12-2003

Computational Magnetohydrodynamic Investigation of Flux Compression and Implosion Dynamics in a Z-pinch Plasma with an Azimuthally Opposed Magnetic Field Configuration

Kyle John Peterson
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Kyle John Peterson entitled "Computational Magnetohydrodynamic Investigation of Flux Compression and Implosion Dynamics in a Z-pinch Plasma with an Azimuthally Opposed Magnetic Field Configuration." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Physics.

Dr. Lloyd M. Davis, Major Professor

We have read this dissertation and recommend its acceptance:

Dr. Dennis Keefer, Dr. Jim Lewis, Dr. Christian Parigger, Dr. Ying-Ling Chen

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Kyle John Peterson entitled *Computational Magnetohydrodynamic Investigation of Flux Compression and Implosion Dynamics in a Z-pinch Plasma with an Azimuthally Opposed Magnetic Field Configuration*. I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Physics.

Dr. Lloyd M. Davis

Dr. Lloyd M. Davis, Major Professor

We have read this thesis
and recommend its acceptance:

Dr. Dennis Keefer

Dr. Jim Lewis

Dr. Christian Parigger

Dr. Ying-Ling Chen

Accepted for the Council:

Anne Mayhew

Vice Provost
and Dean of Graduate Studies

(Original signatures are on file with official student records.)

Computational Magnetohydrodynamic Investigation of Flux Compression and Implosion Dynamics in a Z-pinch Plasma with an Azimuthally Opposed Magnetic Field Configuration

A Dissertation

Presented for the

Doctorate of Philosophy Degree

The University of Tennessee, Knoxville

Kyle John Peterson

December 2003

Dedication

Dr. Larry W. Martin

March 29, 1956 - Oct. 25, 2002

http://www.admin.northpark.edu/lmartin/

This dissertation is dedicated to my undergraduate physics professor and good friend Larry Wendall Martin. Larry was diagnosed with a brain tumor in February of 2002 and lost his battle with cancer on Friday, the 25th of October, 2002.

Acknowledgments

First of all I would like to thank Dr. Dennis Keefer. I am forever in his debt for his guidance, support, technical expertise, and the time he spent to help me with this research. I would also like to thank Dr. Lloyd Davis for serving as my academic advisor and his support throughout this work. I enjoyed several MHD and plasma physics discussions with Dr. Chris Parigger who also served on my committee. I am also grateful for Dr. Jim Lewis and Dr. Ying-Ling Chen for serving on my committee.

I am extremely grateful for the financial support provided by the Sverdrup Technology corporation and for allowing me to pursue this research. Sverdrup also provided support for me to attend several scientific conferences which were extremely helpful in this research. The Decade team was a pleasure to work with. I received much needed help from several Decade team members including Van Kenyon, Amy Hopf, Tim Cotter, Bob Truesdale, Jimmy Payne, Bart Stewart, and especially Michael W. Scott. Mark Babineau and Randy Chapman, now at K-tech, also deserve thanks for their support and encouragement while they were with the Decade team. None of this work would have been completed if it wasn't for John Goyer of Titan. John was extremely helpful and supportive of this work. I need to thank John especially for his 1D Microcap flux compression model.

Numerous other people have also provided help and suggestions. I would like to thank Nino Pereira for wonderful discussion on Z-pinch physics and flux compression. Michael Frese of Numerex has provided me with code updates, suggestions, and other tools for my research. I received many useful comments and wonderful suggestions from several people in the Z-pinch and computational theory group at Sandia National Laboratories. I wish to thank Mathias Bavay for answering several of my questions and for describing his flux compression research. I learned a lot about computational simulations from some wonderful conversations with Bob

Rhodes. Bob helped me solve several nuances in the MACH2 code and other issues that came up along the way.

The most significant support and encouragement, however, came from my wife. I couldn't have finished this work without the continuous love and endless support. I love you! Last but definitely not least, I give thanks and praise to God for giving me the will, perseverance, and ability to complete this research.

Abstract

Magnetic flux compression is a well established technique for the generation of ultrahigh magnetic fields, large currents, and large energy densities. It has been suggested as a means for power density amplification on Z-pinch generators such as Decade Quad, at Arnold Engineering Development Center, and it may be especially suitable as a means for producing higher powers of K-shell radiation from high atomic number loads such as titanium. Although many one-dimensional models of flux compression on Z-pinch generators exist, an improvement in understanding is needed about the physics and implosion dynamics on a two-dimensional level. To this end, a two-dimensional resistive magnetohydrodynamic code was used to study a particular flux compression concept for use on Decade Quad. In the concept under study, compression occurs for self generated opposing azimuthal magnetic fields. In order to provide appropriate boundary conditions for the simulations, a non-linear circuit model was developed to enable calculation of the dynamically changing inductive and resistive impedances of the two coupled current paths such that they are consistent with the developing plasma. Good flux compression is observed despite magnetic flux losses. Two dimensional calculations are shown to match reasonably well with one-dimensional results. However, results also indicate Rayleigh-Taylor instabilities significantly affect implosion dynamics through the creation of isolated magnetic flux pockets, formation of circular currents, and the redistribution of current flow. It is also found that the Aluminum plasma armature shorts out on the stator, and thereby causes a nonideal current distribution in the titanium plasma. Consequently, the titanium plasma does not receive sufficient energy transfer for efficient K-shell radiative emission.

Contents

List of Tables

List of Figures

Nomenclature

t_c	Collisional time scale
χ_p	Planck opacity
χ_r	Rosseland opacity
Δr	Plasma shell thickness just before stagnation
ℓ_c	Collisional mean free path
ϵ_o	Permittivity of free space
η	Charge density
η_d	Electric diffusivity
γ	Instability growth rate
$\hat{t}$	Estimated implosion time
κ	Thermal conductivity
Ł	Characteristic system length scale
$\bar{\mathbf{S}}$	Hydrodynamic stress tensor
τ	Elastic stress deviator
$\mathbf{B}$	Magnetic Induction
$\mathbf{D}$	Electric displacement
$\mathbf{F_e}$	External forces acting on plasma
$\mathbf{H}$	Magnetic field intensity
$\mathbf{J}$	Current density
$\mathbf{K}$	Surface current
$\mathbf{M}$	Magnetization vector
$\mathbf{P}$	Polarization vector
$\mathbf{Q}$	Heat flux vector
$\mathbf{r}$	Position vector
$\mathbf{v}$	Velocity of fluid
μ	Mass per unit length
μ_o	Permeability of free space
Φ	Magnetic Fux
Π_{e-}	Plasma frequency
ρ	Mass density
σ_c	Cross section
σ_{dc}	DC electrical conductivity
σ_s	Surface charge density
τ	Time to peak drive current
τ_{diff}	Magnetic diffusion time scale
τ_{ei}	Electron-ion equilibration time
ε	Internal energy
Ξ	Dimensionless scaling factor in the thin shell model
a	Stephen's constant
$a(t)$	Bulk plasma acceleration
c	Speed of light
e	Unit of electric charge
E_{min}	Minimum energy to ionize into the K-shell

f	Maxwellian distribution function
k	Boltzmann constant
K_{ei}	Electron-ion coupling coefficient
k_m	Wavelength mode
l_d	Debye length
m	Mass
m_i	Ion mass
N	Number of particles
n_{crit}	Critical number of wires required for adequate shell merger at a particular radius
N_o	Number density
p_e	Plasma pressure tensor
r	Plasma shell radius
R_M	Magnetic Reynolds number
r_o	Initial plasma shell radius
T	Mean temperature
t_d	Characteristic time scale
t_p	Total implosion time
V	Characteristic velocity of a fluid element
v_e	Expansion velocity
v_{min}	Minimum implosion velocity for efficient K-shell production
v_p	Peak bulk radial velocity
v_{th}	Thermal velocity
W	Peculiar velocity
1D	One dimensional
2D	Two dimensional
3D	Three dimensional
AEDC	Arnold Engineering Development Center
AFRL	Air Force Research Laboratories
ALE	Arbitrary Lagrangian Eulerian
AN	Atomic Number
C	Capacitance
Decade Quad	DQ
DTRA	Defense Threat Reduction Agency
euv	Extreme Ultraviolet
eV	Electron Volts
I	Circuit Current
ICE	Isentropic compression experiments
ICF	Inertial Confinement Fusion
L	Inductance
L	Self Inductance
LTE	Local Thermodynamic Equilibrium
MA	Megaamperes
MHD	magnetohydrodynamics
MJ	Megajoules
n	Number of wires used in the array
NIF	National Ignition Facility
NRL	Navy Research Laboratories
PRS	Plasma Radiation Source
R	Resistor
RLC	Resistor, Inductor, Capacitor
RT	Rayleigh-Taylor

SNL	Sandia National Laboratories
Ti	Titanium
W	Magnetic energy

Chapter 1

Introduction

1.1 Description of Z-pinches

A Z-pinch is a plasma column that is self-constricted by a large current flowing in the axial or "z" direction. Z-pinch loads consist of a variety of configurations including thin wires configured in cylindrical arrays, supersonic gas expansion through a cylindrical set of nozzles commonly referred to as a gas puff, and thin metallic foils. Each of these load configurations serve to form the initial plasma column as the longitudinal current quickly vaporizes the material and produces an azimuthal magnetic field. Depicted in Figure 1.1, the Lorentz or $\mathbf{J} \times \mathbf{B}$ force drives an implosion of the plasma column onto the z axis as a result of the axial current flow and the induced magnetic field. As the plasma column stagnates on axis, large amounts of radiation, primarily in the soft x-ray and extreme ultraviolet regions (xuv), are produced.

Z-pinches have several applications. The primary application is to function as a plasma radiation source (PRS), a relatively inexpensive intense x-ray generator. Plasma radiation sources are used by the scientific community for radiation science[1] and the study of material properties at extreme conditions of pressure and temperature that could only be previously studied using underground nuclear detonations[2]. Another use for plasma radiation sources is the study of nuclear weapons effects on hardware components and electronics; this is the primary mission of the Decade facility[1] at Arnold Engineering Development Center (AEDC)[3, 4]. In the 1950s, Z-pinches were developed as a potential source of thermonuclear fusion that would heat a deuterium-tritium mixture and confine it until sufficient fusion energy was released [5, 6]. When it was discovered that observed neutrons were not of thermal origin and, rather, were the result of accelerated deuterons produced from plasma instabilities, this research began to fall out of favor in the fusion community[5]. Recently, significant progress in pulsed power technology and a better understanding of plasma instabilities in Z-pinches have renewed interest in Z-pinch inertial confinement fusion (ICF) concepts. Sandia National Laboratories (SNL) is investigating

[1] A Defense Threat Reduction Agency (DTRA) facility

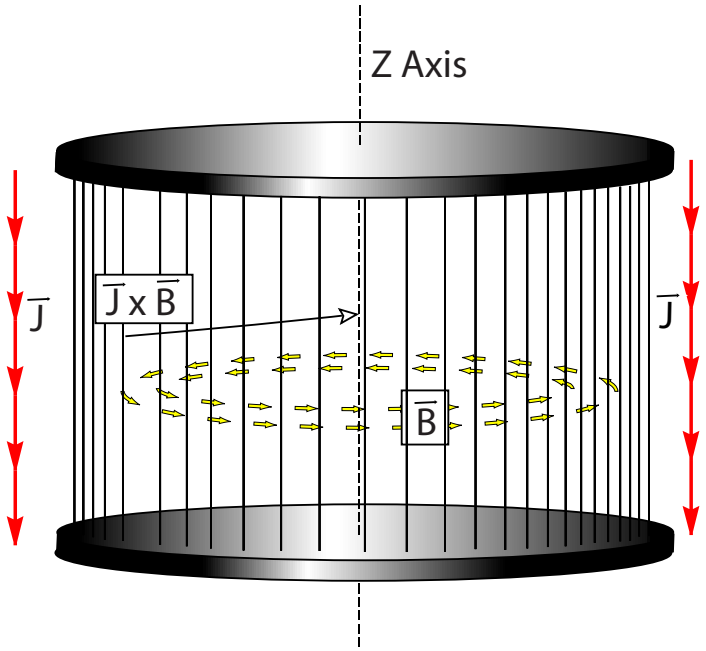

Figure 1.1: Diagram of a simple wire array Z-pinch

ICF concepts using Z-pinches, such as, dynamic[7] and double hohlraum[8] configurations, and fast ignition[9] techniques. As pulsed power technology continues to advance, ICF driven with Z-pinches becomes an attractive alternative to similar laser-driven concepts, such as the National Ignition Facility (NIF), due to the comparatively low cost of the drive energy required. Other applications of Z-pinches include x-ray lithography and microscopy sources, x-ray laser pumping sources, neutron sources, and as magnetic lenses for focusing high-energy particles[10].

1.2 Plasma Radiation Sources

The Z-pinch has been extensively used as an efficient plasma radiation source (PRS). The bulk of the radiation, which lies in the soft x-ray and xuv regions 10-3000 eV, is produced at stagnation[2] of the pinch. Bremsstrahlung radiation, recombination radiation, and line emission radiation all contribute to the observed radiation spectra. The radiation spectra, to some degree, can be tailored by a particular load configuration. Staggering levels of radiated power have been produced on the 20 megaampere (MA) Z accelerator at SNL. Almost 2 megajoule (MJ) in x-ray energy and 340 terrawatts of x-ray power have been achieved experimentally by utilizing nested tungsten wire array loads[11].

[2]Defined as the point at which the bulk plasma reaches the z-axis of symmetry and stagnates.

For some applications, including radiation effects testing at the Decade facility, there is a need to significantly increase the radiation yield from photon energies above 4 keV. Specifically, there is interest in producing K-shell line emissions, that are produced when an electron fills a vacancy in the K-shell of an atom, from high atomic number elements. Large quantities of K-shell radiation are typically produced by large collections of atoms that have been stripped of all but one or two of its electrons. These atoms will then emit hydrogen-like and helium-like spectra. Unfortunately, generating K-shell radiation for elements of increasing atomic number (AN) becomes inefficient due to the additional energy needed to sufficiently ionize the heavier atoms. Decade is a much smaller machine than the Z accelerator at SNL and is capable of producing wire load currents up to 8 MA. Such a current is insufficient to produce sizable K-shell emission from wire loads composed with elements of AN's higher than twenty. With an AN of 22, Titanium (Ti) K-shell emission has a photon energy of approximately 4.8 keV and is of particular interest to the weapons effects community. Thus far, only the Z accelerator at SNL has produced significant yields (∼120 kJ) of Ti K-shell radiation[12].

K-shell radiation production in Z-pinches scale poorly to higher AN for several reasons. First, the energy needed to strip atoms of all but one or two electrons does not vary linearly with higher atomic number elements. Secondly, above an atomic number of twenty, L-shell radiative losses become significant[13]. Lastly, to achieve the necessary kinetic energy in the load, mass must either be reduced, or higher velocities must be obtained with a larger initial radius. Lower mass loads limit radiation flux, while greater radii permit greater plasma instability development and less uniform implosions. A non-uniform implosion is highly detrimental to K-shell radiation flux.

An approximate scaling law has been developed for the minimum kinetic energy required to ionize into the K-shell[13]. The minimum energy, E_{min}, is

$$E_{min} \cong 1.012(\mathrm{AN})^{3.662}\mathrm{eV/ion}, \tag{1.1}$$

where AN is the atomic number of the element. For Ti, the minimum energy for ionization into the K-shell is 83.4 keV/ion. If we assume complete ion kinetic energy transfer into thermal heating upon stagnation, a minimum implosion velocity can be determined for a Z-pinch. The minimum implosion velocity, v_{min}, is given by

$$v_{min} = \left(\frac{2E_{min}}{m_i}\right)^{1/2} \tag{1.2}$$

where m_i is the ion mass. Thus for Ti, the minimum implosion velocity to ionize into the K-shell is approximately 60 cm/μs. The minimum implosion velocity is a primary factor that is considered when designing or choosing a load configuration to maximize K-shell radiation production.

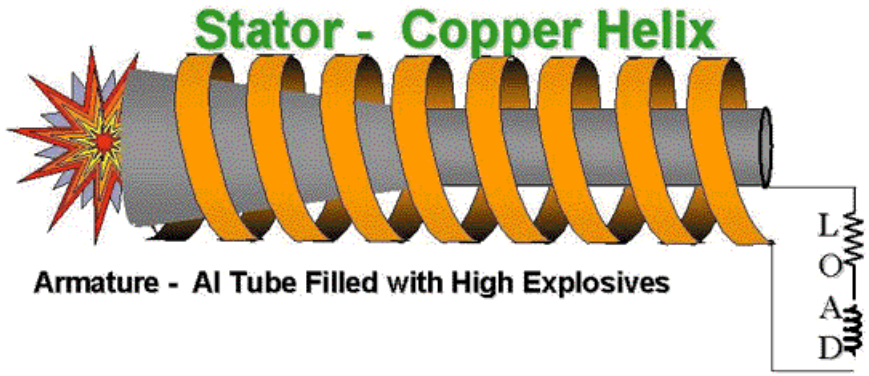

Figure 1.2: Explosive helical flux compression generator. Graphic used with permission from Los Alamos National Laboratory website.

1.3 Flux Compression

Flux compression techniques have been used in pulsed power applications for quite some time as a means to produce large amounts of current (10-100 MA) and magnetic field strengths up to 2800 Tesla[14, 15, 16, 17, 18]. The fundamental concept of flux compression is to quickly reduce the inductance of a circuit while conserving magnetic flux. The magnetic flux through an electrical circuit, Φ, is defined to be

$$\Phi = LI, \tag{1.3}$$

where L is the self inductance of the circuit, which is purely a function of geometry, and I is the current flowing through the circuit.

If the inductance of the circuit is reduced and the magnetic flux conserved, the current must rise proportionally. Chemical energy in the form of explosives is the most commonly used method of flux compression in pulsed power applications. A helical flux compression generator is one such example of this method[19]. As shown in Figure 1.2, the flux compression armature is explosively driven on the helical stator. Consequently, the inductance of the generator connected to the load rapidly decreases producing a corresponding rise in current. Since magnetic energy ($\frac{1}{2}LI^2$) scales as the square of the current, the advantage of magnetic flux compression is a net gain in magnetic energy.

Explosive flux compression is not an option for a Z-pinch PRS as the generator is destroyed in every experiment. Instead, a secondary current source or path is

used to collapse a plasma armature, which compresses a trapped region of magnetic flux. The compressed current pulse allows greater power levels and higher energy densities to be delivered to the load[20]. Since efficient K-shell radiation production scales poorly for higher AN materials, the attractiveness of flux compression is the possibility of producing significant powers of heavy atom K-shell radiation on smaller generators such as Decade Quad (DQ)[3].

1.4 Dissertation Overview and Goals

The primary goal of this dissertation is to investigate the physics of flux compression techniques used in Z-pinches and to evaluate the possibility of increase of radiation yields in these processes. To this end, flux compression experiments are modeled with the 2D Arbitrary Lagrangian Eulerian (ALE) resistive magnetohydrodynamics (MHD) code named MACH2[21]. The MACH2 code is a powerful general purpose MHD code distributed by the Air Force Research Laboratories (AFRL).

In chapter 2, the physics of Z-pinches and flux compression is introduced to give the non-specialist reader adequate background knowledge. The third chapter of the dissertation presents a short introduction to computational MHD and how it is used in this research. A brief description of the MACH2 code, its capabilities, and steps involved in modeling simple Z-pinch experiments, is also discussed in this chapter. Chapter 4 describes, in detail, how flux compression on the DQ is modeled. It also describes the major additions and changes made to the MACH2 code for this research. The fifth chapter presents and discusses flux compression simulation results. Finally, conclusions and suggestions for future work are presented in the sixth and final chapter.

[3]The original concept of Decade was to consist of four separate generators. Decade Quad refers to the only operational generator at this time. Another generator is under construction that is intended to be coupled to Decade Quad. This coupled configuration will be referred to as Decade Half. No plans currently exist to construct the other two generators in the original concept.

Chapter 2

Physics of Z-pinches

2.1 Z-pinch Fundamentals

There are two classes of Z-pinches. Bennett-type equilibrium Z-pinches occur when magnetic forces approximately balance plasma pressures. The pinch is quasi-static and not characterized by high plasma velocities or fast implosions on axis. Dynamic Z-pinches, also referred to as "fast" Z-pinches, are driven by current pulses lasting on the order of the implosion time; a Bennett-type equilibrium cannot exist as the magnetic pressure is much greater than the plasma pressure for the majority of the current pulse[6]. The physics of these two Z-pinches are significantly different, especially when considering plasma instabilities. Dynamic Z-pinches have recently been gaining the attention of the Z-pinch community due to recent advances and breakthroughs in pulsed-power technology[22, 23]. Flux compression concepts for Z-pinches require fast symmetrical implosions to maximize radiation efficiency. For this reason, equilibrium Z-pinches will not be discussed for the remainder of this work and the term "dynamic" will be dropped and assumed to be understood from this point on.

Although the basic principles of Z-pinches are relatively simple, the underlying physics in Z-pinches are extremely complex. The physical processes involved in Z-pinches are still not well understood on many levels and continue to be the source of many scientific disputes[6, 10]. Some of these issues will be discussed in this work. Unfortunately, Z-pinch diagnostics are extremely difficult due to the harsh and violent conditions existing in the implosions. The experimental data thus far has generally limited the physical understanding of Z-pinches to a qualitative and semispeculative level[6]. Consequently, Z-pinches continue to be a very active area of research.

There are several different load configurations for Z-pinches depending on the application. The most common load configurations are thin foil annular shells, supersonic gas jets that create either annular shells or uniformly fill the pinch region, foam cylinders, and cylindrical arrays of thin wires. Supersonic gas jets, or gas puffs as they are commonly called, are generally pre-ionized in order to provide a more uniform current distribution than would occur if the gas were allowed to

breakdown naturally. For this same reason, foam targets are sometimes coated with a conducting material. Cylindrical arrays of thin wires are generally preferred over thin foils since the fabrication and handling of foils on the order of a few hundred nanometers thick without introducing large azimuthal or axial perturbations is very difficult[10]. Wire arrays form more uniform plasma shells, which leads to more efficient radiation production. The particular load material is usually chosen on the basis of desired radiation power or spectrum.

Z-pinch generators exist in a wide variety of sizes and parameters. Table 2.1 shows several key electrical parameters for several different Z-pinch machines using typical wire loads. The Z accelerator at Sandia National Laboratories is the world's most powerful x-ray generator. The efficiency of converting electrical energy into x-ray radiation, on this generator, approaches 20 percent[12]. Decade Quad, shown in Figure 2.1, is a much smaller generator, but still capable of producing hundreds of kilojoules of x-ray energy[3]. DQ is also unique in other respects. Loads are mounted horizontally rather than the typical vertical configuration used on most generators. The front end of the machine is also modular and can be coupled to an array of plasma opening switches and Bremsstrahlung diodes. This configuration, shown in Figure 2.2, allows DQ to produce 20 krads of hard x-rays over areas as large as 2000 cm^2. Small generators, such as MAGPIE at Imperial College in England, are used primarily for basic Z-pinch research, as they can be operated relatively cheaply and require much less maintenance between Z-pinch experiments.

Table 2.1: **Various electrical characteristics of several Z-pinch generators**

	Z [a]	DQ (PRS)[b]	Saturn [c]	MAGPIE [d]	ZR [e]
Energy Store	11.5 MJ	2.0 MJ	5.0 MJ	350 kJ	23 MJ
Peak Load Voltage	5 MV	∼1 MV	3.0 MV	2.4 MV	7 MV
Peak Power	60 TW	∼6-7 TW	5 TW	1 TW	∼180 TW
Energy to Load	3 MJ	∼1.0 MJ	1.2 MJ	240 kJ	6 MJ
Pulse Duration	100 ns	300 ns	200-300 ns	240 ns	100 ns
Peak Current	20 MA	8 MA	11 MA	1 MA	26 MA

[a] Z is the Z accelerator at SNL.
[b] Decade Quad (DQ) is located at AEDC, configured in the PRS mode.
[c] Saturn is a smaller generator also located at SNL. Data is for the Saturn long pulse mode configuration.
[d] MAGPIE is a very small machine used for the study of the physics of Z-pinches at Imperial College in England.
[e] ZR is the refurbishment of the Z accelerator not expected for completion until the summer of 2004.

Figure 2.1: Decade Quad configured in the plasma radiation source mode. Graphic used with permission from the Arnold Engineering Development Center website.

Figure 2.2: Decade Quad configured for the large area Bremsstrahlung source mode. Graphic used with permission from the Arnold Engineering Development Center website.

2.2 Thin Shell Model

The simplest Z-pinch model starts with the assumption of an infinitely thin plasma shell. The shell's dynamics are governed by an azimuthal magnetic field created by an axial drive current flowing through the shell. Thus, the radial position of the plasma column can be approximated by the mass per unit length and the strength of the $\mathbf{J} \times \mathbf{B}$ or Lorentz force. While a crude approximation, this zero dimensional approximation yields favorable estimates for implosion times.

The equation of motion for an infinitely thin plasma shell, with the assumption that magnetic pressure is the only force acting on the plasma, is given by

$$\frac{\mu}{2\pi r}\frac{d^2r}{dt^2} = -\frac{B^2}{2\mu_o} = -\frac{\mu_o I^2(t)}{8\pi^2 r^2}, \tag{2.1}$$

where I is the drive current, μ_o is the permeability of free space, r is the radius of the plasma shell and μ is the mass per unit length of the plasma shell. The equation is typically converted into a dimensionless form by a transformation into the following variables[6],

$$\hat{r} = r/r_o, \tag{2.2}$$

$$\hat{\tau} = t/\tau, \tag{2.3}$$

$$\hat{I} = I/I_{max}, \tag{2.4}$$

where r_o is the initial plasma shell radius, τ is the time required for the current to reach a maximum, and I_{max} is the maximum current. Equation 2.1 then becomes

$$\hat{r}\frac{d^2\hat{r}}{d\hat{\tau}^2} = \Xi\hat{I}^2(\hat{\tau}), \tag{2.5}$$

where Ξ is a dimensionless scaling factor given by

$$\Xi = -\frac{\mu_o I_{max}^2 \tau^2}{4\pi\mu r_o^2}. \tag{2.6}$$

For a particular current profile, $\hat{I}^2(\hat{\tau})$, an optimum value exists for the scaling factor. Ideally, the imploding mass of the plasma shell should arrive at the axis at the time of peak drive current. Too much initial mass will cause the current to peak before the shell reaches the axis and too little mass will cause the shell to stagnate on axis before the peak drive current has been reached. Thus for maximum kinetic energy transfer, the shell mass must be optimized for a particular drive current to ensure maximum x-ray production efficiency of the Z-pinch.

In the case of a pulse of constant current, equation 2.5 can be solved exactly by defining a dimensionless velocity, $\hat{v} = \frac{d\hat{r}}{d\hat{\tau}}$, and integrating in the following manner:

$$\hat{r}\frac{d\hat{v}}{d\tau} = \Xi, \tag{2.7}$$

$$\hat{r}\hat{v}\frac{d\hat{r}}{d\hat{r}} = \Xi, \tag{2.8}$$

$$\int_0^{\hat{v}} \hat{v}d\hat{v} = \int_{\hat{r}}^1 \Xi\frac{d\hat{r}}{\hat{r}}, \tag{2.9}$$

$$\left(\frac{d\hat{r}}{d\tau}\right)^2 = 2\Xi \ln\left(\frac{1}{\hat{r}}\right). \tag{2.10}$$

Another integration yields an estimate of the time required for the shell to compress to a minimum radius. This is called the implosion time, $\hat{t}_{\text{imp}}$, which is given by

$$\int_0^{\hat{t}_{\text{imp}}} d\tau = \int_{\hat{r}_{\text{f}}}^1 \frac{d\hat{r}}{\sqrt{2\Xi \ln\left(\frac{1}{\hat{r}}\right)}}, \tag{2.11}$$

$$\hat{t}_{\text{imp}} = -\frac{\pi}{2\sqrt{\Xi}}\textbf{Erf}\left[\sqrt{\ln\left(\frac{1}{\hat{r}_{\text{f}}}\right)}\right], \tag{2.12}$$

where $\hat{r}_{\text{f}}$ is the minimum implosion radius. Typically, $\hat{r}_{\text{f}}$ is defined in terms of the convergence ratio, C_r, given by

$$C_r = r_o/\hat{r}_{\text{f}}. \tag{2.13}$$

A high quality Z-pinch implosion has a convergence ratio of 10 to 15.

2.3 Plasma Instabilities

Magnetic Rayleigh-Taylor (RT) instabilities are generally believed to be the dominant factor in limiting radiation yield in Z-pinches[6, 10, 24, 25, 26]. This type of plasma instability is akin to hydrodynamic RT instabilities, which are formed at the interface of two incompressible fluids, where a less dense fluid accelerates a fluid of higher density[27]. In the case of Z-pinches, the magnetic field represents the less dense fluid and the plasma shell is the higher density fluid.

RT instabilities correspond to the MHD instability modes m=0 and m=1, shown in Figure 2.3. The m=0, or sausage mode, occurs if an axisymmetric perturbation exists; a bulge in the plasma column will then occur. As the bulge grows, a constricted region of the plasma column also forms. One interesting phenomena of the m=0 mode is the possibility of neutron production that is not of thermonuclear origin[28]. Current disruptions created by sausage mode instabilities produce large voltage potentials, which generate accelerated deuterons and creates neutron radiation bursts[6]. The m=1, or kink mode, occurs when a perturbation causes a slight bend in the plasma column. The magnetic pressure will increase on the inside bends of the plasma column, where the magnetic fields lines are closer together. Likewise, the magnetic pressure will decrease on the outside of the bends.

RT instabilities limit the radiation yield by the broadening of the radial mass density profile on axis, which limits ion compression and lengthens the radiation

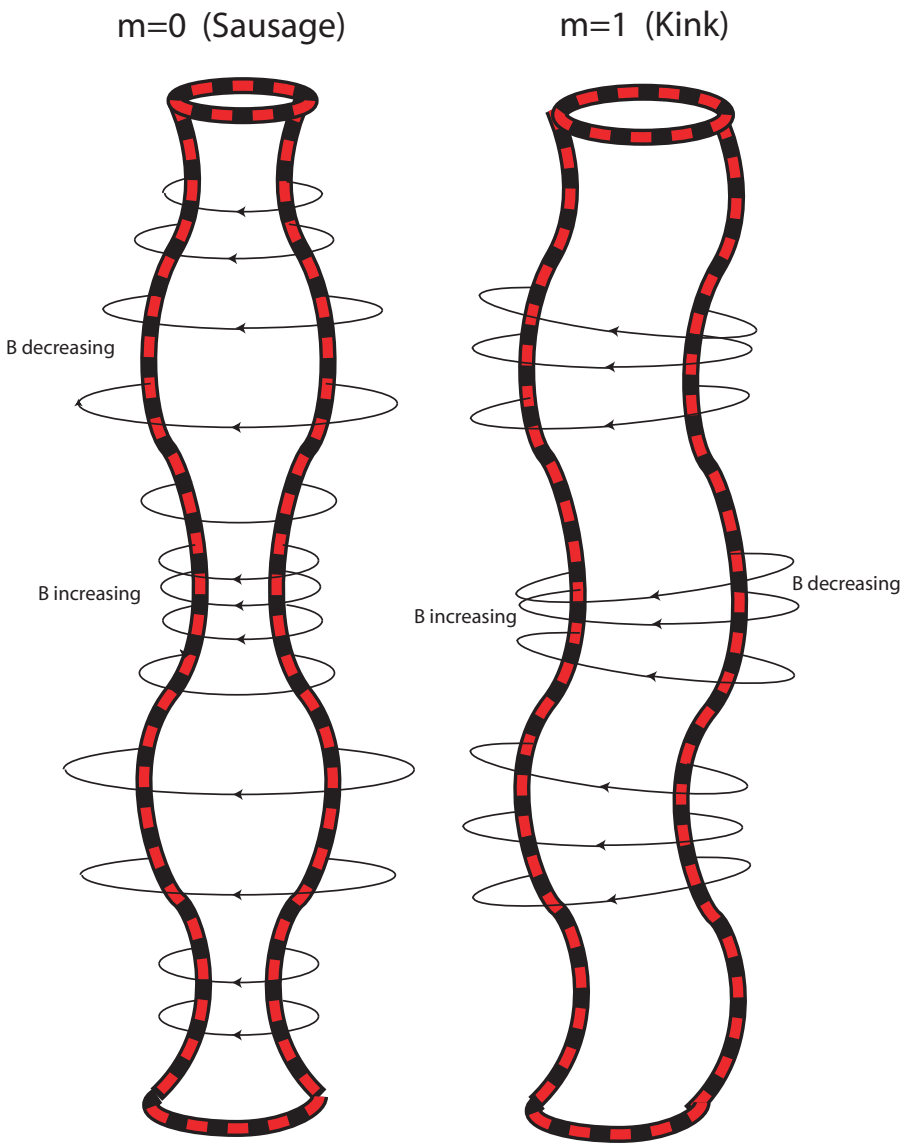

Figure 2.3: Sausage (m=0) and Kink (m=1) instability modes.

pulse. The redistribution of mass also creates inhomogeneous radiation emission along the axis. Numerical simulations show RT instabilities form magnetic bubbles, which are regions of low density containing a pocket of magnetic field penetrating into the plasma. As a result, inductance in the circuit path is increased and the implosion velocity of the bulk plasma is reduced[29]. The m=0 sausage mode instability can also form transient hot-spots[1] defined as localized regions of high mass density and emissivity[10, 30]. The hot-spots radiate energy away from the plasma column and reduce the amount of kinetic energy available to the bulk plasma.

Z-pinch RT mode development is complex due to the large number of possible modes. Several MHD phenomena, such as magnetic diffusion, compressibility, and current distribution, which are not a part of classical RT instability theory, must also be taken into consideration. Computational MHD simulations have been used to investigate single mode and multimode RT instability development in Z-pinch plasmas[29, 31, 32, 33] and their effects on radiation production [34, 35, 36, 37, 38]. In general, RT instability wavelengths are found to have a Gaussian spectral profile during the implosion. As the implosion evolves, the dominate RT instability growth modes transition to longer wavelengths[39] due to mode saturation and coupling effects[31], and exhibit rapid nonlinear growth.

2.4 Heuristic Model

Instabilities aside, the dynamics of wire array Z-pinches are much more complicated than are represented in a thin shell model. The Heuristic model developed by M. G. Haines[40] divides the physics of wire array Z-pinches into four distinct phases. The first stage is referred to as the individual wire and plasma formation phase. A large current is introduced into a circular array of conducting wires, which creates a rapid and uniform constant velocity plasma expansion of the individual wires[41]. At the same time, the current flowing through the wires produces an azimuthal magnetic field. During this phase, the dynamics of the individual wires are considered to be independent. The magnetic field is assumed at this point to be localized and global magnetic field effects are assumed to be negligible. The physics of the individual wires are very similar to single wire current driven explosions until the expansion of the wire plasmas begin to merge into a cylindrical shell.

One interesting aspect of the Heuristic model is the prediction of a critical number of wires at a particular radius in order to ensure adequate merging of plasma into an annular shell. The critical wire number, n_{crit}, is defined to be

$$n_{crit} = \frac{\pi r_o}{v_e t_p}, \tag{2.14}$$

where r_o is the initial wire array radius, v_e is the constant expansion velocity, and t_p is the implosion time. If less than the critical number of wires is used, the inter-wire gaps are too large and a uniform plasma shell will not be formed. Instead, the

[1] Hot-spots are also referred to as bright spots or micropinches by various authors.

individual wire plasmas will "jet" towards the z-axis as a result of the global $\mathbf{J} \times \mathbf{B}$ force, and will produce a highly nonuniform implosion[26]. Experimentally, the effect of a critical wire number has been shown empirically from observed radiation powers[24].

The second phase of the Heuristic model begins when the individual wire expansion is complete and the individual wires have merged into a cylindrical shell. The plasma shell thickness is assumed to be equal to the individual wire expansion diameter. At this point, the global magnetic field must be considered as its effect is much larger than the localized magnetic field around each wire. Several assumptions are then made regarding plasma instabilities. Small density variations in individual wires are assumed to feed the dominant m=0 sausage MHD instability mode in the shell. This instability acts as the seed to Rayleigh-Taylor (RT) instability development in phase three. According to this model, the density variations at the formation of the shell can be reduced as the number of wires in the array are increased. The amount of density variation in the plasma shell is found to scale with wire number as $1/\sqrt{n}$.

As a consequence of the magnetic field and the plasma current density, the $\mathbf{J} \times \mathbf{B}$ force begins to collapse the plasma onto the z-axis. This begins the third phase, also called the run-in phase, where RT instabilities initially exhibit linear growth and, eventually, strong nonlinear growth as the plasma accelerates to the axis. The dominant modes of RT development include the m=0 sausage mode and the m=1 kink mode shown in Figure 2.3.

The fourth phase is plasma stagnation, when the plasma shell approaches the axis of symmetry. During this phase, the plasma kinetic energy is converted to thermal energy and a portion of that energy is radiated away in the form of x-rays. In this model, the main x-ray radiation pulse is primarily a function of plasma density and pinch radius at stagnation. Consequently, the amplitude of RT instabilities significantly affects x-ray power by limiting the minimum pinch radius.

2.5 Wire Initiation

The wire initiation phase of Z-pinches is the subject of intense experimental and theoretical study[6]. An enormous amount of physics is involved in current driven explosions of wires. Equation of state becomes extremely important as the wires undergo phase changes from solid metal to a fully ionized plasma.

Experimental work in pulsed currents through single copper wires by Aranchuk *et al.*[42] showed that shortly after the current pulse begins the majority of current flows through a plasma corona consisting of only 2-7% of the the total wire mass. The 20 μm core of the copper wires remains relatively cool and expands at a relatively constant rate. There is little joule heating in the core, rather the core is heated by thermal diffusion. Other experiments confirm these results in other materials[41, 43, 44, 45, 46, 47, 48]. This suggests that the thin shell model of imploding Z-pinches may not represent the physics as appropriately as an ablative wire model.

Further complicating matters, the behavior of a plasma corona around the wires in Z-pinches is vastly different from a uniform explosion of the individual wires. The plasma corona will be accelerated to the z-axis by the $\mathbf{J} \times \mathbf{B}$ force. This creates a plasma precursor that forms a stable symmetrical cylinder of radiating plasma around the z-axis[49]. The issues of wire numbers in an array and intergap spacing in the array, as discussed in the Heuristic model, significantly affect the physics involved.

Computational models of exploding wires have proven extremely difficult, especially in Z-pinch configurations, due to the complexities of the equation of state. However, J. P. Chittenden *et al.*[44, 50, 51] and T. W. L. Sanford *et al.*[37] have developed "cold start" models of single wires in Z-pinches that predict results in reasonable agreement with experiment[52]. In these models, plasma instabilities are formed from small variations in plasma density along the wires, which lead to a modulation of ablation rates along each wire. The wire array implosion does not begin until a portion of the wire has completely ablated away. The wire plasma starts to break up as the fully ablated portion of the wire begins to implode. Current now flows through the plasma prefill created by ablation of the wire material and drives a magnetic snowplow effect implosion. As a consequence of the breaks in the wire, a significant amount of wire mass gets left behind and does not contribute to the implosion. In fact, there is experimental evidence to conclude that as little as ten percent of the initial wire mass actually participates in the implosion[48]. The wire debris competes for current flow, which further limits maximum current driving the implosion. This model also suggests RT instabilities may be electrothermal rather than electromagnetic in origin[49].

While it is generally accepted that the wires ablate away and create a prefill plasma rather than quickly exploding, the plasma formation stage of Z-pinches is still up for debate from a computational standpoint. Michael H. Frese has investigated Z-pinch wire initiation stages using a Lagrangian MACH2 simulation[53, 54]. This study suggests that under MHD approximations, wire-arrays in Z-pinches do not ablate away and the overall dynamics are more consistent with the thin shell model.

Another item that must be considered is the scaling factor. Experiments showing plasma jetting and prolonged wire ablation have been conducted on relatively low powered machines such as MAGPIE at Imperial College. How do the physics scale to multi-megaampere machines? Is it possible that on large machines with enough power thin wires truly explode rather than ablate away? Obviously more experiments and better diagnostics on machines such as the DQ or the Z accelerator at SNL are needed to resolve these issues.

If it is assumed that the wire ablation model is correct, considerable insight can still be achieved using a thin shell approximation in flux compression simulations. Historically, thin shell approximations have led to the most significant technological advances in Z-pinch radiation yield, especially in nested wire arrays[6]. From a physical perspective, if wire separation is sufficiently small, it has been shown that plasma streamers carry only a negligible amount of current and field with them[55]. Thus, a thin shell model would still be appropriate with the primary difference being

a different distribution of mass and plasma precursor on axis. If flux compression simulations were to be based on the ablative wire model, a 3D model would be needed. This would further increase the complexities of the model.

2.6 Nested Wire Arrays

A significant breakthrough in the reduction of RT instabilities in Z-pinches was achieved with the introduction of the nested wire arrays in 1998 at SNL[11, 49, 56, 57, 58]. Using MHD simulations of the Z-pinch, Melissa Douglas *et al.*, predicted significant reduction of the RT instabilities in nested wire array configurations.

Originally, the mitigation of the RT instabilities was believed to be caused by collision dynamics between two thin annular plasma shells. RT instabilities that form in the outer shell are annilated upon collision with the inner array[11]. Some authors refer to this operating mode as the "hydrodynamic collision" mode.

Further investigation has led to the belief that two distinctly different operating modes exist with nested wire arrays Z-pinch plasmas[25, 58, 59]. The physical mechanisms involved in each mode are postulated to be different. However, each mode yields nearly the same experimental results in terms of radiation yield and pinch diameter. These modes are defined by the length of the inner array wires. As the lengths of the wires increase, the self inductances of the wires increase. Therefore, the two modes are defined as the low L and high L modes. In the high L mode, most of the current initially flows into the outer wire array, which acts as a fast inductive current switch. The inner array remains fixed until the outer wire array has passed through the inner wire array with little interaction. The current is then inductively switched into the inner array and it implodes onto the axis. It is postulated that the rise in x-ray power is a result of RT instabilities having less time to grow in the relatively short implosion time of the inner wire array. In the low L case, a greater fraction of the current is shared with the inner array throughout the implosion. The arrays do not collide, but undergo simultaneous stagnation on axis. In this case, the magnetic flux is trapped between the arrays and acts as a buffer zone. The mechanism for reducing RT growth in this mode is less clear. It is theorized that the trapped magnetic flux mitigates the RT growth rates[25].

2.7 Radiation Modeling

It was previously stated that radiation transport in Z-pinches is largely unknown. This is due in part to the fact that Z-pinch plasmas are generally not in local thermodynamic equilibrium (LTE) at stagnation nor are they entirely optically thin or thick. The MACH2 code provides several different radiation models to choose from, but they are all based on LTE assumptions.

The Z-pinch theory group at Navy Research Laboratories (NRL) takes a fundamental approach to radiation using population kinetics models. 1D Z-pinch results comparing several radiation models demonstrate significant amounts of error when

using simple radiation models based on local thermodynamic equilibrium (LTE) for Z-pinch calculations. Davis argues that using simple radiation models "result in a misinterpretation of experimental observations and a model incapable of addressing issues relevant to the tailoring of x-ray production and of evaluating load performance[60]." While a valid point, NRL's non-LTE models require an enormous amount of computation time compared to simple LTE models. In the future, as the speed of computers seems to exponentially increase, there is no doubt that non-LTE radiation models will be an extremely valuable research tool for Z-pinches.[2]

Due to the complexities in modeling radiation, radiation transport is conceded to be largely unknown at stagnation for this research. However, the assumption will be made that implosion velocity and ion temperatures on axis at the stagnation of the pinch have a direct correlation to K-shell radiation energy. The shell thickness or width of the plasma column just before stagnation is also used to estimate radiation power through the hydrodynamic thermalization time scale. From a physical point of view and available experimental data, this assumption is very reasonable. From an energy conservation standpoint, the flux compression simulations performed in this research will use LTE radiation calculations during the implosion.

[2]A non-LTE radiation model is in development for MACH2

Chapter 3

Computational Magnetohydrodynamics

A basic understanding of magnetohydrodynamics (MHD) and how it is used in computational simulations is important to understanding the physics and approximations used in the flux compression simulations presented in this research. This chapter discusses fundamentals of computational MHD and closes with a detailed description of the capabilities of the MHD code used in this research.

3.1 MHD Fundamental Approximations

The most basic assumption of the MHD model is that collective behavior of a plasma can be modeled as a quasi-neutral electromagnetic fluid. The fluid approximation significantly reduces the complexity of plasma physics equations. In general, a complete representation of the position and momentum of every particle in the system would require a 6N (N = number of particles) dimensional configuration space. Obviously, this space contains much more information than is needed to analyze the statistics of the collection of particles as a whole. In a kinetic theory approach, statistical mechanics is used to reduce the 6N dimensional configuration space to a 6 dimensional phase space. Rather than being concerned with the individual state of every particle in the system, the distribution function of velocities, which represents the density of points in phase space, are used to describe the dynamics of the system. A fluidic approach further simplifies the representation of the system dynamics by considering only a three dimensional spatial configuration space. State variables such as temperature, density, pressure, and mean velocity, rather than using distribution functions, are used to describe the macroscopic properties of the collection of particles as a whole. This approach is intuitively pleasing since the state variables are the quantities that are usually measured experimentally.

The degree to which plasma behavior can be described as a fluid, rather than as a set of particles can be judged from the collisional time scales. In plasmas, three types of collisional processes are of interest: collisions amongst neutral particles,

collisions amongst charged particles, and collisions between neutral particles and charged particles. Charged particle collisions are mediated by long range Coulomb interactions. Typically, the long range of the Coulomb field allows a charged particle to interact with several other charged particles simultaneously. In order to deal with this, the simultaneous interactions are assumed to be a series of consecutive binary collision processes[61]. Collisions that involve a neutral particle are simple binary collisions. The interaction is short range with with low probability of simultaneously interacting with multiple particles.

Binary collisions in plasmas are exemplified by two collisional time scales. The characteristic time scale, t_d, is defined to be

$$t_d = \text{Ł}/v_{th}, \tag{3.1}$$

where Ł is the system length scale and v_{th} is the mean thermal velocity of the plasma species. The collisional time scale, t_c, is defined to be

$$t_c = \ell_c/v_{th}, \tag{3.2}$$

where ℓ_c is the collisional mean free path, defined as

$$\ell_c = 1/(N\sigma_c), \tag{3.3}$$

where σ_c is the cross section for the interaction and N is the number density of particles in the interaction volume.

If $t_c << t_d$ or $\ell_c << \text{Ł}$, then the system of particles is said to be Maxwellian and it will behave collectively as a fluid. The mean time between collisions is sufficiently short that the particles are in equilibrium with a Maxwellian velocity distribution given by

$$p(W)dW = N_o \left(\frac{m}{2\pi kT}\right)^{3/2} e^{-\left(mW^2/2kT\right)}, \tag{3.4}$$

where T is the mean temperature of the species, m is the particle mass, N_o is the number density, k is the Boltzmann constant, and $\mathbf{W}$ is the peculiar velocity, which is defined as

$$\mathbf{W} = \mathbf{w} - \bar{\mathbf{v}}, \tag{3.5}$$

where $\mathbf{w}$ is the particle velocity and $\bar{\mathbf{v}}$ is the mean velocity of the collection of particles. Thermodynamic properties can then be defined for the set of particles and a fluidic approach can be used to analyze collective plasma behavior.

For computations, the collisional mean free path is compared with the characteristic length scale of a single grid cell to determine if the localized fluid approximation is valid. Macroscopic plasma behavior need not extend over the entire range of a computational domain to satisfy these requirements.

There is one exception to the collisionality requirement for MHD approximations to be valid. Interestingly, for collisionless plasmas, MHD approximations are valid for charged particle motions perpendicular, but not parallel to magnetic field lines,

because the resultant energy between the particles and the magnetic field enable the particles to equilibriate. Generally speaking, collisionless plasmas require a modified theory called kinetic-MHD[62] which is beyond the scope of this study.

Since Z-pinch plasmas are fully ionized, the most significant collision processes are Coulomb interactions between charged particles. Coulomb interactions also determine the quasi-neutrality of the plasma. Consider a test charge particle interacting with multiple charged particles simultaneously. Charged particles close to the test particle act as a screen between the test particle and other distant charged particles. The screening effect gives rise to a characteristic length in a plasma that defines the distance in which the test particle trajectory is unaffected by more distant charged particles. This characteristic length is known as the Debye length, l_d, given by

$$l_d = \left(\frac{\epsilon_o kT}{N_o e^2}\right)^{1/2}, \tag{3.6}$$

where ϵ_o is the permittivity of free space and e is the unit charge. Electrostatic potentials exist in plasma regions where the characteristic length scale is on the order of the Debye length or smaller. Particles in such regions must be analyzed individually. Plasma behavior on a length scale greater than the Debye length can be treated as collection of quasi-neutral particles.

Plasma quasi-neutrality can also be affected by the plasma frequency. Since plasmas have a strong tendency to remain electrically neutral, any electric field that arises in the plasma will be counteracted by the motions of electrons trying to return to an equilibrium quasi-neutral charge state. The electron's momentum will overshoot the equilibrium position and establish an electric field in the opposite direction. This motion sets up a characteristic electron oscillation about the equilibrium position with a characteristic frequency called the plasma frequency. Obviously, a characteristic ion frequency will exist as well, but due to the comparatively high mass of ions, the electrons are significantly more mobile. Hence, ion frequencies are generally ignored. The plasma frequency is given by

$$\Pi_{e-} = \sqrt{\left(\frac{N_o e^2}{m_e \epsilon_o}\right)}. \tag{3.7}$$

where m_e is the mass of an electron.

If characteristic time scales or computational time steps are greater than the reciprocal of the plasma frequency, the electric fields due to the oscillating electron motion will not be observed. Thus, the quasi-neutrality assumption will be preserved.

3.2 Fluid Equations

MHD is built on the foundation of fluid mechanics. The fluid equations can be derived on the basis of conservation of mass, momentum, and energy, or from a

kinetic theory approach by taking moments of the Boltzmann equation. These equations are commonly called the Navier-Stokes equations, or the Euler equations when the flow is inviscid. They are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \qquad \text{Continuity Equation} \tag{3.8}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = \mathbf{F_e} - \nabla \cdot \bar{\mathbf{S}}, \qquad \text{Momentum Equation} \tag{3.9}$$

$$\frac{\partial (\rho \varepsilon)}{\partial t} + \nabla \cdot (\rho \varepsilon \mathbf{v}) = -\nabla \cdot \mathbf{Q} - \bar{\mathbf{S}} \nabla \cdot \mathbf{v}, \qquad \text{Energy Equation} \tag{3.10}$$

where $\mathbf{v}$ is the velocity of the fluid, $\mathbf{Q}$ is the heat flux vector, ε is the internal energy, $\bar{\mathbf{S}}$ is the hydrodynamic stress tensor, $\mathbf{F_e}$ is externally applied forces acting on the entire fluid, and ρ is the fluid density. The fluid density is defined as

$$\rho(\mathbf{r}, t) = \sum_{\alpha} m N_{\alpha}(\mathbf{r}, t), \tag{3.11}$$

where $\mathbf{r}$ is the position vector, m is the mass of the plasma species, and $N_{\alpha}(\mathbf{r}, t)$ is the number density of the particle species α.

The hydrodynamic stress tensor, $\mathbf{S}$, is the summation of the individual stress tensors for each species, $\mathbf{S}_{\alpha}$, given by

$$\mathbf{S} = \sum_{\alpha} \mathbf{S}_{\alpha} = \sum_{\alpha} P_{\alpha} + (\rho \nu_{\alpha} \nabla \mathbf{v}) - \tau, \tag{3.12}$$

where the first term is the summation of the hydrodynamic pressures of each species α, $(\rho \nu_{\alpha} \nabla \mathbf{v})$ is the viscosity for each species α, and τ is the elastic stress deviator for modeling solid materials.

For inviscid plasma flows, after substituting for the hydrodynamic stress tensor, the momentum and energy equations can be written as

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla P + \nabla \cdot \bar{\tau}, \tag{3.13}$$

$$\frac{\partial (\rho \varepsilon)}{\partial t} + \nabla \cdot (\rho \varepsilon \mathbf{v}) = -P \nabla \cdot \mathbf{v} - \nabla \cdot \mathbf{Q} + (\bar{\tau} \cdot \nabla)\mathbf{v}. \tag{3.14}$$

Multiple species in plasmas can be dealt with in several different approaches when it comes to fluid modeling. In general, each species in a plasma, including electrons, ions, and neutrals, have their own set of fluid equations. The fluid equations in equations 3.8, 3.9, and 3.10 are for a single fluid or a single plasma specie. Extension of the equations to multiple species is relatively straightforward. However, momentum and energy coupling terms between species must also be included. The most comprehensive MHD model would have a set of fluid equations with the correct collisional and energy coupling terms for every species in the plasma. To date, good MHD multi-fluid models are not available; this is partly because of the lack of

knowledge of collisional coupling terms, which must be determined experimentally.

There is, however, another approach for handling multiple species. Instead of using a multi-fluid model, a single fluid, multi-temperature model can be used. Modeling separate temperatures for a multi-species plasma is not a well defined concept[61], as temperature is defined by having a system in equilibrium. However, the interesting physics lies in areas of nonequilibrium, and thus it is useful to ascribe the concept of multiple species temperature. Most MHD codes use one, two or three temperature fluid models. The three temperature model assigns a temperature for ions, electrons, and radiation, while the two temperature model assumes electrons and radiation to have the same temperature. The single temperature model assumes electrons, ions, and radiation are in thermal equilibrium. All of these models use a single momentum equation, which eliminates the problem of defining collisional coupling terms. However, separate energy equations are used, and these include energy coupling terms.

In the multi-temperature model, electron and ion temperatures, T_e and T_i, respectively, are given by

$$T_e = \frac{\varepsilon_e}{C_{v,e}}, \tag{3.15}$$

$$T_i = \frac{\varepsilon_i}{C_{v,i}}, \tag{3.16}$$

where $C_{v,e}$ and $C_{v,i}$ are the electron and ion pseudo-heat capacities, which are determined by the equation of state for the material. The heat flux vector can be written in terms of the temperature as

$$\mathbf{Q}_e = \kappa_e \nabla T_e, \tag{3.17}$$

and

$$\mathbf{Q}_i = \kappa_i \nabla T_i, \tag{3.18}$$

where κ_e and κ_i are the thermal conductivities of the electrons and ions.

If the radiation temperature is included in the model, an additional radiative coupling term needs to be subtracted from the electron's energy, Φ_{eR}, which is then given by

$$\Phi_{eR} = \chi_p \rho a c \left(T_e^4 - T_r^4\right), \tag{3.19}$$

where a is Stephen's constant, which is related to the Boltzmann-Stephen constant $\sigma = ac/4$, c is the speed of light, χ_p is the mean Planck opacity, and T_r is the radiation temperature. Radiation transport is given by

$$\frac{\partial(aT_r^4)}{\partial t} = -\rho \mathbf{v} \cdot (aT_r^4) - \frac{4}{3}(aT_r^4)\nabla \cdot \mathbf{v} + \nabla \cdot \left[\rho \chi_r \nabla (aT_r^4)\right] + \Phi_{eR}, \tag{3.20}$$

where χ_r is the Rosseland opacity.

