6.51}$$

6.6.2 3D Spectral Stiffness Matrix

Construction of a 3D spectral element stiffness matrix is a straightforward extension of the 2D formulation. The 2D matrix construction is described in Section 6.3.3. Similarly a 3D stiffness matrix K^{ij} is defined as:

$$K^{ij} = \int_{\Omega} \left[\nabla \alpha^i(\xi, \eta, \phi) \cdot \nabla \alpha^j(\xi, \eta, \phi)\right] J d\xi d\eta d\phi = \sum_{k=1}^{nq} \sum_{l=1}^{nq} \sum_{m=1}^{nq} J_{klm} W_{klm} \left[\nabla \alpha^i(\xi_k, \eta_l, \phi_m) \cdot \nabla \alpha^j(\xi_k, \eta_l, \phi_m)\right], \tag{6.52}$$

where nq is the number of quadrature points along each dimension, J_{klm} is the grid Jacobian, and W_{klm} is an array of weights at the quadrature points. Much like in the 2D case, the extreme eigenvalues of this matrix (for condition number calculation) will be found using the SLEPc [50] libraries.

6.6.3 3D Test Problem

A linearized MHD test problem is used in the 3D analysis. The test problem is analogous to the 2D test problem described in Section 6.4.4. A cube geometry is used with the vector $\mathbf{k}$ propagated along the cube diagonal; 45 degrees from all axes. Similarly,

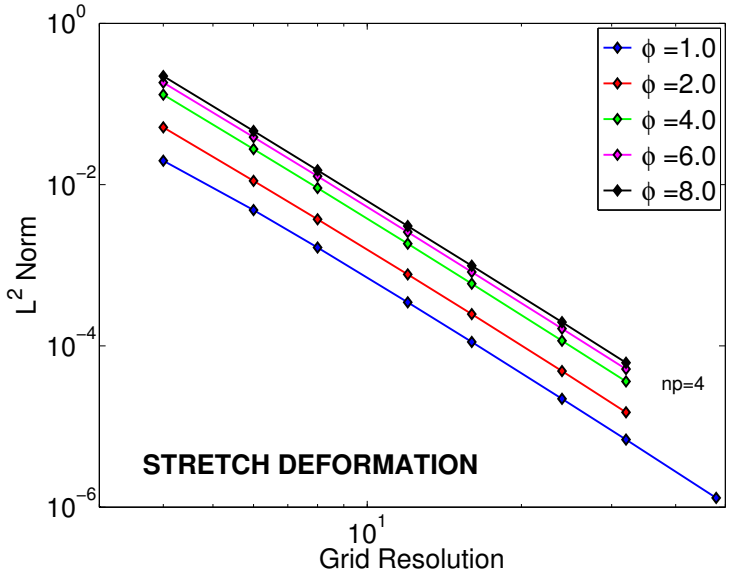


Figure 6.21: Linearized MHD L^2 norm versus grid resolution for varying degrees of stretch for $np = 4$.

a numerical eigensystem is solved and used to initialize the problem. This ensures a pure MHD wave is initialized and the solution is known as the propagation of this wave.

6.7 Results from 3D Analysis

The results from the extension of the metric analysis to 3D confirmed that similar properties exist, and the methodology can be applied in general to 3D meshes. The same spectral convergence is seen, and increase of total error as the mesh deformations increase. Additionally 'random' meshes are included to demonstrate that the specific type of deformation is not important, and may resemble more realistic deformations. Figures 6.21 - 6.23 show the L^2 error norm versus grid resolution for stretched, skewed, and 'random' meshes respectively. Figures 6.24 - 6.26 compare the error norms to various mesh deformation metrics.

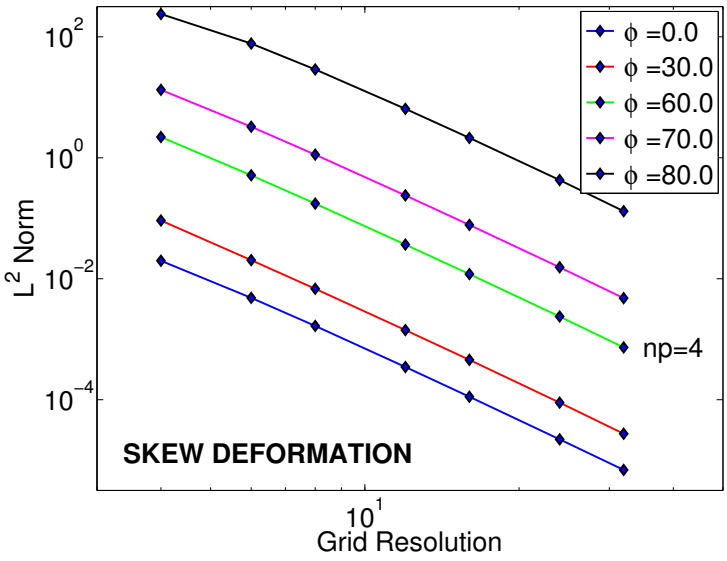


Figure 6.22: Linearized MHD L^2 norm versus grid resolution for varying degrees of skew for $np = 4$.

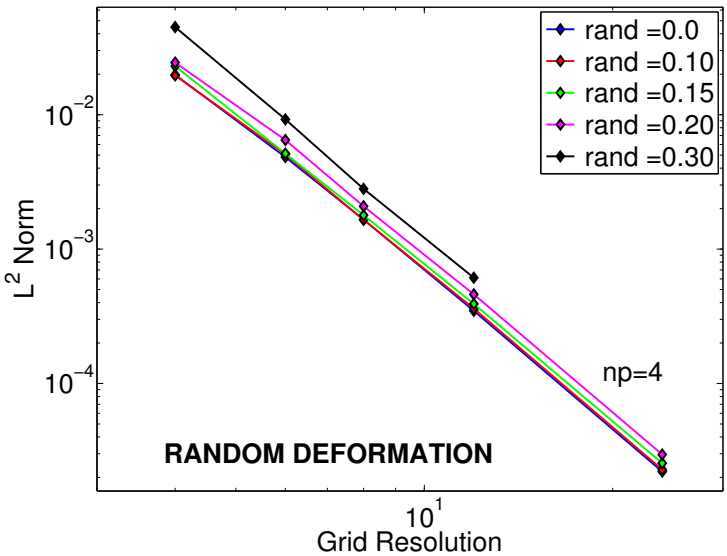


Figure 6.23: Linearized MHD L^2 norm versus grid resolution for varying degrees of randomness for $np = 4$.

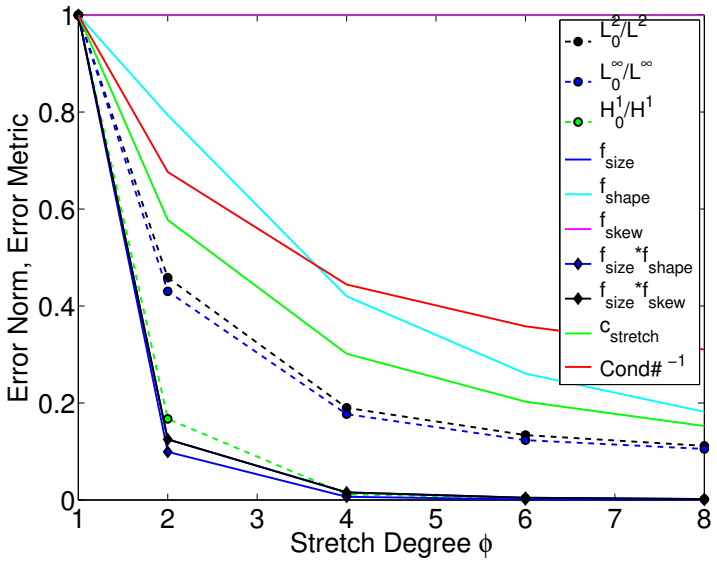


Figure 6.24: Linearized MHD inverse error norms and mesh quality metrics (Section 6.3) for a stretch deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. The inverse L^2 norm, the inverse L^∞ norm, and H^1 semi-norm are plotted.

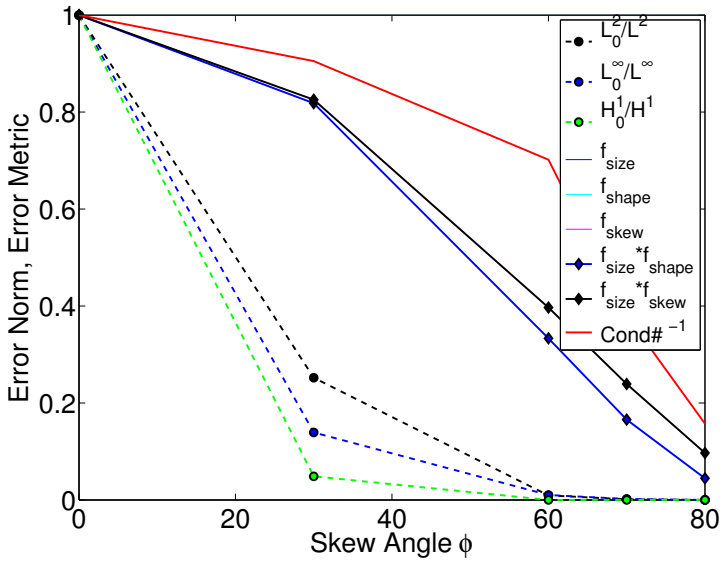


Figure 6.25: Linearized MHD inverse error norms and mesh quality metrics (Section 6.3) for a skew deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. The inverse L^2 norm, the inverse L^∞ norm, and H^1 semi-norm are plotted.

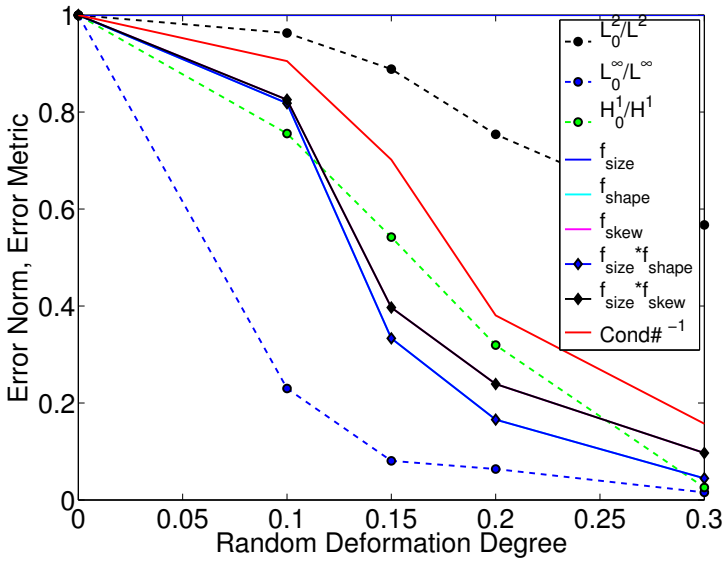


Figure 6.26: Linearized MHD inverse error norms and mesh quality metrics (Section 6.3) for a random deformation. Error norms and metrics are normalized (if necessary) to range from 1 to 0, where 1 is an undeformed element, and 0 is degenerate element. The inverse L^2 norm, the inverse L^∞ norm, and H^1 semi-norm are plotted.

6.8 Mesh Deformation Conclusions

The results of this study show that for the typical types of elemental mesh distortions and representative differential operators (namely, those in the advection, Poisson's, and linearized MHD equations) the spatial order of the spectral element method is preserved with varying degrees of mesh distortion. However, the error is also shown to be a function of overall resolution. By plotting the metrics along with solution error norms, it is observed that the error trend is similar to that of the mesh metrics. This gives a good indication of how the metrics can predict solution error. Knowing that the spatial convergence rates are preserved, an appropriate mesh resolution and spectral element polynomial order can also be chosen for target solution accuracy. This, in turn, suggests that locally increasing the resolution in regions of high grid distortion can improve the overall accuracy of the solution.

The variability of different equation sets and solutions increases the complexity of evaluating the mesh distortion effects, and one should remain cautious about how the mesh might influence their particular problem. Nevertheless, it has been shown that some *a priori* metrics can provide a good estimate to potential error in the solution and are a useful tool in mesh creation and analysis. Both the advection and linearized MHD equations are solved and both are hyperbolic. The linearized MHD is significantly more complicated then the advection equation and the results show meaningful correlations between the solution error and the mesh quality metrics. Despite the differences in the equation sets analyzed, the analysis does indicate general applicability.

One must note that the analysis presented here makes a careful attempt to minimize boundary effects by keeping the solution near zero at boundaries and/or use of periodic boundary conditions. The present analysis also uses a regular tiling of the mesh (with the exception of the 'random' mesh) which amounts to approximately homogenous sized elements. Additionally only smooth isotropic solutions are considered. When one solves more realistic problems that have boundary effects, more complex boundary conditions, non-smooth, and anisotropic solutions on more realistic heterogenous meshes – the error predictions are naturally more difficult to quantify. However,

even under such circumstances, the analysis presented here can lead to a reliable predictor of the location and magnitude of the highest solution error associated with the greatest distortion of the computational mesh.

In this study, the spectral stiffness matrix condition number, and the product of the Knupp metrics provided the most consistent metrics when comparing to the test problem solution error. Both of these metrics are formulated for finite elements (linear finite elements in the Knupp case), while the Kallinderis metric is formulated for a finite volume type approach. The stiffness matrix condition number is a global measure and is the most computationally intensive metric to calculate. Since the Knupp and Kallinderis metrics are local (element wise) measures, they are quick to calculate, which makes them attractive when dealing with large meshes.

Chapter 7

MULTI-BLOCK DEVELOPMENT AND CAD INTERFACE INTEGRATION

7.1 Motivation for Multi-Block Development

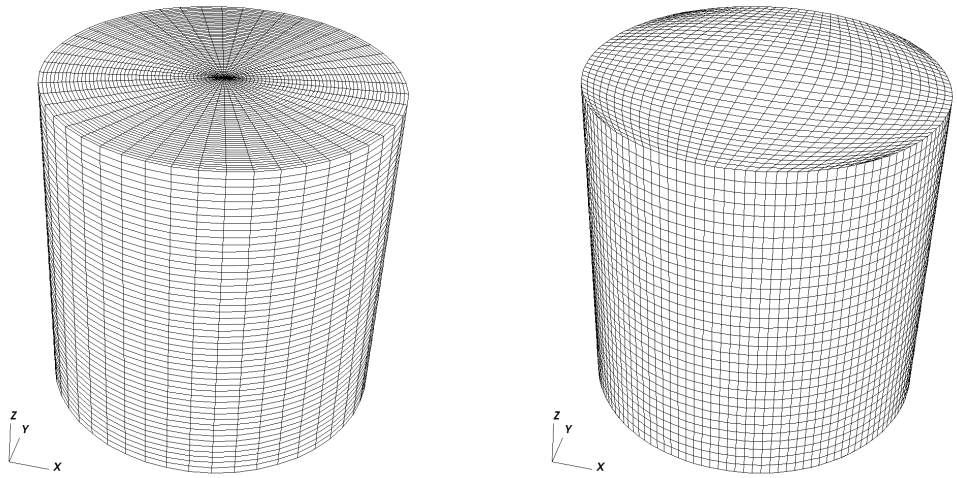
The motivation behind multi-block development is to allow for greater geometric flexibility in computational domains, while preserving the structured mesh organization within each block. A multi-block framework allows for a pseudo-unstructured mesh that can handle complex geometries, but does not use a completely unstructured mesh, and therefore does not suffer from some of the problems associated with unstructured mesh elements.

The original single block HiFi code uses a logical cube as its computational domain, which means any domains shape that the user specifies must also logically be a cube. This greatly limits the geometric possibilities, and can result in severe mesh distortions if one tries a logical-to-physical mapping that results in elements that are close to being degenerate. The multi-block development framework allows for an arbitrary number of blocks, connected in an unstructured fashion to be used as a single computational domain. Many more geometries are possible, like non-axisymmetric, and non-simply connected configurations.

7.1.1 Simple Cylindrical Geometry Improved

The multi-block geometries need not be complex in order to be useful. Take for instance a cylindrical geometry: a logical cube must be severely distorted to create a physical cylinder and degenerate elements result. Two possible configurations for a single block to a cylinder are the 'polar axis' and 'square-to-circle' mappings. Figure 7.1 demonstrates these two possibilities. Notice that in both cases degenerate elements exist. In the 'polar axis' case the center axis of the cylinder is one of the faces of the logical cube collapsed to a line. The hexahedral elements become triangular prisms at the center, and with increasing azimuthal resolution the angle decrease indefinitely. In the 'square-to-circle' case, the logical corners of the mesh now must be fit to the circular cross section. This yields elements where a logical 90 degree angle approaches 180 degrees as the circular curvature is resolved.

The cylindrical mesh can be improved by using multiple logical blocks. Using five



(a) Polar Axis (b) 'Square-to-Circle'

Figure 7.1: Two cylindrical geometries mapped from a single logical block.

blocks, a configuration where no degenerate elements exist can be created. Figure 7.2 shows an example mesh using a five block configuration. Notice that the mesh has no degenerate elements, but is no longer a strictly structured mesh. Unstructured connections exist at the corners of the block in the center of the cylinder. The single block code would not be able to use this geometry and requires the special treatment of the multi-block framework.

7.1.2 Complex Non-Axisymmetric and Non-Simply Connected Geometries

Other more complex configurations are desirable, including non-axisymmetric and non-simply connected geometries. Normally these would require unstructured meshes, but in the multi-block framework they are possible. For example the non-axisymmetric case where a cylinder has extrusions spaced out azimuthally. Figure 7.3 shows an example mesh with this configuration. This configuration uses 76 blocks. The spacing between the extrusions is half the width of the extrusions and the large number of

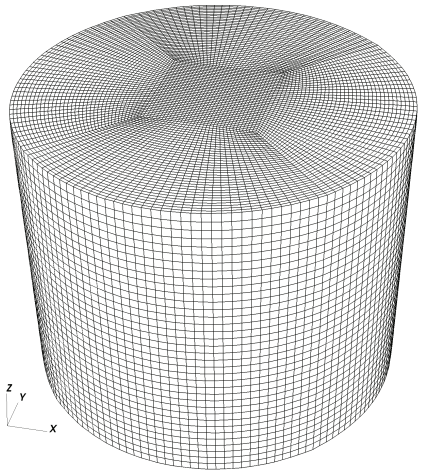


Figure 7.2: A multi-block cylindrical geometry consisting of five blocks.

blocks is used to keep the element sizes regular in the azimuthal plane. The flexibility to use an arbitrary number of blocks and have them connected in a unstructured fashion make this type of mesh possible.

7.2 Multi-Block Collection of Logical Cubes

The multi-block framework uses a collection of logical blocks, which each undergo a coordinate transformation to some physical shape of interest. Consider the diagram in Figure 7.4, where a single uniform logical block with coordinates ξ, η and ψ are transformed into a physical block with coordinates x, y, and z. The shape is no longer a cube, although, logically it is still a cube and can easily accommodate a structured mesh.

In the case of a multi-block geometry, there can be an arbitrary number of blocks, and they can be connected so long as the faces are conformal. In multi-block development, the code must keep track of which block faces are connected to other block faces, and it must also keep track of the relative orientation. Each block has its own logical coordinate system, and it can be arranged in any particular orientation relative to the

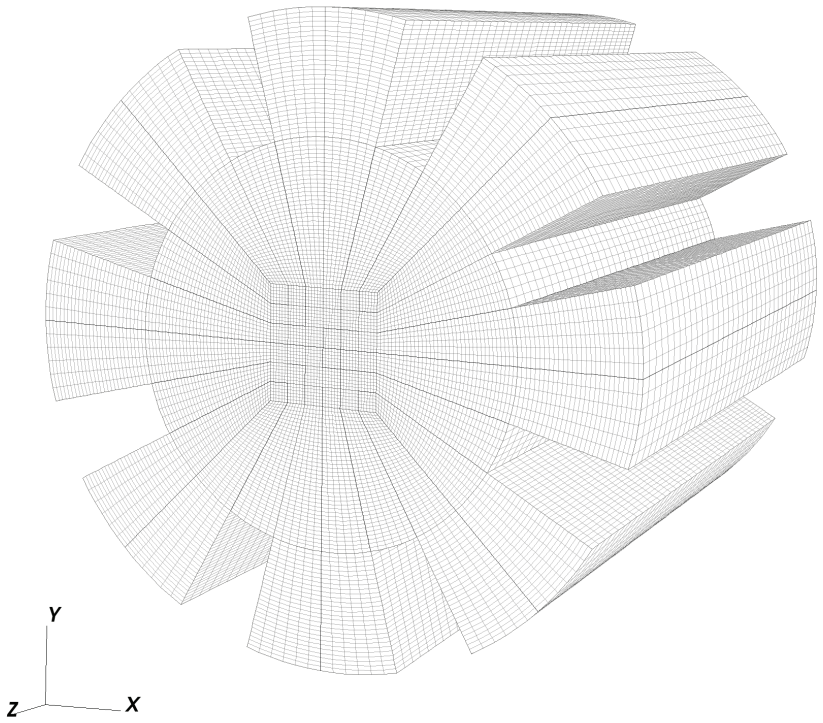


Figure 7.3: A multi-block cylindrical geometry with non-axisymmetric protrusions consisting of 76 blocks.

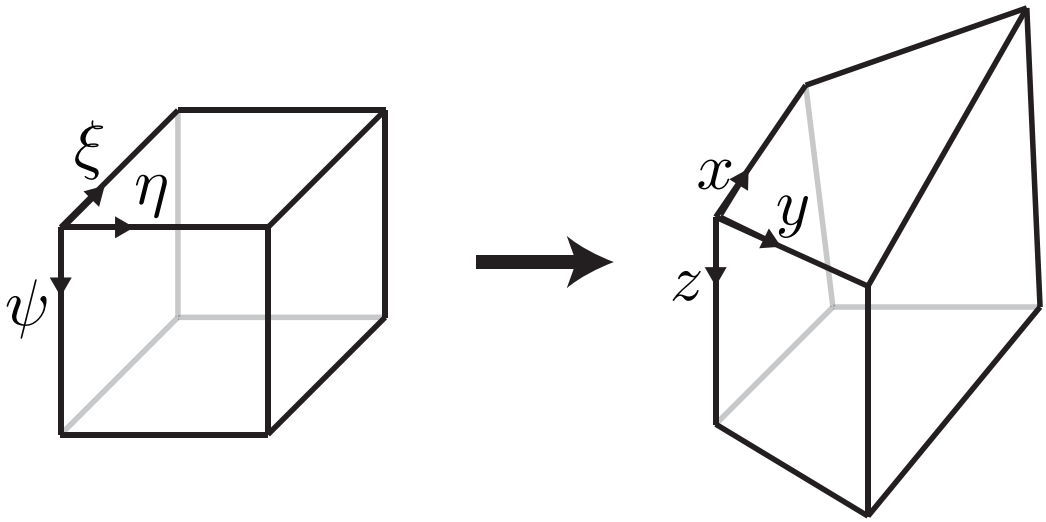


Figure 7.4: Diagram of a logical to physical coordinate transformation.

other blocks. The diagram in Figure 7.5 illustrates a scenario where five blocks are connected and the coordinate systems of each individual block are not aligned. Due to the unique connections in this case, it would be impossible for all logical coordinates on all blocks to always be aligned. This particular block configuration could be mapped to the five block cylinder shown in Figure 7.2.

7.3 CAD Interface and Advanced Mesh Generation with CUBIT

As mentioned in Chapter 3 a user friendly interface is necessary, especially when the computational domain geometry becomes complex. To avoid tedious manual specification of the geometry, a CAD interface using the CUBIT [2] mesh generation software is included in HiFi. The CUBIT mesh generator has the ability to import several different CAD file types including ACIS, IGES, STEP, Granite, AVS, Genesis/Exodus, Facets, STL, Patran, Ideas, Abaqus, Nastran, Catia, and Cubit (Native). Additionally CUBIT has basic CAD capabilities for creating less complex geometries in its native format.

CUBIT has an advanced suite of mesh generating tools, and can create triangular/tetrahedral meshes but it specializes in quadrilateral/hexahedral meshes. It also

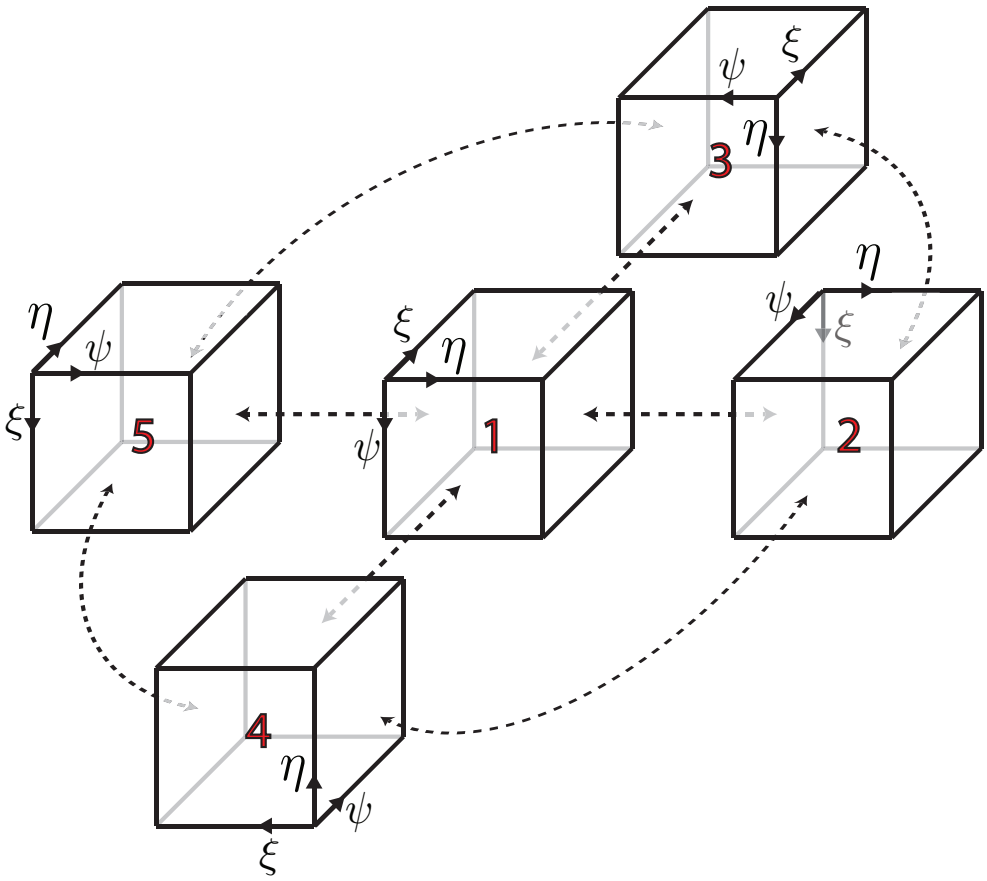


Figure 7.5: Diagram of logical block connections for a 5 block configuration.

has the ability to create both structured and unstructured meshes. HiFi domains are structured and use hexahedral elements, making CUBIT capable of creating the necessary meshes for use in HiFi.

A key feature in CUBIT is its ability to partition a geometry into several blocks (each block is a logical cube or rectangle) and mesh each block independently, but still maintain conforming elements between partitions. This feature is essential to creating useful multi-block meshes. The multi-block development in HiFi is described in Section 7.1. Figure 7.6 shows an example HIT-SI like geometry loaded into CUBIT and its associated mesh. The different colors represent the logical cube partitions that are used in the multi-block framework. A cutaway of the mesh also is shown so it can be seen how each partition is a logical cube and structured mesh.

CUBIT also includes several mesh smoothing algorithms, to minimize tangled grids, and improve distorted elements. In addition to these mesh smoothing algorithms included in CUBIT, the mesh deformation metrics described in Chapter 6 are used during the mesh creation process. After an initial mesh is created, the *a priori* mesh metrics will analyze the mesh and provide some quantification for the quality. The user can then decide if additional smoothing, partitioning, or resolution is necessary to improve the quality of the mesh. This process can also be semi-automated by improving the mesh in certain ways if it does not meet some mesh quality metric criteria.

Figure 7.7 shows a schematic of the CAD interface to the HiFi code. The schematic shows the steps necessary to start with CAD drawing, and end up with an acceptable mesh for use in the HiFi code. It includes a loop where the geometry is partitioned, meshed, and then checked by the mesh quality metrics. If the calculated metrics are acceptable, the resulting mesh can continue for use in HiFi, otherwise it is potentially repartitioned, further resolved, and/or smoothed before the mesh metrics are checked again.

Despite CUBIT's advanced mesh generation and smoother algorithms, it does not create high-order (spectral) element node information. It can at most produce the internal nodes for quadratic elements. It is therefore necessary to create extra internal nodes that correspond to the high order degrees of freedom present in the spectral

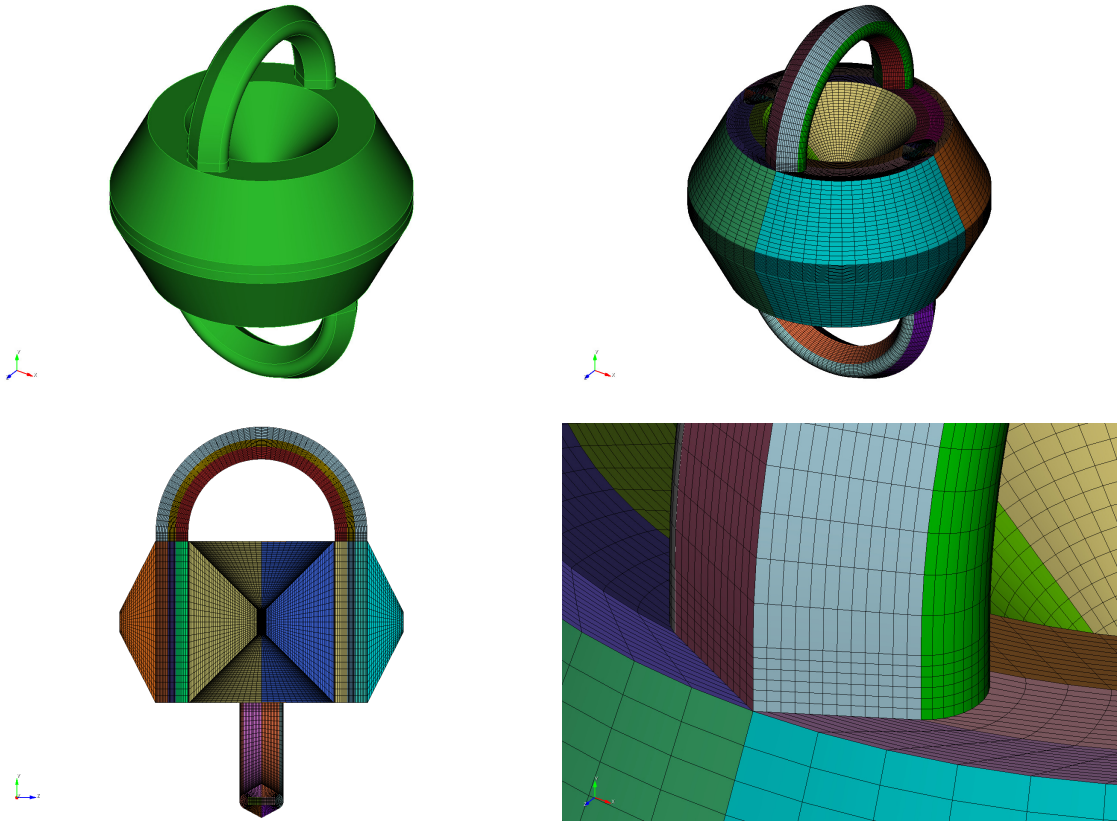


Figure 7.6: HIT-SI geometry and mesh created with CUBIT. Each color represents a logical cube partition or block. This type of mesh can be used in the multi-block framework described in Section 7.1

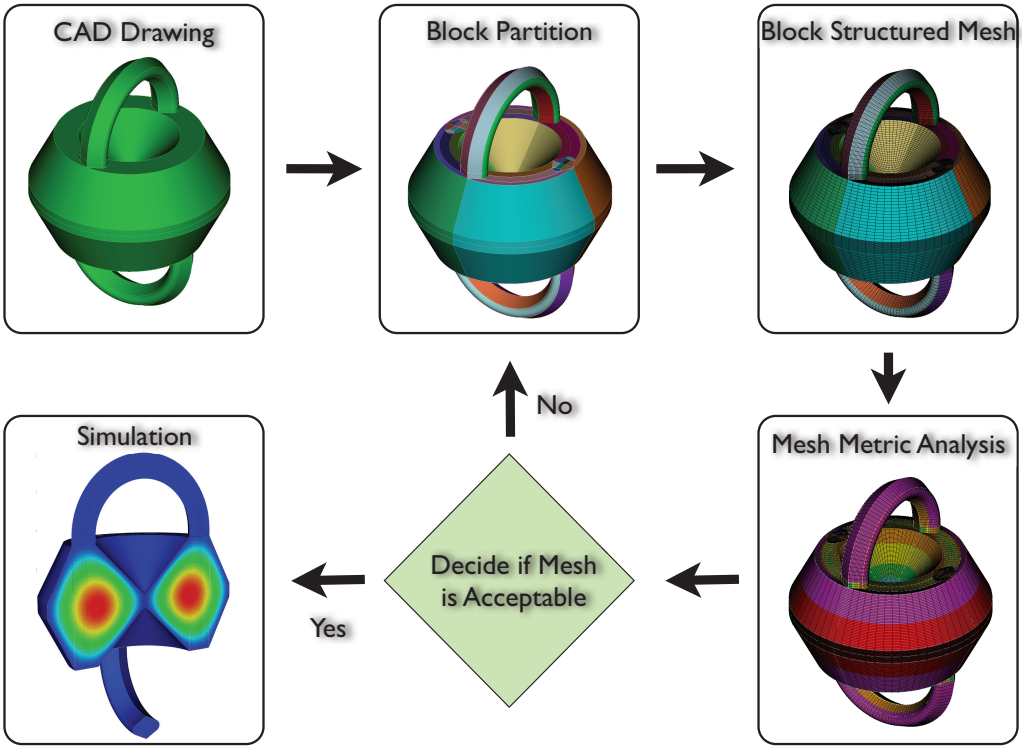


Figure 7.7: A schematic of the CAD interface to the HiFi code

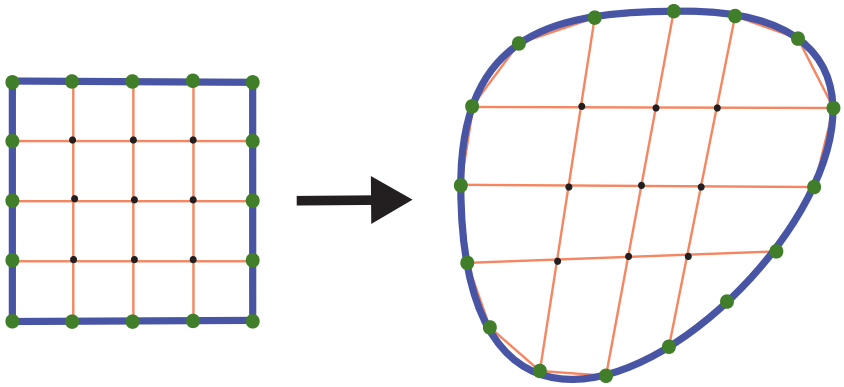


Figure 7.8: A 2D representation of a logical-to-physical transformation of a high-order element. The high-order element boundaries are depicted in blue, and the internal element boundaries that are generated by CUBIT for use in HiFi are in red. HiFi solves a system for each element for the basis function amplitudes based on the CUBIT generated geometry.

method of finite element. One method of achieving this is by creating the extra nodes in CUBIT as linear elements. For instance if a polynomial degree $np = 4$ is chosen, and the number of elements in one dimension $nx = 3$, then a total of $nx * np + 1$ nodes are needed to create the high-order mapping. In CUBIT one would create 12 linear elements in that dimension, rather than the 3 spectral elements that are specified by HiFi. A 2D cartoon shown in Figure 7.8 depicts this situation with the high-order element boundary in blue, and the internal "elements" that are created by CUBIT with red lines. Once the mesh information is read into HiFi, it uses the extra node locations as internal interpolation points for the particular element and solves for a continuous mapping within the element. This mapping information that it has solved for are the amplitudes of the spatial basis functions used in the high-order finite element method (described in Section 4.2). This process allows the use of a low order mesh generator like CUBIT to be used in a high-order code like HiFi.

Chapter 8

APPLICATION TO A Z-PINCH

It is well known that a close-fitting conducting wall has the ability to stabilize a plasma pinch. Early work with arc sources in high temperature gases found a close-fitting wall to provide stability [51, 52]. Subsequently early arcjet research in the 1960s established wall stabilization techniques empirically [53] and many early designs used a constrictor as a wall stabilization technique [54, 55, 56, 57, 58, 59, 60]. Much of this early work focused on finding optimal parameters to operate arcjets as rocket engines, while changing the constrictor diameter, input power, breakdown voltage, gas flow, etc.

The wall stabilization techniques developed throughout the gas arc and arcjet research were extended to the Z-pinches and theta pinches [61, 62, 63, 64, 65, 66], and a more theoretical understanding of the phenomenon was established. Later research by Shumlak and Hartman [67] using a linear stability analysis found that a close-fitting conducting wall around a diffuse pinch provides a stabilizing effect up to a point. When the wall is moved beyond a threshold distance from the pinch, the wall no longer has a stabilizing effect.

Wall stabilization has also been observed in tokamaks [68, 69] and can increase attainable values of β. It is found that the effects of resistivity in the close-fitting wall are destabilizing to tokamaks. Additionally a close-fitting wall in spheromaks are found to stabilize the plasma [70].

The ZaP Z-pinch experiment [71] has made recent design modifications that breaks the previous cylindrical axisymmetry. A 35 cm long section of outer electrode has slots cut, leaving 'rods' of conducting material. Wall stabilization properties for a pinch with a close-fitting conducting wall are well known, and the stabilization properties are investigated in this non-axisymmetric configuration. This design change has coincided with recent advancements in the HiFi [4] modeling code allowing for non-axisymmetric geometries. The HiFi code is used to model the ZaP Z-pinch configuration in the more complex geometry to better understand the stability properties of the pinch.

The Z-pinch $m = 1$ kink mode is explored as a verification of the HiFi code and the multi-block geometry framework that makes non-axisymmetric geometries possible. ZaP like experimental parameters are used in the simulations and compared to a linear stability analysis. The results of the simulation are compared to experimental data

from the ZaP experiment to better understand the implications of the design change.

8.1 Z-Pinch Background

The Z-pinch is a one-dimensional configuration with a purely poloidal magnetic field and axial current density [72]. Its name is derived by the fact that current only flows in the axial $\hat{z}$ direction. An equilibrium can be obtained by solving the radial pressure balance equation

$$\frac{d}{dr}\left(p + \frac{B_\theta^2}{2\mu_0}\right) + \frac{B_\theta^2}{\mu_0 r} = 0 \tag{8.1}$$

where r is the radial dimension, p is the pressure, B_θ is the poloidal magnetic field, and μ_0 is the permeability of free space. This equation represents the balance between magnetic pressure and particle pressure.

8.1.1 $m = 1$ Kink Mode

Z-pinches are classically unstable to the $m = 1$ kink mode instability, and are well known both theoretically and experimentally [73, 74, 75, 76, 77, 67]. Due to the extensive studying of this instability it makes a good test problem for code verification. The kink mode instability grows when a locally higher magnetic field is experienced on one side of the pinch and the $\vec{J} \times \vec{B}$ forces then push the plasma fluid in one direction. A cartoon of this phenomenon is shown in Figure 8.1. The instability grows and is predicted by linear stability analysis [78, 67].

8.2 The ZaP Z-Pinch Experimental Configuration

The ZaP experiment is a simple linear Z-pinch experiment with no applied magnetic fields [71]. The experiment consists of a coaxial plasma accelerator region, beginning at $z = -75$, and a pinch assembly region beginning at $z = 0$ moving from left to right (see Figure 8.2). An approximately 100 cm long, and 1 cm radius plasma is formed in the assembly region by injecting gas into the acceleration region and discharging a capacitor bank across electrodes. Figure 8.3 is a schematic that illustrates the pinch formation process. Typical experimental operating parameters are displayed in Table

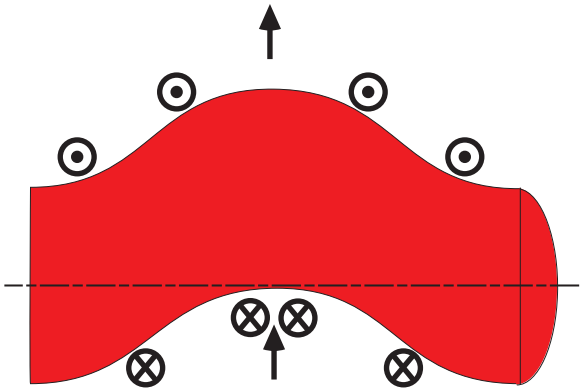


Figure 8.1: Cartoon diagram showing how locally higher magnetic field on one side of a Z-pinch leads to the m=1 kink instability.

