2012-12-07

Advanced Numerical Methods in General Relativistic Magnetohydrodynamics

Michael J. Besselman
Brigham Young University - Provo

Advanced Numerical Methods in General Relativistic Magnetohydrodynamics

Michael Besselman

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Eric W. Hirschmann, Chair
David Neilsen
Jean-François S. Van Huele

Department of Physics and Astronomy

Brigham Young University

December 2012

ABSTRACT

Advanced Numerical Methods in General Relativistic Magnetohydrodynamics

Michael Besselman
Department of Physics and Astronomy, BYU
Master of Science

We show our work to refine the process of evolutions in general relativistic magnetohydrodynamics. We investigate several areas in order to improve the overall accuracy of our results. We test several versions of conversion methodologies between different sets of variables. We compare both single equation and two equations solvers to do the conversion. We find no significant improvement for multiple equation conversion solvers when compared to single equation solvers. We also investigate the construction of initial data and the conversion of coordinate systems between initial data code and evolution code. In addition to the conversion work, we have improved some methodologies to ensure data integrity when moving data from the initial data code to the evolution code. Additionally we add into the system of MHD equations a new field to help control the no monopole constraint. We perform a characteristic decomposition of the system of equations in order to derive the associated boundary condition for this new field. Finally, we implement a WENO (weighted non-oscillatory) system. This is done so we can evolve and track shocks that are generated during an evolution of our GRMHD equations.

Keywords: GRMHD, MHD, characteristic decomposition, neutron stars, initial data

ACKNOWLEDGMENTS

This thesis is dedicated to my family. To Michael and Matheu, while time has taken you away from me you will always be close to my heart. To Benjamin and Lillian, while I only got to hold you for mere moments, you will always be loved. To Samuel and Hazel who make me feel young again. And especially to my wife Reanna who stands by my side and supports me in everything I have wanted to do. I love you all.

Contents

Chapter 1

Introduction

In this thesis we cover several concepts in working with evolving systems of neutron stars. We consider both the gravitational field along with the material that makes up the neutron star. We endeavor to refine our evolution such that we will be able to understand the physics of these systems. We are not looking at the microphysics at this time as we are concerned with the bulk matter.

1.1 Overview

Two years after the first neutron was discovered by James Chadwick in 1932 [1], Baade and Zwicky theorized the existence of neutron stars [2]. The first discovery of a neutron star was not until 1965 when Hewish and Okoye detected a source of high radio brightness in the Crab Nebula [3]. These discoveries have continued and have contributed to some rather fascinating conclusions. Neutron stars have been found to have very large magnetic fields and very short rotational periods. Their density is comparable to the density of an atomic nucleus. In 1974 Taylor and Hulse discovered the first binary pulsar, PSR B1913+16. This binary system contains two neutron stars one of which is a pulsar. They were able to measure the orbital decay of this system and were able to show that the decay was consistent with the theory of general relativity and the emission of gravitational

radiation [4]. For this work they were awarded the 1993 Nobel Prize in Physics.

Nature creates neutron stars at the end of the life of a sufficiently massive main sequence star. As a star ages, the fusible material will be significantly decreased and there will be less fusion in the core. As this happens, the gravitational field on the core will no longer be balanced out by the thermal pressure of the fusion reaction in the core. Eventually the competition between the gravitational force and the gas pressure will be unbalanced to the point the the core will collapse and result in the star going through the process of a nova. The resultant core will collapse into a much smaller object. If the initial star is between 2 and 3 solar masses, the core will collapse to a neutron star. In a neutron star a large portion of the electrons and protons have recombined to form neutrons. Because the core collapse happens very quickly, the core can retain most of its angular momentum and magnetic moment. The result is that neutron stars can have extremely short rotational periods and very large magnetic fields. These are the largest magnetic fields that exist. A study of neutron stars has been done for quite some time and continues to interest scientists. For example, it was believed that the maximum mass for a neutron star was the Chandrasekhar limit of 1.44 solar masses. However, in 2010 Paul Demorest measured a neutron star with a mass of 1.97 ± 0.01 solar masses [5]. There are approximately 2000 known neutron stars in our galaxy.

While we have some information about neutron stars there are still many outstanding questions such as the equation of state, maximum mass and radius, structure and source of the magnetic field, etc. In particular, one may be interested in the structure of their overall magnetic field and the structure of that field internal to the star. Due to our inability to do direct observations, another methodology of discovery needs to be pursued. Because there is already some understanding of the structure and processes of how a neutron star evolves, we can start from that point. We know that they are very massive objects and we need to use an accurate model for the gravitational field. Additionally they are very dense. The material that makes up the neutron star can be modeled as a superfluid that has a large magnetic field. We need to choose appropriate models for the physics

that we know. One can run simulations using the chosen physics models in order to understand the as of yet unanswered questions. When these simulations are run, the results of the simulations can be compared to what is known about neutron stars. If those results match up to what is already known, confidence is gained that the models and evolution are correct. At that point, the results can be investigated to find answers to the unanswered questions. However, if the results do not match our known physics, a refinement of the models that we have used can be done.

We need to choose the models that we wish to use for our physical system. We choose a model of an ideal fluid for the material that makes up the neutron star. Additionally we use general relativity to account for the gravitational fields. Here we give a brief description of some of our chosen models.

1.1.1 Modeling the material

We choose to model the material of the neutron star as high density perfect fluid. With the addition of a magnetic field, we use the model of magnetohydrodynamics or MHD [6]. This has been used extensively in both astrophysics and plasma physics. For this we use the equations of fluid dynamics combined with the equations of electricity and magnetism. To complete the system of equations, we also need an equation of state for the gas. For this we choose the ideal gas law.

Along with the equations of MHD, our usual assumption is that the plasma has infinite conductivity. Physically this means that we assume that the electrons move without resistance such that the electric field vanishes in the comoving frame. This system is known as ideal MHD.

1.1.2 Gravity

Because the density of neutron stars is so large, a Newtonian description of gravity is not accurate for these systems. We need to use instead the description given by general relativity. The equations

for general relativity can be written as

$$G^{ab} = 8\pi T^{ab} \tag{1.1}$$

where G^{ab} is the Einstein tensor that describes how spacetime is curved and T^{ab} is the stress energy tensor that describes the nature of the material in that spacetime. The relation here is that the curvature of our spacetime is directly related to the material in that spacetime. The Einstein tensor is a convenient shorthand for second derivatives of the metric. The metric is the basic mathematical description that gives us the curvature of the spacetime. This equation in effect tells us that the curvature of spacetime depends on the energy and material in that spacetime and that the motion of the material depends on the curvature of that spacetime. To paraphrase relativist John Wheeler, matter tells spacetime how to curve and the curvature of spacetime tells matter how to move.

1.1.3 3+1 decomposition

In the original development of general relativity, time is put on an equal footing with the various spatial dimensions. However, because we wish to understand the Einstein equations as evolution equations, we need to distinguish between the time dimension and spatial dimensions for our spacetime. This necessitates a change in perspective with regard to equation (1.1). The Einstein equations are decomposed with respect to the three dimensional hypersurface made up of the spatial dimensions embedded in the four dimensional spacetime. A vector normal to the spatial hypersurface will necessarily be timelike and is labeled n^a. In the three dimensional hypersurface we refer to the reduced metric appropriate to that hypersurface as h_{ab}. This is distinct from g_{ab} which is the metric of the four dimensional spacetime. Because of the notion of a normal vector to a hypersurface and its associated metric, we understand that the orthogonality relationship will yield $n^a h_{ab} = 0$. This basic form of decomposition of our spacetime is applied to the set of Einstein

and matter equations. In the process we find a set of evolution equations together with constraint equations.

1.2 Converting between sets of variables

When we consider the material that makes up the neutron star, there are a set of variables for that material. We call these the primitive variables and they include quantaties like pressure and density. However, there is another set of variables that is useful that we refer to as the conservative variables. These conservative variables include quantities such as relativistic momentum and energy. In our efforts to solve the equations there are occasions where both sets of variables are needed. Because we use both, it is necessary to convert between them. We endeavor to improve the accuracy of the conversion that is used in our current evolution. This conversion is nonlinear and often times becomes difficult. There are some cases where a solution does exist but the current solver is unable to accurately find that solution.

We implement several different solvers and compare the results. We implement a Monte Carlo type method in an attempt to determine if any solver that we have implemented shows significant improvement over the one that is currently implemented in our evolution code.

1.3 Creating initial data

There are two critical aspects creating a numerical evolution that accurately portrays the evolution of a physical system into the future. In particular, one needs an accurate scheme to evolve the equations into the future. The other component is good initial data. If we have accurate initial data but the evolution scheme is inaccurate, the results will not be a physical system. On the other hand if we have a perfect evolution system but do not have accurate initial data, the results again will be nonphysical. The construction of accurate initial data is paramount for any accurate physical

system.

We describe work we have done to generate initial data that properly solves the constraint equations from the general relativistic magnetohydynamics (GRMHD) equations. We show what is necessary to convert between the initial data in one coordinate system into that coordinate system needed by the evolution code. We additionally give a brief discussion on units and the conversion of units from different systems.

1.4 Boundary conditions and divergence cleaning

As described above the initial data must satisfy the constraint equations from GRMHD. One particular constraint that is present in our evolution comes from the Maxwell equations of electromagnetism known as the no monopole constraint which is written

$$\nabla \cdot \vec{B} = 0. \tag{1.2}$$

The physical interpretation of this equation is that there are no magnetic monopoles. The initial data must satisfy this equation. However, due to computational errors, there are times in the evolution that this constraint may become violated. A major component of any accurate evolution is ensuring that this constraint is handled accurately. If left unchecked this error may cause nonphysical results.

We implement an approach where the constraint is allowed to be slightly violated but we damp out those violations using a new field. To this end we extend our system of equations to include this new field. We start with the eight equations from MHD which include the fluid equations and the Maxwell equations. To these equations we add an equation for the additional field and modify the magnetic field equations in an appropriate way to allow the new field to evolve as a wave. The new field is used to suppress the growth of the constraint violation and propagate the violations to the computational boundaries and off the computational grid. One of the drawbacks of this

implementation is that the boundary conditions must be generalized for this new field. Here, we show our work to build these boundary conditions for our new field.

1.5 Weighted Essentially Non-Oscillatory evolution

The set of equations that we are solving have solutions that produce shocks from smooth initial data. These shocks are found in nature for example when the speed of the solar wind drops due the interaction with the earth's magnetosphere, a shock will be formed. This is known as the earth's bow shock. Because we know that shocks are real we need a system that can evolve them numerically. However, handling shocks in the system can become problematic when solving the equations numerically. Shocks appear in the numerical solutions as discontinuities in the fluid variables. We show our work to construct a system known as WENO that has a high order of accuracy while being able to track and evolve shocks as they get created in the system.

Chapter 2

Solvers

In attempting to better understand neutron stars singly or in binary systems, we need to have a framework in which to model both the material that constitutes the neutron star as well as the strong gravitational fields that it resides in and contributes to. For the gravitational field we use general relativity. For the neutron star matter, we use a perfect fluid. While the latter is not the best model available it is a useful approximation as we are looking at bulk properties of the star and in this work will be less concerned with the microphysics.

We are also interested in the possible effects of an extremely strong magnetic field on the structure and properties of a neutron star. To this end, we consider coupling the electromagnetic field to the fluid. The standard approach is to assume that the fluid has infinite conductivity. This assumption has the result of eliminating any electric field in the comoving frame. The magnetic field is then tied to the fluid elements and evolves with them.

Thus, the model we will work with is general relativistic magnetohydrodynamics or GRMHD. We will use this model extensively throughout the majority of this thesis. In the framework of general relativity, we incorporate the fluid and electromagnetic field using the stress energy tensor.

For our assumed configuration we have

$$T^{ab} = [\rho_o(1+\varepsilon)+P]\,u^a u^b + P g^{ab} + F^{ac}F^{b}{}_{c} - \frac{1}{4} g^{ab} F^{cd} F_{cd} \tag{2.1}$$

where P is the pressure, ε is the internal energy of the fluid, ρ_o is the rest mass density in the comoving frame, g_{ab} is the metric of the spacetime, u^a is the four velocity, and F^{ab} is the Maxwell tensor which is associated with the electric and magnetic fields and is defined as

$$F_{ab} = \partial_a A_b - \partial_b A_a \tag{2.2}$$

where A^b is the gauge potential in four dimensional spacetime.

In the form above, the fluid and electromagnetic fields are coupled only through the curvature. We also want to couple them through a constitutive, or Ohm's law type, relation. In particular, we take

$$J_a + \rho_q u_a = \sigma F_{ab} u^b \tag{2.3}$$

where σ is the conductivity of the fluid, J_a is the current, and ρ_q is the charge density. Note that $F_{ab}u^b$ is the electric field in the comoving frame. If we allow $\sigma \to \infty$, it is equivalent to setting $F_{ab}u^b = 0$, which is the statement that the electric field in the comoving frame vanishes. This additional assumption forms the ideal MHD condition.

The equations in the above form cannot be utilized for evolutions. To do an evolution requires separating out the time component from the equations. The approach to do this is to use the ADM 3+1 formulation. For this, we first consider a spatial hypersurface that is embedded in our spacetime. We evolve the hypersurface through time in order to perform our evolution. The hypersurface will necessarily have a metric and associated curvature. We also construct a normal vector to the hypersurface. Because the hypersurface is purely spatial, the normal vector will necessarily be timelike. We project the equations into the hypersurface and normal to it by using the embedded metric and normal vector. This will generate a set of evolution equations for the

conservative variables. The equations of motion in covariant form that result from this are

$$\partial_t\left(\sqrt{h}D\right)+\partial_i\left[\sqrt{-g}D\left(v^i-\frac{\beta^i}{\alpha}\right)\right] = 0 \tag{2.4}$$

$$\partial_t\left(\sqrt{h}E\right)+\partial_i\left[\sqrt{-g}\left(S^i-\frac{\beta^i}{\alpha}E\right)\right] = \sqrt{-g}\left[(\perp T)^{ab}K_{ab}-\frac{1}{\alpha}S^a\partial_a\alpha\right] \tag{2.5}$$

$$\begin{aligned}\partial_t\left(\sqrt{h}S_b\right)+\partial_i\left[\sqrt{-g}\left((\perp T)^i{}_b-\frac{\beta^i}{\alpha}S_b\right)\right] &= \sqrt{-g}\left[{}^3\Gamma^i_{ab}(\perp T)^a{}_i+\frac{1}{\alpha}S_a\partial_b\beta^a\right]\\ &\quad -\sqrt{-g}\left[\frac{1}{\alpha}\partial_b\alpha E\right]\end{aligned} \tag{2.6}$$

$$\begin{aligned}&\partial_t\left(\sqrt{h}B^b\right)+\partial_i\left[\sqrt{-g}\left(B^b\left(v^i-\frac{\beta^i}{\alpha}\right)\right)\right]\\ &+\partial_i\left[\sqrt{-g}\left(-B^i\left(v^b-\frac{\beta^b}{\alpha}\right)+\phi h^{ib}-\eta\beta^b\sqrt{h}B^i\right)\right] &= (1-\eta)\beta^b\partial_i\left(\sqrt{h}B^i\right)-\eta\sqrt{h}B^i\partial_i\beta^b\\ &\quad +\sqrt{-g}\left[\phi h^{ib}\partial_i ln\alpha-{}^3\Gamma^b_{ij}h^{ij}\phi\right]\end{aligned} \tag{2.7}$$

$$\partial_t\left(\sqrt{h}\phi\right)+\partial_i\left[\sqrt{-g}\left(B^i-\frac{\beta^i}{\alpha}\phi\right)\right] = \sqrt{-g}\left[B^i\partial_i ln\alpha-K\phi-s\phi\right] \tag{2.8}$$

where $D=W\rho_o$ is the relativistic energy density, $E=T_{ab}n^an^b$ is the energy density of matter as measured by an observer whose velocity is given by n^a, $S^i=-T_{ab}n^ah_i{}^b$ is the relativistic momentum, $(\perp T)_{cd}=T_{ab}h_c{}^ah_d{}^b$ is the projected stress energy tensor in the hypersurface, v^i is the fluid velocity, K^{ab} is the extrinsic curvature, K is the trace of the extrinsic curvature, ${}^3\Gamma^i_{jk}$ is the three dimensional Christoffel on the hypersurface, β^i is the shift vector, and α is the lapse. The field ϕ is an auxiliary parameter that is used to enforce the no monopole constraint. Its inclusion will be discussed in a later chapter.

To understand the significance of the conservative forms for partial differential equations, take the advection equation. In one dimension the advection equation describes a left or right moving wave that does not change shape. The advection equation for ψ is

$$\partial_t\psi+c\cdot\partial_x\psi=0 \tag{2.9}$$

where we will assume the velocity c is constant. The significance of the conservative form can be

seen by integrating this equation in space

$$\int_b^a \partial_t \psi dx = -\int_b^a c \cdot \partial_x \psi dx \tag{2.10}$$

The right hand side can be clearly evaluated to the values of ψ at the boundaries. Since we assume that at infinity the wave form $\psi \to 0$, if we allow $a \to \infty$ and $b \to -\infty$, the right hand side will vanish. Changing the order on the left side gives

$$\partial_t \int_\infty^\infty \psi dx = 0. \tag{2.11}$$

Clearly the integral on the left is constant in time. This is an important feature of a conserved quantity. Conservative equations in higher dimensions will demonstrate an analogous property. This conservation property carries over to more complicated equations that are no longer identical to this form. However, the numerical techniques have been shown to remain accurate and robust. For this reason we have chosen to express our GRMHD equations in conservative form.

The time derivative of our quantity ψ is determined wholly by the flux though any region of interest. If that region is all space, the flux is zero and we have a conserved quantity. If we are using a smaller region, we can track the change of a field directly from the fluxes into and out of the region. This property can be exploited in the construction of numerical techniques to solve equations much more complicated than the advection equation but have a similar form.

Conservative schemes can be applied to the equations of GRMHD. These equation are not in the strict conservative form due to the presence of source terms. The generalization of this method for use with source terms is often referred to as balance law. Our methodology is to consider the source terms separately from the flux terms. We will calculate the flux terms and the source terms separately for each of the conserved variables. We then combine them together to update the conserved variables for the next time step. This form has been shown to be useful even in situations where the balanced law form does not hold. The numerical treatment of the equations is otherwise exactly the same.