Written in terms of temperature, the equations for the electron and ion internal energies, ε_e and ε_i, become

$$\begin{aligned}\frac{\partial(\rho\varepsilon_e)}{\partial t} + \nabla\cdot(\rho\varepsilon_e\mathbf{v}) &= -P\nabla\cdot\mathbf{v} - \nabla\cdot(\kappa_e\nabla T_e) + (\bar{\tau}\cdot\nabla)\mathbf{v} \\ &\quad + \rho K_{ei}\left(T_i - T_e\right) - \Phi_{eR},\end{aligned} \tag{3.21}$$

$$\frac{\partial(\rho\varepsilon_i)}{\partial t} + \nabla\cdot(\rho\varepsilon_i\mathbf{v}) = -P\nabla\cdot\mathbf{v} - \nabla\cdot(\kappa_i\nabla T_i) + (\bar{\tau}\cdot\nabla)\mathbf{v} - \rho K_{ei}\left(T_i - T_e\right), \tag{3.22}$$

where K_{ei} is a coefficient that quantifies the rate of coupling of the electron and ion energies. This coefficient is given by

$$K_{ei} = C_{v,e}/\tau_{ei}, \tag{3.23}$$

where τ_{ei} is the electron-ion equilibration time.

3.3 Electromagnetic Equations

The fluid equations are not sufficient by themselves to describe the dynamics of a plasma. Charged particles in the plasma will experience a force in the presence of electric or magnetic induction fields. The Lorentz force, $\mathbf{F}$, on a charged particle in SI units is given by

$$\mathbf{F} = q\left(\mathbf{E} + \mathbf{w}\times\mathbf{B}\right), \tag{3.24}$$

where $\mathbf{w}$ is the particle velocity. This equation can be determined experimentally or derived classically in Hamiltonian or Lagrangian formulations[63]. If the particle is observed in a reference frame that is moving with a velocity equal to that of the particle, the particle would appear to be instantaneously at rest. The Lorentz force on a particle in this reference frame would be

$$\mathbf{F}' = q'\mathbf{E}'. \tag{3.25}$$

Within the context of nonrelativistic mechanics ($|\mathbf{w}| \ll c$), physical laws should be invarient under Galilean transformations. Therefore, the Lorentz force must transform as $\mathbf{F}' = \mathbf{F}$. Electric charge is a fundamental constant that is also invarient under Galilean transformations. Comparing equations 3.24 and 3.25, the electric field transforms as

$$\mathbf{E}' = \mathbf{E} + \mathbf{w}\times\mathbf{B}. \tag{3.26}$$

Consider a small volume element of fluid, δV. The Lorentz force on a fluid element, $\delta\mathbf{F_e}$, is the total electromagnetic force acting on δV. $\delta\mathbf{F_e}$, can be determined by summing all of the individual particle electromagnetic forces in the fluid element, which gives

$$\delta\mathbf{F_e} = \sum_n q_n\left(\mathbf{E}' + \mathbf{W}_n\times\mathbf{B}\right), \tag{3.27}$$

where $\mathbf{W_n} = \mathbf{w}_n - \mathbf{v}$ is the peculiar velocity of particle n, $\mathbf{v}$ is the mean fluid velocity, and $\mathbf{E}' = \mathbf{E} + \mathbf{v} \times \mathbf{B}$ is the electric field as observed in a reference frame moving with fluid velocity $\mathbf{v}$.

The second term in equation 3.27 is evaluated with the assumption that the charged particles can be grouped into separate charged species, s, of charge q_s and summed according to each species. This summation can be represented by

$$\sum_n q_n \mathbf{W_n} = \sum_s q_s \left(\sum_n^{(s)} \mathbf{W_n} \right), \tag{3.28}$$

where the second summation $\sum_n^{(s)} \mathbf{W_n}$ is taken only over the particles in volume δV belonging to the species s[64]. Since there are $(N_s \delta V)$ particles of species s in δV, equation 3.28 can be written as

$$\sum_n q_n \mathbf{W_n} = \sum_s q_s \left(\bar{\mathbf{W}}_s N_s \delta V \right), \tag{3.29}$$

where $\bar{\mathbf{W}}_s$ is the mean peculiar velocity and N_s is the number density of species s.

The electric current density is defined as the net rate that charge is transported over a unit area and is given by[61]

$$\mathbf{J} = \sum_s N_s q_s \bar{\mathbf{v}}_s, \tag{3.30}$$

$$= \sum_s N_s q_s \mathbf{v} + \sum_s N_s q_s \bar{\mathbf{W}}_s, \tag{3.31}$$

$$= \eta \mathbf{v} + \mathbf{J}', \tag{3.32}$$

where $\bar{\mathbf{v}}_s$ is the mean velocity of species s and $\eta = \sum_s N_s q_s$ is the charge density.

The first term, $\eta \mathbf{v}$, in equation 3.32, is the convection current density and represents the transport of space charge moving with the mean mass velocity of the fluid, $\mathbf{v}$. The second term represents the current density measured in a reference frame moving with the mean velocity of the fluid, $\bar{\mathbf{v}}$.

Equation 3.28 can now be written in terms of the conduction current density, which gives

$$\sum_n q_n \mathbf{W}_n = \sum_s q_s \left(\bar{\mathbf{W}}_s N_s \delta V \right) = \mathbf{J}' \delta V. \tag{3.33}$$

The total electromagnetic force acting on a fluid element per unit volume is then

$$\mathbf{F}_e \equiv \frac{\delta \mathbf{F_e}}{\delta V} = \eta \mathbf{E}' + \mathbf{J}' \times \mathbf{B}. \tag{3.34}$$

Equation 3.34 represents an additional term that must be added to the fluid mo-

mentum equation 3.21, which gives

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla P + \eta \mathbf{E}' + \mathbf{J} \times \mathbf{B} + \nabla \cdot \bar{\tau}. \tag{3.35}$$

The rate of work by an electromagnetic field on a collection of n particles, P, is given by

$$P = \sum_n \mathbf{w}_n \cdot \mathbf{F}_n \tag{3.36}$$

$$= \sum_n q_n \mathbf{w}_n \cdot \mathbf{E} \tag{3.37}$$

$$= \sum_n q_n \mathbf{v} + \sum_n q_n \mathbf{W}_n \cdot \mathbf{E}. \tag{3.38}$$

It can further be shown that the power delivered by the electromagnetic field to a fluid per unit volume, P_{tot}, is given by

$$P_{tot} = \eta \mathbf{v} \cdot \mathbf{E} + \mathbf{J}' \cdot \mathbf{E}. \tag{3.39}$$

The thermal energy delivered by the electromagnetic field can be determined by subtracting the directed energy of the fluid, P_{de}, given by

$$P_{de} = \mathbf{v} \cdot \mathbf{F_e} \tag{3.40}$$

$$= \eta \mathbf{v} \cdot \mathbf{E}' + \mathbf{v} \cdot \left(\mathbf{J}' \times \mathbf{B} \right), \tag{3.41}$$

from equation 3.39. After some algebra, the thermal energy added to the fluid as a result of the electromagnetic field is found to be

$$P_{th} = \mathbf{J}' \cdot \mathbf{E}'. \tag{3.42}$$

Equation 3.42 must now be added to the electron energy equation 3.21, which gives

$$\begin{aligned} \frac{\partial(\rho \varepsilon_e)}{\partial t} + \nabla \cdot (\rho \varepsilon_e \mathbf{v}) = -P\nabla \cdot \mathbf{v} - \nabla \cdot (\kappa_e \nabla T_e) + (\bar{\tau} \cdot \nabla)\mathbf{v} + \mathbf{J}' \cdot \mathbf{E}' \\ + \rho K_{ei} \left(T_i - T_e \right) - \Phi_{eR}. \end{aligned} \tag{3.43}$$

The addition of the electromagnetic field interaction with the fluid has introduced four new variables, $\mathbf{E}$, $\mathbf{B}$, $\mathbf{J}$ and η, into the conservative fluid equations. In order to close the fluid equations when electromagnetic fields are present, additional equations are needed. These equations are derived through the fundamental Maxwell's equations and the generalized Ohm's law.

The Maxwell equations are well known and in integral form are given by

$$\oint \mathbf{E} \cdot d\mathbf{l} = -\frac{d}{dt} \int_s \mathbf{B} \cdot \mathbf{n} \, da, \qquad \text{Faraday's Law,} \tag{3.44}$$

$$\oint \mathbf{H} \cdot d\mathbf{l} = -\int_s (\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}) \cdot \mathbf{n}\, da, \quad \text{Maxwell-Ampere's Law,} \tag{3.45}$$

$$\int_s \mathbf{D} \cdot \mathbf{n}\, da = \int_V \eta d^3x, \quad \text{Gauss's Law,} \tag{3.46}$$

$$\int_s \mathbf{B} \cdot \mathbf{n}\, da = 0, \quad \text{No magnetic monopoles.} \tag{3.47}$$

In differential form, the Maxwell equations can be written as

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0, \quad \text{Faraday's Law,} \tag{3.48}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}, \quad \text{Maxwell-Ampere's Law,} \tag{3.49}$$

$$\nabla \cdot \mathbf{D} = \eta, \quad \text{Gauss's Law,} \tag{3.50}$$

$$\nabla \cdot \mathbf{B} = 0, \quad \text{No magnetic monopoles,} \tag{3.51}$$

where the electric displacement $\mathbf{D}$ is defined as $\mathbf{D} = \epsilon_o \mathbf{E} + \mathbf{P}$, and $\mathbf{H}$ is given by $\mathbf{H} = \frac{1}{\mu_o}\mathbf{B} - \mathbf{M}$. Here $\mathbf{M}$ and $\mathbf{P}$ are the magnetization and polarization vectors respectively. For isotropic linear media $\mathbf{B} = \mu\mathbf{H}$ and $\mathbf{D} = \epsilon\mathbf{E}$.

The charge continuity equation is a direct consequence of the Maxwell's equations and can be derived by taking the divergence of Maxwell-Ampere's law,

$$\nabla \cdot \mathbf{J} = -\frac{\partial \eta}{\partial t}, \quad \text{Charge Conservation.} \tag{3.52}$$

Faraday's Law, equation 3.48, and Maxwell-Ampere's Law, equation 3.49, can be used to determine the time variations of the electric field and the magnetic induction terms in the fluid equations. Charge conservation, equation 3.52, can be used to determine the time variation of the charge density, η.

To close the fluid equations, an expression that directly relates the current density to the electric field is still needed. This expression is given by the generalized Ohm's Law,

$$\frac{\mathbf{J}}{\sigma} = \mathbf{E} + \mathbf{v} \times \mathbf{B} - \frac{1}{Ne}(\mathbf{J} \times \mathbf{B}) + \frac{1}{Ne}\nabla p_e, \tag{3.53}$$

where p_e is the plasma pressur tensor, σ_{dc} is the DC electrical conductivity given by

$$\sigma_{dc} = \frac{N_e e^2}{m_e \bar{\nu}_{ei}}, \tag{3.54}$$

N_e is the electron number density, e is the fundamental electric charge constant, m_e is the mass of an electron, and $\bar{\nu}_{ei}$ is the average collision frequency between ions and electrons.

Unfortunately, unlike the Maxwell's equations, the generalized Ohm's law is not fundamental and is only an approximate expression. A complete derivation based on plasma kinetic theory can be found in Holt and Haskell[61] and numerous other

plasma physics texts. In this derivation, several assumptions are made about the plasma. First, the plasma is assumed to be quasi-neutral such that the condition

$$\frac{e\left(N_i - N_e\right)}{eN_e} \ll 1, \tag{3.55}$$

where N_i is the ion number density, is satisfied. Hence, $N = N_e = N_i$ and the charge density η, becomes negligible in the localized area of interest. Since η=0, the current density transforms as

$$\mathbf{J} = \mathbf{J}'. \tag{3.56}$$

As discussed in section 3.1, plasmas have a strong tendency to remain electrically neutral. If any space charge develops in a medium, it will decay exponentially with a time constant ϵ_o/σ_{dc}[61].

The second assumption in the derivation of the generalized Ohm's law is that the fluid is collisionally dominated on the order of the characteristic time scale, t_d. This condition is satisfied if

$$t_d \gg \ell_c = 1/\bar{\nu}_{ei}, \tag{3.57}$$

where ℓ_c is the mean free path between collisions.

The second to last term in the generalized Ohm's law, $\frac{1}{ne}(\mathbf{J} \times \mathbf{B})$, represents the Hall electric field. If the electron cyclotron freqency, $\Omega_e = e\mathbf{B}/m_e$, is substituted into the Hall electric field term, the generalized Ohm's law can be written as

$$\frac{\mathbf{J}}{\sigma_{dc}} = \mathbf{E} + \mathbf{v} \times \mathbf{B} - \frac{1}{\bar{\nu}_{ei}}(\mathbf{J} \times \mathbf{\Omega_e}) + \frac{1}{Ne}\nabla p_e. \tag{3.58}$$

The Hall current will be negligible if

$$\frac{|\Omega_e|}{\bar{\nu}_{ei}} \ll 1, \tag{3.59}$$

which is referred to as the Hall coefficient. Thus, for dense Z-pinch plasmas, the Hall term can generally be neglected.

For an electrically neutral plasma that is in steady state and does not contain a magnetic induction field, the generalized Ohm's law, equation 3.53, reduces to

$$\mathbf{J} = \sigma_{dc}\mathbf{E}' \tag{3.60}$$

$$= \sigma_{dc}(\mathbf{E} + \mathbf{v} \times \mathbf{B}). \tag{3.61}$$

It is at this point the fundamental MHD approximation is made. The convection and displacement currents are neglected since they are small compared to the conduction current. This approximation is justified by dimensional considerations[61]. Consider a space charge η, in a plasma given in terms of electric field. The order of magnitude of η is

$$\{\eta\} = \left\{\frac{\epsilon_0 E}{\mathrm{L}}\right\}, \tag{3.62}$$

where Ł is the characteristic length scale. The ratio of the convection current, $\eta\mathbf{v}$, to the conduction current is

$$\left\{\frac{\eta V}{\sigma_{dc}E}\right\} = \left\{\frac{\epsilon_o EV}{Ł\sigma_{dc}E}\right\} = \left\{\frac{\epsilon_o}{\sigma_{dc}t_c}\right\}, \tag{3.63}$$

where V is the characteristic velocity. The ratio of the displacement current to the conduction current yields the same result given by

$$\left\{\frac{\epsilon_o E}{t_c\sigma_{dc}E}\right\} = \left\{\frac{\epsilon_o}{\sigma_{dc}t_c}\right\}. \tag{3.64}$$

Therefore, in order to be able to neglect the displacement current, the following condition must be imposed on the characteristic time scale

$$t_c \gg \epsilon_o/\sigma_{dc}. \tag{3.65}$$

Upon substitution for the DC conductivity, equation 3.54, and the plasma frequency, equation 3.7, this condition can be restated as

$$t_c \gg \bar{\nu}_{ei}/\Pi_{e-}^2. \tag{3.66}$$

This condition, along with condition 3.57 leads to the requirement

$$t_c \gg 1/\Pi_{e-}^2. \tag{3.67}$$

If these conditions are satisfied, the displacement current is neglible compared to the conductive current in Maxwell's equations. Thus, the set of MHD equations that needs to be solved has been reduced to

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0, \tag{3.68}$$

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) = -\nabla P + \mathbf{J}\times\mathbf{B} + \nabla\cdot\bar{\tau}, \tag{3.69}$$

$$\begin{aligned}\frac{\partial(\rho\varepsilon_e)}{\partial t} + \nabla\cdot(\rho\varepsilon_e\mathbf{v}) = &-P\nabla\cdot\mathbf{v} - \nabla\cdot(\kappa_e\nabla T_e) + (\bar{\tau}\cdot\nabla)\mathbf{v} + \mathbf{J}\cdot\mathbf{E}' \\ &+ \rho K_{ei}\left(T_i - T_e\right) - \Phi_{eR}.\end{aligned} \tag{3.70}$$

$$\frac{\partial(\rho\varepsilon_i)}{\partial t} + \nabla\cdot(\rho\varepsilon_i\mathbf{v}) = -P\nabla\cdot\mathbf{v} - \nabla\cdot(\kappa_i\nabla T_i) + (\bar{\tau}\cdot\nabla)\mathbf{v} - \rho K_{ei}\left(T_i - T_e\right), \tag{3.71}$$

$$\nabla\times\mathbf{E} + \frac{\partial\mathbf{B}}{\partial t} = 0, \tag{3.72}$$

$$\nabla\times\mathbf{B} = \mu_0\mathbf{J}, \tag{3.73}$$

$$\mathbf{J} = \sigma_{dc}\mathbf{E}' + \frac{\sigma_{dc}}{Ne}\nabla p_e, \tag{3.74}$$

which result in a set of closed equations for the electromagnetic fluid when supplemented by equation of state relations.

The Maxwell equation, $\nabla \cdot \mathbf{B} = 0$, is taken to be an initial condition. From the divergence of Faraday's law, it can be seen that $\nabla \cdot \mathbf{B} = 0$ will be satisfied for all time when the electric and magnetic fields are observed in the same reference frame. In MHD, the magnetic field is solved with a dynamical equation rather than the vector potential. This can create errors in finite differencing computations. If poloidal field components exist in the computational domain, the solenoidal constraint, $\nabla \cdot \mathbf{B} = 0$, may not be maintained by the finite difference equations . Most MHD codes, including MACH2, have additional routines to iteratively correct for divergence errors in the magnetic field and ensure the solenoidal constraint is satisfied throughout a computation.

It should also be noted that since the charge density does not appear in equations 3.68 through 3.74. The electric field is determined by the generalized Ohm's law and the condition that

$$\nabla \cdot \mathbf{J} = 0, \tag{3.75}$$

which directly follows by taking the divergence of Ampere's Law 3.73 or from the assumption of charge neutrality. This is in contrast to electrostatics whereas the electric field is determined in terms of the charge density. Upon substitution of Ampere's law, equation 3.73, into the generalized Ohm's law, equation 3.74, and solving for the electric field we have

$$\mathbf{E} = \frac{1}{\sigma_{dc}\mu_o}\nabla \times \mathbf{B} - \mathbf{v} \times \mathbf{B} + \frac{1}{Ne}\nabla p_e. \tag{3.76}$$

Number densities for the flux compression simulations presented in chapter 5 range from $10^{22}\ m^{-3}$ to $10^{26}\ m^{-3}$. For high number densities ($\{N\} \gg \{e\}$), the last term in equation 3.76 can be neglected since it will be small compared to the other terms.

The expression for the electric field, equation 3.76, can also be substituted in the $\mathbf{J} \cdot \mathbf{E}'$ term in the electron energy equation 3.70. This gives

$$\mathbf{J} \cdot \mathbf{E}' = \frac{J^2}{\sigma_{dc}}, \tag{3.77}$$

which is known as the joule or ohmic heating term. This term determines the amount of thermal energy added to electron internal energy as a result of finite electrical conductivity in the plasma.

When Faraday's Law, equation 3.72, is substituted into equation 3.76, an expression for the time evolution of the magnetic field or the induction equation is obtained,

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left[\frac{1}{\sigma_{dc}\mu_o}\nabla \times \mathbf{B} - \mathbf{v} \times \mathbf{B}\right] \tag{3.78}$$

$$= \nabla \times (\mathbf{v} \times \mathbf{B}) + \frac{1}{\sigma_{dc}\mu_o}\nabla^2 \mathbf{B}. \tag{3.79}$$

It is important to realize that as a consequence of neglecting the displacement current, the Maxwell Equations no longer allow wave propagation. From equation 3.78, it can be seen that two primary modes exist that propagate magnetic field in MHD problems. The first term represents magnetic diffusion and the last is material advection. A dimensionless number called the magnetic Reynolds number determines the degree to which of these terms is most responsible for the transport of magnetic field. The magnetic Reynolds number will be discussed in section 4.5.

3.4 MACH2 MHD Code

The MACH2 code is a single fluid multi-temperature general Arbitrary Lagrangian Eulerian (ALE) resistive magnetohydrodynamics code that can be used for a wide variety of plasma physics problems. One of the primary reasons this particular code was chosen is the modular style in which it is written. Users are able to add new physics to the code, as well as modify existing aspects of the code. Another reason this particular code was chosen is the enormous flexibility and wide array of physics available to the user. MACH2 has the ability to use materials in a solid state, track multiple materials and interfaces, use analytical or tabular equations of state, and incorporate multi-temperature models. Another benefit MACH2 has over some of the other MHD codes is the extension the MHD model to include the Hall term in the derivation of Ohm's law. One of the biggest reasons the MACH2 code was chosen was the fact that it has already proven to be robust and remarkably successful in modeling Z-pinch plasmas. In particular, Melissa Douglas of SNL and Norm Roderick of the University of New Mexico have used the MACH2 code to investigate Rayleigh-Taylor instability development in Z-pinches[29, 31, 32, 33] and dynamics of nested wire arrays[11, 65].

All of the physical processes in MACH2 are calculated in a time splitting manner. In this approach, each set of algorithms that describes a physical process is solved in a sequential manner, rather than solving all of the equations simultaneously. After each physical process is calculated, the internal energy is updated. While this approach is not ideal since it can lead to numerical error, it allows for significantly faster convergence of solutions and, more importantly, it allows significant future code modification to be made relatively easily. Numerical error introduced by time-splitting is directly proportional to the size of the time step used. As long as computational time steps are kept relatively small (compared to the time scale of physical processes modeled), the error introduced in time-splitting is also small. In the simulation runs for this research, the maximum time step is 0.5 ns and 10 ps during peak compression, which is sufficient to avoid significant time-splitting error. However, it should also be noted that the order in which the physical processes are solved in the time splitting approach can affect the result to some degree. The code designers of MACH2, over the course of time, have determined the order that is the most accurate. The code solves radiation, thermal diffusion, magnetic diffusion, lagrangian hydrodynamics and finally, convective derivatives in that order.

The MACH2 code uses finite volume differencing methods, which are solved on

a structured grid of arbitrary quadrilateral cells. Only material advection, which uses a 2nd order van Leer model, is calculated explicitly, while all diffusive processes including magnetic, thermal, and radiation diffusion, are calculated implicitly for maximum numerical stability. In general, Lagrangian hydrodynamics are also calculated implicitly using a Newton procedure with a Jacobi iteration but the user can define a time-centering parameter, which can make the calculation fully explicit, or any degree between fully implicit or fully explicit.

The MACH2 code provides the user with several different model choices based on analytic equations or pregenerated tabular lookup tables for equation of state, mean ionization state, classical resistivity, anomalous resistivity, thermal conductivity, photon opacity, and shear velocity and elastic yield parameters. The code also has a variety of model choices for external electromagnetic circuits.

By no means is the MACH2 code the only MHD code well established for looking at Z-pinch plasmas. A classified code called LASNEX has been extensively used, especially when investigating radiation transport[27, 34, 36, 37, 38, 66, 67, 68], and a very capable code developed at Imperial college has yielded considerable insight into the physics of Z-pinch plasmas[44, 46, 50]. Currently in the final stages of development at SNL is a extremely capable 2D and 3D resistive MHD code called ALEGRA.

3.5 Boundary Conditions

The boundary conditions are the most important aspect of any computational MHD model. Hydrodynamic boundary conditions are discussed adequately elsewhere and are relatively straight forward to implement. Electromagnetic boundary conditions, however, are not as straight forward. For example, how does one apply an external circuit to a MHD problem? In this section, a general discussion of electromagnetic boundary conditions is presented. In the next chapter, specifics of how the electromagnetic boundary conditions are applied to the flux compression simulations are given.

The electromagnetic boundary conditions in differential form at interfaces between different media, designated by subscripts 1 and 2, are well known, and are given by[69]

$$\mathbf{n} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{K}, \tag{3.80}$$

$$\mathbf{n} \times (\mathbf{E}_1 - \mathbf{E}_2) = 0, \tag{3.81}$$

$$\mathbf{n} \cdot (\mathbf{B}_1 - \mathbf{B}_2) = 0, \tag{3.82}$$

$$\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \sigma_s, \tag{3.83}$$

where σ_s is the surface charge density, $\mathbf{K}$ is the surface current density, and $\mathbf{n}$ is a unit vector perpendicular to the surface.

Electromagnetic boundary conditions in MACH2 are defined through the magnetic field only, as the magnetic field is a fundamental quantity in the code, whereas

the electric field is not. Therefore equations 3.81 and 3.83 must be transformed in terms of magnetic field.

3.5.1 Perfect Conductor

At a conductor boundary, magnetic field will penetrate a finite distance into the conductor called the skin depth. In the limit of a perfect conductor, the skin depth approaches zero. The magnetic field in a perfect conductor cannot change and is said to be frozen into the conductor. If there is no initial embedded magnetic field in the conductor, the magnetic field, $\mathbf{B}_2$ can be assumed to be zero at all times. Then, equation 3.82 becomes

$$\mathbf{n} \cdot \mathbf{B}_1 = 0, \tag{3.84}$$

when medium 1 has a finite conductivity. The electric field, $\mathbf{E}_2$, inside a perfect conductor is also zero and equation 3.81 becomes

$$\mathbf{n} \times \mathbf{E}_1 = 0. \tag{3.85}$$

For MHD computations, this equation must be written in terms of the magnetic field only. The electric field can be substituted for by rearranging Ohm's law, equation 3.74. We then have

$$\mathbf{n} \times \frac{\mathbf{J_1}}{\sigma_{dc}} - \mathbf{n} \times (\mathbf{v_1} \times \mathbf{B_1}) = 0. \tag{3.86}$$

Using the vector product rule, this equation can be expanded giving

$$\mathbf{n} \times \frac{\mathbf{J_1}}{\sigma_{dc}} - (\mathbf{n} \cdot \mathbf{B_1})\mathbf{v_1} - (\mathbf{n} \cdot \mathbf{v_1})\mathbf{B_1} = 0. \tag{3.87}$$

The second term is zero as a consequence of equation 3.84. If a fixed boundary is assumed, $\mathbf{n} \cdot \mathbf{v} = 0$, and the third term cancels. Therefore, the current is only perpendicular to a perfect conductor boundary layer surface and is given by

$$\mathbf{n} \times \mathbf{J_1} = 0. \tag{3.88}$$

Ampere's law, equation 3.73 can be substituted for the current density and the vector triple product rule can be used to give

$$\mathbf{n} \times (\nabla \times \mathbf{B_1}) = (\mathbf{n} \cdot \nabla)\mathbf{B_1} = 0. \tag{3.89}$$

In axial symmetric cylindrical coordinates, $\partial/\partial\theta = 0$. The toroidal and poloidal fields are now separable and given by

$$\mathbf{n} \cdot \nabla(rB_{1\theta}) = 0 \tag{3.90}$$

$$\mathbf{n} \cdot \nabla(B_{1z}) = 0 \tag{3.91}$$

These equations along with 3.84 represent the boundary conditions at the surface of a perfect conductor.

3.5.2 Insulator

Insulating walls do not allow current flow into or out of the wall. Thus

$$\mathbf{n} \cdot \mathbf{J} = \mathbf{n} \cdot \nabla \times \mathbf{B} = 0, \tag{3.92}$$

which says that the magnetic field is tangent to the insulating surface. In cylindrical coordinates, equation 3.92 is given by

$$\begin{aligned} \mathbf{n} \cdot \nabla \times \mathbf{B} &= \frac{1}{r} n_z \frac{\partial}{\partial z}(r B_\theta) - \frac{1}{r} n_r \frac{\partial}{\partial r}(r B_\theta) && (3.93)\\ &= n_z \frac{\partial}{\partial z}(B_\theta) - \frac{1}{r} n_r \frac{\partial}{\partial r}(r B_\theta) && (3.94)\\ &= 0. && (3.95) \end{aligned}$$

Since the function of derivatives is equal to zero, rB_θ is a constant. The constant can be determined by integration of Ampere's law, equation 3.73,

$$\begin{aligned} \frac{1}{\mu_o} \int_s \mathbf{J} \cdot \mathbf{n} da &= \int_s \nabla \times \mathbf{B} da && (3.96)\\ &= \oint \mathbf{B} \cdot dl && (3.97)\\ &= 2\pi r B_\theta && (3.98)\\ &= \mu_o I, && (3.99) \end{aligned}$$

which gives

$$B_\theta = \frac{\mu_o I}{2\pi r}. \tag{3.100}$$

This condition, which says rB_θ is constant along insulating boundaries, is quite restrictive on the toroidal magnetic field component. For the poloidal field components however, there are no boundary restrictions at all. Therefore, it is usually necessary to define a boundary well outside of the region of interest. Fortunately, for problems with axial symmetry such as the Z-pinch, there are no poloidal field components.

Chapter 4

Flux Compression Model

Computational MHD modeling of Z-pinch flux compression is not a trivial task. In addition to the complexities added to the modeling simple Z-pinches, flux compression adds new boundary condition complexities, multi-component circuit modeling, and equation of state and radiation issues. This chapter will address some of these issues and discuss important aspects of the model developed in MACH2 for flux compression simulations on the DQ.

4.1 Flux Compression on the Decade Quad

The flux compression technique that is studied in this research is a concept by John Goyer of Titan Pulse Sciences designed specifically for the DQ. The primary reason that Goyer's design was chosen for this study is the fact that it is the leading design candidate to be investigated experimentally on the DQ. Another reason this particular nested flux compression design was chosen for this study is the meager theoretical or computational studies of nested wire array flux compression to this point.

Figure 4.1 shows a schematic diagram of how flux compression is implemented in Goyer's design. Current from the DQ generator is fed into the load at two points; one point is the top of the outer anode wall and the other is the top of the conical stator. Current returns to the cathode plate from both circuit paths through the Al wire array, which acts as the armature in this configuration. Current in the secondary path flows in an opposite direction as the current flowing through the primary path. The opposite flow of currents sets up an opposing azimuthal magnetic field configuration separated by the Al plasma armature. The majority of impedance in both circuit paths is inductive rather than resistive. Therefore, current distributions between the two circuit paths are determined by inductance impedance.[1] Not shown in the schematic are anode feed posts which connect to the stator and add an additional 60 nH of inductance to the secondary circuit path.

[1] As long as the plasma armature remains continuous, no magnetic field lines cross both circuit paths; therefore, mutual inductance need not be considered.

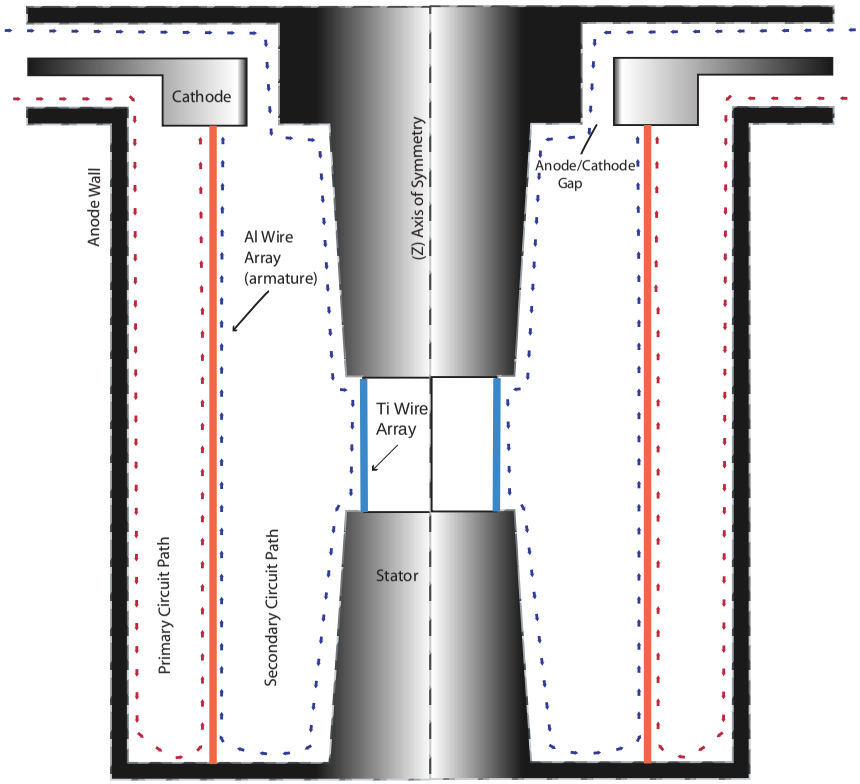

Figure 4.1: Schematic of nested flux compression design for DQ. The primary current paths is designated by red arrows and the secondary current path is designated by blue arrows.

The initial geometric configuration, wire array radii, and anode feed post length can be tailored to adjust the magnetic pressure on both sides of the armature. In this particular design, the initial inductance of the primary path is approximately 5 nH while the inductance of the secondary path is 80 nH (20nH+60nH). Most of the current will initially flow in the primary path due to the smaller circuit path inductance. This allows the Al wire array to vaporize, merge, and begin imploding before the Ti wire array begins to vaporize.

As the Al plasma armature implodes, it begins compressing magnetic flux in the secondary current path between the armature and the stator. At the same time, the inductance in the primary circuit path rises while inductance in the secondary path decreases due to the change in circuit path lengths. When the plasma armature has crossed the anode/cathode or injection gap, it effectively traps the existing magnetic flux in the secondary circuit between the Al plasma, conical stator, and Ti wire array. This is referred to as the crowbar[2] switching phase since the current has now been "crowbarred" from the driving circuit. Magnetic flux in the secondary circuit path continues to be compressed by the momentum of the Al plasma armature. The magnetic flux sustains current flow until all of the magnetic energy has been converted into thermal energy via ohmic heating in the circuit path. If the conductivity of the circuit path is high, the magnetic flux is conserved on the time scale of the armature implosion. Specifically, the armature will continue to compress magnetic flux and raise secondary current levels until the implosion speed is equal to the magnetic diffusion speed in the plasma. At this point the turnaround radius has been achieved and the plasma armature begins accelerating away from the axis.

The nested design allows the Ti wire array to be placed at a much smaller radius than would be required to achieve sufficient velocities in a standard wire array Z-pinch configuration. The greatest benefit of the smaller radius is that it minimizes the time over which RT instabilities can grow. Power is significantly increased in the Ti wire array since flux compression effectively squeezes the current pulse in a manner which significantly increases the pulse amplitude and reduces the secondary pulse rise-time. Theoretically, this should allow the Ti plasma to receive sufficient energy to vaporize and attain the kinetic energy needed for efficient K-shell radiation production without requiring a large initial radius. RT instabilities should also be mitigated to some degree in the Al plasma armature due to the magnetic pressure between the stator and armature. Although RT instabilities in the plasma armature are not as important, from a radiation standpoint, as they are in the Ti plasma, they can affect the dynamics of the flux compression. If RT instabilities are severe, the secondary current path may be shorted, severely limiting flux compression. Armature RT instabilities are the primary reason for the conical nature of the stator. The conical design is intended to help prevent RT instabilities from shorting out the flux compression, and it also helps funnel flux to the Ti plasma during compression.

Numerous parameters can be optimized in this flux compression design. Wire

[2] A crowbar device short-circuits a high voltage to ground.

mass loads, initial radius of wire arrays, position and width of injection gap, wire array lengths, conical stator angle, and secondary anode post inductance are the primary parameters that can be optimized. Of particular note is the angle of the conical stator. This angle is a tradeoff between power of the secondary current pulse and the prevention of shorting out flux compression by RT armature instabilities. 2D simulations may give considerable insight for this particular optimization. It should also be pointed out that pulse shaping of the secondary current can be achieved with careful geometric design of the stator. This could be extremely important when considering flux compression in isentropic compression experiments (ICE), where shocks must be carefully controlled[70].

A couple of other techniques have also been suggested for implementing flux compression with Z-pinches. A French design that has been tested on the Z accelerator at SNL [16] is the basis for the design of the new ECF2 generator at the Centre d'Etudes de Gramat in France[17]. This design is similar to Goyer's design in current flow, but rather than nested wire arrays, the arrays are in a tiered configuration, where flux is compressed along a fixed stator into a separate pinch region. Other flux compression designs use an externally generated axial B_z field, which is compressed[15, 71, 72]. Linear RT instability suggests that a B_z field will significantly reduce RT growth rates[10]. However, evidence suggests that a compressed B_z field limits energy transfer and radiation yeilds[73]. Another design proposed specifically for the DQ is a current doubler flux compression design that uses two stages of power amplification[74]. Goyer's design has several benefits over these designs, including lower total circuit inductance. The opposing magnetic fields should also help mitigate instabilities, to some degree, during the run-in phase. Potential problems of Goyer's design is Al plasma jetting into the stator gap which would lead to highly nonuniform implosions and radiation coupling issues.

A couple of potential problems with flux compression have already been stated, including RT instabilities altering implosion dynamics. Magnetic diffusion into the electrodes, the stator, or the Ti and Al plasmas will produce flux losses that limit flux compression. In order to establish the seed magnetic flux in the secondary circuit path, a small prepulse current must flow. Prepulse currents are believed to be generally disadvantageous for Z-pinches, since the plasma will have greater thermal expansion, which enhances RT instabilities[75]. In the ablative wire model, RT instabilities or an Al precursor plasma could prematurely crowbar the injection gap, which also would severely limit flux compression. Electrode deformation due to the enormous magnetic fields could also present problems. During peak compression, magnetic field strengths can be on the order of 500 Tesla. Lastly, radiation cooling and coupling between the arrays could be factors in the observed Ti K-shell emission.

4.2 Flux Compression Circuit Model

Perhaps the most important aspect of the flux compression simulation is the circuit model used to apply the appropriate magnetic field boundary conditions. It is crucial to model the dynamically changing magnetic boundary conditions accurately if any

type of predictive analysis is to be used from model simulations.

Unfortunately, the existing circuit models in the MACH2 code have key limitations for flux compression simulations. While the MACH2 circuit routines allow multiple current circuits to be defined for a simulation, they must be independent of each other. The physical flux compression circuit under study consists of two current paths which are not independent but are driven by the same external circuit. Ideally, the circuit model should be able to calculate the current distributions in each computational domain current path based on relative impedance. This would enable the model to have a predictive capability. Current distributions obtained from simulations can then be compared with experimental results to test the accuracy of the model.

Previous flux compression simulations have used self-consistent RL circuit models with an applied experimental voltage for the outer wire array or armature[76]. In 2D simulations, the current in the inner array is imposed and given by experimental or 1D circuit data[73]. Both circuit currents are multiplied by a constant factor to match experimentally observed implosion times. This approach has several disadvantages including severely limiting the models predictive capability by relying heavily on experimental data. Since the current in the inner array is imposed and not calculated dynamically, magnetic field interaction effects are not treated self-consistently.

It was therefore decided that flux compression simulations for this study should use a self-consistent circuit model for both wire arrays that would predict the current distributions in each current path from a single generator voltage source. It was also decided that experimental data should not be used to determine model parameters or boundary conditions. Experimental data would be limited to the comparison of simulation results and subsequent adjustment of unknown intrinsic circuit parameters by comparing experimental and simulation results. In order to achieve these objectives, a new circuit model was developed for the MACH2 code.

The circuit model developed for the flux compression simulations presented in this research has four separate components. The first component is a simple external circuit which approximates the electrical characteristics of the DQ, including the Marx Banks, transfer capacitors, etc. The second component calculates the effective inductance of each circuit path in the computational domain by comparing regions of magnetic flux with current values from the previous computational cycle. The third component models the crowbar switch with MACH2's interface tracker and a variable resistance that depends on the average radius of the Al plasma armature. Finally, the fourth component couples the results from the other circuit components into a nonlinear second order circuit solver that parametrically solves for current and voltage on every circuit component at each time step. For a schematic representation, an equivalent circuit for the model is shown in Figure 4.2. The variable resistors in Figure 4.2 represent resistive voltage due to magnetic diffusion in imperfect conductors. Although the resistive impedance is small compared to the inductive impedance during the pinch, resistive impedance should not be neglected during peak flux compression, when the inductance is driven towards zero. Unfortunately,

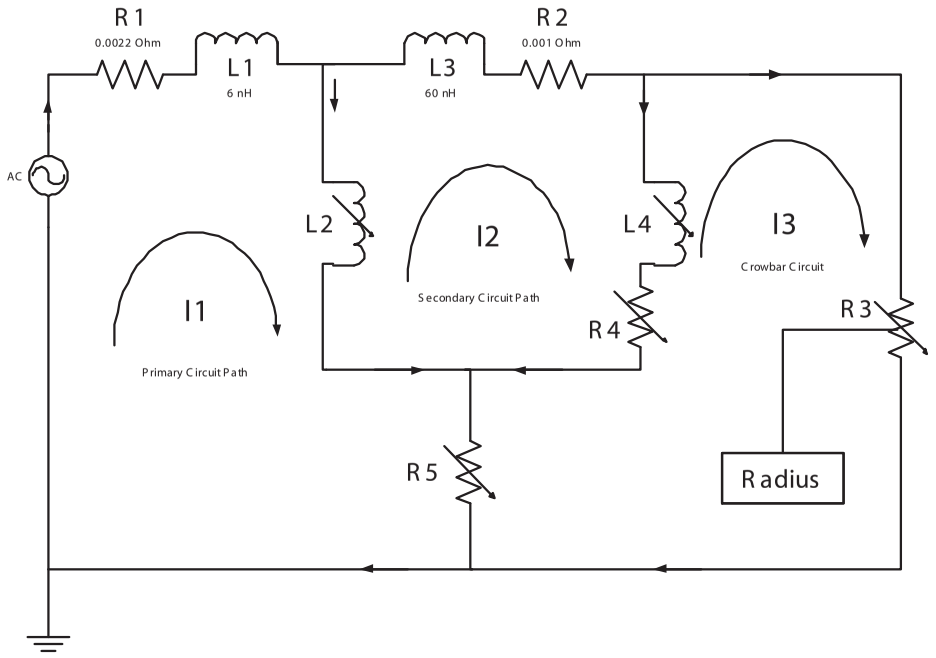

Figure 4.2: Circuit diagram for flux compression circuit model. Current flow is shown three current loops as I1, I2 and I3. The AC source, R1, and L1 represent the external electrical characteristics of the DQ generator. Inductor L3 represents the additional inductance from the anode feed posts. Variable inductors L2 and L4 represent inductance in the circuit paths for the Al and Ti plasma columns. Variable resistors R4 and R5 represent resistive impedance losses due to imperfect conductors. The crowbar switch is represented as a resistor R3 whose value is determined from predetermined parameters and the armature radius. The smaller arrows along the circuit path represent the general flow of current in the circuit.

MACH2's standard routines for calculating these voltages are also inadequate for this problem. In this model, new routines were written so that ohmic heating losses in each circuit path were divided by circuit path current to obtain a characteristic dynamic resistive impedance.

4.2.1 Model Details

The external circuit component of the flux compression model attempts to model the electrical characteristics of the DQ including the Marx banks, transfer capacitors, etc. This is perhaps the most difficult element to model due to several complex pulsed power elements. Detailed circuit models for the DQ have been developed by Navy Research Laboratories (NRL)[3] and Titan Pulsed Sciences Corporation[77]. However, these models are based on engineering design and not necessarily indicative of the physical operating characteristics. One case in point is recent gas puff shots on the DQ show currents twelve to twenty percent lower than predicted by the advanced circuit models developed specifically for the DQ[78]. While the complex circuit models developed for DQ demonstrate good agreement with experimental data in general, they are unnecessarily complex for the needs of flux compression simulations when one considers the uncertainties involved. Another complexity in pulsed power components is the unknown time dependent electrical characteristics[79] which are generally ignored. It was therefore decided that a much simpler external circuit model be developed, albeit somewhat heuristically, from experimental DQ short circuit data.

An RLC circuit is a good approximation for a Marx bank, especially when the load is assumed to have a slight capacitance. In reality, parameters R and L can have significant time dependent effects that should also be considered and are generally ignored. A RLC approach to the external circuit modeling is somewhat misguided when one considers that the DQ consists of not only Marx banks, but transfer capacitors, pulse forming lines, gas switches, and other pulsed power components. Despite this, an attempt was made to determine RLC parameters from experimental short circuit data that would conceivably approximate the entire electrical configuration of the DQ. Only parameters R and L are determined from experimental data. The parameter C is determined by the known capacitance of the Marx Bank ($4.4\mu F$). The initial voltage on the capacitor is determined by the charge voltage on the Marx Banks, which also determines available circuit energy of $85kJ\times$ 24 banks= 2.04 MJ. The experimental short circuit data is obtained by bolting an Al plate across the anode and cathode of the DQ load. Upon determining these parameters, the RLC approximation yields a characteristic impedance that is too high and therefore predicts current risetimes nearly double that which is measured. This level of discrepancy in a fundamental electrical characteristic of the generator is unacceptable.

A better approximation of the DQ electrical configuration is an RLC circuit that models Marx banks connected in series with a pair of ideal lossless transmission lines that model the other pulsed power components. This approach could conceivably

use realistic parameters for the Marx Bank, and the overall impedance and timing could be adjusted in the transmission line parameters. However, it is not trivial to fit experimental data in this approximation due to the need to model voltage reflections accurately. In addition, computational modeling of transmission lines requires the coding of time-value breakpoints, which adds another layer of complexity to the model. However, John Goyer has developed a similar circuit model for the DQ that uses a voltage pulse profile from the Marx Banks coupled to a pair of transmission lines. This particular model closely matches actual amplitudes, timing, and circuit impedance.

In the literature, previous Z-pinch simulations implement a voltage profile coupled to a RL circuit, which is determined to be an equivalent circuit for the external circuit model[29, 33]. It was therefore decided to utilize the same technique for this research. An experimental voltage profile coupled to an RL circuit is used to model all of the combined electrical effects of the DQ. The experimental voltage profile is taken from the output line of soft short circuit shots, which should not significantly vary with different wire array load configurations. The R and L circuit parameters are adjusted to match current amplitude and timing characteristics. This approach is quite heuristic, but turns out to be a reasonable approximation, considering the uncertainties involved.

The key contribution the external circuit model will provide is defining peak current amplitudes and risetimes in the Z-pinch. Therefore, a small degree of error or uncertainty in the external circuit model will have a very minor effect on the flux compression dynamics or physics. As flux compression experimental data becomes available, the external circuit parameters can be adjusted to better match results.

The second component of the flux compression circuit model is the dynamic impedance calculator, in which both the primary and secondary wire array circuit path impedances are determined self-consistently. The MACH2 code, as distributed, provides three different options for calculating computational domain impedance, which is referred to as the port voltage. One method involves calculating resistive $\int \mathbf{E} \cdot d\mathbf{l}$ impedance along the non current-carrying external boundaries. Another method considers inductive impedance and implements Faraday's Law by integrating magnetic flux to determine the port voltage. A third method is based on energy conservation, which is described elsewhere[80]. Since inductive impedance dominates Z-pinch implosions, the Faraday method is a good choice. The $\int \mathbf{E} \cdot d\mathbf{l}$ would be more suited for problems when resistive impedance is more important. Although the Faraday method is the most reasonable choice, the manner in which it was coded assumes that only one circuit path exists in the simulation. The routine only integrates magnetic flux over the entire computational domain. While the flux compression simulations only have one external circuit, they have two current paths, a feature which makes this model impossible to use in this case. This model is also very noisy in voltage calculations and leads to numerous numerical difficulties. Thus, a new variant of the Faraday model was coded.

The biggest problem in modifying the Faraday model was to discriminate which circuit path a particular cell's magnetic flux belonged to in order to accurately

calculate the amount of magnetic flux in each circuit path. A simple approach for this particular problem would be to run the code with Lagrangian block boundaries and predetermine which blocks are associated with each current path. However, this approach neglects magnetic field diffusion in conductors and prevents the use of Eulerian grids. Thus, a routine was developed based on MACH2's material tracker to track regions of magnetic flux based on the material in a cell.

In this routine, vacuum regions in the computational domain that are separated by conductors must be defined as separate materials in the MACH2 code. Although the vacuum regions are separate, their properties are defined identically. Magnetic flux that belongs to a particular circuit path is now defined as the magnetic flux contained in a particular vacuum material.

Another issue arises when the finite conductivity of conductors is considered since magnetic flux can now exist in the skin depth of the conductor. Inside conductors, the sign of the magnetic field is analyzed on a cell by cell basis to determine which circuit path the flux belongs to. Since multiple magnetic flux regions or current loops can now be discriminated, the inductance in each circuit path can be determined with equation 1.3 by dividing the calculated flux by the previous cycle's current.

For the fourth component of the circuit model, inputs from the prior components are put into a general flux compression circuit, which is solved for new magnetic boundary conditions. The equivalent circuit developed for this process is shown in Figure 4.2. Circuit component values in which data is obtained directly from the simulation are shown in red. Resistor R1 and inductor L1 represent the external LR circuit discussed earlier. Resistor R2 helps stabilize the circuit equations and represents a small amount of resistance in the anode current posts. The crowbar resistance is represented by the variable resistor R3. The primary and secondary wire arrays are represented by variable inductors L2 and L4. L3 represents the additional inductance in the secondary wire array due to the anode current posts. Variable resistors R4 and R5 represent resistive impedance due to the imperfect plasma shell conductors. I1, I2, and I3 represent the mesh currents that are to be determined.

Instead of directly calculating the resistive impedance in the simulation, a different approach is taken. First, the joule heating rate is calculated by tracking the joule heating rate of each material. When considering only resistive impedance, the power loss is defined as I^2R. A characteristic resistance is then determined by dividing the magnetic flux by the current squared in the appropriate circuit path.

The crowbar resistance is determined from the radius of the outer plasma shell and two predetermined radii and resistance values defined in the input file. The first resistance value given is the resistance of the crowbar before the radius has reached its first value. Similarly, the second resistance defined is the resistance of the crowbar after the plasma shell has reached the second radius. During the time in which the plasma shell is between the two given radius values, the resistance is calculated as an exponential function between the two defined crowbar resistance values. The radius of the outer plasma shell is calculated using MACH2's built-in interface tracker.

Using Kirchoff's laws and mesh analysis, the equations for this circuit can be written as follows:

$$I_1 R_1 + L_1 \frac{dI_1}{dt} + \frac{d}{dt}\left[L_2\left(I_1 - I_2\right)\right] + R_5\left(I_1 - I_3\right) = V_c(t) \tag{4.1}$$

$$L_3 \frac{dI_2}{dt} + \frac{d}{dt}\left[L_2\left(I_2 - I_1\right)\right] + \frac{d}{dt}\left[L_4\left(I_2 - I_3\right)\right] + I_2 R_2 + R_4(I_2 - I_3) = 0 \tag{4.2}$$

$$I_3 R_3 + \frac{d}{dt}\left[L_4\left(I_3 - I_2\right)\right] + R_5(I_3 - I_1) + R_4(I_3 - I_2) = 0, \tag{4.3}$$

where

$$\frac{d}{dt}(L_i I_j) = L_i \frac{dI_j}{dt} + I_j \frac{dL_i}{dt}, \tag{4.4}$$

and

$$\tag{4.5}$$

$$R_3 = R_3(t), \tag{4.6}$$

$$R_4 = R_4(t), \tag{4.7}$$

$$R_5 = R_5(t), \tag{4.8}$$

$$L_2 = L_2(t), \tag{4.9}$$

$$L_4 = L_4(t). \tag{4.10}$$

4.2.2 Circuit Solver

The first order coupled flux compression circuit equations cannot be solved analytically since nonlinear elements exist. Although the dependent variables, I1, I2, and I3, and its derivatives are of the first degree, not every coefficient depends on only the independent variable, t. The inductors, L2 and L4, have a functional form that depends on not only time, but also on the dependent variable radius. Resistors R3, R4, and R5 are also nonlinear elements since they do not have linear dependence with time or current. An iterative numerical approach is therefore implemented in the code. In order to use this approach, the circuit equations must first be linearized. This is achieved using standard discretization techniques based on the backward Euler approximation[81, 82]:

$$\frac{d}{dt}x^{n+1} = \frac{x^{n+1} - x^n}{\Delta t}. \tag{4.11}$$

One important difference in the implementation of this equation should be noted. The time step (n) is not fixed in the simulations and can vary to maintain optimum numerical speed, accuracy, and stability. Changes in the computational time steps introduces some noise to the finite differencing solutions. However, if changes in time steps between cycles are kept small, the noise introduced is small compared to the solution.

Upon discretization of the circuit equations, the following set of equations is obtained:

$$I_1^{n+1}\left(R_1+\frac{L_1+2L_2^{n+1}-L_2^n}{dt}\right)+I_2^{n+1}\left(\frac{-2L_2^{n+1}+L_2^n}{dt}\right)= \tag{4.12}$$
$$=\frac{L_1I_1^{n+1}}{dt}+\frac{L_2^{n+1}\left(I_1^n-I_2^n\right)}{dt}+V_c-R_5\left(I_1^{n+1}-I_3^{n+1}\right),$$
$$I_1^{n+1}\left(\frac{-2L_2^{n+1}+L_2^n}{dt}\right)+R_2+I_2^{n+1}(\frac{2L_2^{n+1}+L_2^n+L_3-2L_4^{n+1}+L_4^n}{dt})+ \tag{4.13}$$
$$+I_3^{n+1}\left(\frac{-2L_4^{n+1}+L_4^n}{dt}\right)+R_4(I_2^{n+1}-I_3^{n+1})=$$
$$=-\frac{L_3I_2^{n+1}}{dt}+\frac{L_4^{n+1}(I_2^n+I_3^n)}{dt},$$
$$I_2^{n+1}\left(\frac{-2L_4^{n+1}+L_4^n}{dt}\right)+I_3^{n+1}\left(R_3+\frac{2L_4^{n+1}-L_4^n}{dt}\right)= \tag{4.14}$$
$$=-\frac{L_4^{n+1}(I_2^n+I_3^n)}{dt}+R_4(I_3^{n+1}-I_2^{n+1})+R_5(I_1^{n+1}-I_3^{n+1}).$$

The discretized equations can now be formed into a linear matrix equation of the form

$$\bar{\mathbf{A}}\vec{\mathbf{X}}=\vec{\mathbf{B}}, \tag{4.15}$$

which can easily solved computationally for the current values at each time step by the following matrix operations:

$$\bar{\mathbf{A}}\vec{\mathbf{X}}\bar{\mathbf{A}}^{-1}=\vec{\mathbf{B}}\bar{\mathbf{A}}^{-1} \tag{4.16}$$

$$\vec{\mathbf{X}}=\vec{\mathbf{B}}\bar{\mathbf{A}}^{-1}. \tag{4.17}$$

The discretized circuit equations can be put into this form by collecting all (n+1) current terms into matrix $\bar{\mathbf{A}}$ while making vector $\mathbf{X}$ the current vector to be solved at the (n+1) time step. All other terms are collected into the vector $\mathbf{B}$. Thus matrix $\bar{\mathbf{A}}$ will have the form

$$\bar{\mathbf{A}}=\begin{pmatrix} R_{15}+\frac{L_1+2L_2^{n+1}-L_2^n}{dt} & \frac{-2L_2^{n+1}+L_2^n}{dt} & -R_5 \\ \frac{-2L_2^{n+1}+L_2^n}{dt} & R_{24}+\frac{2L_2^{n+1}+L_2^n+L_3-2L_4^{n+1}+L_4^n}{dt} & \frac{-2L_4^{n+1}+L_4^n}{dt}-R_4 \\ -R_5 & \frac{-2L_4^{n+1}+L_4^n}{dt} & R_{345}+\frac{2L_4^{n+1}-L_4^n}{dt} \end{pmatrix}, \tag{4.18}$$

where $R_{15}=R_1+R_5$, $R_{24}=R_2+R_4$, and $R_{345}=R_3+R_4+R_5$. Vectors $\mathbf{B}$ and $\mathbf{X}$

have the form

$$\vec{\mathbf{B}} = \begin{pmatrix} \frac{L_1 I_1^n}{dt} + \frac{L_2^{n+1}(I_1^n - I_2^n)}{dt} \\ -\frac{L_3 I_2^n}{dt} + \frac{L_4^{n+1}(I_2^n + I_3^n)}{dt} \\ -\frac{L_4^{n+1}(I_2^n + I_3^n)}{dt} \end{pmatrix}, \tag{4.19}$$

$$\vec{\mathbf{X}} = \begin{pmatrix} i_1^{n+1} \\ i_2^{n+1} \\ i_3^{n+1} \end{pmatrix}. \tag{4.20}$$

The following sets of operations are executed in the **matrixsolve.F** C.3 subroutine. Since matrix $\bar{\mathbf{A}}$ may contain components that are zero, a sparse linear matrix solver was implemented into the code. The numerical recipes **linbcg** routine[83] was chosen for the sparse linear matrix solver for its ease of implementation and availability.

