Development of Magnetic Nozzle Simulations for Space Propulsion Applications

Colin C. Glesner

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Aerospace Engineering

Bhuvana Srinivasan, Chair
Wayne Scales
Colin Adams

October 26, 2016
Blacksburg, Virginia

Keywords: Numerical Simulation, Plasma, Ideal MHD, Resistive MHD, Divergence Error, Discontinuous Galerkin, Magnetic Nozzle, Detachment

Development of Magnetic Nozzle Simulations for Space Propulsion Applications

Colin C. Glesner

(ABSTRACT)

A means of space propulsion using the channeling of plasma by a divergent magnetic field, referred to as a magnetic nozzle has been explored by a number of research groups. This research develops the capability to apply the high order accurate Runge-Kutta discontinuous Galerkin numerical method to the simulation of magnetic nozzles. The resistive magnetohydrodynamic model of plasma behavior is developed for these simulations. To facilitate this work, several modeling capabilities are developed, including the implementation of appropriate inflow and far-field boundary conditions, the application of a technique for correcting errors that develop in the divergence of the magnetic field, and a split formulation for the magnetic field between the applied and the perturbed component. This model is then applied to perform a scaling study of the performance of magnetic nozzles over a range of β_k and R_m. In addition, the effect of the choice of simulation domain size is investigated. Finally, recommendations for future work are made.

This work was supported in part by VSGC

Development of Magnetic Nozzle Simulations for Space Propulsion Applications

Colin C. Glesner

(GENERAL AUDIENCE ABSTRACT)

In order for human kind to expand our reach beyond the Earth, and create a meaningful human presence throughout the solar system, it is almost certain that the current means of propelling spacecraft will be insufficient. With current propulsion systems that derive their energy from chemical reactions, there is an upper limit to how energetic a chemical reaction can be. This fact, together with the fact that a spacecraft must carry all of its fuel with it, means that interplanetary trips with chemical rockets require an impractically large amount of fuel. Other means of propulsion which do not have an upper limit on the energy that can be used to power the spacecraft are available. One such method, referred to as a Helicon Thruster, or a Magnetic Nozzle thruster, works as follows: first, an inert gas is introduced into the thruster. Then, a powerful antenna is used to break electrons off of the atoms of the gas, which generates ions (atoms that have had one or more of their electrons removed). This results in what is know as a plasma, or a gas made up entirely of charged particles. Common examples of plasma are lightning bolts, the gas inside of fluorescent lights, and the sun. Plasmas differ from normal gasses in that they feel the effects of magnetic and electric fields, and can conduct electricity. Once this plasma has been generated inside of the thruster, another antenna may be used to further heat up the plasma. Then, the plasma is directed by a strong magnetic field out the back of the thruster, propelling the spacecraft forward. This propulsion technique requires much less fuel than traditional chemical approaches, making long interplanetary trips more practical. A number of research questions need to be answered before this propulsion method can be put into practice. This thesis contributes to this effort by building the tools necessary to run a computer simulation of the magnetic field that is used to channel the plasma used by this propulsion technique.

This work was supported in part by VSGC

Dedication

To my loving and patient wife

Acknowledgments

I would like to acknowledge the advice, encouragement, and shared enthusiasm for the research at hand that was given to me by my adviser, Dr. Bhuvana Srinivasan. I would also like to thank the members of my committee, Dr. Colin Adams and Dr. Wayne Scales for their feedback and participation. I would also like to thank the members of my research group, in particular Yang Song and Petr Cagas for their feedback as I prepared my defense and for their help in thinking through research problems through the years. I would also like to thank my parents for their support as I was completing this degree. Most importantly I would like to thank my wife, Ida Cossitt, for her love, support, encouragement and patience as I finished my degree.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

As efforts to expand the human presence in space continue, the development of new and better propulsion systems remains a research area of fundamental importance; two goals of particular importance are increasing payload fractions and reducing time of flight for prospective interplanetary missions. Accomplishing these goals translates into developing high thrust, high specific impulse propulsion systems. Chemical propulsion systems are the most extensively developed thus far, and remain the only viable means of reaching orbit. The largest chemical rockets are capable of generating thrusts in excess of $10^6 N$, but they are limited by the energy density of chemical propellants to specific impulses on the order of hundreds of seconds, placing fundamental restrictions on the mass fraction available for payloads.

In order to obtain a higher specific impulse, it is necessary to decouple the energy source for propulsion from the mass that is being used for propulsion. For example, the heat from nuclear fission can be transferred to hydrogen gas, or electric power generated by solar cells can be used to accelerate an ionized gas by passing it through

an electric field [1]. In these scenarios, the amount of energy that can be imparted to the propellant, and consequently the specific impulse that can be achieved, becomes a technical rather than a fundamental limitation.

One proposed method for achieving high specific impulse, high thrust space propulsion is to use magnetic fields to confine a plasma that is generated by electromagnetic heating of an inert gas. This is in contrast to other electric propulsion concepts in which arcing currents between electrodes are used to generate a plasma. In this concept, because the plasma does not come into contact with the thruster, problems such as electrode erosion can be avoided, and the thruster can operate for long periods at high power levels. Examples of this propulsion concept currently under development include the applied-field magnetoplasmadynamic thruster [2] [3], the helicon thruster [4] [5] [6], and the Variable Specific Impulse Magnetoplasma Rocket (VASIMR) [7]. A well developed example of this type of thruster is the Variable Specific Impulse Magneto-plasma Rocket (VASIMR) that is being developed by the Ad Astra Rocket Company in partnership with NASA. The most recent iteration of the VASIMR concept, the VX-200, has been successfully operated at a power level of 100 kW and a specific impulse of 5000s using argon as the propellant [8].

In order for the plasma to be effectively contained by the magnetic fields while inside the propulsion device, it must be strongly magnetized. In the context of a plasma, this means that the individual electrons and ions of the plasma are gyrating around a given magnetic field line, and are only able to deviate a small distance from the magnetic field line to which they are coupled. Although strong magnetization is desirable while the plasma is being accelerated through the device, once the plasma has left the propulsion device, the plasma must somehow become 'detached' from the magnetic field lines that previously contained it. Because magnetic field lines must close on themselves, plasma that remains coupled to the field would be carried back towards the spacecraft, resulting in spacecraft charging and no net momentum transfer.

A number of detachment mechanisms have been proposed, but an understanding of the dominant detachment mechanisms remains an open research area; particularly with respect to the plasma regimes and magnetic nozzle configurations encountered in relevant experimental devices. The detachment mechanisms that have been proposed fall broadly into two categories: collisional and collisionless.

Collisional detachment theories generally propose that inter-particle collisions serve to jostle particles; causing them to jump from one field line to the next, seen macroscopically as plasma resistivity [9]. Alternatively, it has been proposed that electron-ion recombination may play a role in detachment [10].

A broad range of collisionless detachment mechanisms have also been put forward. One proposal is that the plasma plume, by developing internal currents, could alter the magnetic field around the thruster. This would effectively 'stretch out' the magnetic field of the nozzle, convecting it with the plasma to infinity and allowing the plume to leave the spacecraft without demagnetizing [11]. Another theory notes that ions, due to their relatively large mass, demagnetize much more easily than electrons. In this scenario demagnetized ions attempt to leave the nozzle while the still magnetized electrons attempt to follow the curving field lines. The result is a separation of charge, and the creation of an accompanying electric field. This electric field then couples the ions and electrons, serving as a means by which energy is transferred, giving electrons sufficient energy to detach [12]. Alternatively, it can be hypothesized that magnetic field line reconnection may play a role in detachment. Magnetic reconnection is a phenomenon in which crossing magnetic field lines sever and re-attach, forming a new, lower energy field topology. The reconnection results in the creation of a free loop of magnetic field, and the released magnetic energy is imparted as kinetic energy to the plasma containing the free loop. This is the same mechanism that is considered responsible for coronal mass ejections [13]. Magnetic reconnection provides another means by which plasma could leave the spacecraft without the plasma itself demagnetizing.

In the experimental work that has been performed to date it has been difficult to definitively identify what detachment mechanisms are present [14].

Fundamental research on the topic of plasma detachment remains extremely important while uncertainty about the governing physics persists. Once better understood, a more rigorous approach to the design and optimization of magnetic nozzle thrusters becomes possible, greatly improving the prospects for this space propulsion concept and space exploration requiring new, better means of propulsion.

1.2 Work Performed

This research is carried out using the Washington Approximate Riemann Plasma Solver (WARPX) code. WARPX is a well validated and benchmarked code developed at the Computational Plasma Dynamics Laboratory at the University of Washington [15][16][17]. The numerical method used in this work is the discontinuous Galerkin method with an explicit Runge-Kutta time advance (RKDG) [18]. RKDG is chosen for this work because it is well suited to modeling the behavior of equation systems which form discontinuities, while at the same time providing the potential to capture the complex dispersive behavior of the plasma fluid system. A detailed introduction to the RKDG technique is provided in Chapter 2. A concise overview of the WARPX code is given in Section A.1.

The modeling approach used to perform this work is discussed in Chapter 3. The basic physical model applied is that of magnetohydrodynamics (MHD), a physical model of a plasma which treats the constituent particle populations as a single fluid. The parameter regime of interest for this work suggests that resistivity, a dissipative effect in plasmas, needs to be included in the model, so resistive MHD is used. These physical models are discussed in Sections 3.1 and 3.2. In order to develop the capability to perform simulations in which a plasma is injected into a diverging magnetic field, it is necessary to first develop an adequate inflow boundary condition. This boundary condition is discussed in Section 3.5. The simulations often develop negative pressures very early in their evolution. Several modeling techniques are developed and implemented to improve the ability of simulations to avoid generating negative pressures, which has resulted in a significant increase in the range of simulations that can be performed, but has not entirely eliminated the issue. The first of these is the implementation of a technique for reducing the $\nabla \cdot \mathbf{B}$ error that generally tends to develop in numerical solutions to the MHD equations. This technique is referred to as Hyperbolic Divergence Cleaning. This technique and its implementation in this research is discussed in Section 3.3. Another modeling technique developed for this work separates the magnetic field into a static, applied component, and a dynamic, perturbed component. This is believed to be an advantageous approach for solving simulations in which variations in magnetic field are small compared to the magnitude of the magnetic field itself. Additionally, this is a natural formulation for a simulation in which an externally applied magnetic field, in this case that of the magnetic nozzle, is applied for all time. This technique and its implementation are discussed in Section 3.4. It is also determined that a passive, far-field boundary

condition that could function in the presence of significant gradients in the magnetic field is required. The technique developed to accomplish this is discussed in Section 3.6.

A discussion and analysis of some preliminary simulations of magnetic nozzles is presented in Chapter 4. Simulations are attempted over a broad range of conditions; a subset of which produce useful results for analysis. The remainder of the simulations encounter difficulties with negative pressure issues as described above. Section 4.1 describes the simulations which where carried out. For the successful simulations, the variation of several aspects of the plasma plumes over the range of simulation conditions is considered: the perturbation of the initial magnetic field, the degree of radial expansion of the plumes, and the change in the energy contained in the plume. This analysis is provided in Section 4.2. Finally, in addition to investigating the effect of changing simulation parameters that have physical significance, Section 4.3 details the stark effect of altering a purely non-physical parameter of the simulation: the extent of the simulation domain in the radial direction, and provides a preliminary explanation as to the cause of this behavior.

Lastly, Chapter 5 provides a summary of the findings from this work and details the author's recommendations for future work to improve the ability of this model to simulate magnetic nozzles. The work completed thus far develops the necessary tools and serves as a proof of concept.

Chapter 2

Numerical Methods

Contents

The discontinuous Galerkin (DG) method, (sometimes referred to as discontinuous Galerkin Finite Element Method), is a numerical method for the solution of partial differential equations. It combines aspects of both Finite Volume Methods and traditional Finite Element Methods in an effort to obtain the advantages of each scheme and to leave aside the shortcomings. The discontinuous Galerkin method was first introduced in the 1970s [19] but has recently experienced tremendous growth as the methods were refined and improved to be able to handle a wider variety of problems.

There are, generally speaking, several important properties of the discontinuous Galerkin numerical method:

- The solutions are piece-wise smooth, often polynomial, but discontinuous between elements.
- Boundary conditions and interface continuity are enforced only weakly.
- All operators are local.
- It is well suited to local variation in the order of accuracy and resolution of the solution.

Of these characteristics, the last two are worth highlighting. The locality of the operations in DG means that the "stencil-size", the neighborhood of cells that any given computational operation requires information from, is never larger than the nearest neighbor elements. This is in contrast to finite volume methods which require an ever larger stencil size to reconstruct higher order estimates of the solution at the cell edges to perform flux reconstruction. One of the primary benefits of this restricted stencil size is that parallelization of DG simulations is very efficient [20]. When conveying information from one processing node to another only the outside boundary, a single layer of cells in each simulation domain on each processor must have their information transferred to neighboring processors, regardless of the spatial order of the simulation. The last characteristic mentioned, the potential for purely local refinement of a solution, is one of the greatest strengths of the DG method. It is much easier to do refinement of both mesh-size and spatial order with DG than with any other method.

2.1 Overview of Key Aspects

An overview of the derivation and key elements of the discontinuous Galerkin Scheme is now presented.

2.1.1 Derivation of Modal DG

The 1-D scalar conservation law is

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad x \in [L, R] = \Omega \quad , \tag{2.1}$$

where u is the conserved quantity, f is some (generally non-linear) function defining the flux of the conserved quantity, and Ω is the domain on which the PDE is defined. A unique solution to this PDE requires the definition of an initial condition, given by

$$u(x, 0) = u_0(x) \quad ,$$

and a boundary condition specified for inflow boundaries,

$$\begin{aligned} u(L, t) &= g(t) \quad \text{if} \quad a \geq 0 \quad , \\ u(R, t) &= g(t) \quad \text{if} \quad a \leq 0 \quad . \end{aligned}$$

The domain Ω is discretized into K sub-intervals defined by $x \in [x_l^k, x_r^k] = \mathbf{D}^k$. These sub-intervals share their end-points, but are not overlapping, so that $x_r^{k-1} = x_l^k,\ x_r^k = x_l^{k+1}$, *etc.* On each sub-interval, the Ansatz that the solution to Eq. (2.1) can be represented locally in each the k-th subinterval $\mathbf{D}_k$ as a polynomial of order N is taken. This is expressed as

$$x \in \mathbf{D}^k \ : \ u_h^k(x, t) = \sum_{i=0}^{N} \hat{u}_i^k(t) \psi_i(x) \quad , \tag{2.2}$$

where $u_h^k(x, t) \simeq u(x, t), x \in [x_l^k, x_r^k]$, is the approximate solution to the differential equation in the k-th sub-interval. This is referred to as a modal representation of the solution [21], in which the approximate solution is expressed as a linear combination of some polynomial basis $\psi_i(x)$ which is constant in time, with $\hat{u}_i(t)$ representing the coefficients that multiply the polynomial basis, and which *do* evolve over time. A discussion of the choice of polynomial basis is left until Section 2.1.3.

It is then assumed that the global solution is approximated by a piece-wise N-th order polynomial,

$$u(x, t) \simeq u_h(x, t) = \left\{ u_h^k(x, t) \text{ for } x \in [x_l^k, x_r^k], \quad k = 1, 2, 3, \ldots, K \right. \quad .$$

It is important to note that this definition applies over the endpoints of each sub-interval as well. This means that the boundaries between sub-intervals belong to two sub-intervals, and thus, have two values of u_h associated with them, that can, in general, differ. If Eq. (2.2) is substituted into Eq. (2.1), the residual,

$$\frac{\partial u_h}{\partial t} + \frac{\partial f(u_h)}{\partial x} = \mathcal{R}_h(x,t) \quad ,$$

is formed. In the solution of a partial differential equation, some metric must be defined by which the solution can be said to have converged. This is accomplished by requiring that the residuals vanish. To accomplish this, a set of $N+1$ test functions ϕ_j, are defined, and it is required that the residual be orthogonal to these test functions over each sub-interval, such that

$$\int_{\mathbf{D}^k} \left(\mathcal{R}^k\right) \phi_j dx = 0, \quad j = 0, 1, \ldots, N \quad .$$

This creates $(N+1)$ constraints on the local value of the residual for each of the K sub-intervals. It should be recognized that, assuming they can be evaluated, there are now $(N+1)(K)$ conditions for the evaluation of $(N+1)(K)$ unknown values, (the $\hat{u}_i^k$ values for each mode and sub-interval), providing a clear path to being able to show the existence and uniqueness of this approximate solution to a partial differential equation. It must now be determined how to transform these constraints into explicit equations for the expansion coefficients. Let the test functions be the same as the basis functions. This choice of matching basis and test functions is referred to as the Galerkin approach, and it is this choice that lends its name to DG [21].

One difficulty arrises here. Since the solution is discontinuous at the sub-interval edges, it is not clear that there is a well defined value of $\frac{\partial f(u_h)}{\partial x}$ at these locations; reconstructing a gradient in a meaningful way is difficult. To ameliorate this, integration by parts in the spatial variable, x, is performed to yield:

$$\int_{\mathbf{D}^k} \left(\frac{\partial u_h^k}{\partial t} + \frac{\partial f(u_h^k)}{\partial x} \right) \psi_j dx = 0 \quad ,$$

$$\int_{\mathbf{D}^k} \left(\frac{\partial u_h^k}{\partial t} \psi_j + \frac{\partial f(u_h^k)}{\partial x} \psi_j \right) dx = 0 \quad ,$$

$$\int_{\mathbf{D}^k} \left(\frac{\partial u_h^k}{\partial t} \psi_j - f(u_h^k) \frac{\partial \psi_j}{\partial x} \right) dx = - \left[f(u_h^k) \psi_j \right]_{x_l^k}^{x_r^k} \quad .$$

This results in the more tractable problem of only needing to reconstruct a value for the solution at the edges of the sub-intervals, rather than a derivative of the solution. It is the process of reconstructing the solution, (and in turn the flux that should be produced) that connects each of the K local solutions to each-other, forming a meaningful global solution, as well as being the means by which boundary conditions are imposed on the solution. This flux term that is constructed at the discontinuous boundary is referred to as numerical flux. Because of its role in constructing the global solution, the details of the numerical flux are clearly very significant. It is also where knowledge of the physical behavior of a simulation that is under consideration can and should help inform the formulation of the numerical methods being applied [21]. A brief overview of the topic of numerical flux is given in Section 2.1.5.

Substituting in the modal representation of the local solution for the time derivative term yields

$$\int_{\mathbf{D}^k} \left(\frac{\partial}{\partial t} \left(\sum_{i=0}^{N} \hat{u}_i^k(t) \psi_i(x) \right) \psi_j \right) dx = - \left[f^*(u_h^k) \psi_j \right]_{x_l^k}^{x_r^k} \quad ,$$

$$\mathcal{M}_{ij} \frac{\partial \hat{u}_i^k}{\partial t} = - \left[f^*(u_h^k) \psi_j \right]_{x_l^k}^{x_r^k} + \int_{\mathbf{D}^k} \left(f(u_h^k) \frac{\partial \psi_j}{\partial x} \right) dx \quad ,$$

where $\mathcal{M}_{ij} = \int_{\mathbf{D}^k} \psi_i \, \psi_j \, dx$, is referred to as the mass matrix. The remaining integral term still needs to be dealt with. This is accomplished using the numerical integration technique, *Gaussian-Quadrature*, which is detailed in Section 2.1.4. In this technique

the integrand is evaluated at several discrete locations and then a weighted sum of these values is performed to approximate the value of the integral [21]. With this addition, an explicit, semi-discrete expression for the $(N+1)(K)$ modes is given by

$$\frac{\partial \hat{u}_i^k}{\partial t} = \mathcal{M}_{ij}^{-1}\left[-\left[f^*(u_h^k)\psi_j\right]_{x_l^k}^{x_r^k} + \sum_{i=1}^{N}\left(w_i f(u_h^k(x_i))\frac{\partial \psi_j(x_i)}{\partial x}\right)\right] \quad . \tag{2.3}$$

The result is simply a system of ordinary differential equations. Let the right hand side of Eq. (2.3) be referred to as $\mathcal{L}(u_h(t))$, which is named the discrete spatial operator. This is the most general expression of the 1-D, Modal, discontinuous Galerkin method.

2.1.2 Generalizations

Very few changes are needed to generalize the above discussion to a system of equations and multiple dimensions. Let an arbitrary system of conservation laws be defined by

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{u}) = 0, \quad \mathbf{x} \in \Omega \quad , \tag{2.4}$$

where $\mathbf{u}$ is the vector of conserved quantities, $\mathbf{f}$ is some (generally non-linear) function defining the flux of the conserved quantity, and Ω is the domain on which the PDE is defined. Proceeding as in Section 2.1.1, assume that the solution can be well approximated by a spatially varying polynomial $\mathbf{u_h^k}$ that can be expressed as a linear combination of some polynomial basis on K local sub-domains $\mathbf{D_k} \in \Omega$. Just as before, require that the residual formed by the approximate solution be orthogonal to a set of test functions, generating the system of equations,

$$\int_{\mathbf{D_k}} \left(\frac{\partial \mathbf{u}_h^k}{\partial t}\psi_j + \nabla \cdot \mathbf{f}(\mathbf{u}_h^k)\psi_j\right) d\mathbf{x} = 0, \quad j = 0, 1, 2, \ldots, N \quad .$$

To side step the issue of defining the gradient of the flux at the cell edges, apply the divergence theorem to obtain

$$\int_{\mathbf{D_k}} \left(\frac{\partial \mathbf{u}_h^k}{\partial t}\psi_j - \mathbf{f}(\mathbf{u}_h^k)\nabla\psi_j\right) d\mathbf{x} = -\int_{\partial \mathbf{D_k}} \left(\hat{\mathbf{n}} \cdot \mathbf{f}^*(\mathbf{u}_h^k)\psi_j\right) d\mathbf{x} \quad .$$

Finally, substituting the modal form of the solution in, a system of explicit equations for the evolution of each mode in each cell is obtained:

$$\frac{\partial \hat{\mathbf{u}}_i^k}{\partial t} = \mathcal{M}_{ij}^{-1}\left[\int_{\mathbf{D_k}} \left(\mathbf{f}(\mathbf{u})\nabla\psi_j\right) d\mathbf{x} - \int_{\partial\mathbf{D_k}} \left(\hat{\mathbf{n}}\cdot\mathbf{f}^*(\mathbf{u}_h^k)\psi_j\right) d\mathbf{x}\right] \quad , \tag{2.5}$$

where $\mathcal{M}_{ij} = \int \psi_i\psi_j dx$. The first integral on the right hand side of Eq. (2.5) is a volume integral and the second is a surface integral. Both of these integral terms would need to be evaluated using some method of numerical integration.

It is interesting to note that the finite volume method can be shown to be a special case of the discontinuous Galerkin Method in which the polynomial basis used to represent the solution is limited to polynomials of 0-th order. Specifically, to the basis $\psi = 1$. If this is done then Eq. (2.3) becomes

$$\begin{aligned}
\frac{\partial \hat{u}_i^k}{\partial t} &= \mathcal{M}_{ij}^{-1}\left[-\left[f^*(u_h^k)\psi_j\right]_{x_l^k}^{x_r^k} + \sum_{i=1}^{N}\left(w_i f(u_h^k(x_i))\frac{\partial\psi_j(x_i)}{\partial x}\right)\right] \quad , \\
&= \mathcal{M}_{ij}^{-1}\left[-\left[f^*(u_h^k)\,(1)\right]_{x_l^k}^{x_r^k} + \sum_{i=1}^{N}\left(w_i f(u_h^k(x_i))\,(0)\right)\right] \quad , \\
\frac{\partial \bar{u}^k}{\partial t} &= -\frac{1}{x_r^k - x_l^k}\left(f^*|_{x_r^k} - f^*|_{x_l^k}\right) \quad ,
\end{aligned}$$

where $\bar{u}^k$ is the cell-averaged value in the k-th sub-interval of the domain. Thus, the finite volume method is just a special case of the discontinuous Galerkin method. The finite volume method recovers higher order accuracy by using information from neighboring cells to reconstruct a more accurate estimate of the values of u_h^k at the cell edges.

2.1.3 Polynomial Basis

The first important topic to clarify in the above section is exactly what kind of polynomial basis ψ_i should be chosen. There are no restrictions placed on the choice

of the basis by the scheme outlined in Section 2.1.1, but some options have distinct numerical advantages. Recall that in the final equation derived for the 1-D scalar equation, Eq. (2.3), that the mass matrix, $\mathcal{M}_{ij}$ needed to be inverted to obtain an explicit expression for the modal components of the solution. If for example, $\psi_i = 1,\ x,\ x^2,\ \ldots\ x^N$, is chosen as the polynomial basis, then $\mathcal{M}_{ij}$ quickly becomes very ill-conditioned [21]. The result of this is that, given finite digit arithmetic in an actual simulation, this ill-conditioning makes it impossible to recover accurate values of $\hat{u}_i^k$ when $\mathcal{M}_{ij}$ is inverted. Instead, if an orthogonal set of basis functions is chosen, this ill-conditioning problem vanishes. One such choice of orthogonal polynomials is the Legendre polynomials. The Legendre polynomials are a set of monic polynomials (polynomials for which the coefficient of the leading term is 1), which are orthogonal with respect to the L^2 inner product over the interval [-1,1] [22], defined as

$$\int_{-1}^{1} P_m(x) P_n(x)\, dx = \frac{2}{2n+1}\delta_{mn} \quad , \tag{2.6}$$

where δ_{mn} denotes the Kronecker delta function. This results in a diagonal mass-matrix and therefore a diagonal inverse mass-matrix, given respectively by

$$\mathcal{M}_{ij} = \frac{2}{2i+1}\delta_{ij} \qquad \text{and } \mathcal{M}_{ij}^{-1} = \frac{2i+1}{2}\delta_{ij} \quad . \tag{2.7}$$

The properties of the Legendre polynomials are well understood and a straightforward three term recursion relation known as Bonnet's recursion formula can be used to determine the value of the Legendre polynomial of order n at a value of x [22]. Given $P_0(x) = 1$, $P_1(x) = x$, Bonnet's recursion is

$$(n+1)P_{n+1} = (2n+1)x\, P_n - n\, P_{n-1} \quad . \tag{2.8}$$

The derivative of the Legendre polynomial of order n is given by [22]

$$P_n'(x) = \frac{n\left(xP_n(x) - P_{n-1}(x)\right)}{x^2 - 1} \quad . \tag{2.9}$$

Note that the above definition for the derivatives of the Legendre Polynomials is only valid on the open interval $(-1, 1)$. When the derivative at the end points, is desired, the proper equation is [22]

$$P_n'(1) = \frac{n(n+1)}{2} \qquad P_n'(-1) = (-1)^{n+1}\left[\frac{n(n+1)}{2}\right] \quad . \tag{2.10}$$

Other choices of basis functions are possible, but the property of orthogonality should be retained. For most cases however, the Legendre polynomials prove to be more than adequate [21].