For the set of MHD equations in curved space there are necessarily source terms. On inspection they arise because of the coordinate system and the existence of the curvature of the spacetime. In our case using these methods in this form suggests the use of the variables that appear in the time derivatives in these equations that are the conservative variables, $\{D, S^i, E, B^i\}$. However, the variables that appear in the stress energy tensor are the basic fluid variables. We call these the primitive variables. To solve the equations means we need to solve for both the conservative and primitive variables. To calculate the primitive variables, we need to first obtain the conservative variables using our evolution equations. We then convert the conservative variables into primitive variables. This conversion to primitive variables is highly nontrivial. While algebraic, it has been found that the effort to construct a robust MHD code relies on a good implementation of this transformation [7]. The conservative variables are easily written in terms of the primitive variables. However, the inversion is not expressible in closed form. There have been a variety of approaches to this problem.

2.1 The transformation to primitive variables

The evolution equations are written in terms of the conservative variables. Therefore, at each step in the evolution, the conservative variables are updated in time. Obtaining the primitive variables as stated requires an inversion from the conservative variables. We can write down the equations to construct the conservative variables from the primitive ones. These conservative variables are defined as [8]

$$D = W\rho_o \tag{2.12}$$

$$S_i = \left[h_e W^2 + B^2\right] v_i - (B\cdot v) B_i \tag{2.13}$$

$$\tau = h_e W^2 + B^2 - \frac{\Gamma - 1}{\Gamma}\left[h_e - \frac{D}{W}\right] - \frac{1}{2}\left[(B\cdot v)^2 + \frac{B^2}{W^2}\right] - D \tag{2.14}$$

where B^i is the magnetic field as measured in the hypersurface, v_i is the fluid coordinate velocity, $B \cdot v = B^i v_i$, $W = (1 - v^i v_i)^{-\frac{1}{2}}$ is the Lorentz factor. We note that Γ is the polytropic index, and $h_e = \rho_o(1+\varepsilon) + P$ is the enthalpy.

As can be seen, the definition of the conservative variables in terms of the primitive variables is analytic. If we could write the inverse relation for the primitives in terms of the conservatives we would be done. However, the equations do not admit an obvious inversion. From a numerical standpoint we should be able to write a solver for the primitives in terms of the conservatives.

The primitive variables that we would like to solve for are ρ_o, v_i and P. Clearly, we need to solve a multidimensional root finding problem. We use a standard approach of a multidimensional Newton solver. This have been implemented in a variety of codes and as mentioned serve as a weak point in many such relativistic MHD codes. It is possible to reduce the inversion to finding the roots of a single function. Such a one dimensional Newton solver has been implemented in the current GRMHD code.

It has been seen that there are at least two cases in which the solvers fail. The first mode is when the evolution equations return a set of conservative variables that is nonphysical. There is no hope in resolving this problem with any modification of the inversion code. The other failure mode is where the evolution does return a physical state of the conservative variables but the solvers are not able to do the inversion to the primitive variables. One of the examples of this failure is in the atmosphere surrounding the star. This is an area in which the fluid density is very small and the magnetic field density is very high. We endeavor to improve the accuracy of the inversion where we have the condition of $\frac{B^2}{\rho_o} \gg 1$. An improvement in the accuracy in this regime is likely to improve the overall accuracy of the code.

In order to simplify the construction of the inversion we choose to rescale by the size of the

magnetic field B^2

$$\bar{D} = \frac{D}{B^2} \tag{2.15}$$

$$\bar{S}_i = \frac{S_i}{B^2} \tag{2.16}$$

$$\bar{\tau} = \frac{\tau}{B^2} \tag{2.17}$$

$$\bar{h}_e = \frac{h_e}{B^2}. \tag{2.18}$$

Additionally, we make the following definitions

$$x = \bar{h}_e W^2 \tag{2.19}$$

$$y = \frac{Bv}{|B|} \tag{2.20}$$

$$z = \bar{S}^j v_j \tag{2.21}$$

$$\bar{S}_\parallel = \frac{B_j \bar{S}^j}{|B|} \tag{2.22}$$

$$\bar{S}_\perp^2 = \bar{S}^2 - \bar{S}_\parallel^2. \tag{2.23}$$

With these redefinitions, the above five inversion equations can be reduced to a single equation for x, and the conservative variables. The current code employs just such a single equation solver for x.

$$\begin{aligned} \bar{\tau} + \bar{D} = {} & x\left\{\frac{1}{\Gamma} + \frac{\Gamma-1}{\Gamma}\left(\frac{\bar{S}_\perp^2}{(x+1)^2} + \frac{\bar{S}_\parallel^2}{x^2}\right)\right\} \\ & + \frac{1}{2}\left\{1 + \frac{\bar{S}_\perp^2}{(x+1)^2}\right\} + \frac{\Gamma-1}{\Gamma}\bar{D}\left\{1 - \frac{\bar{S}_\perp^2}{(x+1)^2} - \frac{\bar{S}_\parallel^2}{x^2}\right\}^{1/2}. \end{aligned} \tag{2.24}$$

This has been used to great success in our current code. However, in the atmosphere region of the star we can have $\frac{B^2}{\rho_o} \gg 1$ and this solver will not work or return a physical solution. This is not the only way to solve for the primitive variables. One could construct other solvers to do this inversion.

Our approach has been to reduce the number of equations to two [7]. This adds additional problems from running a multidimensional Newton Raphson solver. While trying to solve a two dimensional root finding problem is more difficult, the benefit is avoiding the necessity of maintaining positiveness of the radical in the equations. We would like to eliminate the radical from the equations to eliminate those associated problems. In the reduction of the original equations the radical arises from the elimination of W from the equations. We therefore do not eliminate W. This leads us to three different sets of equations for three different pair of variables.

The resulting equations in terms of W and x are

$$0 = \frac{1}{W^2} + \frac{\bar{S}_\perp^2}{(x+1)^2} + \frac{\bar{S}_\parallel^2}{x^2} - 1 \tag{2.25}$$

$$0 = \frac{1}{W^2}\left\{\frac{1}{2} + \frac{\Gamma-1}{\Gamma}x\right\} + \bar{\tau} + \bar{D} + \frac{\bar{S}_\parallel^2}{2x^2} - \left\{\frac{\Gamma-1}{\Gamma}\frac{\bar{D}}{W} + x + 1\right\}. \tag{2.26}$$

In terms of W and y, the two equations are

$$0 = \frac{1}{W^2} + \frac{y^2\bar{S}_\perp^2}{(\bar{S}_\parallel + y)^2} + y^2 - 1 \tag{2.27}$$

$$0 = \frac{1}{W^2}\left\{\frac{1}{2} + \frac{\Gamma-1}{\Gamma}\frac{\bar{S}_\parallel}{y}\right\} + \bar{\tau} + \bar{D} + \frac{y^2}{2} - \left\{\frac{\Gamma-1}{\Gamma}\frac{\bar{D}}{W} + \frac{\bar{S}_\parallel}{y} + 1\right\}. \tag{2.28}$$

In terms of W and z, the two equations are

$$0 = (A+C)\left(zA - \bar{S}_\parallel^2 C\right) - \bar{S}_\perp^2 AC \tag{2.29}$$

$$0 = A^2(A+C)\left[1 - \frac{1}{W^2}\right] - C\left[zA^2 + \bar{S}_\parallel^2 C^2\right] \tag{2.30}$$

with the following definitions

$$A = 2(\bar{\tau} + \bar{D}) - \left(z + 1 + 2\frac{\Gamma-1}{\Gamma}\frac{\bar{D}}{W}\right) \tag{2.31}$$

$$C = 1 + \frac{2-\Gamma}{\Gamma}\frac{1}{W^2}. \tag{2.32}$$

There are a few disadvantages of using two equation solvers. First, the process of a two equation solver takes more time computationally than a single equation solver. Second, finding a good

initial guess becomes more difficult. This is because for a single equation solver we can find upper and lower limits for the solver and check to see if the solver goes outside of those bounds. We do not have that ability with more than one equation.

2.2 Testing two equation solvers

In an effort to determine whether a different solution approach to finding primitive variables is beneficial, we want to test these three different two equation solvers and their relative robustness in converting from the conservative variables into the primitive variables. All three of the two equation solvers were coded separately and compiled along with the single equation solver that is currently being used in our evolution code. We set up a standard two dimensional Newton Raphson solver for the above three sets of equations. In order to compare the solvers we need to check the accuracy of each.

We want to construct tests of the solvers independent of the remainder of the evolution code. However, we need to use the evolution code to determine where the current solver fails. We ran the evolution code and collected data as to a variable range where the current solver failed to do an inversion but one existed. We used Maple to very high precision to perform the inversion in order to verify that the data points are indeed a physical solution to the inversion. This allowed us to generate a range of primitive variables where the current solver did not always return an accurate solution. Additionally we took data points where the inversion was working well. This gave us working and nonworking regions of primitive variables where we could test and compare each of the solvers.

With a specific range of primitive variables, we could not write an exhaustive test for all values in the range. For our test we took a Monte Carlo style approach. First we would select a range of variables. For each primitive variable f we can determine from our range and minimum and

maximum value f_{min} and f_{max}. We want to randomly select a value inside those limits. We obtain a random number R between 0 and 1 for each variable. We use this number to obtain a value inside our range in the following way:

If we want to get a linear distribution of values of the variable we use the function:

$$f = R \cdot (f_{max} - f_{min}) + f_{min} \tag{2.33}$$

If however, the failures happen more towards the minimum value we use the function:

$$f = R^2 \cdot (f_{max} - f_{min}) + f_{min} \tag{2.34}$$

Additionally we can push the distribution towards the maximum value with the function:

$$f = (1 - R^2) \cdot (f_{max} - f_{min}) + f_{min} \tag{2.35}$$

We want to construct a test such that we can calculate the accuracy of each of the solvers. In order to do this, we take our set of primitive variables f and construct the conservative variables. We then test each solver by trying to invert the calculated conservative variables back into primitive variables and comparing them to the original values. In this way we know the solution exactly and can compare the solver's ability to solve for the primitive. This gives us a fair comparison of the individual solvers. Because we are using a range of primitive variables and not a single failure point we are testing the solvers more fully. We can determine the error of each variable with

$$f_{error} = \frac{f_{actual\ value} - f_{calculated\ value}}{f_{actual\ value}} \tag{2.36}$$

By taking the sum of the squares of the individual errors we get a measure of the error for each data point. We took several thousand data points to determine the ability of the individual solvers. Additionally, we were able to count the number of events where the solvers were not able to return a valid solution. This was a simple count of failures compared with the number of data points tested in a given region.

2.3 Results

The majority of the testing was done where the single equation solver failed. However, we still did a small portion of testing in regions in which the single equation solver worked and produced no failures. The reason is that we want to test the solvers in places where the system was working so we could compare the performance of new solvers in these known physical states. Even though the two equation solvers may take more time to execute, it may be worth the trade off if the results are more accurate. We found that in these areas there was no significant difference in the accuracy of the two equation solvers over the currently implemented one equations solver. This leads us to the conclusion that using a two equation solver would in fact be a detriment to the code due to the increased time to do the computation.

The results for the regions where the one equation solver failed were not good either. The failure rate was just as frequent and sporadic for the new solvers as the one currently used in the evolution code. The original solver failed no more often than the new solvers.

Since the one equation solver is faster and there is not increased accuracy in the results of the other solvers, it is unwise at this time to replace the current solver in the code.

2.4 Further work

The overall conclusion for this chapter is that this set of equations is very difficult to solve. These sets of equations are basically equivalent in their failure rates. However, we are able to get solutions in these regions using Maple with the conservative variables where the current solver fails. This indicates that the higher precision of Maple could be implemented in cases where the implemented solver fails.

Chapter 3

Initial Data

3.1 Introduction

In this thesis we are particularly interested in addressing several aspects relating to the physics and development of simulations of neutron star systems. In this chapter our focus is on the question of initial data. Even if one could build the perfect evolution system for gravitating magnetized fluids for which there would be no error, our evolution would still be insufficient without good initial data. Another way to say this is that any simulation of any physical system, has two basic components, the construction of initial data and the evolution of that data into the future. Within the defining equations for a physical system there are both evolution equations and constraint equations that are not themselves evolution equations. The constraints provide relations in the physical system that must be satisfied for the initial data and for all time. If at some point, the constraints are no longer satisfied, we no longer have a physical evolution. A simple example of this comes from considering only electromagnetism. One can consider evolving some configuration of charges and currents together with the resultant electromagnetic fields into the future. If the initial configuration of charges and currents do not satisfy the no monopole constraint and Gauss's law, then no

evolution of such initial data will represent a physical system. Further if the initial configuration does satisfy the constraints but the evolution does not preserve those constraints into the future, then the evolution is no longer a physical solution. Fortunately in electromagnetism, at the analytic level, we are guaranteed that if the initial data satisfies the constraints at the initial time, the evolution equations will develop the system that continues to satisfy the constraints at all future time. A similar idea must hold for general relativity and our GRMHD system. Thus one of our first requirements is to construct initial data which satisfies our constraints for general relativity and electromagnetism which is properly gravitating and has a magnetic field which satisfies the no monopole constraint.

Solving for initial data in general relativity is a non-trivial problem. We set before us solving a single rotating magnetized equilibrium solution to the equations of GRMHD. Finding equilibrium solutions for rotating non-magnetized stars in general relativity has a long history going back about 40 years. For strongly magnetized initial data the literature is much more limited. While we are not interested in the history, we can take as a starting point work that was done in the late 1980s. [9] This approach took the constraint equations and solved them using a Green's function approach. The resulting integral equations and iterative method allowed various groups to solve for rotating equilibrium conditions. More recently poloidal and toriodal magnetic fields have been added [10, 11]. However, with the exception of some perturbation methods, there is a complete absence when it comes to combined poloidal and toriodal fields. In particular the Meudon group has produce a publicly available routine [12] which has allowed for the construction of magnetized rotating neutron star initial data with poloidal fields. These have been used extensively in evolution of magnetized neutron stars. [13, 14] Recently an effort has been made to construct magnetized neutron stars possessing both poloidal and toroidal magnetic fields. The effort we detail here describes the transformations that this initial data requires to be evolved with our evolution code.

We need to be able to construct equilibrium configurations that satisfy the constraints. Here we

briefly describe the construction of initial data for a neutron star. We begin with two assumptions. First we assume that the star is axisymmetric. This is equivalent to assuming a Killing vector in the ϕ direction. Second, we assume that the star is stationary. This gives us a timelike Killing vector.

We use these Killing vectors and apply them to the equations of GRMHD. Under these symmetry assumptions our GRMHD equations reduce to three sets of equations. One is set of second order elliptic equations for various metric coefficients and the gauge potential of electromagnetism. Another set of equations include the equations of hydrostatic equilibrium which in effect is the equation that defines the structure of the star. The third set of equations include a number of first order equations that define some of the off diagonal terms in the metric and the fluid and various hydrodynamic relations between fluid equations of the star. On solving all of these equations, we obtain the star, its magnetic properties, and the gravitational field that it creates. Because of the assumed symmetries, our initial data is axisymmetric and stationary. As a result the rotational axis and the magnetic axis must be aligned. This is a noted limitation in this approach. In the future a possible direction for further work would be to develop a system to misalign these.

3.2 TOV as an example

Construction of initial data is illustrated by the construction of a TOV (or Tolman Oppenheimer Volkoff) star. This is a star that is assumed to be made of neutron degenerate matter. [15] A TOV star is one that is static and spherically symmetric. For our code this is also the first step in the construction of any initial data which may generalize a TOV star. For a star with angular momentum and/or magnetic fields, the construction of the TOV star is followed by adding the additional physics into the calculations. Clearly one must solve the relevant equations. For a TOV star the problem reduces to solving a set of ordinary differential equations for the metric and the fluid quantities. To solve this system there are physical quantities which must be given at the

outset. These include possibly the central energy density, the radius, central pressure, and the mass of the star. Indeed of these four pieces of initial data one can choose one freely and the others can be calculated during the construction of the star.

We implement this by choosing the central energy density for the star and calculating all other properties. Upon solving the TOV star, we obtain the metric components and fluid properties of the star for variables such as enthalpy and energy density.

While the equations for equilibrium become more complicated as our star begins to rotate or is magnetized, we have free parameters such as magnetic field strength and angular momentum that allow us to construct families of equilibrium configurations [16].

3.3 Transformation of initial data into BSSN evolution data

The code for constructing the initial data and evolving the initial data are separate codes. Indeed, in the current form the initial data produced cannot be evolved without modification. As an example of this, the initial data is in two dimensions while the evolution code requires three dimensional initial data. In addition, the calculated initial data is found on a compactified grid so that the solution for the star is known throughout the entire space. The other difference is that the initial data solver assumes equatorial plan symmetry. As a result, a significant transformation must be performed on the two dimensional data to obtain three dimensional data. The coordinate system for the initial data is $\{s, \mu\}$ where s is a compactified radial coordinate that ranges from 0 to 1 and $\mu = \cos(\theta)$ where θ is the usual spherical coordinate. Because we use equatorial plane symmetry, μ ranges from 0 to 1 as the zenith angle ranges from $\pi/2$ to 0. We need to transform the two dimensional compactified coordinate system to the three dimensional coordinate system that defines the grid in our evolution code. Additionally we need to convert between the form of the metric that is used to calculate initial data to the metric form considered in the evolution code.

Specifically our code uses the BSSN formulation for the evolution equations [17]. Lastly, in the calculation of initial data we solve for the vector potential. This needs to be converted into the components of the magnetic field for the evolution code.

The conversion of variables from the initial data (or MRNS) code to the BSSN evolution code is a simple transformation between coordinate systems. To do this we describe the coordinate systems of the two different codes.

The MRNS code solves the equation on a rectangular grid using coordinates $\{s,\mu\}$ where $s = \frac{r}{r+r_e}$ (r_e being the coordinate radius of the star) is the compactified radial coordinate, and $\mu = \cos(\theta)$. We can then use the symmetries of the system to get the initial data for the entire star in space. This is done in the following manner. From the initial data, we can reflect across $\mu = 0$ to obtain the southern hemisphere of the star. We then rotate about the $\mu = 1$ axis to obtain the entire space.

