2010

Numerical Solution of the Five-Moment Ideal Two-Fluid Equations in One Dimension

Marcus Scott
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/gradreports
Part of the Mathematics Commons, and the Statistics and Probability Commons

Recommended Citation
Scott, Marcus, "Numerical Solution of the Five-Moment Ideal Two-Fluid Equations in One Dimension" (2010). All Graduate Plan B and other Reports. 1207.
https://digitalcommons.usu.edu/gradreports/1207
NUMERICAL SOLUTION OF THE FIVE-MOMENT IDEAL TWO-FLUID EQUATIONS IN ONE DIMENSION

by

Marcus Scott

A report submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE in
Industrial Mathematics

UTAH STATE UNIVERSITY
Logan, Utah
2010
ABSTRACT

Numerical Solution of the Five-Moment Ideal Two-Fluid Equations in One Dimension

by

Marcus Scott, Master of Science
Utah State University, 2010

Plasmas are frequently treated as a single conducting fluid and modeled using the equations of magnetohydrodynamics. However, this regime works better for low-frequency plasmas. High-frequency plasmas may be modeled using the principles of kinetic theory. For plasmas with frequencies between these two extremes, a two-fluid approach can yield better results. In 2006, Ammar Hakim mathematically modeled a plasma with a set of equations called the five-moment ideal two-fluid equations. An attempt is made to reproduce those results. A derivation of this set of equations by taking moments of the Boltzmann equation is presented. Electric and magnetic fields contribute to the source terms, so Maxwell's equations are coupled to the system. Finally, it is shown that single-fluid results can be obtained by taking suitable limits of the five-moment ideal two-fluid equations.

Assuming a fully ionized plasma of electrons and ions produces a hyperbolic system of sixteen equations. This hyperbolic system is approximately solved for a one-dimensional problem using a finite volume approach. Differing values in adjacent cells are treated as initial data for a Riemann problem and the jump is decomposed into waves. An approximate Riemann solver created through Roe averaging is used to numerically construct these waves. Source terms are solved for by a Strang splitting. The divergence constraints found in
Maxwell's equations can be handled in two different ways. The first approach, which has not been studied by the author, is to use a different set of Maxwell's equations called the perfectly hyperbolic Maxwell's equations (PHM). This approach introduces correction potentials to ensure that the divergence constraints are satisfied throughout the numerical simulation. The second approach is to choose specific values of the charge to mass ratio for the ions and electrons.

The software CLAWPACK is used for all simulations. A simple shock tube problem with one species and no source terms is first solved using sample code to illustrate some behavior of the larger system. Finally, an attempt is made at solving the same shock tube problem with two species and source terms. The resulting time ODE from the Strang splitting is solved using methods of various accuracy. It is shown that the time step required for stability is too small to be practical and thus a different approach to the problem, such as homogenized wavelet refinement, is needed.
ACKNOWLEDGMENTS

I would like to thank the generous people at USTAR for providing the grant money that made this research possible. I would also like to thank Joseph Koebbe, Farrell Edwards, and Eric Held for all of their assistance in helping me understand the concepts studied in this report. Their years of experience have been invaluable. Marcus Scott
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>ix</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Derivation of the Five-Moment Ideal Fluid Equations</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Notation and Definition of the Boltzmann Equation</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Derivation of the Zeroth Moment: Conservation of Matter</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Derivation of the First Moment: Conservation of Momentum</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>Derivation of the Second Moment: Conservation of Energy</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>Maxwell's Equations for the Electric and Magnetic Fields</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Limiting Cases for the Five-Moment Ideal Two-Fluid Equations</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Numerical Methods Discussion</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>One-Dimensional Riemann Problem</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Divergence Constraints</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Flux Jacobian Analysis</td>
<td>23</td>
</tr>
<tr>
<td>3.4</td>
<td>Source Term Analysis</td>
<td>25</td>
</tr>
<tr>
<td>3.5</td>
<td>Second-Order Accurate Method</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>Introduction to CLAWPACK</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Influence of High-Resolution Correction Terms</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Instability From an Inappropriate Courant Number</td>
<td>31</td>
</tr>
<tr>
<td>4.3</td>
<td>Effect of Initial Data on Left- and Right-Going Pulses</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>ONE-DIMENSIONAL SHOCK TUBE PROBLEM</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>Simplified Shock Tube Problem</td>
<td>37</td>
</tr>
<tr>
<td>5.2</td>
<td>Verifying Hakim's Results for the One-Dimensional Shock Tube Problem</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions and Future Work</td>
<td>41</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions</td>
<td>41</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>41</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>APPENDICES</td>
<td></td>
<td>45</td>
</tr>
<tr>
<td>A</td>
<td>Claw1ez.data File</td>
<td>46</td>
</tr>
</tbody>
</table>
B  Riemann Solver Code ........................................... 47
C  Source Term Codes ............................................ 53
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>42</td>
</tr>
<tr>
<td>6.2</td>
<td>42</td>
</tr>
</tbody>
</table>

6.1 Time step size and the number of time steps needed for the simulation to run until $t = 10$ under various assumptions for $x < 0$.

6.2 Time step size and the number of time steps needed for the simulation to run until $t = 10$ under various assumptions for $x > 0$. 
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>30</td>
</tr>
<tr>
<td>4.3</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>32</td>
</tr>
<tr>
<td>4.5</td>
<td>33</td>
</tr>
<tr>
<td>4.6</td>
<td>34</td>
</tr>
<tr>
<td>4.7</td>
<td>34</td>
</tr>
<tr>
<td>4.8</td>
<td>35</td>
</tr>
<tr>
<td>4.9</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>38</td>
</tr>
<tr>
<td>5.2</td>
<td>39</td>
</tr>
<tr>
<td>5.3</td>
<td>39</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

A plasma is a highly ionized gas consisting of electrons, ions, and neutral particles. One method of producing a plasma is to heat a gas to such high temperatures that molecules break down into atoms which in turn are stripped of electrons by the energy exchange from collisions. Given the large amount of energy needed to create and sustain a plasma and the high degree of ionization that occurs, plasmas are sometimes considered to be a fourth state of matter [5].

Particle interactions determine the properties of a plasma. Like a gas, particles in a plasma can interact through collisions. Additionally, the localized charges in a plasma allow particles to simultaneously interact with many others through electromagnetic forces. These are known as collective effects and are another distinguishing characteristic of plasmas. Quantum mechanical effects can typically be ignored as long as the charged particles do not come too close to each other [1].

Mathematically modeling plasmas is a difficult task. A typical approach is to treat the plasma as a single conducting fluid and use the equations of magnetohydrodynamics (MHD). The equations of MHD assume that the plasma is quasineutral, meaning that local charge imbalances can occur, but the plasma remains neutral globally. This is a valid approach for low-frequency plasmas because electrostatic forces lock the charged particles together and the plasma behaves like a single conducting fluid. As the frequency increases, the electrostatic forces are overcome and the ion and electron species are free to move independently. This is where the single-fluid MHD approach is less valid. High-frequency plasmas can be modeled using kinetic theory. At intermediate frequencies, such as those found in a tokamak, the plasma can be modeled as two fluids that penetrate and interact with each other [6, 7]. This report is concerned with plasma phenomena in the intermediate frequency range. One interesting result of modeling a plasma as two fluids is that greater charge separation is allowed since the quasineutrality assumption is no longer necessary [4].

Plasmas have numerous applications to science and technology. The bulk of plasma
research is devoted to controlling thermonuclear fusion. This unsolved problem involves creating and maintaining a plasma with energy in the range of 10 keV. The tokamak and spheromak are examples of machines built in hopes of creating such a plasma. Understanding the dynamics of a plasma can assist in the development of controlled fusion. Another interesting application to astrophysics is how the stars in a galaxy behave in the same manner as particles in a plasma. For a better explanation of the applications mentioned here, along with others, see [3].

When the Pioneer Venus Orbiter began studying the upper atmosphere and ionosphere of Venus in December of 1978, a surprising magnetic structure was discovered. Although Venus has no magnetic field of its own, its ionosphere is magnetized in rope-like strands while the surrounding plasma is nearly free of magnetic fields. These strands are called magnetic flux ropes and are typically observed in the Venusian ionosphere [12]. Plasma research can lead to a greater understanding of these magnetic flux ropes.

The Sun ejects a supersonic plasma called the solar wind. Solar wind is highly conducting and moves at speeds of about 500 km/s. The interaction between the solar wind and a planet depends on the strength of the planet’s ionosphere and magnetic field. Before explaining these interactions, it will be useful to define some terms. A bow shock is a bow-shaped shock front formed when the solar wind encounters a planet’s magnetic field. The magnetopause is a region of space marked by a sudden decrease in a planet’s magnetic field and the magnetosheath is the region separating the bow shock and the magnetopause.

When the solar wind encounters a strong magnetic field, such as the one generated by Earth or Jupiter, a strong bow shock is formed and most of the solar wind plasma in the magnetosheath is unable to penetrate the magnetopause. In this manner, the solar wind exerts pressure which compresses Earth’s magnetic field on the side near the sun and elongates it into a tail on the opposite side.

As the solar wind approaches a planet with a weak or nonexistent magnetic field, such as Venus or Mars, the bow shock formed is much closer to the planet than the one formed in the solar wind-Earth interaction. Although Venus does not have a magnetic field of its
own, the space plasma surrounding it does. Ions in the upper atmosphere and in the solar wind plasma that come into contact with this space plasma gyrate around the magnetic field lines and may be swept away into space or reenter the atmosphere. The curious nature of the plasma surrounding Venus and Mars has raised several questions which will require more study [10].

In 2006, Ammar Hakim of the University of Washington presented a doctoral dissertation on numerically solving the five-moment ideal two-fluid equations. As a test of the model, he solved a generalized version of the Brio-Wu shock tube problem in one dimension. This report presents a literature review to familiarize the author with plasmas and the existing methods of numerically simulating them. Simple problems are solved to calibrate the models and determine the efficacy of the existing methods.

For completeness, Chapter 2 presents a derivation of the five-moment ideal two-fluid equations from a basic equation of kinetic theory. While Hakim, et al. used the Vlasov equation, the derivations in this report begin with the Boltzmann equation so that collision terms may be included. Chapter 3 contains a discussion of the numerical methods tested and an analysis of the flux and source Jacobians. Chapter 4 is an introduction to CLAWPACK, the software package used to numerically solve the system. Chapter 5 begins with a discussion of the shock tube problem and how this problem is a good indicator of the behavior of a numerical approximation. Also included is a discussion of the software used by Hakim to solve the problem. Finally, chapter 6 presents the results and some ideas for future work. Code used by the author is found in the appendices.
CHAPTER 2
DERIVATION OF THE FIVE-MOMENT IDEAL FLUID EQUATIONS

The method of deriving the five-moment ideal equations presented in this chapter can be found in [5] and [7].

2.1 Notation and Definition of the Boltzmann Equation

Derivation of the five-moment ideal fluid equations begins with the Boltzmann equation,

\[ \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \nabla_v f = \frac{\delta_c f}{\delta t}. \]  

At this point, subscripts indicating plasma species will be omitted, so all equations derived will be for a single fluid. When the system of equations is applied to a plasma, each species will have its own set of equations and subscripts indicating species will be important.

The function \( f \) is a distribution function with three spatial coordinates, three velocity coordinates, and time as arguments. It is frequently written as \( f = f(x, \mathbf{v}, t) \) where \( x = (x, y, z) \) and \( \mathbf{v} = (v_x, v_y, v_z) \). The vector \( \nabla f = (\partial f/\partial x, \partial f/\partial y, \partial f/\partial z) \) and the vector \( \nabla_v f \) is defined similarly, except the partial derivatives are with respect to the velocity variables. The force \( \mathbf{F} \) is the Lorentz force, \( \mathbf{F} = q[\mathbf{E} + \mathbf{v} \times \mathbf{B}] \), where \( q \) is the charge, \( \mathbf{E} \) is the electric field, and \( \mathbf{B} \) is the magnetic field. The scalar \( m \) is the particle mass and the dot products that appear are the standard vector scalar product. Finally, the \( \delta_c f/\delta t \) term is an operator on the distribution function \( f \) accounting for Coulomb collisions. An explicit definition of this operator is not given and physical arguments will be used to account for it in the derivations.

Moment equations are obtained by multiplying the Boltzmann equation by a suitable power of \( \mathbf{v} \) and integrating over the velocity space, \( V \). Before doing this, it is useful to define some macroscopic averages that frequently arise in these derivations. Equations for the number density, average velocity, pressure tensor, and heat flow are respectively,

\[ n = \int_V f(x, \mathbf{v}, t) d^3v, \]
(2.3) \[ \mathbf{U} = \frac{1}{n} \int_V \mathbf{v} f(x, \mathbf{v}, t) \, d^3v, \]

(2.4) \[ \mathbf{P} = \int_V m(\mathbf{v} - \mathbf{U})(\mathbf{v} - \mathbf{U}) f(x, \mathbf{v}, t) \, d^3v, \]

and

(2.5) \[ Q_{ijk} = \int_V m(v_i - U_i)(v_j - U_j)(v_k - U_k) f(x, \mathbf{v}, t) \, d^3v. \]

Here the vector-vector product in (2.4) is a second-order tensor defined by

\[
(v - U)(v - U) = \\
\begin{bmatrix}
(v_x - U_x)(v_x - U_x) & (v_x - U_x)(v_y - U_y) & (v_x - U_x)(v_z - U_z) \\
(v_y - U_y)(v_x - U_x) & (v_y - U_y)(v_y - U_y) & (v_y - U_y)(v_z - U_z) \\
(v_z - U_z)(v_x - U_x) & (v_z - U_z)(v_y - U_y) & (v_z - U_z)(v_z - U_z)
\end{bmatrix}
\]

and each \( Q_{ijk} \) is an element of a third-order heat flow tensor. This tensor in its entirety will not appear in these derivations since only the form of its components is needed.