2.1.4 Gaussian Quadrature

As detailed above in section 2.1.1, evaluation of the right hand side of Eq. (2.3) involves the evaluation of an integral over each sub-interval, for each mode of the scheme. This integration is desired to be done in an efficient and accurate manner. The method of choice for this is a technique referred to as *Gaussian Quadrature*. This method uses a judicious choice of both the weights of the integration scheme as well as the locations at which the integrand function is evaluated to obtain a method for which a linear combination of the integrand function at n points will produce an exact result for a function of order up to $2n-1$ [22]. The general definition of a quadrature function is

$$\int_a^b \omega(x) f(x) dx = \sum_{i=1}^{n} w_i f(x_i) \quad , \tag{2.11}$$

where $\omega(x)$ is a weighting function chosen based on the domain of integration and the function to be integrated. For each choice of weighting function $\omega(x)$, there is an associated set of polynomials which are defined over the domain of integration. The n nodes, x_i at which $f(x)$ is to be evaluated are the roots of the associated n-th order polynomial. The weight associated with the i-th node is given by integrating the Lagrange polynomial centered on the i-th node over the region of the quadrature

$$w_i = \int_a^b \prod_{\substack{1 \le j \le n \\ j \neq i}} \frac{x - x_j}{x_i - x_j} dx \tag{2.12}$$

where x_j is the j-th value in x_i. For this scheme, $\omega(x) = 1$ is chosen as the weighting function. This is a reasonable choice because the domain of integration in this

work is over some finite x-interval [a,b], and any physically meaningful solution to a conservation equation should not contain any singularities. Another way of stating this, is that this choice of weight function is appropriate if it is believed that the integrand is approximately polynomial in character over the interval [a,b]. The set of polynomials associated with this choice of $\omega(x)$ is the Legendre Polynomials, whose properties are detailed in section 2.1.3.

The Legendre Polynomials are only orthogonal over the interval $[-1, 1]$. In order to perform Gauss-Legendre quadrature over an arbitrary interval, such as the i-th sub-interval with domain $x \in [x_i, x_{i+1}]$, a change of variables must be made, so that the integral over a given cell is mapped to an equivalent integral over $[-1, 1]$. Note that all cells, regardless of size or location, are mapped to the same interval. Let x_i be the i-th cell edge, so that for a 1-dimensional scheme with n cells $i = 1, 2, \ldots, n+1$ there are $n + 1$ cell edges in total. Let the mapped variable be η. Then, define a function $x = g(\eta)$ such that $g(-1) = x_i$ and $g(1) = x_{i+1}$. A function that would accomplish this is

$$x = g(t) = \frac{x_{i+1} + x_i}{2} + \left(\frac{x_{i+1} - x_i}{2}\right)\eta \quad dx = g'(\eta)d\eta = \frac{x_{i+1} - x_i}{2}\, d\eta \quad . \tag{2.13}$$

Substituting into the expression for the integral over cell i yields

$$\int_{x_i}^{x_{i+1}} f(x)dx = \int_{-1}^{1} f(\, g(\eta)\,)g'(\eta)d\eta \quad , \tag{2.14}$$

$$= \int_{-1}^{1} f\left(\frac{x_{i+1} + x_i}{2} + \left(\frac{x_{i+1} - x_i}{2}\right)\eta\right)\frac{x_{i+1} - x_i}{2}d\eta \quad . \tag{2.15}$$

With this substitution it is now possible to perform quadrature for cell i

$$\int_{x_i}^{x_{i+1}} f(x)dx = \frac{x_{i+1} - x_i}{2}\sum_{k=1}^{m} w_k f\left(\frac{x_{i+1} + x_i}{2} + \left(\frac{x_{i+1} - x_i}{2}\right)\eta_k\right) \tag{2.16}$$

In Gauss-Legendre quadrature of order n, the node locations at which the integrand function is evaluated are the roots of the n-th order Legendre polynomial. For low order quadrature schemes, it is relatively straight forward to determine the roots

by hand and store these values in a table for access by the code implementing the numerical method. However, in practice it is desirable to be able perform quadrature of arbitrary order, so an algorithmic approach has been developed. Two considerations ease the evaluation of the roots of the Legendre polynomials. First, the roots are always symmetric. That is, $x_i = -x_{n-i+1}$ for $i = 1, 2, \ldots, n$, so that at most only $\frac{n}{2}$ or $\frac{n-1}{2}$, roots need to be evaluated for even and odd order polynomials respectively. Second, a reasonable analytic approximation of the roots exists, and can serve as an initial guess that can be efficiently refined to a desired tolerance by use of a Newton-Raphson root finding algorithm. The approximation for the roots of the Legendre polynomial of order n are [22]

$$x_i \approx \cos\left(\pi \frac{i - 1/4}{n + 1/2}\right) \text{ (roots in descending order)} \quad . \tag{2.17}$$

Starting with these initial approximations, the recursive expressions for the Legendre polynomials and their derivatives can be used for performing Newton-Raphson root finding.

For the special case of Gauss-Legendre quadrature, the weights w_i of the quadrature, instead of having to be found from the integral of the Lagrange polynomial, can be evaluated by the much simpler expression [22]

$$w_i = \frac{2}{(1 - x_i^2)\left[P_n'(x_i)\right]^2} \quad . \tag{2.18}$$

2.1.5 Numerical Flux Reconstruction

There are a wide range of methods for reconstructing the flux due to the values at the boundary of each sub-interval, as well as the global-boundary conditions of the equation [21]. In general, it should be required that $f^*(u_h^k(x_l^k))$ should be a function of $[u_h^{k-1}(x_r^{k-1}), u_h^k(x_l^k)]$, and $f^*(u_h^k(x_r^k))$ should be a function of $[u_h^k(x_r^k), u_h^{k+1}(x_l^{k+1})]$. It is desirable, for the definition of numerical flux to be consistent, that if $u_h^{k-1}(x_r^{k-1}) = u_h^k(x_l^k)$, $f^* = f(u_h)$, i.e., if the left and right hand side values of u_h are the same, then the numerical flux is equal to the standard flux that would be calculated for that value of u_h. There are a wide range of methods for determining the numerical flux, many of which have been developed for use with the finite volume method (FVM).

The method that will be implemented here is that of the Lax-Friedrichs flux. This is defined by

$$f^*(a,b) = \frac{f(a)+f(b)}{2} + \frac{|\lambda_{max}|}{2}\,\hat{\mathbf{n}}\cdot(a-b) \quad , \tag{2.19}$$

where a and b represent the interior and exterior values of u_h at the interface respectively, $\hat{\mathbf{n}}$ is the local unit normal at the interface we are considering, and $|\lambda_{max}|$ is the largest local eigenvalue of the flux Jacobian.

2.1.6 Limiting

Limiting is an area of tremendous importance in all higher order numerical schemes, but is not a topic that will be explored in depth here. Briefly, a measure of the oscillations in a solution, called the Total Variation (TV) is defined as

$$|u_h|_{TV} = \sum_{k=1}^{K} |u_h^{k+1} - u_h^k| \quad . \tag{2.20}$$

For stable, first-order numerical schemes, the numerical flux scheme applied is required to be what is called a monotone flux. For a monotone flux scheme, it will be guaranteed that [21]

$$\frac{d}{dt}|u_h|_{TV} \leq 0 \quad . \tag{2.21}$$

This is referred to as a Total Variation Diminishing (TVD) scheme. The Lax-Friedrichs flux scheme chosen for this work is one such monotone flux scheme, and is therefore a TVD scheme [21]. When solutions are represented using a higher-order basis, it is possible for large, spurious oscillations to be introduced to the solution, especially near large discontinuities in the solution (loosely analogous to Gibbs-phenomena). To overcome this, schemes referred to as *limiters* have been developed. Limiters will enforce that a solution is TVD. An acceptable limiting scheme must satisfy the following three constraints [21]

- It must be conservative.

- It must enforce a set of requirements about the slope of the solution in successive sub-intervals.
- It must retain the same formal order of accuracy as the overall scheme.

The limiting scheme implemented in this research is referred to as a component limiting scheme. This limiting scheme is independent of the equation system being solved.

2.1.7 Time Advancing the solution

As detailed in Section 2.1.1, a natural formulation for the DG method is in a semi-discrete, explicit form

$$\frac{\partial u_h(t)}{\partial t} = \mathcal{L}(u_h(t)) \quad , \tag{2.22}$$

where $\mathcal{L}(u_h(t))$ represents the discretization of the spatial derivatives at the current time level. That is, the evaluation of the time derivative of the solution only depends on the value of the solution at the current time level, other wise known as an explicit time advancing scheme. Runge-Kutta explicit time advancing schemes are used. The main considerations in choosing a time integration scheme are accuracy and efficiency. It is desirable to ensure that the time integration scheme employed does not itself add any spurious oscillations to the solution. Stated more formally, this means that a Runge-Kutta (RK) scheme that will maintain the Total Variation Diminishing in the Mean (TVDM) or Total Variance Bounding in the Mean (TVBM) property of the solution, provided the appropriate limiting of spatial oscillations (see Section 2.1.6), and appropriate time step are chosen. A class of schemes called Strong Stability-Preserving Runge-Kutta Schemes (SSP-RK) have been derived for this purpose [23]. Typically, fourth-order RK schemes are used for time integration, but unfortunately it has been shown [23] that it is not possible to construct a fourth-order, four-stage SSP-RK scheme. It is possible to construct a fourth-order, five-stage SSP-RK scheme, and the extra work of additional function evaluations is partially offset in this scheme by the maximum allowed time-step being larger. For this research, another common choice, the optimal third-order three-stage SSP-RK scheme is used. This scheme is optimal in the sense that the time step restriction for stability is the same as that for the forward Euler scheme [23]. The scheme is given as,

$$u^{(1)} = u^n + \Delta t\, \mathcal{L}(u^n, t^n),$$

$$u^{(2)} = \frac{1}{4}\left(3u^n + u^{(1)} + \Delta t\, \mathcal{L}(u^{(1)}, t^n + \Delta t)\right),$$

$$u^{n+1} = \frac{1}{3}\left(u^n + 2u^{(2)} + 2\Delta t\, \mathcal{L}\left(u^{(2)}, t^n + \frac{1}{2}\Delta t\right)\right) \quad .$$

This scheme will provide a solution that is third-order accurate in time.

In order to ensure numerical stability of the time advancing scheme, it must be ensured that the eigenvalues of the discrete spatial operator, $\mathcal{L}(u_h)$, all lie within the region of stability of the time advancing scheme being used [21]. For the simple, forward Euler time advancing scheme, this can be expressed using the Courant-Friedrichs-Levy condition [21],

$$\Delta t = \frac{CFL}{\max(|\lambda|)} \min_K(\Delta x^k), \quad CFL \leq \frac{1}{2N+1} \quad . \tag{2.23}$$

where $\max(|\lambda|)$ represents the absolute value of the largest eigenvalue of the flux-Jacobian of the partial-differential equation, $(\frac{\partial \mathbf{f}}{\partial \mathbf{u}})$, in the *entire* simulation domain, $\min_K(\Delta x^k)$ represents the smallest interval size in the *entire* simulation domain, and N is the order of the modal representation used in this simulation. Thus, this sets a globally defined Δt, which will guarantee stability in the entire simulation, even if in some areas of the simulation this time step is much too conservative. The requirement set by Eq. (2.23) will satisfy the numerical stability requirement for the forward-Euler time advance, and thus, will also be numerically stable for the SSP-RK3 time advance implemented here. Note the dependence on the order of the modal representation of the solution. This means that, all else being equal, higher-order schemes will have more restrictive time steps.

2.1.8 Boundary Conditions

Information from boundary conditions are integrated into the solution in the same manner as information is transferred from neighboring cells: through the numerical flux. Two means of implementing boundary conditions are applied in this work: 'ghost

cell' based boundary conditions and flux based boundary conditions. In principle, one of the primary advantages of discontinuous Galerkin is the ability to develop high order solutions without extrapolation. This has proven to be somewhat difficult to realize in practice when it comes to the case of boundary conditions. This can be realized for a sub-set of boundary conditions when using ghost cells, and should be obtainable for all cases when using flux based boundary conditions. However, it is not trivial to determine how this should be accomplished.

The so called 'ghost cell' is a non-physical cell which exists on the far side of a boundary. The value of the solution in this cell is a function of the value of the solution in the boundary cell. This is the type of boundary condition that is used in finite volume simulations. In finite volume simulations, higher spacial accuracy is accomplished by extrapolating the solution from several neighboring boundary cells to develop a more accurate estimate of the appropriate value of the solution in the ghost cell. For certain cases in DG it is possible to obtain higher order solutions by simply using information from the boundary cell itself. Examples of such conditions are axisymmetric boundary conditions, periodic boundary conditions and wall boundary conditions.

The method used in a flux based boundary condition is to directly specify what values the numerical flux at the boundary should take. This offers the advantage of being able to develop higher-order accuracy boundary conditions without having to reconstruct an appropriate high-order solution in the ghost cell, which is often a non-trivial task.

2.2 Modification for Axisymmetric Case

In coordinate systems other than Cartesian, the differential operator ∇ needs to be modified. For cylindrical coordinate systems, which were used for the axisymmetric simulations performed in this work, it becomes [22]

$$\nabla = \frac{1}{r}\frac{\partial}{\partial r} r\,\hat{r} + \frac{1}{r}\frac{\partial}{\partial \phi}\,\hat{\phi} + \frac{\partial}{\partial z}\,\hat{z} \quad . \tag{2.24}$$

This modification must be applied to all equations in the equation system. Applying this to the induction equation for example results in,

$$\frac{\partial B_r}{\partial t} + \frac{1}{r}\frac{\partial E_z}{\partial \phi} - \frac{\partial E_\phi}{\partial z} = 0 \quad , \tag{2.25}$$

$$\frac{\partial B_\phi}{\partial t} + \frac{\partial E_r}{\partial z} - \frac{\partial E_z}{\partial r} = 0 \quad , \tag{2.26}$$

$$\frac{\partial B_z}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}\left(rE_\phi\right) - \frac{1}{r}\frac{\partial E_r}{\partial \phi} = \frac{\partial B_z}{\partial t} + \frac{\partial E_\phi}{\partial r} - \frac{1}{r}\frac{\partial E_r}{\partial \phi} + \frac{E_\phi}{r} = 0 \quad . \tag{2.27}$$

For the axisymmetric case,

$$\frac{\partial B_r}{\partial t} + -\frac{\partial E_\phi}{\partial z} = 0 \quad , \tag{2.28}$$

$$\frac{\partial B_\phi}{\partial t} + \frac{\partial E_r}{\partial z} - \frac{\partial E_z}{\partial r} = 0 \quad , \tag{2.29}$$

$$\frac{\partial B_z}{\partial t} + \frac{\partial E_\phi}{\partial r} + \frac{E_\phi}{r} = 0 \quad . \tag{2.30}$$

To implement the numerical solution to this equation system, this can be re-cast it into the general form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial r} + \frac{\partial \mathbf{G}}{\partial z} = \mathbf{S} \quad . \tag{2.31}$$

Substituting in the terms from the induction equation yields

$$\frac{\partial}{\partial t}\begin{bmatrix} B_r \\ B_\phi \\ B_z \end{bmatrix} + \frac{\partial}{\partial r}\begin{bmatrix} 0 \\ -E_z \\ E_\phi \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} -E_\phi \\ E_r \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -E_\phi/r \end{bmatrix} \quad . \tag{2.32}$$

The effect of the $\mathbf{v} \times \mathbf{B}$ in the induction equation is already accounted for, and the contribution of resistivity will simply be a direct addition. The contribution of resistivity to the flux for the induction equation is then

$$\frac{\partial}{\partial t}\begin{bmatrix} B_r \\ B_\phi \\ B_z \end{bmatrix} + \frac{\partial}{\partial r}\begin{bmatrix} 0 \\ -\eta\, J_z \\ \eta\, J_\phi \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} -\eta\, J_\phi \\ \eta\, J_r \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\eta\, J_\phi / r \end{bmatrix} \quad . \tag{2.33}$$

These flux and source terms are then appended to the existing flux terms.

2.3 Calculation of Curl

A special application of the discontinuous Galerkin method is to calculate the curl by simply choosing the proper arrangement of terms in the fluxes, and giving the result as the curl instead of a time derivative. This is implemented in this research to calculate current, and is written as

$$\begin{bmatrix} J_r \\ J_\phi \\ J_z \end{bmatrix} + \frac{\partial}{\partial r}\begin{bmatrix} 0 \\ B_z/\mu_0 \\ -B_\phi/\mu_0 \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} B_\phi/\mu_0 \\ -B_r/\mu_0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ B_\phi/(\mu_0 r) \end{bmatrix} \quad . \tag{2.34}$$

This is a time *independent* differential equation that can be solved using the same exact discontinuous Galerkin infrastructure that is used to evolve the solution of a system of PDEs.

Chapter 3

Model Development

Contents

Three broad approaches are typically used to model the behavior of a plasma. The first is to calculate the trajectory of individual charged particles under the influence

of electric and magnetic fields; either externally imposed or interpolated in some approximate manner from the positions and velocities of the other particles constituting the plasma. The second is to apply the tools of statistical mechanics to represent the plasma in terms of a distribution function over position, velocity, and time. The third is to posit the existence of a plasma fluid; a substance which obeys certain conservation laws and is described by continuously varying functions of space and time.

This research employs a model of plasma behavior which falls under the third approach, that of a single fluid, referred to as Magnetohydrodynamics (MHD). This choice is motivated by the macroscopic nature of the phenomena that we wish to investigate. Before proceeding further, it is worthwhile to more precisely define what is meant by macroscopic.

A fluid is a construct of equilibrium thermodynamics. Modeling the behavior of a collection of particles as a fluid introduces the assumption that everywhere within the fluid a state of local thermodynamic equilibrium exists; it is only then that macroscopic thermodynamic quantities, such as temperature and pressure have a well defined meaning. Thermal equilibrium is a result of frequent collisions within a population of particles, and results in a population of particles assuming a Maxwellian velocity distribution. The characteristic time scale over which a population of particles will revert to a condition of thermal equilibrium is referred to as the thermal relaxation time. The thermal relaxation times for electrons and ions are given by the kinetic theory of plasmas as [24]

$$\tau_e \equiv 6\pi\sqrt{2\pi}\epsilon_0^2 \frac{m_e^{1/2}\,(kT_e)^{3/2}}{\ln\Lambda\, Z^2 e^4 n_i} \quad , \tag{3.1}$$

$$\tau_i \equiv 6\pi\sqrt{2\pi}\epsilon_0^2 \frac{m_i^{1/2}\,(kT_i)^{3/2}}{\ln\Lambda\, Z^4 e^4 n_i} \quad . \tag{3.2}$$

These are the time scales at which a population of electrons or ions will relax to a thermodynamically well defined electron or ion *fluid*. For a plasma to be represented as a single fluid however, additional restrictions must be enforced. Firstly as an extension of the above logic, assuming that the plasma is described by a single fluid requires that the individual species of the plasma, the ions and the electrons, are in local thermal equilibrium with one another. This again ensures that this single

plasma fluid has a well defined temperature, pressure, and thermal energy. The time scale for temperature equilibration is [24]

$$\tau_{eq} = \frac{m_i}{2m_e}\tau_e \quad . \tag{3.3}$$

At timescales above this the electrons and ions will assume a common temperature, and there will be no transfer of thermal energy between them. For a single fluid model to be valid, it must further be assumed that

$$\lambda_{MHD} \gg R_i\,, \qquad \tau_{MHD} \gg \Omega_i^{-1}\,, \tag{3.4}$$

where R_i and Ω_i are the ion cyclotron radius and frequency respectively. If phenomena at length and time scales on the order of R_i and Ω_i must be considered, then the single fluid model is not sufficient, and a multi-fluid model becomes necessary.

3.1 Ideal MHD

If a plasma behavior of interest is sufficiently macroscopic, as defined above, but evolves quickly relative to dissipative effects, then dissipation is neglected, and the resulting model is referred to as Ideal MHD. In this model mass, momentum, energy and magnetic flux are conserved. The PDEs describing the time evolution of these conserved quantities are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0\,, \tag{3.5}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \left[\rho \mathbf{v}\mathbf{v} + P\mathbf{I} - \frac{\mathbf{B}\mathbf{B}}{\mu_0} + \frac{B^2}{2\mu_0}\mathbf{I}\right] = 0\,, \tag{3.6}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{B^2}{2\mu_0}\right)\mathbf{v} - \frac{(\mathbf{B}\cdot\mathbf{v})}{\mu_0}\mathbf{B}\right] = 0\,, \tag{3.7}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot [\mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v}] = 0\,, \tag{3.8}$$

$$\tag{3.9}$$

where the total energy density, E, is defined as

$$E = \frac{P}{\gamma - 1} + \frac{\rho v^2}{2} + \frac{B^2}{2\mu_0} \quad , \tag{3.10}$$

and the plasma is assumed to obey an ideal gas law equation of state.

Single-fluid MHD is the core model used in this research. Modifications to this basic set of equations, both for numerical and modeling purposes, are made, the details of which are outlined in the remainder of this chapter, however all of these modifications constitute minor deviations rather than fundamental alterations.

It is often very useful to consider the eigensystem of the system of PDEs used in a particular physical model. This not only gives insight into the mathematical character of the solutions that can be expected, (hyperbolic, parabolic, elliptic, mixed), for the hyperbolic case, it allows us to identify the types and speeds of waves in the model that transmit information throughout the domain. For example, in the case of the Euler equations, there are three distinct eigenvalues, $a - c_s$, a and $a + c_s$, where a is the advection speed of the fluid relative to our frame of reference, and $c_s = \sqrt{\gamma P/\rho}$ is the speed of sound in the fluid. These eigenvalues represent the three waves in the system which transmit information, and represent, respectively, a backwards traveling

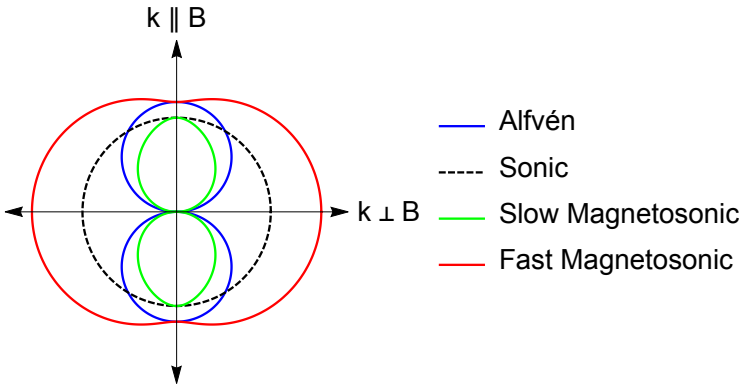


Figure 3.1: A plot of the phase velocity of the three types of waves that occur in Ideal MHD for various orientations of wave vector with B. Note that the sonic speed is not one of the waves in the Ideal MHD system and is included for illustration only.

sound wave, the advective wave, and a forwards traveling sound wave.

The situation for Ideal MHD is substantially more complex. Firstly, in ideal MHD, there are, in general, three distinct types of waves: alfvén waves, fast magneto-sonic waves, and slow magneto-sonic waves. And secondly, these waves propagate in a highly anisotropic manner, as illustrated in Fig. (3.1).

The characteristic speeds are defined as

$$c_o = \sqrt{\frac{\gamma P}{\rho}}\,, \tag{3.11}$$

$$c_a = \frac{B}{\sqrt{\mu_0 \rho}}\,, \tag{3.12}$$

$$c_f = \sqrt{\frac{1}{2}\left(c_o^2 + c_a^2 + \sqrt{\left(c_o^2 + c_a^2\right)^2 - 4c_o^2 c_a^2 \cos^2\theta}\right)} \tag{3.13}$$

$$c_s = \sqrt{\frac{1}{2}\left(c_o^2 + c_a^2 - \sqrt{\left(c_o^2 + c_a^2\right)^2 - 4c_o^2 c_a^2 \cos^2\theta}\right)}\,. \tag{3.14}$$

Where θ is the angle between the direction of the wave propagation and the magnetic field. There are then seven characteristic speeds in the Ideal MHD system, given by

$$a - c_f < a - c_a < a - c_s < a < a + c_s < a + c_a < a + c_f \,. \tag{3.15}$$

This increased number of eigenvalues and the anisotropy, and for several orientations, degeneracy of eigenvalues, leads to a much more complicated mathematical behavior of the ideal MHD equation system compared to the Eulerian equation system for example.