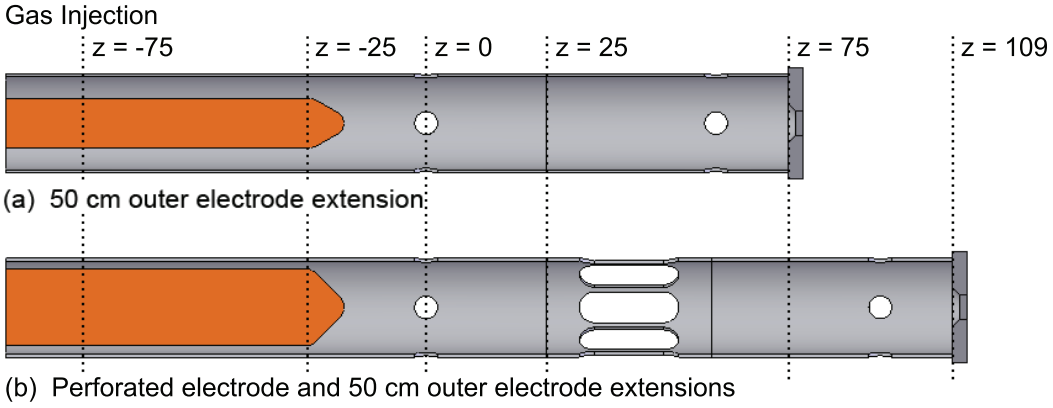


Figure 8.2: Diagram of the ZaP experiment configuration.

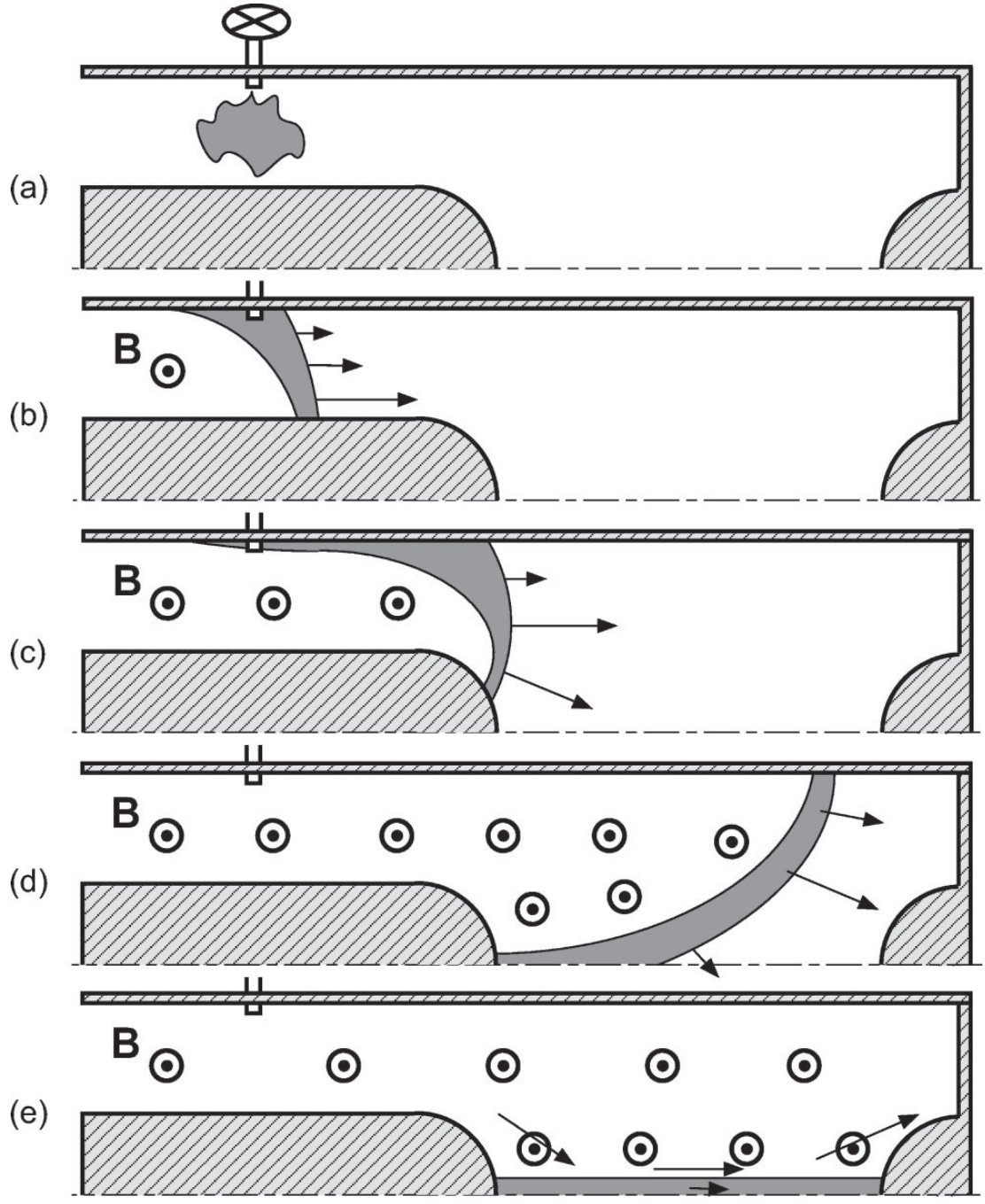


Figure 8.3: Schematic illustrating the Z-pinch formation in the ZaP flow Z-pinch experiment: (a) neutral gas injection, (b) breakdown and ionization of the gas and current accelerates the plasma axially, (c) plasma moves radially towards the pinch axis, (d) plasma forms on along the axis, (e) plasma attaches to outer electrode and end wall, while inertia maintains the axial plasma flow [79].

8.1.

Table 8.1: ZaP Operating Parameters [71]

Parameter	Value
Accelerator Length (cm)	100
Outer Electrode Radius (cm)	10
Inner Electrode Radius (cm)	5 or 8
Pinch Radius, a (cm)	1
Length (cm)	100
Peak Current (kA)	300-400
Electron Density (m^{-3})	10^{22} - 10^{23}
Total Temperature, $T_e + T_i$ (eV)	150 - 250
Magnetic Field (T)	1-2
Pinch Duration (μs)	30-70

The experimental apparatus is shown in Figure 8.2 in two configurations: (a) without a 'rod' extension region, and (b) with a 'rod' electrode extension region. The 'rod' extension configuration has slots cut into the outer electrode, which removes approximately 70% of material, leaving rods of electrode in the section. The purpose of the 'rod' extension is to investigate the impact the removal of the electrode wall material has on the stability of the pinch, as well as to open up optical diagnostic access to the plasma. The 'rod' extension region is precisely where the plasma confinement region loses its axisymmetry and is modeled using the multi-block framework of HiFi. Section 8.4 below describes these simulations.

8.2.1 Diagnostic Configuration

There are three major diagnostics for the ZaP experiment: magnetic probe arrays for magnetic field mode data, two multi-chord interferometers for density gradient measurements, and a fast framing camera for optical light emissions. Each of these diagnostics take measurements as a function of time.

Magnetic Probe Arrays

There are two azimuthal magnetic probe arrays with eight probes each located at approximately $z = 35$ cm and $z = 70$ cm. Additionally an axial magnetic probe array consists of probes located every 5 cm along the length of the pinch assembly region. The azimuthal and axial probe arrays can measure the field fluctuations as a function of time at their respective locations. The data is Fourier decomposed to isolate magnetic mode contribution for the $m = 0$ (average magnetic field), $m = 1$ (radial displacement), $m = 2$ (ellipticity), and $m = 3$ (triangularity).

Multi-Chord Interferometer

A helium-neon, Mach-Zehnder, heterodyne quadrature multi-chord interferometer is used at two locations to measure the chord-integrated density. Both of these locations are in the 'rod' section where there is easy optical access. The measurements are taken at different impact parameters yielding a measurement of the density gradient between the two measurements.

Fast Framing Camera

An Imacon 790 fast framing camera is installed at the 'rod' location for visible light emission measurements. The camera has a 1 μs frame rate allowing for images of the plasma as a function of time. The images are compared to interferometry and magnetic probe array data for consistency and help determine the size, shape, displacement, and stability of the plasma.

8.3 Linear Stability Analysis

A method following from Appert and Gruber [78] and Shumlak and Hartman [67] is used to calculate the theoretical linear growth rates of the $m = 1$ kink mode in diffuse Z-pinches. This method solves linearized ideal MHD equations as an eigenvalue

problem, resulting in a pair of coupled first order differential equations:

$$Xr\frac{\partial p^*}{\partial r} + C_{11}p^* + C_{12}(r\xi_r) = 0, \tag{8.2}$$

$$Xr\frac{\partial (r\xi_r)}{\partial r} + C_{21}p^* + C_{22}(r\xi_r) = 0, \tag{8.3}$$

where r is the radial distance, ξ_r is the radial displacement,

$$p^* = 2\frac{B_\theta^2}{\mu_0}\frac{\xi_r}{r} - \frac{\rho\gamma^2}{Y}\nabla\cdot\xi, \tag{8.4}$$

$$Y = \frac{\rho^2\gamma^4}{X\Gamma p + \rho\gamma^2 B_\theta^2/\mu_0}, \tag{8.5}$$

$$X = \rho\gamma^2 + \frac{F^2}{\mu_0}, \tag{8.6}$$

$$F = \frac{m}{r}B_\theta, \tag{8.7}$$

$$C_{11} = 2\left(\frac{F^2}{\mu_0} + Y\frac{B_\theta^2}{\mu_0}\right), \tag{8.8}$$

$$C_{12} = X^2 - 2X\frac{B_\theta s}{\mu_0}\frac{\partial(B_\theta/r)}{\partial r} - 2C_{11}\frac{B_\theta^2}{r^2\mu_0}, \tag{8.9}$$

$$C_{21} = m^2 + k^2r^2 + r^2Y, \tag{8.10}$$

$$C_{22} = -C_{11}, \tag{8.11}$$

and $\Gamma = 5/3$ is the ratio of specific heats. The boundary conditions used for the cylindrical geometry are

$$\xi_r|_{r=r_{wall}} = 0, \tag{8.12}$$

$$\left.\frac{\partial\xi_r}{\partial r}\right|_{r=0} = 0, \tag{8.13}$$

which corresponds to no displacement at the outer rigid wall, and a Neumann boundary condition on displacement at the axis. The boundary value problem defined by the differential equations 8.2 and 8.3 and the boundary conditions 8.12 and 8.13 are solved using a shooting method. This yields a complex eigenvalue γ, where if γ^2 is greater then zero, the mode is unstable and the growth rate is γ. Otherwise γ^2 is

less then 0 and this represents a stable oscillation with a frequency $\omega = -i\gamma$. This boundary value problem is solved for a variety of conditions and then compared to the simulations results.

8.4 Simulations of a Z-Pinch

A Z-pinch m=1 kink mode is modeled using the advantages of the multi-block features of the HiFi code. A straight circular cylindrical geometry (Figure 8.4(a)) composed of five blocks is used to model a periodic Z-pinch and verify the kink mode growth rates. Simulations with a more complex non-axisymmetric geometry are performed (Figure 8.4(b)). This geometry represents recent changes made to the outer electrode of the ZaP experiment [80] and is described earlier in Section 8.2. In both configurations the kink growth rates are compared to results obtained from a linear stability analysis. Additionally, in the cylindrical geometry, shear flow is introduced to demonstrate the known stabilizing effects and to more closely match the conditions of the ZaP experiment.

8.4.1 MHD Equation Model

A viscous, hyper-resistive MHD with isotropic thermal conduction HiFi module written in 3D Cartesian coordinates provides the basis for performing the tests. The equations are written in flux-source form as:

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\vec{v} \\ \vec{A} \\ \frac{1}{\gamma-1}p \\ 0 \end{bmatrix} + \nabla\cdot\begin{bmatrix} \rho\vec{v} \\ \rho\vec{v}\vec{v} + p\vec{I} - \mu\nabla\vec{v} \\ -\nu\nabla\vec{j} \\ \frac{\gamma}{\gamma-1}p\vec{v} - \kappa\nabla T \\ (\nabla\cdot\vec{A})\vec{I} - \nabla\vec{A} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{j}\times\vec{B} \\ \vec{v}\times\vec{B} - \eta\vec{j} \\ (\vec{v}\cdot\nabla p + \eta\vec{j}\cdot\vec{j} + \mu(\nabla\vec{v}:\nabla\vec{v}) \\ \vec{j} \end{bmatrix}, \tag{8.14}$$

where ρ is mass density, $\vec{v}$ is fluid velocity, $\vec{A}$ is magnetic vector potential, and p is pressure. The symbol : is the tensor double dot product operator defined as

$$\nabla\vec{v} : \nabla\vec{v} = \sum_{i=1}^{3}\sum_{j=1}^{3} \nabla v_{ij} \nabla v_{ji}, \tag{8.15}$$

resulting in a scalar value. The dissipative parameters μ, η, ν, and κ are viscosity, resistivity, hyper-resistivity, and isotropic heat conduction respectively. These parameters represent the inverse of the dimensionless parameters such as the Reynolds number $\mu = 1/Re$ (See Appendix C for more details). The hyper-resistive parameter ν multiplies a fourth order term providing high-order dissipation. This is useful for damping high-frequency noise in the current $\vec{j}$ and providing numerical stability. Additionally the magnetic field intensity $\vec{B} = \nabla \times \vec{A}$, current density $\vec{j} = \nabla \times \nabla \times \vec{A}$ and temperature $T = p/\rho$ are defined.

8.4.2 Bennett Pinch Initial Conditions

A Bennett Z-pinch equilibrium is used as the initial condition in the following analysis and is a solution to the force balance equation 8.1. The form of the equilibrium is:

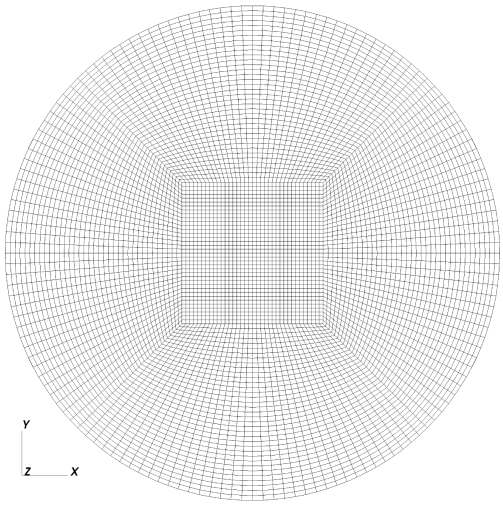
$$B_\theta = \frac{I_0}{2\pi}\frac{r}{r^2 + a^2} \tag{8.16}$$

$$J_z = \frac{I_0}{\mu_0 \pi}\frac{a^2}{(r^2 + a^2)^2} \tag{8.17}$$

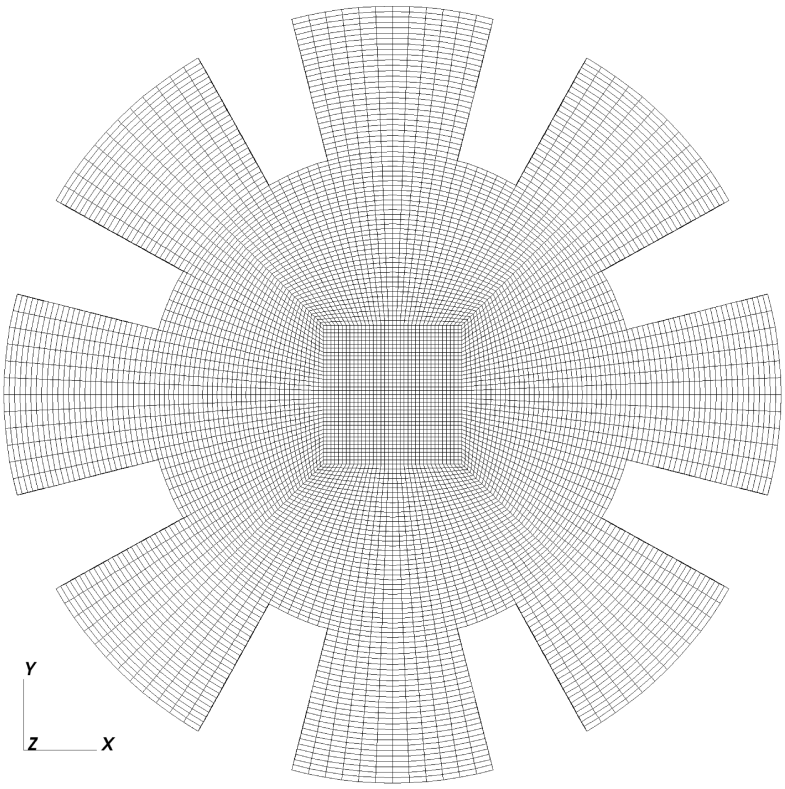
$$p = \frac{I_0^2}{\mu_0 8\pi^2}\frac{a^2}{(r^2 + a^2)^2} \tag{8.18}$$

where I_0 is the total current, r is the radius of the cylinder, and a is the characteristic plasma pinch radius [81]. In order to use this initial equilibrium with equation 8.14, the magnetic field must be represented as vector potential. Since $\vec{B} = \nabla \times \vec{A}$ equation 8.16 can be integrated with respect to r:

$$A_z = \int \frac{I_0}{2\pi}\frac{r}{r^2 + a^2} dr \quad \Rightarrow \quad A_z = -\frac{I_0}{4\pi}\mathbf{Log}[r^2 + a^2]. \tag{8.19}$$



(a) 5 Blocks



(b) 76 Blocks

Figure 8.4: Cross sections of the straight cylinder and cylinder with extrusions geometries.

These equations for A_z, J_z, and p can be used as the initial condition, along with some values for density ρ and momentum $\rho\vec{v}$. The fluid momentum is initialized to zero with a small perturbation to initiate the m=1 kink mode growth. The perturbation has the form

$$\rho v_y = \hat{\delta}\left[\cos(2\pi z/l_z)e^{-(x^2+y^2)/(0.03)^2}\right], \tag{8.20}$$

where $\hat{\delta}$ is the normalized magnitude of the perturbation, l_z is the axial wavelength of the pinch, and the exponential is to ensure the perturbation drops to zero at the radial boundary at $r = r_{wall} = 0.1$. A perturbation normalized by the Alfvén speed of $\hat{\delta} = \delta/v_A = 7.25 \cdot 10^{-9}$ is given and is small enough that the dynamics remain in the linear phase throughout the instability growth. This linear growth can then be compared to theoretical linear growth rates calculated from a linear stability analysis. For these simulations, conditions that closely match the operating parameters of the ZaP experiment are used (see Table 8.1). The ZaP experiment pinch has a radius of approximately $a = 1$ cm, has a peak current of $I = 400$ kA, a number density $n = 10^{23}$ m^{-3}, a magnetic field strength of $B = 2$ T, and the outer electrode radius $r = 10$ cm. In these simulations, a hydrogen plasma is assumed, so the density is $\rho = nM_p = 1.6726 \cdot 10^{-4}$, where $M_p = 1.6726 \cdot 10^{-27}$ is the mass of a proton. Equation 8.14 is non-dimensional, and the normalization can be seen in Appendix C.

8.4.3 Boundary Conditions

The boundary conditions applied to the fluid velocity are perfect slip:

$$\nabla_{\hat{n}} v_{\hat{t}} = 0 \tag{8.21}$$

$$v_{\hat{n}} = 0. \tag{8.22}$$

where $\nabla_{\hat{n}}$ is the normal gradient operator, $v_{\hat{t}}$ is the tangential fluid velocity, and $v_{\hat{n}}$ is the normal fluid velocity. These conditions prevents boundary layers by limiting the normal derivative of flow, and prevents normal flows into the wall. A conducting wall

boundary condition is applied to the at the wall

$$\frac{\partial A_{\hat{t}}}{\partial t} = -E_{\hat{t}} = 0 \tag{8.23}$$

where $A_{\hat{t}}$ is the tangential vector potential, t is time, and $E_{\hat{t}}$ is the tangential electric field. Additionally the density is given a 'natural' boundary condition, and the temperature at the wall is limited to the interior temperature.

$$\frac{\partial}{\partial \hat{n}} \left(p/\rho \right) = 0 \tag{8.24}$$

where $T = p/\rho$.

8.4.4 Minimization of Dissipation

Viscosity, resistivity, hyper-resistivity, and isotropic thermal conduction are included in the equation model described in equation 8.14. The ideal MHD linear stability analysis does not include dissipative terms, and it is therefore important to minimize these terms in the Z-pinch simulations to make a good comparison. The main function of the dissipative terms in the simulations is to maintain numerical stability, and it is found that both resistivity and isotropic thermal conduction are not needed. Only a relatively small amount of both viscosity and hyper-resistivity are included.

The hyper-resistive terms are minimized, and Table 8.2 and Figure 8.5 shows the effect on the kink mode growth rate. At high levels of hyper-resistivity it dramatically affects the behavior of the kink mode. Once the hyper-resistivity is less then 10^{-6} it does not significantly impact the solution to reduce it further.

Similarly the viscosity is reduced to measure the impact on the $m = 1$ kink mode growth rate. Table 8.3 and Figure 8.6 show the results. Since the viscosity operates on the fluid velocity, and thus the kinetic energy, it makes sense that with an increasing amount of viscosity, the growth rates declines. Although, with only relatively small amounts of viscosity, it does not have a drastic impact on the solution, and the kink mode growth is qualitatively fairly consistent. To maintain numerical stability but not

Table 8.2: Normalized growth rates calculated from Z-pinch simulations with varying amount of hyper-resistivity. The characteristic plasma size $a = 1.5$ cm with a $ka = \pi$, cylinder radius $r_{wall} = 10$ cm, viscosity $\mu = 2.5 \cdot 10^{-3}$, on a five block grid with the center block resolution $nx = ny = nz = 12$ and $np = 3$. Linear stability analysis predicts the growth rate $(\gamma/kV_A)_{ls} = 0.38424$

ν	$(\gamma/kV_A)_n$
5.00e-4	0.25156
5.00e-5	0.34885
1.00e-6	0.36793
1.00e-9	0.36828

significantly damp the growth rate, a modest viscosity $\mu = 2.5 \cdot 10^{-3}$ is chosen for the remainder of the simulations.

Table 8.3: Normalized growth rates calculated from Z-pinch simulations with varying amount of viscosity. The characteristic plasma size $a = 1.5$ cm, $ka = \pi$, cylinder radius $r_{wall} = 10$ cm, hyper-resistivity $\nu = 10^{-9}$, on a five block grid with the center block resolution $nx = ny = nz = 12$ and $np = 3$. Linear stability analysis predicts the growth rate $(\gamma/kV_A)_{ls} = 0.38424$

μ	$(\gamma/kV_A)_n$
1.00e-2	0.33115
5.00e-3	0.35491
2.50e-3	0.36676
1.25e-3	0.37242
6.25e-4	0.37512
1.00e-4	0.37701
1.00e-6	0.37714

8.4.5 Convergence of Results

A spatial and temporal convergence study is done to insure the results are accurate. For spatial convergence, the resolution is increased in the cross-section and then in the axial direction. They are done independently of each other to see the impact of each increase in resolution. The five block cylindrical geometry shown in Figure 8.4(a), with varying resolution are used. Throughout the domain $np = 3$ is used. An increase in

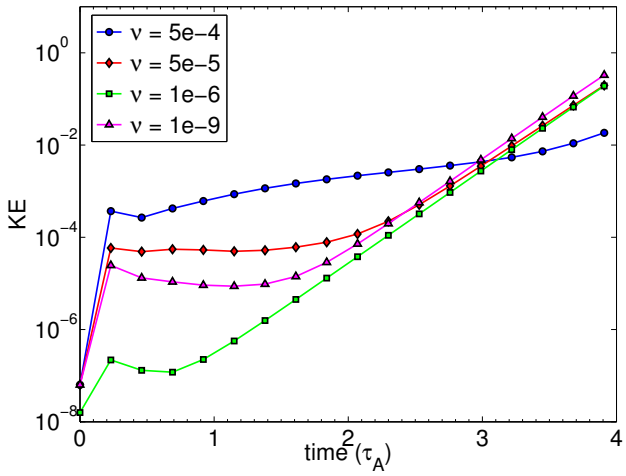


Figure 8.5: Kinetic energy of a Z-pinch simulation in time with varying amounts of hyper-resistivity. The characteristic plasma size $a = 1.5$ cm, $ka = \pi$, cylinder radius $r_{wall} = 10$ cm, viscosity $\mu = 2.5{\cdot}10^{-3}$, on a five block grid with the center block resolution $nx = ny = nz = 12$ and $np = 3$.

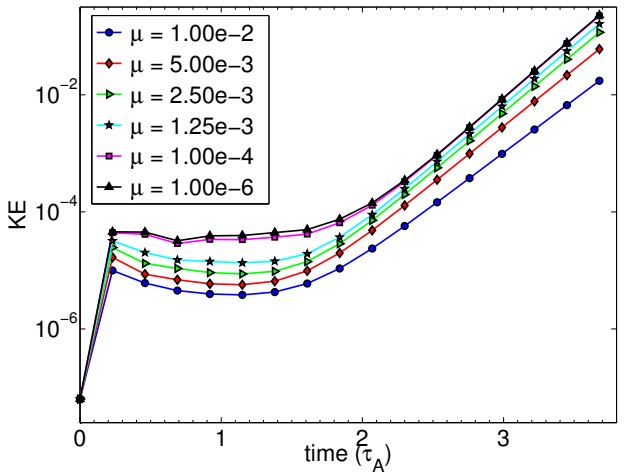


Figure 8.6: Kinetic energy of a Z-pinch simulation in time with varying amounts of viscosity. The characteristic plasma size $a = 1.5$ cm, $ka = \pi$, cylinder radius $r_{wall} = 10$ cm, hyper-resistivity $\nu = 10^{-9}$, on a five block grid with the center block resolution $nx = ny = nz = 12$ and $np = 3$.

cross sectional resolution from $nx = ny = 4$ to $nx = ny = 12$ and then $nx = ny = 18$ is used. Additionally increases in axial resolutions from $nz = 6$ to $nz = 12$ and $nz = 18$ are used. Table 8.4 shows the results. Notice the axial resolution does not impact the growth rate of the $m = 1$ kink mode as significantly as the cross sectional resolution.

Table 8.4: Normalized growth rates calculated from Z-pinch simulations with varying spatial resolutions. The characteristic plasma size $a = 1.5$ cm, $ka = 0.5$, cylinder radius $r_{wall} = 10$ cm, viscosity $\mu = 2.5 \cdot 10^{-3}$, and hyper-resistivity $\nu = 10^{-9}$. Linear stability analysis predicts the growth rate $(\gamma/kV_A)_{ls} = 0.38424$

nx	ny	nz	$(\gamma/kV_A)_n$
4	4	6	0.35040
4	4	12	0.35044
12	12	12	0.36825
12	12	18	0.36825
18	18	12	0.36674

The temporal convergence is studied by decreasing the time step Δt with a fixed spatial resolution. Table 8.5 shows the results. As the time step is reduced the value for the growth rate converges.

Table 8.5: Normalized growth rates calculated from Z-pinch simulations with varying time step. The characteristic plasma size $a = 1.5$ cm, $ka = 0.5$, cylinder radius $r_{wall} = 10$ cm, viscosity $\mu = 2.5 \cdot 10^{-3}$, hyper-resistivity $\nu = 10^{-9}$, and a spatial resolution of $nx = ny = nz = 12$. Linear stability analysis predicts the growth rate $(\gamma/kV_A)_{ls} = 0.38424$

Δt	$(\gamma/kV_A)_n$
8e-8	0.35120
4e-8	0.37384
2e-8	0.37068
1e-8	0.36942
5e-9	0.36885

8.4.6 Circular Cylinder Geometry

A straight circular cylindrical geometry is used to study the Z-pinch $m = 1$ kink mode. The mesh cross section is shown in Figure 8.4(a). Growth rates calculated from numerical simulations of the $m = 1$ kink mode for several pinch conditions are shown in Table 8.6 and are compared to the results from the linear stability analysis. Additionally Figure 8.7 shows results from a linear stability analysis while holding ka fixed at $\pi/4$, $\pi/2$, and π and varying r_{wall}/a. Notice for larger ka the effect of wall stabilization is minimized, even for large characteristic plasma radii.

Table 8.6: Normalized growth rates calculated from a numerical simulation of a Z-pinch $m = 1$ kink $(\gamma/kV_A)_n$ in a straight circular cylindrical geometry, $r_{wall} = 0.1$, compared to growth rates calculated from a linear stability analysis $(\gamma/kV_A)_{ls}$. Several values for the characteristic pinch size a are compared, with values of $ka = \pi/4, \pi/2$ and π.

a (cm)	$(\gamma/kV_A)_n$	$(\gamma/kV_A)_{ls}$	%error
1.0	0.57030	0.58405	2.35
1.5	0.57797	0.58405	1.04
2.0	0.57326	0.58396	1.83
3.0	0.56700	0.57787	1.88
4.0	0.52260	0.54035	3.29
5.0	0.42850	0.43274	0.98
1.0	0.50598	0.52125	2.93
1.5	0.51035	0.52125	2.09
2.0	0.51313	0.52125	1.56
3.0	0.51590	0.52123	1.03
4.0	0.51748	0.52050	0.58
5.0	0.51323	0.51604	0.54
6.0	0.49969	0.50271	0.60
7.0	0.47282	0.47533	0.53
8.0	0.42250	0.42948	1.63
1.0	0.36201	0.38424	5.79
1.5	0.36875	0.38424	4.03
2.0	0.37255	0.38424	3.04
3.0	0.37669	0.38424	1.97
4.0	0.37834	0.38424	1.54
5.0	0.37971	0.38423	1.17
6.0	0.37982	0.38406	1.11
7.0	0.37923	0.38335	1.07
8.0	0.37794	0.38137	0.90

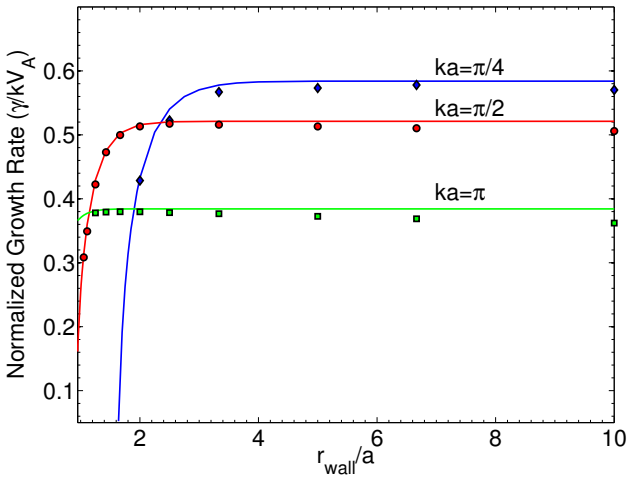
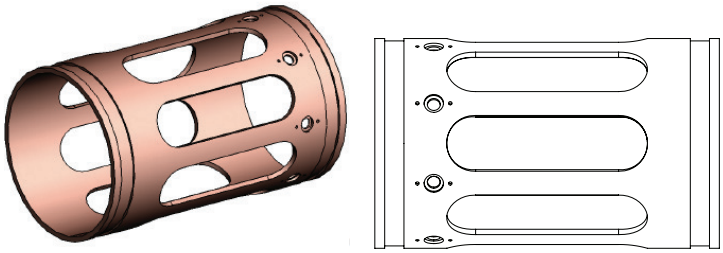


Figure 8.7: Normalized growth rates computed from a linear stability analysis (lines) and from HiFi simulations (markers) for various values of r_{wall}/a while holding constant $ka = \pi/4, \pi/2$ and π.

8.4.7 Modified Outer Electrode Geometry

The modified outer electrode multi-block geometry aims to model the section of the ZaP pinch region where there are cuts in the outer electrode. Figure 8.2 shows the cuts and Figure 8.8 provides a more detailed look. The cuts in the electrode mean that plasma could expand radially out of the original cylindrical volume and further into vacuum vessel. To represent this in a computational domain, extrusions are added to the cylindrical geometry such that in the simulation the plasma can have the ability to move into these regions. The depth of the extrusions are considered to be long enough such that the plasma interaction with ends of the extrusions is minimal.

Simulations with the modified outer electrode geometry are performed with a range of wall radius to characteristic plasma size ratios like the straight cylinder results. The pressure, current density and magnetic field profiles are comparable to the ZaP experiment, and the case with $a = 1.0$ cm most resembles the size pinch seen in the experiment. Additional larger pinch sizes are simulated to better understand how the extrusions may impact the growth rate of the $m = 1$ kink instability.



(a) (b)

Figure 8.8: Three-Dimensional rendering of the modification to the ZaP outer electrode (a) and 2D drawing (b). (Credit J. Rohrbach)

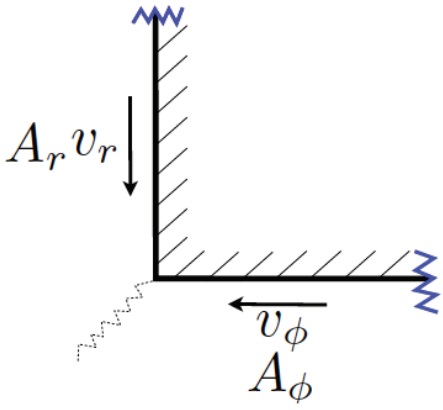


Figure 8.9: Schematic diagram representing a sharp corner region of a geometry.

Special Treatment of Boundary Condition at Sharp Corners

The boundary conditions resulting from the addition of extrusions to the cylindrical geometry require some special attention. There is a possibility for a discontinuity between vector quantities aligned with the azimuthal and radial walls. Consider the 2D diagram shown in Figure 8.9. It is possible to set a velocity boundary condition where the flow must instantaneously change direction at the corner. This would lead to a boundary layer formation that must be resolved, otherwise noise will be introduced into the system. Similar problems could result in other vector quantity boundary con-

ditions.

In the simulations here, a zero slip boundary condition is applied

$$\hat{n} \cdot \vec{v} = 0 \tag{8.25}$$

$$\hat{t} \cdot \vec{v} = 0 \tag{8.26}$$

where $\vec{v}$ is the fluid velocity, and $\hat{n}$ and $\hat{t}$ are the normal and tangential vectors respectively. This boundary condition is used as opposed to a perfect slip to avoid the problem of a velocity discontinuity. The fluid flows in the simulations are small enough at the wall such that boundary layers do not form. The alternative perfect slip condition would lead to a fluid layer propagating 45 degrees inward from the domain corner and is undesirable.

A conducting wall boundary condition is applied

$$\frac{\partial A_{\hat{t}}}{\partial t} = -E_{\hat{t}} = 0 \tag{8.27}$$

where $A_{\hat{t}}$ is the tangential vector potential, t is time, and $E_{\hat{t}}$ is the tangential electric field. A divergence of $\vec{A}$ condition is also applied. In cylindrical coordinates:

$$\nabla \cdot \vec{A} = \frac{1}{r}\frac{\partial}{\partial r}\left(rA_r\right) + \frac{1}{r}\frac{\partial A_\phi}{\partial \phi} + \frac{\partial A_z}{\partial z} = 0. \tag{8.28}$$

If this divergence condition is also applied at the boundary it ensures that the respective A_r and A_ϕ terms go to zero at the boundary, regardless of the orientation of the boundary wall. This condition reenforces the conducting wall condition and is compatible with the sharp corners that exist in the extrusion geometry.

Results of Modified Outer Electrode Simulations

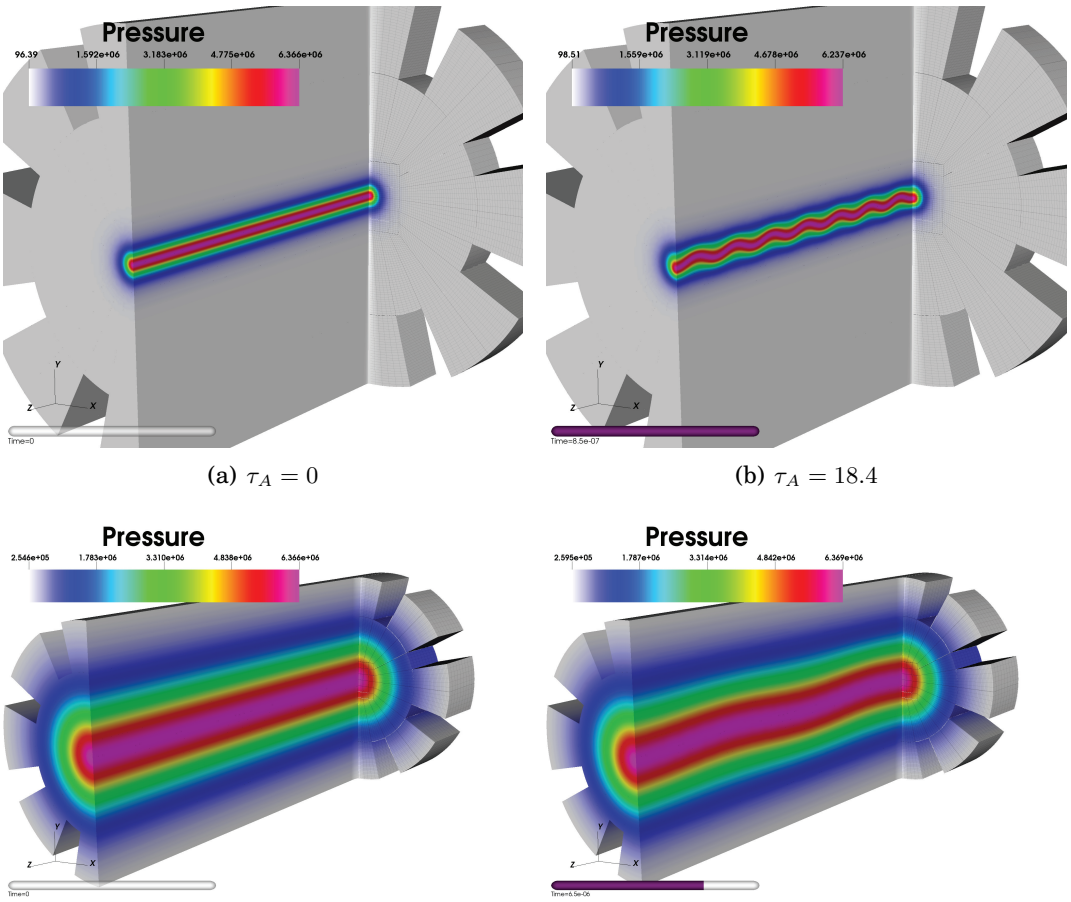
Table 8.7 displays the results of the normalized growth rates computed from HiFi simulations and compared to the linear stability analysis. All results have a wavenumber times pinch size $ka = \pi/2$. It can be seen that the growth rates are negligibly impacted

by the inclusion of the extrusions to the computational domain. With a pinch radius of $a = 8.0$ cm approaching the wall radius, it can be argued that the extrusions have a very small impact, and slightly increase the growth rate of the instability. Figure 8.12 shows the kinetic energy for cases with and without extrusions to demonstrate the minimal effect on growth rate. Notice the case with $r_{wall}/a = 1.25$, early in the linear growth phase, the case with extrusions diverges from the straight cylinder case. This is well before any nonlinear effects dominate, and is most likely a minimal modification to the kinetic energy due to the extrusions.