2.2 Derivation of the Zeroth Moment: Conservation of Matter

The zeroth moment is obtained by multiplying the Boltzmann equation by \( v^0 = 1 \) and integrating over \( V \), yielding an equation of the form

(2.7) \[ \int_V \frac{\partial f}{\partial t} \, d^3v + \int_V \mathbf{v} \cdot \nabla f \, d^3v + \frac{1}{m} \int_V \mathbf{F} \cdot \nabla \mathbf{v} f \, d^3v = \int_V \frac{\delta c f}{\delta t} \, d^3v. \]

Note that in the first term on the left, the integration is with respect to \( v \) while the differentiation is with respect to \( t \). Hence, the order of these operations does not matter. Therefore,

(2.8) \[ \int_V \frac{\partial f}{\partial t} \, d^3v = \frac{\partial}{\partial t} \left( \int_V f \, d^3v \right) = \frac{\partial n}{\partial t}, \]

where the last form was obtained using (2.2).

Since the components of \( \mathbf{v} \) are independent variables, the \( \mathbf{v} \) term in the second integral on the left is a constant with respect to the \( \nabla \) operator. In fact, the differentiations performed by the \( \nabla \) operator are not with respect to the integration variables, so the integral
can be rewritten as

\[ \int_{\mathbf{v}} \mathbf{v} \cdot \nabla f \, d^3v = \nabla \cdot \int_{\mathbf{v}} \mathbf{v} f \, d^3v \]

(2.9) \hspace{1cm} = \nabla \cdot (nU),

where the last form was obtained using (2.3).

To simplify the third integral on the left it must first be shown that with \( \mathbf{F} \) given by the Lorentz force, \( \mathbf{F} \cdot \nabla \mathbf{v} f = \nabla \mathbf{v} \cdot (f \mathbf{F}) \). Since \( \nabla \mathbf{v} \cdot (f \mathbf{F}) = \mathbf{F} \cdot \nabla \mathbf{v} f + f \nabla \mathbf{v} \cdot \mathbf{F} \), this amounts to showing that \( f \nabla \mathbf{v} \cdot \mathbf{F} = 0 \). The electric and magnetic fields are not dependent on the particle velocity, \( \mathbf{v} \), so

\[ f \nabla \mathbf{v} \cdot \mathbf{F} = f q \left[ \nabla \mathbf{v} \cdot \mathbf{E} + \nabla \mathbf{v} \cdot (\mathbf{v} \times \mathbf{B}) \right] \]

\[ = f q \left[ \nabla \mathbf{v} \cdot (\mathbf{v} \times \mathbf{B}) \right] \]

\[ = f q [\mathbf{B} \cdot (\nabla \mathbf{v} \times \mathbf{v}) - \mathbf{v} \cdot (\nabla \mathbf{v} \times \mathbf{B})] \]

\[ = 0. \]

Hence,

(2.10) \hspace{1cm} \mathbf{F} \cdot \nabla \mathbf{v} f = \nabla \mathbf{v} \cdot (f \mathbf{F}).

Therefore,

\[ \int_{\mathbf{v}} \mathbf{F} \cdot \nabla \mathbf{v} f \, d^3v = \int_{\mathbf{v}} \nabla \mathbf{v} \cdot (f \mathbf{F}) \, d^3v \]

\[ = \int (f \mathbf{F}) \cdot d\mathbf{s}_v \]

(2.11) \hspace{1cm} = 0,

where the divergence theorem was used to change the volume integral into a surface integral and \( f \) is such that \( f(\mathbf{v}) \to 0 \) as \( |\mathbf{v}| \to \infty \). This assumes there are no particles with infinite velocity.

Finally, the integral on the right can be evaluated by taking the collision operator outside the integral and using (2.2).

\[ \int_{\mathbf{v}} \delta T f \, d^3v = \frac{\delta}{\delta t} \int_{\mathbf{v}} f \, d^3v \]
because Coulomb collisions do not affect the fluid density. Combining (2.8), (2.9), (2.11), and (2.12) yields the zeroth moment equation

\[
\frac{\partial n}{\partial t} + \nabla \cdot (n \mathbf{U}) = 0,
\]

which is an equation for the conservation of matter.

2.3 Derivation of the First Moment: Conservation of Momentum

The first moment is obtained by multiplying (2.1) by \(m \mathbf{v}\) and integrating with respect to \(\mathbf{v}\). Doing so yields the equation

\[
m \int_V \mathbf{v} \frac{\partial f}{\partial t} \, d^3v + m \int_V \mathbf{v} \cdot \nabla f \, d^3v + \int_V \mathbf{v} (\mathbf{F} \cdot \nabla \mathbf{v}) \, d^3v = m \int_V \frac{\delta f}{\delta t} \, d^3v.
\]

Once again, the integrand of the first term on the left involves differentiation with respect to \(t\) and integration with respect to \(\mathbf{v}\), so the differentiation can be performed after the integration. Using (2.3), the first term can be rewritten as

\[
m \int_V \mathbf{v} \frac{\partial f}{\partial t} \, d^3v = \frac{\partial}{\partial t} \left( m \int_V \mathbf{v} f \, d^3v \right)
\]

\[
(2.15)
\]

The \(\nabla\) operator in the second integral of (2.14) differentiates with respect to \(x\) and is independent of \(\mathbf{v}\), so the integral can be written as

\[
m \int_V \mathbf{v} (\mathbf{v} \cdot \nabla f) \, d^3v = \nabla \cdot \left[ m \int_V \mathbf{v} \mathbf{v} f \, d^3v \right].
\]

The tensor, \(\mathbf{vv}\), can be replaced using the identity

\[
\mathbf{vv} = (\mathbf{v} - \mathbf{U})(\mathbf{v} - \mathbf{U}) + \mathbf{U}^2 + \mathbf{vU} - \mathbf{UU}
\]
where each of these tensors is defined in a manner similar to that of (2.6). Now the right-hand side of (2.16) can be written

\[ \nabla \cdot m \left[ \int_V (v - U)(v - U) f d^3v + \int_V U vf d^3v + \int_V vUf d^3v - \int_V Uuf d^3v \right]. \]  

This first integral is simply the pressure tensor \( \vec{P} \) given by (2.4). The second can be simplified by noting that the quantity \( U \) has no \( v \) dependence since it was obtained by integrating with respect to \( v \). Therefore,

\[ \int_V Uvf d^3v = U \int_V vf d^3v \]

(2.19)

\[ = nUU, \]

where (2.3) was used for the final form. By a similar argument, the third integral can be simplified to \( nUU \). The fourth integral can be written

\[ \int_V Uuf d^3v = UU \int_V f d^3v \]

(2.20)

\[ = nUU. \]

Combining the second, third, and fourth integrals of (2.18) shows that the second integral of (2.14) can be written

\[ m \int_V v(\nabla \cdot f) d^3v = \nabla \cdot \vec{P} + \nabla \cdot [mnUU]. \]

Earlier it was shown that \( F \cdot VvJ = Vv \cdot (JF) \). Because of this, the third integral of (2.14) can be written

\[ \int_V v(F \cdot \nabla f) d^3v = \int_V (v_x, v_y, v_z) \left[ \frac{\partial}{\partial u_x} (F_x f) + \frac{\partial}{\partial u_y} (F_y f) + \frac{\partial}{\partial u_z} (F_z f) \right] d^3v. \]

Nine integrations must be performed, and six are of the form

\[ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v_x dv_x dv_z \int_{-\infty}^{\infty} \frac{\partial}{\partial u_y} (F_y f) dv_y, \]
where the component of \( \mathbf{v} \) in the integrand does not match the component of \( \mathbf{v} \) in the differentiation. This can be integrated to obtain

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v_x \, dv_x dv_z [F_y f]_{-\infty}^{\infty}.
\]

Assuming, as above, that \( f \) is such that \( f(\mathbf{v}) \to 0 \) as \( |\mathbf{v}| \to \infty \), this integral and the five similar to it are all 0. The last three integrals are of the form

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dv_y dv_z \int_{-\infty}^{\infty} v_x \frac{\partial}{\partial v_x} (F_x f) \, dv_z,
\]

where the component of \( \mathbf{v} \) in the integrand and differentiation is the same. Performing an integration by parts yields

\[
\int_{-\infty}^{\infty} v_x \frac{\partial}{\partial v_x} (F_x f) \, dv_x = [v_x F_x f]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} F_x f \, dv_x.
\]

Once again, the first term on the right is 0, given the assumption on \( f \). Combining the second term on the right side of all three of these integrals yields the equation

\[
\int_{V} \mathbf{v} \cdot (\nabla f) \, d^3 v = - \int_{V} \mathbf{F} \cdot d^3 v.
\]

Since \( \mathbf{F} \) in this equation is the Lorentz force, this can be substituted in to obtain the result

\[
\int_{V} \mathbf{v} \cdot (\nabla f) \, d^3 v = -qn(\mathbf{E} + \mathbf{U} \times \mathbf{B}).
\]

Equations for the advancement of \( \mathbf{E} \) and \( \mathbf{B} \) are found in section 2.5.

Last of all, the right-hand side of the equation for the first moment can be written

\[
m \int_{V} \mathbf{v} \cdot \frac{\delta \mathbf{f}}{\delta t} \, d^3 v = \frac{\delta}{\delta t} (m \int_{V} \mathbf{v} f \, d^3 v)
\]

\[
= \frac{\delta}{\delta t} (mn \mathbf{U})
\]

\[
= \frac{\delta \mathbf{P}}{\delta t}
\]

\[
\text{(2.29)}
\]

where the \( \frac{\delta \mathbf{P}}{\delta t} \) term is a drag force due to collisions.

By using (2.15), (2.21), (2.28), and (2.29), the conservation of momentum equation can be written

\[
\frac{\partial}{\partial t} (mn \mathbf{U}) + \nabla \cdot (mn \mathbf{U}) \mathbf{U} = qn(\mathbf{E} + \mathbf{U} \times \mathbf{B}) - \nabla \cdot \mathbf{P} + \frac{\delta \mathbf{P}}{\delta t}.
\]

\[
\text{(2.30)}
\]
In this form, the equation is conservative. As a final simplification, the pressure tensor \( \mathbf{P} \) can be written component-wise as 
\[ P_{ij} = p\delta_{ij} + \Pi_{ij} \]  
where \( p = (P_{xx} + P_{yy} + P_{zz})/3 \) is a scalar pressure and \( \Pi \) is a trace-free stress tensor. This tensor accounts for anisotropic effects, and its non-diagonal entries are quite small. Thus, the entire stress tensor can be taken to be \( \mathbf{0} \), so \( P_{ij} = p\delta_{ij} \) and \( \nabla \cdot \mathbf{P} = \nabla p \). The first moment equation is then

\[ \frac{\partial}{\partial t} (mn\mathbf{U}) + \nabla \cdot (mn\mathbf{U}\mathbf{U}) + \nabla p = qn[\mathbf{E} + \mathbf{U} \times \mathbf{B}] + \frac{\delta \mathbf{P}}{\delta t}. \]

The scalar pressure assumption will also be used in the derivation of the second moment equation.

2.4 Derivation of the Second Moment: Conservation of Energy

Rather than multiplying the Boltzmann equation by \( vv \) and integrating with respect to \( v \), the second moment is obtained using the factor \( (1/2)mv^2 \), where \( v^2 = v_x^2 + v_y^2 + v_z^2 \). Two additional moments \( \mathbf{P} \) and \( \mathbf{Q} \) must first be defined. These are an order 2 and order 3 tensor, respectively, and will be examined component-wise, as this will simplify the derivation of the second moment. The first has components given by

\[ P_{ij} = m \int_V v_i v_j f \, d^3v. \]

These are related to the components of the pressure tensor \( \mathbf{P} \) as follows:

\[ P_{ij} = m \int_V (v_i - U_i)(v_j - U_j) f \, d^3v \]

\[ P_{ij} = m \int_V v_i v_j f \, d^3v - m \int_V v_i U_j f \, d^3v - m \int_V U_i v_j f \, d^3v + m \int_V U_i U_j f \, d^3v \]

\[ P_{ij} = P_{ij} - mU_j \int_V v_i f \, d^3v - mU_i \int_V v_j f \, d^3v + mU_i U_j \int_V f \, d^3v \]

\[ P_{ij} = P_{ij} - mU_j(nU_i) - mU_i(nU_j) + mU_i U_j(n) \]

\[ P_{ij} = P_{ij} - mnU_i U_j. \]

Next, the components of \( \mathbf{Q} \) are given by

\[ Q_{ijk} = m \int_V v_i v_j v_k f \, d^3v. \]
An expression relating these components to the components of the heat flow tensor \( \mathbf{Q} \) and \( \mathbf{P} \) can be derived as follows:

\[
Q_{ijk} = m \int_V (v_i - U_i)(v_j - U_j)(v_k - U_k) f d^3v
\]

\[
= m \int_V (v_i v_j v_k - v_i v_j U_k - v_i U_j v_k + v_i U_j U_k - U_i v_j v_k + U_i v_j U_k + U_i U_j v_k - U_i U_j U_k) f d^3v
\]

\[
= Q_{ijk} - mU_k \int_V v_i v_j f d^3v - mU_j \int_V v_i v_k f d^3v + mU_j U_k \int_V v_i f d^3v
\]

\[
- mU_i \int_V v_j v_k f d^3v + mU_i U_k \int_V v_j f d^3v + mU_i U_j \int_V v_k f d^3v
\]

\[
= Q_{ijk} - U_k P_{ij} - U_j P_{ik} + mnU_i U_j U_k - U_i P_{jk} + mnU_i U_j U_k
\]

\[
+ mnU_i U_j U_k - mnU_i U_j U_k
\]

(2.35) \( Q_{ijk} \) = \( Q_{ijk} + U_k P_{ij} + U_j P_{ik} + U_i P_{jk} - 2mnU_i U_j U_k \).