Two dimensionless parameters that are important for characterizing the behavior of plasmas described by ideal MHD will now be introduced. Where as the characteristic length and time scales mentioned earlier signal where a given physical model is valid, the non-dimensional parameters give insight into the physics occurring in a given model. The two parameters that we wish to introduce are referred to as thermal plasma beta and kinetic plasma beta, β_{th}, and β_k respectively. They are defined as:

$$\beta_{th} = \frac{2\mu_0 P}{B^2} \tag{3.16}$$

$$\beta_k = \frac{\mu_0 \rho v^2}{B^2} \tag{3.17}$$

Thermal Plasma Beta represents the ratio of plasma pressure to magnetic field pressure. Kinetic Plasma Beta represents the ratio of dynamic pressure to magnetic field pressure. The beta parameters are measures of the degree to which the plasma is confined by the magnetic field. When the values of these parameters are smaller than 1, the plasma will only be able to make small perturbations in the magnetic field. When their values are larger than 1, the magnetic field will be forced to follow the motion of the plasma, and will only produce small perturbations in the motion of the plasma compared to the motion of the plasma in the absence of the magnetic field.

3.2 Resistive MHD

In the above discussion of the assumptions inherent in a single fluid approximation of a plasma, it is considered that the electrons and ions are in local thermal equilibrium.

This does not preclude however the development of a mean relative velocity between the electrons and ions. Such a relative velocity will constitute a current density, given by

$$\mathbf{j} = ne\left(\overline{\mathbf{v}}_e - \overline{\mathbf{v}}_i\right) \quad . \tag{3.18}$$

With a single fluid model, there is not direct access to the average velocities of the ions and electrons. Instead, the problem of calculating current may be approached from the electromagnetic direction. The Maxwell-Ampère equation

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \tag{3.19}$$

allows calculation of the current from the electric and magnetic fields. The timescale over which displacement currents develop is much shorter than the hydrodynamic timescale, and therefore the displacement current term can be neglected, recovering the original formulation of Ampère's law

$$\mathbf{j} = \frac{1}{\mu_0} \nabla \times \mathbf{B} \quad . \tag{3.20}$$

This permits the calculation of current purely in terms of the magnetic field, evolved in the MHD model. Note that this means that currents are still present in ideal MHD.

An important dimensionless parameter that indicates the importance of resistivity in a simulation is the Magnetic Reynolds Number, Re_m. The Magnetic Reynolds Number is a dimensionless parameter which quantifies the ratio of the advective and diffusive transport terms in the induction equation. It is defined as

$$Re_m = \frac{\mu_0 l_0 v_0}{\eta} \tag{3.21}$$

Where l_0 is the characteristic length of magnetic field gradients perpendicular to the magnetic field lines, and v_0 is the characteristic advection speed in the simulation of interest. The Magnetic Reynolds Number is a measure of how closely the magnetic filed and the plasma flow is coupled. For $Re_m \gg 1$, the magnetic field is strongly coupled to the plasma, and motion in the plasma cannot occur without motion of the

field lines, and visa versa. For $Re_m \ll 1$ the magnetic field and the plasma become essentially decoupled, and evolve independently.

The generalized Ohm's Law for moving conducting media in the presence of a magnetic field is[24]

$$\mathbf{E} = -\mathbf{v} \times \mathbf{B} + \eta \mathbf{j} \quad . \tag{3.22}$$

This generalized Ohm's Law is then substituted into the induction equation to obtain the evolution equations for the magnetic field of the plasma. If the timescale of interest is much larger than the characteristic timescale of resistivity, either because of sufficiently large time derivatives, or sufficiently low resistivity, then the contribution of $\eta \mathbf{j}$ to the electric field, and thus to the induction equation, can be neglected. Current is still present in ideal MHD, but it does not influence the evolution of the plasma. If the diffusion timescale cannot be neglected, then resistivity and current must be taken into account. How this is done is discussed in the next section.

3.2.1 Modification of the Ideal MHD Equation System

The density and momentum equations remain un-altered with the introduction of resistivity. It is fairly intuitive that resistivity would not affect the total number of charged particles and thus would not affect the density equation, unlike for example, ionization or recombination. Since resistivity only involves collisions between particles that constitute the plasma, it will only involve transfers of momentum from one population of particles in the plasma to another, meaning that there will be no momentum gained or lost, as would be the case for collision with neutral atoms. However, resistivity does alter the equations governing energy and magnetic fields. The physical mechanisms that correspond to these alterations are the conversion of magnetic energy to thermal energy through ohmic heating and magnetic diffusion. The details of these alterations are discussed below.

The modified form of the PDE governing total energy in the plasma is [24]

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(\frac{1}{2}\rho v^2 + \frac{P}{\gamma - 1} + P \right) \mathbf{v} + \frac{1}{\mu_0} \left(-\mathbf{v} \times \mathbf{B} + \eta \mathbf{j} \right) \times \mathbf{B} \right] = 0 \quad . \tag{3.23}$$

This equation states that, with the addition of resistivity, there is no loss of conservation in total energy. Rather, there is now an additional term involving $\eta\,\mathbf{j}$, which, recalling Ampère's law, is equivalent to $\eta/\mu_0 \nabla \times \mathbf{B}$, meaning that we have an altered, but still conservative equation.

It is desirable to recast this equation in a form that will highlight the similarities and differences between the resistive and ideal MHD formulations. First, we add and subtract a factor of $(B^2/\mu_0)\mathbf{v}$ to the terms inside the divergence term

$$\frac{\partial E}{\partial t} + \nabla \cdot \Big[\left(\frac{1}{2}\rho v^2 + \frac{P}{\gamma - 1} + \frac{B^2}{2\mu_0} + P + \frac{B^2}{2\mu_0} \right) \mathbf{v} - \left(\frac{B^2}{\mu_0} \right) \mathbf{v} + \frac{1}{\mu_0} \left(-\mathbf{v} \times \mathbf{B} + \eta\,\mathbf{j} \right) \times \mathbf{B} \Big] = 0 \quad . \tag{3.24}$$

The first term inside of the divergence term is simply equal to the total plasma energy density, E, and can thus be simplified. Further simplifying by distributing the cross product in the second term inside of the divergence term results in

$$\frac{\partial E}{\partial t} + \nabla \cdot \Big[\left(E + P + \frac{B^2}{2\mu_0} \right) \mathbf{v} - \left(\frac{B^2}{\mu_0} \right) \mathbf{v} + \frac{1}{\mu_0} \left(-\mathbf{v} \times \mathbf{B} \right) \times \mathbf{B} + \frac{\eta}{\mu_0} \left(\mathbf{j} \times \mathbf{B} \right) \Big] = 0 \quad . \tag{3.25}$$

Using the vector identity, $(\mathbf{C} \times \mathbf{B}) \times \mathbf{A} = (\mathbf{A} \cdot \mathbf{C})\,\mathbf{B} - (\mathbf{A} \cdot \mathbf{B})\,\mathbf{C}$, the first cross product inside of the divergence term resulting in

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{B^2}{2\mu_0} \right) \mathbf{v} - \left(\frac{\mathbf{B} \cdot \mathbf{v}}{\mu_0} \right) \mathbf{B} + \frac{\eta}{\mu_0} \left(\mathbf{j} \times \mathbf{B} \right) \right] = 0 \quad . \tag{3.26}$$

It is now clear that the energy equation in the resistive case is the same as that of ideal-MHD except for the addition of a single new term, and that accordingly, the dynamics of energy transfer in the plasma are affected by the introduction of resistivity, but the conservation of energy is not. Setting $\eta = 0$ recovers the ideal MHD form of the equation.

The time rate of change of magnetic fields can be obtained from the induction equation,

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0 \quad . \tag{3.27}$$

Substituting Eq. (3.22) into the induction equation results in

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{vB} - \mathbf{Bv}) = -\nabla \times (\eta \mathbf{j}) = -\frac{\eta}{\mu_0} \nabla \times (\nabla \times \mathbf{B}) \quad . \tag{3.28}$$

From the presence of a source term on the right hand side of the equation, it is clear that the function for the magnetic field is no longer conservative. Finally, the entire equation system for resistive MHD is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0 \tag{3.29}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \left[\rho \mathbf{vv} + P\mathbf{I} - \frac{\mathbf{BB}}{\mu_0} + \frac{B^2}{2\mu_0} \mathbf{I} \right] = 0 \tag{3.30}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{B^2}{2\mu_0} \right) \mathbf{v} - \left(\frac{\mathbf{B} \cdot \mathbf{v}}{\mu_0} \right) \mathbf{B} + \frac{\eta}{\mu_0} (\mathbf{j} \times \mathbf{B}) \right] = 0 \tag{3.31}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot [\mathbf{vB} - \mathbf{Bv}] + \nabla \times (\eta \mathbf{j}) = 0 \quad . \tag{3.32}$$

Where the total energy density, E, is defined as

$$E = \frac{P}{\gamma - 1} + \frac{\rho v^2}{2} + \frac{B^2}{2\mu_0} \quad , \tag{3.33}$$

and the plasma is assumed to obey an ideal gas law equation of state.

A note about the Resistive MHD model. Macroscopic dissipative phenomena in plasmas, including diffusion, thermal conductivity, viscosity, and resistivity, can be given a rigorous theoretical foundation using the kinetic theory of plasmas [25]. A full,

rigorous treatment of these phenomena is extremely complex. From the perspective of kinetic theory, the resistive MHD model is, "the crudest approximation [of the transport processes in a plasma] which still retains some form of dissipation" [25].

Resistive MHD is however a useful and very tractable first approximation of dissipative processes in a plasma at the macroscopic scale. However, caution should be exercised when analyzing the results of Resistive MHD simulation. One reason for this, and one of the most significant simplifications is that Resistive MHD assumes that the resistivity of a magnetized plasma is isotropic. This is known to be false, and resistivity perpendicular to magnetic field lines is approximately twice the resistivity along magnetic field lines [25].

The Resistive MHD theory should therefore be considered as providing a first approximation of dissipative effects in a plasma which is very large and evolves very slowly relative to fundamental, microscopic plasma phenomena.

3.2.2 Calculation of Resistivity

Having derived the appropriate system of equations that should be used for modeling resistive MHD, an important question remains in how to calculate an appropriate value for resistivity, which is not a fundamental constant, and highly dependent on the variables of state of the plasma. A reasonably straightforward argument can be laid out for how to arrive at a first approximation of resistivity.

Consider a large number of electrons with number density n_e, in a plasma in which a constant, uniform electric field E has been applied. Assume that the relative drift velocity v_d between electrons and ions is much less than the electron thermal velocity, v_{te}. The electron population will tend to be accelerated by the electric field, and decelerated by collisions with ions, which occur with at an average frequency of $\overline{\nu}_{ei}$. Take the entire population of electrons as the system of interest. This means that only momentum transfers between the electron and ion populations will result in a change in momentum of the system, and electron-electron collisions can be ignored. Assume that the ions are stationary relative to the laboratory frame of reference. The relative drift velocity, v_d is then equal to the bulk velocity of the electrons, v_e. Newton's Second Law for the electron population is then

$$\frac{d}{dt}\left(m_e n_e v_e\right) = -\,e n_e E \;-\; \overline{\nu}_{ei}\, m_e n_e v_e \quad . \tag{3.34}$$

Where m_e is the electron mass and e is the fundamental charge. Once the electrons have achieved a steady state,

$$v_e = \frac{-eE}{\overline{\nu}_{ei}\, m_e} \tag{3.35}$$

Since ions have a much greater mass than electrons, it is reasonable to make the assumption that this steady state will be reached long before significant acceleration of the ions occurs. It is then reasonable to assume that the current density is due entirely to the motion of the electrons, resulting in

$$j = -e\, n_e\, v_e = \frac{n_e e^2 E}{m_e\, \overline{\nu}_{ei}} \quad . \tag{3.36}$$

It is interesting to note that this can be formulated in terms of one of the fundamental plasma parameters, the electron plasma frequency

$$j = \frac{n_e\, e^2}{\epsilon_0\, m_e} \frac{\epsilon_0\, E}{\overline{\nu}_{ei}} = \omega_{pe}^2 \frac{\epsilon_0\, E}{\overline{\nu}_{ei}} \quad . \tag{3.37}$$

Given the assumption stated at the beginning of this section, that $v_d \ll v_{te}$, the mean electron-ion collision frequency is [25],

$$\overline{\nu}_{ei} = \frac{8\pi}{3\sqrt{2\pi}} \left(\frac{Ze^2}{4\pi\epsilon_0}\right)^2 \frac{n_i \ln\Lambda}{\sqrt{m_e}\,(kT)^{3/2}} \quad . \tag{3.38}$$

Where $\ln\Lambda$ is the *Coulomb Logarithm*, a dimensionless parameter that is a slowly varying function of electron temperature and density, and is typically taken to be $\approx 10 - 20$. Resistivity is related to the current density j and electric field E by

$$j = \frac{1}{\eta} E \quad . \tag{3.39}$$

Substituting in the expression for the electric field and current in terms of fundamental parameters, the resistivity can be expressed as

$$\eta = \frac{\overline{\nu}_{ei}}{\epsilon_0 \, \omega_{pe}^2} \quad . \tag{3.40}$$

Substituting in 3.38, and re-expressing ω_{pe} in terms of fundamental constants, the resistivity is given as

$$\eta = \frac{Z \, e^2 \, m_e^{1/2} \, 8\pi \, \ln \Lambda}{(4\pi \, \epsilon_0)^2 \, 3\sqrt{2\pi} \, (k_b T_e)^{3/2}} \quad . \tag{3.41}$$

Note that variation in the resistivity is dominated by temperature. While $\ln \Lambda$, is a function of $T_e and n_e$, is a very slowly varying quantity, so $\eta \propto T_e^{-3/2}$.

Given the strong dependence of resistivity on temperature, some care should be taken in outlining how temperature should be obtained from the conserved quantities used in the resistive MHD equations. The fluid models described and used in this work assume an ideal gas equation of state. This implies that the total thermal pressure in a plasma (considered using a fluid model) is the sum of the partial pressures due to each component species. The assumptions of temperature equilibration and quasi-neutrality allow us to express this as

$$P = (k_b n_e T_e) + (k_b n_i T_i) = 2 k_b n T \quad . \tag{3.42}$$

Using the definition of total energy density, given by Eq. 3.33, the plasma pressure can be expressed in terms of the other conserved variables as

$$P = (\gamma - 1) \left[E - \frac{\rho v^2}{2} - \frac{B^2}{2\mu_0} \right] \quad . \tag{3.43}$$

This expression for pressure can then be substituted into 3.42 to obtain

$$T = \frac{1}{2 k_b n} (\gamma - 1) \left[E - \frac{\rho v^2}{2} - \frac{B^2}{2\mu_0} \right] \quad . \tag{3.44}$$

Number density does not appear explicitly in the governing equations for resistive MHD, but mass density, ρ, is. Multiplying the expression for T on top and bottom by m_i results in

$$T = \frac{m_i}{2k_b\rho_i}(\gamma - 1)\left[E - \frac{\rho v^2}{2} - \frac{B^2}{2\mu_0}\right] \quad , \tag{3.45}$$

Which provides the temperature of the plasma in terms of fundamental constants and variables that can be obtained from the governing equations of Resistive MHD.

3.2.3 Time Step Restriction

A brief detour into numerical considerations is worthwhile at this time. As has been shown above, the introduction of resistivity into the equation system introduces second order spatial derivative terms of the magnetic field components. This changes the mathematical character of the PDEs, and considerations must be made for whether a new restriction must be placed on the time step for the explicit time advance. A heuristic argument for whether resistivity should impose a restriction of the time step can be made by considering the relative magnitude of $\frac{\partial \mathbf{B}}{\partial t}$ due to the advective and resistive terms in the induction equation. Choose a characteristic magnetic field strength, B_0, a characteristic length L_0, and a characteristic time t_0. A non-dimensional set of variables, $\widehat{B} = B/B_0$, $\widehat{L} = L/L_0$, $\widehat{t} = t/t_0$ can then be formed. The contribution of advection and diffusion to the induction equation can then be expressed in terms of non-dimensional variables as,

$$\left.\frac{\widehat{B}}{\widehat{t}}\right|_{adv} = \frac{\lambda_{max}\widehat{B}}{\widehat{L}} \tag{3.46}$$

$$, \left.\frac{\widehat{B}}{\widehat{t}}\right|_{diff} = \frac{\eta\widehat{B}}{\mu_0\,\widehat{L}^2} \quad . \tag{3.47}$$

Where λ_{max} is the maximum wave speed in the system. The characteristic timescales are then, respectively

$$\widehat{t}_a = \frac{\widehat{L}}{\lambda_{max}} \tag{3.48}$$

$$\widehat{t}_d = \frac{\mu_0 \widehat{L}^2}{\eta} \quad . \tag{3.49}$$

Therefore, to capture the physics of both the advection and the diffusion of the magnetic field, the time step dt, chosen must satisfy

$$dt \leq \min\left(\frac{dx}{\lambda_{max}}, \frac{\mu_0\, dx^2}{\eta}\right) \quad . \tag{3.50}$$

3.2.4 Correct Calculation of Characteristic Gradient Length

For accurately calculating the magnetic reynolds number, It is important to determine the correct method for calculating the characteristic scale length of gradients perpendicular to the magnetic field lines. This is defined as

$$L_0 = \frac{|\mathbf{B}|}{\nabla_\perp |\mathbf{B}|}\,, \tag{3.51}$$

where $\nabla_\perp$ is the derivative perpendicular to local magnetic field line. The normal direction to the magnetic field is

$$\hat{\mathbf{b}}_\perp = \frac{\mathbf{B}}{|\mathbf{B}|} \times \hat{\theta}\,, \tag{3.52}$$

where $\hat{\theta}$ is the unit normal in the azimuthal direction, which is perpendicular to all points in the magnetic field for these axisymmetric simulations. The derivative perpendicular to the field line is then given by

$$\nabla_\perp |\mathbf{B}| = \left|\nabla\, |\mathbf{B}| \cdot \hat{\mathbf{b}_\perp}\right| \tag{3.53}$$

3.3 Divergence Error Correction

All numerical schemes for the solution of PDEs involve, in one form or another, generating a discrete approximation to a continuous exact solution. In general, the discretized solution is not equal to the exact solution, and the difference is referred to as the discretization error. Due to discretization error, constraints which physical theory insists must be obeyed exactly, and which are obeyed exactly by the PDEs, may be violated by approximate solutions. An important example in the field of plasma simulation is the development of magnetic fields for which $\nabla \cdot \mathbf{B} \neq 0$. Magnetic monopoles are non-physical, and their presence in simulations leads to non-physical behavior, such as acceleration of plasma parallel to magnetic field lines [26].

It is possible to develop numerical schemes that exactly satisfy the divergence constraint on the magnetic field, but it can be shown that such schemes are no longer conservative [27]. This is a significant drawback if a scheme capable of shock capturing is desired, in which case Rankine-Hugoniot conservation relations across shocks must be strictly satisfied [26].

Rather than requiring that $\nabla \cdot \mathbf{B} = 0$ be satisfied exactly, it is generally considered acceptable if it, and the non-physical behavior associated with it, tend to vanish with increasing grid resolution, just as is the case for the other variables in the simulation [27]. Divergence error correction schemes are thus focused on minimizing the negative effects of divergence error, rather than eliminating it outright.

3.3.1 Hyperbolic Divergence Cleaning

For this research, the method of divergence error correction developed by Dedner et al [28] is implemented. This method is chosen because it is very easy to implement in existing codes. This technique only requires that the equation system being solved be modified slightly, and can be used with any numerical technique. Additionally, this scheme maintains conservation of density, momentum, magnetic induction and total energy.

The basic premise of this technique is to couple the conservation equations of ideal

MHD with the divergence constraint on magnetic fields through the use of a generalized Lagrange Multiplier (GLM). This mathematical technique introduces a non-physical variable ψ, named the *error correction potential* (ECP), which is introduced into the induction equation,

$$\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \cdot (\boldsymbol{uB} - \boldsymbol{Bu}) + \nabla\psi = 0\,, \tag{3.54}$$

and introducing a new equation,

$$\mathcal{D}(\psi) + \nabla \cdot \mathbf{B} = 0\,, \tag{3.55}$$

where $\mathcal{D}$ is a linear differential operator, i.e. some linear combination of partial derivatives of arbitrary order (including zero). The authors of the paper describing the method found that the best results were obtained when a mixed hyperbolic-parabolic differential operator is used, corresponding to both advection and damping of the divergence error in the simulation. This choice results in Eq. (3.55) taking the form,

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \boldsymbol{B} = -\frac{c_h^2}{c_p^2}\psi\,, \tag{3.56}$$

where c_h and c_p are constants corresponding to the rates of advection and damping of divergence error respectively. No other modifications of the ideal MHD or resistive MHD equation systems are required.

The resulting set of characteristic speeds are the same as for ideal MHD but with the addition of two new speeds, $-c_h$, and $+c_h$. It should be noted that this results in an equation system that is no-longer purely Galilean. Dedner et al found that it is possible to derive an alternate formulation where the new characteristic speeds are $a - c_h$, and $a + c_h$, where a is the advection speed, but at the cost of losing the conservation properties of the scheme. The formulation presented here should be adequate, but caution is perhaps advisable if a situation arises where $c_h \approx a$, as this may compromise the ability of the scheme to handle divergence errors.

The presence of a diffusion term in Eq. (3.56) is handled by the use of an operator splitting approach. At time level n, the homogeneous system of differential equations

is solved, (dropping the damping term on the right hand side of 3.56) using an arbitrary numerical technique. The value of ψ that results from the solution of the homogeneous equation system is taken to be an intermediate value, ψ^{n*}. The value of ψ is then modified,

$$\psi^{n+1} = e^{-\Delta t_n c_h^2/c_p^2}\psi^{n*}\,, \tag{3.57}$$

where Δt_n is the size of the time step used in advancing the solution from time level n. Note that this consists of simply multiplying ψ by a constant. This operator splitting approach is shown by the authors to result in an unconditionally stable scheme as long as the CFL condition is obeyed.

It remains to determine how appropriate values for c_h and c_p are to be arrived at. Since divergence should in principle be zero everywhere in the domain, it is reasonable to choose $c_h > \lambda_{max}$, where λ_{max} is the largest characteristic speed in the system, so that any divergence error that accumulates in a simulation will be advected at a faster rate than any of the physically meaningful conserved variables. For the results presented here, c_h is set by the input parameters of the simulation, and is chosen to be some multiple (> 1) of the Alfvén velocity.

The parameter controlling diffusion, c_p, is set by choosing a fixed ratio of diffusive to advective transport, c_r, defined by,

$$c_r = \frac{c_p^2}{c_h}\,. \tag{3.58}$$

Dedner et al reported that for the test cases they examined, $c_r = 0.18$ produced optimal results, and therefore this is the value for c_r used in this research.

3.3.2 Boundary Conditions

It remains to determine an acceptable set of boundary conditions for ψ. The authors who developed this method saw relative insensitivity to boundary conditions in the model problems they investigated [28]. Based on this, preliminary investigations with the divergence error correction scheme were performed using a Dirichlet type boundary condition, $\psi = 0$, on all boundaries. When this is used, it is found that

the scheme does not behave as expected. Rather, it is observed that using the error correction scheme resulted in only a slight decrease in the maximum divergence error in the system, and an *increase* in the average divergence error compared to no correction. A thorough investigation of these issues is performed. The investigations made are briefly summarized below.

For these investigations, the problem of interest in this research, that of a high temperature, high pressure plasma being injected into a diffuse background in a diverging magnetic field is used. For all simulations ψ is initialized to be zero everywhere in the domain. The spatial distribution of $\nabla \cdot \mathbf{B}$, as well as the L_1 an L_∞ norms of divergence error were considered. To allow a fair comparison between simulations, the divergence error measured is normalized by a characteristic magnetic field gradient defined by B_0/L_0, where B_0 is the characteristic magnetic field strength, and L_0 is the characteristic length. A detailed description of how divergence error is calculated in this research is provided in A.3.1.

During the course of these investigations, it is determined that using $\nabla \cdot \psi = 0$ as the boundary condition on the axisymmetric boundary improved the performance of the scheme, but did not completely resolve the issue. In these simulations a Dirichlet boundary condition for ψ is imposed on the boundary in which the plasma flow is being injected. This is believed to be a reasonable choice given that: 1) the inflow boundary condition is Dirichlet for all other variables in the simulation, and 2) that the magnetic field specified by the inflow boundary condition is known to satisfy $\nabla \cdot \mathbf{B} = 0$. After changing the boundary condition applied to ψ at the axisymmetric boundary, it is next hypothesized that the increase in divergence error is due to the particular values chosen for c_h and c_p. However, after a thorough investigation of the behavior of divergence error over a wide range of values, the performance of the correction scheme did not improve. However, when the inflow boundary condition for ψ is changed from Dirichlet to Neumann, the performance of the correction scheme improves dramatically.

Because this strong sensitivity to boundary conditions for ψ is in marked contrast to the findings of Dedner et al, it is decided that a more thorough investigation of this behavior needed to be performed.

An explanation of this unexpected behavior became apparent when spatial distributions of the divergence error were plotted. Comparative simulations were carried out

in which all simulation conditions were kept the same except for two: whether or not divergence correction is performed, and whether the inflow boundary condition is $\psi = 0$ or $\nabla \cdot \psi = 0$. Divergence correction is switched off by setting $c_h = 0$, so that comparisons could be made using the same code.

Figure 3.2 compares the distribution of $\nabla \cdot \mathbf{B}$ in the simulation domain for the case of no error correction (left) and error correction with a $\psi = 0$ boundary condition at the inflow (right). For both cases, a large divergence error has developed along the entire inflow boundary edge. In the simulation without correction, this divergence error remains confined to the boundary. In contrast, in the simulation where the correction scheme is applied, a large amount of divergence error has propagated into the simulation domain as the correction scheme tries to advect away the large divergence error being generated at the inflow boundary. Upon closer inspection of the divergence error at the inlet itself, it can be seen that applying divergence error correction has partially succeeded in correcting for the large divergence error being formed at the inflow boundary. Figure 3.3a shows that use of the error correction scheme results in a 50% reduction in the large error forming at the inflow.