Once all of the current values have been obtained, the correct magnetic boundary conditions are determined and applied to the computational domain. The MACH2 code then proceeds to compute all physical process including magnetic diffusion and hydrodynamics. At this point, a single computational cycle is complete. At the beginning of the next cycle, the new circuit path inductances are determined and circuit equations are solved again for the new magnetic boundary conditions.

Numerically, a question that arises is how to start the simulation since the new circuit model is an iterative process. The inductance is calculated by the flux in the simulation, initial flux in the simulation is determined by the current in each path, and the current in each path is determined by the inductance. Therefore, in order to start the problem an initial magnetic field, initial path inductance, or initial currents must be given. For the described flux compression circuit model, the initial path inductances were chosen as the initial starting point and must be given at the start of the calculation.

4.2.3 Model Tests

Before using the new flux compression circuit model in simulations, the model was rigorously tested for accuracy. An ideal test that could be compared with analytical calculations was developed. This test is similar to a standard coaxial cable but consists of a third outer conducting shell. The currents along the surfaces of each shell, shown in Figure 4.3, are similar to those of a flux compression Z-pinch experiment. In this case, the self inductance of the innermost and outermost conducting shells can be analytically calculated from the magnetic field energy equations since the radius of each shell is fixed and the shells remain perfectly uniform. Due to the sufficient geometric symmetry of the problem, Faraday's law for magnetic induction can be used to calculate the magnetic field between the cylinders, which gives

$$\mathbf{B}_{sec} = \frac{\mu_0 I_{sec}}{2\pi r}\hat{\theta}, \tag{4.21}$$

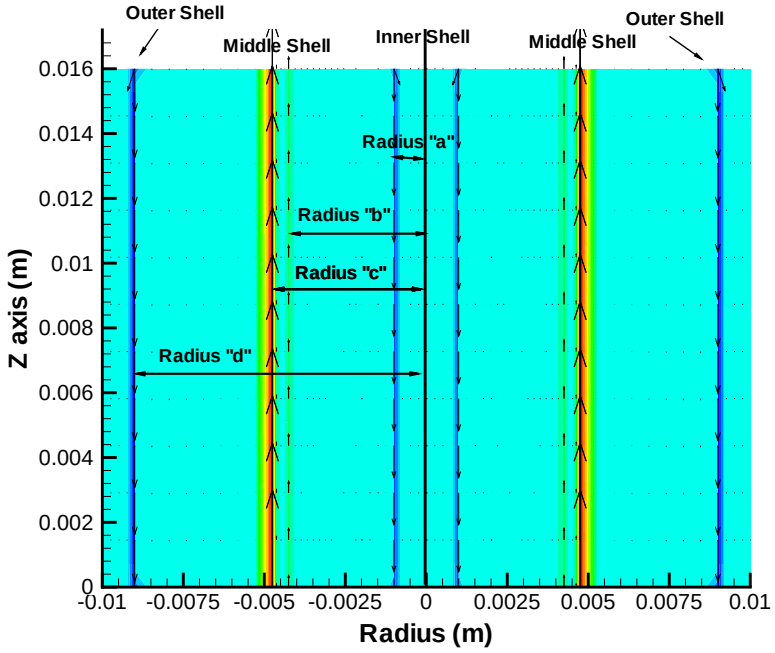

Figure 4.3: Configuration for flux compression circuit model test

$$\mathbf{B}_{pri} = \frac{\mu_0(I_{pri} + I_{sec} - I_{sec})}{2\pi r}\hat{\theta}, \tag{4.22}$$

$$\mathbf{B}_{out} = \frac{\mu_0(I_{pri} + I_{sec} - I_{sec} - I_{pri})}{2\pi r}\hat{\theta} = 0, \tag{4.23}$$

where the indices of the magnetic field represent the areas shown in Figure 4.3. The magnetic field energy, W, is given by

$$W = \frac{1}{2\mu_0}\int_{\text{all space}} \mathbf{B}\cdot\mathbf{B}d^3r \tag{4.24}$$

If we integrate over time from when no current was flowing into the circuit, the magnetic field energy can also be written in terms of the self inductance of the current loop. It is given by

$$W = \frac{1}{2}LI^2. \tag{4.25}$$

Combining equation 4.24 with 4.25 gives the self inductance for the two current paths in this test case, which are

$$L_{pri} = \frac{\mu_0 l}{2\pi}\ln\left[\frac{d}{c}\right] \tag{4.26}$$

$$L_{sec} = \frac{\mu_0 l}{2\pi} \ln \left[\frac{b}{a} \right]. \tag{4.27}$$

This case was tested in the MACH2 code by only implementing magnetic diffusion and not physical processes such as hydrodynamics and radiation. The magnetic field strength and pressure calculated by MACH2 corresponds with analytical calculations, within numerical precision. Also, the calculations of the self inductance were found to be in total agreement with analytical calculations no matter what the initial values of the relative inductances were. The iterative process of the algorithm converges to the correct result for the self inductance of each current loop, which indicates that the circuit routines are stable. The initial values of the inductances only affected the number of cycles needed for convergence during the simulation run. In effect, the algorithm works to ensure conservation of magnetic flux and modifies the self inductance accordingly.

4.3 Initial Conditions

Several initial parameters, such as starting time, plasma column radius, species temperature, amount of initial current flowing through the plasma columns, and the amount of initial cell-to-cell density perturbations are needed for the flux compression simulations. Ideally, separate "cold start"[3]models of the wire initiation phase should be used to determine these parameters. However, for reasons discussed in chapter 2, approximations based on previous experiments and experience are used for the initial parameters.

Both the Al and Ti wire arrays are assumed to have completly vaporized and coalesced into thin annular plasma shells. The Al shell is initially defined to be 1 mm thick at a temperature of 5 eV while the Ti shell is defined initially to be 0.5 mm thick at a temperature of 1.0 eV. Both shell densities are defined with respect to 1D optimized mass calculations. The initial conditions for the Ti shell are, admittedly, troublesome. The Ti wires will vaporize at a much later time than the Al wires due to the inductive current distributions. Since the wires are on the order of one mil in diameter, it is impossible to achieve the resolution required in these simulations for a cold start condition of the wires. Therefore, a thin shell plasma approximation has to be made at the beginning of the simulation for the Ti wire array. Fortunately, this approximation is not unreasonable. During the vaporization of the wires, the Lorentz force is insignificant, and the physics of the vaporization are governed by wire ablation and free-expansion. The Lorentz force on the Ti array only becomes significant later in the simulations long after the array has vaporized.

The equation of state, opacity data, and ionization state used for all materials is obtained using the SESAME unclassified category 1 tables from Los Alamos National Laboratory [84]. Electric and thermal conductivities were obtained using tables generated by Mike Dejasharis at SNL. These tables are based on a modified and improved version[85, 86] of the Lee-More conductivity model[87]. A global floor

[3]Cold start models start with a solid wires at room temperature.

density of 1.0e-4 kg/m^3, was implemented for all materials. Artificial viscosity is not used in any of the simulations to prevent the artificial smoothing of any shock waves that may form in the plasma shells. Although the Hall effect is not implemented in the model, thermally generated magnetic fields are incorporated via the generalized Ohm's law.

4.4 Compuational Grids and Boundary Conditions

In any MHD computation there is a fundamental tradeoff between computational time and grid resolution. For this reason, Lagrangian and ALE gridding schemes are a very attractive choice for these simulations because they minimize the number of cells while concentrating them into regions of interest. Initially, simulations were conducted using an ALE scheme. Plasma interfaces were constrained to be Lagrangian, while vacuum areas were defined as Eulerian. The vacuum grids and block boundaries were allowed to be compressed by the Lagrangian interfaces. Since grid lines move as a function of material velocity, it is extremely difficult to get all of the grid lines moving appropriately in these simulations. Nevertheless, successful runs were completed in the absence of any plasma instabilities. These simulations have very high resolution ($\sim$5 μm) in the region of interest along the stator, but cells become highly elongated, or rectangular. This fact leads to advection errors in the computational domain. Unfortunately, the MACH2 code does not allow for rezoning of grids or the elimination/addition of cells within a block during a simulation.[4] When RT instabilities are added to the simulations, the severe growth rate grossly distorts grid lines and causes numerous problems in grid behavior. An entire simulation with RT instabilities could not be completed.

The results presented in the next chapter all use fixed Eulerian grids. Confidence in advection calculations, as well as elimination of grid behavior problems, far outweighs the penalty of increased computational time for this problem. The baseline Eulerian simulations have 44,800 cells, which corresponds to an average cell resolution of $\sim 175\mu m$. This resolution is sufficient to resolve growth of the most severe RT instability wavelengths[31]. However, this resolution is not sufficient to resolve the turnaround radius of the Al plasma armature. According to 1D simulations, the turnaround radius is approximately 10 to 20 μm away from the stator. In an attempt to resolve this turnaround radius without being penalized in computational time, an order of magnitude more grid resolution; a quadratic grid structure was implemented near the stator. Figure 4.4 shows the grid used in these simulations, which is split into twenty-one separate blocks. Initially, the Al plasma shell is defined in blocks 3, 8, and 12. The Ti plasma shell is defined in block 14. The initial density of the shells is determined from 1D optimized mass calculations and the volume of the initial plasma shell. The initial Al shell density for these simulations is 2.514 kg/m^3; and the initial Ti shell density is 0.8 kg/m^3. Blocks

[4]MACH2 does allow entire blocks of cells to be added and subtracted during a simulation, though.

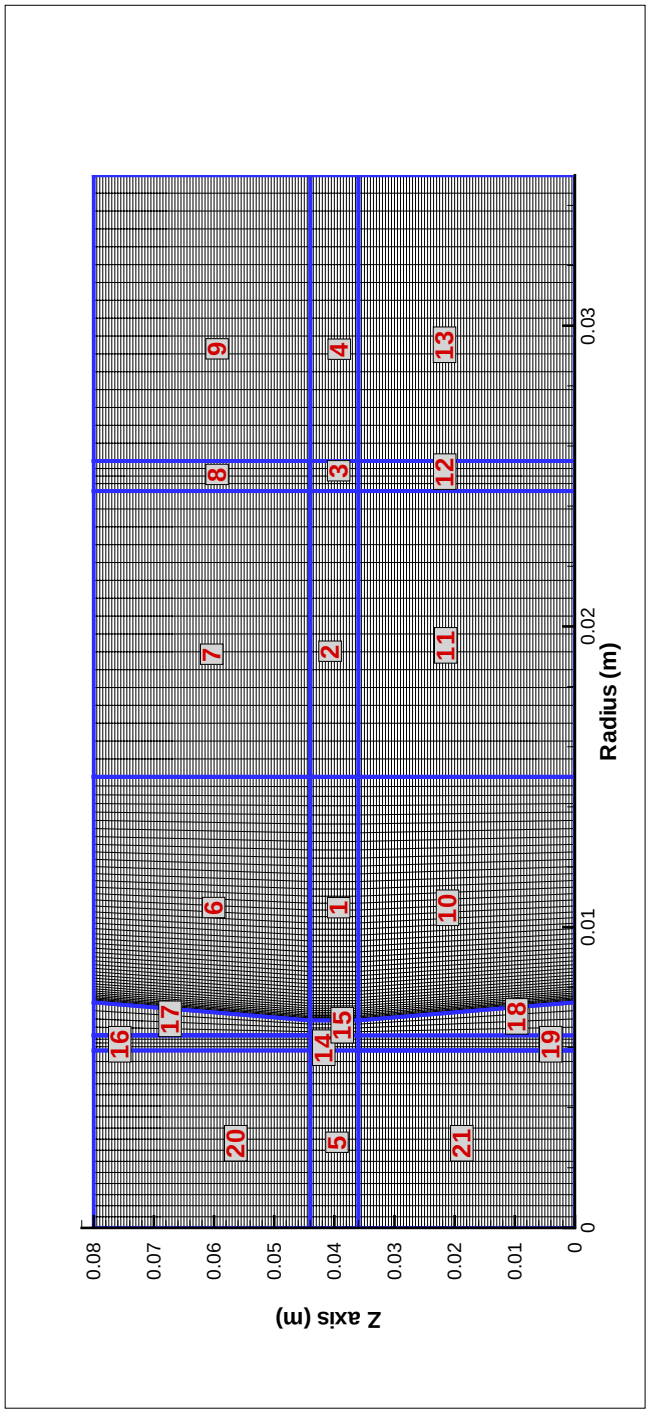

Figure 4.4: Eulerian grid used in the flux compression simulations. Every 4th gridline plotted for readability.

16-21 are part of the conical stator, while the rest of the blocks are initially filled with a pseudo-vacuum fluid.

The pseudo-vacuum fluid is defined to have an artificially fixed electric resistivity of $8.3\times10^{12}\mu\Omega\cdot cm$. This allows the magnetic field to diffuse relatively quickly though the fluid. Larger values would create characteristic diffusion speeds greater than the speed of light and would drive the magnetic diffusion solver unstable. The initial and minimum , or floor, density of fluid is defined to be $1.0\times10^{-3}\ kg/m^3$. Although the floor density could have been lower, it was chosen such the lowest computational times could be achieved without adversely affecting the physics. Smaller values would allow the vacuum fluid to accelerate to such velocities that computational time steps are unreasonably small. A fixed temperature of 1 eV is defined for the fluid, which corresponds to nearly zero hydrodynamic pressure using Al's SESAME derived equation of state. Thermal conductivity in the vacuum material is kept to minimum by choosing an artificially low fixed thermal diffusivity.

Imposed boundary conditions are given in table 4.1. Current is imposed by allowing the magnetic field to diffuse through the vacuum from the computational boundary. The primary circuit path magnetic field boundary conditions are given on side 2 of blocks 13, 14, and 9. The magnetic field quickly diffuses through the highly resistive vacuum material and is slowed upon reaching the Al plasma. The gradient in the magnetic field induces the current in the plasma shell. One drawback of applying current in this method is that current in the Al array is delayed slightly by the small characteristic diffusion time of the vacuum. Boundary conditions for the secondary current path are not quite as evident. The secondary circuit path magnetic field and hydrodynamic boundary conditions are internal and imposed on side 2 of blocks 17 and 18. The inclusion of internal boundary conditions is unfortunate, but necessary, in the geometrical context of the problem to include diffusive losses in the stator.

4.5 Plasma Approximations and Numerical Issues

The magnetic diffusion time scale, τ_{diff}, which is the time required for complete diffusion of an exponentially decaying magnetic field, is defined to be

$$\tau_{\text{diff}} = \mu_o\sigma_{dc}Ł^2 \tag{4.28}$$

$$= \frac{L^2}{\eta_d}, \tag{4.29}$$

where Ł is the characteristic length scale, σ_{dc} is the DC conductivity, and η_d is the electrical diffusivity defined as $\eta_d = 1/\mu_o\sigma_c$. The electric diffusivity of the vacuum is defined to be 1×10^7 m^2/s. It therefore takes 0.13 ns to transport the magnetic field across the entire length of stator, which is 3.6 cm. If the characteristic length is divided by the characteristic diffusion time, a diffusion velocity of 2.8×10^8 m/s is obtained. This velocity, slightly slower than the speed of light, ensures the speed of

Table 4.1: **Boundary conditions used in the flux compression simulations.**

Block[a]	Side[b]	Magnetic (z) BC	Hydro BC	Thermal BC	Radiation BC
4	R	Pri. Current	Wall	Insulator	Conduct
5	L	Axis	Axis	Insulator	Insulator
5	T	Conductor	Wall	Symmetry	Symmetry
5	B	Conductor	Wall	Symmetry	Symmetry
6	T	Conductor	Wall	Insulator	Insulator
7	T	Conductor	Wall	Insulator	Insulator
8	T	Conductor	Wall	Insulator	Insulator
9	T	Conductor	Wall	Insulator	Insulator
9	R	Pri. Current	Wall	Insulator	Conductor
10	B	Conductor	Wall	Insulator	Insulator
11	B	Conductor	Wall	Insulator	Insulator
12	B	Conductor	Wall	Insulator	Insulator
13	B	Conductor	Wall	Insulator	Insulator
13	R	Pri. Current	Wall	Insulator	Conduct
14	T	Conductor	Wall	Symmetry	Symmetry
14	B	Conductor	Wall	Symmetry	Symmetry
15	T	Conductor	Wall	Symmetry	Symmetry
15	B	Conductor	Wall	Symmetry	Symmetry
16	T	Conductor	Wall	Insulator	Insulator
17	R	Sec. Current	Wall	Symmetry	Insulator
17	T	Conductor	Wall	Insulator	Insulator
18	R	Sec. Current	Wall	Symmetry	Insulator
18	B	Conductor	Wall	Insulator	Insulator
19	B	Conductor	Wall	Insulator	Insulator
20	T	Conductor	Wall	Insulator	Insulator
21	B	Conductor	Wall	Insulator	Insulator

[a] Internal block boundaries not shown in table are shared with neighboring block.

[b] R, L, T, and B represent right, left, top and bottom sides, respectively.

magnetic diffusion is faster than the characteristic velocities of the Al or Ti plasmas and keeps the magnetic field solver stable.

Plasma regions which possess relatively high conductivity can cause numerical difficulties with finite diffusion times. The magnetic Reynolds number, R_M, is defined as

$$R_M = \mu_o \sigma \text{Ł} V \tag{4.30}$$

$$= \frac{\text{Ł} V}{\eta_d}, \tag{4.31}$$

where V is the characteristic velocity of the fluid. $R_M \ll 1$ represent low conductivity flows. The psuedovacuum material is one such example where the magnetic field is transported primarily through diffusion. Plasma regions with $R_M \gg 1$ represent high conductivity flows and are such that the magnetic field is strongly coupled to the plasma. Magnetic field transport, in this case, is primarily through advection. From equation 4.28, it can be seen that magnetic diffusion is relatively slow in plasma flows with $R_M \gg 1$. Numerical error in the magnetic field can occur if current rises faster than the magnetic field can diffuse into the plasma. Surface currents will be accurate, but the magnetic field may not be sufficiently diffused into the plasma skin depth. This is similar to the problem of getting magnetic field diffused at the start of an MHD problem.

4.5.1 Quasi-Neutrality

Quasi-neutrality is a fundamental assumption for these MHD simulations. To test the validity of this assumption, a quick check of the Debye length and plasma frequency can be made. Figure 4.5 shows several contour plots that give typical plasma densities, frequencies, and Debye length during the run-in phase of the simulations. The Debye length is on the order of 1 to 10 nm which is four to five orders of magnitude smaller than the cell resolutions used for the simulations. Electron and ion plasma frequencies are in the 10-100 terrahertz region, which yield plasma periods in the femtosecond range. Since time steps range from 1 ns to 10 ps, characteristic plasma oscillations will not affect the quasi-neutrality assumption.

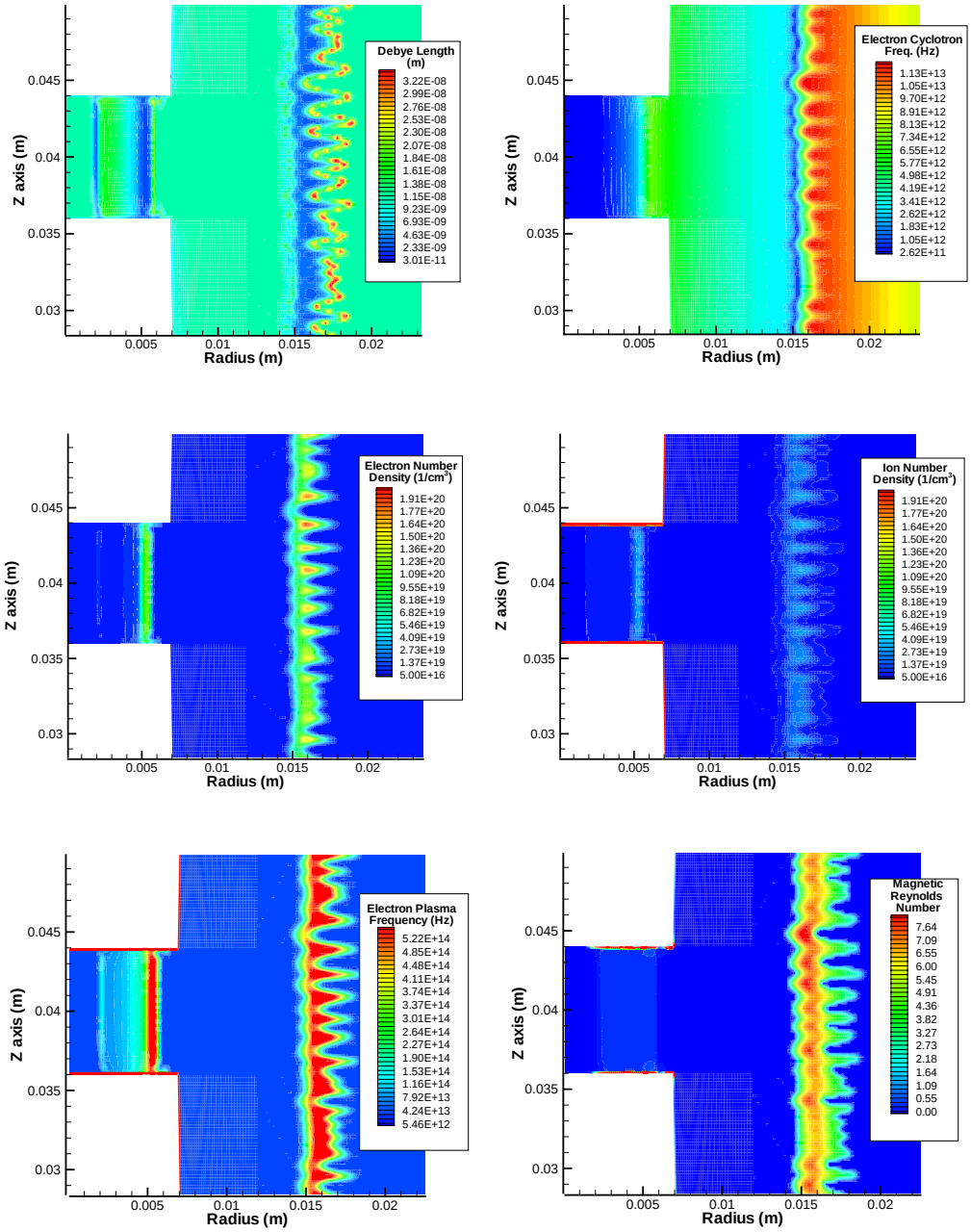

Figure 4.5: Typical plasma number density, electron cyclotron frequency, Debye length, magnetic Reynolds number, and electron plasma frequency for the flux compression simulations.

Chapter 5

Decade Quad Flux Compression Simulations

In this chapter, the significant findings from the flux compression simulations using the MACH2 code will be presented. First, a brief description of a one-dimensional model will be presented in section 5.1. Section 5.2 will describe the results of two-dimensional simulations with ideal plasma shells, which correspond to the absence of plasma instabilities, and compares them with a one-dimensional calculation. Section 5.3 will incorporate Rayleigh Taylor instabilities in the model and show their effects on flux compression. Flux compression results implementing MACH2's material interface tracking routines are presented in section 5.4. Section 5.5 discusses results of various conical stator angles. Finally, results of incorporating MACH2's radiation models will be discussed in section 5.6. Numerical issues including grid resolution, material interface tracking and reconstruction, and various plasma assumptions will also be discussed throughout the chapter.

5.1 1D Model

A one-dimensional flux compression model, developed by John Goyer[1] in a commercially available program called Microcap, was used to compare results from the 2D flux compression model discussed in section 4.2. This model is based on the thin shell model presented in section 2.2. Since magnetic pressure exists on both sides of the Al armature, the equation of radial motion for the armature becomes

$$\mu \frac{d^2 r_1}{dt^2} = -\frac{\mu_o \left[I_{pri}^2(t) - I_{sec}^2(t) \right]}{4\pi r_1}, \tag{5.1}$$

where r_1 is the radial position of the Al plasma armature, I_{pri} and I_{sec} are the primary and secondary circuit path currents respectively. The equation of motion

[1] John Goyer's model was modified to incorporate the same external circuit model as used in the 2D flux compression model.

for the inner Ti plasma shell is simply

$$\mu \frac{d^2 r_2}{dt^2} = -\frac{\mu_o I_{sec}^2(t)}{4\pi r_2}. \tag{5.2}$$

The model solves these equations in an iterative process by integrating these equations to obtain values for the radii of the plasma shells. Then, the inductance in the circuit paths is calculated geometrically depending on calculated radius values. Finally, a nonlinear set of circuit equations is solved to determine current values.

5.2 Ideal Perfect Plasma Shells

As a baseline for the comparison of 1D and 2D results, several simulations of perfectly uniform, i.e. no initial density perturbations, plasma shells of Al and Ti were defined as initial conditions for MACH2 2D simulations. In these simulations, one does not expect to see plasma instabilities unless the plasma shells are perturbed during the simulation by some physical mechanism.

Primary and secondary path inductance calculations are shown in Figure 5.1. The calculated inductance for the primary circuit path starts approximately ten percent lower than 1D Microcap simulations and progressively increases relative to the 1D results throughout the implosion. The secondary circuit path inductance is on four to six percent lower than 1D results initially, but then crosses over and becomes greater than 1D results late in the implosion. The slight discrepancy is explained by the assumption of infinitely thin plasma shells in the 1D case and finite thickness plasma shells in the 2D case. The difference in shell thickness will yield slightly different inductance calculations. The initial inductances can also be varied by altering the initial width and/or position of the plasma shells. The steeper gradient initially observed in the inductance of the secondary current path is a result of plasma shell expansion. All other things being equal, these results suggest primary and secondary current profiles in the 2D simulations will be slightly higher than 1D calculations during most of the run-in phase.

The parametric evolution of the circuit path inductance is also influenced by equation of state parameters. Expansion and contraction of the plasma shells, which can be seen by the radius of each plasma/vacuum interface shown in Figure 5.2, will change the circuit path lengths. Due to the nature by which each material interface radius is calculated in MACH2, the radius data exhibit slight discontinuities as plasma enters a new row of cells.[2] Additionally, occasional errors occur in the radius calculations when high number densities exist in vacuum regions. These errors are easily identifiable in Figure 5.2 as the large peaks in the radius. The specfic cause of these errors has not been determined. The error is not stochastic and occur predictably in simulations with various grid resolutions. Despite these

[2] Lagrangian simulations in which grid lines move precisely with the plasma shells demonstrate smooth radius profiles. Although time consuming, smoother radius profiles in Eulerian simulations can also be extracted from contour plots.

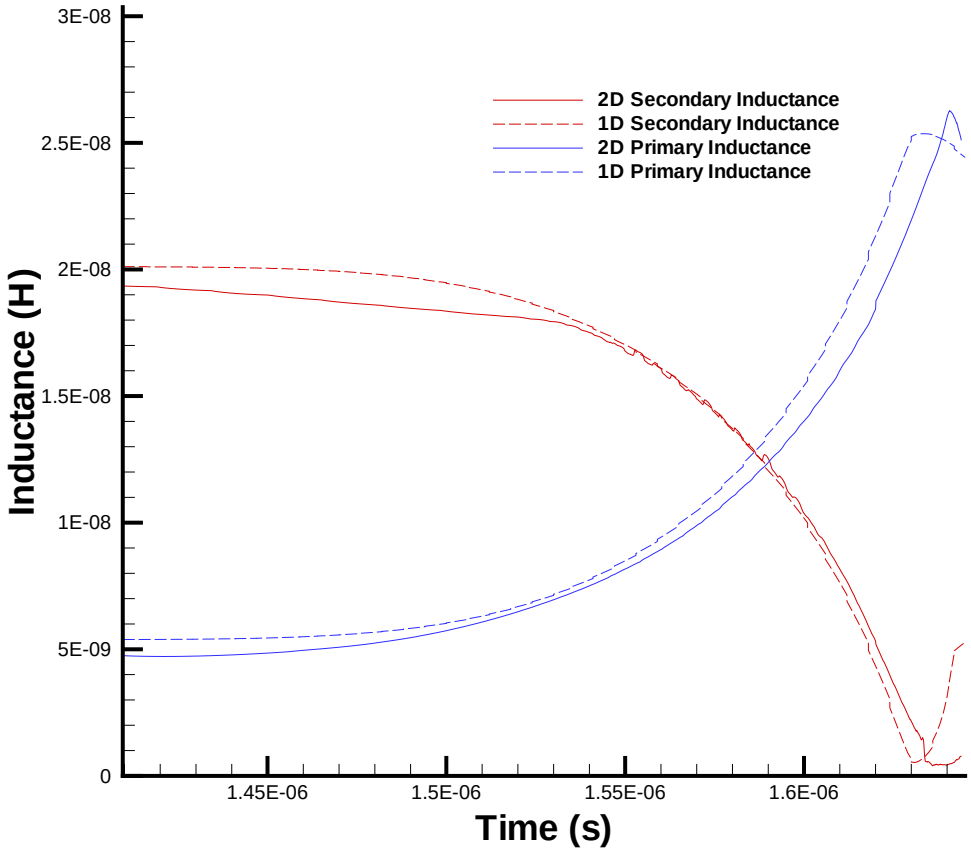

Figure 5.1: Inductance in primary and secondary current paths as a function of time. The 2D calculations are for the ideal case of no plasma instabilities.

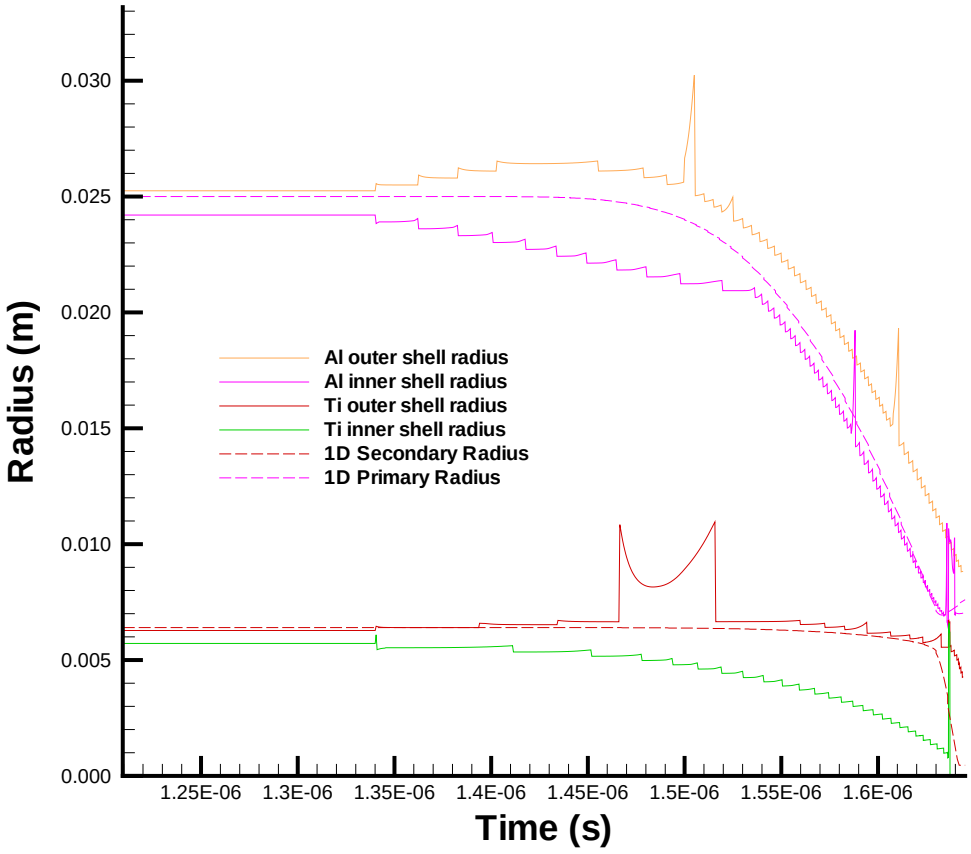

Figure 5.2: Radius of plasma shells as a function of time. The 2D calculations are for the ideal case of no plasma instabilities.

problems, expansion and contraction of the plasma shells can still be seen. The Al plasma liner initially expands, cools, and finally recompresses to a 1.0 mm thick shell, similar to the initial shell width, plasma shell for the remainder of the run-in phase. However, it should also be noted that the initial expansion of the plasma shell is a strong function of the initial temperature.

Current profiles, shown in Figure 5.3, suggest reasonable flux compression has been achieved. Peak current in the secondary current path exceeds twenty megamperes. Overall, excellent agreement with 1D results is evident by similar waveforms and timing characteristics. Clearly, 2D MHD effects in the absence of plasma instabilities are not significantly altering the implosion dynamics. Of particular note is that the peak flux compression is 4.5 ns later than in 1D calculations. The time difference is a result of the difference in plasma armature velocity calculations. The velocity discrepancy results from slightly different different magnetic pressures, which are proportional to the current distributions, on both sides of the plasma armature. Magnetic pressure calculations in the 2D simulations are also lower as a result of magnetic diffusion into the plasma armature.

Also as a result of the current distributions, the magnetic flux is approximately 4 to 5% lower for the primary current path in the 2D case throughout the simulation. Magnetic flux in the secondary path is approximately 10% greater in the 2D case, except during peak compression. After the crowbar switch engages, the magnetic flux in the secondary circuit path remains constant in the 1D simulations, since resistive impedance is not considered. In other words, magnetic flux is completely conserved in the crowbar circuit since no heat can be dissipated during peak compression. In the 2D simulations, magnetic diffusion induces a resistive impedance, which results in a slight loss in magnetic flux as shown in Figure 5.3. The finite conductivity of the stator and Ti plasma causes a slow magnetic flux loss through diffusion. Despite this, magnetic flux in the secondary current path does not significantly limit flux compression. The diffused magnetic flux does, however, allow energy to be dissipated via ohmic heating in the plasma. Ohmic heating rates, shown in Figure 5.4 , rise steadily throughout the implosion, but significantly increase during peak flux compression. Ohmic heating also creates resistive impedance as shown in Figure 5.5 . The resistive impedance must be accounted for in the feedback impedance for the flux compression circuit in order to conserve energy. After the crowbar has engaged, resistive impedance has the greatest effect as the secondary current begins to rise. Although the ohmic heating rate rapidly rises during peak compression, magnetic flux loss does not significantly affect the flux compression. Therefore, it can be concluded that the Ti and Al plasmas possess sufficient conductivity to achieve reasonable flux compression.

It is important to realize that current and voltage profiles shown in Figures 5.3 and 5.5 are calculated values from the flux compression circuit model and only serve as input for simulation boundary conditions. 2D contour plots must be used to investigate any 2D MHD effects, such as non-uniform current distribution. An enormous amount of contour data is produced for each simulation. Only contour plots of relevant or interesting physics are presented in this chapter. Complete contour

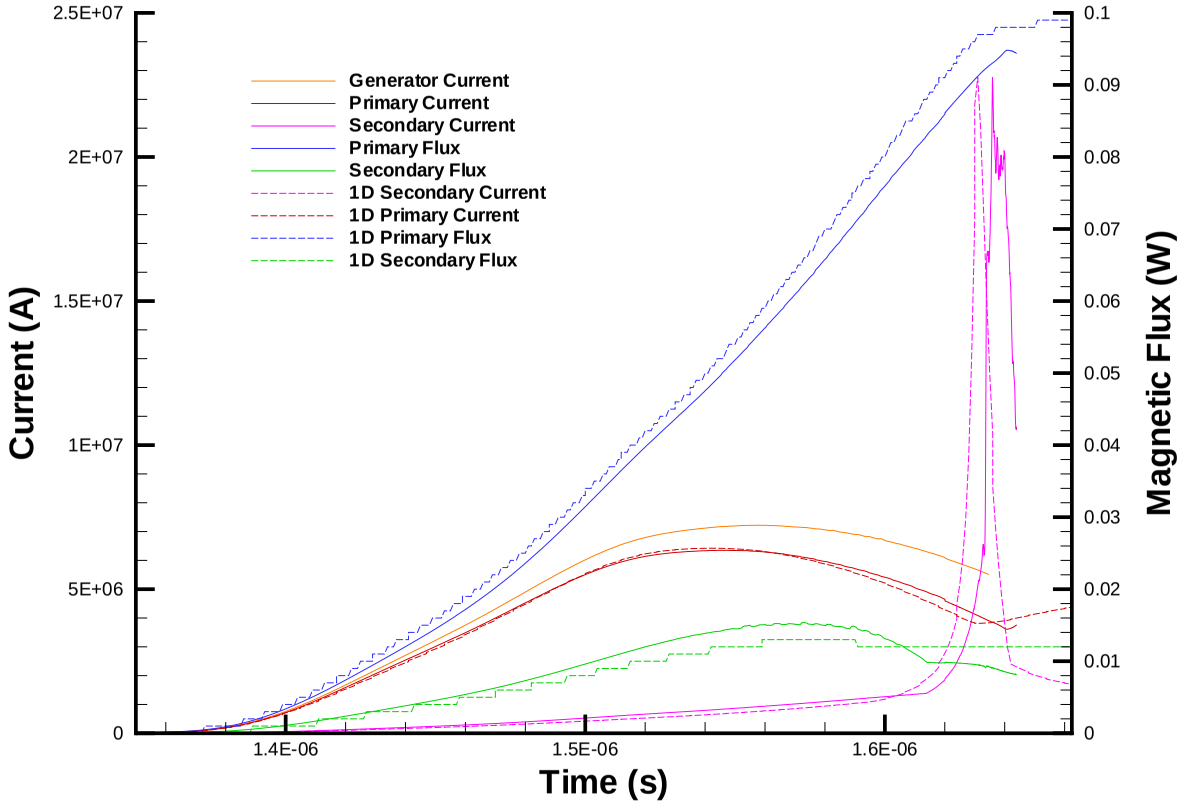

Figure 5.3: Current and magnetic flux profiles as a function of time for ideal case of no plasma instabilities. The crowbar switch engages at t=1.615 μs. The decrease in magnetic flux in the secondary circuit path after the crowbar circuit engages is a result of resistive impedance losses.

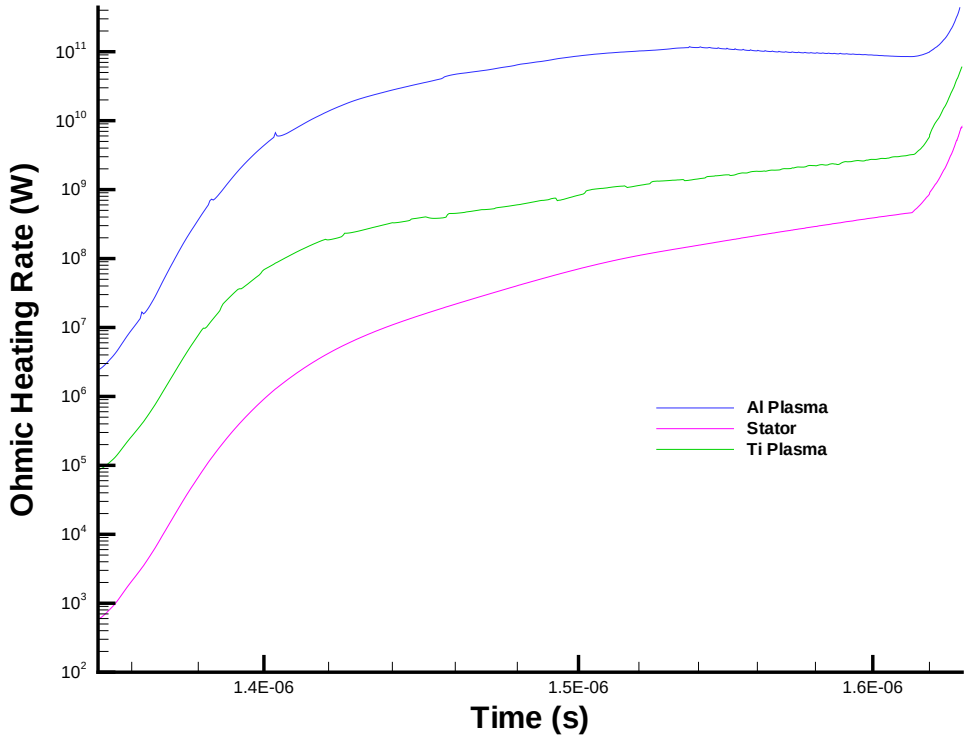

Figure 5.4: Ohmic heating rates for each material during the flux compression simulation. Ohmic heating rates rise steadily throughout the simulation, but rise dramatically after the crowbar engages t=1.615 μs and continues throughout compression of the secondary current flux.

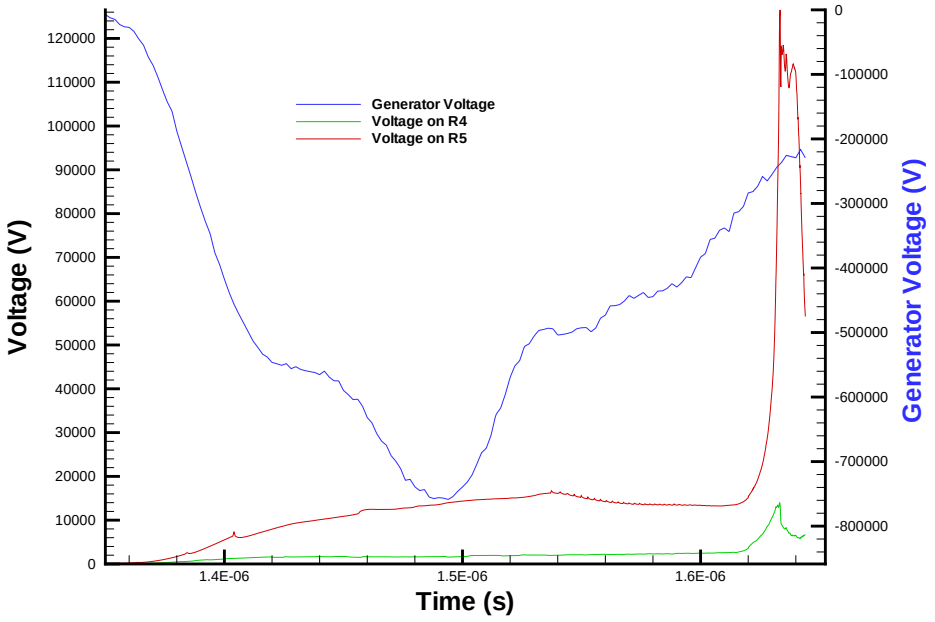

Figure 5.5: DQ voltage profile and resistive impedance voltage drops for primary and secondary current loops.

histories of several important physics parameters for several different simulations can be found in Appendix A.

Figures 5.6, 5.7, 5.8, and 5.9, show a magnified view of several important plasma contour variables at late run-in times. An unexpected 2D phenomena presents itself in the Ti plasma. Both ends of the Ti plasma shell exhibit reduced plasma densities, higher temperatures, and greater joule heating at the stator electrode contact. As a result, the ends of the Ti shell are accelerated ahead of the bulk mass of the plasma shell. These plasma instabilities are induced by the nonuniform magnetic pressure distribution shown in Figure 5.7, which perturb the ends of the Ti plasma shell. Theoretically, electrode interaction should cause the ends of the plasma column to lag rather than accelerate ahead of the bulk plasma. Several mechanisms can account for lag, including friction between the plasma and the electrode, heat flux to the cold electrode, which creates axial variations in temperature, mass influx from the electrode, and frozen magnetic field lines in the stator and Ti plasma. All of these effects should cause the ends of the plasma column to lag behind the bulk plasma.

This issue is resolved by examining the dynamics of the titanium plasma earlier in the implosion, shown in Figures 5.10 and 5.11. Initially, the ends of the Ti plasma column do lag behind the bulk mass as expected. As the Ti plasma implodes, the magnetic field lines must be transferred through the skin layer of the conducting stator, which slightly distorts the magnetic field. The distortion of the magnetic field causes the ends of the plasma column to lag, but at the same time, density decreases in this region.

As the density decreases, the nonuniformity of the magnetic pressure builds and the region experiences greater and greater axial acceleration. At approximately t=1.630 μs, the magnetic pressure shown in Figure 5.7 becomes stronger in the Ti low density regions than behind the bulk Ti plasma. At this point, the low density regions have formed titanium plasma spikes that exhibit rapid nonlinear growth until they stagnate on axis. One consequence of the Ti plasma spikes is the increased circuit path length and inductance in the secondary current path. The spike stagnation is two to three nanoseconds ahead of the bulk Ti mass stagnation, which creates high density localized hot spots, or micropinches, and which can be seen in Figure 5.8 as large electron temperature gradients.

The slice plots on the right side of Figure 5.6 can also be used to analyze magnetic diffusion into the Ti and Al plasmas. Early in the implosion, the rB_θ gradient[3] is nearly linear throughout the Al plasma. As the implosion progresses and the magnetic field builds on the stator side of the Al plasma, the rB_θ gradient becomes steeper and steeper on the same side. The magnetic field created by the secondary current path diffuses further into the Al plasma. This shifts the zero point of the magnetic field away from the axis in the Al plasma. As for the Ti plasma, magnetic diffusion is similar to standard Z-pinch implosions. Note however, that the shell thickness is nearly the same as the skin depth for the bulk Ti; the skin depth is

[3] rB_θ contours can be interpreted as current streamlines.

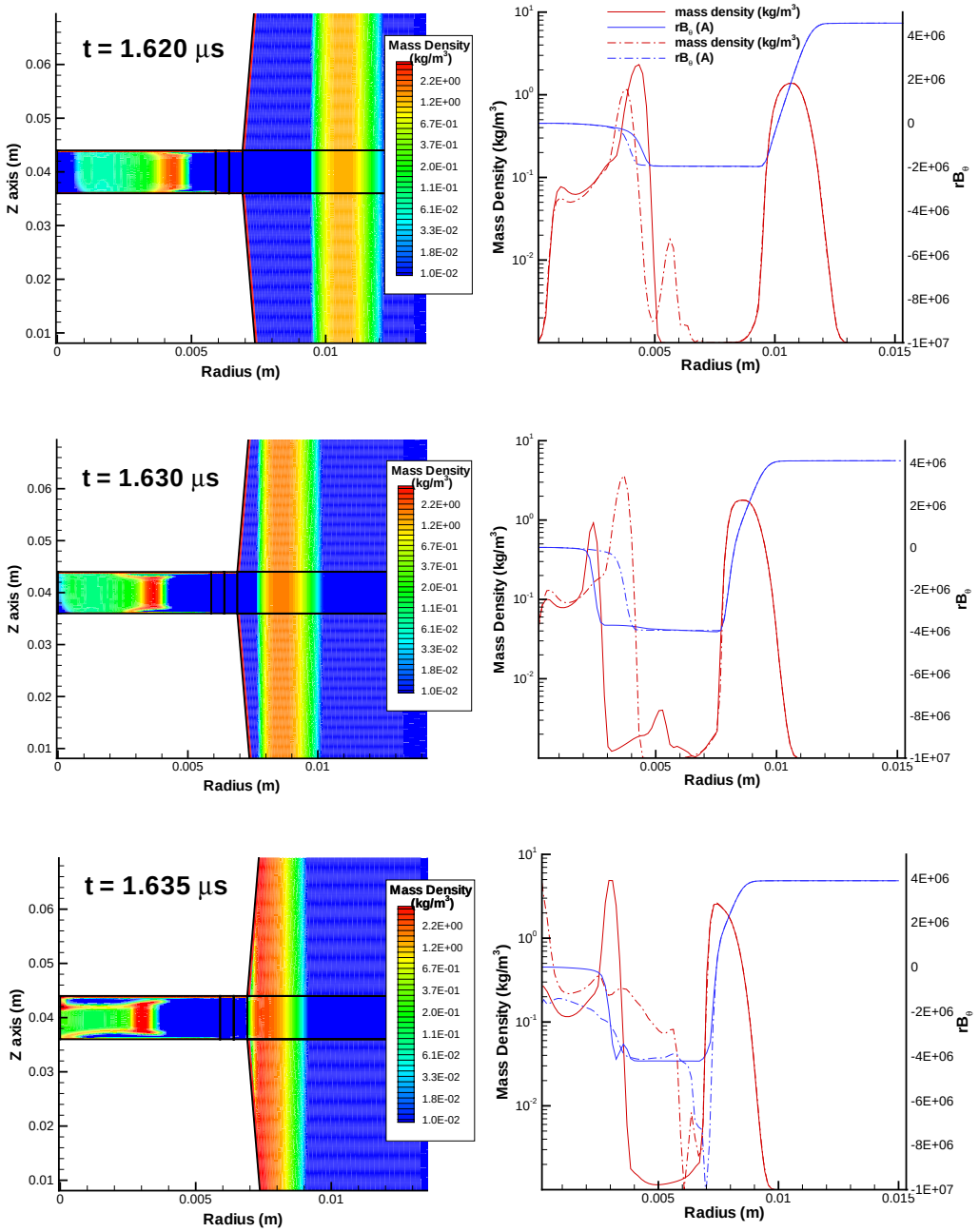

Figure 5.6: Plasma mass density evolution late in the run-in phase. Plots on the right hand side are data slices taken from the plot on the left hand side. The solid curves in the slice plots are from a slice taken at Z=0.04 cm and the dotted dashed curves are from a slice taken very near one end of the conical stator at Z=0.044 cm. rB_θ contours can be interpreted as current streamlines.

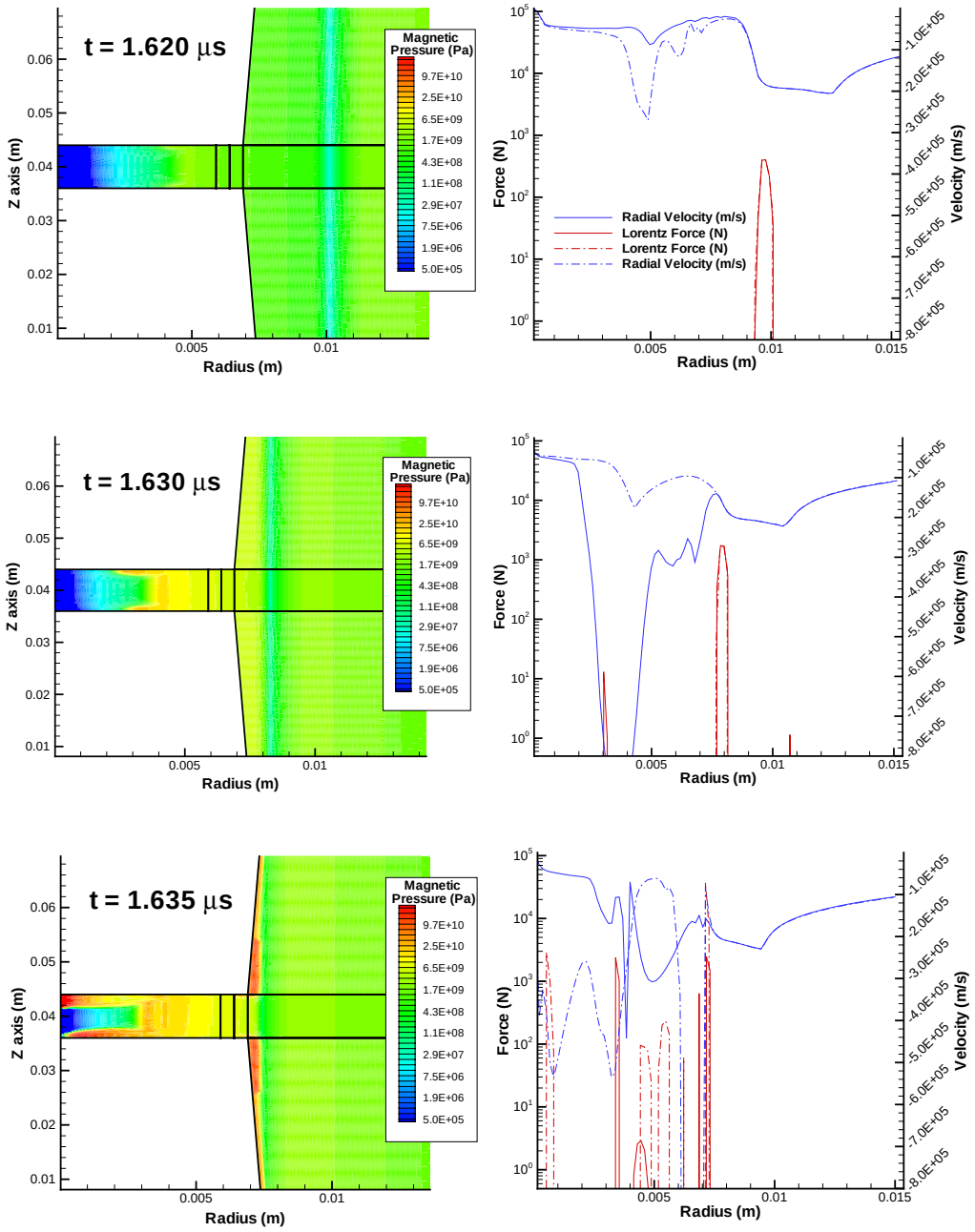

Figure 5.7: Magnetic pressure evolution late in the run-in phase. Plots on the right hand side are data slices taken from the plot on the left hand side. The solid curves in the slice plots are from a slice taken at Z=0.04 cm and the dotted dashed curves are from a slice taken very near one end of the conical stator at Z=0.044 cm.