The BSSN code requires data on a Cartesian grid. Here we use standard $\{x,y,z\}$ coordinates. In order to connect to the above coordinate system, we define $r = \sqrt{x^2+y^2+z^2}$ and $\cos(\theta) = z/r$. The connection between r and s is $r = r_e \frac{s}{(1-s)}$, where r_e is the coordinate radius of the star.

To make the transformations of fields we do an interpolation. We find the r and μ values associated with each point $\{x,y,z\}$ on our evolution grid. We do an interpolation of the initial data to obtain the needed value for the evolution. For scalars like α (the lapse) all we need to do is an interpolation. However, for vectors or higher rank tensors, we need to do a transformation between the coordinate systems. This conversion is relatively simple and is done by the usual transformation law given by

$$A'^i = \frac{\partial x'^i}{\partial x^j} A^j \tag{3.1}$$

with higher rank tensors done in an analogous way.

Because we can write down the conversion between $\{x,y,z\}$ and $\{s,\mu\}$, the derivatives are easy to calculate. We do an interpolation for the value A^j and apply the transformation as described.

There is a subtlety that arises from the above where we have a coordinate singularity in the coordinate system conversion. The derivative of r with respect to x can be written

$$\frac{\partial r}{\partial x} = \frac{x}{r}. \tag{3.2}$$

as $r \to 0$, we naively expect equation (3.2) to blow up. This is an artifact of our coordinate system. This is difficult to enforce numerically. To address this singularity in the coordinate system, we simply use the values at the origin from the initial data routine.

3.3.1 An example

Consider that we want to compute B_z at the center of the star. With the coordinate transformation in the above form we write out the calculation as

$$\begin{aligned} B_z &= \frac{\partial r}{\partial z}B_r + \frac{\partial \theta}{\partial z}B_\theta & (3.3)\\ &= \frac{\partial r}{\partial z}\frac{\partial s}{\partial r}B_s + \frac{\partial \theta}{\partial z}\frac{\partial \mu}{\partial \theta}B_\mu. & (3.4)\end{aligned}$$

Using our previously defined coordinates we find the partial derivatives

$$\begin{aligned} \frac{\partial s}{\partial r} &= \frac{r_e}{(r+r_e)^2} & (3.5)\\ \frac{\partial r}{\partial z} &= \frac{z}{r} = \cos\ \theta & (3.6)\\ \frac{\partial \mu}{\partial \theta} &= -\sin\ \theta = -\sqrt{1-\mu^2} & (3.7)\\ \frac{\partial \theta}{\partial z} &= \frac{-\sin\ \theta}{r} = \frac{-\sqrt{1-\mu^2}}{r}. & (3.8)\end{aligned}$$

We now need to find B_s and B_μ. The components for the magnetic field are calculated from the vector potential. For this formulation we consider the two dimensional metric σ^{bc} which is conformally flat and express it in terms of the usual two dimension polar coordinates. The two dimensional poloidal field components can be defined in terms of the gauge potential as

$$B_a = \frac{1}{r\sin\theta e^{(\gamma-\rho)/2}}\varepsilon_{ab}\sigma^{bc}\Delta_c\left(r^2\sin^2\theta e^{(\gamma-\rho)}A\right) \tag{3.9}$$

where γ and ρ are metric components of the full metric, A is a combination of the four dimensional vector potential, $\varepsilon^{ab} = \sqrt{\det|\sigma_{ab}|}[a\ b]$ is the two dimensional Levi-Civita symbol and $[a\ b]$ is the alternating symbol. With these substitutions we get

$$B_r = e^{(\gamma-\rho)/2}\left(\sin^2\theta A(\gamma-\rho)_{,\mu} + \sin^2\theta A_{,\mu} - 2\mu A\right) \tag{3.10}$$

$$B_z = e^{(\gamma-\rho)/2} r_e s^2 \sin\theta \left(\tfrac{2}{s(1-s)}A + A(\gamma-\rho)_{,s} + A_{,s}\right) \tag{3.11}$$

Combining these equations and transforming to our Cartesian grid we get

$$B_z = -2e^{(\gamma-\rho)/2}A - r\sin\theta\left[\cos\phi\,\partial_x + \sin\phi\,\partial_y\right]\left(Ae^{(\gamma-\rho)/2}\right). \tag{3.12}$$

As $r \to 0$ we get:

$$B_z(r=0) = -2e^{(\gamma-\rho)/2}A\Big|_{r=0} \tag{3.13}$$

Now all we need to do is the interpolations in the initial data for each point in our evolution grid and we can set the magnetic field in that space.

3.4 Units

There is an additional difference in the codes that we need to take into consideration. The units are different in each code. We address this difference in units here.

In general relativity we often use what is referred to as geometric (or geometrized) units. This involves the simple setting

$$c = G = 1 \tag{3.14}$$

As a result of this choice all of the physical quantities in GR have units of length to some power. There are additional units introduced by the presence of the fluid. To make this point clear, let us consider one of the fluid equations that we are solving. In the initial data solver we use the

polytropic relation for the rotating magnetized neutron star. This is a relation between pressure P and rest mass density ρ_o which takes the form

$$P = K \cdot \rho_o^{\Gamma}, \tag{3.15}$$

where K is the polytropic constant and Γ is known as the polytropic index and ranges between 1 and 2. Clearly the polytropic constant K has units and those units depend on the chosen value for Γ. Because our usual choice is $\Gamma = 2$, the following is done with that assignment. If a different value is selected, these computations need to be redone. Because we are looking for the units and value of the constant K we rewrite the equation as

$$K = \frac{P}{\rho_o^2} \tag{3.16}$$

Using the fact that pressure is force per unit area and density is mass per unit volume one can easily derive that the units for K are:

$$[K] = M^{-1}L^5T^{-2}, \tag{3.17}$$

where the square brackets represent the units of the enclosed value, M is mass, L is length, and T is time.

In geometric units both L and T are in linear units of M. From this, one can see that in geometric units we get

$$[K] = M^2. \tag{3.18}$$

Because the equations for GRMHD are all invariant under changes in the length scale, this allows us to use K to set the length scale in the system. For ease in the initial data code, we set the value $K = 1$. We now have a completely dimensionless system.

But what happens when we want to know the results in some well known unit system? The reality is that all of the numerical values have units of $K^x c^y G^z$. The trick is to find the values of

x, y, and z. We can then use the values of K, c, and G in our favorite unit system to do the unit conversion. For the transformation from X to $\hat{X}$, the equation takes the form:

$$\hat{X} = K^x c^y G^z X \tag{3.19}$$

In order to get the values of x, y, and z, one would compare the units of the desired quantity that has dimensions $L^l M^m T^t$. Because we know the units of K, c, and G, we can make the comparison in the following way [7]

$$\begin{aligned}
[\hat{X}] &= [K]^x [c]^y [G]^z && (3.20)\\
&= \left(M^{-1}L^5T^{-2}\right)^x \cdot \left(L^1T^{-1}\right)^y \cdot \left(L^3M^{-1}T^{-2}\right)^z && (3.21)\\
&= M^{x(-1)-z} L^{5x+y+3z} T^{-2x-y-2z} && (3.22)\\
&= M^m L^l T^t. && (3.23)
\end{aligned}$$

From the last two lines it can be shown that:

$$x = \tfrac{l+m+t}{2} \tag{3.24}$$

$$y = 2m - t \tag{3.25}$$

$$z = -\tfrac{l+3m+t}{2}. \tag{3.26}$$

However, we do not know the value of K. Indeed the choice of K is arbitrary. As mentioned, in the initial data code K is set to 1, We can use other considerations to find K. We choose to set $\hat{M}$ to the maximum possible mass of a neutron star. While there is some evidence that the maximum mass of a neutron star may be larger we use the limit of $\hat{M} = 1.44 M_{solar}$. This is known as the Chandrasekhar limit. This is chosen because we are using the TOV model for the neutron star. For a TOV neutron star this is the maximum mass before collapse into a black hole. In the units where $K = 1$ there is a known relation between the central energy density and the mass of a neutron star. This sets the maximum mass of the neutron star in those limits at $M = 0.164$. We write our

transformation equation for mass as

$$\hat{M} = K^x c^y G^z M. \tag{3.27}$$

From our previous discussion, we calculate the values $x = 1/2$, $y = 2$, and $z = -3/2$. We can then solve for K to get $K = 1.060 \cdot 10^5 \frac{cm^5}{g \cdot s^2}$.

Because in our evolution it is important to know time intervals we can ask about the approximate units for time. Let's look at the actual value for the conversion of time. We first calculate the exponents to be $x = 1/2$, $y = -1$, and $z = -1/2$. In this case we can calculate

$$\hat{t} = K^{1/2} c^{-1} G^{-1/2} \tag{3.28}$$

using

$$G = 6.673 \cdot 10^{-8} \frac{cm^3}{g \cdot s^2} \tag{3.29}$$

$$c = 2.99792 \cdot 10^{10} \frac{cm}{s} \tag{3.30}$$

we get

$$\hat{t} = 4.205 \cdot 10^{-5} s \tag{3.31}$$

For every elapsed discrete time interval in the computation of the system we calculate approximately 42μs in coordinate time.

For the magnetic field we have an additional unit that must be dealt with. The units of magnetic induction are in Gauss (or Tesla).

$$10^4\,\mathrm{G} = 1\,\mathrm{T} = 1\,\frac{\mathrm{V \cdot s}}{\mathrm{m^2}} = 1\,\frac{\mathrm{N}}{\mathrm{A \cdot m}} = 1\,\frac{\mathrm{Wb}}{\mathrm{m^2}} = 1\,\frac{\mathrm{kg}}{\mathrm{C \cdot s}} = 1\,\frac{\mathrm{kg}}{\mathrm{A \cdot s^2}} = 1\,\frac{\mathrm{N \cdot s}}{\mathrm{C \cdot m}} \tag{3.32}$$

We will use the first and sixth terms here to do the unit conversion:

$$10^4\,\mathrm{G} = 1\,\frac{\mathrm{kg}}{\mathrm{C \cdot s}} \tag{3.33}$$

This leads us to the question of what a coulomb is in geometric units. To find this we change to statcoulombs. The statcoulomb, which we will label C_s, has units of $\mathrm{g}^{1/2}\mathrm{cm}^{3/2}\mathrm{s}^{-1}$ in cgs. The conversion is

$$1\,\mathrm{C_s} = 3.3356 \cdot 10^{-10}\mathrm{C} \tag{3.34}$$

or

$$1\,\mathrm{C} = 2.99796 \cdot 10^{9}\,\mathrm{g}^{1/2}\mathrm{cm}^{3/2}\mathrm{s}^{-1} \tag{3.35}$$

Using the above equations we can now write

$$1\,\mathrm{G} = 3.3356 \cdot 10^{-11}\,\mathrm{g}^{1/2}\mathrm{cm}^{-3/2} \tag{3.36}$$

We want to convert the value of the magnetic induction between our unitless initial data code system to cgs. We use the same approach as above to solve for $\hat{B}$ from

$$\hat{B} = K^x c^y G^z \cdot B \tag{3.37}$$

with

$$x = -1/2,\ y = 1,\ z = 0 \tag{3.38}$$

Inserting the values of relevant constants we get

$$\hat{B} = 9.2 \cdot 10^{7} \frac{\mathrm{g}^{1/2}}{\mathrm{cm}^{3/2}} \cdot B \tag{3.39}$$

Converting to Gauss we get

$$\hat{B} = B\left(2.76 \cdot 10^{18}\,\mathrm{G}\right) \tag{3.40}$$

There is a factor of $\sqrt{4\pi}$ due to the difference in the Lagrangian between the evolution code and the initial data code that needs to be accounted for in the conversion. Considering this, our final result is

$$\hat{B} = B\left(9.78 \cdot 10^{18}\,G\right) \tag{3.41}$$

We now have the tools to convert between the units of both our evolution code and initial data code to any units system as desired.

3.5 Additional improvements

There has been a good deal of work done in this code. Instead of having a collection of several files for initial data for a star, we have written code to collect all of the data files into one single binary file. This helps in controling the data for a specific initial data set. Additionally, we have added a text block at the end of the file that contains information such as version number, command line used to run mrns, stellar mass, etc. This data can be easily viewed by using the command "tail <filename>". This allows us to eliminate questions about the actual data in the file. This helps us eliminate problems of long term storage and remembering what is in the data file.

Chapter 4

Characteristic Decomposition

4.1 Introduction

When solving the MHD equations, we need to know how to deal with the equations at the boundaries of our computational domain. To determine the boundary conditions for our system of equations, we perform a characteristic decomposition of those equations. This decomposition produces a set of advection equations for the characteristic waves of the system. While we are not going to evolve the system with this set of advection equations, the process of decomposition will lead us to tools needed in order to construct the boundary conditions for the system of equations. Here we implement a method of characteristic decomposition to solve the equations.

We already have two sets of variables that describe the system. That is the primitive variables that we naturally use, and the conservative variables which are the variables in which the equations are written. In a characteristic decomposition we are looking for a new set of variables. For this new set of variables we want to transform from the coupled set of differential equations into a set of decoupled advection equations. Using this set of equations we can transform from a combination of the conservative and primitive variables into the characteristic variables. We then

do the evolution with the much simplified set of advection equations and transform back into the conserved variables.

This method allows a simplified evolution of a set of equations. However there still remains the difficulty of the transformation between the conservative variables and the characteristic variables. This is accomplished in several steps. First we need to calculate the characteristic speeds for the characteristic waves. This is done by solving an eigenvalue problem for the set of equations. We then calculate the left and right eigenvectors for the system. This set of eigenvectors gives us the ability to convert between the conservative and characteristic variables. Here we will go through this method in detail.

Along with the decomposition, we wish to resolve another complicated issue. The Maxwell equations contain the constraint equation $\nabla \cdot \vec{B} = 0$. This equation tells us that there are no magnetic monopoles. It must be satisfied in all of space. However, from the computational perspective, we do not expect this to remain satisfied thoughout the evolution. One can use the constraint as a measure for the accuracy of the evolution.

From the construction of the initial data this equation is indeed satisfied. However, due to several issues including truncation errors and round off errors, there will be instances where there are small deviations from zero. If the violation remained at the level of truncation error, this would generally not be a problem. However, these inconsistencies tend to grow and move around the computational domain. Moreover, as these instabilities reach the computational boundary, they are reflected back into the computational domain. This causes contamination of our solution. There have have been multiple attempts to resolve this issue including constrained transport and parabolic divergence cleaning [18].

We are going to use hyperbolic divergence cleaning. We will consider a new field ϕ which is a measure of the violation of the constraint equation. The purpose of the new equation is to be driven to zero through the evolution. As ϕ is driven to zero, so is the constraint $\nabla \cdot \vec{B}$. With a damped

wave equation we can dynamically handle monopole condition. This will increase the number of equations that we are evolving and add a wave speed into the decomposition.

4.2 Decomposition methodology

We describe here the methodology to decouple a system of differential equations into a set of uncoupled advection equations. We are following an approach similar to the one prescribed in [19]. Additionally we are adding onto work already done in [16]. Suppose we have a system of equations that takes the form

$$\partial_t \vec{u} = -\partial_x (A\vec{u}). \tag{4.1}$$

where A is an $N \times N$ matrix and $\vec{u}$ is a vector of N components that represent the variables of the system. We make the assumption that A is a constant matrix in both space and time.

This is a set of coupled equations. There are several methods to solve this set of equations. A complication in solving such a set of equations is properly determining the conditions of the variables at the boundary of our problem. This point will be discussed later. We choose to simplify the system of equations by diagonalizing the matrix A in order to convert the system of coupled equations into a system of decoupled advection equations. One can view this as a rotation of our system in the space of variables. We do this in the following way. There exists a matrix B such that $BAB^{-1} = D$ where D is a diagonal matrix. We will define $\vec{v} = B\vec{u}$. Substitute this into equation 4.1

we get

$$\partial_t B^{-1}\vec{v} = -\partial_x(AB^{-1}\vec{v}) \tag{4.2}$$

$$B^{-1}\partial_t\vec{v} = -\partial_x(AB^{-1}\vec{v}) \tag{4.3}$$

$$\partial_t\vec{v} = -B\partial_x(AB^{-1}\vec{v}) \tag{4.4}$$

$$\partial_t\vec{v} = -\partial_x(BAB^{-1}\vec{v}) \tag{4.5}$$

$$\partial_t\vec{v} = -\partial_x(D\vec{v}). \tag{4.6}$$

The vector $\vec{v}$ is known as the characteristic variables for the system of equations and D is the diagonal matrix that has the wave speeds along the diagonal. We can write this out as a set of advection equations

$$\partial_t v_i = c_i\partial_x v_i \tag{4.7}$$

where v_i is one of the characteristic variables and c_i is its associated wave speed.

The computation of the matrix B is performed by finding the eigenvalues and associated eigenvectors for the matrix A. The eigenvalues are computed by solving for the solutions to the equation

$$\det\left|A-\lambda\mathbb{I}\right| = 0. \tag{4.8}$$

We then obtain each eigenvector equation

$$A\vec{e}_i = \lambda_i\vec{e}_i, \tag{4.9}$$

where λ_i are the computed eigenvalues and $\vec{e}_i$ are their associated eigenvectors. We construct B such that its columns are the above computed eigenvectors. The matrix B^{-1} can be computed either by directly inverting the matrix B or computing the left eigenvectors with the equation

$$\vec{l}_i A = \lambda_i\vec{l}_i, \tag{4.10}$$

where $\vec{l}_i$ are the left eigenvectors associated with λ_i. The left eigenvectors are the rows of the inverse matrix B^{-1}. Here we compute the left eigenvectors directly.

4.2.1 Computing the boundary conditions

As mentioned above, one of the complications in evolving a set of equations is determining what the values of the fields are at the computational boundaries. For our particular system, the computational domain has fixed boundaries that are not at infinity. This complicates our boundary conditions as compared to boundaries at infinity. We need to consider the waves as they interact at these fixed boundaries. We do not expect any incoming waves into our system. However, due to the use of our computational models, there may be some reflection off our fictitious boundaries. This reflection is a nonphysical addition in the system that looks like incoming waves from outside the computational domain. We need to set boundary conditions such that these incoming waves are suppressed without changing the outgoing waves. In order to this, we follow the same analysis as [20, 21].