Table 8.7: Normalized growth rates calculated from a numerical simulation of a Z-pinch $m = 1$ kink $(\gamma/kV_A)_n$ in a straight circular cylindrical geometry compared with a cylinder with extrusions. Several values for the characteristic pinch size a are compared, while keeping $ka = \pi/2$.

a (cm)	$(\gamma/kV_A)_n$	$(\gamma/kV_A)_n(ext.)$
1.0	0.52125	0.50651
4.0	0.51748	0.51669
6.0	0.49969	0.50136
8.0	0.42250	0.43661

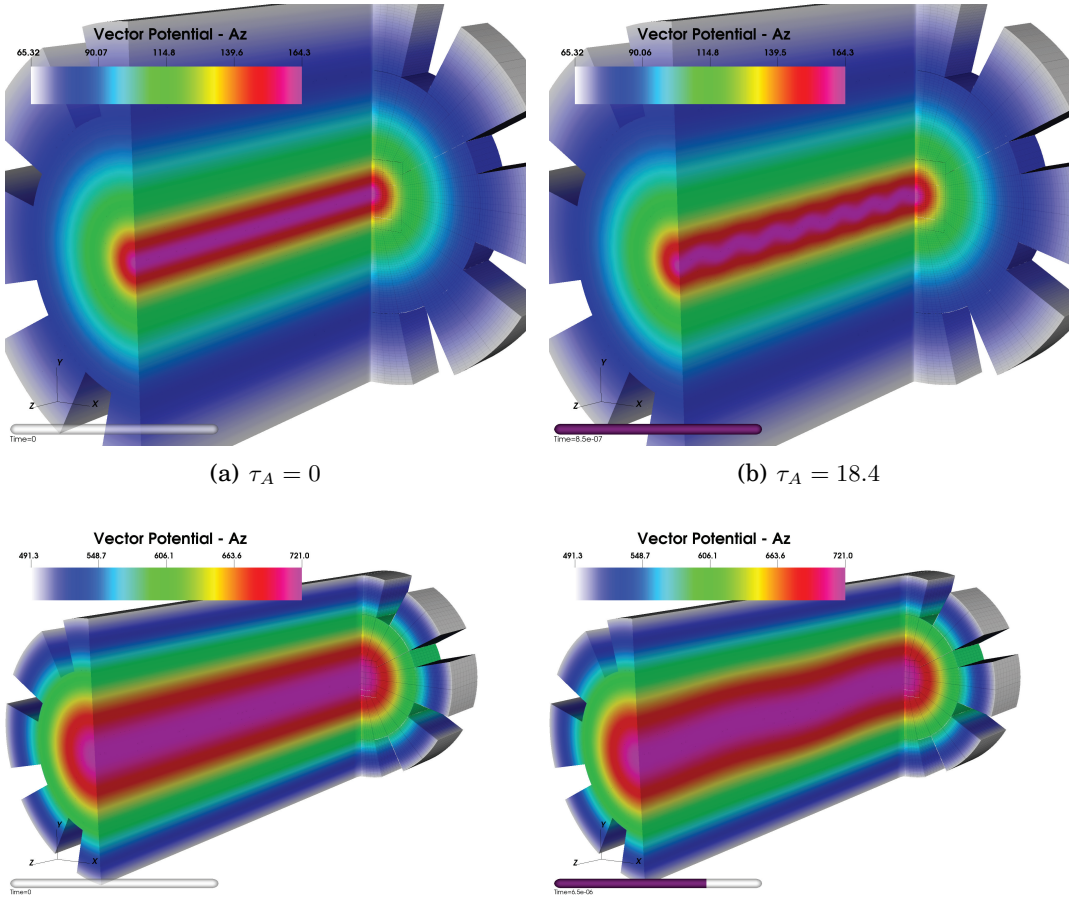
Examining the pressure at the radial boundary shows a more concise distinction between the case with and without extrusions. Table 8.8 shows a comparison of pressure values at the axial kink wavelength midpoint at two azimuthal locations for both the case with and without extrusions. The two azimuthal values are $\theta_{gap} = \pi/2$, and $\theta_{rod} = 3\pi/8$ and are locations for the center of a 'gap' and 'rod' respectively. The 'gap' location is where one of the extrusions is centered, and the 'rod' is centered at one of the locations where there are no extrusions. It is clear that the addition of the extrusions reduces the plasma pressure experienced at both the 'gap' and 'rod' radial wall locations. As the pinch size increases, the ratio of the pressure at the 'gap' to 'rod' locations at first increases, and then decreases with larger pinch sizes. When the pinch size is a larger fraction of the cylindrical volume, the wall plays a more important role. Figure 8.13 shows pressure slices at the midpoint of the kink axial wavelength, at time $\tau_A = 23.2$. It can be seen that for the smaller pinch sizes, the pressure profile is more



(a) $\tau_A = 0$ (b) $\tau_A = 18.4$

(c) $\tau_A = 0$ (d) $\tau_A = 17.6$

Figure 8.10: Pressure p in an extrusion cylinder with a characteristic plasma pinch radius of $a = 1.0$ cm (a), (b) and $a = 8.0$ cm (c),(d). Data are repeated 8 times for the $a = 1.0$ cm case and 2 times for the $a = 8.0$ cm case in the axial direction.



(a) $\tau_A = 0$ (b) $\tau_A = 18.4$

(c) $\tau_A = 0$ (d) $\tau_A = 17.6$

Figure 8.11: Axial vector potential A_z in an extrusion cylinder with a characteristic plasma pinch radius of $a = 1.0$ cm (a), (b) and $a = 8.0$ cm (c),(d). Data are repeated 8 times for the $a = 1.0$ cm case and 2 times for the $a = 8.0$ cm case in the axial direction.

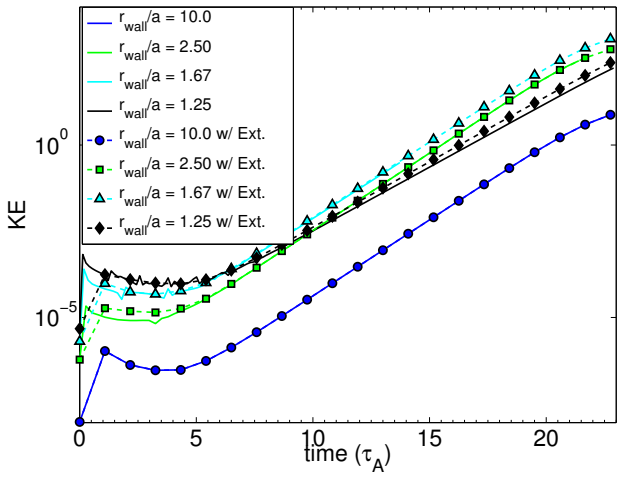
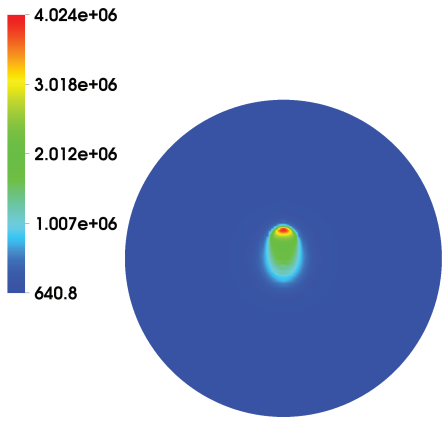


Figure 8.12: Normalized kinetic energy computed from HiFi simulations for various values of r_{wall}/a while holding constant $ka = \pi/2$ for geometries with and without extrusions.

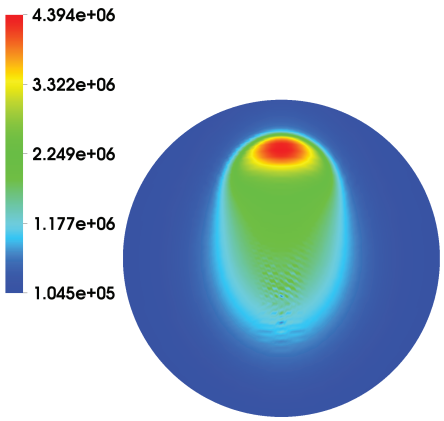
elongated, and for the larger pinch sizes the profile is broadened. When extrusions are added to the radial boundary, the wall effects diminish slightly, but then for larger pinch sizes the wall stabilization returns. With larger pinch sizes, the plasma has more contact with other 'rod' locations, which provide wall stabilization. Figure 8.14 shows pressure slices for the case with extrusions. Notice that for the case with $a = 6.0$ cm (Figure 8.14(c)) the pinch size is large enough to have interactions with the wall and extrusions, but not broad enough to interact with many of the rod locations. In this case the wall stabilization effect is the smallest. It is expected that this will vary depending on the width and frequency of the extrusions compared to the pinch size.

8.5 Inclusion of Shear Flow for Stabilization

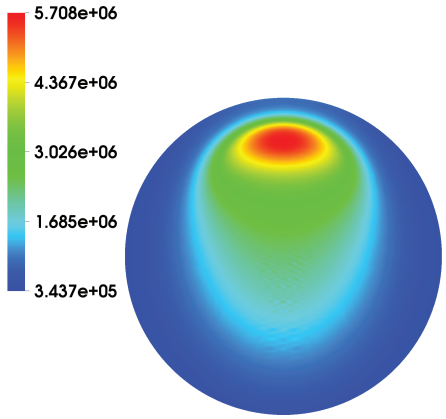
Pure Z-pinches are plagued by both the $m = 0$ sausage, and $m = 1$ kink mode instabilities [77]. Stabilizing the modes by conventional techniques such as controlling the pressure profile, having a close-fitting conducting wall or the addition of an axial field have drawbacks. Flow shear can stabilize these MHD modes and do not have the



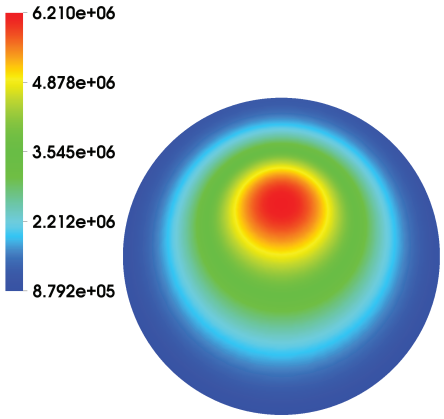
(a) $a = 1.0$ cm



(b) $a = 4.0$ cm

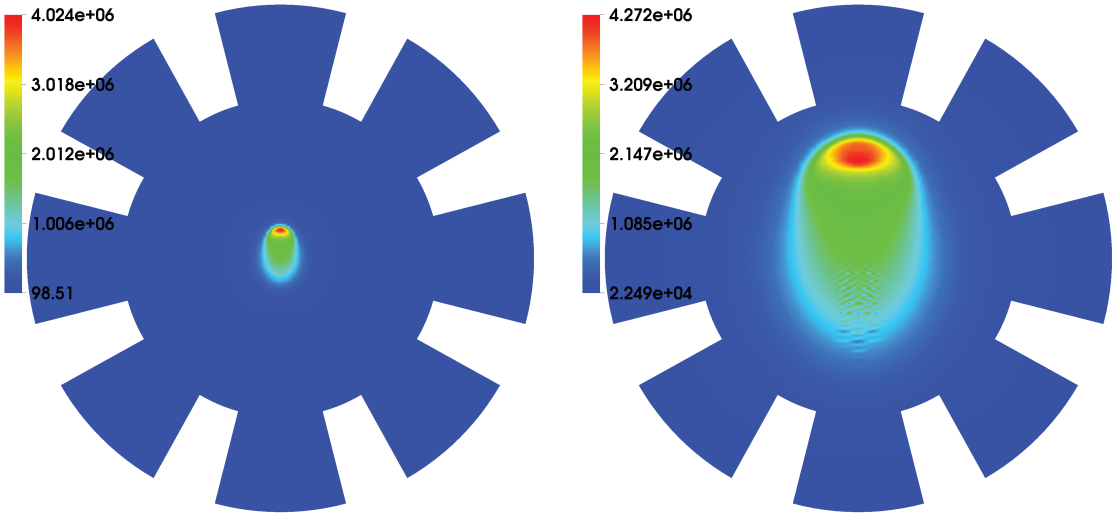


(c) $a = 6.0$ cm

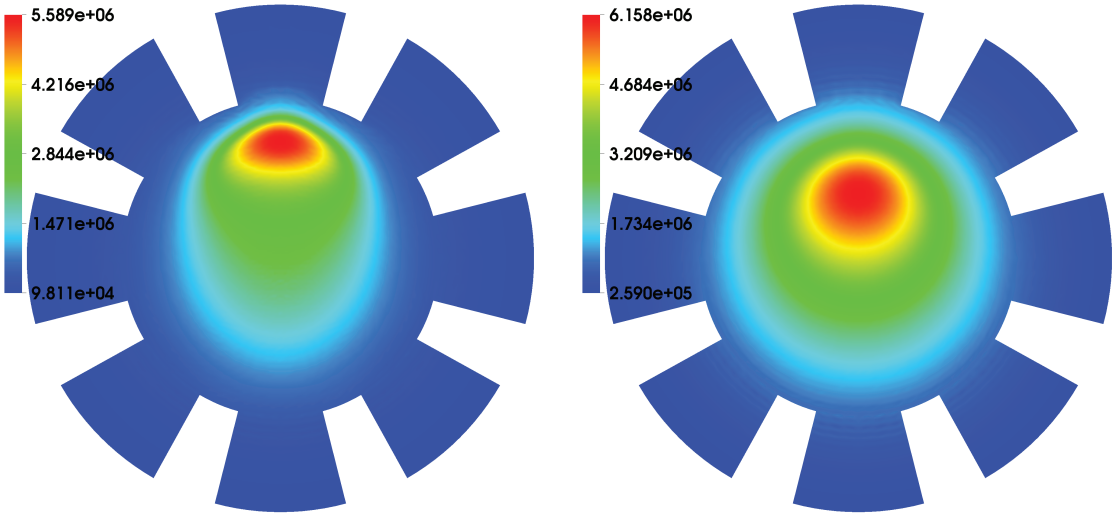


(d) $a = 8.0$ cm

Figure 8.13: Pressure profiles at the midpoint of the axial kink wavelength for characteristic plasma sizes of $a = 1.0$, 4.0, 6.0, and 8.0 cm while keeping $ka = \pi/2$ using a cylindrical cross section. All figures are at time $\tau_A = 23.2$ normalized by the axial wavenumber times the Alfvén speed kV_A.



(a) $a = 1.0$ cm (b) $a = 4.0$ cm



(c) $a = 6.0$ cm (d) $a = 8.0$ cm

Figure 8.14: Pressure profiles at the midpoint of the axial kink wavelength for characteristic plasma sizes of $a = 1.0$, 4.0, 6.0, and 8.0 cm while keeping $ka = \pi/2$ using a cylindrical cross section with extrusions added to the radial wall. All figures are at time $\tau_A = 23.2$ normalized by the axial wavenumber times the Alfvén speed kV_A.

Table 8.8: Pressure values for different pinch sizes a at the kink wavelength midpoint at 'gap' and 'rod' azimuthal locations. Ratios of the pressure at the 'gap' and 'rod' locations are also shown. All results are for the case with $ka = \pi/2$ at a time $\tau_A = 23.2$, which is the end of the linear growth phase.

a (cm)	$p_{\theta_{gap}}$	$p_{\theta_{gap}}$ (ext.)	$p_{\theta_{rod}}$	$p_{\theta_{rod}}$ (ext.)	$\frac{p_{\theta_{gap}}}{p_{\theta_{rod}}}$	$\frac{p_{\theta_{gap}}}{p_{\theta_{rod}}}$ (ext.)
1.0	0.6441e+3	0.6249e+3	0.6432e+3	0.6438e+3	1.0014	0.9706
4.0	0.3071e+6	0.2349e+6	0.2252e+6	0.1541e+6	1.3637	1.5243
6.0	1.0129e+6	0.9917e+6	0.8128e+6	0.5171e+6	1.2462	1.9178
8.0	1.2198e+6	1.2039e+6	1.1812e+6	1.0518e+6	1.0327	1.1446

drawbacks [67, 79, 82].

In the ZaP experiment, the shear flow profile is created during the formation of the pinch. Figure 8.3 shows a schematic of this formation procedure. In the experiment a strong, but very fine shear layer forms near the pinch radius. The profile has an approximate mathematical shape of

$$v_z' \sim v_s/\rho(r/a)^6 \tag{8.29}$$

where r is the radial coordinate and a is the characteristic plasma pinch radius, v_s is the magnitude of the shear flow velocity, and ρ is the fluid density. Figures 8.15 and 8.16 show the evolution of the out of plane momentum ρv_z with a relatively small, $\kappa = v_z'/kV_A = 0.0013$ and slightly larger $\kappa = v_z'/kV_A = 0.0130$ amount of shear flow respectively. It is clear that for a small amount of flow, the shear profile is disrupted and the pinch is not stabilized. Conversely, when the shear flow is significant enough, the profile provides stability to the pinch. This agrees with theory where marginal stability against the $m = 1$ mode is seen as κ approaches 0.10 for pinches with no wall stabilization [67]. Figures 8.17 and 8.18 show the axial current density evolution for the same conditions demonstrating when there is insufficient shear flow, and stabilization when when shear flow is sufficient. Figure 8.19 is a plot of the kinetic energy for the case of no shear flow, and three different magnitudes of shear flow while holding the wall radius to characteristic plasma pinch radius ratio fixed $r_{wall}/a = 10$. With

$r_{wall}/a = 10$ it resembles the size of the pinch in the ZaP experiment.

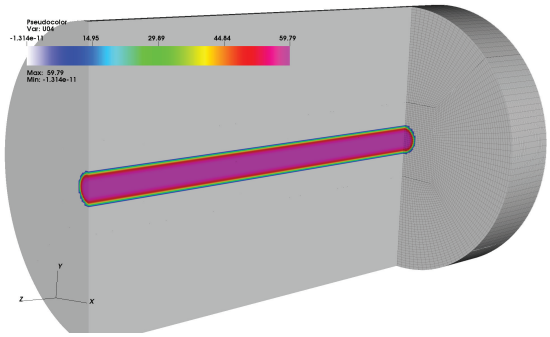
8.6 ZaP Experimental Results

The ZaP experiment operates with a plasma pinch size of approximately $a = 1.0$ cm, and an outer electrode wall radius of $r_{wall} = 10$ cm, giving a ratio $r_{wall}/a = 10$. This ratio is well beyond the wall stabilization limits predicted by the linear stability analysis described in Section 8.3 and the numerical simulation results in Section 8.4, suggesting that the pinch is not wall stabilized. When parts of the outer electrode are removed in the 'rod' configuration described in Section 8.2, it is expected the effect will be minimal due to the fact that the pinch is not wall stabilized. Both the hypothesis that the pinch is not wall stabilized, and that the 'rod' configuration will not significantly impact the pinch stability are backed by experimental data from the ZaP experiment.

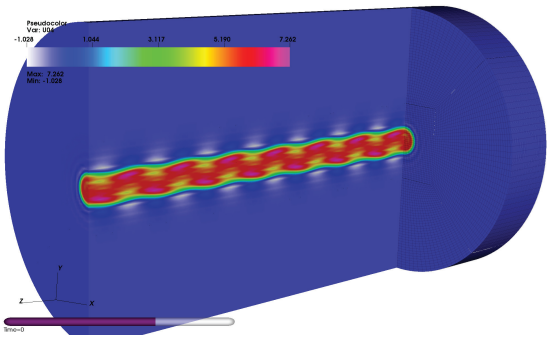
8.6.1 Experimental Evidence of Z-Pinch Stability with and without 'rod' Electrode Configurations

Data from the ZaP experiment show that both with and without the 'rod' configuration a stable pinch is maintained in the device. Figure 8.20 shows Fourier mode data of the pinch magnetic field at both the $z = 0$ cm and $z = 35$ cm locations. Both $m = 0$ (B_0) and the normalized $m = 1$ (B_1/B_0) mode data for each location are shown. The data show that both with and without the 'rod' electrode configurations the magnetic mode data is similar in magnitude and demonstrate similar stability properties. The horizontal green line in the plots for the the $m = 1$ data signifies the ZaP stability definition of $B_1/B_0 \leq 0.2$. When the mode data are below the stability definition it signifies the pinch is well centered and does not have significant portion outside this radial extent. For both with and without the 'rod' section, the pinch has a similar average magnetic field B_0 and show similar stability properties for an same extended period of time of about 30 μs to 85 μs. The similarity of the data suggests the pinch has similar characteristics for both configurations.

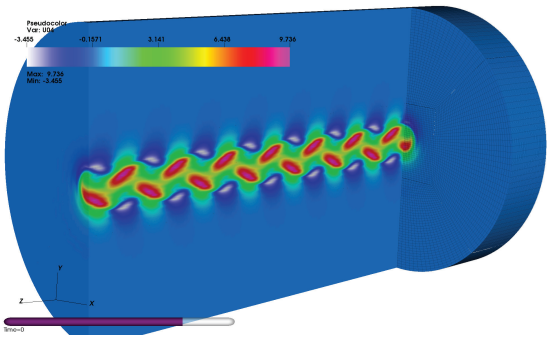
The interferometry data and magnetic probe array data from the 'rod' location in-



(a) $\tau_A = 0$

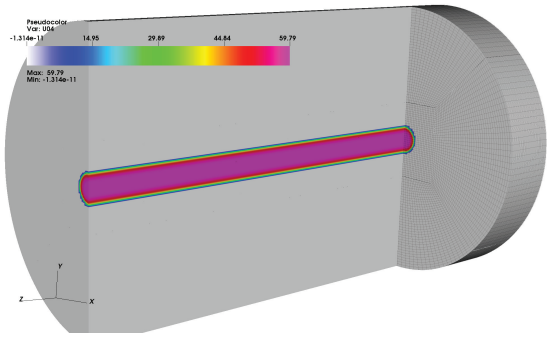


(b) $\tau_A = 1.60$

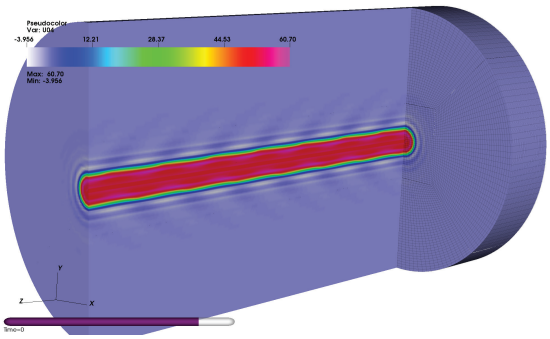


(c) $\tau_A = 1.90$

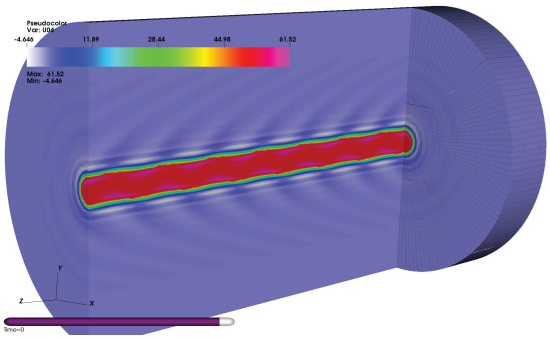
Figure 8.15: Out of plane momentum ρv_y with a small amount of shear flow $\kappa = v_z'/kV_A = 0.0013$ at $\tau_A = 1.6$, and $\tau_A = 1.9$. Data are repeated 8 times in the axial direction.



(a) $\tau_A = 0$

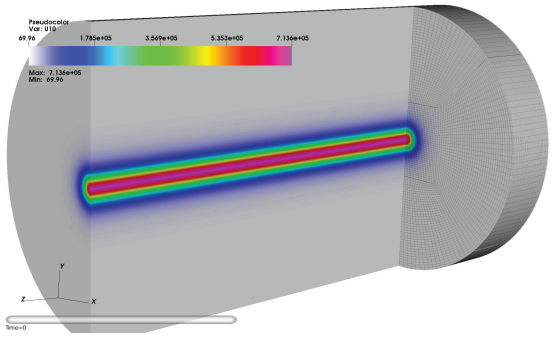


(b) $\tau_A = 4.5$

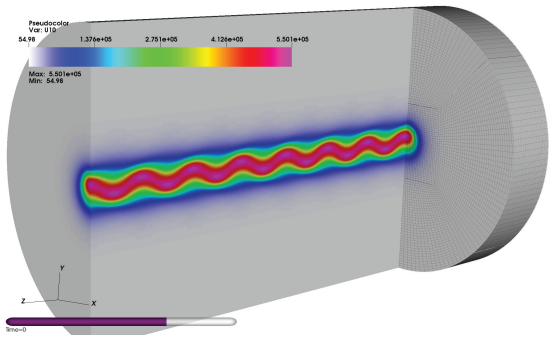


(c) $\tau_A = 5.0$

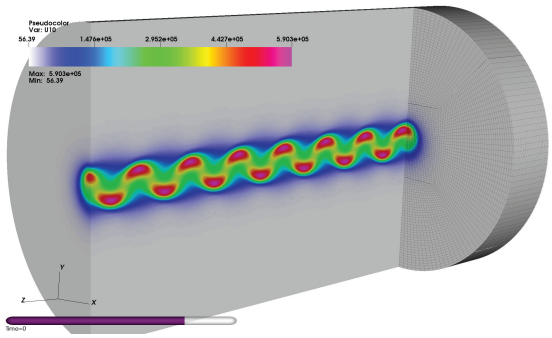
Figure 8.16: Out of plane momentum ρv_y with a larger amount of shear flow $\kappa = v_z'/kV_A = 0.0130$ at $\tau_A = 4.5$, and $\tau_A = 5.0$. Data are repeated 8 times in the axial direction.



(a) $\tau_A = 0$

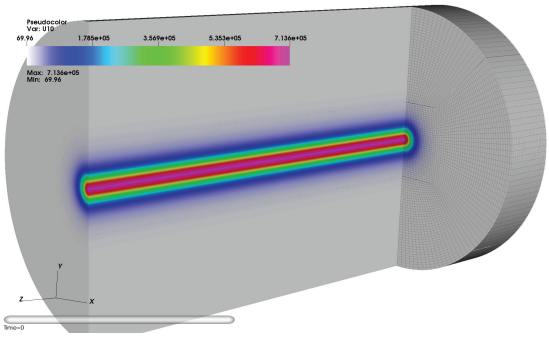


(b) $\tau_A = 1.6$

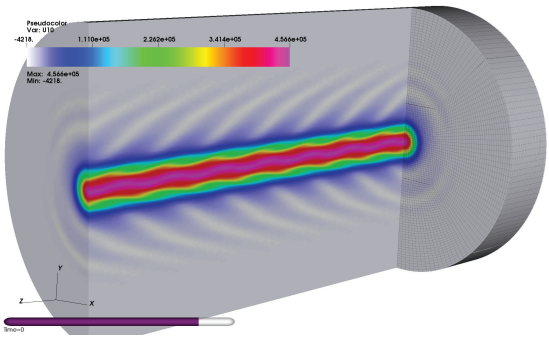


(c) $\tau_A = 1.9$

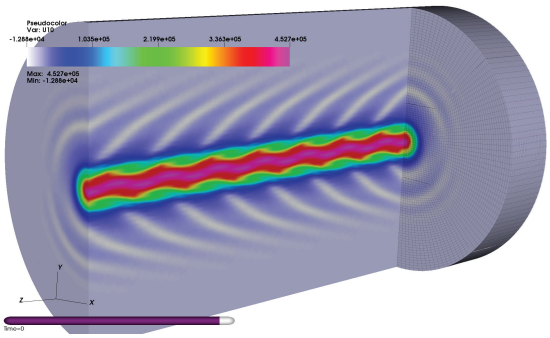
Figure 8.17: Axial current density J_z with a small amount of shear flow $\kappa = v_z'/kV_A = 0.0013$ at $\tau_A = 1.6$, and $\tau_A = 1.9$. Data are repeated 8 times in the axial direction.



(a) $\tau_A = 0$



(b) $\tau_A = 4.5$



(c) $\tau_A = 5.0$

Figure 8.18: Axial current density J_z with a larger amount of shear flow $\kappa = v_z'/kV_A = 0.0130$ at $\tau_A = 4.5$, and $\tau_A = 5.0$. Data are repeated 8 times in the axial direction.

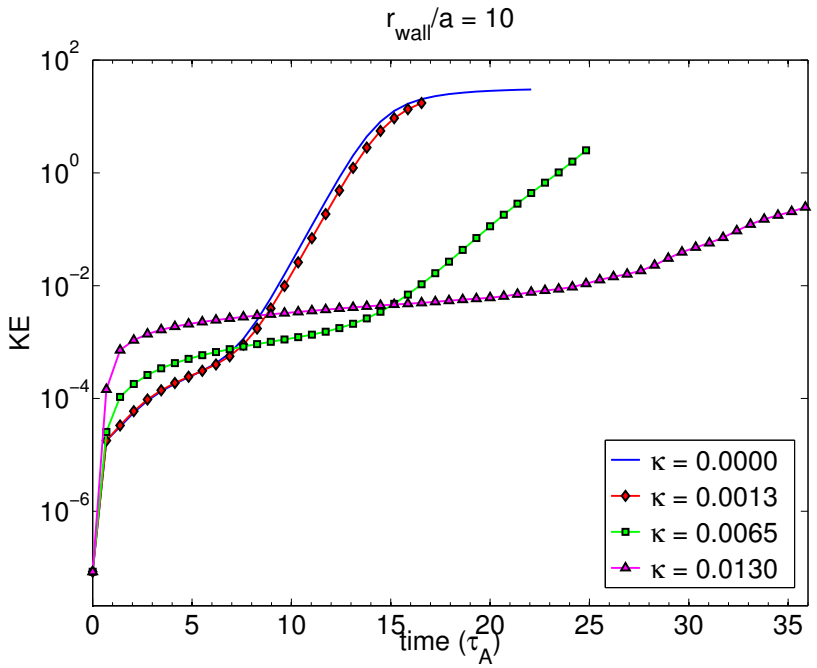
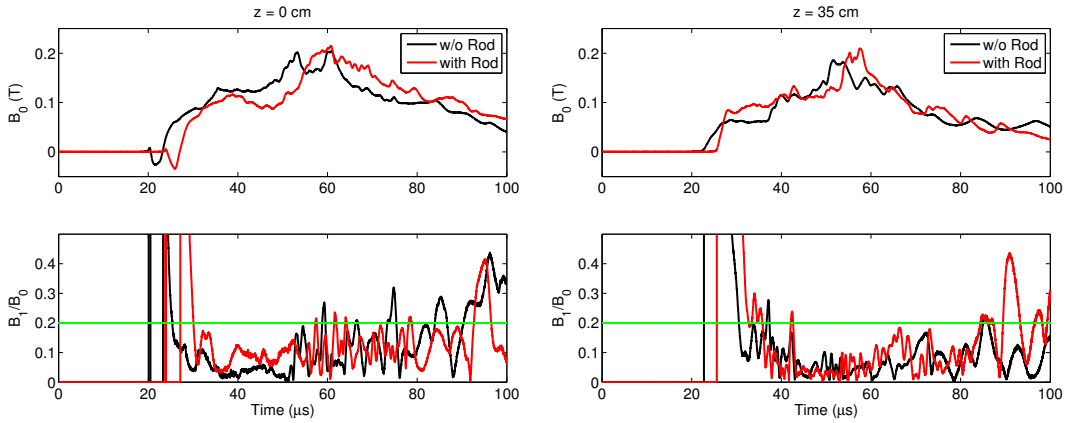


Figure 8.19: Normalized kinetic energy computed from HiFi simulations for various amounts of normalized shear flow $\kappa = v_z'/kv_A$ while holding r_{wall}/a constant.



(a) $z = 0$ cm (b) $z = 35$ cm

Figure 8.20: Fourier mode data from the azimuthal probe arrays located at (a) $z = 0$ cm and (b) $z = 35$ cm. The top figure for each location represents the average magnetic field B_0 as a function of time for cases with and without the 'rod' electrode configuration using similar operating conditions. The bottom plot represents the normalized $m = 1$ mode (B_1/B_0). The horizontal green line indicates the stability definition for the ZaP experiment.

dicate that a coherent pinch forms in the region. Figure 8.21 shows magnetic mode data for the $m = 0$ and $m = 1$ modes as well as interferometry for the same period of time in a similar axial location. The interferometry data show line integrated density information both centered and at a point laterally 1.7 cm away. It can be seen that the densities begin at the same value, and at about 35 μs diverge from each other. The divergence signifies a density gradient and that a coherent pinch has formed. The densities remain separated for a significant amount of time, indicating a stable period. The magnetic mode data agree with the interferometry result and also suggest a stable pinch during the same period of time.

The fast framing optical images shown in Figure 8.22 are consistent with both the magnetic field and density data. In Figure 8.22 a time series of optical images are shown. In the first frame (a) taken at time $t = 35$ μs a relatively uniform light emission is seen, indicating a coherent pinch has not yet formed. The interferometry data at this time also show that a pinch has not yet formed. Optical images from later times

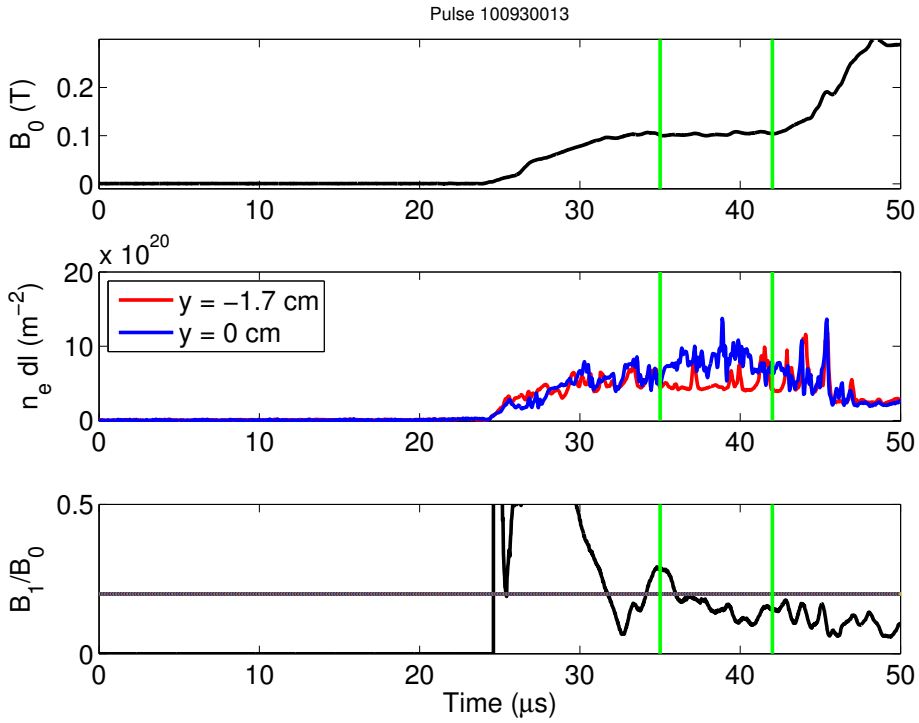


Figure 8.21: Fourier mode data from the azimuthal probe array located at $z = 35$ cm and 2-chord interferometry data from a $z = 40$ cm location. The time is from $t = 0$ to 50 μs with the vertical green lines representing the beginning and end of the fast framing optical images seen in Figure 8.22

clearly show that a coherent pinch has formed and also agree with the interferometry and magnetic mode data.

Further Fourier mode data also suggests that in the region with the 'rod' configuration a stable pinch is maintained. Figure 8.23 shows Fourier mode data for azimuthal probe arrays located at $z = 35$ cm and $z = 70$ cm, which spans the 'rod' section. The data show an average magnetic field B_0 and normalized $m = 1$ mode B_1/B_0 that are consistent with each other. Each of the traces in time show a stable pinch from about 40 μs to about 90 μs. These data also suggest that the 'rod' electrode does not impact the stability of ZaP plasma pinches.

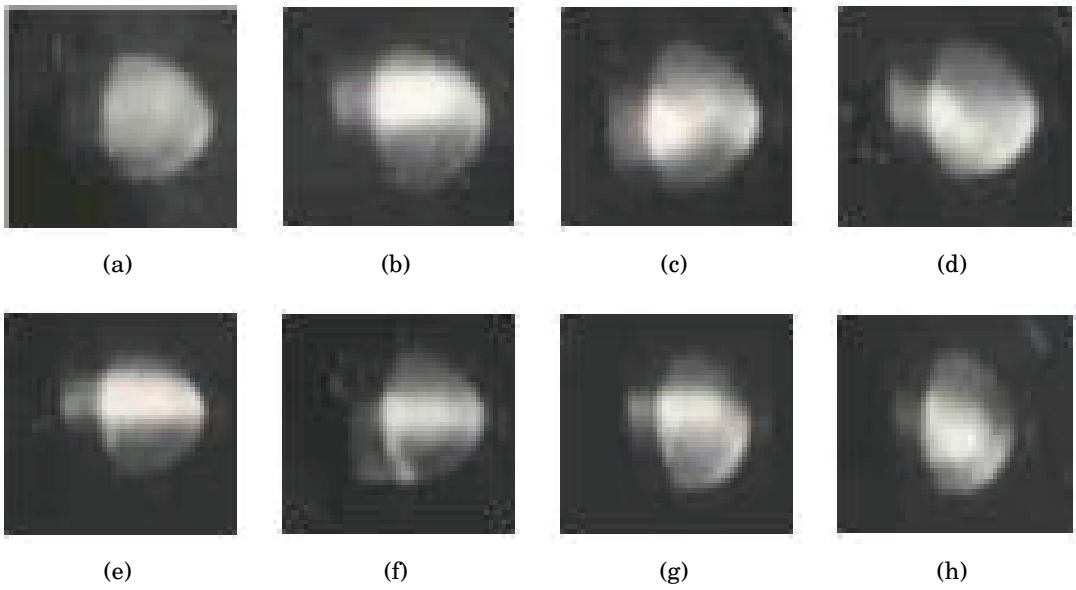


Figure 8.22: Visible light images from fast framing camera from 35 μs to 42 μs with a 1 μs frame rate taken in the 'rod' electrode section of the pinch. The images correspond to the data shown in Figure 8.21.

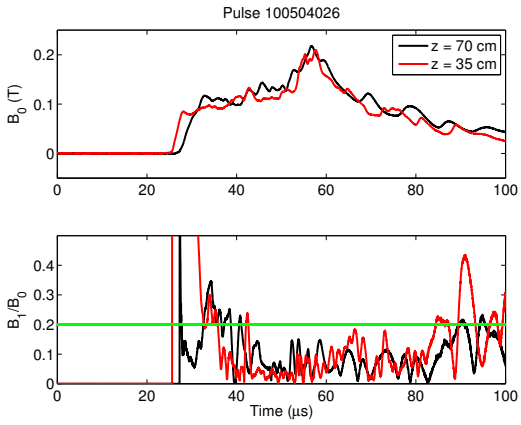


Figure 8.23: Fourier mode data from the azimuthal probe arrays located at $z = 35$ cm and $z = 70$ cm locations (upstream and downstream the 'rod' section). The magnetic data indicates upstream and downstream have similar behavior throughout the pulse. The data also indicates that the stability (quiescent) period is nearly the same for both locations, suggesting axial uniformity of the pinch.

8.7 Z-Pinch Simulation Conclusions

The Z-pinch simulations performed agree with the theoretical results obtained by a linear stability analysis, which is a promising verification of the HiFi code and the multi-block geometry framework. The error between theory and HiFi simulation results is mostly less then a few percent, and in the cases where it is higher a more temporally resolved simulation might improve the result. The simulations using the cylinder with extrusion geometry further verify the codes abilities and make use of a more complex non-axisymmetric multi-block geometry. The results obtained show how the extrusions affect the wall stabilization. No impact is seen when the pinch is small and far enough away from the boundary. When the pinch is large enough to interact with the boundary and on the order of the size of the extrusion gaps, the largest effect is found. The case with the addition of a shear flow profile similar to that measured in the ZaP experiment provides stabilizing effects as expected in theoretical analysis.

An experimental modification to the ZaP Flow Z-Pinch to evaluate the effects of a conducting wall on pinch stability is performed. An outer electrode section is manufactured that has about 70 % of the conducting wall surface removed. For ZaP conditions, the ratio of the pinch radius to the conducting wall radius is in the range in which simulations indicate the wall should have no effect. A coherent pinch structure is observed in the 'rod' electrode section that persists for lengths of time similar to that observed for a solid electrode section. Data from magnetic probe arrays indicate a stable, axially-uniform magnetic structure through the 'rod' electrode section. Optical images and interferometry data also indicate a well-defined pinch structure in the 'rod' electrode section that persists for times much longer than the instability growth time. The data all indicate that the removal of conducting wall material from the outer electrode does not have a significant effect on the stability of the Z-pinch.

Chapter 9

MESH DEFORMATION ANALYSIS APPLIED TO Z-PINCH SIMULATIONS

9.1 ZaP Z-Pinch as Practical Example

Analyzing different test meshes using mesh quality metrics and simple analytic problems as described in Chapter 6 is useful for establishing the feasibility of an *a priori* error check, but a more realistic problem is more convincing. The ZaP Z-pinch geometry is a good example problem that tests the mesh quality metrics with a realistic physical geometry and sufficiently complex nonlinear physics. A careful study of the meshes and resulting solutions demonstrates the *a priori* mesh quality analysis is useful for typical problems that computational physicists might encounter.

The approach chosen here is to study the effect mesh distortions have on a sufficiently complex and relevant problem. The ZaP Z-pinch with similar operating parameters to that of the experiment is used, as it represents a problem of interest to both computational physicists and experimentalists. A Z-pinch $m = 1$ kink mode is modeled similar to the analysis in Section 8.4 with a characteristic pinch size $a = 1.0$ cm. Several test meshes that have a varying amount of distortion are used, and the meshes are characterized using the same mesh metric analysis described in Chapter 6. The solutions are compared to a fully resolved solution on an undeformed mesh. This enables solutions at different degrees of deformation to be compared to the mesh quality metrics and an *a priori* assessment of the mesh can be established based on the metrics. This study also verifies that the analysis can be extended to nonlinear physics models with experimentally relevant parameters.