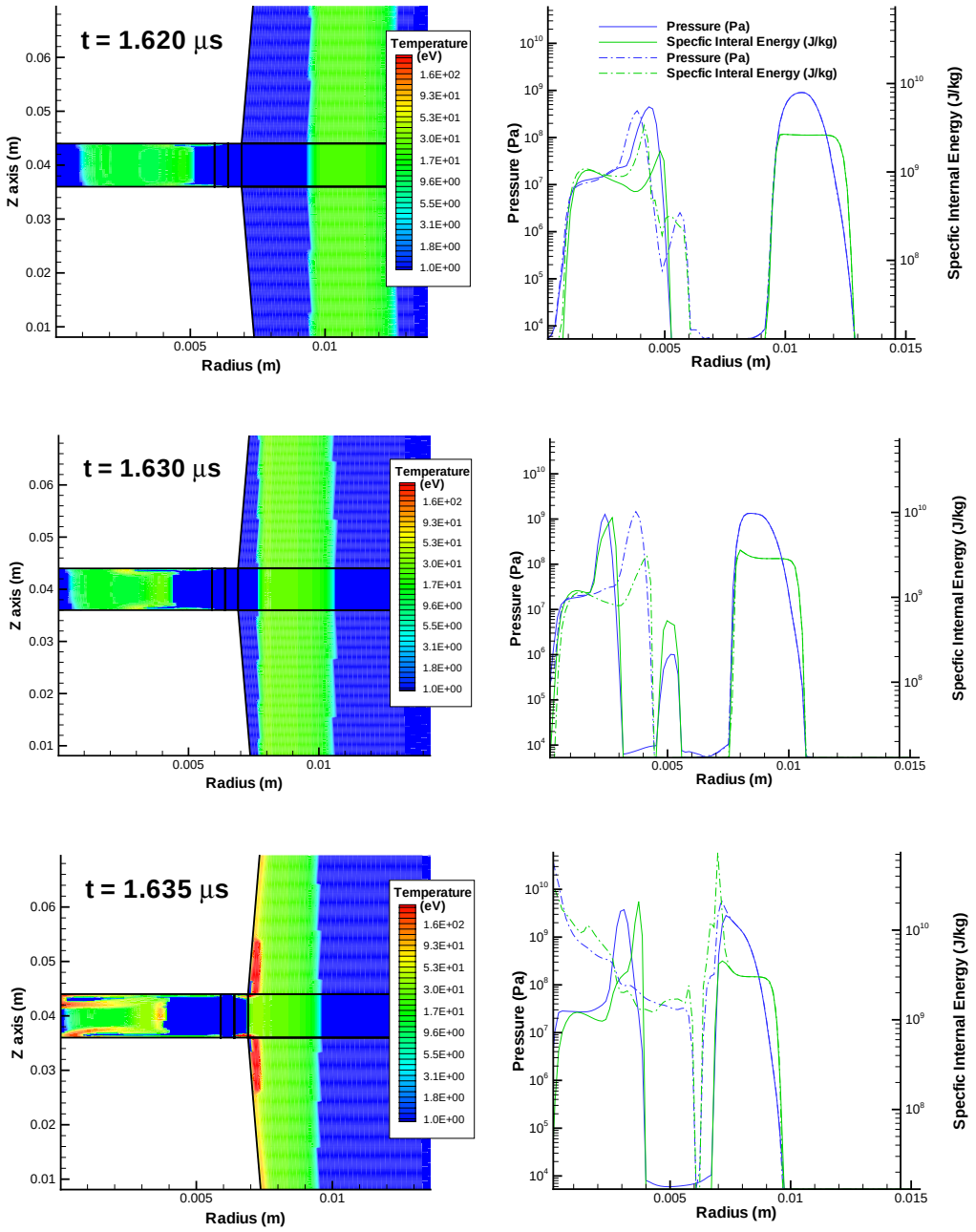

Figure 5.8: Plasma temperature evolution late in the run-in phase. Plots on the right hand side are data slices taken from the plot on the left hand side. The solid curves in the slice plots are from a slice taken at Z=0.04 cm and the dotted dashed curves are from a slice taken very near one end of the conical stator at Z=0.044 cm.

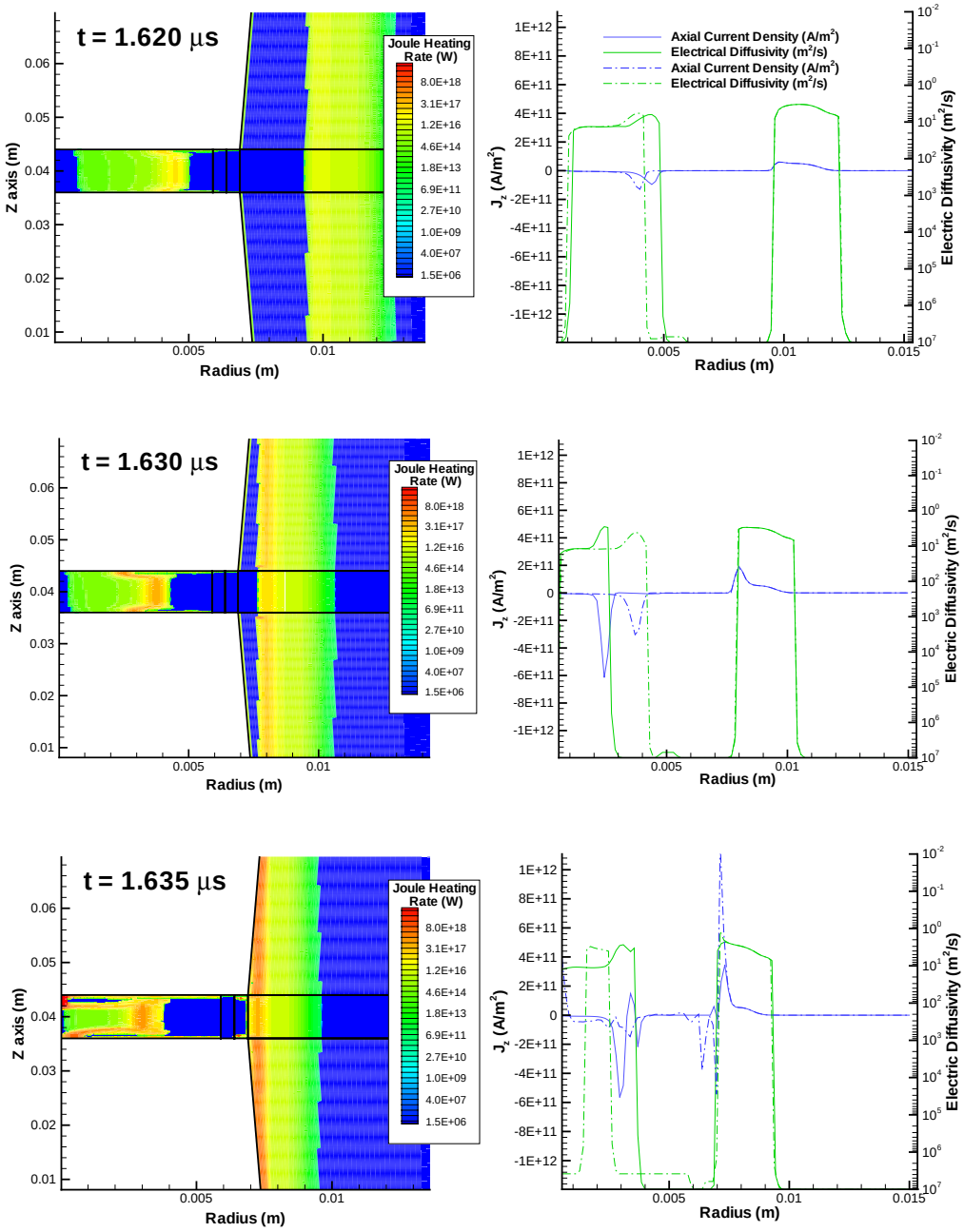

Figure 5.9: Joule heating late in the run-in phase. Plots on the right hand side are data slices taken from the plot on the left hand side. The solid curves in the slice plots are from a slice taken at Z=0.04 cm and the dotted dashed curves are from a slice taken very near one end of the conical stator at Z=0.044 cm.

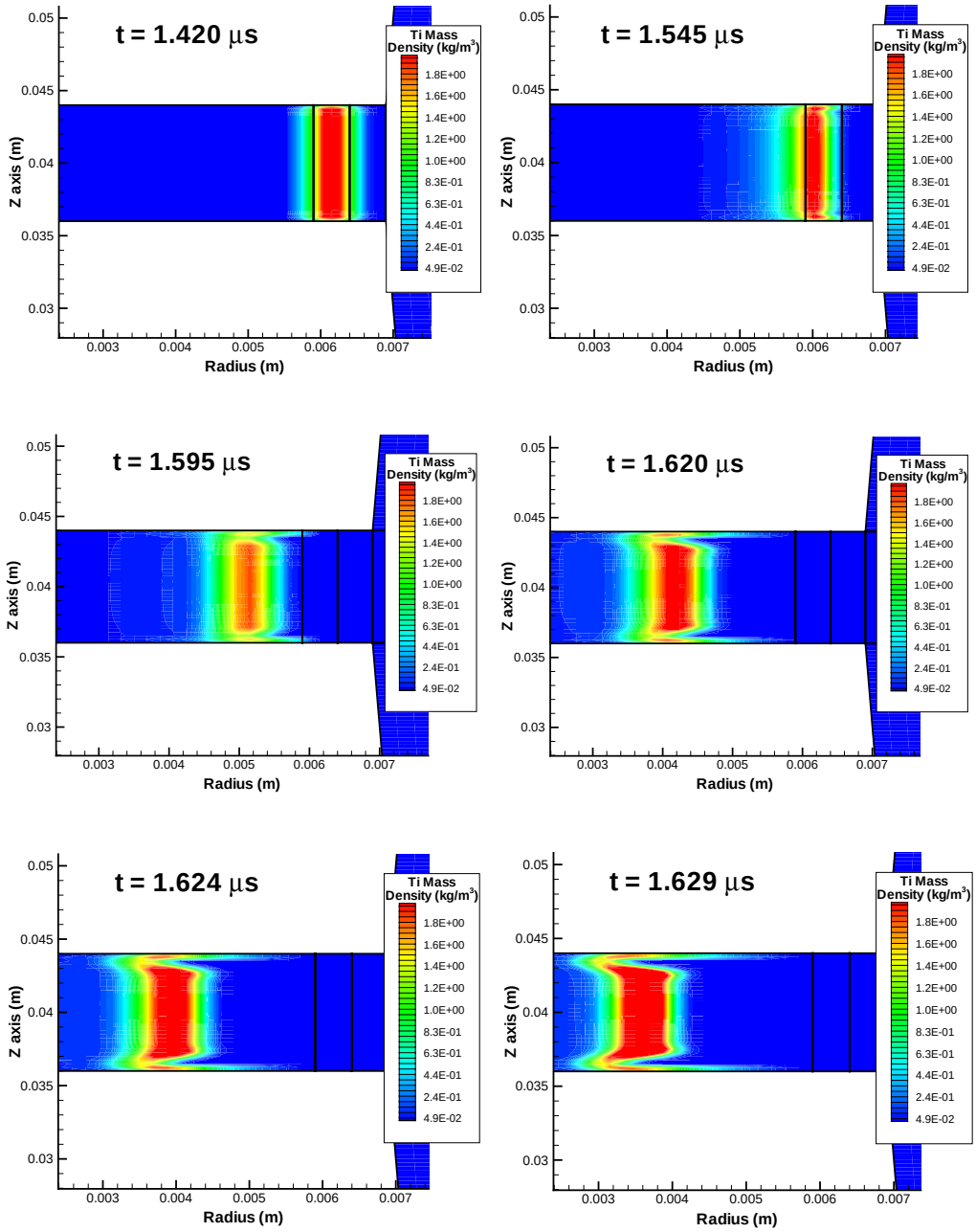

Figure 5.10: Ti plasma mass density evolution during the run-in phase.

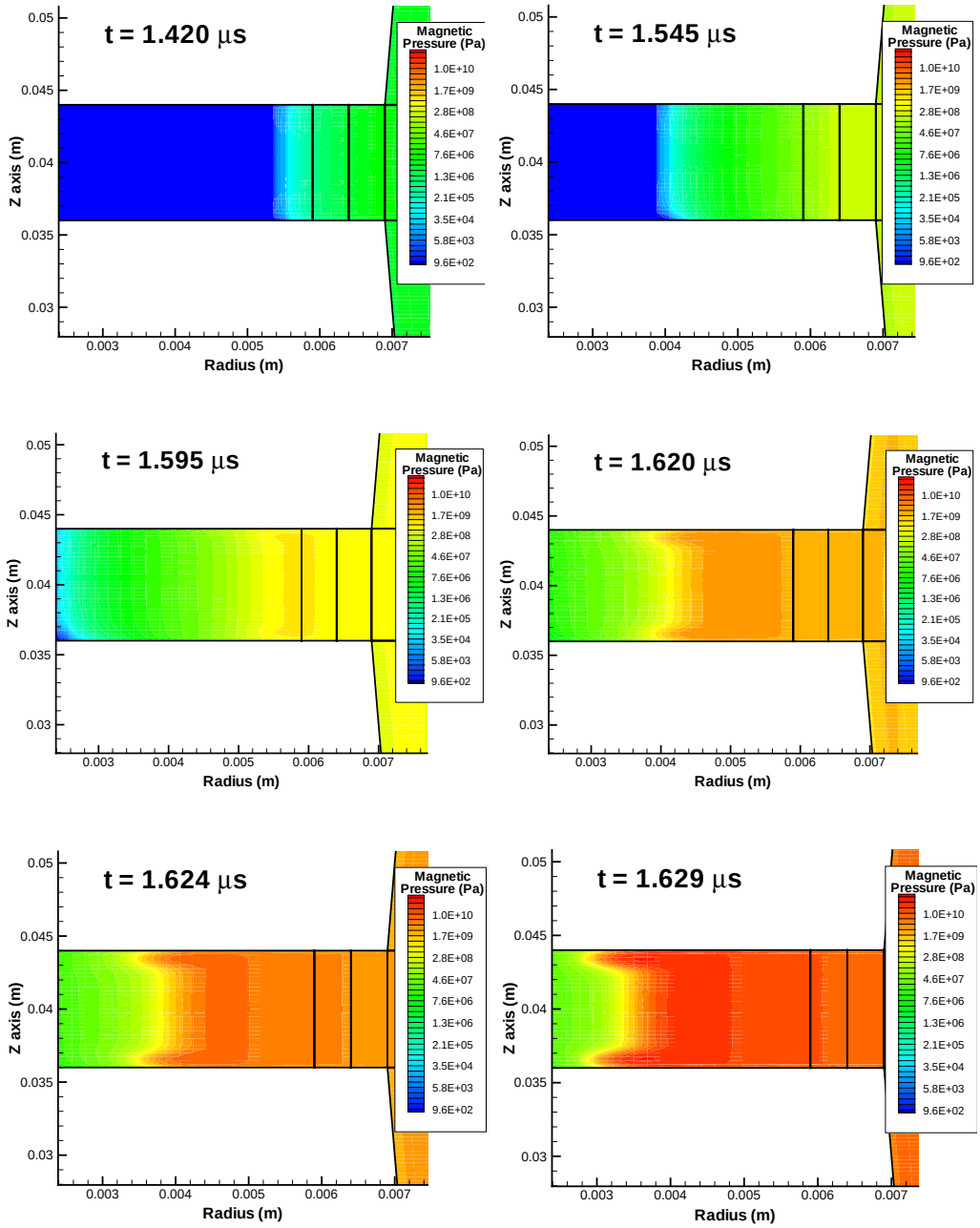

Figure 5.11: Magnetic pressure on Ti plasma during the run-in phase.

larger than the thickness of the plasma spikes. Consequently, there is some magnetic field penetration into the vacuum region.

The rB_θ curves in Figure 5.6 also suggest that maximum current flow in the Ti array peaks around 4 MA. This is significantly lower than the calculated current from the circuit solver in the secondary circuit path, shown in Figure 5.3, that is input into the boundary conditions. Figure 5.12 shows a magnified view of current densities at one inner end of the stator during peak compression. Just before the armature turnaround, the magnetic field has diffused sufficiently into the armature to allow current flow directly between the inner stator tip and the plasma armature. This effectively acts as a current shunt and does not allow the maximum current to flow through the Ti plasma. In order to check this observed phenomena for numerical effects, a simulation was also completed using an order of magnitude more grid resolution along the stator. The results of the simulation can be seen in Figure 5.13. The same current switching process is still observed and begins when the Al armature is approximately 20 to 30 μm away from the stator. Finite magnetic diffusion time does not numerically induce the observed current switching. The computational time step in this simulation was fixed at 10 ps, which allows magnetic field to diffuse 2.8 mm in vacuum regions during one time step. The diffusion speed of 2.8×10^8 m/s ensures sufficient diffusion along the stator as the peak Al implosion velocity is on the order of 1.5×10^5 m/s and enters a new radial cell (10 μm in width) every 7 time steps.

5.2.1 Al Armature Turnaround

1D simulations show a clearly defined Al armature "turnaround" radius approximately 10 μm away from the inner tips of the conical stator. This radius is the point at which magnetic pressure on the inner wall of the armature has become strong enough to begin accelerating the armature away from the z-axis. In 2D simulations, the armature turnaround is not as clearly defined. Since the Al shell thickness is much greater than the skin depth, magnetic pressure decelerates the inner wall of the Al plasma while the outer wall still implodes towards the axis. This causes the Al armature shell to be compressed as it approaches the stator, which is shown in Figure 5.6.

At the moment of peak flux compression, shown in Figure 5.14, the Al plasma near the central ends of the stator is the first to exhibit velocity turnaround. This plasma region now accelerates away from the z-axis. At this point the bulk of the Ti mass is still 3 mm away from the axis with an implosion velocity near 17 cm/μs. The central part of the Al plasma stator behaves quite differently; it continues to implode towards the z-axis. The current distribution in the secondary circuit path has prevented magnetic pressure to reach levels such that enough force is exerted on the Al plasma to produce a turnaround in this region.

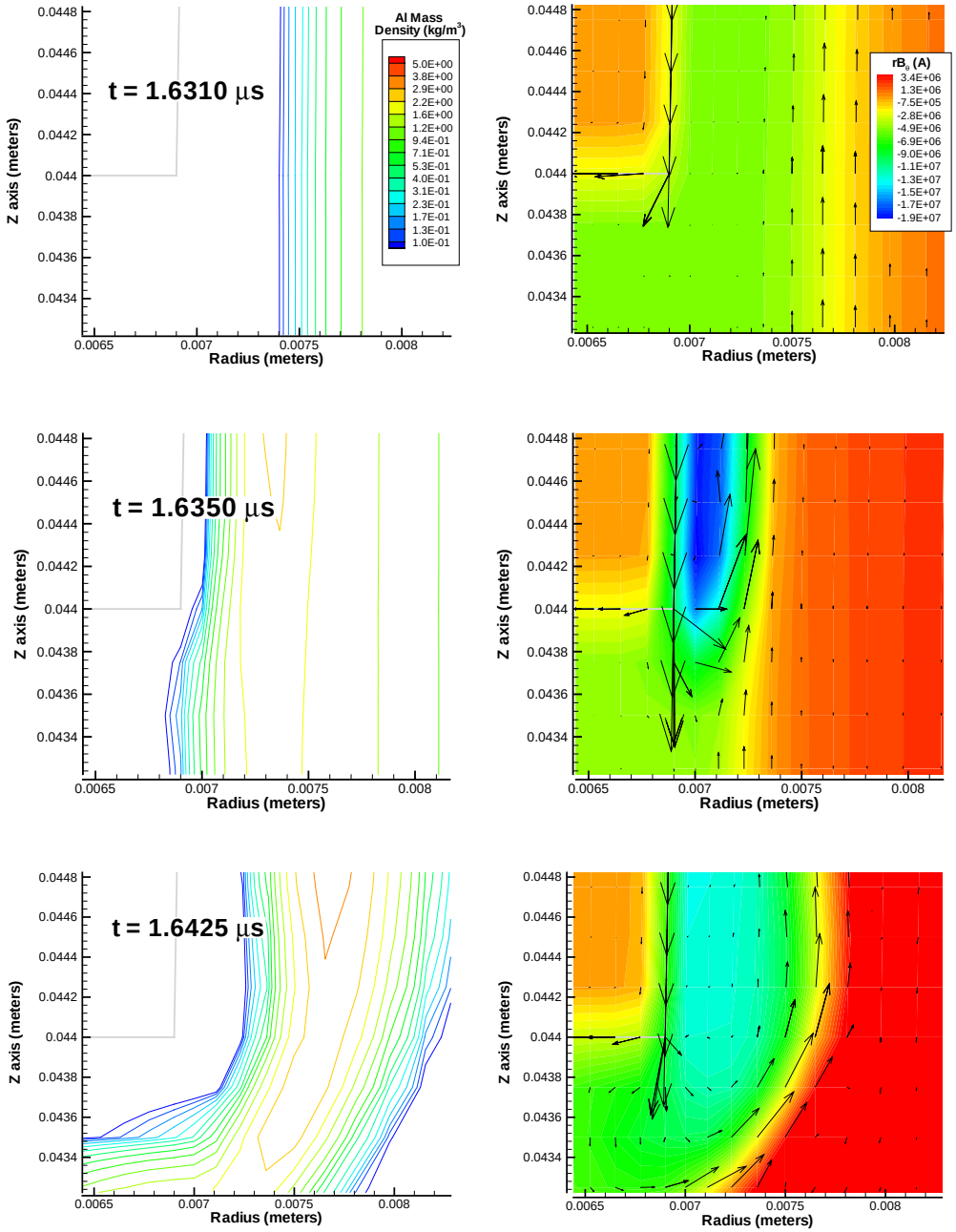

Figure 5.12: Current switching at peak compression at inner ends of the conical stator. Plasma mass density contours are shown in the plots on the left hand side and rB_θ contours and current vectors are shown on the right hand side.

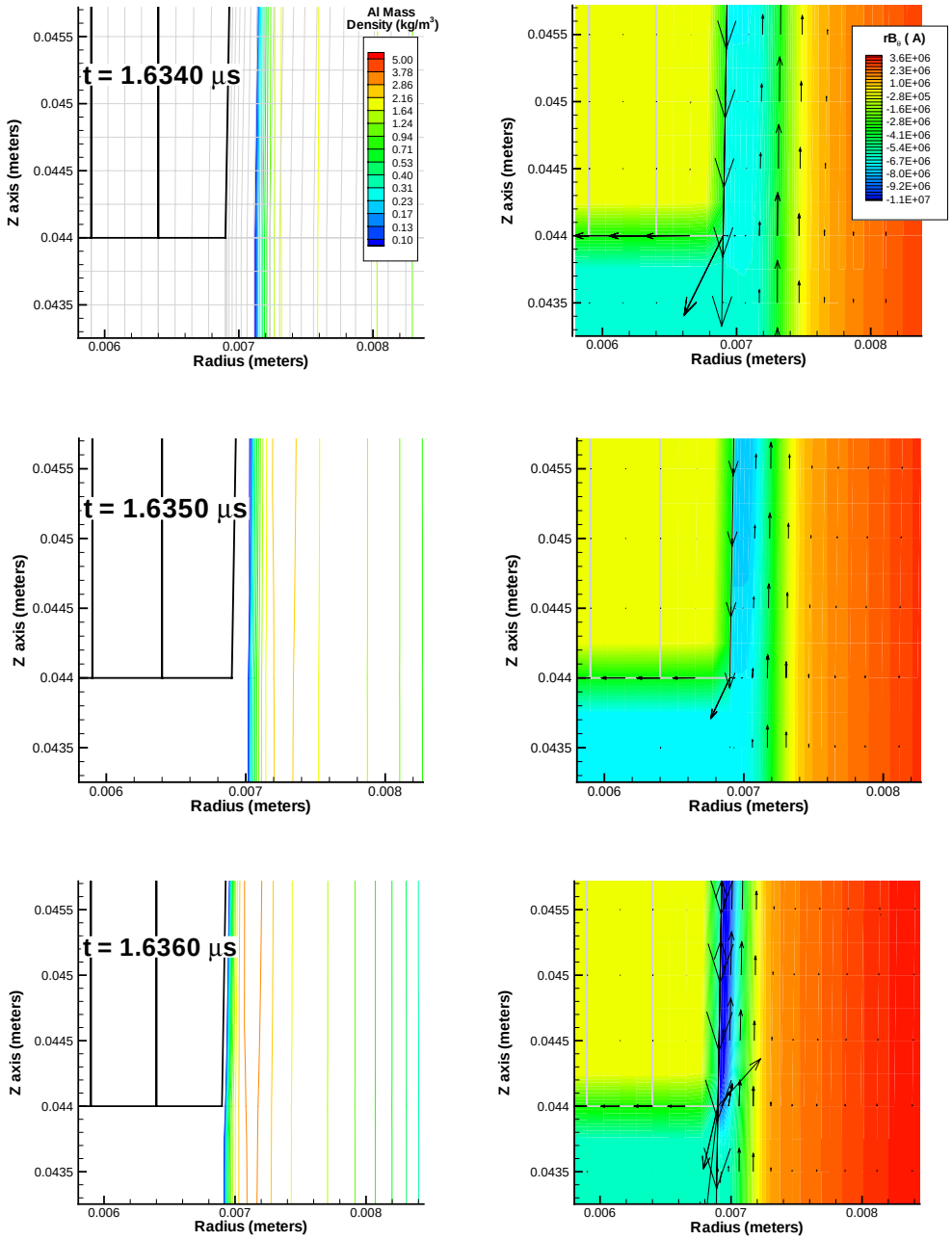

Figure 5.13: Numerical check of current switching at peak compression at inner ends of the conical stator. Grid resolution has been increased by an order of magnitude along the stator. Plasma mass density contours are shown in the plots on the left hand side and rB_θ contours and current vectors are shown on the right hand side.

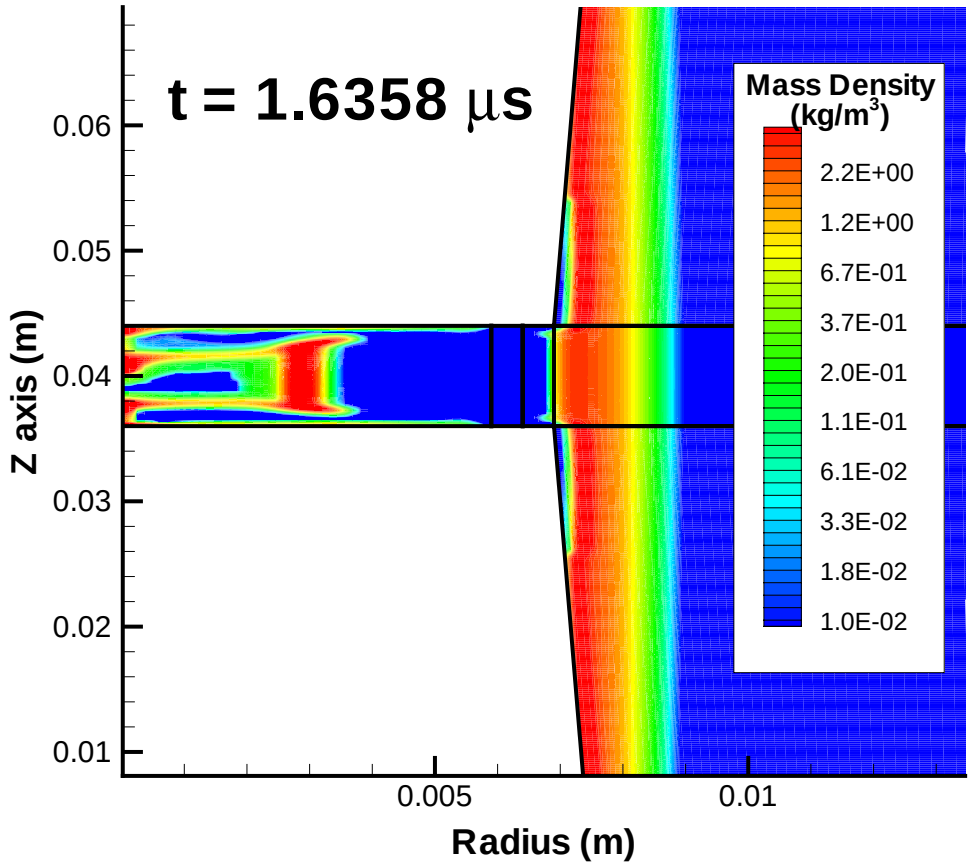

Figure 5.14: Beginning of Al plasma turnaround shown by plasma density contours late in the run-in phase.

5.2.2 Energy Conservation

A good check of any MHD simulation is whether or not energy is conserved. Specifically, energy conservation in this case refers to energy extracted by the external circuit for the simulation versus the sum of the total magnetic, kinetic, radiation, electron and ion internal energies in the computational domain. Some energy will be lost through various numerical floors and caps set in the simulation. However, this will only have a minor impact on energy conservation in a properly designed model. Figure 5.15 shows the degree to which energy is conserved in the 2D flux compression simulations. Very good conservation of energy is observed as the percentage error varies from 1 to 3% for the majority of the simulation. Maximum error reaches only 4% and occurs at peak flux compression.

Figure 5.16 shows the energy distribution for 1D and 2D simulations. Not only is excellent conservation of energy observed, but energy calculations compare favorably with 1D simulations. Electron and ion energy distributions account for the reduced kinetic energy seen in the 2D simulations. Both electron and ion energies spike as the Al plasma armature stagnates on the stator.

5.3 Rayleigh-Taylor Instabilities

In this section, RT instabilities are added to the model by random initial cell-to-cell density perturbations on the plasma shells of varying amounts. In Z-pinch simulations, RT instabilities affect pinch dynamics in several ways, including broadening the plasma shell, reducing implosion velocity, and redistributing mass. RT instabilities in the flux compression design studied here may have additional affects on the dynamics due to the conical stator and opposing azimuthal magnetic field configuration.

5.3.1 Resolution Effects

Grid resolution is directly related to the minimum wavelength of RT instabilities that can be resolved. However, Douglas *et al.* have shown that grid resolution can also affect single mode and multimode RT instability development in Z-pinch simulations[31]. In particular, insufficient grid resolution can introduce an artificial dampening of the fundamental and harmonic RT instability growth rates. It can also considerably affect mode coupling in multimode development. Douglas concludes that RT wavelengths of interest should be resolved by a minimum of ten cells to prevent significant artificial reduction of a particular wavelength amplitude[31]. These conclusions should also be valid for the flux compression simulations presented here. To verify this, identical simulations at three different grid resolutions using various initial cell-to-cell density perturbations are performed. Figures 5.17 and 5.18 show the results of these simulations.

As expected, grid resolution strongly affects the development of RT instabilities. Insufficient grid resolution has a much greater impact for increased initial perturba-

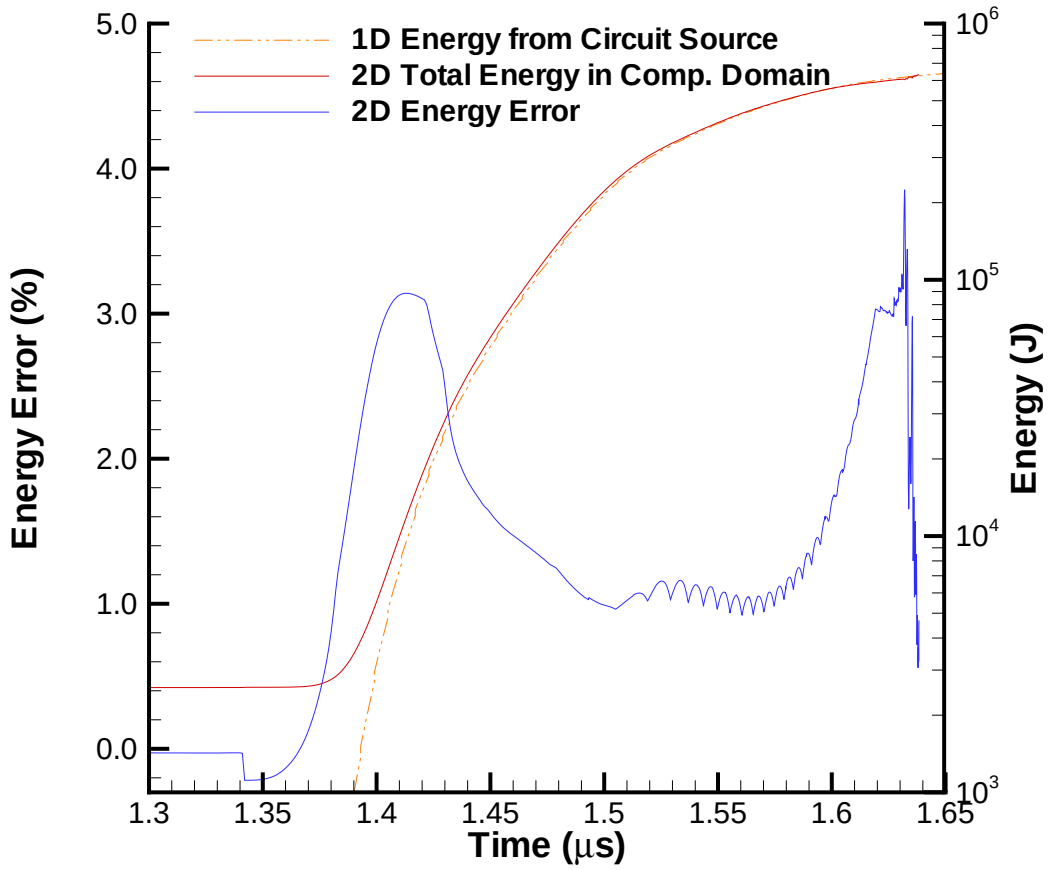

Figure 5.15: Energy conservation percentage error in the 2D flux compression simulations. Maximum error approaches 4% at peak compression. The error is calculated by determining the difference in the energy extracted from the external circuit and the sum of magnetic, kinetic, radiation, and electron and ion internal energies in the computational domain. For comparison reasons, the total energy in 1D and 2D simulations is also shown. Note that the difference in these curves initially is due to the initial energy that exists in the 2D simulations as a result of initial conditions.

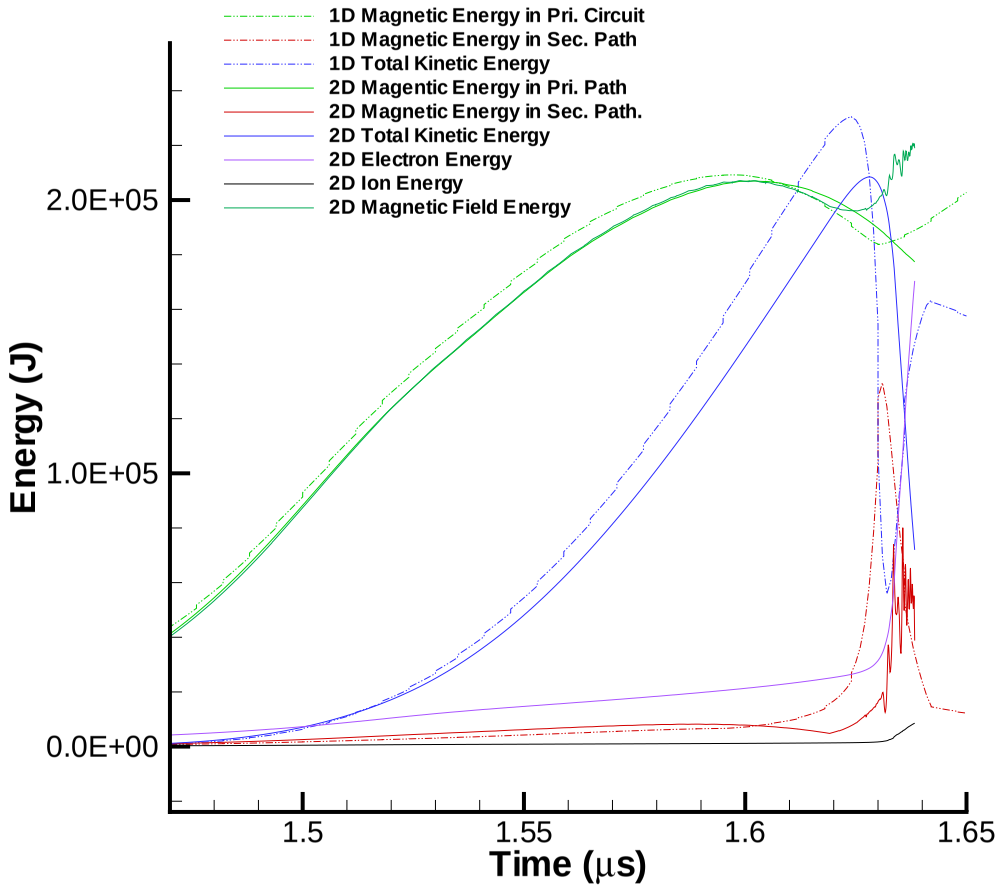

Figure 5.16: Energy distribution for 1D and 2D flux compression simulations.

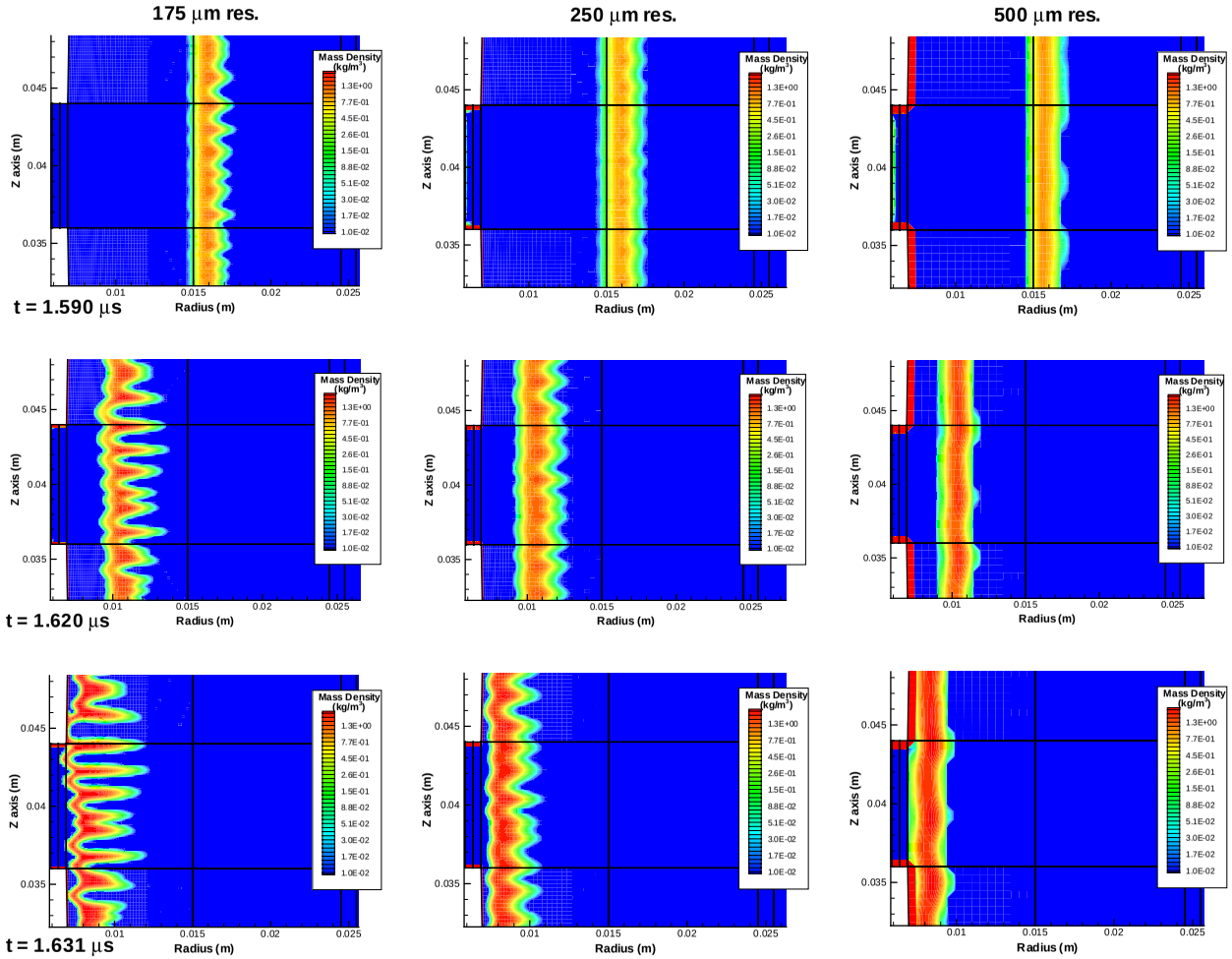

Figure 5.17: Rayleigh-Taylor instability development for various grid resolutions with a 1% cell-to-cell initial density perturbation.

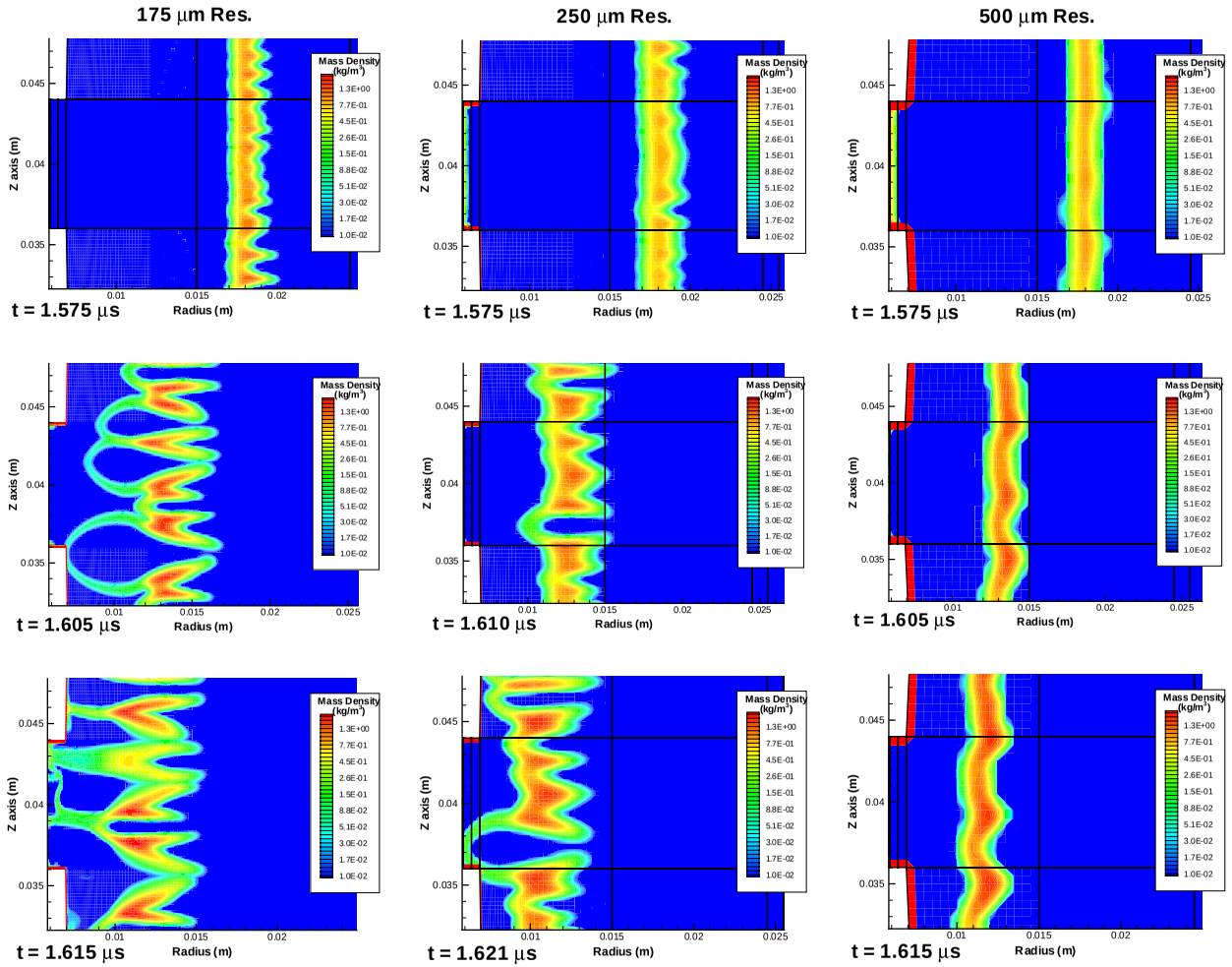

Figure 5.18: Rayleigh-Taylor instability development for various grid resolutions with a 5% cell-to-cell initial density perturbation.

tion levels. In the case of a 5% initial density perturbation, the lowest grid resolution does not capture small wavelength modes coupling into large wavelength modes. In these simulations, the most dominant RT wavelengths are between 1.5 mm and 2.0 mm. The grid resolution required to prevent significant artificial mitigation of the instabilities is on the order of 150 μm to 200 μm. Unfortunately, every factor of 2 increase in grid resolution corresponds approximately to a factor of 4 increase in computation time. The 175 μm simulations require around 72 hours on a 2.5 GHz Pentium IV to complete, excluding radiation diffusion calculations. Therefore, high grid resolutions of 150 μm and 175 μm are used for the study of the RT instabilities and their affect on flux compression implosion dynamics. A lower resolution of 250 μm is used for the study of overall flux compression dynamics and radiation issues.

5.3.2 Initial Perturbation Levels

Without experimental data, choosing an appropriate initial cell-to-cell density perturbation of the plasma shells is difficult. The number, diameter, and material of the wire arrays, as well as the drive current profile, are just some of the factors that would affect the appropriate initial density perturbation. Typically in 2D Z-pinch simulations, the initial density perturbation is chosen to match kimfol x-ray diode data with 2D calculated hydrodynamic thermalization times[33, 34]. The hydrodynamic thermalization time is an approximation of radiated power rise time[4], defined as $\Delta r/v_p$, where Δr is the shell thickness just before stagnation and v_p is the peak bulk radial velocity[29].

Based on computational results of the Saturn long-pulse mode[29], which has relatively similar conditions to DQ of current pulse, wire number, and radius, we expect initial cell-to-cell perturbations between 1 and 5% depending on initial wire array parameters to accurately model RT instability development. Figure 5.19 shows a temporal evolution of RT instabilities for a 1% perturbation. RT instabilities grow very similarly in the Al plasma armature as in standard Z-pinch simulations as the magnetic pressure on the outside of the armature is much greater than the magnetic pressure on the inside for the most of the implosion. Results of simulations comparing 1, 3, and 5% random initial perturbations seeds can be seen in Figures 5.20 and 5.21. RT development is significantly enhanced in the Al plasma armature with an increased perturbation level. As the perturbation level increases, mode coupling increases, which gives rise to increasingly larger RT wavelengths. Large RT wavelengths exhibit the most rapid nonlinear growth late in the run-in phase. In the case of a 5% perturbation level, the RT instabilities have created a large plasma bubble that impacts the stator over 20 ns before the bulk of the plasma reaches the stator.

The Ti plasma shell is much less affected by the initial density perturbation level. This was expected because the small radius of the Ti array does not allow significant RT instability growth. RT instability growth for all perturbation levels

[4]This time scale is used rather than observing computational results of radiation upon stagnation, since the MHD model is not appropriate for on-axis stagnation dynamics.

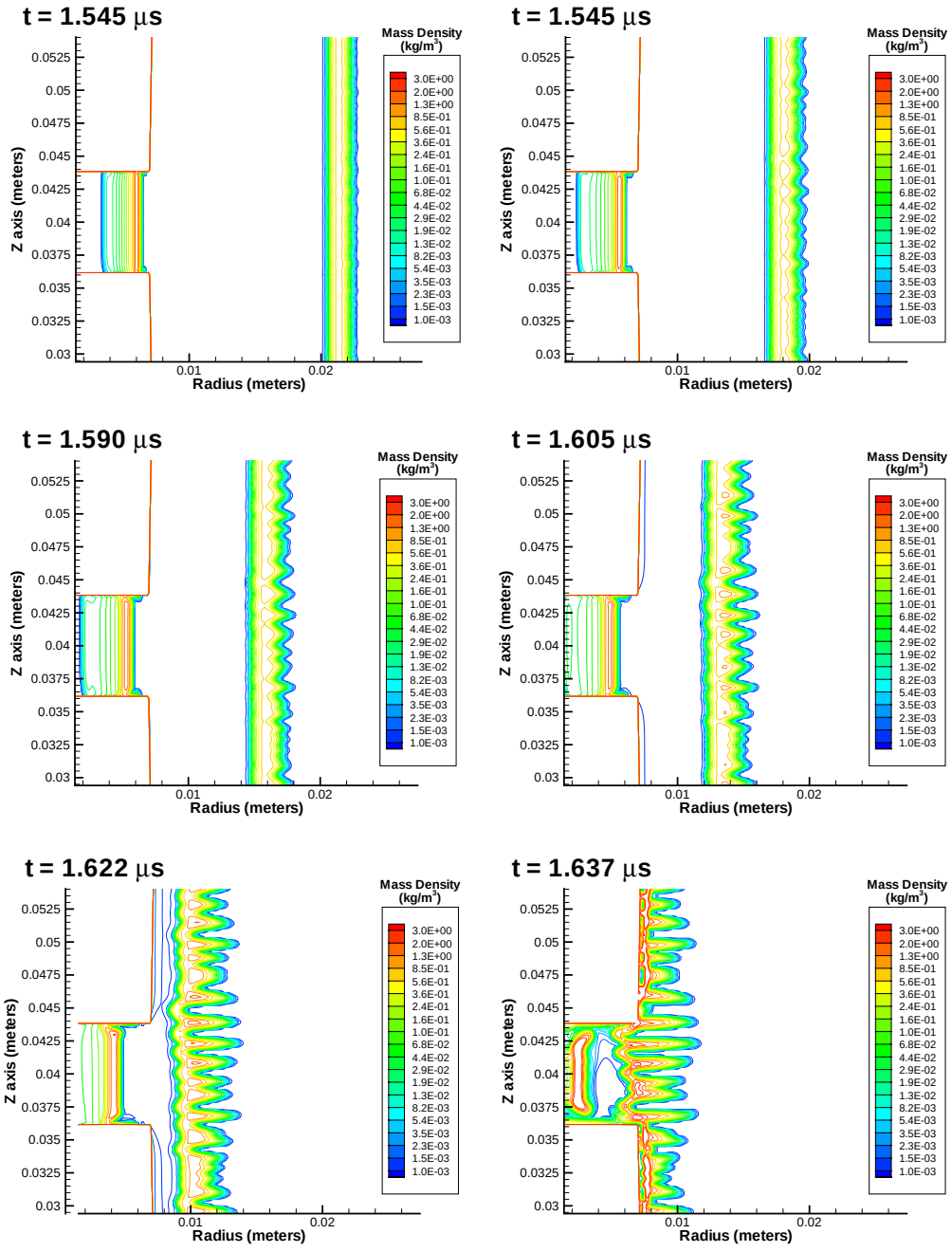

Figure 5.19: Temporal Rayleigh-Taylor development in the Al and Ti plasma shells. A 1% initial cell-to-cell random density perturbation was used for this simulation.

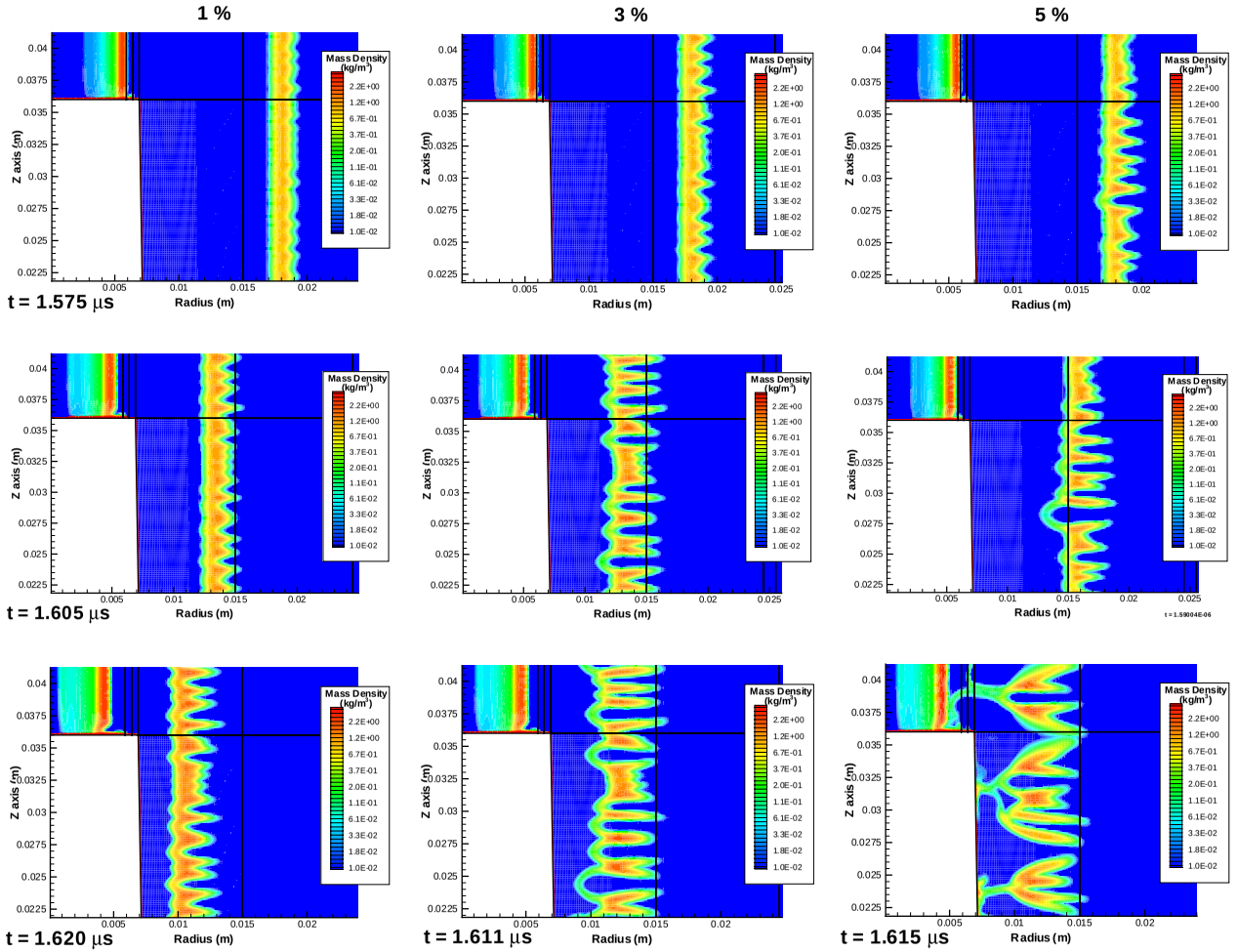

Figure 5.20: Plasma density contours showing Rayleigh-Taylor instability development for 1, 3, and 5% initial cell-to-cell density perturbations. Note 1:1 aspect ratio.

