A 13-Moment Two-Fluid Plasma Physics Model Based on a Pearson Type-IV Distribution Function

Shaun Gilliam

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

Master of Science in Aeronautics and Astronautics

University of Washington

2011

Program Authorized to Offer Degree: Aeronautics & Astronautics

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Shaun Gilliam

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

Committee Members:

____
Uri Shumlak

____
Brian Nelson

Date: ______________________

University of Washington

Abstract

A 13-Moment Two-Fluid Plasma Physics Model Based on a Pearson Type-IV Distribution Function

Shaun Gilliam

Chair of the Supervisory Committee:
Professor Dr. Uri Shumlak
Aeronautics & Astronautics

The two-fluid plasma model describes ions and electrons as two inter-penetrating fluids with Maxwell's equations describing the electric and magnetic fields. Typically, the two-fluid model is based on a 5-moment model. The 13-moment model allows for the simulataneous calculation of fluid and heat transfer. A 13-moment model is derived in three dimensions for a Pearson-IV distribution function and the eigensystem for this three dimensional system is analyzed. Models based on a Pearson-IV distribution function have a recursion relationship that relates higher moments to lower as well as an extended region of hyperbolicity. A simple, one dimensional model is implemented as a second order finite volume simulation to study the system of equations. The differences of the 5 and 13-moment models are investigated for electrostatic two-fluid plasma simulations.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The author would like to express sincere gratitude, first and foremost, to Professor Uri Shumlak for his expertise and guidance in computational plasma physics. The author is also greatful for the time and support of reading committee member Professor Brian Nelson. The author would also like to thank Dr. Manuel Torrilhon for his assistance in understanding his research. This thesis would not have been possible without all their help. The author would like to thank fellow lab members Robert Lilly, Eder Sousa, and Noah Reddell for the discussions that added so much to this thesis. The author would also like to give a special thanks to his parents, Mike and Wanda Gilliam, and his brother, Shane Gilliam, for their lifelong support. Also a great many thanks to the great friends that made graduate school that much better.

Chapter 1
INTRODUCTION

1.1 Overview

At its lowest energy state, matter exists as a solid. As energy is added to it, the covalent, metallic, etc. bonds connecting the atoms slowly break apart until it becomes a liquid and then eventually a gas. If enough energy is added that the atoms smashing into each other begin to break off individual electrons, it becomes a plasma-the fourth state of matter. Two examples of plasmas found in nature are the solar winds ejected from the sun and lightning strikes which heat the surrounding air so much that it turns into plasma. Plasmas are also used in the development of nuclear fusion confinement devices and in space propulsion. What makes this possible is that plasmas can be and are affected by electric and magnetic fields, which can either be externally applied or self generated.

Most gas dynamics problems deal with interactions of individual particles colliding with each other, but the interaction of plasmas with electric and magnetic fields means that they can be affected at much greater distances. This allows for very interesting physics, such as how the hot plasmas of fusion could be contained without making physical contact with a wall. This interesting physics creates unique challenges, however, since plasmas involve physical phenomena of such vastly different time and spatial scales.

A number of analytical tools have been developed to study plasmas. The particle model describes the position and velocity of every particle in a plasma. The Lorentz forces on a particle couple position and velocity with the electric and magnetic fields, which can be calculated by solving Maxwell's equations. A so called PIC, or particle-in-cell, code does exactly that by keeping track of every particle of interest and calculating its interaction with every other particle. It is easy to see how, with this method, a small increase in a problem's complexity can correspond to a massive increase in computational time required. While there are ways to mitigate this, such as using "super-particles" which represent averages of large numbers of individual particles- it is still a fundamentally expensive method of solving problems.

At the opposite end of the spectrum is a single-fluid description of a plasma, known as the MagnetoHydroDynamic or MHD model. The MHD model essentially assumes that plasmas behave as a single, conductive fluid. This is a simple and computationally efficient way to model bulk plasma effects, but it fails to capture phenomenon such as electron inertia, electron waves, and ion currents. Expansions have been developed such as Hall-MHD, where Hall effects are included in the model, but there are still fundamental assumptions made to MHD that will not capture important physics effects.

A good compromise between the computationally expensive particle model and the basic MHD model is the two-fluid plasma model. The two-fluid plasma model assumes that the ions and electrons form two intermixing fluids that can interact with each other. This allows for charge separation, which leads to physics like electrostatic effects, while remaining a fluid code and thus fairly efficient to solve. The objective of this work is to derive a particular two-fluid plasma model in 3D and showcase a few of its features in 1D.

1.2 13-Moment Equations

The first step in any analytical fluid approach is to derive the governing system of equations. While there are many ways to do this, the method chosen for this work is Grad's method of moments [1]. The method of moments takes moments of a differential equation that describes the evolution of a distribution function in time and space. Two common approaches in plasma sciences are to use either the Vlasov equation or the Boltzmann equation as a base; with the results in this paper centered around the Boltzmann equation.

Moments are a form of statistical averaging that estimate parameters of a function. Moments of the distribution function, for example, can determine velocity, pressure, etc. Whereas moments of the Boltzmann equation formally derive the governing equations for continuity, momentum, etc. The derivation of these systems of equations are covered in more detail in Chapter 3. In general terms, moments of the Boltzmann equation contain at a minimum an evolution of time and an evolution in space called the flux. Moment equations in this paper are referred to by the term being evolved in time. In the classical continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{1.1}$$

for example, the general principle behind the conservation of mass (or continuity) is represented by the evolution of the density, or concentration of mass at a particular time. It is also noted that the second term, the flux, contains information from the next moment in the sequence, the momentum.

In theory, every moment equation requires some information from the next moment in the series so that any system of equations requires an infinite number of moments to express. In practice however, we use closure schemes that approximate the behavior of these higher moments. Where this cutoff point is made determines just how robust the moment method is. Extensive references are made in this paper to the 5 and 13-moment equations, where the five and thirteen refer to the number of variables and equations necessary to describe the system. The 5-moment system is the familiar Euler equations where the five variables are density, the three directions of momentum, and the total energy. The 13-moment system has expressions for the density (one variable), momenta (three variables), the pressure (six variables), and the heat flux (three variables) for a total of thirteen.

The thirteen-moment model has a few differences from the five moment model. Most notably, the thirteen moment model directly resolves the transfer of energy as heat. Most codes based on the five moment model that attempt to solve for heat transfer must do so separately from the fluid equations through an auxiliary relation. Another property of moment equations is hyperbolicity. Formally, hyperbolicity is defined mathematically as a region where the flux quantities of the system have only real eigenvalues. More simply, hyperbolicity is a measure of the degree to which the system of equations are valid for a certain set of parameters. The original 13-moments of Grad had a limited region of hyperbolicity centered around an equilibrium state, but Torrilhon [2] proposes using a new distribution function that allows for a much greater region of hyperbolicity. For fluid equations, this means that the system of equations can handle conditions that are significantly far away from equilibrium.

1.3 Two-Fluid Plasma Model

The two-fluid plasma model has been studied extensively by Shumlak, et al. [3, 16, 21]. The model works by assuming that the plasma is described by two inter-penetrating fluids that are individually in thermal equilibrium. These fluids are modeled using the 5-moment Euler equations plus Lorentz force terms and Maxwell's equations to resolve the electromagnetic dynamics. The fluid equations are expressed as conservation of mass, momentum, and energy:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{v} = 0, \tag{1.2}$$

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla \cdot (p + \rho \vec{v}\vec{v}) = \frac{\rho q}{m}\left(\vec{E} + \vec{v} \times \vec{B}\right), \tag{1.3}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (p\vec{v} + \varepsilon \vec{v}) = \frac{\rho q}{m}\left(\vec{v} \cdot \vec{E}\right), \tag{1.4}$$

where $\vec{E}$ and $\vec{B}$ are the electric and magnetic fields, ρ is the density, $\vec{v}$ is the velocity, p is the scalar pressure, q is the charge, m is the mass, and ε is the total energy of the fluid. The total energy here represents the simplifying assumption of the 5-moment model and it is described by the equation

$$\varepsilon = \frac{1}{\gamma - 1} p + \frac{1}{2} \rho \vec{v} \cdot \vec{v}, \tag{1.5}$$

where $\gamma = 5/3$ for the remainder of this paper. To fully describe the physics of the two-fluid plasma model, equations are needed to resolve the electromagnetic fields that are developed by the moving charges in a plasma. These are fully described by Maxwell's equations:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}, \tag{1.6}$$

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t}, \tag{1.7}$$

$$\nabla \cdot \vec{E} = \frac{\rho_c}{\epsilon_0}, \tag{1.8}$$

$$\nabla \cdot \vec{B} = 0. \tag{1.9}$$

Here μ_0 is the permeability of free space, $\vec{J}$ is the current, c is the speed of light, ρ_c is the charge separation, and ϵ_0 is the permittivity of free space. While these equations fully describe the electromagnetic equations, methods of solving them can be a research topic in and of themselves. In this paper we limit ourselves to Gauss's law (Eq. 1.8) and study only electrostatic phenomenon. This simplification allows us to study the source term contributions without the complexity of solving all of Maxwell's equations.

1.4 Objectives

The objective of this paper is to blend the advanced physics captured by the 13-moment fluid model with the advanced physics made possible by the two-fluid plasma scheme. Towards this end, a review is made of the distribution function properties to understand how they are used in the derivation of the 13-moment model. Next, the full

13-moment model is derived by taking moments of the Boltzmann equation resulting in a three-dimensional system of equations. This model is investigated to find the eigensystem that will allow for the solution to the model in three dimensions. Further, the model is reduced to a one-dimensional example to demonstrate the properties of the 13-moment model as both a single and two-fluid simulation.

Chapter 2
PEARSON-IV DISTRIBUTION REVIEW

A new 13-moment model is suggested by Torrilhon [2] that is based on a Pearson-IV distribution function. The Pearson Type-IV distribution function is described in one dimension by Pearson in his original paper [6] and is similar to a Maxwellian distribution function that allows for skewness. An interesting property of the Pearson-IV distribution is that high order moments can be related to lower moments through recursion relations. This section defines the Pearson-IV distribution function in one ($f_\alpha^{(P1)}$) and three dimensions ($f_\alpha^{(P3)}$) based on a velocity space. A detailed derivation of the moment calculations, based on the work in Torrilhon [2], can be found in Appendix A. This section only presents the salient results that are needed to develop closure schemes that are the mathematical foundation of the 13-Moment model.

2.1 Pearson Type-IV Distribution in One Dimension

The one dimensional case is studied first to illustrate the basic properties of the distribution function and how it might be applied. The traditional Pearson-IV distribution is one dimensional, with five real parameters

Parameter	Definition
λ:	Translation from the center of the distribution function
a :	Scale of the distribution function (always > 0)
ν :	Measure of the skewness
m :	Shape factor (always > 0)
k :	Normalization constant for the 1D function

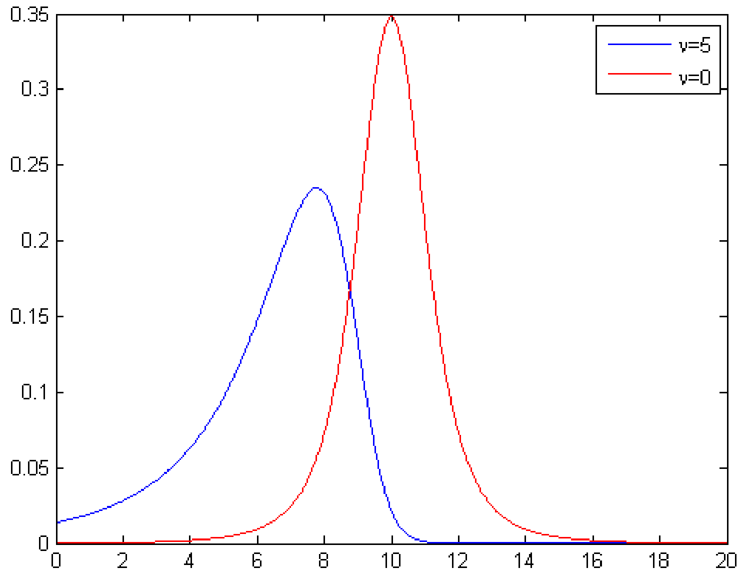


Figure 2.1: Pearson-IV distribution function demonstrating the effect of skewness. Other parameters are $\lambda = 10$, $a = 2$, $m = 2.25$, and k determined by the trapezoidal integration method.

The normalized distribution function in one dimension is defined as

$$f^{(P1)}(c) = \frac{1}{a\,k} \frac{e^{-\nu \tan^{-1}(\frac{c-\lambda}{a})}}{\left[1 + (\frac{c-\lambda}{a})^2\right]^m}. \tag{2.1}$$

Plots of the distribution function based on different parameters are seen in Fig. 2.1.

2.1.1 Moments of the Pearson-IV Distribution Function

In mathematics, moments of a distribution function are used to characterize its general shape, such as the mean, skewness, kurtosis, etc. These are paralleled in the method of moments in physics, which uses moments of a kinetic description of a distribution function to formally develop average variables. The general equation for a moment where n represents the n-th moment, x represents the domain and μ_n the mean value of the n-th moment is

$$\mu_n = \int_{-\infty}^{\infty} (x - \mu_1)^n f(x)\, dx. \tag{2.2}$$

There are two types of moments in mathematics, central and raw. Central moments (as in Eq. 2.2) are defined as the spread and shape of the difference from an average value. Raw moments are defined about the axis and are used to determine this average value μ_1, the average or bulk velocity v, where

$$v \equiv \mu_1 = \frac{1}{a\,k} \int\limits_{-\infty}^{\infty} c \frac{e^{-\nu \tan^{-1}(\frac{c-\lambda}{a})}}{\left[1 + (\frac{c-\lambda}{a})^2\right]^m} dc. \tag{2.3}$$

Using Pearson's suggested simplification $r = 2(m-1)$ and integrating gives the average velocity

$$v = \lambda + \frac{a\,\nu}{r}. \tag{2.4}$$

General moments that are taken beyond this are central moments about this average velocity. Pearson [6] developed a recursion relation for $n \geq 2$ that allows for these moments to be defined in terms of earlier moments in the sequence

$$\mu_n = \frac{a\,(n-1)}{(r+1-n)} \left\{ \left[1 + \left(\frac{\nu}{r}\right)^2\right] a\mu_{n-2} - 2\frac{\nu}{r}\mu_{n-1} \right\}. \tag{2.5}$$

This relationship holds for all moments with the initial starting conditions $\mu_0 = 1$ and $\mu_1 = 0$. The fact that $\mu_1 = 0$ for the central moments allows for higher moments to be categorized by their variance from the average instead of their actual value. Physically, this means that higher moment equations do not need information about the average velocity and instead deal strictly with the independent velocities.

2.1.2 Reduced Third and Fourth Moments

Physically, the second moment is the temperature. To reduce the third and fourth moments to products of the second moment, non-dimensional ratios are introduced

$$\theta = \mu_2, \;\; Q = \frac{\mu_3}{\mu_2^{3/2}}, \;\; D = \frac{\mu_4}{\mu_2^2}. \tag{2.6}$$

The moments found using the recursion Eq. 2.5 are given as

$$\begin{aligned} \mu_2 &= \frac{a^2\left(1+\frac{\nu^2}{r^2}\right)}{r-1}, \\ \mu_3 &= -\frac{4a^3\nu\left(r^2+\nu^2\right)}{r^3\left(2-3r+r^2\right)}, \\ \mu_4 &= \frac{3a^4\left((-2+r)r^4+2r^2(2+r)\nu^2+(6+r)\nu^4\right)}{(-3+r)(-2+r)(-1+r)r^4}. \end{aligned} \tag{2.7}$$

By substitution, the non-dimensional ratios and temperature then become

$$\begin{aligned} \theta &= \frac{a^2\left(1+\frac{\nu^2}{r^2}\right)}{-1+r}, \\ Q &= \frac{\mu_3}{\mu_2^{3/2}} = -\frac{4a\nu}{(-2+r)r\sqrt{\frac{a^2(r^2+\nu^2)}{(-1+r)r^2}}}, \\ D &= \frac{\mu_4}{\mu_2^2} = \frac{3(-1+r)\left((-2+r)r^2+(6+r)\nu^2\right)}{(-3+r)(-2+r)\left(r^2+\nu^2\right)}. \end{aligned} \tag{2.8}$$

Solving for the parameters of the distribution function by inverting the relations of Eq. 2.6 (Appendix A) yields, with the aid of Mathematica

$$\begin{aligned}
r &= \frac{6\left(-1+D-Q^2\right)}{-6+2D-3Q^2}, \\
\nu &= -\frac{Q(-2+r)r}{\sqrt{-Q^2(-2+r)^2+16(-1+r)}}, \\
a &= \frac{1}{4}\sqrt{\theta}\sqrt{-\left(Q^2(-2+r)^2-16(-1+r)\right)}, \\
\lambda &= v+\frac{1}{4}Q(-2+r)\sqrt{\theta}.
\end{aligned} \tag{2.9}$$

2.1.3 Critical Values of D and Q

A realizable Pearson-IV distribution is defined as one for which the parameters are limited to real ν and a , and an m greater than zero. For real ν and a,

$$-Q^2(-2+r)^2+16(-1+r)\geq 0. \tag{2.10}$$

Substituting the expression for r into this equation and solving for D yields the critical values

$$D \geq D_{critical} = \frac{48+39Q^2+6\sqrt{\left(4+Q^2\right)^3}}{32-Q^2} \text{ and } Q^2 \leq 32. \tag{2.11}$$

The first four moments of the Pearson-IV require a closure scheme for the highest moment. Two cases are suggested by Torrilhon [2]. The first is an arbitrary expression that is close to the critical value, but always in the realizable region,

$$\begin{aligned}
D_{realizable} &= \frac{48+39Q^2+3\sqrt{\left(4+Q^2\right)^4}}{32-Q^2} \\
&= 3\left(1+\frac{Q^2\left(22+Q^2\right)}{32-Q^2}\right).
\end{aligned} \tag{2.12}$$

The second is the singular case which does not preserve realizability, but represents similar physics to Grad and Maximum-Entropy closures. A singular distribution is neither a discrete probability distribution nor a probability distribution function, but a special case. It is calculated by solving for D in terms of Q and r and taking the limit as $r \to \infty$,

$$\begin{aligned} D_{singular} &= \lim_{r\to\infty} \frac{3\,(-2-2Q^2+2r+Q^2r)}{2(-3+r)} \\ &= 3\left(1+\frac{1}{2}Q^2\right). \end{aligned} \tag{2.13}$$

The singular closure is a fair approximation of the realizable scheme at points close to equilibrium, and has more modest wave speeds.

2.2 Pearson Type-IV Distribution in Three Dimensions

The Pearson-IV distribution in three dimensions requires modifications and additions to the parameters of the traditional Pearson-IV distribution.

Parameter	Definition
$\vec{c}$:	The velocity is now a vector in three dimensional phase space.
$\vec{\lambda}$:	The translation of the distribution in phase space (vector).
$\overleftrightarrow{A}$:	Scale, a symmetric positive definite 3x3 matrix.
$\hat{n}$:	A direction of skewness (unit vector).
K :	Normalization constant for the 3D function.
ν :	Magnitude of skewness.
m :	Shape factor.

The normalized distribution function in 3-dimensions[2] is

$$f^{(P3)}(\vec{c}) = \frac{1}{\det(\overleftrightarrow{A})\,K} \frac{e^{-\nu\tan^{-1}\left(n^T\left(\overleftrightarrow{A}\right)^{-1}(\vec{c}-\vec{\lambda})\right)}}{\left[1+\left(\vec{c}-\vec{\lambda}\right)^T\left(\overleftrightarrow{A}\overleftrightarrow{A}\right)^{-1}\left(\vec{c}-\vec{\lambda}\right)\right]^m}. \tag{2.14}$$

2.2.1 Moments of the 3D Pearson-IV Distribution Function

Expressions for the moments of three dimensional functions can be quite complex, so for simplicity the remainder of the derivations will be carried out in indicial notation. A continuum mechanics book (e.g. [10]) will have more details on indicial notation, but in short it is a way to represent vector operations more succinctly. For example,

$$\vec{F} \cdot \frac{\partial f_\alpha}{\partial \vec{v}} \rightarrow F_i \partial_{v_i} f_\alpha, \tag{2.15}$$

where the repeated i index represents an implied sum

$$F_i \partial_{v_i} = \sum_{i=1}^{3} F_i \partial_{v_i}. \tag{2.16}$$

Similarly to the 1D case, there is an expression for general moments,

$$M_{i_1 \cdots i_n} = \iiint_{\mathbb{R}^3} (c_{i_1} - \nu_{i_1}) \cdots (c_{i_n} - \nu_{i_n}) f^{(P3)} d\vec{c}, \tag{2.17}$$

where n represents the order of the tensor, $c_{i \cdots n}$ the independent velocities in each principal direction, and $\nu_{i \dots n}$ the skewness in each principal direction. This integral does not evaluate easily, however, and Torrilhon [2] recommends using an extension of the one-dimensional moment equation to three dimensions instead. Here, the coordinate system is rotated (with the local, rotated vector components represented with tilde) such that the skewness unit vector is purely in the direction of $\tilde{n}_1$ and consequently $\tilde{n}_2$ and $\tilde{n}_3$ are zero. The random velocities are chosen such that $\tilde{c}_1$ is in the direction of $\tilde{n}_1$ and that $\tilde{c}_1$, $\tilde{c}_2$ and $\tilde{c}_3$ are orthogonal to each other. The simplification $\tilde{v} = -\frac{\nu}{r}\hat{n}$ is also introduced. This forms the generalized moment equation in three dimensions seen in [2],

$$\tilde{\mu}_n^{p,q} = \frac{1}{K} \iiint_{\mathbb{R}^3} (\tilde{c}_1 - \tilde{v}_1)^n \tilde{c}_2^p \tilde{c}_3^q \frac{\exp(-\nu \arctan(\tilde{c}_1))}{(1 + \tilde{c}_1^2 + \tilde{c}_2^2 + \tilde{c}_3^2)} d\tilde{c}, \tag{2.18}$$

where in this new, rotated frame n represents the order of the moments in the skew direction and p and q the order in the tangential directions. A recursion relation similar to Eq. 2.5 for the higher order moments is shown in detail in [2], with the result for $n \geq 1$

$$\tilde{\mu}_n = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{\nu}{r}\right)^2\right)\tilde{\mu}^{p,q}_{n-2} - (2(n-1)+p+q)\frac{\nu}{r}\tilde{\mu}^{p,q}_{n-1}\right), \tag{2.19}$$

where

$$\tilde{\mu}^{p,q}_{-1} = 0, \tag{2.20}$$

$$\tilde{\mu}^{p,q}_{0} = (p-1)!!(q-1)!!\prod_{k=1}^{\frac{p+q}{2}}\frac{1+\frac{\nu^2}{(2-2k+r)^2}}{1-2k+r}. \tag{2.21}$$

2.2.2 Reduced 3rd and 4th Moments

To relate higher moments to the second as before, it is necessary to define the temperature as a tensor,

$$M_{ij} = \Theta_{ij} = \mu^{0,0}_2 A^2 = \frac{1+\left(\frac{\nu}{r}\right)^2}{r-1}A^2, \tag{2.22}$$

where A^2 represents the matrix product of $\overleftrightarrow{A}$ with itself. The third moment is defined by using the recursion relation

$$M_{ijk} = \mu^{0,0}_3 A_{il}A_{jp}A_{kq}\left(-\frac{1}{2}n_l n_p n_q + \frac{3}{2}n_{(l}\delta_{pq)}\right). \tag{2.23}$$

For convenience, $n_{(l}\delta_{pq)}$ uses the same notation as the symmetrization of a tensor [7, 8],

$$T_{(i_1 i_2 \ldots i_r)} \quad = \quad \frac{1}{r!}\sum_{Permutations} T_{(i_1 i_2 \ldots i_r, i_r \ldots i_2 i_1, etc\ldots)}, \tag{2.24}$$

where the i's represent the indices of a tensor up to order r. This notation allows the ordered permutations of terms to be grouped together, as can be seen in the expansion of

$$3n_{(l}\delta_{pq)} \Rightarrow (n_l\delta_{pq} + n_q\delta_{lp} + n_p\delta_{ql}) \,. \tag{2.25}$$

Symmetrizations of the fourth moment can be found in Appendix A. The fourth moment is similarly defined as

$$M_{ijkl} = A_{il}A_{jp}A_{kq}A_{ls}\left[\left(\mu_0^{4,0} - \mu_4^{0,0}\right) n_l n_p n_q n_s - 2\left(\mu_0^{4,0} - \mu_4^{0,0}\right) n_{(l}n_p\delta_{qs)} + \mu_0^{4,0}\delta_{(lp}\delta_{qs)}\right] . \tag{2.26}$$

Non-dimensional quantities are introduced to reduce the third and fourth moments

$$Q = \frac{\mu_3^{0,0}}{\left(\mu_2^{0,0}\right)^{3/2}} = \frac{4\nu}{r-2}\sqrt{\frac{r-1}{r^2+\nu^2}}, \tag{2.27}$$

$$D = \frac{\mu_4^{0,0}}{\left(\mu_2^{0,0}\right)^{2}} = \frac{3\,(r-1)}{(r-2)\,(r-3)}\left(6 + r - \frac{8r^2}{r^2+\nu^2}\right) . \tag{2.28}$$

An additional definition is used to simplify the expressions,

$$N_i = \Theta_{ij}^{1/2} n_j, \tag{2.29}$$

where $\Theta^{1/2} \cdot \Theta^{1/2} = \Theta$. Substituting into the third moment equation yields

$$M_{ijk} = \frac{1}{2}Q\left(-N_iN_jN_k + 3N_{(l}\Theta_{jk)}\right) . \tag{2.30}$$

Appendix A shows that $\mu_4^{0,0}$ is a multiple of $\mu_0^{4,0}$,

$$\mu_0^{4,0} = \Delta\mu_4^{0,0}. \tag{2.31}$$

Evaluating using the recursion expression,

$$\mu_0^{4,0} = \frac{3\left((-2+r)^2+\nu^2\right)\left(r^2+\nu^2\right)}{(-3+r)(-2+r)^2(-1+r)r^2}, \tag{2.32}$$

$$\mu_4^{0,0} = \frac{3\left(r^2+\nu^2\right)\left((-2+r)r^2+(6+r)\nu^2\right)}{(-3+r)(-2+r)(-1+r)r^4}, \tag{2.33}$$

and solving for Δ yields

$$\Delta = \frac{r^2\left(4-4r+r^2+\nu^2\right)}{(-2+r)\left(-2r^2+r^3+6v^2+r\nu^2\right)}. \tag{2.34}$$

Eliminating ν and r using Gaussian elimination (Appendix A) gives

$$\Delta = 1 - \frac{3}{4}\frac{Q^2}{D} \tag{2.35}$$

or

$$\mu_0^{4,0} = \left(1 - \frac{3}{4}\frac{Q^2}{D}\right)\mu_4^{0,0}. \tag{2.36}$$

Substituting into the equation for the fourth moment gives

$$M_{ijkl} = \left(D - \frac{3}{4}Q^2\right)\Theta_{(ij}\Theta_{kl)} + \frac{3}{4}Q^2\left(-N_iN_jN_kN_l + 2N_{(i}N_j\Theta_{kl)}\right). \tag{2.37}$$

The realizable closure scheme (Eq. 2.12) and the singular closure scheme (Eq. 2.13) from the one dimensional case are again used in the three-dimensional case. In this section, moments of a Pearson-IV distribution function for both the one and three-dimensional cases are derived. Closure schemes that relate high order moments to lower order moments are calculated based on the unique properties of the Pearson-IV distribution function. The information from these moments and their closures will play a critical role in the derivation of the 13-moment system of equations.

Chapter 3
13-MOMENT MODEL DERIVATION

This chapter presents the derivation of the 13-moment model using Grad's method of moments [1]. The fluid variables are derived by taking the moments of the distribution function, while the governing equations are derived by taking moments of a kinetic model. In a two-fluid plasma model, expressions that govern the gas dynamics are combined with Lorentz force terms that couple the fluid to the electromagnetic fields for each species (α). Also included are collision terms that represent an exchange of energy between each species. At each moment of the kinetic model, information from higher moments is required. Closure schemes are presented that make assumptions about these higher moments based on the properties of the Pearson-IV distribution function.

3.1 Moments of the Distribution Function

The fluid variables are formally defined by taking moments of the distribution function $f\left(c_i\right)$ in velocity space, where c_i is the particle velocity. The 13-moment model requires the first four (zeroth - third) moments to define the fluid variables. The zeroth moment of the distribution function

$$\rho = m \iiint_{\mathbb{R}^3} f^{(P3)}\left(c_i\right) d\vec{c}, \tag{3.1}$$

gives the expression for density of each species as a volume integral in phase space. To simplify the expressions in the remainder of this chapter, the triple integrals in phase space will only be represented by a single integral sign. The distinction of the distribution function being three-dimensional is also implied throughout the rest of this chapter and dropped from the notation. The first moment of the distribution function,

$$\rho v_i = m \int c_i f d\vec{c}, \tag{3.2}$$

formally defines the momentum as well as the average velocity v_i. Moments of higher orders are defined by using central moments about the average velocity. This random velocity is denoted by $w_i = c_i - v_i$ and can be seen in the second moment of the distribution function

$$P_{ij} = m \int w_i w_j f d\vec{c}, \tag{3.3}$$

which describes the pressure tensor. The third moment defines the heat flux tensor

$$h_{ijk} = m \int w_i w_j w_k f d\vec{c}. \tag{3.4}$$

The fourth moment of the distribution function,

$$\Delta_{ijkm} = m \int w_i w_j w_k w_m f d\vec{c}, \tag{3.5}$$

is not a fluid variable of the final 13-moment equations, but a variable that will arise in the derivation of the equations of motion. A closure scheme is used to relate this term to the other, lower-order variables.

Other moments arise in the derivation that are either a reduced form of the fluid variables or related to them in some way. The quasi-heat flux vector,

$$q_i = m \int \frac{1}{2} w^2 w_i f d\vec{c}, \tag{3.6}$$

represents the heat transfer in each of the principal directions. Scalar pressure represents the average of the diagonal pressure terms

$$3p = p_{kk}. \tag{3.7}$$

Implicit summation is indicated by the repeated indices as per Eq. 2.16 and will be used extensively throughout this chapter. Using the ideal gas law, the temperature tensor is related to the pressure and density by

$$\Theta_{ij} = \frac{p_{ij}}{\rho} \tag{3.8}$$

and the scalar temperature is the sum of the diagonal temperature terms

$$\theta = 3\Theta_{kk}. \tag{3.9}$$

3.2 Moments of the Boltzmann Equation

As defined by Grad's method of moments [1], the equations of motion are formally derived by taking the raw moments of the Boltzmann equation

$$m\frac{\partial f}{\partial t} + m\vec{c}\cdot\frac{\partial f}{\partial \vec{x}} + q_\alpha(\vec{E} + \vec{c}\times\vec{B})\cdot\frac{\partial f}{\partial \vec{c}} = m\frac{\partial f}{\partial t}\bigg|_{Collisions}, \tag{3.10}$$

which describes the evolution of a distribution function in time and space where q_α is the charge of the species, $\vec{E}$ the electric field, and $\vec{B}$ the magnetic field. For the rest of the derivation, the Boltzmann equation in indicial notation will be used;

$$m\partial_t f + mc_i\partial_{x_i} f + q_\alpha(E_i + \varepsilon_{ijk}c_jB_k)\partial_{c_i} f = m\partial_t f\Big|_c. \tag{3.11}$$

The cross product is represented by the permutation symbol ε_{ijk} as defined [10] by

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{if numerical values of } ijk \text{appear as in the sequence 12312} \\ -1 & \text{if numerical values of } ijk \text{appear as in the sequence 32132} \\ 0 & \text{if numerical values of } ijk \text{appear in any other sequence} \end{cases} \cdot \tag{3.12}$$

3.2.1 Zeroth Moment

Taking the zeroth moment of the Boltzmann equation formally derives the equation of continuity

$$\underbrace{\int m\partial_t f d\vec{c}}_{①} + \underbrace{\int mc_i\partial_{x_i} f d\vec{c}}_{②} + \underbrace{\int q_\alpha(E_i + \varepsilon_{ijk}c_jB_k)\partial_{c_i} f d\vec{c}}_{③} = \underbrace{\int m\partial_t f\Big|_c d\vec{c}}_{④}. \tag{3.13}$$

Each integral (①, ②, etc.) is examined individually. Integral ① is written as

$$① = \partial_t m \int f_\alpha d\vec{c} = \partial_t \rho. \tag{3.14}$$

Eq. 3.14 shows the use of mass density defined by Eq. 3.1 to relate a function of the distribution function to a classically understood variable. Applying the product rule to the integrand of the second integral, $\partial_{x_i}(c_i f) = c_i\partial_{x_i} f + f\partial_{x_i} c_i$, gives

$$\begin{aligned} ② &= m\int [\partial_{x_i}(c_i f) - f\partial_{x_i} c_i] d\vec{c} \\ &= m\int [\partial_{x_i}(c_i f) - \cancelto{0}{f\partial_{x_i} c_i}] d\vec{c}. \end{aligned} \tag{3.15}$$

The second term vanishes because, in phase space, particle velocity c_i is independent of x. Again, a fluid variable definition (Eq. 3.2) is used to simplify the integral of the distribution function,

$$\textcircled{2} \;=\; \partial_{x_i} m \int c_i f d\vec{c} = \partial_{x_i}\rho v_i. \tag{3.16}$$

Integral $\textcircled{3}$ is expressed as

$$\textcircled{3} = \int q_\alpha (E_i + \varepsilon_{ijk} c_j B_k)\partial_{c_i} f d\vec{c}. \tag{3.17}$$

The product rule is used again to expand the Lorentz force term

$$\partial_{c_i}[(E_i + \varepsilon_{ijk} c_j B_k) f] = (E_i + \varepsilon_{ijk} c_j B_k)\partial_{c_i} f + f\partial_{c_i}(E_i + \varepsilon_{ijk} c_j B_k), \tag{3.18}$$

which gives

$$\begin{aligned} \textcircled{3} &= \int q_\alpha\{\partial_{c_i}[(E_i + \varepsilon_{ijk} c_j B_k) f] - f\partial_{c_i}(E_i + \varepsilon_{ijk} c_j B_k)\} d\vec{c} \qquad (3.19) \\ &= \int q_\alpha\{\partial_{c_i}[(E_i + \varepsilon_{ijk} c_j B_k) f] - f\partial_{c_i}(\cancelto{0}{E_i} + \cancelto{0}{\varepsilon_{ijk} c_j B_k})\} d\vec{c}. \end{aligned}$$

Since $\vec{E}$ has no dependence on c_i and the divergence of the cross product term is zero, those terms vanish. The integral reduces to

$$\textcircled{3} \;=\; \int q_\alpha \partial_{c_i}(E_i + \varepsilon_{ijk} c_j B_k) f d\vec{c}. \tag{3.20}$$

Applying the Divergence Theorem converts the volume integral to a surface integral, $\int_\forall \partial_{c_i}\lambda dv = \oint_S \lambda ds$. The consequence of this is that by integrating over a surface with infinite volume (all velocity space), the probability of a particle with infinite velocity is zero (i.e. as $c_i \to \infty$, $f \to 0$). $\therefore$

$$\textcircled{3} \;=\; \oint_S q_\alpha (E_i + \varepsilon_{ijk} c_j B_k)\cancelto{0}{f} n_i ds = 0, \tag{3.21}$$

where n_i is the unit normal of the surface s in phase space.

The assumption is made that particles do not ionize, fuse, etc. As a consequence the collisions of particles to not change the number of particles.

$$\textcircled{4} \quad = \quad \int m\partial_t f_\alpha \Big|_c d\vec{c} = 0. \tag{3.22}$$

Recombining these terms (①, ②, etc.) yields the familiar expression for continuity,

$$\partial_t \rho + \partial_{x_i} \rho v_i = 0. \tag{3.23}$$

This expression contains a time derivative term $\partial_t \rho$ and a flux term $\partial_{x_i} \rho v_i$. The zeroth moment nomenclature refers to the order of the time derivative variable ρ while the flux term requires the next highest moment momentum. The pattern of flux terms requiring information from the next higher moment is a general property of moment approximations.

3.2.2 First Moment

The first moment of the Boltzmann equation is used to describe the evolution of the momentum which is expressed as

$$\underbrace{\int c_i m \partial_t f d\vec{c}}_{\textcircled{1}} + \underbrace{\int c_j m c_i \partial_{x_j} f d\vec{c}}_{\textcircled{2}} + \underbrace{\int c_j q_\alpha (E_i + \varepsilon_{ijk} c_j B_k) \partial_{c_j} f d\vec{c}}_{\textcircled{3}} = \underbrace{\int c_i m \partial_t f \Big|_c d\vec{c}}_{\textcircled{4}}. \tag{3.24}$$

The product rule is used to expand the integrand of the first integral:

$$\begin{aligned} \textcircled{1} \quad &= \int m[\partial_t (c_i f_\alpha) - f_\alpha \partial_t c_i] d\vec{c} \\ &= \int m[\partial_t (c_i f_\alpha) - f_\alpha \cancelto{0}{\partial_t c_i}] d\vec{c}. \end{aligned} \tag{3.25}$$

In phase space, velocity is independent of t so that term vanishes leaving

$$\textcircled{1} \quad = \quad \partial_t m \int c_i f dc = \partial_t \rho v_i. \tag{3.26}$$

Integral $\textcircled{2}$ is written as

$$\begin{aligned} \textcircled{2} \quad &= \quad \int m c_i c_j \partial_{x_j} f dc = \int m c_i [\partial_{x_j}(c_j f) - \cancelto{0}{f \partial_{x_j} c_i}] d\vec{c} \\ &= \quad \int m c_j \partial_{x_j}(c_i f) dc = m \int [\partial_{x_j}(c_i c_j f) - \cancelto{0}{c_i f \partial_{x_j} c_j}] d\vec{c}, \end{aligned} \tag{3.27}$$

where velocity is independent of x, causing those terms to vanish. To express integral $\textcircled{2}$ in terms of the fluid variables defined in Sec. 3.1, the substitution is made such that $w_i = c_i - v_i \Rightarrow c_i = w_i + v_i$.

$$\begin{aligned} \textcircled{2} \quad &= \quad \partial_{x_j} m \int (w_i + v_i)(w_j + v_j) f d\vec{c} \\ &= \quad \partial_{x_j} m \int (w_i w_j + 2 v_i w_j + v_i v_j) f d\vec{c} \\ &= \quad \partial_{x_j} m [\int w_i w_j f_\alpha d\vec{c} + 2 v_i \int w_j f d\vec{c} + \int v_i v_j f d\vec{c}]. \end{aligned} \tag{3.28}$$

A quick proof shows that the central moment of the random velocity vanishes.

$$\begin{aligned} \int w_j f d\vec{c} \quad &= \quad \int (c_i - v_i) f d\vec{c} = \int c_i f d\vec{c} - v_i \int f d\vec{c} \\ &= \quad v_i - v_i = 0. \end{aligned} \tag{3.29}$$

Which further reduces the second integral to

$$\begin{aligned} \textcircled{2} \quad &= \quad \partial_{x_j} m [\int w_i w_j f d\vec{c} + 2 v_i \cancelto{0}{\int w_j f d\vec{c}} + v_i v_j \int f d\vec{c}] \\ &= \quad \partial_{x_j} [P_{ij} + \rho v_i v_j]. \end{aligned} \tag{3.30}$$

As before, $\vec{E}$ has no dependence on c and the divergence of the cross product term is zero, reducing the third integral as

$$\begin{aligned}
③ &= \int c_j q_\alpha (E_i + \varepsilon_{ijk} c_j B_k) \partial_{c_i} f d\vec{c} \qquad (3.31)\\
&= \int c_j q_\alpha \{\partial_{c_i}[(E_i + \varepsilon_{ijk} c_j B_k) f] - f \partial_{c_j} (\cancelto{0}{E_i} + \cancelto{0}{\varepsilon_{ijk} c_j B_k})\} d\vec{c}\\
&= \int c_j q_\alpha \partial_{c_j}[(E_i + \varepsilon_{ijk} c_j B_k) f] d\vec{c}\\
&= q_\alpha \int \{\partial_{c_j}[c_j (E_i + \varepsilon_{ijk} c_j B_k) f] - (E_i + \varepsilon_{ijk} c_j B_k) f \partial_{c_j} c_j\} d\vec{c}.
\end{aligned}$$

Since $\partial_{c_j} c_j = I$, the integral reduces to

$$③ = q_\alpha \int \{\partial_{c_j}[c_j (E_i + \varepsilon_{ijk} c_j B_k) f] - (E_i + \varepsilon_{ijk} c_j B_k) f\} d\vec{c}. \qquad (3.32)$$

Applying the divergence theorem causes the surface integral to vanish as before,

$$\begin{aligned}
③ &= \cancelto{0}{\oint q_\alpha c_j (E_i + \varepsilon_{ijk} c_j B_k) f d\vec{s}} - \int q_\alpha c_j (E_i + \varepsilon_{ijk} c_j B_k) f d\vec{c} \qquad (3.33)\\
&= -q_\alpha E_i \int f d\vec{c} + q_\alpha \varepsilon_{ikj} B_k v_j\\
&= -\frac{q_\alpha}{m} (\rho E_i + \varepsilon_{ijk} v_j B_k).
\end{aligned}$$

The Boltzmann equation allows for the possibility of collisions to change momentum, energy, etc. and drive the solution to an equilibrium state. The simulations eventually assume these to be negligible, but they are included for future expansion of the model with collision operators such as Braginskii [11]. Here, the collisions are split into collisions between like species $\alpha\alpha$ and between separate species $\beta\alpha$.

$$\begin{aligned}
④ &= \int c_i m \partial_t f \Big|_c d\vec{c} \qquad (3.34)\\
&= m \int c_i c_{\alpha\alpha} d\vec{c} + m \int [c_i \sum_{\beta \neq \alpha} c_{\beta\alpha}] d\vec{c}.
\end{aligned}$$

In a fluid description of a plasma, collisions of like particles result in no net change in species momentum.

$$④ = m \int \cancelto{0}{c_i c_{\alpha\alpha}} d\vec{c} + m \int [c_i \sum_{\beta \neq \alpha} c_{\beta\alpha}] d\vec{c}. \tag{3.35}$$

Assuming no sources or sinks indicates that the bulk velocity v_i is not affected:

$$④ = m \sum_{\beta \neq \alpha} \int (w_i + \cancelto{0}{v_i}) c_{\beta\alpha} d\vec{c}. \tag{3.36}$$

Here, we define the frictional collision operator $R_{\alpha\beta i} = m \int w_i c_{\alpha\beta} dc$ for the exchange of momentum between species.

$$④ = \sum_{\beta \neq \alpha} R_{\alpha\beta i}. \tag{3.37}$$

Combining these terms yields the momentum equation,

$$\partial_t \rho v_i + \partial_{x_j}[P_{ij} + \rho v_i v_j] - \frac{q_\alpha}{m}(\rho E_i + \varepsilon_{ijk} v_j B_k) = \sum_{\beta \neq \alpha} R_{\alpha\beta i}. \tag{3.38}$$

3.2.3 Second Moment

The first moment of the Boltzmann equation is used to describe the evolution of the momentum which is expressed as

$$\underbrace{\int c_i c_j m \partial_t f d\vec{c}}_{①} + \underbrace{\int c_i c_j c_k m \partial_{x_i} f d\vec{c}}_{②} + \underbrace{\int c_i c_j q_\alpha (E_k + \varepsilon_{kij} c_i B_j) \partial_{c_k} f d\vec{c}}_{③} = \underbrace{\int c_i c_j m \partial_t f \Big|_c d\vec{c}}_{④}. \tag{3.39}$$

As before, in phase space, velocity is independent of t. The first integral is written as

$$\begin{aligned}
\textcircled{1} &= \int c_i[c_j m\partial_t f]d\vec{c} \\
&= m\int c_i[\partial_t(c_j f_\alpha) - f\cancelto{0}{\partial_t c_j}]d\vec{c} \\
&= m\int [\partial_t(c_i c_j f) - (c_j f)\cancelto{0}{\partial_t c_i}]d\vec{c} \\
&= \partial_t m\int c_i c_j f d\vec{c},
\end{aligned} \tag{3.40}$$

and the integral $\int w_j f dc$ vanishes giving

$$\begin{aligned}
\textcircled{1} &= \partial_t m\int (w_i w_j + \cancelto{0}{2v_i w_j} + v_i v_j) f d\vec{c} \\
&= \partial_t(P_{ij} + \rho v_i v_j).
\end{aligned} \tag{3.41}$$

And, c_i has no dependence on x, which is used to reduce the second integral.

$$\begin{aligned}
\textcircled{2} &= \int c_i c_j c_k m\partial_{x_k} f d\vec{c} = m\int c_i c_j[\partial_{x_k}(c_k f) - f\cancelto{0}{\partial_{x_k} c_k}]d\vec{c} \\
&= \partial_{x_k} m\int c_i c_j c_k f d\vec{c} \\
&= \partial_{x_k} m\int (v_i + w_i)(v_j + w_j)(v_k + w_k) f d\vec{c} \\
&= \partial_{x_k} m\left[v_i v_j v_k \int f d\vec{c} + 3v_{(i}\int w_j w_{k)} f d\vec{c} + \int w_i w_j w_k f d\vec{c}\right] \\
&= \partial_{x_k}(\rho v_i v_j v_k + 3v_{(i}P_{jk)} + h_{ijk}).
\end{aligned} \tag{3.42}$$

The Lorentz force term,

$$\textcircled{3} = \int c_i c_j q_\alpha(E_k + \varepsilon_{kij} c_i B_j)\partial_{c_k} f d\vec{c}, \tag{3.43}$$

is expanded using the product rule such that

$$\partial_{c_k}\left[c_i c_j q_\alpha (E_k + \varepsilon_{kij} c_i B_j) f\right] = c_i c_j q_\alpha (E_k + \varepsilon_{kij} c_i B_j) + f q_\alpha \partial_{c_k}\left[c_i c_j (E_k + \varepsilon_{kij} c_i B_j)\right]. \tag{3.44}$$

The third integral is expanded to

$$③ \quad = \quad \int \left\{\partial_{c_k}\left[c_i c_j q_\alpha (E_k + \varepsilon_{kij} c_i B_j) f\right] - f q_\alpha \partial_{c_k}\left[c_i c_j (E_k + \varepsilon_{kij} c_i B_j)\right]\right\} d\vec{c}. \tag{3.45}$$

Using the Divergence Theorem $\int_{\forall} \partial_{c_i} \lambda dv = \oint_s \lambda ds$, where the probability of a particle with infinite velocity being zero reduces the third integral to

$$\begin{aligned} ③ \quad &= \quad \cancelto{0}{\oint_S c_i c_j q_\alpha (E_k + \varepsilon_{kij} c_i B_j) f d\vec{s}} - \int f q_\alpha \partial_{c_k}\left[c_i c_j (E_k + \varepsilon_{kij} c_i B_j)\right] d\vec{c} \qquad (3.46) \\ &= \quad - \int f q_\alpha \partial_{c_k}\left[c_i c_j E_k\right] d\vec{c} - \int f q_\alpha \partial_{c_k}\left[c_i c_j \varepsilon_{kij} c_i B_j\right] d\vec{c}. \end{aligned}$$

The product rule is used to expand the expressions:

$$\begin{aligned} \partial_{c_k}\left[c_i c_j E_k\right] &= c_i c_j \partial_{c_k} E_k + E_k \partial_{c_k} c_i c_j, \qquad (3.47) \\ \partial_{c_k}\left[c_i c_j \varepsilon_{kij} c_i B_j\right] &= c_i c_j \partial_{c_k}\left[\varepsilon_{kij} c_i B_j\right] + \left(\varepsilon_{kij} c_i B_j\right) \partial_{c_k}\left[c_i c_j\right]. \end{aligned}$$

After substituting, again we see that $\vec{E}$ has no dependence on $c \therefore c_i c_j \partial_{c_k} E_k \to 0$ and $\varepsilon_{kij} c_i B_j$ is orthogonal to c_k, therefore the divergence is zero ($\partial_{c_k}\left[\varepsilon_{kij} c_i B_j\right] \to 0$).

$$\begin{aligned}
\textcircled{3} &= -\int f q_\alpha E_k \partial_{c_k} c_i c_j d\vec{c} - \int f q_\alpha \left(\varepsilon_{kij} c_i B_j\right) \partial_{c_k}\left[c_i c_j\right] d\vec{c} \qquad (3.48)\\
&= -\int f q_\alpha E_k c_i \partial_{c_k} c_j d\vec{c} - \int f q_\alpha E_k c_j \partial_{c_k} c_i d\vec{c}\\
&\quad - \int f q_\alpha \left(\varepsilon_{kij} c_i B_j\right) c_i \partial_{c_k}\left[c_j\right] d\vec{c} - \int f q_\alpha \left(\varepsilon_{kij} c_i B_j\right) c_j \partial_{c_k}\left[c_i\right] d\vec{c}\\
&= -\int f q_\alpha E_k c_i \partial_{c_k} \delta_{kj} c_j d\vec{c} - \int f q_\alpha E_k c_j \partial_{c_k} \delta_{ki} c_i d\vec{c}\\
&\quad - \int f q_\alpha \left(\varepsilon_{kij} c_i B_j\right) c_i \partial_{c_k}\left[\delta_{kj} c_j\right] d\vec{c} - \int f q_\alpha \left(\varepsilon_{kij} c_i B_j\right) c_j \partial_{c_k}\left[\delta_{ki} c_i\right] d\vec{c}\\
&= -\int f q_\alpha E_j c_i \partial_{c_j} c_j d\vec{c} - \int f q_\alpha E_i c_j \partial_{c_i} c_i d\vec{c}\\
&\quad - \int f q_\alpha \left(\varepsilon_{jij} c_i B_j\right) c_i \partial_{c_j}\left[c_j\right] d\vec{c} - \int f q_\alpha \left(\varepsilon_{iij} c_i B_j\right) c_j \partial_{c_i}\left[c_i\right] d\vec{c}.
\end{aligned}$$

With, $\varepsilon_{jij} \to 0$, $\varepsilon_{iij} \to 0$, and $\partial_{c_j}\left[c_j\right] = \partial_{c_i}\left[c_i\right] = 1$ the expression for the third integral is simplified to

$$\begin{aligned}
\textcircled{3} &= -E_j \int f q_\alpha c_i d\vec{c} - E_i \int f q_\alpha c_j d\vec{c} \qquad (3.49)\\
&= -2\frac{q_\alpha}{m_\alpha} \rho E_{(i} v_{j)}.
\end{aligned}$$

Collision between like particles result in no change is species energy, leaving the fourth integral as

$$\begin{aligned}
\textcircled{4} &= \int c_i c_j m \partial_t f\Big|_c d\vec{c} = m \sum_{\beta\neq\alpha} \int c_i c_j c_{\alpha\beta} d\vec{c} \qquad (3.50)\\
&= m \sum_{\beta\neq\alpha} \int (w_i + v_i)(w_j + v_j) c_{\alpha\beta} d\vec{c}\\
&= m \sum_{\beta\neq\alpha} \left[\int w_i w_j c_{\alpha\beta} d\vec{c} + 2 v_{(i} \int w_{j)} c_{\alpha\beta} d\vec{c} + \cancelto{0}{\int v_i v_j c_{\alpha\beta} d\vec{c}}\right].
\end{aligned}$$

After assuming no sources or sinks, we define the viscous heating operator $Q_{\alpha\beta ij} = \frac{1}{2}m\int w_i w_j c_{\alpha\beta} d\vec{c}$ such that

$$\textcircled{4} = \sum_{\beta\neq\alpha}\left[2v_{(i}R_{j)\alpha\beta} + 2Q_{\alpha\beta ij}\right]. \tag{3.51}$$

Combining terms yields the evolution of the energy tensor,

$$\begin{array}{c} \partial_t(P_{ij} + \rho v_i v_j) + \partial_{x_k}(\rho v_i v_j v_k + 3v_{(i}P_{jk)} + h_{ijk}) - 2\frac{q_\alpha}{m_\alpha}\rho E_{(i}v_{j)} \\ = \sum_{\beta\neq\alpha}\left[2v_{(i}R_{j)\alpha\beta} + 2Q_{\alpha\beta ij}\right] \end{array}. \tag{3.52}$$

Here we see the first major difference between the standard two-fluid plasma model and the 13-moment two-fluid plasma model. The second moment of the Boltzmann equation by itself describes the evolution of the energy tensor of a fluid. However, in the Euler equations used in the standard two-fluid plasma model, the equations are simplified by taking the trace and this equation to yield the total energy equation instead.