When finding the boundary conditions for a partial differential equations, there are several methods to obtain them. We have chosen to use constraint preserving boundary conditions as prescribed in [21]. We use the left eigenvectors that correspond to the wave that we wish to set the boundaries for. We combine them in the following way

$$\vec{l}_- \cdot \vec{U} + a\vec{l}_+ \cdot \vec{U} \approx 0, \tag{4.11}$$

where $\vec{l}_\pm$ are the left eigenvectors associated with the eigenvalues that correspond to the field that we are working with. $\vec{U}$ is the vector of characteristic variables. The parameter a is adjustable so that we can modify how much to suppress the incoming wave. The first part of this equation $\vec{l}_- \cdot \vec{U}$ is the outgoing characteristic wave from the system. The second part $\vec{l}_+ \cdot \vec{U}$ is the incoming wave for the same characteristic wave. With a as an adjustable parameter, we can control the amount of incoming wave that we wish to suppress. This model is known as dissipative boundary conditions. If we set $a = 0$ then we have completely dissipative boundary conditions and the incoming wave is completely suppressed.

4.3 Example: The wave equation

To give an example of this procedure, we will use, as a toy problem, the one dimensional wave equation. The wave equation is written as

$$\phi_{tt} = c^2\phi_{xx} \tag{4.12}$$

$$\tag{4.13}$$

where c is the wave speed and ϕ is the field. This is a second order differential equation. We want to write the wave equation as a pair of coupled first order equations. To do this, we make the following definitions

$$\Pi \equiv \phi_t \tag{4.14}$$

$$\Phi \equiv \phi_x \tag{4.15}$$

The above definitions when applied to equation (4.12) give us a system of equations

$$\Pi_t = c^2\Phi_x \tag{4.16}$$

$$\Phi_t = \Pi_x \tag{4.17}$$

We can write these in matrix form:

$$\partial_t V = \partial_x(A \cdot V) \tag{4.18}$$

where $V = (\Pi, \Phi)^T$ and

$$A = \begin{pmatrix} 0 & c^2 \\ 1 & 0 \end{pmatrix}. \tag{4.19}$$

One can easily find that the eigenvalues of the constant matrix A are $\pm c$. This is good since these are the wave speeds for our given wave equation. Additionally, we can calculate the right eigenvectors to be $(c,1)^T$ for the eigenvalue c and $(-c,1)^T$ for the eigenvalue $-c$. We now construct

the matrix of right eigenvectors

$$\begin{pmatrix} c & -c \\ 1 & 1 \end{pmatrix} \tag{4.20}$$

We can obtain the left eigenvector by simply inverting this matrix

$$\frac{1}{2c}\begin{pmatrix} 1 & c \\ -1 & c \end{pmatrix}. \tag{4.21}$$

The rows of this matrix are the left eigenvectors for our system. Using these two left eigenvectors we can compute the characteristic variables for the wave equation as $\frac{\Pi+c\Phi}{2c}$ and $\frac{-\Pi+c\Phi}{2c}$. These characteristic variables satisfy their own advection equations one moving to the left at speed c and one moving to the right at speed c.

To compute the boundary conditions as prescribed with equation (4.11), we use the left eigenvectors we just found. In order to calculate the boundary conditions for the fields we combine the left eigenvectors in the following way

$$\vec{l}_- \cdot \vec{U} + a\vec{l}_+ \cdot \vec{U} = 0 \tag{4.22}$$

where again a is an adjustable parameter such that $|a| < 1$, l_+ is the left eigenvector associated with the right moving wave and l_- is the one for the left moving wave. This equation will allow us to set the boundary conditions for our variables. For the case of the wave equation we get the following relation for our boundary condition

$$\Pi = \Phi\frac{c(a+1)}{a-1}. \tag{4.23}$$

This becomes our boundary condition for Π in terms of Φ. If we set $a = 0$ we have completely dissipative boundary conditions completely suppressing any incoming wave for these fields.

4.4 The equation set

The above example is for the wave equation which is linear and therefore easy to decouple. However, our system is a bit more complicated. Recall our complete system of equations including the source terms:

$$\partial_t\left(\sqrt{h}D\right)+\partial_i\left[\sqrt{-g}D\left(v^i-\frac{\beta^i}{\alpha}\right)\right] = 0 \tag{4.24}$$

$$\partial_t\left(\sqrt{h}E\right)+\partial_i\left[\sqrt{-g}\left(S^i-\frac{\beta^i}{\alpha}E\right)\right] = \sqrt{-g}\left[(\perp T)^{ab}K_{ab}-\frac{1}{\alpha}S^a\partial_a\alpha\right] \tag{4.25}$$

$$\begin{aligned}\partial_t\left(\sqrt{h}S_b\right)+\partial_i\left[\sqrt{-g}\left((\perp T)^i{}_b-\frac{\beta^i}{\alpha}S_b\right)\right] &= \sqrt{-g}\left[{}^3\Gamma^i_{ab}(\perp T)^a{}_i+\frac{1}{\alpha}S_a\partial_b\beta^a\right.\\ &\quad\left.-\frac{1}{\alpha}\partial_b\alpha E\right]\end{aligned} \tag{4.26}$$

$$\begin{aligned}\partial_t\left(\sqrt{h}B^b\right)+\partial_i\left[\sqrt{-g}\left(B^b\left(v^i-\frac{\beta^i}{\alpha}\right)\right.\right.&\\ \left.\left.-B^i\left(v^b-\frac{\beta^b}{\alpha}\right)+\phi h^{ib}-\eta\beta^b\sqrt{h}B^i\right]\right. &= (1-\eta)\beta^b\partial_i\left(\sqrt{h}B^i\right)-\eta\sqrt{h}B^i\partial_i\beta^b\\ &\quad+\sqrt{-g}\left[\phi h^{ib}\partial_i ln\alpha-{}^3\Gamma^b_{ij}h^{ij}\phi\right]\end{aligned} \tag{4.27}$$

$$\partial_t\left(\sqrt{h}\phi\right)+\partial_i\left[\sqrt{-g}\left(B^i-\frac{\beta^i}{\alpha}\phi\right)\right] = \sqrt{-g}\left[B^i\partial_i ln\alpha-K\phi-s\phi\right]. \tag{4.28}$$

Note that the last equation is for the new wave ϕ. This new field along with the freely specifiable variable s will provide us with a damping term that will allow us to evolve away any violations of the no monopole condition $\nabla\cdot\vec{B}=0$. Any violation of that condition will be suppressed with this new field. Other than the geometric terms, the last equation is simply a dispersion equation that relates the time change in ϕ to the divergence of the magnetic field. Additionally there are terms in the fourth equations with the terms η and $(1-\eta)$. These terms are here so that we can determine whether the associated pieces belong as source terms or evolution terms. If we set $\eta=1$ they are treated as flux terms. If we set $\eta=0$ they will be treated as source terms.

To obtain the characteristic structure of the system of equations, we write our balance law equations as a set of conservation, or continuity equations. We define the primitive variables

$\vec{p} = \begin{pmatrix} \rho_0 & v^j & \varepsilon & B^j & \phi \end{pmatrix}^T$. Without source terms we can write the above equations

$$
\begin{aligned}
0 &= \frac{\partial \vec{F}^a(\vec{p})}{\partial x^a} && (4.29)\\
&= \frac{\partial \vec{F}^a(\vec{p})}{\partial \vec{p}} \frac{\partial \vec{p}}{\partial x^a} && (4.30)\\
&= \mathscr{F}^a \frac{\partial \vec{p}}{\partial x^a}, && (4.31)
\end{aligned}
$$

where $\vec{F}^a$ is the vector of conservative variables which are written in terms of the primitive variables. The matrix $\mathscr{F}^a$ which is defined as $\frac{\partial \vec{F}^a(\vec{p})}{\partial \vec{p}}$ is the Jacobian associated with the vector $\vec{F}^a$. We can work with equation 4.31 to get

$$
\begin{aligned}
0 &= \mathscr{F}^0 \frac{\partial \vec{p}}{\partial t} + \mathscr{F}^i \frac{\partial \vec{p}}{\partial x^i} && (4.32)\\
0 &= (\mathscr{F}^0)^{-1} \mathscr{F}^0 \frac{\partial \vec{p}}{\partial t} + (\mathscr{F}^0)^{-1} \mathscr{F}^i \frac{\partial \vec{p}}{\partial x^i} && (4.33)\\
0 &= \frac{\partial \vec{p}}{\partial t} + (\mathscr{F}^0)^{-1} \mathscr{F}^i \frac{\partial \vec{p}}{\partial x^i}. && (4.34)
\end{aligned}
$$

This now is equivalent to equation 4.1 with the matrix being replaced with $(\mathscr{F}^0)^{-1}\mathscr{F}^i$. We now need to solve for the eigenvalues and eigenvectors of that matrix. The eigenvalue equation becomes

$$
\begin{aligned}
0 &= \det\left|(\mathscr{F}^0)^{-1}\mathscr{F}^i - \lambda \mathbb{I}\right| && (4.35)\\
&= \det\left|(\mathscr{F}^0)^{-1}\mathscr{F}^i - \lambda (\mathscr{F}^0)^{-1}\mathscr{F}^0\right| && (4.36)\\
&= \det\left|(\mathscr{F}^0)^{-1}\left(\mathscr{F}^i - \lambda \mathscr{F}^0\right)\right| && (4.37)\\
&= \det\left|(\mathscr{F}^0)^{-1}\right| \det\left|\mathscr{F}^i - \lambda \mathscr{F}^0\right|. && (4.38)
\end{aligned}
$$

From this point on we will focus on a particular direction which we call x^k. We will cycle k through our three spatial dimensions. We consider this direction fixed in our subsequent analysis. Because we know that the determinant of $\mathscr{F}^0$ is not zero, equation 4.38 becomes

$$\det\left|\mathscr{F}^0(-\lambda^k) + \mathscr{F}^k\right| = 0 \tag{4.39}$$

where λ^k is one of several eigenvalues associated with a wave moving in the x^k direction. We additionally need to look at the equation for the eigenvectors. We start with equation (4.9) and do the following

$$\left(\mathscr{F}^0\right)^{-1}\mathscr{F}^i\vec{e}_i = \lambda_i\vec{e}_i \tag{4.40}$$

$$\left(\mathscr{F}^0\right)^{-1}\mathscr{F}^i\vec{e}_i = \lambda_i\left(\mathscr{F}^0\right)^{-1}\mathscr{F}^0\vec{e}_i \tag{4.41}$$

$$\mathscr{F}^i\vec{e}_i = \lambda_i\mathscr{F}^0\vec{e}_i. \tag{4.42}$$

We now write the equations in terms of the primitive variables

$$\vec{F}^0 = \sqrt{h}\begin{pmatrix} D \\ S_i \\ \tau \\ B_i \\ \phi \end{pmatrix} \equiv \sqrt{h}\begin{pmatrix} W\rho_0 \\ (h_eW^2+B^2)v_i-(Bv)B_i \\ h_eW^2+B^2-\frac{1}{2}[(Bv)^2+\frac{B^2}{W^2}]-W\rho_0-P \\ B_i \\ \phi \end{pmatrix} \tag{4.43}$$

$$\vec{F}^k = \alpha\sqrt{h}\begin{pmatrix} W\rho_0\bar{v}^k \\ \Lambda_1 \\ \Lambda_2 \\ -B^k\vec{v}^i+B^i\vec{v}^k+\phi h^{ik}-\eta B^k\frac{\beta^i}{\alpha} \\ B^k-\frac{\beta^k}{\alpha}\phi \end{pmatrix} \tag{4.44}$$

where $B^2 = B^iB_i$, $Bv = B^iv_i$, $\bar{v}^k = v^k - \frac{\beta^k}{\alpha}$ and

$$\begin{aligned}\Lambda_1 &= \left(\left[h_eW^2+B^2\right]v_i-(Bv)B_i\right)\bar{v}^k + h_i^k\left(P+\frac{1}{2}\left[(Bv)^2+\frac{B^2}{W^2}\right]\right) \\ &\quad -\left(\frac{B_i}{W^2}+(Bv)v_i\right)B^k \end{aligned}\tag{4.45}$$

$$\begin{aligned}\Lambda_2 &= \left(h_eW^2+B^2-\frac{1}{2}\left[(Bv)^2+\frac{B^2}{W^2}\right]-W\rho_0-P\right)\bar{v}^k \\ &\quad +\left(P+\frac{1}{2}\left[(Bv)^2+\frac{B^2}{W^2}\right]\right)v^k-(Bv)B^k. \end{aligned}\tag{4.46}$$

The following are some definitions that will assist in the calculations in the following section

$$\frac{\partial W}{\partial v_j} = \frac{\partial}{\partial v_j}\left(1 - v_i v^i\right)^{-\frac{1}{2}} = W^3 v^j \tag{4.47}$$

$$\chi = \frac{\partial P}{\partial \rho_o} \tag{4.48}$$

$$\kappa = \frac{\partial P}{\partial \varepsilon} \tag{4.49}$$

$$\gamma = \frac{\partial h_e}{\partial \rho_o} = 1 + \varepsilon + \chi \tag{4.50}$$

$$h_e c_s^2 = \rho_o \chi + \frac{P}{\rho_o}\kappa \tag{4.51}$$

$$h_e = \rho_o (1 + \varepsilon) + P. \tag{4.52}$$

We take the derivatives of the above vectors to compute the Jacobians.

$$\frac{\mathscr{F}^0}{\sqrt{h}} = \begin{pmatrix} W & W^3 v_j \rho_0 & 0 & 0_j & 0 \\ \gamma W^2 v^i & \Omega_1 & (\rho_o + \kappa) W^2 v^i & 2B_j v^i - (Bv) h^i_j - v_j B^i & 0_j \\ \gamma W^2 - W - \chi & \Omega_2 & (\rho_o + \kappa) W^2 - \kappa & 2B_j - (Bv) v_j - \frac{B_j}{W^2} & 0 \\ 0^i & 0^i_j & 0^i & h^i_j & 0^i \\ 0 & 0_j & 0 & 0_j & 1 \end{pmatrix} \tag{4.53}$$

$$\frac{\mathscr{F}^k}{\alpha\sqrt{h}} = \begin{pmatrix} W\bar{v}^k & W\rho_0 h^k_j + W^3 \rho_0 v_j \bar{v}^k & 0^k & 0^k & 0^k \\ \bar{v}^k \gamma W^2 v^i + \chi h^{ik} & D & \Omega_3 v^i + h^{ik}\kappa & A & 0^{ik} \\ \Omega_5 & E & \Omega_3 + \kappa(v^k - \bar{v}^k) & C & 0^k \\ 0^i & -B^k h^i_j + B^i h^k_j & 0^{ik} & \Omega_4 & h^{ik} \\ 0^k & 0^k_j & 0^k & h^k_j & -\frac{\beta^k}{\alpha} \end{pmatrix} \tag{4.54}$$

with 0^i a column vector of three zeros, 0_j a row vector of three zeros, 0^i_j a 3×3 matrix of zeros,

0^k as a place holder to make sure that the indexing is consistant, and the substitutions

$$\Omega_1 = h^i_j(h_eW^2+B^2)+2h_ew^4v_jv^i-B_jB^i \tag{4.55}$$

$$\Omega_2 = 2W^4v_jh_e-B_j(Bv)+B^2v_j-W^3v_j\rho_0 \tag{4.56}$$

$$\Omega_3 = (\rho_0+\kappa)W^2\bar{v}^k \tag{4.57}$$

$$\Omega_4 = -h^k_j(\bar{v}^i-\eta\frac{\beta^i}{\alpha})+h^i_j\bar{v}^k \tag{4.58}$$

$$\Omega_5 = (\gamma W^2-W)\bar{v}^k+\chi(v^k-\bar{v}^k) \tag{4.59}$$

$$\begin{aligned} A = {} & 2B_jv^i\bar{v}^k-v_jB^i\bar{v}^k-(Bv)h^i_j\bar{v}^k+\left[(Bv)v_j+\frac{B_j}{W^2}\right]h^k_j \\ & -\left(\frac{h^i_j}{W^2}+v^iv_j\right)B^k-\left(\frac{B^i}{W^2}+(Bv)v^i\right)h^k_j \end{aligned} \tag{4.60}$$

$$C = \left(2B_j-(Bv)v_j-\frac{B_j}{W^2}\right)\bar{v}^k+\left[(Bv)v_j+\frac{B_j}{W^2}\right]v^k-(Bv)h^k_j-v_jB^k \tag{4.61}$$

$$\begin{aligned} D = {} & h^k_j([h_eW^2+B^2]v^i-(Bv)B^i)+[h_eW^2+B^2]h^i_j\bar{v}^k+2h_eW^4v_jv^i\bar{v}^k \\ & +(Bv)B_jh^{ik}-(Bv)h^i_jB^k-B_jv^iB^k+2B^iv_jB^k-B_jB^i\bar{v}^k-B^2v_jh^{ik} \end{aligned} \tag{4.62}$$

$$\begin{aligned} E = {} & \bar{v}^k(2h_eW^4v_j-(Bv)B_j+B^2v_j-W^3v_j\rho_0) \\ & +h^k_j(h_eW^2+B^2-W\rho_0)+v^k((Bv)B_j-B^2v_j)-B_jB^k \end{aligned} \tag{4.63}$$

We now have all that we need to solve for the eigenvectors and eigenvalues using equation (4.42). We reiterate it here

$$\mathscr{F}^k\vec{e}-\lambda^k\mathscr{F}^0\vec{e}=0. \tag{4.64}$$

4.5 Finding the eigenvalues

Sparing the details which are long and arduous we give the final equations for the eigenvalues:

$$\begin{aligned} 0 = \quad & \Big[h_e W^4(1-c_s^2)(a^k)^4 + \Big[(a^k)^2(h_e W^2 c_s^2 + B^2 + W^2(Bv)^2) \\ & -c_s^2\left(W(Bv)a^k + \tfrac{B^k}{W}\right)^2\Big]\left[(a^k - v^k)^2 - h^{kk}\right]\Big] \\ & \cdot(\alpha\sqrt{h})^9 h^2 W^3 \rho_0 h_e a^k \Delta^{kk}\Big[(a^k - v^k)[a^k - [(1-\eta)\bar{v}^k + \eta v^k]] - h^{kk}\Big] \end{aligned} \tag{4.65}$$

where

$$\Delta^{kk} = (a^k)^2 Q - 2a^k(Bv)B^k - \frac{1}{W^2}B^k B^k \tag{4.66}$$

$$a^k = \lambda^k - v^k \tag{4.67}$$

$$Q = h_e W^2 + B^2. \tag{4.68}$$

Equation (4.65) is a ninth order polynomial equation. We can separate it into several pieces. The first two lines of this equation are a quartic equation. When this piece is zero, the resultant eigenvalues correspond to the wave speed associated with the magnetosonic waves. When we have $\Delta^{kk} = 0$, we get the wave speeds associated with the Alfven waves. For the term $a^k = 0$ we get the wave speed for the entropy wave. For the last term, $(a^k - v^k)[a^k - [(1-\eta)\bar{v}^k + \eta v^k]] - h^{kk} = 0$, we are solving for the waves speeds associated with the new ϕ wave.