This explains how application of a divergence correction scheme could decrease the maximum divergence error while increasing the average divergence error. The correction scheme is behaving as intended, and is attempting to dissipate the large divergence error being generated at the inflow boundary. It partially succeeds, reducing the maximum error which occurs at the inlet, but in the process introduces a large amount of more diffuse divergence error into the simulation domain.

The large divergence error at the inflow boundary arises from the application of a Dirichlet boundary condition for the magnetic field at the inlet. As the magnetic field tries to relax into a new configuration due to the injection of plasma into the simulation domain, the magnetic field neighboring the inflow boundary can fluctuate relative to the fixed magnetic field imposed by the boundary condition. This creates a jump discontinuity in the component of $\mathbf{B}$ normal to the boundary resulting in $\nabla \cdot \mathbf{B} \neq 0$. This can be observed by plotting a comparison of the change in B_z at the inlet boundary relative to the value of B_z specified by the boundary condition. Figure 3.3b compares the B_z values that occur when there is no correction and when there is a correction with Dirichlet at the inflow applied.

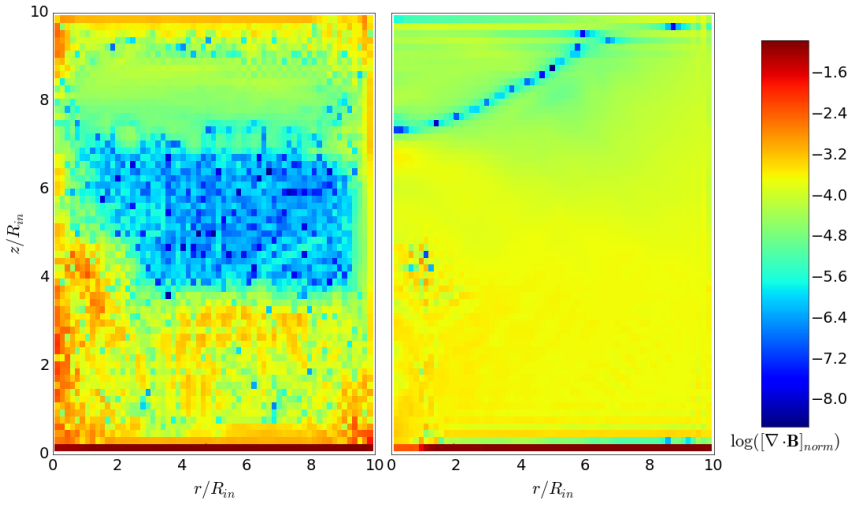
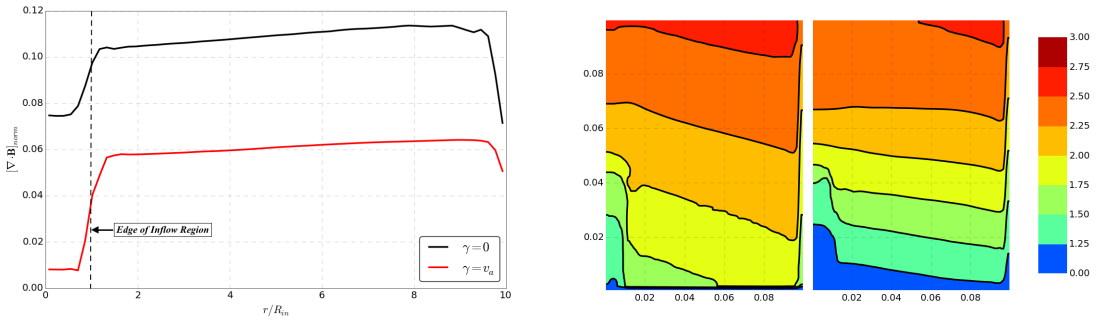


Figure 3.2: Log of divergence error in whole domain shown for $c_h = 0$ (left) and $c_h = v_a$ (right)



(a) (b)

Figure 3.3: Figure 3.3a shows divergence error along inflow boundary (z=0) for the divergence error plots shown in Figure (3.2). Figure 3.3b shows contours of the change in B_Z. $c_h = 0$ is shown on the left, $c_h = 1v_a$ is shown on the right. Change in percent of B_0.

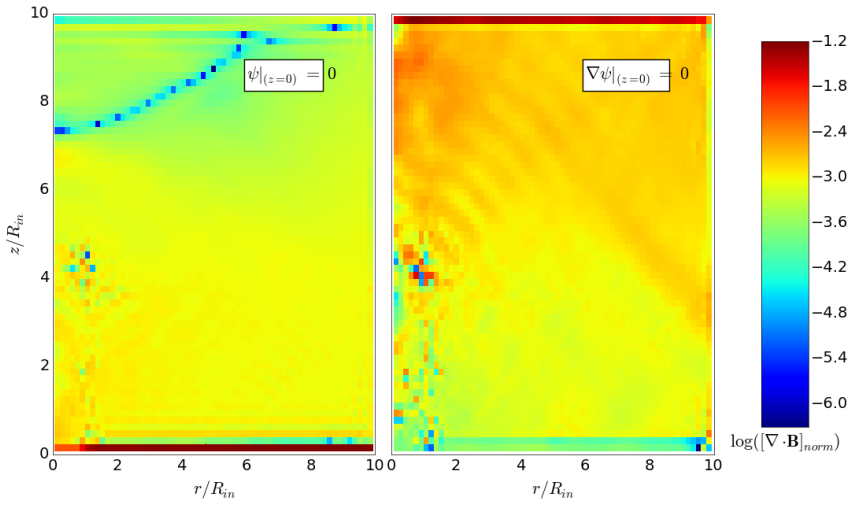


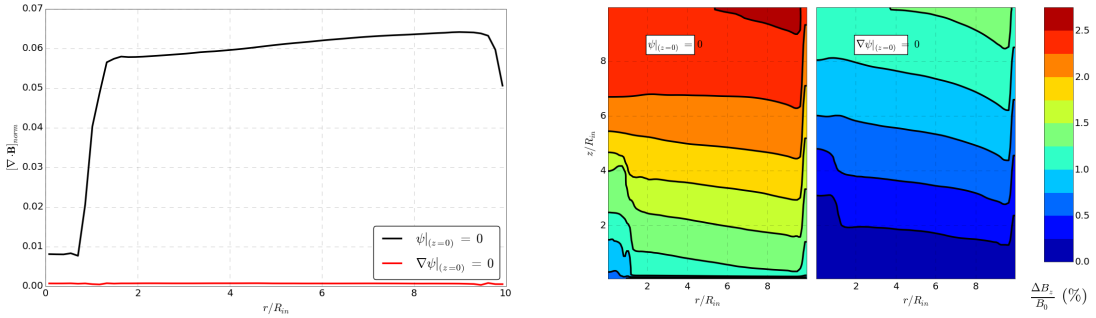
Figure 3.4: Log of divergence error in whole domain shown for $\psi|_{(z=0)} = 0$ (left) and $\nabla\psi_{(z=0)} = 0$ (right) boundary conditions.

From this investigation it is clear that a Neumann boundary condition for ψ performs best for simulations relevant to magnetic nozzle studies. Figure 3.4 shows a comparison of the spatial distribution of divergence error for the case of Dirichlet and Neumann boundary conditions. The main feature to note here is that the large divergence error at the inflow boundary is no longer present in the Neumann case. There are still regions with divergence error in the Neumann case, but the large divergence error seen along the inflow is dramatically reduced, as shown in Figure 3.5a. This agrees with what is observed when comparing contours of B_z, as shown in Figure 3.5b.

A more rigorous and quantitative comparison is to compare the evolution of divergence errors over time in the L_1 and L_∞ norms. This is shown in Figure 3.6 for the cases of no correction, correction with Dirichlet, and correction with Neumann. This shows quite convincingly that the Neumann boundary is the preferred choice, resulting in the maximum divergence error being reduced by two orders of magnitude.

3.4 Static-Dynamic Formulation

This research is interested in investigating plasmas with $\beta_{th}, \beta_k < 1$. In such situations, it is expected that the fluctuations in the magnetic field will be small relative to



(a) (b)

Figure 3.5: Figure 3.5a shows a comparison of divergence error along the z=0 boundary. Figure 3.5b shows contours of the change in B_Z. $\psi|_{(z=0)} = 0$ (left) and $\nabla\psi_{(z=0)} = 0$ (right) boundary conditions. Change is given in percent of B_0

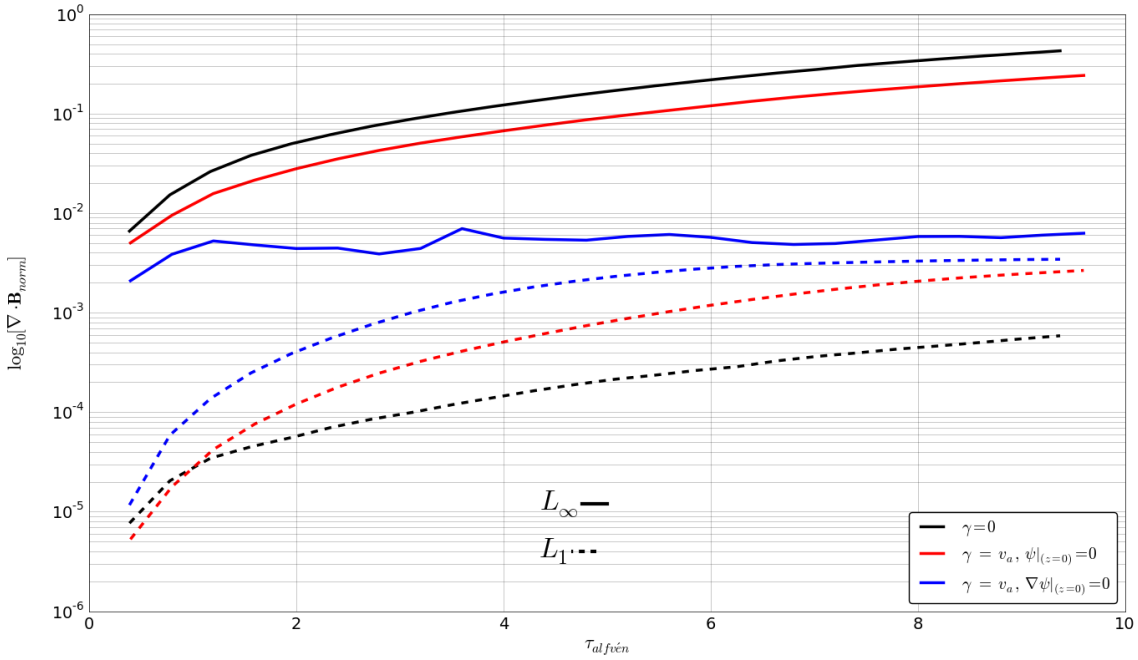


Figure 3.6: Comparison of Divergence error in the L_1 and L_∞ norms over time.

the initial condition of the magnetic field. In some simulations, fluctuations in the magnetic field may be 2 to 3 orders of magnitude smaller than the primary magnetic field. However, throughout the course of this research, it is found that very low β simulations develop negative pressure errors.

In considering how to address this positivity issue it is hypothesized that discretization error is proportional to magnitude in a given variable in a given scheme. For example, this hypothesis would hold that the discretization error in ρ will have a larger absolute value on average if ρ has a larger value on average. Where this ties into the development of negative pressures in simulations is the way in which pressure is defined in plasma simulations. The formulation of the MHD equations means that thermal, kinetic and magnetic energy are not considered separately, but rather the total energy of the plasma is evolved, and the pressure is derived from it. When $\beta_{th} \ll 1$, the ratio of thermal energy to magnetic energy in a plasma is much less than 1, regardless of the choice of variables or normalization. The above hypothesis about the nature of discretization error would then predict that the discretization error in the magnetic field energy would be large compared to the discretization error in other components of the solution, potentially leading to negative pressure. Specifically, the magnetic field components, with their proportionally larger error are evolved independent of the energy term, and then the energy they represent is subtracted, along with the kinetic energy, leaving the thermal energy. If there is sufficient error in the magnetic field energy, this process will result in a negative thermal energy and therefore negative pressure, crashing the simulation.

During the course of this research, a modeling technique is developed that proposes to ameliorate this issue by splitting the magnetic field into two variables: a static or applied magnetic field component, and a dynamic magnetic field component.

3.4.1 Dynamic-Static Magnetic Field Formulation

A formulation of the Ideal MHD equations referred to as Dynamic-Static Magnetic Field (DSB), is described here. This scheme separates the magnetic field into two components: the large, externally applied magnetic field is held static, and represented by $\mathbf{B}_0$. The deviation of the magnetic field from this initial condition is small, allowed to vary, and is represented by $\mathbf{B}_1$. Note that this is not a linearization or any other kind of simplifying approximation. In terms of theoretical behavior, adopting this

formulation for the magnetic field should result in a scheme that is indistinguishable from the standard ideal MHD scheme. Applying this scheme to the ideal MHD equation system results in,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0\,, \quad (3.59)$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \left[\rho \mathbf{v}\mathbf{v} + P\mathbf{I} - \frac{(\mathbf{B}_0 + \mathbf{B}_1)(\mathbf{B}_0 + \mathbf{B}_1)}{\mu_0} + \frac{(B_0 + B_1)^2}{2\mu_0}\mathbf{I} \right] = 0\,, \quad (3.60)$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + P + \frac{(B_0 + B_1)^2}{2\mu_0} \right) \mathbf{v} - \frac{((\mathbf{B}_0 + \mathbf{B}_1) \cdot \mathbf{v})}{\mu_0} (\mathbf{B}_0 + \mathbf{B}_1) \right] = 0\,, \quad (3.61)$$

$$\frac{\partial \mathbf{B}_1}{\partial t} + \nabla \cdot [\mathbf{v}(\mathbf{B}_0 + \mathbf{B}_1) - (\mathbf{B}_0 + \mathbf{B}_1)\mathbf{v}] = 0\,. \quad (3.62)$$

No further alterations to the numerical method or the equation system are made. However, all variables being evolved by the simulation are of approximately the same order, rather than one variable, magnetic field, being much larger than the other variables in the simulation.

3.4.2 Method Validation

To this point, only a preliminary validation of the DSB formulation has been completed. It has been found in several test problems that the DSB formulation produces results which are identical to the results produced by the standard ideal MHD formulation. Further investigation regarding the utility of this scheme needs to be performed, but it has been shown to not produce incorrect results.

3.5 Inflow Boundary Condition

In order to be able to simulate the evolution of a plasma plume in a magnetic nozzle, it is of primary importance to develop a boundary condition which will introduce the plasma plume into the simulation domain, allowing for the simulation to center

on the evolution of a plasma in the magnetic nozzle, and avoid having to consider the manner in which the plasma is generated. Substantial effort was devoted to the development of such a boundary condition, referred to in this work as the inflow boundary condition. This name is however, somewhat deceptive. This boundary condition must allow for plasma to be injected in a specified region of the boundary while at the same time allowing for equilibrium with the background plasma to be maintained in the remainder of the boundary, as well as maintain equilibrium between these two regions tangent to the boundary. Over the course of this research, this boundary condition underwent many iterations. The Section 3.5.1 will detail the current form of the boundary condition, which is the form used in the magnetic nozzle simulations discussed in Chapter 4.

3.5.1 Description of Boundary Condition in its Current Form

The inflow condition currently used is a Dirichlet-type, flux-based boundary condition. The value of the solution specified at the boundary is a function of r, with three distinct regions: the inlet region where $r < r_{in}$, the edge region where $r \approx r_{in}$ and the background region where $r > r_{in}$.

The values of the conserved variables along the inflow boundary are specified in several distinct ways. The density in the inflow region ρ_{in} and the density in the background region, ρ_{bg} are specified from the input file. In the edge region, a *shape function* is used to provide a smooth transition of the value of density between the inflow and the background regions. The shape function is defined as,

$$\Gamma(r) = \frac{1}{2} erfc\left(\frac{r - r_{in}}{\epsilon\,(2r_{in})}\right), \tag{3.63}$$

where ϵ is a dimensionless parameter specifying the fraction of the nozzle radius over which the edge region, and therefore the smoothing due to the shape function, should extend. The magnitude of velocity is defined similarly, where the magnitude of velocity in the inflow region is specified from the input file v_{in}, the background velocity is always zero, and Eq. (3.63) is used to smoothly transition the velocity in the edge region. The direction of the velocity at every point along the boundary is defined so that velocity is parallel to magnetic field lines, by stating that the components of velocity obey

$$\mathbf{v} = |v(r)| \cdot \frac{\mathbf{B}}{|\mathbf{B}|} \,. \tag{3.64}$$

The magnetic field components along the inflow boundary are taken from the initially defined static magnetic field. This is one auxiliary benefit of the DSB formulation as this would be difficult to implement if the initial magnetic field where not already stored by the simulation.

Specifying the plasma pressure along the inflow boundary is the most nuanced of these boundary conditions. To ensure that the total pressure, $P_t = P + B^2/2\mu_0$ is uniform along the boundary, the user specifies a desired minimum plasma pressure P_{min} in the input file. The boundary condition then searches for the maximum magnetic field along the boundary and specifies the total pressure for the boundary to be

$$P_t = P_{min} + \frac{B_{max}^2}{2\mu_0} \,. \tag{3.65}$$

The plasma pressure at every other location is then set such that

$$P\left(r\right) = P_{min} + \frac{B^2(r) - B_{max}^2}{2\mu_0} \,, \tag{3.66}$$

Ensuring that the total pressure along the boundary is uniform.

To assist with the positivity of simulations, a simple ramp-up function is applied to the velocity and density at the inflow. Plasma pressure and magnetic field are un-altered by this ramp up. The ramp-up function is very simple: a time t_{ramp} is specified by the user in the input file. While $t < t_{ramp}$, the velocity and density at the inflow increases linearly from the background to the inflow values. Once $t \geq t_{ramp}$, the density and magnitude of velocity remain fixed at the inflow conditions. This is accomplished by expressing the ramp-up as a dimensionless factor which is multiplied with the density and velocity profiles calculated in the manner detailed above.

3.6 Far-Field Boundary Condition

There are three types of boundary conditions employed in the nozzle simulations carried out for this research. The first is the axisymmetric boundary condition, which has merited no real discussion because it is defined based on symmetry considerations. The second is the inflow condition, described in the previous section, which is of central importance to this research, and has undergone significant development. The third type of boundary condition, discussed here, is needed for the remaining two boundaries in the simulation domain that are located far away from the nozzle inflow region. As opposed to the inflow boundary condition, which drives the behavior of the entire simulation, these boundary conditions are designed with the admonition "do no harm" firmly in mind. This boundary condition interacts with the background plasma, which should have little to no influence on the evolution of the plume. Therefore, these boundary conditions, hereafter referred to as the far-field boundary condition, to the greatest degree possible, should do nothing.

A simple technique for approximating zero gradient boundary conditions is already implemented in WARPX, which accomplishes a simple approximation of a zero gradient boundary by copying the solution in the boundary cell into the ghost cells. However, problems are encountered when this was applied to the magnetic nozzle simulations investigated due to the presence of gradients in the magnetic field strength at the boundary. When this approximate zero gradient boundary condition is applied to simulations, negative pressure is observed in some situations, and in many others the boundary condition drives high velocity plasma flows in the background plasma. At times this results in the background plasma at the boundary having a higher kinetic energy density than the inflowing plasma plume itself! Even when the boundaries are placed several coil-radii away from the inflow region, where the magnetic field gradients diminish rapidly, these issues are often observed.

The source of this unusual behavior stems primarily from the use of the DSB formulation of the magnetic field. With the use of this formulation, the value of the static magnetic field in the ghost cells is calculated from the initialization function that numerically determines the field appropriate at a given location based on the magnetic field of an ideal current loop. This results in a different average value of the magnetic field in the boundary cell and ghost cells. If $\beta \ll 1$, then the change in magnetic pressure between these neighboring cells is large relative to the plasma pressure. Since the copying over of the plasma variables in the approximate zero-gradient boundary

condition occurs using the conserved variables, total plasma energy is copied over into the ghost cell, rather than plasma pressure. When this is done, since there is a substantially different magnetic field pressure in the ghost cell, the plasma pressure calculated in the ghost cell is significantly over or under estimated when the static magnetic field in the ghost cell is weaker or stronger respectively.

This boundary condition is improved by altering it so that when the total energy of the plasma is copied from the boundary cell to the ghost cell, it corrects for the change in the magnetic field energy due to the difference in the static magnetic field between the two cells. That is, the total energy in the ghost cell is defined as

$$E_g = E_b + \left(\frac{B_0^2|_g}{2\mu_0} - \frac{B_0^2|_b}{2\mu_0} \right) \tag{3.67}$$

Where g represents the value as evaluated in the ghost cell, and b represents the value as evaluated in the boundary cell. The remaining conserved variables are copied over from the boundary cell to the ghost cell without modification. At present this scheme has only been implemented to use the P_0 or cell average values of the conserved variables. Nonetheless, marked improvement is observed with the implementation of this boundary condition, and no substantial disturbances in the background plasma develops in the time scales that are considered for the simulations examined.

Chapter 4

Preliminary Results

Contents

The modeling ability developed in the preceding chapter is applied to the problem of interest, that of a magnetic nozzle channeling a high temperature, high pressure plasma for the production of thrust. A preliminary computational study of the performance of the magnetic nozzle propulsion concept is performed using a parameter regime that approximates the computational and experimental work of Winglee et al. [29]. A wide range of the parameters defining the properties of the inflowing plasma plume are investigated. The goal of this investigation is to understand how the behavior of the exhaust plume generated in these simulations is affected by the parameter regime of the plasma introduced into the magnetic nozzle. From these results, trends in the performance of the magnetic nozzle are observed, providing preliminary observations as to which range of parameters will in optimal magnetic nozzle performance, as characterized by the conversion of thermal energy to kinetic energy.

Table 4.1: Parameters Varied for Simulation Campaign

parameter	0	1	2
$n\,(m^{-3})$	2×10^{19}	2×10^{20}	2×10^{21}
$P_0\,(Pa)$	8.35	83.5	835
$V\,(m/s)$	5000	$\sqrt{10}\,5000$	50,000
$r_{coil}\,(m)$	0.01	0.1	1.0

4.1 Simulations Performed

For this preliminary simulation campaign four simulation parameters are each set at one of three different values, resulting in a total of 81 simulations performed. The parameters varied are: n, the number density of the plasma being injected into the simulation domain in m^{-3}, P_0, the minimum pressure used in the construction of the inflow boundary condition in Pa, (see section 3.5 for details), V, the velocity of the inflowing plasma in m/s, and r_{coil}, the radius of the magnetic coil in the simulation in m. The values given to each of these parameters is listed in Table 4.1. Each simulation is given a designator of the form "`SIM_R(n)(p)(v)(r)`" where one of the three levels, 0, 1, 2, for each of the four input parameters is chosen. Simulations will be referred to as such in this document.

All other simulation parameters are held constant, and chosen to match the simulations performed by Winglee et al. There are several differences between the simulations that were performed here with those of Winglee et al. that are worth noting:

- A Helmholtz coil with finite coil thickness was used in the Winglee et al. simulations. The simulations presented here use an ideal current loop of equivalent radius.
- A weaker magnetic field (and thus higher β_{th}) is necessary in the simulations presented here due to issues with maintaining positivity.
- Due to the nature of the inflow boundary condition that is used in this work, (see Section 3.5), it is not possible to place the inflow boundary at the center of the coil; in the simulations here the inflow boundary has been placed $0.2\,r_{coil}$ distance axially from the center of the coil.

- For the simulations performed here, the ion mass is $m_i = m_p$, where m_p is the proton mass. In the simulations performed by Winglee et al., the ion mass is $m_i = 16\, m_p$.

There are two other important things to note about the simulations that are performed in this work:

- The ratio, r_{inlet}/r_{coil} is held constant at 2/3 for all simulations. When the radius of the current coil is altered, the radius of the inlet is also altered. Changing this parameter scales the size of the entire device, which is not a trivial alteration as will be observed. This is due to the fact that the magnetic reynolds number R_m is a function of magnetic field gradient lengths, which scale as the size of the magnetic nozzle changes.
- The background plasma in all simulations is taken to have a density of $n = 2 \times 10^{18}\mathrm{m}^{-3}$, or one tenth of the lowest inflow plasma density. The background pressure is set by the parameter P_0, and so changes with the input parameters. The pressure in the inlet region is larger than the background pressure, but never by a factor of 10. Therefore, for all simulations the background plasma is at a higher temperature than the inflowing plasma, and therefore less resistive.

A complete description of the input parameters used in the simulations performed is provided in the input file in Appendix A.2.

The boundary conditions for these simulations are:

- $\mathbf{Z = 0}$: inflow boundary condition (Section 3.5).
- $\mathbf{Z = 3.25\, r_{in}}$: far-field boundary condition (Section 3.6).
- $\mathbf{R = 0}$: axisymmetric boundary condition.
- $\mathbf{R = 3\, r_{in}}$: far-field boundary condition (Section 3.6).

For this analysis, the attempt was made to run each simulation until the plasma plume had propagated an axial distance of $1.0\, r_{in}$. Of the 81 simulations performed, only 4 meet this criteria, while the remaining simulations experience negative pressure

Table 4.2: Key Parameters of Simulations Examined

Simulation	$\mu\,(m_i/m_p)$	$B\,(mT)$	$n\,(m^{-3})$	$T\,(eV)$	$V_{in}\,(m/s)$	$r_{coil}\,(m)$
SIM_R0110	1	11	2×10^{19}	20.5	1.58×10^4	0.01
SIM_R0111	1	11	2×10^{19}	20.5	1.58×10^4	0.1
SIM_R0112	1	11	2×10^{19}	20.5	1.58×10^4	1.0
SIM_R0102	1	11	2×10^{19}	20.5	5.0×10^3	1.0
*SIM_R1112	1	11	2×10^{20}	2.05	1.58×10^4	1.0
Winglee et al.	16	20	2×10^{19}	4.17	5.0×10^3	0.15

errors before the plume has propagated a sufficient distance.In order to differentiate the plume from the background the plume region was defined as the region in which $\rho > 0.3\rho_{in}$.