3.2.4 Third Moment

The third moment of the Boltzmann equation describes the evolution of the energy flux tensor. It is this equation which allows the 13-moment model to simultaneously solve for the classical fluid equations and heat transfer.

$$\begin{array}{c} \underset{\textcircled{1}}{\int c_i c_j c_k m\partial_t f_\alpha d\vec{c}} + \underset{\textcircled{2}}{\int c_i c_j c_k c_m m\partial_{x_m} f_\alpha d\vec{c}} + \underset{\textcircled{3}}{\int c_i c_j c_k q_\alpha(E_m + \varepsilon_{mij}c_i B_j)\partial_{c_m} f_\alpha d\vec{c}} \\ = \underset{\textcircled{4}}{\int c_i c_j c_k m\partial_t f_\alpha\Big|_c d\vec{c}} \end{array}. \tag{3.53}$$

The substitution $c_i = w_i + v_i$ is used such that the first integral is written as

$$\begin{aligned}
① &= \int c_i c_j c_k m \partial_t f_\alpha d\vec{c} = \partial_t m \int c_i c_j c_k f_\alpha d\vec{c} \\
&= \partial_t m \int (w_i + v_i)(w_j + v_j)(w_k + v_k) f_\alpha d\vec{c} \\
&= \partial_t m \left[\int v_i v_j v_k f_\alpha d\vec{c} + 3 v_{(i} \int w_j w_{k)} f_\alpha d\vec{c} + \int w_i w_j w_k f_\alpha d\vec{c} \right] \\
&= \partial_t \left[\rho v_i v_j v_k + 3 v_{(i} P_{jk)} + h_{ijk} \right].
\end{aligned} \tag{3.54}$$

The flux term ② is similarly expanded as

$$\begin{aligned}
② &= m \int c_i c_j c_k c_m \partial_{x_m} f_\alpha d\vec{c} = \partial_{x_m} m \int c_i c_j c_k c_m f_\alpha d\vec{c} \\
&= \partial_{x_m} m \int (w_i + v_i)(w_j + v_j)(w_k + v_k)(w_m + v_m) f_\alpha d\vec{c}.
\end{aligned} \tag{3.55}$$

With the simplification that the integral $\int w_j f_\alpha d\vec{c}$ is zero, the second integral reduces to

$$\begin{aligned}
② = \partial_{x_m} m \int \Big(& v_i v_j v_k v_m + 6 v_{(i} v_j w_k w_{m)} \\
& + 4 v_{(i} w_j w_k w_{m)} + w_i w_j w_k w_m \Big) f_\alpha d\vec{c}.
\end{aligned}$$

Substituting the definitions for the fluid variables in Sec. 3.1 further reduces integral ② to

$$② = \partial_{x_m} \left(\rho v_i v_j v_k v_m + 6 v_{(i} v_j P_{km)} + 4 v_{(i} h_{jkm)} + \Delta_{ijkm} \right). \tag{3.56}$$

The Lorentz force term integral ③ is

$$③ = \int c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j) \partial_{c_m} f_\alpha d\vec{c}. \tag{3.57}$$

Integral ③ is expanded using the product rule to make the following substitution:

$$\begin{aligned}\partial_{c_m}\left[c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j) f_\alpha\right] &= \left[c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j)\partial_{c_m} f_\alpha\right] \\ &\quad + f_\alpha \partial_{c_m}\left[c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j)\right].\end{aligned} \tag{3.58}$$

Using the Divergence Theorem $\int_\forall \partial_{c_i}\lambda dv = \oint_s \lambda ds$, where the probability of a particle with infinite velocity is zero further simplifies the equations by introducing a surface integral which is evaluated as c goes to infinity where $f = 0$.

$$\begin{aligned}\textcircled{3} &= \int \Bigg\{ \partial_{c_m}\left[c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j) f_\alpha\right] \\ &\qquad - f_\alpha \partial_{c_m}\left[c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j)\right]\Bigg\} d\vec{c} \\ &= \cancelto{0}{\oint_S c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j) f_\alpha d\vec{s}} - \int f_\alpha \partial_{c_m}\left[c_i c_j c_k q_\alpha (E_m + \varepsilon_{mij} c_i B_j)\right] d\vec{c} \\ &= -\int f_\alpha \partial_{c_m} c_i c_j c_k q_\alpha E_m d\vec{c} - \int f_\alpha \partial_{c_m} c_i c_j c_k q_\alpha \varepsilon_{mij} c_i B_j d\vec{c}.\end{aligned} \tag{3.59}$$

The product rule is used to further expand the integrand of each term in integral ③,

$$\begin{aligned}\partial_{c_m} c_i c_j c_k E_m &= \cancelto{0}{c_i c_j c_k \partial_{c_m} E_m} + E_m \partial_{c_m} c_i c_j c_k, \\ \partial_{c_m} c_i c_j c_k \varepsilon_{mij} c_i B_j &= \cancelto{0}{c_i c_j c_k \partial_{c_m}(\varepsilon_{mij} c_i B_j)} + \varepsilon_{mij} c_i B_j \partial_{c_m} c_i c_j c_k.\end{aligned} \tag{3.60}$$

Since $\vec{E}$ has no dependence on c, $\partial_{c_m} E_m \to 0$ and $\varepsilon_{mij} c_i B_j$ is orthogonal to c_m, the divergence is zero ($\partial_{c_m}(\varepsilon_{mij} c_i B_j) \to 0$). Expanding the remaining terms with the product rule,

$$\partial_{c_m} c_i c_j c_k = c_i c_j \partial_{c_m} c_k + c_k c_i \partial_{c_m} c_j + c_j c_k \partial_{c_m} c_i, \tag{3.61}$$

and substituting Eq. 3.60 and Eq. 3.61 yields the expression for integral ③,

$$\begin{aligned}
\textcircled{3} &= -\int f_\alpha q_\alpha E_m \left(c_i c_j \partial_{c_m} c_k + c_k c_i \partial_{c_m} c_j + c_j c_k \partial_{c_m} c_i\right) d\vec{c} \\
&\quad - \int f_\alpha q_\alpha \varepsilon_{mij} c_i B_j \left(c_i c_j \partial_{c_m} c_k + c_k c_i \partial_{c_m} c_j + c_j c_k \partial_{c_m} c_i\right) d\vec{c} \\
&= -\int f_\alpha q_\alpha E_m \left(c_i c_j \partial_{c_m} \delta_{mk} c_k + c_k c_i \partial_{c_m} \delta_{mj} c_j + c_j c_k \partial_{c_m} \delta_{mi} c_i\right) d\vec{c} \\
&\quad - \int f_\alpha q_\alpha \varepsilon_{mij} c_i B_j \left(c_i c_j \partial_{c_m} \delta_{mk} c_k + c_k c_i \partial_{c_m} \delta_{mj} c_j + c_j c_k \partial_{c_m} \delta_{mi} c_i\right) d\vec{c} \\
&= -3 \int f_\alpha q_\alpha E_{(k} c_i c_{j)} \partial_{c_k} c_k d\vec{c} \\
&\quad - \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j \partial_{c_k} c_k\right) d\vec{c} - \int f_\alpha q_\alpha \varepsilon_{(iij)} c_i B_j \left(c_j c_k \partial_{c_i} c_i\right) d\vec{c}.
\end{aligned} \tag{3.62}$$

With, $\epsilon_{jij} \to 0$, $\epsilon_{iij} \to 0$, and $\partial_{c_j}[c_j] = \partial_{c_i}[c_i] = \partial_{c_k}[c_k] = 1$ integral ③ becomes

$$\begin{aligned}
\textcircled{3} &= -3 \int f_\alpha q_\alpha E_{(k} c_i c_{j)} d\vec{c} - \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j\right) d\vec{c} \\
&= -3 q_\alpha E_{(i} \int \left(w_i w_j + \cancelto{0}{2 v_j w_k} + v_j v_k\right) d\vec{c} - \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j\right) d\vec{c} \\
&= -3 \frac{q_\alpha}{m_\alpha} E_{(i} P_{jk)} - 3 \frac{q_\alpha}{m_\alpha} E_{(i} v_j v_{k)} - \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j\right) d\vec{c}.
\end{aligned} \tag{3.63}$$

Substitutions are made to integral ④ for the friction and viscous heating operators to yield the integral

$$\begin{aligned}
\textcircled{4} &= \int c_i c_j c_k m \partial_t f_\alpha \Big|_c d\vec{c} = m \sum_{\beta \neq \alpha} \int c_i c_j c_k c_{\alpha\beta} d\vec{c} \\
&= m \sum_{\beta \neq \alpha} \int (w_i + v_i)(w_j + v_j)(w_k + v_k) c_{\alpha\beta} d\vec{c} \\
&= m \sum_{\beta \neq \alpha} \int \left(v_i v_j v_k + 3 v_i v_j w_k + 3 v_i w_j w_k + w_i w_j w_k\right) c_{\alpha\beta} d\vec{c}.
\end{aligned} \tag{3.64}$$

As before, integral $\int v_i v_j v_k c_{\alpha\beta} dc$ vanishes, simplifying integral ④

$$④ = \sum_{\beta \neq \alpha} \left[3v_{(i} v_j R_{k)\alpha\beta} + \frac{3}{2} v_{(i} Q_{jk)\alpha\beta} + m \int w_i w_j w_k c_{\alpha\beta} d\vec{c} \right]. \tag{3.65}$$

Combining terms yields the equation for the evolution of the energy flux tensor,

$$\begin{aligned} \partial_t \left[\rho v_i v_j v_k + 3v_{(i} P_{jk)} + h_{ijk} \right] + \partial_{x_m} \left(\rho v_i v_j v_k v_m + 6v_{(i} v_j P_{km)} + 4v_{(i} h_{jkm)} + \Delta_{ijkm} \right) \\ -3 \frac{q_\alpha}{m_\alpha} E_{(i} P_{jk)} - 3 \frac{q_\alpha}{m_\alpha} E_{(i} v_j v_{k)} - \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j \right) d\vec{c} \\ = \sum_{\beta \neq \alpha} \left[3v_{(i} v_j R_{k)\alpha\beta} + \frac{3}{2} v_{(i} Q_{jk)\alpha\beta} + m \int w_i w_j w_k c_{\alpha\beta} d\vec{c} \right] \end{aligned} . \tag{3.66}$$

3.3 Closure Scheme

At this point it should be apparent that to evolve any one moment equation in time requires a flux term for the next higher moment. For example, the evolution of the continuity equation requires the flux of the momentum equation. In reality this process goes on forever, which makes it difficult to solve these equations in any practical way. It is for this reason that closure schemes are needed which express the highest term in the system using information from the other moments. Standard approximations from fluids manipulate the energy transfer by assuming the system is isentropic, adiabatic, etc. The 13-moment equations, however, resolve these thermal effects directly.

To close the third moment equation, the trace is taken to reduce the rank three tensors to something more manageable. Taking the time derivative term and setting $j = k$ yields the expression,

$$\mathrm{Tr}_{j=k} \left(\rho v_i v_j v_k + 3v_{(i} P_{jk)} + h_{ijk} \right) = \rho v_i v_k v_k + v_i P_{kk} + 2v_k P_{ki} + h_{ikk}. \tag{3.67}$$

Substituting for the scalar pressure and the quasi-heat flux vector

$$3p = P_{kk} \tag{3.68}$$

$$q_i = \frac{1}{2}m\iiint w^2 w_i f^{(P3)} d\vec{c} = \frac{1}{2}h_{ikk} \tag{3.69}$$

yields,

$$\mathrm{Tr}_{j=k}\left(\rho v_i v_j v_k + 3v_{(i}P_{jk)} + h_{ijk}\right) = \rho v_i v_k^2 + 3v_i p + 2v_k P_{ki} + 2q_i. \tag{3.70}$$

The trace of the divergence term becomes

$$\begin{aligned} &\mathrm{Tr}_{j=k}\left(\rho v_i v_j v_k v_m + 6v_{i(j}P_{km)} + 4v_{(i}h_{jkm)} + \Delta_{ijkm}\right) \qquad (3.71)\\ =\ & \rho v_i v_m v_k^2 + 3p v_m v_i + v_k^2 P_{im} + 4v_{(i}P_{m)k}v_k + 4v_{(i}q_{k)} + 2v_k h_{ikm} + \Delta_{ikkm}. \end{aligned}$$

The trace of the Lorentz force term becomes

$$\begin{aligned} &\mathrm{Tr}_{j=k}\left(-3\frac{q_\alpha}{m_\alpha}E_{(i}P_{jk)} - 3\frac{q_\alpha}{m_\alpha}E_{(i}v_j v_{k)} - \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j\right) dc\right) \qquad (3.72)\\ =\ & \left(-3\frac{q_\alpha}{m_\alpha}E_{(i}P_{kk)} - 3\frac{q_\alpha}{m_\alpha}E_{(i}v_k v_{k)} - \int f_\alpha q_\alpha \varepsilon_{kik} c_i B_k \left(c_i c_k\right) dc\right)\\ =\ & \left(-3\frac{q_\alpha}{m_\alpha}E_{(i}P_{kk)} - 3\frac{q_\alpha}{m_\alpha}E_{(i}v_k v_{k)}\right)\\ =\ & -\frac{q_\alpha}{m_\alpha}E_i P_{kk} - 2\frac{q_\alpha}{m_\alpha}E_k P_{ik} - \frac{q_\alpha}{m_\alpha}E_i v_k^2 - 2\frac{q_\alpha}{m_\alpha}E_k v_i v_k \end{aligned}$$

and the collision terms become

$$\begin{aligned} &\mathrm{Tr}_{j=k}\left(\sum_{\beta\neq\alpha}\left[3v_{(i}v_j R_{k)\alpha\beta} + \frac{3}{2}v_{(i}Q_{jk)\alpha\beta} + m\int w_i w_j w_k c_{\alpha\beta} d\vec{c}\right]\right) \qquad (3.73)\\ =\ & \sum_{\beta\neq\alpha}\left[3v_{(i}v_k R_{k)\alpha\beta} + \frac{3}{2}v_{(i}Q_{kk)\alpha\beta} + m\int w_i w_k^2 c_{\alpha\beta} d\vec{c}\right]. \end{aligned}$$

Collecting terms,

$$\begin{aligned} &\partial_t \left[\rho v_i v_k^2 + 3 v_i p + 2 v_k P_{ki} + 2 q_i\right] \\ &+\partial_{x_j}\left(\rho v_i v_j v_k^2 + 3 p v_j v_i + v_k^2 P_{ij} + 4 v_{(i} P_{j)k} v_k + 4 v_{(i} q_{j)} + 2 v_k h_{ijk} + \Delta_{ijkk}\right) \\ &-\frac{q_\alpha}{m_\alpha} E_i P_{jj} - 2\frac{q_\alpha}{m_\alpha} E_j P_{ij} - \frac{q_\alpha}{m_\alpha} E_i v_j^2 - 2\frac{q_\alpha}{m_\alpha} E_j v_i v_j \\ &= \sum_{\beta\neq\alpha}\left[3 v_{(i} v_j R_{j)\alpha\beta} + \frac{3}{2} v_{(i} Q_{jj)\alpha\beta} + m\int w_i w_j^2 c_{\alpha\beta} d\vec{c}\right] . \end{aligned} \tag{3.74}$$

This leaves two higher order moments which need to be resolved,

$$h_{ijk} = m\int w_i w_j w_k f^{(P3)} d\vec{c} \tag{3.75}$$

$$\Delta_{ijkk} = m\int w_i w_j w_k^2 f^{(P3)} d\vec{c} \tag{3.76}$$

and a collisional heat flux operator is defined as

$$m\int w_i w_k^2 c_{\alpha\beta} d\vec{c} = H_i. \tag{3.77}$$

To resolve the third moment of the distribution function, the reduced heat flux is used. It can be seen that the heat flux is related to the third moment by Eq. 3.69. Using the definition presented in Eq. 2.30, the heat flux is

$$\begin{aligned} q_i &= \frac{\rho}{2} Q \left(\frac{1}{2}\left(3\theta - N_k^2\right)\delta_{ik} + \Theta_{ik}\right) N_k \\ &= \frac{\rho}{2} Q \Theta_{ij}^{1/2}\left(\frac{1}{2}\left(3\theta - n_l \Theta_{lm} n_m\right)\delta_{jk} + \Theta_{jk}\right) n_k. \end{aligned} \tag{3.78}$$

Torrilhon [2] proposes to use the information from the heat flux vector to solve for $\vec{n}$ and Q. Eq. 3.78 is non-linear, and therefore an iterative solution is required. The suggested algorithm for this iterative solution is outlined in (Algorithm 1). This iterative solution is not yet implemented but is presented here for completeness.

Once Q and n_i are known, substituting into Eq. 2.30 gives the third moment

$$h_{ijk} = \frac{\rho}{2} Q \left(-N_i N_j N_k + 3N_{(i}\Theta_{jk)} \right) . \tag{3.88}$$

The trace of the fourth moment is equal to

$$\begin{aligned} \Delta_{ijkk} &= \rho M_{ijkk} \qquad (3.89)\\ &= \rho \left(D - \frac{3}{4} Q^2 \right) \Theta_{(ij}\Theta_{kk)} + \rho \frac{3}{4} Q^2 \left(-N_i N_j N_k N_k + 2N_{(i}N_j\Theta_{kk)} \right) \\ &= \rho \left(D - \frac{3}{4} Q^2 \right) \left(\Theta_{ij}\theta + \frac{2}{3}\Theta_{ij}^2 \right) + \rho \frac{Q^2}{4} \left(3\left(\theta - N^2\right) N_i N_j + N^2 \Theta_{ij} + 4N_k \Theta_{k(i}N_{j)} \right) . \end{aligned}$$

With D being defined by the singular and realizable closure schemes.

3.4 3-D Complete System of Equations

The complete set of equations describing the 13-moment model are compiled here for clarity with the substitution from the ideal gas law that $P_{ij} = \rho\Theta_{ij}$:

$$\partial_t \rho + \partial_{x_i} \rho v_i = 0, \tag{3.90}$$

$$\partial_t \rho v_i + \partial_{x_j} [\rho\Theta_{ij} + \rho v_i v_j] - \frac{q_\alpha}{m} (\rho E_i + \varepsilon_{ijk} v_j B_k) = \sum_{\beta \neq \alpha} R_{i\alpha\beta}, \tag{3.91}$$

$$\begin{aligned} &\partial_t (\rho\Theta_{ij} + \rho v_i v_j) + \partial_{x_k} (\rho v_i v_j v_k + 3\rho v_{(i}\Theta_{jk)} + h_{ijk}) \qquad (3.92)\\ &\qquad -2\frac{q_\alpha}{m_\alpha} \rho E_{(i}v_{j)} = \sum_{\beta \neq \alpha} \left[2v_{(i}R_{j)\alpha\beta} + 2Q_{ij\alpha\beta} \right] \quad , \end{aligned}$$

$$\begin{aligned}
&\partial_t \left[\rho v_i v_k^2 + \frac{\rho\theta}{3} v_i + 2\rho v_k \Theta_{ki} + 2q_i\right] \qquad (3.93)\\
&+\partial_{x_j}\left(\rho v_i v_j v_k^2 + \frac{\rho\theta}{3} v_j v_i + \rho v_k^2 \Theta_{ij} + 2\rho v_{(i}\Theta_{j)k} v_k + 4v_{(i}q_{j)} + 2v_k h_{ijk} + \Delta_{ijkk}\right)\\
&\qquad -\frac{q_\alpha}{m_\alpha}\rho E_i \Theta_{jj} - 2\rho\frac{q_\alpha}{m_\alpha}E_j\Theta_{ij} - \frac{q_\alpha}{m_\alpha}E_i v_j^2 - 2\frac{q_\alpha}{m_\alpha}E_j v_i v_j\\
&\qquad = \sum_{\beta\neq\alpha}\left[3v_{(i}v_j R_{j)\alpha\beta} + \frac{3}{2}v_{(i}Q_{jj)\alpha\beta} + H_i\right] \quad ,
\end{aligned}$$

with

$$\begin{aligned}
h_{ijk} &= \frac{\rho}{2}Q\left(-N_iN_jN_k + 3N_{(i}\Theta_{jk)}\right), \qquad (3.94)\\
\Delta_{ijkk} &= \rho\left(D - \frac{3}{4}Q^2\right)\left(\Theta_{ij}\theta + \frac{2}{3}\Theta_{ij}^2\right) \qquad (3.95)\\
&\quad + \rho\frac{Q^2}{4}\left(3\left(\theta - N^2\right)N_iN_j + N^2\Theta_{ij} + 4N_k\Theta_{k(i}N_{j)}\right),
\end{aligned}$$

and the non-dimensional quantity Q determined iteratively by Alg. 1. The non-dimensional ratio D is based on the realizable or singular closure scheme,

$$D = \begin{cases} 3\left(1 + \frac{Q^2\left(22+Q^2\right)}{32-Q^2}\right) & (D_{realizable}) \\ 3\left(1+\frac{1}{2}Q^2\right) & (D_{singular}) \end{cases} . \qquad (3.96)$$

The collision operators are defined as the collision operator $R_{i\alpha\beta} = m\int w_i c_{\alpha\beta} d\vec{c}$, the viscous heating operator $Q_{ij\alpha\beta} = \frac{1}{2}m\int w_i w_j c_{\alpha\beta} d\vec{c}$, and the heat flux collision operator $H_i = m\int w_i w_k^2 c_{\alpha\beta} d\vec{c}$. It can easily be seen that setting the collision operators and the electric and magnetic fields to zero recovers the single-fluid equations in [2].

Algorithm 1

To solve Eq. 3.78 for the non-dimensional ratio Q and the direction of skewness n_k, a substitution is made such that

$$q_i = \frac{\rho}{2} Q B_{ik}\left(x\right) n_k, \tag{3.79}$$

where

$$B_{ik}(x) = \Theta_{ij}^{1/2}\left(\frac{3\theta - x}{2}\delta_{jk} + \Theta_{jk}\right). \tag{3.80}$$

The initial guess is made by assuming

$$x = n_l^0 \Theta_{lm} n_m^0 = 3\Theta_{ll} = \theta, \tag{3.81}$$

where the superscripts represent the number of iterations each variable has undergone. Substituting in and solving for the intermediate value $r_k^1 = Q^0 n_k^0$ gives

$$r_k^1 = \frac{2}{\rho} B_{ik}^{-1}\left(x\right) q_i. \tag{3.82}$$

Q is the magnitude of the vector r_k^1,

$$Q^1 = \left\| r_k^1 \right\|. \tag{3.83}$$

Calculating unit vector n_k requires normalizing r_k by its magnitude,

$$n_k^1 = \frac{r_k^1}{Q^1}. \tag{3.84}$$

Successive approximations (s) are given by the recursion relation

$$r_i^{(s)} = \frac{2}{\rho} B_{ik}^{-1}\left(n_l^{(s-1)} \Theta_{lm} n_m^{(s-1)}\right) q_i, \tag{3.85}$$

$$Q^{(s)} = \left\| r_k^{(s)} \right\|, \tag{3.86}$$

$$n_k^{(s)} = \frac{r_k^{(s)}}{Q^{(s)}}. \tag{3.87}$$

Torrilhon [2] makes the assertion that $s = 1$ gives a reasonable approximation with a deviation of less than 2% from the exact solution.

Chapter 4
SOLUTIONS TO HYPERBOLIC EQUATIONS IN 3D

Solving hyperbolic equations is necessary to analyze fluid systems. The Riemann problem is modeled with a hyperbolic equation system given by the equation

$$\partial_t U + \partial_{x_i} F(U) = 0, \tag{4.1}$$

where U is a vector of the conserved variables and $F(U)$ are the flux quantities, with the initial value problem a piecewise function

$$U(x, t = 0) = \begin{cases} U_R, & x > 0 \\ U_L, & x < 0 \end{cases}, \tag{4.2}$$

where U_R and U_L represent the right and left side of the domain. One way to solve hyperbolic equations numerically involves the use of an approximate Riemann solver based on the work of Roe [12]. In Roe's approximate Riemann solution, the equation is expressed as

$$\partial_t U + \hat{A}_c \partial_{x_i} U = S, \tag{4.3}$$

where $\hat{A}_c$ is a locally constant linearization of the Jacobian $A_c = \frac{\partial F}{\partial U}$. Additional details on the properties of Roe solvers and application to the 10-moment model can be found in Brown [13]. In short, however, the Roe solver approximates the difference in fluxes at the left and right interface as

$$F_R - F_L = \sum_{k=1}^{n} \hat{\alpha}_k \hat{\lambda}_k \hat{r}_k, \tag{4.4}$$

where $\hat{\lambda}_k$ are the eigenvalues, $\hat{r}_k$ are the corresponding right eigenvectors, and $\hat{\alpha}_k = \hat{L}_p \Delta V$ the wave strengths of the approximate coefficient matrix $\hat{A}_c$, with $\hat{L}_p$ the left primitive eigenvectors. A 13-moment Roe solver in 3-dimensions is not implemented here, but the calculation of the eigenvalues, eigenvectors, and Jacobians necessary are included to start such an effort.

4.1 Conservative vs. Primitive Equations

The equations thus far derived for the 13-moment model and presented in Eqs. (3.90-3.93) are classified as conservative equations. These represent the evolution in time and space of conserved quantities U (momentum, energy, etc.) due to the corresponding fluxes $F(U)$.

$$U = \begin{pmatrix} \rho \\ \rho v_i \\ \rho\Theta_{ij} + \rho v_i v_j \\ \rho v_i v_k^2 + \frac{\rho\theta}{3} v_i + 2\rho v_k \Theta_{ki} + 2q_i \end{pmatrix} \tag{4.5}$$

and

$$F = \begin{pmatrix} \rho v_i \\ \rho\Theta_{ij} + \rho v_i v_j \\ \rho v_i v_j v_k + 3\rho v_{(i}\Theta_{jk)} + h_{ijk} \\ \rho v_i v_j v_k^2 + \frac{\rho\theta}{3} v_j v_i + \rho v_k^2 \Theta_{ij} + 2\rho v_{(i}\Theta_{j)k} v_k + 4v_{(i}q_{j)} + 2v_k h_{ijk} + \Delta_{ijkk} \end{pmatrix}, \tag{4.6}$$

along with the source terms

$$S = \begin{pmatrix} 0 \\ \frac{q_\alpha}{m}(\rho E_i + \varepsilon_{ijk} v_j B_k) + \sum_{\beta\neq\alpha} R_{i\alpha\beta} \\ 2\frac{q_\alpha}{m_\alpha}\rho E_{(i}v_{j)} + \sum_{\beta\neq\alpha}\left[2v_{(i}R_{j)\alpha\beta} + 2Q_{ij\alpha\beta}\right] \\ \frac{q_\alpha}{m_\alpha}\rho E_i \Theta_{jj} + 2\rho\frac{q_\alpha}{m_\alpha} E_j \Theta_{ij} + \frac{q_\alpha}{m_\alpha} E_i v_j^2 + 2\frac{q_\alpha}{m_\alpha} E_j v_i v_j + \cdots \\ \sum_{\beta\neq\alpha}\left[3v_{(i}v_j R_{j)\alpha\beta} + \frac{3}{2}v_{(i}Q_{jj)\alpha\beta} + H_i\right] \end{pmatrix}. \tag{4.7}$$

These equations can also be expressed in terms of the more traditional variables (pressure, velocity, etc.) called primitive form as

$$\partial_t V + A_p \partial_{x_i} V = S \tag{4.8}$$

where, for the 13-moment equations,

$$V = \left(\rho, v_x, v_y, v_z, \Theta_{xx}, \Theta_{xy}, \Theta_{yy}, \Theta_{yz}, \Theta_{zz}, \Theta_{zx}, q_x, q_y, q_z\right)^T . \tag{4.9}$$

The advantage of converting to primitive equations is that for more complex, higher moment systems it is difficult to compute the eigensystem of the conservative equations analytically, while it is relatively straightforward for primitive equations. In the 13-moment system, the equations are not truly hyperbolic since they include a source term. Later, it is discussed how the equations are advanced by solving the system as if it were hyperbolic and then adding the contributions of the source terms separately.

4.2 Converting to Primitive Equations

Converting to primitive equations is a relatively straightforward, but tedious process. Mathematica's computer algebra software is used to help keep track of this process and more details of the code used can be found in Appendix B. The product rule is used to expand the conservative equations and then substitutions are made to reduce the amount of redundant information in the expressions. An example of redundant information might be that the evolution of density is contained in each of the higher moment equations, but since the zeroth moment contains this information there is no added benefit to leaving it in the higher moments.

4.2.1 Conservation of Mass Moment

The first equation in the 13-moment system in the conservative formulation is given by Eq. 3.90 as

$$\partial_t \rho + \partial_{x_i} (v_i \rho) = 0. \tag{4.10}$$

It is easy to see that applying the product rule gives the primitive equation

$$\partial_t \rho + \rho \partial_{x_i} (v_i) + v_i \partial_{x_i} \rho = 0. \tag{4.11}$$

4.2.2 Conservation of Momentum Moment

The second equation for the 13-moment system in conservative form is given by Eq. 3.91 as

$$\partial_t (\rho \mathrm{v}_i) + \partial_{x_j} (\rho \Theta_{ij} + \rho \mathrm{v}_i v_j) = \lambda_i, \tag{4.12}$$

where the source terms are collected into the term

$$\lambda_i = \frac{q_\alpha}{m} (\rho E_i + \varepsilon_{ijk} v_j B_k) + \sum_{\beta \neq \alpha} R_{i\alpha\beta}. \tag{4.13}$$

The product rule is used to derive the expression for the unreduced second primitive equation

$$v_i \partial_t \rho + \rho \partial_t (v_i) + v_i v_j \partial_{x_j} \rho + \Theta_{ij} \partial_{x_j} \rho + \rho v_j \partial_{x_j} (v_i) + \rho v_i \partial_{x_j} (v_j) + \rho \partial_{x_j} (\Theta_{ij}) = \lambda_i. \tag{4.14}$$

The first primitive equation is substituted to simplify the system to,

$$\partial_t (v_i) + \frac{\Theta_{ij}}{\rho} \partial_{x_j} \rho + v_j \partial_{x_j} (v_i) + \partial_{x_j} (\Theta_{ij}) = \frac{\lambda_i}{\rho}. \tag{4.15}$$

4.2.3 Conservation of Energy Moment

The third equation in a conservative formulation is given by Eq. 3.92 and modified to

$$\partial_t \left(\rho\Theta_{ij} + \rho v_i v_j\right) + \partial_{x_k} \left(\rho v_i v_j v_k + 3\rho v_{(i}\Theta_{jk)}\right) = \psi_{ij} - \partial_{x_k} h_{ijk}, \tag{4.16}$$

with the source term ψ_{ij} defined as

$$\psi_{ij} = 2\frac{q_\alpha}{m_\alpha}\rho E_{(i}v_{j)} + \sum_{\beta\neq\alpha}\left[2v_{(i}R_{j)\alpha\beta} + 2Q_{ij\alpha\beta}\right]. \tag{4.17}$$

Note here that the divergence of the heat-flux tensor $\partial_{x_k} h_{ijk}$ has been moved to the right hand side as if it were a source term. Solving for the eigenvectors with the divergence term proved to be prohibitively difficult. This term and a few others are therefore treated as source terms and are updated accordingly. The consequence of this is that large shocks in these quantities are not captured by this model.

The product rule is used to expand the equations. The resulting equation is simplified by substituting the expressions for the first and second primitive equations to give the third primitive equation

$$\partial_t \left(\Theta_{ij}\right) + \Theta_{jk}\partial_{x_k}\left(v_i\right) + \Theta_{ik}\partial_{x_k}\left(v_j\right) + v_k\partial_{x_k}\left(\Theta_{ij}\right) = \frac{\psi_{ij} - v_j\lambda_i - v_i\lambda_j - \partial_{x_k}\left(h_{ijk}\right)}{\rho}. \tag{4.18}$$

4.2.4 Conservation of Energy Flux Moment

The fourth conservative equation before the trace is taken is given by Eq. 3.93 as

$$\begin{aligned}&\partial_t \left(\rho v_i v_j v_k + 3\rho v_{(i}\Theta_{jk)} + h_{ijk}\right) + \partial_{x_m}\left(\rho v_i v_j v_k v_m + 6\rho v_{(i}v_j\Theta_{km)} + 4v_{(i}h_{jkm)}\right) \\ &-h_{kmi}\partial_{x_m}\left(v_j\right) - h_{mij}\partial_{x_m}\left(v_k\right) = \gamma_{ijk} - \partial_{x_m}\left(\Delta_{ijkm}\right) - h_{kmi}\partial_{x_m}\left(v_j\right) - h_{mij}\partial_{x_m}\left(v_k\right)\quad ,\end{aligned} \tag{4.19}$$

with the source terms represented by γ_{ijk}

$$\begin{aligned}\gamma_{ijk} &= 3\frac{q_\alpha}{m_\alpha}E_{(i}P_{jk)} + 3\frac{q_\alpha}{m_\alpha}E_{(i}v_jv_{k)} + \int f_\alpha q_\alpha \varepsilon_{kij} c_i B_j \left(c_i c_j\right) d\vec{c} \\ &+ \sum_{\beta\neq\alpha}\left[3v_{(i}v_jR_{k)\alpha\beta} + \frac{3}{2}v_{(i}Q_{jk)\alpha\beta} + m\int w_iw_jw_kc_{\alpha\beta}d\vec{c}\right].\end{aligned} \tag{4.20}$$

These complicated expressions simplify considerably when the trace of the fourth moment is taken, but are included here for completeness. It again should be noted that terms are moved to the right hand side of the equation to be computed with the source terms. Also, the terms $h_{kmi}\partial_{x_m}(v_j)$ and $h_{mij}\partial_{x_m}(v_k)$ are subtracted from both sides and vanish from the left hand side when the trace is taken. The temperature tensor is introduced and the product rule is used as before. Similarly the first, second and third primitive equations are substituted to remove redundant information which gives the complete fourth primitive equation. The expression is reduced to its usable form by taking its trace $j \to k$ and then changing the free indices $m \to j$ to give the fourth primitive equation used in the rest of the derivation

$$\begin{aligned}&\partial_t(q_i) - \Theta_{ki}\Theta_{kj}\partial_{x_j}\rho - \frac{1}{2}\Theta_{ij}\Theta_{kk}\partial_{x_j}\rho + v_j\partial_{x_j}(q_i) + q_j\partial_{x_j}(v_i) \\ &+q_i\partial_{x_j}(v_j) - \frac{1}{2}\rho\Theta_{kk}\partial_{x_j}(\Theta_{ij}) - \rho\Theta_{ki}\partial_{x_j}(\Theta_{kj}) = \frac{\gamma_{ikk}}{2} + \frac{1}{2}v_k^2\lambda_i + v_iv_k\lambda_k \\ &-\lambda_k\Theta_{ki} - \frac{1}{2}\lambda_i\Theta_{kk} - v_k\psi_{ik} - \frac{1}{2}v_i\psi_{kk} - \frac{1}{2}\partial_{x_j}(\Delta_{ijkk}) - h_{ijk}\partial_{x_j}(v_k).\end{aligned} \tag{4.21}$$

4.3 Creation of A_p, B_p and C_p Matrices

The primitive hyperbolic equation is separated into its components in the x, y and z directions to give the equation

$$\partial_tV^{13} + A_p^{13}\partial_xV^{13} + B_p^{13}\partial_yV^{13} + C_p^{13}\partial_xV^{13} = S^{13}, \tag{4.22}$$

where A_p, B_p and C_p represent the Jacobians for the x, y and z-directions respectively and the superscript thirteen represents the length of each vector. The four primitive equations from the previous section are now expanded from their compact indicial form to give the complete primitive equations for the 13-moment model.

4.3.1 Primitive Form of the Density Equation

The first primitive equation in indicial notation is given by Eq. 4.11 as

$$\partial_t \rho + \rho \partial_{x_i} (v_i) + v_i \partial_{x_i} \rho = 0, \tag{4.23}$$

which is expanded by summing over i to give

$$\partial_t \rho + v_x \partial_x \rho + \rho \partial_x (v_x) + v_y \partial_y \rho + \rho \partial_y (v_y) + v_z \partial_z \rho + \rho \partial_z (v_z) \quad = \quad 0. \tag{4.24}$$

Expressions in the remaining sections are left in their expanded forms to make the derivation of the matrix coefficients more clear.

4.3.2 Primitive Form of the Momentum Equation

The second primitive equation in indicial notation is given by Eq. 4.15 as

$$\partial_t (v_i) + \frac{\Theta_{ij}}{\rho} \partial_{x_j} \rho + v_j \partial_{x_j} (v_i) + \partial_{x_j} (\Theta_{ij}) = \frac{\lambda_i}{\rho}, \tag{4.25}$$

which is expanded by first summing over j to give

$$\begin{aligned} \partial_t (v_i) + \frac{\Theta_{ix} \partial_x \rho}{\rho} + v_x \partial_x (v_i) + \partial_x (\Theta_{ix}) + \frac{\Theta_{iy} \partial_y \rho}{\rho} + v_y \partial_y (v_i) + \partial_y (\Theta_{iy}) \\ + \frac{\Theta_{iz} \partial_z \rho}{\rho} + v_z \partial_z (v_i) + \partial_z (\Theta_{iz}) = \frac{\lambda_i}{\rho}. \end{aligned} \tag{4.26}$$

Expressing Eq. 4.26 for each value of i gives the primitive equations for each velocity component.

$$\begin{aligned}\partial_t\left(v_x\right)+\frac{\Theta_{xx}\partial_x\rho}{\rho}+v_x\partial_x\left(v_x\right)+\partial_x\left(\Theta_{xx}\right)+\frac{\Theta_{xy}\partial_y\rho}{\rho}+v_y\partial_y\left(v_x\right)+\partial_y\left(\Theta_{xy}\right)\\+\frac{\Theta_{zx}\partial_z\rho}{\rho}+v_z\partial_z\left(v_x\right)+\partial_z\left(\Theta_{zx}\right)=\frac{\lambda_x}{\rho},\end{aligned}\tag{4.27}$$

$$\begin{aligned}\partial_t\left(v_y\right)+\frac{\Theta_{xy}\partial_x\rho}{\rho}+v_x\partial_x\left(v_y\right)+\partial_x\left(\Theta_{xy}\right)+\frac{\Theta_{yy}\partial_y\rho}{\rho}+v_y\partial_y\left(v_y\right)+\partial_y\left(\Theta_{yy}\right)\\+\frac{\Theta_{yz}\partial_z\rho}{\rho}+v_z\partial_z\left(v_y\right)+\partial_z\left(\Theta_{yz}\right)=\frac{\lambda_y}{\rho},\end{aligned}\tag{4.28}$$

$$\begin{aligned}\partial_t\left(v_z\right)+\frac{\Theta_{zx}\partial_x\rho}{\rho}+v_x\partial_x\left(v_z\right)+\partial_x\left(\Theta_{zx}\right)+\frac{\Theta_{yz}\partial_y\rho}{\rho}+v_y\partial_y\left(v_z\right)+\partial_y\left(\Theta_{yz}\right)\\+\frac{\Theta_{zz}\partial_z\rho}{\rho}+v_z\partial_z\left(v_z\right)+\partial_z\left(\Theta_{zz}\right)=\frac{\lambda_z}{\rho}.\end{aligned}\tag{4.29}$$

4.3.3 Primitive Form of The Energy Equations

The third primitive equation in indicial notation is given by Eq. 4.18 as

$$\partial_t\left(\Theta_{ij}\right)+\Theta_{jk}\partial_{x_k}\left(v_i\right)+\Theta_{ik}\partial_{x_k}\left(v_j\right)+v_k\partial_{x_k}\left(\Theta_{ij}\right)=\frac{\psi_{ij}-v_j\lambda_i-v_i\lambda_j-\partial_{x_k}\left(h_{ijk}\right)}{\rho}\tag{4.30}$$

and is first expanded by summing over k. Summing over the remaining indices and seeing that the temperature tensor Θ_{ij} is symmetric ($\Theta_{ij}=\Theta_{ji}$) gives

$$\begin{aligned}\partial_t\left(\Theta_{xx}\right)+2\Theta_{xx}\partial_x\left(v_x\right)+v_x\partial_x\left(\Theta_{xx}\right)+2\Theta_{xy}\partial_y\left(v_x\right)+v_y\partial_y\left(\Theta_{xx}\right)\\+2\Theta_{zx}\partial_z\left(v_x\right)+v_z\partial_z\left(\Theta_{xx}\right)=-\frac{2v_x\lambda_x}{\rho}+\frac{\psi_{xx}}{\rho}-\frac{\partial_x\left(h_{xxx}\right)}{\rho}-\frac{\partial_y\left(h_{xxy}\right)}{\rho}-\frac{\partial_z\left(h_{xxz}\right)}{\rho},\end{aligned}\tag{4.31}$$

$$\begin{aligned}\partial_t\left(\Theta_{xy}\right)+\Theta_{xy}\partial_x\left(v_x\right)+\Theta_{xx}\partial_x\left(v_y\right)+v_x\partial_x\left(\Theta_{xy}\right)+\Theta_{yy}\partial_y\left(v_x\right)+\Theta_{xy}\partial_y\left(v_y\right)\\ +v_y\partial_y\left(\Theta_{xy}\right)+\Theta_{yz}\partial_z\left(v_x\right)+\Theta_{zx}\partial_z\left(v_y\right)+v_z\partial_z\left(\Theta_{xy}\right)\\ =-\frac{v_y\lambda_x}{\rho}-\frac{v_x\lambda_y}{\rho}+\frac{\psi_{xy}}{\rho}-\frac{\partial_x\left(h_{xyx}\right)}{\rho}-\frac{\partial_y\left(h_{xyy}\right)}{\rho}-\frac{\partial_z\left(h_{xyz}\right)}{\rho},\end{aligned} \tag{4.32}$$

$$\begin{aligned}\partial_t\left(\Theta_{yy}\right)+2\Theta_{xy}\partial_x\left(v_y\right)+v_x\partial_x\left(\Theta_{yy}\right)+2\Theta_{yy}\partial_y\left(v_y\right)+v_y\partial_y\left(\Theta_{yy}\right)\\ +2\Theta_{yz}\partial_z\left(v_y\right)+v_z\partial_z\left(\Theta_{yy}\right)=-\frac{2v_y\lambda_y}{\rho}+\frac{\psi_{yy}}{\rho}-\frac{\partial_x\left(h_{yyx}\right)}{\rho}-\frac{\partial_y\left(h_{yyy}\right)}{\rho}-\frac{\partial_z\left(h_{yyz}\right)}{\rho},\end{aligned} \tag{4.33}$$

$$\begin{aligned}\partial_t\left(\Theta_{yz}\right)+\Theta_{zx}\partial_x\left(v_y\right)+\Theta_{xy}\partial_x\left(v_z\right)+v_x\partial_x\left(\Theta_{yz}\right)+\Theta_{yz}\partial_y\left(v_y\right)+\Theta_{yy}\partial_y\left(v_z\right)\\ +v_y\partial_y\left(\Theta_{yz}\right)+\Theta_{zz}\partial_z\left(v_y\right)+\Theta_{yz}\partial_z\left(v_z\right)+v_z\partial_z\left(\Theta_{yz}\right)\\ =-\frac{v_z\lambda_y}{\rho}-\frac{v_y\lambda_z}{\rho}+\frac{\psi_{yz}}{\rho}-\frac{\partial_x\left(h_{yzx}\right)}{\rho}-\frac{\partial_y\left(h_{yzy}\right)}{\rho}-\frac{\partial_z\left(h_{yzz}\right)}{\rho},\end{aligned} \tag{4.34}$$

$$\begin{aligned}\partial_t\left(\Theta_{zz}\right)+2\Theta_{zx}\partial_x\left(v_z\right)+v_x\partial_x\left(\Theta_{zz}\right)+2\Theta_{yz}\partial_y\left(v_z\right)+v_y\partial_y\left(\Theta_{zz}\right)\\ +2\Theta_{zz}\partial_z\left(v_z\right)+v_z\partial_z\left(\Theta_{zz}\right)=-\frac{2v_z\lambda_z}{\rho}+\frac{\psi_{zz}}{\rho}-\frac{\partial_x\left(h_{zzx}\right)}{\rho}-\frac{\partial_y\left(h_{zzy}\right)}{\rho}-\frac{\partial_z\left(h_{zzz}\right)}{\rho},\end{aligned} \tag{4.35}$$

$$\begin{aligned}\partial_t\left(\Theta_{zx}\right)+\Theta_{zx}\partial_x\left(v_x\right)+\Theta_{xx}\partial_x\left(v_z\right)+v_x\partial_x\left(\Theta_{zx}\right)+\Theta_{yz}\partial_y\left(v_x\right)+\Theta_{xy}\partial_y\left(v_z\right)\\ +v_y\partial_y\left(\Theta_{zx}\right)+\Theta_{zz}\partial_z\left(v_x\right)+\Theta_{zx}\partial_z\left(v_z\right)+v_z\partial_z\left(\Theta_{zx}\right)\\ =-\frac{v_z\lambda_x}{\rho}-\frac{v_x\lambda_z}{\rho}+\frac{\psi_{zx}}{\rho}-\frac{\partial_x\left(h_{zxx}\right)}{\rho}-\frac{\partial_y\left(h_{zxy}\right)}{\rho}-\frac{\partial_z\left(h_{zxz}\right)}{\rho}.\end{aligned} \tag{4.36}$$

4.3.4 Primitive Form of the Energy Flux Equations

The fourth primitive equation in indicial notation is given by Eq. 4.21 as

$$\begin{aligned}\partial_t\left(q_i\right) - \Theta_{ki}\Theta_{kj}\partial_{x_j}\rho - \frac{1}{2}\Theta_{ij}\Theta_{kk}\partial_{x_j}\rho + v_j\partial_{x_j}\left(q_i\right) + q_j\partial_{x_j}\left(v_i\right) + q_i\partial_{x_j}\left(v_j\right) \qquad (4.37)\\ -\frac{1}{2}\rho\Theta_{kk}\partial_{x_j}\left(\Theta_{ij}\right) - \rho\Theta_{ki}\partial_{x_j}\left(\Theta_{kj}\right) = \frac{\gamma_{ikk}}{2} + \frac{1}{2}v_k^2\lambda_i + v_iv_k\lambda_k - \lambda_k\Theta_{ki} - \frac{1}{2}\lambda_i\Theta_{kk} - v_k\psi_{ik}\\ -\frac{1}{2}v_i\psi_{kk} - \frac{1}{2}\partial_{x_j}\left(\Delta_{ijkk}\right) - h_{ijk}\partial_{x_j}\left(v_k\right)\end{aligned}$$

and is expanded by first summing over k and then over j. The final three primitive equations are computed by summing over i, with the first of the three being

$$\begin{aligned}\partial_t\left(q_x\right) + \left(-\frac{3}{2}\Theta_{xx}^2 - \Theta_{xy}^2 - \frac{1}{2}\Theta_{xx}\Theta_{yy} - \Theta_{zx}^2 - \frac{1}{2}\Theta_{xx}\Theta_{zz}\right)\partial_x\rho \qquad (4.38)\\ +\left(-\frac{3}{2}\rho\Theta_{xx} - \frac{1}{2}\rho\Theta_{yy} - \frac{1}{2}\rho\Theta_{zz}\right)\partial_x\left(\Theta_{xx}\right) - \rho\Theta_{xy}\partial_x\left(\Theta_{xy}\right) - \rho\Theta_{zx}\partial_x\left(\Theta_{zx}\right)\\ +\left(-\frac{3}{2}\Theta_{xx}\Theta_{xy} - \frac{3}{2}\Theta_{xy}\Theta_{yy} - \Theta_{yz}\Theta_{zx} - \frac{1}{2}\Theta_{xy}\Theta_{zz}\right)\partial_y\rho + v_y\partial_y\left(q_x\right) + q_y\partial_y\left(v_x\right)\\ +q_x\partial_y\left(v_y\right) + \left(-\frac{3}{2}\rho\Theta_{xx} - \frac{1}{2}\rho\Theta_{yy} - \frac{1}{2}\rho\Theta_{zz}\right)\partial_y\left(\Theta_{xy}\right) - \rho\Theta_{xy}\partial_y\left(\Theta_{yy}\right)\\ -\rho\Theta_{zx}\partial_y\left(\Theta_{yz}\right) + \left(-\Theta_{xy}\Theta_{yz} - \frac{3}{2}\Theta_{xx}\Theta_{zx} - \frac{1}{2}\Theta_{yy}\Theta_{zx} - \frac{3}{2}\Theta_{zx}\Theta_{zz}\right)\partial_z\rho\\ +v_x\partial_x\left(q_x\right) + v_z\partial_z\left(q_x\right) + q_z\partial_z\left(v_x\right) + q_x\partial_z\left(v_z\right) - \rho\Theta_{xy}\partial_z\left(\Theta_{yz}\right)\\ +2q_x\partial_x\left(v_x\right) + \left(-\frac{3}{2}\rho\Theta_{xx} - \frac{1}{2}\rho\Theta_{yy} - \frac{1}{2}\rho\Theta_{zz}\right)\partial_z\left(\Theta_{zx}\right) - \rho\Theta_{zx}\partial_z\left(\Theta_{zz}\right)\\ = \frac{1}{2}\left[\gamma_{xxx} + \gamma_{xyy} + \gamma_{xzz} + 3v_x^2\lambda_x + v_y^2\lambda_x - 2\lambda_k\Theta_{kx} + \lambda_x\left(v_z^2 - \Theta_{xx} - \Theta_{yy} - \Theta_{zz}\right)\right]\\ +\frac{1}{2}\left[-2v_y\psi_{xy} - 2v_z\psi_{xz} + v_x\left(2v_y\lambda_y + 2v_z\lambda_z - 3\psi_{xx} - \psi_{yy} - \psi_{zz}\right) - \partial_x\left(\Delta_{xxkk}\right)\right]\\ \frac{1}{2}\left[-2h_{xxx}\partial_x\left(v_x\right) - 2h_{xxy}\partial_x\left(v_y\right) - 2h_{xxz}\partial_x\left(v_z\right) - \partial_y\left(\Delta_{xykk}\right) - 2h_{xyx}\partial_y\left(v_x\right) - 2h_{xyy}\partial_y\left(v_y\right)\right]\\ \frac{1}{2}\left[-2h_{xyz}\partial_y\left(v_z\right) - \partial_z\left(\Delta_{xzkk}\right) - 2h_{xzx}\partial_z\left(v_x\right) - 2h_{xzy}\partial_z\left(v_y\right) - 2h_{xzz}\partial_z\left(v_z\right)\right].\end{aligned}$$

The other equations are equally as complex, and it does little good to reproduce them here. The final equations in matrix form are somewhat simpler and the final results can be seen there.

4.3.5 Flux Jacobian Matrices

The final A_p matrix is presented here with a few substitutions.

$$A_p = \begin{pmatrix}
v_x & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{xx}}{\rho} & v_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{xy}}{\rho} & 0 & v_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{zx}}{\rho} & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 2\Theta_{xx} & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \Theta_{xy} & \Theta_{xx} & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2\Theta_{xy} & 0 & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Theta_{zx} & \Theta_{xy} & 0 & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2\Theta_{zx} & 0 & 0 & 0 & 0 & v_x & 0 & 0 & 0 & 0 \\
0 & \Theta_{zx} & 0 & \Theta_{xx} & 0 & 0 & 0 & 0 & 0 & v_x & 0 & 0 & 0 \\
\mu_1 & 2q_x & 0 & 0 & -\frac{1}{2}\rho\left(\theta + 2\Theta_{xx}\right) & -\rho\Theta_{xy} & 0 & 0 & 0 & -\rho\Theta_{zx} & v_x & 0 & 0 \\
\mu_2 & q_y & q_x & 0 & -\rho\Theta_{xy} & -\frac{1}{2}\rho\left(\theta + 2\Theta_{yy}\right) & 0 & 0 & 0 & -\rho\Theta_{yz} & 0 & v_x & 0 \\
\mu_3 & q_z & 0 & q_x & -\rho\Theta_{zx} & -\rho\Theta_{yz} & 0 & 0 & 0 & -\frac{1}{2}\rho\left(\theta + 2\Theta_{zz}\right) & 0 & 0 & v_x
\end{pmatrix}, \tag{4.39}$$

where

$$\mu_1 = \frac{1}{2}\left(-2\left(\Theta_{xy}^2 + \Theta_{zx}^2\right) - \Theta_{xx}\left(3\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right)\right), \tag{4.40}$$

$$\mu_2 = \frac{1}{2}\left(-2\Theta_{yz}\Theta_{zx} - \Theta_{xy}\left(3\left(\Theta_{xx} + \Theta_{yy}\right) + \Theta_{zz}\right)\right), \tag{4.41}$$

$$\mu_3 = \frac{1}{2}\left(-2\Theta_{xy}\Theta_{yz} - \Theta_{zx}\left(3\Theta_{xx} + \Theta_{yy} + 3\Theta_{zz}\right)\right). \tag{4.42}$$

The remaining matrices can be found in Appendix B.

4.4 Eigensystem Analysis

The eigensystems are determined symbolically by use of Mathematica's Eigenvector[] and Eigenvalue[] functions, the code for which can be found in Appendix B.

4.4.1 Eigenvalues

The eigenvalues for the A_p, B_p and C_p matrices (λ_A, λ_B and λ_C respectively) are

$$\lambda_A = \begin{pmatrix} v_x \\ v_x \\ v_x \\ v_x \\ v_x \\ v_x \\ v_x \\ v_x - \sqrt{\Theta_{xx}} \\ v_x - \sqrt{\Theta_{xx}} \\ v_x + \sqrt{\Theta_{xx}} \\ v_x + \sqrt{\Theta_{xx}} \\ v_x - \sqrt{3}\sqrt{\Theta_{xx}} \\ v_x + \sqrt{3}\sqrt{\Theta_{xx}} \end{pmatrix}, \quad \lambda_B = \begin{pmatrix} v_y \\ v_y \\ v_y \\ v_y \\ v_y \\ v_y \\ v_y \\ v_y - \sqrt{\Theta_{yy}} \\ v_y - \sqrt{\Theta_{yy}} \\ v_y + \sqrt{\Theta_{yy}} \\ v_y + \sqrt{\Theta_{yy}} \\ v_y - \sqrt{3}\sqrt{\Theta_{yy}} \\ v_y + \sqrt{3}\sqrt{\Theta_{yy}} \end{pmatrix}, \quad \text{and } \lambda_C = \begin{pmatrix} v_z \\ v_z \\ v_z \\ v_z \\ v_z \\ v_z \\ v_z \\ v_z - \sqrt{\Theta_{zz}} \\ v_z - \sqrt{\Theta_{zz}} \\ v_z + \sqrt{\Theta_{zz}} \\ v_z + \sqrt{\Theta_{zz}} \\ v_z - \sqrt{3}\sqrt{\Theta_{zz}} \\ v_z + \sqrt{3}\sqrt{\Theta_{zz}} \end{pmatrix}. \tag{4.43}$$

4.4.2 A_p Eigenvectors

The right eigenvectors for the A_p matrix are presented here, with the remaining found in Appendix B. Many terms share certain similarities and are simplified by a matrix coefficient function defined as

$$\bar{Q}_i\left(c_1, c_2, c_3, c_4\right) = 2q_i + c_1 * \rho\sqrt{\Theta_{\hat{e}\hat{e}}}\left(c_2\Theta_{xx} + c_3\Theta_{yy} + c_4\Theta_{zz}\right). \tag{4.44}$$

For example,

$$\bar{Q}_x[1,1,3,1] = 2q_x + \rho\sqrt{\Theta_{xx}}\left(\Theta_{xx} + 3\Theta_{yy} + \Theta_{zz}\right). \tag{4.45}$$

The right eigenvectors for the A_p matrix are presented here with the following simplifications

$$\begin{aligned} \alpha &= \bar{Q}_x[1,1,1,3], & \beta &= \bar{Q}_x[1,1,3,1], \\ \chi &= \bar{Q}_x[-1,1,3,1], & \sigma &= \bar{Q}_x[-1,1,1,3]. \end{aligned} \tag{4.46}$$

$$r_{A1} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\1 \end{pmatrix}, \quad r_{A2} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\0 \end{pmatrix}, \quad r_{A3} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0 \end{pmatrix}, \quad r_{A4} = \begin{pmatrix} -\rho\\0\\0\\0\\\Theta_{xx}\\\Theta_{xy}\\0\\0\\0\\\Theta_{zx}\\0\\0\\0 \end{pmatrix}, \quad r_{A5} = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0\\0\\0 \end{pmatrix}, \tag{4.47}$$

$$r_{A6} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{A7} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{A8} = \begin{pmatrix} 0 \\ 0 \\ -2\alpha\sqrt{\Theta_{xx}} \\ 4\rho\Theta_{xx}\Theta_{yz} \\ 0 \\ 2\alpha\Theta_{xx} \\ 4\alpha\Theta_{xy} \\ 2\left(-2\rho\sqrt{\Theta_{xx}}\Theta_{xy}\Theta_{yz} + \alpha\Theta_{zx}\right) \\ -8\rho\sqrt{\Theta_{xx}}\Theta_{yz}\Theta_{zx} \\ -4\rho\Theta_{xx}^{3/2}\Theta_{yz} \\ 2\left(\alpha\rho\sqrt{\Theta_{xx}}\Theta_{xy} - 2\rho^2\Theta_{xx}\Theta_{yz}\Theta_{zx}\right) \\ \beta^2 - 4\rho^2\Theta_{xx}\Theta_{yz}^2 + 2\beta\rho\sqrt{\Theta_{xx}}\left(-\Theta_{yy} + \Theta_{zz}\right) \\ 0 \end{pmatrix}, \tag{4.48}$$

$$r_{A9} = \begin{pmatrix} 0 \\ 0 \\ -2\alpha\sqrt{\Theta_{xx}} \\ 4\rho\Theta_{xx}\Theta_{yz} \\ 0 \\ 2\alpha\Theta_{xx} \\ 4\alpha\Theta_{xy} \\ 2\left(-2\rho\sqrt{\Theta_{xx}}\Theta_{xy}\Theta_{yz} + \alpha\Theta_{zx}\right) \\ -8\rho\sqrt{\Theta_{xx}}\Theta_{yz}\Theta_{zx} \\ -4\rho\Theta_{xx}^{3/2}\Theta_{yz} \\ 2\left(\alpha\rho\sqrt{\Theta_{xx}}\Theta_{xy} - 2\rho^2\Theta_{xx}\Theta_{yz}\Theta_{zx}\right) \\ \beta^2 - 4\rho^2\Theta_{xx}\Theta_{yz}^2 + 2\beta\rho\sqrt{\Theta_{xx}}\left(-\Theta_{yy} + \Theta_{zz}\right) \\ 0 \end{pmatrix}, \tag{4.49}$$

$$r_{A10} = \begin{pmatrix} 0 \\ 0 \\ 4\rho\Theta_{xx}\Theta_{yz} \\ 2\chi\sqrt{\Theta_{xx}} \\ 0 \\ 4\rho\Theta_{xx}^{3/2}\Theta_{yz} \\ 8\rho\sqrt{\Theta_{xx}}\Theta_{xy}\Theta_{yz} \\ 2\chi\Theta_{xy} + 4\rho\sqrt{\Theta_{xx}}\Theta_{yz}\Theta_{zx} \\ 4\chi\Theta_{zx} \\ 2\chi\Theta_{xx} \\ -4\rho^2\Theta_{xx}\Theta_{xy}\Theta_{yz} - 2\rho\chi\sqrt{\Theta_{xx}}\Theta_{zx} \\ 0 \\ \chi^2 - 4\rho^2\Theta_{xx}\Theta_{yz}^2 + 2\rho\chi\sqrt{\Theta_{xx}}\left(\Theta_{yy} - \Theta_{zz}\right) \end{pmatrix}, \qquad (4.50)$$

$$r_{A11} = \begin{pmatrix} 0 \\ 0 \\ 2\sigma\sqrt{\Theta_{xx}} \\ 4\rho\Theta_{xx}\Theta_{yz} \\ 0 \\ 2\sigma\Theta_{xx} \\ 4\sigma\Theta_{xy} \\ 4\rho\sqrt{\Theta_{xx}}\Theta_{xy}\Theta_{yz} + 2\sigma\Theta_{zx} \\ 8\rho\sqrt{\Theta_{xx}}\Theta_{yz}\Theta_{zx} \\ 4\rho\Theta_{xx}^{3/2}\Theta_{yz} \\ -2\rho\sigma\sqrt{\Theta_{xx}}\Theta_{xy} - 4\rho^2\Theta_{xx}\Theta_{yz}\Theta_{zx} \\ \sigma^2 - 4\rho^2\Theta_{xx}\Theta_{yz}^2 + 2\rho\sigma\sqrt{\Theta_{xx}}\left(-\Theta_{yy} + \Theta_{zz}\right) \\ 0 \end{pmatrix}, \qquad (4.51)$$

$$r_{A12} = \begin{pmatrix} 2\sqrt{3}\rho\Theta_{xx} \\ -6\Theta_{xx}^{3/2} \\ -6\sqrt{\Theta_{xx}}\Theta_{xy} \\ -6\sqrt{\Theta_{xx}}\Theta_{zx} \\ 4\sqrt{3}\Theta_{xx}^2 \\ 4\sqrt{3}\Theta_{xx}\Theta_{xy} \\ 4\sqrt{3}\Theta_{xy}^2 \\ 4\sqrt{3}\Theta_{xy}\Theta_{zx} \\ 4\sqrt{3}\Theta_{zx}^2 \\ 4\sqrt{3}\Theta_{xx}\Theta_{zx} \\ 6\rho\sqrt{\Theta_{xx}}\left(\Theta_{xy}^2+\Theta_{zx}^2\right)+2\sqrt{3}\Theta_{xx}\bar{Q}_x\left[\frac{\sqrt{3}}{2},3,1,1\right] \\ 2\sqrt{3}q_y\Theta_{xx}+6\rho\sqrt{\Theta_{xx}}\Theta_{yz}\Theta_{zx}+\sqrt{3}\Theta_{xy}\bar{Q}_x\left[\sqrt{3},3,3,1\right] \\ 2\sqrt{3}q_z\Theta_{xx}+6\rho\sqrt{\Theta_{xx}}\Theta_{xy}\Theta_{yz}+\sqrt{3}\Theta_{zx}\bar{Q}_x\left[\sqrt{3},3,1,3\right] \end{pmatrix}, \tag{4.52}$$

and

$$r_{A13} = \begin{pmatrix} 2\sqrt{3}\rho\Theta_{xx} \\ 6\Theta_{xx}^{3/2} \\ 6\sqrt{\Theta_{xx}}\Theta_{xy} \\ 6\sqrt{\Theta_{xx}}\Theta_{zx} \\ 4\sqrt{3}\Theta_{xx}^2 \\ 4\sqrt{3}\Theta_{xx}\Theta_{xy} \\ 4\sqrt{3}\Theta_{xy}^2 \\ 4\sqrt{3}\Theta_{xy}\Theta_{zx} \\ 4\sqrt{3}\Theta_{zx}^2 \\ 4\sqrt{3}\Theta_{xx}\Theta_{zx} \\ -6\rho\sqrt{\Theta_{xx}}\left(\Theta_{xy}^2+\Theta_{zx}^2\right)+2\sqrt{3}\Theta_{xx}\bar{Q}_y\left[-\frac{\sqrt{3}}{2},3,1,1\right] \\ 2\sqrt{3}q_y\Theta_{xx}-6\rho\sqrt{\Theta_{xx}}\Theta_{yz}\Theta_{zx}+\sqrt{3}\Theta_{xy}\bar{Q}_y\left[-\sqrt{3},3,3,1\right] \\ 2\sqrt{3}q_z\Theta_{xx}-6\rho\sqrt{\Theta_{xx}}\Theta_{xy}\Theta_{yz}+\sqrt{3}\Theta_{z,x}\bar{Q}_y\left[-\sqrt{3},3,1,3\right] \end{pmatrix}. \tag{4.53}$$

4.5 Transformation Matrix

The transformation matrix is the Jacobian of the conserved variables with respect to the primitive variables.

$$M = \frac{\partial U}{\partial V}, \tag{4.54}$$

where V is defined as before, and U is expanded as

$$U = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho\Theta_{xx} + \rho v_x v_x \\ \rho\Theta_{xy} + \rho v_x v_y \\ \rho\Theta_{yy} + \rho v_y v_y \\ \rho\Theta_{yz} + \rho v_y v_z \\ \rho\Theta_{zz} + \rho v_z v_z \\ \rho\Theta_{zx} + \rho v_z v_x \\ \rho v_x \left(v_x^2 + v_y^2 + v_z^2\right) + \rho v_x \left(\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right) + 2\rho \left(v_x\Theta_{xx} + v_y\Theta_{xy} + v_z\Theta_{xz}\right) + 2q_x \\ \rho v_y \left(v_x^2 + v_y^2 + v_z^2\right) + \rho v_y \left(\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right) + 2\rho \left(v_x\Theta_{xy} + v_y\Theta_{yy} + v_z\Theta_{yz}\right) + 2q_y \\ \rho v_z \left(v_x^2 + v_y^2 + v_z^2\right) + \rho v_z \left(\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right) + 2\rho \left(v_x\Theta_{xz} + v_y\Theta_{yz} + v_z\Theta_{zz}\right) + 2q_z \end{pmatrix}. \tag{4.55}$$

M is calculated with the following substitutions,

$$v^2 = v_x^2 + v_y^2 + v_z^2, \tag{4.56}$$

$$\frac{\theta}{3} = \Theta_{xx} + \Theta_{yy} + \Theta_{zz}, \tag{4.57}$$

$$\zeta_i = v^2 v_i + \frac{\theta v_i}{3} + 2\left(v_x \Theta_{xi} + v_y \Theta_{iy} + v_z \Theta_{iz}\right), \tag{4.58}$$

$$\eta_i = v^2 \rho + \frac{\theta \rho}{3} + 2\rho v_i^2 + 2\rho \Theta_{ii}, \tag{4.59}$$

$$\mathcal{F}_1 = 2\rho v_x v_y + 2\rho \Theta_{xy}, \tag{4.60}$$

$$\mathcal{F}_2 = 2\rho v_x v_z + 2\rho \Theta_{xz}, \tag{4.61}$$

$$\mathcal{F}_3 = 2\rho v_y v_z + 2\rho \Theta_{yz}, \tag{4.62}$$

$$M = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_x & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_y & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_z & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_x^2 + \Theta_{xx} & 2\rho v_x & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_x v_y + \Theta_{xy} & \rho v_y & \rho v_x & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_y^2 + \Theta_{yy} & 0 & 2\rho v_y & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\
v_y v_z + \Theta_{yz} & 0 & \rho v_z & \rho v_y & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 \\
v_z^2 + \Theta_{zz} & 0 & 0 & 2\rho v_z & 0 & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 \\
v_x v_z + \Theta_{zx} & \rho v_z & 0 & \rho v_x & 0 & 0 & 0 & 0 & 0 & \rho & 0 & 0 & 0 \\
\zeta_x & \eta_x & \mathcal{F}_1 & \mathcal{F}_2 & 3\rho v_x & 2\rho v_y & \rho v_x & 0 & \rho v_x & 0 & 2 & 0 & 0 \\
\zeta_y & \mathcal{F}_1 & \eta_y & \mathcal{F}_3 & \rho v_y & 2\rho v_x & 3\rho v_y & 2\rho v_z & \rho v_y & 0 & 0 & 2 & 0 \\
\zeta_z & \mathcal{F}_2 & \mathcal{F}_3 & \eta_z & \rho v_z & 0 & \rho v_z & 2\rho v_y & 3\rho v_z & 0 & 0 & 0 & 2
\end{pmatrix}. \tag{4.63}$$

Chapter 5
1-D SIMULATION

In order to study the behavior of the system of equations, it is useful to reduce the system to the one-dimensional case. The formal derivation of the one dimensional system of equations is presented. Then, a second order accurate finite volume method is introduced. Details of the finite volume method such as the method for determining flux quantities are discussed. The behavior of the system is first studied without sources to compare to Torrilhon [2]. The equations are non-dimensionalized using the procedure outlined in [3] and the contributions of the source terms are added. An electrostatic solver is implemented to allow for a direct comparison to results from the 5-moment model.