In the derivation of the second moment equation, \( Q_{ijk} \) will arise as \( Q_{iij} \), where the first and second indices are the same. In this situation,

(2.36) \( Q_{iij} = Q_{iij} + U_i P_{ii} + 2U_i P_{ij} - 2mnU_i U_i U_j \).

Factoring \( 2U_i \) out of the last two terms and using (2.33) shows that

(2.37) \( Q_{iij} = Q_{iij} + U_i P_{ii} + 2U_i P_{ij} \).

With these relationships for \( P_{ij} \) and \( Q_{iij} \), the second moment equation can be derived. Multiplying (2.1) by \((1/2)mv^2\) yields

(2.38) \( \frac{1}{2} m \int_V v^2 \frac{\partial f}{\partial t} d^3v + \frac{1}{2} m \int_V v^2 (\mathbf{v} \cdot \nabla f) d^3v + \frac{1}{2} \int_V v^2 \mathbf{F} \cdot \nabla f f d^3v = \frac{1}{2} m \int_V v^2 \frac{\delta f}{\delta t} d^3v. \)

All terms in the first integral of (2.38) can be taken inside the time derivative, so

(2.39) \( \frac{1}{2} m \int_V v^2 \frac{\partial f}{\partial t} d^3v = \frac{\partial}{\partial t} \left( \frac{1}{2} m \int_V v^2 f d^3v \right) \)

\( \frac{\partial \mathcal{E}}{\partial t} \).
where

\[ \mathcal{E} \equiv \frac{1}{2} m \int_V v^2 f \, d^3v \]

is the total fluid energy.

Since \( f \) is the only term in the second integral of (2.38) with \( x \) dependence, the integral can be written

\[ \frac{1}{2} m \int_V v^2 (\mathbf{v} \cdot \nabla f) \, d^3v = \nabla \cdot \left[ \int_V \frac{1}{2} m v^2 \mathbf{v} f \, d^3v \right]. \]  

This integral is a vector, and in component form is given by

\[ \int_V \frac{1}{2} m v^2 \mathbf{v}^2 f \, d^3v = \left[ \int_V \frac{1}{2} m v_x^2 v_x f \, d^3v + \int_V \frac{1}{2} m v_y^2 v_y f \, d^3v + \int_V \frac{1}{2} m v_z^2 v_z f \, d^3v \right]. \]

By (2.34), this is

\[ \frac{1}{2} \int_V \frac{1}{2} m v^2 \mathbf{v}^2 f \, d^3v = \left[ \frac{1}{2} Q_{xxx} + \frac{1}{2} Q_{xyy} + \frac{1}{2} Q_{zzz} \right]. \]

Note that since each term in the vector is of the form \( Q_{ii} \), they can be expanded using

\[ 2(Q_{xxx} + Q_{yy} + Q_{zzz}) + \frac{1}{2} \mathbf{U} \cdot (\mathbf{P}_{xx} + \mathbf{P}_{yy} + \mathbf{P}_{zz}) + U_x P_{xx} + U_y P_{yy} + U_z P_{zz} \]

From the definition of \( \mathcal{E} \) comes the relationship

\[ \mathcal{E} = \frac{1}{2} m \int_V v^2 f \, d^3v = \frac{1}{2} \left( m \int_V v_x^2 f \, d^3v + m \int_V v_y^2 f \, d^3v + m \int_V v_z^2 f \, d^3v \right) = \frac{1}{2} (P_{xx} + P_{yy} + P_{zz}). \]  

The scalar pressure introduced at the end of section 2.3 shows that \( P_{xx} = P_{yy} = P_{zz} = p \) and all other \( P_{ij} = 0 \). Finally, the \( Q_{ii} \) terms are taken to be 0 to close the system. Now
the vector can be written

\[ \int \frac{1}{2} m v^2 \mathbf{v} \cdot \mathbf{f} d^3 v = \left[ \begin{array}{c} U_x E + U_x p \\ U_y E + U_y p \\ U_z E + U_z p \end{array} \right] = \mathbf{U} E + \mathbf{U} p \]  

(2.45)

and therefore,

\[ \frac{1}{2} m \int v^2 (\mathbf{v} \cdot \nabla f) d^3 v = \nabla \cdot (\mathbf{U} E + \mathbf{U} p). \]  

(2.46)

For the third integral of (2.38), the \( \mathbf{F} \cdot \nabla \mathbf{v} \) term is replaced with \( \nabla \cdot (\mathbf{F} \mathbf{f}) \) as was shown in (2.10) and the integral now looks like

\[ \frac{1}{2} \int \int v^2 \mathbf{F} \cdot \nabla \mathbf{v} \cdot (\mathbf{f} \mathbf{f}) d^3 v = \frac{1}{2} \int \int v^2 \nabla \cdot (\mathbf{F} \mathbf{f}) d^3 v \]

\[ = \frac{1}{2} \int \int \left[ \frac{\partial}{\partial v_x} (F_x f) + \frac{\partial}{\partial v_y} (F_y f) + \frac{\partial}{\partial v_z} (F_z f) \right] d^3 v \]

\[ = \frac{1}{2} \left[ \int \int v^2 \frac{\partial}{\partial v_x} (F_x f) d^3 v + \int \int v^2 \frac{\partial}{\partial v_y} (F_y f) d^3 v + \int \int v^2 \frac{\partial}{\partial v_z} (F_z f) d^3 v \right]. \]

(2.47)

Integrating by parts gives

\[ \int \int v^2 \frac{\partial}{\partial v_x} (F_x f) d^3 v = \int_{-\infty}^{\infty} \left[ F_x f v^2 \right]_{v_x}^{v_x} - \int_{-\infty}^{\infty} F_x f \frac{\partial}{\partial v_x} (v^2) d v_x \]

\[ = - \int \int F_x f \frac{\partial}{\partial v_x} (v^2) d^3 v. \]  

(2.48)

Similar terms result from \( \int \int v^2 \frac{\partial}{\partial v_y} (F_y f) d^3 v \) and \( \int \int v^2 \frac{\partial}{\partial v_z} (F_z f) d^3 v \), so

\[ \frac{1}{2} \int \int v^2 \mathbf{F} \cdot \nabla \mathbf{v} \cdot (\mathbf{f} \mathbf{f}) d^3 v = \frac{1}{2} \left( \int \int F_x f \frac{\partial}{\partial v_x} (v^2) d^3 v + \int \int F_y f \frac{\partial}{\partial v_y} (v^2) d^3 v \right. \]

\[ + \int \int F_z f \frac{\partial}{\partial v_z} (v^2) d^3 v \]

\[ = \frac{1}{2} \left( \int \int \mathbf{F} \cdot \nabla (v^2) d^3 v \right). \]  

(2.49)

Further simplification is done by substituting the Lorentz force for \( \mathbf{F} \) and noting that

\[ \nabla \cdot (v^2) = 2v. \]

\[ \frac{1}{2} \int \int v^2 \mathbf{F} \cdot \nabla f d^3 v = - \frac{1}{2} \left( \int \int f q [\mathbf{E} + (\mathbf{v} \times \mathbf{B})] \cdot 2v d^3 v \right) \]

\[ = - \int \int f q [\mathbf{E} \cdot \mathbf{v} + (\mathbf{v} \times \mathbf{B}) \cdot \mathbf{v}] d^3 v \]  

(2.50)
The quantity \((v \times B)\) is orthogonal to \(v\) and \(E\) is independent of \(v\), so

\[
\frac{1}{2} \int_V v^2 F \cdot \nabla v f \, d^3v = -qE \cdot \int_V f v \, d^3v
\]

\(\quad = -qnE \cdot U.\)  

(2.51)

Finally, the right-hand side of (2.38) can be written

\[
\frac{1}{2} m \int_V v^2 \frac{\delta \rho f}{\delta t} \, d^3v = \frac{\delta \rho}{\delta t} \left( \frac{1}{2} m \int_V v^2 f \, d^3v \right)
\]

\(\quad = \frac{\delta \rho E}{\delta t}.\)  

(2.52)

In a completely ionized plasma, collisions merely transfer energy without affecting the total energy, so

\[
\frac{1}{2} m \int_V v^2 \frac{\delta \rho f}{\delta t} \, d^3v = 0.
\]

(2.53)

The combination of (2.39), (2.46), (2.51), and (2.53) yield the second moment equation

\[
\frac{\partial E}{\partial t} + \nabla \cdot (UE + Up) = qnE \cdot U,
\]

which is an equation for the conservation of energy. A relationship between the scalar pressure \(p\) and the total fluid energy \(E\) is necessary because this equation contains derivatives of both quantities. The equation \(E = \frac{1}{2}(P_{xx} + P_{yy} + P_{zz})\) can be rewritten using (2.33) as

\[
E = \frac{1}{2}(P_{xx} + P_{yy} + P_{zz} + mnUU_x + mnUU_y + mnUU_z).
\]

(2.55)

From the definition of the scalar pressure given in section 2.3, \(P_{xx} + P_{yy} + P_{zz} = 3p\), so

\[
E = \frac{3}{2}p + \frac{1}{2}mnU^2.
\]

(2.56)

In summary, the five-moment ideal fluid equations for each species \(s\) in the plasma are

\[
\frac{\partial n_s}{\partial t} + \nabla \cdot (n_s U_s) = 0
\]

(2.57)

\[
\frac{\partial}{\partial t} (m_s n_s U_s) + \nabla \cdot (m_s n_s U_s U_s + \nabla p_s) = q_s n_s [E + U_s \times B] + \frac{\delta \rho_s}{\delta t}
\]

(2.58)

\[
\frac{\partial E_s}{\partial t} + \nabla \cdot (U_s E_s + U_s p_s) = q_s n_s E \cdot U_s.
\]

(2.59)
2.5 Maxwell's Equations for the Electric and Magnetic Fields

The electric field, \( \mathbf{E} \), and the magnetic field, \( \mathbf{B} \), contribute to the source terms of the fluid equations, so Maxwell's equations

\[
\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t},
\]

\[
\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t},
\]

\[
\nabla \cdot \mathbf{E} = \frac{\rho_c}{\epsilon_0},
\]

and

\[
\nabla \cdot \mathbf{B} = 0
\]

can be used to determine these fields. The constants \( \mu_0 \) and \( \epsilon_0 \) are the permeability and permittivity of free space and the speed of light, \( c \), is defined by \( c = (\mu_0 \epsilon_0)^{-1/2} \). Last of all, the current density, \( \mathbf{J} \), and charge density, \( \rho_c \), are defined by:

\[
\mathbf{J} = \sum_s q_s n_s \mathbf{U}_s,
\]

\[
\rho_c = \sum_s q_s n_s,
\]

where \( s \) represents each species in the plasma. The two curl equations contribute six equations for the six components of \( \mathbf{E} \) and \( \mathbf{B} \), which combined with the five-moment ideal fluid equations give \( 5s + 6 \) equations for a plasma with \( s \) species.

2.6 Limiting Cases for the Five-Moment Ideal Two-Fluid Equations

The usefulness of the five-moment ideal two-fluid equations is demonstrated by taking limits of certain quantities to obtain results from different plasma regimes. To see this, a general vorticity equation will be derived, starting from (2.58). Dropping the subscript \( s \) and ignoring collisions, the left-hand side can be expanded as

\[
(2.66) \quad m \frac{\partial}{\partial t} (m \mathbf{U}) + U \frac{\partial}{\partial t} (mn) + \nabla \cdot (mn \mathbf{U}) \mathbf{U} + (mn \mathbf{U} \cdot \nabla) \mathbf{U} = nq(\mathbf{E} + \mathbf{U} \times \mathbf{B}) - \nabla \rho_c.
\]

The second and third terms of the left-hand side sum to 0 by (2.57). Dividing by \( n \) yields

\[
(2.67) \quad \frac{\partial}{\partial t} (m \mathbf{U}) + m \mathbf{U} \cdot \nabla \mathbf{U} = q(\mathbf{E} + \mathbf{U} \times \mathbf{B}) - \frac{\nabla \rho_c}{n}.
\]
By introducing a scalar potential, \( \phi \), and a vector potential, \( A \), the \( E \) and \( B \) fields in Maxwell's equations can be written

\[
E = -\nabla \phi - \frac{\partial A}{\partial t}
\]
\[
B = \nabla \times A.
\]

A generalized momentum, \( P \), is then defined as

\[
P = mU + qA.
\]

Taking the curl of \( P \) gives the generalized vorticity, \( \Omega \) which can be expressed as

\[
\Omega = \nabla \times P
\]
\[
= \nabla \times (mU) + \nabla \times (qA)
\]
\[
= m(\nabla \times U) + q(\nabla \times A)
\]

\[
(2.71)
\]

where \( \omega = \nabla \times U \).