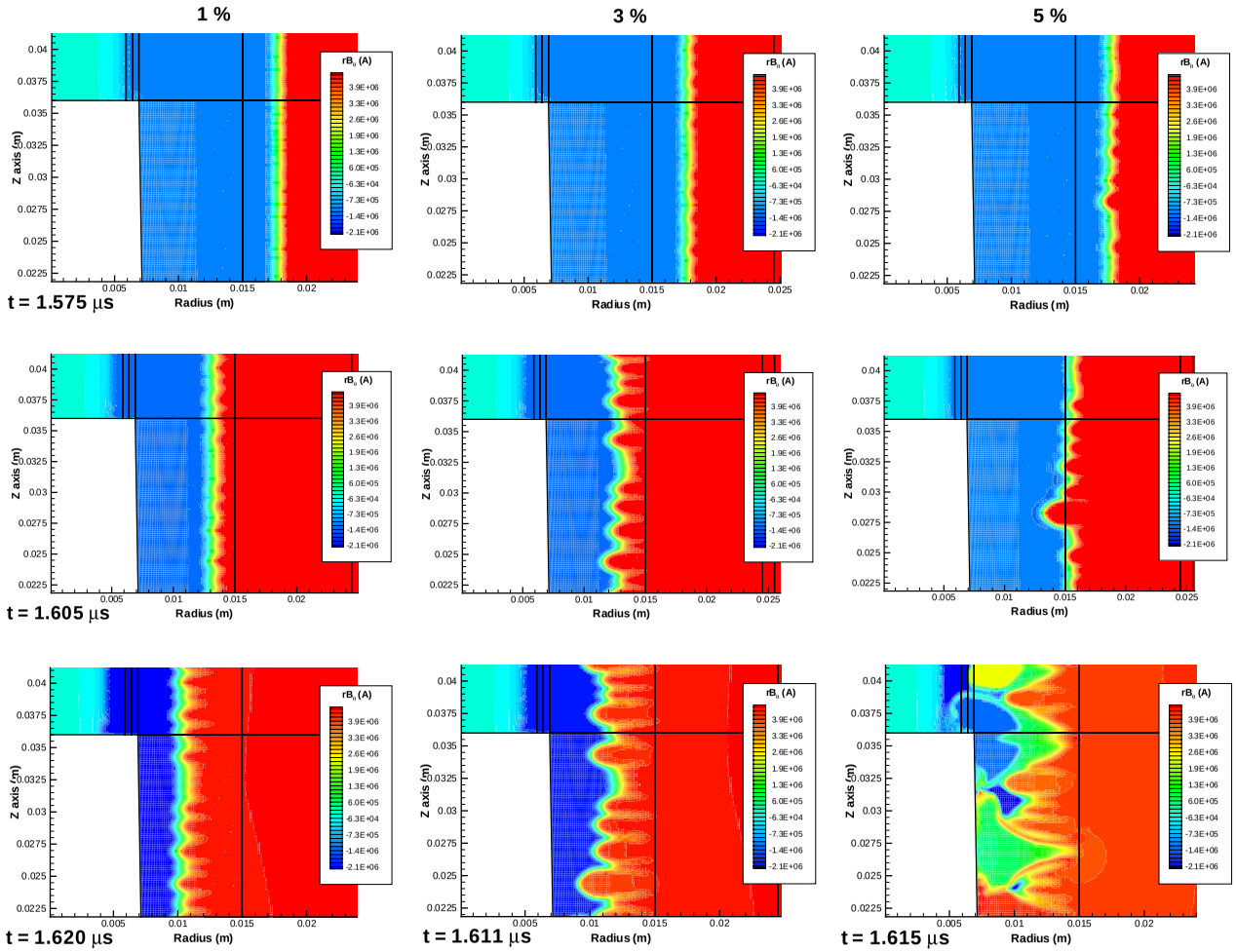

Figure 5.21: rB_θ current contours showing Rayleigh-Taylor instability development for 1, 3, and 5% initial cell-to-cell density perturbations. Note 1:1 aspect ratio.

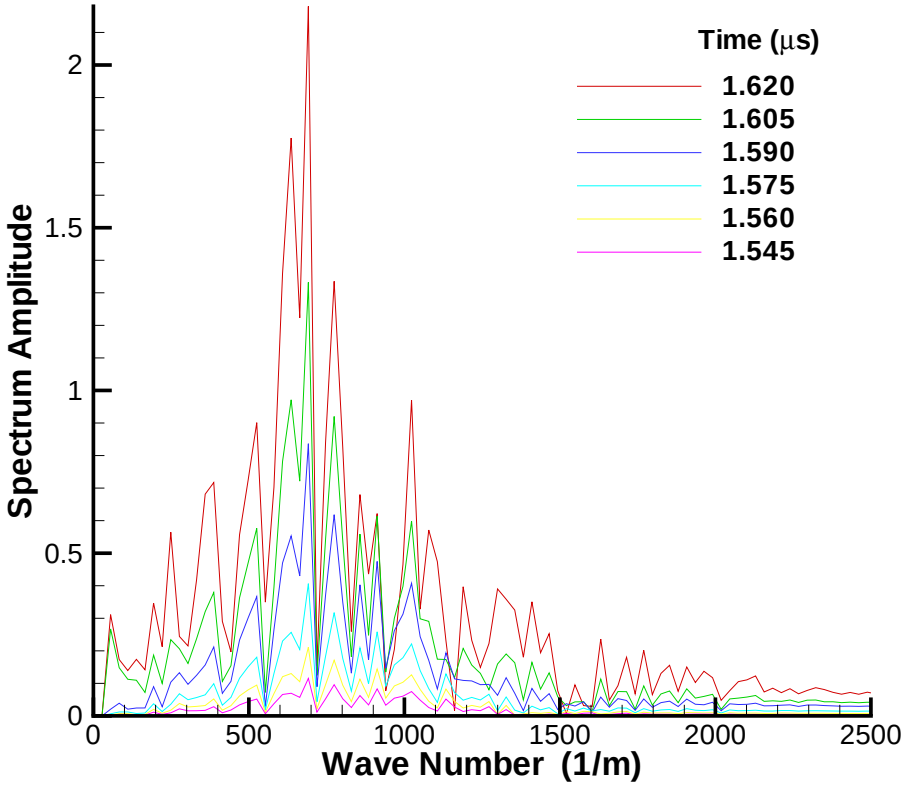

Figure 5.22: Fourier transform of axial density variations for a 1% instability.

simulated were small compared to the accelerated ends of the Ti plasma column.

In an attempt to quantify RT growth rates, the Fourier transform of the axial averaged, integrated radial mass deviation was evaluated for the case of 1% initial density perturbation. Figure 5.22 shows the result of this calculation at 15 ns intervals. The most dominant mode of RT instability growth is around 1.4 mm at all times. Interestingly, the spectral evolution of the FFT profile does not show a shift to longer and longer wavelengths as the implosion progress. This is in contrast to what has been shown in standard Z-pinch simulations[31]. It was later found that some artificial dampening of RT instabilities occurs due to insufficient grid resolution, which prevents the spectral shift. When a 5% initial density perturbation is used, the spectral shift to longer wavelengths is clearly evident, as shown in Figure 5.23. Higher grid resolution must therefore be used for smaller initial density perturbations.

Figure 5.24 plots the dominate wavelength for the 1% perturbation case. The growth rate can be quantified by fitting the data to the exponential curve $A_o e^{\gamma t}$.

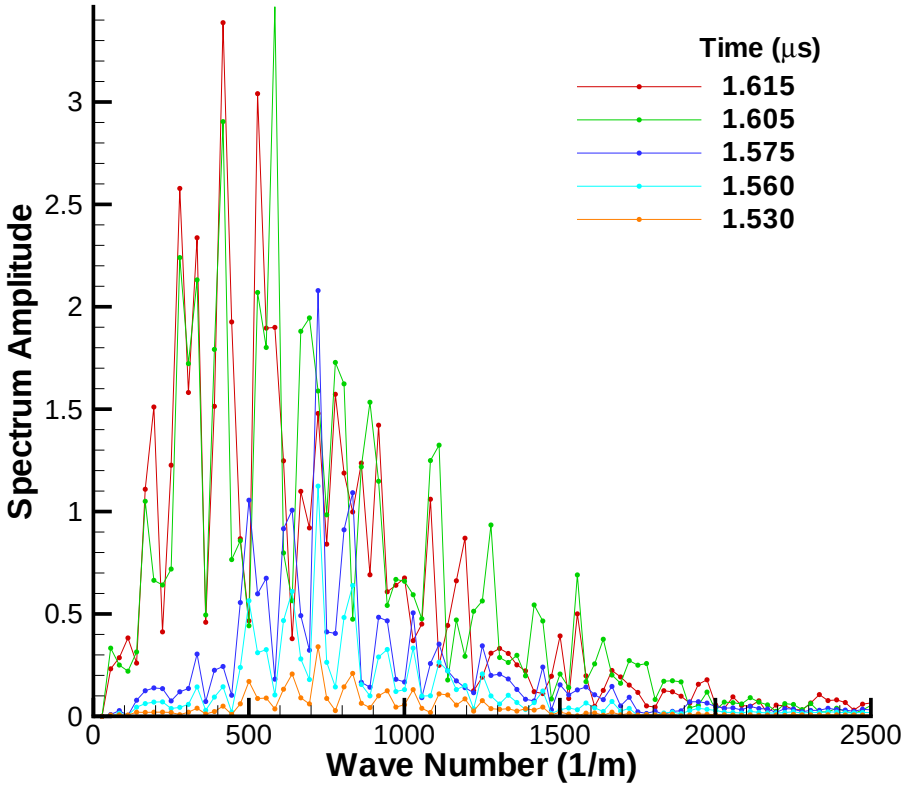

Figure 5.23: Fourier transform of axial density variations for a 5% instability.

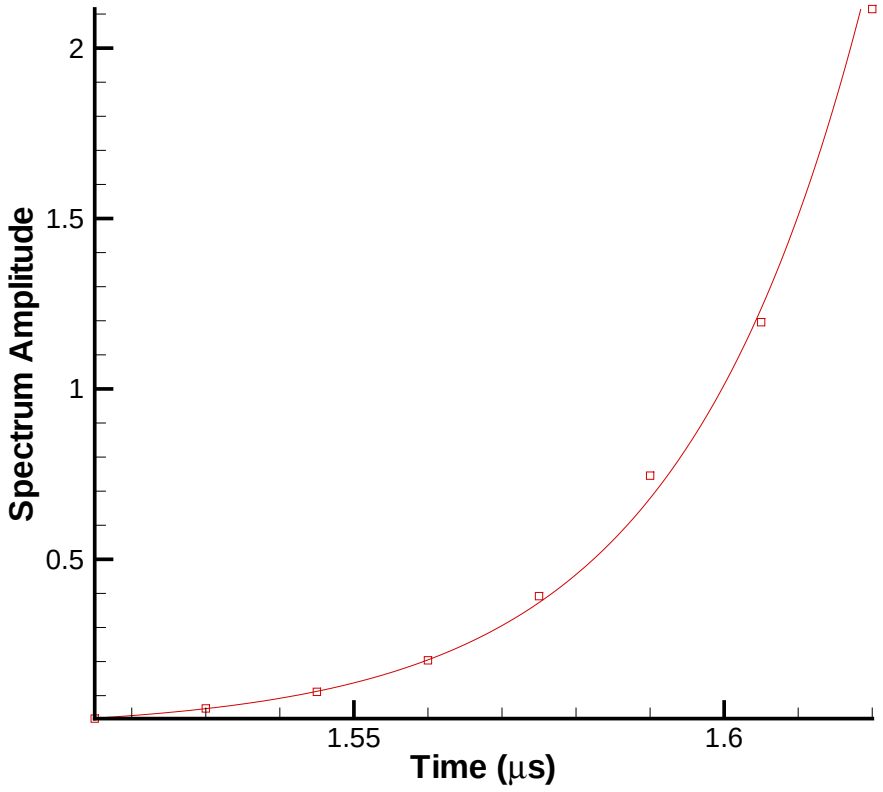

Figure 5.24: Peak amplitude of Rayleigh Taylor FFT spectrum for 1% instability.

The gamma parameter, which has units of inverse time, is 0.04 ns^{-1} for this set of data. Classical RT instability theory predicts classical growth rates of the form $\gamma = \sqrt{a(t)k_m}$, where $a(t)$ is the acceleration of the plasma and k_m is the wavelength mode[88]. The acceleration of the Al and Ti plasma shells in 1D Microcap simulations can be seen in Figure 5.25. Linear theory predicts a gamma parameter of 0.01 ns^{-1} early in the implosion and 0.05 ns^{-1} near peak compression for the Al plasma shell. Therefore, the observed 2D RT growth rate in the Al plasma armature is similar to that predicted by classical RT theory.

5.3.3 Current Distribution

As seen in Figure 5.17, significant development of RT instabilities occurs in the Al plasma armature with initial density perturbations as small as 1%. As one might expect, the instabilities have significant impact on current distributions. Figures 5.26 and 5.27 show density, electron temperature, ohmic heating, and rB_θ current

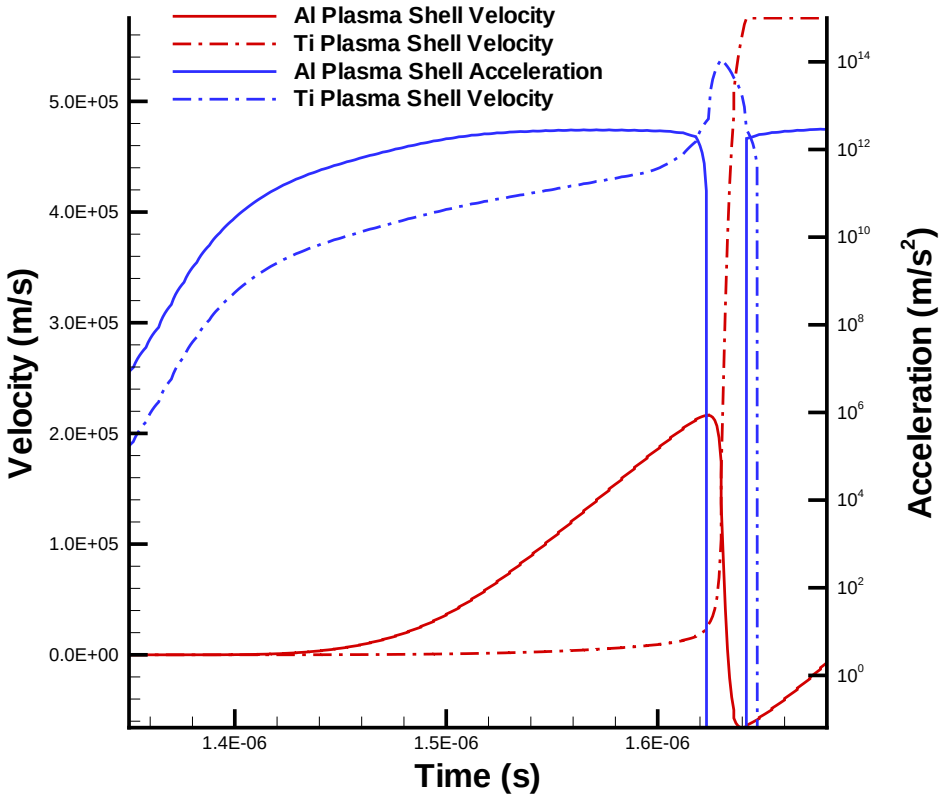

Figure 5.25: Radial velocity and acceleration of Al and Ti plasma shells. These results were produced with the 1D Microcap flux compression model.

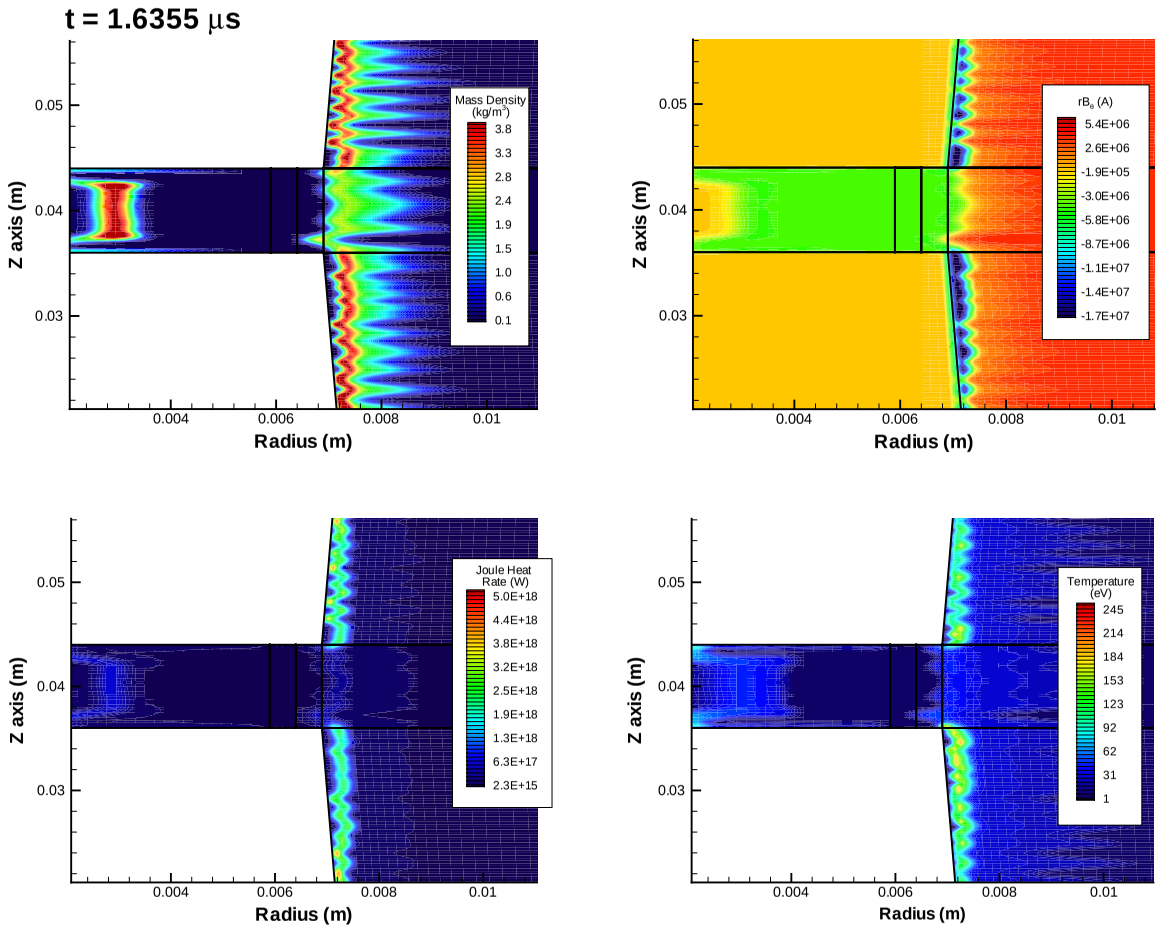

Figure 5.26: Density, current, joule heat, and electron temperature contours near peak flux compression with 1% initial density perturbation. Current vectors demonstrate the development of circulation currents around existing magnetic field lines.

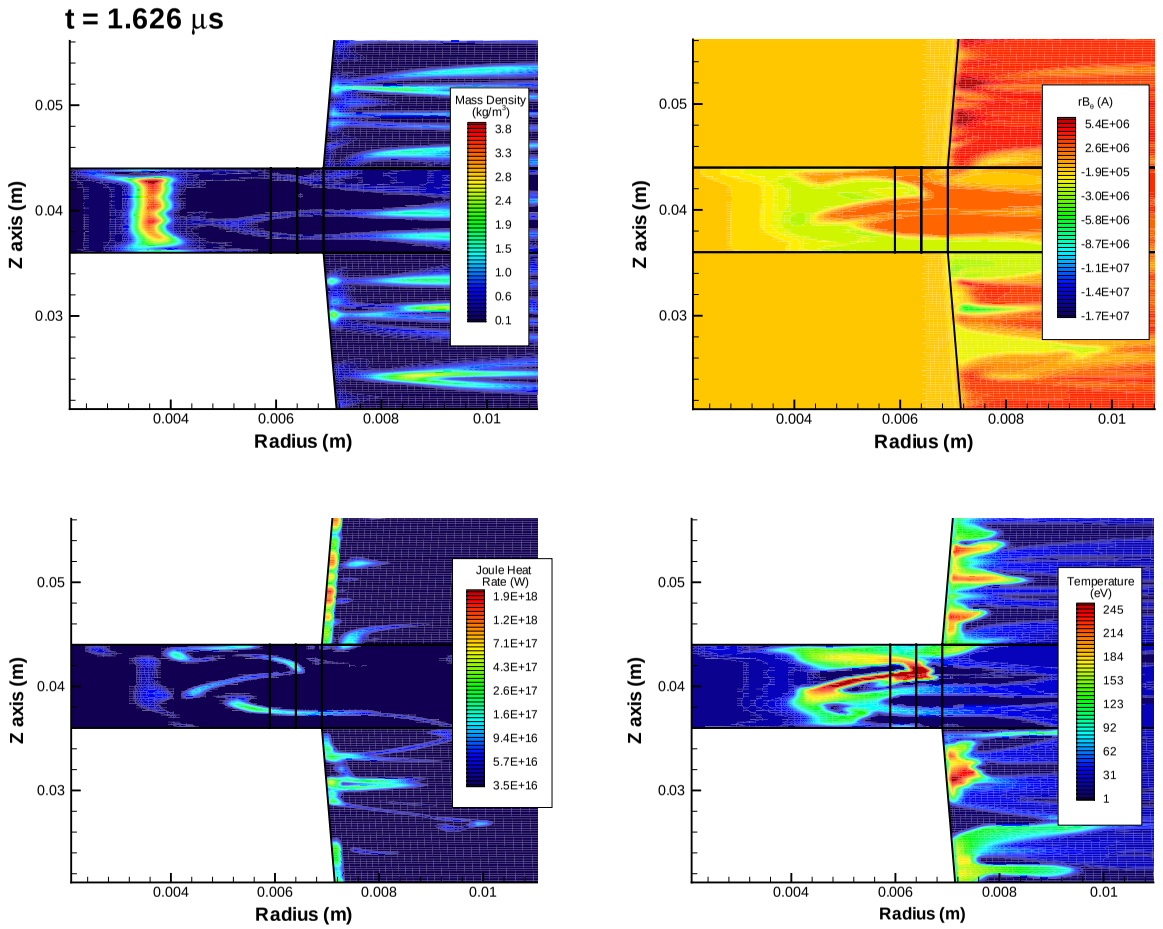

Figure 5.27: Density, current, joule heat, and electron temperature contours near peak flux compression with 5% initial density perturbation. Current vectors demonstrate the development of circulation currents around existing magnetic field lines.

contours, at peak compression for initial density perturbations of 1 and 5%, respectively. RT instabilities redistribute plasma mass and form magnetic bubbles between thin Al density spikes. Since the mass density is much lower surrounding the magnetic bubbles, the magnetic bubbles are accelerated at a faster rate than the bulk Al plasma column and have sufficient momentum to impact the stator. Consequently, isolated regions of opposing magnetic flux become trapped between RT spikes as shown in Figure 5.28. The trapped flux pockets form new magnetic bubbles with an opposing azimuthal magnetic field.

When the initial density perturbations are increased, RT instabilities can trap isolated regions of magnetic flux by a magnetic bubble burst or magnetic bubble merging. Figure 5.29 shows a magnified view of the armature just before stagnation with a initial 5% density perturbation. RT mode coupling gives rise to a large magnetic bubble formation that exhibits rapid nonlinear growth. The magnetic bubble bursts shortly after impacting the stator as a result of increasing axial $\mathbf{J} \times \mathbf{B}$ forces. Mass from the bubble continues to accelerate axially and a large pocket of magnetic flux is formed along the stator. Also seen in Figure 5.29 are two smaller magnetic bubbles folding over and merging. This also traps magnetic flux into an isolated region in which a circular current distribution flows around it.

Current distributions are shunted into the peripheries of the trapped magnetic flux pockets and current flow begins to circulate around them. The aluminum armature continues to compress these regions, which further enhances the current. Electron temperatures show localized high temperature regions, induced by adiabatic and ohmic heating, on the order of 100 to 300 eV. With proper diagnostics, an experimental flux compression shot would be able to detect radiation from these localized high temperature Al regions.

Unfortunately, the circulation currents formed along the stator reduce the magnetic flux available for compression in the Ti array since it is no longer part of the secondary circuit path. The circulation currents surround magnetic bubbles that contain isolated azimuthally opposed magnetic flux. As the bubbles are compressed, they undergo independent localized magnetic flux compression. In addition, the circulation currents accelerate the rate of magnetic to thermal energy conversion due to joule heating, which also enhances overall magnetic flux loss.

Circulation currents, however, are not the biggest source of magnetic flux loss occurs due to RT instabilities. In the formation of the magnetic bubbles, RT instabilities prematurely short out the conical stator. Consequently, only a fraction of the flux compression current flows through the Ti array plasma. This process is similar to the current switching observed in section 5.2, but occurs at earlier times depending on the perturbation level. In general, as the perturbation level is increased, the amount of trapped magnetic flux available for kinetic energy transfer to the Ti plasma is decreased.

From analysis of the current distributions, it must be concluded that the current versus time plots, such as Figure 5.3, are not indicative of the levels of current that flow through the Ti wire array plasma. The current histories in these plots must be interpreted as current input for the boundary conditions only.

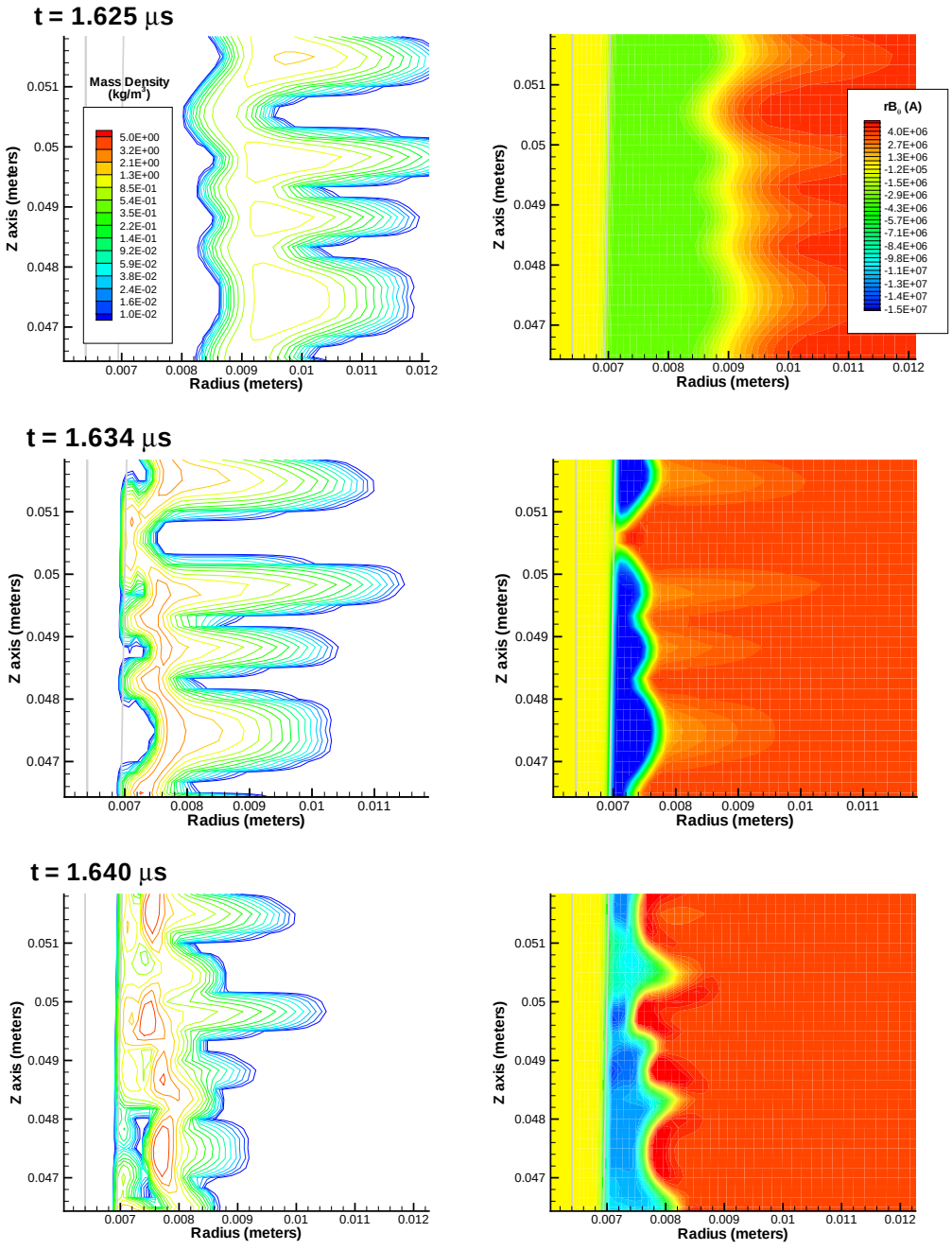

Figure 5.28: Trapping of magnetic flux due to impact of RT instabilities along the stator. Initial density perturbation is 1%.

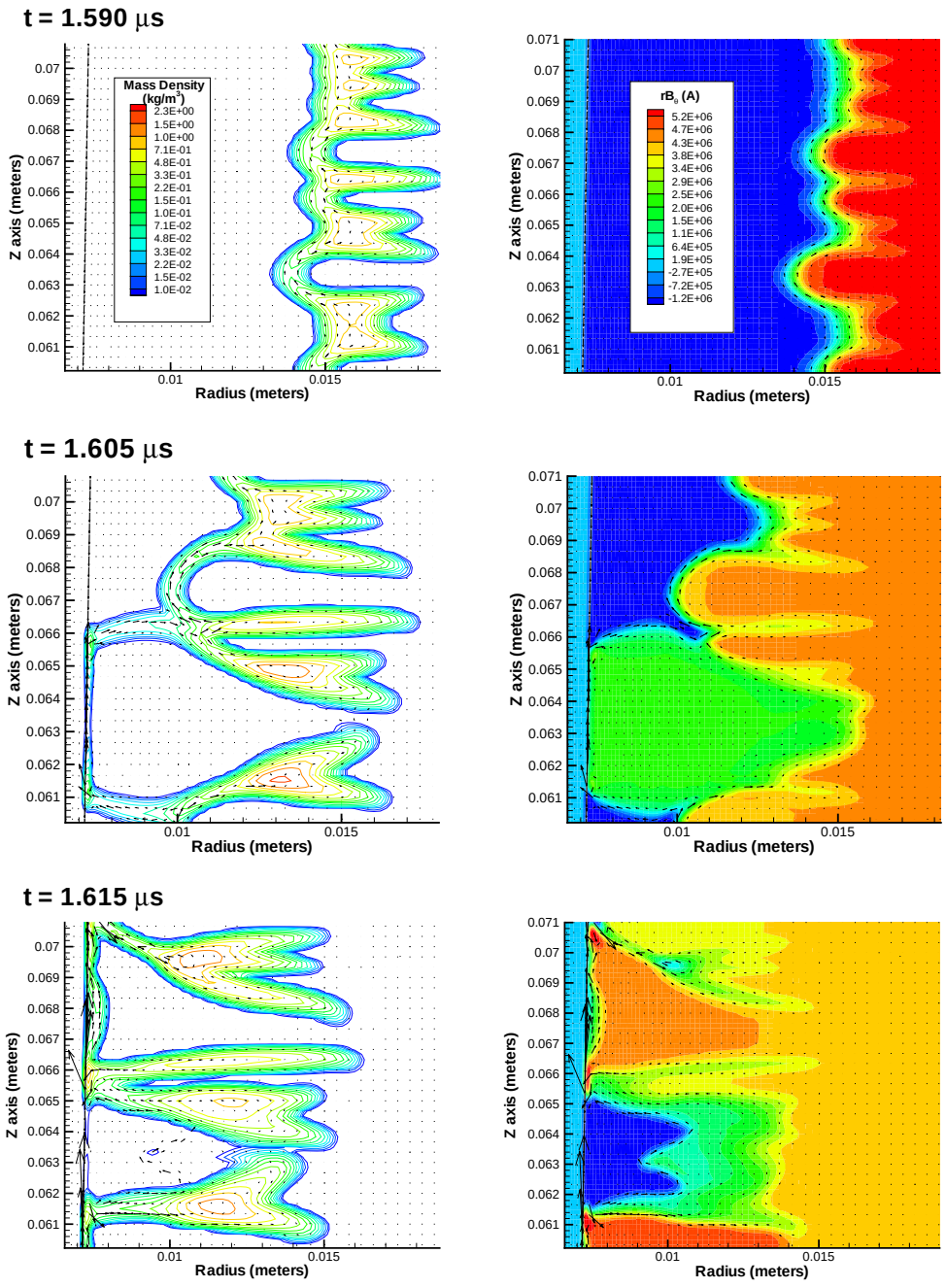

Figure 5.29: Magnified view of magnetic bubble development, burst, and the trapping of magnetic flux. Initial density perturbation is 5%.

5.3.4 Hydrodynamic Thermalization

Previously, it was estimated that efficient Ti K-shell radiation production would be achieved with ion velocities greater than 60 $cm/\mu s$. In the flux compression simulations presented here, the current distribution is such that only a fraction of the secondary path current flows into the Ti plasma. Large K-shell radiation yield is, therefore, not expected. Figure 5.30 shows Ti velocities just before stagnation. Ti velocities approach only 40% of the velocity required for efficient K-shell radiation production.

5.4 Material/Interface Tracking

Material interfaces are intrinsic to Lagrangian simulations. However, material interfaces are not clearly defined in ALE and Eulerian simulations and therefore require special algorithms. All of the simulations in this research are fixed grid Eulerian calculations. For compressible flows, the MACH2 code provides users with two different algorithms for handling multiple materials and material interfaces. In the non-tracking mode, material interfaces are treated as mixed zones and thus each cell's transport and equation of state variables are determined by the material with the greatest mass ratio in each cell. In the tracking mode, material interfaces are reconstructed at every time step using the Simple Line Interface Calculation (SLIC) developed by Noh and Woodward[89]. This algorithm is based on keeping track of mass, volume, and energy fractions throughout the simulation. This option is especially useful in modeling solid materials when the material interfaces are assumed to be continuous throughout the simulation.

The SLIC algorithm is one of the earliest interface tracking and reconstructing algorithms. It reconstructs interfaces by integrating line segments parallel and perpendicular to the computational grid[90]. Unfortunately, when advecting objects are not directly oriented with the grid, interfaces with discontinuities can exist and object geometries can become distorted. The degree of distortion directly affects the accuracy of the solution. In addition, new numerical stability issues are introduced as well. Modern interface tracking and reconstruction algorithms would be a welcomed addition to the MACH2 code.

Is the tracking or non-tracking version of the MACH2 code more appropriate for flux compression simulations? The answer to this question depends on whether or not the plasma/vacuum interfaces are assumed to be continuous without any mixing zones. It is more logical to assume a traditional compressible flow approach for mixed zones at the interface. Indeed, most high density plasma physics simulations, including Z-pinch simulations with similar plasma/vacuum interfaces, performed in MACH2 use the non-tracking mode[91]. However, since the vacuum material in these simulations acts merely to transport magnetic field, MACH2's tracking mode is also appropriate.

Identical simulations in both modes may provide insight into any numerical issues that may exist in handling multiple materials. In particular, different results may be

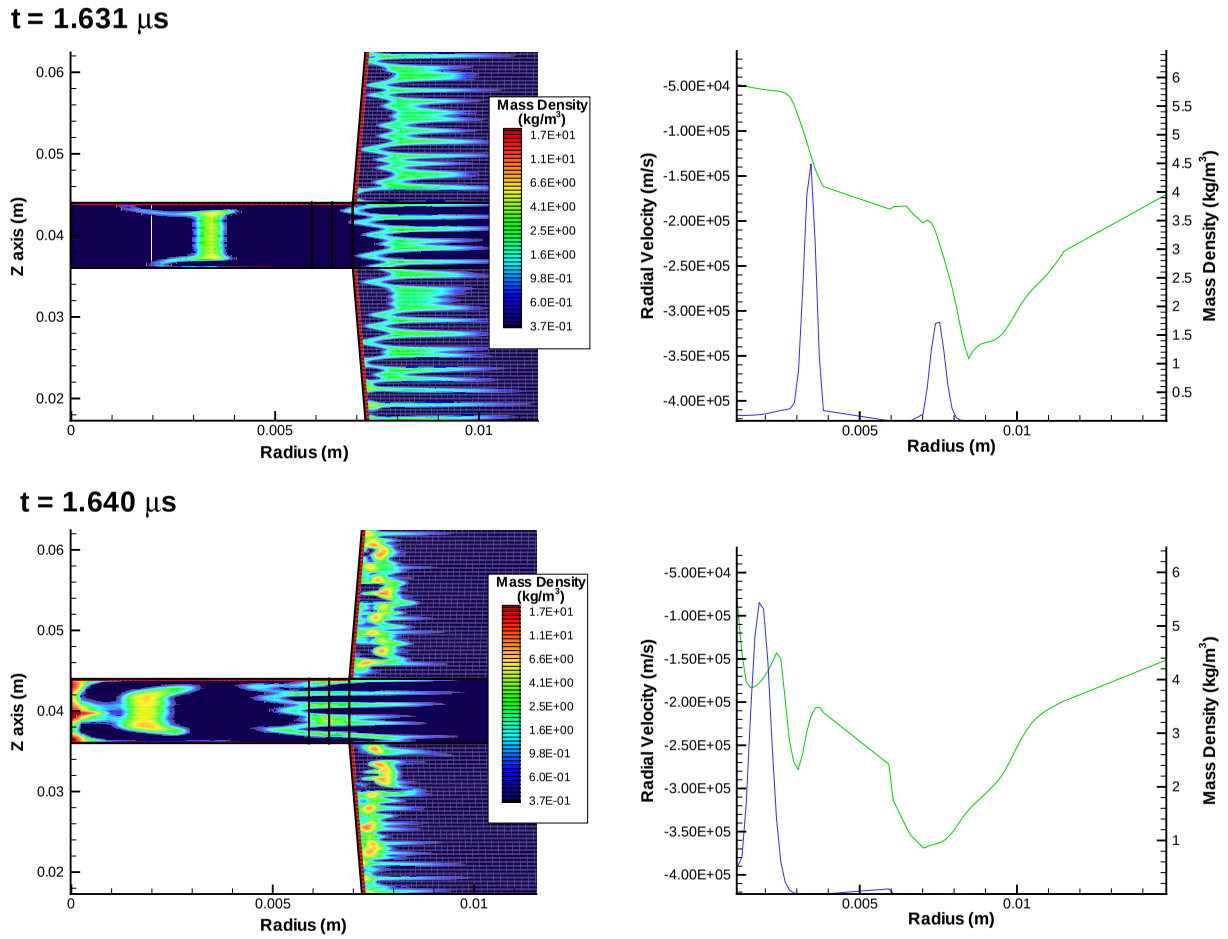

Figure 5.30: Velocity and density of Ti plasma just before stagnation with 1% initial cell-to-cell density perturbation.

obtained with difference settings of the tracking tolerance. The tracking tolerance is defined as the minimum material mass to cell volume ratio the code will keep track of. Lower settings of the tracking tolerance in general should produce more accurate results. However, computational time steps must be decreased with lower tracking tolerances[5], which significantly increases computation time required for the simulation. Additionally, machine precision errors and errors in tabular equation of state from the interpolation to very small densities become greater.

Figure 5.31 shows density contours of identical simulations run in MACH2's tracking[6] and non-tracking modes at a time t=1.624 μs in the implosion. The greater plasma densities and more pronounced RT instabilities in the tracking mode are somewhat surprising. Resistive impedance voltage profiles, as shown in Figure 5.32, seem to indicate the discrepancy can be explained by the numerical calculation of ohmic heating alone. The method in which ohmic heating of the materials is calculated in the tracking mode is different, since equation of state variables must be partitioned in the mixed cells. Higher ohmic heating rates calculated in the tracking mode create higher plasma temperatures and raise the conductivity of the plasma. The increase in conductivity decreases the amount of magnetic diffusion into the plasma and produces more localized regions of current density. This could account for the greater levels of RT instabilities seen in the tracking case.

Current and flux history comparisons in Figure 5.33 support this theory as evidenced by the greater magnetic flux loss in the secondary array for the non-tracking case after the crowbar circuit engages. A conclusive analysis could be made by altering the new code routines to turn off ohmic heating and then running the simulations again.

5.5 Conical Stator Angle

Simulations presented thus far have a stator angle of 0.95 degrees, which corresponds to the original design. The original design angle was chosen somewhat arbitrarily as a tradeoff between peak K-shell powers and premature shorting due to plasma instabilities. 2D simulation results presented thus far demonstrate non-ideal current distributions due to the armature plasma shorting out the flux compression on the inner Ti wire array plasma. A logical conclusion would be that the angle on the conical angle should be increased to achieve maximum energy and power transfer to the Ti plasma. To investigate this, identical simulations were run with stator angles of 0.95, 1.75, and 2.5 degrees.

Calculated inductances for various stator angles can be seen in Figure 5.34 for 1D and 2D simulations. Both 1D and 2D simulations demonstrate similar reductions in secondary circuit path inductance as the conical stator angle is increased. Since the additional anode post inductance is much higher than the secondary current path

[5] As the material fraction decreases, it becomes more and more likely that the material mass can be accelerated to a velocity that would not place the material in an adjacent cell in the next time step.

[6] The tracking tolerance that was used was trktol=1.0e-4.

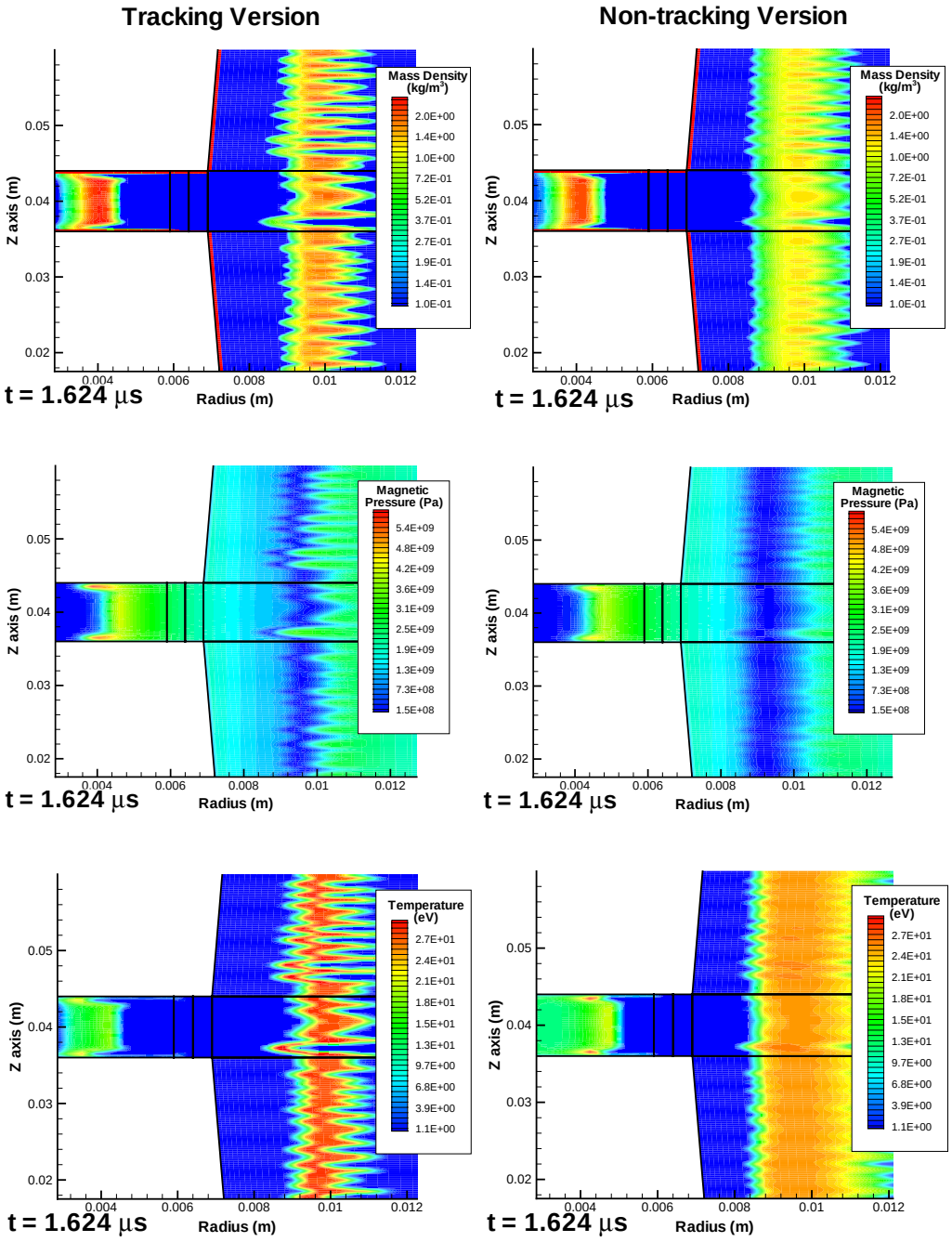

Figure 5.31: Comparison of tracking and non-tracking MACH2 flux compression simulations with an initial 1% cell-to-cell density perturbation.

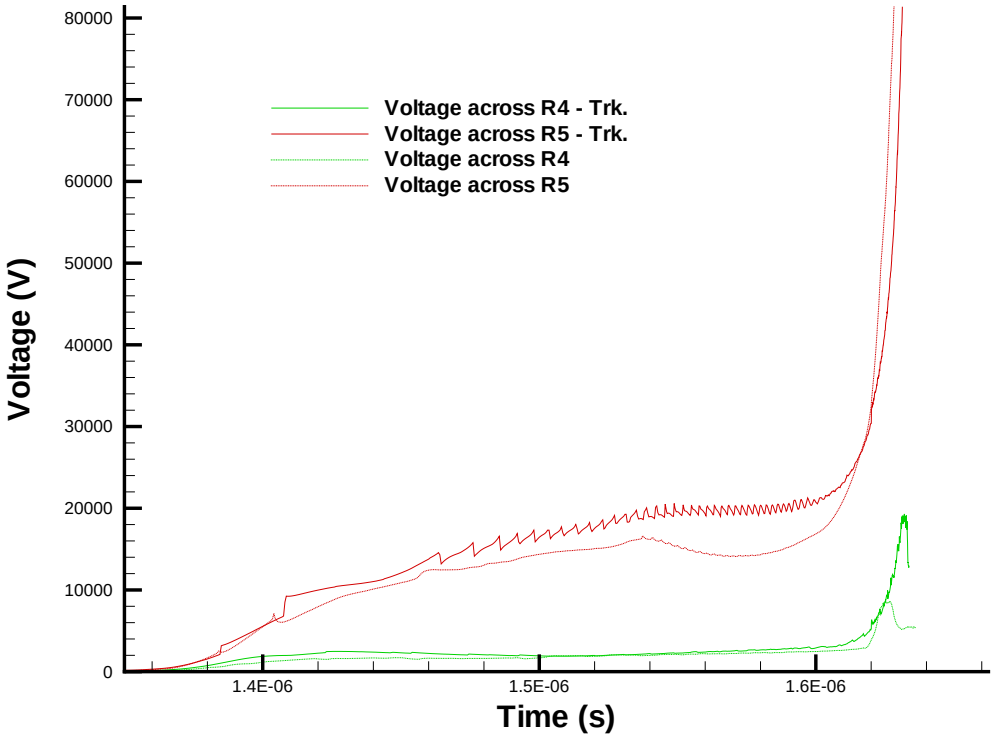

Figure 5.32: Voltage comparison in tracking and non-tracking simulations with an initial 1% cell-to-cell density perturbation.

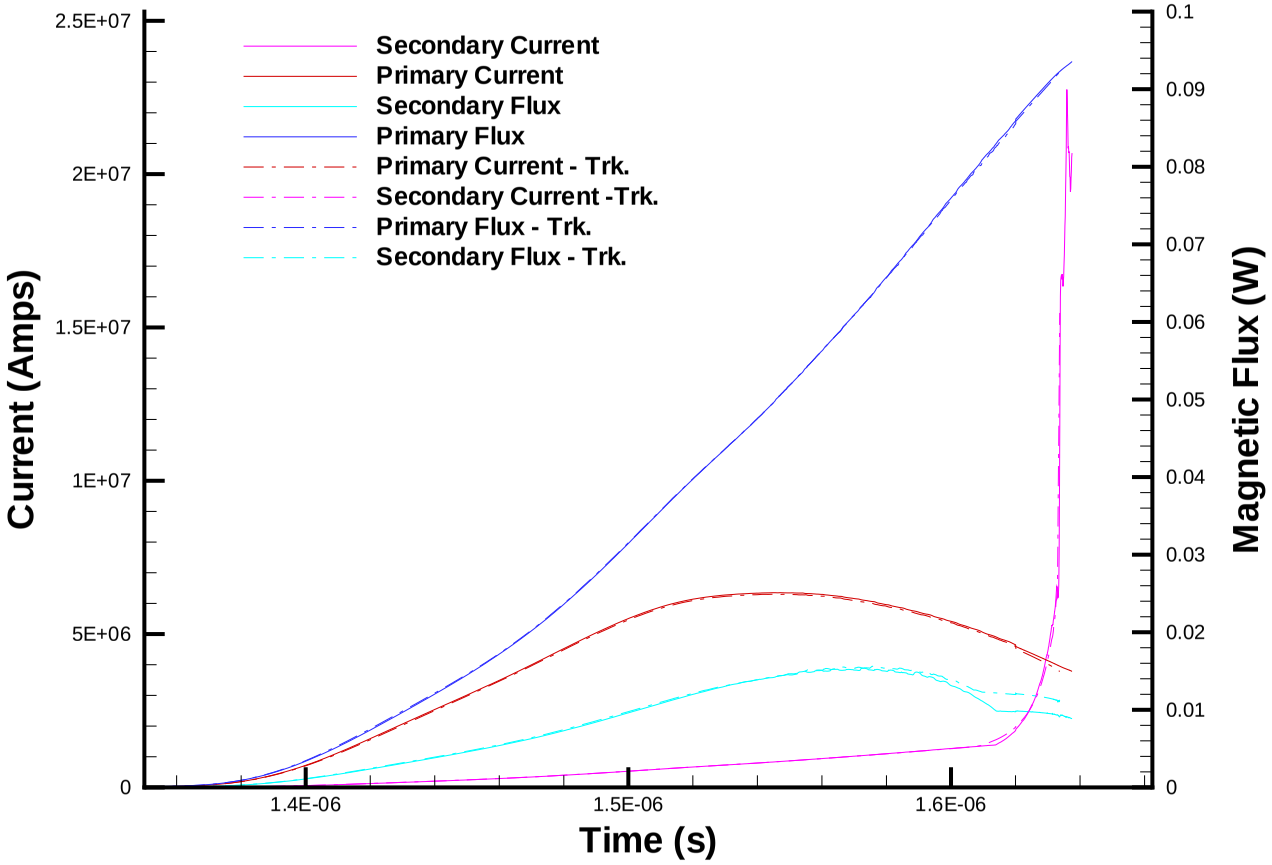

Figure 5.33: Comparison of current and magnetic flux profiles in the tracking and non-tracking MACH2 flux compression simulations with an initial 1% cell-to-cell density perturbation.

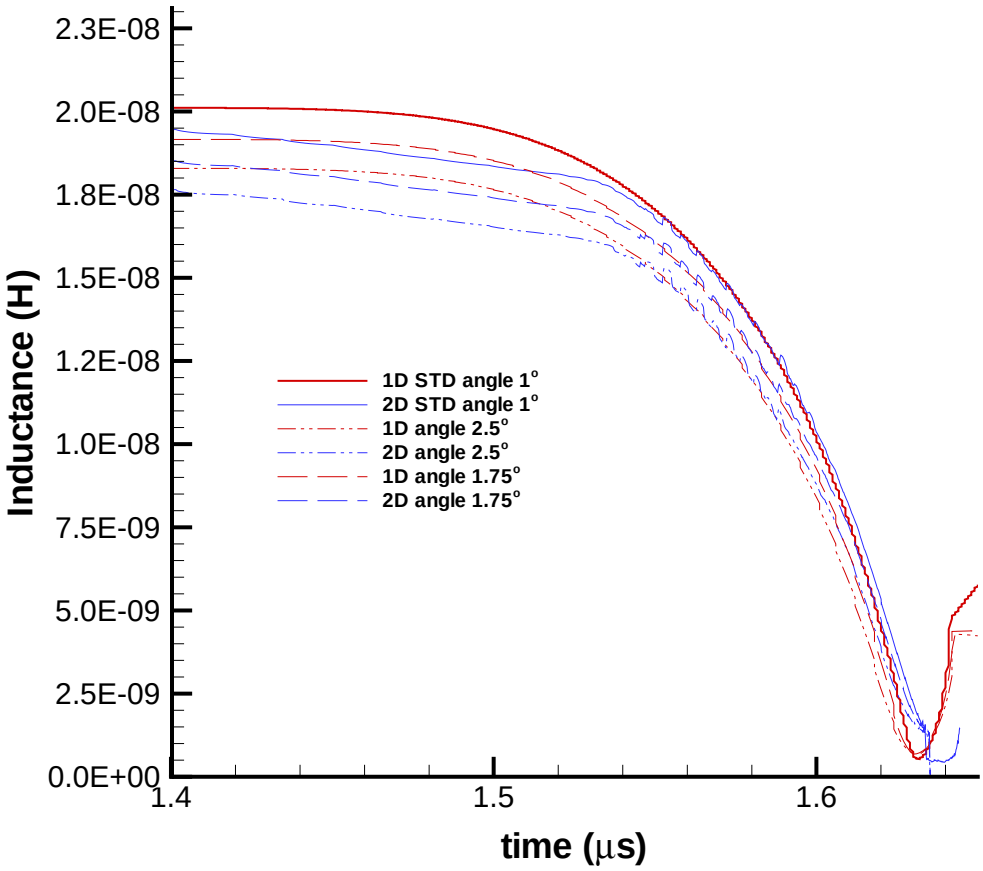

Figure 5.34: 1D and 2D secondary circuit path inductance calculations for various conical stator angles. As the angle is increased, a corresponding reduction in evident in both 1D and 2D simulations. The initial slope difference between 1D and 2D simulations is due to shell thickness and expansion approximation differences.

inductance, the current levels in the secondary path, before the crowbar engages, are only marginally higher (1 to 2 %), as shown in Figure 5.35. Therefore, magnetic flux in the secondary current path is reduced with increased stator angle, which results in less magnetic energy in the secondary array.