5.1 *Reducing 13-Moment Equations to One Dimension*

To reduce the moments to the one dimensional case, only quantities that vary in one dimension (in this case x) are considered. This leaves five primitive variables: density, velocity in the x-direction, scalar pressure, pressure in the x-direction, and heat flux in the x-direction to capture full gas dynamics;

$$V = \{\rho, v_x, p, P_{xx}, q_x\}. \tag{5.1}$$

The variance in P_{yy} and P_{zz} are captured by the scalar pressure $p = P_{xx} + P_{yy} + P_{zz}$, where $P_{yy} = P_{zz}$. The moment equations are formally reduced to one dimension by only allowing the indices $i, j, k, etc.$ to be x and setting all other quantities to zero.

For example, for the continuity equation reduces as

$$\partial_t \rho + \partial_{x_i} (v_i \rho) = 0, \tag{5.2}$$

$$\partial_t \rho + \partial_x (v_x \rho) = 0. \tag{5.3}$$

Two further simplifications are made to the source terms by limiting the simulation to electrostatics, where the magnetic fields are set to zero, and by assuming that they are collisionless, where $R, Q, etc.$ are set to zero. As with the zeroth moment, the first - third moments are reduced to

$$\partial_t (\rho v_x) + \partial_x \left(P_{xx} + \rho v_x^2\right) = \frac{q_\alpha}{m}(\rho E_x), \tag{5.4}$$

$$\partial_t \left(P_{xx} + \rho v_x^2\right) + \partial_x \left(\rho v_x^3 + P_{xx} v_x + m_{xxx}\right) = 2\frac{q_\alpha}{m_\alpha} \rho E_x v_x, \tag{5.5}$$

$$\begin{aligned} \partial_t \left(\left(\frac{1}{2}\rho v_x^2 + \frac{3}{2}p\right) v_x + P_{xx} v_x + q_x \right) + & \\ \partial_x \left(\left(\frac{1}{2}\rho v_x^2 + \frac{3}{2}p\right) v_x^2 + \frac{5}{2} P_{xx} v_x^2 + 2 q_x v_x + m_{xxx} v_x + \frac{1}{2}\Delta_{xxkk} \right) & \\ = \frac{3}{2}\frac{q_\alpha}{m} \rho E_x p + \frac{q_\alpha}{m} \rho E_x P_{xx} + \frac{3}{2}\frac{q_\alpha}{m}\rho E_x v_x^2. & \end{aligned} \tag{5.6}$$

This process gives only four equations for five unknowns, so another expression is needed. This is done by taking the trace of the second moment equation $(i \to j)$

$$\partial_t (p_{jj} + \rho v_j v_j) + \partial_{x_k} (\rho v_j v_j v_k + 3 v_{(j} p_{jk)} + h_{jjk}) = 2\frac{q_\alpha}{m_\alpha} \rho E_{(j} v_{j)}. \tag{5.7}$$

Substitutions are then made for scalar pressure $3p = P_{kk}$ and the heat vector $h_{jkk} = 2q_j$, giving the final 1D equation

$$\partial_t (\frac{3}{2}p + \frac{1}{2}\rho v_x^2) + \partial_x (\left(\frac{1}{2}\rho v_x^2 + \frac{3}{2}p\right) v_x + p_{xx} v_x + q_x) = \frac{q_\alpha}{m_\alpha} \rho E_x v_x. \tag{5.8}$$

5.2 Finite Volume Methods

The 1D system of equations is simulated using a finite volume method [14, 15]. Finite volume (FV) methods are ways of solving differential equations on a discrete mesh, with each “node” surrounded by a cell volume. While there are many ways to implement a FV approach, the way pursued here was to use a predictor-corrector approach based on Heun’s method to achieve second order accuracy [18]. The fluxes are calculated using a First ORder CEntered (FORCE) flux method [14] with a Van Leer limiter to duplicate the results of Torrilhon [2] and then for the two-fluid simulations. The sources are updated using an unsplit method and are evolved in time with the flux terms.

5.2.1 Heun’s Method

Heun’s method is one method in the family of Runge-Kutta predictor-corrector schemes. It uses an Euler method to make an initial prediction about the solution to the PDE and then uses a trapezoidal method to correct this intermediate value for second order accuracy. The principles of this scheme are to take the piecewise constant (i.e. each cell has uniform properties) data and reconstruct it using a higher order function. That means that locally, the discrete values within a cell are represented by a linear function provided by interpolating with neighboring cells. That information is used to determine the values at the boundary of each cell which are used in the Riemann solver [14, 15].

The basic Heun’s method solves hyperbolic conservation laws of the form

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S\left(U\right), \tag{5.9}$$

where U are the conserved variables, $F(U)$ are the fluxes, and $S\left(U\right)$ are the source terms. An individual time step for this scheme would involve the following steps:

Step 1: Data Reconstruction. The conserved variables are converted algebraically to primitive variables. The values from the primitive variables are then represented in each local cell by a piece-wise linear function

$$V_i\left(x\right) = V_i^n + \frac{x - x_i}{\Delta x}\Delta_i, \;\; x \in \left[0, \Delta x\right] \tag{5.10}$$

where the subscript i denotes the value at the center of the volume; equal to the discrete cell value, the superscript n represents the current time and Δ_i is the difference between cell values

$$\Delta_i \equiv V_{i+1}^n - V_{i-1}^n. \tag{5.11}$$

The values at $x = 0$ and $x = \Delta x$ are the boundary extrapolated values as

$$V_i^L = V_i^n - \frac{1}{2}\Delta_i \; ; \;\; V_i^R = V_i^n + \frac{1}{2}\Delta_i. \tag{5.12}$$

The superscripts L and R denote the values at the left and right cell interfaces, respectively. This information is used to solve the Riemann problem and calculate the flux at the cell interfaces

$$F_{i+\frac{1}{2}} = F\left(V_{i+1}^L\right) - F\left(V_i^R\right). \tag{5.13}$$

Step 2: Predictor Step. Once the flux quantities are known, the solution is predicted using Euler's method and including the source term contributions.

$$\overline{U}_i = U_i + \frac{\Delta t}{\Delta x}\left[F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right] + \triangle t \cdot S\left(U_i\right) \tag{5.14}$$

Step 3: Data Reconstruction. The data reconstruction process is repeated with the predicted solution $\overline{U}_i$. The flux quantities are calculated by solving the

Riemann problem Eqs. 5.13 and 5.14 and denoted as $\overline{F}_{i+\frac{1}{2}}$.

Step 4: Corrector Step. The corrector step uses the trapezoidal method to calculate the second order accurate evolution of the conserved variables.

$$U_i^n = U_i + \frac{1}{2}\frac{\Delta t}{\Delta x}\left[F^{\star}_{i+\frac{1}{2}} - F^{\star}_{i-\frac{1}{2}}\right] + \triangle t \cdot S\left(\overline{U}_i\right), \tag{5.15}$$

where $F^{\star}_{i+\frac{1}{2}} = F_{i+\frac{1}{2}} + \overline{F}_{i+\frac{1}{2}}$.

5.2.2 Riemann Solvers

Various Riemann solvers exist to solve these hyperbolic equations, but two are discussed here: The HLL solver and the FORCE flux solver.

The HLL Solver

The HLL Solver [14, 15] is an approximate Riemann solver created by Harten, Lax and van Leer (HLL). The HLL method assumes a two-wave exact solution and approximates the intercell fluxes with calculated wave speeds. The intercell flux for the HLL method using a Gudunov update is

$$F^{HLL}_{i+\frac{1}{2}} \begin{cases} F_L & \text{, if } 0 \leq S_L \\ \frac{S_R F_L - S_L F_R + S_L S_R (U_R - U_L)}{S_R - S_L} & \text{, if } S_L \leq 0 \leq S_R \\ F_R & \text{, if } 0 \geq S_R \end{cases}, \tag{5.16}$$

where F_L is the flux at the left interface of the right cell, F_R is the flux at the right interface of the left cell, S_L is the left wave speed of the right cell, S_R is the right wave speed of the left cell and $U_{L,R}$ are the conserved quantities at each time step. To make use of this equation, a function that estimates the left and right wave speeds is needed. The wave speeds used in Torrilhon [2] are based on a constant multiplied by the square root of the temperature

$$S_{L,R} = \pm k\sqrt{\frac{P_{L,R}}{\rho_{L,R}}}, \tag{5.17}$$

where k is a parameter that varies with each problem and is determined by experimentation. The HLL method is fairly simple to implement but does suffer from a few drawbacks. One, the wave speeds must be known ahead of time. For the Euler equations, this is well explored, but not for the 13-moment equations. Two, the HLL method assumes a two-wave solution, with three uniform, intermediate states. This means that it will not capture contact discontinuities, shear waves, or any other type of non-two-wave phenomena.

FORCE flux solver

The FORCE scheme is an approximate Riemann solver introduced by Toro [14]. This First ORder CEntered scheme has a CFL restriction of unity in one spatial dimension, where

$$CFL = \frac{\Delta t}{\Delta x} S_{max} \tag{5.18}$$

is the Courant-Friedrichs-Lewy number. This is described as the ratio of the wave speed of the system and the grid speed $\Delta x/\Delta t$ due to the domain discritization[14]. The FORCE flux is derived by using integral averages of Riemann problem solutions which coincidentally reduce to the average of the Richtmyer and Lax-Friedrichs schemes

$$U^{RI}_{i+\frac{1}{2}} = \frac{1}{2}\left(U_L + U_R\right) + \frac{1}{2}\frac{\triangle t}{\triangle x}\left[F\left(U_L\right) - F\left(U_R\right)\right], \tag{5.19}$$

$$F^{RI}_{i+\frac{1}{2}} = F\left(U^{RI}_{i+\frac{1}{2}}\right), \tag{5.20}$$

$$F^{LF}_{i+\frac{1}{2}} = \frac{1}{2}\left[F\left(U_L\right) + F\left(U_R\right)\right] + \frac{1}{2}\frac{\triangle x}{\triangle t}\left(U_L - U_R\right), \tag{5.21}$$

$$F^{FORCE}_{i+\frac{1}{2}} = \frac{1}{2}\left[F^{LF}_{i+\frac{1}{2}}\left(U_L, U_R\right) + F^{RI}_{i+\frac{1}{2}}\left(U_L, U_R\right)\right]. \tag{5.22}$$

In general, Toro [14] states that the FORCE flux is less accurate than the Gudunov flux, but superior to Lax-Friedrichs while being fairly simple and efficient. In the problems studied here, it was comparable to the HLL flux when using Heun's method to integrate in time and has the added benefit that estimates of the wave speeds are not needed.

5.2.3 Slope Limiters

One issue with higher order (than first) schemes are that they have difficulty resolving shocks. In regions with strong discontinuities, these solutions tend to oscillate in a non-physical way. One method to reduce these spurious oscillations is to use a slope limiter that limits the slope in a way that the values remain physical. For the Heun scheme, the difference between cell values (Δ_i) is modified so that

$$\overline{\Delta_i} = \phi \Delta_i. \tag{5.23}$$

The slope limiter used here is known as the Van Leer limiter, where

$$r = \frac{\Delta_{i-\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}}, \tag{5.24}$$

$$\phi_{vl}(r) = \begin{cases} 0 & r \leq 0 \\ \frac{2r}{1+r} & r > 0 \end{cases}, \tag{5.25}$$

with $\Delta_{i-\frac{1}{2}} = V_i^n - V_{i-1}^n$ and $\Delta_{i+\frac{1}{2}} = V_{i+1}^n - V_i^n$. The limiter acts to drop the scheme's accuracy to first order in regions of discontinuities and approaches second order in regions of smooth solutions.

5.2.4 Poisson Solver

The electric field appears in the source terms of Eqs. (5.3-5.8) and must be solved from a model that relates the field to the fluid variables. For an electrostatic problem, Gauss's Law must be satisfied

$$\nabla \cdot E = \frac{\rho_v}{\epsilon_0}, \tag{5.26}$$

where ϵ_0 is the permittivity of free space and ρ_v is the charge density,

$$\rho_v = \sum_\alpha n_\alpha q_\alpha. \tag{5.27}$$

In practice it is usually easier to solve for the electrostatic potential (ϕ), which is related to the charge density by Poisson's equation.

$$\nabla^2 \phi = -\frac{\rho_v}{\epsilon_0}, \tag{5.28}$$

and the electric field is the gradient of the electric potential

$$\vec{E} = -\nabla \phi. \tag{5.29}$$

In the one dimensional case, Poisson's equation reduces to

$$\frac{\partial^2 \phi}{\partial x^2} = -\frac{\rho_v}{\epsilon_0}. \tag{5.30}$$

The electric potential is calculated by using a second order central differencing [17] scheme, where

$$\frac{\partial^2 f}{\partial x^2} = \frac{f\left(x+\Delta x\right) - 2f\left(x\right) + f\left(x-\Delta x\right)}{\Delta x^2} \tag{5.31}$$

for any arbitrary function $f\left(x\right)$. Boundary conditions must be specified for the electrostatic potential. The boundary conditions that produced comparable results to [3, 16] were found to be

$$\begin{aligned} \phi'\left(x_L\right) &= 0, & (5.32) \\ \phi\left(x_R\right) &= 0, & (5.33) \end{aligned}$$

where x_L is the left boundary and x_R is the right boundary. The equation can be represented as a matrix problem where $\mathbf{A}\mathbf{x} = \mathbf{b}$, with the matrices equal to

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & -2 & 0 & \cdots & & & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & & \cdots & 0 & \vdots \\ \vdots & -1 & 2 & -1 & 0 & \cdots & & \vdots & \\ & 0 & \ddots & \ddots & \ddots & & & & \\ & \vdots & & & & & & \vdots & \\ & & & & & & & 0 & \\ & \vdots & & & & \ddots & \ddots & -1 & \vdots \\ \vdots & 0 & \cdots & & & 0 & -1 & 2 & -1 \\ 0 & \cdots & & & & & \cdots & 0 & 1 \end{bmatrix} \tag{5.34}$$

and

$$\mathbf{x} = \begin{bmatrix} \phi\left(x_L\right) \\ \phi\left(x_1\right) \\ \vdots \\ \phi\left(x_{R-1}\right) \\ \phi\left(x_R\right) \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ -\Delta x^2 \rho_v\left(x_1\right) \\ \vdots \\ -\Delta x^2 \rho_v\left(x_{R-1}\right) \\ 0 \end{bmatrix}. \tag{5.35}$$

The electric field is then computed by taking the gradient of the electrostatic potentials (ϕ's) as

$$E^n_{x_i} = \frac{\phi^n_{i+1} - \phi^n_{i-1}}{2\Delta x} \tag{5.36}$$

for each time step, which can be used to compute the source terms.

5.3 Non-Dimensionalization of the Equations

It is often useful to non-dimensionalize the equations in terms of unitless ratios. This enables different phenomenon to be captured without computationally expensive resolutions. For example, varying the characteristic scale of the domain allows one to "zoom" in or out without having to change grid size. The similitude property allows for simulations where proper scales are defined. The non-dimensionalization here is designed to match that of [3, 16]. In the two-fluid model presented by Shumlak [3], the length is non-dimensionalized by the characteristic plasma length x_o, the electron and ion velocities by the characteristic ion thermal velocity $v_{th,i}$, etc. The table below shows the complete non-dimensionalization.

Variable	Non-Dimensional Parameter	Characteristic	Symbol	Expression
Density	Ion Mass Density	$n_0 m_i$	$\rho^\star$	$= \frac{\rho}{n_0 m_i}$
Velocity	Ion Thermal Velocity	$v_{th,i}$	$u^\star$	$= \frac{u}{v_{th,i}}$
Pressure	Ion Dynamic Pressure	$n_0 m_i v_{th,i}^2$	$p^\star$	$= \frac{p}{n_0 m_i v_{th,i}^2}$
Time	Ion Transit Time	$x_o/v_{th,i}$	$t^\star$	$= \frac{t}{x_o/v_{th,i}}$
Length	Plasma Scale	x_0	$x^\star$	$= \frac{x}{x_0}$
Charge	Ion Charge	q_i	$q^\star$	$= \frac{q}{q_i}$
Magnetic Field	Magnetic Field Strength	B_0	$B^\star$	$= \frac{B}{B_0}$
Electric Field	–	$B_0 v_{th,i}$	$E^\star$	$= \frac{E}{B_0 v_{th,i}}$
Mass	Ion Mass	m_i	$m^\star$	$= \frac{m}{m_i}$

To non-dimensionalize, the variables are simply substituted into the original expressions. For example, for the non-dimensionalized density the expression is $\rho^\star = \frac{\rho}{n_0 m_i}$ or by re-arranging the original expression for density is $\rho = \rho^\star n_0 m_i$. Using the parameters above to non-dimensionalize the momentum equation,

$$\frac{\partial \rho_\alpha u_\alpha}{\partial t} + \nabla \cdot \left(\rho_\alpha u_\alpha^2 + p\right) = \frac{\rho_\alpha q_\alpha}{m_\alpha} \vec{E} \tag{5.37}$$

gives the following substitutions in 1-dimension

$$\frac{\partial \rho^\star n_0 m_i u_\alpha}{\partial t^\star \left(x_o/v_{th,i}\right)} + \frac{\partial}{\partial x^\star x_0}\left(\rho^\star n_0 m_i \left(u^\star v_{th,i}\right)^2 + p^\star n_0 m_i v_{th,i}^2\right) = \frac{\rho^\star n_0 m_i q^\star q_i}{m^\star m_i} E^\star v_{th,i} B_0. \tag{5.38}$$

Simplifying and dropping the $\star$'s give the complete non-dimensionalized equation

$$\partial_t \rho u + \partial_x \left(\rho u^2 + p\right) = \pm \left(\frac{x_0}{r_{gi}}\right) n \vec{E}, \tag{5.39}$$

where r_{gi} is the ion Larmor radius $\frac{m_i}{q_i}\frac{v_{th,i}}{B_0}$. In this electrostatic simulation, the change in Larmor radius corresponds to a change in characteristic Debye length. For the 13-moment equations, an additional non-dimensionalization is made for the heat-flux

q_x, which is non-dimensionalized by the ion heat flow

$$q_x^\star = \frac{q_x}{n_o m_i v_{th,i}^3}. \tag{5.40}$$

Following the same procedure as for the momentum equation, the complete set of equations are very similar except for a scaling factor for the source terms

$$\partial_t U + \partial_x F(U) = \pm \left(\frac{x_0}{r_{gi}}\right) S(U). \tag{5.41}$$

The sources are positive for the ion fluid, and negative for the electron fluid. Poisson's equation becomes

$$\nabla \cdot \vec{E} = \left(\frac{x_0}{r_{gi}}\right)\left(\frac{r_{gi}}{\lambda_d}\right)^2 (n_i - n_e) \tag{5.42}$$

with the Debye length defined as

$$\lambda_d = \left(\frac{\epsilon_0 m_e v_{th,e}^2}{n_\emptyset q_e^2}\right)^{1/2}, \tag{5.43}$$

where $q_e = e$. Factors for $\frac{x_0}{r_{gi}}$ and $\frac{r_{gi}}{\lambda_d}$ are included as coefficients in the numerical simulation.

5.4 Simulation Algorithm

A one-dimensional, second order finite volume code is written using the elements presented here. The code is listed in Appendix **C**. The flow-charts presented here demonstrate the basic code structure.

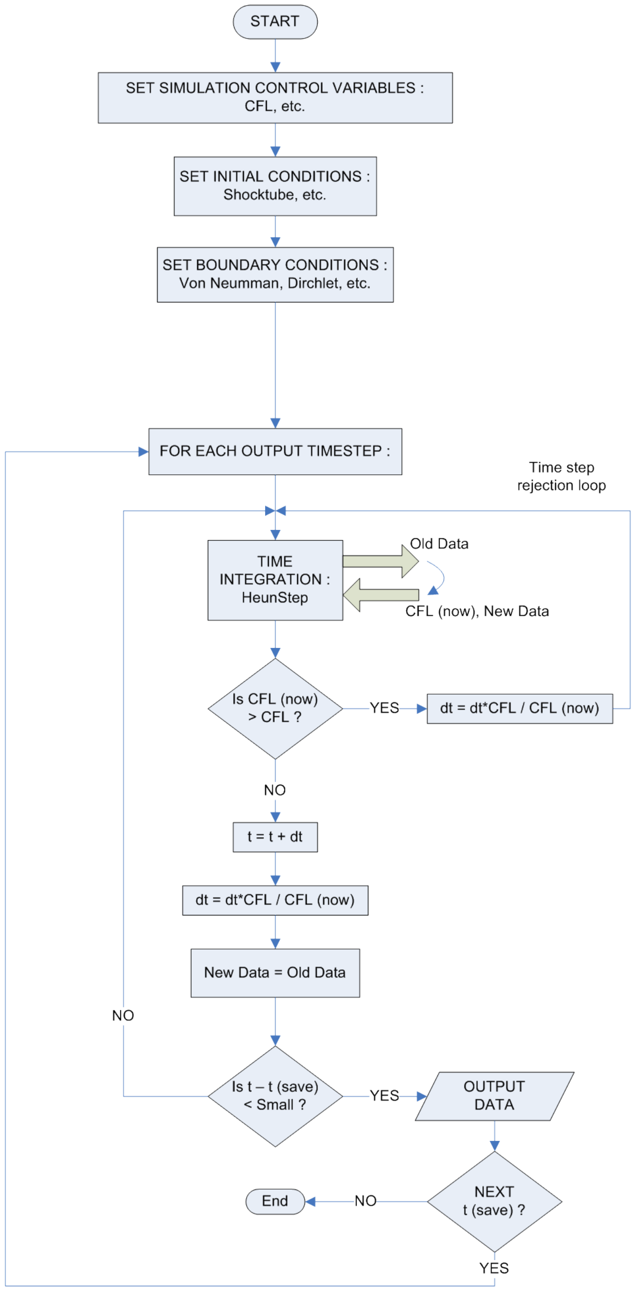

START
SET SIMULATION CONTROL VARIABLES : CFL, etc.
SET INITIAL CONDITIONS : Shocktube, etc.
SET BOUNDARY CONDITIONS : Von Neumman, Dirchlet, etc.
FOR EACH OUTPUT TIMESTEP :
Time step rejection loop
TIME INTEGRATION : HeunStep
Old Data
CFL (now), New Data
Is CFL (now) > CFL ?
YES
dt = dt*CFL / CFL (now)
NO
t = t + dt
dt = dt*CFL / CFL (now)
New Data = Old Data
NO
Is t – t (save) < Small ?
YES
OUTPUT DATA
NEXT t (save) ?
NO
End
YES

Figure 5.1: Flow chart showing the main program loop.

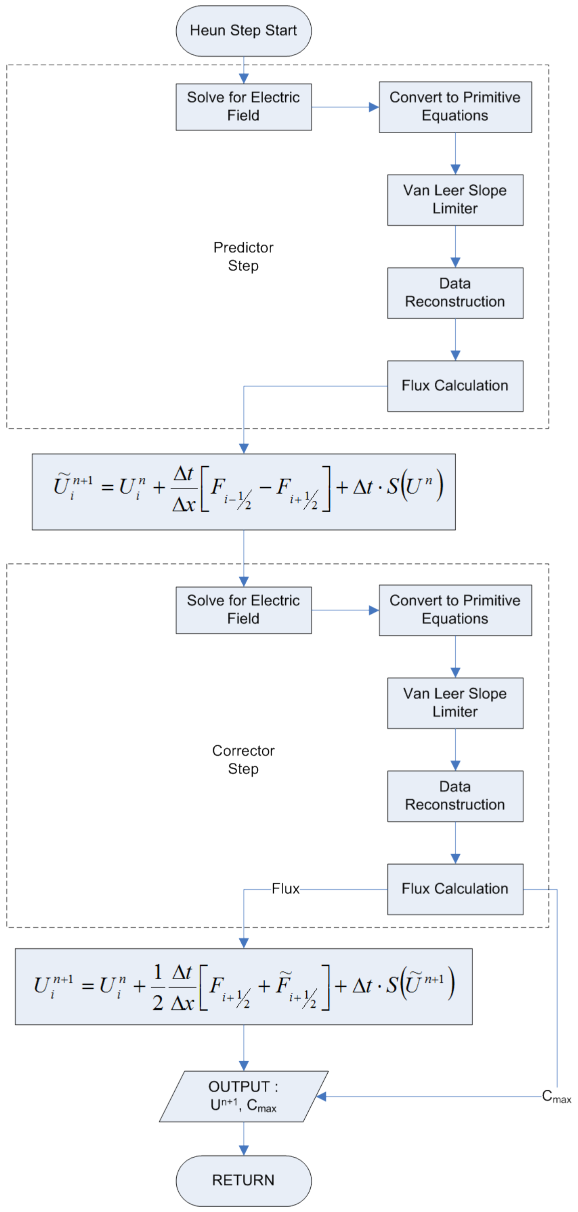

Heun Step Start
Solve for Electric Field
Convert to Primitive Equations
Van Leer Slope Limiter
Data Reconstruction
Flux Calculation
Predictor Step
$\tilde{U}_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x}\left[F_{i-1/2} - F_{i+1/2}\right] + \Delta t \cdot S(U^n)$
Solve for Electric Field
Convert to Primitive Equations
Van Leer Slope Limiter
Data Reconstruction
Flux Calculation
Corrector Step
Flux
$U_i^{n+1} = U_i^n + \frac{1}{2}\frac{\Delta t}{\Delta x}\left[F_{i+1/2} + \tilde{F}_{i+1/2}\right] + \Delta t \cdot S(\tilde{U}^{n+1})$
C_{max}
OUTPUT : U^{n+1}, C_{max}
RETURN

Figure 5.2: Flow chart showing the time advance of the electrostatic two-fluid plasma equations using a Heun method.

Chapter 6

NUMERICAL SIMULATIONS

6.1 13-Moment Single Fluid Results

To ensure the correct computer implementation of the model, it is important to benchmark the results to those of Torrilhon [2]. To simplify the two-fluid plasma code presented in Appendix C to handle single-fluid gas dynamics phenomenon, it is only necessary to set the source term coefficients ($\frac{x_0}{r_{gi}}$ and $\frac{r_{gi}}{\lambda_d}$) to a very small number, such as 10^{-16} (to avoid dividing by zero) and give each fluid the same initial conditions for density, velocity, pressure, etc. The results from each fluid are identical. To increase or decrease the time step, characteristic speeds can be modified in the flux calculations as necessary. Appendix D contains the Mathematica code necessary to determine the characteristic speeds. Initial conditions for the "Strong Shock" problem [2] are given by

	$x < 0$	$x \geq 0$
ρ	50	1
u	0	0
p	50	1
P_{xx}	50	1
q_x	0	0

6.1.1 Strong Shock Tube Simulation with Singular Closure

The strong shock tube problem is run using the FORCE flux method described in Chapter 5 with a grid resolution of $\Delta x = 7.5 \times 10^{-4}$ and a CFL number of 0.9 until $t = 0.4$. The time step is calculated using information from the estimated maximum

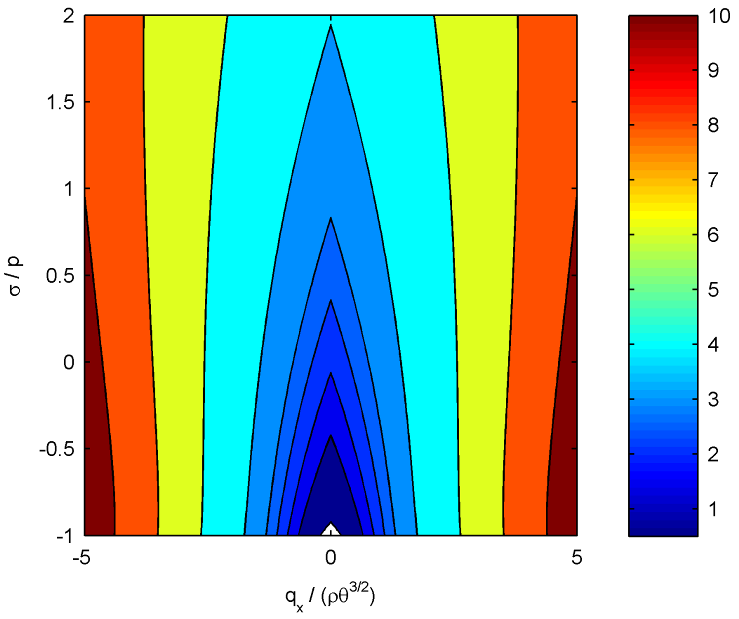


Figure 6.1: Contour plot of the characteristic wave speeds for the singular closure scheme showing c_{max} for a given set of parameters.

wave speed, which was chosen as $c_{max} = 5\sqrt{\theta}$ to match the conditions in Torrilhon[2]. Figure 6.1 shows the characteristic wave speeds for the singular closure scheme used to estimate c_{max}.

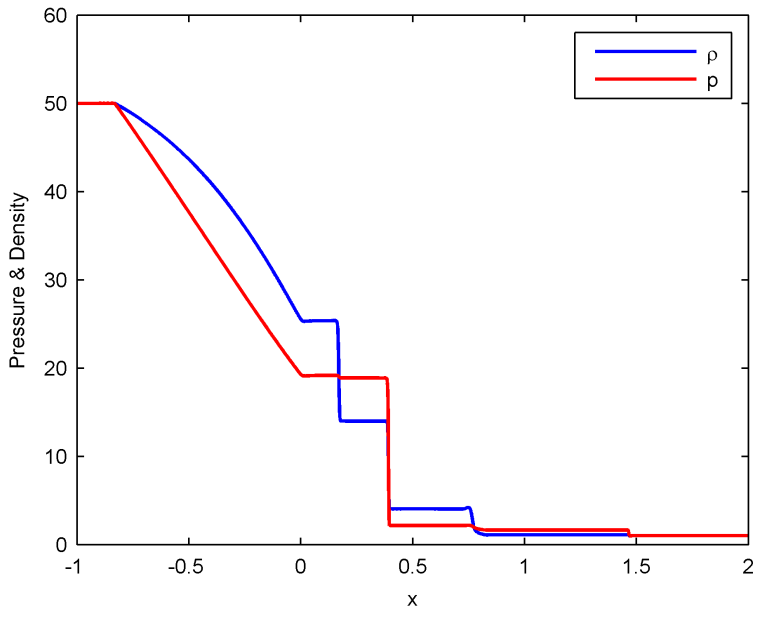


Figure 6.2: Pressure and density for the strong shock tube problem using the singular closure scheme at $t = 0.4$ for the 13-moment single-fluid gas dynamics model.

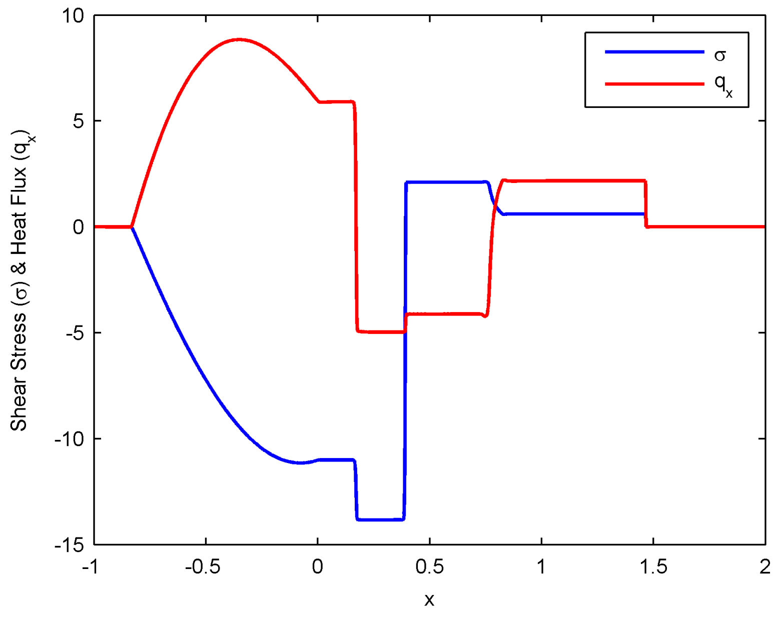


Figure 6.3: Shear stress ($\sigma = P_{xx} - p$) and heat flux for the strong shock tube problem using the singular closure scheme at $t = 0.4$ for the 13-moment single-fluid gas dynamics model.

The results seen in Figures 6.2 and 6.3 show good agreement with the results Torrilhon presents. The switch to using a FORCE scheme here in comparison to an HLL scheme does not seem to have made a large impact on the quality of the results. The strong non-equilibrium effects in the shear and heat flux of Fig. 6.3 are due to the lack of a collision operator to drive the model towards an equilibrium.

Fig. 6.2 shows significant differences between the 5 and 13-moment models. The 13-moment model has several waves not captured in the 5-moment case, seen as having more plateaus ($-0.5 \leq x \leq 1$) than predicted by the Euler equations. The 13-moment model's faster wave speeds can be seen at $x \sim -1$ and $x \sim 1.5$.

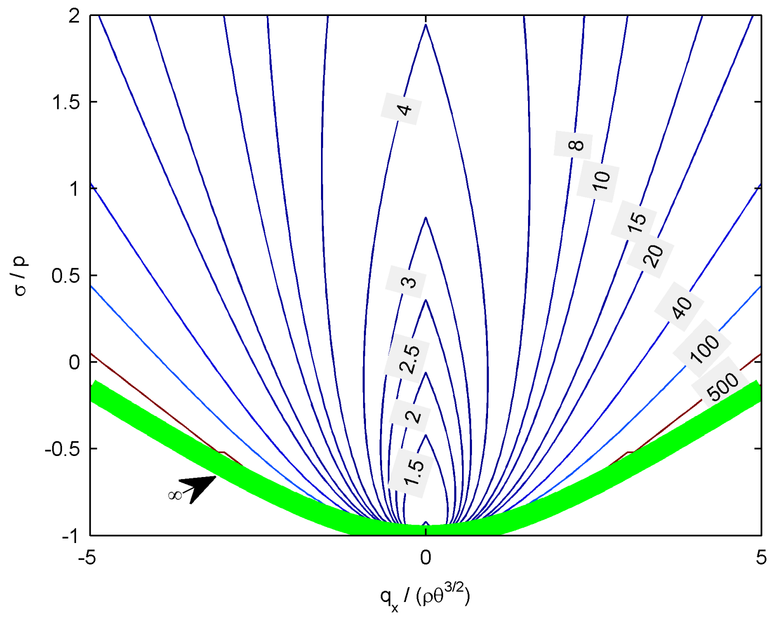


Figure 6.4: Contours of the characteristic wave speeds for the realizable closure scheme showing c_{max} for a given set of parameters. The green line represents a trend towards infinite wave speeds not seen in the singular closure scheme.

6.1.2 Strong Shock Tube Simulation with Realizable Closure

As with the single-fluid results for the singular closure, the realizable results are run using the FORCE flux method with a grid resolution of $\Delta x = 7.5 \times 10^{-4}$ and a CFL number of 0.9 until $t = 0.4$. The time step is calculated using information from the estimated maximum wave speed, which is chosen as $c_{max} = 30\sqrt{\theta}$, as per Torrilhon [2]. Fig. 6.4 shows the wave speeds for the realizable closure scheme.

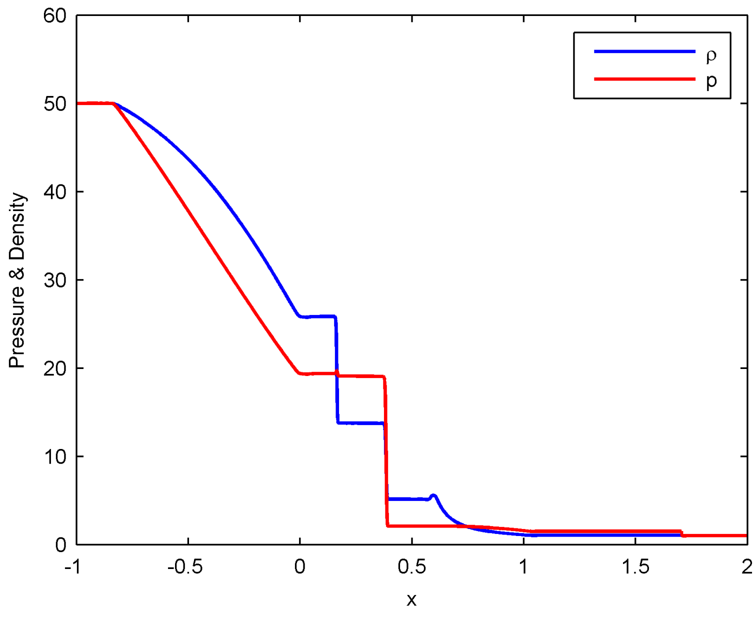


Figure 6.5: Pressure and density for the strong shock tube problem using the realizable closure scheme.

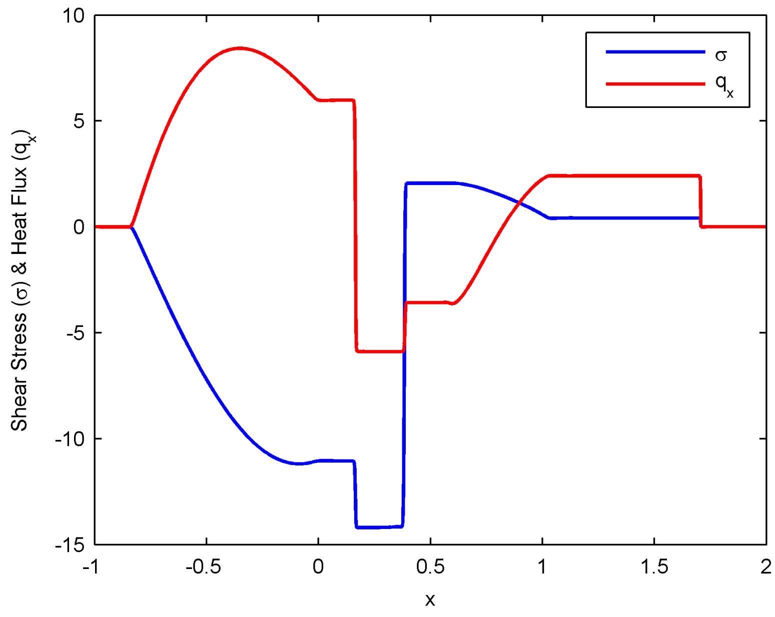


Figure 6.6: Shear stress and heat flux for the strong shock tube problem using the realizable closure scheme.

Again Figures 6.5 and 6.6 show the results seen in [2]. There seems to be little difference attributed to our use of the FORCE scheme over the HLL scheme. The shear and heat flux results seen in Fig. 6.6 show real differences between the singular and realizable closure schemes. The realizable closure has wave speeds that increase exponentially and that can be seen as the right shock waves ($x \sim 1.5 - 2$) have progressed further than their singular closure counterparts over the same amount of time. These faster waves require a significantly smaller time step to resolve, which becomes computationally expensive.

6.2 13-Moment Two-Fluid Plasma Results

Two-fluid plasma results are presented for a comparison to the 5-moment electrostatic simulations present in [3, 16]. These results are the two-fluid plasma equivalent to the Brio-Wu [22] MHD shock tube simulation, which has become a standard test case in plasma physics codes. The initial conditions for the 13-moment two-fluid plasma system are

	$x < 0$	$x \geq 0$
ρ_i	1.0	0.125
u_i	0	0
p_i	0.5	.05
$p_{xx,i}$	0.5	.05
$q_{x,i}$	0	0
ρ_e	$1.0\frac{m_e}{m_i}$	$0.125\frac{m_e}{m_i}$
u_e	0	0
p_e	0.5	.05
$p_{xx,e}$	0.5	.05
$q_{x,e}$	0	0

6.2.1 13-Moment Singular Closure Shocktube with $\frac{x_0}{r_{gi}} = 0.1$

The first 13-moment electrostatic test case is for the singular closure with $\frac{x_0}{r_{gi}} = 0.1$ and $\frac{r_{gi}}{\lambda_d} = 100$. For the case of an electrostatic simulation, the Larmor radius r_{gi} is infinite and the ratio x_0/r_{gi} is instead analogous to the number of Debye lengths per characteristic length x_0. The nomenclature, however, is chosen to match published results [3, 16]. The simulation is run at a resolution of $\Delta x = 2.5\times10^{-4}$, a CFL number of 0.9, and using the singular closure scheme. To match the setup for the 5-moment electrostatic model[3, 16], the simulation is advanced 10 light transit times. With a normalized speed of light $c = 100v_{th,i}$, this is equivalent to $t = 0.1$ for normalized time.

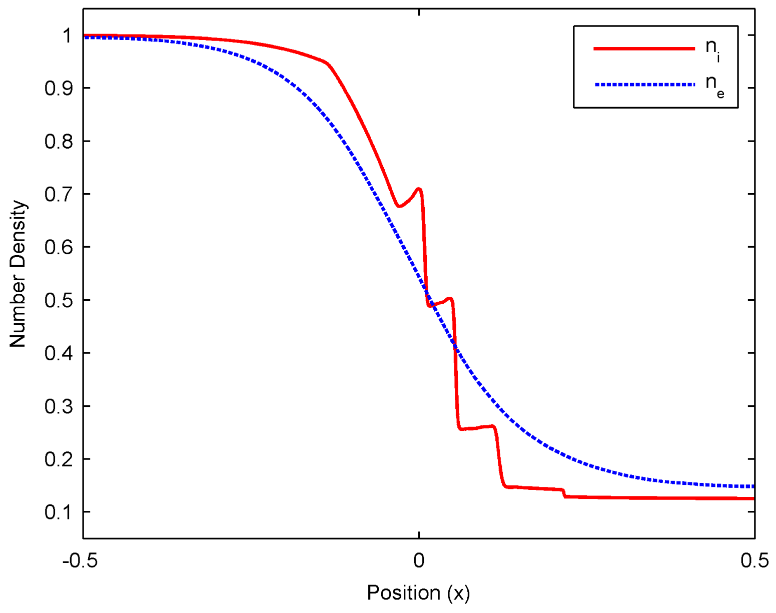


Figure 6.7: Number densities for the 13-moment electrostatic shock tube problem using the singular closure and with $\frac{x_0}{r_{gi}} = 0.1$

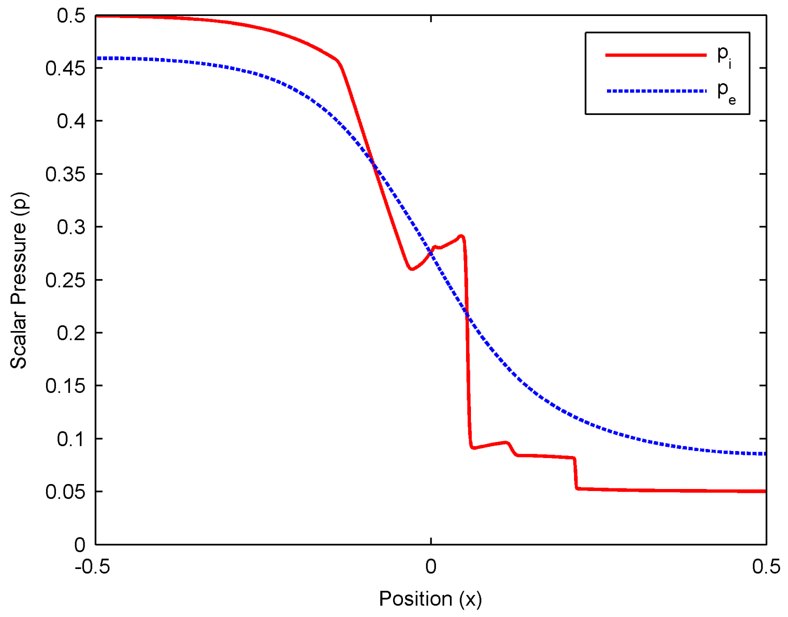


Figure 6.8: Pressures for the 13-moment electrostatic shock tube problem using the singular closure and with $\frac{x_0}{r_{gi}} = 0.1$

Figures 6.7 and 6.8 show significant charge separation. Without electrostatic effects, the electron fluid would have reached equilibrium much faster due to the lower mass and faster wave speeds of the electrons. Here we see the electron motion retarded by the ion fluid.

6.2.2 *Direct Comparison of Singular and Realizable Closure Schemes* ($\frac{x_0}{r_{gi}} = 0.1$)

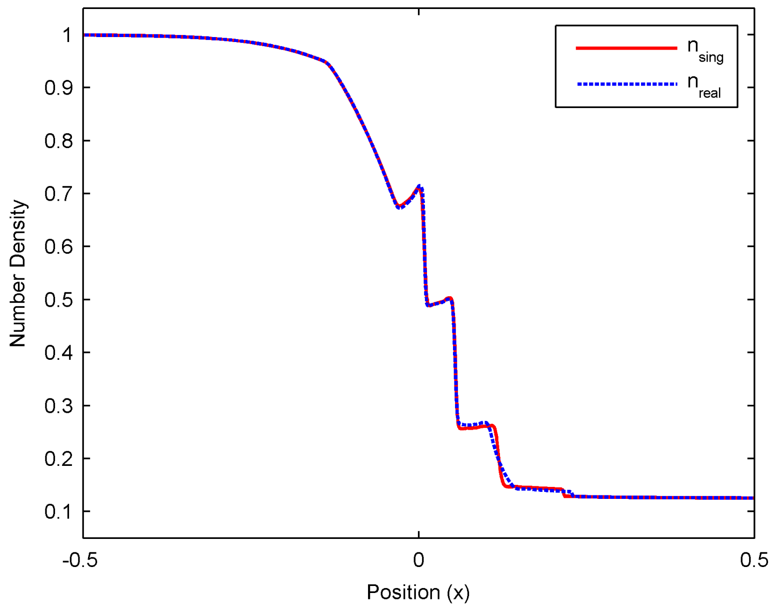


Figure 6.9: Comparison of ion densities for the singular and realizable closures for $\frac{x_0}{r_{gi}} = 0.1$

A direct comparison between the singular and realizable closures for $\frac{x_0}{r_{gi}} = 0.1$ and $\frac{r_{gi}}{\lambda_d} = 100$ can be seen in Fig. 6.9. Any differences between the two solutions is minor. This is to be expected since the conditions are not far from equilibrium and therefore this problem does not extend to the regions far from equilibrium where the vastly different wave speeds cause the solutions to diverge.

6.2.3 *13-Moment Singular Closure Shocktube with* $\frac{x_0}{r_{gi}} = 1$

To understand the effects of the source terms on the model, successive simulations are run for increasing plasma scales ($\frac{x_0}{r_{gi}}$). These results represent a simulation of the singular closure scheme with $\frac{x_0}{r_{gi}} = 1$ and $\frac{r_{gi}}{\lambda_d} = 100$ run at a resolution of $\Delta x = 2.5 \times 10^{-4}$. In Fig. 6.10 it can be seen that the electron fluid has almost entirely approached the ion fluid.

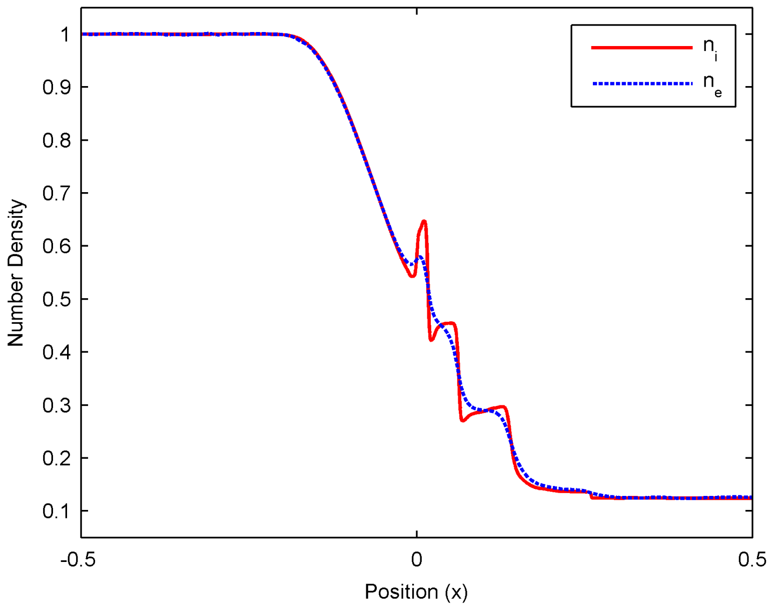


Figure 6.10: Number densities for the 13-moment electrostatic shock tube problem with the singular closure and $\frac{x_0}{r_{gi}} = 1$.

6.2.4 13-Moment Singular Closure Shocktube with $\frac{x_0}{r_{gi}} = 10$

With another order of magnitude change to the plasma scale ($\frac{x_0}{r_{gi}}$), Fig. 6.11 shows that there is no discernable separation between the electron and ion fluids. The ion fluid shows what would be expected for a solution from gas dynamics, and the electron fluid is so much lighter that the source terms force it to follow the ion motion in an ambipolar sense.

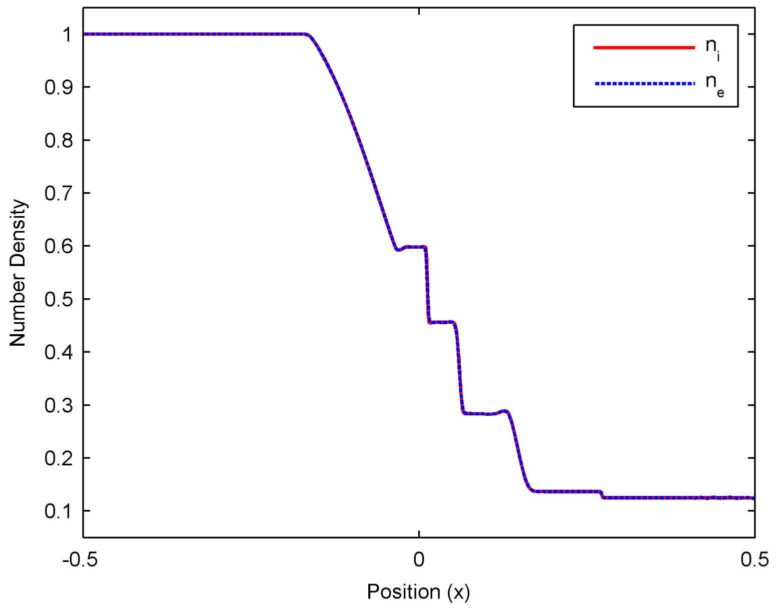


Figure 6.11: Number density for the electrostatic shock tube problem with the singular closure and $\frac{x_0}{r_{gi}} = 10$.

6.2.5 Direct Comparisons to 5-Moment Model

It is relevant at this point to discuss the differences seen between the 5 and 13-moment models. Simulations are run for both models at $\frac{x_0}{r_{gi}} = 0.1$ and $\frac{r_{gi}}{\lambda_d} = 100$ with a resolution of $\Delta x = 2.5 \times 10^{-4}$. The ion number densities best showcase the waves of both simulations and Fig. 6.12 shows that there are significant differences between the two models. Several additional wave structures are seen in the 13-moment model that are not present in the 5-moment model, which can also be seen in the purely gas-dynamical solutions. Of additional interest are the differences in the maximum wave speeds between the two models. It can be seen from the left side of the domain ($x \sim -0.2$), for example, that the 13-moment model generates a faster wave speed for the left rarefaction wave than the 5-moment model. Similarly, the fast shock on

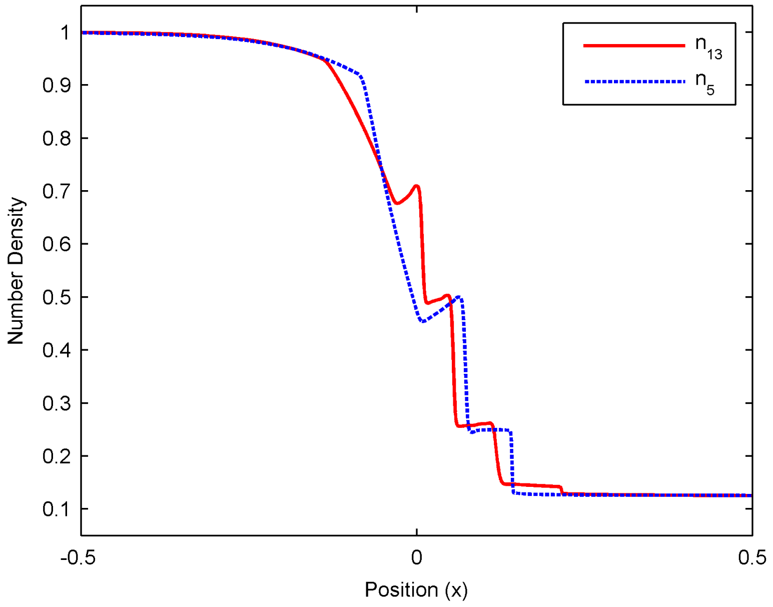


Figure 6.12: Direct comparison of the ion number densities for the 5 and 13-Moment models. $\frac{x_0}{r_{gi}} = 0.1$

the right ($x \sim 0.2$) is not present in the 5-moment model results.