A fifth simulation which is very similar to the main simulation campaign carried out is included in this analysis because of its success in simulating to much longer propagation distances. This simulation differs from the `SIM_R1112` simulation in the main campaign only in that it uses a background density of $2.0 \times 10^{19} m^{-3}$, a factor of ten larger than the background density used in the rest of the simulations. The background pressure remains the same, resulting in a background plasma that is a factor of ten lower in temperature than the background plasma in the other simulations discussed. This will result in a more resistive, i.e. diffusive behavior in the background plasma. When results using this simulation are presented they will be signified with an (*). A summary of the key parameters of these successful simulations and that of Winglee et al. are given in Table 4.2.

The three dimensionless parameters which most characterize the behavior of resistive MHD nozzle simulations are the ratio of plasma pressure to magnetic pressure, referred to as the thermal plasma beta, β_{th}, the ratio of dynamic or kinetic pressure to magnetic pressure, referred to as the kinetic plasma beta β_k, and the ratio of advective to diffusive transport of magnetic flux, referred to as the magnetic reynolds number, R_m. The values of these dimensionless parameters are calculated for the five simulations presented in this analysis and for the simulations performed by Winglee et al., and are presented in Table 4.3 and Figure 4.1. From these it can be observed that, while it was not possible to perform simulations at a value of β_{th} as low as that

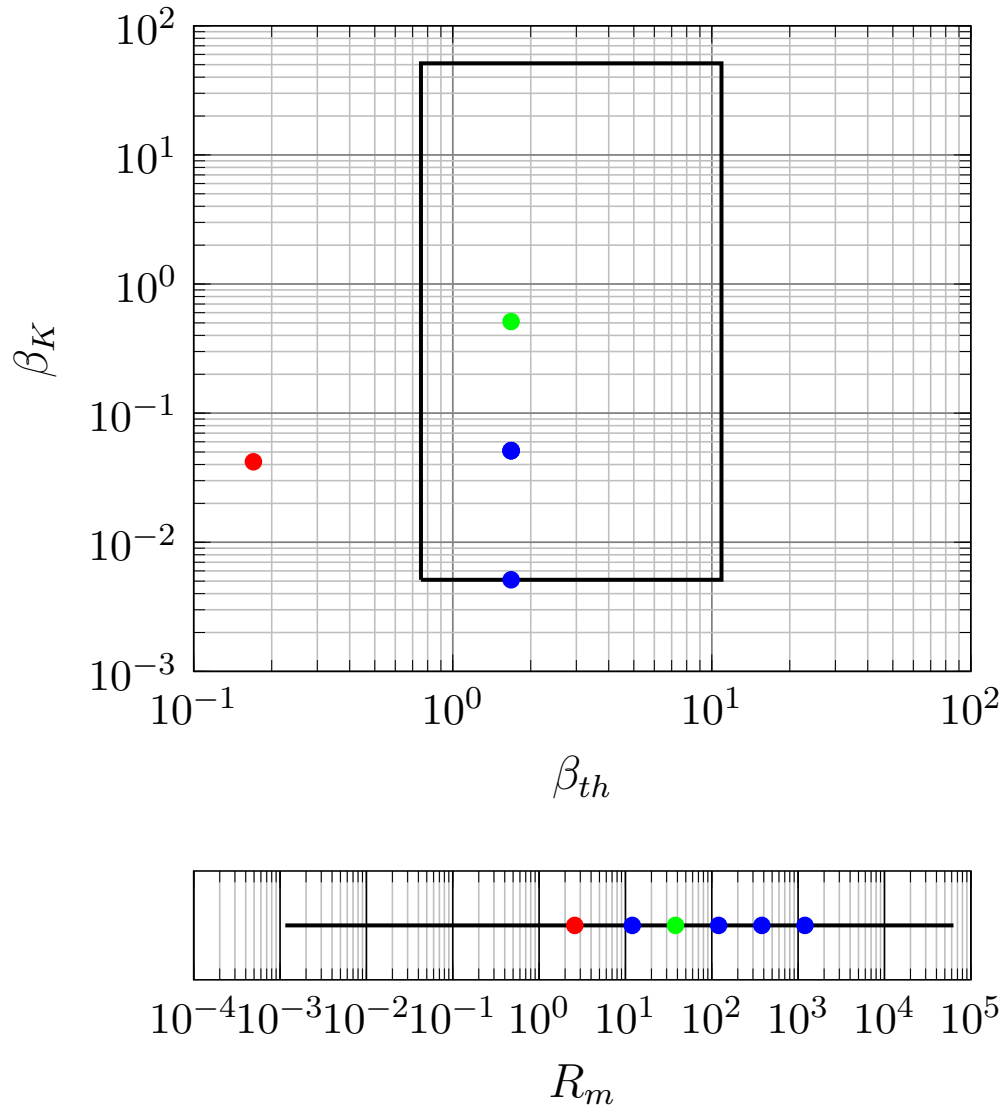


Figure 4.1: Key dimensionless parameters of simulations. Blue points represent successful simulations, green represents a similar successful simulation, and red represents the simulation performed by Winglee et al. The black bounded region represents the entire region in which simulations were attempted.

Table 4.3: Key Dimensionless Parameters of Simulations Examined

Simulation	β_{th}	β_k	R_m
SIM_R0110	1.68	5.1×10^{-2}	12
SIM_R0111	1.68	5.1×10^{-2}	120
SIM_R0112	1.68	5.1×10^{-1}	1200
SIM_R0102	1.68	5.1×10^{-3}	380
*SIM_R1112	1.68	5.1×10^{-1}	38
Winglee et al.	0.17	4.2×10^{-2}	2.6

discussed in Winglee et al., that simulations ranging over two orders of magnitude in β_k and two orders of magnitude in R_m are successfully performed. This permits a scaling study of the behavior of the plasma plume with changes in R_m and β_{th}, which is presented.

One thing to note about Winglee et al.'s simulations and analysis is that, although their chosen simulation parameters result in a magnetic Reynolds number on the order of unity, no resistivity is considered in the model used for their simulations. The non-negligible resistive diffusion in magnetic nozzles is the primary motivation for resistive MHD simulations.

4.2 Simulation Results

For the primary batch of simulations, two of the three dimensionless simulation parameters, β_k and R_m are varied to understand their effect on plume expansion and the changes in thermal and kinetic energy observed in the plume. A statistically significant sample of β_k variation is challenging to obtain due to positivity issues of the high order DG scheme. Because of this, and because of the substantially higher β_k values that these simulations are performed at, the analysis carried out in this section focuses on identifying preliminary trends in the behavior of the plasma plume and the effectiveness of the nozzle as a propulsion system as these dimensionless parameters change. The nominal magnetic Reynolds number is varied over two orders of magnitude while the other dimensionless parameters are held constant by scaling the entire simulation by two orders of magnitude in size. β_k and R_m are not completely independent of each other, but the magnetic Reynolds number is constrained to vary only over

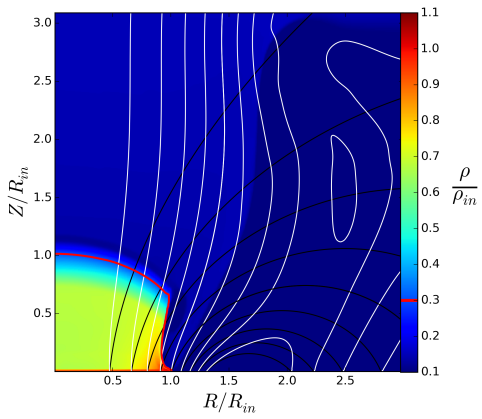
a single order of magnitude while varying the value of β_k over two orders of magnitude.

For each simulation, three different analyses are carried out: 1.) the perturbation of the magnetic field lines is studied as compared to their initial condition when the plume has propagated to $z = 1\, r_{in}$, 2.) the degree of radial expansion of each plume after having propagated to $z = 1\, r_{in}$ is compared, and 3.) the evolution of three scalar quantities obtained by integrating over the plume as it propagates. The three quantities measured are the velocity of the plume's center of mass v_{cm}, the plume's average kinetic energy per unit mass ke, and the plume's average thermal energy per unit mass te. In order to make a fair comparison between simulations in which the injection velocity and density vary over several orders of magnitude, the quantities are normalized to the nominal properties of that simulation. Therefore, v_{cm} is normalized by v_{in}, kinetic energy per unit mass is normalized by $1/2\, V_{in}^2$, and thermal energy per unit mass is normalized by $P_{in}/\rho_{in}\,(\gamma - 1)$, where P_{in} is the plasma pressure of the inflowing plasma.

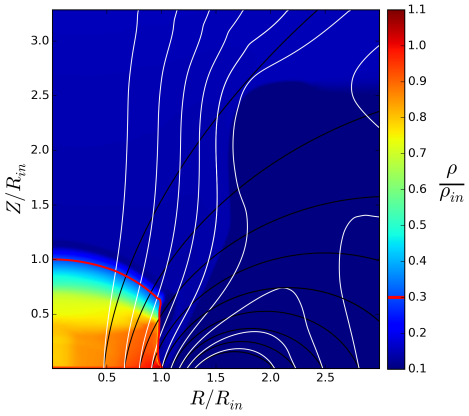
For the simulations in which only β_k is varied, as shown in Figure 4.2, the primary difference in the magnetic field configuration is that with decreasing kinetic beta, the magnetic field-lines downstream from the plume become increasingly collimated. For the simulations where only R_m number is varied, shown in Figure 4.3, there does not seem to be any significant variation in the magnetic field configurations.

When considering the radial expansion of the plume as shown in Figure 4.4, it is observed that with increasing beta kinetic, the radial expansion of the plume steadily increases. At the lowest value of β_k the plume actually contracts in radius by approximately 10% before expanding again. The highest β_k simulation, in contrast, expands continually. The variation between simulations with increasing R_m, shown in Figure 4.5, is less pronounced with higher magnetic Reynolds numbers resulting in slightly greater axial expansion.

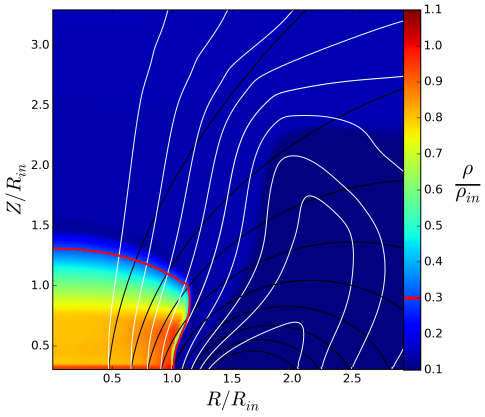
Finally, the evolution of the integrated quantities is considered. Because of the large variation in the input parameters of these simulations, the evolution of the plasma plumes they produce occur over very disparate time scales. For this reason it is decided that comparing the plumes at equivalent times, even when the times have been normalized to some characteristic velocity, does not provide a reasonable comparison between simulations. Instead, the evolution of normalized measures of v_{cm}, ke and te are measured with respect to the distance of plume propagation, normalized to r_{in}, a



(a) $\beta_k = 5.1 \times 10^{-3}$, $R_m = 3.8 \times 10^2$

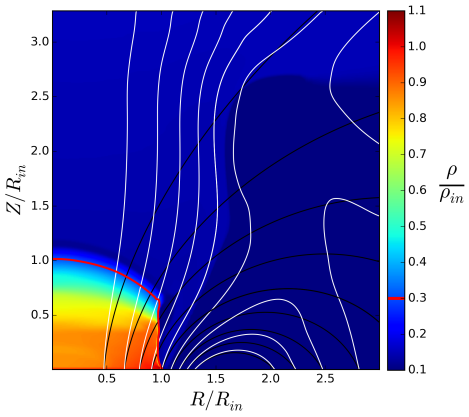


(b) $\beta_k = 5.1 \times 10^{-2}$, $R_m = 1.2 \times 10^2$

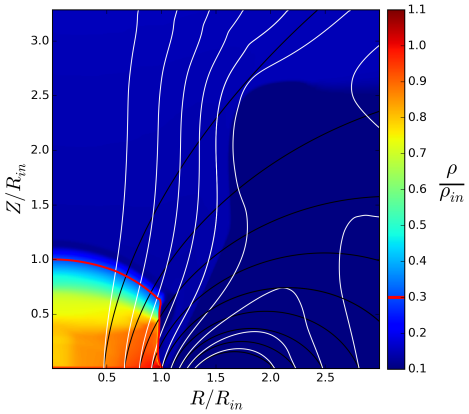


(c) $\beta_k = 5.1 \times 10^{-1}$, $R_m = 3.8 \times 10^1$

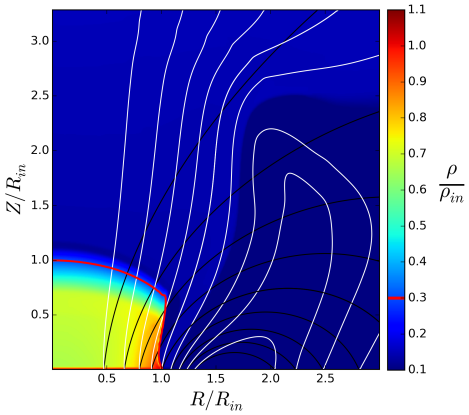
Figure 4.2: Density and magnetic field configuration as β_k is varied. As β_k decreases, increasing collimation of the magnetic field in the background plasma is observed. Additionally, with increasing R_m, a lower density is observed in the interior of the plasma plume.



(a) $R_m = 12$



(b) $R_m = 120$



(c) $R_m = 1200$

Figure 4.3: Density and magnetic field configuration as R_m is varied. The perturbation of the magnetic field in each case is approximately the same, with a somewhat smaller perturbation in the background for the $R_m = 1200$ case. The more striking difference is the decrease in density in the interior of the plasma plume with increasing R_m.

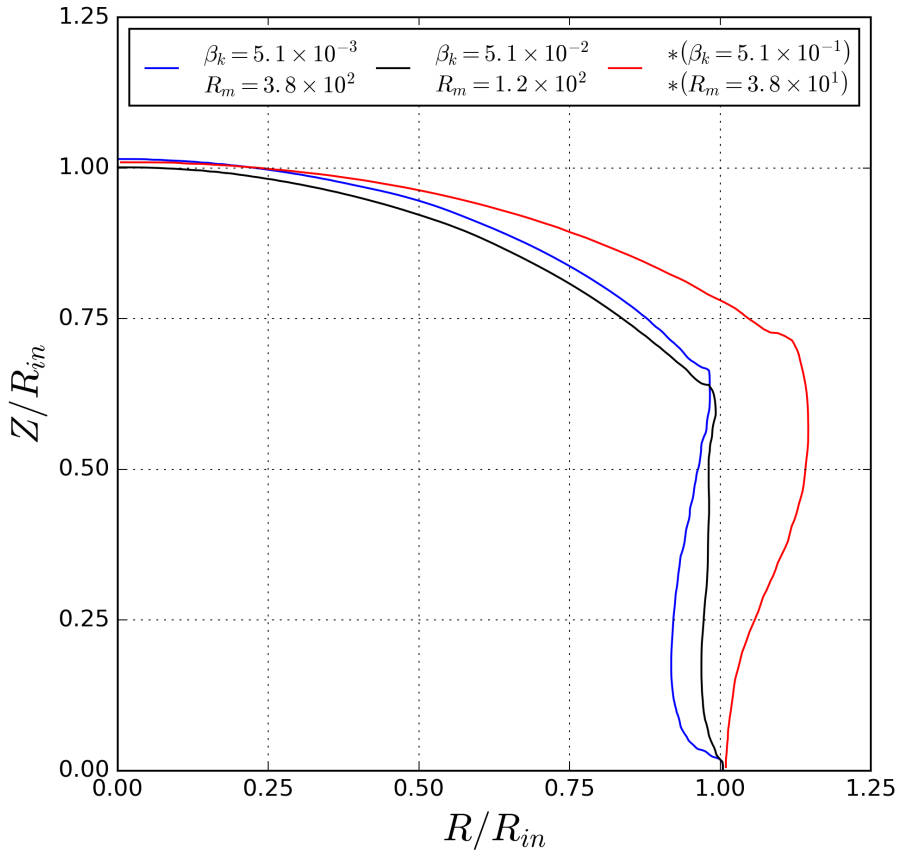


Figure 4.4: Comparison of axial plume expansion as β_k is varied. As β_k increases the radial expansion of the plume increases by approximately 10%. at distances further away from the center of the plume the plume front is observed to have propagated approximately 20% further.

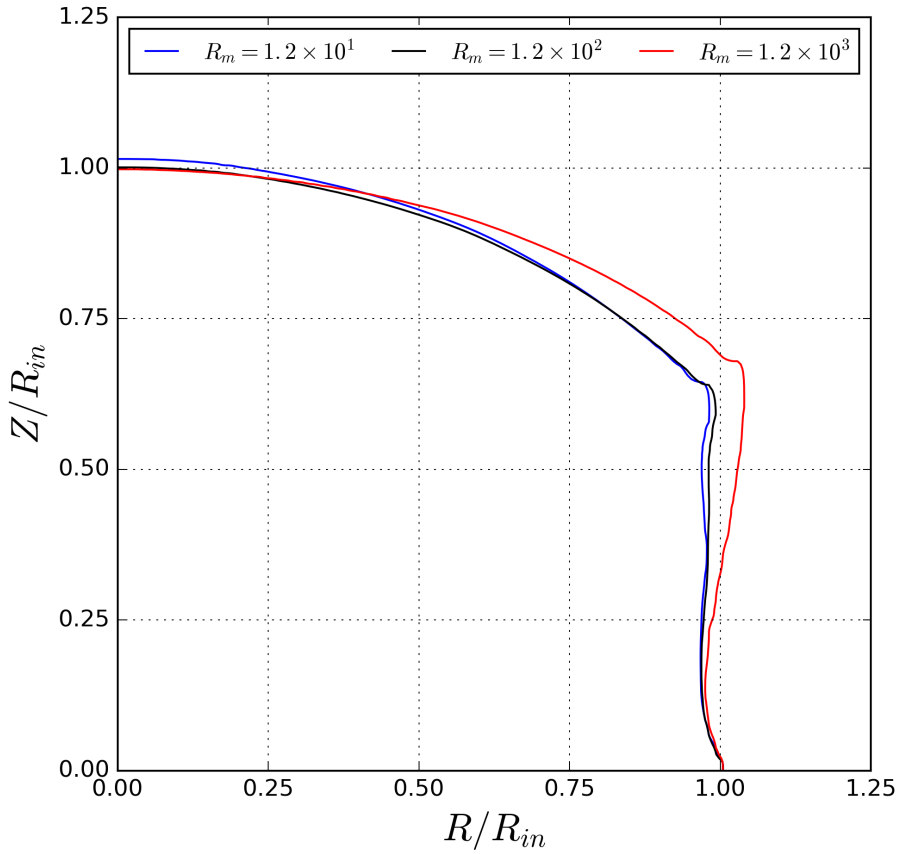


Figure 4.5: Comparison of axial plume expansion as R_m is varied. As R_m increases the degree of radial expansion increases, but in a non-uniform manner, with virtually no difference between the $R_m = 1.2 \times 10^1$ and $R_m = 1.2 \times 10^2$ cases, but an approximately 5% increase in the radial expansion of the plume for the $R_m = 1.2 \times 10^3$ case.

characteristic length scale of the simulation. The quantities plotted are: 1.) $overline v_z$, the velocity of the plume's center of mass, v_{cm} normalized to the nominal inflow velocity v_{in}, 2.) $\overline{ke}$, the specific kinetic energy of the plume, ke, normalized to the nominal specific kinetic energy, $\frac{1}{2}(v_{in})^2$, and 3.) $\overline{te}$, the specific thermal energy normalized to the nominal specific thermal energy, $P_{in}/\rho_{in}\,(\gamma - 1)$.

First, the cases in which only β_k varies, shown in Figure 4.6, are considered. There is an initial spike in v_{cm} and ke for the low β_k simulations, representing an initial acceleration of the plasma plume until it has propagated to $z = 0.1\,r_{in}$. For the remainder of the plumes propagation deceleration is experienced to a greater or lesser degree. For the lowest β_k simulation, the value of $\beta_{th}/\beta_k \approx 10^3$. This indicates that the majority of the energy in the plasma that is not due to the magnetic field is stored in the internal thermal energy, rather than the kinetic energy of bulk motion. Therefore, the acceleration of the plume is likely due to the expansion of the high pressure plasma into the low pressure background. Furthermore, because of the low β_k for the simulation shown in blue, the factors normalizing v_{cm} and ke are also relatively small, which would explain why these quantities are much larger than the other cases. For the highest beta kinetic case, the initial increase in v_{cm} and ke appears to occur over a significantly longer distance. This may be explained by the higher background density used in this simulation. The te of the plumes converge once they have propagated to $z = 1.0\,r_{in}$.

For the cases in which only R_m is varied, shown in Figure 4.7, much smaller differences are observed in the evolution of all quantities. In particular, the propagation distance over which the initial spike in v_{cm}, ke and te occurs is the same in all cases. Considering the region after this initial peak, two trends are observed: 1.) with increasing R_m, the amount of v_{cm} and ke that are retained from the initial peak increases, and 2.) with increasing R_m, the te of the plume drops off at a marginally faster rate, but results in the same final value of te at $z = 1.0\,r_{in}$.

4.3 Sensitivity to Radial Domain Size

For all simulations in the previous section, the size of the simulation domain in the radial direction is fixed at $r_{max} = 2\,r_{coil}$, $r_{max} = 3\,r_{in}$. The effect of altering this parameter, while leaving all other aspects of the simulation the same, is investigated for `SIM_R0112`. To this end, two additional simulations are carried out. For the first,

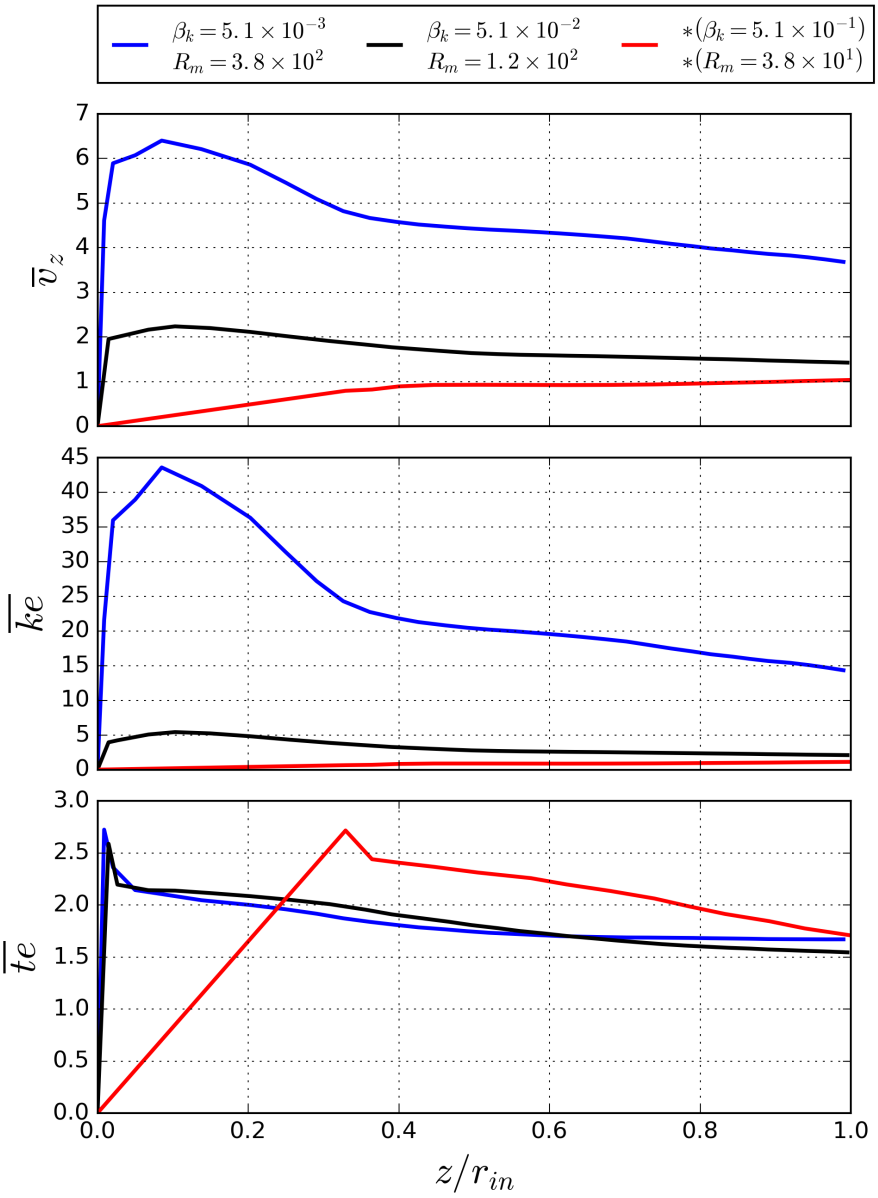


Figure 4.6: Comparison of integrated quantities as β_k is varied. For the each case $\beta_{th} = 1.68$. With increasing β_k the increase in $\overline{v}_z$ and $\overline{ke}$ becomes smaller, with the $\beta_k = 5.1 \times 10^{-1}$ case differing from the other two in that it does not reach a maximum in these values followed by a decline. The $\overline{te}$ appears to behave approximately the same for the two lower β_k simulations but behaves markedly differently for the highest beta case, taking a much longer time to reach its initial peak.