9.2 Choice of Pinch and Mesh Size

The ZaP Z-pinch experiment described in Section 8.4 has a characteristic pinch radius of approximately 1 cm and an outer electrode radius of 10 cm and these sizes are used to create the computational domain. The same cylindrical geometry as in Section 8.4 is used with a five block configuration. Each block has a resolution of $12 \times 12 \times 12$, with $np = 3$, giving an effective resolution of $nx \times ny \times nz = 36 \times 36 \times 12$, $np = 3$. The effective resolution is defined as the resolution across all blocks in the multi-block configuration. In the five block case, three blocks span each physical dimension in the

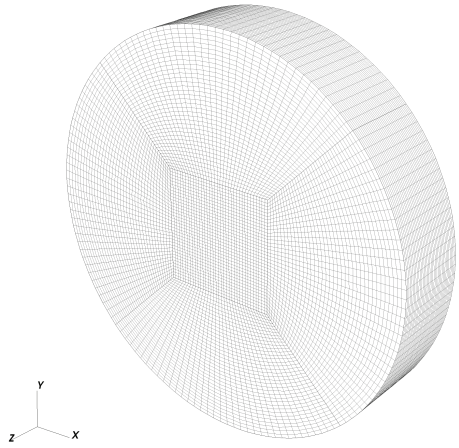
axial cross section giving $nx = ny = 3 \times 12$ elements. This resolution is sufficient to resolve all the necessary physics. For error norm comparison a very fine effective mesh resolution of $72 \times 72 \times 24$, with $np = 3$ is used. A periodic axial boundary condition is chosen with a constant $ka = \pi/2$, giving the cylindrical domain height of four times the characteristic plasma pinch size a. Figure 9.1 shows both the 3D meshes (a), (b) as well as a 2D cross section showing the detail of the mesh near the pinch (c), (d).

9.3 Meshes with a 'Random' Distortion

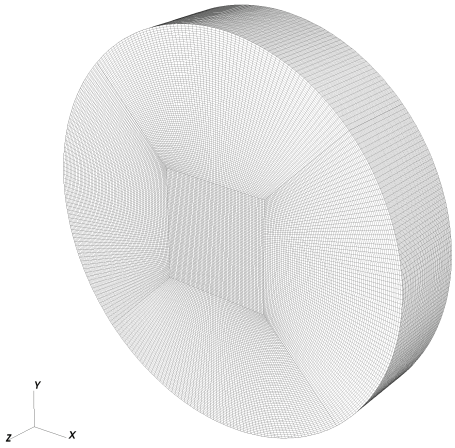
The Z-pinch $m = 1$ kink mode instability is a linear phenomenon, but the HiFi model evolves the full nonlinear physics. The pinch is initialized and given a small perturbation of the desired wavelength to kick off the growth of the instability. This allows the linear mode to grow while keeping the nonlinear terms small. However, noise in the axial direction can trigger growth of higher order modes. These higher order modes are undesirable in this case because the aim of this study is to analyze one particular mode structure solution. As a result of this issue, the problem is particularly sensitive to the distortions in the axial direction and a small amount of dissipation is necessary to suppress grid level noise, such that higher order modes do not grow.

A feature in the CUBIT [2] mesh generator is a 'randomized' mesh smoother. The smoother moves mesh nodes a random distance and direction up to a maximum tolerance set by the user. The tolerance is set using a parameter $rand$ that controls the degree of possible mesh distortions. It is a percentage measure of allowed random movement of node locations from the original uniformly spaced mesh. With a higher value of $rand$ the mesh smoother has the ability to move node locations further away from the uniform case. This feature is used to create meshes that have increasingly severe distortions that characterize many commonly encountered distortion types (e.g. stretch, shear, skew, etc). Several meshes with the same resolution, but a differing degree of the 'random' deformation are created and used to study the Z-pinch $m = 1$ kink mode problem.

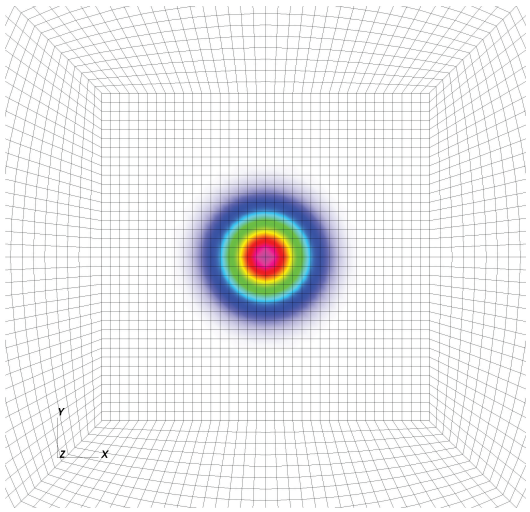
It does not make sense to have distortions in the axial direction that increase the



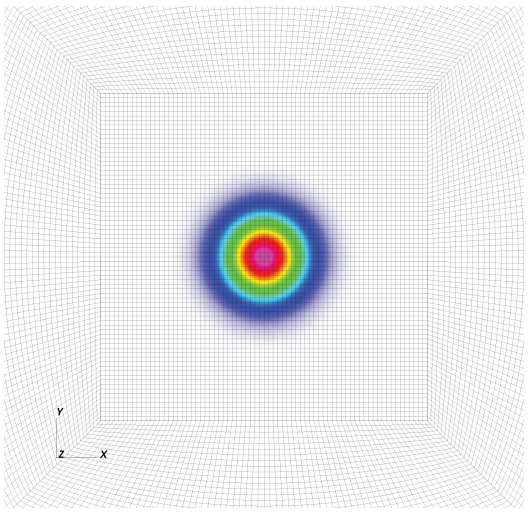
(a) $36 \times 36 \times 12$, $np = 3$



(b) $72 \times 72 \times 24$, $np = 3$



(c) $36 \times 36 \times 12$, $np = 3$



(d) $72 \times 72 \times 24$, $np = 3$

Figure 9.1: 3D mesh and cross sections of the radial dimension $x - y$ plane showing the fine mesh (a), (c) and very fine mesh (b), (d) used in the mesh deformation analysis of the Z-pinch $m = 1$ kink mode. The kinetic energy of the pinch is plotted on the mesh as reference to the size of the pinch.

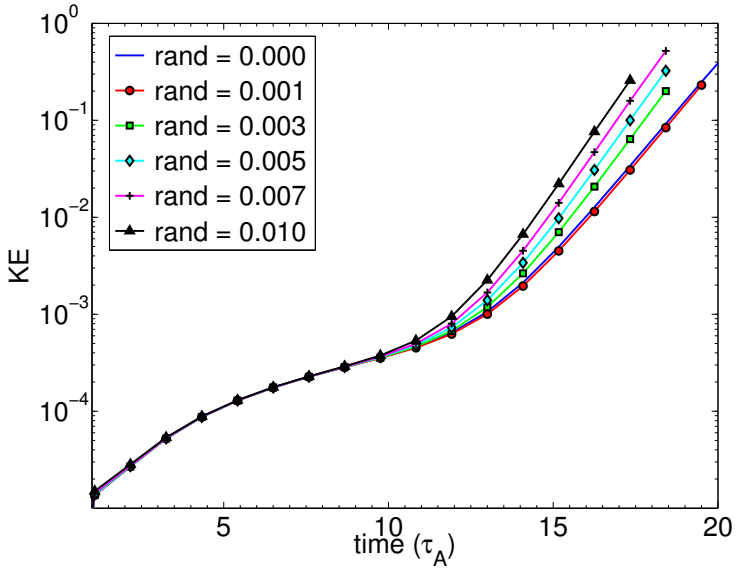
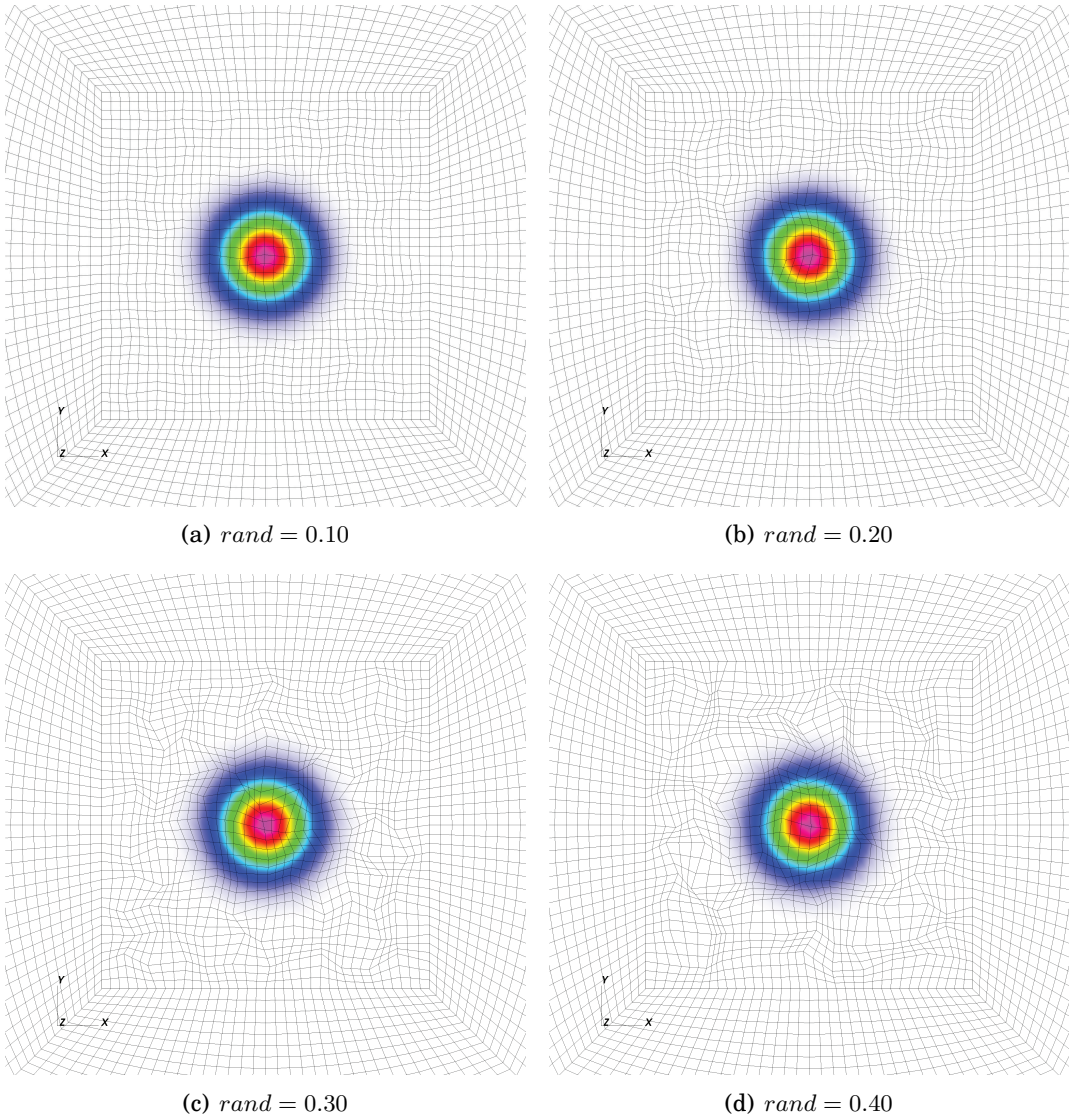


Figure 9.2: Kinetic energy versus time for a $m = 1$ kink mode simulation with varying amount of axial mesh distortion. The variable $rand$ is a percentage measure of allowed random movement of node locations from a uniformly spaced mesh.

likelihood of noise that can result in higher order mode growth. Figure 9.2 demonstrates this effect. Only a small amount of axial mesh distortion is necessary to impact the growth rate, since other higher order modes are non negligible due to the noise generated. Consequently, in the subsequent analysis, mesh deformations are applied only to the radial $(x - y)$ plane. Figure 9.3 shows mesh cross sections near the pinch center for various levels of the 'random' distortion. In each case the pinch is sufficiently resolved, although there are increasingly severe distortions that lead to errors in the solution.

9.4 Results

A Z-pinch $m = 1$ kink mode simulation is performed on an undeformed mesh as well as several meshes with a 'random' deformation. These solutions are compared to the solution computed on the very fine mesh (Figure 9.1(b), 9.1(d)). In order to ensure that the



(a) $rand = 0.10$ (b) $rand = 0.20$

(c) $rand = 0.30$ (d) $rand = 0.40$

Figure 9.3: Mesh cross sections of the radial ($x - y$) plane showing various levels of the 'random' mesh deformation. The kinetic energy of the pinch is plotted on the mesh as reference to the size of the pinch. The variable $rand$ is a percentage measure of allowed random movement of node locations from a uniformly spaced mesh.

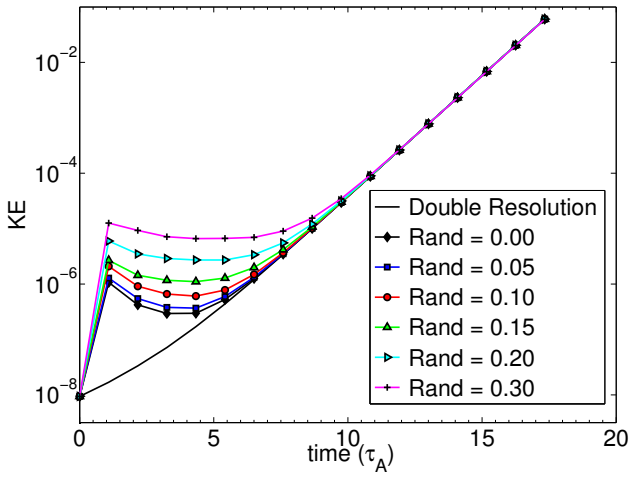


Figure 9.4: Normalized kinetic energy computed from HiFi simulations for various levels of a 'random' deformation in the $x - y$ plane. The meshes used have an effective resolution of $36 \times 36 \times 12$ with $np = 3$. A simulation with no deformation and double resolution in all dimensions ($72 \times 72 \times 24$ with $np = 3$) is shown for comparison and considered to be a fully resolved solution. The final time is well into the linear growth of the kink instability, but significantly before saturation and the nonlinear phase.

correct physical problem is solved, the kinetic energy is plotted for each of the meshes in Figure 9.4. The kinetic energy is computed by integrating kinetic energy over the entire simulation volume as a function of time. The final time is well into the linear growth of the kink instability, but significantly before saturation and the nonlinear phase. Notice that for all of the meshes with the exception of the fully resolved case, there is an initial jump in the kinetic energy. This jump is the solution adjusting to errors generated by the under resolved mesh, and is exacerbated by mesh distortions. Dissipative terms in the equations damp this initial jump and keep the simulation numerically stable. As the simulation progresses, the kinetic energy growth rates for all meshes converge to the same physical result.

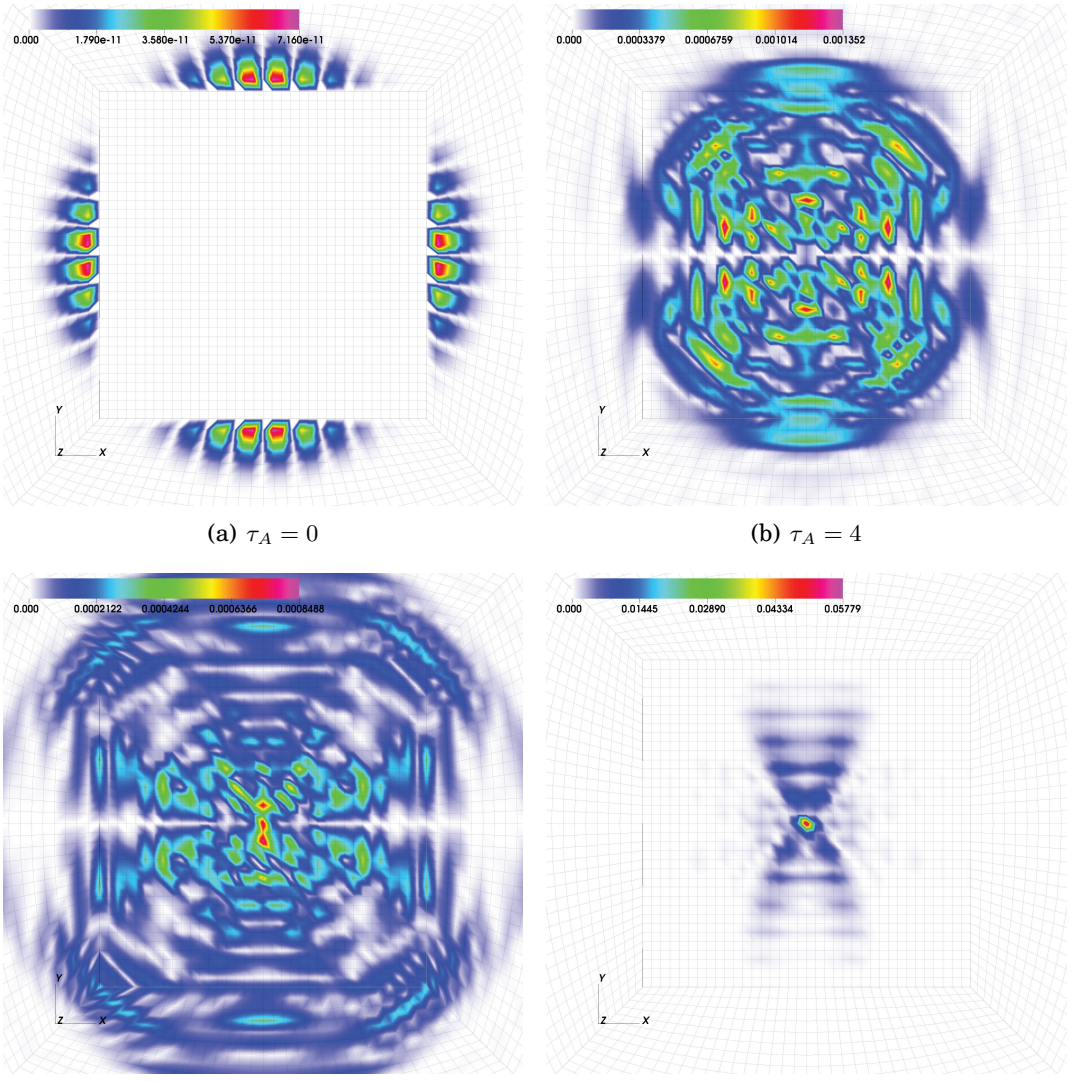
To explore the error associated with mesh distortions a L^∞ norm is calculated. Figures 9.5 through 9.9 show four time slices of a cross section of the L^∞ norm plotted on their respective meshes. Additionally the 3D structure of the L^∞ error norm can be

seen in Figures 9.10 and 9.11 for the cases of no mesh deformation and a random mesh deformation ($rand = 0.030$) respectively. Notice that in areas of high mesh deformation significant errors can result.

The L^∞ norm is useful for studying the absolute error associated with a solution on a mesh. It does not however, take into account the size of the mesh elements. The L^2 norm can be calculated by integrating the absolute error over the domain volume. This yields a quantitative result for the total global error of the solution. Figures 9.12 plots the L^2 error norm of the $\hat{y}$-momentum (ρv_y) in time for several of the different meshes with increasing amount of distortion. Additionally Figure 9.13 shows a semilog plot of the L^2 error norm of the kinetic energy in time for the same meshes. In both of these figures it is clear that having larger mesh distortions yields solutions with larger errors. This fact remains true throughout the duration of the simulation.

The aim of this study is not only to show how distortions impact the solution error, but to be able to quantify the error *a priori* from the mesh itself. Similar to the study in Chapter 6, various mesh metrics are computed for each of the meshes, and the results compared to the global error norms. Figure 9.14 shows a comparison of the computed mesh metrics to the normalized values of the global error norms for increasing degree of mesh deformation. Notice that the error norms follows the trend of the product of Knupp [39] metics well as in the analysis in Chapter 6.

Another important aspect of these results is the fact that the error metrics increase when moving from the deformation degree of $rand = 0.20$ to $rand = 0.25$. This is merely because the deformation applied is random, and by chance when creating the mesh, it did not have as significant distortions as the $rand = 0.20$ case. If several meshes with this amount of 'random' deformation were created, and the results for the error metrics averaged, it would be expected to become more monotonic with respect to the $rand$ parameter. Figure 9.14 shows how the f_{size} metric increases when moving from a deformation degree of $rand = 0.20$ to $rand = 0.25$, suggesting a better quality mesh. The simulation L^2 and L^∞ error norms from $rand = 0.20$ to $rand = 0.25$ confirm the more well behaved mesh predicted by the metrics.



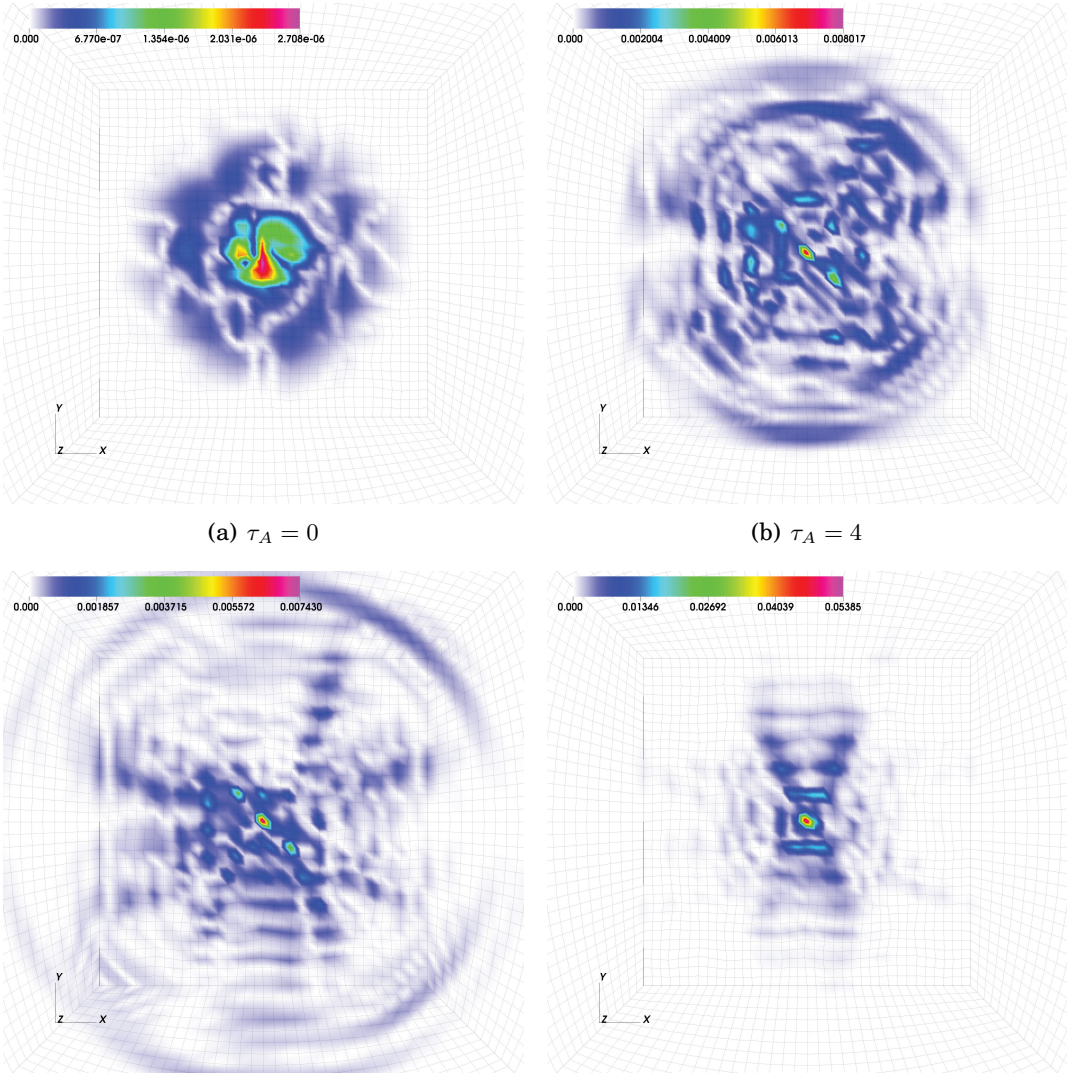
(a) $\tau_A = 0$

(b) $\tau_A = 4$

(c) $\tau_A = 8$

(d) $\tau_A = 16$

Figure 9.5: Time sequence of the cross sections of the radial dimension $x - y$ showing the L^∞ error norm for the case of no deformation.



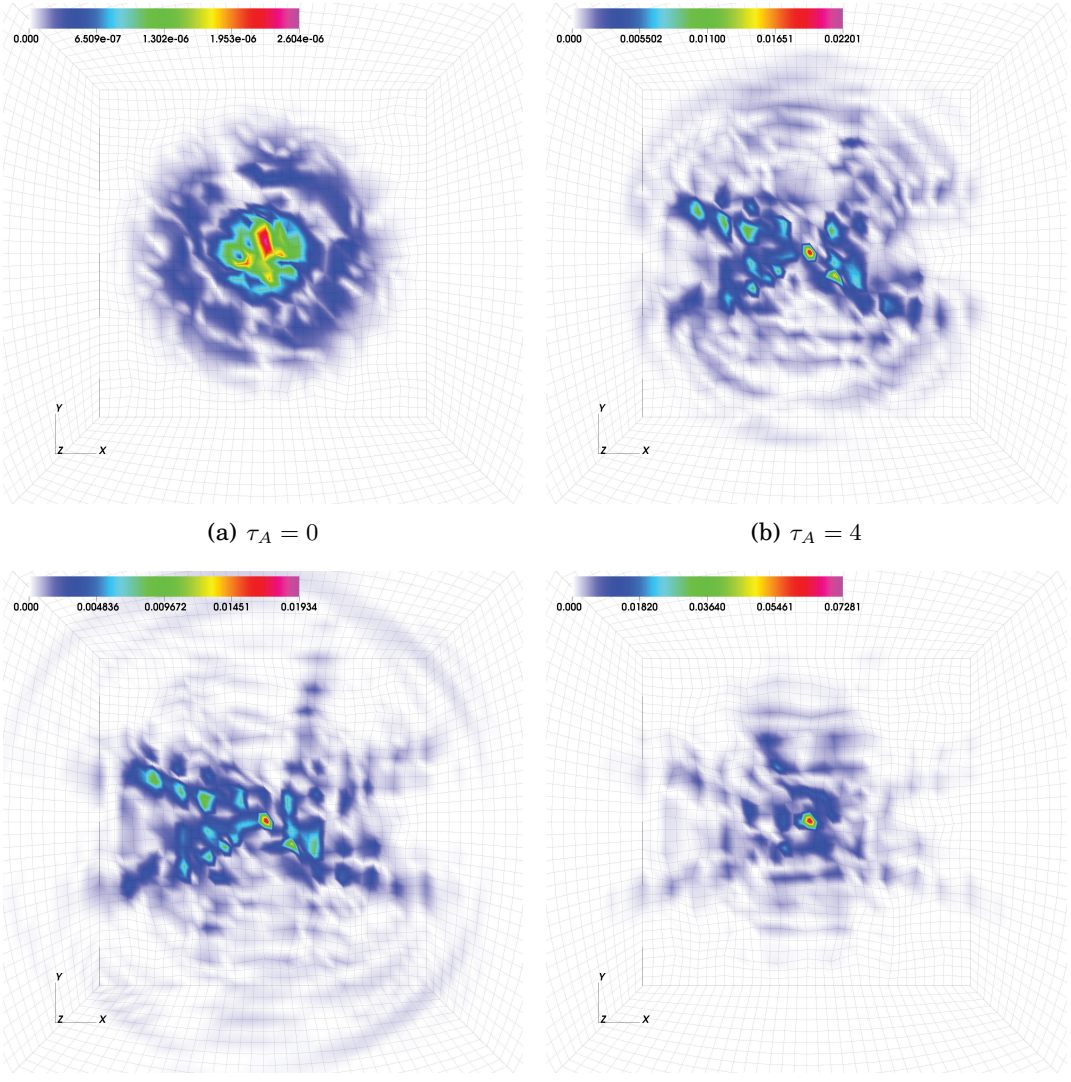
(a) $\tau_A = 0$

(b) $\tau_A = 4$

(c) $\tau_A = 8$

(d) $\tau_A = 16$

Figure 9.6: Time sequence of the cross sections of the radial dimension $x - y$ showing the L^∞ error norm for the case with a 'random' mesh deformation ($rand = 0.010$).



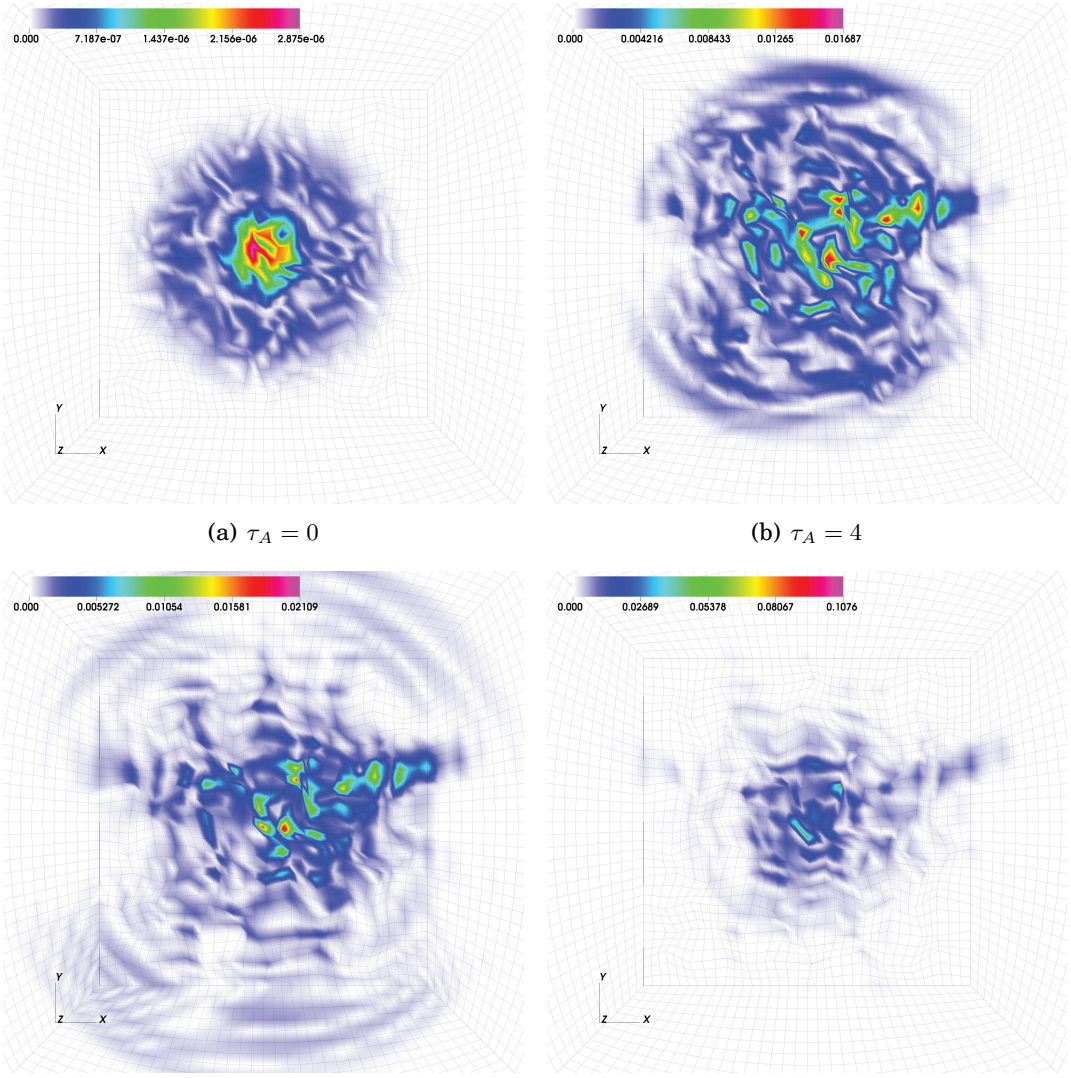
(a) $\tau_A = 0$

(b) $\tau_A = 4$

(c) $\tau_A = 8$

(d) $\tau_A = 16$

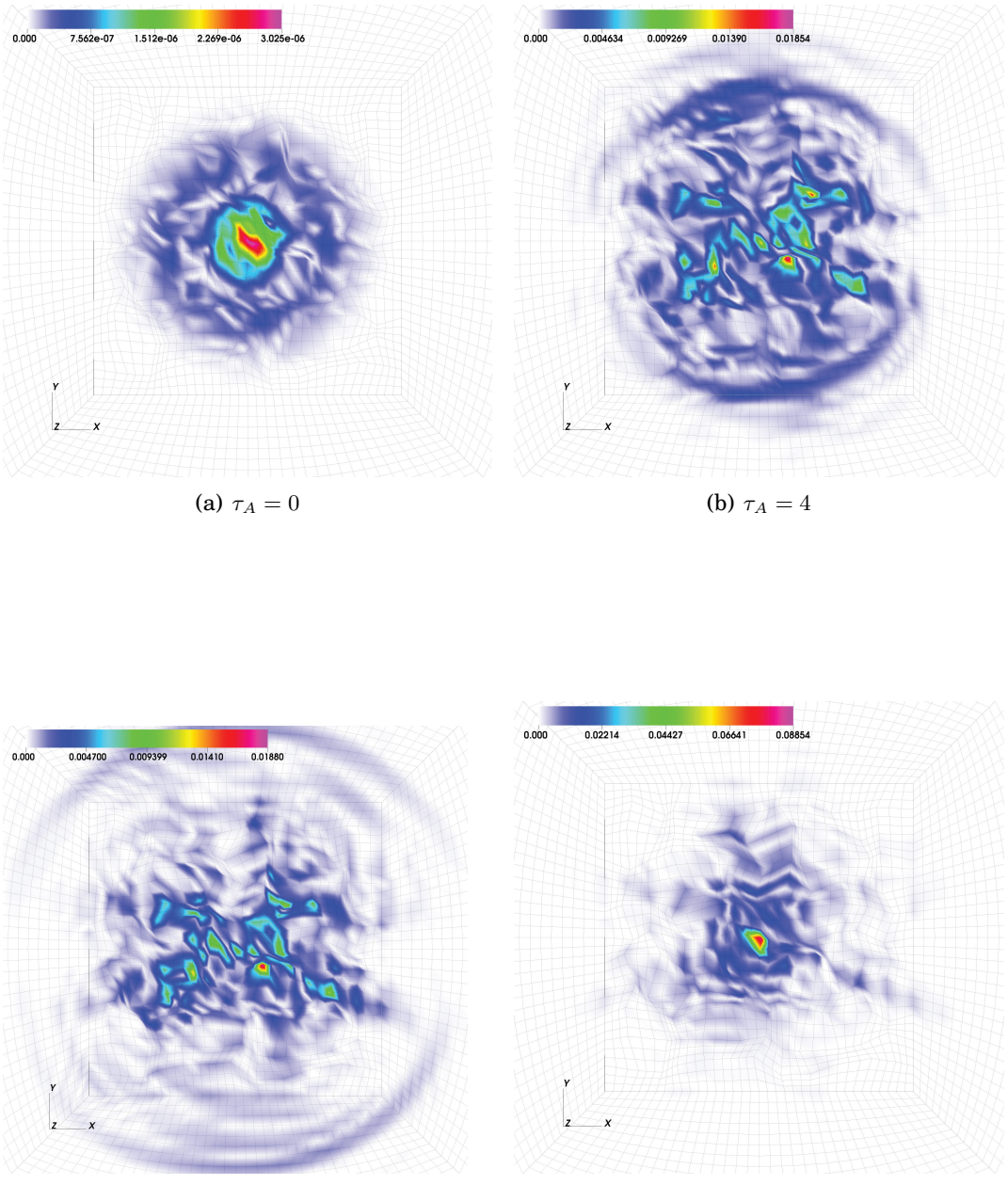
Figure 9.7: Time sequence of the cross sections of the radial dimension $x - y$ showing the L^∞ error norm for the case with a 'random' mesh deformation ($rand = 0.020$).



(a) $\tau_A = 0$ (b) $\tau_A = 4$

(c) $\tau_A = 8$ (d) $\tau_A = 16$

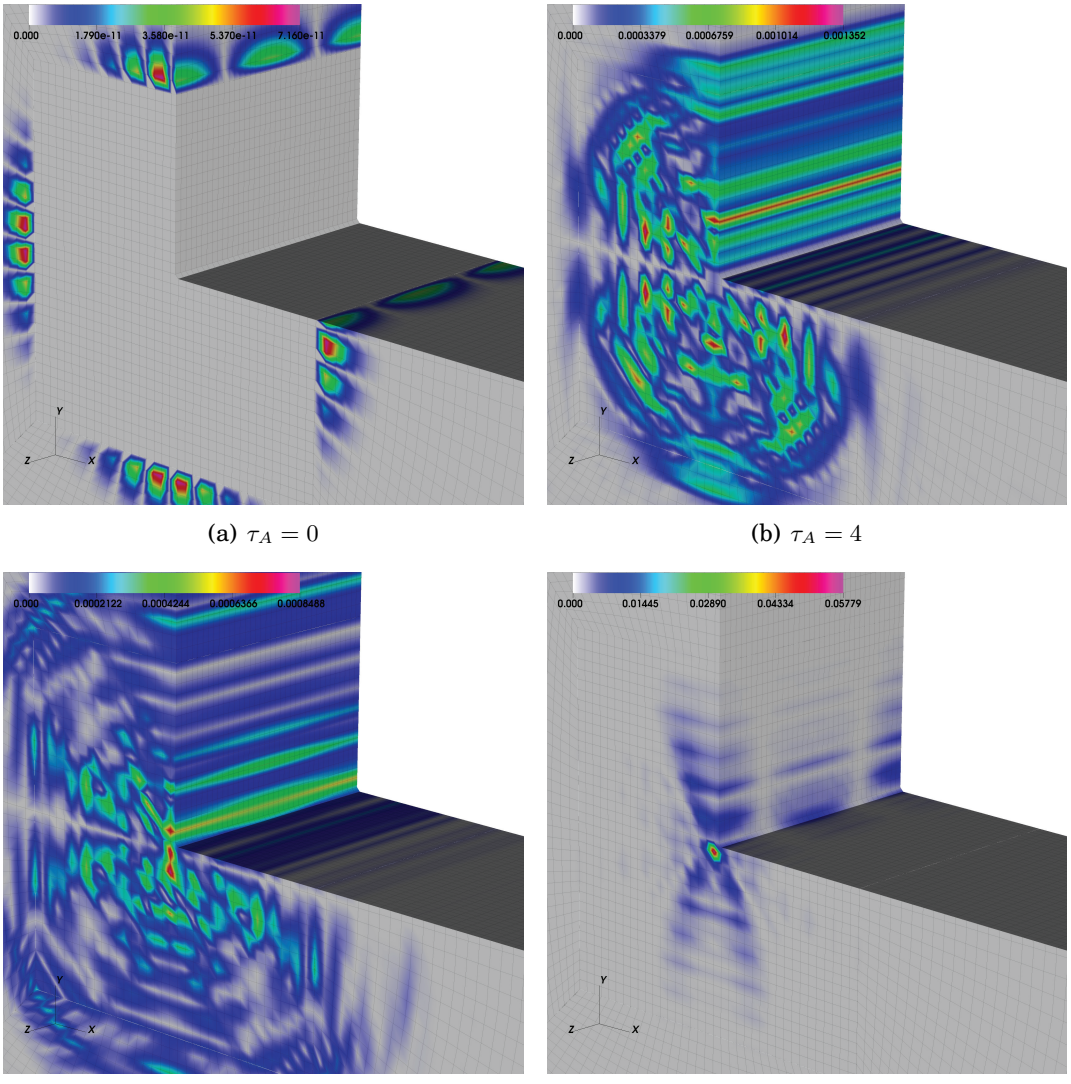
Figure 9.8: Time sequence of the cross sections of the radial dimension $x - y$ showing the L^∞ error norm for the case with a 'random' mesh deformation ($rand = 0.030$).