The momentum equation can be written using the generalized momentum as

\[
\frac{\partial}{\partial t} \left( P - qA \right) + mU \cdot \nabla U = qE + qU \times B - \frac{\nabla \cdot P}{n}
\]
\[
\frac{\partial P}{\partial t} - \frac{\partial A}{\partial t} + mU \cdot \nabla U = qE + qU \times B - \frac{\nabla \cdot P}{n}
\]
\[
\frac{\partial P}{\partial t} + q(E + \nabla \phi) + mU \cdot \nabla U = qE + qU \times B - \frac{\nabla \cdot P}{n}
\]
\[
\frac{\partial P}{\partial t} + mU \cdot \nabla U = -\frac{\nabla \cdot P}{n} - \nabla (\phi q) + qU \times B
\]
\[
\frac{\partial P}{\partial t} + \nabla \left( \frac{mU^2}{2} \right) - mU \times \omega = -\frac{\nabla \cdot P}{n} - \nabla (\phi q) + qU \times B
\]
\[
\frac{\partial P}{\partial t} - U \times (m \omega + qB) = -\frac{\nabla \cdot P}{n} - \nabla \left( \frac{mU^2}{2} + q\phi \right)
\]
\[
(2.72)
\]
\[
\frac{\partial P}{\partial t} - U \times \Omega = -\frac{\nabla \cdot P}{n} - \nabla \left( \frac{mU^2}{2} + q\phi \right).
\]
Taking the curl of (2.72) gives the generalized vorticity equation

\[ \frac{\partial \Omega}{\partial t} - \nabla \times (U \times \Omega) = -\nabla \times \left( \frac{\nabla p}{n} \right). \]

Two-fluid theory is more general than the single-fluid approaches of Euler (neutral) fluid theory, Hall MHD, and ideal MHD. In the limit as \( B \to 0 \), (2.73) becomes

\[ \frac{\partial \omega}{\partial t} - \nabla \times (U \times \omega) = -\nabla \times \left( \frac{\nabla p}{\rho} \right). \]

In Euler (neutral) fluid theory, the flow velocities of each species become a bulk velocity for a single fluid. This bulk velocity, \( v \), is defined by

\[ v = \frac{m_e n_e U_e + m_i n_i U_i}{m_e n_e + m_i n_i}. \]

Dividing (2.74) by \( m \) and using the definition of \( v \) gives the Euler fluid result

\[ \frac{\partial \omega}{\partial t} - \nabla \times (v \times \omega) = -\nabla \times \left( \frac{\nabla p}{\rho} \right). \]

An equation from Hall MHD can be obtained by writing (2.73) for the electrons and dividing by the ion mass to get

\[ \frac{\partial}{\partial t} \left( \frac{m_e \omega_e + q_e B}{m_i} \right) - \nabla \times \left[ U_e \times \left( \frac{m_e \omega_e + q_e B}{m_i} \right) \right] = -\nabla \times \left( \frac{\nabla p_e}{m_i n_e} \right). \]

Letting the mass ratio \( m_e/m_i \to 0 \) and multiplying by \( m_i/q_e \) gives the Hall MHD result

\[ \frac{\partial B}{\partial t} - \nabla \times (U_e \times B) = -\nabla \times \left( \frac{\nabla p_e}{q_e n_e} \right). \]

Finally, an ideal MHD equation can be derived from (2.73) by taking the limit \( \omega/(qB/m) \to 0 \), where \( \omega = |\omega| \) and \( B = |B| \). The quantity \( qB/m \) is called the cyclotron frequency and is denoted by \( \omega_c \). Factoring \( \omega_c \) out of \( \Omega \) gives

\[ \frac{\partial}{\partial t} \left[ \frac{\omega_c}{m} \left( \frac{\omega \hat{\omega}}{\omega_c} + m \hat{b} \right) \right] - \nabla \times \left( U \times \left( \frac{qB}{m} \left( \frac{\omega \hat{\omega}}{\omega_c} + m \hat{b} \right) \right) \right) = -\nabla \times \left( \frac{\nabla p_e}{m_i n_e} \right), \]

where \( \hat{\omega} \) and \( \hat{b} \) are unit vectors in the direction of \( \omega \) and \( B \). Under the assumption that \( \omega/\omega_c \to 0 \), the equation becomes

\[ \frac{\partial (qB)}{\partial t} - \nabla \times (U \times qB) = -\nabla \times \left( \frac{\nabla p}{n} \right). \]
One of the main assumptions for ideal MHD to be valid is that the conductivity, $\sigma$, of the fluid is large. The conductivity can be expressed with the relation

\begin{equation}
\frac{1}{\sigma} = \frac{m_e \nu_{ei}}{n q_e^2},
\end{equation}

where $\nu_{ei}$ is the frequency of collisions between electrons and ions. Solving for $n$ and substituting into (2.80) gives

\begin{equation}
\frac{\partial (qB)}{\partial t} - \nabla \times (U \times qB) = -\nabla \times \left( \frac{q_e^2}{\sigma m_e \nu_{ei}} \nabla \rho \right).
\end{equation}

The collision frequency $\nu_{ei} \neq 0$, so as $\sigma \to \infty$, the right-hand side approaches 0. Using this limit, replacing $U$ by $v$, and dividing by $q$ gives the ideal MHD equation

\begin{equation}
\frac{\partial B}{\partial t} - \nabla \times (v \times B) = 0.
\end{equation}

The charge to mass ratio of the ions, $r_i$, is a parameter in this problem. As this ratio approaches $\infty$, two-fluid effects become less important and the two-fluid problem approaches the ideal MHD limit. Simulations presented in this report use $r_i = 10$. For more details about this and the other concepts discussed in this section, see [6] and [7].
CHAPTER 3
NUMERICAL METHODS DISCUSSION

3.1 One-Dimensional Riemann Problem

Hakim first solved a one-dimensional shock tube problem for a proton-electron plasma while ignoring collisions. A complete discussion of a shock tube problem will be given in chapter 5. In this situation, one-dimensional means each component only depends on \( x \) and \( t \), so partial derivatives with respect to \( y \) or \( z \) are 0. Hence, \( \nabla \cdot \mathbf{a} = \partial a_x / \partial x \). The zeroth moment equation is multiplied by \( m_s \) and the mass density \( \rho_s \equiv m_s n_s \) is defined. Here \( u, v, \) and \( w \) are the components of \( \mathbf{U} \). For all simulations presented in this report, the speed of light \( c \) and permittivity constant \( \varepsilon_0 \) are set to 1. Using the relationship \( c = (\varepsilon_0 \mu_0)^{-1/2} \), it can be shown that the magnetic permeability of free space, \( \mu_0 \), is also 1. Finally, the right-hand side of the electric field equations is multiplied by \( m_s / m_s \) and the charge to mass ratio, \( r_s \equiv q_s / m_s \), is defined. Under these assumptions, the system to be solved is

\[
\begin{align*}
(3.1) \quad \frac{\partial \rho_s}{\partial t} + \frac{\partial}{\partial x} (\rho_s u_s) &= 0 \\
(3.2) \quad \frac{\partial}{\partial t} (\rho_s u_s) + \frac{\partial}{\partial x} (\rho_s u_s^2) &= q_s n_s (E_x + v_s B_z - w_s B_y) \\
(3.3) \quad \frac{\partial}{\partial t} (\rho_s v_s) + \frac{\partial}{\partial x} (\rho_s u_s v_s) &= q_s n_s (E_y + w_s B_x - u_s B_z) \\
(3.4) \quad \frac{\partial}{\partial t} (\rho_s w_s) + \frac{\partial}{\partial x} (\rho_s u_s w_s) &= q_s n_s (E_z + u_s B_y - v_s B_x) \\
(3.5) \quad \frac{\partial E_x}{\partial t} + \frac{\partial}{\partial x} (u_s p_s + u_s E_x) &= q_s n_s (E_x u_s + E_y v_s + E_z w_s) \\
(3.6) \quad \frac{\partial B_x}{\partial t} &= 0 \\
(3.7) \quad \frac{\partial B_y}{\partial t} + \frac{\partial}{\partial x} (-E_x) &= 0 \\
(3.8) \quad \frac{\partial B_z}{\partial t} + \frac{\partial}{\partial x} (E_y) &= 0 \\
(3.9) \quad \frac{\partial E_x}{\partial t} &= - \sum_s r_s \rho_s u_s \\
(3.10) \quad \frac{\partial E_y}{\partial t} + \frac{\partial}{\partial x} (B_z) &= - \sum_s r_s \rho_s v_s \\
(3.11) \quad \frac{\partial E_z}{\partial t} + \frac{\partial}{\partial x} (-B_y) &= - \sum_s r_s \rho_s w_s
\end{align*}
\]
as a one-dimensional Riemann problem. Equation (2.56) is used to relate energy and pressure as before. Equations (3.1)–(3.5) have to be solved for electrons and ions, so there are a total of 16 equations in the system. The entire system is hyperbolic [6], [7] and has been written in a divergence form.

The Riemann problem is an initial value problem consisting of a system of hyperbolic equations with initial data of the form

\[
q(x, 0) = \begin{cases} 
q_l & \text{if } x < 0, \\
q_r & \text{if } x > 0,
\end{cases}
\]

where \( q_l \) and \( q_r \) are constant vectors. In the numerical scheme, the \( x \)-axis is discretized into cells and the values of \( q \) are approximated in each cell by the cell averages, so they are constant. For the simulations in this paper, 50,000 cells are used between -0.5 and 0.5. At the interface between adjacent cells, \( q \) will likely have different values on either side. By treating the interface as \( x = 0 \), advancing the solution to the next time step amounts to solving a Riemann problem and updating the cells [9].

For \( x < 0 \), the initial conditions for this simulation are:

\[
\begin{bmatrix}
\rho_e \\
\rho_e u_e \\
\rho_e v_e \\
\rho_e w_e \\
\mathcal{E}_e \\
\rho_i \\
\rho_i u_i \\
\rho_i v_i \\
\rho_i w_i \\
\mathcal{E}_i \\
B_x \\
B_y \\
B_z \\
E_x \\
E_y \\
E_z
\end{bmatrix} = \begin{bmatrix}
1.0 \frac{m_e}{m_i} \\
0 \\
0 \\
0 \\
7.5 \times 10^{-5} \\
1.0 \\
0 \\
0 \\
0 \\
7.5 \times 10^{-5} \\
0.75 \\
1.0 \\
0 \\
0 \\
0
\end{bmatrix},
\]
and for $x > 0$, the initial conditions are

$$
\begin{bmatrix}
\rho_e \\
\rho_e u_e \\
\rho_e v_e \\
\rho_e w_e \\
\mathcal{E}_e \\
\rho_i \\
\rho_i u_i \\
\rho_i v_i \\
\rho_i w_i \\
\mathcal{E}_i \\
B_x \\
B_y \\
B_z \\
E_x \\
E_y \\
E_z
\end{bmatrix}
= 
\begin{bmatrix}
0.125 \\
0 \\
0 \\
0 \\
7.5 \times 10^{-6} \\
0.125 \\
0 \\
0 \\
0 \\
7.5 \times 10^{-6} \\
0.75 \\
-1.0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} 
$$

This initial data is similar to that used by Brio and Wu for their classic shock-tube problem [2]. The ratio of the electron and ion masses, $m_e/m_i$, has the constant value 1/1832.6 for a proton-electron plasma.

In general, the process of numerically solving the equations governing a plasma begins with determining whether each species will be treated as a fluid or as a set of particles. This simulation treats electrons and ions as fluids. For simplicity, the plasma is assumed to be fully ionized. If the plasma is modeled using kinetic theory, then each species must be treated as a set of particles. Second, the nature of the electric and magnetic fields must be considered. Electromagnetic fields are solved differently than electrostatic fields. Next, attention must be paid to the boundary conditions, initial conditions, and geometry of the system. For example, a toroidal geometry would require that the boundary conditions be periodic. For the simulations presented here, the geometry is a one-dimensional tube and the equations have constant values on the boundaries. Initially, the gas on the left has a higher density and pressure than the gas on the right. Once all of these preliminary assumptions are in place, the system is solved through a recursive process of moving the fluids, accounting for the source terms, and solving for the fields. Finally, the results are analyzed to determine if the numerical solution makes sense [10].
3.2 Divergence Constraints

Satisfying the two divergence equations for \( \mathbf{E} \) and \( \mathbf{B} \) is another issue when numerically solving plasma equations. Typically, the divergence equations are satisfied initially and assumed to remain satisfied as the solution progresses. Equation (2.63) is the easier of the two to handle and is satisfied from the one-dimensionality of the problem and by having \( B_x \) initially constant as shown by the following [8]:

\[
\begin{align*}
\nabla \cdot \mathbf{B} &= 0 \\
\frac{\partial}{\partial x} (B_x) &= 0 \\
\frac{\partial}{\partial x} (\text{constant}) &= 0 \\
(3.13)\quad 0 &= 0.
\end{align*}
\]

Furthermore, equation (3.6) shows that \( B_x \) remains constant, so \( \nabla \cdot \mathbf{B} = 0 \) will be satisfied the entire time.

Equation (2.62) poses a greater challenge and there are two approaches to consider. Using the definition of \( \rho_c \) in equation (2.62) and multiplying the right-hand side by \( m_e/m_s \) gives

\[
(3.14) \quad \nabla \cdot \mathbf{E} = \frac{1}{\epsilon_0} (r_e \rho_e + r_i \rho_i).
\]

Hakim uses the true value \( r_e = -1.7588 \times 10^{11} \) and \( r_i = 10 \). With these values and the assumption \( \epsilon_0 = 1 \), equation (2.62) is not satisfied initially since the left-hand side is 0. To remedy this, he uses an altered set of Maxwell's equations known as the perfectly hyperbolic Maxwell's equations (PHM) which include correction potentials to ensure that the divergence constraints are satisfied as the solution progresses. These equations are:

\[
\begin{align*}
(3.15) \quad &\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} + \nu \nabla \psi = 0, \\
(3.16) \quad &\epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{B} + \chi \nabla \phi = -\mu_0 \mathbf{J}, \\
(3.17) \quad &\frac{1}{\chi} \frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{E} = \frac{\rho_c}{\epsilon_0}, \\
\text{and} \quad &\frac{\epsilon_0 \mu_0}{\nu} \frac{\partial \psi}{\partial t} + \nabla \cdot \mathbf{B} = 0.
\end{align*}
\]
The quantities $\psi$ and $\phi$ are correction potentials and $\chi$ and $\nu$ are error propagation speeds. Greater accuracy is achieved by letting $\chi, \nu \to \infty$. Setting $\chi = \nu = 2c = 2$ gives satisfactory results [6], [7]. If $\epsilon_0 = 1$ and the true value of $r_e$ is used, the time step for the numerical method must be $O(10^{-12})$. An explanation of how this time step is calculated is found in section 3.4 and its exact value is found in tables 6.1 and 6.2. Furthermore, using the PHM equations adds two equations to the system.