Chapter 7
CONCLUSIONS AND FUTURE WORK

7.1 Concluding Remarks

In this paper, a 13-moment two-fluid plasma model based on a Pearson type-IV distribution function is developed. The basic qualities of the 13-moment fluid model and the two-fluid plasma physics model are explored, as well as a review of the Pearson-IV distribution function and the essential mathematics necessary for the 13-moment derivation. The 13-moment model is formally derived from moments of the Boltzmann equation to include Lorentz force and collisional source terms. Eigensystem analysis derives the eigenvalues and eigenvectors of the full three-dimensional 13-moment plasma model. Finally, a few examples are conducted in one dimension, including the single-fluid gas dynamic and two-fluid plasma models.

The derivation of the 13-moment model includes several collisional operators that would drive a system towards equilibrium. In the one dimensional model implemented here, those terms are neglected for a collisionless plasma. While this assumption is valid for a variety of plasmas, it does limit the simulation to a specific class of plasmas. The exclusion of magnetic fields in the simulation also limits the model such that it can only capture electrostatic physics phenomena. While both of these are fairly stringent assumptions, the purpose of this model is to build a first foray into using the thirteen moment model for two-fluid physics and as such it does create a baseline that can be improved upon.

Further analysis of the simulation shows that there is good agreement between the 13-moment model electrostatic simulations and those of the 5-moment model seen in current literature [3, 16]. Differences can still be seen between the waves seen in the plasma and gas dynamics models and the 13-moment model clearly captures more physics. The differences in the two closure schemes are demonstrated in the single-fluid gas dynamic shock tube, but the electrostatic shock does not venture into a region where the differences in wave speeds between these two closure schemes is readily apparent. However, the goal is to understand the differences based on a very specific set of conditions.

In the end, the 13-moment model successfully demonstrates a two-fluid plasma solution to an electrostatic shock tube problem. Even without exercising the model for these extended regions of hyperbolicity it shows fundamental physics not captured by the 5-moment model. Though these tests are by no means an exhaustive study of the 13-moment model, they demonstrate that the 13-moment equations can be useful in solving a two-fluid plasma model.

7.2 Future Work

The potential of the 13-moment two-fluid plasma model is still largely unexplored at this point. The simulations presented in this paper are of limited application without further work, especially the limitation to only electrostatic problems. The first extension of this model should probably include a complete solver for Maxwell's equations that will allow for comparisons to an electromagnetic shock tube found as a classic test case for two-fluid plasma codes. The inclusion of source terms to drive solutions towards an equilibrium, Maxwellian distribution would also greatly extend the validity region to include collisional plasmas and is an important step towards further validating the model.

Currently the simulations are limited to problems that can be solved by the 5-moment models so that direct comparisons can be made. It would be interesting, however, to see more simulations such as the strong shock tube that, because of their limited regions of hyperbolicity, are outside of the capabilities of current models to resolve. Also, a further exploration of the consequences of choosing a closure scheme need to be explored to fully understand each scheme's limitations.

The eventual goal is to implement this model in three dimensions. For that, an efficient solver capable of parallel processing will be needed. Towards that end, the target code for this model is the WARPX (Washington Approximate Riemann Problem Solver Version X) simulation developed by the Computational Plasma Dynamics Lab at the University of Washington, led by Dr. Shumlak. The eigenvalues and eigenvectors calculated in this paper are the first steps required to implement a new system of equations in this software, and it is expected that the next steps will soon follow.

Finally, it is likely that the finite volume method presented in this paper will eventually give way to a higher order scheme like discontinuous Galerkin. The different wave speeds seen in plasma physics have shown themselves to be difficult to resolve in a finite volume scheme. Promising research has shown that higher order methods can resolve these phenomenon accurately and efficiently [19].

Appendix A
DETAILS OF THE PEARSON-IV DERIVATIONS

A.1 In One Dimension

The non-normalized distribution function in one dimension can be defined as

$$f^{(nn)} = \frac{e^{-\nu \tan^{-1}(\frac{c-\lambda}{a})}}{\left[1+(\frac{c-\lambda}{a})^2\right]^m}. \tag{A.1}$$

A.1.1 Zeroth Moment

Taking the zeroth raw moment (μ_0) with area under the non-normalized distribution curve (α),

$$\alpha\mu_0 = \int_{-\infty}^{\infty} \frac{e^{-\nu \tan^{-1}(\frac{c-\lambda}{a})}}{\left[1+(\frac{c-\lambda}{a})^2\right]^m} dc \tag{A.2}$$

and letting

$$\begin{aligned} \tilde{c} &= \frac{c-\lambda}{a} \quad \Leftrightarrow \quad c = a\tilde{c}+\lambda \\ d\tilde{c} &= \frac{1}{a}dc \quad \Leftrightarrow \quad dc = a\,d\tilde{c} \end{aligned} \tag{A.3}$$

and substituting into the zeroth moment yields

$$\alpha\mu_0 = a\int_{-\infty}^{\infty} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c}. \tag{A.4}$$

Defining the one dimensional normalization constant (k) as,

$$k = \int\limits_{-\infty}^{\infty} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c}. \tag{A.5}$$

By definition, the area under the curve of the zeroth moment of a normalized distribution function is 1 (i.e. $\mu_0 = 1$). Dividing by the area of the non-normalized function (α) gives a normalized distribution function as stated in [2, 6].

$$f^{(P1)} = \frac{1}{a\,k} \frac{e^{-\nu \tan^{-1}(\frac{c-\lambda}{a})}}{\left[1+(\frac{c-\lambda}{a})^2\right]^m}. \tag{A.6}$$

A.1.2 First Moment

The first moment is a raw moment of the distribution function and defines the average or bulk velocity v

$$v = \frac{1}{a\,k} \int\limits_{-\infty}^{\infty} c \frac{e^{-\nu \tan^{-1}(\frac{c-\lambda}{a})}}{\left[1+(\frac{c-\lambda}{a})^2\right]^m} dc. \tag{A.7}$$

Substituting Eq. A.3 as before,

$$\begin{aligned}
v &= \frac{1}{a\,k} \int\limits_{-\infty}^{\infty} (a\tilde{c}+\lambda) \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} a d\tilde{c} \qquad \text{(A.8)} \\
&= \frac{1}{a\,k} \int\limits_{-\infty}^{\infty} \left(a^2\tilde{c}\right) \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c} + \frac{1}{a\,k} \int\limits_{-\infty}^{\infty} (a\lambda) \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c} \\
&= \frac{a}{k} \int\limits_{-\infty}^{\infty} \tilde{c} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c} + \lambda \cancel{\frac{1}{k} \int\limits_{-\infty}^{\infty} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m}} d\tilde{c} \\
&= \lambda + \frac{a}{k} \int\limits_{-\infty}^{\infty} \tilde{c} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c},
\end{aligned}$$

and letting

$$\begin{aligned} x &= \tan^{-1}(\tilde{c}) \quad \Leftrightarrow \quad \tilde{c} = \tan x \\ dx &= \frac{1}{1+\tilde{c}^2} d\tilde{c} \end{aligned} \tag{A.9}$$

and changing the limits of integration such that $x = \tan^{-1}(\infty) = \frac{\pi}{2}$ and $x = \tan^{-1}(-\infty) = -\frac{\pi}{2}$,

$$v = \lambda + \frac{a}{k} \int\limits_{-\pi/2}^{\pi/2} \tan(x) \frac{e^{-\nu x}}{[1+\tan^2(x)]^m} \frac{1}{\cos^2(x)} dx. \tag{A.10}$$

Using the trigonometry identity $\frac{1}{1+\tan^2 x} = \cos^2 x$,

$$\begin{aligned} v &= \lambda + \frac{a}{k} \int\limits_{-\pi/2}^{\pi/2} \frac{\sin(x)\cos^{2(m-1)-1}(x)}{e^{\nu x}} dx \\ &= \lambda + \frac{a}{k} \int\limits_{-\pi/2}^{\pi/2} \frac{\sin(x)\cos^{2(m-1)-1}(x)}{e^{\nu x}} dx, \end{aligned} \tag{A.11}$$

where $\cos^{2(m-1)-1}(x) = [\cos(x)]^{2(m-1)-1}$. Integrating by parts $\int \mathbf{u}\, d\mathbf{v} = \mathbf{uv} - \int \mathbf{v}\, d\mathbf{u}$ and letting

$$\begin{aligned} \mathbf{u} &= e^{-\nu x} \quad \text{and} \quad & d\mathbf{v} &= \sin(x)\cos^{2(m-1)-1}(x) dx \\ d\mathbf{u} &= -\nu e^{-\nu x} & \mathbf{v} &= \frac{\cos^{2(m-1)}(x)}{2(m-1)} \end{aligned} \tag{A.12}$$

gives the expression

$$v = \lambda + \frac{a}{k} \left[\cancel{\left. \frac{\cos^{2(m-1)}(x)}{e^{\nu x} 2(m-1)} \right|_{-\frac{\pi}{2}}^{\frac{\pi}{2}}} + \frac{\nu}{2(m-1)} \int\limits_{-\pi/2}^{\pi/2} \frac{\cos^{2(m-1)}(x)}{e^{\nu x}} dx \right]. \tag{A.13}$$

Transforming back into velocity space results in

$$\begin{aligned} v &= \lambda + \frac{a}{k}\left[\frac{\nu}{2(m-1)}\int\limits_{-\infty}^{\infty}\frac{(1+\tan^2(\tilde{c}))}{(1+\tan^2(\tilde{c}))^m}\cdot e^{-\nu\tan^{-1}(\tilde{c})}\cdot\frac{1}{(1+\tan^2(\tilde{c}))}d\tilde{c}\right] \quad \text{(A.14)} \\ &= \lambda + \frac{a}{k}\frac{\nu}{2(m-1)}\int\limits_{-\infty}^{\infty}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m}d\tilde{c} = \lambda + \frac{a}{\cancel{k}}\frac{\nu}{2(m-1)}\cancel{k}. \end{aligned}$$

Therefore, the average or bulk velocity is equal to

$$v = \lambda + \frac{a\,\nu}{2(m-1)}. \quad \text{(A.15)}$$

A.1.3 General Moments

The remaining nth moments are central moments taken about the average velocity v

$$\alpha\mu_n(c) = \int\limits_{-\infty}^{\infty}(c-v)^n\frac{e^{-\nu\tan^{-1}(\frac{c-\lambda}{a})}}{\left[1+(\frac{c-\lambda}{a})^2\right]^m}dc. \quad \text{(A.16)}$$

Substituting for v and letting

$$\begin{aligned} \tilde{c} &= \frac{c-\lambda}{a} \quad \Leftrightarrow \quad c = a\tilde{c}+\lambda \quad \text{(A.17)} \\ d\tilde{c} &= \frac{1}{a}dc \quad \Leftrightarrow \quad dc = a\,d\tilde{c}, \end{aligned}$$

$$\begin{aligned} \alpha\mu_n(\tilde{c}) &= \int\limits_{-\infty}^{\infty}(a\,\tilde{c}+\cancel{\lambda}-\cancel{\lambda}+\frac{a\,\nu}{2(m-1)})^n\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m}a\,d\tilde{c} \quad \text{(A.18)} \\ &= a^{n+1}\int\limits_{-\infty}^{\infty}(\tilde{c}-\tilde{v})^n\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m}d\tilde{c}. \end{aligned}$$

With $\tilde{v} = -\frac{\nu}{2(m-1)}$, it is easy to see the beginnings of the recursion relationship

$$\begin{aligned}
\int_{-\infty}^{\infty} (\tilde{c}-\tilde{v})^{n-1} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c} &= \frac{\alpha\mu_{n-1}}{a^n}, \qquad \text{(A.19)}\\
\int_{-\infty}^{\infty} (\tilde{c}-\tilde{v})^{n-2} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c} &= \frac{\alpha\mu_{n-2}}{a^{n-1}}.
\end{aligned}$$

Continuing from before,

$$\begin{aligned}
\alpha\mu_n(\tilde{c}) &= a^{n+1} \int_{-\infty}^{\infty} (\tilde{c}-\tilde{v})^{n} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} a\, d\tilde{c} \qquad \text{(A.20)}\\
&= a^{n+1} \int_{-\infty}^{\infty} (\tilde{c}-\tilde{v})(\tilde{c}-\tilde{v})^{n-1} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c}\\
&= \underbrace{a^{n+1} \int_{-\infty}^{\infty} \tilde{c}(\tilde{c}-\tilde{v})^{n-1} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c}}_{ⓐ} - \underbrace{a^{n+1}\tilde{v} \int_{-\infty}^{\infty} (\tilde{c}-\tilde{v})^{n-1} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c}}_{ⓑ},
\end{aligned}$$

where the equations are broken into integrals that are handled individually, e.g.

$$ⓐ = \int_{-\infty}^{\infty} \tilde{c}(\tilde{c}-\tilde{v})^{n-1} \frac{e^{-\nu \tan^{-1}(\tilde{c})}}{[1+(\tilde{c})^2]^m} d\tilde{c}. \qquad \text{(A.21)}$$

Letting

$$\begin{aligned}
\theta = \tan^{-1}(\tilde{c}) \quad &\Leftrightarrow \quad \tilde{c} = \tan\theta \qquad \text{(A.22)}\\
d\theta = \frac{1}{1+\tilde{c}^2} d\tilde{c} \quad &\Leftrightarrow \quad d\tilde{c} = \frac{1}{\cos^2\theta} d\theta
\end{aligned}$$

and changing the limits of integration such that $\theta = \tan^{-1}(\infty) = \frac{\pi}{2}$ and $\theta = \tan^{-1}(-\infty) = -\frac{\pi}{2}$, integral ⓐ becomes

$$\begin{aligned}
ⓐ &= \int_{-\pi/2}^{\pi/2} \tan\theta(\tan\theta - \tilde{v})^{n-1} \frac{e^{-\nu\theta}}{[1+\tan^2\theta]^m} \frac{1}{\cos^2\theta} d\theta \qquad \text{(A.23)} \\
&= \int_{-\pi/2}^{\pi/2} \frac{\sin\theta}{\cos\theta} \frac{(\sin\theta - \tilde{v})^{n-1}}{\cos^{n-1}\theta} \frac{\cos^{2(m-1)}\theta}{e^{\nu\theta}} d\theta \\
&= \int_{-\pi/2}^{\pi/2} \sin\theta \cos^{r-n}\theta \frac{(\sin\theta - \tilde{v}\cos\theta)^{n-1}}{e^{\nu\theta}} d\theta,
\end{aligned}$$

with $r = 2(m-1)$ and integrating by parts $\int \mathbf{u}\, d\mathbf{v} = \oint \mathbf{u}\mathbf{v} - \int \mathbf{v}\, d\mathbf{u}$,

$$\begin{aligned}
\mathbf{u} &= e^{-\nu\theta} \qquad \text{(A.24)} \\
d\mathbf{v} &= \sin(\theta)\cos^{r-n}(\theta) d\theta.
\end{aligned}$$

To find the differential element $d\mathbf{u}$ it is necessary to use the product rule:

$$\begin{aligned}
(f\,g)' &= f'g + g'f = du, \qquad \text{(A.25)} \\
f &= (\sin\theta - \tilde{v}\cos\theta)^{n-1}, \\
f' &= (n-1)(\sin\theta - \tilde{v}\cos\theta)^{n-1}(\sin\theta + \tilde{v}\cos\theta), \\
g &= e^{-\nu\theta}, \\
g' &= -\nu e^{-\nu\theta}.
\end{aligned}$$

Substituting gives the remaining terms for integration by parts

$$\begin{aligned} d\mathbf{u} &= \Big[(n-1)(\sin\theta - \tilde{v}\cos\theta)^{n-1}(\sin\theta + \tilde{v}\cos\theta)e^{-\nu\theta} \\ &\qquad - \nu e^{-\nu\theta}(\sin\theta - \tilde{v}\cos\theta)^{n-1}\Big]d\theta, \\ \mathbf{v} &= \int \sin(\theta)\cos^{r-n}(\theta)d\theta. \end{aligned} \tag{A.26}$$

Integrating $\mathbf{v}$ requires substitution, with

$$\mathbf{u} = \cos^{r-n+1}(\theta) \quad \Leftrightarrow \quad d\mathbf{u} = -\sin\theta d\theta \tag{A.27}$$

giving

$$\begin{aligned} \mathbf{v} &= -\int u^{r-n}du = -\frac{u^{r-n+1}}{r-n+1}. \\ \Rightarrow \mathbf{v} &= \frac{\cos^{r-n+1}(\theta)}{n-r-1}. \end{aligned} \tag{A.28}$$

Substituting and continuing from before,

$$ⓒ = \int \frac{\cos^{r-n+1}(\theta)}{n-r-1}\left[(n-1)(\sin\theta - \tilde{v}\cos\theta)^{n-1}(\cos\theta + \tilde{v}\sin\theta)e^{-\nu\theta}\right]d\theta, \tag{A.29}$$

$$ⓓ = \int \frac{\cos^{r-n+1}(\theta)}{n-r-1}\left[\nu e^{-\nu\theta}(\sin\theta - \tilde{v}\cos\theta)^{n-1}\right]d\theta, \tag{A.30}$$

$$ⓐ = \frac{(\sin\theta - \tilde{v}\cos\theta)^{n-1}}{e^{\nu\theta}}\frac{\cos^{r-n+1}(\theta)}{n-r-1}\Bigg|_{-\frac{\pi}{2}}^{\frac{\pi}{2}} - ⓒ + ⓓ, \tag{A.31}$$

$$ⓓ = \nu\int_{-\pi/2}^{\pi/2}\frac{\cos^{r-n+1}(\theta)}{n-r-1}e^{-\nu\theta}\cos^{n-1}\theta(\tan\theta - \tilde{v})^{n-1}d\theta. \tag{A.32}$$

Using the substitutions

$$\begin{aligned} \theta = \tan^{-1}(\tilde{c}) &\Leftrightarrow \tilde{c} = \tan\theta \\ d\theta = \frac{1}{1+\tilde{c}^2}d\tilde{c} &\Leftrightarrow d\tilde{c} = \frac{1}{\cos^2\theta}d\theta \\ \cos^n &= \frac{1}{(1+\tilde{c}^2)^{n/2}} \end{aligned} \tag{A.33}$$

$$\begin{aligned} ⓓ &= \frac{\nu}{n-r-1}\int_{-\infty}^{\infty}\frac{e^{-\nu\tan^{-1}(\tilde{c})}(\tilde{c}-\tilde{v})^{n-1}}{(1+\tilde{c}^2)^m}d\theta \\ &= \frac{\nu}{n-r-1}\frac{\alpha\mu_{n-1}}{a^n}, \end{aligned} \tag{A.34}$$

$$\begin{aligned} ⓒ &= \int_{-\pi/2}^{\pi/2}\frac{\cos^{r-n+1}(\theta)}{n-r-1}(n-1)(\sin\theta-\tilde{v}\cos\theta)^{n-1}(\cos\theta+\tilde{v}\sin\theta)e^{-\nu\theta}d\theta \\ &= ⓔ + ⓕ, \end{aligned} \tag{A.35}$$

$$\begin{aligned} ⓔ &= \int_{-\pi/2}^{\pi/2}\frac{\cos^{r-n+1}(\theta)}{n-r-1}(n-1)(\sin\theta-\tilde{v}\cos\theta)^{n-1}\cos\theta e^{-\nu\theta}d\theta \\ &= \int_{-\pi/2}^{\pi/2}\frac{(n-1)}{n-r-1}\cos^r(\theta)(\tan\theta-\tilde{v})^{n-2}e^{-\nu\theta}d\theta, \end{aligned} \tag{A.36}$$

$$\begin{aligned} ⓕ &= \int_{-\pi/2}^{\pi/2}\frac{\cos^{r-n+1}(\theta)}{n-r-1}(n-1)(\sin\theta-\tilde{v}\cos\theta)^{n-1}\tilde{v}\sin\theta e^{-\nu\theta}d\theta \\ &= \int_{-\pi/2}^{\pi/2}\frac{\tilde{v}(n-1)}{n-r-1}\tan\theta\cos^r(\theta)(\tan\theta-\tilde{v})^{n-2}e^{-\nu\theta}d\theta. \end{aligned} \tag{A.37}$$

Using,

$$\begin{aligned} \theta = \tan^{-1}(\tilde{c}) &\Leftrightarrow \tilde{c} = \tan\theta \\ d\theta = \frac{1}{1+\tilde{c}^2}d\tilde{c} &\Leftrightarrow d\tilde{c} = \frac{1}{\cos^2\theta}d\theta \\ \cos^n(\theta) &= \frac{1}{(1+\tilde{c}^2)^{n/2}} \end{aligned} \tag{A.38}$$

$$\begin{aligned}
\textcircled{e} &= \frac{(n-1)}{n-r-1}\int_{-\infty}^{\infty}(\tilde{c}-\tilde{v})^{n-2}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{(1+\tilde{c}^2)^m}d\tilde{c} = \frac{(n-1)}{n-r-1}\frac{\alpha\mu_{n-2}}{a^{n-1}}, && \text{(A.39)}\\
\textcircled{f} &= \frac{\tilde{v}(n-1)}{n-r-1}\int_{-\infty}^{\infty}\tilde{c}(\tilde{c}-\tilde{v})^{n-2}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{(1+\tilde{c}^2)^m}d\tilde{c} && \text{(A.40)}\\
&= \frac{\tilde{v}(n-1)}{n-r-1}\int_{-\infty}^{\infty}\left\{(\tilde{c}-\tilde{v})(\tilde{c}-\tilde{v})^{n-2}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{(1+\tilde{c}^2)^m}d\tilde{c}+\tilde{v}(\tilde{c}-\tilde{v})^{n-2}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{(1+\tilde{c}^2)^m}d\tilde{c}\right\}\\
&= \frac{\tilde{v}(n-1)}{n-r-1}\int_{-\infty}^{\infty}(\tilde{c}-\tilde{v})^{n-1}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{(1+\tilde{c}^2)^m}d\tilde{c}+\frac{\tilde{v}^2(n-1)}{n-r-1}\int_{-\infty}^{\infty}(\tilde{c}-\tilde{v})^{n-2}\frac{e^{-\nu\tan^{-1}(\tilde{c})}}{(1+\tilde{c}^2)^m}d\tilde{c}\\
\textcircled{f} &= \tilde{v}\frac{(n-1)}{n-r-1}\frac{\alpha\mu_{n-1}}{a^n}+\tilde{v}^2\frac{(n-1)}{n-r-1}\frac{\alpha\mu_{n-2}}{a^{n-1}}
\end{aligned}$$

Recombining,

$$\begin{aligned}
\alpha\mu_n &= a^{n+1}\left[-\left(\textcircled{e}+\textcircled{f}\right)+\textcircled{d}\right]-\textcircled{B} && \text{(A.41)}\\
\cancel{\alpha}\mu_n &= a^{n+1}\Bigg[-\left(\frac{(n-1)}{n-r-1}\frac{\cancel{\alpha}\mu_{n-2}}{a^{n-1}}+\tilde{v}\frac{(n-1)}{n-r-1}\frac{\cancel{\alpha}\mu_{n-1}}{a^n}+\tilde{v}^2\frac{(n-1)}{n-r-1}\frac{\cancel{\alpha}\mu_{n-2}}{a^{n-1}}\right)\\
&\qquad +\frac{\nu}{n-r-1}\frac{\cancel{\alpha}\mu_{n-1}}{a^n}\Bigg]-a\tilde{v}\cancel{\alpha}\mu_{n-1}\\
\mu_n &= \frac{a^n}{(r+1-n)}\left[(n-1)\left(\frac{\mu_{n-2}}{a^{n-2}}+\tilde{v}\frac{\mu_{n-1}}{a^{n-1}}+\tilde{v}^2\frac{\mu_{n-2}}{a^{n-2}}\right)-\nu\frac{\mu_{n-1}}{a^{n-1}}\right]-\tilde{v}a\mu_{n-1}. && \text{(A.42)}
\end{aligned}$$

With $\tilde{v} = -\frac{\nu}{2(m-1)} = -\frac{\nu}{r}$,

$$\begin{aligned}
\mu_n &= \frac{a^n}{(r+1-n)}\left[(n-1)\left(\frac{\mu_{n-2}}{a^{n-2}}-\frac{\nu}{r}\frac{\mu_{n-1}}{a^{n-1}}+\left(\frac{\nu}{r}\right)^2\frac{\mu_{n-2}}{a^{n-2}}\right)-\nu\frac{\mu_{n-1}}{a^{n-1}}\right] && \text{(A.43)}\\
&\qquad +\frac{\nu}{r}a\mu_{n-1}.
\end{aligned}$$

Simplifying yields the final recursion relationship for the general one dimensional moments

$$\mu_n = \frac{a\,(n-1)}{(r+1-n)}\left\{\left[1+\left(\frac{\nu}{r}\right)^2\right]a\mu_{n-2}-2\frac{\nu}{r}\mu_{n-1}\right\}. \qquad \text{(A.44)}$$

This relationship holds for the all moments with the initial starting conditions $\mu_0 = 1$ and $\mu_1 = 0$.

A.1.4 Using Mathematica to Generate Reduced Third and Fourth Moments

The computer algebra system Mathematica was used to carry out the Gaussian eliminations for these derivations. Sample code is included here for completeness.

- Solving for r, ν, λ, and a using the recursion relationship for the generalized moment equations: $\mu_n = \frac{a(n-1)}{r-(n-1)}\left(\left(1+\left(\frac{\nu}{r}\right)^2\right)a\mu_{n-2} - 2\frac{\nu}{r}\mu_{n-1}\right)$

`Remove["Global‘*"]`

$\mu_0 = 1$

$\mu_1 = 0$

$\mu_2 = \texttt{FullSimplify}\left[\frac{a(n-1)}{r-(n-1)}\left(\left(1+\left(\frac{\nu}{r}\right)^2\right)a\mu_{n-2} - 2\frac{\nu}{r}\mu_{n-1}\right)/.\{n \to 2\}\right]$
$\mu_3 = \texttt{FullSimplify}\left[\frac{a(n-1)}{r-(n-1)}\left(\left(1+\left(\frac{\nu}{r}\right)^2\right)a\mu_{n-2} - 2\frac{\nu}{r}\mu_{n-1}\right)/.\{n \to 3\}\right]$
$\mu_4 = \texttt{FullSimplify}\left[\frac{a(n-1)}{r-(n-1)}\left(\left(1+\left(\frac{\nu}{r}\right)^2\right)a\mu_{n-2} - 2\frac{\nu}{r}\mu_{n-1}\right)/.\{n \to 4\}\right]$
$\texttt{que} = \texttt{FullSimplify}\left[\frac{\mu_3}{\mu_2^{3/2}}\right]$
$\texttt{dee} = \texttt{FullSimplify}\left[\frac{\mu_4}{\mu_2^2}\right]$

- Use Gaussian elimination to solve for r

$\texttt{ans} = \texttt{Eliminate}\left[\theta == \mu_2\&\&q == \texttt{que}\&\&d == \texttt{dee}, \{a, \nu, \theta\}\right];$

$\texttt{Solve}[\texttt{ans}, r]$

- Use Gaussian elimination to solve for ν

$\texttt{ans} = \texttt{Eliminate}\left[\theta == \mu_2\&\&q == \texttt{que}\&\&d == \texttt{dee}, \{a, d, \theta\}\right];$

$\texttt{ans} = \texttt{FullSimplify}[\texttt{Solve}[\texttt{ans}, \nu]];$

$\texttt{ans} = \texttt{Part}[\texttt{ans}, 1]$`(* Only concerned with real values of `$\nu$`*)`

- Use Gaussian elimination to solve for a

$\texttt{ans} = \texttt{Eliminate}\left[\theta == \mu_2\&\&q == \texttt{que}\&\&d == \texttt{dee}, \{\nu, d\}\right];$

$\texttt{ans} = \texttt{FullSimplify}[\texttt{Solve}[\texttt{ans}, a]];$

$\texttt{ans} = \texttt{Part}[\texttt{ans}, 2]$ `(* only positive values of a *)`

A.2 In Three Dimensions

The non-normalized three dimensional distribution function then becomes,

$$f^{nn-3D} = \frac{e^{-\nu \tan^{-1}\left(n^T A^{-1}(c-\lambda)\right)}}{\left[1 + (c-\lambda)^T A^{-2} (c-\lambda)\right]^m}. \tag{A.45}$$

A.2.1 Zeroth Moment

Taking the zeroth, raw moment

$$\alpha\mu_0 = \iiint_{\mathbb{R}^3} \frac{e^{-\nu \tan^{-1}\left(n^T A^{-1}(c-\lambda)\right)}}{\left[1 + (c-\lambda)^T A^{-2} (c-\lambda)\right]^m} d\vec{c}, \tag{A.46}$$

and letting

$$\begin{aligned} \tilde{c} = A^{-1}(\vec{c} - \vec{\lambda}) \quad &\Leftrightarrow \quad \vec{c} = A\tilde{c} + \lambda \\ d\vec{c} \quad &= \quad d(A\tilde{c}) \\ &= \quad \det(A) d\tilde{c}, \end{aligned} \tag{A.47}$$

and substituting yields

$$\alpha\mu_0 = \det(A) \iiint_{\mathbb{R}^3} \frac{e^{-\nu \tan^{-1}\left(n^T \tilde{c}\right)}}{\left[1 + \tilde{c}^T \tilde{c}\right]^m} d\tilde{c}. \tag{A.48}$$

Using the definition that $\mu_0 = 1$, the normalization constant in three dimensions (K) is defined as

$$\begin{aligned} K &\equiv \iiint \frac{e^{-\nu \tan^{-1}\left(n^T \tilde{c}\right)}}{\left[1 + \tilde{c}^T \tilde{c}\right]^m} d\tilde{c} \\ &= \frac{1}{\det(A)} \iiint_{\mathbb{R}^3} \frac{e^{-\nu \tan^{-1}\left(n^T A^{-1}(c-\lambda)\right)}}{\left[1 + (c-\lambda)^T A^{-2} (c-\lambda)\right]^m} dc. \end{aligned} \tag{A.49}$$

The normalized distribution function is

$$f^{(P3)} = \frac{1}{\det(A)\,K} \frac{e^{-\nu \tan^{-1}\left(n^T A^{-1}(c-\lambda)\right)}}{\left[1 + (c-\lambda)^T A^{-2} (c-\lambda)\right]^m}. \tag{A.50}$$

A.2.2 Factorization of normalization constant K

The normalization constant can be factored into multiples of the one dimensional normalization constant by choosing a $\tilde{c}_1 = n^T c$, $\tilde{c}_{2,3}$ orthogonal to n, and $\tilde{c}^T \tilde{c} = \tilde{c}_1^2 + \tilde{c}_2^2 + \tilde{c}_3^2$. Applying the substitutions

$$\hat{c}_3 = \frac{\tilde{c}_3}{1 + \tilde{c}_1^2 \tilde{c}_2^2},\ \hat{c}_2 = \frac{\tilde{c}_2}{\sqrt{1 + \tilde{c}_1^2}},\ \hat{c}_1 = \tilde{c}_1, \tag{A.51}$$

$$\begin{aligned} K &= \iiint \frac{e^{-\nu \tan^{-1}\left(n^T \hat{c}_1\right)}}{[1 + \tilde{c}_1^2 + \tilde{c}_2^2 + \tilde{c}_3^2]^m} d\tilde{c} & \text{(A.52)} \\ &= \iiint e^{-\nu \tan^{-1}\left(n^T \hat{c}_1\right)} \frac{1}{(1 + \tilde{c}_1^2 + \tilde{c}_2^2)^m} \frac{1}{\left(1 + \frac{\tilde{c}_3^2}{1 + \tilde{c}_1^2 + \tilde{c}_2^2}\right)^m} d\tilde{c}_1 d\tilde{c}_2 d\tilde{c}_3 \\ &= \int_{\mathbb{R}} \frac{e^{-\nu \tan^{-1}\left(n^T \hat{c}_1\right)}}{(1 + \hat{c}_1^2)^{m-1}} d\hat{c}_1 \int_{\mathbb{R}} \frac{1}{(1 + \hat{c}_2^2)^{m-\frac{1}{2}}} d\hat{c}_2 \int_{\mathbb{R}} \frac{1}{(1 + \hat{c}_3^2)^m} d\hat{c}_3 \\ K &= K_1(m,\nu) K_2(m) K_3(m) & \text{(A.53)} \\ &= k(m-1,\nu) k(m - \frac{1}{2}, 0) k(m, 0), \end{aligned}$$

where lower case k represents normalization constant in one dimension. This step is important conceptually, as [2] explains that it allows the three-dimensional moment equations to be represented as extensions of the one-dimensional equations.

A.2.3 Velocity, First Raw Moment

$$\begin{aligned} v &= \iiint_{\mathbb{R}^3} c f^{(P3)} dc \\ &= \frac{1}{\det(A)\,K} \iiint_{\mathbb{R}^3} c \frac{e^{-\nu \tan^{-1}\left(n^T A^{-1}(c-\lambda)\right)}}{\left[1+(c-\lambda)^T A^{-2}(c-\lambda)\right]^m} dc \end{aligned} \tag{A.54}$$

and letting

$$\begin{aligned} \tilde{c} = A^{-1}(c-\lambda) &\Leftrightarrow c = A\tilde{c} + \lambda \\ dc &= d(A\tilde{c}) \\ &= \det(A) d\tilde{c} \end{aligned} \tag{A.55}$$

and substituting gives

$$\begin{aligned} v &= \frac{1}{\cancel{\det(A)}\,K} \iiint_{\mathbb{R}^3} (A\tilde{c}+\lambda) \frac{e^{-\nu \tan^{-1}\left(n^T \tilde{c}\right)}}{[1+\tilde{c}^T\tilde{c}]^m} \cancel{\det(A)} d\tilde{c} \\ &= \frac{A}{K} \iiint_{\mathbb{R}^3} \tilde{c} \frac{e^{-\nu \tan^{-1}\left(n^T \tilde{c}\right)}}{[1+\tilde{c}^T\tilde{c}]^m} d\tilde{c} + \frac{\lambda}{K} \iiint_{\mathbb{R}^3} \tilde{c} \frac{e^{-\nu \tan^{-1}\left(n^T \tilde{c}\right)}}{[1+\tilde{c}^T\tilde{c}]^m} d\tilde{c}. \end{aligned} \tag{A.56}$$

From [2], a coordinate frame is chosen such that $c_1 = n^T c$.

$$\begin{aligned} M_l &= \iiint_{\mathbb{R}^3} n_l \left(\tilde{c}_l - \tilde{v}_l\right) f^{(P3)} d\tilde{c} \\ &= n_l \iiint_{\mathbb{R}^3} \left(\tilde{c}_l - \tilde{v}_l\right) f^{(P3)} d\tilde{c} \\ &= \alpha n_l. \end{aligned} \tag{A.57}$$

$$\therefore \alpha\vec{n} = \iiint_{\mathbb{R}^3} \left(\tilde{c}_l - \tilde{v}_l\right) \frac{e^{-\nu \tan^{-1}\left(n^T \tilde{c}\right)}}{[1+\tilde{c}^T\tilde{c}]^m} d\tilde{c} \tag{A.58}$$

so,

$$\begin{aligned}
\alpha &= \frac{1}{K}\iiint n^T\tilde{c}\frac{e^{-\nu\tan^{-1}\left(n^T\tilde{c}\right)}}{[1+\tilde{c}^T\tilde{c}]^m}d\tilde{c} \qquad (A.59)\\
&= \frac{1}{K_1\cancel{K_2}\cancel{K_3}}\int_{\mathbb{R}}\hat{c}_1\frac{e^{-\nu\tan^{-1}\left(n^T\hat{c}_1\right)}}{\left(1+\hat{c}_1^2\right)^{m-1}}d\hat{c}_1\cancel{\int_{\mathbb{R}}\frac{1}{\left(1+\hat{c}_2^2\right)^{m-\frac{1}{2}}}d\hat{c}_2}\cancel{\int_{\mathbb{R}}\frac{1}{\left(1+\hat{c}_3^2\right)^{m}}d\hat{c}_3}\\
&= \frac{1}{K_1}\int_{-\infty}^{\infty}\hat{c}_1\frac{e^{-\nu\tan^{-1}\left(n^T\hat{c}_1\right)}}{\left(1+\hat{c}_1^2\right)^{m-1}}d\hat{c}_1
\end{aligned}$$

and letting

$$\begin{aligned}
x = \tan^{-1}(\hat{c}) \quad &\Leftrightarrow \quad \hat{c} = \tan x \qquad (A.60)\\
dx &= \frac{1}{1+\tilde{c}^2}d\hat{c}
\end{aligned}$$

and changing the limits of integration such that $x = \tan^{-1}(\infty) = \frac{\pi}{2}$ and $x = \tan^{-1}(-\infty) = -\frac{\pi}{2}$ gives

$$\begin{aligned}
\alpha &= \frac{1}{K_1}\int_{-\pi/2}^{\pi/2}\tan x\frac{e^{-\nu\tan^{-1}x}}{\left(1+\tan^2 x\right)^{m-1}}\frac{dx}{\cos^2 x} \qquad (A.61)\\
&= \frac{1}{K_1}\int_{-\pi/2}^{\pi/2}\frac{\sin x\cos^{2(m-1)-3}x}{e^{\nu x}}dx.
\end{aligned}$$

Integrating by parts $\int \mathbf{u}\,d\mathbf{v} = \oint \mathbf{uv} - \int \mathbf{v}\,d\mathbf{u}$, letting

$$\begin{aligned}
\mathbf{u} = e^{-\nu x} \quad &\text{and} \quad d\mathbf{v} = \sin(x)\cos^{2(m-2)}(x)dx \qquad (A.62)\\
d\mathbf{u} = -\nu e^{-\nu x} \quad &\qquad \mathbf{v} = -\frac{\cos^{2(m-2)}(x)}{2(m-2)},
\end{aligned}$$

and substituting gives

$$\begin{aligned}
\alpha &= \frac{1}{K_1}\left[\cancel{-\frac{\cos^{2(m-2)}(x)}{e^{\nu x}2(m-2)}\Bigg|_{-\frac{\pi}{2}}^{\frac{\pi}{2}}} + \frac{\nu}{2(m-2)}\int_{-\pi/2}^{\pi/2}\frac{\cos^{2(m-1)}(x)}{e^{\nu x}}dx\right] \qquad (A.63)\\
&= -\frac{\nu}{2(m-2)}\frac{1}{K_1}\int_{-\infty}^{\infty}\frac{e^{-\nu\tan^{-1}\left(n^T\hat{c}_1\right)}}{\left(1+\hat{c}_1^2\right)^{m-1}}d\hat{c}_1 = -\frac{\nu}{2(m-2)}.
\end{aligned}$$

Substituting gives

$$v = \lambda - \frac{\nu A\vec{n}}{2(m-2)}. \qquad (A.64)$$

With the simplification that $r = 2(m-2)$, the expression becomes

$$v = \lambda - \frac{\nu}{r}A\vec{n}. \qquad (A.65)$$

A.2.4 Second to Fourth Moments

Second Moment

Beginning with the second moment of the Pearson-IV in 3D,

$$M_{ij} = A_{ik}A_{jl}\iiint_{\mathbb{R}^3}(\tilde{c}_k-\tilde{\nu}_k)(\tilde{c}_l-\tilde{\nu}_l)f^{(P3)}d\tilde{c}, \qquad (A.66)$$

assume the equation is normalized and rotate the coordinate frame such that $c_1 = n^T\tilde{c}$

$$\begin{aligned}
\tilde{M}_{kl} &= \iiint_{\mathbb{R}^3} n_k n_l(\tilde{c}_k-\tilde{\nu}_k)(\tilde{c}_l-\tilde{\nu}_l)f^{(P3)}d\tilde{c} \qquad (A.67)\\
&= \iiint_{\mathbb{R}^3} n_k n_l\tilde{c}_k\tilde{c}_l f^{(P3)}d\tilde{c} - \cancel{2n_k n_l\tilde{\nu}_k\iiint_{\mathbb{R}^3}\tilde{c}_l f^{(P3)}d\tilde{c}} + \iiint_{\mathbb{R}^3} n_k n_l\tilde{\nu}_k\tilde{\nu}_l f^{(P3)}d\tilde{c}\\
&= \alpha n_k n_l + \beta\delta_{kl}.
\end{aligned}$$

Multiply both sides by $n_k n_l$,

$$\iiint_{\mathbb{R}^3} n_k n_l(\tilde{c}_k - \tilde{\nu}_k)(\tilde{c}_l - \tilde{\nu}_l) f^{(P3)} d\tilde{c} = \alpha n_k n_l n_k n_l + \beta n_k n_l \delta_{kl}. \tag{A.68}$$

Making the summation explicit, on the left side we have

$$\iiint_{\mathbb{R}^3} \sum_{k=1}^{3} \sum_{l=1}^{3} n_k n_l(\tilde{c}_k - \tilde{\nu}_k)(\tilde{c}_l - \tilde{\nu}_l) f^{(P3)} d\tilde{c} = \tag{A.69}$$

$$n_1^2\,(c_1 - v_1)^{\,2} + 2n_1 n_2\,(c_1 - v_1)\,(c_2 - v_2) + n_2^2\,(c_2 - v_2)^{\,2} + 2n_1 n_3\,(c_1 - v_1)\,(c_3 - v_3) \ldots$$

$$+2n_2 n_3\,(c_2 - v_2)\,(c_3 - v_3) + n_3^2\,(c_3 - v_3)^{\,2}.$$

Here, the coordinate system is rotated such that the skewness unit vector is purely in the direction of n_1 and consequently n_2 and n_3 are zero. The random velocities are chosen such that c_1 is in the direction of n_1 and that c_1, c_2 and c_3 are orthogonal to each other. Thus, the average velocity components ν_2 and ν_3 are zero.

$$\iiint_{\mathbb{R}^3} n_1^2(\tilde{c}_1 - \tilde{\nu}_1)^2 f^{(P3)} d\tilde{c}. \tag{A.70}$$

From Eq. 2.18 we have the generalization of moments to three dimensions:

$$\mu_n^{p,q} = \frac{1}{K} \iiint_{\mathbb{R}^3} (\tilde{c}_1 - \tilde{\nu}_1)^n \tilde{c}_2^p \tilde{c}_3^q \frac{\exp(-\nu \arctan(\tilde{c}_1))}{(1 + \tilde{c}_1^2 + \tilde{c}_2^2 + \tilde{c}_3^2)} d\tilde{c}. \tag{A.71}$$

From here it is easy to see the expression is identical to $\mu_2^{0,0}$. On the right hand side of the equation, with explicit summation, the resulting expression is

$$\alpha \sum_{k=1}^{3} \sum_{l=1}^{3} n_k n_l n_k n_l + \beta \sum_{k=1}^{3} \sum_{l=1}^{3} n_k n_l \delta_{kl} = \alpha \left(n_1^2 + n_2^2 + n_3^2\right)^{\,2} + \beta(n_1^2 + n_2^2 + n_3^2) = \alpha + \beta. \tag{A.72}$$

With n being a unit vector, combining the right and left hand sides yields

$$\mu_2^{0,0} = \alpha + \beta. \tag{A.73}$$

Multiplying the normalized second moment by δ_{kl} yields

$$\iiint_{\mathbb{R}^3} \sum_{k=1}^{3}\sum_{l=1}^{3} \delta_{kl}(\tilde{c}_k - \tilde{\nu}_k)(\tilde{c}_l - \tilde{\nu}_l) f^{(P3)} d\tilde{c} = \alpha \sum_{k=1}^{3}\sum_{l=1}^{3} \delta_{kl} n_k n_l + \beta \sum_{k=1}^{3}\sum_{l=1}^{3} \delta_{kl}\delta_{kl}, \tag{A.74}$$

with explicit summation. Evaluating yields

$$\iiint_{\mathbb{R}^3} \left[(\tilde{c}_1 - \tilde{v}_1)^2 + (\tilde{c}_2 - \tilde{v}_2)^2 + (\tilde{c}_3 - \tilde{v}_3)^2 \right] f^{(P3)} d\tilde{c} = \alpha(n_1^2 + n_2^2 + n_3^2) + \beta(3). \tag{A.75}$$

It is easy to see that this can be expressed in terms of the generalized moments in three dimensions

$$\mu_2^{0,0} + \mu_0^{2,0} + \mu_0^{0,2} = \alpha + 3\beta. \tag{A.76}$$

Further, $\mu_2^{0,0} = \mu_0^{2,0}$, so that

$$3\mu_2^{0,0} = \alpha + 3\beta. \tag{A.77}$$

Solving for α and β yields $\alpha = 0$ and $\beta = \mu_2^{0,0}$. Substituting into the second moment gives

$$M_{ij} = \Theta_{ij} = \mu_2^{0,0} A^2 = \frac{1 + \left(\frac{\nu}{r}\right)^2}{r - 1} A^2. \tag{A.78}$$

Third Moment

Beginning with the third moment of the Pearson-IV in 3D,

$$M_{ijk} = A_{il} A_{jp} A_{kq} \iiint_{\mathbb{R}^3} (\tilde{c}_l - \tilde{\nu}_l)(\tilde{c}_p - \tilde{\nu}_p)(\tilde{c}_q - \tilde{\nu}_q) f^{(P3)} d\tilde{c}, \tag{A.79}$$

assume the equation is normalized and rotate the coordinate frame such that $c_1 = n^T\tilde{c}$,

$$\begin{aligned}\tilde{M}_{lpq} &= \iiint_{\mathbb{R}^3} n_l n_p n_q (\tilde{c}_l - \tilde{\nu}_l)(\tilde{c}_p - \tilde{\nu}_p)(\tilde{c}_q - \tilde{\nu}_q) f^{(P3)} d\tilde{c} \qquad (A.80)\\ &= \iiint_{\mathbb{R}^3} n_l n_p n_q \left[\tilde{c}_l\tilde{c}_p\tilde{c}_q - \tilde{v}_l\tilde{v}_p\tilde{v}_q\right] f^{(P3)} d\tilde{c} - \iiint_{\mathbb{R}^3} n_{(l} n_p n_q \tilde{v}_l \tilde{c}_p \tilde{c}_{q)} f^{(P3)} d\tilde{c}\\ &= \alpha\, n_l n_p n_q + \beta\, n_{(l}\delta_{pq)}.\end{aligned}$$

For convenience, $n_{(l}\delta_{pq)}$ uses the same notation as the symmetrization of a tensor [7, 8],

$$T_{(i_1 i_2 \ldots i_r)} = \frac{1}{r!} \sum_{Permutations} T_{(i_1 i_2 \ldots i_r, i_r \ldots i_2 i_1, etc\ldots)}, \qquad (A.81)$$

where the i's represent the indices of a tensor up to order r. This notation allows the ordered permutations of terms to be grouped together, as can be seen in the expansion of

$$3 n_{(l}\delta_{pq)} = \left(n_l\delta_{pq} + n_q\delta_{lp} + n_p\delta_{ql}\right). \qquad (A.82)$$

After making the summation explicit, multiplying both sides by $n_l n_p n_q$ yields a left hand side equal to

$$\iiint_{\mathbb{R}^3} \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} n_l n_p n_q (\tilde{c}_l - \tilde{\nu}_l)(\tilde{c}_p - \tilde{\nu}_p)(\tilde{c}_q - \tilde{\nu}_q) f^{(P3)} d\tilde{c}. \qquad (A.83)$$

Here, the coordinate system is rotated such that the skewness unit vector is purely in the direction of n_1 and consequently n_2 and n_3 are zero. The random velocities are chosen such that c_1 is in the direction of n_1 and that c_1, c_2 and c_3 are orthogonal to each other. Thus, the average velocity components ν_2 and ν_3 are zero. From the generalized moment equation in three dimensions,

$$\mu_n^{p,q} = \frac{1}{K}\iiint_{\mathbb{R}^3} (\tilde{c}_1 - \tilde{\nu}_1)^n \tilde{c}_2^p \tilde{c}_3^q \frac{\exp(-\nu\arctan(\tilde{c}_1))}{(1+\tilde{c}_1^2+\tilde{c}_2^2+\tilde{c}_3^2)} d\tilde{c}, \qquad (A.84)$$

it is obvious that, with $n_1 = 1$,

$$\iiint_{\mathbb{R}^3} n_1^3(\tilde{c_1} - \tilde{\nu_1})^3 f^{(P3)} d\tilde{c} = \mu_3^{0,0}. \tag{A.85}$$

The right hand side of the equation with explicit summation is equal to

$$\alpha \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} n_l n_p n_q n_l n_p n_q + \beta \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} n_l n_p n_q n_{(l}\delta_{pq)} \tag{A.86}$$

$$= \alpha \left(n_1^2 + n_2^2 + n_3^2\right)^3 + \beta \left(n_1^2 + n_2^2 + n_3^2\right)^2.$$

The combined equation becomes

$$\mu_3^{0,0} = \alpha + \beta. \tag{A.87}$$

Multiplying the normalized third moment by $n_l\delta_{pq}$ yields a left hand side

$$\iiint_{\mathbb{R}^3} \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} n_l\delta_{pq}(\tilde{c_l} - \tilde{\nu_l})(\tilde{c_p} - \tilde{\nu_p})(\tilde{c_q} - \tilde{\nu_q}) f^{(P3)} d\tilde{c}. \tag{A.88}$$

As before, setting ν_2, ν_3, n_2 and n_3 to zero,

$$\iiint_{\mathbb{R}^3} [c_2^2 n_1 \left(c_1 - v_1\right) + c_3^2 n_1 \left(c_1 - v_1\right) + n_1 \left(c_1 - v_1\right)^3] f^{(P3)} d\tilde{c}. \tag{A.89}$$

it is easy to see from the general moment equation that this is equivalent to

$$\mu_3^{0,0} + \mu_1^{2,0} + \mu_1^{0,2}. \tag{A.90}$$

The right hand side with explicit summation is equal to

$$\alpha \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} n_l\delta_{pq} n_l n_p n_q + \beta \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3} n_l\delta_{pq} n_{(l}\delta_{pq)} \tag{A.91}$$

$$= \alpha \left(n_1^2 + n_2^2 + n_3^2\right)^{2} + \frac{5}{3}\beta(n_1^2 + n_2^2 + n_3^2).$$

Combining these equations yields

$$\mu_3^{0,0} + \mu_1^{2,0} + \mu_1^{0,2} = \alpha + \frac{5}{3}\beta. \tag{A.92}$$

To solve for the constants α and β, it is necessary to relate the moments $\mu_3^{0,0}$ and $\mu_1^{2,0}$ using the recursion relationship

$$\tilde{\mu}_n = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{\nu}{r}\right)^{2}\right)\tilde{\mu}_{n-2}^{p,q} - (2(n-1)+p+q)\frac{\nu}{r}\tilde{\mu}_{n-1}^{p,q}\right), \tag{A.93}$$

where

$$\tilde{\mu}_{-1}^{p,q} = 0, \tag{A.94}$$

$$\tilde{\mu}_0^{p,q} = (p-1)!!(q-1)!!\prod_{k=1}^{\frac{p+q}{2}} \frac{1+\frac{\nu^2}{(2-2k+r)^2}}{1-2k+r}. \tag{A.95}$$

After evaluating, the moments are found to be

$$\mu_1^{2,0} = -\frac{2v\left(r^2+v^2\right)}{(-2+r)(-1+r)r^3} \tag{A.96}$$

and

$$\mu_3^{0,0} = -\frac{4v\left(1+\frac{v^2}{r^2}\right)}{(-2+r)(-1+r)r}. \tag{A.97}$$

It is easy to see from these expressions that $\mu_1^{2,0} = \mu_1^{0,2} = \frac{1}{2}\mu_3^{0,0}$. Using these relationships, the system of equations for the third moment becomes

$$\mu_3^{0,0} = \alpha + \beta, \tag{A.98}$$

$$2\mu_3^{0,0} = \alpha + \frac{5}{3}\beta. \tag{A.99}$$

Solving these equations for α and β gives $\alpha = -\frac{1}{2}\mu_3^{0,0}$ and $\beta = \frac{3}{2}\mu_3^{0,0}$. Substituting back in, the third moment is

$$M_{ijk} = \mu_3^{0,0} A_{il} A_{jp} A_{kq} \left(-\frac{1}{2} n_l n_p n_q + \frac{3}{2} n_{(l} \delta_{pq)} \right). \tag{A.100}$$

Fourth Moment

Beginning with the fourth moment of the Pearson-IV in 3D,

$$M_{ijkl} = A_{il} A_{jp} A_{kq} A_{ls} \iiint_{\mathbb{R}^3} (\tilde{c_l} - \tilde{v_l})(\tilde{c_p} - \tilde{v_p})(\tilde{c_q} - \tilde{v_q})(\tilde{c_s} - \tilde{v_s}) f^{(P3)} d\tilde{c}, \tag{A.101}$$

assume the equation is normalized and rotate the coordinate frame such that $c_1 = n^T \tilde{c}$,

$$\begin{aligned}
\tilde{M}_{ijkl} &= \iiint_{\mathbb{R}^3} (\tilde{c_l} - \tilde{v_l})(\tilde{c_p} - \tilde{v_p})(\tilde{c_q} - \tilde{v_q})(\tilde{c_s} - \tilde{v_s}) f^{(P3)} d\tilde{c} \qquad \text{(A.102)} \\
&= \iiint_{\mathbb{R}^3} n_l n_p n_q n_s (\tilde{c_l} - \tilde{v_l})(\tilde{c_p} - \tilde{v_p})(\tilde{c_q} - \tilde{v_q})(\tilde{c_s} - \tilde{v_s}) f^{(P3)} d\tilde{c} \\
&= \iiint_{\mathbb{R}^3} n_l n_p n_q n_s \left[\tilde{c}_l \tilde{c}_p \tilde{c}_q \tilde{c}_s + \tilde{v}_l \tilde{v}_p \tilde{v}_q \tilde{v}_s \right] f^{(P3)} d\tilde{c} \\
&\quad + \iiint_{\mathbb{R}^3} n_{(l} n_p n_q n_s \tilde{c}_l \tilde{c}_p \tilde{c}_q \tilde{v}_{s)} f^{(P3)} d\tilde{c} - \iiint_{\mathbb{R}^3} n_{(l} n_p n_q n_s \tilde{c}_l \tilde{c}_p \tilde{v}_q \tilde{v}_{s)} f^{(P3)} d\tilde{c} \\
&= \alpha\, n_l n_p n_q n_s + \beta\, n_{(l} n_p \delta_{qs)} + \gamma\, \delta_{(lp} \delta_{qs)}.
\end{aligned}$$

Again for convenience, $n_{(l} n_p \delta_{qs)}$ and $\delta_{(lp} \delta_{qs)}$ use the same notation as the symmetrization of a tensor

$$T_{(i_1 i_2 \ldots i_r)} = \frac{1}{r!} \sum_{Permutations} T_{(i_1 i_2 \ldots i_r, i_r \ldots i_2 i_1, etc \ldots)}. \tag{A.103}$$

After multiplying both sides of the moment equation by $n_l n_p n_q n_s$ and making the summation explicit, the left hand side is equal to

$$\sum_{l=1}^{3} \sum_{p=1}^{3} \sum_{q=1}^{3} \sum_{s=1}^{3} n_l n_p n_q n_s \left(c_l - v_l \right) \left(c_p - v_p \right) \left(c_q - v_q \right) \left(c_s - v_s \right) f^{(P3)} d\tilde{c}. \tag{A.104}$$

Here, the coordinate system is rotated such that the skewness unit vector is purely in the direction of n_1 and consequently n_2 and n_3 are zero. The random velocities are chosen such that c_1 is in the direction of n_1 and that c_1, c_2 and c_3 are orthogonal to each other. Thus, the average velocity components ν_2 and ν_3 are zero. From the generalized moment equation in three dimensions,

$$\mu_n^{p,q} = \frac{1}{K} \iiint_{\mathbb{R}^3} (\tilde{c}_1 - \tilde{\nu}_1)^n \tilde{c}_2^p \tilde{c}_3^q \frac{\exp(-\nu \arctan(\tilde{c}_1))}{(1 + \tilde{c}_1^2 + \tilde{c}_2^2 + \tilde{c}_3^2)} d\tilde{c}, \tag{A.105}$$

it can be seen that

$$\iiint_{\mathbb{R}^3} n_1^4 (\tilde{c}_1 - \tilde{\nu}_1)^4 f^{(P3)} d\tilde{c} = \mu_4^{0,0}. \tag{A.106}$$

The right hand side of the equation with explicit summation is equal to

$$\begin{aligned} &= \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p n_q n_s \alpha n_l n_p n_q n_s \qquad \text{(A.107)}\\ &\quad + \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p n_q n_s \beta n_{(l} n_p \delta_{qs)} \\ &\quad + \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p n_q n_s \gamma \delta_{(lp} \delta_{qs)} \\ &= \alpha + \beta + \gamma. \qquad \text{(A.108)} \end{aligned}$$

The combined equation becomes

$$\mu_4^{0,0} = \alpha + \beta + \gamma. \tag{A.109}$$

Multiplying the normalized fourth moment by $n_l n_p \delta_{qs}$ yields a left hand side

$$\sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p \delta_{qs} \left(c_l - v_l\right)\left(c_p - v_p\right)\left(c_q - v_q\right)\left(c_s - v_s\right) f^{(P3)} d\tilde{c}. \tag{A.110}$$

As before, setting ν_2, ν_3, n_2 and n_3 to zero yields

$$\iiint_{\mathbb{R}^3} \left[c_2^2 \left(c_1 - v_1\right)^2 + c_3^2 \left(c_1 - v_1\right)^2 + \left(c_1 - v_1\right)^4\right] f^{(P3)} d\tilde{c}. \tag{A.111}$$

From the generalized moment equation, this is equivalent to

$$\mu_2^{2,0} + \mu_2^{0,2} + \mu_4^{0,0}. \tag{A.112}$$

The right hand side with explicit summation is equal to

$$\begin{aligned}
&= \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p \delta_{qs} \alpha n_l n_p n_q n_s && \text{(A.113)}\\
&\quad + \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p \delta_{qs} \beta n_{(l} n_p \delta_{qs)} && \text{(A.114)}\\
&\quad + \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l n_p \delta_{qs} \gamma \delta_{(lp} \delta_{qs)} \\
&= \alpha + \frac{4\beta}{3} + \frac{5\gamma}{3}.
\end{aligned}$$

Combining these equations leads to

$$\mu_2^{2,0} + \mu_2^{0,2} + \mu_4^{0,0} = \alpha + \frac{4\beta}{3} + \frac{5\gamma}{3}. \tag{A.115}$$

Multiplying the equation by $\delta_{(lp}\delta_{qs)}$ yields a left hand side equal to

$$\sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} \delta_{(lp}\delta_{qs)} \left(c_l - v_l\right)\left(c_p - v_p\right)\left(c_q - v_q\right)\left(c_s - v_s\right) f^{(P3)} d\tilde{c}. \tag{A.116}$$

As before, setting ν_2, ν_3, n_2 and n_3 to zero

$$\sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}\left[c_2^4+2c_2^2c_3^2+c_3^4+2c_2^2\left(c_1-v_1\right)^2+2c_3^2\left(c_1-v_1\right)^2+\left(c_1-v_1\right)^4\right]f^{(P3)}d\tilde{c}. \tag{A.117}$$

Simplifying from the expression for general moments gives

$$\mu_0^{4,0}+2\mu_0^{2,2}+\mu_0^{0,4}+2\mu_2^{2,0}+2\mu_2^{0,2}+\mu_4^{0,0}. \tag{A.118}$$

The right hand side with explicit summation is equal to

$$\begin{aligned} &= \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}\delta_{lp}\delta_{qs}\alpha n_l n_p n_q n_s+\sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}\delta_{lp}\delta_{qs}\beta n_{(l}n_p\delta_{qs)} \qquad \text{(A.119)}\\ &\quad +\sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}\delta_{lp}\delta_{qs}\gamma\delta_{(lp}\delta_{qs)}\\ &= \alpha+\frac{5\beta}{3}+5\gamma. \end{aligned}$$

Combining these equations gives

$$\mu_0^{4,0}+2\mu_0^{2,2}+\mu_0^{0,4}+2\mu_2^{2,0}+2\mu_2^{0,2}+\mu_4^{0,0}=\alpha+\frac{5\beta}{3}+5\gamma. \tag{A.120}$$

To solve for the unknowns it is first necessary to relate as many moments as possible using the recursion relationship

$$\tilde{\mu}_n=\frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{\nu}{r}\right)^2\right)\tilde{\mu}_{n-2}^{p,q}-(2(n-1)+p+q)\frac{\nu}{r}\tilde{\mu}_{n-1}^{p,q}\right), \tag{A.121}$$

where

$$\tilde{\mu}_{-1}^{p,q}=0, \tag{A.122}$$

$$\tilde{\mu}_0^{p,q}=(p-1)!!(q-1)!!\prod_{k=1}^{\frac{p+q}{2}}\frac{1+\frac{\nu^2}{(2-2k+r)^2}}{1-2k+r}. \tag{A.123}$$

Evaluation shows the moments to be

$$\begin{aligned}
\mu_4^{0,0} &= \frac{3\left(r^2+v^2\right)\left((-2+r)r^2+(6+r)v^2\right)}{(-3+r)(-2+r)(-1+r)r^4}, && \text{(A.124)}\\
\mu_2^{0,2}=\mu_2^{2,0} &= \frac{(-2+r)r^4+2r^2(2+r)v^2+(6+r)v^4}{(-3+r)(-2+r)(-1+r)r^4}, && \text{(A.125)}\\
\mu_0^{2,2} &= \frac{\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)r^2}, && \text{(A.126)}\\
\mu_4^{4,0} &= \frac{3\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)r^2}. && \text{(A.127)}
\end{aligned}$$

The relationship between moments is therefore shown to be

$$\begin{aligned}
\mu_4^{0,0} &= 3\mu_2^{0,2}=3\mu_2^{2,0}, && \text{(A.128)}\\
\mu_4^{4,0} &= \mu_4^{0,4}=3\mu_0^{2,2}. && \text{(A.129)}
\end{aligned}$$

Using these expressions, the full system of equations for the fourth moment is shown to be

$$\begin{aligned}
\mu_4^{0,0} &= \alpha+\beta+\gamma, && \text{(A.130)}\\
\frac{1}{3}\mu_4^{0,0}+\frac{1}{3}\mu_4^{0,0}+\mu_4^{0,0} &= \alpha+\frac{4\beta}{3}+\frac{5\gamma}{3}, && \text{(A.131)}\\
\mu_0^{4,0}+2\frac{1}{3}\mu_4^{4,0}+\mu_0^{0,4}+2\frac{1}{3}\mu_4^{0,0}+2\frac{1}{3}\mu_4^{0,0}+\mu_4^{0,0} &= \alpha+\frac{5\beta}{3}+5\gamma. && \text{(A.132)}
\end{aligned}$$

Solving these equations for the constants α, β, and γ yields

$$\begin{aligned}
\alpha &= \mu_0^{4,0}-\mu_4^{0,0}, && \text{(A.133)}\\
\beta &= -2\left(\mu_0^{4,0}-\mu_4^{0,0}\right), && \text{(A.134)}\\
\gamma &= \mu_0^{4,0}. && \text{(A.135)}
\end{aligned}$$

Substituting back into the original moment equation yields

$$M_{ijkl} = A_{il}A_{jp}A_{kq}A_{ls}\left[\left(\mu_0^{4,0} - \mu_4^{0,0}\right) n_l n_p n_q n_s - 2\left(\mu_0^{4,0} - \mu_4^{0,0}\right) n_{(l} n_p \delta_{qs)} + \mu_0^{4,0}\delta_{(lp}\delta_{qs)}\right]. \tag{A.136}$$

A.2.5 Using Mathematica to Solve the 4th Moment of the 3-D Pearson-IV function

Mathematica is used to carry through the indicial notation for this derivation. An example for the source code used in the derivation of the fourth moment is presented here for completeness.