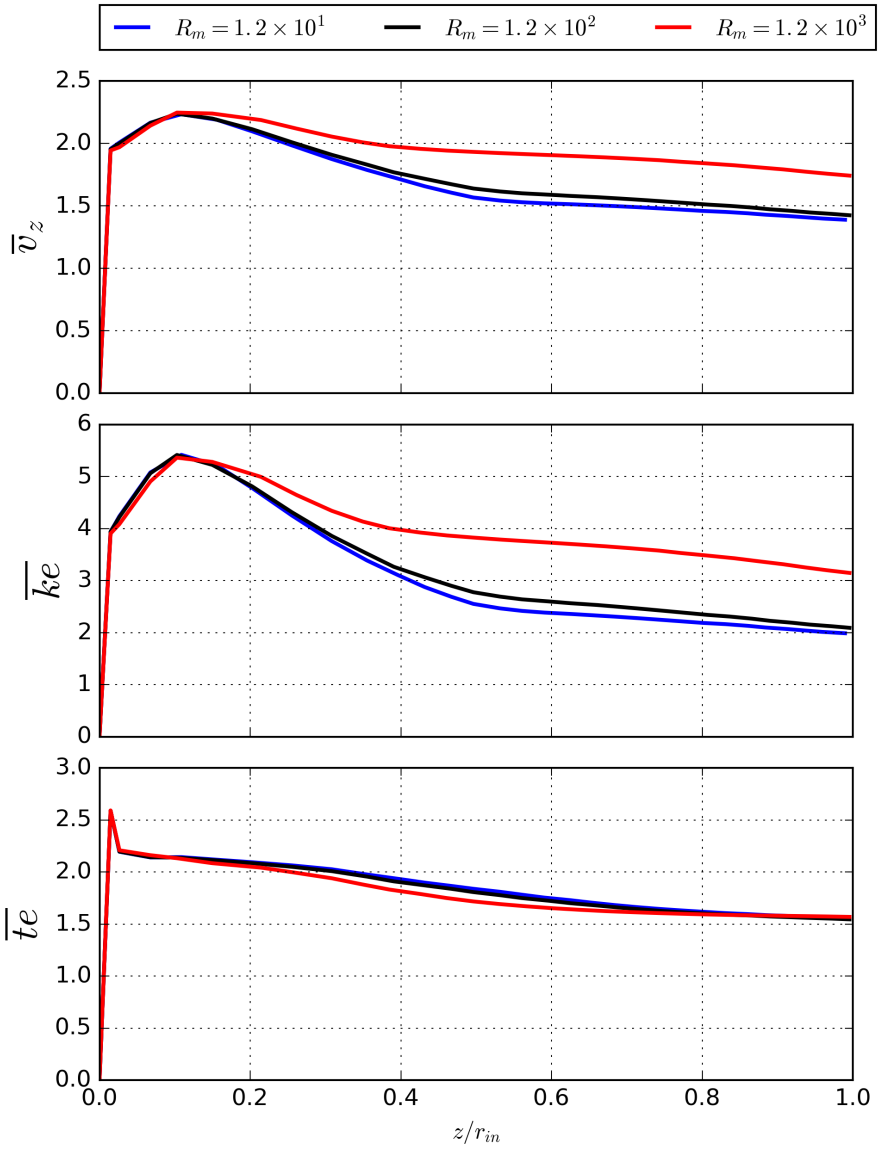
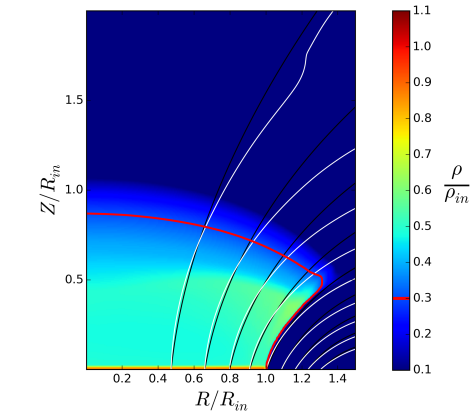


Figure 4.7: Comparison of integrated quantities as R_m is varied. All quantities measured for all simulations behave the same until the plume has propagated to approximately $z = 0.1\, r_{in}$, after which Both $\overline{v}_z$ and $\overline{ke}$ show a greater reduction from their initial peaks with decreasing R_m. The plots of $\overline{te}$ show the reverse trend, with the value of $\overline{te}$ undergoes larger reductions as R_m increases, though this variation is much less pronounced, that for the other two quantities measured.

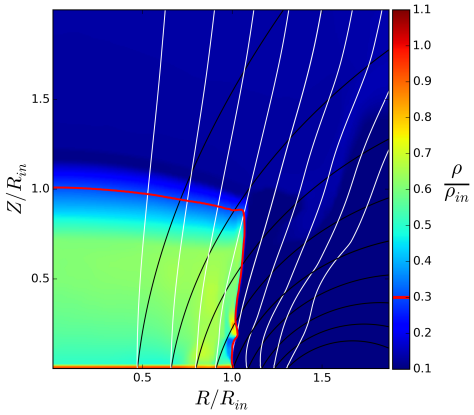
$r_{max} = 5/4\, r_{coil}$, $r_{max} = 15/8\, r_{in}$. For the second, $r_{max} = 1\, r_{coil}$, $r_{max} = 1.5\, r_{in}$.

The `SIM_R0112` simulation, like the other simulations discussed in this work, nominally has a thermal plasma beta of $\beta_{th} = 1.68$. Given this, it is contrary to expectations that such large perturbations in the magnetic field, both inside the plume as well as outside, occur. However, when the maximum radius is decreased, particularly when $r_{max} = 1\, r_{coil}$, it can be observed that the distortion of the magnetic field decreases dramatically, as shown in Figure 4.8. This radically different behavior is explained by considering the spatial variation of β_{th}. Figures 4.9 and 4.10 show the perturbed magnetic field lines in blue, the initial magnetic field lines in black and contours of β_{th} in green for the $r_{max} = 2.0\, r_{coil}$ and $r_{max} = 1.0\, r_{coil}$ cases respectively. The contours of β_{th} are labeled with the value of $\log_{10}(\beta_{th})$ at that contour. From this it is clear that, while β_{th} remains of order unity at the edge of the inlet, where the nominal values of the dimensionless parameters have been calculated, plasma beta is several orders of magnitude larger in the region flanking the inflow region, where the largest distortion of the magnetic field is observed in the simulations. For the $r_{max} = 1.0\, r_{coil}$ simulation, the value of β_{th} is also 1 to 2 orders of magnitude larger at further distances from the plume but is substantially lower in the high magnetic field region just outside of the inflow region. Since the magnetic field strength and geometry is independent of the size of the simulation domain, this significant change in the β_{th} values indicates that significant differences in the plasma pressure must be developing for different values of r_{max}.

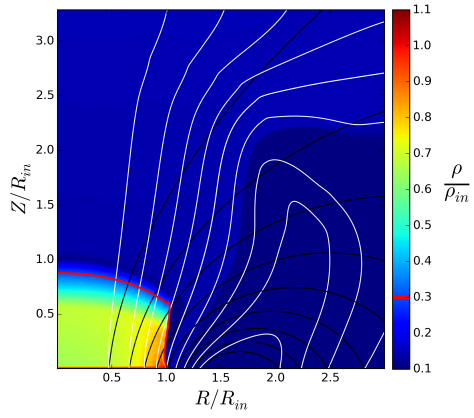
To investigate this, plots of the spatial distribution of plasma pressure, P_p, and total pressure, P_t for the $r_{max} = 1.0\, r_{coil}$ and $r_{max} = 2.0\, r_{coil}$ cases are examined, both early in time, when the plume has propagated to $z \approx 0.2\, r_{in}$, and later in time after the plume has propagated to $z \approx 0.7\, r_{in}$. Figure 4.11 shows that early in time, for the case with the standard $r_{max} = 2\, r_{coil}$, a starkly visible wave of high pressure has appeared from regions of larger r traveling towards the inlet region. by comparison, for the $r_{max} = 1\, r_{coil}$ case, this wave is absent. As the plume evolves further in time, Figure 4.12 shows that the difference in r_{max} results in a starkly different picture in terms both of P_p and P_t. For the standard case, the pressure outside the plume is higher than that inside the plume, and as a result, one can observe that the plume with the standard r_{max} has been retarded compared to the $r_{max} = 1\, r_{coil}$, as both images show the same point in time, and the standard radius plume has only traveled 90% as far as the smaller r_{max} case. This behavior could potentially explain why simulations at higher values of β_{th}, which should be less susceptible to negative pressure issues don't result in plume propagation to at least $z = 1.0\, r_{in}$. This is despite the fact



(a) $r_{max} = r_{coil} = 1.5\, r_{in}$



(b) $r_{max} = 1.25\, r_{coil} = 1.875\, r_{in}$.



(c) $r_{max} = 2\, r_{coil} = 3\, r_{in}$.

Figure 4.8: Density and magnetic field configuration as r_{max} is varied. The perturbation of the magnetic field lines is quite different for the $r_{max} = 1.0\, r_{coil}$ case from the other two simulations. It shows only a small perturbation of the applied magnetic field lines, while the other two simulations show the large scale re-arrangement of the magnetic field that is seen in the other simulations that have been performed in this work.

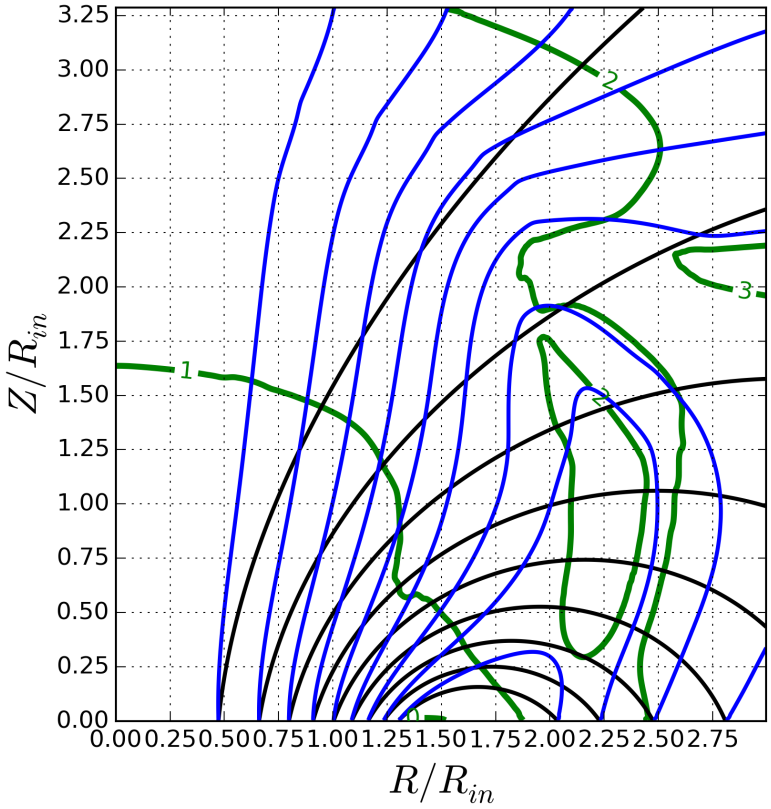


Figure 4.9: The distortion of lines of constant magnetic flux for $r_{max} = 2\,r_{coil}$ (corresponding to Figure 4.8(c)). In this figure more contours have been draw to show more clearly the change in the magnetic field. Blue lines represent the perturbed magnetic field and the black lines the applied magnetic field. The green lines represent contours of β_{th}, and are labeled with the value of $\log_{10}(\beta_{th})$ at that contour. For this simulation, the nominal value of $\log_{10}(\beta_{th}) = 0.23$. However, it is observed that β_{th} is as much as a thousand times larger in regions of the background where the largest changes in the magnetic field are observed.

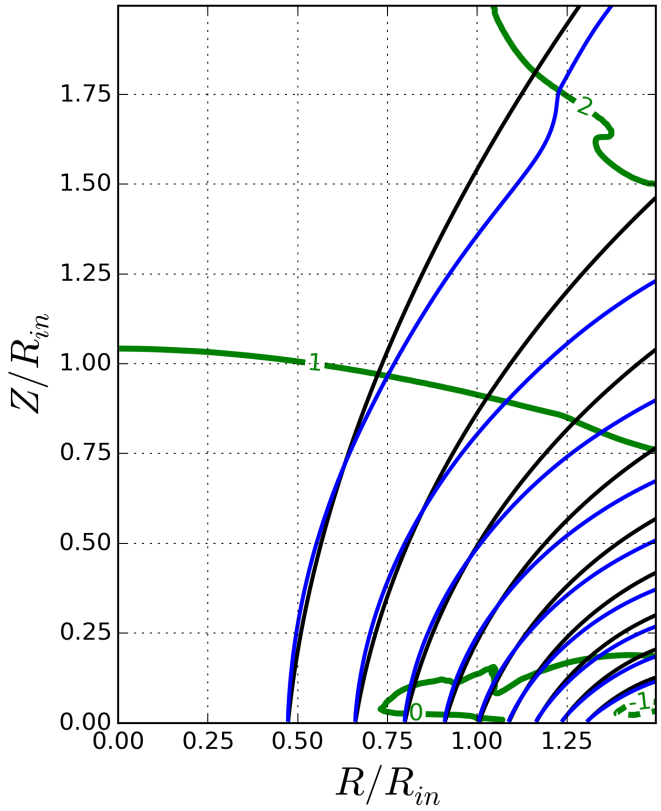
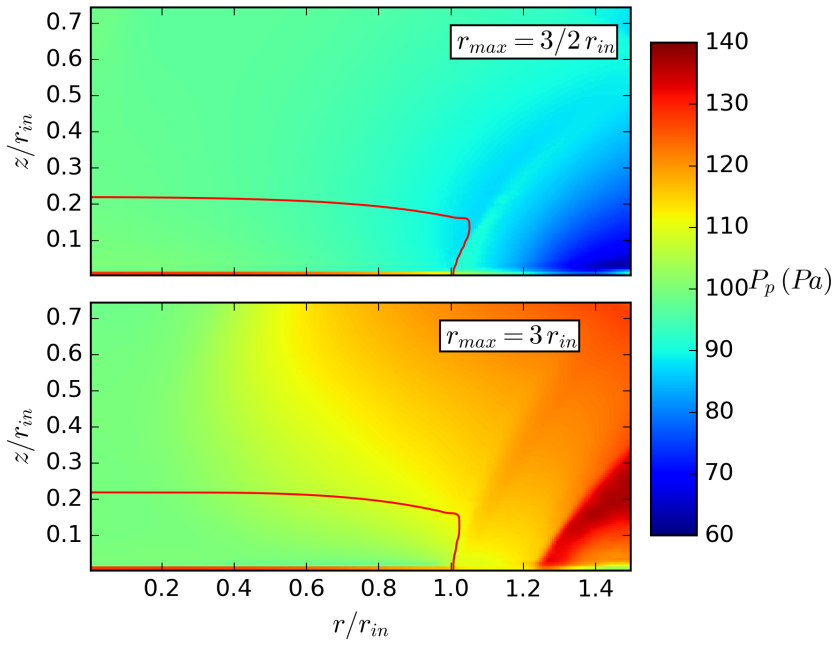
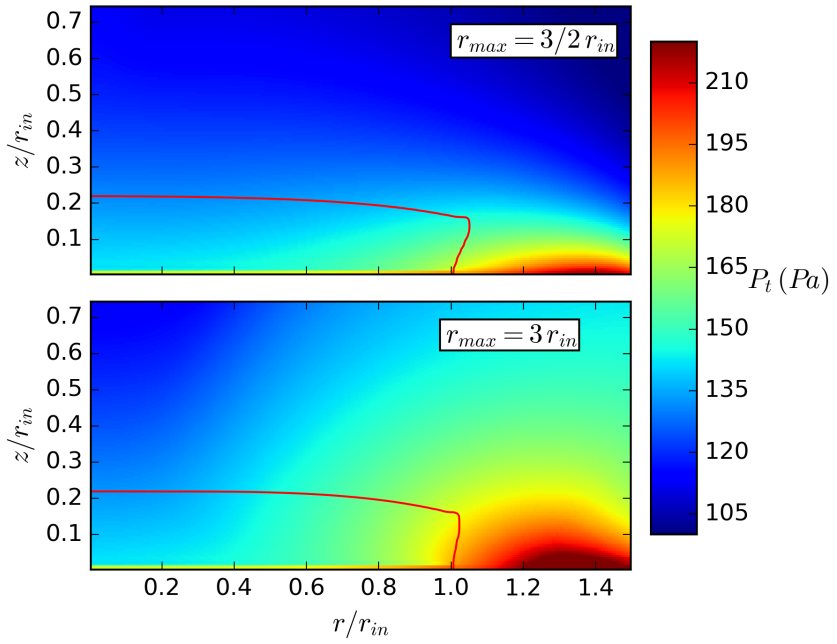


Figure 4.10: The distortion of lines of constant magnetic flux for $r_{max} = 1\, r_{coil}$ (corresponding to Figure 4.8(a)). In this figure more contours have been draw to show more clearly the change in the magnetic field. Blue lines represent the perturbed magnetic field and the black lines the applied magnetic field. The green lines represent contours of β_{th}, and are labeled with the value of $\log_{10}(\beta_{th})$ at that contour. For this simulation, the nominal value of $\log_{10}(\beta_{th}) = 0.23$. Larger values of β_{th} are also observed in the background for this simulation, just as is seen in Fig. 4.9 but they are uniformly lower by an order of magnitude.

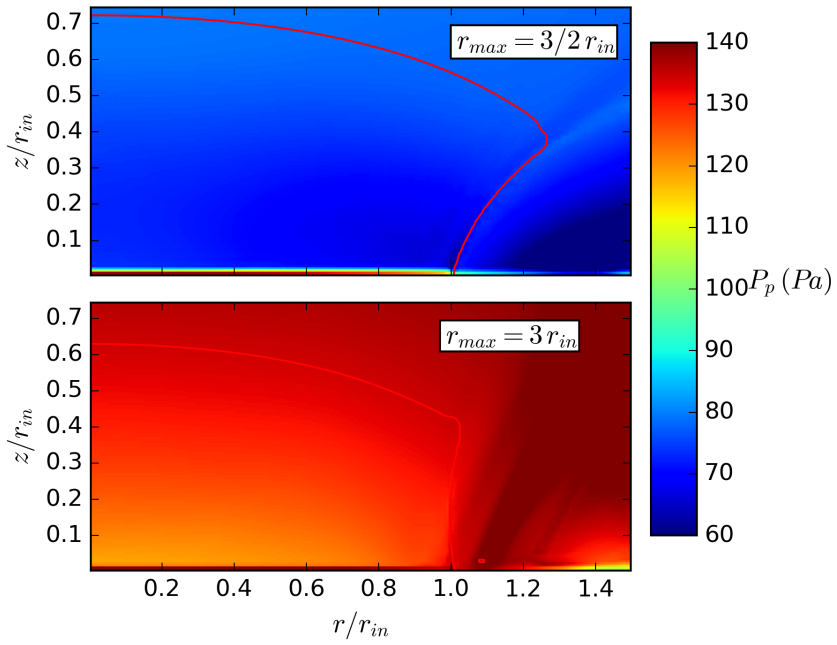


(a) plasma pressure

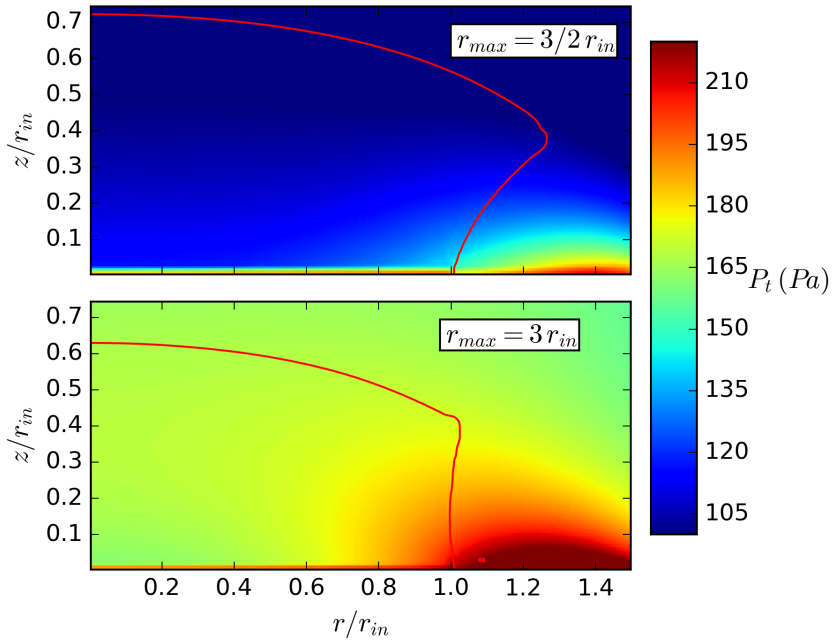


(b) total pressure

Figure 4.11: Comparison of plasma pressure, and total pressure for different values of r_{max} early in time. The red contour line represents the boundary of the plume. It is clear for both plasma pressure and total pressure that a higher pressure region has emerged from the right hand side of the simulation domain. For the plasma pressure in particular, the distribution of pressure is quite different, with regions of lowest pressure in the $r_{max} = 1.0\, r_{coil}$ being the regions of highest pressure in the $r_{max} = 2.0\, r_{coil}$.



(a) plasma pressure



(b) total pressure

Figure 4.12: Comparison of plasma pressure, and total pressure for different values of r_{max} late in time. The red contour line represents the boundary of the plume. The plasma pressure between the two cases is completely different, with the pressure in the $r_{max} = 2.0\, r_{coil}$ case having approximately double the pressure everywhere of the $r_{max} = 1.0\, r_{coil}$ case. The total pressure is also uniformly higher for the $r_{max} = 2.0\, r_{coil}$ case. It should also be noted that the $r_{max} = 1.0\, r_{coil}$ case has undergone approximately 30% greater radial expansion and has propagated approximately 10% further.

that these simulations are able to run to much longer times (as much as 10x) than successful simulations that use the same value for v_{in}. If the deceleration observed here is present in those simulations, the plume could be sufficiently decelerated by the background plasma that it never reaches $z = 1.0\, r_{in}$.

This phenomena is most likely explained by the formulation for the inflow boundary condition that has been applied here. Recall from Section 3.5 that in order to contend with issues that would develop at the edge of the injection region of the inflow, the boundary condition applied at the inflow is modified to impose a total pressure balance along the entire length of the boundary, ensuring that the inflowing plume of plasma will not try to expand radially, which was believed to be the underling issue. Recall also that, as discussed in Section 3.6, the far-field boundary conditions were found to have difficulty functioning properly if they were imposed in a region where the magnetic field gradient was large. It was for this reason that r_{max} was chosen to be substantially larger than the radius of the coil itself; because the magnetic field drops of rapidly for $r > r_{coil}$, a choice of $r_{max} = 2.0\, r_{coil}$ placed the far-field boundary condition in a region where the magnetic field gradients were sufficiently small for it to function properly. However, this meant that the inflow boundary condition, and the total pressure balance condition that it imposed, was also imposed in this low magnetic field region outside of the coil. Because the magnetic field in this region was weaker than the magnetic field in the plasma injection region, a total pressure balance resulted in a larger plasma pressure in this part of the background than in the plasma plume itself. When the simulation domain was reduced to $r_{max} = 1.0\, r_{in}$, the weakest magnetic field in the region outside of the coil remains larger than the magnetic field in the inlet region, and therefore the plasma pressure in this region is lower than in the inlet region.

Previous figures have indicated that the radial expansion of the plume increases dramatically for the $r_{max} = 1\, r_{coil}$ case. This is shown more clearly by Figure 4.13, in which a third intermediate value for r_{max} has also been included. This clearly shows that greater radial expansion occurs at equivalent propagation distances for the $r_{max} = 1\, r_{coil}$, which is consistent with the much lower pressure background that is observed for this case compared to the standard cases.

Considering the integrated quantities, shown in Figure 4.14, for the simulation in which the smallest r_{max} is used, rather than the center of mass velocity and specific kinetic energy decreasing after an initial hump, they continue to increase before

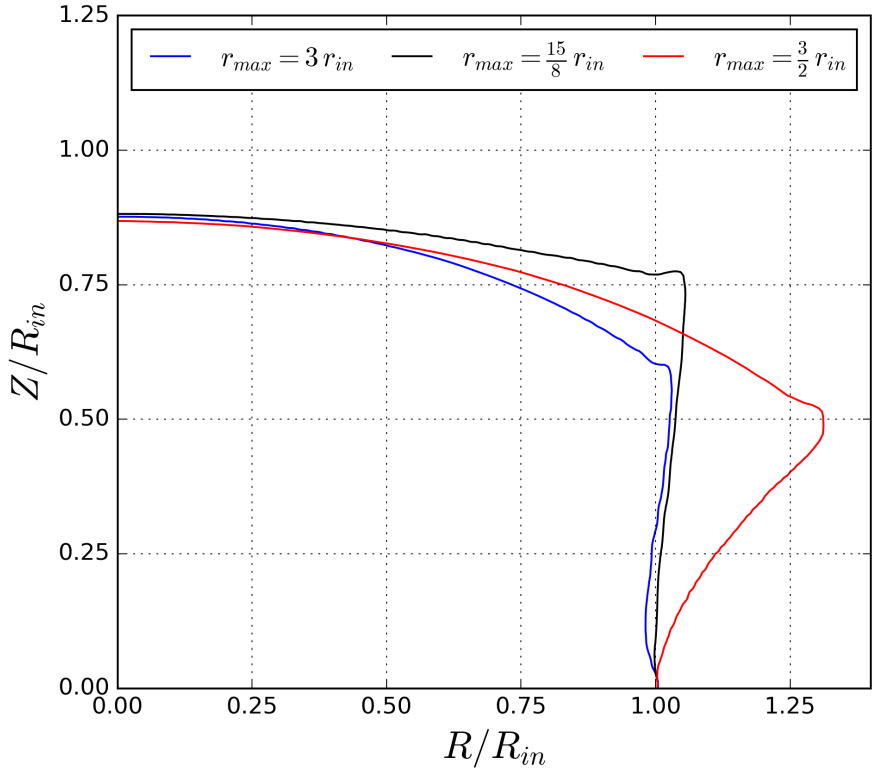


Figure 4.13: comparison of axial plume expansion r_{max} is varied. A small increase in radial expansion is observed as r_{max} is decreased from $2.0\, r_{coil}$ to $1.25\, r_{coil}$. In contrast, a substantially larger radial expansion is observed for the $r_{max} = 1.0\, r_{coil}$ case.

leveling out at a much higher level, showing clearly that the plasma plume does not under go any deceleration in contrast with the simulations in which the standard r_{max} value is used. Rather, the center of mass of the plume continues to accelerate, though more slowly until it has propagated to a distance of $z = 0.4\, r_{in}$. Additionally, the specific thermal energy drops off more rapidly for this simulation, suggesting that greater expansion is allowed, and thus that the plasma has cooled and given up more of its thermal energy.