Using a time step that small will require approximately 2 trillion time steps to reproduce Hakim's results. The computational power available to the author is insufficient to handle such a long-running simulation, so a different approach is needed. A clever choice of $r_e$ will satisfy equation (2.62) initially. With zero for each component of the electric field, the equation becomes

\[
\nabla \cdot \mathbf{E} = \frac{1}{\epsilon_0} (q_e n_e + q_i n_i) \\
0 = r_e \rho_e + r_i \rho_i \\
r_e = -r_i \frac{\rho_i}{\rho_e} \\
r_e = -r_i \left( \frac{m_i}{m_e} \right) \\
r_e = -1832.6 r_i.
\]

(3.19)

This choice of $r_e$ results in an $O(10^{-5})$ time step for the simulation, which is much more feasible.

3.3 Flux Jacobian Analysis

One immediate concern is the question of the existence of a solution to the system satisfying the initial conditions. Proving the existence and uniqueness of a solution can be an arduous task, as shown in [13]. Furthermore, once the existence and uniqueness of a solution are demonstrated, it is usually impossible to construct the solution. In practice, a solution is often assumed to exist, numerically computed, and then analyzed to see if it makes sense physically. Given the hyperbolicity of the system and the nature of the initial conditions, the assumed solution will be computed using a Riemann solver.
A Riemann solver works by first calculating the eigenvalues ($\lambda_m$) and eigenvectors ($r_m$) of the flux Jacobian ($A$) of an $n$-dimensional hyperbolic system. The jump between the left and right states can then be decomposed as

\begin{equation}
q_r - q_l = \alpha_1 r_1 + \alpha_2 r_2 + \cdots + \alpha_n r_n.
\end{equation}

This can be written as a system of equations with the form

\begin{equation}
R\alpha = q_r - q_l,
\end{equation}

where $R$ is a matrix with $r_m$ as its $m^{th}$ column and $q_r - q_l$ is the jump between the left and right states. Therefore the vector $\alpha$ can be expressed

\begin{equation}
\alpha = R^{-1}(q_r - q_l).
\end{equation}

The components of $\alpha$ are called the projection coefficients. Since the rows of $R^{-1}$ are the left eigenvectors of $A$, $\alpha$ can be determined. Now the $m^{th}$ wave is given by

\begin{equation}
\mathcal{W}_m = \alpha_m r_m,
\end{equation}

with speed $\lambda_m$. Negative eigenvalues correspond to left-going waves and positive eigenvalues correspond to right-going waves. These waves update the values of $q$ in each cell and the process is repeated at each time step [9].

An inherent difficulty in calculating the solution to a Riemann problem is that the eigenvalues and eigenvectors must be known. For a nonlinear problem, eigenvalues and eigenvectors can only be computed locally for a linearized system. In addition to this, a closed form solution to the Riemann problem may not exist. Therefore, the best approach is to approximate the solution of the Riemann problem in such a way that the approximation becomes better as the mesh is refined. In this paper, Roe averages are used for this approximation. Roe averaging creates a Roe matrix that linearly approximates the flux Jacobian, so calculating eigenvalues and their corresponding eigenvectors becomes a linear algebra problem.
The $16 \times 16$ Roe matrix is block diagonal with the first $5 \times 5$ block corresponding to the electrons, the middle $5 \times 5$ block corresponding to the ions, and the remaining $6 \times 6$ block corresponding to the magnetic and electric fields. The eigenvalues, eigenvectors, and projection coefficients for the electron and ion blocks can be found in [11]. Eigenvectors corresponding to the magnetic and electric fields have 0 for the leading 10 entries and the remaining 6 entries of each eigenvector are, in matrix form,

$$
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & -1/\sqrt{2} & 0 & 0 & -1/\sqrt{2} & 0 \\
1/\sqrt{2} & 0 & 0 & 0 & 0 & 1/\sqrt{2} \\
0 & 0 & 0 & 1 & 0 & 0 \\
-1/\sqrt{2} & 0 & 0 & 0 & 1/\sqrt{2} & 0 \\
0 & -1/\sqrt{2} & 0 & 0 & 1/\sqrt{2} & 0
\end{bmatrix},
$$

with the eigenvalues given by $\lambda_{11} = -1, \lambda_{12} = -1, \lambda_{13} = 0, \lambda_{14} = 0, \lambda_{15} = 1$, and $\lambda_{16} = 1$.

The projection coefficients are given by

$$
\begin{align*}
\alpha_{11} & = \frac{1}{2}(B_{xr} - B_{zl}) - \frac{1}{2}(E_{yr} - E_{yt}) \\
\alpha_{12} & = -\frac{1}{2}(B_{yr} - B_{yl}) - \frac{1}{2}(E_{zr} - E_{zt}) \\
\alpha_{13} & = B_{xr} - B_{zl} \\
\alpha_{14} & = E_{xr} - E_{xl} \\
\alpha_{15} & = \frac{1}{2}(E_{zr} - E_{zt}) - \frac{1}{2}(B_{yr} - B_{yt}) \\
\alpha_{16} & = \frac{1}{2}(B_{zr} - B_{zt}) + \frac{1}{2}(E_{yr} - E_{yt}),
\end{align*}
$$

where $B_{zr}$ is the $z$-component of the magnetic field to the right of the cell interface and $B_{zl}$ is the $z$-component of the magnetic field to the left of the interface. All other terms in the projection coefficients are interpreted in a similar fashion.

### 3.4 Source Term Analysis

Many of the approximate Riemann solvers in the literature have been developed for systems without source terms, that is, the right-hand side is 0. This is not the case for the system to be solved in this report, so an analysis of the source terms and their effects on the numerical scheme is necessary.
Source terms can best be analyzed by setting spatial derivatives to zero to obtain a system of ordinary differential equations (ODEs) of the form

\begin{equation}
\frac{dq}{dt} = s
\end{equation}

and computing eigenvalues of the source Jacobian, \( \frac{\partial s}{\partial q} \). Setting the spatial derivatives to zero in the conservation of mass equation yields

\begin{equation}
\frac{\partial n_s}{\partial t} = 0.\end{equation}

Even though the conservation of mass equation has no source term, this result is important because it will be used in the momentum equation. Initially, the momentum equations are

\begin{equation}
m_s \frac{\partial}{\partial t} \begin{bmatrix} n_s u_s \\ n_s v_s \\ n_s w_s \end{bmatrix} = q_s n_s \begin{bmatrix} E_x + v_s B_z - w_s B_y \\ E_y + w_s B_x - u_s B_z \\ E_z + u_s B_y - v_s B_x \end{bmatrix}.
\end{equation}

After expanding the \( \frac{\partial}{\partial t} (n_s u_s) \), \( \frac{\partial}{\partial t} (n_s v_s) \), and \( \frac{\partial}{\partial t} (n_s w_s) \) terms according to the product rule and using the result of (3.32), the equations can be written as

\begin{equation}
m_s n_s \frac{\partial}{\partial t} \begin{bmatrix} u_s \\ v_s \\ w_s \end{bmatrix} = q_s n_s \begin{bmatrix} E_x + v_s B_z - w_s B_y \\ E_y + w_s B_x - u_s B_z \\ E_z + u_s B_y - v_s B_x \end{bmatrix}.
\end{equation}

Dividing by \( m_s n_s \) introduces an \( r_s \) term as defined earlier. These three equations are for each species in the plasma. The examples presented in this report focus on a two-species ion-electron plasma, so six equations are contributed to the system \( dq/dt = s \). Although the energy equation possesses source terms, it decouples from the system because energy does not appear in the source terms of the other equations and can be ignored.

Equation (2.61) contributes the remaining source terms to the system. Zeroing out spatial derivatives and using \( c \equiv (\mu_0 \epsilon_0)^{-1/2} \) results in

\begin{equation}
\mu_0 J + \frac{1}{c^2} \frac{\partial E}{\partial t} = 0
\end{equation}

\begin{equation}
\frac{\partial E}{\partial t} = -c^2 \mu_0 J
\end{equation}

\begin{equation}
\frac{\partial E}{\partial t} = -\frac{1}{\epsilon_0} J.
\end{equation}
This provides three more equations to the system.

After writing $\mathbf{J}$ in component form, the final system is

\[
\frac{d}{dt} \begin{bmatrix}
    u_e \\
    v_e \\
    w_e \\
    u_i \\
    v_i \\
    w_i \\
    E_x \\
    E_y \\
    E_z
\end{bmatrix} = \begin{bmatrix}
    r_e (E_x + v_e B_z - w_e B_y) \\
    r_e (E_y + w_e B_x - u_e B_z) \\
    r_e (E_z + u_e B_y - v_e B_x) \\
    r_i (E_x + v_i B_z - w_i B_y) \\
    r_i (E_y + w_i B_x - u_i B_z) \\
    r_i (E_z + u_i B_y - v_i B_x) \\
    -\frac{1}{c_0} (q_e n_e u_e + q_i n_i u_i) \\
    -\frac{1}{c_0} (q_e n_e v_e + q_i n_i v_i) \\
    -\frac{1}{c_0} (q_e n_e w_e + q_i n_i w_i)
\end{bmatrix}
\]

(3.36)

with source Jacobian

\[
\frac{\partial S}{\partial \mathbf{q}} = \begin{bmatrix}
    0 & r_e B_z & -r_e B_y & 0 & 0 & 0 & r_e & 0 & 0 \\
    -r_e B_z & 0 & r_e B_y & 0 & 0 & 0 & 0 & 0 & r_e \\
    r_e B_y & -r_e B_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & -r_i B_z & 0 & r_i B_y & 0 & 0 & r_i \\
    0 & 0 & -r_i B_y & 0 & r_i B_x & 0 & 0 & 0 & r_i \\
    -\frac{\tau_{\phi e}}{c_0} & 0 & 0 & -\frac{\tau_{\phi i}}{c_0} & 0 & 0 & 0 & 0 & 0 \\
    0 & -\frac{\tau_{\phi e}}{c_0} & 0 & 0 & -\frac{\tau_{\phi i}}{c_0} & 0 & 0 & 0 & 0 \\
    0 & 0 & -\frac{\tau_{\phi e}}{c_0} & 0 & 0 & -\frac{\tau_{\phi i}}{c_0} & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

(3.37)

Three of the source Jacobian eigenvalues are 0 and $\pm i \omega_p \equiv \sqrt{\omega_{pe}^2 + \omega_{pi}^2}$ where $i = \sqrt{-1}$ and $\omega_{pi}^2 \equiv (n_ip_i^2)/(\epsilon_0 m_i)$. The characteristic polynomial for the remaining six eigenvalues is an even function with positive coefficients, so all of its roots are purely imaginary. Therefore, the source terms are characterized by undamped oscillations and they add no dissipation to the solution. Furthermore, the eigenvalues of the source Jacobian are frequencies, so a characteristic time scale for the problem is given by $T_c = \max_i \{2\pi/|\omega_i|\}$, where $\omega$ is one of these eigenvalues. This puts an additional constraint on the time step because now the time step must be chosen so that it satisfies the CFL condition ($\max_m \{|\lambda_m \Delta t/\Delta x| \leq 1\}$ and is smaller than $T_c$ [6], [7].

3.5 Second-Order Accurate Method

The system of $m$ equations will be solved using a second-order accurate method with the formula

\[
Q_i^{n+1} = Q_i - \frac{\Delta t}{\Delta x} \left( A^- \Delta Q_{i+1/2} + A^+ \Delta Q_{i-1/2} \right) - \frac{\Delta t}{2\Delta x} \left( \tilde{F}_{i+1/2} - \tilde{F}_{i-1/2} \right),
\]

(3.38)
where

\begin{equation}
A^+ \Delta Q_{i-1/2} = \sum_{p=1}^{m} (s^p)^+ \mathcal{W}^p_{i-1/2}
\end{equation}

and

\begin{equation}
A^- \Delta Q_{i-1/2} = \sum_{p=1}^{m} (s^p)^- \mathcal{W}^p_{i-1/2}.
\end{equation}

The notation $(s^p)^+ = \max\{s^p, 0\}$ and $(s^p)^- = \min\{s^p, 0\}$ where $s^p$ is the $p^{th}$ eigenvalue of the Roe-averaged flux Jacobian. Also, $\tilde{F}_{i-1/2}$ is given by

\begin{equation}
\tilde{F}_{i-1/2} = \frac{1}{2} \sum_{p=1}^{m} |s^p_{i-1/2}| \left( 1 - \frac{\Delta t}{\Delta x} |s^p_{i-1/2}| \right) \tilde{\mathcal{W}}_{i-1/2}.
\end{equation}

The quantity $\tilde{\mathcal{W}}_{i-1/2}$ is a wave that has been limited by the monotonized centered flux-limiter. This is because second-order accurate methods have spurious oscillations near discontinuities. A flux-limiter reduces the method to first-order accuracy near discontinuities so that these oscillations do not occur. Furthermore, the source term cannot be directly incorporated into the Riemann solver because the source terms are characterized by undamped oscillations. Doing so would result in an unstable method [6], [7]. Therefore, the source terms will be handled by performing a Strang splitting and solving the resulting time ODE. The accuracy of the approximate solution to this time ODE affects the accuracy of the overall solution. Code for a first-order explicit Euler method, a second-order Runge-Kutta method, and a fourth-order Runge-Kutta method can be found in the appendices. Detailed explanations of all of these concepts can be found in [9].
All simulations presented in this paper will be performed using CLAWPACK (Conservation Law Package), written by Randall Leveque of the University of Washington. The information on CLAWPACK and sample problems presented here were taken from [9]. Documentation for CLAWPACK, along with the software package can be found at www.amath.washington.edu/~clawpack. CLAWPACK utilizes the techniques of finite volume methods for the solution of hyperbolic problems, so it is ideal for solving the five-moment ideal two-fluid equations.