4.6 Solving for the right eigenvectors

The approach for solving for the eigenvectors is standard. We pick a particular value of a^k that is a solution to the eigenvalue equation. We then solve the set of equations for the components of the associated eigenvector. The right eigenvectors are of the form

$$\vec{e} = \begin{pmatrix} e^0 & e^j & e^4 & \hat{e}^j & e^8 \end{pmatrix}^T. \tag{4.69}$$

We now solve the system of equations for a particular eigenvalue λ^k. In this form, our system of equations becomes

$$
\begin{aligned}
0 &= e^0 W a^k + (ve) W^3 \rho_0 a^k + e^k W \rho_0 && (4.70)\\
0 &= e^0 \left[W^2 \gamma v_i a^k + \chi h_i^k \right] + e_i \left[Q a^k - (Bv) B^k \right] + (ve) \left[2 h_e W^4 v_i a^k - h_i^k B^2 + 2 B_i B^k \right] \\
&\quad + (Be) \left[-B_i a^k + h_i^k (Bv) - v_i B^k \right] + e^k \left[Q v_i - (Bv) B_i \right] \\
&\quad + e^4 \left[(\rho_0 + \kappa) W^2 a^k v_i + h_i^k \kappa \right] \\
&\quad + \hat{e}_i \left[-(Bv) a^k - \frac{B^k}{W^2} \right] + (v\hat{e}) \left[-B_i a^k - v_i B^k + h_i^k (Bv) \right] + (B\hat{e}) \left[2 v_i a^k + \frac{h_i^k}{W^2} \right] \\
&\quad - \hat{e}^k \left[(Bv) v_i + \frac{B_i}{W^2} \right] && (4.71)\\
0 &= e^0 \left[(W^2 \gamma - W - \chi) a^k + \chi v^k \right] + (ve) \left[(2 h_e W^4 + B^2 - W^3 \rho_0) a^k - B^2 v^k \right] \\
&\quad + (Be) \left[-(Bv) a^k + (Bv) v^k - B^k \right] + e^k \left[Q - W \rho_0 \right] \\
&\quad + e^4 \left[(\rho_0 + \kappa) W^2 a^k + \kappa (v^k - a^k) \right] \\
&\quad + (v\hat{e}) \left[-(Bv) a^k + (Bv) v^k - B^k \right] + (B\hat{e}) \left[2 a^k + \frac{v^k - a^k}{W^2} \right] - \hat{e}^k \left[(Bv) \right] && (4.72)\\
0 &= B^i e^k - e^i B^k + \hat{e}^i a^k - \hat{e}^k \bar{v}^i - \eta \frac{\beta^i}{\alpha} \hat{e}^k + h^{ik} e^8 && (4.73)\\
0 &= \hat{e}^k - e^8 \left(\frac{\beta^i}{\alpha} + \lambda^k \right) && (4.74)
\end{aligned}
$$

where $Be = B_i e^i$, $B\hat{e} = B_i \hat{e}^i$, $ve = v_i e^i$, $v\hat{e} = v_i \hat{e}^i$. It is instructive if we first look at equations (4.73) and (4.74). On contracting equation (4.73) with h_i^k we get

$$
0 = \hat{e}^k \left(a^k - \bar{v}^k - \eta \frac{\beta^k}{\alpha} \right) + h^{kk} e^8 . \tag{4.75}
$$

We now use this equation and equation (4.74) and look for solutions for e^8 and $\hat{e}^k$.

$$
\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a^k - \bar{v}^k - \eta \frac{\beta^k}{\alpha} & h^{kk} \\ 1 & -\left(\frac{\beta^k}{\alpha} + \lambda^k \right) \end{pmatrix} \cdot \begin{pmatrix} \hat{e}^k \\ e^8 \end{pmatrix} . \tag{4.76}
$$

We can see that if the determinant of the above matrix is not equal to zero then e^8 and $\hat{e}^k$ must be zero. If we evaluate the determinant of the above matrix using the fact that $\lambda^k = \bar{v}^k - a^k$ we get

$$\det \begin{vmatrix} a^k - \bar{v}^k - \eta \frac{\beta^k}{\alpha} & h^{kk} \\ 1 & -\left(\frac{\beta^k}{\alpha} + \lambda^k\right) \end{vmatrix} = \left(a^k - v^k\right)\left[a^k - \left[(1-\eta)\bar{v}^k + \eta v^k\right]\right] - h^{kk} \tag{4.77}$$

This is the factor of the eigenvalue equation that represents wave speeds for the ϕ wave. This will be zero when we consider the eigenvectors associated with the ϕ waves. The only way to resolve this in other cases is for the components e^k and e^8 of all the other waves vanish. In other words, since we are first going to look at the eigenvalues other than the two associated with this equation, for those eigenvectors we will have $\hat{e}^k = e^8 = 0$.

4.6.1 The Entropy wave

We begin by solving for the eigenvector for the entropy wave. The entropy wave, as we have mentioned, corresponds to the eigenvalue $a^k = 0$. Equation (4.70) becomes

$$0 = e^k W \rho_o. \tag{4.78}$$

Therefore we have $e^k = 0$. From this we can look at Equation (4.73) and get

$$0 = -e^i B^k. \tag{4.79}$$

Therefore we find $e^i = 0$. Because we can write

$$Be = B_i e^i, \tag{4.80}$$

we conclude that $Be = 0$. A similar argument is made to see that $ve = 0$. Contracting equation (4.71) with B^i we get:

$$0 = e^0 \chi B^k + e^4 \kappa B^k + (B\hat{e})\left[-\frac{B^k}{W^2} + \frac{B^k}{W^2}\right] + (v\hat{e})\left[-(Bv)B^k + (Bv)B^k\right]. \tag{4.81}$$

Note that the last four terms cancel out leaving $e^4 = -\frac{\chi}{\kappa}e^0$. Contracting equation (4.71) with $\hat{e}^i$ we get

$$0 = \left(\frac{\hat{e}_i\hat{e}^i}{W^2} + (v\hat{e})^2\right) B^k \tag{4.82}$$

Both terms in the parenthesis are positive definite and therefore B^k must be zero. Additionally, if we have two positive definite things adding to zero, both must be zero. Therefore:

$$\hat{e}_i\hat{e}^i = 0 = (v\hat{e})^2 \tag{4.83}$$

consequently, $\hat{e}^i = 0$. Combining all of these together we get the final eigenvector for the entropy wave as

$$\vec{e} = \begin{pmatrix} e_0 \\ 0^j \\ -e_0\frac{\chi}{\kappa} \\ \hat{0}^j \\ 0 \end{pmatrix} \tag{4.84}$$

When solving eigenvalue problems, there is always a normalization parameter that is freely adjustable. In this case, that parameter is e^0. When we set $e^0 = 1$ we get the following right eigenvector for our system:

$$\vec{e} = \begin{pmatrix} 1 \\ 0^j \\ -\frac{\chi}{\kappa} \\ \hat{0}^j \\ 0 \end{pmatrix} \tag{4.85}$$

The entropy waves are compression waves in the fluid. We expect these to have fluctuations in the pressure and density terms. This vector has components that deal only with the fluid components of the primitive variables as expected.

4.6.2 Alfven waves

Recall that the term of the eigenvalue equation associated with the Alfven waves is when $\Delta^{kk} = 0$. To solve for the Alfven waves, we can start with equation (4.73). We contract it with B_i and solve for $B\hat{e}$ to get

$$B\hat{e} = (Be)\frac{B^k}{a^k} - e^k\frac{B^2}{a^k}. \tag{4.86}$$

Contracting equation (4.73) with v_i and solve for $v\hat{e}$ to get

$$v\hat{e} = (ve)\frac{B^k}{a^k} - e^k\frac{Bv}{a^k}. \tag{4.87}$$

Note that these two equations will also hold for the magnetosonic waves described below.

From equation (4.70) one gets

$$e^k = e^0\left[\frac{-a^k}{\rho_o}\right] - (ve)W^2a^k. \tag{4.88}$$

After quite a bit of manipulation using equations (4.70, 4.71, and 4.72) one can arrive at the following equation

$$e^4 = e^0\frac{P}{\rho_o^2}. \tag{4.89}$$

From equation (4.73) one gets

$$e^i = \frac{1}{B^k}\left[B^ie^k + \hat{e}^ia^k\right]. \tag{4.90}$$

In order to get equations for $\hat{e}^1_\perp$ and $\hat{e}^2_\perp$, which are the components of the vector $\hat{e}$ in an arbitrary plain perpendicular to the direction of $\vec{e}^k$, one can consider the following

$$v\hat{e} = v_1\hat{e}^1_\perp + v_2\hat{e}^2_\perp + v_k\hat{e}^k \tag{4.91}$$

$$B\hat{e} = B_1\hat{e}^1_\perp + B_2\hat{e}^2_\perp + B_k\hat{e}^k. \tag{4.92}$$

If we recall that $\hat{e}^k = 0$ we can rewrite these two equations to arrive at

$$\hat{e}^1_\perp = \frac{1}{v_1 B_2 - B_1 v_2}(B_2(v\hat{e}) - v_2(B\hat{e})) \tag{4.93}$$

$$\hat{e}^2_\perp = \frac{1}{v_1 B_2 - B_1 v_2}(v_1(B\hat{e}) - B_1(v\hat{e})) \tag{4.94}$$

All that is needed now are the values of ve and Be. In order to calculate these, we combine our above knowledge and write equations (4.71) and (4.72) in matrix form as follows:

$$\begin{pmatrix} h_e W^4 a^k - C^k + W^2 E^k & D^k \\ h_e W^4 a^k (Bv) & h_e W^2 a^k \end{pmatrix} \begin{pmatrix} ve \\ Be \end{pmatrix} = -\frac{e^0}{\rho_o} \begin{pmatrix} h_e c_s^2 (v^k - a^k + W^2 a^k) + E^k \\ h_e c_s^2 (B^k + W^2 a^k (Bv)) \end{pmatrix} \tag{4.95}$$

with the following definitions

$$C^k \equiv (v^k - a^k)\left[B^2 - \frac{(Bv)B^k}{a^k}\right] + \frac{B^k B^k}{a^k} \tag{4.96}$$

$$D^k \equiv (v^k - a^k)\left[(Bv) + \frac{B^k}{a^k W^2}\right] + B^k \tag{4.97}$$

$$E^k \equiv (v^k - a^k)\left[(Bv)^2 + \frac{B^2}{W^2}\right] + B^2 a^k - (Bv)B^k. \tag{4.98}$$

To solve this we need to invert the matrix on the left hand side. It is easy to show that the determinant of the matrix is $h_e W^4 \Delta^{kk}$ and therefore cannot be inverted because $\Delta^{kk} = 0$ for Alfven waves. Because the matrix is singular we conclude that the right hand side of the equation is zero and therefore we must have $e^0 = 0$ in order to be consistent.

The second component of equation (4.95) can be written as

$$(ve)W^2(Bv) + Be = 0 \tag{4.99}$$

and used to eliminate *Be* from the eigenvector. The final solution to these equations is:

$$ve\begin{pmatrix} 0 \\ \frac{1}{B^k}\left[-B^1W^2a^k+\frac{a^k}{v_1B_2-B_1v_2}\left(B_2\left[\frac{B^k}{a^k}+W^2(Bv)\right]-v_2\left[W^2B^2-W^2\frac{B^k}{a^k}(Bv)\right]\right)\right] \\ \frac{1}{B^k}\left[-B^2W^2a^k+\frac{a^k}{v_1B_2-B_1v_2}\left(v_1\left[-W^2\frac{B^k}{a^k}(Bv)+W^2B^2\right]-B_1\left[W^2(Bv)+\frac{B^k}{a^k}\right]\right)\right] \\ -W^2a^k \\ 0 \\ \frac{1}{v_1B_2-B_1v_2}\left[B_2\left[\frac{B^k}{a^k}+W^2(Bv)\right]-v_2\left[W^2B^2-W^2\frac{B^k}{a^k}(Bv)\right]\right] \\ \frac{1}{v_1B_2-B_1v_2}\left[-B_1\left[\frac{B^k}{a^k}+W^2(Bv)\right]+v_1\left[W^2B^2-W^2\frac{B^k}{a^k}(Bv)\right]\right] \\ 0 \\ 0 \end{pmatrix} \tag{4.100}$$

Alfven waves are produced by the transverse motion of the magnetic field lines. The Maxwell equations provide a restoring force to the field lines. Therefore we expect for an Alfven wave the components that are effected we be in the direction perpendicular to the wave. Here we have such a structure as expected.

4.6.3 Magnetosonic waves

Recall the wave speeds for magnetosonic waves are the solutions for the quartic portion of the eigenvalue euation are zero. The magnetosonic waves are developed exactly as the Alfven waves with the exception that $\Delta^{kk} \neq 0$. We return to that point of the calculation and continue. In this case we branch at the inversion of the above matrix in equation (4.95)

$$\begin{pmatrix} h_eW^4a^k-C^k+W^2E^k & D^k \\ h_eW^4a^k(Bv) & h_eW^2a^k \end{pmatrix}\begin{pmatrix} ve \\ Be \end{pmatrix}=-\frac{e^0}{\rho_o}\begin{pmatrix} h_ec_s^2(v^k-a^k+W^2a^k)+E^k \\ h_ec_s^2(B^k+W^2a^k(Bv)) \end{pmatrix} \tag{4.101}$$

In this case the matrix is invertible and we can solve for ve and Be.

$$\begin{aligned}
\begin{pmatrix} ve \\ Be \end{pmatrix} &= -\frac{e^0}{\rho_o} \begin{pmatrix} h_e W^4 a^k - C^k + W^2 E^k & D^k \\ h_e W^4 a^k (Bv) & h_e W^2 a^k \end{pmatrix}^{-1} \\
&\quad \cdot \begin{pmatrix} h_e c_s^2 (v^k - a^k + W^2 a^k) + E^k \\ h_e c_s^2 (B^k + W^2 a^k (Bv)) \end{pmatrix} \qquad (4.102) \\
\begin{pmatrix} ve \\ Be \end{pmatrix} &= -\frac{e^0}{\rho_o h_e W^4 \Delta^{kk}} \begin{pmatrix} h_e W^2 a^k & -D^k \\ -h_e W^4 a^k (Bv) & h_e W^4 a^k - C^k + W^2 E^k \end{pmatrix} \\
&\quad \cdot \begin{pmatrix} h_e c_s^2 (v^k - a^k + W^2 a^k) + E^k \\ h_e c_s^2 (B^k + W^2 a^k (Bv)) \end{pmatrix} \qquad (4.103) \\
\begin{pmatrix} ve \\ Be \end{pmatrix} &= -e^0 \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \qquad (4.104)
\end{aligned}$$

This is done with the following definition for α and β. (NOTE: these are not the same as the shift β^i and the lapse α):

$$\begin{aligned}
\alpha = & \quad \tfrac{-1}{\rho_0 h_e W^4 \Delta^{kk}} \left[h_e^2 W^2 c_s^2 a^k (v^k - a^k + W^2 a^k) + h_e W^2 a^k E^k \right. \\
& \quad \left. - D^k h_e c_s^2 (B^k + (Bv) W^2 a^k) \right] \qquad (4.105) \\
\beta = & \quad \tfrac{-1}{\rho_0 h_e W^4 \Delta^{kk}} \left[-h_e^2 W^4 c_s^2 a^k (Bv)(v^k - a^k + W^2 a^k) + h_e^2 W^4 a^k c_s^2 (B^k + (Bv) W^2 a^k) \right] \\
& \quad + \tfrac{-1}{\rho_0 h_e W^4 \Delta^{kk}} \left[-C^k h_e c_s^2 (B^k + (Bv) W^2 a^k) \right. \\
& \quad \left. + W^2 h_e c_s^2 E^k (B^k + (Bv) W^2 a^k) - h_e W^4 a^k (Bv) E^k \right]. \qquad (4.106)
\end{aligned}$$

We can now write

$$\begin{aligned}
B\hat{e} = & \quad e^0 \left(\beta \tfrac{B^k}{a^k} + B^2 \left(\tfrac{1}{\rho_0} + \alpha W^2 \right) \right) \qquad (4.107) \\
v\hat{e} = & \quad e^0 \left(\alpha \left(\tfrac{B^k}{a^k} + W^2 (Bv) \right) + \tfrac{Bv}{\rho_0} \right) \qquad (4.108)
\end{aligned}$$

The components of the eigenvector associated with the magnetosonic waves become

$$
\begin{aligned}
e^0 &= e^0 && (4.109)\\
e^i &= \frac{1}{B^k}\left[-B^i e^0\left(\frac{a^k}{\rho_0}+\alpha W^2 a^k\right)+\hat{e}^i_\perp a^k\right] && (4.110)\\
e^k &= -e^0\left(\frac{a^k}{\rho_0}-\alpha W^2 a^k\right) && (4.111)\\
e^4 &= e^0\frac{P}{\rho_0^2} && (4.112)\\
\hat{e}^1_\perp &= \frac{e^0}{v_1B_2-B_1v_2}\left[B_2\left(\alpha\left(\frac{B^k}{a^k}+W^2(Bv)\right)+\frac{Bv}{\rho_0}\right)\right.\\
&\quad \left.-v_2\left(\beta\frac{B^k}{a^k}+B^2\left(\frac{1}{\rho_0}+\alpha W^2\right)\right)\right] && (4.113)\\
\hat{e}^2_\perp &= \frac{e^0}{v_1B_2-B_1v_2}\left[-B_1\left(\alpha\left(\frac{B^k}{a^k}+W^2(Bv)\right)+\frac{Bv}{\rho_0}\right)\right.\\
&\quad \left.+v_1\left(\beta\frac{B^k}{a^k}+B^2\left(\frac{1}{\rho_0}+\alpha W^2\right)\right)\right] && (4.114)\\
\hat{e}^k &= 0 && (4.115)\\
e^8 &= 0 && (4.116)
\end{aligned}
$$

e^0 is our adjustable parameter that we will set to 1. Magnetosonic waves are composed of both the magnetic field and the matter of the system. We expect both magnetic field components as well as fluid components to be perturbed with this type of wave. The components of this eigenvector do indeed show the expected properties.