4.4 Potential for Longer Simulations

The discussion of the previous section clearly shows that further development of the simulations, in particular of the inflow boundary condition, is needed. However, if this can be done successfully, then simulations with beams that propagate much further should be obtainable. At present, Figure 4.15 is the primary example that has been obtained demonstrating this. This shows the furthest propagation that was obtained by the *`SIM_R1112` simulation, in which the plume has propagated to a distance of $z \approx 2.9\, r_{in}$.

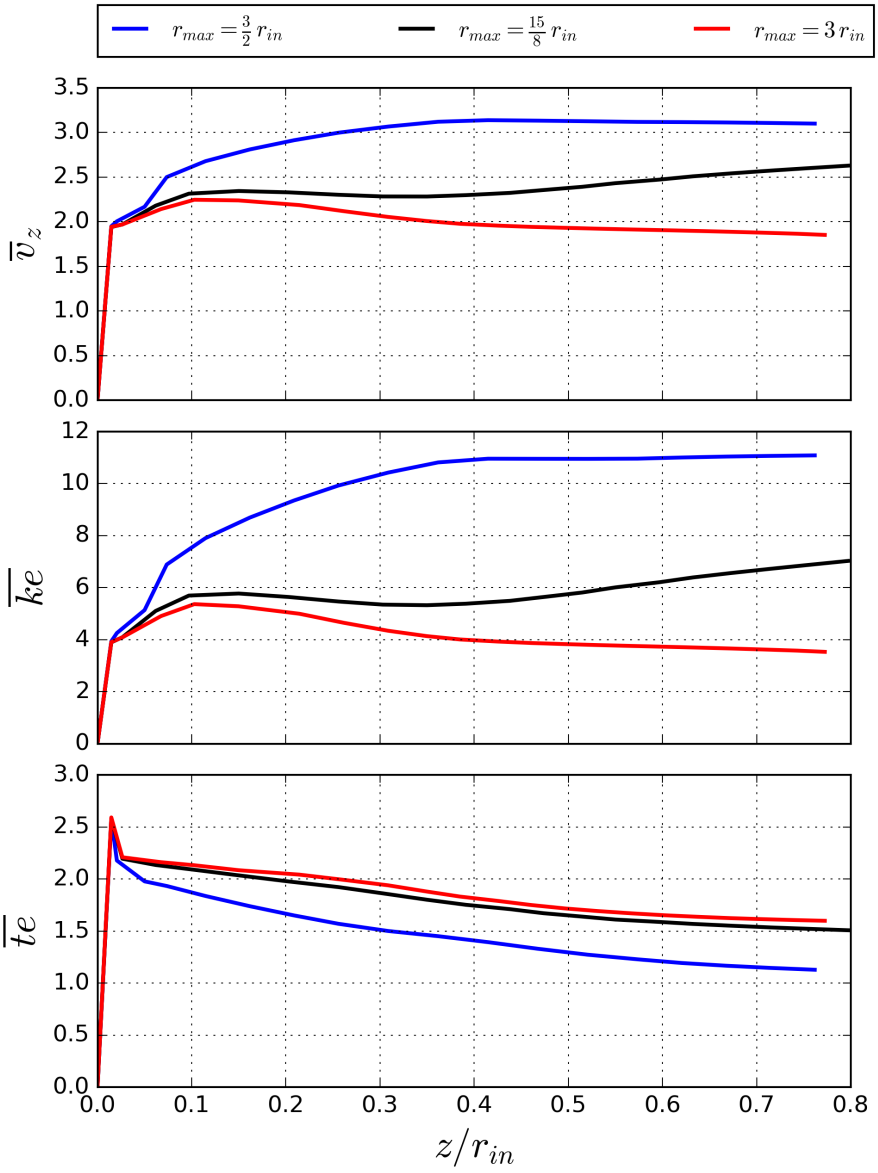


Figure 4.14: Comparison of integrated quantities as r_{max} is varied. For $\overline{v}_z$ and $\overline{ke}$, as r_{max} is decreased, the peak value reached increases, with the $r_{max} = 1.0\, r_{coil}$ never decreasing, indicating that center of mass of the plume never undergoes deceleration. Comparing the $\overline{te}$ of the simulations shows that a substantially larger decrease occurs for the $r_{max} = 1.0\, r_{coil}$ case, which is consistent with the expansion of the plume.

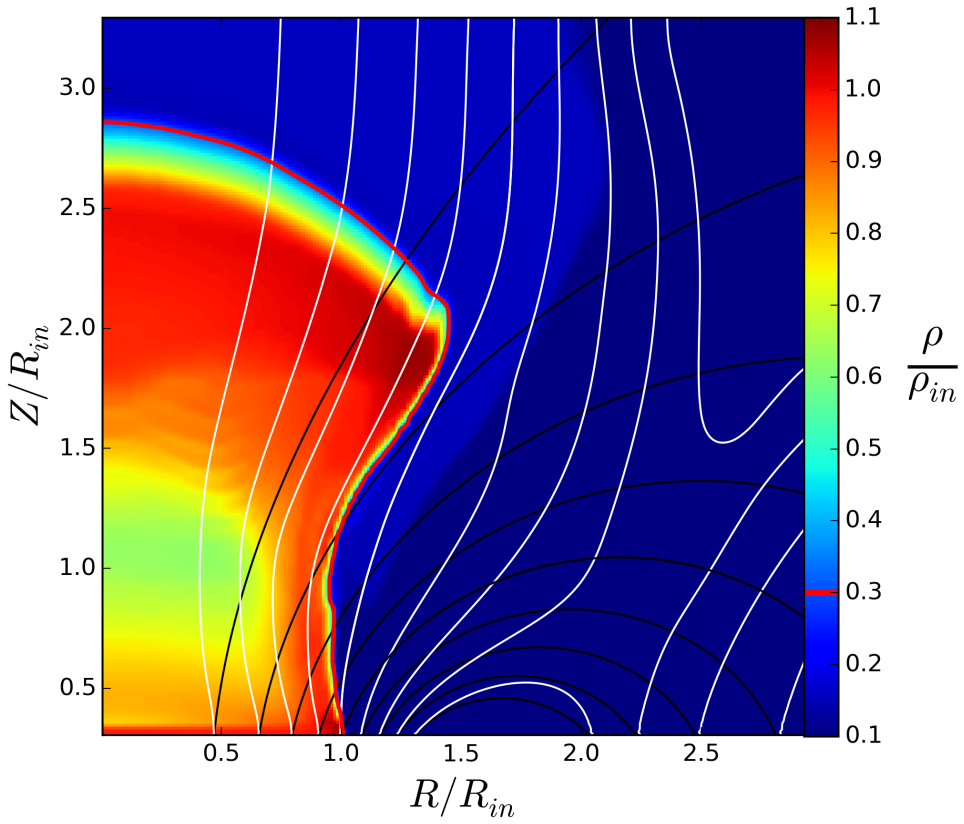


Figure 4.15: Density and contours of magnetic flux for a plasma plume that has propagated to a distance of $z \approx 2.9\, r_{in}$. This result was generated by the *`SIM_R1112` case. One of the interesting features of this simulation is the radial contraction and low density region that has developed in the region around $z = 1.0\, r_{in}$.

Chapter 5

Conclusion

Contents

This research has laid the groundwork for investigating the physics of plasma detachment from magnetic nozzles, and consequently the tools to investigate the performance of magnetic nozzle propulsion systems. The ability to carry out such simulations has been demonstrated, but the parameter range in which such simulations can be carried out must be broadened. Additionally, Quantifying and minimizing the influence of a background density or pressure that may be higher than realistic should be prioritized. Improvements in simulation robustness will broaden the reach of this simulation technique to include the important $\beta_{th} < 1$ cases. Improved understanding of the effect of the background plasma conditions and the boundary conditions will improve confidence in the validity of simulation results. As progress is made on these twin

goals this model will provide an increasingly powerful and versatile tool for the study and development of magnetic nozzles propulsion systems.

5.1 Summary of Work Completed

5.1.1 Model Development

This research makes a number of improvements to the ability to perform detailed, highly accurate simulations of magnetic nozzle systems. A partial inflow boundary condition useful for nozzle simulations is created, a technique for divergence error correction is implemented, a dynamic-static formulation of the magnetic field was developed, and an improved far-field boundary condition is added.

5.1.2 Preliminary investigations

A series of simulations which spanned a wide range of the important dimensionless parameters for resistive MHD simulations, β_k, β_{th}, and R_m are performed. Of these simulations, many suffer from density or pressure positivity violation or the background pressure prevents plume propagation. Of the simulations which are successfully performed, it is observed that some radial expansion of the plumes occur, and this expansion is larger for simulations with a higher β_k. Disturbances of the magnetic field are observed as being suspiciously large given the range of β_k and β_{th} values which are investigated. Further investigation shows that the size of the simulation domain in the radial direction can have a dramatic impact on simulation behavior. It is shown that this behavior is likely due to the nature of the inflow boundary condition imposed on the simulations. More precisely, it is found that when the size of the simulation domain in the radial direction is reduced to locations where the magnetic field is close to its peak strength, as opposed to where it is weak, the simulations behave as expected. This includes significant plume expansion and minimal perturbation of magnetic field lines. With a reduced radial domain size, a clear acceleration of the plume's center of mass, and conversion of thermal energy to kinetic energy is observed.

5.2 Recommendations for Future Investigations

This work presents a proof of concept, and lays the ground work for using WARPX to investigate the performance of magnetic nozzle technologies. It is clear from the analyses presented in Chapter 4, particularly the sensitivity to radial domain size discussed in Section 4.3, that further work developing the ability to model magnetic nozzles and validate the results of these simulations is needed before confident predictions about the realistic behavior of magnetic nozzles can be made. In order to achieve this, five areas to focus future work are identified: Implementation of a positivity preserving scheme, further improvement of the inflow boundary condition, a study of the effect of the background plasma, implementation of a resistive two-fluid model, and further evaluation of the static-dynamic magnetic field formulation.

5.2.1 Positivity Preserving Scheme

As discussed in Chapter 4, virtually all attempts at simulating magnetic nozzles have either immediately resulted in negative pressure errors or have resulted in negative pressure errors at later times. It is possible that these negative pressure errors develop inside of a single cell, or, due to the nature of the discontinuous Galerkin numerical method, it is possible that a negative pressure develops at a single node of a single cell.

A simple "positivity-preserving scheme" has been developed to address both of these issues. This scheme can be summarized as follows:

1. Verify that the cell average value of density and pressure are both positive.
   (a) If either one is not, then the offending grid cell, and the four neighboring cells are reduced to P_0 (solution is constant across grid cell)
   (b) The new value at the offending grid cell is recalculated.
2. Verify that the solution at each node in the cell results in positive density and pressure.
   (a) If it does not, then the solution of the offending grid cell is reduced to P_0.
3. The value is then accepted and the time advance proceeds.

This scheme is conservative, meaning that it will result in a positive pressure and density without the addition of mass or energy to the simulation. This scheme has been successfully implemented in WARPX for the Euler equation system, and in this setting it has shown improvements in positivity. As of this writing, a partial implementation has been carried out for the MHD equation system.

Finishing the implementation of this positivity-preserving scheme should be a top priority for this work, as it should allow for more robust simulations so that more results can be obtained over a wider range of simulation parameters.

5.2.2 Inflow Boundary Condition

As demonstrated in Section 4.3, there is more work to be done in the development of the inflow boundary condition. There are two primary sets of issues with the current formulation of the magnetic nozzle inflow condition: Issues that can develop from the use of a total pressure balance condition, and issues that can develop from the use of an ideal current loop to generate the magnetic field of the nozzle. The total pressure balance condition can interfere with the pressure in the background and has a minimum value of β_{th} allowable for a given magnetic field configuration. These issues could best be addressed if a boundary condition could be successfully implemented in which the p-total balance is applied only in the inlet region, and a constant plasma pressure is applied in the background region. A less direct means of achieving the same end result would be to implement the ability to use a non-uniform grid; with the grid spanning only the inflow region at the simulation inlet, and extending out to allow for radial expansion of the beam. The use of an ideal current loop is problematic because regions near to or including the current loop result in prohibitively large magnetic fields. Because the time step for advancing the simulation is chosen globally, these very high magnetic field regions near the coil arrest the evolution of the simulation. The practical consequence of this is that the magnetic field configurations that can be applied at present are only those in which the current loop is some distance away from the simulation domain. This constraint, in conjunction with the uniform, rectangular grid that is used in this work makes it impractical to perform simulations from the center of the nozzle; instead simulations are performed at some distance downstream, between $.2\,r_{coil}$ and $.3\,r_{coil}$. The solution is to implement the use of a current loop of finite thickness, resulting in a more realistic magnetic field. There would then be no issue associated with including the coil in the simulation domain once a means of handling the grid cells where the coil is located is developed. Finally, it would be

beneficial to implement a more sophisticated model of the inflowing plasma plume. In particular, a more rigorous description of the radial variation in the properties of the inflowing plasma would be desirable. It is recommended that work by Ahedo [30] be used as a basis for developing such a model.

5.2.3 Influence of Background

From the preliminary nozzle simulation results presented in Chapter 4, it appears that the background plasma, even when at a low density relative to the inflow, has an influence on the ability of some simulations to maintain positivity. For example, *`SIM_R1112` achieves significant plume propagation when the background density is 10% the inflow density, but is not able to develop at all when the background density is 1% the inflow density. It is also clear from the comparison of the kinetic and thermal energy plots of this simulation that the difference in the background density affects the evolution of the plume, causing the propagation of the plume into the domain to occur over a much longer time period. A study of the effect of the background density and pressure on the plume propagation is necessary.

5.2.4 Two Fluid Physics

In the WARPX code, the ability to perform full, two-fluid plasma simulations has been developed [16] [17]. This is a powerful capability that should be applied to these simulations as well. In particular, in plume edge region, where velocity, temperature, and magnetic field gradients will be largest, the full two-fluid plasma model is likely to prove useful.

5.2.5 Evaluation of Dynamic-Static Formulation

The effect of applying the dynamic-static formulation of the magnetic field, described in Section 3.4, should be further explored. In order to provide a more quantitative evaluation of this technique, the following investigation is proposed:

The circularly polarized Alfvén wave is a useful model problem in which an exact, non-linear solution to the MHD equations is known, and that solution is amenable to arbitrarily large or small perturbations in the magnetic field relative to the applied or

static magnetic field. Both the divergence error and the discretization error for the standard and the static-dynamic formulation should be investigated for different magnitude perturbations of this circularly polarized Alfvén wave problem. A description of the1 use of the circularly polarized Alfvén wave as a model problem, is described in Section 6.3.1 of the comprehensive review of divergence error correction techniques provided by Tóth [27].

Another high priority task for future development of the static-dynamic formulation is developing the ability to ensure that the static magnetic field is always generated with exactly zero divergence error. Algorithms exist for finite volume and finite difference to initialize magnetic fields with divergence error equal to machine precision. This must be a high priority for any simulations using the static-dynamic magnetic field formulation. If a divergence error is frozen into the static magnetic field component, this may limit the effectiveness of divergence error cleaning techniques.

Appendix A

Details of Code Implementation

A.1 Overview of WARPX

WARPX (Washington Approximate Riemann Plasma) is a powerful, multi-dimensional, general geometry, structured code for the solution of partial differential equations. A variety of numerical methods have been implemented in WARPX, with an emphasis on the use of the Runge-Kutta Discontinuous Galerkin algorithm for the solution of unsteady, hyperbolic PDEs. WARPX requires minimal dependencies. All that is needed is a comprehensive installation of python 2.7, (the 'Anaconda' package maintained by Continuum Analytics is highly recommended), the software construction tool 'SCons', an MPI package such as openMPI, and a parallel HDF5 package compatible with the chosen MPI package. WARPX is capable of using other packages such as PETSc and lapack, but these are not required. WARPX is implemented in C++, and takes full advantage of the object oriented approach to software development.

The source code is contained in eight sub-directories:

1. `lib`
2. `etc`
3. `solvers`
4. `hyper`

5. `hyperapps`
6. `variables`
7. `xwarpx`
8. `buildconfig`

Of these sub-directories, `solvers, hyper` and `hyperapps`, which contain the source code which implements the algorithms used for simulation, were the only portions of WARPX that were modified and added to in to in the course of this research. The remaining sub-directories contain the source code which establishes the framework that these algorithms are implemented on top of, handles input file parsing, output file generation, and interaction with the parallelization library, among other tasks unrelated to the numerical simulation.