An illustration of how CLAWPACK works will be given by solving the linearized acoustics equations for a one-dimensional system initially at rest. Changes to the problem will be made in the next three subsections. Specifically, the system looks like

\[
\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0
\]

where \( p, u, K_0, \) and \( \rho_0 \) are the pressure, velocity, bulk modulus, and density, respectively. For this example, \( K_0 = 4.0, \rho_0 = 1.0, \) and the initial conditions for \( p \) and \( u \) are

\[
p = \begin{cases} e^{-50(x-0.3)^2} \cos(20(x - 0.3)) & \text{if } x \leq 0.3 \\ 0 & \text{otherwise} \end{cases}
\]

and

\[
u = 0.
\]

At time \( t = 0 \) the pressure and velocity are shown in figure 4.1. Both graphs then split into left- and right-going pulses as shown in figure 4.2. The boundary at \( x = 1 \) allows for outflow while the boundary at \( x = -1 \) is a reflecting wall. Eventually, the right-going pulse leaves the viewing window and the left-going pulse becomes right-going after it is reflected off the \( x = -1 \) boundary (see figure 4.3).
Figure 4.1. Initial conditions for pressure ($q(1)$) and velocity ($q(2)$).

Figure 4.2. Left- and right-going pulses.
4.1 Influence of High-Resolution Correction Terms

The methods that CLAWPACK uses to solve hyperbolic problems are determined by the file claw1ez.data. For example, values read in from this file determine the number of grid cells, Courant numbers, order of the method, and types of boundary conditions. A sample of the code contained in claw1ez.data, along with a description of its functions can be found in Appendix A. The order of the method can be changed by setting the value of method(2). For method(2)=2, higher order correction terms are used and for method(2)=1, a first-order upwind method is used. Figures 4.4 and 4.5 will show how the previous example looks under this upwind method. Notice that the waves are not as steep and are more rounded at the peaks. This is caused by the “smearing” effect that odd-ordered methods have.

4.2 Instability From an Inappropriate Courant Number

This next sample problem investigates the role Courant numbers play in the stability of a numerical scheme. The Courant number is defined by

\[ v = \frac{\Delta t}{\Delta x} \max_m |\lambda_m|, \]  

(4.4)
where $\Delta t$ is the time step, $\Delta x$ is the space step, and $\lambda_m$ is the $m$th eigenvalue of the hyperbolic system. For the first-order upwind method, Courant numbers no greater than 1 result in a stable method. In this sample problem, the Courant number is set to 1.1 and the resulting instability can be seen after the first time step as shown in figure 4.6. Figure 4.7 shows how the instability gets even worse after the second time step.

4.3 Effect of Initial Data on Left- and Right-Going Pulses

The final sample problem provides insight into how the Riemann solver works and illustrates how the initial conditions affect the solution. The goal of this sample problem is to modify the initial data of $u(x)$ so that only a left-going pulse appears. An appropriate initial condition for $u(x)$ can be found by examining how the solution is formed in the Riemann solver.

The eigenvalues of the linear acoustics equations are given by

$$\lambda_1 = -c_0 \text{ and } \lambda_2 = c_0.$$
where
\[ c_0 = \sqrt{\frac{K_0}{\rho_0}}. \]

After defining the impedance, \( Z_0 \) by
\[ Z_0 = \rho_0 c_0, \]
the eigenvectors are given by
\[ r_1 = \begin{bmatrix} -Z_0 \\ 1 \end{bmatrix} \quad \text{and} \quad r_2 = \begin{bmatrix} Z_0 \\ 1 \end{bmatrix}. \]

Therefore, \( \alpha_1 \) and \( \alpha_2 \) are given by
\[ \alpha_1 = \frac{(p_r - p_l) + Z_0 (u_r - u_l)}{2Z_0}, \]
\[ \alpha_2 = \frac{(p_r - p_l) + Z_0 (u_r - u_l)}{2Z_0}. \]

The right-going wave has eigenvector coefficient \( \alpha_2 \), so it can be set to 0 by choosing the initial condition for \( u \) to be the negative of the initial condition for \( p \) scaled by a factor of \( 1/Z_0 \). In this sample problem, \( Z_0 = 2.0 \). Setting the initial data for \( u \) as
\[ u = \begin{cases} -\frac{1}{2}e^{-50(x-0.3)^2} \cos(20(x-0.3)) & \text{if } x \leq 0.3 \\ 0 & \text{otherwise} \end{cases} \]
Figure 4.6. Instability after the first time step with a Courant number of 1.1.

Figure 4.7. Instability after the second time step with a Courant number of 1.1.
will eliminate the right-going pulse. Graphs of the initial conditions for $p$ and $u$ are shown in figure 4.8. Figure 4.9 verifies that there is no right-going pulse under these initial conditions.
Figure 4.9. Pulse is purely left-going.
As discussed in the previous chapter, the numerical solution of the five-moment ideal two-fluid equations depends on the solution of a Riemann problem. Evolving the solution of a Riemann problem in time creates rarefaction waves and shock waves. Although such waves do not occur in plasmas, a shock tube problem does exhibit such behavior and is therefore a good starting point for testing plasma-modeling code [8].

A shock tube is an experimental setup in which two gases at different states are separated by a thin membrane. Initially, the gases are at rest, so they differ only in density and pressure. At time $t = 0$, the membrane is instantaneously removed and the gases are allowed to interact. The solution involves a shock wave moving into the gas at lower pressure and a rarefaction wave moving into the gas at higher pressure [9]. Shock tubes have been studied extensively over the years and famous results have been obtained by Sod [14] and Brio and Wu [2]. Another reason why shock tube problems are used to test plasma code is that any results obtained can be compared to these classic results.

5.1 Simplified Shock Tube Problem

Sample code for solving the Euler equations of gas dynamics in a one-dimensional shock tube is included in the CLAWPACK download. This is a simpler system with only one species and no source terms, but its behavior is qualitatively similar to the more complex system solved by Hakim. The equations are [9]

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho u) & = 0, \\
\frac{\partial (\rho u)}{\partial t} + \frac{\partial}{\partial x} (\rho u^2 + p) & = 0,
\end{align*}
\]

and

\[
\frac{\partial \mathcal{E}}{\partial t} + \frac{\partial}{\partial x} [(\mathcal{E} + p)u] = 0.
\]
As with the five-moment ideal two-fluid equations, the system is closed by using the equation of state
\begin{equation}
\mathcal{E} = \frac{3}{2}p + \frac{1}{2}\rho u^2.
\end{equation}

A simulation was run with the initial conditions
\[
\begin{bmatrix}
\rho \\
\rho u \\
\mathcal{E}
\end{bmatrix}
= \begin{bmatrix}
1.0_{\text{m}}m_i \\
0 \\
7.5 \times 10^{-5}
\end{bmatrix}
\]
for \( x < 0 \) and
\[
\begin{bmatrix}
\rho \\
\rho u \\
\mathcal{E}
\end{bmatrix}
= \begin{bmatrix}
0.125_{\text{m}}m_i \\
0 \\
7.5 \times 10^{-6}
\end{bmatrix}
\]
for \( x > 0 \). Results are presented for the mass density, \( \rho \). The initial condition for \( \rho \) can be seen in figure 5.1. Immediately, waves start to appear. The slanted region of figure 5.2 is a rarefaction wave moving into the gas at higher pressure, as mentioned earlier. The front of the stair-step region is a shock wave moving into the gas at lower pressure. The rarefaction and shock waves continue to propagate through the shock tube and figure 5.3 shows the simulation at a later time. Hakim’s simulations were run until time \( t = 10.0 \), but these were stopped at time \( t = 1.0 \) because the shock leaves the viewing window sometime after \( t = 1.0 \)
Figure 5.2. Rarefaction wave and slow shock.

Figure 5.3. Rarefaction wave and slow shock at a later time.
leaving only the rarefaction wave. Source terms in Hakim's simulation will slow down the propagation speed of the shock. Also, the mass densities in Hakim's simulation should exhibit similar behavior to the mass density in this simpler problem; namely, a rarefaction wave and a shock wave should form.

5.2 Verifying Hakim's Results for the One-Dimensional Shock Tube Problem

The Computational Fluid Dynamics Laboratory at the University of Washington is developing a software package called WarpX for solving plasma equations. Hakim's dissertation was an initial test run of this software to see if it could solve some classic problems, such as the shock tube studied here. WarpX is written in C++ with Python scripts and makes use of the Message Passing Library to allow parallel computing. Hakim ran the software on the Ladon Supercomputer owned by the Department of Mechanical Engineering at the University of Washington [7]. More information on WarpX can be found at warpx.org/wiki/index.php?title=Main_Page. An attempt was made to replicate Hakim's results by using CLAWPACK. Ultimately, the required time step was too small for the simulation to be feasible, as discussed in the next chapter. Code for the Riemann solver can be found in Appendix B and Appendix C contains the code for various methods of solving the time ODE from the Strang splitting.
CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In the end, the CLAWPACK code was unable to reproduce Hakim's results using the $O(10^{-5})$ time step mentioned in Chapter 3. The assumptions necessary to use this time step were not physically correct, so instabilities appeared. Tables 6.1 and 6.2 show the different time steps needed under various assumptions. Also shown is the number of time steps necessary to reach $t = 10$, the time at which Hakim displayed his results. For most of the cases presented in these tables, the computational power available to the author is insufficient to run the simulations and the one that can be run is unstable. Another limitation is that the WarpX code run by Hakim is designed and optimized for solving plasma fluid equations, while CLAWPACK is a generic package for solving systems of hyperbolic conservation laws.

6.2 Future Work

The computational limitations discussed above demonstrate that a different approach is needed for solving the five-moment ideal two-fluid equations. As future work, the techniques of homogenization theory will be used to handle the different time and length scales that control the behavior of each plasma species. Also, the more modern technique of homogenized wavelet refinement (HWR) will be used to model the wave nature of plasmas. HWR is a fast transform method for solving systems of partial differential equations. The idea is to perform a dyadic refinement at each step, find the adaptive wavelets, and homogenize them. The analysis for solving a one-dimensional electrostatic problem using this method has already been performed and the code has been written. Solving the two-fluid equations in one dimension with this method is the next step. From here, everything will be in place to solve the one-dimensional shock tube problem presented in this paper using HWR. Extending these results to higher dimensions is the final step. There is an algorithm for the two-dimensional elliptic problem and work is being done for the three-dimensional...
case, specifically to solve equations of the form $\nabla \cdot (k \nabla h) = f$.

Table 6.1. Time step size and the number of time steps needed for the simulation to run until $t = 10$ under various assumptions for $x < 0$.

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Step size</th>
<th>Number of steps to $t = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_e = -\left(\frac{m_i}{m_e}\right) r_i$ and $\epsilon_0 = 1$</td>
<td>$4.3639 \times 10^{-5}$</td>
<td>$2.2915 \times 10^5$</td>
</tr>
<tr>
<td>$r_e = -\left(\frac{m_i}{m_e}\right) r_i$ and $\epsilon_0 = 8.8542 \times 10^{-12}$</td>
<td>$6.9484 \times 10^{-9}$</td>
<td>$1.4392 \times 10^9$</td>
</tr>
<tr>
<td>$r_e = -1.7588 \times 10^{11}$ and $\epsilon_0 = 1$</td>
<td>$4.5470 \times 10^{-12}$</td>
<td>$2.1993 \times 10^{12}$</td>
</tr>
<tr>
<td>$r_e = -1.7588 \times 10^{11}$ and $\epsilon_0 = 8.8542 \times 10^{-12}$</td>
<td>$7.2420 \times 10^{-16}$</td>
<td>$1.3808 \times 10^{16}$</td>
</tr>
</tbody>
</table>

Table 6.2. Time step size and the number of time steps needed for the simulation to run until $t = 10$ under various assumptions for $x > 0$.

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>Step size</th>
<th>Number of steps to $t = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_e = -\left(\frac{m_i}{m_e}\right) r_i$ and $\epsilon_0 = 1$</td>
<td>$4.3652 \times 10^{-5}$</td>
<td>$2.2908 \times 10^5$</td>
</tr>
<tr>
<td>$r_e = -\left(\frac{m_i}{m_e}\right) r_i$ and $\epsilon_0 = 8.8542 \times 10^{-12}$</td>
<td>$1.9650 \times 10^{-8}$</td>
<td>$5.0890 \times 10^8$</td>
</tr>
<tr>
<td>$r_e = -1.7588 \times 10^{11}$ and $\epsilon_0 = 1$</td>
<td>$4.5484 \times 10^{-12}$</td>
<td>$2.1986 \times 10^{12}$</td>
</tr>
<tr>
<td>$r_e = -1.7588 \times 10^{11}$ and $\epsilon_0 = 8.8542 \times 10^{-12}$</td>
<td>$2.0480 \times 10^{-15}$</td>
<td>$4.8827 \times 10^{15}$</td>
</tr>
</tbody>
</table>
REFERENCES


APPENDICES
APPENDIX A
CLAW1EZ.DATA FILE

This appendix contains a copy of the file claw1ez.data, which specifies how CLAWPACK is
to solve the numerical system. Examples include setting how high the CFL can be, what
kind of source term splitting should be performed, how the waves should be limited, and
how the boundary conditions should be handled. For complete documentation, see the
website mentioned in chapter 4.