4.6.4 Divergence waves

Now all we have left to calculate is the last two eigenvectors for the ϕ waves. These are the ones that are associated with the eigenvalues when the determinant in equation (4.77) vanishes. This process is extremely long and complicated.

First contract equation (4.73) with h_i^k to get

$$0 = \hat{e}^k \left[a^k - \bar{v}^k - \eta \left(v^k - \bar{v}^k \right) \right] + e^8 h^{kk} \tag{4.117}$$

We then set

$$\xi = a^k - \bar{v}^k - \eta \left(v^k - \bar{v}^k \right) \tag{4.118}$$

to get

$$\hat{e}^k = -\frac{h^{kk}}{\xi} e^8 \tag{4.119}$$

One can do a similar procedure from equation (4.74) and get:

$$\hat{e}^k = -\frac{h^{kk}}{\xi'} e^8 \tag{4.120}$$

$$\xi' = -\frac{h^{kk}}{v^k - a^k} \tag{4.121}$$

Because the values ξ and ξ' are numerically identical for at least flat space, either can be used.

Now take equation (4.73) and contract with B^i and v^i respectively. If we define μ and ν as follows:

$$\mu = \xi B^k + h^{kk} \left((B\bar{v}) + \eta (Bv - B\bar{v}) \right) \tag{4.122}$$

$$\nu = \xi v^k + h^{kk} \left((v\bar{v}) + \eta (v^2 - v\bar{v}) \right) \tag{4.123}$$

we will get:

$$B\hat{e} = (Be)\frac{B^k}{a^k} - B^2 \frac{e^k}{a^k} - e^8 \frac{\mu}{a^k \xi} \tag{4.124}$$

$$v\hat{e} = (ve)\frac{B^k}{a^k} - Bv \frac{e^k}{a^k} - e^8 \frac{\nu}{a^k \xi} \tag{4.125}$$

At this point, we choose to use e^8 as our adjustable parameter for these vectors.

If we contract equation (4.71) with B^i we get the following

$$0 = (ve)\Gamma + (Be) h_e W^2 a^k + e^4 \Theta + e^k \Lambda + e^8 \frac{\Xi}{\xi} \tag{4.126}$$

with

$$\Gamma = -W^4\rho_0\gamma(Bv)a^k - W^2\rho_0\chi B^k + 2h_eW^4(Bv)a^k \tag{4.127}$$

$$\Theta = (\rho_0+\kappa)W^2(Bv)a^k + \kappa B^k \tag{4.128}$$

$$\Lambda = h_eW^2(Bv) - \rho_0W^2\gamma(Bv) - \chi\rho_0\frac{B^k}{a^k} \tag{4.129}$$

$$\Xi = -(Bv)\mu - B^2\nu + h^{kk}(Bv)^2 + h^{kk}\frac{B^2}{W^2} \tag{4.130}$$

If we contract the same equation with v^i we get

$$0 = (ve)\Delta + (Be)\zeta + e^4\Pi + e^k\Sigma + e^8\frac{\tau}{\xi} \tag{4.131}$$

with

$$\Delta = -W^4\rho_0\gamma v^2a^k - \chi W^2\rho_0v^k + Qa^k + 2h_eW^4v^2a^k + h_eW^2a^k \tag{4.132}$$

$$+\left[-B^2 + (Bv)\frac{B^k}{a^k}\right](v^k - a^k) - \frac{B^kB^k}{a^k} \tag{4.133}$$

$$\zeta = (v^k - a^k)\left[(Bv) + \frac{B^k}{W^2a^k}\right] + B^k \tag{4.134}$$

$$\Pi = (\rho_0+\kappa)W^2v^2a^k + \kappa v^k \tag{4.135}$$

$$\Sigma = \nu\left[(Bv) + \frac{B^k}{a^k} - \frac{Bv}{a^k}(v^k - a^k)\right] + \mu\left[-2v^2 - \frac{v^k}{W^2a^k}\right] + (Bv)h^{kk} \tag{4.136}$$

$$\tau = h_eW^2v^2 + \left[\frac{B^2}{W^2a^k} + \frac{(Bv)^2}{a^k}\right](a^k - v^k) + (Bv)\frac{B^k}{a^k} - B^2 - \rho_0W^2\gamma v^2 - \rho_0\chi\frac{v^k}{a^k} \tag{4.137}$$

Now we contract the same equation with h^{ik} and get:

$$0 = (ve)A + (Be)C + e^4D + e^kE + e^8\frac{F}{\xi} \tag{4.138}$$

with

$$\begin{aligned}
A &= -\rho_0 W^4 \gamma v^k a^k - \rho_0 W^2 \chi h^{kk} + 2h_e W^4 v^k a^k - h^{kk} B^2 \\
&\quad + B^k B^k \frac{a^k - v^k}{a^k} + h^{kk} \frac{(Bv)}{a^k} && (4.139)\\
C &= -B^k a^k + h^{kk}(Bv) + v^k B^k + \frac{B^k h^{kk}}{W^2 a^k} && (4.140)\\
D &= (\rho_0 + \kappa) W^2 v^k a^k + h^{kk} \kappa && (4.141)\\
E &= -\rho_0 W^2 \gamma v^k - \rho_0 \chi \frac{h^{kk}}{a^k} + Q v^k - (Bv) B^k + Q v^k \\
&\quad + (Bv) B^k \frac{v^k}{a^k} - h^{kk} \frac{(Bv)^2}{a^k} - 2B^2 v^k - \frac{B^2 h^{kk}}{W^2 a^k} && (4.142)\\
F &= h^{kk}(Bv) a^k + \frac{B^k h^{kk}}{W^2} + (Bv) h^{kk} v^k + \frac{B^k h^{kk}}{W^2} \\
&\quad + \nu \left[B^k + \frac{B^k v^k - (Bv) h^{kk}}{a^k} \right] - \mu \left[2v^k + \frac{h^{kk}}{a^k W^2} \right] && (4.143)
\end{aligned}$$

If we rewrite equation (4.72) we get:

$$0 = (ve) G + (Be) H + e^4 J + e^k M + e^8 \frac{N}{\xi} \tag{4.144}$$

with

$$\begin{aligned}
G &= (W^2 \gamma - W - \chi)(-W^2 \rho_0 a^k) - \chi W^2 \rho_0 v^k + (2h_e W^4 + B^2 - W^3 \rho_0) a^k \\
&\quad - B^2 v^k + \frac{(Bv) B^k}{a^k} (v^k - a^k) + \frac{B^k B^k}{a^k} && (4.145)\\
H &= (v^k - a^k) \left[(Bv) + \frac{B^k}{a^k W^2} \right] + B^k && (4.146)\\
J &= (\rho_0 + \kappa) W^2 a^k + \kappa (v^k - a^k) && (4.147)\\
M &= Q - W \rho_0 - 2B^2 + (Bv) \frac{B^k}{a^k} + (a^k - v^k) \left[\frac{W^2 (Bv)^2 + B^2}{a^k W^2} \right] \\
&\quad - \frac{\rho_0}{a^k} \left[(W^2 \gamma - W - \chi) a^k + \chi v^k \right] && (4.148)\\
N &= (Bv) h^{kk} + \frac{\nu}{a^k} [(Bv)(a^k - v^k) + B^k] - \mu \left[2 + \frac{v^k - a^k}{W^2 a^k} \right] && (4.149)
\end{aligned}$$

Our goal now is to combine the above four equations to eliminate e^4 and e^k and find solutions for Be and ve. Since we have used a good deal of the letters above and need to continue to do so, I

will use in this part lower case letters. We can write the following for *ve* and *Be*:

$$Be = -\frac{e^8}{\xi}\frac{rs-up}{qs-tp} \tag{4.150}$$

$$ve = \frac{e^8}{\xi}\frac{rt-qu}{qs-tp} \tag{4.151}$$

with

$$p = cA\Theta + lD + dE\Theta \quad q = cC\Theta + mD + fE\Theta \quad r = cF\Theta + nD + gE\Theta \tag{4.152}$$

$$s = cG\Theta + lJ + dM\Theta \quad t = cH\Theta + mJ + fM\Theta \quad u = cN\Theta + nJ + gM\Theta \tag{4.153}$$

$$c = \Theta\tau - \Lambda\Pi \qquad d = \Gamma\Pi - \Theta\Delta \qquad f = \Pi \cdot h_e W^2 a^k - \Theta\zeta \tag{4.154}$$

$$g = \Xi\Pi - \Sigma\Theta \qquad l = -c\Gamma - d\Lambda \qquad m = -c \cdot h_e W^2 a^k - f\Lambda \tag{4.155}$$

$$n = -g\Lambda - c\Xi \tag{4.156}$$

We can now collect where we are at and write down the components of our eigenvector:

$$\hat{e}^k = e^8\left[-\frac{h^{kk}}{\xi}\right] \tag{4.157}$$

$$e^4 = \frac{e^8}{c\Theta\xi T_2}\Upsilon \tag{4.158}$$

$$e^k = \frac{e^8}{c\xi T_2}\Phi \tag{4.159}$$

$$e^0 = \frac{e^8}{c\xi T_2}\Psi \tag{4.160}$$

with

$$\Upsilon = (rt-qu)l - (rs-up)m + T_2 n \tag{4.161}$$

$$\Phi = (rt-qu)d - (rs-up)f + T_2 g \tag{4.162}$$

$$\Psi = -(rt-qu)c \cdot W^2\rho_0 - \Phi\frac{\rho_0}{a^k} \tag{4.163}$$

$$T_2 = sq - tp \tag{4.164}$$

In order to obtain the remaining four components of the eigenvector, one must calculate the values of $B\hat{e}$ and $v\hat{e}$. We get:

$$B\hat{e} = \frac{e^8}{cT_2\xi}\aleph \tag{4.165}$$

$$v\hat{e} = \frac{e^8}{cT_2\xi}\beth \tag{4.166}$$

with

$$\aleph = -c(rs-up)\frac{B^k}{a^k} - \Phi\frac{B^2}{a^k} - \frac{cT_2\mu}{a^k} \tag{4.167}$$

$$\beth = c(rt-qu)\frac{B^k}{a^k} - \Phi\frac{Bv}{a^k} - \frac{cT_2\nu}{a^k} \tag{4.168}$$

We can now write the last four components:

$$\hat{e}^1_\perp = \frac{e^8}{c\xi T_1 T_2}\Omega \tag{4.169}$$

$$\hat{e}^1_\perp = \frac{e^8}{c\xi T_1 T_2}\delta \tag{4.170}$$

$$e^1 = \frac{e^8}{B^k c\xi T_1 T_2}\gimel \tag{4.171}$$

$$e^2 = \frac{e^8}{B^k c\xi T_1 T_2}\daleth \tag{4.172}$$

with

$$\Omega = v_2\aleph - B_2\beth + h^{kk}cT_2[v_2B^k - B_2v^k] \tag{4.173}$$

$$\delta = -v_1\aleph + B_1\beth + h^{kk}cT_2[-v_1B^k + B_1v^k] \tag{4.174}$$

$$\gimel = B_1T_1\Phi + a^k\Omega + cT_1T_2h^{kk}[\bar{v}_1 + \eta(v_1-\bar{v}_1)] \tag{4.175}$$

$$\daleth = B_2T_1\Phi + a^k\delta + cT_1T_2h^{kk}[\bar{v}_2 + \eta(v_2-\bar{v}_2)] \tag{4.176}$$

We now have complete eigenvectors for the ϕ waves.

4.7 The left eigenvectors

The left eigenvectors are calculated with a similar approach. It is sketched out here, but we only perform the construction of eigenvectors associated with the ϕ waves.

In order to find the left eigenvectors, we must consider the row vector:

$$\vec{l} = \begin{pmatrix} e^0 & e^i & e^4 & \hat{e}^i & e^8 \end{pmatrix} \tag{4.177}$$

and solve the system of equations

$$\vec{l}\mathscr{F}^k = \lambda^k \vec{l}\mathscr{F}^0 \tag{4.178}$$

Writing out this set of equations in component form, we get

$$0 = e^0[W\bar{a}^k] + (ve)[\gamma W^2\bar{a}^k] + e^k[\chi] + e^4[(\gamma W^2 - W - \chi)\bar{a}^k + \chi v^k] \tag{4.179}$$

$$\begin{aligned} 0 = {} & e^0[W\rho_o h_j^k + W^3\rho_o v_j\bar{a}^k] + (ve)[Qh_j^k + 2h_e W^4 v_j\bar{a}^k - B_j B^k] \\ & + (Be)[-(Bv)h_j^k + 2v_j B^k - B_j\bar{a}^k] + e_j[Q\bar{a}^k - (Bv)B^k] + e^k[(Bv)B_j - v_j B^2] \\ & + e^4[(2h_e W^4 v_j - (Bv)B_j + B^2 v_j - W^3 v_j\rho_o)\bar{a}^k + (Q - W\rho_o)h_j^k + ((Bv)B_j - B^2 v_j)v^k - B_j B^k] \\ & + \hat{e}_j[-B^k] + (B\hat{e})[h_j^k] \end{aligned} \tag{4.180}$$

$$0 = (ve)[(\rho_o + \kappa)W^2\bar{a}^k] + e^k[\kappa] + e^4[((\rho_o + \kappa)W^2 - \kappa)\bar{a}^k + \kappa v^k] \tag{4.181}$$

$$\begin{aligned} 0 = {} & (ve)[2B_j\bar{a}^k - v_j B^k - (Bv)h_j^k] + (Be)\left[-v_j\bar{a}^k - \frac{h_j^k}{W^2}\right] + e_j\left[-(Bv)\bar{a}^k - \frac{B^k}{W^2}\right] \\ & + e^4\left[\left(2B_j - (Bv)v_j - \frac{B_j}{W^2}\right)\bar{a}^k + v^k\left((Bv)v_j + \frac{B_j}{W^2}\right) - (Bv)h_j^k - v_j B^k\right] + \hat{e}_j\bar{a}^k \\ & + h_j^k[\eta(\bar{v}\hat{e} - v\hat{e}) - \bar{v}\hat{e}] + e^8 h_j^k + e^k\left[(Bv)v_j + \frac{B_j}{W^2}\right] \end{aligned} \tag{4.182}$$

$$0 = \hat{e}^k + e^8[\bar{a}^k - v^k] \tag{4.183}$$

The structure of these equations is simular to the equations that are used to compute the right eigenvector components. However, because our system is not symmetric, they require us to start over with our solutions for the different types of waves.

4.7.1 Divergence Waves

For the ϕ waves, we could go through the rather lengthy calculations to show that

$$e^0 = e^i = e^4 = \hat{e}^1_\perp = \hat{e}^2_\perp = 0 \tag{4.184}$$

or we could follow a much easier path. If we look at the matrix whose columns are the right eigenvectors of the system we notice that due to the zeros in the $\hat{e}^k_\perp$ and e^8 positions, the matrix has a lower 2×2 block. If we were to invert this, we would get the same structure. Because the left eigenvectors are the inverse of this matrix, we can conclude that for the ϕ waves, the above holds true and we only need to find a relation between $\hat{e}^k_\perp$ and e^8. To find this relation, we simply modify equation (4.183) to get:

$$\hat{e}^k = -e^8[\bar{a}^k - v^k]. \tag{4.185}$$

This is the left eigenvector for the ϕ waves. There are no fluid components involved in this eigenvector.

4.8 The Boundaries

As decribed previously, for us to construct the boundary condition associated with the field ϕ, we need to left eigenvectors that we just computed. We combine these left eigenvectors as describes in equation 4.11. We write out the left eigenvectors explicitly as

$$\vec{l}_- = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{-\sqrt{h^{kk}}}{2} \end{pmatrix} \tag{4.186}$$

$$\vec{l}_+ = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{\sqrt{h^{kk}}}{2} \end{pmatrix}. \tag{4.187}$$

Using these in the equation 4.11 gives us an equation for the boundary condition

$$(1+a)B^k = (1-a)\sqrt{h^{kk}}\phi. \tag{4.188}$$

Solving for ϕ we get

$$\phi = \frac{1+a}{1-a}\frac{B^k}{\sqrt{h^{kk}}}. \tag{4.189}$$

If we choose to use maximally dissipative boundary conditions thereby supressing the incoming wave, we set $a = 0$ and get

$$\phi = \frac{B^k}{\sqrt{h^{kk}}}. \tag{4.190}$$

We now have the necessary boundary condition for ϕ. We can simply set the value of ϕ as perscribed by this equation to eliminate reflections off of the boundary.

4.9 Conclusion

The characteristic decomposition of the GRMHD equations was completed in [22,23]. The decomposition work we have done here while important, is not our specific goal in section. We choose to evolve the system of equations using the conserved variables. We are particularly interested is the boundary condition for the ϕ wave as shown in Figure 4.1. This is because while we can select any method that we choose to use for the evolution we still need to have a good boundary condition for the new field.