" 4th Moment of the Pearson-IV 3D Equation";

"Starting with the equation:";

"$M_{lpqs}=\alpha\ n_l n_p n_q n_s\ +\ \beta\ n_{(l} n_p \delta_{qs)}\ +\ \gamma\ \delta_{(lp}\delta_{qs)}$";

```
" ********************************************* ";
" Build Symmetric Part of 4th Rank Tensor ";
"All Possible Combinations:";
P = Permutations[{l, p, q, s}];
Symm1 = 0; (* n_(l n_p δ_qs) *)
For[index = 1, index < 25,
  {
    i = P[[index, 1]];
    j = P[[index, 2]];
    k = P[[index, 3]];
    h = P[[index, 4]];
    dummy = n_i n_h KroneckerDelta[j, k];
    Symm1 = Symm1 + dummy;
   } index++];
Symm1 = Symm1 / (4 !); (* n_(l n_p δ_qs) *)


Symm2 = 0; (* δ_(lp δ_qs) *)
For[index = 1, index < 25,
  {
    i = P[[index, 1]];
    j = P[[index, 2]];
    k = P[[index, 3]];
    h = P[[index, 4]];
    dummy = KroneckerDelta[i, h] KroneckerDelta[j, k];
    Symm2 = Symm2 + dummy;
   } index++];
Symm2 = Symm2 / (4 !); (* δ_(lp δ_qs) *)
" ********************************************* ";
```

"Multiply both sides by $n_l n_p n_q n_s$";

" NOTE: This is inside the integrand ";

"Explicit Summation";

$$A\ =\ \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l\, n_p\, n_q\, n_s\,(c_l - v_l)\left(c_p - v_p\right)\left(c_q - v_q\right)(c_s - v_s)$$

$$n_1^4\,(c_1-v_1)^4 + 4\,n_1^3\,n_2\,(c_1-v_1)^3\,(c_2-v_2) + 6\,n_1^2\,n_2^2\,(c_1-v_1)^2\,(c_2-v_2)^2 + 4\,n_1\,n_2^3\,(c_1-v_1)\,(c_2-v_2)^3 + n_2^4\,(c_2-v_2)^4 + \\ 4\,n_1^3\,n_3\,(c_1-v_1)^3\,(c_3-v_3) + 12\,n_1^2\,n_2\,n_3\,(c_1-v_1)^2\,(c_2-v_2)\,(c_3-v_3) + 12\,n_1\,n_2^2\,n_3\,(c_1-v_1)\,(c_2-v_2)^2\,(c_3-v_3) + \\ 4\,n_2^3\,n_3\,(c_2-v_2)^3\,(c_3-v_3) + 6\,n_1^2\,n_3^2\,(c_1-v_1)^2\,(c_3-v_3)^2 + 12\,n_1\,n_2\,n_3^2\,(c_1-v_1)\,(c_2-v_2)\,(c_3-v_3)^2 + \\ 6\,n_2^2\,n_3^2\,(c_2-v_2)^2\,(c_3-v_3)^2 + 4\,n_1\,n_3^3\,(c_1-v_1)\,(c_3-v_3)^3 + 4\,n_2\,n_3^3\,(c_2-v_2)\,(c_3-v_3)^3 + n_3^4\,(c_3-v_3)^4$$

"Simplification Step 1: ν_2 and ν_3 are both zero.";

$B\ =\ A\ /.\ \{v_2 \to 0,\ v_3 \to 0\}$

$$c_2^4\,n_2^4 + 4\,c_2^3\,c_3\,n_2^3\,n_3 + 6\,c_2^2\,c_3^2\,n_2^2\,n_3^2 + 4\,c_2\,c_3^3\,n_2\,n_3^3 + c_3^4\,n_3^4 + 4\,c_2^3\,n_1\,n_2^3\,(c_1-v_1) + \\ 12\,c_2^2\,c_3\,n_1\,n_2^2\,n_3\,(c_1-v_1) + 12\,c_2\,c_3^2\,n_1\,n_2\,n_3^2\,(c_1-v_1) + 4\,c_3^3\,n_1\,n_3^3\,(c_1-v_1) + 6\,c_2^2\,n_1^2\,n_2^2\,(c_1-v_1)^2 + \\ 12\,c_2\,c_3\,n_1^2\,n_2\,n_3\,(c_1-v_1)^2 + 6\,c_3^2\,n_1^2\,n_3^2\,(c_1-v_1)^2 + 4\,c_2\,n_1^3\,n_2\,(c_1-v_1)^3 + 4\,c_3\,n_1^3\,n_3\,(c_1-v_1)^3 + n_1^4\,(c_1-v_1)^4$$

"Simplification Step 2: n_2 and n_3 are both zero.";

$F\ =\ B\ /.\ \{n_2 \to 0,\ n_3 \to 0\}$

$$n_1^4\,(c_1-v_1)^4$$

"From this it is clear that the left side of the eq. is $\mu_4^{0,0}$";

"$n_l n_p n_q n_s \alpha\ n_l n_p n_q n_s$ Explicit Summation:";

$$A\ =\ \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_l\, n_p\, n_q\, n_s\, \alpha\, n_l\, n_p\, n_q\, n_s$$

$$\alpha\,n_1^8 + 4\,\alpha\,n_1^6\,n_2^2 + 6\,\alpha\,n_1^4\,n_2^4 + 4\,\alpha\,n_1^2\,n_2^6 + \alpha\,n_2^8 + 4\,\alpha\,n_1^6\,n_3^2 + 12\,\alpha\,n_1^4\,n_2^2\,n_3^2 + \\ 12\,\alpha\,n_1^2\,n_2^4\,n_3^2 + 4\,\alpha\,n_2^6\,n_3^2 + 6\,\alpha\,n_1^4\,n_3^4 + 12\,\alpha\,n_1^2\,n_2^2\,n_3^4 + 6\,\alpha\,n_2^4\,n_3^4 + 4\,\alpha\,n_1^2\,n_3^6 + 4\,\alpha\,n_2^2\,n_3^6 + \alpha\,n_3^8$$

B = Factor[A]

$\alpha\left(n_1^2+n_2^2+n_3^2\right)^4$

B /. $\left\{n_1^2+n_2^2+n_3^2\to 1\right\}$ **(* n is a unit vector *)**

α

"$n_1 n_p n_q n_s \beta\ n_{(1} n_p \delta_{qs)}$ Explicit Summation:";

$$A = \sum_{1=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_1\, n_p\, n_q\, n_s\, \beta\ \mathbf{Symm1}$$

$$\beta n_1^6+2\beta n_1^4 n_2^2+2\beta n_1^2 n_2^4+\beta n_2^6+\frac{1}{4}\beta n_1^2 n_2^2\left(4 n_1^2+4 n_2^2\right)+2\beta n_1^4 n_3^2+6\beta n_1^2 n_2^2 n_3^2+$$
$$2\beta n_2^4 n_3^2+2\beta n_1^2 n_3^4+2\beta n_2^2 n_3^4+\beta n_3^6+\frac{1}{4}\beta n_1^2 n_3^2\left(4 n_1^2+4 n_3^2\right)+\frac{1}{4}\beta n_2^2 n_3^2\left(4 n_2^2+4 n_3^2\right)$$

B = Factor[A]

$\beta\left(n_1^2+n_2^2+n_3^2\right)^3$

B /. $\left\{n_1^2+n_2^2+n_3^2\to 1\right\}$ **(* n is a unit vector *)**

β

"$n_1 n_p n_q n_s \gamma\ \delta_{(1p}\delta_{qs)}$ Explicit Summation:";

$$A = \sum_{1=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_1\, n_p\, n_q\, n_s\, \gamma\ \mathbf{Symm2}$$

$$\gamma n_1^4+2\gamma n_1^2 n_2^2+\gamma n_2^4+2\gamma n_1^2 n_3^2+2\gamma n_2^2 n_3^2+\gamma n_3^4$$

B = Factor[A]

$\gamma\left(n_1^2+n_2^2+n_3^2\right)^2$

B /. $\left\{n_1^2+n_2^2+n_3^2\to 1\right\}$

γ

"Combining these equations yields:";
"$\mu_4^{0,0}= \alpha + \beta + \gamma$";

"";**

"Multiply the Equation by $n_1 n_p \delta_{qs}$";
" NOTE: This is inside the integrand ";
"Explicit Summation";

$$A = \sum_{1=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_1\, n_p\ \mathbf{KroneckerDelta[q,\ s]}\ (c_1-v_1)\left(c_p-v_p\right)\left(c_q-v_q\right)(c_s-v_s)$$

$$n_1^2(c_1-v_1)^4+2n_1n_2(c_1-v_1)^3(c_2-v_2)+n_1^2(c_1-v_1)^2(c_2-v_2)^2+n_2^2(c_1-v_1)^2(c_2-v_2)^2+2n_1n_2(c_1-v_1)(c_2-v_2)^3+$$
$$n_2^2(c_2-v_2)^4+2n_1n_3(c_1-v_1)^3(c_3-v_3)+2n_2n_3(c_1-v_1)^2(c_2-v_2)(c_3-v_3)+2n_1n_3(c_1-v_1)(c_2-v_2)^2(c_3-v_3)+$$
$$2n_2n_3(c_2-v_2)^3(c_3-v_3)+n_1^2(c_1-v_1)^2(c_3-v_3)^2+n_3^2(c_1-v_1)^2(c_3-v_3)^2+2n_1n_2(c_1-v_1)(c_2-v_2)(c_3-v_3)^2+$$
$$n_2^2(c_2-v_2)^2(c_3-v_3)^2+n_3^2(c_2-v_2)^2(c_3-v_3)^2+2n_1n_3(c_1-v_1)(c_3-v_3)^3+2n_2n_3(c_2-v_2)(c_3-v_3)^3+n_3^2(c_3-v_3)^4$$

"Simplification Step 1: v_2 and v_3 are both zero.";

B = A /. {$v_2 \to 0$, $v_3 \to 0$}

$$c_2^4 n_2^2 + c_2^2 c_3^2 n_2^2 + 2 c_2^3 c_3 n_2 n_3 + 2 c_2 c_3^3 n_2 n_3 + c_2^2 c_3^2 n_3^2 + c_3^4 n_3^2 + 2 c_2^3 n_1 n_2 (c_1 - v_1) + 2 c_2 c_3^2 n_1 n_2 (c_1 - v_1) + 2 c_2^2 c_3 n_1 n_3 (c_1 - v_1) + 2 c_3^3 n_1 n_3 (c_1 - v_1) + c_2^2 n_1^2 (c_1 - v_1)^2 + c_3^2 n_1^2 (c_1 - v_1)^2 + c_2^2 n_2^2 (c_1 - v_1)^2 + 2 c_2 c_3 n_2 n_3 (c_1 - v_1)^2 + c_3^2 n_3^2 (c_1 - v_1)^2 + 2 c_2 n_1 n_2 (c_1 - v_1)^3 + 2 c_3 n_1 n_3 (c_1 - v_1)^3 + n_1^2 (c_1 - v_1)^4$$

"Simplification Step 2: n_2 and n_3 are both zero.";

F = B /. {$n_2 \to 0$, $n_3 \to 0$}
"Simplification Step 3: n is a unit vector"

$$c_2^2 n_1^2 (c_1 - v_1)^2 + c_3^2 n_1^2 (c_1 - v_1)^2 + n_1^2 (c_1 - v_1)^4 \;/.\; \{n_1^2 \to 1\}$$

$$c_2^2 (c_1 - v_1)^2 + c_3^2 (c_1 - v_1)^2 + (c_1 - v_1)^4$$
" This is clearly equal to: $\mu_2^{2,0}$ + $\mu_2^{0,2}$ + $\mu_4^{0,0}$ ";

"$n_1 n_p \delta_{qs} \alpha$ $n_1 n_p n_q n_s$ Explicit Summation:";

$$A = \sum_{1=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_1 n_p \,\text{KroneckerDelta}[q, s]\, \alpha\, n_1 n_p n_q n_s$$

$$\alpha n_1^6 + 3 \alpha n_1^4 n_2^2 + 3 \alpha n_1^2 n_2^4 + \alpha n_2^6 + 3 \alpha n_1^4 n_3^2 + 6 \alpha n_1^2 n_2^2 n_3^2 + 3 \alpha n_2^4 n_3^2 + 3 \alpha n_1^2 n_3^4 + 3 \alpha n_2^2 n_3^4 + \alpha n_3^6$$

B = Factor[A]

$$\alpha \left(n_1^2 + n_2^2 + n_3^2\right)^3$$

B /. $\{n_1^2 + n_2^2 + n_3^2 \to 1\}$

$$\alpha$$

"$n_1 n_p \delta_{qs} \beta$ $n_{(1} n_p \delta_{qs)}$ Explicit Summation:";

$$A = \sum_{1=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_1 n_p \,\text{KroneckerDelta}[q, s]\, \beta\, \text{Symm1}$$

$$\beta n_1^4 + \frac{7}{3} \beta n_1^2 n_2^2 + \beta n_2^4 + \frac{1}{24} \beta n_1^2 \left(4 n_1^2 + 4 n_2^2\right) + \frac{1}{24} \beta n_2^2 \left(4 n_1^2 + 4 n_2^2\right) + \frac{7}{3} \beta n_1^2 n_3^2 + \frac{7}{3} \beta n_2^2 n_3^2 + \beta n_3^4 + \frac{1}{24} \beta n_1^2 \left(4 n_1^2 + 4 n_3^2\right) + \frac{1}{24} \beta n_3^2 \left(4 n_1^2 + 4 n_3^2\right) + \frac{1}{24} \beta n_2^2 \left(4 n_2^2 + 4 n_3^2\right) + \frac{1}{24} \beta n_3^2 \left(4 n_2^2 + 4 n_3^2\right)$$

B = Factor[A]

$$\frac{4}{3} \beta \left(n_1^2 + n_2^2 + n_3^2\right)^2$$

B /. $\{n_1^2 + n_2^2 + n_3^2 \to 1\}$

$$\frac{4 \beta}{3}$$

"$n_1 n_p \delta_{qs} \gamma$ $\delta_{(1p} \delta_{qs)}$ Explicit Summation:";

$$A = \sum_{1=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3} n_1 n_p \,\text{KroneckerDelta}[q, s]\, \gamma\, \text{Symm2}$$

$$\frac{5 \gamma n_1^2}{3} + \frac{5 \gamma n_2^2}{3} + \frac{5 \gamma n_3^2}{3}$$

B = Factor[A]

$$\frac{5}{3} \gamma \left(n_1^2 + n_2^2 + n_3^2\right)$$

```
B /. {n_1^2 + n_2^2 + n_3^2 → 1}
```

$\frac{5\gamma}{3}$

```
"Combining these equations yields:";
```

"$\mu_2^{2,0} + \mu_2^{0,2} + \mu_4^{0,0} = \alpha + \frac{4\beta}{3} + \frac{5\gamma}{3}$";

```
"*************************************************";

"Multiply the Equation by δ_{lp}δ_{qs}";

"  NOTE: This is inside the integrand  ";
"Explicit Summation";
```

$A = \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}$ KroneckerDelta[l, p] KroneckerDelta[q, s] $(c_l - v_l)(c_p - v_p)(c_q - v_q)(c_s - v_s)$

$(c_1 - v_1)^4 + 2(c_1 - v_1)^2(c_2 - v_2)^2 + (c_2 - v_2)^4 + 2(c_1 - v_1)^2(c_3 - v_3)^2 + 2(c_2 - v_2)^2(c_3 - v_3)^2 + (c_3 - v_3)^4$

"Simplification Step 1: v_2 and v_3 are both zero.";

B = A /. {$v_2 \to 0$, $v_3 \to 0$}

$c_2^4 + 2c_2^2c_3^2 + c_3^4 + 2c_2^2(c_1 - v_1)^2 + 2c_3^2(c_1 - v_1)^2 + (c_1 - v_1)^4$

" This is clearly equal to: $\mu_0^{4,0} + 2\mu_0^{2,2} + \mu_0^{0,4} + 2\mu_2^{2,0} + 2\mu_2^{0,2} + \mu_4^{0,0}$ ";

"$\delta_{lp}\delta_{qs}\alpha\ n_l n_p n_q n_s$ Explicit Summation:";

$A = \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}$ KroneckerDelta[l, p] KroneckerDelta[q, s] $\alpha\, n_l\, n_p\, n_q\, n_s$

$\alpha n_1^4 + 2\alpha n_1^2 n_2^2 + \alpha n_2^4 + 2\alpha n_1^2 n_3^2 + 2\alpha n_2^2 n_3^2 + \alpha n_3^4$

B = Factor[A]

$\alpha(n_1^2 + n_2^2 + n_3^2)^2$

B /. {$n_1^2 + n_2^2 + n_3^2 \to 1$}

α

"$\delta_{lp}\delta_{qs}\beta\ n_{(l}n_p\delta_{qs)}$ Explicit Summation:";

```
A = Symm1 * KroneckerDelta[l, p] * KroneckerDelta[q, s];

B = Expand[A]
```

$\frac{1}{6}$ KroneckerDelta[l, p] KroneckerDelta[q, s]^2 $n_l n_p$ +

$\frac{1}{6}$ KroneckerDelta[l, p] KroneckerDelta[p, s] KroneckerDelta[q, s] $n_l n_q$ +

$\frac{1}{6}$ KroneckerDelta[l, p] KroneckerDelta[l, s] KroneckerDelta[q, s] $n_p n_q$ +

$\frac{1}{6}$ KroneckerDelta[l, p] KroneckerDelta[p, q] KroneckerDelta[q, s] $n_l n_s$ +

$\frac{1}{6}$ KroneckerDelta[l, p] KroneckerDelta[l, q] KroneckerDelta[q, s] $n_p n_s$ +

$\frac{1}{6}$ KroneckerDelta[l, p]^2 KroneckerDelta[q, s] $n_q n_s$

```
A = ∑_{l=1}^{3} ∑_{p=1}^{3} ∑_{q=1}^{3} ∑_{s=1}^{3} β B
```

$$\beta\, n_1^2 + \beta\, n_2^2 + 2\,\beta\left(\frac{n_1^2}{6} + \frac{n_2^2}{6}\right) + \beta\, n_3^2 + 2\,\beta\left(\frac{n_1^2}{6} + \frac{n_3^2}{6}\right) + 2\,\beta\left(\frac{n_2^2}{6} + \frac{n_3^2}{6}\right)$$

```
B = Factor[A]
```

$$\frac{5}{3}\,\beta\,\left(n_1^2 + n_2^2 + n_3^2\right)$$

```
B /. {n_1^2 + n_2^2 + n_3^2 → 1}
```

$$\frac{5\,\beta}{3}$$

"$\delta_{lp}\delta_{qs}\gamma\ \delta_{(lp}\delta_{qs)}$ **Explicit Summation:";**

$$A = \sum_{l=1}^{3}\sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{s=1}^{3}\text{KroneckerDelta}[l,p]\ \text{KroneckerDelta}[q,s]\ \gamma\ \text{Symm2}$$

5γ

"Combining these equations yields:";

"$\mu_0^{4,0} + 2\mu_0^{2,2} + \mu_0^{0,4} + 2\mu_2^{2,0} + 2\mu_2^{0,2} + \mu_4^{0,0} = \alpha + \frac{5\beta}{3} + 5\gamma$";

"";**

"The combined system of equations is:"

"$\mu_4^{0,0} = \alpha + \beta + \gamma$";

"$\mu_2^{2,0} + \mu_2^{0,2} + \mu_4^{0,0} = \alpha + \frac{4\beta}{3} + \frac{5\gamma}{3}$";

"$\mu_0^{4,0} + 2\mu_0^{2,2} + \mu_0^{0,4} + 2\mu_2^{2,0} + 2\mu_2^{0,2} + \mu_4^{0,0} = \alpha + \frac{5\beta}{3} + 5\gamma$";

"";**

"Using the Recursion Relation, Find an Expression for $\mu_4^{0,0}$"

$\mu_{-1} = 0;$

$$\mu_0 = \text{FullSimplify}\left[(p-1)!!\ (q-1)!!\prod_{k=1}^{\frac{p+q}{2}}\frac{1+\frac{v^2}{(2-2k+r)^2}}{1-2k+r}\ /.\ \{p\to 0,\ q\to 0\}\right];$$

$$\mu_1 = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{v}{r}\right)\hat{}2\right)\mu_{n-2} - (2\,(n-1)+p+q)\ \frac{v}{r}\ \mu_{n-1}\right)\ /.\ \{q\to 0,\ p\to 0,\ n\to 1\};$$

$$\mu_2 = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{v}{r}\right)\hat{}2\right)\mu_{n-2} - (2\,(n-1)+p+q)\ \frac{v}{r}\ \mu_{n-1}\right)\ /.\ \{q\to 0,\ p\to 0,\ n\to 2\};$$

$$\mu_3 = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{v}{r}\right)\hat{}2\right)\mu_{n-2} - (2\,(n-1)+p+q)\ \frac{v}{r}\ \mu_{n-1}\right)\ /.\ \{q\to 0,\ p\to 0,\ n\to 3\};$$

$$\mu_4 = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{v}{r}\right)\hat{}2\right)\mu_{n-2} - (2\,(n-1)+p+q)\ \frac{v}{r}\ \mu_{n-1}\right)\ /.\ \{q\to 0,\ p\to 0,\ n\to 4\};$$

FullSimplify[μ_4]

$$\frac{3\left(r^2+v^2\right)\left((-2+r)\,r^2+(6+r)\,v^2\right)}{(-3+r)\,(-2+r)\,(-1+r)\,r^4};$$

$$\text{"}\quad \mu_4^{0,0} = \frac{3\left(r^2+v^2\right)\left((-2+r)\,r^2+(6+r)\,v^2\right)}{(-3+r)\,(-2+r)\,(-1+r)\,r^4}\quad\text{"};$$

"Using the Recursion Relation, Find an Expression for $\mu_2^{0,2}$ ";

$\mu_{-1} = 0;$

$$\mu_0 = \text{FullSimplify}\left[(p-1)!!\ (q-1)!!\prod_{k=1}^{\frac{p+q}{2}}\frac{1+\frac{v^2}{(2-2k+r)^2}}{1-2k+r}\ /.\ \{p\to 0,\ q\to 2\}\right];$$

$$\mu_1 = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{v}{r}\right)\hat{}2\right)\mu_{n-2} - (2\,(n-1)+p+q)\ \frac{v}{r}\ \mu_{n-1}\right)\ /.\ \{q\to 2,\ p\to 0,\ n\to 1\};$$

$$\mu_2 = \frac{1}{1-n-p-q+r}\left((n-1)\left(1+\left(\frac{v}{r}\right)\hat{}2\right)\mu_{n-2} - (2\,(n-1)+p+q)\ \frac{v}{r}\ \mu_{n-1}\right)\ /.\ \{q\to 2,\ p\to 0,\ n\to 2\};$$

```
FullSimplify[μ2]
```

$$\frac{(-2+r)\, r^4 + 2 r^2 (2+r)\, v^2 + (6+r)\, v^4}{(-3+r)(-2+r)(-1+r)\, r^4}$$

" $\mu_2^{0,2} = \mu_2^{2,0} = \dfrac{(-2+r)\, r^4 + 2 r^2 (2+r)\, v^2 + (6+r)\, v^4}{(-3+r)(-2+r)(-1+r)\, r^4}$ "

"Using the Recursion Relation, Find an Expression for $\mu_0^{2,2}$ ";

$\mu_{-1} = 0;$

$\mu_0 = \text{FullSimplify}\left[(p-1)!!\,(q-1)!!\prod_{k=1}^{\frac{p+q}{2}} \frac{1+\frac{v^2}{(2-2k+r)^2}}{1-2k+r} \text{ /. } \{p \to 2,\ q \to 2\}\right];$

```
FullSimplify[μ0]
```

$$\frac{\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)\, r^2}$$

" $\mu_0^{2,2} = \dfrac{\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)\, r^2}$ "

"Using the Recursion Relation, Find an Expression for $\mu_0^{4,0}$"

$\mu_{-1} = 0;$

$\mu_0 = \text{FullSimplify}\left[(p-1)!!\,(q-1)!!\prod_{k=1}^{\frac{p+q}{2}} \frac{1+\frac{v^2}{(2-2k+r)^2}}{1-2k+r} \text{ /. } \{p \to 4,\ q \to 0\}\right];$

```
FullSimplify[μ0]
```

$$\frac{3\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)\, r^2}$$

" $\mu_4^{4,0} = \dfrac{3\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)\, r^2}$ "

"**"

"List of All Moments:"

" $\mu_4^{4,0} = \mu_4^{0,4} = \dfrac{3\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)\, r^2}$ "

" $\mu_0^{2,2} = \dfrac{\left((-2+r)^2+v^2\right)\left(r^2+v^2\right)}{(-3+r)(-2+r)^2(-1+r)\, r^2}$ "

" $\mu_2^{0,2} = \mu_2^{2,0} = \dfrac{(-2+r)\, r^4 + 2 r^2 (2+r)\, v^2 + (6+r)\, v^4}{(-3+r)(-2+r)(-1+r)\, r^4}$ "

" $\mu_4^{0,0} = \dfrac{3\left(r^2+v^2\right)\left((-2+r)\, r^2 + (6+r)\, v^2\right)}{(-3+r)(-2+r)(-1+r)\, r^4}$ ";

FullSimplify$\left[\left(\dfrac{3\left(r^2+v^2\right)\left((-2+r)\,r^2+(6+r)\,v^2\right)}{(-3+r)\,(-2+r)\,(-1+r)\,r^4}\right)\Big/\left(\dfrac{(-2+r)\,r^4+2\,r^2\,(2+r)\,v^2+(6+r)\,v^4}{(-3+r)\,(-2+r)\,(-1+r)\,r^4}\right)\right]$

```
3
```

```
"*********************************************";
"Relations Between Moments:"
```

" $\mu_4^{0,0} = 3\mu_2^{0,2} = 3\mu_2^{2,0}$ "

" $\mu_4^{4,0} = \mu_4^{0,4} = 3\mu_0^{2,2}$ "

```
"*********************************************";
```

```
"The combined system of equations is:"
```

"$\mu_4^{0,0} = \alpha + \beta + \gamma$";

"$\frac{1}{3}\mu_4^{0,0} + \frac{1}{3}\mu_4^{0,0} + \mu_4^{0,0} = \alpha + \frac{4\beta}{3} + \frac{5\gamma}{3}$";

"$\mu_0^{4,0} + 2\frac{1}{3}\mu_4^{4,0} + \mu_0^{0,4} + 2\frac{1}{3}\mu_4^{0,0} + 2\frac{1}{3}\mu_4^{0,0} + \mu_4^{0,0} = \alpha + \frac{5\beta}{3} + 5\gamma$";

```
"**************************************************";
```

```
" Solving this system of equations : ";
```

" $x = \mu_4^{0,0}$, $y = \mu_0^{4,0}$ "

ans = Solve[

$\quad x == \alpha + \beta + \gamma$ &&

$\quad (1/3)\,x + (1/3)\,x + x == \alpha + \frac{4}{3}\beta + \frac{5}{3}\gamma$ &&

$\quad y + 2\,(1/3)\,y + y + 2\,(1/3)\,x + 2\,(1/3)\,x + x == \alpha + \frac{5}{3}\beta + 5\gamma$

$\quad , \{\alpha, \beta, \gamma\}]$;

$x = \mu_4^{0,0}$, $y = \mu_0^{4,0}$

```
FullSimplify[ans]
```

$\{\{\alpha \to -x + y,\ \beta \to 2\,(x - y),\ \gamma \to y\}\}$

```
" This gives: ";
```

" $\alpha = \mu_0^{4,0} - \mu_4^{0,0}$ ";

" $\beta = -2(\mu_0^{4,0} - \mu_4^{0,0})$";

" $\gamma = \mu_0^{4,0}$ ";

Appendix B

DETAILS OF THE SOLUTION TO HYPERBOLIC EQUATIONS IN 3D

B.1 Converting to Primitive Equations (Mathematica)

Mathematica is used to convert from conservative to primitive equations in indicial notation. The source code used is presented here.

Derivation of Primitive Equations :

1 st Equation :

As a conservative equation :

$$\partial_t \rho + \partial_{x_i} (v_i \rho) = 0 \qquad (1)$$

```
cEq1 = ∂_t ρ[t, x] + ∂_x (ρ[t, x] v_i[t, x]) == 0;
pEq1 = cEq1 /. {
    v_i[t, x] → v_i,
    ρ^(0,1)[t, x] → ρ^(0,1),
    ρ[t, x] → ρ,
    (v_i)^(0,1)[t, x] → (v_i)^(0,1),
    ρ^(1,0)[t, x] → ρ^(1,0)
  }
```

$$v_i \rho^{(0,1)} + \rho (v_i)^{(0,1)} + \rho^{(1,0)} == 0$$

Giving the primitive equation :

$$\partial_t \rho + \rho \partial_{x_i} (v_i) + v_i \partial_{x_i} \rho = 0 \qquad (2)$$

2 nd Equation :

As a conservative equation :

$$\partial_t (\rho v_i) + \partial_{x_j} \left(P_{i,j} + \rho v_i v_j\right) = \lambda_i \qquad (3)$$

Substitute $\rho \Theta_{ij} = P_{ij}$

```
cEq2L = ∂_t (ρ[t, x] v_i[t, x]) + ∂_x (ρ[t, x] Θ_{i,j}[t, x] + ρ[t, x] v_i[t, x] v_j[t, x]);
cEq2R = λ_i;
pEq2Step1 = cEq2L /. {
    v_i[t, x] → v_i,
    ρ^(0,1)[t, x] → ρ^(0,1),
    ρ[t, x] → ρ,
    (v_i)^(0,1)[t, x] → (v_i)^(0,1),
    ρ^(1,0)[t, x] → ρ^(1,0),
    Θ_{i,j}[t, x] → Θ_{i,j},
    (v_i)^(1,0)[t, x] → (v_i)^(1,0),
    (v_j)^(0,1)[t, x] → (v_j)^(0,1),
    v_j[t, x] → v_j,
    (Θ_{i,j})^(0,1)[t, x] → (Θ_{i,j})^(0,1)
  }
pEq2Step2 = Collect[pEq2Step1, v_i] /. {v_j ρ^(0,1) + ρ (v_j)^(0,1) + ρ^(1,0) → 0};
pE32Step3 = pEq2Step2 == cEq2R
pEq2 = Expand[pEq2Step2 / ρ] == cEq2R / ρ
```

$$v_i v_j \rho^{(0,1)} + \Theta_{i,j} \rho^{(0,1)} + \rho v_j (v_i)^{(0,1)} + \rho v_i \left(v_j\right)^{(0,1)} + \rho \left(\Theta_{i,j}\right)^{(0,1)} + v_i \rho^{(1,0)} + \rho (v_i)^{(1,0)}$$

$$\Theta_{i,j} \rho^{(0,1)} + \rho v_j (v_i)^{(0,1)} + \rho \left(\Theta_{i,j}\right)^{(0,1)} + \rho (v_i)^{(1,0)} == \lambda_i$$

$$\frac{\Theta_{i,j} \rho^{(0,1)}}{\rho} + v_j (v_i)^{(0,1)} + \left(\Theta_{i,j}\right)^{(0,1)} + (v_i)^{(1,0)} == \frac{\lambda_i}{\rho}$$

Giving the primitive equation (dividing by ρ) :

$$\frac{\Theta_{ij}}{\rho}\,\partial_{x_j}\rho + v_j\,\partial_{x_j}(v_i) + \partial_{x_j}\left(\Theta_{i,j}\right) + \partial_t(v_i) == \frac{\lambda_i}{\rho} \tag{4}$$

3 rd Equation :

As a conservative equation :

$$\partial_t\left(P_{ij} + \rho v_i\, v_j\right) + \partial_{x_k}\left(\rho v_i\, v_j\, v_k + 3\, v_{(i}\, P_{jk)}\right) = \psi_{ij} - \nabla\cdot h_{ijk} \tag{5}$$

Substitute $\rho\,\Theta_{ij} = P_{ij}$

```
cEq3L = ∂_t (ρ[t, x] Θ_{i,j}[t, x] + ρ[t, x] v_i[t, x] v_j[t, x]) + ∂_x (ρ [t, x] v_i[t, x] v_j[t, x] v_k[t, x] +
     v_i[t, x] ρ[t, x] Θ_{j,k}[t, x] + v_j[t, x] ρ[t, x] Θ_{i,k}[t, x] + v_k[t, x] ρ[t, x] Θ_{i,j}[t, x]);
cEq3R =
  ψ_{i,j} -
   (h_{ijk})^(0,1);

pEq3Step1 = cEq3L /. {
    v_i[t, x] → v_i,
    ρ^(0,1)[t, x] → ρ^(0,1),
    ρ[t, x] → ρ,
    (v_i)^(0,1)[t, x] → (v_i)^(0,1),
    ρ^(1,0)[t, x] → ρ^(1,0),
    Θ_{i,j}[t, x] → Θ_{i,j},
    (v_i)^(1,0)[t, x] → (v_i)^(1,0),
    (v_j)^(0,1)[t, x] → (v_j)^(0,1),
    v_j[t, x] → v_j,
    (Θ_{i,j})^(0,1)[t, x] → (Θ_{i,j})^(0,1),
    v_k[t, x] → v_k,
    (v_k)^(0,1)[t, x] → (v_k)^(0,1),
    Θ_{i,k}[t, x] → Θ_{i,k},
    (Θ_{i,j})^(1,0)[t, x] → (Θ_{i,j})^(1,0),
    (v_j)^(1,0)[t, x] -> (v_j)^(1,0),
    (Θ_{i,k})^(0,1)[t, x] → (Θ_{i,k})^(0,1),
    Θ_{j,k}[t, x] → Θ_{j,k},
    (Θ_{j,k})^(0,1)[t, x] → (Θ_{j,k})^(0,1)
   };
pEq3Step2 = Collect[pEq3Step1, Θ_{i,j}] /. {v_k ρ^(0,1) + ρ (v_k)^(0,1) + ρ^(1,0) → 0};
pEq3Step3 = Collect[pEq3Step2, v_i v_j] /. {v_k ρ^(0,1) + ρ (v_k)^(0,1) + ρ^(1,0) → 0};
pEq3Step4 = Collect[pEq3Step3 , v_j] /. {Θ_{i,k} ρ^(0,1) + ρ v_k (v_i)^(0,1) + ρ (Θ_{i,k})^(0,1) + ρ (v_i)^(1,0) → λ_i};
pEq3Step5 = Collect[pEq3Step4, v_i ] /. {Θ_{j,k} ρ^(0,1) + ρ v_k (v_j)^(0,1) + ρ (Θ_{j,k})^(0,1) + ρ (v_j)^(1,0) → λ_j};
pEq3Step6 = pEq3Step5 - (v_j λ_i + v_i λ_j) == cEq3R - (v_j λ_i + v_i λ_j)
```

$$\rho\,\Theta_{j,k}\,(v_i)^{(0,1)} + \rho\,\Theta_{i,k}\,(v_j)^{(0,1)} + \rho\, v_k\,(\Theta_{i,j})^{(0,1)} + \rho\,(\Theta_{i,j})^{(1,0)} == -v_j\,\lambda_i - v_i\,\lambda_j + \psi_{i,j} - (h_{ijk})^{(0,1)}$$

```
pEq3 = Expand[(pEq3Step5 - (v_j λ_i + v_i λ_j)) / ρ] == (cEq3R - (v_j λ_i + v_i λ_j)) / ρ
```

$$\Theta_{j,k}\,(v_i)^{(0,1)} + \Theta_{i,k}\,(v_j)^{(0,1)} + v_k\,(\Theta_{i,j})^{(0,1)} + (\Theta_{i,j})^{(1,0)} == \frac{-v_j\,\lambda_i - v_i\,\lambda_j + \psi_{i,j} - (h_{ijk})^{(0,1)}}{\rho}$$

Giving the primitive equation (dividing by ρ) :

$$\Theta_{jk}\,\partial_{x_k}(v_i) + \Theta_{ik}\,\partial_{x_k}\left(v_j\right) + v_k\,\partial_{x_k}\left(\Theta_{ij}\right) + \partial_t\,\partial_{x_k}\left(\Theta_{ij}\right) = \frac{\psi_{i,j} - v_j\,\lambda_i - v_i\,\lambda_j - \partial_{x_k}\left(h_{ijk}\right)}{\rho} \tag{6}$$

4 th Equation :

As a conservative equation (with the source term containing $\nabla\cdot\left(\Delta_{\text{ijkm}}\right)$, $h_{\text{kmi}}\,\nabla\cdot\left(v_j\right)$ and $h_{\text{mij}}\,\nabla\cdot\left(v_k\right)$) :

$$\partial_t\left(\rho\, v_i\, v_j\, v_k + 3\, v_{(i}\, P_{jk)} + h_{ijk}\right) + \partial_{x_m}\left(\rho\, v_i\, v_j\, v_k\, v_m + 6\, v_{(i}\, v_j\, P_{km)} + 4\, v_{(i}\, h_{jkm)}\right) - h_{kmi}\, \nabla\cdot\left(v_j\right) - h_{mij}\, \nabla\cdot(v_k) = \gamma_{ijk} - \nabla\cdot\left(\Delta_{ijkm}\right) - h_{kmi}\, \nabla\cdot\left(v_j\right) - h_{mij}\, \nabla\cdot(v_k)$$

```
cEq4L = ∂_t (ρ[t, x] v_i[t, x] v_j[t, x] v_k[t, x] +
        (ρ[t, x] v_i[t, x] Θ_{j,k}[t, x] + ρ[t, x] v_k[t, x] Θ_{i,j}[t, x] + ρ[t, x] v_j[t, x] Θ_{k,i}[t, x]) + h_{ijk}[t, x]) +
    ∂_x (ρ[t, x] v_i[t, x] v_j[t, x] v_k[t, x] v_m[t, x] + (ρ[t, x] v_i[t, x] v_j[t, x] Θ_{k,m}[t, x] +
          ρ[t, x] v_i[t, x] v_k[t, x] Θ_{j,m}[t, x] + ρ[t, x] v_k[t, x] v_j[t, x] Θ_{i,m}[t, x] + ρ[t, x] v_i[t, x] v_m[t, x] Θ_{j,k}[t, x] +
          ρ[t, x] v_m[t, x] v_j[t, x] Θ_{k,i}[t, x] + ρ[t, x] v_k[t, x] v_m[t, x] Θ_{i,j}[t, x]) +
        (v_i[t, x] h_{jkm}[t, x] + v_m[t, x] h_{ijk}[t, x] + v_k[t, x] h_{mij}[t, x] + v_j[t, x] h_{kmi}[t, x])) -
    h_{kmi}[t, x] ∂_x (v_j[t, x]) - h_{mij}[t, x] ∂_x (v_k[t, x]);
cEq4R = γ_{ijk} - h_{kmi}[t, x] ∂_x (v_j[t, x]) - h_{mij}[t, x] ∂_x (v_k[t, x]) - ∂_x (Δ_{ijkm}[t, x]) /. {h_{mij}[t, x] → h_{mij},
     h_{kmi}[t, x] → h_{kmi},  (v_k)^(0,1)[t, x] → (v_k)^(0,1),  (v_j)^(0,1)[t, x] → (v_j)^(0,1), (Δ_{ijkm})^(0,1)[t, x] → (Δ_{ijkm})^(0,1)};
pEq4Step1 = cEq4L /. {
     v_i[t, x] → v_i,
     ρ^(0,1)[t, x] → ρ^(0,1),
     ρ[t, x] → ρ,
     (v_i)^(0,1)[t, x] → (v_i)^(0,1),
     ρ^(1,0)[t, x] → ρ^(1,0),
     Θ_{i,j}[t, x] → Θ_{i,j},
     (v_i)^(1,0)[t, x] → (v_i)^(1,0),
     (v_j)^(0,1)[t, x] → (v_j)^(0,1),
     v_j[t, x] → v_j,
     (Θ_{i,j})^(0,1)[t, x] → (Θ_{i,j})^(0,1),
     v_k[t, x] → v_k,
     (v_k)^(0,1)[t, x] → (v_k)^(0,1),
     Θ_{i,k}[t, x] → Θ_{i,k},
     (Θ_{i,j})^(1,0)[t, x] → (Θ_{i,j})^(1,0),
     (v_j)^(1,0)[t, x] -> (v_j)^(1,0),
     (Θ_{i,k})^(0,1)[t, x] → (Θ_{i,k})^(0,1),
     Θ_{j,k}[t, x] → Θ_{j,k},
     (Θ_{j,k})^(0,1)[t, x] → (Θ_{j,k})^(0,1),
     h_{jkm}[t, x] → h_{jkm},
      v_m[t, x] →  v_m,
     Θ_{i,m}[t, x] → Θ_{i,m},
     Θ_{j,m}[t, x] → Θ_{j,m},
     Θ_{k,m}[t, x] → Θ_{k,m},
     Θ_{k,i}[t, x] → Θ_{k,i},
     (v_k)^(1,0)[t, x] → (v_k)^(1,0),
     (v_m)^(0,1)[t, x] → (v_m)^(0,1),
     (Θ_{k,m})^(0,1)[t, x] → (Θ_{k,m})^(0,1),
     h_{ijk}[t, x] → h_{ijk},
     (h_{ijk})^(0,1)[t, x] → (h_{ijk})^(0,1),
     (Θ_{k,i})^(0,1)[t, x] → (Θ_{k,i})^(0,1),
     (Θ_{k,i})^(1,0)[t, x] → (Θ_{k,i})^(1,0),
     (Θ_{j,k})^(1,0)[t, x] → (Θ_{j,k})^(1,0),
     (h_{ijk})^(1,0)[t, x] → (h_{ijk})^(1,0),
      (h_{kmi})^(0,1)[t, x] →  (h_{kmi})^(0,1),
     (Θ_{i,m})^(0,1)[t, x] → (Θ_{i,m})^(0,1),
```

$$\left(h_{mij}\right)^{(0,1)}[t,x] \to \left(h_{mij}\right)^{(0,1)},$$
$$\left(\Theta_{j,m}\right)^{(0,1)}[t,x] \to \left(\Theta_{j,m}\right)^{(0,1)},$$
$$\left(h_{jkm}\right)^{(0,1)}[t,x] \to \left(h_{jkm}\right)^{(0,1)}$$
$$\};$$

$$\mathtt{pEq4Step2} = \mathtt{Collect}\left[\mathtt{pEq4Step1},\ v_i\, v_j\, v_k\right] \mathbin{/.} \left\{v_m\, \rho^{(0,1)} + \rho\, (v_m)^{(0,1)} + \rho^{(1,0)} \to 0\right\};$$
$$\mathtt{pEq4Step3} = \mathtt{Collect}\left[\mathtt{pEq4Step2},\ v_k\, \Theta_{i,j}\right] \mathbin{/.} \left\{v_m\, \rho^{(0,1)} + \rho\, (v_m)^{(0,1)} + \rho^{(1,0)} \to 0\right\};$$
$$\mathtt{pEq4Step4} = \mathtt{Collect}\left[\mathtt{pEq4Step3},\ v_i\, \Theta_{j,k}\right] \mathbin{/.} \left\{v_m\, \rho^{(0,1)} + \rho\, (v_m)^{(0,1)} + \rho^{(1,0)} \to 0\right\};$$
$$\mathtt{pEq4Step5} = \mathtt{Collect}\left[\mathtt{pEq4Step4},\ v_j\, \Theta_{k,i}\right] \mathbin{/.} \left\{v_m\, \rho^{(0,1)} + \rho\, (v_m)^{(0,1)} + \rho^{(1,0)} \to 0\right\};$$
$$\mathtt{pEq4Step6} = \mathtt{Collect}\left[\mathtt{pEq4Step5},\ v_j\, v_k\right] \mathbin{/.} \left\{\Theta_{i,m}\, \rho^{(0,1)} + \rho\, v_m\, (v_i)^{(0,1)} + \rho\, (\Theta_{i,m})^{(0,1)} + \rho\, (v_i)^{(1,0)} \to \lambda_i\right\};$$
$$\mathtt{pEq4Step7} = \mathtt{Collect}[\mathtt{pEq4Step6},\ v_i\, v_k] \mathbin{/.} \left\{\Theta_{j,m}\, \rho^{(0,1)} + \rho\, v_m\, \left(v_j\right)^{(0,1)} + \rho\, \left(\Theta_{j,m}\right)^{(0,1)} + \rho\, \left(v_j\right)^{(1,0)} \to \lambda_j\right\};$$
$$\mathtt{pEq4Step8} = \mathtt{Collect}\left[\mathtt{pEq4Step7},\ v_i\, v_j\right] \mathbin{/.} \left\{\Theta_{k,m}\, \rho^{(0,1)} + \rho\, v_m\, (v_k)^{(0,1)} + \rho\, (\Theta_{k,m})^{(0,1)} + \rho\, (v_k)^{(1,0)} \to \lambda_k\right\};$$
$$\mathtt{pEq4Step9} = \mathtt{Collect}[\mathtt{pEq4Step8},\ v_k] \mathbin{/.}$$
$$\left\{\left(h_{mij}\right)^{(0,1)} + \rho\, \Theta_{j,m}\, (v_i)^{(0,1)} + \rho\, \Theta_{i,m}\, \left(v_j\right)^{(0,1)} + \rho\, v_m\, \left(\Theta_{i,j}\right)^{(0,1)} + \rho\, \left(\Theta_{i,j}\right)^{(1,0)} \to -v_j\, \lambda_i - v_i\, \lambda_j + \psi_{i,j}\right\};$$
$$\mathtt{pEq4Step10} = \mathtt{Collect}[\mathtt{pEq4Step9},\ v_i] \mathbin{/.}$$
$$\left\{\left(h_{jkm}\right)^{(0,1)} + \rho\, \Theta_{k,m}\, \left(v_j\right)^{(0,1)} + \rho\, \Theta_{j,m}\, (v_k)^{(0,1)} + \rho\, v_m\, \left(\Theta_{j,k}\right)^{(0,1)} + \rho\, \left(\Theta_{j,k}\right)^{(1,0)} \to -v_j\, \lambda_k - v_k\, \lambda_j + \psi_{k,j}\right\};$$
$$\mathtt{pEq4Step11} = \mathtt{Collect}\left[\mathtt{pEq4Step10},\ v_j\ \right] \mathbin{/.}$$
$$\left\{(h_{kmi})^{(0,1)} + \rho\, \Theta_{k,m}\, (v_i)^{(0,1)} + \rho\, \Theta_{i,m}\, (v_k)^{(0,1)} + \rho\, v_m\, (\Theta_{k,i})^{(0,1)} + \rho\, (\Theta_{k,i})^{(1,0)} \to -v_k\, \lambda_i - v_i\, \lambda_k + \psi_{i,k}\right\};$$
$$\mathtt{pEq4Step12} = \mathtt{Collect}\left[\mathtt{pEq4Step11},\ \Theta_{j,k}\right] \mathbin{/.} \left\{\rho\, v_m\, (v_i)^{(0,1)} + \rho\, (v_i)^{(1,0)} \to \lambda_i - \Theta_{i,m}\, \rho^{(0,1)} - \rho\, (\Theta_{i,m})^{(0,1)}\right\};$$
$$\mathtt{pEq4Step13} = \mathtt{Collect}[\mathtt{pEq4Step12},\ \Theta_{k,i}] \mathbin{/.} \left\{\rho\, v_m\, \left(v_j\right)^{(0,1)} + \rho\, \left(v_j\right)^{(1,0)} \to \lambda_j - \Theta_{j,m}\, \rho^{(0,1)} - \rho\, \left(\Theta_{j,m}\right)^{(0,1)}\right\};$$
$$\mathtt{pEq4Step14} = \mathtt{Collect}\left[\mathtt{pEq4Step13},\ \Theta_{i,j}\right] \mathbin{/.} \left\{\rho\, v_m\, (v_k)^{(0,1)} + \rho\, (v_k)^{(1,0)} \to \lambda_k - \Theta_{k,m}\, \rho^{(0,1)} - \rho\, (\Theta_{k,m})^{(0,1)}\right\};$$
$$\mathtt{pEqStep15} = \mathtt{Expand}\left[\mathtt{pEq4Step14} - \left(-v_j\, v_k\, \lambda_i - v_i\, v_k\, \lambda_j - v_i\, v_j\, \lambda_k + \lambda_k\, \Theta_{i,j} + \lambda_i\, \Theta_{j,k} + \lambda_j\, \Theta_{k,i} + v_k\, \psi_{i,j} + v_j\, \psi_{i,k} + v_i\, \psi_{k,j}\right) == \right.$$
$$\left. \mathtt{cEq4R} - \left(-v_j\, v_k\, \lambda_i - v_i\, v_k\, \lambda_j - v_i\, v_j\, \lambda_k + \lambda_k\, \Theta_{i,j} + \lambda_i\, \Theta_{j,k} + \lambda_j\, \Theta_{k,i} + v_k\, \psi_{i,j} + v_j\, \psi_{i,k} + v_i\, \psi_{k,j}\right)\right]$$

$$-\Theta_{i,m}\, \Theta_{j,k}\, \rho^{(0,1)} - \Theta_{j,m}\, \Theta_{k,i}\, \rho^{(0,1)} - \Theta_{i,j}\, \Theta_{k,m}\, \rho^{(0,1)} + v_m\, \left(h_{ijk}\right)^{(0,1)} +$$
$$h_{jkm}\, (v_i)^{(0,1)} + h_{ijk}\, (v_m)^{(0,1)} - \rho\, \Theta_{j,k}\, (\Theta_{i,m})^{(0,1)} - \rho\, \Theta_{k,i}\, \left(\Theta_{j,m}\right)^{(0,1)} - \rho\, \Theta_{i,j}\, (\Theta_{k,m})^{(0,1)} + \left(h_{ijk}\right)^{(1,0)} ==$$
$$\gamma_{ijk} + v_j\, v_k\, \lambda_i + v_i\, v_k\, \lambda_j + v_i\, v_j\, \lambda_k - \lambda_k\, \Theta_{i,j} - \lambda_i\, \Theta_{j,k} - \lambda_j\, \Theta_{k,i} - v_k\, \psi_{i,j} - v_j\, \psi_{i,k} - v_i\, \psi_{k,j} - h_{kmi}\, \left(v_j\right)^{(0,1)} - h_{mij}\, (v_k)^{(0,1)} - \left(\Delta_{ijkm}\right)^{(0,1)}$$