Figures 5.36 and 5.37 show how increased stator angles affect the current distribution into the Ti array when plasma instabilities are not considered. rB_θ contours indicate current levels in the Ti array reach levels of 5.35 MA and 6.0 MA for stator angles of 2.5° and 1.75°, respectively. These levels are slightly higher than the 5.0 MA levels shown in simulations using a 1.0° stator angle in Figures 5.12 and 5.13.

Current switching is still observed in both simulations and is the primary loss mechanism for energy transfer to the Ti plasma array. In fact, current switching in these simulations lead to the formation of a large circulation current, 200 ps after the last contour plot in Figures 5.36 and 5.37, at the inner tips of the stator. The large circulation currents immediately drive the circuit solver unstable from the large amount of feedback into the circuit. The point at which the circuit goes unstable is evident in Figure 5.38. Secondary current pulse broadening can also be seen in the same figure. 1D simulations show conical stator angles of 1.75° and 2.5° would reduce secondary current levels from 22 MA to 18 MA and 15 MA, respectively. Unfortunately, pulse broadening in 2D simulations is difficult to determine since the circuit solver becomes unstable at peak compression and the simulation is unable to continue from that point.

It is plausible that increasing the stator angle will be more beneficial when RT instabilities are implemented into the simulations since it may prevent or delay premature armature current shorting on the stator. Figure 5.39 shows mass density contours and Figure 5.40 shows rB_θ current contours at three different times near peak compression with an initial perturbation on 1%. The data indicates higher currents are delivered to the Ti plasma with increased stator angle. Current levels are in excess of 7 MA for a stator angle of 2.5° which is higher than that observed in the case without plasma instabilities.

Premature armature shorting cannot be avoided with greater conical stator angles. Greater angles can delay the point at which armature shorting occurs and allow a greater current level to flow through the Ti plasma. However, maximum current levels in the Ti plasma are still far below that needed for efficient K-shell radiation production.

5.6 Radiation Cooling

Although detailed radiation modeling is not an emphasis area of this research, simulations were run using MACH2's radiation emission and single group non-equilibrium radiation diffusion models in order to investigate potential effects of radiation cooling during the run-in phase. The radiation emission model, which is only accurate in the optically thin limit, is based on mean Planck opacities. MACH2 removes

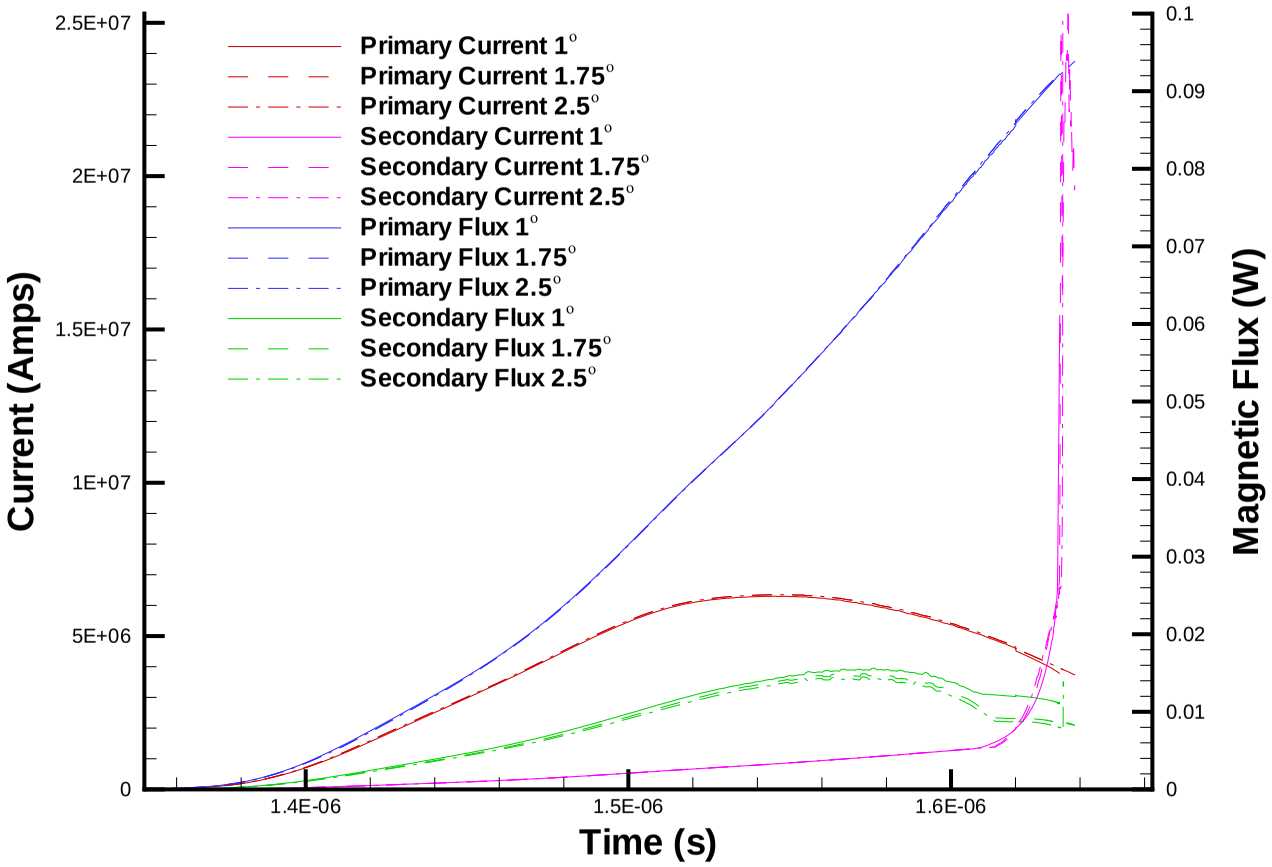

Figure 5.35: Boundary condition current and magnetic flux evolution for various stator angles. A 1% initial cell-to-cell perturbation level was used for these simulations.

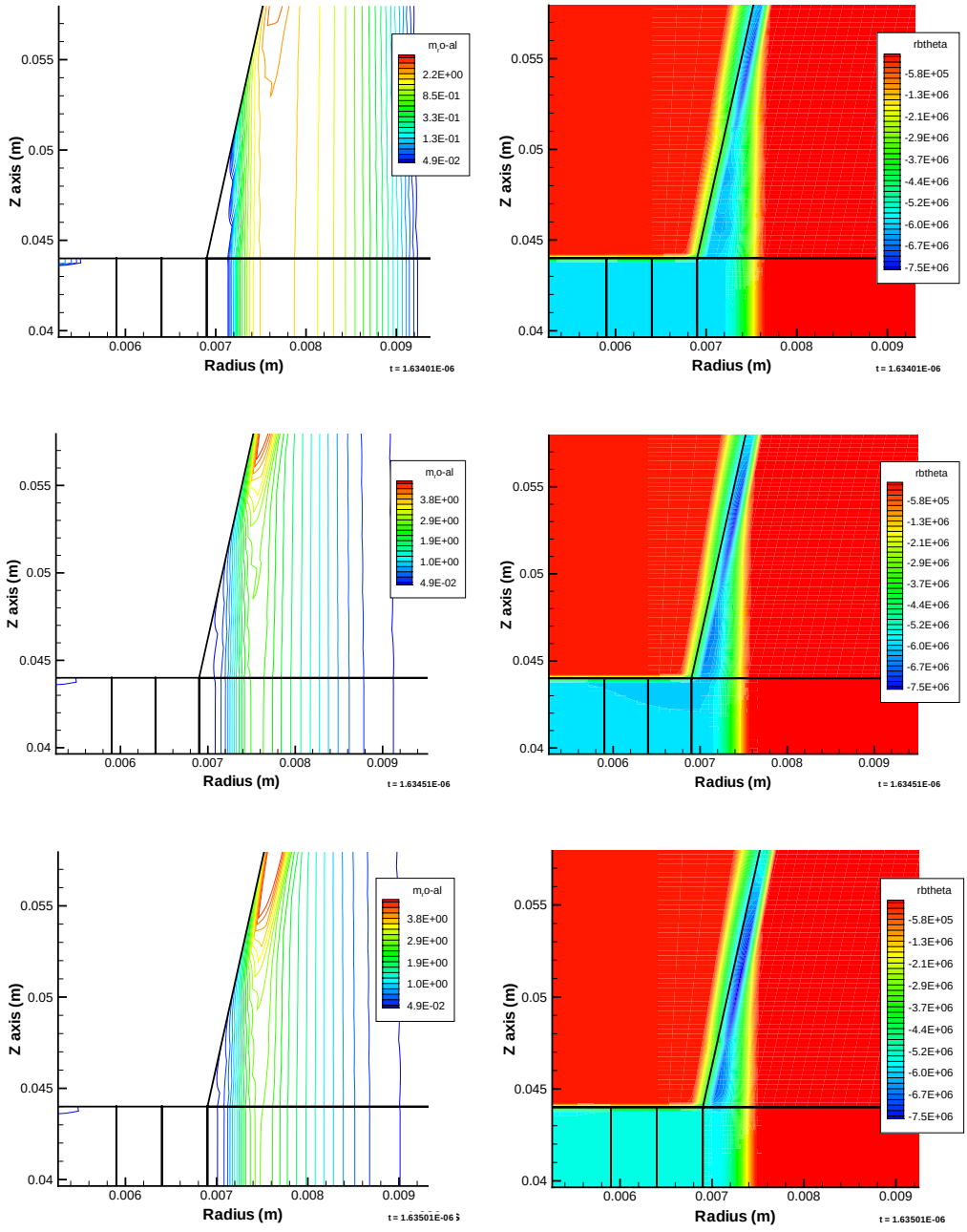

Figure 5.36: rB_θ and plasma density contours near peak compression for an increased stator angle of 1.75°.

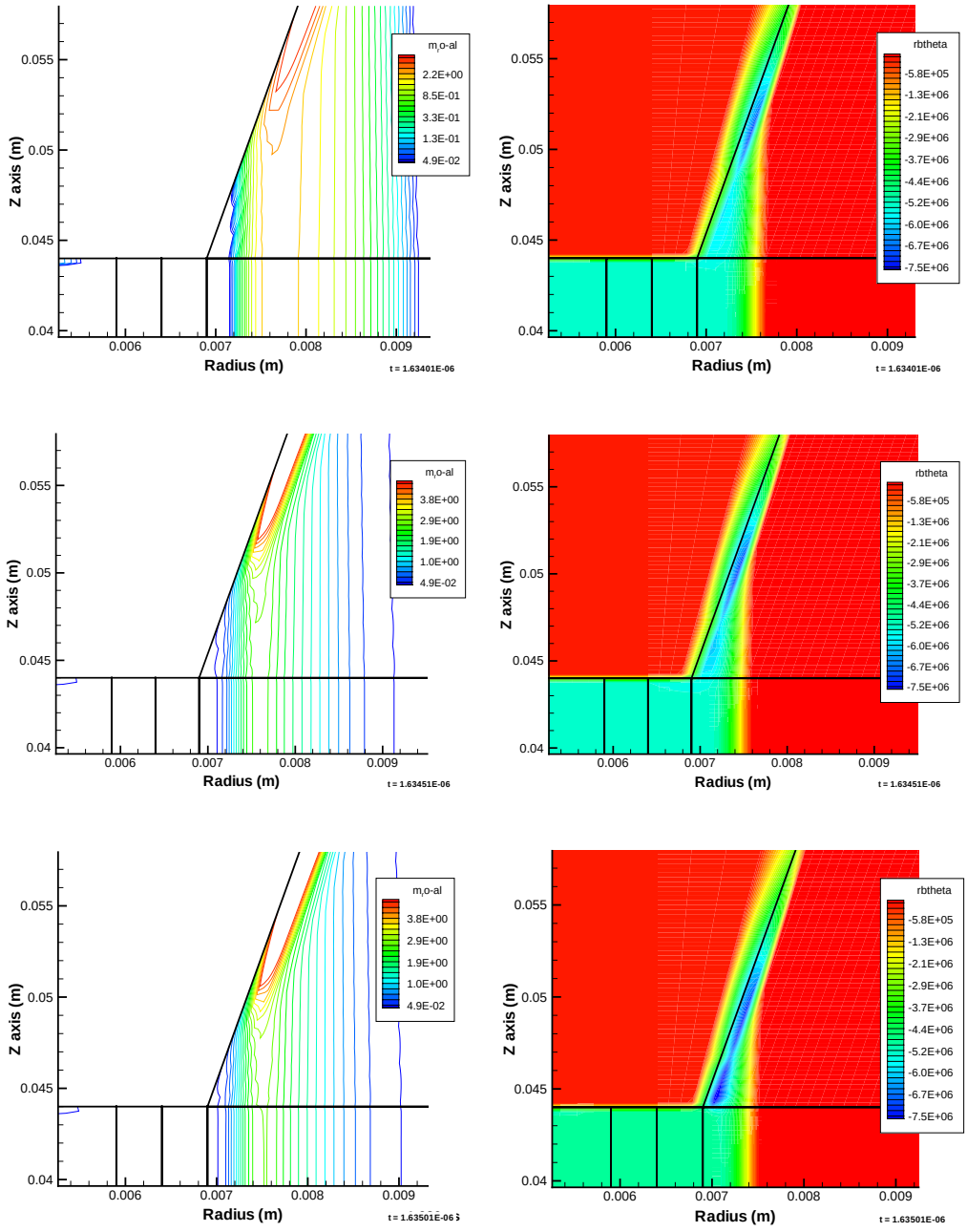

Figure 5.37: rB_θ and plasma density contours near peak compression for an increased stator angle of 2.5°.

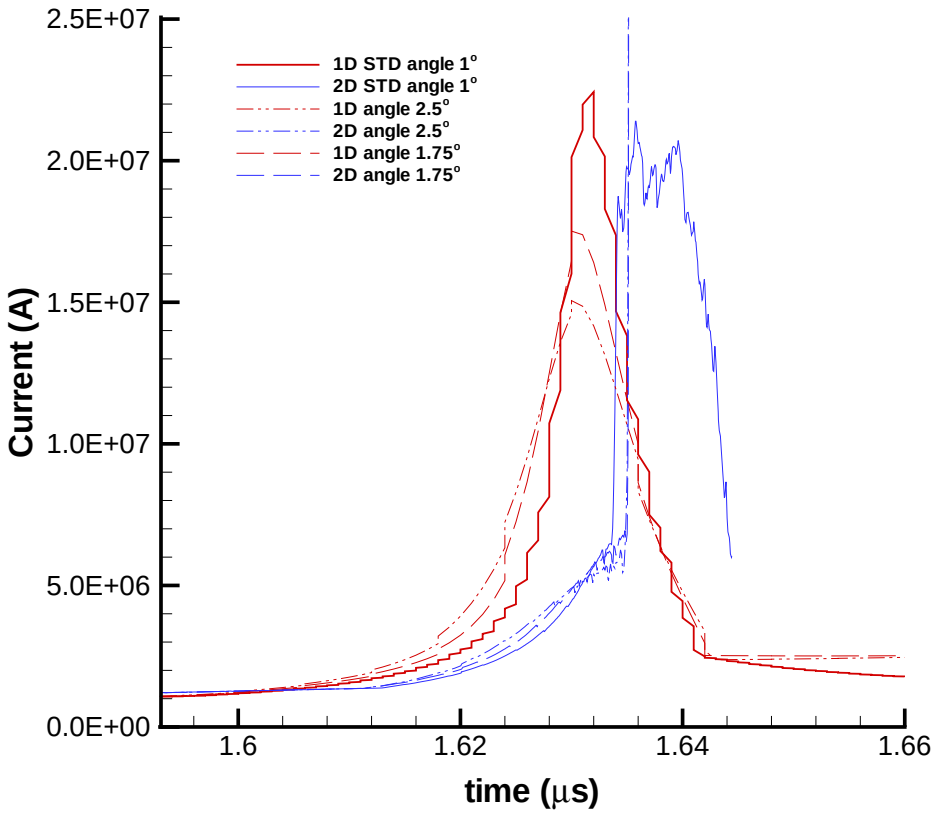

Figure 5.38: 1D and 2D secondary current calculations for various conical stator angles. Pulse broadening is clearly evident in 1D simulations which leads to lower current levels when the conical stator angle is increased. 2D simulations become unstable at t=1.3152 μs due to circulation currents formed on the inner tips of the stator. As a result, pulse broadening in 2D simulations cannot be determined.

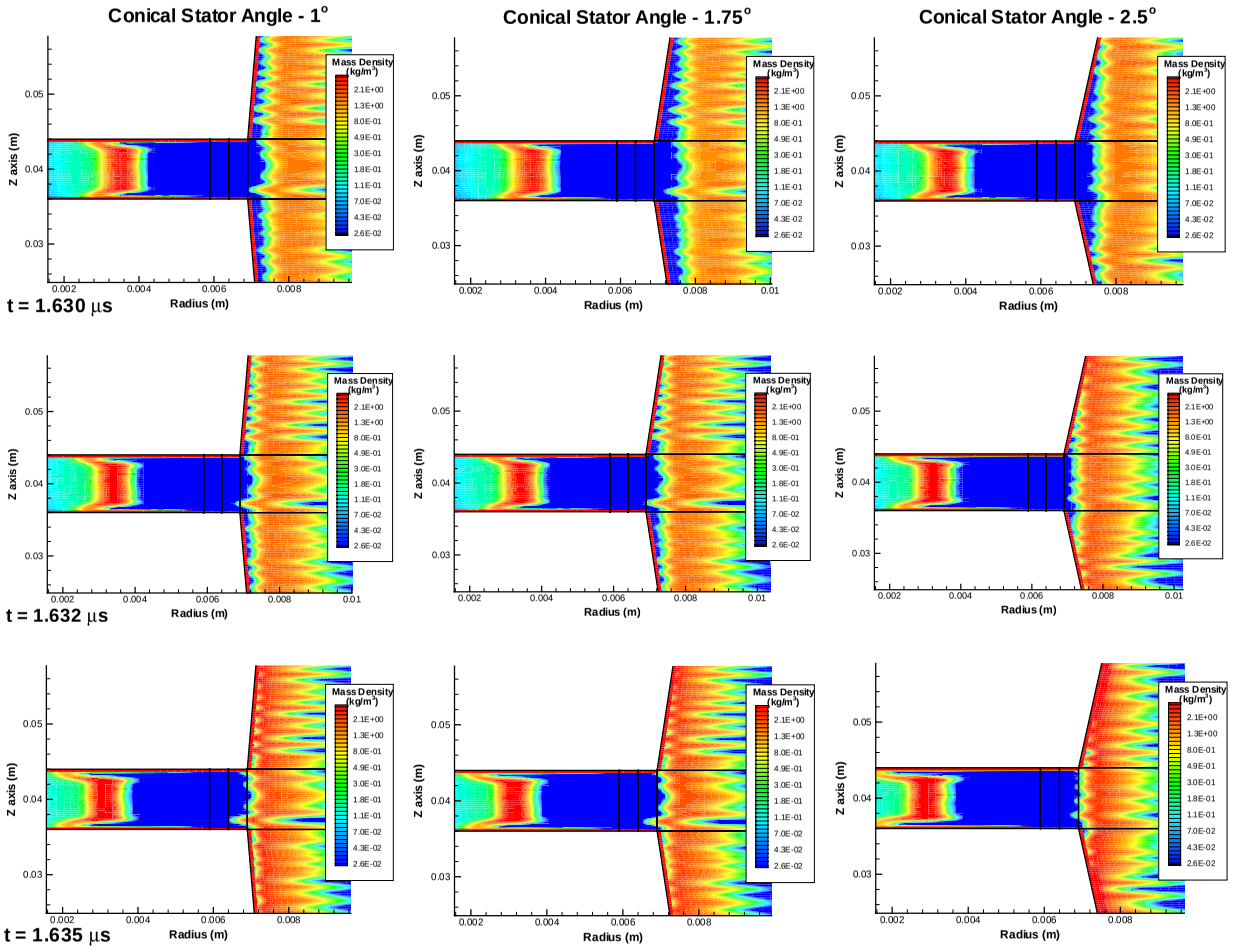

Figure 5.39: Density contour comparison of various stator angles. A 1% initial cell-to-cell density perturbation was used for these simulations.

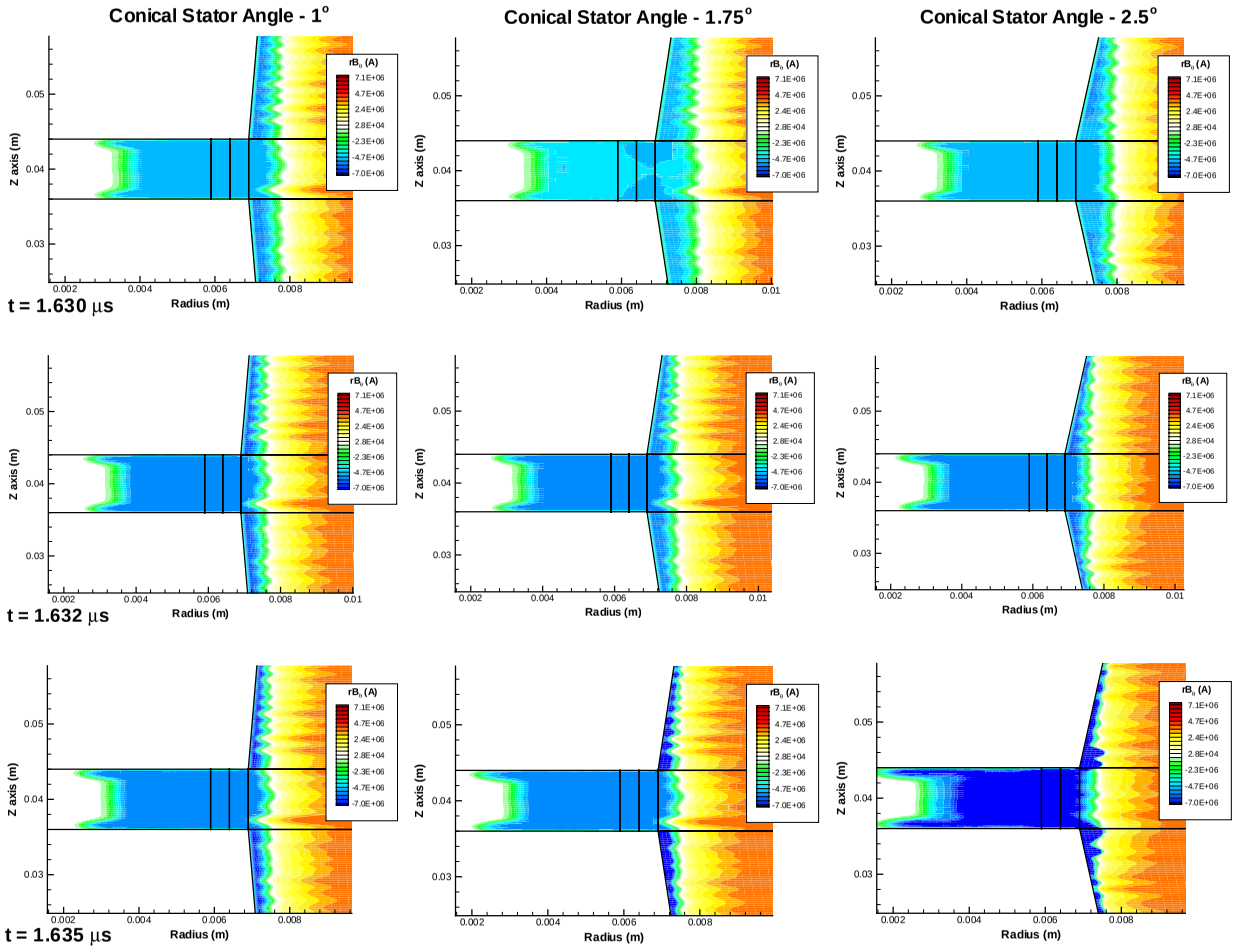

Figure 5.40: rB_θ contour comparison of various stator angles. A 1% initial cell-to-cell density perturbation was used for these simulations.

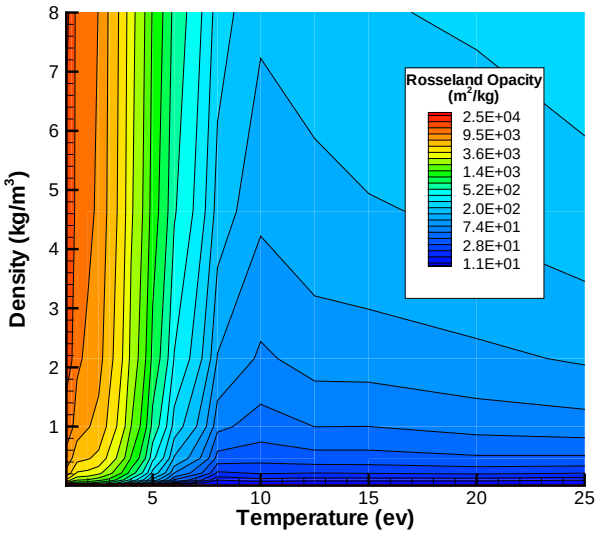

Figure 5.41: Al Rosseland opacity contours from SESAME database

heat, Ψ, from a cell according to

$$\Psi = ac\rho\chi T_e^4, \tag{5.3}$$

where a is Stephen's constant, c is the speed of light, χ is the mean Planck opacity, ρ is the mass density, and T_e is the electron temperature. Z-pinch plasmas are generally optically thick during plasma formation and stagnation, and are only approximately optically thin during the run-in phase. However, if a good estimate of the radiative cooling can be estimated with the simple emission model, it would be extremely useful in regards to simulating radiation energy transport. The radiation emission model adds little to computation time, whereas more complicated models like the nonequilibrium diffusion model significantly increase computation time.

In order to determine the validity of the radiation emission model, Planck and Rosseland mean free path lengths were compared to radial thickness of the plasma column. Figures 5.41 and 5.42 show SESAME Rosseland and Planck opacity contours for plasma temperatures and densities of interest during the run-in phase of the implosion. Rosseland mean free paths are on the order of the plasma radius for the majority of the run-in phase as shown in Figure 5.43 while Planckian mean free paths shown in Figure 5.44 are approximately an order of magnitude lower. As mentioned previously, MACH2's radiation emission model uses Planckian opacities and we therefore expect the optically thin plasma approximation to be only somewhat valid.

A better choice for a radiation model is MACH2's non-equilibrium radiation diffusion model based on both Planck and Rosseland opacities. However, this model

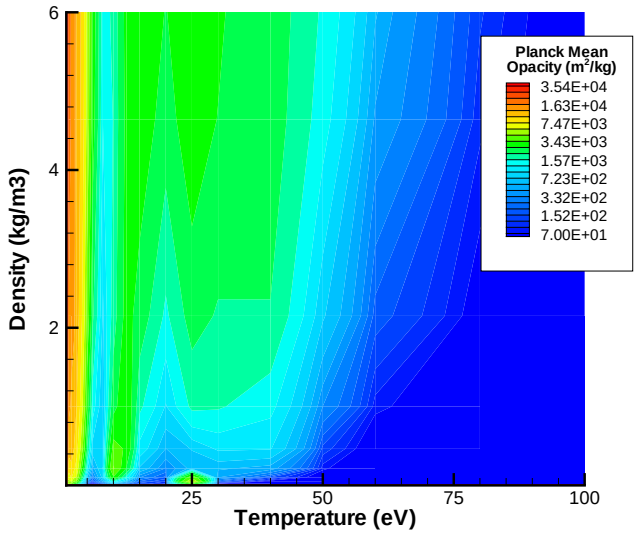

Figure 5.42: Al Planckian opacity contours from SESAME database

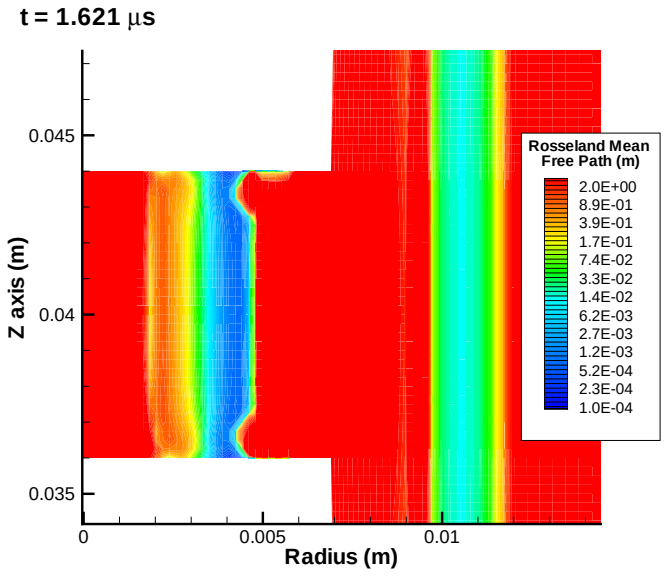

Figure 5.43: Rosseland mean free path

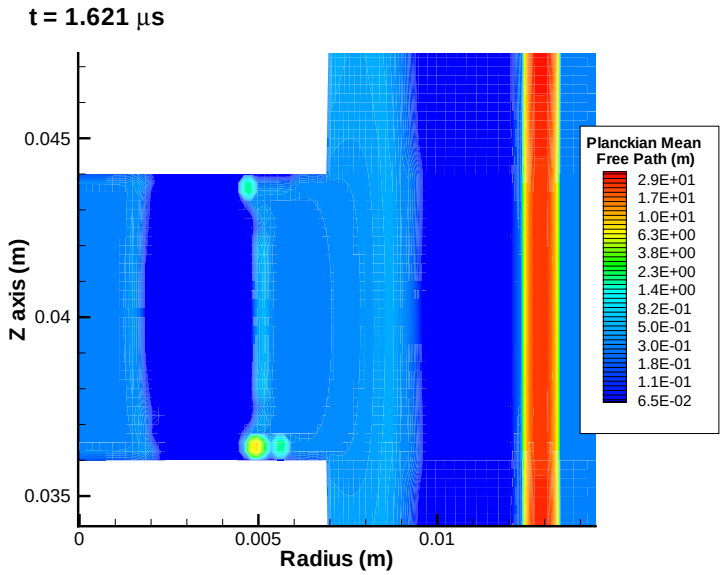

Figure 5.44: Planckian mean free path

significantly increases computational time and was found to be somewhat numerically unstable for flux compression simulations. Despite this, several simulations were completed using this model.

Figure 5.45 shows the calculated electron, ion, and radiation temperature contours. Ion and electron temperatures are such that the electrons and ions can be considered to be in thermal equilibrium during the run-in phase. The temperature of the radiation field in the Al and Ti plasmas is slightly lower than electron and ion temperatures. As the Al plasma nears the stator, the radiation temperature remains relatively constant as the electron and ion temperatures rise. Electron temperature contour comparisons for simulations with no radiation calculation and with MACH2's nonequilibrium diffusion model are shown in Figure 5.46. Some radiation cooling is evident although it is not significant to alter implosion dynamics. Current and magnetic flux profiles, shown in Figures 5.47 and 5.48, are nearly identical for simulations with and without radiation calculations.

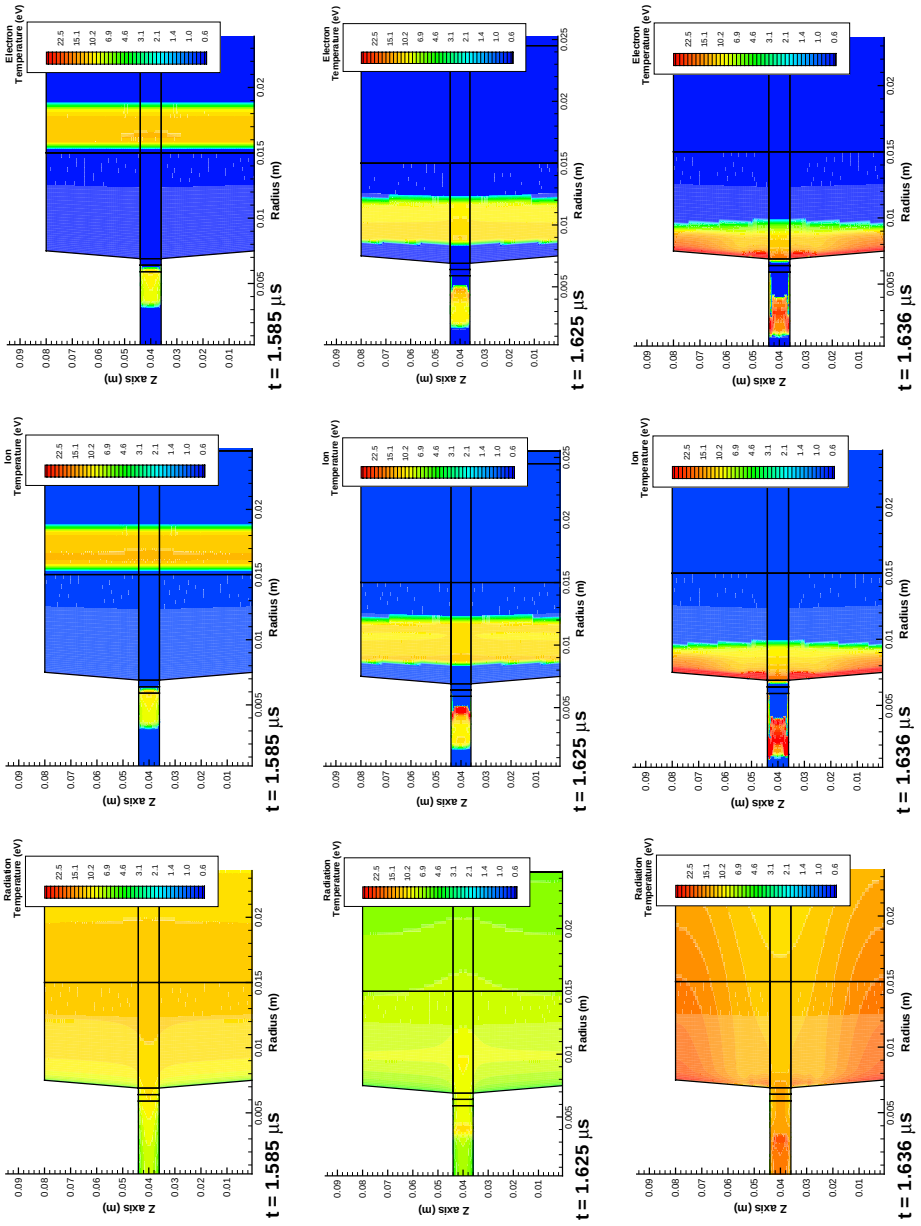

Figure 5.45: Electron, ion and radiation temperatures using MACH2's single group nonequilibrium diffusion model.

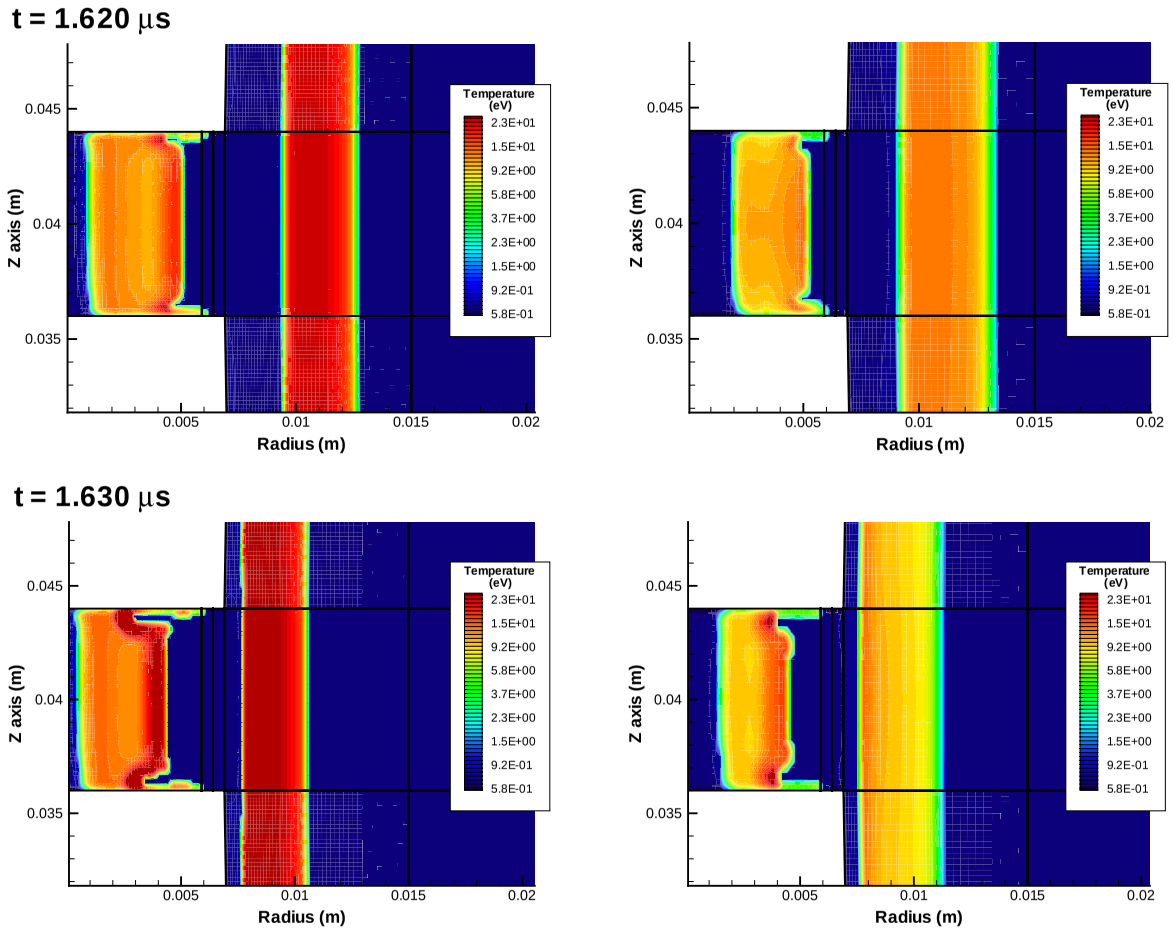

Figure 5.46: Electron temperature contours for identical simulation with and without MACH2's nonequilibrium radiation diffusion model.

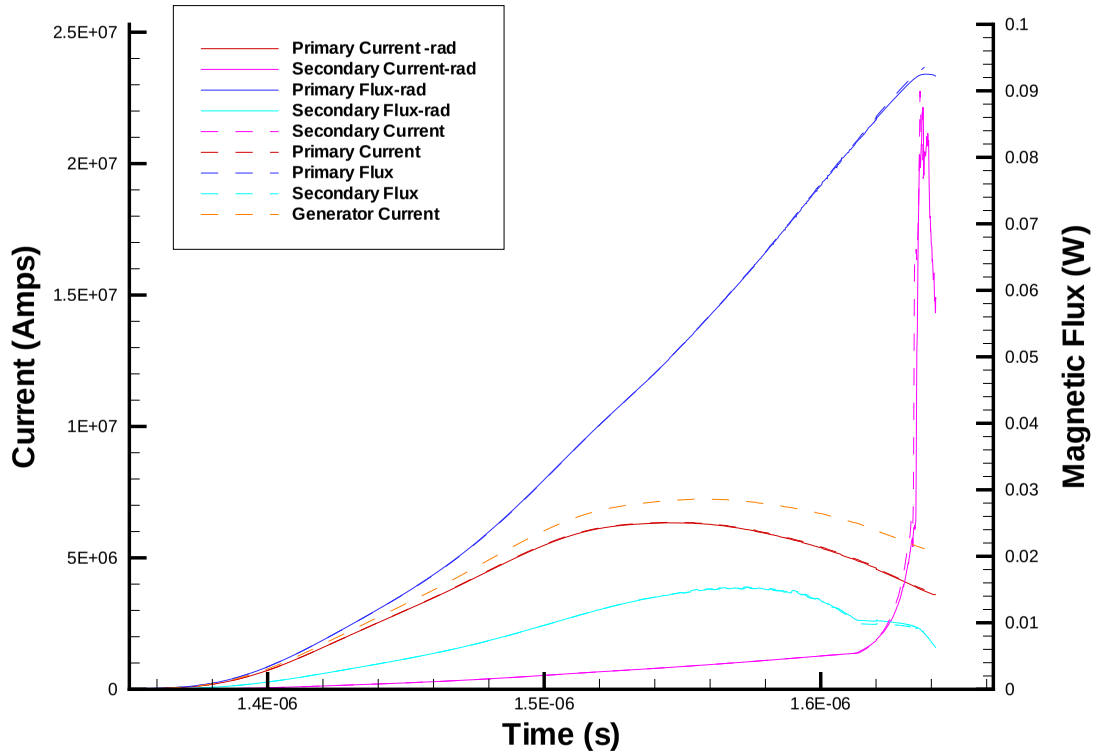

Figure 5.47: Boundary Condition current and magnetic flux evolution in identical simulations with and without MACH2's nonequilibrium radiation diffusion model.

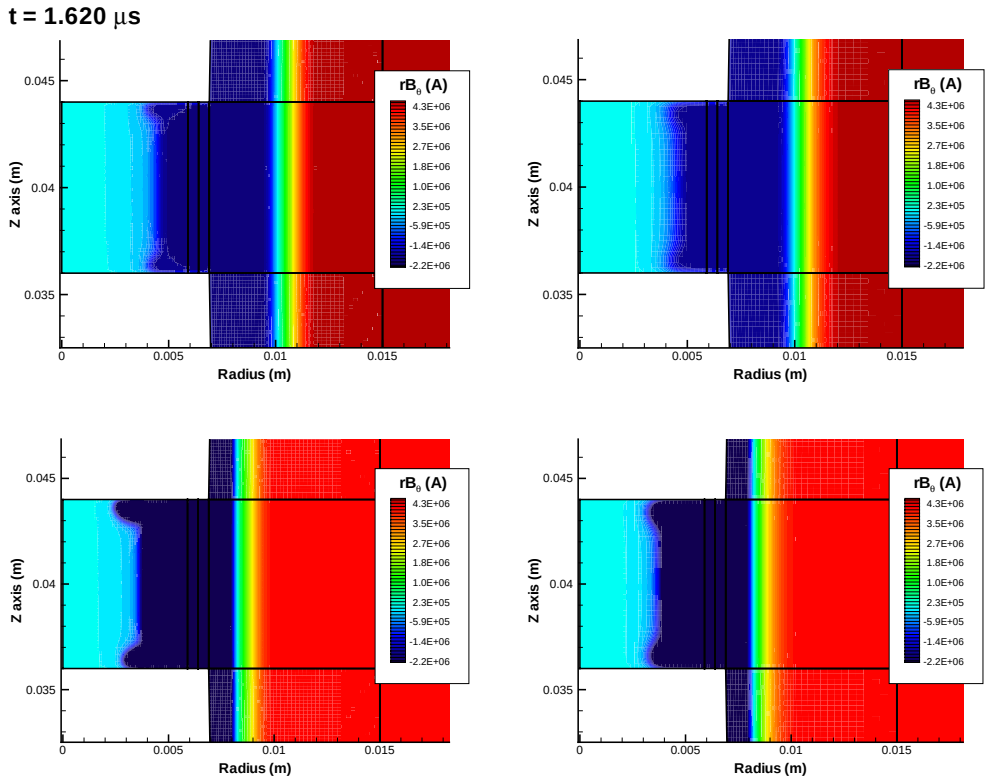

Figure 5.48: rB_θ current contours for identical simulation with and without MACH2's nonequilibrium radiation diffusion model.

Chapter 6

Conclusions

6.1 MACH2 Modifications

For this work, new circuit and isolated flux tracking routines have been successfully integrated into the MACH2 framework. The new routines have provided the capability for self-consistent current boundary conditions in Z-pinch flux compression simulations. In addition, a method of incorporating resistive and inductive impedance feedback into an external circuit that is numerically stable has been developed. The scope of applicability of the new routines is not limited to flux compression, but rather the new routines could be applied to any MHD simulation in which multiple current paths exist and for which recursive calculation of inductance is needed. One such example would be standard nested wire array Z-pinch simulations.

The ability to determine inductance self-consistently is extremely important when plasma instabilities are present, or when plasma flow geometries are complex and it is necessary to predict the current into a load. Up to this point, it appears there has not been any MHD simulations of nested wire arrays or flux compression that has determined currents self-consistently from a single energy source. Instead, all previous simulations have some form of prescribed current or voltage waveforms from experimental data.

6.2 Decade Quad Flux Compression Design

The data from the simulations presented in this research suggest the flux compression design studied would not yield efficient K-shell production of Ti or any other high AN element. Current distribution along the conical stator nearly match 1D predictions. However, only a small fraction of the stator current actually flows through the Ti inner wire array. Even without any initial density perturbations, that correspond to a ideal imploding plasma shell with no plasma instabilities, the Al plasma shorts out the flux compression process at the inner tips of the stator. As a result, the plasma effectively traps and isolates magnetic flux into two or more

separate regions. The majority of the magnetic flux lies between the stator and the Al plasma, and thus energy transfer to the Ti plasma is limited. Only a relatively small portion of the magnetic flux gets trapped between the Ti and Al plasmas. Magnetic flux compression and peak power is significantly reduced in the area of importance.

A tiered flux compression design similar to the flux compression experiments on the Z at SNL[16] or ECF2[17] would most likely prevent shorting out of the flux compression on the stator. This design has some drawbacks, but could produce a much higher current in the inner Ti wire array.

6.3 Flux Compression Physics

Perhaps the most important finding in this research is the predicted formation of magnetic bubbles that circulate current in an eddy-current-like-fashion along the stator. These bubbles rob the magnetic flux from the compression process and dissipate large amounts of magnetic energy in the form of joule heat.

Although an increase in the conical stator angle reduces the capable peak power, it allows more magnetic flux to be compressed along the Ti plasma. As the conical angle increases, it takes more time for the Al plasma to short out the stator. This allows a better current distribution in the Ti plasma and, consequently, a better energy transfer. The simulations presented in this research demonstrate an angle of 2.5 degrees allows 35% higher currents in the Ti plasma than the original design of 1 degree.

The predicted dynamics of the Al plasma armature were somewhat unexpected near peak compression. Due to the plasma armature shorting out on the stator, the magnetic pressures between the Al and Ti plasmas do not build high enough to produce a plasma turnaround. Therefore, the central section of the Al plasma armature continues to implode towards the z-axis after peak compression. The upper and lower ends of the the Al plasma do have a plasma turnaround. The magnetic pressure is highest at the inner tips of the stator, and the Al plasma in that region begins to move away from the z-axis. The Al plasma separates and axial symmetry is destroyed.

6.4 Thin Shell Approximation

Wire array implosion physics is three dimensional in nature, which requires 3D MHD codes for a comprehensive model of implosion dynamics. This is currently a very active area of research. With the advances in computational speed and cluster computing, a much better understanding of wire array implosions will be gained in the near future. In the meantime, a great deal of insight can be achieved using 1D and 2D simulations as demonstrated in this research.

One particular area of concern, in regards to the model assumptions, is that of thin plasma shells. While the complex nature and physics of wire array implo-

sions are still not completely understood, a substantial amount of evidence has been mounting suggesting that individual wires ablate away relatively slowly, forming precursor plasma streamers that extend to the axis of the pinch. This picture is substantially different than the thin shell approximation, in which no plasma precursors exist and all of the wire mass participates in the implosion. However, with a sufficient number of wires, i.e. small interwire gap spacing, the implosion should be relatively uniform and the thin shell model works well as a first order approximation. The major issue that needs to be explored is how a plasma precursor changes the physics of flux compression. A paramount issue is what the conductivity and magnetic Reynolds number of the precursor plasma would be. In order to estimate these parameters, the precursor plasma density, and even plasma composition, must be determined. There is no doubt a significant amount of experimental investigation is still needed in this area.

Despite the concerns of the plasma shell assumption, flux compression experiments have produced large amounts of current that have matched reasonably well with simulations making plasma shell assumptions[17]. Potential problems of premature gap closure and large diffusion loses that could be magnified by the presence of a precursor plasma do not significantly affect experimental current distributions.

6.5 Experiments

Shot #496 on the DQ was to provide a proof of principle test of the flux compression technique. Instead of a nested wire array configuration, the inner Ti wire array and conical stator were replaced with a stainless steel rod. The purpose of the experiment was to test whether flux compression could be achieved at all.

The shot was fielded at a time when the DQ was first testing the PRS configuration and still having operational problems, leading to unacceptable amounts of systematic error. In addition, only a limited amount of diagnostics were used. Figure 6.1 shows the anode feed current results, taken from the average of 4 b-dot probes. The current is approximately 30% lower than expected. The most likely hypothesis is that an electrical problem occurred with the DQ during the shot. Therefore, the data from flux compression Shot #496 can support no conclusions. Little insight was gained about flux compression during this experiment and the results have been labeled inconclusive.

It was hoped that several flux compression shots, including the nested shots model in this research, would have been performed on the DQ by the time of this writing. Unfortunately, due to changes in funding priority of the Defense Threat Reduction Agency (DTRA), no further flux compression shots have been fielded on the DQ. Obviously, without experimental data, the validity of the model, as to whether it actually models reality cannot be determined. Due to the numerous complex and intertwined physics involved in this problem, the results presented in this research may be received with skepticism. While justified, it is the author's hope that future flux compression experiments will be eventually fielded to compare with simulation results.

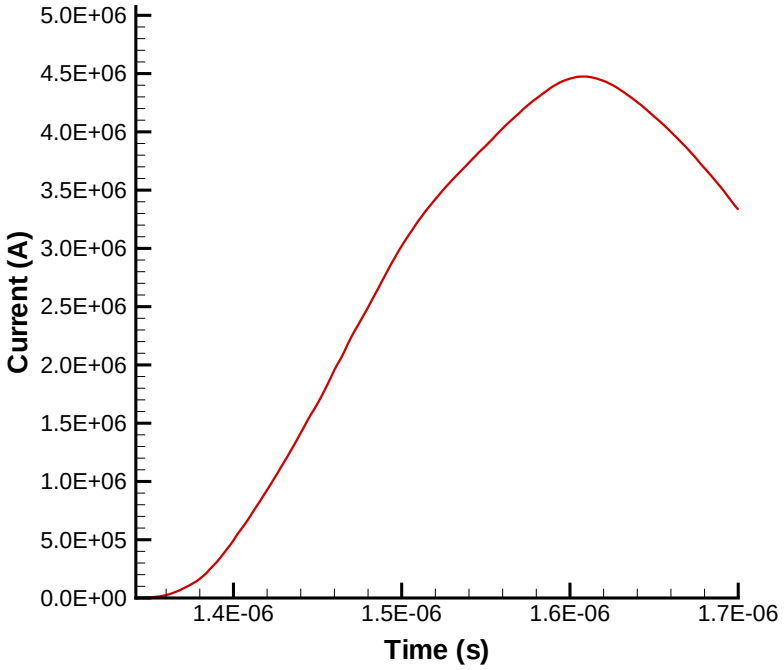

Figure 6.1: Current profile from flux compression shot #496. The current value is obtained by averaging integrated data from 4 b-dot probes located on the anode current feed. Peak current is approximately 30% lower than expected.

6.6 Suggestions for Future Work

Numerous issues could be investigated in future work. A detailed study of the initial wire explosion would be extremely useful in these simulations. 2D wire initiation simulations in the $R - \theta$ plane could be used as input conditions to the 2D R-Z plane simulations presented here. This would help eliminate initial plasma density, temperature, and timing issues.

The Rayleigh Taylor instabilities in this problem could be studied in much greater depth. A more complete and detailed Fourier analysis of the RT modes would be helpful in quantifing growth rates. At this point, growth rates could be compared to analytical models of RT instability growth. Of particular interest is whether or not flux compression mitigates RT instabilities in low L nested wire array Z-pinch implosions as proposed. While significantly more study is needed, the data presented here suggests minimal mitigation of RT instabilities in flux compression schemes. A quantitative study of flux compression ratio to RT growth rates should clarify the issue.

Finally, as MHD codes and computers evolve, a comprehensive 3D simulation of the flux compression scheme could shed new light on implosion dynamics. Also, the addition of advanced radiation models, such as multi-group non LTE radiation transport models, would enable a much better understanding of the radiation physics involved in this problem.

Bibliography

Bibliography

[1] J. E. Bailey, G. A. Chandler, D. Cohen, M. E. Cuneo, M. E. Ford, R. F. Heeter, D. Jobe, P. W. Lake, J. J. MacFarlane, T. J. Nash, D. S. Nielson, and R. Smelser. Radiation Science using Z-Pinch X-rays. *Phys. Plasmas*, 9(5):2186–2194, May 2002.

[2] Sandia National Laboratories Website. http://www.sandia.gov/pulspowr/facilities/zaccelerator.html.

[3] P. Sincerny, S. Ashby, K. Childers, C. Deeney, D. Drury, J. Goyer, D. Kortbawi, I. Roth, and C. Stallings. The Decade High Power Generator. In *Digest of Technical Papers 1987-1995 IEEE International Pulsed Power Conferences*, pages 880–883. IEEE, 1993.

[4] P. Sincerny, S. Ashby, K. Childers, J. Goyer, D. Kortbawi, I. Roth, C. Stallings, and J. Dempsey. Performance of Decade Module 1 (DM1) and the Status of the Decade MACHine. In *Digest of Technical Papers 1987-1995 IEEE International Pulsed Power Conferences*, pages 405–416. IEEE, 1995.

[5] M. G. Haines, S. V. Lebedev, J. P. Chittenden, F. N. Beg, S. N. Bland, and A. E. Dangor. The Past, Present and Future of Z Pinches. *Phys. Plasmas*, 7(5):1672–1680, May 2000.

[6] D. D. Ryutov, M. S. Derzon, and M. K. Matzen. The Physics of Fast Z Pinches. *Rev. Mod. Phys.*, 72(1):167–223, Jan 2000.

[7] S. A. Slutz, M. R. Douglas, J. S. Lash, R. A. Vesey, G. A. Chandler, T. J. Nash, and M. S. Derzon. Scaling and optimization of the radiation temperature in dynamic hohlraums. *Phys. Plasmas*, 8(5):1673–1691, May 2001.

[8] M. E. Cuneo, R. A. Vesey, Jr. J. L. Porter, G. A. Chandler, D. L. Fehl, T. L. Gilliland, D. L. Hanson, J. S. McGurn, P. G. Reynolds, L. E. Ruggles, H. Seamen, R. B. Spielman, K. W. Struve, W. A. Stygar, W. W. Simpson, J. A. Torres, D. F. Wenger, J. H. Hammer, P. W. Rambo, D. L. Peterson, and G. C. Idzorek. Development and characterization of a Z-pinch driven hohlraum high-yield inertial confinement fusion target concept. *Phys. Plasmas*, 8(5):2257–2267, May 2001.

[9] Stephen A. Slutz and Mark C. Herrmann. Radiation driven capsules for fast ignition fusion. *Phys. Plasmas*, 10(1):224–240, Jan 2003.

[10] M. A. Liberman, J. S. DeGroot, A. Toor, and R. B. Spielman. *Physics of High-Density Z-Pinch Plasmas*. Springer, 1999.