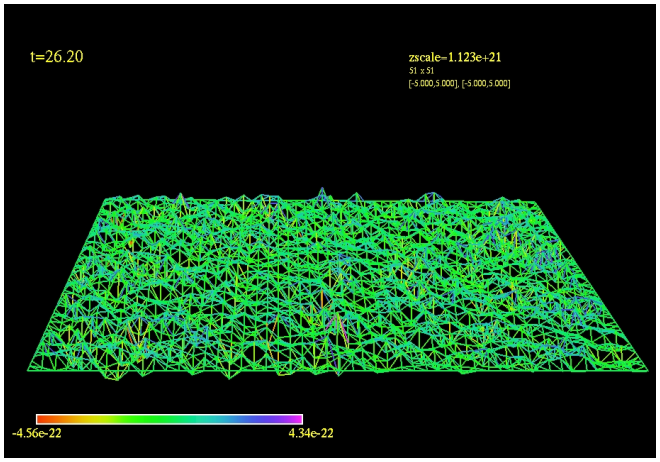

Figure 4.1 This is the divergence cleaning field ϕ after running an evolition for a long time We can see that the error is well below truncation error of the evolution code.

Chapter 5

Weighted Essentially Non-Oscillatory (WENO) evolution

5.1 Overview

The equations of magnetohydrodynamics (MHD) admit weak solutions. In other words, shocks will be produced from smooth initial data. These shocks can be handled analytically. However, they become difficult to deal with respect to certain computational approaches. This is because these approaches assume smoothness of the fields. These techniques break down when there is a discontinuity or shock in the fields. There have been many attempts at addressing the issues of shocks in computational models. For example, some models use a method known as artificial viscosity [24]. Artificial viscosity has the effect of smoothing out discontinuities, sometimes to the point where they may not even be formed in an evolution.

We wish to implement a system in which we can preserve and follow a shock through the computational domain. We also want our scheme to have the ability to have a high order of accuracy away from any shock. One system, known as ENO (essentially non-oscillatory) has been put

forth as one method which satisfies the first requirement [25]. The virtue of this scheme, as given by its name, is the ability to track the shock without producing high frequency oscillations in the vicinity of the shock. However, the accuracy drops off very quickly in regions near such a shock. We have chosen to implement an improvement on ENO called WENO (weighted essentially non-oscillatory). WENO improves the accuracy around the shock while still maintaining the virtues of ENO. Using WENO we are able to maintain high order accuracy without increasing the number of grid points necessary to perform the computation.

As a description of WENO we assume a continuum toy problem described below. Our first step is to take the continuum where our model exists and discretize it along points in out computational domain. These points will be spaced some distance apart and represent locations where our functions will be evolved in both space and time. On this grid we want to implement WENO. This overreaching methodology is used to determine the derivative of a function at a point labeled i by calculating the values of the function at points $i \pm \frac{1}{2}$. To do this we will use k grid points in the vicinity of i. In smooth regions the accuracy of the WENO model is expected to be of order $(\Delta x)^{2k-1}$ where Δx is the distance between computational points.

For our grid we will use the notation of [26]. We define cells I_i, and cell centers x_i by

$$I_i \equiv \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right] \tag{5.1}$$

$$x_i \equiv \frac{1}{2}\left(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}\right) \tag{5.2}$$

$$i = 1, 2, ..., N. \tag{5.3}$$

With a fixed grid our cell sizes are designated by Δx.

We wish to apply this to a simple toy problem namely the linear advection equation given by

$$\partial_t v = -c\partial_x v \tag{5.4}$$

$$v(x,0) = v_o(x). \tag{5.5}$$

We want to approximate the continuum function $v(x,t)$ at a particular time t over the grid points

x_i. $v_o(x)$ is the initial profile of our propagating wave and c is the right moving wave speed. We suppress the explicit time dependence t in our notation but realize that this is indeed an evolution in time that we are approximating. In order to do a numerical evolution of such a system, we can populate the above grid with the values of $v(x_i,0)$. For each step of the evolution we need to then calculate the derivative of the function at those grid points. In order to do this we therefore need to know the values at the half points $x_{i\pm\frac{1}{2}}$. The goal is to create interpolation polynomials that will approximate the values of the field at those half points.

Due to the nonlinear nature of the fluid equations in general relativistic magnetohydrodynamics (GRMHD), shocks may be formed in the field v during the evolution. While this can be dealt with on an analytical level, shocks lead to computational problems due to the large discontinuities that are formed on the computational grid. We wish to use WENO to track and evolve shocks through the system while maintaining high order accuracy away from the shock.

Using the WENO method we will consider a number of points over which to compute the interpolation of the function v to the half points between our gridpoints. We have to do the interpolation around a specific point i. We first select the number of points k such that the accuracy of our WENO scheme will be of order $2k-1$. Our collection of stencils will range over $2k$ points from $i-(k-1)$ to $i+k$. For example, if we select $k=3$, the points that we will use will be $(i-2,i-1,i,i+1,i+2,i+3)$. We will consider the complete set of substencils with consecutive points of size k. Figure 5.1 shows the substencils for a system where $k=3$. We calculate an interpolation polynomial for each substencil to obtain several approximations for the value of the field v at the half points. For each substencil we will compute an interpolation polynomial to find the value of $v(x_{i+\frac{1}{2}})$. The final approximation of the value of the field at the half point will be done with a weighted average of the interpolation polynomials from the substencils that do not contain any discontinuities.

Our goal is to determine the derivative of the function $v(x)$ at a particular grid point i. In order

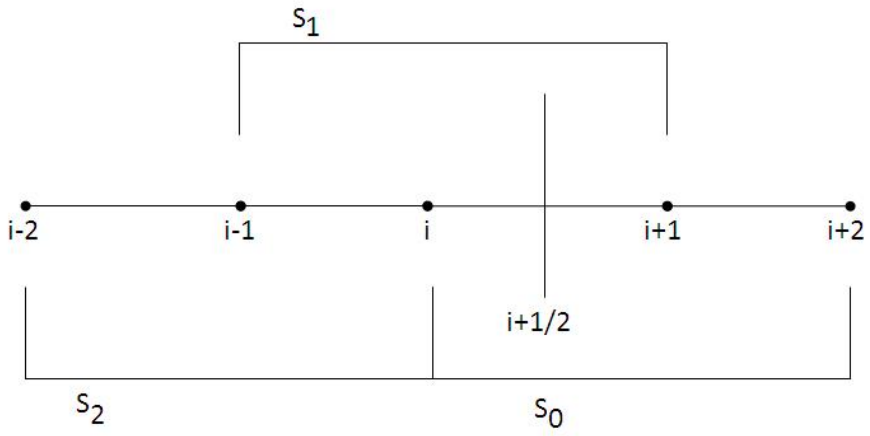

Figure 5.1 Stencil and associated substencils for a WENO system where $k = 3$. These are the substencils that are used to find the flux at the position $x_{i+1/2}$ for a right moving wave.

to do this, one needs a good approximation around that point. This requires us to get a good approximation inside the cell I_i that contains our grid point i. We define the sliding average $\bar{v}_i$ to be

$$\bar{v}_i \equiv \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(\xi)\, d\xi. \tag{5.6}$$

This sliding average is the value that we shall track in our evolution. Approximating the derivative of $\bar{v}(x)$ with respect to x we get

$$\partial_x \bar{v}(x_i) \approx \frac{1}{\Delta x}[v(x_{i+\frac{1}{2}}) - v(x_{i-\frac{1}{2}})]. \tag{5.7}$$

We can now put this into equation (5.4)

$$\partial_t \bar{v} \approx -\frac{c}{\Delta x}[v(x_{i+\frac{1}{2}}) - v(x_{i-\frac{1}{2}})]. \tag{5.8}$$

Now that we have a system and understand what we want to calculate we now need to construct the interpolation polynomials.

5.2 Construction of polynomials for a specific stencil

We want to find a polynomial, $p_s(x)$, on the substencil of degree at most $k-1$ such that inside the interval I_i, $p_s(x)$ is a good interpolation of $v(x)$. The subscript s is to indicate that this polynomial is specific to a particular substencil labeled s. The polynomial is an order $k-1$ approximation of $v(x)$. We will compare different approximation polynomials around a specific location x_i. Therefore i will be fixed in this discussion. We define r as the number of points to the left of point i. We can therefore uniquely specify a particular polynomial only by the r value for the specific substencil from the group of substencils associated with the point i. We will call this polynomial $p_r(x)$ where r is the number of points to the left of i. For the following discussion, the subscript is not needed and we will drop it for the time being.

An interpolation polynomial of degree $k-1$ takes the form $p(x_i) = v(x_i) + O(\Delta x^k)$. We are particularly interested in the interpolation at the end points of our cells I, namely

$$p^R_{i+\frac{1}{2}} = v(x_{i+\frac{1}{2}}) + O(\Delta x^k) \tag{5.9}$$

$$p^L_{i-\frac{1}{2}} = v(x_{i-\frac{1}{2}}) + O(\Delta x^k) \tag{5.10}$$

where R designates the function on the right boundary of the cell and L is the function on the left boundary of the cell.

We will now define a substencil upon which we will calculate the above polynomials. We now work with substencils that have points to the left and right of i. For clarity, we have i which is the point around which we are calculating. Again, r is the number of points to the left of i for a stencil. s is the number of points to the right of i for a stencil. This gives an equation for k as $k = r+s+1$. We can label a particular stencil for a given k, i, and r. Our substencil will cover the following range

$$S_{i,r} \equiv \{I_{i-r}, ..., I_{i+k-r-1}\} \tag{5.11}$$

That is, a substencil will cover a range of k intervals around the point i. Our interpolation polynomial is therefore a unique polynomial, $p(x)$, of degree at most $k-1$. Such that

$$\frac{1}{\Delta x}\int_{I_n} p(\xi)\,d\xi = \bar{v}_{I_n} \tag{5.12}$$

$$n = i-r,\ldots,i-r+k-1. \tag{5.13}$$

Thus the integral of p over the cell gives us our sliding average over the cell.

We want to interpolate the value of the function at the cell boundaries. To do this, we need to consider flow both to the left and right across the cell boundaries at the half points. To calculate the derivative at point i, we need the value of the function at the boundaries $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$. We are calculating these values just inside the cell boundaries and we shall designate the left and right side of the boundaries of a single cell with L and R respectively. As a key part of the approximation we assume that we have a linear relation between the cell averages within a stencil to the values of v at the cell boundaries. To effect this approximation we assume that there are constants $c_{r,j}$ and $\tilde{c}_{r,j}$ such that:

$$p^L_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{r,j}\bar{v}_{i-r+j} \tag{5.14}$$

$$p^R_{i-\frac{1}{2}} = \sum_{j=0}^{k-1} \tilde{c}_{r,j}\bar{v}_{i-r+j}. \tag{5.15}$$

To calculate our function at the cell boundaries, we need $c_{r,j}$ and $\tilde{c}_{r,j}$.

In order to caculate $c_{r,j}$ and $\tilde{c}_{r,j}$ we consider two substencils, namely $S_{i,r}$ and $S_{i+1,r+1}$. It can easily be seen that these two substencils cover the same points. Because the interpolation polynomial is unique for a given substencil, we can equate the v^L and v^R at a point and thereby equate the constants in the following way:

$$\tilde{c}_{r,j} = c_{r-1,j} \tag{5.16}$$

As long as we can find the above constants, we can interpolate the values of the function at the half way points

$$p_{i+\frac{1}{2}} = \sum_{j=0}^{k-1} c_{r,j} \bar{v}_{i-r+j} \tag{5.17}$$

with the order of accuracy given by

$$p_{i+\frac{1}{2}} = v(x_{i+\frac{1}{2}}) + O(\Delta x^k). \tag{5.18}$$

We will now define P as the derivative of the interpolation polynomial and V as the integral of the continuum function $v(x)$

$$P(x) \equiv p'(x) \tag{5.19}$$

$$V(x) \equiv \int_{-\infty}^{x} v(\xi)\, d\xi. \tag{5.20}$$

We can approximate V using the cell averages

$$V(x_{i+\frac{1}{2}}) = \sum_{j=-\infty}^{i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(\xi)\, d\xi \approx \sum_{j=-\infty}^{i} \bar{v}_j \Delta x_j. \tag{5.21}$$

Knowing the cell averages, gives us an approximation of $V(x)$ at the cell boundaries. We can now verify equation (5.12):

$$\begin{aligned}
\frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} P(\xi)\, d\xi &= \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p'(\xi)\, d\xi && (5.22)\\
&= \frac{1}{\Delta x} \left(p(x_{j+\frac{1}{2}}) - p(x_{j-\frac{1}{2}}) \right) && (5.23)\\
&= \frac{1}{\Delta x} \left(V(x_{j+\frac{1}{2}}) - V(x_{j-\frac{1}{2}}) \right) && (5.24)\\
&= \frac{1}{\Delta x} \left(\int_{-\infty}^{x_{j+\frac{1}{2}}} v(\xi)\, d\xi - \int_{-\infty}^{x_{j-\frac{1}{2}}} v(\xi)\, d\xi \right) && (5.25)\\
&= \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(\xi)\, d\xi && (5.26)\\
&= \bar{v}_j && (5.27)
\end{aligned}$$

The third equality is because $p(x)$ interpolates $V(x)$ at the points $x_{j\pm\frac{1}{2}}$ when $j = i-r,...,i-r+k-1$. Thus $p(x)$ is the polynomial we are looking for. So we have:

$$p'(x) = V'(x) + O(\Delta x^k), \qquad x \in I_i \tag{5.28}$$

Now we are finally able to construct the constants $c_{r,j}$. To do this we use the Lagrange form of the interpolation polynomial

$$p(x) = \sum_{j=1}^{n} p_j(x) \tag{5.29}$$

$$p_j(x) = y_j \prod_{\substack{k=1 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k} \tag{5.30}$$

where y_j is the value of the function at the grid points. We write this for our system as

$$p(x) = \sum_{j=1}^{n} p_j(x) \tag{5.31}$$

$$p_j(x) = V(x_{i-r+m-\frac{1}{2}}) \prod_{\substack{l=0 \\ l\neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}}. \tag{5.32}$$

All we have done is changed notation and reindexed our system. We can write this as a single equation

$$p(x) = \sum_{m=0}^{k} V(x_{i-r+m-\frac{1}{2}}) \prod_{\substack{l=0 \\ l\neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}}. \tag{5.33}$$

Our goal is to get this into the form of equation (5.17) so that we can compare and determine the values of the coefficients $c_{r,j}$.

Note that we are trying to interpolate to the point $x_{i+\frac{1}{2}}$ and in that case it can easily be show that

$$\sum_{m=0}^{k} \prod_{\substack{l=0 \\ l\neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}} = 1. \tag{5.34}$$

We subtract $V(x_{i-r-\frac{1}{2}})$ from the above equation multiplied by this particular version of 1 from the above equation to get

$$p(x) - V(x_{i-r-\frac{1}{2}}) = \sum_{m=0}^{k} \left(V(x_{i-r+m-\frac{1}{2}}) - V(x_{i-r-\frac{1}{2}}) \right) \prod_{\substack{l=0 \\ l \neq m}}^{k} \frac{x - x_{i-r+l-\frac{1}{2}}}{x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}}} \tag{5.35}$$

Taking the derivative of the above equation with respect to x and noticing that

$$V(x_{i-r+m-\frac{1}{2}}) - V(x_{i-r-\frac{1}{2}}) = \sum_{j=0}^{m-1} \bar{v}_{i-r+j} \Delta x_{i-r+j} \tag{5.36}$$

we get:

$$P(x) = \sum_{m=0}^{k} \sum_{j=0}^{m-1} \bar{v}_{i-r+j} \Delta x_{i-r+j} \left(\frac{\sum_{\substack{l=0 \\ l \neq m}}^{k} \prod_{\substack{q=0 \\ q \neq m,l}}^{k} \left(x - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l \neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \tag{5.37}$$

We are interested in the values at the cell boundaries we therefore evaluate the above at $x = x_{i+\frac{1}{2}}$ to obtain:

$$v_{i+\frac{1}{2}} = P(x_{i+\frac{1}{2}}) = \sum_{j=0}^{k-1} \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l \neq m}}^{k} \prod_{\substack{q=0 \\ q \neq m,l}}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l \neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \bar{v}_{i-r+j} \tag{5.38}$$

We can compare this to equation(5.17) to see that $c_{r,j}$ is given by

$$c_{r,j} = \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l \neq m}}^{k} \prod_{\substack{q=0 \\ q \neq m,l}}^{k} \left(x_{i+\frac{1}{2}} - x_{i-r+q-\frac{1}{2}} \right)}{\prod_{\substack{l=0 \\ l \neq m}}^{k} \left(x_{i-r+m-\frac{1}{2}} - x_{i-r+l-\frac{1}{2}} \right)} \right) \Delta x_{i-r+j} \tag{5.39}$$

Because we have a uniform grid, i.e. $\Delta x_i = \Delta x$, the expression for $c_{r,j}$ will no longer depend on Δx or i and can be simplified to:

$$c_{r,j} = \sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l\neq m}}^{k} \prod_{\substack{q=0 \\ q\neq m,l}}^{k} (r-q+1)}{\prod_{\substack{l=0 \\ l\neq m}}^{k} (m-l)}. \tag{5.40}$$

With the constants we now have polynomials $p_r(x)$ that will allow us to interpolate $v_{i+\frac{1}{2}}$ up to any order accuracy that we want by selecting the size of our overall stencil.

5.3 Calculating Undivided Newtonian Differences, Smoothness and Weights

Now that we have obtained the interpolation polynomials, we now need to find which, if any, of the substencils contain a discontinuity. A discontinuity in a substencil will be indicated in a large change in the grid function across that substencil when compared to the other substencils. We need a measure of the changes across the substencil. We use what is called an undivided Newtonian difference. This is a simplified approximation of a derivative across the substencil. For the undivided Newtonian difference we use the operator Δ. The simplest form of the difference is defined across two points as

$$\Delta[v_l] \equiv v_{l+1} - v_l. \tag{5.41}$$

This is the numerator of a simple derivative approximation across two points. We can have higher orders of this operator by using the recursive relationship of the operator which is defined as

$$\Delta^k[v_l] \equiv \Delta^{k-1}[v_{l+1}] - \Delta^{k-1}[v_l] \tag{5.42}$$

For substencils with more than two points, the following can easily be calculated. For $k = 2$ we have

$$\Delta^2 [v_l] \quad = \quad u_{l+2} - 2u_{l+1} + u_l \tag{5.43}$$

which is the numerator for a second derivative approximation. For $k = 3$ we have

$$\Delta^3 [v_l] \quad = \quad u_{l+3} - 3u_{l+2} + 3u_{l+1} - u_l \tag{5.44}$$

which is likewise the numerator for a third derivative appriximation. Because these will be calculated for each substencil we can then compare them to determine if there is indeed a discontinuity in the overall stencil. Knowing which of the substencils contain the discontinuity, we can weight the polynomials associated with the other substencils to use those over the ones with the discontinuity.