50000  mx = cells in x direction
10     nout = number of output times to print results
1      outstyle = style of specifying output times
10.d0  tfinal = final time

1.8d-5 dtv(1) = initial dt (used in all steps if method(1)=0)
1.0d99 dtv(2) = max allowable dt
1.0d0  cflv(1) = max allowable Courant number
0.9d0  cflv(2) = desired Courant number
3000   nv(1) = max number of time steps per call to claw1

1  method(1) = 1 for variable dt
2  method(2) = order
0  method(3) = not used in one dimension
0  method(4) = verbosity of output
2  method(5) = source term splitting
0  method(6) = mcapa
0  method(7) = maux (should agree with parameter in driver)

16  meqn = number of equations in hyperbolic system
16  mwaves = number of waves in each Riemann system
4 4 4 4 4 4 4 4 4 4 4 4 4 4 mthl(mw) = wave limiter

0.40  t0 = initial time
-0.5d0 xlower = left edge of computational domain
0.5d0  xupper = right edge of computational domain

2  mbc = number of ghost cells at each boundary
1  mthbc(1) = type of boundary conditions at left
1  mthbc(2) = type of boundary conditions at right
APPENDIX B
RIEMANN SOLVER CODE

This appendix presents the code for the Riemann solver used in the simulations. The primary goal of the Riemann solver is to calculate the fluctuations, denoted in the code as amdq and apdq, which are used for the first-order approximation to the solution. Waves and wave speeds obtained while computing amdq and apdq are used for the second-order approximation.

```fortran
C =========================================================
subroutine rp1(maxmx,meqn,mwaves,mbc,mx,ql,qr,auxl,auxr,
                & wave,s,amdq,apdq)
C =========================================================
C
C # solve Riemann problems for the 1D Euler equations using Roe's
C # approximate Riemann solver.
C
C # On input, ql contains the state vector at the left edge of each cell
C # qr contains the state vector at the right edge of each cell
C # On output, wave contains the waves,
C # s the speeds,
C # amdq the left-going flux difference A^- \Delta q
C # apdq the right-going flux difference A^-+ \Delta q
C
C # Note that the i'th Riemann problem has left state qr(i-1,:) and right state ql(i,:)
C # From the basic clawpack routine step1, rp is called with ql = qr = q.
C
implicit double precision (a-h,o-z)
dimension ql(1-mbc:maxmx+mbc, meqn)
dimension qr(1-mbc:maxmx+mbc, meqn)
dimension s(1-mbc:maxmx+mbc, mwaves)
dimension wave(1-mbc:maxmx+mbc, meqn, mwaves)
dimension amdq(1-mbc:maxmx+mbc, meqn)
dimension apdq(1-mbc:maxmx+mbc, meqn)

C # local storage
-------------
parameter (max2 = 50002) !# assumes at most 50000 grid points with mbc=2
dimension delta(meqn)
dimension ue(-1:max2)
dimension ve(-1:max2)
dimension we(-1:max2)
dimension ente(-1:max2)
dimension qe(-1:max2)
dimension ce(-1:max2)
dimension ui(-1:max2)
dimension vi(-1:max2)
dimension wi(-1:max2)
```
if (mx+mbc .gt. max2) then
  write(6,*) '*** Error *** need to increase max2 in rpleu'
  stop
endif

# Compute Roe-averaged quantities:

do 20 i=2-mbc,mx+mbc

  # electron quantities-------------------------------
  rhoe = dsqrt(qr(i-1,1)) + dsqrt(ql(i,1))
  ue(i) = (qr(i-1,2)/dsqrt(qr(i-1,1)) + ql(i,2)/
            dsqrt(ql(i,1))) / rhoe
  ve(i) = (qr(i-1,3)/dsqrt(qr(i-1,1)) + ql(i,3)/
            dsqrt(ql(i,1))) / rhoe
  we(i) = (qr(i-1,4)/dsqrt(qr(i-1,1)) + ql(i,4)/
            dsqrt(ql(i,1))) / rhoe
  hle = ((5.d0/3.d0)*qr(i-1,5) - (qr(i-1,2)**2 + qr(i-1,3)**2
          + qr(i-1,4)**2)/(3.d0*qr(i-1,1)))/dsqrt(qr(i-1,1))
  hre = ((5.d0/3.d0)*ql(i,5) - (ql(i,2)**2 + ql(i,3)**2 +
          ql(i,4)**2)/(3.d0*ql(i,1)))/dsqrt(ql(i,1))
  ente(i) = (hle + hre)/rhoe
  qe(i) = ue(i)**2 + ve(i)**2 + we(i)**2
  ce(i) = dsqrt((2.d0/3.d0)*(ente(i) - 0.5d0*qe(i)))

  # ion quantities-----------------------------------
  rhoi = dsqrt(qr(i-1,6)) + dsqrt(ql(i,6))
  ui(i) = (qr(i-1,7)/dsqrt(qr(i-1,6)) + ql(i,7)/
           dsqrt(ql(i,6))) / rhoi
  vi(i) = (qr(i-1,8)/dsqrt(qr(i-1,6)) + ql(i,8)/
           dsqrt(ql(i,6))) / rhoi
  wi(i) = (qr(i-1,9)/dsqrt(qr(i-1,6)) + ql(i,9)/
           dsqrt(ql(i,6))) / rhoi
  hli = ((5.d0/3.d0)*qr(i-1,10) - (qr(i-1,7)**2 + qr(i-1,8)**2
& + qr(i-1,9)**2)/(3.d0*qr(i-1,6))/dsqrt(qr(i-1,6))
& hri = ((5.d0/3.d0)*ql(i,10) - (ql(i,7)**2 + ql(i,8)**2 +
& ql(i,9)**2)/(3.d0*ql(i,6)))/dsqrt(ql(i,6))
&
& enti(i) = (hli + hri)/rhoi
& qi(i) = ui(i)**2 + vi(i)**2 + wi(i)**2
& ci(i) = dsqrt((2.d0/3.d0)*(enti(i) - 0.5d0*qi(i)))
&
20 continue
&
do 30 i=2-mbc,mx+mbc
&
& # build the matrix of eigenvectors at this interface
&
& # electron block-----------------------------------
&
eig(1,1) = 1.d0
& eig(2,1) = ue(i) - ce(i)
& eig(3,1) = ve(i)
& eig(4,1) = we(i)
& eig(5,1) = ente(i) - ue(i)*ce(i)
& eig(3,2) = 1.d0
& eig(5,2) = ve(i)
& eig(4,3) = 1.d0
& eig(5,3) = we(i)
& eig(1,4) = 1.d0
& eig(2,4) = ue(i)
& eig(3,4) = ve(i)
& eig(4,4) = we(i)
& eig(5,4) = 0.5d0*qe(i)
& eig(1,5) = 1.d0
& eig(2,5) = ue(i) + ce(i)
& eig(3,5) = ve(i)
& eig(4,5) = we(i)
& eig(5,5) = ente(i) + ue(i)*ce(i)
&
& # ion block---------------------------------------
&
eig(6,6) = 1.d0
& eig(7,6) = ui(i) - ci(i)
& eig(8,6) = vi(i)
& eig(9,6) = wi(i)
& eig(10,6) = enti(i) - ui(i)*ci(i)
& eig(8,7) = 1.d0
& eig(10,7) = vi(i)
& eig(9,8) = 1.d0
& eig(10,8) = wi(i)
& eig(6,9) = 1.d0
& eig(7,9) = ui(i)
& eig(8,9) = vi(i)
& eig(9,9) = wi(i)
& eig(10,9) = 0.5d0*qi(i)
& eig(6,10) = 1.d0
& eig(7,10) = ui(i) + ci(i)
\[
eig(8,10) = vi(i)
\]
\[
eig(9,10) = wi(i)
\]
\[
eig(10,10) = enti(i) + ui(i)\ast ci(i)
\]
\[
\text{# electromagnetic block-----------------------------}
\]
\[
eig(13,11) = \text{dsqrt}(0.5d0)
\]
\[
eig(15,11) = -\text{dsqrt}(0.5d0)
\]
\[
eig(12,12) = -\text{dsqrt}(0.5d0)
\]
\[
eig(16,12) = -\text{dsqrt}(0.5d0)
\]
\[
eig(11,13) = 1.d0
\]
\[
eig(14,14) = 1.d0
\]
\[
eig(12,15) = -\text{dsqrt}(0.5d0)
\]
\[
eig(16,15) = \text{dsqrt}(0.5d0)
\]
\[
eig(13,16) = \text{dsqrt}(0.5d0)
\]
\[
eig(15,16) = \text{dsqrt}(0.5d0)
\]
\[
\text{# find the eigenvector coefficients:}
\]
\[
do j=1,meqn
\]
\[
delta(j) = ql(i,j) - qr(i-1,j)
\]
\[
enddo
\]
\[
\text{# electron block coefficients------------------------}
\]
\[
sume = ue(i)*\delta(2) + ve(i)*\delta(3) + we(i)*\delta(4)
\]
\[
alpha(4) = (2.d0/(3.d0*ce(i)**2))*((enti(i) - qe(i))\ast
\]
\[
& \delta(1) + sume - \delta(5))
\]
\[
alpha(2) = \delta(3) - ve(i)*\delta(1)
\]
\[
alpha(3) = \delta(4) - we(i)*\delta(1)
\]
\[
alpha(5) = (\delta(2) + (ce(i) - ue(i))*\delta(1)
\]
\[
& - ce(i)*alpha(4)) / (2.d0*ce(i))
\]
\[
alpha(1) = \delta(1) - alpha(4) - alpha(5)
\]
\[
\text{# ion block coefficients--------------------------}
\]
\[
sumi = ui(i)*\delta(7) + vi(i)*\delta(8) + wi(i)*\delta(9)
\]
\[
alpha(9) = (2.d0/(3.d0*ci(i)**2))*((enti(i) - qi(i))\ast
\]
\[
& \delta(6) + sumi - \delta(10))
\]
\[
alpha(7) = \delta(8) - vi(i)*\delta(6)
\]
\[
alpha(8) = \delta(9) - wi(i)*\delta(6)
\]
\[
alpha(10) = (\delta(7) + (ci(i) - ui(i))*\delta(6)
\]
\[
& - ci(i)*alpha(9)) / (2.d0*ci(i))
\]
\[
alpha(6) = \delta(6) - alpha(9) - alpha(10)
\]
\[
\text{# electromagnetic block coefficients-----------------}
\]
\[
alpha(11) = \text{dsqrt}(0.5d0)*\delta(13) - \delta(15))
\]
\[
alpha(12) = -\text{dsqrt}(0.5d0)*\delta(12) + \delta(16))
\]
\[
alpha(13) = \delta(11)
\]
\[
alpha(14) = \delta(14)
\]
\[
alpha(15) = \text{dsqrt}(0.5d0)*\delta(16) - \delta(12))
\]
\[
alpha(16) = \text{dsqrt}(0.5d0)*\delta(13) + \delta(15))
\]
\[
\text{# Compute the waves.}
\]
\[
do j=1,meqn
\]
\[
\text{wave}(i,j,1) = alpha(1)*eig(j,1)
\]
\[
\text{s}(i,1) = ue(i) - ce(i)
\]
wave(i,j,2) = alpha(2)*eig(j,2)
s(i,2) = ue(i)

wave(i,j,3) = alpha(3)*eig(j,3)
s(i,3) = ue(i)

wave(i,j,4) = alpha(4)*eig(j,4)
s(i,4) = ue(i)

wave(i,j,5) = alpha(5)*eig(j,5)
s(i,5) = ue(i) + ce(i)

wave(i,j,6) = alpha(6)*eig(j,6)
s(i,6) = ui(i) - ci(i)

wave(i,j,7) = alpha(7)*eig(j,7)
s(i,7) = ui(i)

wave(i,j,8) = alpha(8)*eig(j,8)
s(i,8) = ui(i)

wave(i,j,9) = alpha(9)*eig(j,9)
s(i,9) = ui(i)

wave(i,j,10) = alpha(10)*eig(j,10)
s(i,10) = ui(i) + ci(i)

wave(i,j,11) = alpha(11)*eig(j,11)
s(i,11) = -1.d0

wave(i,j,12) = alpha(12)*eig(j,12)
s(i,12) = -1.d0

wave(i,j,13) = alpha(13)*eig(j,13)
s(i,13) = 0.d0

wave(i,j,14) = alpha(14)*eig(j,14)
s(i,14) = 0.d0

wave(i,j,15) = alpha(15)*eig(j,15)
s(i,15) = 1.d0

wave(i,j,16) = alpha(16)*eig(j,16)
s(i,16) = 1.d0

enddo

30 continue

# compute Godunov flux f0:

# amdq = SUM s*wave  over left-going waves
# apdq = SUM s*wave  over right-going waves
do 100 m=1,meqn
    do 100 i=2-mbc, mx+mbc
        amdq(i,m) = 0.d0
        apdq(i,m) = 0.d0
    do 90 mw=1,mwaves
        if (s(i,mw) .lt. 0.d0) then
            amdq(i,m) = amdq(i,m) + s(i,mw)*wave(i,m,mw)
        else
            apdq(i,m) = apdq(i,m) + s(i,mw)*wave(i,m,mw)
        endif
    90    continue
100    continue