(a) $\tau_A = 0$ (b) $\tau_A = 4$

(c) $\tau_A = 8$ (d) $\tau_A = 16$

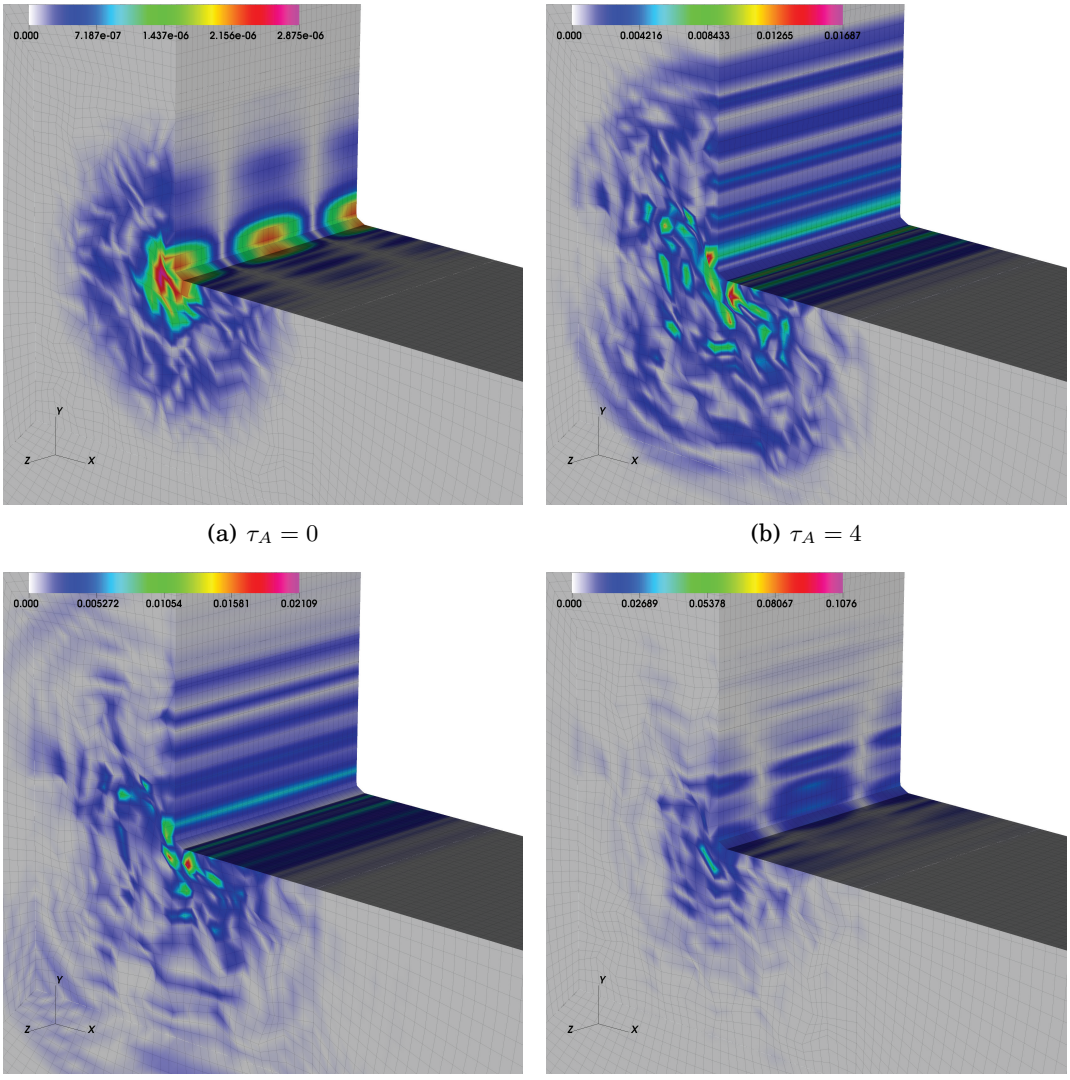
Figure 9.9: Time sequence of the cross sections of the radial dimension $x - y$ showing the L^∞ error norm for the case with a 'random' mesh deformation ($rand = 0.040$).



(a) $\tau_A = 0$ (b) $\tau_A = 4$

(c) $\tau_A = 8$ (d) $\tau_A = 16$

Figure 9.10: Time sequence the L^∞ error norm for the case of no deformation showing a cutaway of the 3D geometry.



(a) $\tau_A = 0$ (b) $\tau_A = 4$

(c) $\tau_A = 8$ (d) $\tau_A = 16$

Figure 9.11: Time sequence the L^∞ error norm for the case of a 'random' deformation ($rand = 0.30$) showing a cutaway of the 3D geometry.

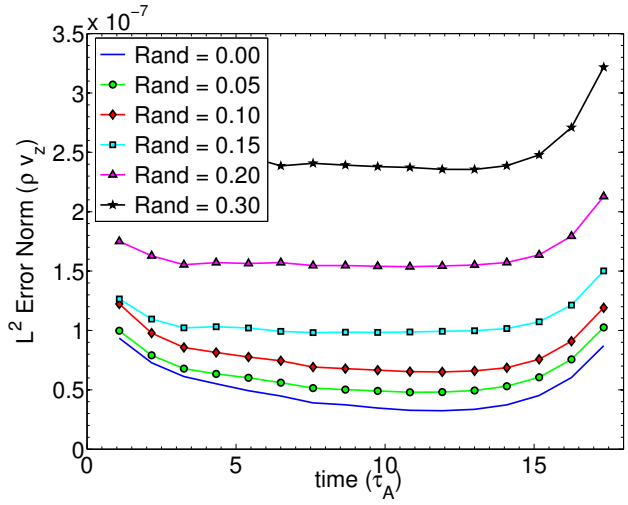


Figure 9.12: L^2 error norm of the y-momentum, ρv_y in time for a nonlinear simulation of a Z-pinch $m = 1$ kink mode simulation with varying levels of 'random' deformations in the $x - y$ plane. The meshes used have an effective resolution of $36 \times 36 \times 12$ with $np = 3$.

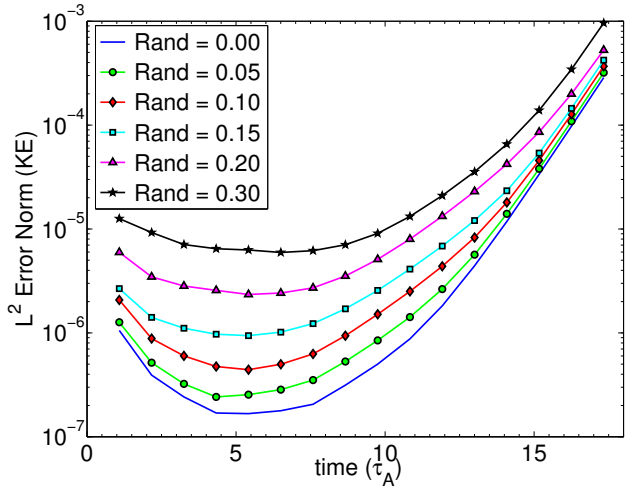


Figure 9.13: L^2 error norm of the kinetic energy in time for a nonlinear simulation of a Z-pinch $m = 1$ kink mode simulation with varying levels of 'random' deformations in the $x - y$ plane. The meshes used have an effective resolution of $36 \times 36 \times 12$ with $np = 3$.

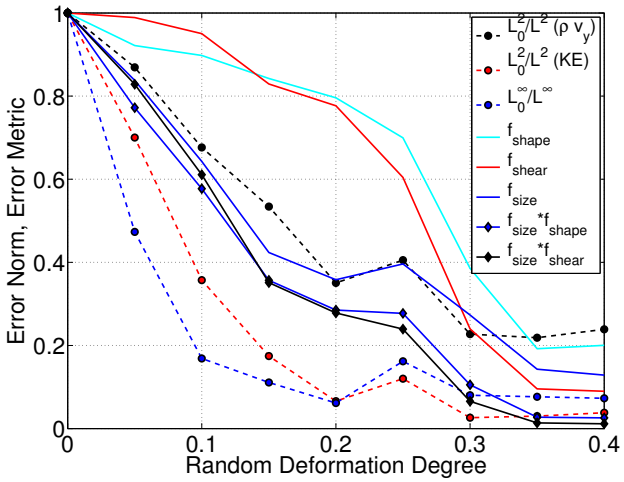


Figure 9.14: Inverse error norms and mesh quality metrics for varying 'random' deformation. Error norms and metrics are normalized to range from 1 to 0, where 1 is an undeformed element, and 0 is a degenerate element. Both the inverse L^2 norm and the inverse L^∞ norm are plotted. The Knupp f_{size}, f_{shape}, and f_{skew} as well as products of the Knupp metrics are plotted. This result is for a normalized time $\tau_A = 5.417$.

9.5 Conclusions

The technique here can be used in practice by computing the mesh metrics of a potential computational mesh, and then estimating the expected increase in solution error for those particular metrics. Consider the case where the degree of random deformation is $rand = 0.20$ (Figure 9.14). This correlates to an approximate metric value $f_{size}f_{shape} = 0.3$. Since inverse error is plotted the expected increase in error due to this metric value is $0.3^{-1} = 3.33$. The computed value of the normalized inverse L^2 error norm for this case is about 0.35, which corresponds to an error approximately 2.86 times the undeformed case. This result is close enough to be useful in predicting solution error and can be used as guide for determining *a priori* whether or not a mesh is of acceptable quality.

The results here are promising as it provides consistency between the different problems analyzed using this technique in chapter 6. Additionally it validates its via-

bility for use in realistic problems with experimentally relevant solution magnitudes.

Chapter 10

APPLICATION TO A HIT-SI GEOMETRY

A major motivation throughout the recent developments of multi-block geometric capabilities is to have the ability to model non-axisymmetric, and non-simply connected complex geometries. The HIT-SI geometry fits this description, and the ability to model HIT-SI with an accurate computational domain is a significant achievement.

All of the development advancements described in this dissertation are necessary to accurately and successfully model problems with the HIT-SI geometry. The geometry is sufficiently complex that a CAD drawing is needed for the geometric specification. Figure 10.1 shows a CAD drawing of the HIT-SI computational domain imported into CUBIT. Along with the ability to read the complex geometry, advanced mesh generation is needed. CUBIT will provide the necessary means of partitioning and meshing the geometry. The geometry will also require an unstructured multi-block description in order for HiFi to handle the domain. Additionally, since the geometry is complex, mesh distortions are likely, and a means of quantifying the severity of the distortions is necessary through the mesh quality metric analysis. All these pieces must be present to have a successful simulation, and makes the simulation of HIT-SI geometry a good milestone to meet in the development of a useful engineering design tool.

A detailed physics study of the HIT-SI experiment is not planned, but rather simulations that demonstrates all the code developments made that are described in this dissertation. A study of the HIT-SI experiment requires a significant amount of spatial and temporal resolution, and the amount of time and computational resources necessary to perform a study is out of the scope of this dissertation. Although, an accurate 3D HIT-SI geometry is represented in the multi-block HiFi framework and has been proven to work with MHD equations, and conducting wall boundary conditions. This is a major milestone and is intended to be the ground work for future detailed physics studies of HIT-SI with an accurate 3D representation.

10.1 HIT-SI Geometry Creation

In order to handle the complex geometry shown in Figure 10.1 for use in the multi-block framework, the geometry is partitioned into blocks. These blocks must be logical

Figure 10.1: Sample CAD file of HIT-SI geometry imported into CUBIT using an ACIS file.

cubes, and their faces conformal with the other blocks. With complex geometries, such as the HIT-SI geometry, this can sometimes require many blocks. The CUBIT software handles the partitioning and more details of this process are included in Appendix A. Figure 10.2 shows both a top and side view of a resulting partitioning of the HIT-SI geometry described in the CAD file. This particular case uses 101 blocks, and captures all geometric aspects of the HIT-SI experimental configuration.

10.2 Mesh Assessment and Improvement

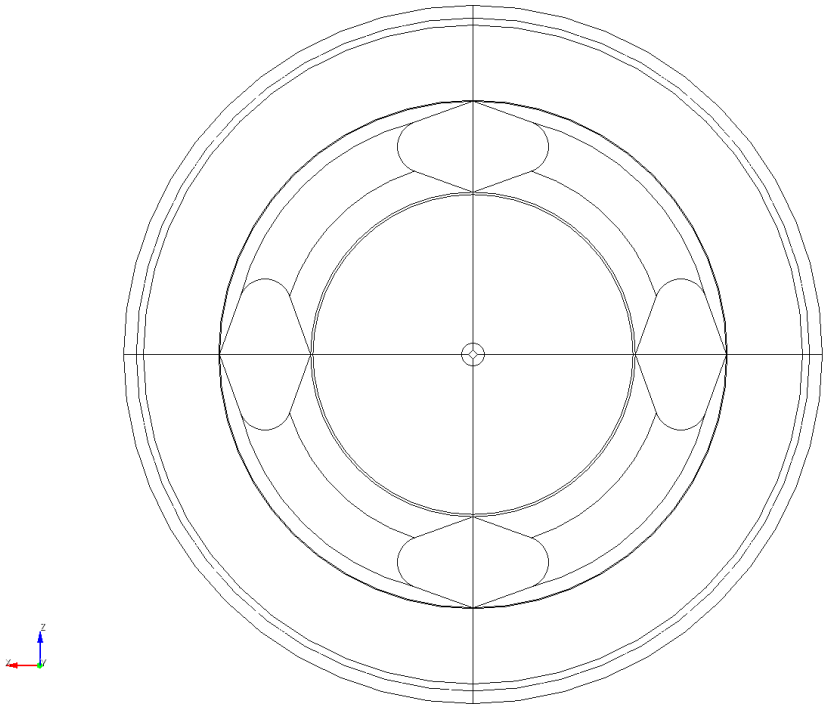
After creation of the block partitions of the geometry, the blocks are meshed to create a global mesh of the domain. During this process careful consideration of the mesh quality must be made in order to minimize the potential for introducing errors in the solution.

In order to make the process of creating a mesh with minimal distortions more efficient, some automation of the process is desirable. This is possible during some of the stages of creating a mesh by using smoothers, a change in resolution, or a change in block partitioning. Due to the fact that the multi-block framework was designed for structured block elements only, it still requires manual partitioning or repartitioning of the geometry. The mesh metric analysis can however guide the process of repartitioning by calculating the mesh metrics of the original geometry, and identifying areas that need improvement.

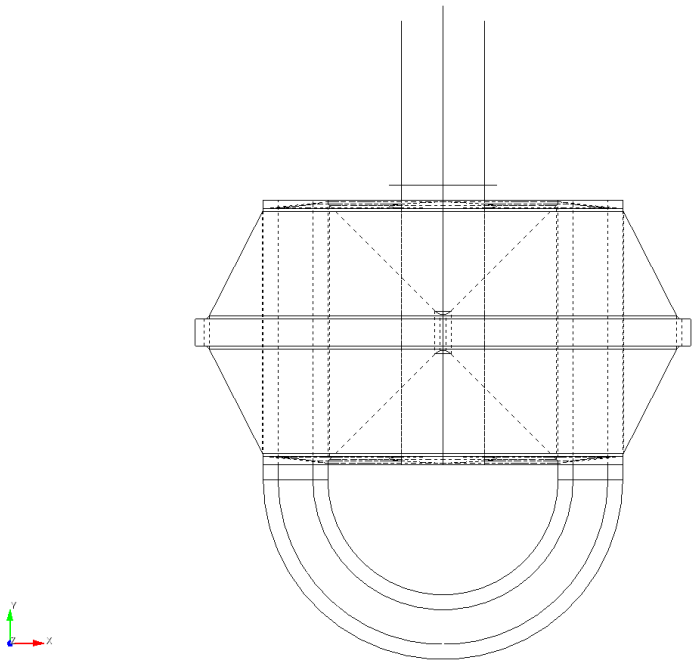
10.2.1 Identification of Problematic Mesh Areas

The first step to improving a mesh quality is to identify the problematic areas. One can do this by visual inspection, but a more useful technique is to use mesh quality metrics to analyze the mesh. A first check one could use is the Jacobian metric. This metric has the property of being less then or equal to zero when elements are degenerate. Having degenerate elements will typically not allow solvers to converge to a solution.

An example of where degenerate elements can occur is the region of the HIT-SI geometry where the injectors meet the main confinement region. An isolated view of



(a)



(b)

Figure 10.2: Top and side view of block partitions for the HIT-SI geometry.

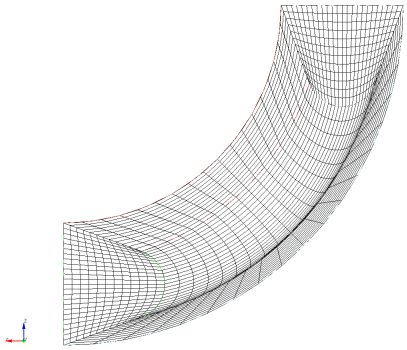


Figure 10.3: Isolated view of a region of the HIT-SI mesh with degenerate elements.

the surface mesh resulting in this region are shown in Figure 10.3. The Jacobian and Shape metrics [2] are shown in Figure 10.4. Notice the Jacobian metric is negative in the region, and subsequently the Shape metric is zero signifying the mesh elements are degenerate.

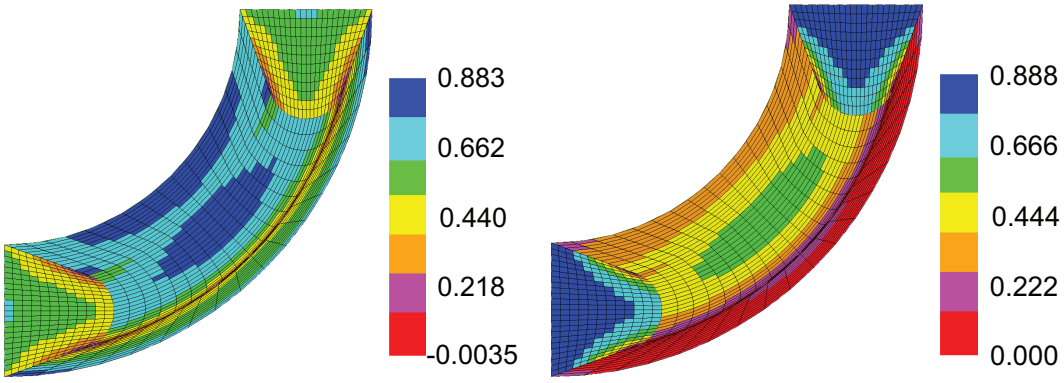
Additionally, the mesh shown in Figure 10.3 has a simple injector footprint, with two symmetrical blocks. This block shape is similar to that of the 'square-to-circle' mapping, and is prone to highly distorted elements, especially as the resolution is refined. Table 10.1 shows the minimum mesh metric values for the Shape, Shear, and

Table 10.1: Minimum mesh metric values for the injector footprint surface mesh with various resolutions.

Resolution	Shape	Shear	Jacobian
2×4	0.0850	0.105	0.1460
4×8	0.0437	0.0537	0.0188
8×16	0.0229	0.0282	0.00246
16×32	0.0125	0.0154	0.000336

Jacobian metrics computed on various resolution of the injector footprint mesh. Notice all of the metrics decline with increased resolution, and therefore the mapping for this particular block is flawed.

The surface surrounding the injector footprint, as well as the injector footprint itself



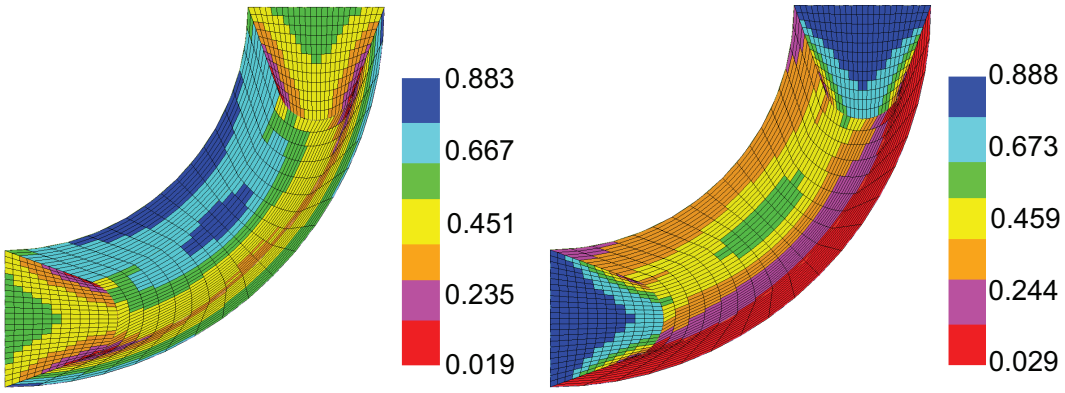
(a) Jacobian Metric (b) Shape Metric

Figure 10.4: Jacobian and Shape metrics computed on a region of the HIT-SI mesh with degenerate elements.

both have problematic meshes. The level of distortion is determined by examining the mesh metrics for these areas. In the case of the blocks surrounding the injector, the mesh Jacobian is negative signifying that something must be done to improve the mesh, otherwise the solution will not converge. In the case of the injector footprint, the metric values are small, signifying a poor mesh. Additionally when the mesh is refined the metrics decrease further, suggesting the area is problematic.

10.2.2 Mesh Improvement by Smoothers

A simple means of correcting the degenerate elements identified around the injector footprint (Figure 10.3) is to apply a smoother to the mesh. An elliptic Winslow smoother [2] is applied, and the mesh is no longer degenerate. Figure 10.5 shows the Jacobian and Shape metrics computed after applying a Winslow smoother. The smoothing process improved the mesh and eliminated degenerate elements. Other mesh smoothers could be used, and several of the available smoothers in the CUBIT software were tested with this particular mesh. Table 10.2 shows results for various smoothers and the resulting minimum mesh metric values for each.



(a) Jacobian Metric (b) Shape Metric

Figure 10.5: Jacobian and Shape metrics computed on a region of the HIT-SI mesh after applying a Winslow smoother to degenerate elements.

Table 10.2: Minimum mesh metric values for various smoothers applied to the surface with degenerate elements shown in Figure 10.3. The resolution of the surface is 12×18.

Smoother	Shape	Shear	Jacobian
No Smoothing	0.0000	0.0000	-0.00732
Centroid Area	0.0379	0.0606	0.1230
Winslow	0.0516	0.0766	0.1420
Smart Laplacian	0.0000	0.0000	-0.2340
Mean Ratio	0.0953	0.1200	0.1750
Laplacian	0.0423	0.0620	0.1130
Condition Number	0.1440	0.1330	0.1660

10.2.3 Mesh Improvement by Introducing More Block Partitions

The block configuration shown in Figure 10.2 is acceptable for creating a mesh, but the metric values indicate a poor mesh, and a refinement of the mesh further decreases the metrics values. Smoothers in this case also do not improve the mesh. Another means of improving the mesh is further partitioning of the geometry into more favorable block shapes. The improved shape of the blocks subsequently can improve the resulting mesh and the quality of the simulations on these meshes. Figure 10.6 shows a configuration where the footprint of the injectors have increased partitioning. This increases

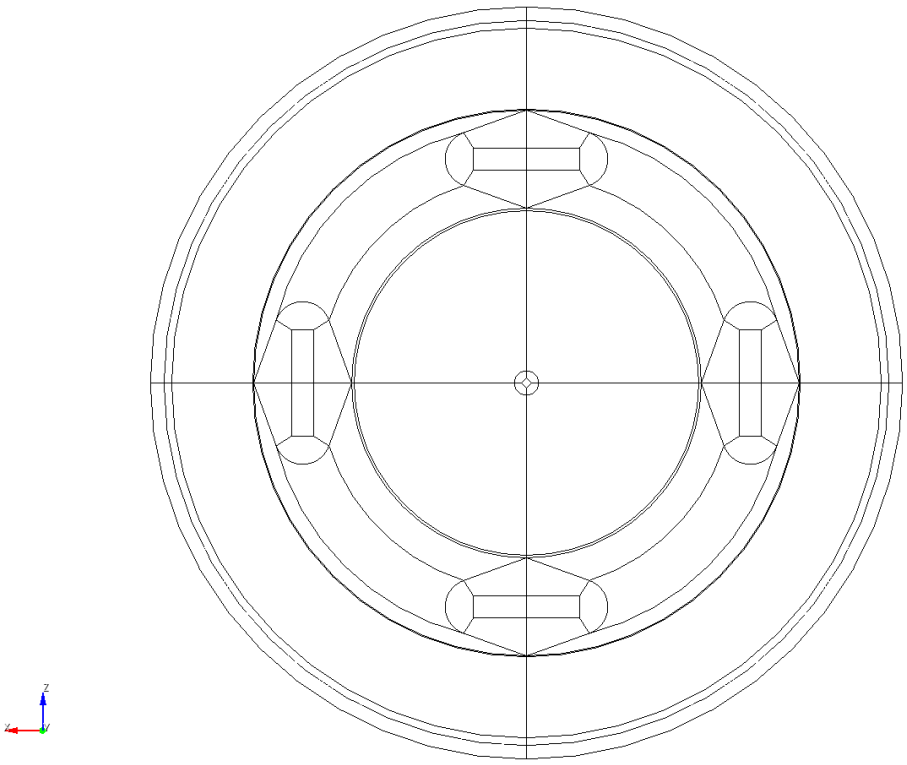
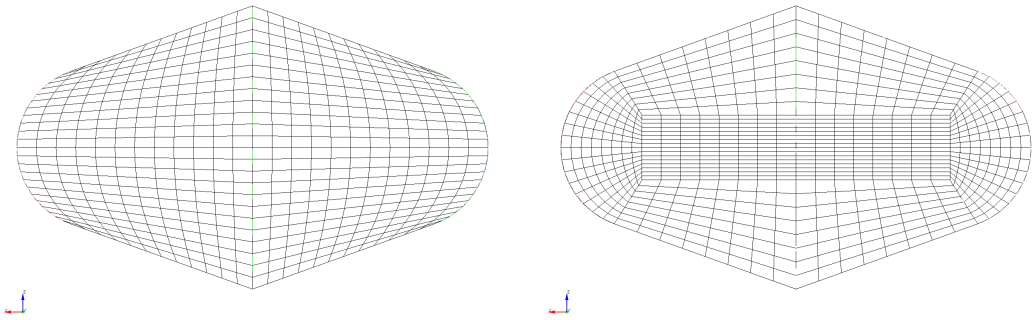


Figure 10.6: Top view of an alternative block partitioning for the HIT-SI geometry where the injector footprints have been further partitioned.

the total block count, but overall has the potential to reduce the simulation error when using the mesh. Figure 10.7 shows example meshes of the original injector footprint, and the more complex partitioning.

In order to improve the mesh in this circumstance, blocks are introduced to further partition into shapes that have more favorable mesh mappings. Similar to the five block cylinder mesh described in Section 8.4 and shown in Figure 8.4(a), the injector footprint can be partitioned as shown in Figure 10.6. Figure 10.7 shows example meshes with both the simple and more complex injector footprint. With the same number of elements the minimum mesh metric values for Shape, Shear and Jacobian are 0.406, 0.533, and 0.0659 respectively. This is about a five times increase for the Shape and Shear metrics, and it is expected that more accurate simulations result on this



(a) (b)

Figure 10.7: Examples meshes with a simple 2 block injector footprint and a more complex partitioning with 8 blocks.

mesh. Table 10.3 shows the comparison between the two injector footprint configurations.

Table 10.3: Minimum mesh metric values for the two injector footprint surface meshes shown in Figure 10.7.

Injector	Shape	Shear	Jacobian
2 Block	0.0850	0.1050	0.1460
8 Block	0.4060	0.5330	0.0659

An example mesh using the 209 block configuration is shown in Figure 10.8, and details of the mesh are shown in Figure 10.9.

10.3 Heat Equation Solution using a HIT-SI Mesh

An anisotropic heat equation solution is computed on the mesh shown in Figures 10.8, and 10.9 to demonstrate the ability of the multi-block framework to handle the complex geometric shape of the HIT-SI experiment. A Gaussian torus shape is initialized in the domain, and falls off such that temperature is initially zero at the boundary. The heat conduction anisotropy is aligned in the $\hat{y}$-direction and is considered the parallel

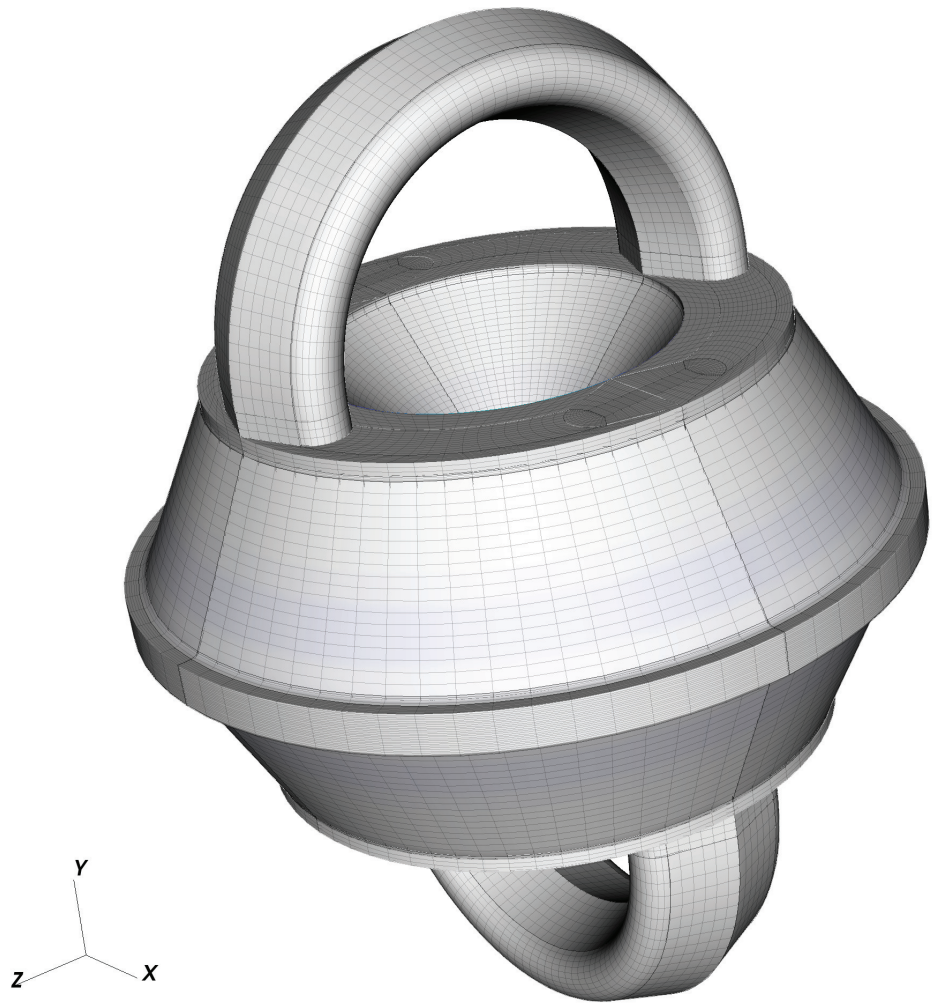


Figure 10.8: Multi-block mesh of a HIT-SI geometry using 209 blocks and 676,512 linear finite elements

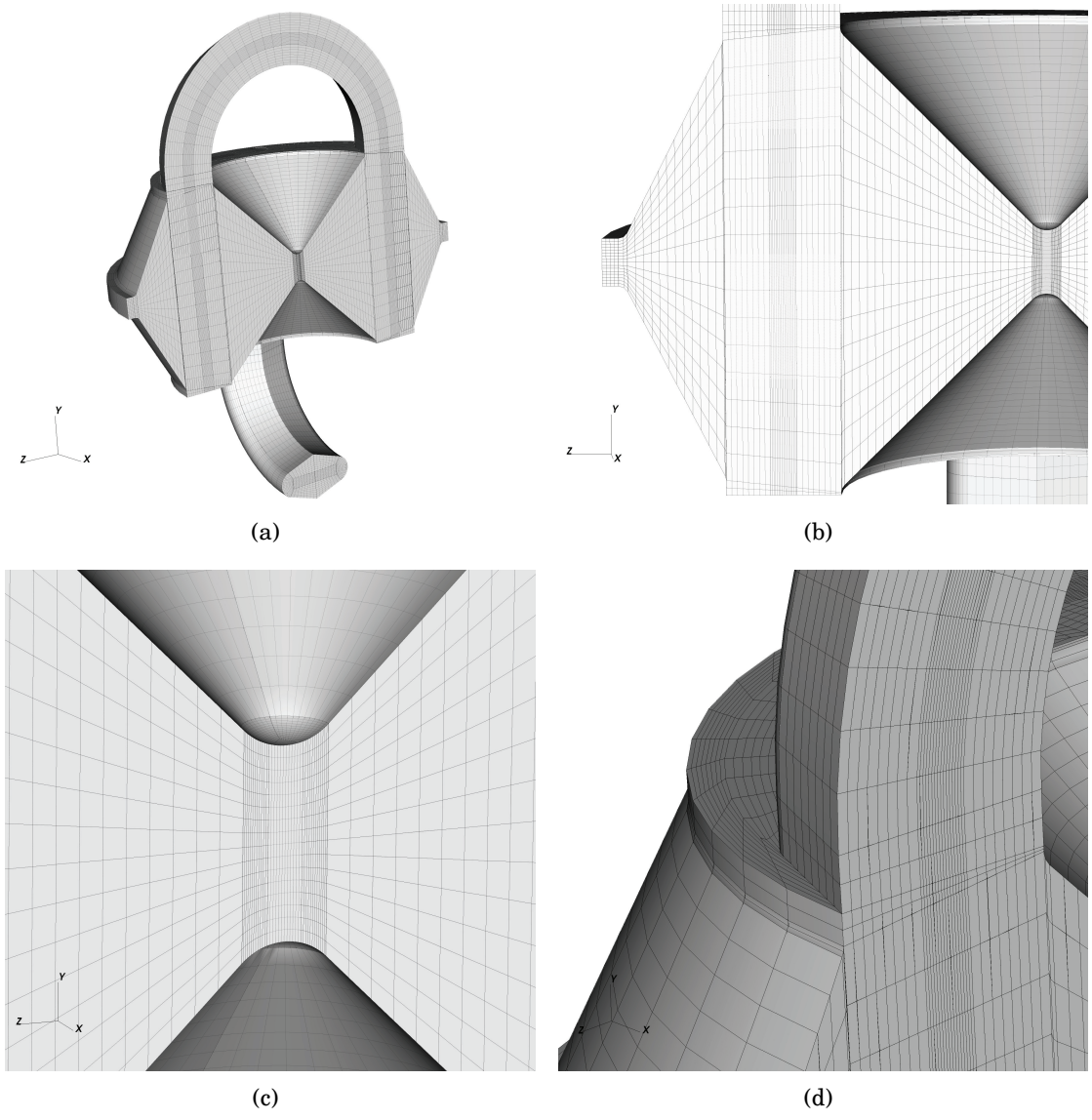


Figure 10.9: Cutaway and detail of the HIT-SI multi-block mesh

direction. The thermal conduction is preferential along the parallel direction and is 10 times that of the perpendicular directions.

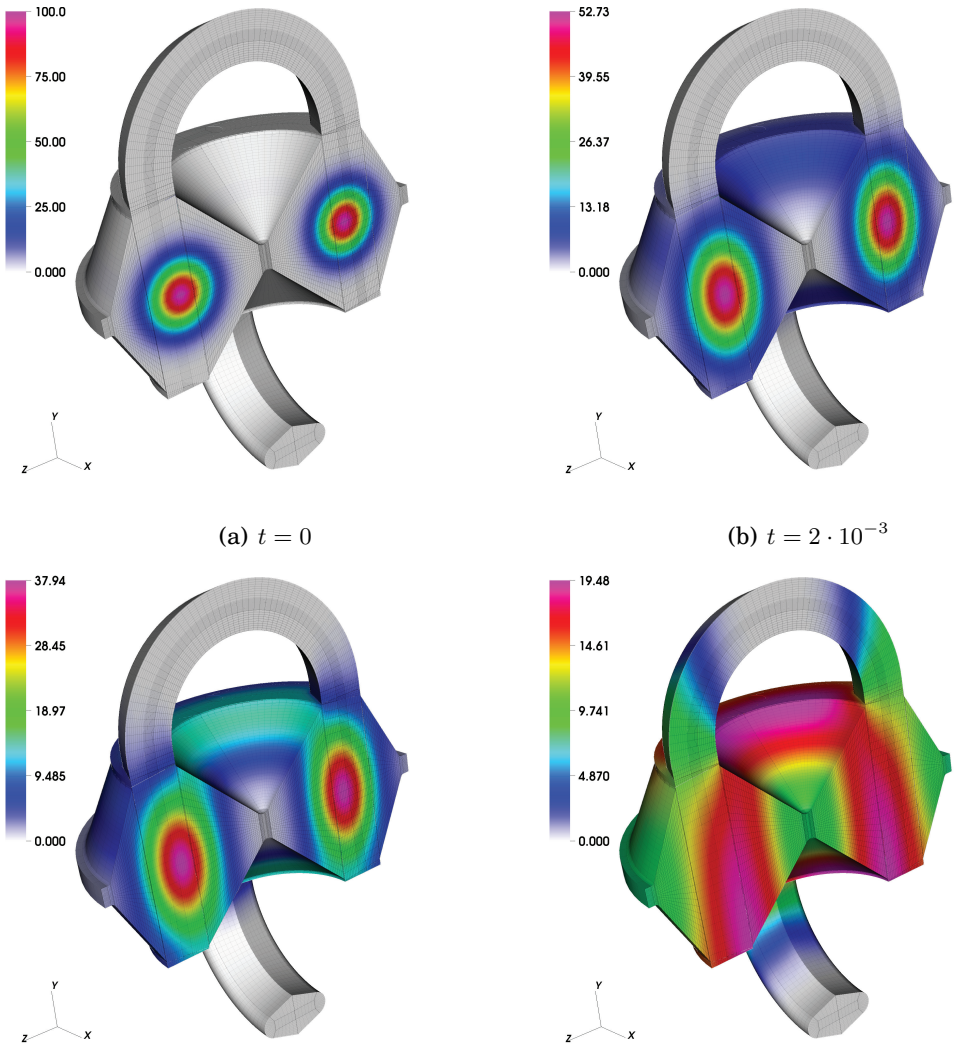
Figure 10.10 shows the solution evolution at 0, $2 \cdot 10^{-3}$, 10^{-2}, and 10^{-1} diffusion times. The diffusion time is approximately $\tau_D \approx D/L^2$, where D is the magnitude of the diffusion operator, and L is the length scale of the domain. It can be seen that as the solution evolves the initial torus shape elongates more in the $\hat{y}$-direction. A zeroflux boundary condition prevents thermal conduction through the wall, and is evident as significant temperature reaches the boundary. By the end of the simulation, the temperature has made it into the injectors, demonstrating that the multi-block framework can deal with the complex geometry.

10.4 Global Error Reduction with Improved Mesh

The idea behind the improvements made to a mesh as described in Section 10.2 is to have a reduction in total global error of the solution. This is tested by comparing the error in solution when comparing the simple 2 block injector footprint partitioning with that of the 8 block partitioning. The test problem is the same anisotropic heat conduction equation as described above in Section 10.3. Both meshes are identical otherwise and both have smoother applied to mesh blocks that have degenerate elements. The only major difference is the injector footprint blocks, which penetrate through the entire domain. Table 10.4 shows results for the different mesh metric values for each of the meshes, and the corresponding global error norm $|L^2|$.

Table 10.4: Minimum mesh metric values for the global volume and injector footprint surface mesh with for the two injector footprint configurations.

Mesh Metric	2 Block	8 Block
Global Shear	0.0692	0.0692
Global Shape	0.0420	0.0420
Injector Shear	0.0850	0.5330
Injector Shape	0.1050	0.4060
Normalized Global Error $\|L^2\|$	**0.3852**	**0.3731**



(a) $t = 0$

(b) $t = 2 \cdot 10^{-3}$

(c) $t = 10^{-2}$

(d) $t = 10^{-1}$

Figure 10.10: Solution to the anisotropic heat equation using the 209 block HIT-SI mesh. The anisotropy is aligned in the $\hat{y}$-direction, where the heat conduction is 10 times that of perpendicular directions.

A noticeable reduction in the total global error is computed with the improved mesh. The error reduction could be larger if the injector footprints were a larger part of the global domain. The injector footprint blocks are only a fraction of the total volume, and the rest of the mesh is identical to that of the simpler block configuration.

10.5 Further Mesh Improvement for MHD Simulations

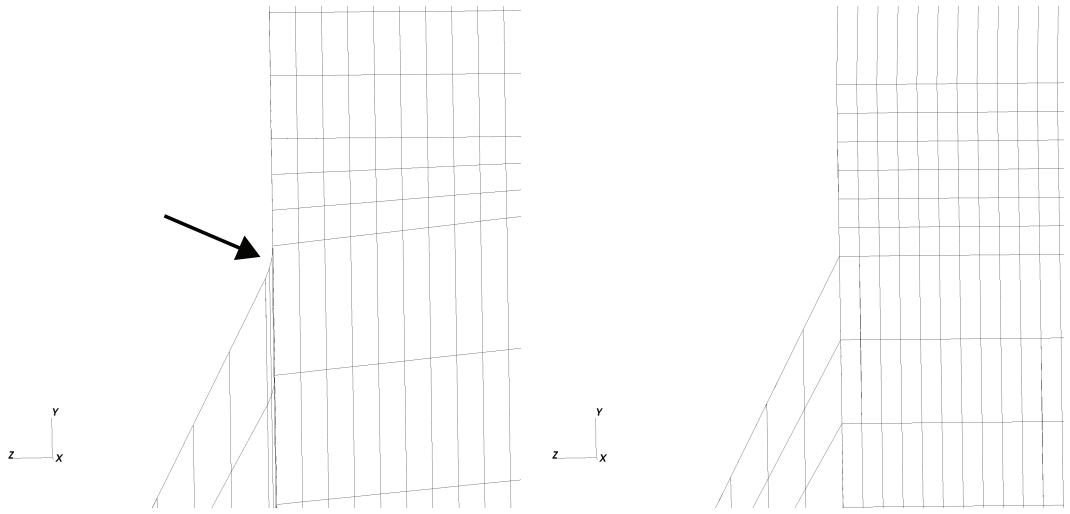
The previous sections use the heat equation as a test problem. The heat equation is a simple dissipative scalar system that can easily converge on distorted meshes. More complex coupled equation systems with vector variables and dispersive operators have a more difficult time converging on distorted meshes. With the MHD equation system as described in Chapter 5 and 8.4.1, the same mesh used for the heat equations is found to be problematic and does not properly converge to a solution.