A.2 Input File Generation

The input file used in this research is listed below.

```
# -*- python -*-

# Version 3
# Created by Colin Glesner
# Last Edited on September 28th, 2016
# Input file used for resistive MHD
# magnetic nozzle simulations used in thesis.

# Changes from Version 2:
# V2 resulted in simulation domains which were often
# much, much larger than necissary. Want to go back to
# having constant axial domain size.


################################################
#------IMPORTANT:------------------------------#
#----------MACRO-FILE-USED-TO-EXPAND-----------#
#----------THIS-INPUT-FILE-IS:-----------------#
################################################

#include "resistivephdsbidealmacro-radial.pin" #


#################################################################
#-----------PARAMETERS-VARRIED-IN-THESE-SIMULATIONS----------#
#################################################################
```

```
27
28 # (~) coulomb logarithm (acts as switch for resistivity)
29 LOGLAMBDA = $L
30
31 # (1/m^3) number density of inflow
32 Nin = $N
33
34 # (N/m^2) minimum pressure at inflow
35 PRES0 = $P
36
37 # (m/s) velocity of inflow
38 VELin = $V
39
40 # (m) radius of ideal current loop
41 Rcoil = $R
42
43 ################################################################
44 #--------------------PHYSICAL-CONSTANTS-----------------------#
45 ################################################################
46 import math
47 PI = math.pi
48
49 ME        = 9.11e-31  # (kg) electron mass
50 MI        = 1.67e-27  # (kg) hydrogen ion mass
51 GAS_GAMMA = 5./3.     # (~) ratio of specific heats
52 MU0       = PI*4.0e-7 # (H/m) pemiability of free space
53 KB        = 1.38e-23  # (J/K) Boltzman's constant
54 Z         = 1.        # (~) ion charge number
55 QE        = 1.602e-19 # (C) fundamental charge
56 EPS0      = 8.854e-12 # (F/m) vacuum permitivity
57 SAFETY_FACTOR = 4.    # FoS for resistive time step restriction
58
59
60
61 ################################################################
62 #----------------MISCELLANEOUS-PARAMETERS-------------------#
63 ################################################################
64
65 # number of equations in each equation system
66 MEQN    = 9 # resistive MHD with error correction
67 MEQN_B0 = 3 # applied/static magnetic field components
68 MEQN_J  = 3 # current components
69
70 # Standard WARPX parameter used in
71 # time step calculation
72 FUDGE = 1.0
73
74 #~~~~~~~~~~~~~~~~~~~(BEGIN-DEPRICATED)~~~~~~~~~~~~~~~~~~~~~~#
75 #
76 # WARPX takes a parameter called spatial order
77 # and represents the solution in DG using a polynomial
78 # of order (SP_ORDER - 1).
79 # In 2-D DG simulations, spatial order of accuracy is
80 # in general not equal to (polynomial_order + 1).
81 # It would be more correct to state the polynomial
82 # order used to represent the solution. In this case,
83 # polynomial order 1 was used.
84 #
```

```
SP_ORDER = 2 # (coresponds to polynomial order 1)
#
#
# The inflow boundary condition solver
# currently in use does not specify the pressure
# of the inflowing plasma, and does not need
# the corresponding (PRESin) parameter. However, at the moment
# the code still expects this variable to be passed to
# the setup method for the inflow boundary condition, and
# it is used by the ramp time calculation in the set up.
#
PBmax = 132.25 # SPECIFIC TO Zcoil=-0.2*Rcoil
PBin = 81.36   #           ..
PRESin = (PBmax - PBin) + PRES0
#
#~~~~~~~~~~~~~~~~~~~~(END-DEPRICATED)~~~~~~~~~~~~~~~~~~~~~~#



#################################################################
#--------------FIXED-AND-DERIVED-PARAMETERS-------------------#
#################################################################

# background
n0   = 2.0e18 # (1/m^3) number density of background
RHO0 = n0*MI  # (kg/m^3) mass density of background

# ideal current loop
coil_field = 0.011 # (T) magnetic field at center of loop
N_STEPS = 1024     # (~) steps taken in Biot-Savart calculation
Zcoil = -0.2*Rcoil # (m) axial location of center of loop

# inflow condition
RHOin = Nin*MI          # (kg/m^3) mass density of inflow
INLET_MAX = 2./3.*Rcoil # (m) radius of inlet region
EPSILON = 0.01          # (~) edge smoothing factor
RAMP_FACTOR = 1.        # (~) inflow ramp-up factor

# divergence correction
Binlet = 0.0143 # (Tesla) magnetic field (determined by magnetic field condition)
a = math.sqrt(GAS_GAMMA*PRESin/RHOin)
t1 = a*a + (Binlet*Binlet)/(MU0*RHOin)
t2 = math.sqrt(t1*t1 - 4.0*a*a*(Binlet*Binlet)/(MU0*RHOin))
cf = math.sqrt(0.5*(t1+t2))         # (m/s) fast magnetosonic speed
GAMMA = 2.0*(cf+VELin)              # (m/s) speed of error propogation
RATIO = 0.18                        # (~) ratio of diffusive to advective transport

# determine characteristic times
ADVECTIVETIME = INLET_MAX/VELin     # (s) characteristic time for advective wave
WAVETIME = INLET_MAX/(cf+VELin)     # (s) characteristic time for fastest wave
RAMPTIME = RAMP_FACTOR*INLET_MAX/cf # (s) ramp-up time for inflow condition
DELTA = WAVETIME/10.0               # (s) reasonable time between outputs

# simulation domain
ZDESIRED = 1.0*INLET_MAX         # (m) desired distance that we want the plume to
    propogate to
RMIN = 0.                        # (m) location in R of lower edge (must be zero!)
RMAX = 2.0*Rcoil                 # (m) location in R of upper edge
```

```
ZMIN = 0.                          # (m) location in Z of lower edge
FOS  = 3.0                         # (~) factor of safety on domain size
ZMAX = FOS*ZDESIRED                # (m) location in Z of upper edge
dR   = Rcoil/128.                  # (m) length of grid cells in R direction
dZ   = Rcoil/128.                  # (m) length of grid cells in Z direction
RCELLS = int((RMAX-RMIN)/dR)       # (~) number of grid cells in the R direction
ZCELLS = int((ZMAX-ZMIN)/dZ)       # (~) number of grid cells in the Z direction

# time
TEND =  RAMPTIME+ZDESIRED/VELin  # (s) end time for simulation
NOUT =  int(TEND/DELTA)          # (~) number of outputs the simulation should
    generate.


#################################################################################
#------------------BEGIN-MAIN-BODY-OF-INPUT-FILE----------------------#
#################################################################################

# begin WARPX input file definition
<warpx>

 ###################################
 #-----DEFINE-BASIC-PARAMETERS----#
 ###################################

 # simulation name
 Simulation = ThesisNozzleSimulations

 # simulation verbosity:
 # debug:    all messages written to STDOUT and .log
 # info:     all messages written to .log, step completion written to STDOUT
 # warning:
 # error:
 # critical:
 # disabled:
 Verbosity = info

 # begin simulation definition
 <ThesisNozzleSimulations>
   Type = WxSolver
   Kind = comboSolver

   Time = [0.0, TEND]
   Out = NOUT
   Dt = 1.0

   # grid on which to solve equations
   <grid>
     Type = WxGridBox

     Lower = [RMIN, ZMIN]
     Upper = [RMAX, ZMAX]
     Cells = [RCELLS, ZCELLS]

   </grid>

   # variable array definitions
   <[PAR_ARRAY("qold", "grid", MEQN*SP_ORDER*SP_ORDER)]>
```

```
<[PAR_ARRAY_NOWRITE("qnew", "grid", MEQN*SP_ORDER*SP_ORDER)]>
<[PAR_ARRAY_NOWRITE("dq", "grid", MEQN*SP_ORDER*SP_ORDER)]>
<[PAR_ARRAY_NOWRITE("q1", "grid", MEQN*SP_ORDER*SP_ORDER)]>
<[PAR_ARRAY("b0", "grid", MEQN_B0*SP_ORDER*SP_ORDER)]>
<[PAR_ARRAY("j", "grid", MEQN_J*SP_ORDER*SP_ORDER)]>



#################################################################
#--------------------DEFINE-SUBSOLVERS-----------------------#
#################################################################

# Initialize qold,qnew
<initialcond>
  Type = WxSubSolver
  Kind = funcDGArraySetter
  WriteVars = [qold, qnew]
  OnGrid = grid
  spatialOrder = SP_ORDER
  meqn = MEQN

  <func>
    Type = WxFunction
    Kind = phDsbIdealMhdIdealCurrentLoop

    MU0 = MU0
    GAS_GAMMA = GAS_GAMMA

    COIL_RADIUS = Rcoil
    COIL_CTR = Zcoil
    N_STEPS = N_STEPS

    RHOBG = RHO0
    COIL_FIELD = coil_field
    PRESBG = PRES0
  </func>

</initialcond>

# initialize the static magnetic field component
<initialb0>
  Type = WxSubSolver
  Kind = funcDGArraySetterExtBox
  WriteVars = [b0]
  OnGrid = grid
  spatialOrder = SP_ORDER
  meqn = MEQN_B0

  <func>
    Type = WxFunction
    Kind = DsbIdealMhdAuxIdealCurrentLoop

    COIL_RADIUS = Rcoil
    COIL_CTR = Zcoil
    N_STEPS = N_STEPS

    COIL_FIELD = coil_field
  </func>
```

```
257
258   </initialb0>
259
260   # define a subsolver for the frequency time checker.
261   <checkrestime>
262     Type = WxSubSolver
263     Kind = checkResistiveTime
264
265     ReadVars = [qold, b0]
266     OnGrid = grid
267
268     gas_gamma = GAS_GAMMA
269     mu0 = MU0
270     gamma = GAMMA
271     ion_charge_number = Z
272     electron_mass = ME
273     ion_mass = MI
274     fundamental_charge = QE
275     permitivity = EPS0
276     boltzmann = KB
277     coulomb_log = LOGLAMBDA
278     safety_factor = SAFETY_FACTOR
279   </checkrestime>
280
281   # use qold to compute dq
282   <[DGRHS("dgrhs_qold", "qold", "b0","j","dq")]>
283
284   # use qold to compute q1
285   <rk_stage1>
286     Type = WxSubSolver
287     Kind = linearCombiner
288
289     OnGrid = grid
290
291     ReadVars = [qold, dq]
292     coeffs   = [1.0, 1.0]
293     WriteVars = [q1]
294
295   </rk_stage1>
296
297   # subsolver to apply limiter to q1
298   <[DGLIMIT("q1_limit", "component", "q1")]>
299
300   # use q1 to calculate dq
301   <[DGRHS("dgrhs_q1", "q1", "b0","j","dq")]>
302
303   # use dq q1 and qold to compute qnew
304   <rk_stage2>
305     Type = WxSubSolver
306     Kind = linearCombiner
307
308     OnGrid = grid
309
310     ReadVars = [qold, q1, dq]
311     coeffs   = [0.5, 0.5, 0.5]
312     WriteVars = [qnew]
313
314   </rk_stage2>
```

```
315
316   # subsolver to apply limiter to qnew
317   <[DGLIMIT("qnew_limit", "component", "qnew")]>
318
319   # subsolvers to calculate current before each RKDG step.
320   <[JAUX("jqold", "j", "qold", "b0")]>
321   <[JAUX("jq1", "j", "q1", "b0")]>
322
323   # subsolvers to apply limiter to j
324   <[CURRENTLIMIT("j_limit","component","j")]>
325
326   # define Rmin boundary condition applicators
327   <[BC("RminBCqold", "bcAxisPHDsbIdealMhdDG2d", "qold", 0, "lower")]>
328   <[BC("RminBCq1", "bcAxisPHDsbIdealMhdDG2d", "q1", 0, "lower")]>
329   <[BC("RminBCqnew", "bcAxisPHDsbIdealMhdDG2d", "qnew", 0, "lower")]>
330
331   # define Rmax boundary condition applicator
332   <[PRIMITIVECOPYBC("RmaxBCqold", "qold", "b0", 0, "upper")]>
333   <[PRIMITIVECOPYBC("RmaxBCq1", "q1", "b0", 0, "upper")]>
334   <[PRIMITIVECOPYBC("RmaxBCqnew", "qnew", "b0", 0, "upper")]>
335
336   # define Zmin boundary conditions
337   <[PTOTALBALNOGHOST("ZminBCqold","qold","b0","j",1,"lower", "RHOin","VELin","
       PRESin","RHO0","PRES0","coil_field","INLET_MAX","EPSILON","RAMP_FACTOR")]>
338   <[PTOTALBALNOGHOST("ZminBCq1","q1","b0","j",1,"lower", "RHOin","VELin","PRESin","
       RHO0","PRES0","coil_field","INLET_MAX","EPSILON","RAMP_FACTOR")]>
339
340   # define dummy zmin boundary condition
341   <[BC("ZminBCdummyqold","bcCopy","qold",1,"lower")]>
342   <[BC("ZminBCdummyq1","bcCopy","q1",1,"lower")]>
343   <[BC("ZminBCdummyqnew","bcCopy","qnew",1,"lower")]>
344
345   # define Zmax boundary condition
346   <[PRIMITIVECOPYBC("ZmaxBCqold", "qold", "b0", 1, "upper")]>
347   <[PRIMITIVECOPYBC("ZmaxBCq1", "q1", "b0", 1, "upper")]>
348   <[PRIMITIVECOPYBC("ZmaxBCqnew", "qnew", "b0", 1, "upper")]>
349
350   # define boundary conditions for current density
351   <[BC("Rminj","currentbcAxisDsbIdealMhdDG2d","j",0,"lower")]>
352   <[BC("Rmaxj","bcCopy","j",0,"upper")]>
353   <[BC("Zminj","bcCopy","j",1,"lower")]>
354   <[BC("Zmaxj","bcCopy","j",1,"upper")]>
355
356   # define phdiffusion subsolver
357   <[PHDIFFUSION("phdiffusionqnew","qnew","true")]>
358
359   # Set qnew back to qold so it can be used in hyperbolicSolve
360   <copier>
361     Type = WxSubSolver
362     Kind = linearCombiner
363
364     OnGrid = grid
365     ReadVars = [qnew]
366     coeffs   = [1.0]
367     WriteVars = [qold]
368   </copier>
369
370   # define step to check for NANs in qold
```

```
371   <isnan>
372     Type = WxSubSolver
373     Kind = isNANCheck
374
375     OnGrid = grid
376     ReadVars = [qold]
377   </isnan>
378
379
380
381   ###################################################################
382   #------------------DEFINE-SUBSOLVER-STEPS--------------------#
383   ###################################################################
384
385   # initialize the array
386   <[SUBSOLVERSTEP("initArrays", "initialcond, initialb0", "qnew, qold, b0")]>
387
388   # compute RK stage 1
389   <[SUBSOLVERSTEP("rk_stage1_step", "dgrhs_qold, ZminBCqold, rk_stage1")]>
390
391   # limit q1
392   <[SUBSOLVERSTEP("limit_q1_step", "q1_limit", "q1")]>
393
394   # compute RK stage 2
395   <[SUBSOLVERSTEP("rk_stage2_step", "dgrhs_q1, ZminBCq1, rk_stage2")]>
396
397   # limit qnew
398   <[SUBSOLVERSTEP("limit_qnew_step", "qnew_limit", "qnew")]>
399
400   # apply boundary conditions
401   <[SUBSOLVERSTEP("applyBCqold", "RminBCqold, RmaxBCqold, ZminBCdummyqold,
       ZmaxBCqold")]>
402   <[SUBSOLVERSTEP("applyBCqnew", "RminBCqnew, RmaxBCqnew, ZminBCdummyqnew,
       ZmaxBCqnew")]>
403   <[SUBSOLVERSTEP("applyBCq1", "RminBCq1, RmaxBCq1, ZminBCdummyq1, ZmaxBCq1")]>
404
405   # Calculate current and apply boundary conditions to it and limit it.
406   <[SUBSOLVERSTEP("current_qold", "jqold")]>
407   <[SUBSOLVERSTEP("current_q1", "jq1")]>
408   <[SUBSOLVERSTEP("apply_current_BCs","Rminj,Rmaxj,Zminj,Zmaxj")]>
409   <[SUBSOLVERSTEP("limit_current","j_limit","j")]>
410
411   # Check for the resistive time restriction
412   <[SUBSOLVERSTEP("check_R_time","checkrestime")]>
413
414   # apply diffusion to error correction potential to qnew.
415   <[SUBSOLVERSTEP("applydiffusionqnew","phdiffusionqnew","qnew")]>
416
417   # copy step
418   <[SUBSOLVERSTEP("copy", "copier")]>
419
420   # check variables for nans
421   <[SUBSOLVERSTEP("isnancheck", "isnan")]>
422
423
424   ###################################################################
425   #---------------------DEFINE-SOLVER-SEQUENCE----------------------#
426   ###################################################################
```

```
427
428   <SolverSequence>
429     Type = WxSolverSequence
430
431     StartOnly = [initArrays, applyBCqold]
432     PerStep = [isnancheck, \
433                current_qold, limit_current, apply_current_BCs, \
434                check_R_time, \
435                rk_stage1_step, limit_q1_step, applyBCq1, \
436                current_q1, limit_current, apply_current_BCs, \
437                check_R_time, \
438                rk_stage2_step, limit_qnew_step, applyBCqnew, \
439                applydiffusionqnew, \
440                copy]
441
442   </SolverSequence>
443
444 # end simulation definition
445 </ThesisNozzleSimulations>
446
447 # end WARPX input file definition
448 </warpx>
```

A.3 Post Processing

A.3.1 Calculation of Divergence Error

A scheme to calculate $\nabla \cdot \boldsymbol{B}$ is presented below. Begining with the divergence theorem,

$$\iiint_V \nabla \cdot \boldsymbol{B}\, dV = \oiint_S \boldsymbol{B} \cdot d\boldsymbol{S}\,, \tag{A.1}$$

it is desired to develop an algorithm for calculating the divergence of B in each cell with the greatest accuracy permitted by the Discontinuous Glaerkin scheme. Consider a unit cell in cylindrical coordinates (ρ, θ, z), centered at (ρ_0, θ_0, z_0), and with dimensions $\Delta\rho$, $\Delta\theta$, Δz. imposing an axisymmetric condition, so that $\frac{\partial \boldsymbol{B}}{\partial \theta} = \boldsymbol{0}$, and carrying out the trivial integration with respect to θ, A.1 can be writen as

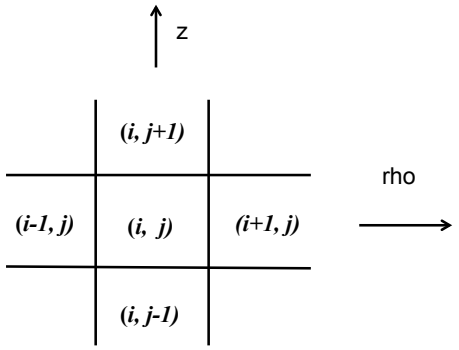


Figure A.1: simple illustration of unitcell, cell $(i,\, j)$ is located at $(\rho_0,\, z_0)$, and has dimensions $(\Delta\rho,\, \Delta z)$

$$
\begin{aligned}
(\nabla \cdot \boldsymbol{B})_{avg}\ \ \rho_0\, \Delta\rho\, \Delta\theta,\, \Delta z \quad &= -\left(\rho_0 - \frac{\Delta\rho}{2}\right) \Delta\theta \int_{z_0-\frac{\Delta z}{2}}^{z_0+\frac{\Delta z}{2}} B_r\big|_{\left(\rho=\rho_0-\frac{\Delta\rho}{2}\right)}\, dz \\
&+ \quad \left(\rho_0 + \frac{\Delta\rho}{2}\right) \Delta\theta \int_{z_0-\frac{\Delta z}{2}}^{z_0+\frac{\Delta z}{2}} B_r\big|_{\left(\rho=\rho_0+\frac{\Delta\rho}{2}\right)}\, dz \\
&- \quad \Delta\theta \int_{\rho_0-\frac{\Delta\rho}{2}}^{\rho_0+\frac{\Delta\rho}{2}} B_z\big|_{\left(z=z_0-\frac{\Delta z}{2}\right)}\, \rho\, d\rho \\
&+ \quad \Delta\theta \int_{\rho_0-\frac{\Delta\rho}{2}}^{\rho_0+\frac{\Delta\rho}{2}} B_z\big|_{\left(z=z_0+\frac{\Delta z}{2}\right)}\, \rho\, d\rho\,.
\end{aligned}
\tag{A.2}
$$

These are the integrals over the two ρ normal and z normal faces respectively. Note that a factor of ρ is retained inside of the integrals over the z-normal faces. These expressions can be simplified by first canceling $\Delta\theta$. In the Discontinuous Glaerkin Scheme, the existence of discontinuities at the cell edge is allowed for. Therefore, an estimated value for the magnetic field at the cell edge must be reconstructed from the left and right handed values. Here the value of the magnetic field at a given edge

is taken to be to be the average of the values calculated at the edge of the two cells neighboring that edge. Using the indexing shown in Figure A.1, the magnetic field terms in equation A.2 become:

$$B_r\big|_{\left(\rho=\rho_0-\frac{\Delta\rho}{2}\right)} = \frac{1}{2}\left[B_r^{(i,j)}\big|_{\left(\rho=\rho_0-\frac{\Delta\rho}{2}\right)} + B_r^{(i-1,j)}\big|_{\left(\rho=\rho_0-\frac{\Delta\rho}{2}\right)}\right] \tag{A.3}$$

$$B_r\big|_{\left(\rho=\rho_0+\frac{\Delta\rho}{2}\right)} = \frac{1}{2}\left[B_r^{(i,j)}\big|_{\left(\rho=\rho_0+\frac{\Delta\rho}{2}\right)} + B_r^{(i+1,j)}\big|_{\left(\rho=\rho_0+\frac{\Delta\rho}{2}\right)}\right] \tag{A.4}$$

$$B_z\big|_{\left(z=z_0-\frac{\Delta z}{2}\right)} = \frac{1}{2}\left[B_z^{(i,j)}\big|_{\left(z=z_0-\frac{\Delta z}{2}\right)} + B_z^{(i,j-1)}\big|_{\left(z=z_0-\frac{\Delta z}{2}\right)}\right] \tag{A.5}$$

$$B_z\big|_{\left(z=z_0+\frac{\Delta z}{2}\right)} = \frac{1}{2}\left[B_z^{(i,j)}\big|_{\left(z=z_0+\frac{\Delta z}{2}\right)} + B_z^{(i,j+1)}\big|_{\left(z=z_0+\frac{\Delta z}{2}\right)}\right] \tag{A.6}$$

In Discontinuous Galerkin schemes, the solution in a given cell is expressed in terms of a set of basis functions, in this case the Legendre polynomials, which are only defined over the interval $[-1,1]$. For each cell, the solution is mapped from the global coordinate system ρ, z to a local coordinate system $u \in [-1,1], v \in [-1,1]$. For second order accurate Discontinuous Galerkin, the solution on a given cell is given by

$$B^{(i,j)} = B_0^{(i,j)} + B_x^{(i,j)}u + B_y^{(i,j)}v + B_{xy}^{(i,j)}uv \tag{A.7}$$

The integral can then be expresed in terms of a modal representation as

$$B_r\big|_{\left(\rho=\rho_0-\frac{\Delta\rho}{2}\right)} = \frac{1}{2}\left[B_r^{(i,j)}\big|_{(u=-1)} + B_r^{(i-1,j)}\big|_{(u=1)}\right] \tag{A.8}$$

$$B_r\big|_{\left(\rho=\rho_0+\frac{\Delta\rho}{2}\right)} = \frac{1}{2}\left[B_r^{(i,j)}\big|_{(u=1)} + B_r^{(i+1,j)}\big|_{(u=-1)}\right] \tag{A.9}$$

$$B_z\big|_{\left(z=z_0-\frac{\Delta z}{2}\right)} = \frac{1}{2}\left[B_z^{(i,j)}\big|_{(v=-1)} + B_z^{(i,j-1)}\big|_{(v=1)}\right] \tag{A.10}$$

$$B_z\big|_{\left(z=z_0+\frac{\Delta z}{2}\right)} = \frac{1}{2}\left[B_z^{(i,j)}\big|_{(v=1)} + B_z^{(i,j+1)}\big|_{(v=-1)}\right]. \tag{A.11}$$

Returning to equation A.2, we can greatly simplify things by making a similar coordinate transformation from ρ, z to u, v (with respect to the cell (i, j). Specifically, let:

$$u = \frac{2}{\Delta\rho}\left(\rho - \rho_0\right) \qquad\qquad v = \frac{2}{\Delta z}\left(z - \Delta z\right)$$

$$du = \frac{2}{\Delta\rho}d\rho \qquad\qquad dv = \frac{2}{\Delta z}dz$$

Then, along with dividing through by the element volume on the right hand side, equation A.2 simplifies to:

$$
\begin{aligned}
(\nabla \cdot \boldsymbol{B})_{avg} \quad &= \quad -\left(\frac{1}{2\Delta\rho} - \frac{1}{4\rho_0}\right) \int_{-1}^{+1} B_r\big|_{\left(\rho = \rho_0 - \frac{\Delta\rho}{2}\right)} \, dv \\
&+ \quad \left(\frac{1}{2\Delta\rho} + \frac{1}{4\rho_0}\right) \int_{-1}^{+1} B_r\big|_{\left(\rho = \rho_0 + \frac{\Delta\rho}{2}\right)} \, dv \\
&- \quad \frac{1}{2\rho_0 \, \Delta z} \int_{-1}^{+1} B_z\big|_{\left(z = z_0 - \frac{\Delta z}{2}\right)} \left(\frac{\Delta\rho}{2} u + \rho_0\right) du \\
&+ \quad \frac{1}{2\rho_0 \, \Delta z} \int_{-1}^{+1} B_z\big|_{\left(z = z_0 + \frac{\Delta z}{2}\right)} \left(\frac{\Delta\rho}{2} u + \rho_0\right) du
\end{aligned}
\tag{A.12}
$$

The once we substitute in our reconstructions of B at the edges, the integrals here are very straightforward to perform, and will not be explicitly shown. However, to help have a sense of whether I am obtaining a reasonable final result, I have compared the final algorithm that we obtain to what we would expect in the limit of smooth, axisymmetric B, which is:

$$
\nabla \cdot \boldsymbol{B} = \frac{1}{\rho} B_r \, + \, \frac{\partial B_r}{\partial r} + \frac{\partial B_z}{\partial z} \tag{A.13}
$$

Indeed, the resulting algorithm I obtain does correspond to this, with one extra term that doesn't have a clear 1-to-1 correspondence. The final algorithm is:

$$
\begin{aligned}
(\nabla \cdot \boldsymbol{B})_{avg} \quad &= \underbrace{\frac{1}{\rho_0}\left[\frac{2B_{r_0}^{(i,j)} + \left(B_{r_0}^{(i-1,j)} + B_{r_x}^{(i-1,j)}\right) + \left(B_{r_0}^{(i+1,j)} - B_{r_x}^{(i+1,j)}\right)}{4}\right]}_{\frac{1}{\rho_0}B_r} \\
&+ \underbrace{\frac{1}{\Delta\rho}\left[\frac{2B_{r_x}^{(i,j)} + \left(B_{r_0}^{(i+1,j)} - B_{r_x}^{(i+1,j)}\right) - \left(B_{r_0}^{(i-1,j)} + B_{r_x}^{(i-1,j)}\right)}{2}\right]}_{\frac{\partial B_r}{\partial \rho}} \\
&+ \underbrace{\frac{1}{\Delta z}\left[\frac{2B_{z_y}^{(i,j)} + \left(B_{z_0}^{(i,j+1)} - B_{z_y}^{(i,j+1)}\right) - \left(B_{z_0}^{(i,j-1)} + B_{z_y}^{(i,j-1)}\right)}{2}\right]}_{\frac{\partial B_z}{\partial z}} \\
&+ \underbrace{\frac{\Delta\rho}{\rho_0 \Delta z}\left[\frac{2B_{z_{xy}}^{(i,j)} + \left(B_{z_x}^{(i,j+1)} - B_{z_{xy}}^{(i,j+1)}\right) - \left(B_{z_x}^{(i,j-1)} + B_{z_{xy}}^{(i,j-1)}\right)}{12}\right]}_{cross-term}
\end{aligned}
\tag{A.14}
$$

Bibliography

[1] George P Sutton and Oscar Biblarz. *Rocket propulsion elements*. John Wiley & Sons, 2010.

[2] Y Arakawa and A Sasoh. Electromagnetic effects in an applied-field magnetoplasmadynamic thruster. *Journal of Propulsion and Power*, 8(1):98–102, 1992.

[3] Gerd Kr-uacute, lle, Monika Auweter-Kurtz, and Akihiro Sasoh. Technology and application aspects of applied field magnetoplasmadynamic propulsion. *Journal of Propulsion and Power*, 14(5):754–763, 1998.

[4] Christine Charles, Rod Boswell, et al. Current-free double-layer formation in a high-density helicon discharge. 2003.

[5] Samuel A Cohen, Xuan Sun, Nathaniel M Ferraro, Earl E Scime, Mahmood Miah, Sy Stange, Nicholas S Siefert, and Robert F Boivin. On collisionless ion and electron populations in the magnetic nozzle experiment (mnx). *IEEE Transactions on Plasma Science*, 34(3):792–803, 2006.

[6] Oleg V Batishchev. Minihelicon plasma thruster. *IEEE Transactions on plasma science*, 37(8):1563–1571, 2009.

[7] Alexey V Arefiev and Boris N Breizman. Theoretical components of the vasimr plasma propulsion concept. *Physics of Plasmas (1994-present)*, 11(5):2942–2949, 2004.

[8] Christopher S Olsen, Maxwell G Ballenger, Mark D Carter, Franklin R Chang Díaz, Matthew Giambusso, Timothy W Glover, Andrew V Ilin, Jared P Squire, Benjamin W Longmier, and Paul A Cloutier11. An experimental study of plasma detachment from a magnetic nozzle in the plume of the vasimr® engine. In *Proceedings of the 33rd International Electric Propulsion Conference*, 2013.

[9] Ronald W Moses Jr, Richard A Gerwin, and Kurt F Schoenberg. Resistive plasma detachment in nozzle based coaxial thrusters. In *Proceedings of the ninth symposium on space nuclear power systems*, volume 246, pages 1293–1303. AIP Publishing, 1992.

[10] Kurt F Schoenberg, Richard A Gerwin, Cris W Barnes, Ivars Henins, Robert Mayo, Ronald Moses Jr, Richard Scarberry, and Glen Wurden. Coaxial plasma thrusters for high specific impulse propulsion. 1991.

[11] Alexey V Arefiev and Boris N Breizman. Magnetohydrodynamic scenario of plasma detachment in a magnetic nozzle. *Physics of Plasmas (1994-present)*, 12(4):043504, 2005.

[12] EB Hooper. Plasma detachment from a magnetic nozzle. *Journal of Propulsion and Power*, 9(5):757–763, 1993.

[13] Jiong Qiu, Haimin Wang, CZ Cheng, and Dale E Gary. Magnetic reconnection and mass acceleration in flare-coronal mass ejection events. *The Astrophysical Journal*, 604(2):900, 2004.

[14] D Gregory Chavers, Chris Dobson, Jonathan Jones, Michael Lee, Adam Martin, Judith Gregory, Jim Cecil, Roger D Bengtson, Boris Breizman, Alexey Arefiev, et al. Status of magnetic nozzle and plasma detachment experiment. In *SPACE TECH. & APPLIC. INT. FORUM-STAIF 2006: 10th Conf Thermophys Applic Microgravity; 23rd Symp Space Nucl Pwr & Propulsion; 4th Conf Human/Robotic Tech & Nat'l Vision for Space Explor.; 4th Symp Space Coloniz.; 3rd Symp on New Frontiers & Future Concepts*, volume 813, pages 465–473. AIP Publishing, 2006.

[15] Uri Shumlak, R Lilly, N Reddell, E Sousa, and B Srinivasan. Advanced physics calculations using a multi-fluid plasma model. *Computer Physics Communications*, 182(9):1767–1770, 2011.

[16] B Srinivasan and U Shumlak. Analytical and computational study of the ideal full two-fluid plasma model and asymptotic approximations for hall-magnetohydrodynamics. *Physics of Plasmas (1994-present)*, 18(9):092113, 2011.

[17] Bhuvana Srinivasan, Ammar Hakim, and Uri Shumlak. Numerical methods for two-fluid dispersive fast mhd phenomena. *Communications in Computational Physics*, 10(01):183–215, 2011.

[18] Bernardo Cockburn and Chi-Wang Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16(3):173–261, 2001.

[19] Wm H Reed and TR Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479*, 1973.

[20] Rupak Biswas, Karen D Devine, and Joseph E Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14(1-3):255–283, 1994.

[21] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[22] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.

[23] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing runge-kutta schemes. *Mathematics of computation of the American Mathematical Society*, 67(221):73–85, 1998.

[24] Johan Peter Goedbloed and Stefaan Poedts. *Principles of magnetohydrodynamics: with applications to laboratory and astrophysical plasmas*. Cambridge university press, 2004.

[25] R. Balescu. *Classical Transport*, volume 1. Elsevier Science Publishing Company, Inc., 1988.

[26] J.U Brackbill and D.C Barnes. The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations. *Journal of Computational Physics*, 35(3):426 – 430, 1980.

[27] Gábor Tóth. The $\nabla \cdot \mathbf{B} = 0$ constraint in shock-capturing magnetohydrodynamics codes. *Journal of Computational Physics*, 161(2):605 – 652, 2000.

[28] A. Dedner, F. Kemm, D. Krner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the mhd equations. *Journal of Computational Physics*, 175(2):645 – 673, 2002.

[29] R Winglee, T Ziemba, L Giersch, J Prager, J Carscadden, and BR Roberson. Simulation and laboratory validation of magnetic nozzle effects for the high power helicon thruster. *Physics of Plasmas (1994-present)*, 14(6):063501, 2007.

[30] Eduardo Ahedo. Parametric analysis of a magnetized cylindrical plasma. *Physics of Plasmas (1994-present)*, 16(11):113503, 2009.