[11] C. Deeney, M. R. Douglas, R. B. Spielman, T. J. Nash, D. L. Peterson, P. L. Eplattenier, and G. A. Chandler. Enhancement of X-Ray Power from a Z Pinch Using Nested-Wire Arrays. *Phys. Rev. Lett.*, 81(22):4883–4886, Nov 1998.

[12] C. Deeney, C. A. Coverdale, M. R. Douglas, T. J. Nash, R. B. Spielman, K. G. Whitney, J. W. Thornhill, J. P. Apruzese, R. W. Clark, J. Davis, F. N. Beg, and J. Ruiz-CaMACHo. Titanium K-shell X-ray Production from High Velocity Wire Array Implosions on the 20-MA Z Accelerator. *Phys. Plasmas*, 6(5):2081–2088, May 1999.

[13] J. W. Thornhill, K. G. Whitney, J. Davis, and J. P. Apruzese. Investigation of K-shell Emission from Moderate-Z, low-η (-velocity), Z-pinch Implosions. *J. Appl. Phys.*, 80(2):710–718, Jul 1996.

[14] Y. H. Matsuda, F. Herlach, S. Ikeda, and N. Miura. Generation of 600 T by electromagnetic flux compression with improved implosion symmetry. *Rev. Sci. Instrum.*, 73(12):4288–4294, Dec 2002.

[15] R. K. Appartaim. Large Magnetic Fields Generated by Z-pinch Flux Compression. *J. Appl. Phys.*, 84(8):4170–4175, Oct 1988.

[16] J.F. Leon, R. B. Spielman, J. R. Asay, C. A. Hall, W. A. Stygar, and P. L'Eplattenier. Flux Compression Experiments on the Z Accelerator. In *1999 IEEE Internation Plasma Science Conference Proceedings*, pages 275–278. IEEE, 1999.

[17] P. L'Eplattenier, F. Lassalle, C. Mangeant, F. Hamann, M. Bavay, F. Bayol, D. Huet, A. Morell, P. Monjaux, G. Avrillaud, and B. Lalle. ECF2: A pulsed power generator based on magnetic flux compression for K-shell radiation production. In *Dense Z-Pinches 5th International Conference* , pages 51–54. AIP, 2002.

[18] B A. Boyko, A. I. Bykov, M. I. Dolotenko, N. P. Kolokolchikov, I. M. Markevtsev, O. M. Tatsenko, and K. Shuvalov. With record magnetic fields to the 21st Century . In *Pulsed Power Conference 1999 Digest of Technical Papers*, pages 746–749. IEEE, 1999.

[19] M. Lehr, L. Bamert, K. Bell, T. Cavazos, D. Chama, S. Coffey, J. Degnan, D. Gale, G. Kiuttu, P. Pellitier, and W. Sommars. Helical explosive flux compression generator research at the Air Force research laboratory . In *Pulsed Power Plasma Science 2001 Digest of Technical Papers* , pages 339–342. IEEE, 2001.

[20] M. Bavay, P. L'Eplattenier, C. Mangeant, F. Hamann, M. Mazarakis, Ph. Monjaux, F. Lassalle, F. Bayol, D. Huet, G. Avrillaud, and B. Lalle. The magnetic flux compression scheme as a power amplification and pulse shaping stage . In *International Conference on Plasma Science 2002 IEEE Conference Record* , pages 343–346. IEEE, 2002.

[21] Robert E. Peterkin Jr., Michael H. Frese, and Carl R. Sovinec. Transport of Magnetic Flux in an Arbitrary Coordinate ALE Code. *J. Comp. Phys.*, 140(1):148–171, Feb 1998.

[22] Nino R. Pereira, Jack Davis, and Peter E. Pulsifer, editors. *Dense Z-Pinches, Fourth International Conference*. AIP, American Institute of Physics, 1997. AIP Conference Proceedings 409.

[23] Nino R. Pereira, Jack Davis, and Christopher Deeney, editors. *Dense Z-Pinches, Fifth International Conference*. AIP, American Institute of Physics, 2002. AIP Conference Proceedings 651.

[24] T. W. L. Sanford, G. O. Allshouse, B. M. Marder, T. J. Nash, R. C. Mock, R. B. Spielman, J. F. Seaman, J. S. McGurn, D. Jobe, T. L. Gilliland, M. Vargas, K. W. Struve, W. A. Stygar, M. R. Douglas, M. K. Matzen, J. H. Hammer, J. S. De Groot, J. L. Eddleman, D. L. Peterson, D. Mosher, K. G. Whitney, J. W. Thornhill, P. E. Pulsifer, J. P. Apruzese, and Y. Maron. The Different Dynamical Modes of Nested Wire Array Z-pinches. *Phys. Plasmas*, 8(3):675–678, Mar 2001.

[25] S. V. Lebedev, R. Aliaga-Rossel, S. N. Bland, J. P. Chittenden, A. E. Dangor, M. G. Haines, and M. Zakaullah. Two Different Modes of Nested Wire Array Z-Pinch Implosions. *Phys. Rev. Lett.*, 84(8):1708–1711, Feb 2000.

[26] M. P. Desjarlais and B. M. Marder. Theory of Wire Number Scaling in Wire-Array Z Pinches. *Phys. Plasmas*, 6(5):2057–2064, May 1999.

[27] D. L. Peterson, R. L. Bowers, J. H. Brownell, A. E. Grenne, K. D. McLenithan, T. A. Oliphant, N. F. Roderick, and A. J. Scannapieco. Two-dimensional Modeling of Magnetically Driven Rayleigh-Taylor Instabilities in Cyclindrical Z Pinches. *Phys. Plasmas*, 3(1):368–381, Jan 1996.

[28] Ferdinand R. Cap. *Handbook on Plasma Instabilities*. Academic Press, Inc., 1976.

[29] M. R. Douglas, C. Deeney, R. B. Spielman, C. A. Coverdale, N. F. Roderick, and M. G. Haines. Long Implosion Time Tungsten Wire Array Z Pinches on the Saturn Generator. *Phys. Plasmas*, 7(7):2945–2958, Jul 2000.

[30] S. A. Pikuz, T. A. Shelkovenko, D. B. Sinars, K. M. Chandler, D. A. Hammer, G. V. Ivanenkov, W. Stepniewski, and I. Yu. Skobelev. High Energy Deposition

Z-Pinch Plasma Conditions with Picosecond Time Resolution. *Phys. Rev. Lett.*, 89(3):350031–350034, Jul 2002.

[31] M. R. Douglas, C. Deeney, and N. F. Roderick. Computational Investigation of Single Mode vs Multimode Rayleigh-Taylor Seeding in Z-pinch Implosions . *Phys. Plasmas*, 5(12):4183–4198, Dec 1998.

[32] M. R. Douglas, C. Deeney, and N. F. Roderick. The Effect of Load Thickness on the Performance of High Velocity, Annular Z-pinch Implosions. *Phys. Plasmas*, 8(1):238–248, Jan 2001.

[33] M. R. Douglas, C. Deeney, R. B. Spielman, C. A. Coverdale, N. F. Roderick, and D. L. Peterson. Implosion Dynamics of Long-pulse Wire Array Z Pinches. *Phys. Plasmas*, 7(5):1935–1944, May 2000.

[34] W. Matuska, R. L. Bowers, J. H. Brownell, H. Lee, C. M. Lund, D. L. Peterson, and N. F. Roderick. Two-dimensional Modeling of the X-radiation Output from Preturbed Z Pinches. *Phys. Plasmas*, 3(4):1415–1429, Apr 1996.

[35] R. Benattar, S. V. Zakharov, A. F. Nikiforov, V. G. Novikov, V. A. Gasilov, A. Yu. Krukovskii, and V. S. Zakharov. Influence of Magnetohydrodynamic Rayleigh-Taylor Instability on Radiation of Imploded Heavy Ion Plasmas. *Phys. Plasmas*, 6(1):175–187, Jan 1999.

[36] J. H. Hammer, J. L. Eddleman, P. T. Springer, M. Tabak, A. Toor, K. L. Wong, G. B. Zimmerman, C. Deeney, R. Humphreys, T. J. Nash, T. W. L. Sanford, and J. S. De Groot. Two-dimensional Radiation-magnetohydrodynamic Simulations of SATURN Imploding Z Pinches. *Phys. Plasmas*, 3(5):2063–2068, May 1996.

[37] T. W. L. Sanford, R. C. Mock, R. B. Spielman, M. G. Haines, J. P. Chittenden, K. G. Whitney, J. P. Apruzese, D. L. Peterson, J. B. Greenly, D. B. Sinars, D. B. Reisman, and D. Mosher. Wire Array Z-pinch Insights for Enhanced X-ray Production. *Phys. Plasmas*, 6(5):2030–2040, May 1999.

[38] D. L. Peterson, R. L. Bowers, K. D. McLenithan, C. Deeney, G. A. Chandler, R. B. Spielman, M. K. Matzen, and N. F. Roderick. Characterization of Energy Flow and Instability Development in Two-dimensional Simulations of Hollow Z Pinches . *Phys. Plasmas*, 5(9):3302–3310, Sept 1998.

[39] C. Wahlberg, H. G. Eriksson, and Z. X. Jiang. Nonlinear Dynamics of an Elliptic Magnetic Stagnation Line. *Phys. Plasmas*, 4(7):2397–2405, Jul 1997.

[40] M. G. Haines. A Heuristic Model of the Wire Array Z-Pinch. *IEEE Transactions on Plasma Science*, 26(4):1275–1281, Aug 1998.

[41] S. A. Pikuz, T. A. Shelkovenko, D. B. Sinars, J. B. Greenly, B. R. Kusse, Min Hu, K. M. Chandler, and D. A. Hammer. Experiments Measuring the Initial Energy Deposition, Expansion Rates, and Morphology of Exploding Wires with about 1 kA/wire. *Phys. Plasmas*, 8(1):216–230, Jan 2001.

[42] L. E. Aranchuk. *Sov. J. Plasma Phys.*, 12:765–, 1986.

[43] S. A. Pikuz, T. A. Shelkovenko, D. B. Sinars, J. B. Greenly, Y. S. Dimant, and D. A. Hammer. Multiphase Foamlike Structure of Exploding Wire Cores. *Phys. Rev. Lett.*, 83(21):4313–4316, Nov 1999.

[44] J. P. Chittenden, S. V. Lebedev, J. Ruiz-Camacho, F. N. Begs, S. N. Bland, C. A. Jennings, A. R. Bell, and M. G. Haines. Plasma formation in metallic wire Z pinches. *Phys. Rev. E*, 61(4):4370–4380, Apr 2000.

[45] S. V. Lebedev, F. N. Beg, J. P. Chittenden, A. E. Dangor, M. G. Haines, S. A. Pikuz, and T. A. Shelokovenko. Effect of Core-Corona Plasma Structure of Seeding of Instabilities in Wire Array Z Pinches. *Phys. Rev. Lett.*, 85(1):98–101, Jul 2000.

[46] J. P. Chittenden, S. V. Lebedev, A. R. Bell, R. Aliaga-Rossel, S. N. Bland, and M. G. Haines. Plasma formation and Implosion Structure in Wire Array Z Pinches. *Phys. Rev. Lett.*, 83(1):100–103, Jul 1999.

[47] S. V. Lebedev, F. N. Beg, S. N. Bland, J. P. Chittenden, A. E. Dangor, M. G. Haines, K. H. Kwek, S. A. Pikuz, and T. A. Shelkovenko. Effect of Discrete Wires on the Implosion Dynamics of Wire Array Z Pinches. *Phys. Plasmas.*, 8(8):3734–3747, Aug 2001.

[48] S. V. Lebedev, J. P. Chittenden, S. N. Bland, D. J. Ampleford, C. Jennings, and M. G. Haines. Effect of Discrete Wires on the Implosion Dynamics of Wire Array Z-Pinches. In *Dense Z-Pinches 5th International Conference*, pages 65–70. AIP, 2002.

[49] M. G. Haines, S. V. Lebedev, J. P. Chittenden, F. N. Beg, S. N. Bland, and M. Sherlock. Why do Wire-Array Z-Pinches Give Such a Sharp and Efficient X-Ray Pulse? In *Dense Z-Pinches 5th International Conference*, pages 345–349. AIP, 2002.

[50] J. P. Chittenden, S. V. Lebedev, S. N. Bland, F. N. Beg, and M. G. Haines. One-, two-, and three-dimensional modeling of the different phases of wire array Z-pinch evolution. *Phys. Plasmas*, 8(5):2305–2314, May 2001.

[51] J. P. Chittenden, R. Aliaga Rossel, S. V. Lebedev, I. H. Mitchell, M. Tatarakis, A. R. Bell, and M. G. Haines. Two-dimensional Magneto-hydrodynamic Modeling of Carbon Fiber Z-pinch Experiments. *Phys. Plasmas*, 4(12):4309–4317, Dec 1997.

[52] S. V. Lebedev, D. J. Ampleford, S. N. Bland, J. P. Chittenden, and M. G. Haines. Ablation Rate of Wire Cores in Wire Array Z-Pinch Experiments. In *Dense Z-Pinches 5th International Conference*, pages 71–74. AIP, 2002.

[53] Michael H. Frese, Sherry D. Frese, Stephen E. Rosenthal, Melissa R. Douglas, and Norman F. Roderick. Transport of Magnetic Flux in an Arbitrary Coordinate ALE Code. *IEEE Transactions on Plasma Science*, 30(2):593–603, Apr 2002.

[54] Michael H. Frese and Sherry D. Frese. Three Dimensional Resistive Wire Array Implosions Simulations Continued from Two Dimensional R-θ Initiation Simulations. In *Dense Z-Pinches 5th International Conference*, pages 376–379. AIP, 2002.

[55] S. N. Bland, S. V. Lebedev, F. N. Beg, H. Kwek, J. P. Chittenden, and M. G. Haines. The Effect of Array Configuration on Current Distribution in a Wire Array Z-Pinch. In *Dense Z-Pinches 5th International Conference*, pages 83–86. AIP, 2002.

[56] R. E. Terry, J. Davis, C. Deeney, and A. L. Velikovich. Current Switching and Mass Interpenetration Offer Enhanced Power from Nested-Array Z Pinches. *Phys. Rev. Lett.*, 83(21):4305–4308, Nov 1999.

[57] J. Davis, N. A. Gondarenko, and A. L. Velikovich. Fast Commutation of High Current in Double Wire Array Z-pinch Loads. *Appl. Phys. Lett.*, 70(2):170–172, Jan 1997.

[58] J. P. Chittenden, S. V. Lebedev, S. N. Bland, A. Ciardi, and M. G. Haines. The Different Dynamical Modes of Nested Wire Array Z-pinches. *Phys. Plasmas*, 8(3):675–678, Mar 2001.

[59] S. N. Bland, S. V. Lebedev, J. P. Chittenden, and M. G. Haines. Implosion Dynamics and X-ray Characteristics of Nested Wire Array Z-Pinches. In *Dense Z-Pinches 5th International Conference*, pages 75–78. AIP, 2002.

[60] J. Davis, J. L. Giuliani, J. P. Apruzese, R. W. Clark, J. W. Thornhill, K. G. Whitney, A. Velikovich, Y. K. Chong, C. A. Coverdale, C. Deeney, and P. D. LePell. Pitfalls in Radiation Modeling of Z-Pinch Plasmas. In *Dense Z-Pinches 5th International Conference*, pages 339–344. AIP, 2002.

[61] E. H. Holt and R. E. Haskell. *Foundations of Plama Dynamics*. The Macmillian Comoany, 1965.

[62] Richard Fitzpatrick. Assistant Professor of Physics, Plasma Physics Lecture Notes, Institute for Fusion Studies, University of Texas at Austin.

[63] R. Shankar. *Principles of Quantum Mechanics* . Plenum Press, 1994.

[64] M. Mitchner and Charles H. Kruger Jr. *Partially Ionized Gases*. John Wiley and Sons, Inc., 1992.

[65] Thomas W. L. Sanford, Norman F. Roderick, Raymond C. Mock, Kenneth W. Struve, and Darrel L. Peterson. Azimuthal Structure in Wire-Array Z Pinch Experiments. *IEEE Transactions on Plasma Science*, 30(2):538–546, Apr 2002.

[66] S. Maxon, J. H. Hammer, J. L. Eddleman, M. Tabak, G. B. Zimmerman, W. E. Alley, T. J. Nash, T. W. L. Sanford, and J. S. De Groot. Magnetohydrodynamic Solution for a Z Pinch Showing the Production of a Hot Spot . *Phys. Plasmas*, 3(5):1737–1740, May 1996.

[67] D. L. Peterson, R. L. Bowers, W. Matuska, K. D. McLenithan, G. A. Chandler, C. Deeney, M. S. Derzon, M. R. Douglas, M. K. Matzen, T. J. Nash, R. B. Spielman, K. W. Struve, W. A. Stygar, and N. F. Roderick. Insights and Applications of Two-dimensional Simulations to Z-pinch Experiments . *Phys. Plasmas*, 6(5):2178–2184, May 1999.

[68] B. M. Marder, T. W. L. Sanford, and G. O. Allshouse. Numerical Simulations of Annular Wire-array Z-pinches in (x,y),(r,θ) and (r,z) Geometries. *Phys. Plasmas*, 5(8):2997–3005, Aug 1998.

[69] John David Jackson. *Classical Electrodynamics - 3rd Edition* . John Wiley and Sons, Inc., 1999.

[70] Clint Hall. Isentropic compression experiments on the Sandia Z accelerator. *Phys. Plasmas*, 7(5):2069–2075, May 2000.

[71] F. S. Felber, M. A. Liberman, and A. L. Velikovich. Magnetic Flux Compression by Dynamic Plasmas. I. Subsonic self-similar Compression of a Magnetized Plasma-filled Liner. *Phys. Fluids*, 31(12):3675–3682, Dec 1988.

[72] F. S. Felber, M. A. Liberman, and A. L. Velikovich. Magnetic Flux Compression by Dynamic Plasmas. II. Supersonic self-similar solutions for Magnetic Cumulation. *Phys. Fluids*, 31(12):3683–3689, Dec 1988.

[73] M. Bavay. *Compression de flux magnétique dans le régime sub-microseconde pour l'obtention de hautes pressions et de rayonnement X intense*. PhD thesis, le Centre d'*É*tudes de Gramat (CEG), 2001.

[74] L. I. Rudakov and Eduardo M. Waismann. Current Doubler Flux compression device for power amplification in vacuum. In *1999 IEEE Internation Pulsed Power Conference Proceedings*, pages 1719–1722. IEEE, 1999.

[75] F. N. Beg, S. V. Lebedev, S. N. Bland, J. P. Chittenden, A. E. Dangor, and M. G. Haines. The Effect of Current Prepulse on Wire Array Z-Pinch Implosions. *Phys. Plasmas*, 9(1):375–377, Jan 2002.

[76] P. L'Eplattenier, M. Bavay, G. Avrillaud, and B. Lalle. 0D numerical modelisation and optimization of flux compression experiments . In *Pulsed Power Plasma Science 2001 Digest of Technical Papers* , pages 665–668. IEEE, 2001.

[77] John Goyer. Private Communication.

[78] J. S. Levine, M. A. Babineau, J. Banister, D. Bell, C. Enis, B. H. Failor, V. Kenyon, H. M. Sze, and T. Worley. Z-Pinch Experiments on Decade Quad Using Double-Shell Gas Puff. In *Dense Z-Pinches 5th International Conference*, pages 109–112. AIP, 2002.

[79] S. T. Pai and Qi Zhang. *Introduction to High Power Pulse Technology* . World Scientific, 1995.

[80] Sherry D. Frese and Michael H. Frese. Recent Improvements to MACH2 and MACH23 for Fast Z-Pinch Modeling. In *Dense Z-Pinches 5th International Conference*, pages 380–383. AIP, 2002.

[81] John D. Anderson Jr. *Compuational Fluid Dynamics*. McGraw-Hill Inc., 1995.

[82] Donald A. Calahan, Alan B. Macnee, and E. Lawrence McMahon. *Introduction to Modern Circuit Analysis*. Holt Rinehart and Winston Inc., 1974.

[83] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in Fortran 77. The Art of Scientific Computing Second Edition*. Cambridge University Press, 1992.

[84] EOS and Mechanics of Materials Group (T1) of the Theoretical Division at Los Alamos National Laboratory.

[85] M. P. Desjarlais. Practical Improvements to the Lee-More Conductivity Near the Metal-Instulator Transition. *Contrib. Plasmas Phys.*, 41(2-3):267–270, 2001.

[86] M. P. Desjarlais, J. D. Kress, and L. A. Collins. Electrical conductivity for warm, dense aluminum plasmas and liquids. *Phys. Rev. E*, 66, 2002.

[87] Y. T. Lee and R. M. More. An electron conductivity model for dense plasmas. *Phys. Fluids*, 27:1273–1286, 1984.

[88] E. G. Harris. Rayleigh-Taylor Instabilities of a Collapsing Cylinderical Shell in a Magnetic Field. *Phys. Fluids*, 5(9):1057–1062, Sep 1962.

[89] W. F. Noh and P. Woodward. SLIC (Simple Line Interface Calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Mechanics* . Springer Verlag, 1976.

[90] Kathleen S. Bonnell, Daniel R. Schikore, Kenneth I. Joy, Mark Duchaineau, and Bernd Hamann. Constructing Material Interfaces From Data Sets With Volume-Fraction Information. Center for Image Processing and Integrated Computing (CIPIC).

[91] Michael H. Frese. Private Communication.

Appendix

Appendix A

Contour Data for Selected Simulations

1% Initial Cell-to-Cell Density Perturbation
Density Contours
Magnetic Field Contours
Axial Current Density Contours
Temperature Contours

5% Initial Cell-to-Cell Density Perturbation
Density Contours
Magnetic Field Contours
Axial Current Density Contours
Temperature Contours

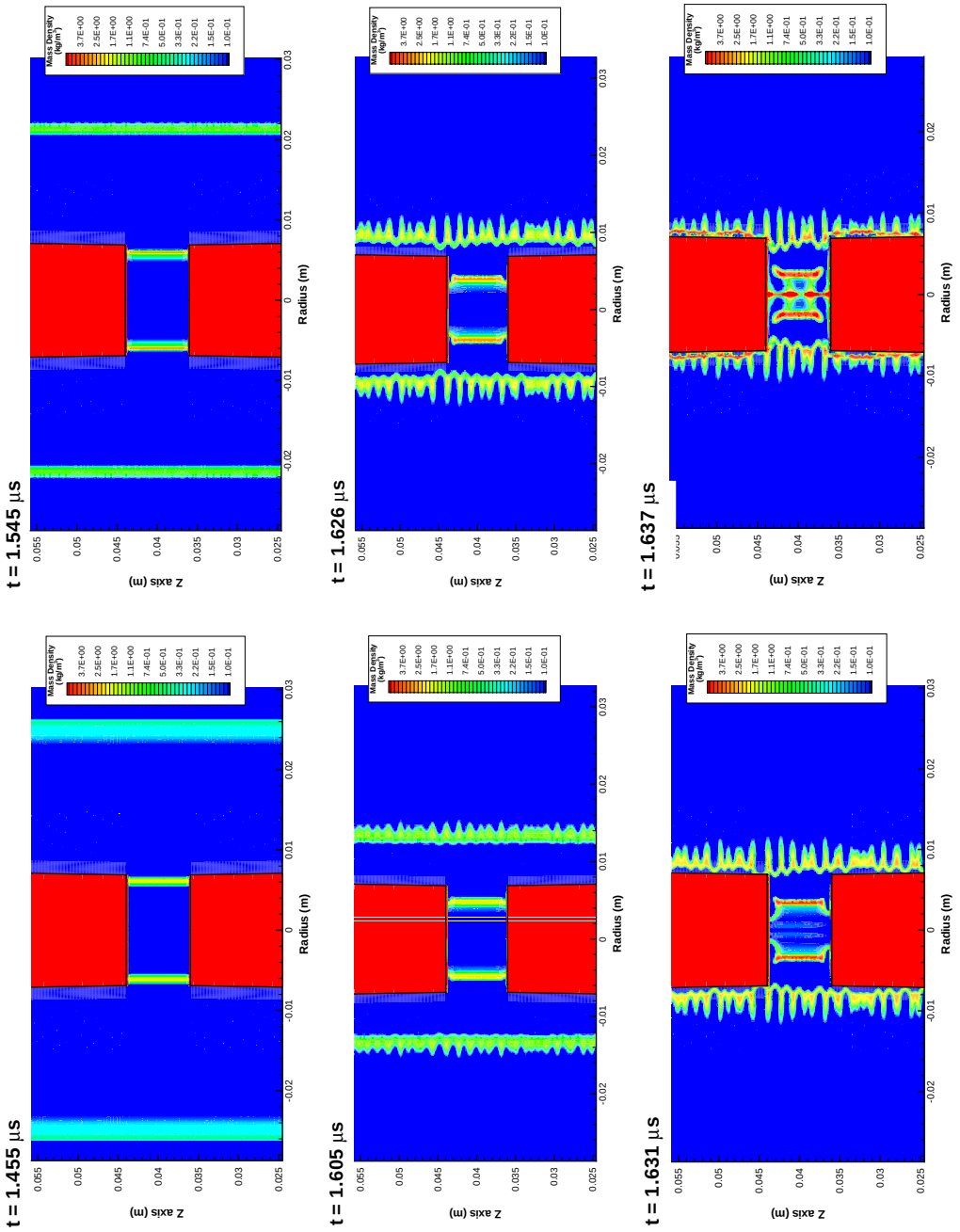

Figure 8.1: Mass density contour evolution with a 1% initial density perturbation.

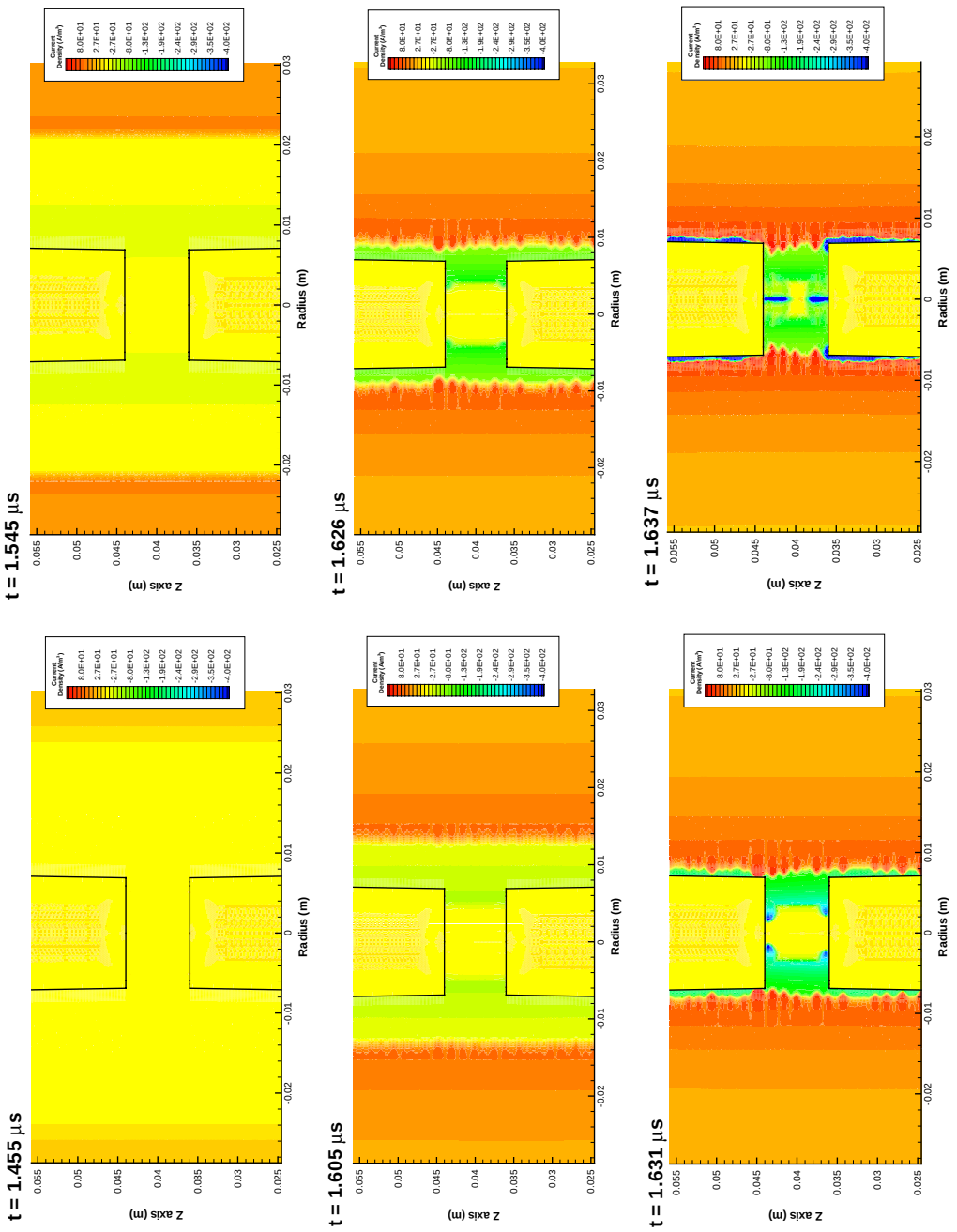

Figure 8.2: Magnetic field contour evolution with a 1% initial density perturbation.

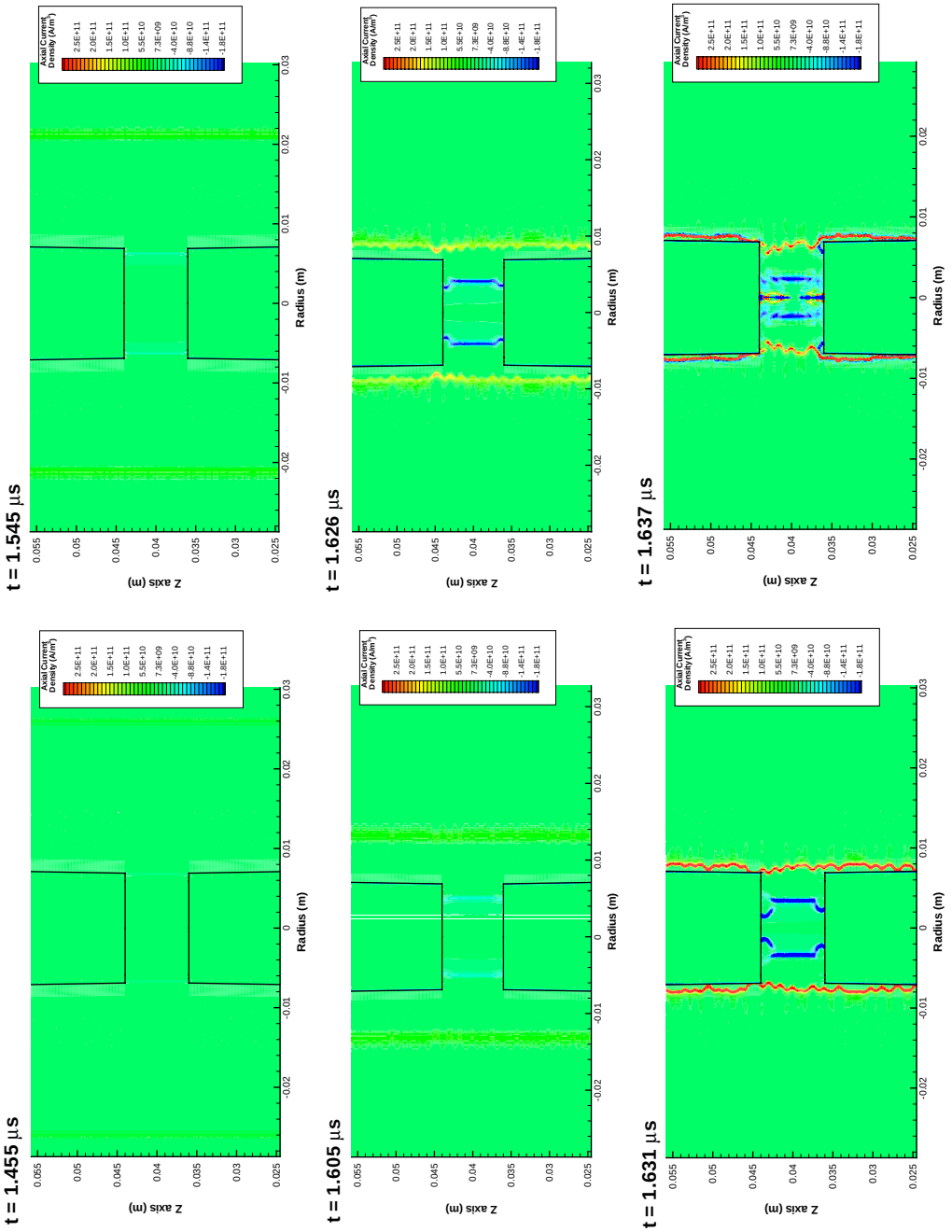

Figure 8.3: Axial current density contour evolution with a 1% initial density perturbation.

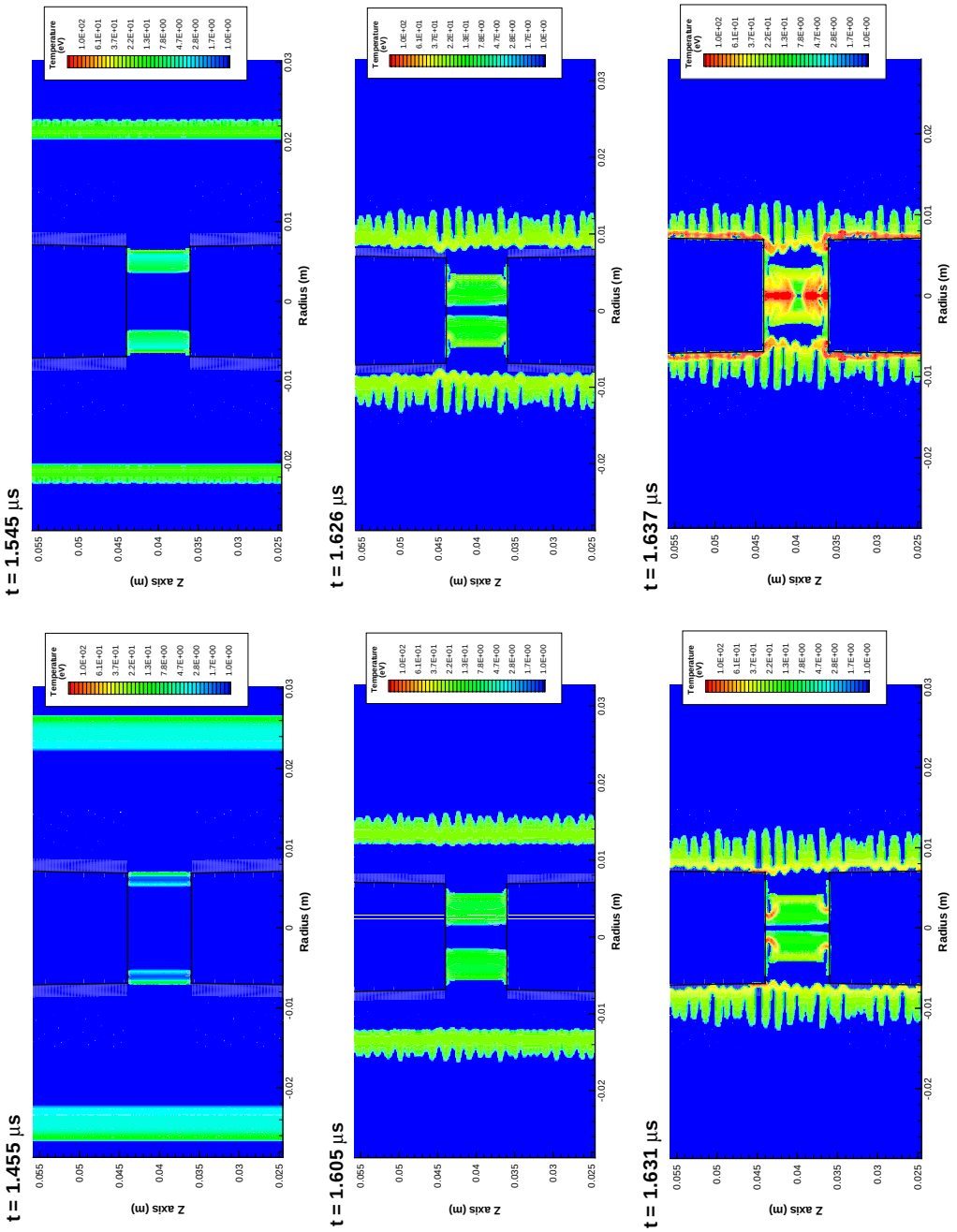

Figure 8.4: Temperature contour evolution with a 1% initial density perturbation.

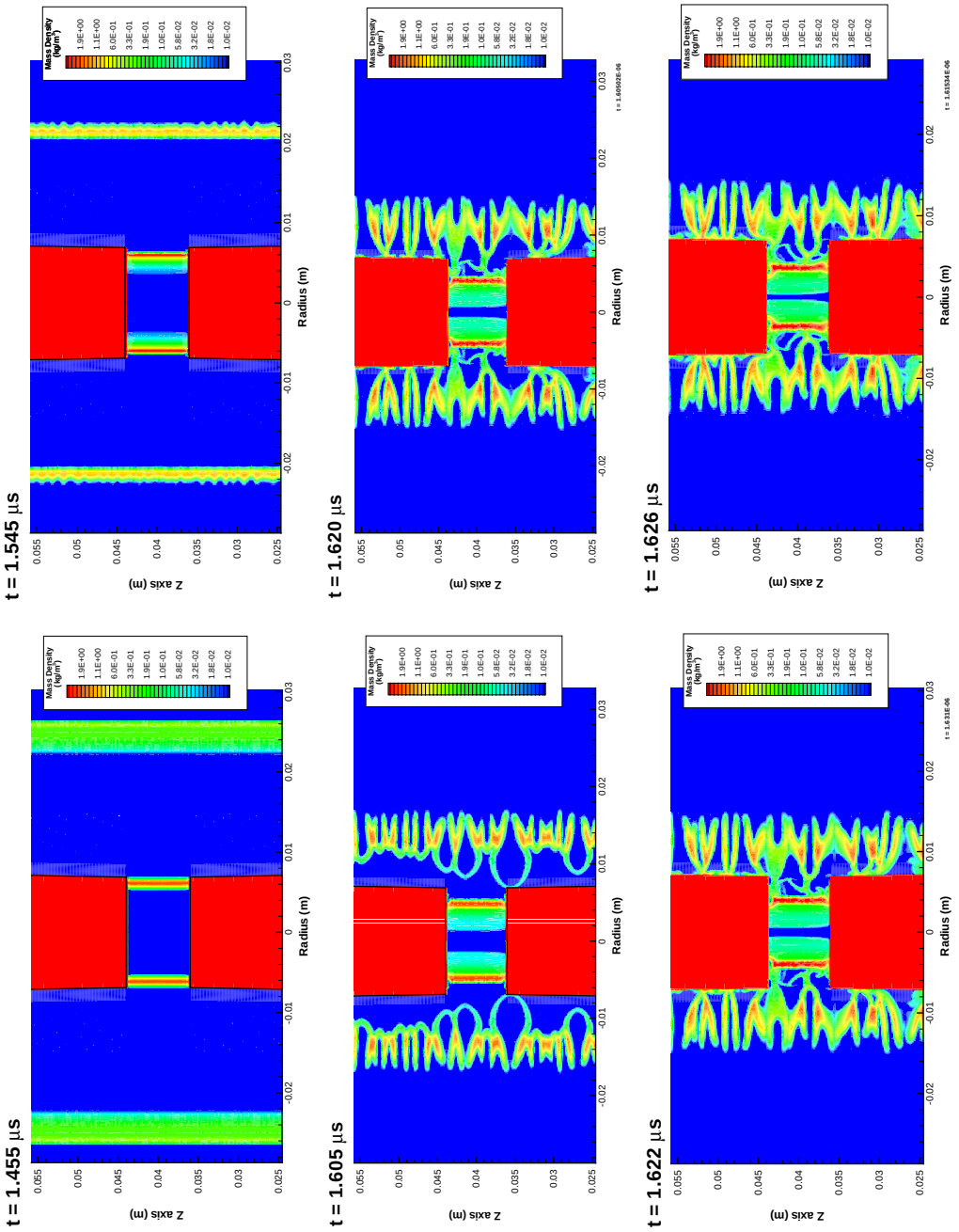

Figure 8.5: Mass density contour evolution with a 5% initial density perturbation.

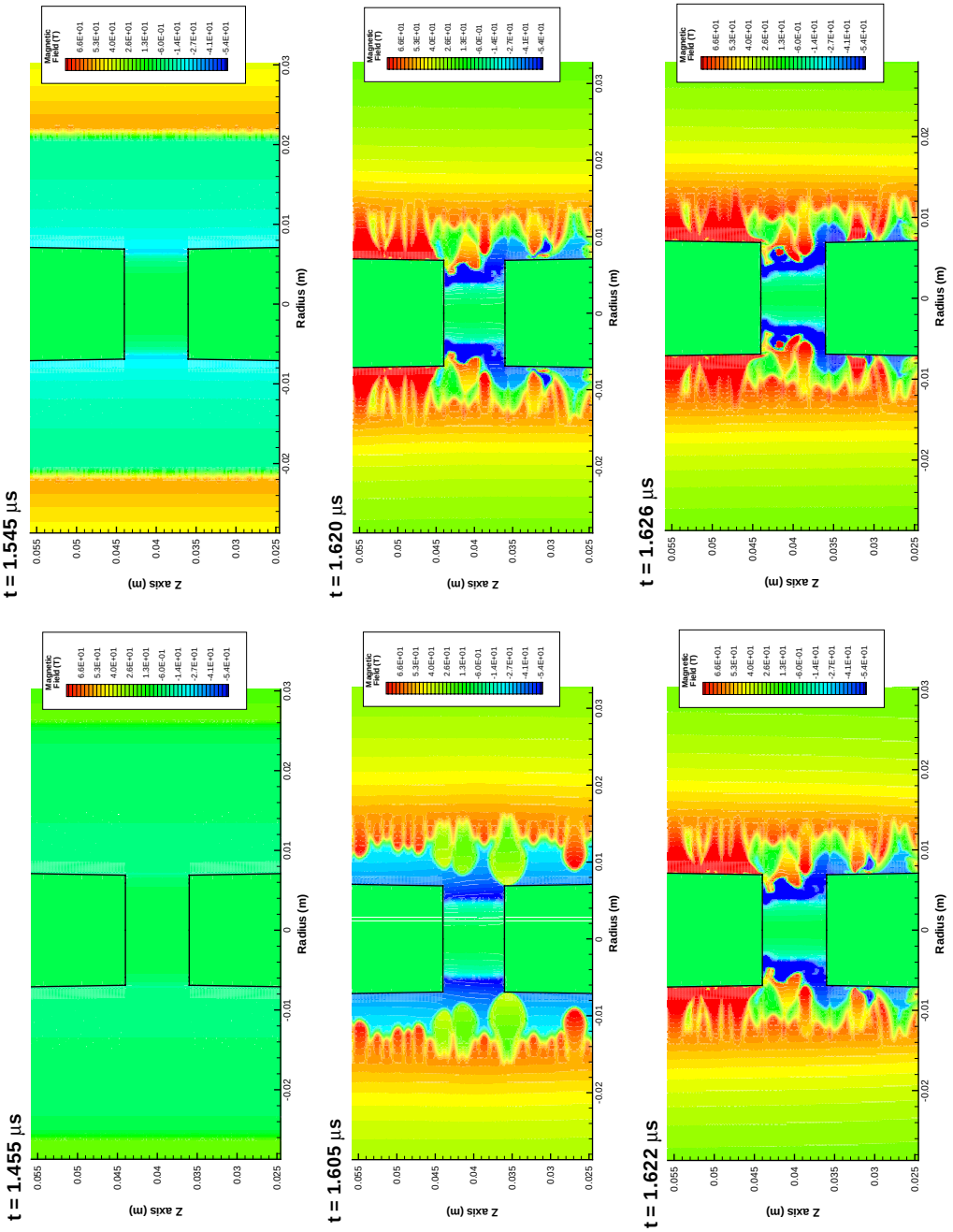

Figure 8.6: Magnetic field contour evolution with a 5% initial density perturbation.

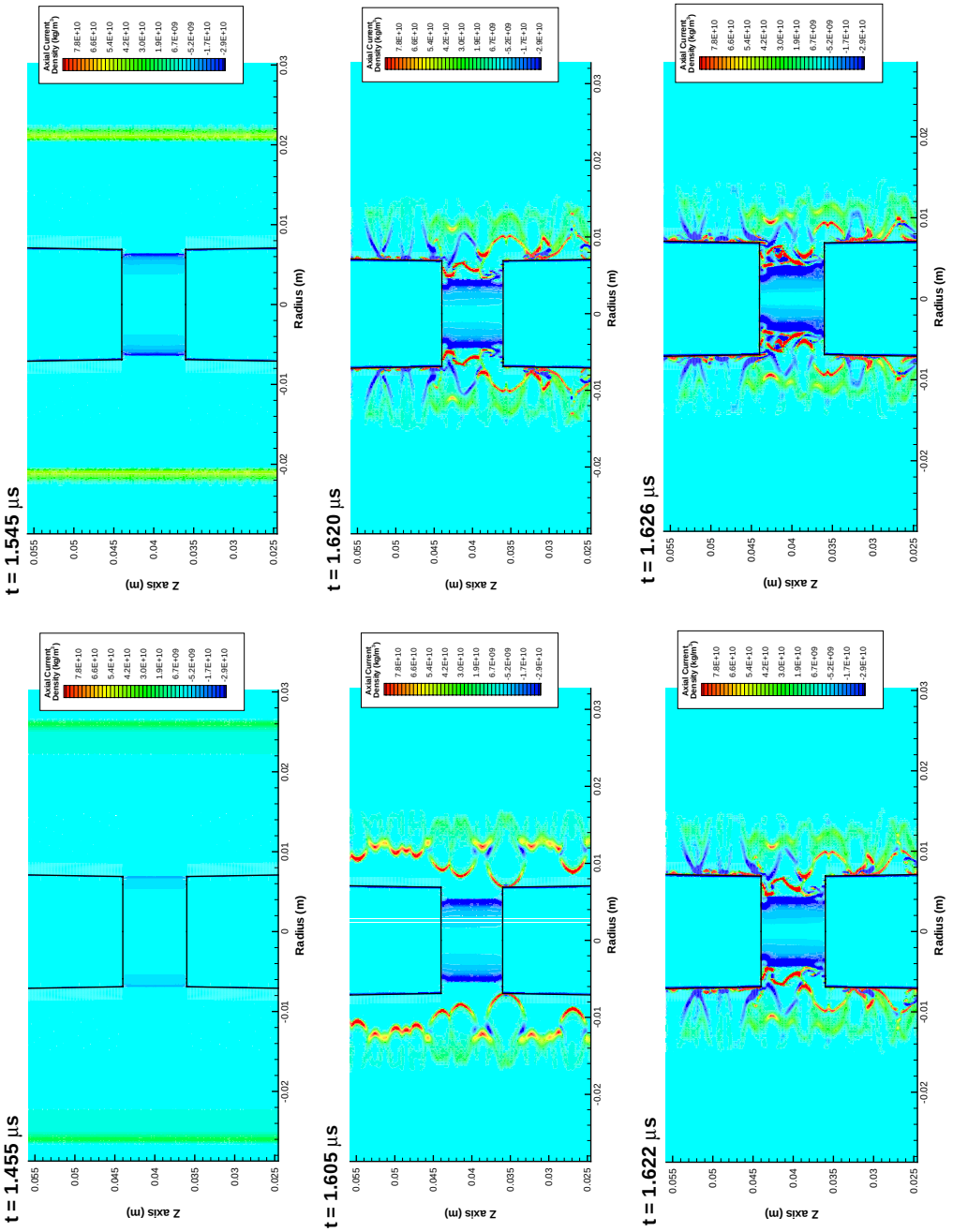

Figure 8.7: Axial current density contour evolution with a 5% initial density perturbation.

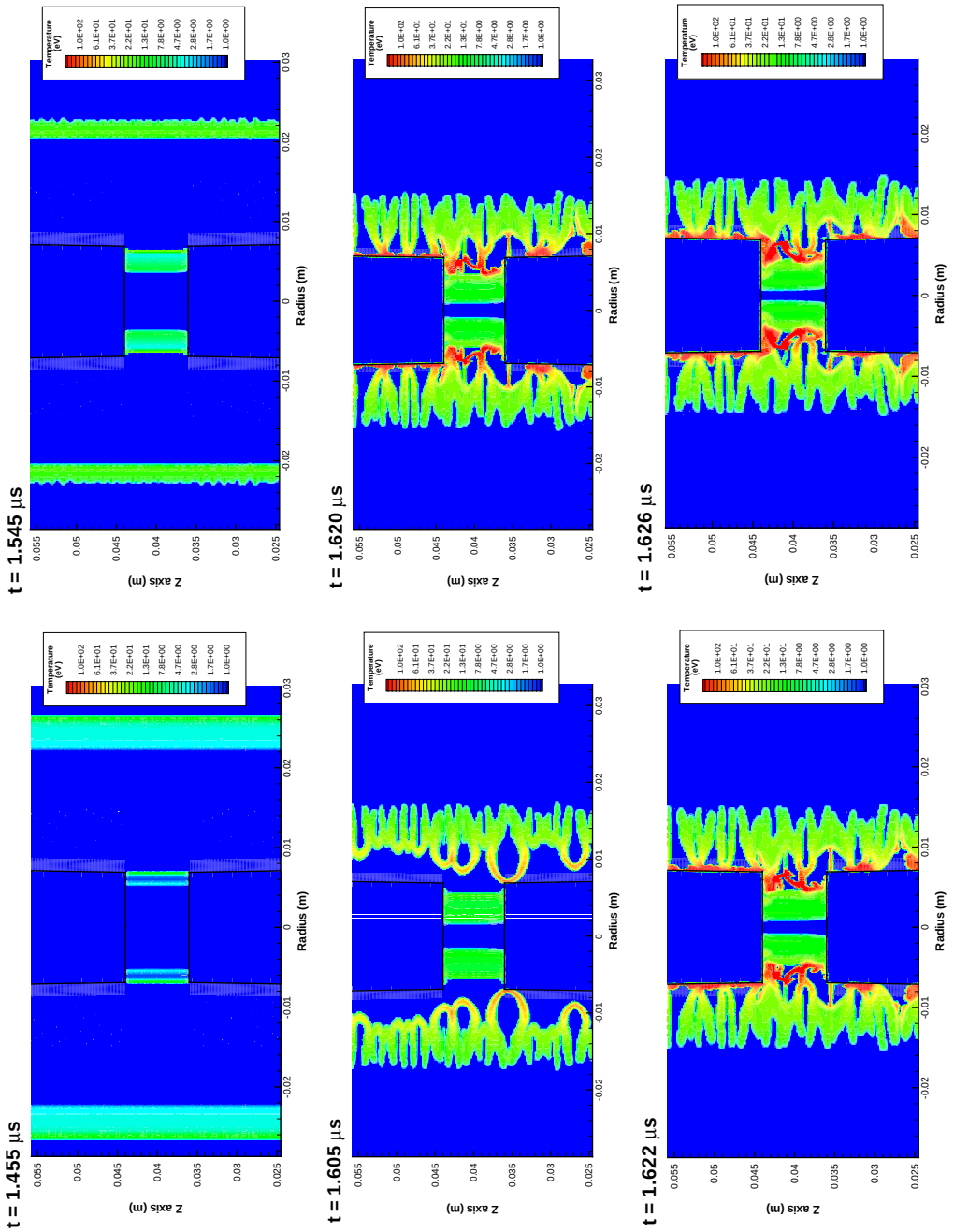

Figure 8.8: Temperature contour evolution with a 5% initial density perturbation.