After further investigation using the mesh metric analysis, a problem area of the mesh was identified. The detailed HIT-SI mesh included features with rapid curvature in a small area. Figure 10.11(a) shows a closeup of a region with the curvature. In this scenario in order to have more well behaved elements either the curvature must be resolved, or the curvature removed. In the case of the HIT-SI mesh, it would create an unreasonably large mesh to fully resolve the curvature in the geometry, and therefore makes the most sense to remove these regions.

The mesh metric values at these regions of rapid curvature proved to have the most degenerate elements in the total mesh. After truncating these features from the mesh, the metrics improved. Table 10.5 shows the values for the metrics before and after the rapid curvature regions are truncated. Figure 10.11 shows the mesh with and without the curvature. The shape metric about a 6.5 times larger then the case with poor curvature, signifying an overall improvement of the mesh. With this mesh in place, the MHD equation system converges properly.

Table 10.5: Minimum mesh metric values for the global volume with and without the fine curvature features in the geometry.

Mesh Metric	w/ Curvature Features	w/o Curvature Features
Global Shear	0.0692	0.0992
Global Shape	0.0420	0.2740



(a) Mesh with Poor Curvature Geometry (b) Mesh with Poor Curvature Removed from Geometry

Figure 10.11: A closeup cutaway view of a region of the HIT-SI geometry mesh where the injector meets the main confinement region. Meshes with and without small curvature features are shown and the arrow points to the region where the curvature is poor. The mesh metrics are improved in the mesh without the curved region.

10.6 Toroidal MHD Pinch Simulation in the HIT-SI Geometry

A test problem to exercise the complex three dimensional geometry with the MHD equation system is devised. It involves a pinch wrapped into a torus shape within the HIT-SI geometry. This problem is sufficient to demonstrate that there are no geometry and boundary issues associated with the complex mesh, and that the MHD solution converges properly and provides the correct qualitative solution.

A torus is initialized with a pinch of magnetic field (via vector potential) and given a uniform density and pressure. The initial condition is advanced in time and immedi-

ately a $\vec{j} \times \vec{B}$ force from the toroidal current and poloidal magnetic field begins to pinch the plasma. Additionally a pressure profile builds to balance the magnetic pressure satisfying

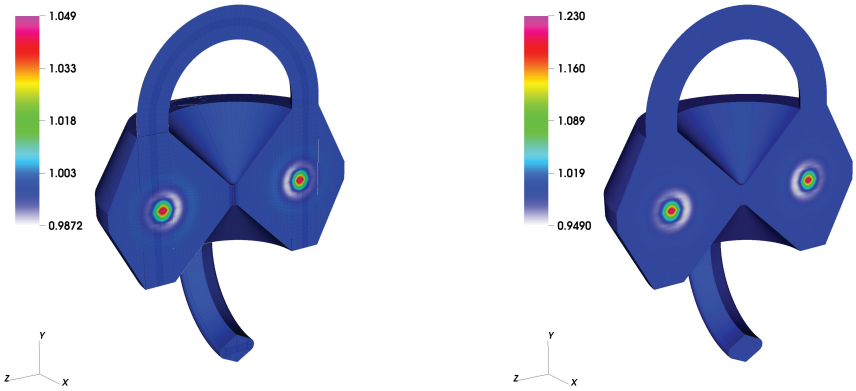
$$\nabla p = \vec{j} \times \vec{B}. \tag{10.1}$$

After the pressure and magnetic field balance, the pinch begins to resistively decay. Figure 10.12 shows a cutaway of the pressure in the domain over time. It can be seen that the pressure rises to a peak pressure at $t = 3.3$ while pinching down on its axis. After the pinch is in equilibrium, it can be seen that the pressure drops due to the fact that the pinch is resistively decaying. Similarly the magnitude of the current density is shown in Figure 10.13 for the same time sequence. By looking at the peak current magnitude it can be seen that it peaks at $t = 3.3$ and then begins to drop due to resistive decay. Cutaways profiles of the pressure and current density are shown in Figures 10.12 and 10.13 because the solution is approximately axisymmetric around the torus.

10.7 Spheromak MHD Simulation in HIT-SI Geometry

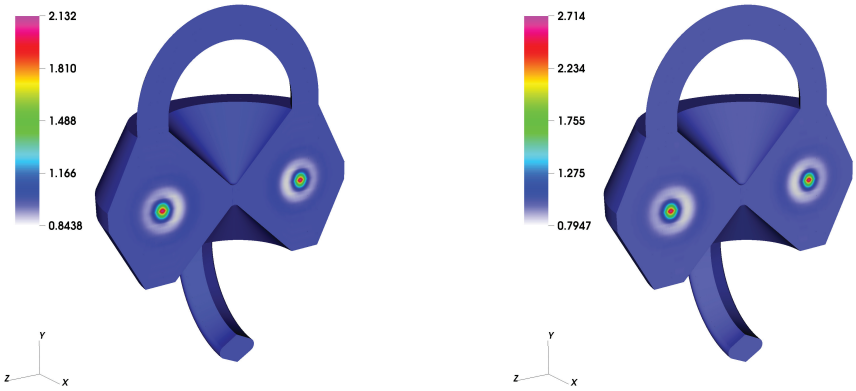
The HIT-SI experiment creates a spheromak in the confinement region and thus a more interesting test problem in the geometry is evolving a spheromak equilibrium. The full three-dimensional HIT-SI geometry with the injectors does not have a simple analytical description for a spheromak equilibrium, but one can be found by solving a Taylor minimum energy state eigenvalue problem. The Taylor minimum energy state problem solution can be found numerically for a specific geometry, but requires a specialized solver like PSITET. Interpolation errors when importing a PSITET equilibrium solution into HiFi are problematic and requires significant mesh resolution. An approximate equilibrium can be found by ignoring the injectors and solving the Grad-Shafranov equation in a 2D toroidal cross section, but this also involves a numerical solution.

An even simpler approximation is the Grad-Shafranov solution for a cylindrical geometry, which has an exact analytic description. The cylindrical equilibrium can



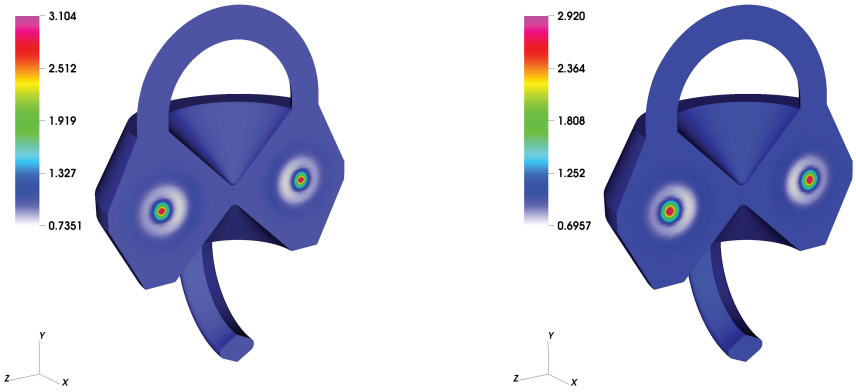
(a) $t = 0.5$

(b) $t = 1.0$



(c) $t = 2.0$

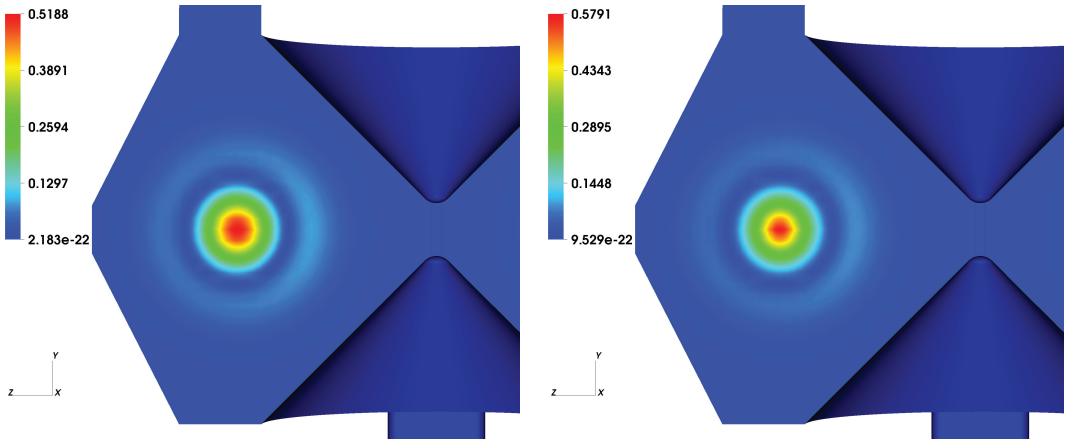
(d) $t = 2.5$



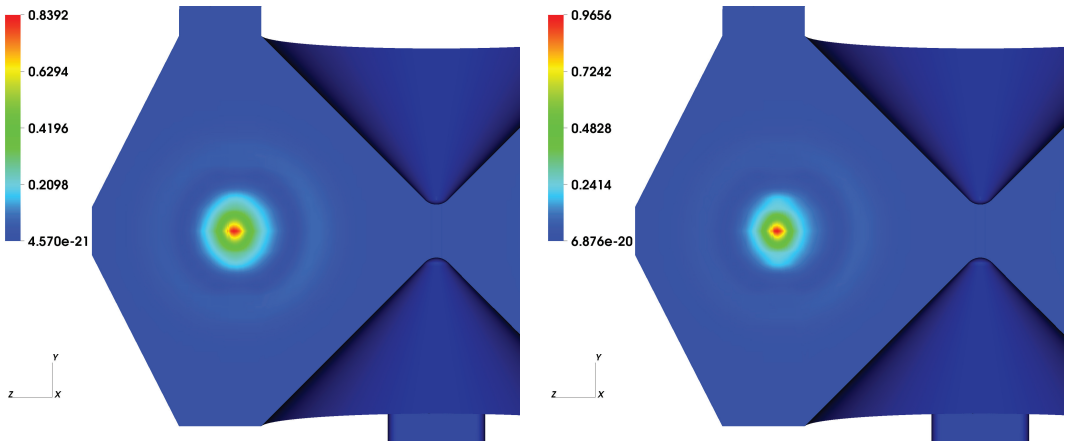
(e) $t = 3.3$

(f) $t = 4.1$

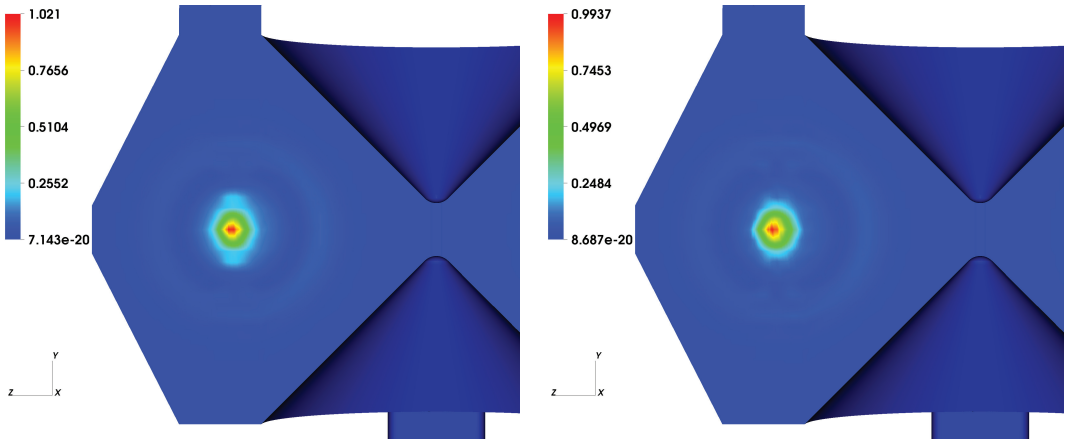
Figure 10.12: Pressure values over time for a pinch in a torus shape. The initial pressure is uniform at unity and over time it pinches to a balance the $\vec{j} \times \vec{B}$ forces at $t = 3.3$. After the plasma reaches force balance, the pinch begins to resistively decay.



(a) $t = 0.5$ (b) $t = 1.0$



(c) $t = 2.0$ (d) $t = 2.5$



(e) $t = 3.3$ (f) $t = 4.1$

Figure 10.13: Current magnitude over time for a pinch in a torus shape. The $\vec{j} \times \vec{B}$ forces pinch the plasma until a pressure balance is met at $t = 3.3$ and then it begins to resistively decay.

be placed into the HIT-SI shape, and then a elliptic relaxation problem solved as described in section 5.2 for the vector potential with additional boundary conditions. This ensures the vector potential does not have a tangential component at the boundary and gives good approximate spheromak equilibrium in the HIT-SI geometry.

Figure 10.14 shows a cutaway of the $\hat{x}$-component of the vector potential in the HIT-SI geometry. Notice the field drops to zero at the boundaries. Figure 10.15 shows a vector field representation of the vector potential $\vec{A}$, contours of the magnitude of vector potential $|\vec{A}|$, and streamlines of the current density $\vec{J}$. This solution is after approximately one tenth of an Alfvén time, which is not enough time to expect interesting dynamics, but it does demonstrate the ability to solve MHD in a complex geometry, with an experimentally relevant initial condition.

From here there are many possibilities to simulate relevant and interesting phenomenon to the HIT-SI experiment. One could improve the initial condition to include the injectors, drive the spheromak by including an RFP in the injectors, or add perturbations to drive various instabilities. This simulation is meant to show that it is possible to run effective simulations in the complex fully three-dimensional multi-block geometry and is intended to be a springboard for future investigations.

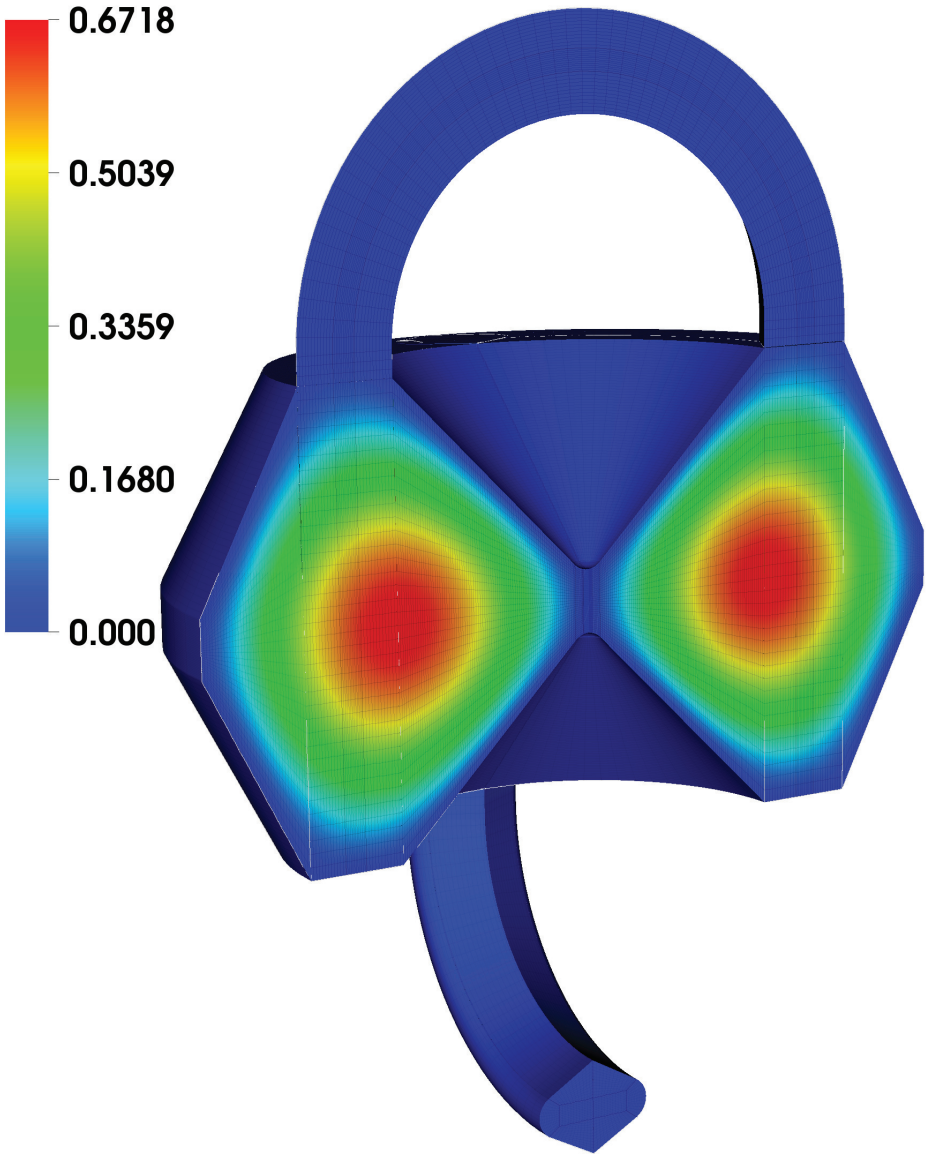


Figure 10.14: Cutaway of the azimuthal component of the vector potential A_θ for a spheromak equilibrium in the HIT-SI geometry

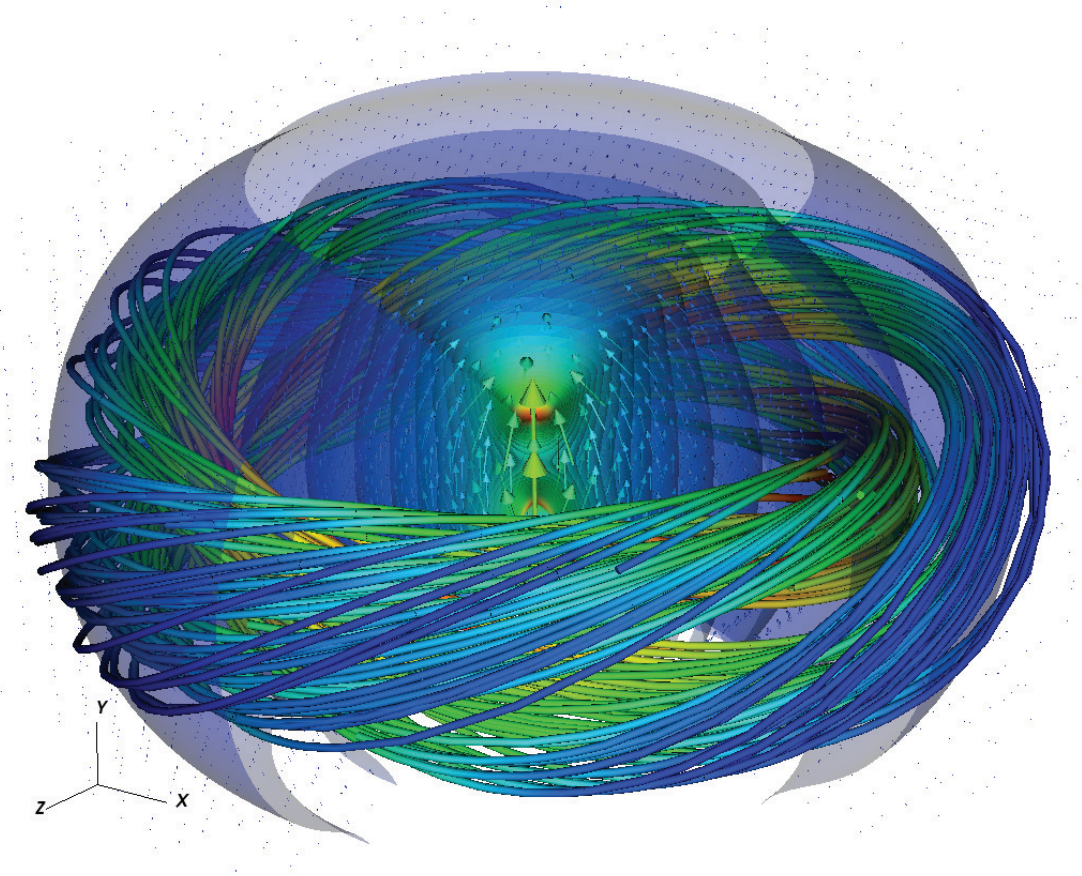


Figure 10.15: Vector plot of vector potential $\vec{A}$, contours of the vector potential magnitude $|\vec{A}|$, and streamlines of the current density $\vec{J}$ of a spheromak initial condition advanced in time with the MHD equations. The simulation time is $t \approx 0.2\tau_A$

Chapter 11

CONCLUSIONS

The main motivation for the work in this dissertation is to be able to model physical systems of interest with predictive capabilities in a user friendly manner for use as an effective engineering design tool. Its primary focus is a design and research tool for plasma physics, but its applicability extends beyond the plasma science community. The hope is that one will find the tool and methodology effective for solving a broad range of scientific problems.

The work in this dissertation began with a two dimensional academic research code. After the accomplishments made throughout this work, the code now allows a user to use the developments as an engineering design tool with a three dimensional representation, a CAD interface, mesh generator capabilities, and an *a priori* mesh quality estimation technique to produce reliable and accurate simulations with complex geometries and equation systems. The individual pieces of development described are important for an engineering design tool, but the methodology of starting from a CAD drawing and completing an accurate and reliable solution is also a significant achievement. This dissertation can be used as a guideline for following this methodology. Each of the major pieces are summarized.

11.1 HiFi Verification

The three-dimensional HiFi code had only recently been expanded from the SEL code when beginning this dissertation, and no physics modules had been written. A resistive MHD model was written for the 3D HiFi code and therefore needed to be verified before it could be accepted for general use. A linear MHD wave problem and nonlinear spheromak tilt instability were chosen as test problems. Both test problems exhibited the correct behavior and thus verified the numerical model.

Dissipative MHD waves were analyzed and confirmed to have the correct wave speed and decay rates. Pure waves were initialized by first linearizing the equation system and solving for the eigensystem. These initialized waves have known analytic solutions and could be compared to the results computed by HiFi. The solution errors when compared to an analytic solution were almost always less then 1% error.

A nonlinear evolution of a spheromak was also chosen to verify the HiFi resistive MHD model. A spheromak Taylor state was initialized in a cylindrical flux conserver geometry and given a small perturbation. The spheromak demonstrated the tilt instability and relaxed to the correct Taylor minimum energy state.

11.2 A priori Mesh Quality Error Estimation Analysis

The main finding from the mesh deformation analysis is that for increasing amount of distortion in a mesh, the total error increased, but the spectral convergence rates remained intact. Additionally it is shown that the global error in simulations correlated to the mesh quality metrics, and they could be used *a priori* to estimate the solution error. The product of the Knupp metrics proved to be the most useful metrics in determining the approximate solution error. The method was initially performed in 2D and extended to 3D with similar behavior. Several equation systems and test problems were investigated with similar conclusions, suggesting the analysis has broad applicability.

11.3 Multi-Block Development

The multi-block development is successfully implemented in the HiFi code, and thoroughly tested with several test problems. The framework is mostly invisible to the user, and is a useful numerical tool for simulations with complex geometries. Simple geometries like a five block cylinder are successfully tested, as well as more complex configurations like a cylinder with extrusions or the HIT-SI like geometry.

11.4 Application to a Z-Pinch

The multi-block development is put to use by studying a Z-pinch $m = 1$ kink mode with ZaP experimental parameters. The results of simulations are compared to a linear stability analysis and are shown to be in good agreement. A more complex cylindrical geometry with extrusions is used to model a recent design change to the ZaP experiment. The results are consistent with experimental observations. Additionally simulations

with axial shear flow above and below the theoretical stabilizing criteria demonstrates stabilization and confirms the model.

11.5 Z-Pinch Mesh Deformation Analysis

The success of the initial mesh deformation analysis laid the foundation for using the same analysis on a more realistic problem. A Z-pinch $m = 1$ kink mode simulation with ZaP experimental parameters is analyzed in a multi-block geometry. Similar results are obtained to that of the initial more simplistic mesh deformation analysis, which is promising. The Z-pinch problem is evolved using the full nonlinear hyper-resistive MHD equations in a multi-block geometry with experimentally relevant operating parameters. The fact that the metrics provide a good correlation to solution error in this configuration is promising for general applicability of the technique.

11.6 Application to HIT-SI

A major motivation for the multi-block development and mesh deformation metrics is to be able to create and use a complex mesh that results from the HIT-SI geometry. Using the developments of the multi-block framework and CAD interface a block structured mesh is created for the HIT-SI geometry. Due to the complex features in the mesh, degeneracies and poor mesh regions were discovered that would either cause the solver to fail to converge or generate inaccurate results. The mesh deformation analysis is then used to identify regions that need repair or improvement. It is shown that with mesh improvement the solution accuracy improved in the HIT-SI geometry solving the heat equation. It was also shown that for more complex equation systems like MHD, that tighter tolerances on the metrics are needed and that poor regions of the mesh need to be improved or eliminated. This work ties together much of the previous work in this dissertation.

11.7 Final Summary

The work in this dissertation accomplishes the goal of creating an effective engineering design tool by combining several development pieces into a single coherent analysis method. The research shows that one can now use the tool with a CAD representation, an automatic mesh generator, and an *a priori* error analysis technique to estimate the solution errors and identify problematic regions. The meshes can be refined through the analyses to minimize the solution error. This gives experimentalists, computationalists and other scientists an effective way of analyzing their particular problem with confidence the results will be accurate.

BIBLIOGRAPHY

[1] Plasma science and innovation center (psi-center) website. http://www.psicenter.org/, 2011.

[2] T.D. Blacker, W.J. Bohnhoff, and T.L. Edwards. CUBIT mesh generation environment. Volume 1: Users manual. Technical report, Sandia National Labs., Albuquerque, NM (United States), 1994.

[3] A.H. Glasser and X.Z. Tang. The SEL macroscopic modeling code. *Computer physics communications*, 164(1-3):237–243, 2004.

[4] V.S. Lukin, A.H. Glasser, W. Lowrie, and E. Meier. HiFi–Adaptive, Implicit, High Order Finite Element Code for General Multi-Fluid Applications. In *AAS/Solar Physics Division Meeting*, volume 40, 2009.

[5] R.W. Clough. The finite element method in plane stress analysis. 1960.

[6] K.I. Gerhardt and G.W. Leibniz. *Leibnizens mathematische Schriften*. Verlag von A. Asher & Comp., 1855.

[7] K. Schellbach. Probleme der Variationsrechnung. *Journal für die reine und angewandte Mathematik*, 1851(41):293–363, 1851.

[8] A. Hrennikoff. Solution of problems in elasticity by the framework method. *Journal of Applied Mechanics*, 8(4):169–175, 1941.

[9] J.T. Oden. Historical comments on finite elements. In *A history of scientific computing*, pages 152–166. ACM, 1990.

[10] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin (New Series) of the American Mathematical Society*, 49(1):1–23, 1943.

[11] J.H. Agyris and S. Kelsey. Energy theorems and structural analysis. *Aircraft Engineering*, 1955, 1954.

[12] R.W. Clough. Thoughts about the origin of the finite element method. *Computers and Structures*, 79(22):2029–2030, 2001.

[13] C.A. Felippa. A historical outline of matrix structural analysis: a play in three acts. *Computers & Structures*, 79(14):1313–1324, 2001.

[14] M.J. Turner, R.W. Clough, H.C. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *J. Aero. Sci*, 23(9):805–823, 1956.

[15] O.C. Zienkiewicz. The birth of the finite element method and of computational mechanics. *International Journal for Numerical Methods in Engineering*, 60(1):3–10, 2004.

[16] J.T. Oden and D. Somogyi. Finite element applications in fluid dynamics. *Journal of Engineering Mechanics*, 95:821–826, 1968.

[17] J.T. Oden. Finite-Element Analogue of Navier-Stokes Equation. *Journal of the Engineering Mechanics Division*, 96(4):529–534, 1970.

[18] J.C. Heinrich, P.S. Huyakorn, O.C. Zienkiewicz, and A.R. Mitchell. An upwindfinite element scheme for two-dimensional convective transport equation. *International Journal for Numerical Methods in Engineering*, 11(1):131–143, 1977.

[19] R. Gruber and J. Rappaz. *Finite Element Methods in Linear Ideal Magnetohydrodynamics*. Springer Verlag, 1985.

[20] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, and M. Litke. Flow simulation and high performance computing. *Computational Mechanics*, 18(6):397–412, 1996.

[21] Top500 supercomputing sites website. http://www.top500.org, 2010.

[22] CNC website. http://www.computerhistory.org, 2010.

[23] S.G. Nash. *A History of scientific computing*. ACM New York, NY, USA, 1990.

[24] P.E. Ceruzzi. *A history of modern computing*. The MIT press, 2003.

[25] Y. Oyanagi. Future of supercomputing. *Journal of computational and applied mathematics*, 149(1):147–153, 2002.

[26] L. Lapointe and J. Morse. Message Passing Interface Forum. MPI: A Message Passing Interface Standard. In *J. Combin. Theory Ser. A*. Citeseer, 1994.

[27] D.W. Walker and J.J. Dongarra. MPI: a standard message passing interface. *Supercomputer*, 12:56–68, 1996.

[28] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhan. PETSc website. http://www.mcs.anl.gov/petsc, 2011.

[29] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc users manual. Technical report, Citeseer, 2004.

[30] T.R. Jarboe. The spheromak confinement device. *Physics of plasmas*, 12:058103, 2005.

[31] T.R. Jarboe, W.T. Hamp, G.J. Marklin, B.A. Nelson, R.G. ONeill, A.J. Redd, P.E. Sieck, R.J. Smith, and J.S. Wrobel. Spheromak formation by steady inductive helicity injection. *Physical review letters*, 97(11):115003, 2006.

[32] A.J. Redd, T.R. Jarboe, W.T. Hamp, B.A. Nelson, R.G. ONeill, P.E. Sieck, R.J. Smith, G.L. Sutphin, and J.S. Wrobel. Overview of the helicity injected torus (hit) program. *Journal of fusion energy*, 26(1):163–168, 2007.

[33] P.E. Sieck, W.T. Hamp, V.A. Izzo, T.R. Jarboe, B.A. Nelson, R.G. O'Neill, A.J. Redd, and R.J. Smith. Initial studies of steady inductive helicity injection on the hit-si experiment. *Plasma Science, IEEE Transactions on*, 33(2):723–728, 2005.

[34] V.S. Lukin. *Computational Study of the Internal Kink Mode Evolution and Associated Magnetic Reconnection Phenomena*. PhD thesis, Princeton University, 2007.

[35] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover publications, 1964.

[36] W. Lowrie, V.S. Lukin, and U. Shumlak. A priori mesh quality metric error analysis applied to a high-order finite element method. *Journal of Computational Physics*, 230(14):5564–5586, 2011.

[37] P.M. Knupp. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.*, 23:193–218, January 2001.

[38] P.M. Knupp. Remarks on mesh quality. In *45th AIAA Sciences Meeting and Exhibit, Reno NV (January 7-10, 2007)*, 2007.

[39] P.M. Knupp. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, 39(3):217 – 241, 2003.

[40] M. Berzins. Mesh quality: a function of geometry, error estimates or both? *Engineering with Computers*, 15(3):236–247, 1999.

[41] B. Cockburn and P.A. Gremaud. A priori error estimates for numerical methods for scalar conservation laws. Part I: The general approach. *Mathematics of computation*, 65(214):533–574, 1996.

[42] Y. Kallinderis and C. Kontzialis. A priori mesh quality estimation via direct relation between truncation error and mesh distortion. *Journal of Computational Physics*, 228(3):881–902, 2009.

[43] K. Okamoto, G. Klopfer, and J. Chattot. Assessing grid quality of structured meshes by truncation error analysis. In *Fifteenth International Conference on Numerical Methods in Fluid Dynamics*, pages 340–345. Springer, 1997.

[44] I. Babuška and A.K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13:214, 1976.

[45] P.M. Knupp. Label-invariant mesh quality metrics. *Proceedings of the 18th International Meshing Roundtable*, pages 139–155, 2009.

[46] L.A. Freitag and P.M. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *International Journal for Numerical Methods in Engineering*, 53(6):1377–1391, 2002.

[47] I. Tsukerman. A general accuracy criterion for finite element approximation. *Magnetics, IEEE Transactions on*, 34(5):2425–2428, 1998.

[48] F.X. Zgainski, Y. Marechal, J.L. Coulomb, M.G. Vanti, and A. Raizer. An a priori indicator of finite element quality based on the condition number of the stiffness matrix. *Magnetics, IEEE Transactions on*, 33(2):1748–1751, 1997.

[49] M.L. Bittencourt and T.G. Vazquez. A nodal spectral stiffness matrix for the finite-element method. *IMA Journal of Applied Mathematics*, 73(6):837, 2008.

[50] V. Hernandez, J.E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):351–362, 2005.

[51] H. Maecker. Elektronendichte und temperatur in der säule des hochstromkohlebogens. *Zeitschrift für Physik A Hadrons and Nuclei*, 136(2):119–136, 1953.

[52] J.B. Shumaker. Arc source for high temperature gas studies. *Review of Scientific Instruments*, 32(1):65–67, 1961.

[53] R.G. Jahn. *Physics of Electric Propulsion*. Dover Publications, Inc., Mineola, NY, 1968.

[54] R.R. John, S. Bennett, and J.F. Connors. Arcjet engine performance: experiment and theory. *AIAA (Am. Inst. Aeron. Astronaut.) J.*, 1, 1963.

[55] L.E. Wallner and J. Czika. *NASA TN D-2868*, volume 2868. National Aeronautics and Space Administration, 1965.

[56] V.R. Watson and E.B. Pegot. NASA technical note. Technical report, TN-D-4042, 1967.

[57] H.A. Stine, V.R. Watson, and C.E. Shepard. Effect of axial flow on the behavior of the wall-constricted arc. Technical report, National Aeronautics and Space Administration, Moffett Field, Calif. Ames Research Center, 1964.

[58] R.A. Graves and W.L. Wells. Preliminary study of a wall-stabilized constricted arc. *NASA TM X-2700*, 1973.

[59] R.C. Preston. Spectroscopic studies of a plasma temperature and radiation standard based on a wall-stabilized arc. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 18(3):337–360, 1977.

[60] K.E. Clark and R.G. Jahn. The magnetoplasmadynamic arcjet. *Astronautica Acta*, 13:315–325, 1967.

[61] R.W. Kilb, H. Hurwitz Jr, and W.F. Westendorp. Wall stabilization effects in theta pinch configurations. *Physics of Fluids (US)*, 5, 1962.

[62] R.W. Kilb, H. Hurwitz Jr, and W.F. Westendorp. Wall stabilization effects in theta-pinch configurations. *Physics of Fluids*, 6:1332, 1963.

[63] F.A. Haas and J.A. Wesson. Stability of the theta-pinch. *Physics of Fluids*, 9:2472, 1966.

[64] F.A. Haas and J.A. Wesson. Stability of the theta-pinch. ii. *Physics of Fluids*, 10:2245, 1967.

[65] B.B. Kadomtsev. Reviews of plasma physics, edited by ma leontovich (consultant bureau, new york). *Vol*, 2:153, 1966.

[66] F.L. Ribe. Lasl controlled thermonuclear research program. progress report for a 12-month period ending december 1972. Technical report, comp.; Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[67] U. Shumlak and C. Hartman. Sheared Flow Stabilization of the m=1 Kink Mode in Z-Pinches. *Physical Review Letters*, 75:3285–3288, 1995.

[68] J.P. Goedbloed, D. Pfirsch, and H. Tasso. Instability of a pinch surrounded by a resistive wall. *Nuclear Fusion*, 12:649, 1972.

[69] E.J. Strait and et al. Wall stabilization of high beta tokamak discharges in DIII-D. *Phys. Rev. Lett.*, 74:2483, 1995.

[70] M.N. Rosenbluth and M.N. Bussac. Mhd stability of spheromak. *Nuclear Fusion*, 19:489, 1979.

[71] U. Shumlak. ZaP experiment website. http://www.aa.washington.edu/research/zap, 2011.

[72] J.P. Freidberg. Ideal magnetohydrodynamic theory of magnetic fusion systems. *Reviews of Modern Physics*, 54(3):801, 1982.

[73] W.H. Bennett. Magnetically self-focussing streams. *Physical Review*, 45(12):890, 1934.

[74] A.S. Bishop and US Atomic Energy Commission. *Project Sherwood: the US program in controlled fusion*. Addison-Wesley Pub. Co., 1958.

[75] W.A. Newcomb. Hydrodynamic stability of a diffuse linear pinch. *Ann. Phys*, 10:232–67, 1960.

[76] R.J. Bickerton. Pinch research. *Nuclear Fusion*, 20:1072, 1980.

[77] U. Shumlak, J.M. Blakely, B.-J. Chan, R.P. Golingo, S.D. Knecht, B.A. Nelson, R.J. Oberto, M.R. Sybouts, G.V. Vogman, and D.J. Den Hartog. Stabilization in the ZaP Flow Z-Pinch. *Journal of Fusion Energy*, 28:208–211, 2009.

[78] K. Appert, R. Gruber, and J. Vaclavik. Continuous spectra of a cylindrical magnetohydrodynamic equilibrium. *Phys. Fluids, v. 17, no. 7, pp. 1471-1472*, 17(7), 1974.

[79] U. Shumlak, B.A. Nelson, R.P. Golingo, S.L. Jackson, E.A. Crawford, and D.J. Den Hartog. Sheared flow stabilization experiments in the ZaP flow Z pinch. *Physics of Plasmas*, 10:1683, 2003.

[80] U. Shumlak, J. Chadney, R.P. Golingo, D.J. Den Hartog, M.C. Hughes, S.D. Knecht, B.A. Nelson, W. Lowrie, R.J. Oberto, M.P. Ross, J.L. Rohrbach, and G.V. Vogman. The Sheared-Flow Stabilized Z-Pinch. *Proceedings of the 15th International Conference on Emerging Nuclear Energy Systems*, 2011.

[81] J.P. Freidberg. Ideal magnetohydrodynamic theory of magnetic fusion systems. *Reviews of Modern Physics*, 54(3):801, 1982.

[82] U. Shumlak, R.P. Golingo, B.A. Nelson, and D.J. Den Hartog. Evidence of Stabilization in the Z-Pinch. *Physical review letters*, 87(20):205005, 2001.

[83] The hdf group. hierarchical data format version 5. http://www.hdfgroup.org/HDF5, 2000-2010.

[84] Netcdf website. http://www.unidata.ucar.edu/software/netcdf/, 2011.

[85] Cubit website. http://cubit.sandia.gov, 2011.

Appendix A

MULTI-BLOCK HIFI USER MANUAL

A.1 Obtaining HiFi-mb

The HiFi code can be downloaded via a subversion repository after communicating with one of the primary developers and completing a user agreement form. The form is displayed on the next page.

HiFi (SEL) User Agreement Form:

HiFi (SEL) is an open source code development project for solving systems of coupled non-linear PDEs on (semi-)structured logically hexahedral (rectangular) grids that abides by the following BSD-style license. The project has been supported, in part, by the U.S. Department of Energy. Before having access to the code, you must agree to the conditions of the license that serve as additional protections for the HiFi (SEL) code.

Terms of Agreement:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3) Neither the name of the project nor the names of its developers may be used to endorse, promote, or publish products derived from this software without specific prior written permission by one of the principle developers.
4) Publications or figures made using results of the HiFi (SEL) code calculations will acknowledge the HiFi (SEL) code.
5) It is understood that the HiFi (SEL) code is still under development and thus may not contain all features that users may need/want for their problem of interest.
6) It is understood that the HiFi (SEL) project does not guarantee that support will always be available to users of the code. In addition, it is understood that extensive support from a HiFi (SEL) team member on a particular application generally implies that any publication derived from the application will include that team's member(s) as a co-author.

THIS SOFTWARE IS PROVIDED BY HiFi team ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL neither any HiFi team member, nor the University of Washington, nor the United States Government, nor any agency thereof, nor any of their employees BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

If you agree to the above terms and would like to download and/or use the HiFi (SEL) code, please fill out this form and e-mail it to Vyacheslav Lukin at vlukin1@mailaps.org.

HiFi (SEL) Agreement form.

Name: _____
Organization: _____
Mailing Address: _____
Mailing Address: _____
Email Address: _____
Requested username for Web access: _____
Any comments or special requests:

I agree to the above Terms of Agreement and certify that the information submitted in the form is true.
NAME: _____
DATE: _____

A.2 Structure of the HiFi Code

The HiFi code has two major components: The main solver, and a physics module. The main solver is compiled into a library called 'libhifi.a'. The physics module is compiled separately and then linked to this main solver library.