Which gives a primitive equation of :

$$\begin{aligned} &-\Theta_{i,m}\, \Theta_{j,k}\, \partial_{x_m}\rho - \Theta_{j,m}\, \Theta_{k,i}\, \partial_{x_m}\rho - \Theta_{i,j}\, \Theta_{k,m}\, \partial_{x_m}\, \rho + v_m\, \partial_{x_m}\left(h_{ijk}\right) + h_{jkm}\, \partial_{x_m}(v_i) + h_{ijk}\, \partial_{x_m}(v_m) - \\ &\rho\, \Theta_{j,k}\, \partial_{x_m}(\Theta_{i,m}) - \rho\, \Theta_{k,i}\, \partial_{x_m}\left(\Theta_{j,m}\right) - \rho\, \Theta_{i,j}\, \partial_{x_m}(\Theta_{k,m}) + \partial_t\left(h_{ijk}\right) == \gamma_{ijk} + v_j\, v_k\, \lambda_i + v_i\, v_k\, \lambda_j + v_i\, v_j\, \lambda_k - \\ &\lambda_k\, \Theta_{i,j} - \lambda_i\, \Theta_{j,k} - \lambda_j\, \Theta_{k,i} - v_k\, \psi_{i,j} - v_j\, \psi_{i,k} - v_i\, \psi_{k,j} - h_{kmi}\, \partial_{x_m}\left(v_j\right) - h_{mij}\, \partial_{x_m}(v_k) - \partial_{x_m}\left(\Delta_{ijkm}\right) \end{aligned} \tag{8}$$

Now, the trace of the 4 th equation must be taken :

$$j \to k \text{ and } m \to j$$

$$\mathtt{pEq4Step16} = \mathtt{Expand}\left[\mathtt{pEq4Step14} - \left(-v_j\, v_k\, \lambda_i - v_i\, v_k\, \lambda_j - v_i\, v_j\, \lambda_k + \lambda_k\, \Theta_{i,j} + \lambda_i\, \Theta_{j,k} + \lambda_j\, \Theta_{k,i} + v_k\, \psi_{i,j} + v_j\, \psi_{i,k} + v_i\, \psi_{k,j}\right)\right] \mathbin{/.}$$
$$\left\{j \to k,\ h_{ijk} \to h_{ikk},\ h_{jkm} \to h_{kkm}\right\} \mathbin{/.} \left\{m \to j,\ \ h_{kkm} \to 2\, q_j,\ \ h_{ikk} \to 2\, q_i\right\};$$
$$\mathtt{pEq4Step17} = \mathtt{Expand}\left[\mathtt{cEq4R} - \left(-v_j\, v_k\, \lambda_i - v_i\, v_k\, \lambda_j - v_i\, v_j\, \lambda_k + \lambda_k\, \Theta_{i,j} + \lambda_i\, \Theta_{j,k} + \lambda_j\, \Theta_{k,i} + v_k\, \psi_{i,j} + v_j\, \psi_{i,k} + v_i\, \psi_{k,j}\right)\right] \mathbin{/.}$$
$$\left\{j \to k,\ h_{ijk} \to h_{ikk},\ h_{jkm} \to h_{kkm},\ h_{mij} \to h_{kmi},\ \Delta_{ijkm} \to \Delta_{ikkm},\ \gamma_{ijk} \to \gamma_{ikk}\right\} \mathbin{/.} \left\{m \to j,\ h_{kmi} \to h_{ijk},\ \Delta_{ikkm} \to R_{ij}\ \right\};$$
$$\mathtt{pEq4} = \mathtt{Expand}[\mathtt{pEq4Step16}/2 == \mathtt{pEq4Step17}/2] \mathbin{/.} \{\Theta_{i,k} \to \Theta_{k,i}\}$$

$$-\Theta_{k,i}\, \Theta_{k,j}\, \rho^{(0,1)} - \frac{1}{2}\, \Theta_{i,j}\, \Theta_{k,k}\, \rho^{(0,1)} + \frac{1}{2}\, v_j\, (2\, q_i)^{(0,1)} + q_j\, (v_i)^{(0,1)} + q_i\, \left(v_j\right)^{(0,1)} - \frac{1}{2}\, \rho\, \Theta_{k,k}\, \left(\Theta_{i,j}\right)^{(0,1)} - \rho\, \Theta_{k,i}\, \left(\Theta_{k,j}\right)^{(0,1)} +$$
$$\frac{1}{2}\, (2\, q_i)^{(1,0)} == \frac{\gamma_{ikk}}{2} + \frac{1}{2}\, v_k^2\, \lambda_i + v_i\, v_k\, \lambda_k - \lambda_k\, \Theta_{k,i} - \frac{1}{2}\, \lambda_i\, \Theta_{k,k} - v_k\, \psi_{i,k} - \frac{1}{2}\, v_i\, \psi_{k,k} - \frac{1}{2}\, \left(R_{ij}\right)^{(0,1)} - h_{ijk}\, (v_k)^{(0,1)}$$

Giving the final primitive equation :

$$\begin{aligned} &-\Theta_{ki}\, \Theta_{kj}\, \partial_{x_j}\rho - \frac{1}{2}\, \Theta_{ij}\, \Theta_{kk}\, \partial_{x_j}\, \rho + v_j\, \partial_{x_j}(\,q_i) + q_j\, \partial_{x_j}(v_i) + q_i\, \partial_{x_j}\left(v_j\right) - \frac{1}{2}\, \rho\, \Theta_{kk}\, \partial_{x_j}\left(\Theta_{ij}\right) - \rho\, \Theta_{ki}\, \partial_{x_j}\left(\Theta_{kj}\right) + \\ &\partial_t\,(q_i) == \frac{\gamma_{ikk}}{2} + \frac{1}{2}\, v_k^2\, \lambda_i + v_i\, v_k\, \lambda_k - \lambda_k\, \Theta_{ki} - \frac{1}{2}\, \lambda_i\, \Theta_{kk} - v_k\, \psi_{ik} - \frac{1}{2}\, v_i\, \psi_{kk} - \frac{1}{2}\, \partial_{x_j}\left(R_{ij}\right) - h_{ijk}\, \partial_{x_j}(v_k) \end{aligned} \tag{9}$$

B.2 Derivation of Eigensystem and Jacobian Matrices (Mathematica)

Derivation of A_p Matrix :

```
Remove["Global`*"];
LHS = List[];
RHS = List[];
```

```
Remove::rmnsm : There are no symbols matching "Global`*". ≫
```

Primitive Equation 1 :

$$v_i\,\partial_t\rho+\rho\,\partial_{x_i}(v_i)+\partial_{x_i}\rho=0$$

$$\mathbf{Eq1L1}\;=\;v_i\,\rho^{(0,i)}+\rho\,(v_i)^{(0,i)}+\rho^{(1,0)};$$
$$\mathbf{Eq1R}\;=\;0;$$

Sum on i :

$$\begin{aligned}&\mathbf{Eq1L}\;=\;\mathbf{Expand}\Big[\mathbf{Eq1L1}\;/.\;\Big\{\\&\quad v_i\,\rho^{(0,i)}\to\left(v_x\,\rho^{(0,x)}+v_y\,\rho^{(0,y)}+v_z\,\rho^{(0,z)}\right),\\&\quad(v_i)^{(0,i)}\to\left((v_x)^{(0,x)}+\left(v_y\right)^{(0,y)}+(v_z)^{(0,z)}\right)\\&\Big\}\Big];\end{aligned}$$
$$\mathbf{Eq1}\;=\;\mathbf{Eq1L}\;==\;\mathbf{Eq1R}$$

$$v_x\,\rho^{(0,x)}+\rho\,(v_x)^{(0,x)}+v_y\,\rho^{(0,y)}+\rho\,\left(v_y\right)^{(0,y)}+v_z\,\rho^{(0,z)}+\rho\,(v_z)^{(0,z)}+\rho^{(1,0)}==0$$

```
LHS = Append[LHS, Eq1L];
RHS = Append[RHS, Eq1R];
```

Primitive Equations 2 - 4 :

$$\frac{\Theta_{ij}}{\rho}\,\partial_{x_j}\,\rho+v_j\,\partial_{x_j}(v_i)+\partial_{x_j}\left(\Theta_{i,j}\right)+\partial_t(v_i)==\frac{\lambda_i}{\rho}$$

$$\mathbf{Eq2L1}\;=\;\frac{\Theta_{i,j}\,\rho^{(0,j)}}{\rho}+v_j\,(v_i)^{(0,j)}+\left(\Theta_{i,j}\right)^{(0,j)}+(v_i)^{(1,0)};$$

$$\mathbf{Eq2R2}\;=\;\frac{\lambda_i}{\rho};$$

First, sum on j :

$$\begin{aligned}&\mathbf{Eq2L2}=\mathbf{Expand}\Big[\mathbf{Eq2L1}\;/.\;\Big\{\\&\quad\Theta_{i,j}\,\rho^{(0,j)}\to\left(\Theta_{i,x}\,\rho^{(0,x)}+\Theta_{i,y}\,\rho^{(0,y)}+\Theta_{i,z}\,\rho^{(0,z)}\right),\\&\quad v_j\,(v_i)^{(0,j)}\to\left(v_x\,(v_i)^{(0,x)}+v_y\,(v_i)^{(0,y)}+v_z\,(v_i)^{(0,z)}\right),\\&\quad\left(\Theta_{i,j}\right)^{(0,j)}\to\left((\Theta_{i,x})^{(0,x)}+\left(\Theta_{i,y}\right)^{(0,y)}+(\Theta_{i,z})^{(0,z)}\right)\\&\Big\}\Big]\end{aligned}$$

$$\frac{\Theta_{i,x}\,\rho^{(0,x)}}{\rho}+v_x\,(v_i)^{(0,x)}+(\Theta_{i,x})^{(0,x)}+\frac{\Theta_{i,y}\,\rho^{(0,y)}}{\rho}+v_y\,(v_i)^{(0,y)}+\left(\Theta_{i,y}\right)^{(0,y)}+\frac{\Theta_{i,z}\,\rho^{(0,z)}}{\rho}+v_z\,(v_i)^{(0,z)}+(\Theta_{i,z})^{(0,z)}+(v_i)^{(1,0)}$$

Now, sum on i (for 3 separate equations) :

- **Primitive Equation 2 with (i = x) :**

```
xEq2L = Eq2L2 /. {i → x} /. {Θy,x → Θx,y, Θx,z → Θz,x, Θz,y → Θy,z}
xEq2R = Eq2R2 /. {i → x}
```

$$\frac{\Theta_{x,x}\,\rho^{(0,x)}}{\rho} + v_x\,(v_x)^{(0,x)} + (\Theta_{x,x})^{(0,x)} + \frac{\Theta_{x,y}\,\rho^{(0,y)}}{\rho} + v_y\,(v_x)^{(0,y)} + (\Theta_{x,y})^{(0,y)} + \frac{\Theta_{z,x}\,\rho^{(0,z)}}{\rho} + v_z\,(v_x)^{(0,z)} + (\Theta_{z,x})^{(0,z)} + (v_x)^{(1,0)}$$

$$\frac{\lambda_x}{\rho}$$

```
LHS = Append[LHS, xEq2L];
RHS = Append[RHS, xEq2R];
```

- **Primitive Equation 3 with (i = y) :**

```
yEq3L = Eq2L2 /. {i → y} /. {Θy,x → Θx,y, Θx,z → Θz,x, Θz,y → Θy,z}
yEq3R = Eq2R2 /. {i → y}
```

$$\frac{\Theta_{x,y}\,\rho^{(0,x)}}{\rho} + v_x\,(v_y)^{(0,x)} + (\Theta_{x,y})^{(0,x)} + \frac{\Theta_{y,y}\,\rho^{(0,y)}}{\rho} + v_y\,(v_y)^{(0,y)} + (\Theta_{y,y})^{(0,y)} + \frac{\Theta_{y,z}\,\rho^{(0,z)}}{\rho} + v_z\,(v_y)^{(0,z)} + (\Theta_{y,z})^{(0,z)} + (v_y)^{(1,0)}$$

$$\frac{\lambda_y}{\rho}$$

```
LHS = Append[LHS, yEq3L];
RHS = Append[RHS, yEq3R];
```

- **Primitive Equation 4 with (i = z) :**

```
zEq4L = Eq2L2 /. {i → z} /. {Θy,x → Θx,y, Θx,z → Θz,x, Θz,y → Θy,z}
zEq4R = Eq2R2 /. {i → z}
```

$$\frac{\Theta_{z,x}\,\rho^{(0,x)}}{\rho} + v_x\,(v_z)^{(0,x)} + (\Theta_{z,x})^{(0,x)} + \frac{\Theta_{y,z}\,\rho^{(0,y)}}{\rho} + v_y\,(v_z)^{(0,y)} + (\Theta_{y,z})^{(0,y)} + \frac{\Theta_{z,z}\,\rho^{(0,z)}}{\rho} + v_z\,(v_z)^{(0,z)} + (\Theta_{z,z})^{(0,z)} + (v_z)^{(1,0)}$$

$$\frac{\lambda_z}{\rho}$$

```
LHS = Append[LHS, zEq4L];
RHS = Append[RHS, zEq4R];
```

Primitive Equations 5 - 10 :

$$\Theta_{jk}\,\partial_{x_k}(v_i) + \Theta_{ik}\,\partial_{x_k}(v_j) + v_k\,\partial_{x_k}(\Theta_{ij}) + \partial_t\,\partial_{x_k}(\Theta_{ij}) = \frac{\psi_{i,j} - v_j\,\lambda_i - v_i\,\lambda_j - \partial_{x_k}(h_{ijk})}{\rho}$$

$$\mathbf{Eq3L1} = \Theta_{j,k}\,(v_i)^{(0,k)} + \Theta_{i,k}\,(v_j)^{(0,k)} + v_k\,(\Theta_{i,j})^{(0,k)} + (\Theta_{i,j})^{(1,0)};$$

$$\mathbf{Eq3R1} = \frac{-v_j\,\lambda_i - v_i\,\lambda_j + \psi_{i,j} - (h_{ijk})^{(0,k)}}{\rho};$$

First sum on k :

```
Eq3L2 = Expand[Eq3L1 /. {
      Θ_{j,k} (v_i)^(0,k) → (Θ_{j,x} (v_i)^(0,x) + Θ_{j,y} (v_i)^(0,y) + Θ_{j,z} (v_i)^(0,z)),
      Θ_{i,k} (v_j)^(0,k) → (Θ_{i,x} (v_j)^(0,x) + Θ_{i,y} (v_j)^(0,y) + Θ_{i,z} (v_j)^(0,z)),
      v_k (Θ_{i,j})^(0,k) → (v_x (Θ_{i,j})^(0,x) + v_y (Θ_{i,j})^(0,y) + v_z (Θ_{i,j})^(0,z))
    }] /. {Θ_{y,x} → Θ_{x,y}, Θ_{x,z} → Θ_{z,x}, Θ_{z,y} → Θ_{y,z}}
Eq3R2 = Expand[Eq3R1 /. {
      (h_{ijk})^(0,k) → ((h_{ijx})^(0,x) + (h_{ijy})^(0,y) + (h_{ijz})^(0,z))
    }] /. {Θ_{y,x} → Θ_{x,y}, Θ_{x,z} → Θ_{z,x}, Θ_{z,y} → Θ_{y,z}}
```

$$\Theta_{j,x}(v_i)^{(0,x)} + \Theta_{i,x}(v_j)^{(0,x)} + v_x(\Theta_{i,j})^{(0,x)} + \Theta_{j,y}(v_i)^{(0,y)} + \Theta_{i,y}(v_j)^{(0,y)} + v_y(\Theta_{i,j})^{(0,y)} + \Theta_{j,z}(v_i)^{(0,z)} + \Theta_{i,z}(v_j)^{(0,z)} + v_z(\Theta_{i,j})^{(0,z)} + (\Theta_{i,j})^{(1,0)}$$

$$-\frac{v_j\lambda_i}{\rho} - \frac{v_i\lambda_j}{\rho} + \frac{\psi_{i,j}}{\rho} - \frac{(h_{ijx})^{(0,x)}}{\rho} - \frac{(h_{ijy})^{(0,y)}}{\rho} - \frac{(h_{ijz})^{(0,z)}}{\rho}$$

Now, sum on i and j (for 6 equations) :

- **Primitive Equation 5 with (i = x, j = x) :**

```
xxEq5L =  Eq3L2 /. {i → x,  j → x} /. {Θ_{y,x} → Θ_{x,y}, Θ_{x,z} → Θ_{z,x}, Θ_{z,y} → Θ_{y,z}}
xxEq5R = Eq3R2 /. {i → x,  j → x,  h_{ijx} → h_{xxx},  h_{ijy} → h_{xxy},  h_{ijz} → h_{xxz} }
```

$$2\,\Theta_{x,x}(v_x)^{(0,x)} + v_x(\Theta_{x,x})^{(0,x)} + 2\,\Theta_{x,y}(v_x)^{(0,y)} + v_y(\Theta_{x,x})^{(0,y)} + 2\,\Theta_{z,x}(v_x)^{(0,z)} + v_z(\Theta_{x,x})^{(0,z)} + (\Theta_{x,x})^{(1,0)}$$

$$-\frac{2\,v_x\lambda_x}{\rho} + \frac{\psi_{x,x}}{\rho} - \frac{(h_{xxx})^{(0,x)}}{\rho} - \frac{(h_{xxy})^{(0,y)}}{\rho} - \frac{(h_{xxz})^{(0,z)}}{\rho}$$

```
LHS = Append[LHS, xxEq5L];
RHS = Append[RHS, xxEq5R];
```

- **Primitive Equation 6 with (i = x, j = y) :**

```
xyEq6L =  Eq3L2 /. {i → x,  j → y} /. {Θ_{y,x} → Θ_{x,y}, Θ_{x,z} → Θ_{z,x}, Θ_{z,y} → Θ_{y,z}}
xyEq6R = Eq3R2 /. {i → x,  j → y,  h_{ijx} → h_{xyx},  h_{ijy} → h_{xyy},  h_{ijz} → h_{xyz} }
```

$$\Theta_{x,y}(v_x)^{(0,x)} + \Theta_{x,x}(v_y)^{(0,x)} + v_x(\Theta_{x,y})^{(0,x)} + \Theta_{y,y}(v_x)^{(0,y)} + \Theta_{x,y}(v_y)^{(0,y)} + v_y(\Theta_{x,y})^{(0,y)} + \Theta_{y,z}(v_x)^{(0,z)} + \Theta_{z,x}(v_y)^{(0,z)} + v_z(\Theta_{x,y})^{(0,z)} + (\Theta_{x,y})^{(1,0)}$$

$$-\frac{v_y\lambda_x}{\rho} - \frac{v_x\lambda_y}{\rho} + \frac{\psi_{x,y}}{\rho} - \frac{(h_{xyx})^{(0,x)}}{\rho} - \frac{(h_{xyy})^{(0,y)}}{\rho} - \frac{(h_{xyz})^{(0,z)}}{\rho}$$

```
LHS = Append[LHS, xyEq6L];
RHS = Append[RHS, xyEq6R];
```

- **Primitive Equation 7 with (i = y, j = y) :**

```
yyEq7L =  Eq3L2 /. {i → y,  j → y} /. {Θ_{y,x} → Θ_{x,y}, Θ_{x,z} → Θ_{z,x}, Θ_{z,y} → Θ_{y,z}}
yyEq7R = Eq3R2 /. {i → y,  j → y,  h_{ijx} → h_{yyx},  h_{ijy} → h_{yyy},  h_{ijz} → h_{yyz} }
```

$$2\,\Theta_{x,y}(v_y)^{(0,x)} + v_x(\Theta_{y,y})^{(0,x)} + 2\,\Theta_{y,y}(v_y)^{(0,y)} + v_y(\Theta_{y,y})^{(0,y)} + 2\,\Theta_{y,z}(v_y)^{(0,z)} + v_z(\Theta_{y,y})^{(0,z)} + (\Theta_{y,y})^{(1,0)}$$

$$-\frac{2\,v_y\lambda_y}{\rho} + \frac{\psi_{y,y}}{\rho} - \frac{(h_{yyx})^{(0,x)}}{\rho} - \frac{(h_{yyy})^{(0,y)}}{\rho} - \frac{(h_{yyz})^{(0,z)}}{\rho}$$

```
LHS = Append[LHS, yyEq7L];
RHS = Append[RHS, yyEq7R];
```

- **Primitive Equation 8 with (i = y, j = z) :**

$$\texttt{yzEq8L = Eq3L2 /. \{i} \to \texttt{y, j} \to \texttt{z\} /. } \{\Theta_{y,x} \to \Theta_{x,y},\ \Theta_{x,z} \to \Theta_{z,x},\ \Theta_{z,y} \to \Theta_{y,z}\}$$
$$\texttt{yzEq8R = Eq3R2 /. } \{i \to y,\ j \to z,\ h_{ijx} \to h_{yzx},\ h_{ijy} \to h_{yzy},\ h_{ijz} \to h_{yzz}\}$$

$$\Theta_{z,x}\left(v_y\right)^{(0,x)} + \Theta_{x,y}\left(v_z\right)^{(0,x)} + v_x\left(\Theta_{y,z}\right)^{(0,x)} + \Theta_{y,z}\left(v_y\right)^{(0,y)} + \Theta_{y,y}\left(v_z\right)^{(0,y)} + v_y\left(\Theta_{y,z}\right)^{(0,y)} + \Theta_{z,z}\left(v_y\right)^{(0,z)} + \Theta_{y,z}\left(v_z\right)^{(0,z)} + v_z\left(\Theta_{y,z}\right)^{(0,z)} + \left(\Theta_{y,z}\right)^{(1,0)}$$

$$-\frac{v_z\,\lambda_y}{\rho} - \frac{v_y\,\lambda_z}{\rho} + \frac{\psi_{y,z}}{\rho} - \frac{\left(h_{yzx}\right)^{(0,x)}}{\rho} - \frac{\left(h_{yzy}\right)^{(0,y)}}{\rho} - \frac{\left(h_{yzz}\right)^{(0,z)}}{\rho}$$

```
LHS = Append[LHS, yzEq8L];
RHS = Append[RHS, yzEq8R];
```

- **Primitive Equation 9 with (i = z, j = z) :**

$$\texttt{zzEq9L = Eq3L2 /. \{i} \to \texttt{z, j} \to \texttt{z\} /. } \{\Theta_{y,x} \to \Theta_{x,y},\ \Theta_{x,z} \to \Theta_{z,x},\ \Theta_{z,y} \to \Theta_{y,z}\}$$
$$\texttt{zzEq9R = Eq3R2 /. } \{i \to z,\ j \to z,\ h_{ijx} \to h_{zzx},\ h_{ijy} \to h_{zzy},\ h_{ijz} \to h_{zzz}\}$$

$$2\,\Theta_{z,x}\left(v_z\right)^{(0,x)} + v_x\left(\Theta_{z,z}\right)^{(0,x)} + 2\,\Theta_{y,z}\left(v_z\right)^{(0,y)} + v_y\left(\Theta_{z,z}\right)^{(0,y)} + 2\,\Theta_{z,z}\left(v_z\right)^{(0,z)} + v_z\left(\Theta_{z,z}\right)^{(0,z)} + \left(\Theta_{z,z}\right)^{(1,0)}$$

$$-\frac{2\,v_z\,\lambda_z}{\rho} + \frac{\psi_{z,z}}{\rho} - \frac{\left(h_{zzx}\right)^{(0,x)}}{\rho} - \frac{\left(h_{zzy}\right)^{(0,y)}}{\rho} - \frac{\left(h_{zzz}\right)^{(0,z)}}{\rho}$$

```
LHS = Append[LHS, zzEq9L];
RHS = Append[RHS, zzEq9R];
```

- **Primitive Equation 10 with (i = z, j = x) :**

$$\texttt{zxEq9L = Eq3L2 /. \{i} \to \texttt{z, j} \to \texttt{x\} /. } \{\Theta_{y,x} \to \Theta_{x,y},\ \Theta_{x,z} \to \Theta_{z,x},\ \Theta_{z,y} \to \Theta_{y,z}\}$$
$$\texttt{zxEq9R = Eq3R2 /. } \{i \to z,\ j \to x,\ h_{ijx} \to h_{zxx},\ h_{ijy} \to h_{zxy},\ h_{ijz} \to h_{zxz}\}$$

$$\Theta_{z,x}\left(v_x\right)^{(0,x)} + \Theta_{x,x}\left(v_z\right)^{(0,x)} + v_x\left(\Theta_{z,x}\right)^{(0,x)} + \Theta_{y,z}\left(v_x\right)^{(0,y)} + \Theta_{x,y}\left(v_z\right)^{(0,y)} + v_y\left(\Theta_{z,x}\right)^{(0,y)} + \Theta_{z,z}\left(v_x\right)^{(0,z)} + \Theta_{z,x}\left(v_z\right)^{(0,z)} + v_z\left(\Theta_{z,x}\right)^{(0,z)} + \left(\Theta_{z,x}\right)^{(1,0)}$$

$$-\frac{v_z\,\lambda_x}{\rho} - \frac{v_x\,\lambda_z}{\rho} + \frac{\psi_{z,x}}{\rho} - \frac{\left(h_{zxx}\right)^{(0,x)}}{\rho} - \frac{\left(h_{zxy}\right)^{(0,y)}}{\rho} - \frac{\left(h_{zxz}\right)^{(0,z)}}{\rho}$$

```
LHS = Append[LHS, zxEq9L];
RHS = Append[RHS, zxEq9R];
```

Primitive Equations 11 - 13 :

$$-\Theta_{ki}\,\Theta_{kj}\,\partial_{x_j}\rho - \frac{1}{2}\,\Theta_{ij}\,\Theta_{kk}\,\partial_{x_j}\,\rho + v_j\,\partial_{x_j}\left(q_i\right) + q_j\,\partial_{x_j}\left(v_i\right) + q_i\,\partial_{x_j}\left(v_j\right) - \frac{1}{2}\,\rho\,\Theta_{kk}\,\partial_{x_j}\left(\Theta_{ij}\right) - \rho\,\Theta_{ki}\,\partial_{x_j}\left(\Theta_{kj}\right) + \partial_t\left(q_i\right) ==$$
$$\frac{\Upsilon_{ikk}}{2} + \frac{1}{2}\,v_k^2\,\lambda_i + v_i\,v_k\,\lambda_k - \lambda_k\,\Theta_{ki} - \frac{1}{2}\,\lambda_i\,\Theta_{kk} - v_k\,\psi_{ik} - \frac{1}{2}\,v_i\,\psi_{kk} - \frac{1}{2}\,\partial_{x_j}\left(R_{ij}\right) - h_{ijk}\,\partial_{x_j}\left(v_k\right)$$

Eq4L1 =

$$-\Theta_{k,i}\,\Theta_{k,j}\,\rho^{(0,j)} - \frac{1}{2}\,\Theta_{i,j}\,\Theta_{k,k}\,\rho^{(0,j)} + v_j\left(q_i\right)^{(0,j)} + q_j\left(v_i\right)^{(0,j)} + q_i\left(v_j\right)^{(0,j)} - \frac{1}{2}\,\rho\,\Theta_{k,k}\left(\Theta_{i,j}\right)^{(0,j)} - \rho\,\Theta_{k,i}\left(\Theta_{k,j}\right)^{(0,j)} + \left(q_i\right)^{(1,0)};$$

$$\textbf{Eq4R1} = \frac{\Upsilon_{ikk}}{2} + \frac{1}{2}\,v_k^2\,\lambda_i + v_i\,v_k\,\lambda_k - \lambda_k\,\Theta_{k,i} - \frac{1}{2}\,\lambda_i\,\Theta_{k,k} - v_k\,\psi_{i,k} - \frac{1}{2}\,v_i\,\psi_{k,k} - \frac{1}{2}\left(R_{ij}\right)^{(0,j)} - h_{ijk}\left(v_k\right)^{(0,j)};$$

First, sum on k :

$$\begin{aligned}
&\mathbf{Eq4L2} = \mathbf{Expand}\Big[\mathbf{Eq4L1}\ /.\ \Big\{\\
&\qquad \Theta_{k,i}\,\Theta_{k,j} \to \left(\Theta_{x,i}\,\Theta_{x,j} + \Theta_{y,i}\,\Theta_{y,j} + \Theta_{z,i}\,\Theta_{z,j}\right),\\
&\qquad \Theta_{k,k} \to \left(\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z}\right),\\
&\qquad \Theta_{k,i}\left(\Theta_{k,j}\right)^{(0,j)} \to \left(\Theta_{x,i}\left(\Theta_{x,j}\right)^{(0,j)} + \Theta_{y,i}\left(\Theta_{y,j}\right)^{(0,j)} + \Theta_{z,i}\left(\Theta_{z,j}\right)^{(0,j)}\right)\\
&\quad \Big\}\Big];
\end{aligned}$$

$$\begin{aligned}
&\mathbf{Eq4R2} = \mathbf{Expand}\Big[\mathbf{Eq4R1}\ /.\ \Big\{\\
&\quad \Upsilon_{ikk} \to \left(\Upsilon_{ixx} + \Upsilon_{iyy} + \Upsilon_{izz}\right),\\
&\quad v_k^2 \to \left(v_x^2 + v_y^2 + v_z^2\right),\\
&\quad \Theta_{k,k} \to \left(\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z}\right),\\
&\quad \psi_{k,k} \to \left(\psi_{x,x} + \psi_{y,y} + \psi_{z,z}\right),\\
&\quad v_i\,v_k\,\lambda_k \to \left(v_i\,v_x\,\lambda_x + v_i\,v_y\,\lambda_y + v_i\,v_z\,\lambda_z\right),\\
&\quad v_k\,\psi_{i,k} \to \left(v_x\,\psi_{i,x} + v_y\,\psi_{i,y} + v_z\,\psi_{i,z}\right),\\
&\quad h_{ijk}\left(v_k\right)^{(0,j)} \to \left(h_{ijx}\left(v_x\right)^{(0,j)} + h_{ijy}\left(v_y\right)^{(0,j)} + h_{ijz}\left(v_z\right)^{(0,j)}\right)\\
&\Big\}\Big]
\end{aligned}$$

$$\frac{\Upsilon_{ixx}}{2} + \frac{\Upsilon_{iyy}}{2} + \frac{\Upsilon_{izz}}{2} + \frac{1}{2}v_x^2\,\lambda_i + \frac{1}{2}v_y^2\,\lambda_i + \frac{1}{2}v_z^2\,\lambda_i + v_i\,v_x\,\lambda_x + v_i\,v_y\,\lambda_y + v_i\,v_z\,\lambda_z - \lambda_k\,\Theta_{k,i} - \frac{1}{2}\lambda_i\,\Theta_{x,x} - \frac{1}{2}\lambda_i\,\Theta_{y,y} - \frac{1}{2}\lambda_i\,\Theta_{z,z} - v_x\,\psi_{i,x} - v_y\,\psi_{i,y} - v_z\,\psi_{i,z} - \frac{1}{2}v_i\,\psi_{x,x} - \frac{1}{2}v_i\,\psi_{y,y} - \frac{1}{2}v_i\,\psi_{z,z} - \frac{1}{2}\left(R_{ij}\right)^{(0,j)} - h_{ijx}\left(v_x\right)^{(0,j)} - h_{ijy}\left(v_y\right)^{(0,j)} - h_{ijz}\left(v_z\right)^{(0,j)}$$

Now, sum on j :

$$\begin{aligned}
&\mathbf{Eq4L3} = \mathbf{Eq4L2}\ /.\ \Big\{\\
&\quad \Theta_{x,j}\,\rho^{(0,j)} \to \left(\Theta_{x,x}\,\rho^{(0,x)} + \Theta_{x,y}\,\rho^{(0,y)} + \Theta_{x,z}\,\rho^{(0,z)}\right),\\
&\quad \Theta_{i,j}\,\Theta_{x,x}\,\rho^{(0,j)} \to \left(\Theta_{i,x}\,\Theta_{x,x}\,\rho^{(0,x)} + \Theta_{i,y}\,\Theta_{x,x}\,\rho^{(0,y)} + \Theta_{i,z}\,\Theta_{x,x}\,\rho^{(0,z)}\right),\\
&\quad \Theta_{y,j}\,\rho^{(0,j)} \to \left(\Theta_{y,x}\,\rho^{(0,x)} + \Theta_{y,y}\,\rho^{(0,y)} + \Theta_{y,z}\,\rho^{(0,z)}\right),\\
&\quad \Theta_{i,j}\,\Theta_{y,y}\,\rho^{(0,j)} \to \left(\Theta_{i,x}\,\Theta_{y,y}\,\rho^{(0,x)} + \Theta_{i,y}\,\Theta_{y,y}\,\rho^{(0,y)} + \Theta_{i,z}\,\Theta_{y,y}\,\rho^{(0,z)}\right),\\
&\quad \Theta_{z,j}\,\rho^{(0,j)} \to \left(\Theta_{z,x}\,\rho^{(0,x)} + \Theta_{z,y}\,\rho^{(0,y)} + \Theta_{z,z}\,\rho^{(0,z)}\right),\\
&\quad \Theta_{i,j}\,\Theta_{z,z}\,\rho^{(0,j)} \to \left(\Theta_{i,x}\,\Theta_{z,z}\,\rho^{(0,x)} + \Theta_{i,y}\,\Theta_{z,z}\,\rho^{(0,y)} + \Theta_{i,z}\,\Theta_{z,z}\,\rho^{(0,z)}\right),\\
&\quad v_j\left(q_i\right)^{(0,j)} \to \left(v_x\left(q_i\right)^{(0,x)} + v_y\left(q_i\right)^{(0,y)} + v_z\left(q_i\right)^{(0,z)}\right),\\
&\quad q_j\left(v_i\right)^{(0,j)} \to \left(q_x\left(v_i\right)^{(0,x)} + q_y\left(v_i\right)^{(0,y)} + q_z\left(v_i\right)^{(0,z)}\right),\\
&\quad \left(v_j\right)^{(0,j)} \to \left(\left(v_x\right)^{(0,x)} + \left(v_y\right)^{(0,y)} + \left(v_z\right)^{(0,z)}\right),\\
&\quad \left(\Theta_{i,j}\right)^{(0,j)} \to \left(\left(\Theta_{i,x}\right)^{(0,x)} + \left(\Theta_{i,y}\right)^{(0,y)} + \left(\Theta_{i,z}\right)^{(0,z)}\right),\\
&\quad \left(\Theta_{x,j}\right)^{(0,j)} \to \left(\left(\Theta_{x,x}\right)^{(0,x)} + \left(\Theta_{x,y}\right)^{(0,y)} + \left(\Theta_{x,z}\right)^{(0,z)}\right),\\
&\quad \left(\Theta_{y,j}\right)^{(0,j)} \to \left(\left(\Theta_{y,x}\right)^{(0,x)} + \left(\Theta_{y,y}\right)^{(0,y)} + \left(\Theta_{y,z}\right)^{(0,z)}\right),\\
&\quad \left(\Theta_{z,j}\right)^{(0,j)} \to \left(\left(\Theta_{z,x}\right)^{(0,x)} + \left(\Theta_{z,y}\right)^{(0,y)} + \left(\Theta_{z,z}\right)^{(0,z)}\right)\\
&\Big\};
\end{aligned}$$

$$\begin{aligned}
&\mathbf{Eq4R3} = \mathbf{Eq4R2}\ /.\ \Big\{\\
&\quad \left(R_{ij}\right)^{(0,j)} \to \left(\left(R_{ix}\right)^{(0,x)} + \left(R_{iy}\right)^{(0,y)} + \left(R_{iz}\right)^{(0,z)}\right),\\
&\quad h_{ijx}\left(v_x\right)^{(0,j)} \to \left(h_{ixx}\left(v_x\right)^{(0,x)} + h_{iyx}\left(v_x\right)^{(0,y)} + h_{izx}\left(v_x\right)^{(0,z)}\right),\\
&\quad h_{ijy}\left(v_y\right)^{(0,j)} \to \left(h_{ixy}\left(v_y\right)^{(0,x)} + h_{iyy}\left(v_y\right)^{(0,y)} + h_{izy}\left(v_y\right)^{(0,z)}\right),\\
&\quad h_{ijz}\left(v_z\right)^{(0,j)} \to \left(h_{ixz}\left(v_z\right)^{(0,x)} + h_{iyz}\left(v_z\right)^{(0,y)} + h_{izz}\left(v_z\right)^{(0,z)}\right)\\
&\Big\};
\end{aligned}$$

Now, sum on i (for 3 Equations) :

- **Primitive Equation 11 with (i = x) :**

$$\mathbf{xEq11L1 = Expand}\left[\mathbf{Eq4L3\ /.\ \{i \to x\}\ /.\ }\left\{\Theta_{y,x} \to \Theta_{x,y},\ \Theta_{x,z} \to \Theta_{z,x},\ \Theta_{z,y} \to \Theta_{y,z}\right\}\right];$$

$$\mathbf{xEq11R1 = Eq4R3\ /.\ }\{i \to x,\ R_{ix} \to R_{xx},\ R_{iy} \to R_{xy},\ R_{iz} \to R_{xz},\ h_{ixx} \to h_{xxx},\ h_{ixy} \to h_{xxy},\ h_{ixz} \to h_{xxz},\ h_{iyx} \to h_{xyx},\ h_{iyy} \to h_{xyy},\ h_{iyz} \to h_{xyz},\ h_{izx} \to h_{xzx},\ h_{izy} \to h_{xzy},\ h_{izz} \to h_{xzz}\};$$

$$\begin{aligned}\mathbf{xEq11L = Collect}\Big[\mathbf{xEq11L1},\ \Big\{&\rho^{(0,x)},\ (v_x)^{(0,x)},\ (v_y)^{(0,x)},\ (v_z)^{(0,x)},\ (\Theta_{x,x})^{(0,x)},\ (\Theta_{x,y})^{(0,x)},\\ &(\Theta_{y,y})^{(0,x)},\ (\Theta_{y,z})^{(0,x)},\ (\Theta_{z,z})^{(0,x)},\ (\Theta_{z,x})^{(0,x)},\ (q_x)^{(0,x)},\ (q_y)^{(0,x)},\ (q_z)^{(0,x)},\\ &\rho^{(0,y)},\ (v_x)^{(0,y)},\ (v_y)^{(0,y)},\ (v_z)^{(0,y)},\ (\Theta_{x,x})^{(0,y)},\ (\Theta_{x,y})^{(0,y)},\ (\Theta_{y,y})^{(0,y)},\\ &(\Theta_{y,z})^{(0,y)},\ (\Theta_{z,z})^{(0,y)},\ (\Theta_{z,x})^{(0,y)},\ (q_x)^{(0,y)},\ (q_y)^{(0,y)},\ (q_z)^{(0,y)},\\ &\rho^{(0,z)},\ (v_x)^{(0,z)},\ (v_y)^{(0,z)},\ (v_z)^{(0,z)},\ (\Theta_{x,x})^{(0,z)},\ (\Theta_{x,y})^{(0,z)},\ (\Theta_{y,y})^{(0,z)},\\ &(\Theta_{y,z})^{(0,z)},\ (\Theta_{z,z})^{(0,z)},\ (\Theta_{z,x})^{(0,z)},\ (q_x)^{(0,z)},\ (q_y)^{(0,z)},\ (q_z)^{(0,z)}\Big\}\Big]\end{aligned}$$

$$\mathbf{xEq11R = FullSimplify[xEq11R1]\ /.\ }\left\{\gamma_{ixx} + \gamma_{iyy} + \gamma_{izz} \to \gamma_{xxx} + \gamma_{xyy} + \gamma_{xzz}\right\}$$

$$\begin{aligned}&\left(-\frac{3}{2}\Theta_{x,x}^2 - \Theta_{x,y}^2 - \frac{1}{2}\Theta_{x,x}\Theta_{y,y} - \Theta_{z,x}^2 - \frac{1}{2}\Theta_{x,x}\Theta_{z,z}\right)\rho^{(0,x)} + v_x\,(q_x)^{(0,x)} + 2\,q_x\,(v_x)^{(0,x)} + \left(-\frac{3}{2}\rho\,\Theta_{x,x} - \frac{1}{2}\rho\,\Theta_{y,y} - \frac{1}{2}\rho\,\Theta_{z,z}\right)(\Theta_{x,x})^{(0,x)} -\\ &\rho\,\Theta_{x,y}\,(\Theta_{x,y})^{(0,x)} - \rho\,\Theta_{z,x}\,(\Theta_{z,x})^{(0,x)} + \left(-\frac{3}{2}\Theta_{x,x}\Theta_{x,y} - \frac{3}{2}\Theta_{x,y}\Theta_{y,y} - \Theta_{y,z}\Theta_{z,x} - \frac{1}{2}\Theta_{x,y}\Theta_{z,z}\right)\rho^{(0,y)} +\\ &v_y\,(q_x)^{(0,y)} + q_y\,(v_x)^{(0,y)} + q_x\,(v_y)^{(0,y)} + \left(-\frac{3}{2}\rho\,\Theta_{x,x} - \frac{1}{2}\rho\,\Theta_{y,y} - \frac{1}{2}\rho\,\Theta_{z,z}\right)(\Theta_{x,y})^{(0,y)} - \rho\,\Theta_{x,y}\,(\Theta_{y,y})^{(0,y)} -\\ &\rho\,\Theta_{z,x}\,(\Theta_{y,z})^{(0,y)} + \left(-\Theta_{x,y}\Theta_{y,z} - \frac{3}{2}\Theta_{x,x}\Theta_{z,x} - \frac{1}{2}\Theta_{y,y}\Theta_{z,x} - \frac{3}{2}\Theta_{z,x}\Theta_{z,z}\right)\rho^{(0,z)} + v_z\,(q_x)^{(0,z)} + q_z\,(v_x)^{(0,z)} +\\ &q_x\,(v_z)^{(0,z)} - \rho\,\Theta_{x,y}\,(\Theta_{y,z})^{(0,z)} + \left(-\frac{3}{2}\rho\,\Theta_{x,x} - \frac{1}{2}\rho\,\Theta_{y,y} - \frac{1}{2}\rho\,\Theta_{z,z}\right)(\Theta_{z,x})^{(0,z)} - \rho\,\Theta_{z,x}\,(\Theta_{z,z})^{(0,z)} + (q_x)^{(1,0)}\end{aligned}$$

$$\begin{aligned}\frac{1}{2}\Big(&\gamma_{xxx} + \gamma_{xyy} + \gamma_{xzz} + 3\,v_x^2\,\lambda_x + v_y^2\,\lambda_x - 2\,\lambda_k\,\Theta_{k,x} + \lambda_x\left(v_z^2 - \Theta_{x,x} - \Theta_{y,y} - \Theta_{z,z}\right) - 2\,v_y\,\psi_{x,y} - 2\,v_z\,\psi_{x,z} +\\ &v_x\left(2\,v_y\,\lambda_y + 2\,v_z\,\lambda_z - 3\,\psi_{x,x} - \psi_{y,y} - \psi_{z,z}\right) - (R_{xx})^{(0,x)} - 2\,h_{xxx}\,(v_x)^{(0,x)} - 2\,h_{xxy}\,(v_y)^{(0,x)} - 2\,h_{xxz}\,(v_z)^{(0,x)} - (R_{xy})^{(0,y)} -\\ &2\,h_{xyx}\,(v_x)^{(0,y)} - 2\,h_{xyy}\,(v_y)^{(0,y)} - 2\,h_{xyz}\,(v_z)^{(0,y)} - (R_{xz})^{(0,z)} - 2\,h_{xzx}\,(v_x)^{(0,z)} - 2\,h_{xzy}\,(v_y)^{(0,z)} - 2\,h_{xzz}\,(v_z)^{(0,z)}\Big)\end{aligned}$$

```
LHS = Append[LHS, xEq11L];
RHS = Append[RHS, xEq11R];
```

- **Primitive Equation 12 with (i = y) :**

```
yEq12L1 = Expand[Eq4L3 /. {i → y} /. {Θ_{y,x} → Θ_{x,y}, Θ_{x,z} → Θ_{z,x}, Θ_{z,y} → Θ_{y,z}}];
yEq12R1 = Eq4R3 /. {i → y, R_{ix} → R_{yx}, R_{iy} → R_{yy}, R_{iz} → R_{yz} , h_{ixx} → h_{yxx},
     h_{ixy} → h_{yxy}, h_{ixz} → h_{yxz}, h_{iyx} → h_{yyx}, h_{iyy} → h_{yyy}, h_{iyz} → h_{yyz}, h_{izx} → h_{yzx}, h_{izy} → h_{yzy}, h_{izz} → h_{yzz}};
yEq12L = Collect[yEq12L1, {
    ρ^(0,x), (v_x)^(0,x), (v_y)^(0,x), (v_z)^(0,x), (Θ_{x,x})^(0,x), (Θ_{x,y})^(0,x),
    (Θ_{y,y})^(0,x), (Θ_{y,z})^(0,x), (Θ_{z,z})^(0,x), (Θ_{z,x})^(0,x), (Θ_{x,y})^(0,x), (q_x)^(0,x), (q_y)^(0,x), (q_z)^(0,x),
    ρ^(0,y), (v_x)^(0,y), (v_y)^(0,y), (v_z)^(0,y), (Θ_{x,x})^(0,y), (Θ_{x,y})^(0,y), (Θ_{y,y})^(0,y), (Θ_{y,z})^(0,y),
    (Θ_{z,z})^(0,y), (Θ_{z,x})^(0,y), (Θ_{x,y})^(0,y), (q_x)^(0,y), (q_y)^(0,y), (q_z)^(0,y),
    ρ^(0,z), (v_x)^(0,z), (v_y)^(0,z), (v_z)^(0,z), (Θ_{x,x})^(0,z), (Θ_{x,y})^(0,z), (Θ_{y,y})^(0,z), (Θ_{y,z})^(0,z),
    (Θ_{z,z})^(0,z), (Θ_{z,x})^(0,z), (Θ_{x,y})^(0,z), (q_x)^(0,z), (q_y)^(0,z), (q_z)^(0,z)}]
yEq12R = FullSimplify[yEq12R1] /. {γ_{ixx} + γ_{iyy} + γ_{izz} → γ_{yxx} + γ_{yyy} + γ_{yzz}}
```

$$\left(-\frac{3}{2}\Theta_{x,x}\Theta_{x,y}-\frac{3}{2}\Theta_{x,y}\Theta_{y,y}-\Theta_{y,z}\Theta_{z,x}-\frac{1}{2}\Theta_{x,y}\Theta_{z,z}\right)\rho^{(0,x)}+v_x\left(q_y\right)^{(0,x)}+q_y\left(v_x\right)^{(0,x)}+q_x\left(v_y\right)^{(0,x)}-\rho\,\Theta_{x,y}\left(\Theta_{x,x}\right)^{(0,x)}+$$
$$\left(-\frac{1}{2}\rho\,\Theta_{x,x}-\frac{3}{2}\rho\,\Theta_{y,y}-\frac{1}{2}\rho\,\Theta_{z,z}\right)\left(\Theta_{x,y}\right)^{(0,x)}-\rho\,\Theta_{y,z}\left(\Theta_{z,x}\right)^{(0,x)}+\left(-\Theta_{x,y}^2-\frac{1}{2}\Theta_{x,x}\Theta_{y,y}-\frac{3\,\Theta_{y,y}^2}{2}-\Theta_{y,z}^2-\frac{1}{2}\Theta_{y,y}\Theta_{z,z}\right)\rho^{(0,y)}+$$
$$v_y\left(q_y\right)^{(0,y)}+2\,q_y\left(v_y\right)^{(0,y)}-\rho\,\Theta_{x,y}\left(\Theta_{x,y}\right)^{(0,y)}+\left(-\frac{1}{2}\rho\,\Theta_{x,x}-\frac{3}{2}\rho\,\Theta_{y,y}-\frac{1}{2}\rho\,\Theta_{z,z}\right)\left(\Theta_{y,y}\right)^{(0,y)}-$$
$$\rho\,\Theta_{y,z}\left(\Theta_{y,z}\right)^{(0,y)}+\left(-\frac{1}{2}\Theta_{x,x}\Theta_{y,z}-\frac{3}{2}\Theta_{y,y}\Theta_{y,z}-\Theta_{x,y}\Theta_{z,x}-\frac{3}{2}\Theta_{y,z}\Theta_{z,z}\right)\rho^{(0,z)}+v_z\left(q_y\right)^{(0,z)}+q_z\left(v_y\right)^{(0,z)}+$$
$$q_y\left(v_z\right)^{(0,z)}+\left(-\frac{1}{2}\rho\,\Theta_{x,x}-\frac{3}{2}\rho\,\Theta_{y,y}-\frac{1}{2}\rho\,\Theta_{z,z}\right)\left(\Theta_{y,z}\right)^{(0,z)}-\rho\,\Theta_{x,y}\left(\Theta_{z,x}\right)^{(0,z)}-\rho\,\Theta_{y,z}\left(\Theta_{z,z}\right)^{(0,z)}+\left(q_y\right)^{(1,0)}$$

$$\frac{1}{2}\Big(\gamma_{yxx}+\gamma_{yyy}+\gamma_{yzz}+v_x^2\lambda_y+3\,v_y^2\lambda_y-2\,\lambda_k\Theta_{k,y}+\lambda_y\left(v_z^2-\Theta_{x,x}-\Theta_{y,y}-\Theta_{z,z}\right)+2\,v_x\left(v_y\lambda_x-\psi_{y,x}\right)-2\,v_z\psi_{y,z}+$$
$$v_y\left(2\,v_z\lambda_z-\psi_{x,x}-3\,\psi_{y,y}-\psi_{z,z}\right)-\left(R_{yx}\right)^{(0,x)}-2\,h_{yxx}\left(v_x\right)^{(0,x)}-2\,h_{yxy}\left(v_y\right)^{(0,x)}-2\,h_{yxz}\left(v_z\right)^{(0,x)}-\left(R_{yy}\right)^{(0,y)}-$$
$$2\,h_{yyx}\left(v_x\right)^{(0,y)}-2\,h_{yyy}\left(v_y\right)^{(0,y)}-2\,h_{yyz}\left(v_z\right)^{(0,y)}-\left(R_{yz}\right)^{(0,z)}-2\,h_{yzx}\left(v_x\right)^{(0,z)}-2\,h_{yzy}\left(v_y\right)^{(0,z)}-2\,h_{yzz}\left(v_z\right)^{(0,z)}\Big)$$

```
LHS = Append[LHS, yEq12L];
RHS = Append[RHS, yEq12R];
```

- **Primitive Equation 13 with (i = z) :**

$$\mathtt{zEq13L1 = Expand}\left[\mathtt{Eq4L3\ /.\ \{i \to z\}\ /.\ }\left\{\Theta_{y,x} \to \Theta_{x,y},\ \Theta_{x,z} \to \Theta_{z,x},\ \Theta_{z,y} \to \Theta_{y,z}\right\}\right];$$

$$\mathtt{zEq13R1 = Eq4R3\ /.\ }\left\{i \to z,\ R_{ix} \to R_{zx},\ R_{iy} \to R_{zy},\ R_{iz} \to R_{zz},\ h_{ixx} \to h_{zxx},\right.$$
$$\left. h_{ixy} \to h_{zxy},\ h_{ixz} \to h_{zxz},\ h_{iyx} \to h_{zyx},\ h_{iyy} \to h_{zyy},\ h_{iyz} \to h_{zyz},\ h_{izx} \to h_{zzx},\ h_{izy} \to h_{zzy},\ h_{izz} \to h_{zzz}\right\};$$

$$\mathtt{zEq13L = Collect}\Big[\mathtt{zEq13L1},\ \Big\{$$
$$\rho^{(0,x)},\ (v_x)^{(0,x)},\ (v_y)^{(0,x)},\ (v_z)^{(0,x)},\ (\Theta_{x,x})^{(0,x)},\ (\Theta_{x,y})^{(0,x)},$$
$$(\Theta_{y,y})^{(0,x)},\ (\Theta_{y,z})^{(0,x)},\ (\Theta_{z,z})^{(0,x)},\ (\Theta_{z,x})^{(0,x)},\ (\Theta_{x,y})^{(0,x)},\ (q_x)^{(0,x)},\ (q_y)^{(0,x)},\ (q_z)^{(0,x)},$$
$$\rho^{(0,y)},\ (v_x)^{(0,y)},\ (v_y)^{(0,y)},\ (v_z)^{(0,y)},\ (\Theta_{x,x})^{(0,y)},\ (\Theta_{x,y})^{(0,y)},\ (\Theta_{y,y})^{(0,y)},\ (\Theta_{y,z})^{(0,y)},$$
$$(\Theta_{z,z})^{(0,y)},\ (\Theta_{z,x})^{(0,y)},\ (\Theta_{x,y})^{(0,y)},\ (q_x)^{(0,y)},\ (q_y)^{(0,y)},\ (q_z)^{(0,y)},$$
$$\rho^{(0,z)},\ (v_x)^{(0,z)},\ (v_y)^{(0,z)},\ (v_z)^{(0,z)},\ (\Theta_{x,x})^{(0,z)},\ (\Theta_{x,y})^{(0,z)},\ (\Theta_{y,y})^{(0,z)},\ (\Theta_{y,z})^{(0,z)},$$
$$(\Theta_{z,z})^{(0,z)},\ (\Theta_{z,x})^{(0,z)},\ (\Theta_{x,y})^{(0,z)},\ (q_x)^{(0,z)},\ (q_y)^{(0,z)},\ (q_z)^{(0,z)}\Big\}\Big]$$

$$\mathtt{zEq13R = FullSimplify[zEq13R1]\ /.\ }\left\{\gamma_{ixx} + \gamma_{iyy} + \gamma_{izz} \to \gamma_{zxx} + \gamma_{zyy} + \gamma_{zzz}\right\}$$

$$\left(-\Theta_{x,y}\,\Theta_{y,z} - \frac{3}{2}\,\Theta_{x,x}\,\Theta_{z,x} - \frac{1}{2}\,\Theta_{y,y}\,\Theta_{z,x} - \frac{3}{2}\,\Theta_{z,x}\,\Theta_{z,z}\right)\rho^{(0,x)} + v_x\,(q_z)^{(0,x)} + q_z\,(v_x)^{(0,x)} +$$
$$q_x\,(v_z)^{(0,x)} - \rho\,\Theta_{z,x}\,(\Theta_{x,x})^{(0,x)} - \rho\,\Theta_{y,z}\,(\Theta_{x,y})^{(0,x)} + \left(-\frac{1}{2}\,\rho\,\Theta_{x,x} - \frac{1}{2}\,\rho\,\Theta_{y,y} - \frac{3}{2}\,\rho\,\Theta_{z,z}\right)(\Theta_{z,x})^{(0,x)} +$$
$$\left(-\frac{1}{2}\,\Theta_{x,x}\,\Theta_{y,z} - \frac{3}{2}\,\Theta_{y,y}\,\Theta_{y,z} - \Theta_{x,y}\,\Theta_{z,x} - \frac{3}{2}\,\Theta_{y,z}\,\Theta_{z,z}\right)\rho^{(0,y)} + v_y\,(q_z)^{(0,y)} + q_z\,(v_y)^{(0,y)} + q_y\,(v_z)^{(0,y)} - \rho\,\Theta_{z,x}\,(\Theta_{x,y})^{(0,y)} -$$
$$\rho\,\Theta_{y,z}\,(\Theta_{y,y})^{(0,y)} + \left(-\frac{1}{2}\,\rho\,\Theta_{x,x} - \frac{1}{2}\,\rho\,\Theta_{y,y} - \frac{3}{2}\,\rho\,\Theta_{z,z}\right)(\Theta_{y,z})^{(0,y)} + \left(-\Theta_{y,z}^2 - \Theta_{z,x}^2 - \frac{1}{2}\,\Theta_{x,x}\,\Theta_{z,z} - \frac{1}{2}\,\Theta_{y,y}\,\Theta_{z,z} - \frac{3\,\Theta_{z,z}^2}{2}\right)\rho^{(0,z)} +$$
$$v_z\,(q_z)^{(0,z)} + 2\,q_z\,(v_z)^{(0,z)} - \rho\,\Theta_{y,z}\,(\Theta_{y,z})^{(0,z)} - \rho\,\Theta_{z,x}\,(\Theta_{z,x})^{(0,z)} + \left(-\frac{1}{2}\,\rho\,\Theta_{x,x} - \frac{1}{2}\,\rho\,\Theta_{y,y} - \frac{3}{2}\,\rho\,\Theta_{z,z}\right)(\Theta_{z,z})^{(0,z)} + (q_z)^{(1,0)}$$

$$\frac{1}{2}\,\Big(\gamma_{zxx} + \gamma_{zyy} + \gamma_{zzz} + v_x^2\,\lambda_z + v_y^2\,\lambda_z + 3\,v_z^2\,\lambda_z - 2\,\lambda_k\,\Theta_{k,z} - \lambda_z\,\Theta_{x,x} - \lambda_z\,\Theta_{y,y} - \lambda_z\,\Theta_{z,z} - v_z\,\psi_{x,x} - v_z\,\psi_{y,y} + 2\,v_x\,(v_z\,\lambda_x - \psi_{z,x}) +$$
$$2\,v_y\,(v_z\,\lambda_y - \psi_{z,y}) - 3\,v_z\,\psi_{z,z} - (R_{zx})^{(0,x)} - 2\,h_{zxx}\,(v_x)^{(0,x)} - 2\,h_{zxy}\,(v_y)^{(0,x)} - 2\,h_{zxz}\,(v_z)^{(0,x)} - (R_{zy})^{(0,y)} -$$
$$2\,h_{zyx}\,(v_x)^{(0,y)} - 2\,h_{zyy}\,(v_y)^{(0,y)} - 2\,h_{zyz}\,(v_z)^{(0,y)} - (R_{zz})^{(0,z)} - 2\,h_{zzx}\,(v_x)^{(0,z)} - 2\,h_{zzy}\,(v_y)^{(0,z)} - 2\,h_{zzz}\,(v_z)^{(0,z)}\Big)$$

```
LHS = Append[LHS, zEq13L ];
RHS = Append[RHS, zEq13R];
```