For each substencil we combine the undivided differences from order 1 to order k. We can then compare these in order to determine if there is a discontinuity over the stencil. We call this the smoothness function across the substencil. We expect that in the absence of any discontunity in the stencil, the smoothness function will be the same across all substencils. However, in the presence of a discontinuity one or more of the smoothness functions will be significantly larger than the others. To define the notion of smoothness, we consider a sum of undivided differences squared. In effect this is a norm of undivided differences. For WENO this can be written [27]

$$\mathfrak{s}_{i,r} = \sum_{l=1}^{k-1} \sum_{m=1}^{l} \frac{\left(\Delta^{k-l} [u_{i-r-1+m}]\right)^2}{l}, \tag{5.45}$$

where $\mathfrak{s}_{i,r}$ is the smoothness associated with the interpolation polynomial $p_{i,r}(x)$. For a particular order k, we have multiple terms that can be roughly thought of as 1^{st}, 2^{nd} and higher order derivatives which are squared and summed. In the presence of a discontinuity one or more of these will be large in comparison to the others. However if there is no discontinuity we expect all of the derivatives to be approximatly the same and therefore we expect the same for their associated smoothness functions.

If we have a system where for $k = 2$, the smoothness functions are

$$\begin{aligned} \mathfrak{s}_{i,r} &= (\Delta[v_{i-r}])^2 && (5.46)\\ &= (v_{i-r+1} - v_{i-r})^2 && (5.47)\\ &= v_{i-r+1}^2 - 2v_{i-r}v_{i-r+1} - v_{i-r}^2. && (5.48) \end{aligned}$$

For a system where $k = 3$, the smoothness function becomes

$$\begin{aligned} \mathfrak{s}_{i,r} &= ((\Delta[v_{j-2}])^2 + (\Delta[v_{j-1}])^2)/2 + (\Delta^2[v_{j-2}])^2 && (5.49)\\ &= \frac{1}{2}(2v_{i-r+1}^2 + v_{i-r}^2 - 2v_{i-r}v_{i-r+1} + v_{i-r+2}^2 - 2v_{i-r+2}v_{i-r+1}) \\ &\quad +(v_{i-r+2} - 2v_{i-r-1} + v_{i-r})^2. && (5.50) \end{aligned}$$

The smoothness function gives us a measure of which substencils are more or less likely to have a discontinuity. If we were using an ENO scheme, we simply choose the polynomial associated with the substencil that has the lowest smoothness function and we are finished. However, in a WENO system, we want to use all of the polynomials and weight them to consider a subset of the stencils with the smallest smoothness. This weighting in WENO gives us the desired ability to evolve and track discontinuities as well as the higher accuracy away from discontinuities. With the smoothness functions, we are almost ready to construct the overall interpolation polynomial which we will label $\mathfrak{P}_i(x)$. The last step is to calculates weights ω_r for our interpolation polynomials $p_r(x)$ and construct the final polynomial as

$$\mathfrak{P}_i(x) = \sum_{r=-1}^{k-1} \omega_r p_r(x). \tag{5.51}$$

These weights are calculated from the ENO property labeled α_r by

$$\omega_r = \frac{\alpha_r}{\sum_{l=-1}^{k-1} \alpha_l}. \tag{5.52}$$

This ensures that the weights add up to 1. The ENO property α_r has the following conditions:

(i) If p_r is in a smooth regions we have:

$$\frac{\alpha_r}{\sum_{l=-1}^{k-1} \alpha_l} = O(1) \tag{5.53}$$

(ii) If p_r is in a discontinuous region, we have:

$$\frac{\alpha_r}{\sum_{l=-1}^{k-1} \alpha_l} = O(\Delta x^k) \tag{5.54}$$

To accomplish this, we define α_r as

$$\alpha_r = \frac{C_r}{(\varepsilon + \mathfrak{s}_r)^k} \tag{5.55}$$

We will discuss the constants C_r in a moment. ε is a tolerance to ensure that α does not blow up when the smoothness function is 0. It can be seen that in the absence of a discontinuity, we have for the ENO property

$$\frac{\alpha_r}{\sum_{l=-1}^{k-1} \alpha_l} = O(1). \tag{5.56}$$

Additionally we can see that if p_r is in a discontinuous region, we would have $\mathfrak{s}_r$ be large and therefore we would have for that particular ENO property α_r,

$$\frac{\alpha_r}{\sum_{l=-1}^{k-1} \alpha_l} \leq \max(O(\varepsilon), O(\Delta x^{2k}). \tag{5.57}$$

So we have the ENO properties as stated above. We now need to determine the values of the constants C_r in equation (5.55). There is a construction of these constants that is described in [27]. However there is an easier way to construct these constants. Consider a region where all of the smoothness functions are identical. Then when we normalize the weights with equation (5.52) the denominators of the αs will be divided away. This will leave us with just the constants which will

add to 1. In this case we want the final polynomial $\mathfrak{P}_i(x)$ to be the interpolation polynomial that is of order $2k-1$. We therefore select constants C_r such that when we combine the interpolation polynomials in equation (5.51) we compute the larger interpolation polynomial. We can simply compare the polynomials across the substencils and the higher order polynomial to construct the necessary values for C_r. If we select $k=3$, then we compare the above calculated interpolation polynomials on the substencils with the 5^{th} order interpolation polynomial. In this case for a right moving wave we get

$$C_2 = \frac{1}{10} \tag{5.58}$$
$$C_1 = \frac{6}{10} \tag{5.59}$$
$$C_0 = \frac{3}{10} \tag{5.60}$$
$$C_{-1} = 0. \tag{5.61}$$

Likewise for a left moving wave we get

$$C_{-1} = \frac{1}{10} \tag{5.62}$$
$$C_0 = \frac{6}{10} \tag{5.63}$$
$$C_1 = \frac{3}{10} \tag{5.64}$$
$$C_2 = 0. \tag{5.65}$$

We can now use the above interpolations to approximate the values of v at the half points and calculate the derivative of our function at the point x_i with

$$\partial_x v(x_i) = \frac{v\left(x_{i+\frac{1}{2}}\right) - v\left(x_{i-\frac{1}{2}}\right)}{\Delta x}. \tag{5.66}$$

With all of these steps completed, we now have a WENO system that is order $2k-1$ accurate away from discontinuities. While we lose accuracy around discontinuities, we can evolve and track them through the system.

5.4 Evolution of the 1-D advection equation and the wave equation with WENO

Our goal here is to fully implement a code that implements WENO. We want a 5^{th} order system so we will set $k = 3$. For the initial code, we will implement periodic boundary conditions. This eliminates some of the complication of the evolution, but will need to be relaxed for more complicated systems.

First to make the subsequent equations more tractable, we will label the grid points starting at $i-2$

$$a \equiv \bar{v}_{i-2} \tag{5.67}$$

$$b \equiv \bar{v}_{i-1} \tag{5.68}$$

$$c \equiv \bar{v}_i \tag{5.69}$$

$$d \equiv \bar{v}_{i+1} \tag{5.70}$$

$$e \equiv \bar{v}_{i+2} \tag{5.71}$$

$$f \equiv \bar{v}_{i+3}. \tag{5.72}$$

Using these labels, we can write down the polynomials p_r from equation (5.17) and calculate the constants from equation (5.40) for a given stencil. This gives

$$p_2 = \frac{1}{3}a - \frac{7}{6}b + \frac{11}{6}c \tag{5.73}$$

$$p_1 = -\frac{1}{6}b + \frac{5}{6}c + \frac{1}{3}d \tag{5.74}$$

$$p_0 = \frac{1}{3}c + \frac{5}{6}d - \frac{1}{6}e \tag{5.75}$$

$$p_{-1} = \frac{11}{6}d - \frac{7}{6}e + \frac{1}{3}f. \tag{5.76}$$

We also write done the smoothness functions $\mathfrak{s}_r$ for each as follows:

$$\mathfrak{s}_2 = (c-2b+a)^2+\frac{1}{2}(a^2+2b^2+c^2-2b(a+c)) \tag{5.77}$$

$$\mathfrak{s}_1 = (d-2c+b)^2+\frac{1}{2}(b^2+2c^2+d^2-2c(b+d)) \tag{5.78}$$

$$\mathfrak{s}_0 = (e-2d+c)^2+\frac{1}{2}(c^2+2d^2+e^2-2d(c+e)) \tag{5.79}$$

$$\mathfrak{s}_{-1} = (f-2e+d)^2+\frac{1}{2}(d^2+2e^2+f^2-2e(d+f)). \tag{5.80}$$

The ENO properties α_r depend on the direction of the flow of the wave. The weights for a right moving wave are

$$\alpha_2 = \frac{1}{10(\varepsilon+\mathfrak{s}_2)^3} \tag{5.81}$$

$$\alpha_1 = \frac{6}{10(\varepsilon+\mathfrak{s}_1)^3} \tag{5.82}$$

$$\alpha_0 = \frac{3}{10(\varepsilon+\mathfrak{s}_0)^3} \tag{5.83}$$

$$\alpha_{-1} = 0 \tag{5.84}$$

and for a left moving wave we get:

$$\alpha_{-1} = \frac{1}{10(\varepsilon+\mathfrak{s}_{-1})^3} \tag{5.85}$$

$$\alpha_0 = \frac{6}{10(\varepsilon+\mathfrak{s}_0)^3} \tag{5.86}$$

$$\alpha_1 = \frac{3}{10(\varepsilon+\mathfrak{s}_1)^3} \tag{5.87}$$

$$\alpha_2 = 0 \tag{5.88}$$

We ran these evolutions for the advection equation using a Gaussian for the wave and the system is 5^{th} order accurate. We evolved a square wave and found that while the evolution was less accurate at the discontinuity and there was some dissipation of the discontinuity at the beginning of the evolution as expected, we were able to track its motion during the evolution.

Because our system of equations is more complicated than the advection equation we needed to test it on a more complicated system. We decomposed the wave equation as discussed in the

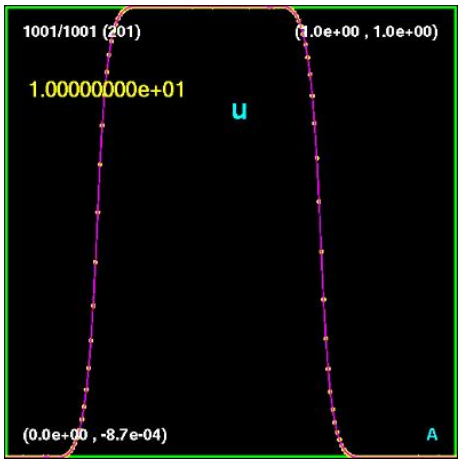

Figure 5.2 After running a square wave several times though our evolution domain with periodic boundary condition. We can see some dissipation that is expected. However, the form and location of the discontinuity can be seen quite well.

previous chapter starting at equation (4.12). We used the same code to evolve both of the resulting advection equations simultaneously. We did this in three dimensions and had very similar examples as we did with the single advection equation in one dimension.

5.5 Results

We have fully evolved the wave equation using a WENO system and periodic boundary conditions. This was done with several wave forms including a square wave and triangle wave. The end result of the square wave can be seen in Figure 5.2. We have tested the system and believe that the system can be applied to our full system of equations in the evolution. However, this implementation was not completed due to the fact that we need both of the complete sets of the left and right eigenvectors as discussed in Chapter 4. While this has the possibility to resolve some issues in the current evolution code, it will not lead to new science.

Chapter 6

Conclusion and Further Work

In Chapter 2, we were able to show that implementation of additional solvers were not effective in improving the accuracy of the code. The use of two equations solvers did not decrease the failure rate of the conversion between variables. However, there is still work that can be accomplished. The fact that Maple could indeed find solutions where our code could not is concerning. One could work on implementing solvers that have the ability to adjust their precision when a solution is difficult to find.

For initial data in Chapter 3, we were able to convert the data that is generated in the initial data code into the evolution code. Additionally we created a methodology to have single files that contained all of the data for a particular star. There is more work that needs to be completed. We need to be able to generate a star with differential rotation, convection, poloidal and toridal magnetic fields. As mentioned, there is also work to be done that would allow the rotational and magnetic axes to be misaligned.

In the setting up of the divergence cleaning in Chapter 4, we were able to add in the additional field into our evolution equations. We were able to find the boundary conditions for this new field. One can continue this work by finding the left eigenvectors for this system. This would be particularly helpful for creating a WENO system for the evolution equations.

Finally we were able to create a completely working WENO system in Chapter 5. We have tested this system for the three dimensional wave equation to show that it indeed performs as expected. On calculating the left eignevectors, we could implement this system and evolve the complete system of MHD equations.

Bibliography

[1] J. Chadwick, "Bakerian Lecture. The Neutron," Royal Society of London Proceedings Series A **142,** 1–25 (1933).

[2] W. Baade and F. Zwicky, "Remarks on Super-Novae and Cosmic Rays," Phys. Rev. **46,** 76–77 (1934).

[3] A. Hewish and S. E. Okoye, "Evidence for an Unusual Source of High Radio Brightness Temperature in the Crab Nebula," Nature **207,** 59–60 (1965).

[4] J. M. Weisberg, J. H. Taylor, and L. A. Fowler, "Gravitational waves from an orbiting pulsar," Scientific American **245,** 74–82 (1981).

[5] P. B. Demorest, T. Pennucci, S. M. Ransom, M. S. E. Roberts, and J. W. T. Hessels, "A two-solar-mass neutron star measured using Shapiro delay," Nature **467,** 1081–1083 (2010).

[6] H. Alfven, "Existence of electromagnetic-hydrodynamic waves," Nature 150 (1942).

[7] S. C. Noble, "Primitive Variable Solvers for Conservative General Relativistic Magnetohydrodynamics," Astrophysical Journal **641,** 626–637 (2006).

[8] J. D. Anderson, *Computational Fluid Dynamics, The Basics With Applications* (McGraw Hill, 1995).

[9] http://www.lorene.obspm.fr/.

[10] R. Ciolfi, S. K. Lander, G. M. Manca, and L. Rezzolla, "Instability-driven evolution of poloidal magnetic fields in relativistic stars," Astrophysical Journal Letters 736 (2011).

[11] K. Kiuchi, S. Yoshida, and Y. Eriguchi, "Equilibrium configurations of magnetized rotating polytropes: Effects of strong toroidal magnetic fields in addition to poloidal magnetic fields," The Astrophysical Journal Supplement Series **164,** 156–172 (2006).

[12] S. Bonazzola, E. Gourgoulhon, and J. Marck, "Numerical approach for high precision 3-D relativistic star models," Physical Review D 58 (1998).

[13] S. Chawla, M. Anderson, M. Besselman, L. Lehner, S. L. Liebling, P. M. Motl, and D. Neilsen, "Mergers of Magnetized Neutron Stars with Spinning Black Holes: Disruption, Accretion, and Fallback," Phys. Rev. Lett. **105,** 111101 (2010).

[14] C. Palenzuela, M. Anderson, L. Lehner, S. L. Liebling, and D. Neilsen, "Binary Black Holes' Effects on Electromagnetic Fields," Phys. Rev. Lett. **103,** 081101 (2009).

[15] R. C. Tolman, "Static Solutions of Einstein's Field Equations for Spheres of Fluid," Phys. Rev. **55,** 364–373 (1939).

[16] E. Hirschmann, notes on GRMHD (unpublished).

[17] T. W. Baumgarte and S. L. Shapiro, "Numerical integration of Einstein's field equations," Phys. Rev. D **59,** 024007 (1998).

[18] G. Tóth, "The $\nabla \cdot B=0$ Constraint in Shock-Capturing Magnetohydrodynamics Codes," Journal of Computational Physics 161 (2000).

[19] M. Brio and C. Wu, "An upwind differencing scheme for the equations of ideal magnetohydrodynamics," Journal of Computational Physics **75,** 400–422 (1988).

[20] M. Cecere, L. Lehner, and O. Reula, "Constraint preserving boundary conditions for the Ideal Newtonian MHD equations," Computer Physics Communications **179,** 545–554 .

[21] M. Alcubierre, *Introduction to 3+1 Numerical Relativity* (Oxford University Press, 2008).

[22] L. Anton, J. A. Miralles, J. M. Marti, J. M. Ibanez, M. A. Aloy, and P. Mimica, "Relativistic Magnetohydrodynamics: Renormalized eigenvectors and full wave decomposition Riemann solver," The Astrophysical Journal Supplement Series 188 (2010).

[23] J. M. Ibanez, M. A. Aloy, P. Mimica, L. Anton, J. A. Miralles, and J. M. Marti, "A Roe-type Riemann Solver Based on the Spectral Decomposition of the Equations of Relativistic Magnetohydrodynamics," Numerical Modeling of Space Plasma Flows: ASTRONUM-2010 **444,** 217–222 .

[24] M. L. Wilkins, "Use of artificial viscosity in multidimensional fluid dynamic calculations," Journal of Computational Physics **36,** 281–303 (1980).

[25] S. Osher, "Efficient implementation of essentially non-oscillatory shock-capturing schemes," Journal of Computational Physics **77,** 439–471 (1988).

[26] C.-W. Shu, "Essentially Non-Oscillatory and Weighted Essentially non-Oscillatory Schemes for Hyperbolic Conservation Law," NASA/CR-97-206253 ICASE Report 97-65 (1997).

[27] X.-D. Liu, S. Osher, and T. Chan, "Weighted Essentially Non-oscillatory Schemes," Journal of Computational Physics **115,** 200–212 (1994).