The main solver requires three major external libraries to work properly. The PETSc [29, 28] library is essential to the code, as it relies on it for massively parallel linear and nonlinear solvers, preconditioners, matrix and vector data structures, and the majority of the MPI parallelism. The HDF5 [83] library is necessary for parallel I/O of the HiFi output and restart (checkpoint) data. Additionally the NetCDF [84] library is used for reading external mesh files created by the CUBIT [2] software. These libraries need to be compiled prior to compiling libhifi.a. They are common enough that many of the large HPC machines (e.g. Hopper and Franklin at NERSC) have these libraries compiled and are accessed by loading the appropriate modules using a command like `module load pets/3.1.04 .`

A.3 Compiling Main Solver Library

The main solver library can be compiled in a unix/linux environment using the '`make`' command with one of the included 'makefiles'. It requires a Fortran compiler, and below is an example makefile for use on the Hopper machine at NERSC. In the header the appropriate modules for loading the required external libraries are listed. One must make sure be sure to check the compiler flags, which are included in the `FFLAGS` environment variable. In this particular case the Portland Group (PGI) compiler is used. Other Fortran compiler such as Cray, Intel, and GNU have also been tested and successfully compile the code.

```
# This is the makefile for libhifi.a on Hopper.
# Before compiling, load the necessary modules with
# the following commands:
#
# module load petsc/3.1.04
# module load hdf5-parallel
# module load netcdf
#

FFLAGS = -O3 -Munroll=c:4 -Msave \
```

```
        $(HDF5) $(NETCDF) -I$(HDF5_DIR)
FC = ftn $(FFLAGS)
F90 = ftn $(FFLAGS)

OBJECTS = \
        io.o \
        local.o \
        debug.o \
        spline.o \
        bicube.o \
        jacobi.o \
        bessel.o \
        cubit3d.o \
        beltrami.o \
        job3.o \
        p3_hifi.o \
        p3_ct.o \
        p3_grid.o \
        p3_condense.o \
        p3_interior.o \
        p3_face.o \
        p3_rj.o \
        p3_diagnose.o \
        p3_snes.o \
        p3_advance.o \
        driver.o

hifi: $(OBJECTS) chkopts
        ar -r libhifi.a $(OBJECTS)
        rm -f *.cpp *.i

include $(PETSC_DIR)/conf/variables
include $(PETSC_DIR)/conf/rules

#dependencies

local.o: io.o
debug.o: local.o
spline.o: local.o
bicube.o: spline.o
jacobi.o: local.o
bessel.o: local.o
cubit3d.o: local.o
beltrami.o: jacobi.o bicube.o
job3.o: beltrami.o cubit3d.o
p3_hifi.o: job3.o debug.o
p3_diagnose.o: p3_hifi.o
p3_ct.o: p3_diagnose.o
p3_condense.o: p3_hifi.o
p3_interior.o: p3_ct.o
p3_face.o: p3_ct.o
p3_rj.o: p3_condense.o p3_interior.o p3_face.o
p3_snes.o: p3_rj.o
p3_grid.o: p3_snes.o
p3_advance.o: p3_grid.o
driver.o: p3_advance.o

realclean: clean
        rm -f *.o *.mod *.out *.bin *.dat *.fld *.diff *.err *~ temp*
```

A.4 Physics Module Creation and Compiling

A physics module that links with the HiFi library is the main user interface. It allows the user to specify parameters, initial conditions, boundary conditions, the equation system, and the geometry. This separates the user from the core code, such that they can focus on their specific equation system and unique problem of interest.

First the physics module must be created. A template physics module is included with the code called '`physics_templ.f`'. This file is a template for all the required functions that are called by the main solver. Additional complete physics modules (e.g. vmhd.f, heat.f, spheromak.f) are also included with the code.

For compiling the physics module, and linking to the main solver, and the external libraries, there is an included makefile. It is similar to the main solver makefile, but in this case it only compiles the desired physics modules, and then links to the main solver and external libraries. Prior to invoking the `make` command one must specify the environment variable `PHYSICS`, which should have the same name as the `<physics>.f` file. An example makefile is included below. The resulting executable file will be named from the `PHYSICS` environment variable.

```
# This is the makefile for HiFi on Hopper.
# Before compiling HiFi, load the necessary modules with
# the following commands:
#
# module load petsc/3.1.04
# module load hdf5-parallel
# module load netcdf
#
# Then, export PHYSICS environment variable to be the name of
# the [physics_templ].f application file you would like to
# compile with the following command:
#
# export PHYSICS=physics_templ
#
# where "physics_templ" should be replaced with the name of
# your physics application file.
#

FFLAGS = -O3 -Munroll=c:4 -Msave \
         -I./solver \
         $(HDF5) $(NETCDF) -I$(HDF5_DIR)
FC = ftn $(FFLAGS)
F90 = ftn $(FFLAGS)

# libraries

LIBS = \
```

```
        -L./solver -lhifi \
        $(PETSC_FORTRAN_LIB) \
        $(PETSC_LIB) \
        $(HDF5) \
        $(NETCDF)

# objects

OBJECTS = \
        $(PHYSICS).o

# targets

all: libhifi $(PHYSICS)

libhifi:
        cd ./solver; make

$(PHYSICS): $(OBJECTS) chkopts
        $(FLINKER) -o $(PHYSICS) $(OBJECTS) $(LIBS)
        rm -f *.cpp *.i

include $(PETSC_DIR)/conf/variables
include $(PETSC_DIR)/conf/rules

#dependencies

$(OBJECTS): solver/libhifi.a

realclean: clean
        rm -f *.o *.mod *.out *.bin *.dat *.fld *.diff *.err *~ temp* \
        HiFi.o* $(PHYSICS)
```

A.5 CAD Model Import and Mesh Creation With CUBIT

In many cases, especially with complex geometries, one will want to use an external mesh file rather then analytically specifying the logical to physical mapping for the domain geometry. In the multi-block formulation each block must have a logical to physical mapping, further necessitating the need for an external mesh generation. One way to do this is with the CUBIT software, which can read in CAD files, partition the volume into blocks, and create a hexahedral mesh. This appendix section will be a tutorial on how accomplish the creation of a successful mesh.

A.5.1 CAD file input

Importing CAD files in CUBIT is simple. The native format used in CUBIT is ACIS, which is a common format that many CAD programs can export. One can either use the GUI menus (File $\rightarrow$ Import...) command line, or a script file to give the import

command. The command line/script statement looks like:
import acis ‘ ‘/ path/to/ file /HIT_SI_CAD.sat’ attributes_on separate_bodies’,
and will create a volume in CUBIT. Figure A.1 shows a screen shot of resulting volume imported into the software.

A.5.2 Alternative: Geometry Creation Within CUBIT

An alternative method of creating a geometry is to use the native CAD tools within the CUBIT program. One can either use the GUI interface or a command line/script interface. The documentation for how to use the software is found on the the CUBIT website [85].

An example script of geometry creation is shown below. In this particular case the HIT-SI confinement region is created. It first creates the cross-sectional surface, and then rotates the surface to define the volume.

```
##########################
## Create HIT-SI Geometry
##########################
# CUBIT 13.0 Build 46571
##########################
reset

#{rad=1.796051}
#{sq_ratio=1.75}

######################
## Create Surface
######################
# Vertices
create vertex 0 0 0
create vertex 0 4.03430245 0
create vertex 1.79605122 4.77825122 0
create vertex 25.46802561 28.45022561 0
create vertex 25.84 29.34825122 0
create vertex 25.84 29.6731 0
create vertex 40.34294648 29.6731 0
create vertex 40.34294648 27.75568243 0
create vertex 40.48152483 27.17880719 0
create vertex 52.45754091 3.69032476 0
create vertex 53.58896256 2.9972 0
create vertex 55.60218540 2.9972 0
create vertex 55.60218540 0 0

# Arc Centers
create vertex 0 6.574302 0
create vertex 24.57 29.348251 0
create vertex 41.612946 27.755682 0
create vertex 53.588963 4.2672 0
```

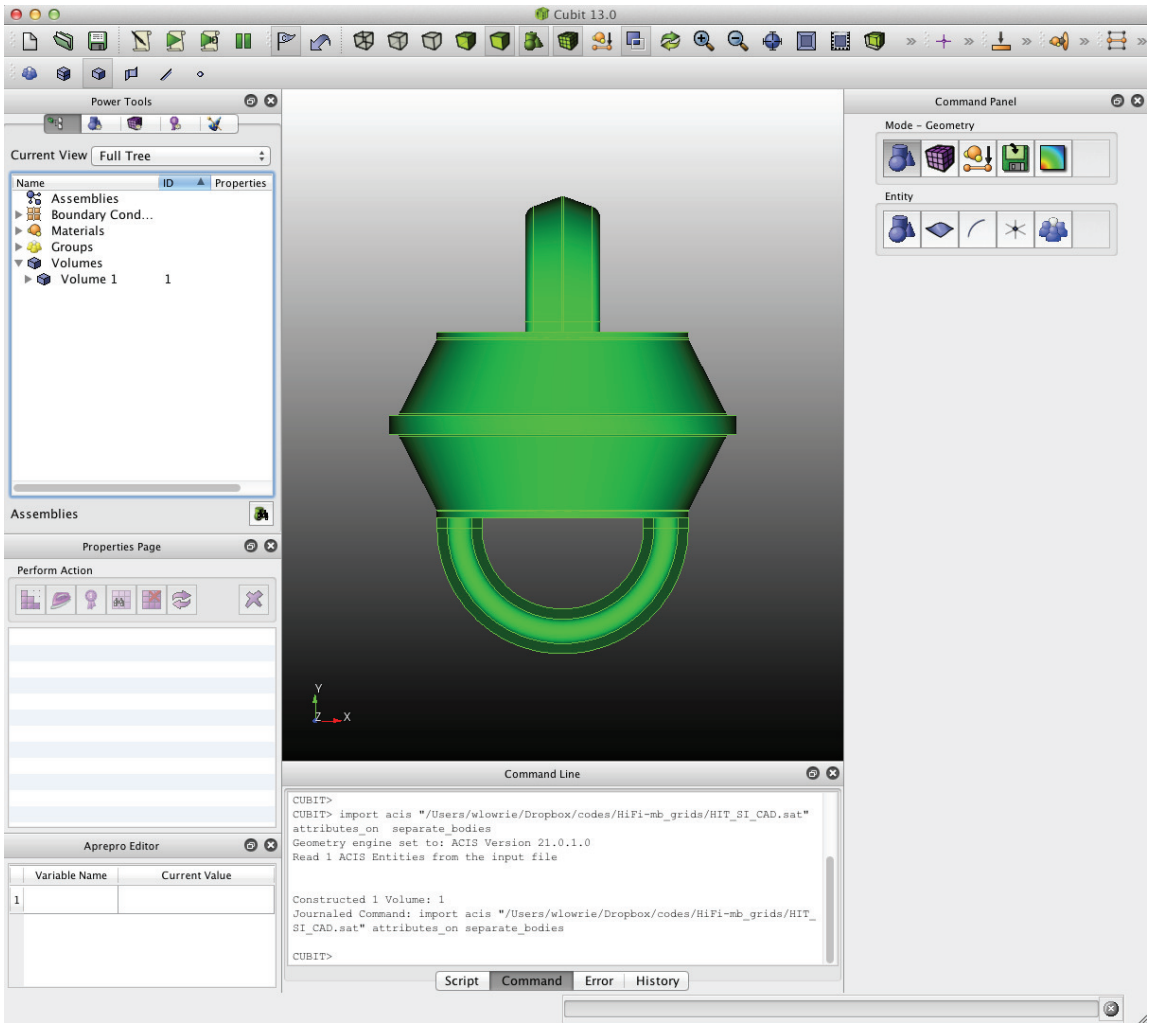


Figure A.1: CUBIT Screenshot showing a CAD volume that has been imported

```
# Curves
create curve vertex 1 vertex 2
create curve arc center vertex 14  2  3  radius 2.54
create curve vertex 3 vertex 4
create curve arc center vertex 15 4 5 radius 1.27
create curve vertex 5 vertex 6
create curve vertex 6 vertex 7
create curve vertex 7 vertex 8
create curve arc center vertex 16 8 9 radius 1.27
create curve vertex 9 vertex 10
create curve arc center vertex 17 10 11 radius 1.27
create curve vertex 11 vertex 12
create curve vertex 12 vertex 13
create curve vertex 13 vertex 1

# Delete Arc Center Vertices
delete vertex 14 15 16 17

# Surface
create surface curve 1 2 3 4 5 6 7 8 9 10 11 12 13

## Reflect Surfaces
Surface 1  copy reflect y nomesh

## Unite the Surfaces
unite volume 1 2

## Rotate the Surface to Create the Volume
sweep surface 1 yaxis angle 360
```

A.5.3 Geometry Partitioning

The geometry partitioning is a more involved process, and requires careful consideration of the topological shape of each partition. The multi-block framework requires that each partition, or block be a logical cube, such that a structured mapped mesh can be created for each.

There are several tools in the CUBIT software to accomplish a successful partitioning. The main tool used is called a `webcut`. This tool allows one to cut up a volume or collection of volumes into partitions using several geometric constructs. These include planes, other volumes, swept surfaces, swept curves, and more. A full list of possible ways to `webcut` a volume is in the CUBIT documentation [85]. Like any of the features in CUBIT, they can be accomplished either through the GUI or command line/script. Below is an example script that partitions the HIT-SI geometry into logical cube blocks.

```
##########################
## Create HIT–SI Geometry
##########################
# CUBIT 13.0 Build 46571
```

```
##########################
reset

#{rad=1.796051}
#{sq_ratio=1.75}
#{inj_sq_ratio=1.25}
#{inj_sq=4.1529/inj_sq_ratio}


##########################
## Import the CAD Geometry
##########################
import acis "/Users/wlowrie/Dropbox/codes/HiFi-mb_grids/HIT_SI_CAD.sat" attributes_on
    separate_bodies

imprint all
merge all

# Delete Injectors From Geometry (They Have a Small Error)
webcut volume 1  with plane yplane offset 29.6731 noimprint nomerge
webcut volume 1  with plane yplane offset -29.6731 noimprint nomerge
delete Volume 2
delete Volume 1

## Create the Injectors with Correct Geometry
create vertex  9.293423  29.6731 29.195499
create vertex  0  29.6731 25.84 on curve 35
create vertex -9.293423  29.6731  29.195499
create vertex -9.293423  29.6731  37.000398
create vertex 0  29.6731 40.342946 on curve 36
create vertex  9.293423  29.6731  37.000398
create vertex -7.873047 29.6731 33.097949
create vertex 7.873047 29.6731 33.097949

create curve vertex 85  vertex 86 on surface 2
create curve vertex 86 vertex 87 on surface 2
create curve arc center vertex 91 87 88 radius 4.1529 on surface 2
create curve vertex 88 vertex 89 on surface 2
create curve vertex 89 vertex 90 on surface 2
create curve arc center vertex 92 90 85 radius 4.1529 on surface 2

create vertex -7.873047 29.6731 {33.097949+inj_sq/2}
create vertex -7.873047 29.6731 {33.097949-inj_sq/2}
create vertex 7.873047 29.6731 {33.097949+inj_sq/2}
create vertex 7.873047 29.6731 {33.097949-inj_sq/2}

create curve vertex 88  vertex 99  on surface 2
create curve vertex 99  vertex 100  on surface 2
create curve vertex 100  vertex 87  on surface 2
create curve vertex 99  vertex 101  on surface 2
create curve vertex 101  vertex 90  on surface 2
create curve vertex 101  vertex 102  on surface 2
create curve vertex 102  vertex 85  on surface 2
create curve vertex 102  vertex 100  on surface 2

create surface curve 135 141 140 139  on surface 2
create surface curve 142 140 146 144  on surface 2
create surface curve 143 138 145 144  on surface 2
create surface curve 145 133 146 134 141  on surface 2
create surface curve 142 139 136 137 143  on surface 2
delete Vertex 91 92

sweep surface 77 76 75 78 74  vector 0 1 0 distance 3.3782
```

```
sweep surface 77 76 75 78 74 axis 0 33.0513 0 1 0 0 angle -180
sweep surface 77 76 75 78 74  vector 0 -1 0 distance 3.3782

Volume 4 5 6 7 8  copy rotate 180 about z nomesh
rotate Volume 9 10 11 12 13 angle 90  about Y include_merged

imprint all
merge all

webcut volume 3  sweep surface 89  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 84  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 105  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 94  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 100  vector 0 -1 0 distance 60

webcut volume 3  sweep surface 77  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 76  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 75  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 74  vector 0 -1 0 distance 60
webcut volume 3  sweep surface 78  vector 0 -1 0 distance 60

webcut volume 3  sweep surface 208  vector 0 1 0 distance 60
webcut volume 3  sweep surface 191  vector 0 1 0 distance 60
webcut volume 3  sweep surface 177  vector 0 1 0 distance 60
webcut volume 3  sweep surface 163  vector 0 1 0 distance 60
webcut volume 3  sweep surface 225  vector 0 1 0 distance 60

webcut volume 3  sweep surface 207  vector 0 1 0 distance 60
webcut volume 3  sweep surface 162  vector 0 1 0 distance 60
webcut volume 3  sweep surface 176  vector 0 1 0 distance 60
webcut volume 3  sweep surface 190  vector 0 1 0 distance 60
webcut volume 3  sweep surface 224  vector 0 1 0 distance 60

imprint all
merge all

########################
## Partition the Geometry
########################
## Make Radial Cuts with Center Cylinder/Square
brick x {rad/sq_ratio} y {2*4.7782512} z {rad/sq_ratio}
rotate Volume 34  angle 45  about Y
chop volume 3  with volume 34

webcut volume 36 28 24 26 29 31 33 with plane zplane noimprint nomerge
webcut volume 37 23 21 19 36 15 17 18 with plane xplane noimprint nomerge

imprint all
merge all

webcut volume 36 48 44 37  with cylinder radius 53.58896256 axis y
webcut volume 52 53 54 55  with cylinder radius 52.45754091 axis y
webcut volume 56 57 58 59  with cylinder radius 40.342946 axis y
webcut volume 60 61 62 63  with cylinder radius 25.84 axis y
webcut volume 64 65 66 67  with cylinder radius 1.79605122 axis y
webcut volume 64 65 66 67  with cylinder radius 25.46802561 axis y
webcut volume 56 57 58 59  with cylinder radius 40.48152483 axis y

imprint all
merge all

webcut volume 60 61 62 63  with cylinder radius 30.63894373 axis y
webcut volume 60 61 63 62  with cylinder radius 38.14966793 axis y
```

```
imprint all
merge all

create curve vertex 509  vertex 472
create curve vertex 510  vertex 471
webcut volume 85 81 61 32 29 31 33 43 42 41 30 87 62 82 22 46 45 47 19 23 21 20 63 86 83
    27 39 40 38 24 26 28 25 80 84 60 14 15 17 18 49 50 16 51  sweep curve 1709  yaxis
    angle 360
webcut volume 87 62 82 41 30 42 43 29 31 33 32 85 61 81 16 51 50 49 18 17 15 14 84 60 80
    25 24 26 28 38 40 39 27 83 86 63 23 20 45 21 19 47 46 22  sweep curve 1710  yaxis
    angle 360
delete Curve 1709 1710

imprint all
merge all

webcut volume 5 7 8 4 6  with plane xplane noimprint nomerge
webcut volume 11 13 10 12 9  with plane zplane noimprint nomerge
webcut volume 179 9 181 180 12 11 10 13  with plane yplane offset -33.0513 noimprint
    nomerge
webcut volume 177 4 176 178 8 5 7 6  with plane yplane offset 33.0513 noimprint nomerge

imprint all
merge all
```

Notice the 'imprint all' and 'merge all' statements. They simplify the model by combining vertices, curves, and faces that are conformal into a single entity. This is essential before meshing, so that CUBIT knows to match the mesh on conformal geometric features. Figure A.2 shows a screenshot of a CUBIT geometry partitioned into blocks. Each block is represented by a different color.

A.5.4 Mesh Creation

Once the geometry is partitioned into logical blocks, the mesh can be created. The partitions are created as logical blocks, such that a mapped structured mesh can be created within each one. Each block is effectively meshed separately, but will depend on any conformal adjacent block edges and faces that have been previously meshed. In order to make sure one creates the desired mesh, the intervals of each edge can be specified explicitly. Additionally one must make careful consideration of the intervals on of the block and make sure it is devisable by intended polynomial degree np to be used in the simulation. When the meshing process is scripted, one can make the intervals an input parameter such that many different resolution can be created with minimal effort. Below is an example script of the mesh creation process.

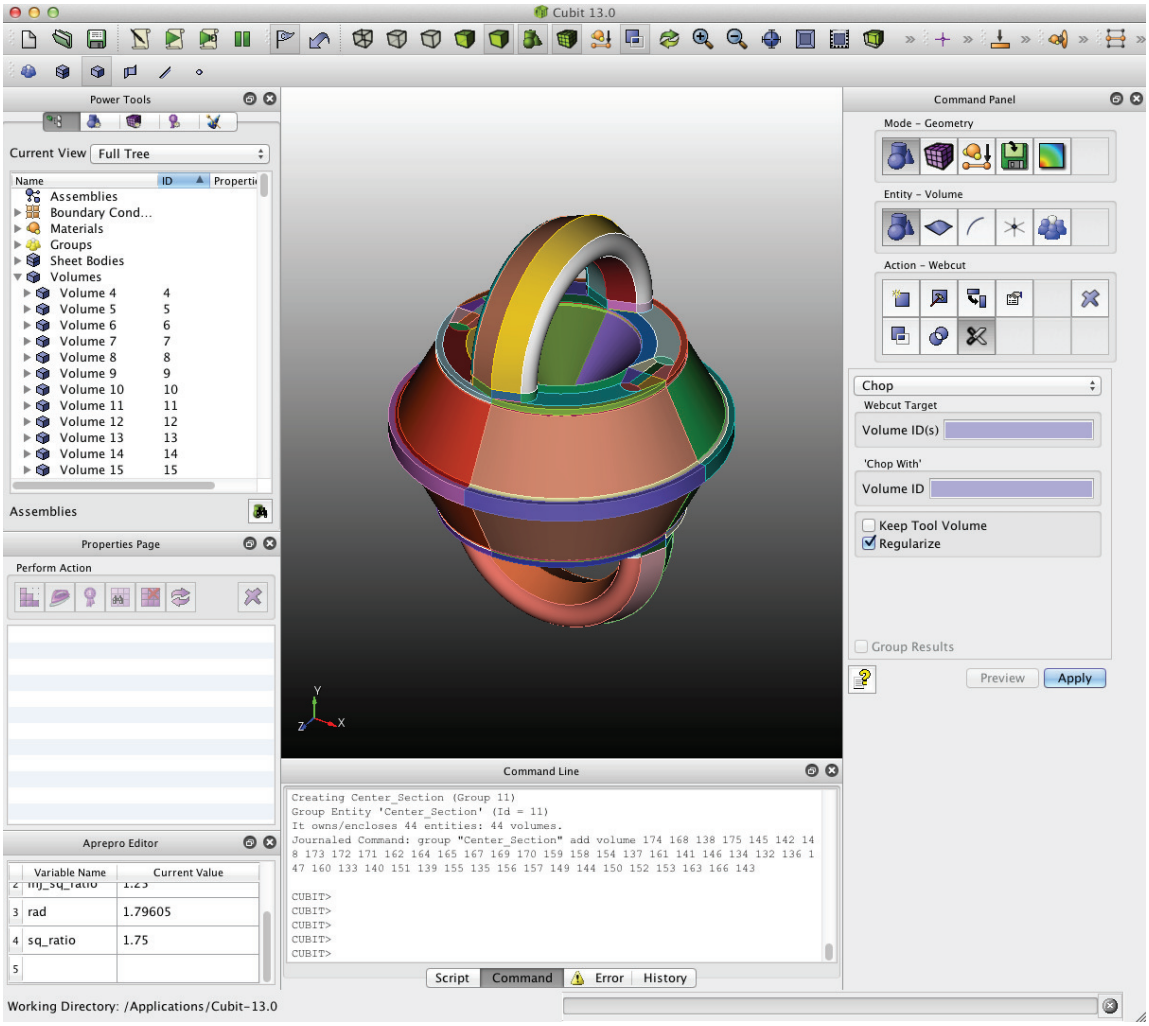


Figure A.2: CUBIT Screenshot showing a partitioned volume, where each color represents a different block.

```
########################
## Create HIT-SI Mesh
########################
# CUBIT 13.0 Build 46571
########################
reset

########################
## Import the Geometry
########################
open "/Users/wlowrie/Dropbox/codes/HiFi-mb_grids/HIT_SI_Geometry_InjSq1250.cub"

####################
## Set the Mesh Sizes
####################
#{np=3}
#{sq_nxz=6*np}
#{diag_nxz1=6*np}
#{diag_nxz2=12*np}
#{diag_nxz3=6*np}

#{ctr_diag1=4*np}
#{ctr_diag2=8*np}

#{diag_dnstc=2*np}

#{inj_rad=4*np}
#{inj_gap_ny=1*np}
#{inj_riser_ny=1*np}
#{inj=18*np}

#{ny=8*np}



####################
## Center Cylinder
####################
# Top Side
curve 892 894 885 889 1109 1143 1249 1211  interval {sq_nxz}
curve 892 894 885 889 1109 1143 1249 1211  scheme bias factor 1.0
curve 1090 916 1195 910  interval {diag_nxz1}
curve 1090 916 1195 910  scheme bias factor 1.0
surface 554 441 575 644 616  scheme map
mesh surface 554 441 575 644 616

# y-Length
curve 1458 1475 884 888 886 1468 1457 891  interval {ny}
curve 1458 1475 884 888 886 1468 1457 891  scheme bias factor 1.0

# Bottom Side
curve 1114 887 893 1244 890 1142 1219 883  interval {sq_nxz}
curve 1114 887 893 1244 890 1142 1219 883  scheme bias factor 1.0
curve 1089 909 1196 917  interval {diag_nxz1}
curve 1089 909 1196 917  scheme bias factor 1.0

# Volumes
volume 35 70 71 68 69  scheme Map
mesh volume 35 70 71 68 69

####################
## Conic Section
####################
```

```
# Top Side
curve 1212 692 1108 688 1144 777 1250 775  interval {sq_nxz}
curve 1212 692 1108 688 1144 777 1250 775  scheme bias factor 1.0
curve 1194 911 1091 915  interval {diag_nxz2}
curve 1194 911 1091 915  scheme bias factor 1.0
curve 912 1193 914 1092  interval {np}
curve 912 1193 914 1092  scheme bias factor 1.0
surface 576 553 617 643  scheme map
mesh surface 576 553 617 643
surface 645 577 618 552  scheme map
mesh surface 645 577 618 552

# y-Length
curve 684 533 592 768  interval {ny}
curve 684 533 592 768  scheme bias factor 1.0
curve 1487 1498 1488 1505  interval {ny}
curve 1487 1498 1488 1505  scheme bias factor 1.0

# Bottom Side
curve 1141 1218 1115 1245 691 690 778 773 interval {sq_nxz}
curve 1141 1218 1115 1245 691 690 778 773 scheme bias factor 1.0
curve 918 1094 908 1197  interval {diag_nxz2}
curve 918 1094 908 1197  scheme bias factor 1.0
curve 913 1192 907 1093  interval {np}
curve 913 1192 907 1093 scheme bias factor 1.0

# Volumes
volume 72 75 74 73  scheme Map
mesh volume 72 75 74 73

# Thin Volumes
volume 64 65 66 67  scheme Map
mesh volume 64 65 66 67


########################
## Center Section
########################
# Top Side
curve 692 2187 2196 749 688 2040 2029 752 777 1871 1869 835 775 1713 1711 839  interval {
    sq_nxz}
curve 692 2187 2196 749 688 2040 2029 752 777 1871 1869 835 775 1713 1711 839  scheme bias
     factor 1.0
curve 2055 2126 2186 1727 1728 1840 1897 1968  interval {ctr_diag1}
curve 2055 2126 2186 1727 1728 1840 1897 1968  scheme bias factor 1.0
curve 2169 2199 1712 1714 1856 1872 2011 2041  interval {ctr_diag2}
curve 2169 2199 1712 1714 1856 1872 2011 2041  scheme bias factor 1.0
curve 2212 1739 1738 1809 1885 1981 2027 2155  interval {ctr_diag1}
curve 2212 1739 1738 1809 1885 1981 2027 2155  scheme bias factor 1.0

surface 1239 931 1045 1151  scheme map
mesh surface 1239 931 1045 1151
surface 1247 921 1027 1141  scheme map
mesh surface 1247 921 1027 1141
surface 1257 939 1037 1133  scheme map
mesh surface 1257 939 1037 1133

# y-Length
curve 533 684 592 768  interval {ny}
curve 533 684 592 768  scheme bias factor 1.0
curve 2689 2658 2523 2821 2851 2358 2357 2460  interval {ny}
curve 2689 2658 2523 2821 2851 2358 2357 2460  scheme bias factor 1.0
curve 2687 745 2791 2625 575 2521 2491 831 2355 2356 659 2864  interval {ny}
```

```
curve 2687 745 2791 2625 575 2521 2491 831 2355 2356 659 2864  scheme bias factor 1.0

# Bottom Side
curve 778 2509 2511 836 753 2677 2675 691 690 2842 2854 750 773 2343 2345 838  interval {
    sq_nxz}
curve 778 2509 2511 836 753 2677 2675 691 690 2842 2854 750 773 2343 2345 838  scheme bias
     factor 1.0
curve 2445 2372 2370 2843 2809 2704 2643 2538  interval {ctr_diag1}
curve 2445 2372 2370 2843 2809 2704 2643 2538  scheme bias factor 1.0
curve 2496 2512 2660 2678 2828 2855 2344 2346  interval {ctr_diag2}
curve 2496 2512 2660 2678 2828 2855 2344 2346  scheme bias factor 1.0
curve 2526 2612 2692 2778 2869 2359 2360 2478  interval {ctr_diag1}
curve 2526 2612 2692 2778 2869 2359 2360 2478  scheme bias factor 1.0

# Smooth Surfaces that Need It
surface 939 921 931 1257 1247 1239 1037 1027 1045 1133 1141 1151 smooth scheme winslow
smooth surface 939 921 931 1257 1247 1239 1037 1027 1045 1133 1141 1151
surface 939 921 931 1257 1247 1239 1037 1027 1045 1133 1141 1151 smooth scheme mean ratio
    cpu 10
smooth surface 939 921 931 1257 1247 1239 1037 1027 1045 1133 1141 1151

# Volumes
volume 154  redistribute nodes on
volume 154  autosmooth target off
volume 154  scheme Sweep  source surface 1247  target surface 1557  rotate off
volume 154  sweep smooth Copy
mesh volume 154
volume 166  redistribute nodes on
volume 166  autosmooth target off
volume 166  scheme Sweep  source surface 1141  target surface 1671  rotate off
volume 166  sweep smooth Copy
mesh volume 166
volume 132  redistribute nodes on
volume 132  autosmooth target off
volume 132  scheme Sweep  source surface 1027  target surface 1345  rotate off
volume 132  sweep smooth Copy
mesh volume 132
volume 143  redistribute nodes on
volume 143  autosmooth target off
volume 143  scheme Sweep  source surface 921  target surface 1451  rotate off
volume 143  sweep smooth Copy
mesh volume 143

volume 156 155 165 167 134 133 145 144  scheme Map
mesh volume 156 155 165 167 134 133 145 144

########################
## Injector Footprints
########################
# Top Side
curve 2300 2286 2254 2241 1780 1767 1824 1811 1926 1924 1983 1970 2083 2096 2142 2128
    interval {ctr_diag1}
curve 2300 2286 2254 2241 1780 1767 1824 1811 1926 1924 1983 1970 2083 2096 2142 2128
    scheme bias factor 1.0
curve 1712 2299 2253 2224 2199 2169 2141 2095 2068 2041 2011 1997 1923 1910 1872 1856 1825
     1779 1751 1714  interval {ctr_diag2}
curve 1712 2299 2253 2224 2199 2169 2141 2095 2068 2041 2011 1997 1923 1910 1872 1856 1825
     1779 1751 1714  scheme bias factor 1.0
curve 1954 1969 1984 1939 1907 1909 1841 1766 1812 1795 1752 1750 2285 2313 2269 2240 2223
     2225 2127 2158 2111 2065 2067 2082  interval {inj_rad}
curve 1954 1969 1984 1939 1907 1909 1841 1766 1812 1795 1752 1750 2285 2313 2269 2240 2223
     2225 2127 2158 2111 2065 2067 2082  scheme bias factor 1.0
surface 967 957 1007 997 987 1017 947 977  scheme map
```

```
mesh surface 967 957 1007 997 987 1017 947 977
surface 1305 1315 1335 1325 1295 1285 1275 1265  scheme map
mesh surface 1305 1315 1335 1325 1295 1285 1275 1265
surface 1229 1199 1209 1219 1189 1179 1169 1159  scheme map
mesh surface 1229 1199 1209 1219 1189 1179 1169 1159
surface 1093 1123 1113 1103 1073 1063 1083 1053  scheme map
mesh surface 1093 1123 1113 1103 1073 1063 1083 1053

# y-Length
curve 2561 2559 2575 2626 2642 2591  interval {ny}
curve 2561 2559 2575 2626 2642 2591  scheme bias factor 1.0
curve 2808 2792 2758 2742 2728 2726  interval {ny}
curve 2808 2792 2758 2742 2728 2726  scheme bias factor 1.0
curve 2893 2908 2939 2891 2973 2926  interval {ny}
curve 2893 2908 2939 2891 2973 2926  scheme bias factor 1.0
curve 2396 2409 2394 2426 2475 2459  interval {ny}
curve 2396 2409 2394 2426 2475 2459  scheme bias factor 1.0

# Bottom Side
curve 2447 2464 2414 2384 2961 2914 2881 2930 2794 2780 2746 2732 2628 2614 2565 2582
    interval {ctr_diag1}
curve 2447 2464 2414 2384 2961 2914 2881 2930 2794 2780 2746 2732 2628 2614 2565 2582
    scheme bias factor 1.0
curve 2496 2461 2415 2400 2346 2344 2975 2929 2896 2855 2828 2793 2747 2678 2660 2627 2581
     2550 2512  interval {ctr_diag2}
curve 2496 2461 2415 2400 2346 2344 2975 2929 2896 2855 2828 2793 2747 2678 2660 2627 2581
     2550 2512  scheme bias factor 1.0
curve 2381 2448 2479 2430 2399 2383 2962 2943 2897 2882 2880 2913 2812 2779 2762 2729 2715
     2713 2646 2613 2564 2549 2547 2595  interval {inj_rad}
curve 2381 2448 2479 2430 2399 2383 2962 2943 2897 2882 2880 2913 2812 2779 2762 2729 2715
     2713 2646 2613 2564 2549 2547 2595  scheme bias factor 1.0

# Volumes
volume 170 168 171 174 173 172 169 175 149 148 147 150 151 152 153 146 135 137 138 136 139
     140 141 142 164 162 163 158 159 157 160 161  scheme Map
mesh volume 170 168 171 174 173 172 169 175 149 148 147 150 151 152 153 146 135 137 138
    136 139 140 141 142 164 162 163 158 159 157 160 161


########################
## Outter Section
########################
# Top Side
curve 1242 1215 1137 1112  interval {sq_nxz}
curve 1242 1215 1137 1112  scheme bias factor 1.0
curve 1206 920 1101 936  interval {diag_nxz3}
curve 1206 920 1101 936  scheme bias factor 1.0
curve 1100 919 1198 928  interval {np}
curve 1100 919 1198 928  scheme bias factor 1.0
curve 1243 1136 1214 1113  interval {sq_nxz}
curve 1243 1136 1214 1113  scheme bias factor 1.0
surface 620 558 634 568  scheme map
mesh surface 620 558 634 568
surface 559 619 635 567  scheme map
mesh surface 559 619 635 567

# y-Length
curve 1518 1517 1535 1528  interval {ny}
curve 1518 1517 1535 1528  scheme bias factor 1.0
curve 1528 1428 1517 1417 1518 1418 1535 1435  interval {ny}
curve 1528 1428 1517 1417 1518 1418 1535 1435  scheme bias factor 1.0

# Bottom Side
```

```
curve 1146 1247 1209 1105  interval {sq_nxz}
curve 1146 1247 1209 1105  scheme bias factor 1.0
curve 926 1200 930 1098  interval {diag_nxz3}
curve 926 1200 930 1098  scheme bias factor 1.0
curve 929 1099 927 1199  interval {np}
curve 929 1099 927 1199  scheme bias factor 1.0
curve 1210 1248 1147 1104  interval {sq_nxz}
curve 1210 1248 1147 1104  scheme bias factor 1.0

# Volumes
volume 79 78 77 76  scheme Map
mesh volume 79 78 77 76
volume 56 59 58 57  scheme Map
mesh volume 56 59 58 57

########################
## Diagnostic Gap
########################
# Top Side
curve 1102 935 1205 921  interval {np}
curve 1102 935 1205 921  scheme bias factor 1.0
curve 1216 1111 1138 1241  interval {sq_nxz}
curve 1216 1111 1138 1241  scheme bias factor 1.0
surface 569 557 633 621  scheme map
mesh surface 569 557 633 621
curve 934 1204 922 1103  interval {diag_dnstc}
curve 934 1204 922 1103  scheme bias factor 1.0
curve 1110 1139 1239 1217  interval {sq_nxz}
curve 1110 1139 1239 1217  scheme bias factor 1.0
surface 570 632 556 622  scheme map
mesh surface 570 632 556 622

# y-Length
curve 1398 1387 1388 1405  interval {ny}
curve 1398 1387 1388 1405  scheme bias factor 1.0
curve 1095 923 1203 933  interval {ny}
curve 1095 923 1203 933  scheme bias factor 1.0

# Bottom Side
curve 931 1097 925 1201  interval {np}
curve 931 1097 925 1201  scheme bias factor 1.0
curve 1246 1207 1145 1106  interval {sq_nxz}
curve 1246 1207 1145 1106  scheme bias factor 1.0
curve 924 1202 932 1096  interval {diag_dnstc}
curve 924 1202 932 1096  scheme bias factor 1.0
curve 1107 1140 1208 1240  interval {sq_nxz}
curve 1107 1140 1208 1240  scheme bias factor 1.0

# Volumes
volume 53 52 54 55  scheme Map
mesh volume 53 52 54 55
volume 44 37 36 48  scheme Map
mesh volume 44 37 36 48

########################
## Injector Gaps
########################
# Top Gap - Top Side
curve 689 1611 1639 748 687 1633 1673 751 776 1627 1689 834 774 1620 1657 837  interval {
    sq_nxz}
curve 689 1611 1639 748 687 1633 1673 751 776 1627 1689 834 774 1620 1657 837  scheme bias
     factor 1.0
```

```
curve 686 1034 1022 747 743 1016 1027 682 133 1261 1274 137 136 1282 1266 134 772 1054
    1068 833 829 1076 1060 766 252 1178 1163 272 269 1158 1170 243  interval {ctr_diag1}
curve 686 1034 1022 747 743 1016 1027 682 133 1261 1274 137 136 1282 1266 134 772 1054
    1068 833 829 1076 1060 766 252 1178 1163 272 269 1158 1170 243  scheme bias factor 1.0

curve 735 725 1038 712 708 138 144 1269 140 135 819 809 1063 798 797 279 264 1168 256 260
     interval {ctr_diag2}
curve 735 725 1038 712 708 138 144 1269 140 135 819 809 1063 798 797 279 264 1168 256 260
     scheme bias factor 1.0
curve 733 1015 710 680 1024 675 246 253 1156 1183 250 274 764 794 1078 1053 770 821 139
    1284 143 145 1259 141  interval {inj_rad}
curve 733 1015 710 680 1024 675 246 253 1156 1183 250 274 764 794 1078 1053 770 821 139
    1284 143 145 1259 141  scheme bias factor 1.0

# y-Length
curve 744 2166 2151 2137 2136 2110 2092 685 2036 2079 2077 2050  interval {inj_gap_ny}
curve 744 2166 2151 2137 2136 2110 2092 685 2036 2079 2077 2050  scheme bias factor 1.0
curve 660 1938 1937 1992 1993 1978 1979 1116 1883 1921 1919 1881  interval {inj_gap_ny}
curve 660 1938 1937 1992 1993 1978 1979 1116 1883 1921 1919 1881  scheme bias factor 1.0
curve 943 1794 1778 769 1820 1821 1836 1851 1725 1762 1764 1723  interval {inj_gap_ny}
curve 943 1794 1778 769 1820 1821 1836 1851 1725 1762 1764 1723  scheme bias factor 1.0
curve 1726 2309 2296 1724 1220 2268 2252 1213 2195 2238 2236 2210  interval {inj_gap_ny}
curve 1726 2309 2296 1724 1220 2268 2252 1213 2195 2238 2236 2210  scheme bias factor 1.0

# Volumes
volume 88  redistribute nodes on
volume 88  autosmooth target off
volume 88  scheme Sweep  source surface 921  target surface 900  rotate off
volume 88  sweep smooth Copy
mesh volume 88
volume 122  redistribute nodes on
volume 122  autosmooth target off
volume 122  scheme Sweep  source surface 1247  target surface 888  rotate off
volume 122  sweep smooth Copy
mesh volume 122
volume 111  redistribute nodes on
volume 111  autosmooth target off
volume 111  scheme Sweep  source surface 1141  target surface 909  rotate off
volume 111  sweep smooth Copy
mesh volume 111
volume 99  redistribute nodes on
volume 99  autosmooth target off
volume 99  scheme Sweep  source surface 1027  target surface 919  rotate off
volume 99  sweep smooth Copy
mesh volume 99

volume 110 121 123 124 127 126 125 128 129 131 90 130 89 94 93 92 91 97 96 95 98 100 101
    103 105 106 112 107 104 102 108 109 113 114 115 116 119 118 117 120  scheme Map
mesh volume 110 121 123 124 127 126 125 128 129 131 90 130 89 94 93 92 91 97 96 95 98 100
    101 103 105 106 112 107 104 102 108 109 113 114 115 116 119 118 117 120

# Bottom Gap - Bottom Side
curve 601 1632 1671 665 603 1626 1687 662 537 1619 1655 578 536 1612 1641 579  interval {
    sq_nxz}
curve 601 1632 1671 665 603 1626 1687 662 537 1619 1655 578 536 1612 1641 579  scheme bias
     factor 1.0
curve 400 1033 1023 439 437 1017 1028 402 531 1260 1275 577 573 1280 1267 535 406 1069
    1055 371 384 1061 1074 410 589 1180 1164 657 661 1157 1169 594  interval {ctr_diag1}
curve 400 1033 1023 439 437 1017 1028 402 531 1260 1275 577 573 1280 1267 535 406 1069
    1055 371 384 1061 1074 410 589 1180 1164 657 661 1157 1169 594  scheme bias factor 1.0
curve 624 622 1166 637 649 287 294 1065 325 353 557 556 1271 510 514 367 339 1036 308 313
     interval {ctr_diag2}
```

```
curve 624 622 1166 637 649 287 294 1065 325 353 557 556 1271 510 514 367 339 1036 308 313
     scheme bias factor 1.0
curve 626 1154 1182 600 647 596 291 1051 350 343 1080 297 527 553 1286 512 507 1257 364
    369 1013 311 314 1026  interval {inj_rad}
curve 626 1154 1182 600 647 596 291 1051 350 343 1080 297 527 553 1286 512 507 1257 364
    369 1013 311 314 1026  scheme bias factor 1.0

# y-Length
curve 1583 2884 2932 593 2347 2966 2349 2917 2857 2886 2899 2844  interval {inj_gap_ny}
curve 1583 2884 2932 593 2347 2966 2349 2917 2857 2886 2899 2844  scheme bias factor 1.0
curve 2784 2783 2799 2814 1552 2749 2733 1557 2680 2719 2717 2682  interval {inj_gap_ny}
curve 2784 2783 2799 2814 1552 2749 2733 1557 2680 2719 2717 2682  scheme bias factor 1.0
curve 576 2568 2584 534 2617 2618 2633 2649 2516 2554 2552 2514  interval {inj_gap_ny}
curve 576 2568 2584 534 2617 2618 2633 2649 2516 2554 2552 2514  scheme bias factor 1.0
curve 832 2417 2385 1571 2451 2452 2468 2484 2348 2402 2387 2350  interval {inj_gap_ny}
curve 832 2417 2385 1571 2451 2452 2468 2484 2348 2402 2387 2350  scheme bias factor 1.0

# Volumes
volume 87  redistribute nodes on
volume 87  autosmooth target off
volume 87  scheme Sweep  source surface 1345  target surface 918  rotate off
volume 87  sweep smooth Copy
mesh volume 87
volume 85  redistribute nodes on
volume 85  autosmooth target off
volume 85  scheme Sweep  source surface 1451  target surface 898  rotate off
volume 85  sweep smooth Copy
mesh volume 85
volume 86  redistribute nodes on
volume 86  autosmooth target off
volume 86  scheme Sweep  source surface 1671  target surface 908  rotate off
volume 86  sweep smooth Copy
mesh volume 86
volume 84  redistribute nodes on
volume 84  autosmooth target off
volume 84  scheme Sweep  source surface 1557  target surface 889  rotate off
volume 84  sweep smooth Copy
mesh volume 84

volume 62 82 30 43 42 41 29 31 33 61 32 81 51 16 49 50 15 18 17 14 60 80 25 28 26 24 38 40
     39 27 63 23 21 19 20 45 46 47 22 83 scheme Map
mesh volume 62 82 30 43 42 41 29 31 33 61 32 81 51 16 49 50 15 18 17 14 60 80 25 28 26 24
    38 40 39 27 63 23 21 19 20 45 46 47 22 83


#######################
## Injector 1
#######################
# Top Side - Injector Riser
curve 186 3014 3016 160 158 3026 3028 188 202 3015 3013 228 230 3027 3025 199  interval {
    ctr_diag1}
curve 186 3014 3016 160 158 3026 3028 188 202 3015 3013 228 230 3027 3025 199  scheme bias
     factor 1.0
curve 168 172 3225 180 196 209 214 3226 222 238  interval {ctr_diag2}
curve 168 172 3225 180 196 209 214 3226 222 238  scheme bias factor 1.0
curve 212 204 3206 3237 232 208 162 3205 155 190 3236 170  interval {inj_rad}
curve 212 204 3206 3237 232 208 162 3205 155 190 3236 170  scheme bias factor 1.0
curve 169 171 163 161 159 3012 3007 187 189 179 157 156 255 245 3010 3009 254 247 244 249
    262 273 251  interval {inj_riser_ny}
curve 169 171 163 161 159 3012 3007 187 189 179 157 156 255 245 3010 3009 254 247 244 249
    262 273 251  scheme bias factor 1.0

# Riser Volumes
```

```
volume 206 212 209 211 198 202 204 200 199 201 205 210 203 208 207 213  scheme Map
mesh volume 206 212 209 211 198 202 204 200 199 201 205 210 203 208 207 213

# Injector
curve 203 200 205 201 213 207 3011 221 3008 231 229  interval {inj}
curve 203 200 205 201 213 207 3011 221 3008 231 229  scheme bias factor 1.0

# Injector Volumes
volume 8 178 6 4 7 177 176 5  scheme Map
mesh volume 8 178 6 4 7 177 176 5

########################
## Injector 2
########################
# Bottom Side - Injector Riser
curve 427 3077 3099 387 390 3089 3067 425 385 3098 3076 412 408 3066 3088 373  interval {
    ctr_diag1}
curve 427 3077 3099 387 390 3089 3067 425 385 3098 3076 412 408 3066 3088 373  scheme bias
     factor 1.0
curve 356 328 3154 306 303 289 296 3153 326 354  interval {ctr_diag2}
curve 356 328 3154 306 303 289 296 3153 326 354  scheme bias factor 1.0
curve 300 305 3144 3120 359 362 3143 3119 298 293 352 345  interval {inj_rad}
curve 300 305 3144 3120 359 362 3143 3119 298 293 352 345  scheme bias factor 1.0
curve 288 292 295 290 374 3083 3063 407 346 318 323 351 310 307 309 312 436 3065 3085 399
    363 335 338 366  interval {inj_riser_ny}
curve 288 292 295 290 374 3083 3063 407 346 318 323 351 310 307 309 312 436 3065 3085 399
    363 335 338 366  scheme bias factor 1.0

# Riser Volumes
volume 184 187 189 183 196 194 190 192 186 191 197 193 195 185 182 188  scheme Map
mesh volume 184 187 189 183 196 194 190 192 186 191 197 193 195 185 182 188

# Injector
curve 299 301 424 304 302 3064 3084 327 355 386 329 357  interval {inj}
curve 299 301 424 304 302 3064 3084 327 355 386 329 357  scheme bias factor 1.0

# Injector Volumes
volume 181 12 10 13 11 9 179 180  scheme Map
mesh volume 181 12 10 13 11 9 179 180
```