Appendix B

Sample Input File

MACH2 code input file for flux compression simulations

```
 mach2 run of 14 Apr 2003, started at 10:30:07: input file image
RT01-ang0-shr-quadeta-trns

 $contrl

! Basic Parameters
     dt = 1.0e-11,
     dtmax = 100.0e-9,
     twfn = 1.68e-6,
     cyl = 1.,
     trackon = .false.,
     trktol=1.0e-4,
     courmax = 100.0,
!     volratm = 0.8,
!     rmvolrm = 0.5,
     eps = 1.e-4,

!    **** Physics ****
!    Turn off hydro initially

     hydron = .false.,
     itopt = 80.,
     tsplit = 2,
     jouhtmlt = 1.0,
     eoson = .true.,
!     viscosty = .true.,
       mu = 5.6,
!     radiate = .true.,
!     thmldif = .true.,
     strength = .true.,
     meshon = .false.,
!     filter = .true.,
     aresvac = 1.0e5,
     htsclfac = 1,

! External circuit model. See $curnt.
     ciron = .true.,

! Magnetic diffusion parameters
      bdiff = .true.,
!      btherm = .true.,
      mgmode = 'vcycle',
      multgrd = .true.,
```

```
      rdtol = 1.0e-5,
      itpot = 1.,
        magrintp = 'precond',
        mglintp = 'condintp',
!        mgmode = 'converg2',
        vtxeta = 'condctnc',
       nvcycmx = 15,
       ncorplv = 12,
!      brbzon =.true.,
!      mglmax = 3,
       rdrelax = 1.0,
       itmaxrd = 2000.,


! Conserve energy ?
!     conserv = 1,

 $end

! External constant current of 8.5 MA.
 $curnt
     circtype(1) = 'rlc',
     volttype(1) ='genfile',
     exres(1) = 0.0022,
     exind(1) = 6.0e-9,
     volt(1) = 1.0e-5,
     capac(1) = 0.0,

     voltmodl = 'fluxcomp',
       pripathmat = 2,
       secpathmat = 3,
      addsecpathind = 60.0e-9,

     crowrad(1) = 1.45e-2
     crowrad(2) = 1.40e-2
     crowval(1) = 1.0e7
     crowval(2) = 1.0e-3

     cirl(2) = 20.0e-11,
     cirl(1) = 5.0e-11,

     fluxmat(2) = 1,
     fluxmat(3) = 2,
     nmixedmat(1) = 2,
     pmixedmat(1) = 1,
     fluxmat(6) = 2,
     fluxmat(4) = 2,
     matintf(1) = 1,
     matintf(2) = 3,
     calcrvolt = .true.,
 $end

 $ezphys
      siecoldg = -1.e99,
```

```
      gdvlg = 0.0,
      icellsg = 48,
      jcellsg = 216,
      roig = 1.0e-4,
      tempig = 1.0,
!      arsmodlg = 'vacuum',
       rofradg = 1.0e99,
    rofsiecg = 1.0e-4,

 $end

! Set up Grid Geometry....

 $ezgeom
     nblk = 21,
     npnts = 32,

     pointx(1) = 0.0,
     pointy(1) = 8.0e-2,

     pointx(2) = 0.75e-2,
     pointy(2) = 8.0e-2,

     pointx(3) = 1.5e-2,
     pointy(3) = 8.0e-2,

     pointx(4) = 2.45e-2,
     pointy(4) = 8.0e-2,

     pointx(5) = 2.55e-2,
     pointy(5) = 8.0e-2,

     pointx(6) = 3.5e-2,
     pointy(6) = 8.0e-2,

     pointx(7) = 3.5e-2,
     pointy(7) = 0.0,

     pointx(8) = 2.55e-2,
     pointy(8) = 0.0,

     pointx(9) = 2.45e-2,
     pointy(9) = 0.0,

     pointx(10) = 1.5e-2,
     pointy(10) = 0.0,

     pointx(11) = 0.75e-2,
     pointy(11) = 0.0,

     pointx(12) = 0.0,
     pointy(12) = 0.0,

     pointx(13) = 0.0,
```

```
pointy(13) = 3.6e-2,

pointx(14) = 0.69e-2,
pointy(14) = 3.6e-2,

pointx(15) = 0.69e-2,
pointy(15) = 4.4e-2,

pointx(16) = 0.0,
pointy(16) = 4.4e-2,

pointx(17) = 1.5e-2,
pointy(17) = 4.4e-2,

pointx(18) = 2.45e-2,
pointy(18) = 4.4e-2,

pointx(19) = 2.55e-2,
pointy(19) = 4.4e-2,

pointx(20) = 1.5e-2,
pointy(20) = 3.6e-2,

pointx(21) = 2.45e-2,
pointy(21) = 3.6e-2,

pointx(22) = 2.55e-2,
pointy(22) = 3.6e-2,

pointx(23) = 3.5e-2,
pointy(23) = 3.6e-2,

pointx(24) = 3.5e-2,
pointy(24) = 4.4e-2,

pointx(25) = 0.64e-2,
pointy(25) = 4.4e-2,

pointx(26) = 0.64e-2,
pointy(26) = 3.6e-2,

pointx(27) = 0.64e-2,
pointy(27) = 8.0e-2,

pointx(28) = 0.64e-2,
pointy(28) = 0.0,

pointx(29) = 0.59e-2,
pointy(29) = 4.4e-2,


pointx(30) = 0.59e-2,
pointy(30) = 8.0e-2,
```

```
     pointx(31) = 0.59e-2,
     pointy(31) = 0.0,

     pointx(32) = 0.59e-2,
     pointy(32) = 3.6e-2,

     corners(1,1) = 15,17,20,14,
     corners(1,2) = 17,18,21,20,
     corners(1,3) = 18,19,22,21,
     corners(1,4) = 19,24,23,22,
     corners(1,5) = 16,29,32,13,
     corners(1,6) = 2,3,17,15,
     corners(1,7) = 3,4,18,17,
     corners(1,8) = 4,5,19,18,
     corners(1,9) = 5,6,24,19,
     corners(1,10) = 14,20,10,11,
     corners(1,11) = 20,21,9,10,
     corners(1,12) = 21,22,8,9,
     corners(1,13) = 22,23,7,8,
     corners(1,14) = 25,15,14,26,
     corners(1,15) = 29,25 ,26 ,32,

     corners(1,16) = 30,27,25,29,
     corners(1,17) = 27,2,15,25,
     corners(1,18) = 26,14,11,28,
     corners(1,19) = 32,26 ,28 ,31,
          corners(1,20) = 1,30,29,16,
          corners(1,21) = 13,32,31,12
 $end

 $inmesh
!      dirintp(18) = 'intp4t2q',
!      rntplen(18) = -0.0001,

!      dirintp(17) = 'intp4t2q',
!      rntplen(17) = -0.0001,
      dirintp(1) = 'intp4t2q',
      rntplen(1) = 0.0001,

      dirintp(6) = 'intp4t2q',
      rntplen(6) = 0.0001,

      dirintp(10) = 'intp4t2q',
      rntplen(10) = 0.0001,
    icells(8) = 8,
     icells(3) =  8,
     icells(12) = 8,
     icells(14) = 8,
     icells(15) = 8,
     icells(16) = 8,
     icells(17) = 8,
     icells(18) = 8,
     icells(19) = 8,
```

```
     icells(5) = 48,
     icells(20) = 48,
     icells(21) = 48,

     jcells(5) = 48,
     jcells(1) = 48,
     jcells(2) = 48,
     jcells(14) =48,
     jcells(3) = 48,
     jcells(4) = 48,
     jcells(15) = 48,

!     jcells(16) = 2,


     icells(9) = 48,
     icells(4)  =48,
     icells(13) = 48,


! Set materials in blocks.
     matnami(1) = 'vac2',
     matnami(2) = 'vac2',
     matnami(3) = 'al',
     matnami(4) = 'vac1',
     matnami(5) = 'vac3',
     matnami(6) = 'vac2',
     matnami(7) = 'vac2',
     matnami(8) = 'al',
     matnami(9) = 'vac1',
     matnami(10) = 'vac2',
     matnami(11) = 'vac2',
     matnami(12) = 'al',
     matnami(13) = 'vac1',
     matnami(14) = 'vac2',
     matnami(15) = 'ti',

     matnami(16) = 'al2',
     matnami(17) = 'al2',
     matnami(18) = 'al2',
     matnami(19) = 'al2',
     matnami(20) = 'al2',
     matnami(21) = 'al2',


! Hydro boundary conditions.
     hydbc(1,7)  = 'wall',
     hydbc(1,8) = 'wall',
     hydbc(1,9) = 'wall',
     hydbc(3,10) = 'wall',
     hydbc(3,11) = 'wall',
     hydbc(3,12) = 'wall',
     hydbc(3,13) = 'wall',
     hydbc(4,5)  = 'axis',
```

```
     hydbc(2,9) = 'wall',
     hydbc(2,4)  = 'wall',
     hydbc(2,13) = 'wall',

!     roflow(2,9) = 5e-4,
!     tflow(2,9) = 1.0,
!     roflow(2,4) = 5e-4,
!     tflow(2,4) = 1.0,
!     roflow(2,13) = 5e-4,
!     tflow(2,13) = 1.0,

     hydbc(4,6) = 'wall',
     hydbc(4,10) = 'wall',
     hydbc(1,5) ='wall',
     hydbc(3,5) ='wall',
     hydbc(1,14) ='wall',
     hydbc(3,14) ='wall',
     hydbc(1,15) ='wall',
     hydbc(3,15) ='wall',


     velcc(3,17) = 'no slip',
     velcc(2,18) = 'no slip',

! Initial conditions.
! Initial Plama temperature 5 eV

     tempi(8) = 5.0,
     tempi(3)  = 5.0,
     tempi(12) = 5.0,
     tempi(15) = 1.0,
     tempi(16) = 2.5e-2,
     tempi(17) = 2.5e-2,
     tempi(18) = 2.5e-2,
     tempi(19) = 2.5e-2,
     tempi(20) = 2.5e-2,
     tempi(21) = 2.5e-2,
     roi(8) = 0.8,
     roi(3) = 0.8,
     roi(12) = 0.8,
     roi(15) = 2.514,
     roi(16) = 2.7e3,
     roi(17) = 2.7e3,
     roi(18) = 2.7e3,
     roi(19) = 2.7e3,
     roi(20) = 2.7e3,
     roi(21) = 2.7e3,

! Density Perturbations
     ropert(8) = 'random',
     ropert(3) = 'random',
     ropert(12) = 'random',
     roprtam(8) = 0.01,
     roprtam(3) = 0.01,
```

```
     roprtam(12) = 0.01,
     ropert(15) = 'random',
     roprtam(15) = 0.01,
     intrbc(2,4) = .true.,

! Magnetic boundary conditions.

     imagseq(1,17) = 1,4,3,2
     imagseq(1,18) = 3,4,1,2
     magbc(2,9) = 'fluxpri'
     magbc(2,13) = 'fluxpri'
     magbc(2,4)  = 'fluxpri',
     currcir(2,4) = 1,


     magzbc(4,6) = 'fluxsec',
     magzbc(4,10) = 'fluxsec',


!     rstbc(4,6) = 'vacuum',
!     rstbc(4,10) = 'vacuum',
!     magbc(1,14) = 'conductr',
!     magbc(3,14) = 'conductr',
!     magbc(1,14) = 'fluxsec',
!     magbc(3,14) = 'fluxsec',

!      rstbc(1,14)= 'symmetry',
!      rstbc(3,14)= 'symmetry',
!      rstbc(3,16)= 'symmetry',
!      rstbc(1,19)='symmetry',

     magbc(1,6)  = 'contnutv',
!     magbc(1,6)  = 'insulatr',

     magbc(1,7)  = 'contnutv',
     magbc(1,8) = 'contnutv',
     magbc(1,9) = 'contnutv',
     magbc(3,10) = 'contnutv',
     magbc(3,11) = 'contnutv',
     magbc(3,12) = 'contnutv',
     magbc(3,13) = 'contnutv',
     magzbc(4,5) = 'axis',
!     magbc(1,5) = 'conductr',
!     magbc(1,15) = 'conductr',
!     magbc(3,5) = 'conductr',
!     magbc(3,15) = 'conductr',


      radbc(2,9) = 'conduct',
      radbc(2,13) = 'conduct',
      radbc(2,4) = 'conduct',
      tradbdy(2,9) = 1.0,
      tradbdy(2,13) = 1.0,
      tradbdy(2,4) = 1.0,
```

```
      rstbc(4,6) = 'vacuum',
      rstbc(4,10)= 'vacuum',
 $end
 $matmdl
! 6061 aluminum.
     matnam(1) = 'al',
     sesanam(1) = 'al-new',
     eosmodl(1) = 'tabular',

       resmodl(1) = 'tabular',
!      resmodl(1) = 'spitzer',
       arsmodl(1) = 'vacuum',
       opacity(1) ='tabular',
!      resmodl(1) = 'constant',
!       nfemodl(1) = 'saha_h',
        nfemodl(1) = 'zfree2',
!       eta0(1) = 10.,
       tcnmodl(1) = 'spitzer',
       an(1) = 27,
       aw(1) = 26.98,
       yield(1) = 9.e8,
       elpmodl(1) = 'steinb-g',
       elpname(1) = 'al-6061',

! - Floors/Ceilings
       rofsiecp(1) = 1.0e-3,
       teflr(1) = 2.5e-3,
!       ticap(1) = 1.0e2,
!       tecap(1) = 150.,
       tiflr(1) = 2.5e-3,
!       rofjoule(1) = 1.e-4,
       rof(1) = 1.0e-4,
       rofhyd(1) = 1.0e-4,
       rofanom(1) = 1.0e-4,
       rofrad(1) = 1.0e-4,

! Air-outside
    matnam(2) = 'vac1',
    sesanam(2) = 'al-new',
      eosmodl(2) = 'tabular',
      resmodl(2) = 'constant',
      nfemodl(2) = 'constant',
      opacity(2) = 'none',
      eta0(2) = 1e7,
!      arsmodl(2) = 'vacuum',
      tcnmodl(2) = 'constant'
      etc0(2) = 1.0e-2,
      itc0(2) = 1.0e-2,
       nfe0(3) = 0,
! Floors/Ceilings
      rofsiecp(2) = 1.0,
      tecap(2) = 1.0,
      ticap(2) = 1.0,
       tiflr(2) = 1.0,
```

```
      teflr(2) = 1.0,
      rofjoule(2) = 2.41e99,
      rofhyd(2) = 1.0e-3,
      rof(2) = 1.0e-3,
!       rofvac(2) = 1.0,
!      rofanom(2) = 2.41e-4,
      nu(2) = 1.0,

! Air-inside
    matnam(3) = 'vac2',
    sesanam(3) = 'al-new',
     eosmodl(3) = 'tabular',
     nfemodl(3) = 'constant',
      resmodl(3) = 'constant',
      opacity(3) = 'none',
      eta0(3) = 1e7,
!      arsmodl(3) = 'vacuum',
      tcnmodl(3) = 'constant'
      etc0(3) = 1.0e-2,
      itc0(3) = 1.0e-2,
      nfe0(3) = 0,
! Floors/Ceilings
      rofsiecp(3) = 1.0,
      tecap(3) = 1.0,
      ticap(3) = 1.0,
      tiflr(3) = 1.0,
      teflr(3) = 1.0,
     rofjoule(3) = 2.41e99
      rofhyd(3) = 1.0e-3,
      rof(3) = 1.0e-3,
!       rofvac(3) = 1.0,
!      rofanom(3) = 2.41e-4,
      nu(3) = 1.0,


! Air-inside
    matnam(5) = 'vac3',
    sesanam(5) = 'al-new',
    nfemodl(5) = 'constant',
     eosmodl(5) = 'tabular',
      resmodl(5) = 'constant',
      opacity(5) = 'none',
      eta0(5) = 1e7,
!      arsmodl(5) = 'vacuum',
      tcnmodl(5) = 'constant'
      etc0(5) = 1.0e-2,
      itc0(5) = 1.0e-2,
       nfe0(5) = 0,
! Floors/Ceilings
      rofsiecp(5) = 1.0,
      tecap(5) = 1.0,
       tiflr(5) = 1.0,
      ticap(5) = 1.0,
      teflr(5) = 1.0,
```

```
     rofjoule(5) = 2.41e99,
      rofhyd(5) = 1.0e-3,
      rof(5) = 1.0e-3,
!       rofvac(5) = 1.0,
!      rofanom(5) = 2.41e-4,
       nu(5) = 1.0,


! 6061 aluminum.
     matnam(4) = 'al2',
     sesanam(4) = 'al-new',
     eosmodl(4) = 'tabular',

! - Select Resistivity Model...
       resmodl(4) = 'tabular',
!       arsmodl(4) = 'none',
!       resmodl(4) = 'constant',
       mattabs(3,24) = 29373,
       mattabs(2,24) = 19371,
!       nfemodl(4) = 'tabular',
       opacity(4) = 'none',
       eta0(4) = 26,
! - Select Thermal Conductivity Model..
       tcnmodl(4) = 'tabular',
!       yield(4) = 9.e8,
!       elpmodl(4) = 'steinb-g',
!       elpname(4) = 'al-6061',
! - Floors/Ceilings
       rofsiecp(4) = 2.7e3,
       teflr(4) = 2.5e-2,
       tecap(4) = 2.5e-2,
       tiflr(4) = 2.5e-2,
       ticap(4) = 2.5e-2,
       rofjoule(4) = 1.0e2,
       rof(4) = 2.71e3,
       rofhyd(4) = 1.0e99,


! 6061 aluminum.
     matnam(6) = 'ti',
     sesanam(6) = 'ti',
     eosmodl(6) = 'tabular',

! - Select Resistivity Model...
       resmodl(6) = 'tabular',
       nfemodl(6) = 'zfree2',
!       arsmodl(6) = 'vacuum',
!      resmodl(6) = 'constant',
       opacity(6) ='tabular',
!       eta0(6) = 0.5,
       an(6) = 22,
       aw(6) = 47.9,
! - Select Thermal Conductivity Model..
       tcnmodl(6) = 'tabular',
```

```
!       yield(6) = 9.e8,
!       elpmodl(6) = 'steinb-g',
!       elpname(6) = 'al-6061',
! - Floors/Ceilings
       rofsiecp(6) = 1.0e-3,
       teflr(6) = 2.5e-2,
       tecap(6) = 150.,
       ticap(6) = 150.,
       tiflr(6) = 2.5e-3,
       mattabs(3,46) = 29964,
       mattabs(2,46) = 12960,
       rofjoule(6) = 1.e-2,
       rof(6) = 1.0e-4,
       rofhyd(6) = 1.0e-4,
       rofanom(6) = 1.0e-4,
       rofrad(6) = 1.0e-4,
 $end

 $output
!     dtrst = .5e-6,
!     dtpost = 1.e-8,
     intty = 'edits,1',
     ncyctty = 1,
!     ncyctec = 50,
     ncycrst = 50,
!     poston = .true.,
     tectype = 'binary',
!     tecmode = 'both',
!     tecwidth = 'all',
      histfrmt = 'tec',
     histenrg = .true.,
     histeerr = .true.,
     histflux = .true.,
     tplot(0) = 'plasma',
     tplot(16) = 'pressure',
     tplot(17) = 'density',
     tplot(18) = 'rbtheta',
     tplot(19) = 'efield',
!     tplot(20) = 'numvis',
     tplot(21) = 'totpress',
     tplot(22) = 'diffusiv',
     tplot(23) = 'joulheat',
     tplot(24) = 'force',
     tplot(25) = 'accelb',
     tplot(26) = 'accelp',
     tplot(27) = 'accel',
!     tplot(28) = 'thrmflux',
!     tplot(29) = 'thmratio',
     tplot(30) = 'sie',
!     tplot(31) = 'radheat',
!     tplot(32) = 'opacp',
     tplot(33) = 'forceb',
     tplot(34) = 'forcep',
     tplot(35) = 'sound',
```

```
     tplot(36) = 'alfven',
!     tplot(37) = 'm_te',
     tplot(38) = 'm_ro',
!     tplot(39) = 'm_vel',
     tplot(40) = 'magp',
!     tplot(41) = 'm_dffusv',
!     tplot(42) = 'm_ohmht',
!     tplot(43) = 'material',
!     tplot(44) = 'cellmass',
     tplot(45) = 'massfrac',
!     tplot(46) = 'divb',
!     tplot(47) = 'dive',
!     tplot(48) = 'divv',
!     tplot(49) = 'vxb',
!     tplot(50) = 'nrgflxv',
!     tplot(51) = 'poynting',
!     tplot(52) = 'bline',
     tplot(53) = 'nfe',
     tplot(54) = 'tion',
     tplot(55) = 'sieion',
!     tplot(56) = 'trad',
!     tplot(57) = 'erad',
!     tplot(58) = 'radflux',
!     tplot(59) = 'kinden',
!      tplot(60) = 'dirke',
!      tplot(61) = 'tcone',
!      tplot(62) = 'tconi',
!      tplot(63) = 'photmfpp',
!      tplot(64) = 'photmfpr',
!      tplot(65) = 'opacr',
     intrfac = 1,
     ncychist = 20,
     dthist = 1.0e-10,
     dttec = 50.e-6,

 $end

 $modtime
   tmod= 1.3e-6,
 $end

 $contrl
   dtmax= 5.0e-9,

! Magnetic diffusion parameters
      bdiff = .true.,
      mgmode = 'vcycle',
!      multgrd = .true.,
      rdtol = 1.0e-5,
      itpot = 1.,
        magrintp = 'precond',
        mglintp = 'condintp',
!        mgmode = 'converg2',
        vtxeta = 'condctnc',
```

```
       nvcycmx = 15,
       ncorplv = 10,
!      brbzon =.true.,
!       mglmaxi=2,
!       mglmaxj=4,
!      mglmax = 3,
       rdrelax = 1.0,
      itmaxrd = 2000.,
        mgmodete = 'converge',
       tdtole = 1.0e-4,
        mgmodeti = 'converge',
       tdtoli = 1.0e-4,
 $end

 $modtime
   tmod= 1.34e-6,
 $end

 $contrl
   dtmax = 5.0e-10,
      hydron=.true.,
      btherm=.true.,
       thmldif = .true.,
!      radiate=.true.,
!       mu = 2.0,
!         viscosty = .true.,
!      radmodl = 'neqdiff',
!       radcupl = 'integrtd',
!       radflxlt = .false.,
!          mgmodetr = 'vcycle',
!           tdtolr = 1.e-5,
!       nvcymxtr = 10,
!       ncrplvtr = 10,
       nvcycmx = 12,
       ncorplv = 10,
!       tdtolr = 1.e-4,
!       it2tolrd = 500,
!       itmaxrd = 4000,
       nthrmax = 4000,
!       nvcymxtr = 4,
!       ncrplvtr = 100,
         tdrelax=1.0,
!      thmrintp ='precond',
 $end
 $output
     dttec = 15.e-9,
 $end
 $modtime
   tmod = 1.545e-6,
 $end

 $contrl
   dtmax = 2.5e-10,
!   rmvolrm = 0.05,
```

```
       nvcycmx = 10,
       ncorplv = 6,

 $end
 $output
   ncycrst = 200,
 $end

 $modtime
   tmod = 1.570e-6,
 $end

 $contrl

!   rmvolrm = 0.03,
       nvcycmx = 6,
       ncorplv = 6,

 $end
 $modtime
   tmod= 1.620e-6,
 $end
 $contrl
   dtmax = 1.0e-11,
!   rmvolrm = 0.01,
 $end
 $output
   dttec = 5.0e-10,
   ncycrst=75,
 $end
```

Appendix C

Flux Compression Code Modules for MACH2

C.1 Ciredl4.F

```
      subroutine ciredl4(edotdl,icur)

cdir$ nolist
      include 'common.h'
      include 'inputcom.h'
      include 'pointer.h'
      include 'circom.h'
      include 'intrcom.h'
cdir$ list

      logical crowengage

c----- Zero out variables at start of cycle

      flux = zero
      fluxpri = zero
      fluxsec = zero
      primag  = zero
      secmag  = zero
      priarea = zero
      secarea = zero
      difflux = zero


c----- Determine flux compression crowbar switch resistance

            if (radius(1).gt.crowrad(1) .or. ncyc.lt.5) then
             if (crowengage.ne. .true.) then
              crowresist = crowval(1)
             endif
            else if(radius(1).lt.crowrad(1).and.
     %             radius(1).gt.crowrad(2)) then

c----- exponential decay of crowresistance

             decay = (log(crowval(2)/crowval(1)))/
     %                (crowrad(1)-crowrad(2)+tiny)
             crowresist = crowval(1)*exp(decay*(crowrad(1)-
```

```
     %                    radius(1)))
            else
             crowresist = crowval(2)
             crowengage = .true.
            endif

c ---    Get shell radii from material tracker...

         radius(1)=rint(interface(1))
         radius(2)=rint(interface(2))

      do 10 lblk=1,nblk
         call setblk
c ---  Calculate magnetic flux regions
         call ciredl4a(difflux)
  10  continue

c ---  Prevent divide by zero errors....

      if (fluxsec .eq.zero) then
         fluxsec=tiny
      endif

c ---  Find Inductance for current paths

       if (ncyc .gt. 0) then
          oldlpri = cirlpri
          oldlsec = cirlsec

c ---  Calculate inductance for circuit loops
       do 20 i=1, mxmats
          cirl(i)= cirflux(i) / (fluxcurr(i)+tiny)
  20   continue

       endif

c ---  Define Specific flux compression inductance variables
       cirlpri = cirl(1)
       cirlsec = cirl(2)

c ---  Calculate resistive impendence in the form of voltage

       xpnd = htsclfac * tpi

       if (ncyc.gt.1 ) then

         edotdlcop=abs((xpnd*(ohmheat(1))/dtold)/
     %    (current1(1)+abs(fluxcurr(4))))
         edotdlsec=abs((xpnd*(ohmheat(6)+ohmheat(4))/dtold)/
     %     abs(fluxcurr(2)))

c ---  Calculate edotdl on pri circuit for compatibility of port voltage

         edotdl = edotdlcop + htsclfac*((fluxpri-oldflux)/dtold)
```

```
       endif

c ---  Solve Circuit Equations

       call matrixsolve(edotdl)

c --- Set oldflux variable for compatibility

       oldflux = fluxpri


       return
       end
```

C.2 Ciredl4a.F

```
      subroutine ciredl4a(difflux)

c-----calculate regions of magnetic flux based on material number
c-----Materials with regions of opposing magnetic flux treated
c-----specially

cdir$ nolist
      include 'common.h'
      include 'inputcom.h'
      include 'pointer.h'
      include 'circom.h'
      include 'paramcom.h'
cdir$ list

      int mattol

c --- Define Material Tracking Tolerance

      mattol = 0.5

c        i,j+1      c       i+1,j+1
c           ><--------------\
c          /                 \
c      a  /                   \  d
c        /         i,j         \
c       /                       >
c   i,j------------------------> i+1,j
c                   b

c-----Area = 0.5d0*( a x b + d x c)*cylbnd
      do 10 j=1,jcels
        do 10 i=1,icels
         ax = x(i,j+1)-x(i,j)
         ay = y(i,j+1)-y(i,j)
         bx = x(i+1,j)-x(i,j)
         by = y(i+1,j)-y(i,j)
```

```
         cx = x(i,j+1)-x(i+1,j+1)
         cy = y(i,j+1)-y(i+1,j+1)
         dx = x(i+1,j)-x(i+1,j+1)
         dy = y(i+1,j)-y(i+1,j+1)

c ---- Calculate mag flux in cell

         flux1 = zero
         flux1 = cylbnd*0.5d0*( bx*ay - ax*by
     %                            +cx*dy - dx*cy )*bzn(i,j)

         do 10 k=1, mxmats

c ---    Fluxmat(mat) defines the circuit path associated with material

         if (massfrac(i,j,k).gt.mattol .and. fluxmat(k) .ne. zero) then
           cirflux(fluxmat(k)) = cirflux(fluxmat(k)) + flux1
           cirarea(fluxmat(k)) = cirarea(fluxmat(k)) + abs(cylbnd*0.5d0*(
     %       bx*ay - ax*by +cx*dy - dx*cy ))
           cirmag(fluxmat(k))  = cirmag(fluxmat(k)) + bzn(i,j)

c ---    Treat regions of opposing flux separatly
c ---     mixedmatp(mat) defines which circuit path positive flux belongs to
c ---     mixedmatn(mat) defines which circuit path negative flux belongs to
c ---     If either are zero, then do nothing

         else if (flux1 .gt. 0.0 .and. massfrac(i,j,k).gt.mattol .and.
             fluxmat(k).eq. zero)
           cirflux(mixedmatp(k)) = cirflux(mixedmatp(k)) + flux1
           cirarea(mixedmatp(k)) = cirarea(mixedmatp(k)) + abs(cylbnd*0.5d0*(
     %       bx*ay - ax*by +cx*dy - dx*cy ))
           cirmag(mixedmatp(k))  = cirmag(mixedmatp(k)) + bzn(i,j)
         else if (flux1 .lt. 0.0 .and. massfrac(i,j,k).gt.mattol)
           cirflux(mixedmatn(k)) = cirflux(mixedmatn(k)) + flux1
           cirarea(mixedmatn(k)) = cirarea(mixedmatn(k)) + abs(cylbnd*0.5d0*(
     %       bx*ay - ax*by +cx*dy - dx*cy ))
           cirmag(mixedmatn(k))  = cirmag(mixedmatn(k)) + bzn(i,j)
         else
c ---      do nothing
        endif

  10  continue

      return
      end
```

C.3 Matrixsolve.F

```
      subroutine matrixsolve(edotdl)


cdir$ nolist
```

```
      include 'common.h'
      include 'inputcom.h'
      include 'pointer.h'
      include 'circom.h'
cdir$ list

      INTEGER np,nmax,itol,itmax
      DOUBLE PRECISION mtol
      PARAMETER (np=3,nmax=1000,itol=1,tol=1.d-9,itmax=75)

      COMMON /mat/ sa(NMAX),ija(NMAX),vectx(NMAX)
      INTEGER i,iter,msize,ija
      DOUBLE PRECISION amatrix(np,np),sa,vectx
      DOUBLE PRECISION vectb(np),bcmp(np),calcerr

c --- Prevent large initial inductance shock to circuit equations...

      if (oldlpri.eq.zero) then
        oldlpri = cirlpri
      endif
      if (oldlsec.eq.zero) then
        oldlsec = cirlsec
      endif

c ---  Define Linear Circuit Matrix

      amatrix(1,1) = exres(1) + (exind(1)/dtold)+(2*cirlpri/dtold)
     %       -(oldlpri/dtold)
      amatrix(1,2) = -(2* cirlpri/dtold) + (oldlpri/dtold)
      amatrix(1,3) = 0.0

      amatrix(2,1) = -(2* cirlpri/dtold) + (oldlpri/dtold)
      amatrix(2,2) =(2*cirlpri/dtold)-(oldlpri/dtold)+(2*cirlsec/dtold)
     %   - (oldlsec/dtold) + 0.001 +  (addsecpathind/dtold)
         amatrix(2,3) = - (2*cirlsec/dtold) + (oldlsec/dtold)

      amatrix(3,1)=  0.0
      amatrix(3,2) = - (2*cirlsec/dtold) + (oldlsec/dtold)
      amatrix(3,3) = (2*cirlsec/dtold) + crowresist - (oldlsec/dtold)


c --- Create matrix into row-indexed sparce storage mode
      call sprsin(amatrix,np,np,0.5,nmax,sa,ija)

c --- Initialize Circuit on cycle zero
       if (ncyc.eq.0) then
         ciractv(1) = .true.
         fluxen = zero

c --- Initialize Solution Vector
        do 20 i=1,np
          vectx(i)=0.0
 20     continue
       endif
```

```
c --- See if user wants to incorporate resistive impendence

      if (calcrvolt.eq. .false) then
        edotdlcop = zero
        edotdlsec = zero
      endif

c --- Define Source vector

         vectb(1) = (exind(1)*current1(1)/dtold) + ((cirlpri/dtold)*
     %       fluxcurr(1))-volt(1)-edotdlcop
         vectb(2) = -(addsecpathind*fluxcurr(3)/dtold)-((cirlsec/dtold)*
     %       fluxcurr(2)) + ((cirlpri/dtold)*fluxcurr(1))-
     %       edotdlsec
         vectb(3) =  (cirlsec/dtold)*fluxcurr(2)+edotdlsec+edotdlcop

c --- Solve linear matrix equation by iterative biconjugate method
       call linbcg(np,vectb,vectx,3,tol,itmax,iter,calcerr)

c --- Circuit Solver Diagnostic Data...
c      write(*,'(/1x,a,e15.6)') 'Estimated Circuit error:',calcerr
c      write(*,'(/1x,a,i6)') 'Circuit Iterations needed:',iter
c      write(*,'(/1x,a)') 'Solution vector:'
c      write(*,*) (vectx(i),i=1,np)
c      call dsprsax(sa,ija,vectx,bcmp,np)
c      write(*,'(1x,a/t8,a,t22,a)') 'Test of solution vector:','a*x','b'
c      do 12 i=1,np
c        write(*,*) bcmp(i),vectb(i)
c 12    continue

c ---  Take solved mesh analysis currents and calculate real circuit paths

       current1(1) = vectx(1)
       fluxcurr(3) = vectx(2)
       fluxcurr(4) = vectx(3)
       fluxcurr(1) = (vectx(1) - vectx(2))
       fluxcurr(2) = -(vectx(2) - vectx(3))

c ---  Filter Current Data?

c       call cirfilt(vectx)

      end
```

C.4 Circuit.F

```
      subroutine circuit


c--------------NEW VERSION-----------------------------------------
c-----compute the current on up to MXCIRC boundaries conected to an
c----- external circuit
```

```
c-----the user may specify a current waveform or may have the
c----- code compute the current from a self-consistent circuit model
c----- that is driven by external voltages, iductances, resistances,
c----- and the E dot dl along the current-carrying boundary.

cdir$ nolist
       include 'common.h'
       include 'inputcom.h'
       include 'pointer.h'
       include 'circom.h'
cdir$ list

      dimension delin(mxcirc)
      save logical initfil
      data initfil /.false./

c-----See if we need to read a file & do so
c-----Once data are read, the circuit is treated as linear
c-----Note: all circtype=genfile circuits will use the same input file
c------ and all volttype=genfile circuits will use the same input file
      if ( .not. initfil ) then
        do 20 icur = 1,numcirc
          if (volttype(icur) .eq. 'genfile') then
            call circrdfl('volt',volttime(1,icur),voltvalu(1,icur))
            volttype(icur) = 'linear'

          endif
          if (circtype(icur) .eq. 'genfile') then
            call circrdfl('crnt',currtime(1,icur),currvalu(1,icur))
            circtype(icur) = 'linear'
          endif
  20    continue
        initfil = .true.
      endif

c-----compute the average timestep and its reciprocal
      dtav = 0.5 * ( dtold + dt )
      rdt  = one / dtav

c-----compute E dot dl and delin as needed
      call ciredln(delin)
c      write(*,*) "*****cnum****",numcirc
c-----loop over all external circuits
      do 1000 icur=1,numcirc
c        write(*,*) "*****start loop****",capac(1),ciractv(1)
        current0 = current1(icur)
        totcurr = current0
        curb0 = curb1(icur)
        if (faspr(icur) .gt. zero) call cirfusr(icur)
        exind(icur) = exind(icur) + delin(icur)
        exldt(icur) = delin(icur)/dtold

c-------If not the first time through this circuit,
c-------compute the voltage
```

```
        if (ciractv(icur)) then
c           write(*,*) "*****called cirvolt****",capac(1)
          if (capac(icur) .eq. zero) then
c---------compute the source voltage
            call cirvolt(icur)
c               write(*,*) "*****called cirvolt****"
          else
c-----------compute the voltage across a capacitor
            if (circtype(icur) .eq. 'rlc') then
              cs = current0
            elseif (circtype(icur) .eq. 'rlc2loop') then
              cs = curb0
            elseif (circtype(icur) .eq. 'pfn') then
              cs = zero
            endif
            volt(icur) = volt(icur) - cs*dtold/capac(icur)
          endif
        endif

        if (voltmodl.ne.'fluxcomp') then

c-------compute current0, curb0 , and trest
        call circrnt(icur,rdt,dtav,current0,curb0,trest)

        endif

c-------currents at t
        current(icur) = 0.5 * (current0 + current1(icur))
        curb(icur)    = 0.5 * (curb0    + curb1(icur)   )

c-------compute the energy presently in, and extracted from, the circuit
#ifdef NEWCIRC
        deltace = current1(icur) * edl(icur) * dt
#else
        deltace = current1(icur) * edl(icur) * dt -
     %            0.5 * current1(icur)**2 * delin(icur)
#endif
        enfrmcir(icur) = enfrmcir(icur) + deltace
        if ( (circtype(icur) .eq. 'rlc')      .or.
     %       (circtype(icur) .eq. 'rlc2loop')      ) then
          ec   = 0.5 * capac(icur) * volt(icur)**2
          el   = 0.5 * ( exind(icur) * current(icur)**2 +
     %                   bind(icur)  * curb(icur)**2 +
     %                   tind(icur)  * (curb(icur)-current(icur))**2 )
          exresnrg(icur) = exresnrg(icur) + dtold *
     %                     ( exres(icur) * current0**2 +
     %                       bres(icur)  * curb0**2 +
     %                       trest       * ( curb0 - current0 )**2 )
          enfus(icur) = fsie(icur) * fmas(icur)
          encir(icur) = ec + el + exresnrg(icur) + enfus(icur)
        else
          encir(icur) = encir(icur) - deltace
        endif
c----- inner wire array flux compression energy
```

```
       fluxen = fluxin + (fluxcurr(2)*(fluxsec-ofluxsec)/dtold)*dt


 1000 continue

      dtold = dt
#ifdef ENRGEDIT
      if (((dtenrg.ne.big)       .or. (ncycenrg.ne.ihuge)      ) .and.
     %    ((mod(t,dtenrg).le.dt) .or. (mod(ncyc,ncycenrg).eq.0)))
     %     call enrgacc('circuit ')
#endif

      return
      end
```

C.5 Circrnt.F

```
      subroutine circrnt(icur,rdt,dtav,current0,curb0,trest)

c-----compute the current on this circuit

cdir$ nolist
       include 'common.h'
       include 'inputcom.h'
       include 'pointer.h'
       include 'circom.h'
cdir$ list


      if (circtype(icur) .eq. 'linear') then
c-------apply a piecewise-linear current
        curb1(icur) = zero
        do 200 k=1,ncurpiec
          kk=k
          if (currtime(k,icur) .gt. t) goto 300
  200   continue
  300   alf = (currtime(kk,icur) - t) /
     %        (currtime(kk,icur) - currtime(kk-1,icur))
        current1(icur) = alf      * currvalu(kk-1,icur) +
     %                   (one-alf) * currvalu(kk  ,icur)

        if (.not. ciractv(icur)) then
c---------on the first pass through this routine,
c--------- initialize the current0 for t - dtold/2
          current0 = currvalu(1,icur)
          ciractv(icur) = .true.
        endif

      elseif (circtype(icur) .eq. 'sine') then
c-------current goes as the sine function
        curb1(icur) = zero
        current1(icur) = currvalu(1,icur) *
```

```
     %                  sin(0.5d0 * pi * t / currtime(1,icur))

      elseif (circtype(icur) .eq. 'sine2') then
c-------current goes as the sine squared function
        curb1(icur) = zero
        current1(icur) = currvalu(1,icur) *
     %                   sin(0.5d0 * pi * t / currtime(1,icur)) *
     %                   sin(0.5d0 * pi * t / currtime(1,icur))

      elseif (circtype(icur) .eq. 'rlc') then
c-------solve the single-loop LRC circuit equation
c------- with a time-centered scheme
c-------calculate the current at t + dt/2 (current1) from the value
c------- at t - dtold/2 (current0) and at t (current).
        curb1(icur) = zero

c-------save the total resistance
        resis = exres(icur) + fres(icur)

        if (.not. ciractv(icur)) then
c---------on the first pass through this routine,
c--------- initialize the current0 for t - dtold/2
          cl = 2.0d0 * rdt * exind(icur)
          current0 = ( edl(icur) - volt(icur) +
     %                 current(icur) * (resis + cl) ) / cl
          ciractv(icur) = .true.
        endif

c-------current1 at t + dt/2
        cl = rdt * exind(icur)
        current1(icur) = ( volt(icur) - edl(icur) -
     %                    current0*(0.5d0*resis-cl) ) / (0.5d0*resis+cl)

      elseif (circtype(icur) .eq. 'fuzeload') then
c-------first current through fuze
c-------time dependent resistivity
        if ( t .le. scrtch(805) ) then
          fresis = scrtch(801)
        elseif ( t .le. scrtch(806) ) then
          fresis = scrtch(801)+(scrtch(802)-scrtch(801))
     %          *(t-scrtch(805))/(scrtch(806)-scrtch(805))
        elseif ( t .le. scrtch(807) ) then
          fresis = scrtch(802)+(scrtch(803)-scrtch(802))
     %          *(t-scrtch(806))/(scrtch(807)-scrtch(806))
        elseif ( t .le. scrtch(808) ) then
          fresis = scrtch(803)+(scrtch(804)-scrtch(803))
     %          *(t-scrtch(807))/(scrtch(808)-scrtch(807))
        else
          fresis = scrtch(808)
        endif

        resis = exres(icur)

        if (.not. ciractv(icur)) then
```

```
c---------on the first pass through this routine,
c--------- initialize the current0 for t - dtold/2
          cl = 2.0d0 * rdt * exind(icur)
          dl = 2.0d0 * rdt * scrtch(800)
          curb0 = ( edl(icur) +
     %              curb(icur) * (fresis + dl) ) / dl
          current0 = curb0 + ( edl(icur) +
     %               (current(icur)-curb(icur))*(resis + cl) )/cl
          ciractv(icur) = .true.
        endif

c-------current1 at t + dt/2
        cl = rdt * exind(icur)
        dl = rdt * scrtch(800)
        curb1(icur) = (-edl(icur)-curb0*(0.5d0*fresis-dl))
     %               /(0.5d0*fresis+dl)
        current1(icur) = curb1(icur) + (-edl(icur) - (0.5d0*resis-cl)
     %                    *(current0-curb0) )/(0.5d0*resis+cl)

      elseif (circtype(icur) .eq. 'rlc2loop') then
c-------solve the two-loop LRC circuit equation
c------- with a time-centered scheme
c-------calculate the current at t + dt/2 (curb1, current1) from
c------- the value at t - dtold/2 (curb0, current0) and at t (curb,
c------- current). assume curb and current are zero on the first pass

c-------time-dependent tres obtion for mcg modeling
        if (volttype(icur) .eq. 'mcg') then
          trest = tres(icur) * t
        else
          trest = tres(icur)
        endif

        if (.not. ciractv(icur)) then
c--------on the first pass through this routine,
c-------- initialize the currents at t + dt/2 and t - dtold/2
          cl  = exind(icur) / (exind(icur) + tind(icur))
          dl  = tind(icur)  / (exind(icur) + tind(icur))
c---------current from the bank
          curb1(icur) = curb(icur) + 0.5d0*dtav*(
     %                volt(icur) - edl(icur)*(one-cl) -
     %                curb(icur)*(bres(icur)+trest*cl) -
     %                current(icur)*(exres(icur)*dl-trest*cl) ) /
     %                (bind(icur) + tind(icur)*cl)
          curb0 = 2.d0*curb(icur) - curb1(icur)
c---------current to the problem domain
          current1(icur) = current(icur) + (curb1(icur)-curb(icur))*dl -
     %         0.5d0*dtav*(
     %                   edl(icur) +
     %                   current(icur)*(exres(icur)+trest) -
     %                   curb(icur)*trest ) /
     %                   ( tind(icur) + exind(icur) )
          current0 = 2.d0*current(icur) - current1(icur)
          ciractv(icur) = .true.
```

```
        else
c---------currents at t + dt/2
          cbp = 0.5d0 * bres(icur)  + rdt * bind(icur)
          cbm = 0.5d0 * bres(icur)  - rdt * bind(icur)
          ctp = 0.5d0 * trest       + rdt * tind(icur)
          ctm = 0.5d0 * trest       - rdt * tind(icur)
          cp  = 0.5d0 * exres(icur) + rdt * exind(icur)
          cm  = 0.5d0 * exres(icur) - rdt * exind(icur)
          cr  = rdt * (exres(icur)*tind(icur)-exind(icur)*trest)
          curb1(icur) = ( volt(icur)*(ctp+cp) - edl(icur)*ctp -
     %                   current0*cr - curb0*(cbm*ctp + cp*(cbm+ctm)) )/
     %                  ( cbp*ctp + cp*(cbp+ctp) )
          current1(icur) = ( curb1(icur)*ctp + curb0*ctm -
     %                       current0*(ctm+cm) - edl(icur) ) /
     %                       ( ctp + cp )
        endif

       elseif (circtype(icur) .eq. 'pfn') then
         call cirpfn(icur)

       else
c-------don't change the current

       endif

       return
       end
```

C.6 Bfldzbc.F

```
       subroutine bfldzbc(bz)

c-----fill non-neighbor boundary ghost cells with bz values

cdir$ nolist
       include 'common.h'
       include 'bccommon.h'
       include 'circom.h'
       include 'inputcom.h'
       include 'pointer.h'
       include 'celcptrs.h'
cdir$ list
       dimension bz(0:ip2,0:jp2)
       dimension temp(1:mxmats)
       pointer ( kpbznbr , bznbr )

       lbz = lindex( bz )

c-----ghost cell data creation

       do 100 i=1,4
         ibdry = ibzseq(i,lblk)
         lnbr = knbr(ibdry,lblk)
```

```
        if (magzbc(ibdry,lblk) .eq. 'conductr') then
c                              from       to       range
          call bcpntrs(ibdry,this,edge,this,ghst,all,cell)
          call bcxtcsc(bz,rc,bz,rc,1)
c         if (matbc(ibdry,lblk).ne.'       ') then
c           do 255 imat = 1,nummats
c              temp(imat) = 0.
c 255        continue
c            temp(nmatbc(ibdry,imat)) = 1.
c            call bcsetscm(areafrac,temp)
c            call bcsetscm(flxzfrac,temp)
c          else
c            call bcxtcscm(areafrac,rc,areafrac,rc,1)
c            call bcxtcscm(flxzfrac,rc,flxzfrac,rc,1)
c          endif
        elseif (magzbc(ibdry,lblk) .eq. 'insulatr') then
          if (currcir(ibdry,lblk).eq. 0) then
c                              from       to       range
            call bcpntrs(ibdry,this,edge,this,ghst,all,cell)
            call bcxtcsc(bz,rc,bz,rc,1)
c           call bcxtcscm(areafrac,rc,areafrac,rc,1)
c           call bcxtcscm(flxzfrac,rc,flxzfrac,rc,1)
c                              from       to       range
            call bcpntrs(ibdry,this,ghst,this,ghst,all,cell)
            call bcmltsc(bz,bz,-1.d0)
c           call bcmltscm(areafrac,areafrac,-1.d0)
c           call bcmltscm(flxzfrac,flxzfrac,-1.d0)
          else
c                              from       to       range
            call bcpntrs(ibdry,none,none,this,ghst,all,cell)
            call bccurnt(current(currcir(ibdry,lblk)),rc,bz,m0,tpi)
c           do 200 imat = 1,nummats
c200           temp(imat) = 0.
c           temp(nmatbc(ibdry,lblk)) = 1.
c           call bcsetscm(areafrac,temp)
c           call bcsetscm(flxzfrac,temp)
          endif
        elseif (magzbc(ibdry,lblk) .eq. 'matinsul') then
          if (currcir(ibdry,lblk).eq. 0) then
c                              from       to       range
            call bcpntrs(ibdry,this,edge,this,ghst,all,cell)
            call bcxtcsc(bz,rc,bz,rc,1)
c                              from       to       range
            call bcpntrs(ibdry,this,ghst,this,ghst,all,cell)
            call bcmltsc(bz,bz,-1.d0)
          else
c                              from       to       range
            call bcpntrs(ibdry,none,none,this,ghst,all,cell)
            call bcmtcur(current(currcir(ibdry,lblk)),conimat,rc,bz,m0
     %                  ,pi,cyl)
c           do 205 imat = 1,nummats
c205           temp(imat) = 0.
c           temp(nmatbc(ibdry,lblk)) = 1.
```

```
c           call bcsetscm(areafrac,temp)
c           call bcsetscm(flxzfrac,temp)
          endif
        elseif (magzbc(ibdry,lblk) .eq. 'ininsult') then
          if (currcir(ibdry,lblk) .ne. 0) then
c                               from       to       range
            call bcpntrs(ibdry,none,none,this,edge,all,cell)
            call bccurnt(current(currcir(ibdry,lblk)),rc,bz,m0,tpi)
          endif
          call setnbrb(lnbr)
          kpbznbr = lpoint(lbz,lnbr)
c                               from       to       range
          call bcpntrs(ibdry,nebr,edge,this,ghst,all,cell)
          call bccpysc(bznbr,bz)
        elseif (magzbc(ibdry,lblk) .eq. 'axis') then
c                               from       to       range
          call bcpntrs(ibdry,this,edge,this,ghst,all,cell)
          call bccpysc(bz,bz)
          call bcxtcsc2(rc,bz,rc,-1)
c         call bcxtcscm(areafrac,rc,areafrac,rc,-1)
c         call bcxtcscm(flxzfrac,rc,flxzfrac,rc,-1)
c                               from       to       range
          call bcpntrs(ibdry,this,ghst,this,ghst,all,cell)
          call bcmltsc(bz,bz,-1.d0)
c         call bcmltscm(areafrac,areafrac,-1.d0)
c         call bcmltscm(flxzfrac,flxzfrac,-1.d0)
        elseif (magzbc(ibdry,lblk) .eq. 'symmetry') then
c                               from       to       range
          call bcpntrs(ibdry,this,edge,this,ghst,all,cell)
          call bccpysc(bz,bz)
c         call bccpyscm(areafrac,areafrac)
c         call bccpyscm(flxzfrac,flxzfrac)
        elseif (magzbc(ibdry,lblk) .eq. 'contnutv') then
c                               from       to       range
          call bcpntrs(ibdry,this,edge,this,ghst,all,cell)
          call bcxtcsc(bz,rc,bz,rc,1)
c         call bcxtcscm(areafrac,rc,areafrac,rc,1)
c         call bcxtcscm(flxzfrac,rc,flxzfrac,rc,1)
        elseif (magzbc(ibdry,lblk) .eq. 'specfied') then
c                             from      to        range
          call bcpntrs(ibdry,none,none,this,ghst,all,cell)
          call bcsetsc(bz,bzbdy(ibdry,lblk))
c         do 210 imat = 1,nummats
c210         temp(imat) = 0.
c         temp(nmatbc(ibdry,lblk)) = 1.
c         call bcsetscm(areafrac,temp)
c         call bcsetscm(flxzfrac,temp)
        elseif (magzbc(ibdry,lblk) .eq. 'fluxpri' ) then

            call bcpntrs(ibdry,none,none,this,ghst,all,cell)
            call bccurnt(fluxcurr(1),rc,bz,m0,tpi)

        elseif (magzbc(ibdry,lblk) .eq. 'fluxsec' ) then
```

```
            call bcpntrs(ibdry,none,none,this,ghst,all,cell)
            call bccurnt(fluxcurr(2),rc,bz,m0,tpi)

         elseif (magzbc(ibdry,lblk) .eq. 'none') then
c---------do nothing
         elseif ((lnbr .ne. 0) .and.
     %          (magzbc(ibdry,lblk) .eq. '        ')) then
           call setnbrb(lnbr)
           kpbznbr = lpoint(lbz,lnbr)
c                              from       to       range
           call bcpntrs(ibdry,nebr,edge,this,ghst,all,cell)
           call bccpysc(bznbr,bz)
c          call bccpyscm(afracnbr,areafrac)
c          call bccpyscm(flxzcnbr,flxzfrac)
         else
           stop 'bfldzbc'
         endif
  100 continue

      return
      end
```

C.7 Ciredln.F

```
      subroutine ciredln(delin)
c-----compute E dot dl and delin based on voltmodl
c-----Note: numcirc  LE 0 is checked in cirset

cdir$ nolist
       include 'common.h'
       include 'inputcom.h'
       include 'pointer.h'
       include 'circom.h'
cdir$ list

      real*8 delin(mxcirc)

      if (voltmodl .eq. 'int_edl') then

        do 100 icur = 1,numcirc
c---------zero E dot dl and delin for all external circuits
          edl(icur) = zero
          delin(icur) = zero
          do 100 ii = 1,numbdycr(icur)
c---------compute the boundary voltage and change of external inductance
            lblk = lblkcur(icur,ii)
            call setblk
            call bfldzbc(bzn)
            call ciredl(edl(icur),icur,ii)
            call cirind(delin(icur),icur,ii)
            call ciredl2b(flux)   ! may need flux for output, not used for this voltmodl
  100   continue
        oldefldt = tbe + tbhyde + totheat  ! in case we change to voltmodl='energy'
```

```
        oldflux = flux

      elseif ((voltmodl .eq. 'flux' .or. voltmodl .eq. 'energy') .and.
     %         numcirc .eq. 1 ) then
c-------zero E dot dl and delin for all external circuits
        icur = 1
        edl(icur) = zero
        delin(icur) = zero
        if (voltmodl .eq. 'flux') then
c---------calculate the potential along all external boundaries
c----------except circuit boundary
         call ciredl2(edl(icur),icur)




#ifdef NEWCIRC
        elseif (voltmodl .eq. 'energy') then
          call ciredl3(edl(icur),icur)
#else
        else
          call ciredl3(edl(icur),icur)
#endif
        endif
        do 110 ii = 1,numbdycr(icur)
          lblk = lblkcur(icur,ii)
          call setblk
          call bfldzbc(bzn)
          call cirind(delin(icur),icur,ii)
 110    continue

      elseif ( numcirc .gt. 1 ) then
        write (buffer, 5005)
 5005   format (
     &  'Error 1060: Cannot use energy or flux calc for > 1 circuit. ')
        call writemsg (buffer)
        stop 'ciredln1: Cannot use energy or flux calc for > 1 circuit.'

      elseif (voltmodl .eq. 'fluxcomp' .and. numcirc .eq. 1 ) then

        icur = 1
        edl(icur) = zero
        delin(icur) = zero
        call ciredl4(edl(icur),icur)
        do 111 ii = 1,numbdycr(icur)
          lblk = lblkcur(icur,ii)
          call setblk
          call bfldzbc(bzn)
          call cirind(delin(icur),icur,ii)
 111      continue



      else
```

```
        write (buffer, 5006) voltmodl
 5006   format ('Error 1060: Bad voltmodl =',a8)
        call writemsg (buffer)
        stop 'ciredln2: Bad voltmodl'
      endif

      return
      end
```

C.8 Circom.h

```
*cd circom
c-----circuit common
c-----fuse common
      common /circ/ tfit(nbk),bfit(nbk-nord),fres(mxcirc)
     % ,faspr(mxcirc),fsie(mxcirc),fmas(mxcirc)
     % ,fresmax(mxcirc),fdens(mxcirc),flength(mxcirc)
     % ,fwidth(mxcirc),fthick(mxcirc),oldflux
c-----pulse-forming network common
      common /circ/ vdot(nvals),vlt(nvals),crnt(nvals),pfnLi(nvals)
     % ,pfnCi(nvals),pfnRi(nvals),pfnLp(nvals),pfnRp(nvals)

      common /circ/ fluxcurr(4)
     %             ,addsecpathind
     %             ,blkflux(mxblktot), cirflux(mxmats)
     %             ,ofluxpri,ofluxsec, totcurr
     %             ,cirlpri, cirlsec, fluxen
     %             ,crowrad(2),crowval(2)
     %             ,oldlpri,oldlsec,crowresist
     %             ,cirarea(mxmats),cirmag(mxmats)
     %             ,radius(mxmats*2),edotdlcop,edotdlsec
     %             ,velo(mxmats*2)

c-----general circuit common
      common /circ/ current(mxcirc),current1(mxcirc),curb(mxcirc)
     % ,curb1(mxcirc),exind(mxcirc),exres(mxcirc)
     % ,exldt(mxcirc),capac(mxcirc),volt(mxcirc)
     % ,edl(mxcirc),exresnrg(mxcirc),enfus(mxcirc)
     % ,encir(mxcirc),dtold,xcrnew(mxcirc,mxcirc,mxij)
     % ,ycrnew(mxcirc,mxcirc,mxij),currvalu(ncurpiec,mxcirc)
     % ,currtime(ncurpiec,mxcirc),bres(mxcirc)
     % ,volttime(ncurpiec,mxcirc),voltvalu(ncurpiec,mxcirc)
     % ,tres(mxcirc),bind(mxcirc),tind(mxcirc),oldefldt !recent fix
     % ,encirtot,tefrmcir,crntscl,voltscl,enfrmcir(mxcirc) !1

      common /circi/ numcirc,numbdycr(mxcirc),lblkcur(mxcirc,mxcirc)
     % ,ibdycur(mxcirc,mxcirc),fluxmat(mxmats)
     % , interface(mxmats*2),numinter
     % ,currcir(4,mxblktot),nloops
      integer currcir,fluxmat(mxmats),interface
      common /circi/ ciractv(mxcirc),calcrvolt,ciron
      logical ciron, ciractv,calcrvolt
c-----enddeck circom
```

Author Index

Vita

Kyle Peterson was born in Chicago, Illinois on September 8, 1975. He graduated from Fort Calhoun High School in May of 1994. In August of 1994, Kyle entered North Park University in Chicago, Illinois where he received a Bachelor of Science degree in Physics in May of 1998. In the fall of that same year, he enrolled in the Master's program in Physics at the University of Tennessee Space Institute. Two years later, Kyle received his Master's degree in May of 2000. Kyle immediately began pursuing his doctoral research in high density Z-pinch plasmas at the Decade facility at Arnold Engineering Development Center. In August of 2003, Kyle was awarded a Doctorate of Philosophy in Physics at the University of Tennessee Space Institute.

Kyle is presently employed at Sandia National Laboratories and researching Inertial Confinement Fusion concepts.