Sorting of Equations into Matricies :

```
EquationList = LHS;
VariableList = List[ρ^(0,x), (v_x)^(0,x), (v_y)^(0,x), (v_z)^(0,x), (Θ_x,x)^(0,x), (Θ_x,y)^(0,x),
    (Θ_y,y)^(0,x), (Θ_y,z)^(0,x), (Θ_z,z)^(0,x), (Θ_z,x)^(0,x), (Θ_x,y)^(0,x), (q_x)^(0,x), (q_y)^(0,x), (q_z)^(0,x),
    ρ^(0,y), (v_x)^(0,y), (v_y)^(0,y), (v_z)^(0,y), (Θ_x,x)^(0,y), (Θ_x,y)^(0,y), (Θ_y,y)^(0,y), (Θ_y,z)^(0,y),
    (Θ_z,z)^(0,y), (Θ_z,x)^(0,y), (Θ_x,y)^(0,y), (q_x)^(0,y), (q_y)^(0,y), (q_z)^(0,y),
    ρ^(0,z), (v_x)^(0,z), (v_y)^(0,z), (v_z)^(0,z), (Θ_x,x)^(0,z), (Θ_x,y)^(0,z), (Θ_y,y)^(0,z), (Θ_y,z)^(0,z),
    (Θ_z,z)^(0,z), (Θ_z,x)^(0,z), (Θ_x,y)^(0,z), (q_x)^(0,z), (q_y)^(0,z), (q_z)^(0,z)];
xDerivativeVariableList = List[
    ρ^(0,x), (v_x)^(0,x), (v_y)^(0,x), (v_z)^(0,x), (Θ_x,x)^(0,x), (Θ_x,y)^(0,x),
    (Θ_y,y)^(0,x), (Θ_y,z)^(0,x), (Θ_z,z)^(0,x), (Θ_z,x)^(0,x), (q_x)^(0,x), (q_y)^(0,x), (q_z)^(0,x)];
yDerivativeVariableList = List[
    ρ^(0,y), (v_x)^(0,y), (v_y)^(0,y), (v_z)^(0,y), (Θ_x,x)^(0,y), (Θ_x,y)^(0,y),
    (Θ_y,y)^(0,y), (Θ_y,z)^(0,y), (Θ_z,z)^(0,y), (Θ_z,x)^(0,y), (q_x)^(0,y), (q_y)^(0,y), (q_z)^(0,y)];
zDerivativeVariableList = List[
    ρ^(0,z), (v_x)^(0,z), (v_y)^(0,z), (v_z)^(0,z), (Θ_x,x)^(0,z), (Θ_x,y)^(0,z),
    (Θ_y,y)^(0,z), (Θ_y,z)^(0,z), (Θ_z,z)^(0,z), (Θ_z,x)^(0,z), (q_x)^(0,z), (q_y)^(0,z), (q_z)^(0,z)];
DerivativeVariableList = List[xDerivativeVariableList, yDerivativeVariableList, zDerivativeVariableList] ;

vl = DerivativeVariableList;
vl // MatrixForm;
el = EquationList;
len = Length[el];

Ap = ConstantArray[0, {len, 13}];
Bp = Ap;
Cp = Ap;
rawAp = Ap;
rawBp = Ap;
rawCp = Ap;
(* **************************************************************** *)
(* REMOVES DERIVATIVES FROM MATRICIES *)
Der = List[];
For[kVAR = 1, kVAR ≤ 3, {
   For[jVAR = 1, jVAR ≤ 13, {
      Der = Append[Der, Part[vl, kVAR, jVAR] → 1];
     }; jVAR++];
  }; kVAR++];
(* **************************************************************** *)(* MATRICIES *)
For[kVAR = 1, kVAR ≤ 3, {
   mOUT = ConstantArray[0, {len, 13}];
   For[iVAR = 1, iVAR ≤ len, {
     tempEq = Part[el, iVAR];
     EqLen = Length[tempEq];
     For[jVAR = 1, jVAR ≤ EqLen, {
       term = Part[tempEq, jVAR];
       For[hVAR = 1, hVAR ≤ 13, {
         If[Length[Position[term, Part[vl, kVAR, hVAR]]] > 0, {
            col = hVAR;
            mOUT[[iVAR, col]] = term}];
        }; hVAR++];
      }; jVAR++];
    }; iVAR++];
   If[kVAR == 1, {rawAp = mOUT}];
   If[kVAR == 2, {rawBp = mOUT}];
   If[kVAR == 3, {rawCp = mOUT}];
  }; kVAR++];

Ap = FullSimplify[rawAp] /. Der;
Bp = FullSimplify[rawBp] /. Der;
Cp = FullSimplify[rawCp] /. Der;
(* **************************************************************** *)
```

$$
\begin{aligned}
&\mathbf{Ap\ /.\ \Big\{}\\
&\quad \frac{1}{2}\left(-2\left(\Theta_{x,y}^2+\Theta_{z,x}^2\right)-\Theta_{x,x}\left(3\,\Theta_{x,x}+\Theta_{y,y}+\Theta_{z,z}\right)\right)\to\mu_1,\\
&\quad \frac{1}{2}\left(-2\,\Theta_{y,z}\,\Theta_{z,x}-\Theta_{x,y}\left(3\left(\Theta_{x,x}+\Theta_{y,y}\right)+\Theta_{z,z}\right)\right)\to\mu_2,\\
&\quad \frac{1}{2}\left(-2\,\Theta_{x,y}\,\Theta_{y,z}-\Theta_{z,x}\left(3\,\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)\to\mu_3,\\
&\quad -\frac{1}{2}\rho\left(3\,\Theta_{x,x}+\Theta_{y,y}+\Theta_{z,z}\right)\to\frac{\rho\,(2\,\Theta_{x,x}+\theta)}{-2},\\
&\quad -\frac{1}{2}\rho\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\to\frac{\rho\left(2\,\Theta_{y,y}+\theta\right)}{-2},\\
&\quad -\frac{1}{2}\rho\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\to\frac{\rho\,(2\,\Theta_{z,z}+\theta)}{-2}\\
&\mathbf{\Big\}\ //\ MatrixForm}
\end{aligned}
$$

$$
\begin{pmatrix}
v_x & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{x,x}}{\rho} & v_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{x,y}}{\rho} & 0 & v_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{z,x}}{\rho} & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 2\,\Theta_{x,x} & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \Theta_{x,y} & \Theta_{x,x} & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 2\,\Theta_{x,y} & 0 & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & \Theta_{z,x} & \Theta_{x,y} & 0 & 0 & 0 & v_x & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 2\,\Theta_{z,x} & 0 & 0 & 0 & 0 & v_x & 0 & 0 & 0 & 0\\
0 & \Theta_{z,x} & 0 & \Theta_{x,x} & 0 & 0 & 0 & 0 & 0 & v_x & 0 & 0 & 0\\
\mu_1 & 2\,q_x & 0 & 0 & -\frac{1}{2}\rho\,(\theta+2\,\Theta_{x,x}) & -\rho\,\Theta_{x,y} & 0 & 0 & 0 & -\rho\,\Theta_{z,x} & v_x & 0 & 0\\
\mu_2 & q_y & q_x & 0 & -\rho\,\Theta_{x,y} & -\frac{1}{2}\rho\left(\theta+2\,\Theta_{y,y}\right) & 0 & 0 & 0 & -\rho\,\Theta_{y,z} & 0 & v_x & 0\\
\mu_3 & q_z & 0 & q_x & -\rho\,\Theta_{z,x} & -\rho\,\Theta_{y,z} & 0 & 0 & 0 & -\frac{1}{2}\rho\,(\theta+2\,\Theta_{z,z}) & 0 & 0 & v_x
\end{pmatrix}
$$

Bp /. $\Big\{$

$$\frac{1}{2}\left(-2\,\Theta_{x,y}^2-2\,\Theta_{y,z}^2-\Theta_{y,y}\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)\to\mu_4,$$

$$\frac{1}{2}\left(-2\,\Theta_{y,z}\,\Theta_{z,x}-\Theta_{x,y}\left(3\left(\Theta_{x,x}+\Theta_{y,y}\right)+\Theta_{z,z}\right)\right)\to\mu_2,$$

$$\frac{1}{2}\left(-2\,\Theta_{x,y}\,\Theta_{z,x}-\Theta_{y,z}\left(\Theta_{x,x}+3\left(\Theta_{y,y}+\Theta_{z,z}\right)\right)\right)\to\mu_5,$$

$$-\frac{1}{2}\rho\left(3\,\Theta_{x,x}+\Theta_{y,y}+\Theta_{z,z}\right)\to\frac{\rho\,(2\,\Theta_{x,x}+\theta)}{-2},$$

$$-\frac{1}{2}\rho\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\to\frac{\rho\left(2\,\Theta_{y,y}+\theta\right)}{-2},$$

$$-\frac{1}{2}\rho\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\to\frac{\rho\,(2\,\Theta_{z,z}+\theta)}{-2}$$

$\Big\}$ **// MatrixForm**

$$\begin{pmatrix}
v_y & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{x,y}}{\rho} & v_y & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{y,y}}{\rho} & 0 & v_y & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{y,z}}{\rho} & 0 & 0 & v_y & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 2\,\Theta_{x,y} & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \Theta_{y,y} & \Theta_{x,y} & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 2\,\Theta_{y,y} & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & \Theta_{y,z} & \Theta_{y,y} & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 2\,\Theta_{y,z} & 0 & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0\\
0 & \Theta_{y,z} & 0 & \Theta_{x,y} & 0 & 0 & 0 & 0 & 0 & v_y & 0 & 0 & 0\\
\mu_2 & q_y & q_x & 0 & 0 & -\frac{1}{2}\rho\,(\theta+2\,\Theta_{x,x}) & -\rho\,\Theta_{x,y} & -\rho\,\Theta_{z,x} & 0 & 0 & v_y & 0 & 0\\
\mu_4 & 0 & 2\,q_y & 0 & 0 & -\rho\,\Theta_{x,y} & -\frac{1}{2}\rho\left(\theta+2\,\Theta_{y,y}\right) & -\rho\,\Theta_{y,z} & 0 & 0 & 0 & v_y & 0\\
\mu_5 & 0 & q_z & q_y & 0 & -\rho\,\Theta_{z,x} & -\rho\,\Theta_{y,z} & -\frac{1}{2}\rho\,(\theta+2\,\Theta_{z,z}) & 0 & 0 & 0 & 0 & v_y
\end{pmatrix}$$

$$
\texttt{Cp /. }\Big\{
\begin{aligned}
&\tfrac{1}{2}\left(-2\,\Theta_{y,z}^2-2\,\Theta_{z,x}^2-\Theta_{z,z}\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)\to\mu_6,\\
&\tfrac{1}{2}\left(-2\,\Theta_{x,y}\,\Theta_{y,z}-\Theta_{z,x}\left(3\,\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)\to\mu_3,\\
&\tfrac{1}{2}\left(-2\,\Theta_{x,y}\,\Theta_{z,x}-\Theta_{y,z}\left(\Theta_{x,x}+3\left(\Theta_{y,y}+\Theta_{z,z}\right)\right)\right)\to\mu_5,\\
&-\tfrac{1}{2}\,\rho\left(3\,\Theta_{x,x}+\Theta_{y,y}+\Theta_{z,z}\right)\to\frac{\rho\,(2\,\Theta_{x,x}+\theta)}{-2},\\
&-\tfrac{1}{2}\,\rho\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\to\frac{\rho\left(2\,\Theta_{y,y}+\theta\right)}{-2},\\
&-\tfrac{1}{2}\,\rho\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\to\frac{\rho\,(2\,\Theta_{z,z}+\theta)}{-2}
\end{aligned}
\Big\}\ \texttt{// MatrixForm}
$$

$$
\begin{pmatrix}
v_z & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{z,x}}{\rho} & v_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
\frac{\Theta_{y,z}}{\rho} & 0 & v_z & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
\frac{\Theta_{z,z}}{\rho} & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 2\,\Theta_{z,x} & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & \Theta_{y,z} & \Theta_{z,x} & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 2\,\Theta_{y,z} & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & \Theta_{z,z} & \Theta_{y,z} & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 2\,\Theta_{z,z} & 0 & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0\\
0 & \Theta_{z,z} & 0 & \Theta_{z,x} & 0 & 0 & 0 & 0 & 0 & v_z & 0 & 0 & 0\\
\mu_3 & q_z & 0 & q_x & 0 & 0 & 0 & -\rho\,\Theta_{x,y} & -\rho\,\Theta_{z,x} & -\frac{1}{2}\,\rho\,(\theta+2\,\Theta_{x,x}) & v_z & 0 & 0\\
\mu_5 & 0 & q_z & q_y & 0 & 0 & 0 & -\frac{1}{2}\,\rho\left(\theta+2\,\Theta_{y,y}\right) & -\rho\,\Theta_{y,z} & -\rho\,\Theta_{x,y} & 0 & v_z & 0\\
\mu_6 & 0 & 0 & 2\,q_z & 0 & 0 & 0 & -\rho\,\Theta_{y,z} & -\frac{1}{2}\,\rho\,(\theta+2\,\Theta_{z,z}) & -\rho\,\Theta_{z,x} & 0 & 0 & v_z
\end{pmatrix}
$$

Eigensystem Analysis :

- **Eigenvalues :**

```
ApEigVals = FullSimplify[Eigenvalues[Ap]];
ApEigVals // MatrixForm
```

$$
\begin{pmatrix}
v_x\\
v_x\\
v_x\\
v_x\\
v_x\\
v_x\\
v_x\\
v_x-\sqrt{\Theta_{x,x}}\\
v_x-\sqrt{\Theta_{x,x}}\\
v_x+\sqrt{\Theta_{x,x}}\\
v_x+\sqrt{\Theta_{x,x}}\\
v_x-\sqrt{3}\,\sqrt{\Theta_{x,x}}\\
v_x+\sqrt{3}\,\sqrt{\Theta_{x,x}}
\end{pmatrix}
$$

```
BpEigVals = FullSimplify[Eigenvalues[Bp]];
BpEigVals // MatrixForm
```

$$\begin{pmatrix} v_y \\ v_y \\ v_y \\ v_y \\ v_y \\ v_y \\ v_y \\ v_y - \sqrt{\Theta_{y,y}} \\ v_y - \sqrt{\Theta_{y,y}} \\ v_y + \sqrt{\Theta_{y,y}} \\ v_y + \sqrt{\Theta_{y,y}} \\ v_y - \sqrt{3}\ \sqrt{\Theta_{y,y}} \\ v_y + \sqrt{3}\ \sqrt{\Theta_{y,y}} \end{pmatrix}$$

```
CpEigVals = FullSimplify[Eigenvalues[Cp]];
CpEigVals // MatrixForm
```

$$\begin{pmatrix} v_z \\ v_z \\ v_z \\ v_z \\ v_z \\ v_z \\ v_z \\ v_z - \sqrt{\Theta_{z,z}} \\ v_z - \sqrt{\Theta_{z,z}} \\ v_z + \sqrt{\Theta_{z,z}} \\ v_z + \sqrt{\Theta_{z,z}} \\ v_z - \sqrt{3}\ \sqrt{\Theta_{z,z}} \\ v_z + \sqrt{3}\ \sqrt{\Theta_{z,z}} \end{pmatrix}$$

- **Matrix Coefficient Function :**

In these matricies, certain terms share certain similarities. These are simplified by use of a coefficient function $\overline{Q}$.

$$\overline{Q}_{\hat{e}}\ (c_1,\ c_2,\ c_3,\ c_4) = 2\, q_{\hat{e}} + c_1 * \rho \sqrt{\Theta_{\hat{e},\hat{e}}}\ \left(c_2\, \Theta_{x,x} + c_3\, \Theta_{y,y} + c_4\, \Theta_{z,z}\right)$$

For example :

$$\beta = \overline{Q}_x[1,\ 1,\ 3,\ 1] = 2\, q_x + \rho \sqrt{\Theta_{x,x}}\ \left(\Theta_{x,x} + 3\, \Theta_{y,y} + \Theta_{z,z}\right)$$

■ **Ap Eigenvectors :**

```
ApEigVecs = FullSimplify[Eigenvectors[Ap]];
ApEigVecs // MatrixForm
```

$$\left(\begin{array}{ccc}
0 & 0 & \\
0 & 0 & \\
0 & 0 & \\
-\frac{\rho}{\Theta_{z,x}} & 0 & \\
0 & 0 & \\
0 & 0 & \\
0 & 0 & \\
0 & 0 & \frac{}{4\,q_x^2+4\,\rho\,q_x\,\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x}+\right.} \\
0 & 0 & \frac{}{-\frac{2\,q_x}{\sqrt{\Theta_{x,}}}} \\
0 & 0 & \frac{}{4\,q_x^2-4\,\rho\,q_x\,\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x}+\right.} \\
0 & 0 & \frac{}{\frac{2\,q_x}{\sqrt{\Theta_{x,x}}}} \\
\frac{2\sqrt{3}\,\rho\,\Theta_{x,x}}{2\sqrt{3}\,q_z\,\Theta_{x,x}+2\sqrt{3}\,q_x\,\Theta_{z,x}+3\,\rho\,\sqrt{\Theta_{x,x}}\,\left(2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(3\,\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{6\,\Theta_{x,x}^{3/2}}{2\sqrt{3}\,q_z\,\Theta_{x,x}+2\sqrt{3}\,q_x\,\Theta_{z,x}+3\,\rho\,\sqrt{\Theta_{x,x}}\,\left(2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(3\,\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{}{2\sqrt{3}\,q_z\,\Theta_{x,x}+} \\
\frac{2\sqrt{3}\,\rho\,\Theta_{x,x}}{2\sqrt{3}\,q_z\,\Theta_{x,x}+2\sqrt{3}\,q_x\,\Theta_{z,x}-3\,\rho\,\sqrt{\Theta_{x,x}}\,\left(2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(3\,\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{6\,\Theta_{x,x}^{3/2}}{-2\sqrt{3}\,q_z\,\Theta_{x,x}-2\sqrt{3}\,q_x\,\Theta_{z,x}+3\,\rho\,\sqrt{\Theta_{x,x}}\,\left(2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(3\,\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{}{-2\sqrt{3}\,q_z\,\Theta_{x,x}-}
\end{array}\right.$$

```
ApR1 = Take[ApEigVecs, {1, 1}];
ApR2 = Take[ApEigVecs, {2, 2}];
ApR3 = Take[ApEigVecs, {3, 3}];
ApR4 = Take[ApEigVecs, {4, 4}];
ApR5 = Take[ApEigVecs, {5, 5}];
ApR6 = Take[ApEigVecs, {6, 6}];
ApR7 = Take[ApEigVecs, {7, 7}];
ApR8 = Take[ApEigVecs, {8, 8}];
ApR9 = Take[ApEigVecs, {9, 9}];
ApR10 = Take[ApEigVecs, {10, 10}];
ApR11 = Take[ApEigVecs, {11, 11}];
ApR12 = Take[ApEigVecs, {12, 12}];
ApR13 = Take[ApEigVecs, {13, 13}];
```

```
ApR1
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}}
```

```
ApR2
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0}}
```

```
ApR3
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0}}
```

```
ApR4
```

$$\left\{\left\{-\frac{\rho}{\Theta_{z,x}},\ 0,\ 0,\ 0,\ \frac{\Theta_{x,x}}{\Theta_{z,x}},\ \frac{\Theta_{x,y}}{\Theta_{z,x}},\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0\right\}\right\}$$

```
ApR5
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0}}
```

```
ApR6
```

```
{{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0}}
```

ApR7

```
{{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}}
```

$$\mathbf{FullSimplify}\Big[\mathbf{Transpose[ApR8]}\;*\;\Big(4\,q_x^2+4\,\rho\,q_x\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x}+2\,(\Theta_{y,y}+\Theta_{z,z})\right)+\rho^2\,\Theta_{x,x}\left(-4\,\Theta_{y,z}^2+(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z})\,(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z})\right)\Big)\Big]\;/.\;\Big\{$$

$$2\,q_x+\rho\sqrt{\Theta_{x,x}}\,(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z})\rightarrow\beta,$$

$$2\,q_x\sqrt{\Theta_{x,x}}+\rho\,\Theta_{x,x}\,(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z})\rightarrow\beta\sqrt{\Theta_{x,x}},$$

$$2\,q_x\,\Theta_{x,y}+\rho\sqrt{\Theta_{x,x}}\,\left(-2\,\Theta_{y,z}\,\Theta_{z,x}+\Theta_{x,y}\,(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z})\right)\rightarrow\beta\,\Theta_{x,y}-2\,\rho\sqrt{\Theta_{x,x}}\,\Theta_{y,z}\,\Theta_{z,x},$$

$$2\,\rho\,q_x\sqrt{\Theta_{x,x}}\,\Theta_{z,x}+\rho^2\,\Theta_{x,x}^2\,\Theta_{z,x}+\rho^2\,\Theta_{x,x}\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,(3\,\Theta_{y,y}+\Theta_{z,z})\right)\rightarrow\beta\,\rho\sqrt{\Theta_{x,x}}\,\Theta_{z,x}-2\,\rho^2\,\Theta_{x,x}\,\Theta_{x,y}\,\Theta_{y,z},$$

$$4\,q_x^2+4\,\rho\,q_x\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x}+2\,(\Theta_{y,y}+\Theta_{z,z})\right)+\rho^2\,\Theta_{x,x}\left(-4\,\Theta_{y,z}^2+(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z})\,(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z})\right)\rightarrow$$

$$\beta^2-4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2+2\,\rho\sqrt{\Theta_{x,x}}\,\beta\,(\Theta_{z,z}-\Theta_{y,y})$$

$$\Big\}\;//$$

MatrixForm

$$\begin{pmatrix}
0\\
0\\
4\,\rho\,\Theta_{x,x}\,\Theta_{y,z}\\
-2\,\beta\sqrt{\Theta_{x,x}}\\
0\\
-4\,\rho\,\Theta_{x,x}^{3/2}\,\Theta_{y,z}\\
-8\,\rho\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\,\Theta_{y,z}\\
2\left(\beta\,\Theta_{x,y}-2\,\rho\sqrt{\Theta_{x,x}}\,\Theta_{y,z}\,\Theta_{z,x}\right)\\
4\,\beta\,\Theta_{z,x}\\
2\,\beta\,\Theta_{x,x}\\
2\left(-2\,\rho^2\,\Theta_{x,x}\,\Theta_{x,y}\,\Theta_{y,z}+\beta\,\rho\sqrt{\Theta_{x,x}}\,\Theta_{z,x}\right)\\
0\\
\beta^2-4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2+2\,\beta\,\rho\sqrt{\Theta_{x,x}}\,(-\Theta_{y,y}+\Theta_{z,z})
\end{pmatrix}$$

$$\begin{aligned}
&\mathbf{FullSimplify}\Big[\mathbf{Transpose[ApR9]} * \\
&\quad \Big(4\, q_x^2 + 4\,\rho\, q_x \sqrt{\Theta_{x,x}}\ \left(\Theta_{x,x} + 2\left(\Theta_{y,y} + \Theta_{z,z}\right)\right) + \rho^2\,\Theta_{x,x}\left(-4\,\Theta_{y,z}^2 + \left(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}\right)\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right)\Big)\Big]\ /.\ \Big\{ \\
&\ 4\, q_x^2 + 4\,\rho\, q_x \sqrt{\Theta_{x,x}}\ \left(\Theta_{x,x} + 2\left(\Theta_{y,y} + \Theta_{z,z}\right)\right) + \rho^2\,\Theta_{x,x}\left(-4\,\Theta_{y,z}^2 + \left(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}\right)\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \to \\
&\quad \beta^2 - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2 + 2\,\rho\sqrt{\Theta_{x,x}}\ \beta\left(\Theta_{z,z} - \Theta_{y,y}\right), \\
&\ 2\, q_x + \rho\sqrt{\Theta_{x,x}}\ \left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right) \to \alpha, \\
&\ 2\, q_x \sqrt{\Theta_{x,x}} + \rho\,\Theta_{x,x}\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right) \to \alpha\sqrt{\Theta_{x,x}}\ , \\
&\ 2\, q_x\,\Theta_{z,x} + \rho\sqrt{\Theta_{x,x}}\ \left(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \to \alpha\,\Theta_{z,x} - 2\,\rho\sqrt{\Theta_{x,x}}\ \Theta_{x,y}\,\Theta_{y,z}, \\
&\ 2\,\rho\, q_x \sqrt{\Theta_{x,x}}\ \Theta_{x,y} + \rho^2\,\Theta_{x,x}\left(-2\,\Theta_{y,z}\,\Theta_{z,x} + \Theta_{x,y}\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \to \alpha\left(\rho\sqrt{\Theta_{x,x}}\ \Theta_{x,y}\right) - 2\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}\,\Theta_{z,x} \\
&\Big\}\ //\ \mathbf{MatrixForm}
\end{aligned}$$

$$\begin{pmatrix}
0 \\
0 \\
-2\,\alpha\sqrt{\Theta_{x,x}} \\
4\,\rho\,\Theta_{x,x}\,\Theta_{y,z} \\
0 \\
2\,\alpha\,\Theta_{x,x} \\
4\,\alpha\,\Theta_{x,y} \\
2\left(-2\,\rho\sqrt{\Theta_{x,x}}\ \Theta_{x,y}\,\Theta_{y,z} + \alpha\,\Theta_{z,x}\right) \\
-8\,\rho\sqrt{\Theta_{x,x}}\ \Theta_{y,z}\,\Theta_{z,x} \\
-4\,\rho\,\Theta_{x,x}^{3/2}\,\Theta_{y,z} \\
2\left(\alpha\,\rho\sqrt{\Theta_{x,x}}\ \Theta_{x,y} - 2\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}\,\Theta_{z,x}\right) \\
\beta^2 - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2 + 2\,\beta\,\rho\sqrt{\Theta_{x,x}}\ \left(-\Theta_{y,y} + \Theta_{z,z}\right) \\
0
\end{pmatrix}$$

```
FullSimplify[Transpose[ApR10] *
    (4 q_x^2 - 4 ρ q_x √(Θ_{x,x}) (Θ_{x,x} + 2 (Θ_{y,y} + Θ_{z,z})) + ρ^2 Θ_{x,x} (-4 Θ_{y,z}^2 + (Θ_{x,x} + 3 Θ_{y,y} + Θ_{z,z}) (Θ_{x,x} + Θ_{y,y} + 3 Θ_{z,z})))] /. {
  2 q_x - ρ √(Θ_{x,x}) (Θ_{x,x} + 3 Θ_{y,y} + Θ_{z,z}) → χ,
  4 q_x √(Θ_{x,x}) - 2 ρ Θ_{x,x} (Θ_{x,x} + 3 Θ_{y,y} + Θ_{z,z}) → 2 χ (√(Θ_{x,x})),
  4 q_x Θ_{x,y} - 2 ρ √(Θ_{x,x}) (-2 Θ_{y,z} Θ_{z,x} + Θ_{x,y} (Θ_{x,x} + 3 Θ_{y,y} + Θ_{z,z})) → 2 χ (Θ_{x,y}) + 4 ρ √(Θ_{x,x}) Θ_{y,z} Θ_{z,x},
  -2 Θ_{x,x} (-2 q_x + ρ √(Θ_{x,x}) (Θ_{x,x} + 3 Θ_{y,y} + Θ_{z,z})) → (2 Θ_{x,x}) χ,
  2 (-2 ρ q_x √(Θ_{x,x}) Θ_{z,x} + ρ^2 Θ_{x,x}^2 Θ_{z,x} + ρ^2 Θ_{x,x} (-2 Θ_{x,y} Θ_{y,z} + Θ_{z,x} (3 Θ_{y,y} + Θ_{z,z}))) → χ (-2 ρ √(Θ_{x,x}) Θ_{z,x}) - 4 ρ^2 Θ_{x,x} Θ_{x,y} Θ_{y,z},
  4 q_x^2 - 4 ρ q_x √(Θ_{x,x}) (Θ_{x,x} + 2 (Θ_{y,y} + Θ_{z,z})) + ρ^2 Θ_{x,x} (-4 Θ_{y,z}^2 + (Θ_{x,x} + 3 Θ_{y,y} + Θ_{z,z}) (Θ_{x,x} + Θ_{y,y} + 3 Θ_{z,z})) →
   χ^2 + 2 χ ρ √(Θ_{x,x}) (Θ_{y,y} - Θ_{z,z}) - 4 ρ^2 Θ_{x,x} Θ_{y,z}^2
 } //
 MatrixForm
```

$$\begin{pmatrix}
0 \\
0 \\
4\,\rho\,\Theta_{x,x}\,\Theta_{y,z} \\
2\,\chi\,\sqrt{\Theta_{x,x}} \\
0 \\
4\,\rho\,\Theta_{x,x}^{3/2}\,\Theta_{y,z} \\
8\,\rho\,\sqrt{\Theta_{x,x}}\;\Theta_{x,y}\,\Theta_{y,z} \\
2\,\chi\,\Theta_{x,y} + 4\,\rho\,\sqrt{\Theta_{x,x}}\;\Theta_{y,z}\,\Theta_{z,x} \\
4\,\chi\,\Theta_{z,x} \\
2\,\chi\,\Theta_{x,x} \\
-4\,\rho^2\,\Theta_{x,x}\,\Theta_{x,y}\,\Theta_{y,z} - 2\,\rho\,\chi\,\sqrt{\Theta_{x,x}}\;\Theta_{z,x} \\
0 \\
\chi^2 - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2 + 2\,\rho\,\chi\,\sqrt{\Theta_{x,x}}\;\left(\Theta_{y,y} - \Theta_{z,z}\right)
\end{pmatrix}$$

$$\mathbf{FullSimplify}\Big[\mathbf{Transpose[ApR11]} \ast$$
$$\left(4\,q_x^2 - 4\,\rho\,q_x\,\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x} + 2\,\left(\Theta_{y,y} + \Theta_{z,z}\right)\right) + \rho^2\,\Theta_{x,x}\,\left(-4\,\Theta_{y,z}^2 + \left(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}\right)\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right)\right)\Big]\ /.\ \Big\{$$
$$2\,q_x - \rho\,\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right) \to \sigma,$$
$$4\,q_x\,\sqrt{\Theta_{x,x}} - 2\,\rho\,\Theta_{x,x}\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right) \to \left(2\,\sqrt{\Theta_{x,x}}\right)\,\sigma,$$
$$-2\,\Theta_{x,x}\,\left(-2\,q_x + \rho\,\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \to \sigma\,(2\,\Theta_{x,x}),$$
$$-2\,\left(-2\,q_x\,\Theta_{z,x} + \rho\,\sqrt{\Theta_{x,x}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right)\right) \to \sigma\,(2\,\Theta_{z,x}) + 4\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\,\Theta_{y,z},$$
$$2\,\left(-2\,\rho\,q_x\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y} + \rho^2\,\Theta_{x,x}\,\left(-2\,\Theta_{y,z}\,\Theta_{z,x} + \Theta_{x,y}\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right)\right) \to \sigma\,\left(-2\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\right) - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}\,\Theta_{z,x},$$
$$4\,q_x^2 - 4\,\rho\,q_x\,\sqrt{\Theta_{x,x}}\,\left(\Theta_{x,x} + 2\,\left(\Theta_{y,y} + \Theta_{z,z}\right)\right) + \rho^2\,\Theta_{x,x}\,\left(-4\,\Theta_{y,z}^2 + \left(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}\right)\,\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \to$$
$$\sigma^2 + 2\,\rho\,\sqrt{\Theta_{x,x}}\,\sigma\,\left(\Theta_{z,z} - \Theta_{y,y}\right) - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2$$
$$\Big\}\ //$$
$$\mathbf{MatrixForm}$$

$$\begin{pmatrix}
0 \\
0 \\
2\,\sigma\,\sqrt{\Theta_{x,x}} \\
4\,\rho\,\Theta_{x,x}\,\Theta_{y,z} \\
0 \\
2\,\sigma\,\Theta_{x,x} \\
4\,\sigma\,\Theta_{x,y} \\
4\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\,\Theta_{y,z} + 2\,\sigma\,\Theta_{z,x} \\
8\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{y,z}\,\Theta_{z,x} \\
4\,\rho\,\Theta_{x,x}^{3/2}\,\Theta_{y,z} \\
-2\,\rho\,\sigma\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y} - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}\,\Theta_{z,x} \\
\sigma^2 - 4\,\rho^2\,\Theta_{x,x}\,\Theta_{y,z}^2 + 2\,\rho\,\sigma\,\sqrt{\Theta_{x,x}}\,\left(-\Theta_{y,y} + \Theta_{z,z}\right) \\
0
\end{pmatrix}$$

$$\texttt{FullSimplify}\Big[\texttt{Transpose[ApR12]} * \Big(2\sqrt{3}\, q_z\, \Theta_{x,x} + 2\sqrt{3}\, q_x\, \Theta_{z,x} + 3\,\rho\,\sqrt{\Theta_{x,x}}\,\big(2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(3\,\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z})\big)\Big)\ \texttt{/.}\ \Big\{$$

$$\sqrt{\Theta_{x,x}}\,\Big(4\sqrt{3}\, q_x\,\sqrt{\Theta_{x,x}} + 3\,\rho\,\big(2\,(\Theta_{x,y}^2 + \Theta_{z,x}^2) + \Theta_{x,x}\,(3\,\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\big)\Big) \to$$

$$2\sqrt{3}\,\Theta_{x,x}\,\bar{Q}_x\Big[\frac{3}{2\sqrt{3}},\ 3,\ 1,\ 1\Big] + 6\,\rho\,\sqrt{\Theta_{x,x}}\,\big(\Theta_{x,y}^2 + \Theta_{z,x}^2\big),$$

$$2\sqrt{3}\, q_y\,\Theta_{x,x} + 2\sqrt{3}\, q_x\,\Theta_{x,y} + 3\,\rho\,\sqrt{\Theta_{x,x}}\,\big(2\,\Theta_{y,z}\,\Theta_{z,x} + \Theta_{x,y}\,(3\,(\Theta_{x,x} + \Theta_{y,y}) + \Theta_{z,z})\big) \to$$

$$\bar{Q}_x\Big[\frac{3}{\sqrt{3}},\ 3,\ 3,\ 1\Big] * \sqrt{3}\,\Theta_{x,y} + 2\sqrt{3}\, q_y\,\Theta_{x,x} + 6\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{y,z}\,\Theta_{z,x},$$

$$2\sqrt{3}\, q_z\,\Theta_{x,x} + 2\sqrt{3}\, q_x\,\Theta_{z,x} + 3\,\rho\,\sqrt{\Theta_{x,x}}\,\big(2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(3\,\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z})\big) \to$$

$$\bar{Q}_x\Big[\frac{3}{\sqrt{3}},\ 3,\ 1,\ 3\Big] * \sqrt{3}\,\Theta_{z,x} + 2\sqrt{3}\, q_z\,\Theta_{x,x} + 6\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\,\Theta_{y,z}\Big\}\Big]\ \texttt{// MatrixForm}$$

$$\begin{pmatrix}
2\sqrt{3}\,\rho\,\Theta_{x,x} \\
-6\,\Theta_{x,x}^{3/2} \\
-6\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y} \\
-6\,\sqrt{\Theta_{x,x}}\,\Theta_{z,x} \\
4\sqrt{3}\,\Theta_{x,x}^2 \\
4\sqrt{3}\,\Theta_{x,x}\,\Theta_{x,y} \\
4\sqrt{3}\,\Theta_{x,y}^2 \\
4\sqrt{3}\,\Theta_{x,y}\,\Theta_{z,x} \\
4\sqrt{3}\,\Theta_{z,x}^2 \\
4\sqrt{3}\,\Theta_{x,x}\,\Theta_{z,x} \\
6\,\rho\,\sqrt{\Theta_{x,x}}\,\big(\Theta_{x,y}^2 + \Theta_{z,x}^2\big) + 2\sqrt{3}\,\Theta_{x,x}\,\bar{Q}_x\Big[\frac{\sqrt{3}}{2},\ 3,\ 1,\ 1\Big] \\
2\sqrt{3}\,q_y\,\Theta_{x,x} + 6\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{y,z}\,\Theta_{z,x} + \sqrt{3}\,\Theta_{x,y}\,\bar{Q}_x\big[\sqrt{3},\ 3,\ 3,\ 1\big] \\
2\sqrt{3}\,q_z\,\Theta_{x,x} + 6\,\rho\,\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\,\Theta_{y,z} + \sqrt{3}\,\Theta_{z,x}\,\bar{Q}_x\big[\sqrt{3},\ 3,\ 1,\ 3\big]
\end{pmatrix}$$

```
FullSimplify[Transpose[ApR13] * (2 √3 q_z Θ_{x,x} + 2 √3 q_x Θ_{z,x} - 3 ρ √Θ_{x,x} (2 Θ_{x,y} Θ_{y,z} + Θ_{z,x} (3 Θ_{x,x} + Θ_{y,y} + 3 Θ_{z,z})))] /. {
  √Θ_{x,x} (4 √3 q_x √Θ_{x,x} - 3 ρ (2 (Θ_{x,y}^2 + Θ_{z,x}^2) + Θ_{x,x} (3 Θ_{x,x} + Θ_{y,y} + Θ_{z,z}))) →
   Q̄_y[- 3/(2 √3), 3, 1, 1] * (2 √3 Θ_{x,x}) - 6 ρ √Θ_{x,x} (Θ_{x,y}^2 + Θ_{z,x}^2),
  2 √3 q_y Θ_{x,x} + 2 √3 q_x Θ_{x,y} - 3 ρ √Θ_{x,x} (2 Θ_{y,z} Θ_{z,x} + Θ_{x,y} (3 (Θ_{x,x} + Θ_{y,y}) + Θ_{z,z})) →
   Q̄_y[- 3/√3, 3, 3, 1] * (√3 Θ_{x,y}) + 2 √3 q_y Θ_{x,x} - 6 ρ √Θ_{x,x} Θ_{y,z} Θ_{z,x},
  2 √3 q_z Θ_{x,x} + 2 √3 q_x Θ_{z,x} - 3 ρ √Θ_{x,x} (2 Θ_{x,y} Θ_{y,z} + Θ_{z,x} (3 Θ_{x,x} + Θ_{y,y} + 3 Θ_{z,z})) →
   Q̄_y[- 3/√3, 3, 1, 3] * (√3 Θ_{z,x}) + 2 √3 q_z Θ_{x,x} - 6 ρ √Θ_{x,x} Θ_{x,y} Θ_{y,z}
 } //
 MatrixForm
```

$$\begin{pmatrix}
2\sqrt{3}\,\rho\,\Theta_{x,x} \\
6\,\Theta_{x,x}^{3/2} \\
6\sqrt{\Theta_{x,x}}\,\Theta_{x,y} \\
6\sqrt{\Theta_{x,x}}\,\Theta_{z,x} \\
4\sqrt{3}\,\Theta_{x,x}^{2} \\
4\sqrt{3}\,\Theta_{x,x}\Theta_{x,y} \\
4\sqrt{3}\,\Theta_{x,y}^{2} \\
4\sqrt{3}\,\Theta_{x,y}\Theta_{z,x} \\
4\sqrt{3}\,\Theta_{z,x}^{2} \\
4\sqrt{3}\,\Theta_{x,x}\Theta_{z,x} \\
-6\rho\sqrt{\Theta_{x,x}}\left(\Theta_{x,y}^{2}+\Theta_{z,x}^{2}\right)+2\sqrt{3}\,\Theta_{x,x}\,\overline{Q}_{y}\left[-\frac{\sqrt{3}}{2},\,3,\,1,\,1\right] \\
2\sqrt{3}\,q_{y}\Theta_{x,x}-6\rho\sqrt{\Theta_{x,x}}\,\Theta_{y,z}\Theta_{z,x}+\sqrt{3}\,\Theta_{x,y}\,\overline{Q}_{y}\left[-\sqrt{3},\,3,\,3,\,1\right] \\
2\sqrt{3}\,q_{z}\Theta_{x,x}-6\rho\sqrt{\Theta_{x,x}}\,\Theta_{x,y}\Theta_{y,z}+\sqrt{3}\,\Theta_{z,x}\,\overline{Q}_{y}\left[-\sqrt{3},\,3,\,1,\,3\right]
\end{pmatrix}$$

- **Bp Eigenvectors :**

```
BpEigVecs = FullSimplify[Eigenvectors[Bp]];
BpEigVecs // MatrixForm
```

$$\begin{pmatrix}
0 & 0 & \\
0 & 0 & \\
0 & 0 & \\
0 & 0 & \\
0 & 0 & \\
-\frac{\rho}{\Theta_{y,z}} & 0 & \\
0 & 0 & \\
0 & \frac{2\sqrt{\Theta_{y,y}}\ \Theta_{y,z}}{2\ q_y\ \Theta_{x,y}+\rho\sqrt{\Theta_{y,y}}\ (-2\ \Theta_{y,z}\ \Theta_{z,x}+\Theta_{x,y}\ (\Theta_{x,x}+\Theta_{y,y}+3\ \Theta_{z,z}))} & \\
0 & \frac{1}{\rho\left(-\Theta_{x,y}+\frac{2\rho\sqrt{\Theta_{y,y}}\ \Theta_{y,z}\ \Theta_{z,x}}{2\ q_y+\rho\sqrt{\Theta_{y,y}}\ (\Theta_{x,x}+\Theta_{y,y}+3\ \Theta_{z,z})}\right)} & \\
0 & \frac{2\sqrt{\Theta_{y,y}}\ \Theta_{y,z}}{-2\ q_y\ \Theta_{x,y}+\rho\sqrt{\Theta_{y,y}}\ (-2\ \Theta_{y,z}\ \Theta_{z,x}+\Theta_{x,y}\ (\Theta_{x,x}+\Theta_{y,y}+3\ \Theta_{z,z}))} & \\
0 & \frac{1}{\rho\left(-\Theta_{x,y}+\frac{2\rho\sqrt{\Theta_{y,y}}\ \Theta_{y,z}\ \Theta_{z,x}}{-2\ q_y+\rho\sqrt{\Theta_{y,y}}\ (\Theta_{x,x}+\Theta_{y,y}+3\ \Theta_{z,z})}\right)} & \\
\frac{2\sqrt{3}\ \rho\ \Theta_{y,y}}{2\sqrt{3}\ q_z\ \Theta_{y,y}+2\sqrt{3}\ q_y\ \Theta_{y,z}+3\ \rho\sqrt{\Theta_{y,y}}\ (2\ \Theta_{x,y}\ \Theta_{z,x}+\Theta_{y,z}\ (\Theta_{x,x}+3\ (\Theta_{y,y}+\Theta_{z,z})))} & -\frac{6\ \Theta_{x,y}\sqrt{\Theta_{y,y}}}{2\sqrt{3}\ q_z\ \Theta_{y,y}+2\sqrt{3}\ q_y\ \Theta_{y,z}+3\ \rho\sqrt{\Theta_{y,y}}\ (2\ \Theta_{x,y}\ \Theta_{z,x}+\Theta_{y,z}\ (\Theta_{x,x}+3\ (\Theta_{y,y}+\Theta_{z,z})))} & -\frac{}{2\sqrt{3}\ q_z\ \Theta_{y,y}+2\sqrt{3}} \\
\frac{2\sqrt{3}\ \rho\ \Theta_{y,y}}{2\sqrt{3}\ q_z\ \Theta_{y,y}+2\sqrt{3}\ q_y\ \Theta_{y,z}-3\ \rho\sqrt{\Theta_{y,y}}\ (2\ \Theta_{x,y}\ \Theta_{z,x}+\Theta_{y,z}\ (\Theta_{x,x}+3\ (\Theta_{y,y}+\Theta_{z,z})))} & -\frac{6\ \Theta_{x,y}\sqrt{\Theta_{y,y}}}{-2\sqrt{3}\ q_z\ \Theta_{y,y}-2\sqrt{3}\ q_y\ \Theta_{y,z}+3\ \rho\sqrt{\Theta_{y,y}}\ (2\ \Theta_{x,y}\ \Theta_{z,x}+\Theta_{y,z}\ (\Theta_{x,x}+3\ (\Theta_{y,y}+\Theta_{z,z})))} & -\frac{}{-2\sqrt{3}\ q_z\ \Theta_{y,y}-2\sqrt{3}}
\end{pmatrix}$$

```
BpR1 = Take[BpEigVecs, {1, 1}];
BpR2 = Take[BpEigVecs, {2, 2}];
BpR3 = Take[BpEigVecs, {3, 3}];
BpR4 = Take[BpEigVecs, {4, 4}];
BpR5 = Take[BpEigVecs, {5, 5}];
BpR6 = Take[BpEigVecs, {6, 6}];
BpR7 = Take[BpEigVecs, {7, 7}];
BpR8 = Take[BpEigVecs, {8, 8}];
BpR9 = Take[BpEigVecs, {9, 9}];
BpR10 = Take[BpEigVecs, {10, 10}];
BpR11 = Take[BpEigVecs, {11, 11}];
BpR12 = Take[BpEigVecs, {12, 12}];
BpR13 = Take[BpEigVecs, {13, 13}];
```

```
BpR1
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}}
```

```
BpR2
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0}}
```

```
BpR3
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0}}
```

```
BpR4
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0}}
```

```
BpR5
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0}}
```

```
BpR6
```

$$\left\{\left\{-\frac{\rho}{\Theta_{y,z}},\ 0,\ 0,\ 0,\ 0,\ \frac{\Theta_{x,y}}{\Theta_{y,z}},\ \frac{\Theta_{y,y}}{\Theta_{y,z}},\ 1,\ 0,\ 0,\ 0,\ 0,\ 0\right\}\right\}$$

BpR7

```
{{0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0}}
```

$$\mathbf{FullSimplify}\Big[\mathbf{Transpose[BpR8]} * \Big(2\, q_y\, \Theta_{x,y} + \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big)\Big)\Big] \;/.\; \Big\{$$
$$2\, q_y\, \Theta_{x,y} + \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big) \to \overline{Q}_y[1,\ 1,\ 1,\ 3] * \Theta_{x,y} - 2\, \rho \sqrt{\Theta_{y,y}}\, \Theta_{y,z}\, \Theta_{z,x},$$
$$-2\, q_y\, \Theta_{y,z} - \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{x,y}\, \Theta_{z,x} + \Theta_{y,z}\, (3\, \Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\big) \to \overline{Q}_y[1,\ 3,\ 1,\ 1] * (-\Theta_{y,z}) + 2\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x}$$
$$\Big\}\; //\ \mathbf{MatrixForm}$$

$$\begin{pmatrix} 0 \\ 2 \sqrt{\Theta_{y,y}}\, \Theta_{y,z} \\ 0 \\ -2\, \Theta_{x,y} \sqrt{\Theta_{y,y}} \\ -4\, \Theta_{x,y}\, \Theta_{y,z} \\ -2\, \Theta_{y,y}\, \Theta_{y,z} \\ 0 \\ 2\, \Theta_{x,y}\, \Theta_{y,y} \\ 4\, \Theta_{x,y}\, \Theta_{y,z} \\ 2\, (\Theta_{x,y}^2 - \Theta_{y,z}^2) \\ 2\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x} - \Theta_{y,z}\, \overline{Q}_y[1,\ 3,\ 1,\ 1] \\ 0 \\ -2\, \rho \sqrt{\Theta_{y,y}}\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, \overline{Q}_y[1,\ 1,\ 1,\ 3] \end{pmatrix}$$

$$\mathbf{FullSimplify}\Big[\mathbf{Transpose[BpR9]} * \Big(2\, \rho\, q_y\, \Theta_{x,y} \sqrt{\Theta_{y,y}} + \rho^2\, \Theta_{y,y}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big)\Big)\Big] \;/.\; \Big\{$$
$$2\, q_y + \rho \sqrt{\Theta_{y,y}}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z}) \to \alpha,$$
$$-2\, q_y \sqrt{\Theta_{y,y}} - \rho\, \Theta_{y,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z}) \to \alpha\, \big(-1 \sqrt{\Theta_{y,y}}\big),$$
$$2\, q_y\, \Theta_{y,y} + \rho\, \Theta_{y,y}^{3/2}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z}) \to \alpha * \Theta_{y,y},$$
$$2\, q_y\, \Theta_{y,z} + \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{x,y}\, \Theta_{z,x} + \Theta_{y,z}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big) \to \alpha * \Theta_{y,z} - 2\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x},$$
$$\frac{1}{2}\, \Big(4\, q_y^2 + 4\, \rho\, q_y \sqrt{\Theta_{y,y}}\, (2\, \Theta_{x,x} + \Theta_{y,y} + 2\, \Theta_{z,z}) + \rho^2\, \Theta_{y,y}\, \big(-4\, \Theta_{z,x}^2 + (3\, \Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big)\Big) \to$$
$$\frac{1}{2}\, (\alpha)^2 + (\alpha) * \big(\rho * (\Theta_{x,x} - \Theta_{z,z}) \sqrt{\Theta_{y,y}}\big) - 2\, \rho^2\, \Theta_{y,y}\, \Theta_{z,x}^2,$$
$$2\, \rho\, q_y\, \Theta_{x,y} \sqrt{\Theta_{y,y}} + \rho^2\, \Theta_{y,y}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big) \to (\alpha) * \big(\rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\big) - 2\, \rho^2\, \Theta_{y,y}\, \Theta_{y,z}\, \Theta_{z,x}$$
$$\Big\}\; //\ \mathbf{MatrixForm}$$

$$\begin{pmatrix} 0 \\ -\alpha \sqrt{\Theta_{y,y}} \\ 0 \\ 2\, \rho\, \Theta_{y,y}\, \Theta_{z,x} \\ 2\, \alpha\, \Theta_{x,y} \\ \alpha\, \Theta_{y,y} \\ 0 \\ -2\, \rho\, \Theta_{y,y}^{3/2}\, \Theta_{z,x} \\ -4\, \rho \sqrt{\Theta_{y,y}}\, \Theta_{y,z}\, \Theta_{z,x} \\ \alpha\, \Theta_{y,z} - 2\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x} \\ \frac{\alpha^2}{2} - 2\, \rho^2\, \Theta_{y,y}\, \Theta_{z,x}^2 + \alpha\, \rho \sqrt{\Theta_{y,y}}\, (\Theta_{x,x} - \Theta_{z,z}) \\ \alpha\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}} - 2\, \rho^2\, \Theta_{y,y}\, \Theta_{y,z}\, \Theta_{z,x} \\ 0 \end{pmatrix}$$

$$\texttt{FullSimplify}\Big[\Big(\texttt{Transpose[BpR10]} * \Big(2\, q_y\, \Theta_{x,y} - \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big)\Big)\Big)\Big]\ /.\ \Big\{$$

$$2\, q_y\, \Theta_{x,y} - \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big) \to \overline{Q}_y[-1,\ 1,\ 1,\ 3]\, \Theta_{x,y} + 2\, \rho \sqrt{\Theta_{y,y}}\, \Theta_{y,z}\, \Theta_{z,x},$$

$$-2\, q_y\, \Theta_{y,z} + \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{x,y}\, \Theta_{z,x} + \Theta_{y,z}\, (3\, \Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\big) \to \overline{Q}_y[-1,\ 3,\ 1,\ 1]\, (-\Theta_{y,z}) - 2\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x}$$

$$\Big\}\ //\ \texttt{MatrixForm}$$

$$\begin{pmatrix}
0 \\
-2 \sqrt{\Theta_{y,y}}\, \Theta_{y,z} \\
0 \\
2\, \Theta_{x,y} \sqrt{\Theta_{y,y}} \\
-4\, \Theta_{x,y}\, \Theta_{y,z} \\
-2\, \Theta_{y,y}\, \Theta_{y,z} \\
0 \\
2\, \Theta_{x,y}\, \Theta_{y,y} \\
4\, \Theta_{x,y}\, \Theta_{y,z} \\
2\, (\Theta_{x,y}^2 - \Theta_{y,z}^2) \\
-2\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x} - \Theta_{y,z}\, \overline{Q}_y[-1,\ 3,\ 1,\ 1] \\
0 \\
2\, \rho \sqrt{\Theta_{y,y}}\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, \overline{Q}_y[-1,\ 1,\ 1,\ 3]
\end{pmatrix}$$

$$\texttt{FullSimplify}\Big[\texttt{Transpose[BpR11]} * \Big(2\, \Big(-2\, \rho\, q_y\, \Theta_{x,y} \sqrt{\Theta_{y,y}} + \rho^2\, \Theta_{y,y}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big)\Big)\Big)\Big]\ /.\ \Big\{$$

$$-4\, \Theta_{x,y}\, \Big(-2\, q_y + \rho \sqrt{\Theta_{y,y}}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\Big) \to \beta\, (4\, \Theta_{x,y}),$$

$$-2\, \Theta_{y,y}\, \Big(-2\, q_y + \rho \sqrt{\Theta_{y,y}}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\Big) \to \beta\, (2\, \Theta_{y,y}),$$

$$4\, q_y \sqrt{\Theta_{y,y}} - 2\, \rho\, \Theta_{y,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z}) \to \beta\, \Big(2 \sqrt{\Theta_{y,y}}\Big),$$

$$4\, q_y\, \Theta_{y,z} - 2\, \rho \sqrt{\Theta_{y,y}}\, \big(-2\, \Theta_{x,y}\, \Theta_{z,x} + \Theta_{y,z}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big) \to \beta\, (2\, \Theta_{y,z}) + 4\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x},$$

$$4\, q_y^2 - 4\, \rho\, q_y \sqrt{\Theta_{y,y}}\, (2\, \Theta_{x,x} + \Theta_{y,y} + 2\, \Theta_{z,z}) + \rho^2\, \Theta_{y,y}\, \big(-4\, \Theta_{z,x}^2 + (3\, \Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big) \to$$

$$(\beta)^2 + (\beta) * \Big(2\, \rho\, (\,-\Theta_{x,x}\ +\ \Theta_{z,z})\, \sqrt{\Theta_{y,y}}\Big) - 4\, \rho^2\, \Theta_{y,y}\, \Theta_{z,x}^2,$$

$$2\, \Big(-2\, \rho\, q_y\, \Theta_{x,y} \sqrt{\Theta_{y,y}} + \rho^2\, \Theta_{y,y}\, \big(-2\, \Theta_{y,z}\, \Theta_{z,x} + \Theta_{x,y}\, (\Theta_{x,x} + \Theta_{y,y} + 3\, \Theta_{z,z})\big)\Big) \to (\beta) * \Big(-2\, \rho\ \Theta_{x,y} \sqrt{\Theta_{y,y}}\Big) - 4\, \rho^2\, \Theta_{y,y}\, \Theta_{y,z}\, \Theta_{z,x}$$

$$\Big\}\ //\ \texttt{MatrixForm}$$

$$\begin{pmatrix}
0 \\
2\, \beta \sqrt{\Theta_{y,y}} \\
0 \\
4\, \rho\, \Theta_{y,y}\, \Theta_{z,x} \\
4\, \beta\, \Theta_{x,y} \\
2\, \beta\, \Theta_{y,y} \\
0 \\
4\, \rho\, \Theta_{y,y}^{3/2}\, \Theta_{z,x} \\
8\, \rho \sqrt{\Theta_{y,y}}\, \Theta_{y,z}\, \Theta_{z,x} \\
2\, \beta\, \Theta_{y,z} + 4\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}}\, \Theta_{z,x} \\
\beta^2 - 4\, \rho^2\, \Theta_{y,y}\, \Theta_{z,x}^2 + 2\, \beta\, \rho \sqrt{\Theta_{y,y}}\, (-\Theta_{x,x} + \Theta_{z,z}) \\
-2\, \beta\, \rho\, \Theta_{x,y} \sqrt{\Theta_{y,y}} - 4\, \rho^2\, \Theta_{y,y}\, \Theta_{y,z}\, \Theta_{z,x} \\
0
\end{pmatrix}$$

$$\mathbf{FullSimplify}\Big[\mathbf{Transpose[BpR12]} * \Big(2\sqrt{3}\,q_z\,\Theta_{y,y} + 2\sqrt{3}\,q_y\,\Theta_{y,z} + 3\,\rho\,\sqrt{\Theta_{y,y}}\,\big(2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,\big(\Theta_{x,x} + 3\,\big(\Theta_{y,y} + \Theta_{z,z}\big)\big)\big)\Big)\Big] \;//\; \mathbf{MatrixForm}$$

$$\begin{pmatrix}
2\sqrt{3}\,\rho\,\Theta_{y,y} \\
-6\,\Theta_{x,y}\sqrt{\Theta_{y,y}} \\
-6\,\Theta_{y,y}^{3/2} \\
-6\sqrt{\Theta_{y,y}}\,\Theta_{y,z} \\
4\sqrt{3}\,\Theta_{x,y}^{2} \\
4\sqrt{3}\,\Theta_{x,y}\,\Theta_{y,y} \\
4\sqrt{3}\,\Theta_{y,y}^{2} \\
4\sqrt{3}\,\Theta_{y,y}\,\Theta_{y,z} \\
4\sqrt{3}\,\Theta_{y,z}^{2} \\
4\sqrt{3}\,\Theta_{x,y}\,\Theta_{y,z} \\
2\sqrt{3}\,q_y\,\Theta_{x,y} + 2\sqrt{3}\,q_x\,\Theta_{y,y} + 3\,\rho\sqrt{\Theta_{y,y}}\,\big(2\,\Theta_{y,z}\,\Theta_{z,x} + \Theta_{x,y}\,\big(3\,\big(\Theta_{x,x} + \Theta_{y,y}\big) + \Theta_{z,z}\big)\big) \\
\sqrt{\Theta_{y,y}}\,\big(4\sqrt{3}\,q_y\sqrt{\Theta_{y,y}} + 3\,\rho\,\big(2\,\Theta_{x,y}^{2} + 2\,\Theta_{y,z}^{2} + \Theta_{y,y}\,\big(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}\big)\big)\big) \\
2\sqrt{3}\,q_z\,\Theta_{y,y} + 2\sqrt{3}\,q_y\,\Theta_{y,z} + 3\,\rho\sqrt{\Theta_{y,y}}\,\big(2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,\big(\Theta_{x,x} + 3\,\big(\Theta_{y,y} + \Theta_{z,z}\big)\big)\big)
\end{pmatrix}$$

$$\mathbf{FullSimplify}\Big[\mathbf{Transpose[BpR13]} * \Big(2\sqrt{3}\,q_z\,\Theta_{y,y} + 2\sqrt{3}\,q_y\,\Theta_{y,z} - 3\,\rho\,\sqrt{\Theta_{y,y}}\,\big(2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,\big(\Theta_{x,x} + 3\,\big(\Theta_{y,y} + \Theta_{z,z}\big)\big)\big)\Big)\Big] \;//\; \mathbf{MatrixForm}$$

$$\begin{pmatrix}
2\sqrt{3}\,\rho\,\Theta_{y,y} \\
6\,\Theta_{x,y}\sqrt{\Theta_{y,y}} \\
6\,\Theta_{y,y}^{3/2} \\
6\sqrt{\Theta_{y,y}}\,\Theta_{y,z} \\
4\sqrt{3}\,\Theta_{x,y}^{2} \\
4\sqrt{3}\,\Theta_{x,y}\,\Theta_{y,y} \\
4\sqrt{3}\,\Theta_{y,y}^{2} \\
4\sqrt{3}\,\Theta_{y,y}\,\Theta_{y,z} \\
4\sqrt{3}\,\Theta_{y,z}^{2} \\
4\sqrt{3}\,\Theta_{x,y}\,\Theta_{y,z} \\
2\sqrt{3}\,q_y\,\Theta_{x,y} + 2\sqrt{3}\,q_x\,\Theta_{y,y} - 3\,\rho\sqrt{\Theta_{y,y}}\,\big(2\,\Theta_{y,z}\,\Theta_{z,x} + \Theta_{x,y}\,\big(3\,\big(\Theta_{x,x} + \Theta_{y,y}\big) + \Theta_{z,z}\big)\big) \\
\sqrt{\Theta_{y,y}}\,\big(4\sqrt{3}\,q_y\sqrt{\Theta_{y,y}} - 3\,\rho\,\big(2\,\Theta_{x,y}^{2} + 2\,\Theta_{y,z}^{2} + \Theta_{y,y}\,\big(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}\big)\big)\big) \\
2\sqrt{3}\,q_z\,\Theta_{y,y} + 2\sqrt{3}\,q_y\,\Theta_{y,z} - 3\,\rho\sqrt{\Theta_{y,y}}\,\big(2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,\big(\Theta_{x,x} + 3\,\big(\Theta_{y,y} + \Theta_{z,z}\big)\big)\big)
\end{pmatrix}$$