A.5.5 Block Creation

An important step before exporting the mesh to a file is the specification of block numbers. The user can decide what ordering they would like for the blocks, and this will be how they are identified in HiFi. The following commands in CUBIT are used to create the block numberings.

```
########################
## Assign Blocks
########################
set duplicate block elements off
# Center Cylinder
block 1 volume 35
block 2 volume 69
block 3 volume 70
block 4 volume 71
```

```
block 5 volume 68

# Conic Section
block 6 volume 73
block 7 volume 74
block 8 volume 75
block 9 volume 72
block 10 volume 65
block 11 volume 66
block 12 volume 67
block 13 volume 64

# Center Section
block 14 volume 145
block 15 volume 134
block 16 volume 165
block 17 volume 156
block 18 volume 143
block 19 volume 132
block 20 volume 166
block 21 volume 154
block 22 volume 144
block 23 volume 133
block 24 volume 167
block 25 volume 155

block 26 volume 142
block 27 volume 139
block 28 volume 140
block 29 volume 141
block 30 volume 135
block 31 volume 137
block 32 volume 138
block 33 volume 136

block 34 volume 175
block 35 volume 173
block 36 volume 174
block 37 volume 170
block 38 volume 172
block 39 volume 171
block 40 volume 168
block 41 volume 169

block 42 volume 164
block 43 volume 163
block 44 volume 162
block 45 volume 161
block 46 volume 158
block 47 volume 159
block 48 volume 160
block 49 volume 157

block 50 volume 153
block 51 volume 152
block 52 volume 151
block 53 volume 150
block 54 volume 149
block 55 volume 148
block 56 volume 147
block 57 volume 146

# Outer Section
```

```
block 58 volume 77
block 59 volume 78
block 60 volume 79
block 61 volume 76

block 62 volume 57
block 63 volume 58
block 64 volume 59
block 65 volume 56

# Diagnostic Gap
block 66 volume 53
block 67 volume 54
block 68 volume 55
block 69 volume 52

block 70 volume 48
block 71 volume 44
block 72 volume 37
block 73 volume 36

# Injector Gap 1
block 74 volume 89
block 75 volume 101
block 76 volume 112
block 77 volume 121
block 78 volume 88
block 79 volume 99
block 80 volume 111
block 81 volume 122
block 82 volume 90
block 83 volume 100
block 84 volume 110
block 85 volume 123

block 86 volume 91
block 87 volume 92
block 88 volume 93
block 89 volume 94
block 90 volume 97
block 91 volume 96
block 92 volume 95
block 93 volume 98

block 94 volume 102
block 95 volume 105
block 96 volume 103
block 97 volume 104
block 98 volume 106
block 99 volume 108
block 100 volume 107
block 101 volume 109

block 102 volume 113
block 103 volume 114
block 104 volume 115
block 105 volume 116
block 106 volume 117
block 107 volume 118
block 108 volume 119
block 109 volume 120

block 110 volume 124
```

```
block 111 volume 125
block 112 volume 126
block 113 volume 127
block 114 volume 128
block 115 volume 129
block 116 volume 131
block 117 volume 130

# Injector Gap 2
block 118 volume 81
block 119 volume 82
block 120 volume 83
block 121 volume 80

block 122 volume 85
block 123 volume 87
block 124 volume 86
block 125 volume 84

block 126 volume 61
block 127 volume 62
block 128 volume 63
block 129 volume 60

block 130 volume 32
block 131 volume 29
block 132 volume 31
block 133 volume 33
block 134 volume 41
block 135 volume 42
block 136 volume 43
block 137 volume 30

block 138 volume 22
block 139 volume 47
block 140 volume 46
block 141 volume 45
block 142 volume 19
block 143 volume 21
block 144 volume 23
block 145 volume 20

block 146 volume 27
block 147 volume 39
block 148 volume 40
block 149 volume 38
block 150 volume 24
block 151 volume 26
block 152 volume 28
block 153 volume 25

block 154 volume 14
block 155 volume 15
block 156 volume 17
block 157 volume 18
block 158 volume 49
block 159 volume 50
block 160 volume 51
block 161 volume 16

# Injector 1
block 162 volume 201
block 163 volume 199
```

```
block 164 volume 203
block 165 volume 205
block 166 volume 210
block 167 volume 208
block 168 volume 207
block 169 volume 213

block 170 volume 212
block 171 volume 211
block 172 volume 209
block 173 volume 206
block 174 volume 198
block 175 volume 202
block 176 volume 204
block 177 volume 200

block 178 volume 4
block 179 volume 177
block 180 volume 176
block 181 volume 178
block 182 volume 7
block 183 volume 5
block 184 volume 8
block 185 volume 6

# Injector 2
block 186 volume 193
block 187 volume 191
block 188 volume 195
block 189 volume 197
block 190 volume 186
block 191 volume 188
block 192 volume 182
block 193 volume 185

block 194 volume 184
block 195 volume 187
block 196 volume 189
block 197 volume 183
block 198 volume 190
block 199 volume 194
block 200 volume 196
block 201 volume 192

block 202 volume 11
block 203 volume 12
block 204 volume 10
block 205 volume 13
block 206 volume 181
block 207 volume 180
block 208 volume 179
block 209 volume 9
```

A.5.6 Mesh File Export

The last step in CUBIT before the mesh can be used in HiFi is exporting the mesh and block data. This is a simple process and like any of the other processes in CUBIT can

be done graphically via the GUI or by command. The export command can be included at the end of the mesh script, and the command is shown below.

```
##########################
## Write the Mesh to File
##########################
set large exodus file off
export Genesis  "/Users/wlowrie/Desktop/HIT_SI_Mesh_209Block_InjSq1250_Sq06_MR_W_np3.g"
    dimension 3 block all
```

When exporting the mesh file to a Genesis filetype, it is important to make sure it's exported in 3D with the dimension 3 keyword, and all blocks are included with block all keyword. This ensures that the file is in the correct format for HiFi, all dimensional data is included, and all the block numberings are included in the output file.

A.6 Input File (hifi.in) Specification

The hifi.in input file is similar to the single block version. The major difference is in the 'block_input' list section. This includes a variable to specify the number of processor cores per block. It is a list separated by spaces for each of the blocks. Additionally one can optionally specify the block resolution and processor partitioning variables nx, ny, nz, and nbx, and nbz. These are not required and are automatically determined from the CUBIT mesh.

```
&algorithm_input

        solve_type="condense"
        step_type="theta"
        theta=.5

        adapt_dt=f
        errtol=1.e-4
        ksp_restart=500
        always_pc_reset=t
        mat_prealloc=t

        itmax=100
        itmax_incr=4
        itmax_decr=5
        dt_incr=1.2
        dt_decr=.6

        nodal=f
        quad_type="gl0"

        grid_type="cubit"
```

```
        grid_inv_type="uniform"
        gridtol1=5e-2
        gridtol2=1.

        monitor=t
        fd_test=f
        mesh_metrics=f

        outfile_type="hdf5"
        parallel_write=t
        parallel_read=t
/

&universal_input

        np=3
        nq=5

        dt=1.e-9
        dtmax=1.e-1
        tmax=250.
        nstep=2000
        dmout=1

        cubit_BCs=f
        cubit_file="../../grids/HIT_SI_Meshes/HIT_SI_Mesh_209Block_InjSq1250_Sq06_MR_W_np3
            .g"
        outdir="../../results/H20111017_209blk_001/raw"
        restart_run=f
        restart_init=f
        restart_dir="."
        restart_step=1
        restart_np=3
        grid_step=0
/

&block_input

        mb_diagnostics=t
        proc_per_blk= 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
             32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
             32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 04 04
             04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04
             04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04
             04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04
             04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 72 72
             72 72 72 72 72 72 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 72 72 72 72
             72 72 72 72
/

&vmhd_list

        init_type="HIT-SI"
        source=f

        lx=0.1
        ly=0.1
        lz=0.320
        a=0.080

        etar=0.e-3
        mu=0e-3
        muz=0e-3
```

```
        nu=0e-6
        kap=0.e-1
        rhodiff=0
        B0=2.0
        rho0=1.0
        p0=1.0
        lambda=0.103366
        kx=1.
        ky=1.
        vz_shear=0.0
        delta=0.e-3
/
```

A.7 Running HiFi-mb

Like HiFi, the HiFi-mb executables are run with whatever MPI executable commands the particular system is using. For example one could use 'mpirun' or 'mpiexec' to initiate the HiFi-mb executable. On large shared systems like those seen at NERSC an other facilities, a resource manager and job scheduler are used. The user is required to submit the job for execution using whatever system they have in place. For example on the Hopper system at NERSC one would use a script like the one below to initiate a job.

```
#PBS -N             HiFi-mb
#PBS -A             m489
#PBS -j             oe
#PBS -S             /bin/bash
#PBS -m             abe

#PBS -q             debug
#PBS -l  walltime=00:10:00
#PBS -l  mppwidth=3984
#PBS -l  mppnppn=24

export CRAY_ROOTFS=DSL
ulimit -S -c 0
cd $PBS_O_WORKDIR

aprun -n 3968 -N 24 ./vmhd \
-ksp_type fgmres \
-pc_type asm \
-pc_asm_overlap 1 \
-sub_pc_type lu \
-sub_pc_factor_mat_solver_package superlu_dist \
-ksp_rtol 1e-10 \
-snes_monitor \
-ksp_monitor \
-options_left \
-log_summary > log.out 2> log.err
```

Appendix B

EQUATION LINEARIZATION AND EIGENSYSTEM

B.1 Linearization of the dissipative MHD equations and solving for the resulting eigensystem

The linearization starts with the full dissipative MHD equation system:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\vec{v}\\ \vec{A}\\ \frac{1}{\gamma-1}p\end{bmatrix}+\nabla\cdot\begin{bmatrix}\rho\vec{v}\\ \rho\vec{v}\vec{v}-\vec{B}\vec{B}+\left(p+\frac{1}{2}\vec{B}\cdot\vec{B}\right)\vec{I}-\mu\nabla\vec{v}\\ -\eta\nabla\vec{B}\\ \frac{\gamma}{\gamma-1}p\vec{v}-\kappa\nabla T\end{bmatrix}=\begin{bmatrix}0\\ 0\\ \vec{v}\times\vec{B}\\ (\vec{v}\cdot\nabla p+\eta\vec{j}\cdot\vec{j}+\mu(\nabla\vec{v}:\nabla\vec{v})\end{bmatrix} \tag{B.1}$$

For clarity it is expanded out of its vector form as:

$$
\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho v_x\\ \rho v_y\\ \rho v_z\\ A_x\\ A_y\\ A_z\\ \frac{1}{\gamma-1}p\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho v_x\\ \rho v_x^2+p+\frac{1}{2}\vec{B}\cdot\vec{B}-(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})^2-\mu\frac{\partial}{\partial x}(\frac{\rho v_x}{\rho})\\ \rho v_x v_y-(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})-\mu\frac{\partial}{\partial x}(\frac{\rho v_y}{\rho})\\ \rho v_x v_z-(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})-\mu\frac{\partial}{\partial x}(\frac{\rho v_z}{\rho})\\ +\eta\nabla(\frac{\partial A_y}{\partial y}+\frac{\partial A_z}{\partial z})\\ -\eta\nabla\frac{\partial A_y}{\partial x}\\ -\eta\nabla\frac{\partial A_z}{\partial x}\\ \frac{\gamma}{\gamma-1}pv_x-\kappa\frac{\partial}{\partial x}(\frac{p}{\rho})\end{bmatrix}
$$

$$
+\frac{\partial}{\partial y}\begin{bmatrix}\rho v_y\\ \rho v_y v_x-(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})-\mu\frac{\partial}{\partial y}(\frac{\rho v_x}{\rho})\\ \rho v_y^2+p+\frac{1}{2}\vec{B}\cdot\vec{B}-(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})^2-\mu\frac{\partial}{\partial y}(\frac{\rho v_y}{\rho})\\ \rho v_y v_z-(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})-\mu\frac{\partial}{\partial x}(\frac{\rho v_z}{\rho})\\ -\eta\nabla\frac{\partial A_x}{\partial y}\\ +\eta\nabla(\frac{\partial A_x}{\partial x}+\frac{\partial A_z}{\partial z})\\ -\eta\nabla\frac{\partial A_z}{\partial y}\\ \frac{\gamma}{\gamma-1}pv_y-\kappa\frac{\partial}{\partial y}(\frac{p}{\rho})\end{bmatrix}+\frac{\partial}{\partial z}\begin{bmatrix}\rho v_z\\ \rho v_z v_x-(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})-\mu\frac{\partial}{\partial z}(\frac{\rho v_x}{\rho})\\ \rho v_z v_y-(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})-\mu\frac{\partial}{\partial z}(\frac{\rho v_y}{\rho})\\ \rho v_z^2+p+\frac{1}{2}\vec{B}\cdot\vec{B}-(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})^2-\mu\frac{\partial}{\partial z}(\frac{\rho v_z}{\rho})\\ -\eta\nabla\frac{\partial A_x}{\partial z}\\ -\eta\nabla\frac{\partial A_y}{\partial z}\\ +\eta\nabla(\frac{\partial A_x}{\partial x}+\frac{\partial A_y}{\partial y})\\ \frac{\gamma}{\gamma-1}pv_z-\kappa\frac{\partial}{\partial z}(\frac{p}{\rho})\end{bmatrix}
$$

$$
=\begin{bmatrix}0\\ 0\\ 0\\ 0\\ v_y(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})-v_z(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})\\ v_z(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})-v_x(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y})\\ v_x(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x})-v_y(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z})\\ (v_x\frac{\partial}{\partial x}+v_y\frac{\partial}{\partial y}+v_z\frac{\partial}{\partial z})p+\eta\vec{j}\cdot\vec{j}+\mu(\nabla\vec{v}:\nabla\vec{v})\end{bmatrix} \tag{B.2}
$$

where,

$$
\vec{B}\cdot\vec{B}=(\nabla\times\vec{A})\cdot(\nabla\times\vec{A})=\left(\frac{\partial A_z}{\partial y}-\frac{\partial A_y}{\partial z}\right)^2+\left(\frac{\partial A_x}{\partial z}-\frac{\partial A_z}{\partial x}\right)^2+\left(\frac{\partial A_y}{\partial x}-\frac{\partial A_x}{\partial y}\right)^2. \tag{B.3}
$$

From here each primary variable is assumed to be composed of a background quantity,

and a perturbation quantity:

$$u = u_0 + \delta\tilde{u}, \tag{B.4}$$

with δ assumed in the form

$$\delta \equiv e^{-i(\vec{k}\cdot\vec{x}-\omega t)}. \tag{B.5}$$

In this particular case, the background density and pressure are assumed to be uniform, and the background velocity is zero. For the magnetics, a uniform magnetic field intensity aligned in the $\hat{x}$-direction is assumed, and the other directions are zero. Equation B.4 is substituted into equation B.2. All terms higher than the first order perturbation are removed, and the result is converted into a spectral space where $\frac{\partial}{\partial t} = -i\omega$, $\frac{\partial}{\partial x} = ik_x$, $\frac{\partial}{\partial y} = ik_y$, and $\frac{\partial}{\partial z} = ik_z$. The equations are simplified and result as:

- Equation 1:

$$-i\omega\tilde{\rho} + ik_x\rho_0\tilde{v}_x + ik_y\rho_0\tilde{v}_y + ik_z\rho_0\tilde{v}_z = 0 \tag{B.6}$$

- Equation 2:

$$\begin{aligned} &-i\omega\rho_0\tilde{v}_x + ik_x\tilde{p} + \mu k^2\tilde{v}_x \\ &-ik_xB_{x_0}(ik_y\tilde{A}_z - ik_z\tilde{A}_y) + ik_yB_{x_0}(ik_z\tilde{A}_{x_1} - ik_x\tilde{A}_z) - ik_zB_{x_0}(ik_x\tilde{A}_y - ik_y\tilde{A}_x) = 0 \end{aligned} \tag{B.7}$$

- Equation 3:

$$\begin{aligned} &-i\omega\rho_0\tilde{v}_y + ik_y\tilde{p} + \mu k^2\tilde{v}_y \\ &-ik_xB_{x_0}(ik_z\tilde{A}_x - ik_x\tilde{A}_z) + ik_yB_{x_0}(ik_y\tilde{A}_z - ik_z\tilde{A}_y) = 0 \end{aligned} \tag{B.8}$$

- Equation 4:

$$\begin{aligned} &-i\omega\rho_0\tilde{v}_z + ik_z\tilde{p} + \mu k^2\tilde{v}_z \\ &-ik_xB_{x_0}(ik_x\tilde{A}_y - ik_y\tilde{A}_x) + ik_zB_{x_0}(ik_y\tilde{A}_z - ik_z\tilde{A}_y) = 0 \end{aligned} \tag{B.9}$$

- Equation 5:

$$-i\omega\tilde{A}_x - k_x k_y \eta \tilde{A}_y - k_x k_z \eta \tilde{A}_z + k_y^2 \eta \tilde{A}_x + k_z^2 \eta \tilde{A}_x = 0 \tag{B.10}$$

- Equation 6:

$$-i\omega\tilde{A}_y - k_y k_x \eta \tilde{A}_x - k_y k_z \eta \tilde{A}_z + k_x^2 \eta \tilde{A}_y + k_z^2 \eta \tilde{A}_y + i\tilde{v}_z B_{x_0} = 0 \tag{B.11}$$

- Equation 7:

$$-i\omega\tilde{A}_z - k_z k_x \eta \tilde{A}_x - k_z k_y \eta \tilde{A}_y + k_x^2 \eta \tilde{A}_z + k_y^2 \eta \tilde{A}_z - i\tilde{v}_y B_{x_0} = 0 \tag{B.12}$$

- Equation 8:

$$-i\omega\tilde{p} + ik_x\gamma p_0 \tilde{v}_x + ik_y\gamma p_0 \tilde{v}_y + ik_z\gamma p_0 \tilde{v}_z + (\gamma-1)\kappa k^2 \frac{1}{\rho_0}\tilde{p} - (\gamma-1)\kappa k^2 \frac{p_0}{\rho_0^2}\tilde{\rho} = 0 \tag{B.13}$$

B.1.1 Solve for Eigensystem

The simplified system of equations in spectral space can now be organized into an eigenvalue problem of the form:

$$\mathbf{A}\vec{\tilde{u}} = \omega\vec{\tilde{u}} \tag{B.14}$$

where,

$$\vec{\tilde{u}} = \begin{bmatrix} \tilde{\rho} \\ \tilde{v}_x \\ \tilde{v}_y \\ \tilde{v}_z \\ \tilde{A}_x \\ \tilde{A}_y \\ \tilde{A}_z \\ \tilde{p} \end{bmatrix} \tag{B.15}$$

and

$$\mathbf{A} = \begin{bmatrix}
0 & k_x\rho_0 & k_y\rho_0 & k_z\rho_0 & 0 & 0 & 0 & 0 \\
0 & -\frac{i\mu k^2}{\rho_0} & 0 & 0 & 0 & 0 & 0 & \frac{k_x}{\rho_0} \\
0 & 0 & -\frac{i\mu k^2}{\rho_0} & 0 & -\frac{ik_xk_zB_{x_0}}{\rho_0} & -\frac{ik_yk_zB_{x_0}}{\rho_0} & +\frac{i(k_x^2+k_y^2)B_{x_0}}{\rho_0} & \frac{k_y}{\rho_0} \\
0 & 0 & 0 & -\frac{i\mu k^2}{\rho_0} & +\frac{ik_xk_yB_{x_0}}{\rho_0} & -\frac{i(k_y^2+k_z^2)B_{x_0}}{\rho_0} & +\frac{ik_zk_yB_{x_0}}{\rho_0} & \frac{k_z}{\rho_0} \\
0 & 0 & 0 & 0 & -i(k_y^2+k_z^2)\eta & ik_xk_y\eta & ik_xk_z\eta & 0 \\
0 & 0 & 0 & +iB_{x_0} & +ik_zk_x\eta & +ik_zk_y\eta & -i(k_x^2+k_y^2)\eta & 0 \\
0 & 0 & -iB_{x_0} & 0 & +ik_zk_x\eta & +ik_zk_y\eta & -i(k_x^2+k_y^2)\eta & 0 \\
\frac{i(\gamma-1)\kappa k^2p_0}{\rho_0^2} & k_x\gamma p_0 & k_y\gamma p_0 & k_z\gamma p_0 & 0 & 0 & 0 & -\frac{i(\gamma-1)\kappa k^2}{\rho_0}
\end{bmatrix} \tag{B.16}$$

Equation B.14 can be solved as a eigenvalue problem. The resulting eigensystem from this equation will find 8 different eigenvalues and eigenvectors. For the linearized MHD case, 2 eigenvalues will be null, and the other six represent the forward and backward propagating slow magnetosonic, fast magnetosonic, and shear Alfvén waves. The corresponding eigenvectors represent the perturbation necessary to initialize the particular wave.

Appendix C

EQUATION NORMALIZATION

C.1 Equations

C.1.1 Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0 \tag{C.1}$$

$$\frac{\rho_0}{t_0}\frac{\partial \tilde{\rho}}{\partial \tilde{t}} + \frac{\rho_0 v_0}{L_0}\tilde{\nabla} \cdot [\tilde{\rho}\tilde{\mathbf{v}}] = 0 \tag{C.2}$$

$$\frac{L_0}{t_0 v_0}\frac{\partial \tilde{\rho}}{\partial \tilde{t}} + \tilde{\nabla} \cdot [\tilde{\rho}\tilde{\mathbf{v}}] = 0 \tag{C.3}$$

$$\frac{L_0}{t_0 v_0} = 1 \quad \Rightarrow \quad \boxed{t_0 = \frac{L_0}{v_0}} \tag{C.4}$$

$$\boxed{\frac{\partial \tilde{\rho}}{\partial \tilde{t}} + \tilde{\nabla} \cdot [\tilde{\rho}\tilde{\mathbf{v}}] = 0} \tag{C.5}$$

C.1.2 Momentum Equation

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot [\rho \mathbf{v}\mathbf{v} + p\mathbf{I} - \mu \nabla \mathbf{v}] = \mathbf{j} \times \mathbf{B} \tag{C.6}$$

$$\frac{\rho_0 v_0}{t_0}\frac{\partial (\tilde{\rho}\tilde{\mathbf{v}})}{\partial \tilde{t}} + \frac{1}{L_0}\tilde{\nabla} \cdot \left[\rho_0 v_0^2 \tilde{\rho}\tilde{\mathbf{v}}\tilde{\mathbf{v}} + p_0 \tilde{p}\mathbf{I} - \frac{v_0}{L_0}\mu \tilde{\nabla}\tilde{\mathbf{v}}\right] = j_0 B_0 (\tilde{\mathbf{j}} \times \tilde{\mathbf{B}}) \tag{C.7}$$

$$\boxed{p_0 = \rho_0 v_0^2}, \quad \boxed{\tilde{\mu} = \frac{\mu t_0}{\rho_0 L_0^2}}$$

$$\rho_0 v_0^2 \frac{\partial (\tilde{\rho}\tilde{\mathbf{v}})}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left[\rho_0 v_0^2 \tilde{\rho}\tilde{\mathbf{v}}\tilde{\mathbf{v}} + \rho_0 v_0^2 \tilde{p}\mathbf{I} - \frac{v_0}{L_0}\mu \tilde{\nabla}\tilde{\mathbf{v}}\right] = j_0 B_0 L_0 (\tilde{\mathbf{j}} \times \tilde{\mathbf{B}}) \tag{C.8}$$

$$\frac{\partial (\tilde{\rho}\tilde{\mathbf{v}})}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left[\tilde{\rho}\tilde{\mathbf{v}}\tilde{\mathbf{v}} + \tilde{p}\mathbf{I} - \tilde{\mu}\tilde{\nabla}\tilde{\mathbf{v}}\right] = \frac{j_0 B_0 L_0}{\rho_0 v_0^2}(\tilde{\mathbf{j}} \times \tilde{\mathbf{B}}) \tag{C.9}$$

$$\mu_0 \mathbf{j} = \nabla \times \mathbf{B} \quad \Rightarrow \quad \frac{\mu_0 j_0 L_0}{B_0}\tilde{\mathbf{j}} = \tilde{\nabla} \times \tilde{\mathbf{B}} \tag{C.10}$$

$$\frac{\mu_0 j_0 L_0}{B_0} = 1 \quad \Rightarrow \quad \boxed{j_0 = \frac{B_0}{\mu_0 L_0}}$$

$$\frac{B_0^2}{\mu_0 \rho_0 v_0^2} = 1 \quad \Rightarrow \quad v_A^2 = v_0^2 = \frac{B_0^2}{\mu_0 \rho_0} \quad \Rightarrow \quad \boxed{v_A = v_0 = \frac{B_0}{\sqrt{\mu_0 \rho_0}}}$$

$$\boxed{\frac{\partial(\tilde{\rho}\tilde{\mathbf{v}})}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left[\tilde{\rho}\tilde{\mathbf{v}}\tilde{\mathbf{v}} + \tilde{p}\mathbf{I} - \tilde{\mu}\tilde{\nabla}\tilde{\mathbf{v}}\right] = (\tilde{\mathbf{j}} \times \tilde{\mathbf{B}})} \tag{C.11}$$

C.1.3 Ohm's Law (Vector Potential) Equation

$$\frac{\partial \mathbf{A}}{\partial t} + \nabla \cdot [-\nu \nabla \mathbf{j}] = \mathbf{v} \times \mathbf{B} - \eta \mathbf{j} \tag{C.12}$$

$$\frac{A_0}{t_0}\frac{\partial \tilde{\mathbf{A}}}{\partial \tilde{t}} + \frac{1}{L_0}\tilde{\nabla} \cdot \left[-\frac{j_0}{L_0}\nu \tilde{\nabla}\tilde{\mathbf{j}}\right] = v_0 B_0 \tilde{\mathbf{v}} \times \tilde{\mathbf{B}} - j_0 \eta \tilde{\mathbf{j}} \tag{C.13}$$

$$A_0 v_0 \frac{\partial \tilde{\mathbf{A}}}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left[-\frac{j_0}{L_0}\nu \tilde{\nabla}\tilde{\mathbf{j}}\right] = v_0 B_0 L_0 \tilde{\mathbf{v}} \times \tilde{\mathbf{B}} - j_0 L_0 \eta \tilde{\mathbf{j}} \tag{C.14}$$

$$\mathbf{B} = \nabla \times \mathbf{A} \quad \Rightarrow \quad \frac{B_0 L_0}{A_0}\tilde{\mathbf{B}} = \tilde{\nabla} \times \tilde{\mathbf{A}} \tag{C.15}$$

$$\frac{B_0 L_0}{A_0} = 1 \quad \Rightarrow \quad \boxed{A_0 = B_0 L_0} \tag{C.16}$$

$$B_0 L_0 v_0 \frac{\partial \tilde{\mathbf{A}}}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left[-\frac{j_0}{L_0}\nu \tilde{\nabla}\tilde{\mathbf{j}}\right] = v_0 B_0 L_0 \tilde{\mathbf{v}} \times \tilde{\mathbf{B}} - \frac{B_0}{\mu_0}\eta \tilde{\mathbf{j}} \tag{C.17}$$

$$\boxed{\tilde{\nu} \equiv \frac{\nu}{\mu_0 v_0 L_0^3}}, \quad \boxed{\tilde{\eta} \equiv \frac{\eta}{\mu_0 v_0 L_0}}$$

$$\boxed{\frac{\partial \tilde{\mathbf{A}}}{\partial \tilde{t}} + \tilde{\nabla} \cdot \left[-\tilde{\nu}\tilde{\nabla}\tilde{\mathbf{j}}\right] = \tilde{\mathbf{v}} \times \tilde{\mathbf{B}} - \tilde{\eta}\tilde{\mathbf{j}}} \tag{C.18}$$

C.1.4 Pressure Equation

$$\left(\frac{1}{\gamma-1}\right)\frac{\partial p}{\partial t}+\nabla\cdot\left[\frac{\gamma}{\gamma-1}p\mathbf{v}-\kappa\nabla T\right]=\mathbf{v}\cdot\nabla p+\eta\mathbf{j}\cdot\mathbf{j}+\mu(\nabla\mathbf{v}:\nabla\mathbf{v}) \tag{C.19}$$

$$\rho_0 v_0^3\left(\frac{1}{\gamma-1}\right)\frac{\partial\tilde{p}}{\partial\tilde{t}}+\tilde{\nabla}\cdot\left[\rho_0 v_0^3\frac{\gamma}{\gamma-1}\tilde{p}\tilde{\mathbf{v}}-\kappa\frac{T_0}{L_0}\tilde{\nabla}\tilde{T}\right]=\rho_0 v_0^3\tilde{\mathbf{v}}\cdot\tilde{\nabla}\tilde{p}+\frac{B_0^2}{\mu_0^2 L_0}\eta\tilde{\mathbf{j}}\cdot\tilde{\mathbf{j}}+\frac{v_0^2}{L_0}\mu(\tilde{\nabla}\tilde{\mathbf{v}}:\tilde{\nabla}\tilde{\mathbf{v}}) \tag{C.20}$$

$$\boxed{\tilde{\kappa}\equiv\frac{\kappa T_0}{\rho_0 v_0^3 L_0}}$$

$$\boxed{\left(\frac{1}{\gamma-1}\right)\frac{\partial\tilde{p}}{\partial\tilde{t}}+\tilde{\nabla}\cdot\left[\frac{\gamma}{\gamma-1}\tilde{p}\tilde{\mathbf{v}}-\tilde{\kappa}\tilde{\nabla}\tilde{T}\right]=\tilde{\mathbf{v}}\cdot\tilde{\nabla}\tilde{p}+\tilde{\eta}\tilde{\mathbf{j}}\cdot\tilde{\mathbf{j}}+\tilde{\mu}(\tilde{\nabla}\tilde{\mathbf{v}}:\tilde{\nabla}\tilde{\mathbf{v}})} \tag{C.21}$$

C.1.5 Current Density Equation

$$\nabla\cdot[(\nabla\cdot\mathbf{A})\mathbf{I}-\nabla\mathbf{A}]=\mathbf{j} \tag{C.22}$$

$$\frac{1}{L_0}\tilde{\nabla}\cdot\left[\frac{A_0}{L_0}(\tilde{\nabla}\cdot\tilde{\mathbf{A}})\mathbf{I}-\frac{A_0}{L_0}\tilde{\nabla}\tilde{\mathbf{A}}\right]=\frac{B_0}{\mu_0 L_0}\tilde{\mathbf{j}} \tag{C.23}$$

$$\boxed{\tilde{\nabla}\cdot\left[(\tilde{\nabla}\cdot\tilde{\mathbf{A}})\mathbf{I}-\tilde{\nabla}\tilde{\mathbf{A}}\right]=\frac{1}{\mu_0}\tilde{\mathbf{j}}} \tag{C.24}$$

C.2 Summary of Normalizations

$$\boxed{t_0=\frac{L_0}{v_0}},\quad \boxed{p_0=\rho_0 v_0^2},$$

$$\boxed{j_0=\frac{B_0}{\mu_0 L_0}},\quad \boxed{v_A=v_0=\frac{B_0}{\sqrt{\mu_0\rho_0}}},\quad \boxed{A_0=B_0 L_0}$$

$$\boxed{\tilde{\mu}=\frac{\mu t_0}{\rho_0 L_0^2}},\quad \boxed{\tilde{\nu}\equiv\frac{\nu}{\mu_0 v_0 L_0^3}},\quad \boxed{\tilde{\eta}\equiv\frac{\eta}{\mu_0 v_0 L_0}},\quad \boxed{\tilde{\kappa}\equiv\frac{\kappa T_0}{\rho_0 v_0^3 L_0}}$$

C.3 Initial Condition - Bennett Z-Pinch Profile

C.3.1 Vector Potential

$$\tilde{B}_\theta = \frac{I_0}{2\pi}\frac{r}{r^2+a^2}\frac{1}{B_0} \quad \Longrightarrow \quad \tilde{A}_z = -\frac{I_0}{4\pi}\mathbf{Log}[r^2+a^2]\frac{1}{A_0} \tag{C.25}$$

$$\boxed{\tilde{A}_z = -\frac{I_0}{4\pi}\mathbf{Log}[r^2+a^2]\frac{1}{v_A\sqrt{\mu_0\rho_0}L_0}} \tag{C.26}$$

C.3.2 Current Density

$$\tilde{J}_z = \frac{I_0}{\mu_0\pi}\frac{a^2}{(r^2+a^2)^2}\frac{1}{j_0} \tag{C.27}$$

$$\boxed{\tilde{J}_z = \frac{I_0}{\mu_0\pi}\frac{a^2}{(r^2+a^2)^2}\frac{\sqrt{\mu_0}L_0}{v_A\sqrt{\rho_0}}} \tag{C.28}$$

C.3.3 Pressure

$$\tilde{p} = \frac{I_0^2}{\mu_0 8\pi^2}\frac{a^2}{(r^2+a^2)^2}\frac{1}{p_0} \tag{C.29}$$

$$\boxed{\tilde{p} = \frac{I_0^2}{\mu_0 8\pi^2}\frac{a^2}{(r^2+a^2)^2}\frac{1}{\rho_0 v_A^2}} \tag{C.30}$$

VITA

Weston Lowrie grew up in Andover, Massachusetts. He earned his B.S. in Mechanical Engineering from Union College in Schenectady, NY in 2003. During his undergraduate education he interned at companies involved with gas turbines and propulsion systems for nuclear submarines. In 2005 he moved to Seattle to attend graduate school and recieved his M.S. and Ph.D from the Aeronautics and Astronautics Department at the University of Washington. Along with his studies he was able to take advantage of the many activities in the Pacific Northwest including backcountry skiing and ski mountaineering, nordic skiing, road and mountain biking, backpacking, and much more.