continue
return
end
One of the goals of this report is to examine the difference in the results from solving
the time ODE using methods of different accuracy. This code is for a first-order accurate
explicit Euler method.

```fortran
subroutine src1(maxmx,meqn,mbc,mx,xlower,dx,q,maux,aux,t,dt)
  implicit double precision (a-h,o-z)
  dimension q(1:mbc:maxmx+mbc, meqn)
  common /csrc/ ctme,ctmi
  ! Explicit Euler Method
  ! use the exact solution operator
  dimension qstar(1:mbc:maxmx+mbc, meqn)
  do i=1,mx
    qstar(i,1) = 0.d0
    qstar(i,2) = ctme*(q(i,1)*q(i,14) + q(i,3)*q(i,13)
      & - q(i,4)*q(i,12))
    qstar(i,3) = ctme*(q(i,1)*q(i,15) + q(i,4)*q(i,11)
      & - q(i,2)*q(i,13))
    qstar(i,4) = ctme*(q(i,1)*q(i,16) + q(i,2)*q(i,12)
      & - q(i,3)*q(i,11))
    qstar(i,5) = ctme*(q(i,14)*q(i,2) + q(i,15)*q(i,3)
      & + q(i,16)*q(i,4))
    qstar(i,6) = 0.d0
    qstar(i,7) = ctmi*(q(i,6)*q(i,14) + q(i,8)*q(i,13)
      & - q(i,9)*q(i,12))
    qstar(i,8) = ctmi*(q(i,6)*q(i,15) + q(i,9)*q(i,11)
      & - q(i,7)*q(i,13))
    qstar(i,9) = ctmi*(q(i,6)*q(i,16) + q(i,7)*q(i,12)
      & - q(i,8)*q(i,11))
    qstar(i,10) = ctmi*(q(i,14)*q(i,7) + q(i,15)*q(i,8)
      & + q(i,16)*q(i,9))
    qstar(i,11) = 0.d0
    qstar(i,12) = 0.d0
    qstar(i,13) = 0.d0
    qstar(i,14) = -ctmi*q(i,7) - ctme*q(i,2)
    qstar(i,15) = -ctmi*q(i,8) - ctme*q(i,3)
    qstar(i,16) = -ctmi*q(i,9) - ctme*q(i,4)
  enddo
  do j=1,meqn
    q(i,j) = q(i,j) + dt*qstar(i,j)
  enddo
end
```
This code solves the source term ODE using a second-order Runge-Kutta method.

```fortran
subroutine src1(maxmx, meqn, mbc, mx, xlower, dx, q, maux, aux, t, dt)

C implicit double precision (a-h,o-z)
C dimension q(1-mbc:maxmx+mbc, meqn)
C common /csrc/ ctme, ctmi
C
C # Fourth-order Runge-Kutta method.
C # vl* and v2* are the components of the vectors
C # in the Runge-Kutta method.
C # w1-w16 are the components of a temporary vector used as the
C # argument in the source function.
C
do i=1,mx

v11 = 0.d0
v12 = ctme*(q(i,1)*q(i,14) + q(i,3)*q(i,13) - q(i,4)*q(i,12))
v13 = ctme*(q(i,1)*q(i,15) + q(i,4)*q(i,11) - q(i,2)*q(i,13))
v14 = ctme*(q(i,1)*q(i,16) + q(i,2)*q(i,12) - q(i,3)*q(i,11))
v15 = ctme*(q(i,2)*q(i,14) + q(i,3)*q(i,15) + q(i,4)*q(i,11))
v16 = 0.d0
v17 = ctmi*(q(i,6)*q(i,14) + q(i,8)*q(i,13) - q(i,9)*q(i,12))
v18 = ctmi*(q(i,6)*q(i,15) + q(i,9)*q(i,11) - q(i,7)*q(i,13))
v19 = ctmi*(q(i,6)*q(i,16) + q(i,7)*q(i,12) - q(i,8)*q(i,11))
v210 = ctmi*(q(i,7)*q(i,14) + q(i,8)*q(i,15) + q(i,9)*q(i,16))
v11 = 0.d0
v112 = 0.d0
v113 = 0.d0
v114 = -ctme*q(i,2) - ctmi*q(i,7)
v115 = -ctme*q(i,3) - ctmi*q(i,8)
v116 = -ctme*q(i,4) - ctmi*q(i,9)

w1 = q(i,1) + 0.5d0*dt*v11
w2 = q(i,2) + 0.5d0*dt*v12
w3 = q(i,3) + 0.5d0*dt*v13
w4 = q(i,4) + 0.5d0*dt*v14
w5 = q(i,5) + 0.5d0*dt*v15
w6 = q(i,6) + 0.5d0*dt*v16
w7 = q(i,7) + 0.5d0*dt*v17
w8 = q(i,8) + 0.5d0*dt*v18
w9 = q(i,9) + 0.5d0*dt*v19
w10 = q(i,10) + 0.5d0*dt*v110
w11 = q(i,11) + 0.5d0*dt*v111
w12 = q(i,12) + 0.5d0*dt*v112
w13 = q(i,13) + 0.5d0*dt*v113
w14 = q(i,14) + 0.5d0*dt*v114
w15 = q(i,15) + 0.5d0*dt*v115
w16 = q(i,16) + 0.5d0*dt*v116

v21 = 0.d0
v22 = ctme*(w1*w14 + w3*w13 - w4*w12)
v23 = ctme*(w1*w15 + w4*w11 - w2*w13)
v24 = ctme*(w1*w16 + w2*w12 - w3*w11)
v25 = ctme*(w2*w14 + w3*w15 + w4*w16)
```

54
This code solves the source term ODE using a fourth-order Runge-Kutta method.

```
C subroutine src1(maxmx,meqn,mbc, mx,xlower,dx,q,maux, aux,t,dt)
C  
C # Fourth-order Runge-Kutta method.
C # v1*, v2*, v3*, and v4* are the components of the 4 vectors
C # in the Runge-Kutta method.
C # w1-w16 are the components of a temporary vector used as the
C # argument in the source function.
C do i=1,mx
C  
C v11 = 0.d0
C v12 = ctme*(q(i,1)*q(i,14) + q(i,3)*q(i,13) - q(i,4)*q(i,12))
C v13 = ctme*(q(i,1)*q(i,15) + q(i,4)*q(i,11) - q(i,2)*q(i,13))
```

\[ v_{14} = \text{ctme} \cdot (q(i,1) \cdot q(i,16) + q(i,2) \cdot q(i,12) - q(i,3) \cdot q(i,11)) \]
\[ v_{15} = \text{ctme} \cdot (q(i,2) \cdot q(i,14) + q(i,3) \cdot q(i,15) + q(i,4) \cdot q(i,16)) \]
\[ v_{16} = 0.0 \]
\[ v_{17} = \text{ctmi} \cdot (q(i,6) \cdot q(i,14) + q(i,8) \cdot q(i,13) - q(i,9) \cdot q(i,12)) \]
\[ v_{18} = \text{ctmi} \cdot (q(i,6) \cdot q(i,15) + q(i,9) \cdot q(i,11) - q(i,7) \cdot q(i,13)) \]
\[ v_{19} = \text{ctmi} \cdot (q(i,6) \cdot q(i,16) + q(i,7) \cdot q(i,12) - q(i,8) \cdot q(i,11)) \]
\[ v_{210} = \text{ctmi} \cdot (q(i,7) \cdot q(i,14) + q(i,8) \cdot q(i,15) + q(i,9) \cdot q(i,16)) \]
\[ v_{111} = 0.0 \]
\[ v_{112} = 0.0 \]
\[ v_{113} = 0.0 \]
\[ v_{114} = -\text{ctme} \cdot q(i,2) - \text{ctmi} \cdot q(i,7) \]
\[ v_{115} = -\text{ctme} \cdot q(i,3) - \text{ctmi} \cdot q(i,8) \]
\[ v_{116} = -\text{ctme} \cdot q(i,4) - \text{ctmi} \cdot q(i,9) \]
\[ c \]
\[ w_1 = q(i,1) + 0.5d0 \cdot dt \cdot v_{11} \]
\[ w_2 = q(i,2) + 0.5d0 \cdot dt \cdot v_{12} \]
\[ w_3 = q(i,3) + 0.5d0 \cdot dt \cdot v_{13} \]
\[ w_4 = q(i,4) + 0.5d0 \cdot dt \cdot v_{14} \]
\[ w_5 = q(i,5) + 0.5d0 \cdot dt \cdot v_{15} \]
\[ w_6 = q(i,6) + 0.5d0 \cdot dt \cdot v_{16} \]
\[ w_7 = q(i,7) + 0.5d0 \cdot dt \cdot v_{17} \]
\[ w_8 = q(i,8) + 0.5d0 \cdot dt \cdot v_{18} \]
\[ w_9 = q(i,9) + 0.5d0 \cdot dt \cdot v_{19} \]
\[ w_{10} = q(i,10) + 0.5d0 \cdot dt \cdot v_{110} \]
\[ w_{11} = q(i,11) + 0.5d0 \cdot dt \cdot v_{111} \]
\[ w_{12} = q(i,12) + 0.5d0 \cdot dt \cdot v_{112} \]
\[ w_{13} = q(i,13) + 0.5d0 \cdot dt \cdot v_{113} \]
\[ w_{14} = q(i,14) + 0.5d0 \cdot dt \cdot v_{114} \]
\[ w_{15} = q(i,15) + 0.5d0 \cdot dt \cdot v_{115} \]
\[ w_{16} = q(i,16) + 0.5d0 \cdot dt \cdot v_{116} \]
\[ c \]
\[ v_{21} = 0.0 \]
\[ v_{22} = \text{ctme} \cdot (w_1 \cdot w_{14} + w_3 \cdot w_{13} - w_4 \cdot w_{12}) \]
\[ v_{23} = \text{ctme} \cdot (w_1 \cdot w_{15} + w_4 \cdot w_{11} - w_2 \cdot w_{13}) \]
\[ v_{24} = \text{ctme} \cdot (w_1 \cdot w_{16} + w_2 \cdot w_{12} - w_3 \cdot w_{11}) \]
\[ v_{25} = \text{ctme} \cdot (w_2 \cdot w_{14} + w_3 \cdot w_{15} + w_4 \cdot w_{16}) \]
\[ v_{26} = 0.0 \]
\[ v_{27} = \text{ctmi} \cdot (w_6 \cdot w_{14} + w_8 \cdot w_{13} - w_9 \cdot w_{12}) \]
\[ v_{28} = \text{ctmi} \cdot (w_6 \cdot w_{15} + w_9 \cdot w_{11} - w_7 \cdot w_{13}) \]
\[ v_{29} = \text{ctmi} \cdot (w_6 \cdot w_{16} + w_7 \cdot w_{12} - w_8 \cdot w_{11}) \]
\[ v_{30} = \text{ctmi} \cdot (w_7 \cdot w_{14} + w_8 \cdot w_{15} + w_9 \cdot w_{16}) \]
\[ v_{31} = 0.0 \]
\[ v_{32} = 0.0 \]
\[ v_{33} = 0.0 \]
\[ c \]
\[ w_1 = q(i,1) + 0.5d0 \cdot dt \cdot v_{21} \]
\[ w_2 = q(i,2) + 0.5d0 \cdot dt \cdot v_{22} \]
\[ w_3 = q(i,3) + 0.5d0 \cdot dt \cdot v_{23} \]
\[ w_4 = q(i,4) + 0.5d0 \cdot dt \cdot v_{24} \]
\[ w_5 = q(i,5) + 0.5d0 \cdot dt \cdot v_{25} \]
\[ w_6 = q(i,6) + 0.5d0 \cdot dt \cdot v_{26} \]
\[ w_7 = q(i,7) + 0.5d0 \cdot dt \cdot v_{27} \]
\[ \begin{align*}
w8 &= q(i,8) + 0.5d0*dt*v28 \\
w9 &= q(i,9) + 0.5d0*dt*v29 \\
w10 &= q(i,10) + 0.5d0*dt*v210 \\
w11 &= q(i,11) + 0.5d0*dt*v211 \\
w12 &= q(i,12) + 0.5d0*dt*v212 \\
w13 &= q(i,13) + 0.5d0*dt*v213 \\
w14 &= q(i,14) + 0.5d0*dt*v214 \\
w15 &= q(i,15) + 0.5d0*dt*v215 \\
w16 &= q(i,16) + 0.5d0*dt*v216 \\
\end{align*} \]
v412 = 0.d0
v413 = 0.d0
v414 = -ctme*w2 - ctmi*w7
v415 = -ctme*w3 - ctmi*w8
v416 = -ctme*w4 - ctmi*w9

c
q(i,1) = q(i,1) + (dt/6.d0)*(v11 + 2.d0*v21 + 2.d0*v31 + v41)
q(i,2) = q(i,2) + (dt/6.d0)*(v12 + 2.d0*v22 + 2.d0*v32 + v42)
q(i,3) = q(i,3) + (dt/6.d0)*(v13 + 2.d0*v23 + 2.d0*v33 + v43)
q(i,4) = q(i,4) + (dt/6.d0)*(v14 + 2.d0*v24 + 2.d0*v34 + v44)
q(i,5) = q(i,5) + (dt/6.d0)*(v15 + 2.d0*v25 + 2.d0*v35 + v45)
q(i,6) = q(i,6) + (dt/6.d0)*(v16 + 2.d0*v26 + 2.d0*v36 + v46)
q(i,7) = q(i,7) + (dt/6.d0)*(v17 + 2.d0*v27 + 2.d0*v37 + v47)
q(i,8) = q(i,8) + (dt/6.d0)*(v18 + 2.d0*v28 + 2.d0*v38 + v48)
q(i,9) = q(i,9) + (dt/6.d0)*(v19 + 2.d0*v29 + 2.d0*v39 + v49)
q(i,10) = q(i,10) + (dt/6.d0)*(v110 + 2.d0*v210 + 2.d0*v310
  & + v410)
q(i,11) = q(i,11) + (dt/6.d0)*(v111 + 2.d0*v211 + 2.d0*v311
  & + v411)
q(i,12) = q(i,12) + (dt/6.d0)*(v112 + 2.d0*v212 + 2.d0*v312
  & + v412)
q(i,13) = q(i,13) + (dt/6.d0)*(v113 + 2.d0*v213 + 2.d0*v313
  & + v413)
q(i,14) = q(i,14) + (dt/6.d0)*(v114 + 2.d0*v214 + 2.d0*v314
  & + v414)
q(i,15) = q(i,15) + (dt/6.d0)*(v115 + 2.d0*v215 + 2.d0*v315
  & + v415)
q(i,16) = q(i,16) + (dt/6.d0)*(v116 + 2.d0*v216 + 2.d0*v316
  & + v416)
c
c
return
end