■ Cp Eigenvectors :

```
CpEigVecs = FullSimplify[Eigenvectors[Cp]];
CpEigVecs // MatrixForm
```

$$\begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
-\frac{\rho}{\Theta_{z,x}} & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & \frac{1}{\rho\left(-\Theta_{z,x}+\frac{2\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}}}{2\,q_z+\rho\sqrt{\Theta_{z,z}}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)}\right)} & \frac{2\,\Theta_{x,y}\sqrt{\Theta_{z,z}}}{2\,q_z\,\Theta_{z,x}+\rho\sqrt{\Theta_{z,z}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)} \\
0 & \frac{2\,\Theta_{y,z}\sqrt{\Theta_{z,z}}}{2\,q_z\,\Theta_{z,x}+\rho\sqrt{\Theta_{z,z}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)} & -\frac{2\,\Theta_{z,x}\sqrt{\Theta_{z,z}}}{2\,q_z\,\Theta_{z,x}+\rho\sqrt{\Theta_{z,z}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)} \\
0 & \frac{1}{\rho\left(-\Theta_{z,x}+\frac{2\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}}}{-2\,q_z+\rho\sqrt{\Theta_{z,z}}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)}\right)} & -\frac{2\,\Theta_{x,y}\sqrt{\Theta_{z,z}}}{2\,q_z\,\Theta_{z,x}-\rho\sqrt{\Theta_{z,z}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)} \\
0 & -\frac{2\,\Theta_{y,z}\sqrt{\Theta_{z,z}}}{2\,q_z\,\Theta_{z,x}-\rho\sqrt{\Theta_{z,z}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)} & -\frac{2\,\Theta_{z,x}\sqrt{\Theta_{z,z}}}{-2\,q_z\,\Theta_{z,x}+\rho\sqrt{\Theta_{z,z}}\,\left(-2\,\Theta_{x,y}\,\Theta_{y,z}+\Theta_{z,x}\,\left(\Theta_{x,x}+3\,\Theta_{y,y}+\Theta_{z,z}\right)\right)} \\
\frac{2\rho\sqrt{\Theta_{z,z}}}{4\,q_z\sqrt{\Theta_{z,z}}+\sqrt{3}\,\rho\left(2\,\Theta_{y,z}^2+2\,\Theta_{z,x}^2+\Theta_{z,z}\,\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{2\sqrt{3}\,\Theta_{z,x}}{4\,q_z\sqrt{\Theta_{z,z}}+\sqrt{3}\,\rho\left(2\,\Theta_{y,z}^2+2\,\Theta_{z,x}^2+\Theta_{z,z}\,\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{2\sqrt{3}\,\Theta_{y,z}}{4\,q_z\sqrt{\Theta_{z,z}}+\sqrt{3}\,\rho\left(2\,\Theta_{y,z}^2+2\,\Theta_{z,x}^2+\Theta_{z,z}\,\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} \\
-\frac{2\rho\sqrt{\Theta_{z,z}}}{-4\,q_z\sqrt{\Theta_{z,z}}+\sqrt{3}\,\rho\left(2\,\Theta_{y,z}^2+2\,\Theta_{z,x}^2+\Theta_{z,z}\,\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{2\sqrt{3}\,\Theta_{z,x}}{-4\,q_z\sqrt{\Theta_{z,z}}+\sqrt{3}\,\rho\left(2\,\Theta_{y,z}^2+2\,\Theta_{z,x}^2+\Theta_{z,z}\,\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right)} & -\frac{2\sqrt{3}\,\Theta_{y,z}}{-4\,q_z\sqrt{\Theta_{z,z}}+\sqrt{3}\,\rho\left(2\,\Theta_{y,z}^2+2\,\Theta_{z,x}^2+\Theta_{z,z}\,\left(\Theta_{x,x}+\Theta_{y,y}+3\,\Theta_{z,z}\right)\right.}
\end{pmatrix}$$

```
CpR1 = Take[CpEigVecs, {1, 1}];
CpR2 = Take[CpEigVecs, {2, 2}];
CpR3 = Take[CpEigVecs, {3, 3}];
CpR4 = Take[CpEigVecs, {4, 4}];
CpR5 = Take[CpEigVecs, {5, 5}];
CpR6 = Take[CpEigVecs, {6, 6}];
CpR7 = Take[CpEigVecs, {7, 7}];
CpR8 = Take[CpEigVecs, {8, 8}];
CpR9 = Take[CpEigVecs, {9, 9}];
CpR10 = Take[CpEigVecs, {10, 10}];
CpR11 = Take[CpEigVecs, {11, 11}];
CpR12 = Take[CpEigVecs, {12, 12}];
CpR13 = Take[CpEigVecs, {13, 13}];
```

```
CpR1
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}}
```

```
CpR2
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0}}
```

```
CpR3
```

```
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0}}
```

```
CpR4
```

$$\left\{\left\{-\frac{\rho}{\Theta_{z,x}},\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ \frac{\Theta_{y,z}}{\Theta_{z,x}},\ \frac{\Theta_{z,z}}{\Theta_{z,x}},\ 1,\ 0,\ 0,\ 0\right\}\right\}$$

```
CpR5
```

```
{{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}}
```

```
CpR6
```

```
{{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0}}
```

CpR7

```
{{0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0}}
```

$$\begin{aligned}
&\mathbf{FullSimplify}\Big[\mathtt{Transpose[CpR8]} * \Big(4\,\rho\,q_z\,\Theta_{z,x}\sqrt{\Theta_{z,z}} + 2\,\rho^2\,\Theta_{z,z}\,\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big)\Big)\Big] \;/.\; \Big\{\\
&\quad 2\,q_z + \rho\sqrt{\Theta_{z,z}}\;(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}) \to \beta,\\
&\quad -2\,\Big(2\,q_z\sqrt{\Theta_{z,z}} + \rho\,\Theta_{z,z}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\Big) \to (\beta) * \Big(-2\sqrt{\Theta_{z,z}}\Big),\\
&\quad 2\,\Big(2\,q_z\,\Theta_{y,z} + \rho\sqrt{\Theta_{z,z}}\;\big(-2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big)\Big) \to (\beta) * \big(2\,\Theta_{y,z}\big) - 4\,\rho\,\Theta_{x,y}\,\Theta_{z,x}\sqrt{\Theta_{z,z}},\\
&\quad 4\,q_z^2 + 4\,\rho\,q_z\sqrt{\Theta_{z,z}}\;\big(2\,(\Theta_{x,x} + \Theta_{y,y}) + \Theta_{z,z}\big) + \rho^2\,\Theta_{z,z}\,\big(-4\,\Theta_{x,y}^2 + (3\,\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\;(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to\\
&\qquad (\beta)^2 * (1) + (\beta) * \Big(2\,\rho\;\sqrt{\Theta_{z,z}}\;(\Theta_{x,x} - \Theta_{y,y})\Big) - 4\,\rho^2\,\Theta_{x,y}^2\,\Theta_{z,z},\\
&\quad 4\,\rho\,q_z\,\Theta_{z,x}\sqrt{\Theta_{z,z}} + 2\,\rho^2\,\Theta_{z,z}\,\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to (\beta) * \Big(2\,\rho\;\Theta_{z,x}\sqrt{\Theta_{z,z}}\Big) - 4\,\rho^2\,\Theta_{x,y}\,\Theta_{y,z}\,\Theta_{z,z}\\
&\Big\} \;//\; \mathbf{MatrixForm}
\end{aligned}$$

$$\begin{pmatrix}
0\\
-2\,\beta\sqrt{\Theta_{z,z}}\\
4\,\rho\,\Theta_{x,y}\,\Theta_{z,z}\\
0\\
4\,\beta\,\Theta_{z,x}\\
2\,\beta\,\Theta_{y,z} - 4\,\rho\,\Theta_{x,y}\,\Theta_{z,x}\sqrt{\Theta_{z,z}}\\
-8\,\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}}\\
-4\,\rho\,\Theta_{x,y}\,\Theta_{z,z}^{3/2}\\
0\\
2\,\beta\,\Theta_{z,z}\\
\beta^2 + 2\,\beta\,\rho\,\big(\Theta_{x,x} - \Theta_{y,y}\big)\sqrt{\Theta_{z,z}} - 4\,\rho^2\,\Theta_{x,y}^2\,\Theta_{z,z}\\
0\\
2\,\beta\,\rho\,\Theta_{z,x}\sqrt{\Theta_{z,z}} - 4\,\rho^2\,\Theta_{x,y}\,\Theta_{y,z}\,\Theta_{z,z}
\end{pmatrix}$$

$$\begin{aligned}
&\mathbf{FullSimplify}\Big[\mathtt{Transpose[CpR9]} * \Big(2\,q_z\,\Theta_{z,x} + \rho\sqrt{\Theta_{z,z}}\;\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big)\Big)\Big] \;/.\; \Big\{\\
&\quad 2\,q_z\,\Theta_{z,x} + \rho\sqrt{\Theta_{z,z}}\;\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to \overline{Q}_y\mathtt{[1,\ 1,\ 3,\ 1]} * (\Theta_{z,x}) - 2\,\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}},\\
&\quad -2\,q_z\,\Theta_{y,z} - \rho\sqrt{\Theta_{z,z}}\;\big(-2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,(3\,\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\big) \to \overline{Q}_y\mathtt{[1,\ 3,\ 1,\ 1]} * (-\,\Theta_{y,z}) + 2\,\rho\,\Theta_{x,y}\,\Theta_{z,x}\sqrt{\Theta_{z,z}}\\
&\Big\} \;//\; \mathbf{MatrixForm}
\end{aligned}$$

$$\begin{pmatrix}
0\\
2\,\Theta_{y,z}\sqrt{\Theta_{z,z}}\\
-2\,\Theta_{z,x}\sqrt{\Theta_{z,z}}\\
0\\
-4\,\Theta_{y,z}\,\Theta_{z,x}\\
-2\,\Theta_{y,z}^2 + 2\,\Theta_{z,x}^2\\
4\,\Theta_{y,z}\,\Theta_{z,x}\\
2\,\Theta_{z,x}\,\Theta_{z,z}\\
0\\
-2\,\Theta_{y,z}\,\Theta_{z,z}\\
2\,\rho\,\Theta_{x,y}\,\Theta_{z,x}\sqrt{\Theta_{z,z}} - \Theta_{y,z}\,\overline{Q}_y[1,\ 3,\ 1,\ 1]\\
-2\,\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}} + \Theta_{z,x}\,\overline{Q}_y[1,\ 1,\ 3,\ 1]\\
0
\end{pmatrix}$$

$$\texttt{FullSimplify}\Big[\texttt{Transpose[CpR10]} * \Big(-4\,\rho\,q_z\,\Theta_{z,x}\sqrt{\Theta_{z,z}} + 2\,\rho^2\,\Theta_{z,z}\,\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big)\Big)\Big] \;\texttt{/.}\; \Big\{$$

$$-4\,\Theta_{z,x}\Big(-2\,q_z + \rho\sqrt{\Theta_{z,z}}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\Big) \to (\chi) * (4\,\Theta_{z,x}),$$

$$4\,q_z\sqrt{\Theta_{z,z}} - 2\,\rho\,\Theta_{z,z}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z}) \to (\chi) * \Big(2\sqrt{\Theta_{z,z}}\Big),$$

$$4\,q_z\,\Theta_{y,z} - 2\,\rho\sqrt{\Theta_{z,z}}\,\big(-2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to (\chi) * \Big(2\sqrt{\Theta_{z,z}}\Big),$$

$$-2\,\Theta_{z,z}\Big(-2\,q_z + \rho\sqrt{\Theta_{z,z}}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\Big) \to (\chi) * (2\,\Theta_{z,z}),$$

$$4\,q_z^2 - 4\,\rho\,q_z\sqrt{\Theta_{z,z}}\,\big(2\,(\Theta_{x,x} + \Theta_{y,y}) + \Theta_{z,z}\big) + \rho^2\,\Theta_{z,z}\,\big(-4\,\Theta_{x,y}^2 + (3\,\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to$$

$$(\chi)^2 * (1) + (\chi) * 2\,\rho\sqrt{\Theta_{z,z}}\,(\Theta_{y,y} - \Theta_{x,x}) - 4\,\rho^2\,\Theta_{x,y}^2\,\Theta_{z,z},$$

$$-4\,\rho\,q_z\,\Theta_{z,x}\sqrt{\Theta_{z,z}} + 2\,\rho^2\,\Theta_{z,z}\,\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to (\chi) * \Big(-2\,\rho\,\Theta_{z,x}\sqrt{\Theta_{z,z}}\Big) - 4\,\rho^2\,\Theta_{x,y}\,\Theta_{y,z}\,\Theta_{z,z}$$

$$\Big\}\;\texttt{// MatrixForm}$$

$$\begin{pmatrix}
0 \\
2\,\chi\sqrt{\Theta_{z,z}} \\
4\,\rho\,\Theta_{x,y}\,\Theta_{z,z} \\
0 \\
4\,\chi\,\Theta_{z,x} \\
2\,\chi\sqrt{\Theta_{z,z}} \\
8\,\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}} \\
4\,\rho\,\Theta_{x,y}\,\Theta_{z,z}^{3/2} \\
0 \\
2\,\chi\,\Theta_{z,z} \\
\chi^2 + 2\,\rho\,\chi\,(-\Theta_{x,x} + \Theta_{y,y})\sqrt{\Theta_{z,z}} - 4\,\rho^2\,\Theta_{x,y}^2\,\Theta_{z,z} \\
0 \\
-2\,\rho\,\chi\,\Theta_{z,x}\sqrt{\Theta_{z,z}} - 4\,\rho^2\,\Theta_{x,y}\,\Theta_{y,z}\,\Theta_{z,z}
\end{pmatrix}$$

$$\texttt{FullSimplify}\Big[\texttt{Transpose[CpR11]} * \Big(2\,q_z\,\Theta_{z,x} - \rho\sqrt{\Theta_{z,z}}\,\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big)\Big)\Big] \;\texttt{/.}\; \Big\{$$

$$-2\,q_z\,\Theta_{y,z} + \rho\sqrt{\Theta_{z,z}}\,\big(-2\,\Theta_{x,y}\,\Theta_{z,x} + \Theta_{y,z}\,(3\,\Theta_{x,x} + \Theta_{y,y} + \Theta_{z,z})\big) \to \overline{Q}_y\texttt{[-1, 3, 1, 1]} * (-\Theta_{y,z}) - 2\,\rho\,\Theta_{x,y}\,\Theta_{z,x}\sqrt{\Theta_{z,z}},$$

$$2\,q_z\,\Theta_{z,x} - \rho\sqrt{\Theta_{z,z}}\,\big(-2\,\Theta_{x,y}\,\Theta_{y,z} + \Theta_{z,x}\,(\Theta_{x,x} + 3\,\Theta_{y,y} + \Theta_{z,z})\big) \to \overline{Q}_y\texttt{[-1, 1, 3, 1]} * (\Theta_{z,x}) + 2\,\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}}$$

$$\Big\}\;\texttt{// MatrixForm}$$

$$\begin{pmatrix}
0 \\
-2\,\Theta_{y,z}\sqrt{\Theta_{z,z}} \\
2\,\Theta_{z,x}\sqrt{\Theta_{z,z}} \\
0 \\
-4\,\Theta_{y,z}\,\Theta_{z,x} \\
-2\,\Theta_{y,z}^2 + 2\,\Theta_{z,x}^2 \\
4\,\Theta_{y,z}\,\Theta_{z,x} \\
2\,\Theta_{z,x}\,\Theta_{z,z} \\
0 \\
-2\,\Theta_{y,z}\,\Theta_{z,z} \\
-2\,\rho\,\Theta_{x,y}\,\Theta_{z,x}\sqrt{\Theta_{z,z}} - \Theta_{y,z}\,\overline{Q}_y[-1,\ 3,\ 1,\ 1] \\
2\,\rho\,\Theta_{x,y}\,\Theta_{y,z}\sqrt{\Theta_{z,z}} + \Theta_{z,x}\,\overline{Q}_y[-1,\ 1,\ 3,\ 1] \\
0
\end{pmatrix}$$

```
FullSimplify[Transpose[CpR12] * (Sqrt[Θz,z] (6 ρ Θy,z^2 + 6 ρ Θz,x^2 + 4 Sqrt[3] qz Sqrt[Θz,z] + 3 ρ Θz,z (Θx,x + Θy,y + 3 Θz,z)))] // MatrixForm
```

$$\begin{pmatrix} 2\sqrt{3}\,\rho\,\Theta_{z,z} \\ -6\,\Theta_{z,x}\sqrt{\Theta_{z,z}} \\ -6\,\Theta_{y,z}\sqrt{\Theta_{z,z}} \\ -6\,\Theta_{z,z}^{3/2} \\ 4\sqrt{3}\,\Theta_{z,x}^2 \\ 4\sqrt{3}\,\Theta_{y,z}\,\Theta_{z,x} \\ 4\sqrt{3}\,\Theta_{y,z}^2 \\ 4\sqrt{3}\,\Theta_{y,z}\,\Theta_{z,z} \\ 4\sqrt{3}\,\Theta_{z,z}^2 \\ 4\sqrt{3}\,\Theta_{z,x}\,\Theta_{z,z} \\ 2\sqrt{3}\,q_z\,\Theta_{z,x} + 3\rho\left(2\,\Theta_{x,y}\,\Theta_{y,z} + \left(3\,\Theta_{x,x} + \Theta_{y,y}\right)\Theta_{z,x}\right)\sqrt{\Theta_{z,z}} + 2\sqrt{3}\,q_x\,\Theta_{z,z} + 9\,\rho\,\Theta_{z,x}\,\Theta_{z,z}^{3/2} \\ 2\sqrt{3}\,q_z\,\Theta_{y,z} + 3\rho\left(\left(\Theta_{x,x} + 3\,\Theta_{y,y}\right)\Theta_{y,z} + 2\,\Theta_{x,y}\,\Theta_{z,x}\right)\sqrt{\Theta_{z,z}} + 2\sqrt{3}\,q_y\,\Theta_{z,z} + 9\,\rho\,\Theta_{y,z}\,\Theta_{z,z}^{3/2} \\ \sqrt{\Theta_{z,z}}\left(6\,\rho\,\Theta_{y,z}^2 + 6\,\rho\,\Theta_{z,x}^2 + 4\sqrt{3}\,q_z\sqrt{\Theta_{z,z}} + 3\,\rho\,\Theta_{z,z}\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \end{pmatrix}$$

```
FullSimplify[Transpose[CpR13] * (Sqrt[Θz,z] (6 ρ Θy,z^2 + 6 ρ Θz,x^2 - 4 Sqrt[3] qz Sqrt[Θz,z] + 3 ρ Θz,z (Θx,x + Θy,y + 3 Θz,z)))] // MatrixForm
```

$$\begin{pmatrix} -2\sqrt{3}\,\rho\,\Theta_{z,z} \\ -6\,\Theta_{z,x}\sqrt{\Theta_{z,z}} \\ -6\,\Theta_{y,z}\sqrt{\Theta_{z,z}} \\ -6\,\Theta_{z,z}^{3/2} \\ -4\sqrt{3}\,\Theta_{z,x}^2 \\ -4\sqrt{3}\,\Theta_{y,z}\,\Theta_{z,x} \\ -4\sqrt{3}\,\Theta_{y,z}^2 \\ -4\sqrt{3}\,\Theta_{y,z}\,\Theta_{z,z} \\ -4\sqrt{3}\,\Theta_{z,z}^2 \\ -4\sqrt{3}\,\Theta_{z,x}\,\Theta_{z,z} \\ -2\sqrt{3}\,q_z\,\Theta_{z,x} + 3\rho\left(2\,\Theta_{x,y}\,\Theta_{y,z} + \left(3\,\Theta_{x,x} + \Theta_{y,y}\right)\Theta_{z,x}\right)\sqrt{\Theta_{z,z}} - 2\sqrt{3}\,q_x\,\Theta_{z,z} + 9\,\rho\,\Theta_{z,x}\,\Theta_{z,z}^{3/2} \\ -2\sqrt{3}\,q_z\,\Theta_{y,z} + 3\rho\left(\left(\Theta_{x,x} + 3\,\Theta_{y,y}\right)\Theta_{y,z} + 2\,\Theta_{x,y}\,\Theta_{z,x}\right)\sqrt{\Theta_{z,z}} - 2\sqrt{3}\,q_y\,\Theta_{z,z} + 9\,\rho\,\Theta_{y,z}\,\Theta_{z,z}^{3/2} \\ \sqrt{\Theta_{z,z}}\left(6\,\rho\,\Theta_{y,z}^2 + 6\,\rho\,\Theta_{z,x}^2 - 4\sqrt{3}\,q_z\sqrt{\Theta_{z,z}} + 3\,\rho\,\Theta_{z,z}\left(\Theta_{x,x} + \Theta_{y,y} + 3\,\Theta_{z,z}\right)\right) \end{pmatrix}$$

Transformation Matrix M :

The transformation matrix M is defined as the Jacobian

$$M = \frac{\partial U}{\partial V}$$

Where V are the primitive variables and U are the conserved variables :

$$\mathbf{V} = \begin{pmatrix} \rho \\ v_x \\ v_y \\ v_z \\ \Theta_{xx} \\ \Theta_{xy} \\ \Theta_{yy} \\ \Theta_{yz} \\ \Theta_{zz} \\ \Theta_{zx} \\ q_x \\ q_y \\ q_z \end{pmatrix};$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho * v_x \\ \rho * v_y \\ \rho * v_z \\ \rho * \Theta_{xx} + \rho * v_x * v_x \\ \rho * \Theta_{xy} + \rho * v_x * v_y \\ \rho * \Theta_{yy} + \rho * v_y * v_y \\ \rho * \Theta_{yz} + \rho * v_y * v_z \\ \rho * \Theta_{zz} + \rho * v_z * v_z \\ \rho * \Theta_{zx} + \rho * v_z * v_x \\ \rho * v_x * \left(v_x^2 + v_y^2 + v_z^2\right) + \rho * v_x * \left(\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right) + 2 * \rho * \left(v_x * \Theta_{xx} + v_y * \Theta_{xy} + v_z * \Theta_{xz}\right) + 2\, q_x \\ \rho * v_y * \left(v_x^2 + v_y^2 + v_z^2\right) + \rho * v_y * \left(\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right) + 2\, \rho * \left(v_x * \Theta_{xy} + v_y * \Theta_{yy} + v_z * \Theta_{yz}\right) + 2\, q_y \\ \rho * v_z * \left(v_x^2 + v_y^2 + v_z^2\right) + \rho * v_z * \left(\Theta_{xx} + \Theta_{yy} + \Theta_{zz}\right) + 2\, \rho * \left(v_x * \Theta_{xz} + v_y * \Theta_{yz} + v_z * \Theta_{zz}\right) + 2\, q_z \end{pmatrix};$$

```
M = ConstantArray[0, {13, 13}];
For[iVAR = 1, iVAR ≤ 13, {
   For[jVAR = 1, jVAR ≤ 13, {
      term = ∂Part[V, jVAR, 1] (Part[U, iVAR, 1]);
      M[[iVAR, jVAR]] = term;
     }; jVAR++];
  }; iVAR++];
```

$$\mathbf{M}\ /.\ \left\{v_x^2 + v_y^2 + v_z^2 \to v^2,\ \Theta_{xx} + \Theta_{yy} + \Theta_{zz} \to \frac{\theta}{3}\right\}\ /.\ \Big\{$$

$$v^2\, v_x + \frac{\theta\, v_x}{3} + 2\,\left(v_x\, \Theta_{xx} + v_y\, \Theta_{xy} + v_z\, \Theta_{xz}\right) \to \zeta[v_x],$$

$$v^2\, v_y + \frac{\theta\, v_y}{3} + 2\,\left(v_x\, \Theta_{xy} + v_y\, \Theta_{yy} + v_z\, \Theta_{yz}\right) \to \zeta[v_y],$$

$$v^2\, v_z + \frac{\theta\, v_z}{3} + 2\,\left(v_x\, \Theta_{xz} + v_y\, \Theta_{yz} + v_z\, \Theta_{zz}\right) \to \zeta[v_z],$$

$$v^2\, \rho + \frac{\theta\, \rho}{3} + 2\, \rho\, v_x^2 + 2\, \rho\, \Theta_{xx} \to \eta[\mathbf{x}],$$

$$v^2\, \rho + \frac{\theta\, \rho}{3} + 2\, \rho\, v_y^2 + 2\, \rho\, \Theta_{yy} \to \eta[\mathbf{y}],$$

$$v^2\, \rho + \frac{\theta\, \rho}{3} + 2\, \rho\, v_z^2 + 2\, \rho\, \Theta_{zz} \to \eta[\mathbf{z}],$$

$$2\, \rho\, v_x\, v_y + 2\, \rho\, \Theta_{xy} \to \mathcal{F}_1,$$

$$2\, \rho\, v_x\, v_z + 2\, \rho\, \Theta_{xz} \to \mathcal{F}_2,$$

$$2\, \rho\, v_y\, v_z + 2\, \rho\, \Theta_{yz} \to \mathcal{F}_3$$

$$\Big\}\ //\ \mathbf{MatrixForm}$$

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_x & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_y & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_z & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_x^2 + \Theta_{xx} & 2\rho v_x & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_x v_y + \Theta_{xy} & \rho v_y & \rho v_x & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_y^2 + \Theta_{yy} & 0 & 2\rho v_y & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\
v_y v_z + \Theta_{yz} & 0 & \rho v_z & \rho v_y & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 \\
v_z^2 + \Theta_{zz} & 0 & 0 & 2\rho v_z & 0 & 0 & 0 & 0 & \rho & 0 & 0 & 0 & 0 \\
v_x v_z + \Theta_{zx} & \rho v_z & 0 & \rho v_x & 0 & 0 & 0 & 0 & 0 & \rho & 0 & 0 & 0 \\
\zeta[v_x] & \eta[x] & F_1 & F_2 & 3\rho v_x & 2\rho v_y & \rho v_x & 0 & \rho v_x & 0 & 2 & 0 & 0 \\
\zeta[v_y] & F_1 & \eta[y] & F_3 & \rho v_y & 2\rho v_x & 3\rho v_y & 2\rho v_z & \rho v_y & 0 & 0 & 2 & 0 \\
\zeta[v_z] & F_2 & F_3 & \eta[z] & \rho v_z & 0 & \rho v_z & 2\rho v_y & 3\rho v_z & 0 & 0 & 0 & 2
\end{pmatrix}$$

B.3 B_p and C_p Matrices and Eigenvectors

B.3.1 B_p Matrix

The B_p matrix is presented here with the following substitutions
$B_p =$

$$\begin{pmatrix}
v_y & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{xy}}{\rho} & v_y & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{yy}}{\rho} & 0 & v_y & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{yz}}{\rho} & 0 & 0 & v_y & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 2\Theta_{xy} & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \Theta_{yy} & \Theta_{xy} & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2\Theta_{yy} & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Theta_{yz} & \Theta_{yy} & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2\Theta_{yz} & 0 & 0 & 0 & 0 & v_y & 0 & 0 & 0 & 0 \\
0 & \Theta_{yz} & 0 & \Theta_{xy} & 0 & 0 & 0 & 0 & 0 & v_y & 0 & 0 & 0 \\
\mu_2 & q_y & q_x & 0 & 0 & -\frac{1}{2}\rho\left(\theta+2\Theta_{xx}\right) & -\rho\Theta_{xy} & -\rho\Theta_{zx} & 0 & 0 & v_y & 0 & 0 \\
\mu_4 & 0 & 2q_y & 0 & 0 & -\rho\Theta_{xy} & -\frac{1}{2}\rho\left(\theta+2\Theta_{yy}\right) & -\rho\Theta_{yz} & 0 & 0 & 0 & v_y & 0 \\
\mu_5 & 0 & q_z & q_y & 0 & -\rho\Theta_{zx} & -\rho\Theta_{yz} & -\frac{1}{2}\rho\left(\theta+2\Theta_{zz}\right) & 0 & 0 & 0 & 0 & v_y
\end{pmatrix}, \tag{B.1}$$

where,

$$\mu_2 = \frac{1}{2}\left(-2\Theta_{yz}\Theta_{zx} - \Theta_{xy}\left(3\left(\Theta_{xx}+\Theta_{yy}\right)+\Theta_{zz}\right)\right) \tag{B.2}$$
$$\mu_4 = \frac{1}{2}\left(-2\Theta_{xy}^2 - 2\Theta_{yz}^2 - \Theta_{yy}\left(\Theta_{xx}+3\Theta_{yy}+\Theta_{zz}\right)\right) \tag{B.3}$$
$$\mu_5 = \frac{1}{2}\left(-2\Theta_{xy}\Theta_{zx} - \Theta_{yz}\left(\Theta_{xx}+3\left(\Theta_{yy}+\Theta_{zz}\right)\right)\right) \tag{B.4}$$

B.3.2 C_p Matrix

The C_p matrix is presented here with the following substitutions

$C_p =$

$$\begin{pmatrix}
v_z & 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{zx}}{\rho} & v_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\frac{\Theta_{yz}}{\rho} & 0 & v_z & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{\Theta_{zz}}{\rho} & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 2\Theta_{zx} & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \Theta_{yz} & \Theta_{zx} & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2\Theta_{yz} & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Theta_{zz} & \Theta_{yz} & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2\Theta_{zz} & 0 & 0 & 0 & 0 & v_z & 0 & 0 & 0 & 0 \\
0 & \Theta_{zz} & 0 & \Theta_{zx} & 0 & 0 & 0 & 0 & 0 & v_z & 0 & 0 & 0 \\
\mu_3 & q_z & 0 & q_x & 0 & 0 & 0 & -\rho\Theta_{xy} & -\rho\Theta_{zx} & -\frac{1}{2}\rho\left(\theta+2\Theta_{xx}\right) & v_z & 0 & 0 \\
\mu_5 & 0 & q_z & q_y & 0 & 0 & 0 & -\frac{1}{2}\rho\left(\theta+2\Theta_{yy}\right) & -\rho\Theta_{yz} & -\rho\Theta_{xy} & 0 & v_z & 0 \\
\mu_6 & 0 & 0 & 2q_z & 0 & 0 & 0 & -\rho\Theta_{yz} & -\frac{1}{2}\rho\left(\theta+2\Theta_{zz}\right) & -\rho\Theta_{zx} & 0 & 0 & v_z
\end{pmatrix}, \tag{B.5}$$

where,

$$\mu_3 = \frac{1}{2}\left(-2\Theta_{xy}\Theta_{yz} - \Theta_{zx}\left(3\Theta_{xx} + \Theta_{yy} + 3\Theta_{zz}\right)\right) \tag{B.6}$$

$$\mu_5 = \frac{1}{2}\left(-2\Theta_{xy}\Theta_{zx} - \Theta_{yz}\left(\Theta_{xx} + 3\left(\Theta_{yy} + \Theta_{zz}\right)\right)\right) \tag{B.7}$$

$$\mu_6 = \frac{1}{2}\left(-2\Theta_{yz}^2 - 2\Theta_{zx}^2 - \Theta_{zz}\left(\Theta_{xx} + \Theta_{yy} + 3\Theta_{zz}\right)\right) \tag{B.8}$$

B.3.3 B_p Eigenvectors

The right eigenvectors for the B_p matrix are presented here with the following simplifications

$$\begin{aligned} \alpha &= \bar{Q}_y[1,1,1,3], & \beta &= \bar{Q}_y[-1,1,1,3], \\ \chi &= \bar{Q}_y[-1,1,3,1], & \sigma &= \bar{Q}_y[-1,1,1,3]. \end{aligned} \tag{B.9}$$

$$r_{B1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad r_{B2} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad r_{B3} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad r_{B4} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{B5} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{B.10}$$

$$r_{B6} = \begin{pmatrix} -\rho \\ 0 \\ 0 \\ 0 \\ 0 \\ \Theta_{xy} \\ \Theta_{yy} \\ \Theta_{yz} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{B7} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{B8} = \begin{pmatrix} 0 \\ 2\sqrt{\Theta_{yy}}\Theta_{yz} \\ 0 \\ -2\Theta_{xy}\sqrt{\Theta_{yy}} \\ -4\Theta_{xy}\Theta_{yz} \\ -2\Theta_{yy}\Theta_{yz} \\ 0 \\ 2\Theta_{xy}\Theta_{yy} \\ 4\Theta_{xy}\Theta_{yz} \\ 2\left(\Theta_{xy}^2 - \Theta_{yz}^2\right) \\ 2\rho\Theta_{xy}\sqrt{\Theta_{yy}}\Theta_{zx} - \Theta_{yz}\bar{Q}_y[1,3,1,1] \\ 0 \\ -2\rho\sqrt{\Theta_{yy}}\Theta_{yz}\Theta_{zx} + \Theta_{xy}\bar{Q}_y[1,1,1,3] \end{pmatrix}, \tag{B.11}$$

$$r_{B9} = \begin{pmatrix} 0 \\ -\alpha\sqrt{\Theta_{yy}} \\ 0 \\ 2\rho\Theta_{yy}\Theta_{zx} \\ 2\alpha\Theta_{xy} \\ \alpha\Theta_{yy} \\ 0 \\ -2\rho\Theta_{yy}^{3/2}\Theta_{zx} \\ -4\rho\sqrt{\Theta_{yy}}\Theta_{yz}\Theta_{zx} \\ \alpha\Theta_{yz} - 2\rho\Theta_{xy}\sqrt{\Theta_{yy}}\Theta_{zx} \\ \frac{\alpha^2}{2} - 2\rho^2\Theta_{yy}\Theta_{zx}^2 + \alpha\rho\sqrt{\Theta_{yy}}\left(\Theta_{xx} - \Theta_{zz}\right) \\ \alpha\rho\Theta_{xy}\sqrt{\Theta_{yy}} - 2\rho^2\Theta_{yy}\Theta_{yz}\Theta_{zx} \\ 0 \end{pmatrix}, \tag{B.12}$$

$$r_{B10} = \begin{pmatrix} 0 \\ -2\sqrt{\Theta_{yy}}\Theta_{yz} \\ 0 \\ 2\Theta_{xy}\sqrt{\Theta_{yy}} \\ -4\Theta_{xy}\Theta_{yz} \\ -2\Theta_{yy}\Theta_{yz} \\ 0 \\ 2\Theta_{xy}\Theta_{yy} \\ 4\Theta_{xy}\Theta_{yz} \\ 2\left(\Theta_{xy}^2 - \Theta_{yz}^2\right) \\ -2\rho\Theta_{xy}\sqrt{\Theta_{yy}}\Theta_{zx} - \Theta_{yz}\bar{Q}_y[-1,3,1,1] \\ 0 \\ 2\rho\sqrt{\Theta_{yy}}\Theta_{yz}\Theta_{zx} + \Theta_{xy}\bar{Q}_y[-1,1,1,3] \end{pmatrix}, \tag{B.13}$$

$$r_{B11} = \begin{pmatrix} 0 \\ 2\beta\sqrt{\Theta_{yy}} \\ 0 \\ 4\rho\Theta_{yy}\Theta_{zx} \\ 4\beta\Theta_{xy} \\ 2\beta\Theta_{yy} \\ 0 \\ 4\rho\Theta_{yy}^{3/2}\Theta_{zx} \\ 8\rho\sqrt{\Theta_{yy}}\Theta_{yz}\Theta_{zx} \\ 2\beta\Theta_{yz} + 4\rho\Theta_{xy}\sqrt{\Theta_{yy}}\Theta_{zx} \\ \beta^2 - 4\rho^2\Theta_{yy}\Theta_{zx}^2 + 2\beta\rho\sqrt{\Theta_{yy}}\left(-\Theta_{xx} + \Theta_{zz}\right) \\ -2\beta\rho\Theta_{xy}\sqrt{\Theta_{yy}} - 4\rho^2\Theta_{yy}\Theta_{yz}\Theta_{zx} \\ 0 \end{pmatrix}, \tag{B.14}$$

$$r_{B12} = \begin{pmatrix} 2\sqrt{3}\rho\Theta_{yy} \\ -6\Theta_{xy}\sqrt{\Theta_{yy}} \\ -6\Theta_{yy}^{3/2} \\ -6\sqrt{\Theta_{yy}}\Theta_{yz} \\ 4\sqrt{3}\Theta_{xy}^2 \\ 4\sqrt{3}\Theta_{xy}\Theta_{yy} \\ 4\sqrt{3}\Theta_{yy}^2 \\ 4\sqrt{3}\Theta_{yy}\Theta_{yz} \\ 4\sqrt{3}\Theta_{yz}^2 \\ 4\sqrt{3}\Theta_{xy}\Theta_{yz} \\ 2\sqrt{3}q_y\Theta_{xy} + 2\sqrt{3}q_x\Theta_{yy} + 3\rho\sqrt{\Theta_{yy}}\left(2\Theta_{yz}\Theta_{zx} + \Theta_{xy}\left(3\left(\Theta_{xx} + \Theta_{yy}\right) + \Theta_{zz}\right)\right) \\ \sqrt{\Theta_{yy}}\left(4\sqrt{3}q_y\sqrt{\Theta_{yy}} + 3\rho\left(2\Theta_{xy}^2 + 2\Theta_{yz}^2 + \Theta_{yy}\left(\Theta_{xx} + 3\Theta_{yy} + \Theta_{zz}\right)\right)\right) \\ 2\sqrt{3}q_z\Theta_{yy} + 2\sqrt{3}q_y\Theta_{yz} + 3\rho\sqrt{\Theta_{yy}}\left(2\Theta_{xy}\Theta_{zx} + \Theta_{yz}\left(\Theta_{xx} + 3\left(\Theta_{yy} + \Theta_{zz}\right)\right)\right) \end{pmatrix}, \tag{B.15}$$

and

$$r_{B13} = \begin{pmatrix} 2\sqrt{3}\rho\Theta_{yy} \\ 6\Theta_{xy}\sqrt{\Theta_{yy}} \\ 6\Theta_{yy}^{3/2} \\ 6\sqrt{\Theta_{yy}}\Theta_{yz} \\ 4\sqrt{3}\Theta_{xy}^2 \\ 4\sqrt{3}\Theta_{xy}\Theta_{yy} \\ 4\sqrt{3}\Theta_{yy}^2 \\ 4\sqrt{3}\Theta_{yy}\Theta_{yz} \\ 4\sqrt{3}\Theta_{yz}^2 \\ 4\sqrt{3}\Theta_{xy}\Theta_{yz} \\ 2\sqrt{3}q_y\Theta_{xy} + 2\sqrt{3}q_x\Theta_{yy} - 3\rho\sqrt{\Theta_{yy}}\left(2\Theta_{yz}\Theta_{zx} + \Theta_{xy}\left(3\left(\Theta_{xx} + \Theta_{yy}\right) + \Theta_{zz}\right)\right) \\ \sqrt{\Theta_{yy}}\left(4\sqrt{3}q_y\sqrt{\Theta_{yy}} - 3\rho\left(2\Theta_{xy}^2 + 2\Theta_{yz}^2 + \Theta_{yy}\left(\Theta_{xx} + 3\Theta_{yy} + \Theta_{zz}\right)\right)\right) \\ 2\sqrt{3}q_z\Theta_{yy} + 2\sqrt{3}q_y\Theta_{yz} - 3\rho\sqrt{\Theta_{yy}}\left(2\Theta_{xy}\Theta_{zx} + \Theta_{yz}\left(\Theta_{xx} + 3\left(\Theta_{yy} + \Theta_{zz}\right)\right)\right) \end{pmatrix}. \tag{B.16}$$

B.3.4 C_p Eigenvectors

The right eigenvectors for the C_p matrix are presented here with the following simplifications

$$\beta = \bar{Q}_z[1,1,3,1], \qquad \chi = \bar{Q}_z[-1,1,3,1], \tag{B.17}$$

$$r_{C1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad r_{C2} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad r_{C3} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad r_{C4} = \begin{pmatrix} -\rho \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Theta_{yz} \\ \Theta_{zz} \\ \Theta_{zx} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{C5} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{B.18}$$

$$r_{C6} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{C7} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad r_{C8} = \begin{pmatrix} 0 \\ -2\beta\sqrt{\Theta_{zz}} \\ 4\rho\Theta_{xy}\Theta_{zz} \\ 0 \\ 4\beta\Theta_{zx} \\ 2\beta\Theta_{yz} - 4\rho\Theta_{xy}\Theta_{zx}\sqrt{\Theta_{zz}} \\ -8\rho\Theta_{xy}\Theta_{yz}\sqrt{\Theta_{zz}} \\ -4\rho\Theta_{xy}\Theta_{zz}^{3/2} \\ 0 \\ 2\beta\Theta_{zz} \\ \beta^2 + 2\beta\rho\left(\Theta_{xx} - \Theta_{yy}\right)\sqrt{\Theta_{zz}} - 4\rho^2\Theta_{xy}^2\Theta_{zz} \\ 0 \\ 2\beta\rho\Theta_{zx}\sqrt{\Theta_{zz}} - 4\rho^2\Theta_{xy}\Theta_{yz}\Theta_{zz} \end{pmatrix}, \tag{B.19}$$

$$r_{C9} = \begin{pmatrix} 0 \\ 2\Theta_{yz}\sqrt{\Theta_{zz}} \\ -2\Theta_{zx}\sqrt{\Theta_{zz}} \\ 0 \\ -4\Theta_{yz}\Theta_{zx} \\ -2\Theta_{yz}^2 + 2\Theta_{zx}^2 \\ 4\Theta_{yz}\Theta_{zx} \\ 2\Theta_{zx}\Theta_{zz} \\ 0 \\ -2\Theta_{yz}\Theta_{zz} \\ 2\rho\Theta_{xy}\Theta_{zx}\sqrt{\Theta_{zz}} - \Theta_{yz}\bar{Q}_y[1,3,1,1] \\ -2\rho\Theta_{xy}\Theta_{yz}\sqrt{\Theta_{zz}} + \Theta_{zx}\bar{Q}_y[1,1,3,1] \\ 0 \end{pmatrix}, \tag{B.20}$$

$$r_{C10} = \begin{pmatrix} 0 \\ 2\chi\sqrt{\Theta_{zz}} \\ 4\rho\Theta_{xy}\Theta_{zz} \\ 0 \\ 4\chi\Theta_{zx} \\ 2\chi\sqrt{\Theta_{zz}} \\ 8\rho\Theta_{xy}\Theta_{yz}\sqrt{\Theta_{zz}} \\ 4\rho\Theta_{xy}\Theta_{zz}^{3/2} \\ 0 \\ 2\chi\Theta_{zz} \\ \chi^2 + 2\rho\chi\left(-\Theta_{xx} + \Theta_{yy}\right)\sqrt{\Theta_{zz}} - 4\rho^2\Theta_{xy}^2\Theta_{zz} \\ 0 \\ -2\rho\chi\Theta_{zx}\sqrt{\Theta_{zz}} - 4\rho^2\Theta_{xy}\Theta_{yz}\Theta_{zz} \end{pmatrix}, \tag{B.21}$$

$$r_{C11} = \begin{pmatrix} 0 \\ -2\Theta_{yz}\sqrt{\Theta_{zz}} \\ 2\Theta_{zx}\sqrt{\Theta_{zz}} \\ 0 \\ -4\Theta_{yz}\Theta_{zx} \\ -2\Theta_{yz}^2 + 2\Theta_{zx}^2 \\ 4\Theta_{yz}\Theta_{zx} \\ 2\Theta_{zx}\Theta_{zz} \\ 0 \\ -2\Theta_{yz}\Theta_{zz} \\ -2\rho\Theta_{xy}\Theta_{zx}\sqrt{\Theta_{zz}} - \Theta_{yz}\bar{Q}_y[-1,3,1,1] \\ 2\rho\Theta_{xy}\Theta_{yz}\sqrt{\Theta_{zz}} + \Theta_{zx}\bar{Q}_y[-1,1,3,1] \\ 0 \end{pmatrix}, \tag{B.22}$$

$$r_{C12} = \begin{pmatrix} 2\sqrt{3}\rho\Theta_{zz} \\ -6\Theta_{zx}\sqrt{\Theta_{zz}} \\ -6\Theta_{yz}\sqrt{\Theta_{zz}} \\ -6\Theta_{zz}^{3/2} \\ 4\sqrt{3}\Theta_{zx}^2 \\ 4\sqrt{3}\Theta_{yz}\Theta_{zx} \\ 4\sqrt{3}\Theta_{yz}^2 \\ 4\sqrt{3}\Theta_{yz}\Theta_{zz} \\ 4\sqrt{3}\Theta_{zz}^2 \\ 4\sqrt{3}\Theta_{zx}\Theta_{zz} \\ 2\sqrt{3}q_z\Theta_{zx} + 3\rho\left(2\Theta_{xy}\Theta_{yz} + \left(3\Theta_{xx} + \Theta_{yy}\right)\Theta_{zx}\right)\sqrt{\Theta_{zz}} + 2\sqrt{3}q_x\Theta_{zz} + 9\rho\Theta_{zx}\Theta_{zz}^{3/2} \\ 2\sqrt{3}q_z\Theta_{yz} + 3\rho\left(\left(\Theta_{xx} + 3\Theta_{yy}\right)\Theta_{yz} + 2\Theta_{xy}\Theta_{zx}\right)\sqrt{\Theta_{zz}} + 2\sqrt{3}q_y\Theta_{zz} + 9\rho\Theta_{yz}\Theta_{zz}^{3/2} \\ \sqrt{\Theta_{zz}}\left(6\rho\Theta_{yz}^2 + 6\rho\Theta_{zx}^2 + 4\sqrt{3}q_z\sqrt{\Theta_{zz}} + 3\rho\Theta_{zz}\left(\Theta_{xx} + \Theta_{yy} + 3\Theta_{zz}\right)\right) \end{pmatrix}, \tag{B.23}$$

and

$$r_{C13} = \begin{pmatrix} -2\sqrt{3}\rho\Theta_{zz} \\ -6\Theta_{z,x}\sqrt{\Theta_{zz}} \\ -6\Theta_{y,z}\sqrt{\Theta_{zz}} \\ -6\Theta_{zz}^{3/2} \\ -4\sqrt{3}\Theta_{zx}^2 \\ -4\sqrt{3}\Theta_{yz}\Theta_{zx} \\ -4\sqrt{3}\Theta_{yz}^2 \\ -4\sqrt{3}\Theta_{yz}\Theta_{zz} \\ -4\sqrt{3}\Theta_{zz}^2 \\ -4\sqrt{3}\Theta_{zx}\Theta_{zz} \\ -2\sqrt{3}q_z\Theta_{zx} + 3\rho\left(2\Theta_{xy}\Theta_{yz} + \left(3\Theta_{xx} + \Theta_{yy}\right)\Theta_{zx}\right)\sqrt{\Theta_{zz}} - 2\sqrt{3}q_x\Theta_{zz} + 9\rho\Theta_{zx}\Theta_{zz}^{3/2} \\ -2\sqrt{3}q_z\Theta_{yz} + 3\rho\left(\left(\Theta_{xx} + 3\Theta_{yy}\right)\Theta_{yz} + 2\Theta_{xy}\Theta_{zx}\right)\sqrt{\Theta_{zz}} - 2\sqrt{3}q_y\Theta_{zz} + 9\rho\Theta_{yz}\Theta_{zz}^{3/2} \\ \sqrt{\Theta_{zz}}\left(6\rho\Theta_{yz}^2 + 6\rho\Theta_{zx}^2 - 4\sqrt{3}q_z\sqrt{\Theta_{zz}} + 3\rho\Theta_{zz}\left(\Theta_{xx} + \Theta_{yy} + 3\Theta_{zz}\right)\right) \end{pmatrix}. \tag{B.24}$$

Appendix C
FINITE VOLUME CODE IN MATLAB

C.1 13-Moment Electrostatic Simulation

C.1.1 main.m

```
function main
% Name:
%   main
% Description:
%   Simulation to solve Electro-Static two-fluid simulations in 1D. 2nd
%   Order Accurate Finite Volume Method is used. Heun's Method is used to
%   integrate in time and a FORCE flux method is used to calculate
%   intercell fluxes. Source terms are treated as UN-SPLIT and included in
%   the time integration. Finite Differencing is used to solve Poisson'e
%   equation and the Electric Field is the gradient of the potential. Code
%   works for both 13-Moment and standard 5-Moment two-fluid models with
%   small modifications - Code that is used in each model contains a tag at
%   the beginning specifying 13-Moment or 5-Moment. Switching these
%   functions switches models.
% Algorithms :
%   Gudonov Scheme - LeVeque, Toro
%   Huen's Method - Leader
%   Finite Differences - Kutz
% References :
%   [1] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid
%       Dynamics Third Edition, Springer, New York (2009).
%   [2] R. J. LeVeque, Finite Volume Methods for Hyperbolic Problems,
%       Cambridge University Press, New York (2002).
%   [3] J. J. Leader, Numerical Analysis and Scientific Computation,
%       Pearson Assison Wesley, Boston (2004).
%   [4] J. N. Kutz, AMATH 581 Practical Scientific Computing, Dept. of App.
%       Math., Version 1.1, (2005).
%   [5] U. Shumlak, Approximate Riemann solver for the two-fluid plasma
%       model, Journ. of Comp. Phys., 187 (2003), p. 620-638.
%   [6] J. Loverich, A Finite Volume Algorithm for the Two-Fluid Plasma
%       System, 16th AIAA Comp. Fluid Dyn. Conf., June 23-26 (2003).
%   [7] M. Torrilhon, Hyperbolic Moment Equations in Kinetic Gas Theory
%       Based on Multi-Variate Pearson-IV-Distributions, Commun. Comput.
%       Phys., 7 (2010), p. 639-673.
%   [8] B. Srinivasan, Numerical Methods for 3-Dimensional Magnetic
%       Confinement Configurations using Two-Fluid Plasma Equations,
%       Dissertation, University of Washington, (2010).
% Functions :
%   InitialConditions
%   BoundaryConditions
%   AdvanceToPoint - Incrementally advances solution up to a final time
%   Output - Saves variables to data files
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   July 17, 2011 - Document Updated to Electro-Static Two-Fluid Model
%   Plots script graphs outputs.
%   Note:   A special thanks to Dr. Torrilhon (Aachen University) for
%           his help in troubleshooting this code. His original code
%           for the 13-Moment w/ Grad's Closure is the basis for much
%           of what's presented here. My graditude to him cannot be
%           overstated!
%
clc; clear all; close all;  % Housekeeping commands

    % ***************************************************************
    % ***                       SWITCHES                          ***
    % ***************************************************************
    % Switches (Some Declared as Global Variables) :
        % Select Closure Scheme :
        global SetClosure;
            % 'sing' : Singular Closure Scheme (Default)
            % 'real' : Realizable Closure Scheme
            SetClosure = 'sing';

        % Select Simulation :
            % By default : two-fluid simulation
            %SetSimulation = 'TwoFluid';
            %SetSimulation = 'StrongShock';
            %SetSimulation = 'TestCase1';    % C1 = .1
            %SetSimulation = 'TestCase2';    % C1 = 1
            SetSimulation = 'TestCase3';    % C1 = 10

        % Clear Directory?
            % This flag being true will delete all the .dat files in the
            % SaveName directory,
            SetClearDirectory = true;

    % ****************************************************************
    % ***                    CONTROL VARIABLES                     ***
    % ****************************************************************

%     % Domain Variables
        nc = 1000;  % Number of Cells
        xl = -0.5;    % Left Boundary x-coordinate
        xr = 0.5;     % Right Boundary x-coordinate

    % Time Step Control Variables
        dt = 1e-3;    % Initial Step Value (best guess)
        t = 0.0;        % Starting time

        CFLmax = 1.0;   % Maximum Allowable CFL number
        CFLset = .90;   % Target CFL number (~ Average CFL number)

        tfinal = 0.1;   % Simulation ending time

        % Vector of times to be saved to output
        tsaves = linspace(0,tfinal,10+1);
            % e.g. [0, 0.1, ... 0.7]

    % Output Control Variables
        % Usage : directory / fileprefix
        %   (If directory does not exist or some other error occurs,
        %   code keeps running and an error line is generated.)
        SaveName = 'saves/TwoFluid13_tc4_lo_';
        SaveNum = 0;

    % ****************************************************************
    % ***                    INITIAL CONDITIONS                    ***
    % ****************************************************************

    % Get Initial Conditions
        [IONS, ELEC, AppliedEx, xc, CON, outputStr] = ...
            InitialConditions(nc, xl, xr, SetSimulation);
        % Pull Constants
        first = CON(4);
        last = CON(5);
    % Set Boundary Conditions
        IONS = BoundaryConditions(IONS, first, last);
        ELEC = BoundaryConditions(ELEC, first, last);

    % ****************************************************************
    % ***                      SIMULATION LOOP                     ***
    % ****************************************************************

    % Clear Files?
    if SetClearDirectory
        recycle on;
        delStr = strcat(SaveName, '*','.dat');
        fprintf( ' Deleting : %s\n',delStr);
        delete(delStr);
        recycle off;
    end

    % For each time in the list of saved times, advance to that point.
    for i = 1:length(tsaves)
        % Get Target Time
        tend = tsaves(i);

        % Advance Solution
        [t, IONS, ELEC, Ex] = AdvanceToPoint( IONS, ELEC, AppliedEx, ...
            xc, CON, dt, t, tend, CFLmax, CFLset, tfinal );

        % Output Solution
        Output(IONS, ELEC, Ex, xc, CON, t, ...
            CFLmax, CFLset, SaveName, SaveNum);
        SaveNum = SaveNum + 1;
    end
    beep;
end % End main function
% ----------------------------------------------------------------
```

C.1.2 InitialConditions.m

```
% --------         13 - Moment        ------

% ****************************************************************
% ***                  INITIAL CONDITIONS                    ***
% ****************************************************************

function [IONS, ELEC, Ex, xc, CON, outputStr] = ...
    InitialConditions(nc, xl, xr, type)
% Name:
%   InitialConditions
% Description:
%   Sets the initial conditions based on the type passed.
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   July 1st, 2011 - Updated for Two-Fluid Model
%

% Set Number of Ghost Cells (Don't change unless necessary.)
NumGhostCells = 2; % (On Each Side)

nc = nc + NumGhostCells*2;  % Updated Number of Cells
nf = nc+1;                  % Number of faces
xf = linspace(xl,xr,nf);% Vector of face x-positions
dx = mean(diff(xf));        % Spatial resolution

first = NumGhostCells + 1;  % Beginning of Domain
last = nc - NumGhostCells;  % End of Domain

% Vector of center positions
xc = xl+dx/2:dx:xr-dx/2;
outputStr = '';
% For each simulation type,
    switch lower(type)
        case 'strongshock'
            outputStr = 'Strong Shock Simulation';
            for i=1:60  fprintf('*'); end;  fprintf('\n');
            fprintf('%40s\n','Shock Tube Simulation');
            for i=1:60  fprintf('*'); end;  fprintf('\n');
            fprintf('\n\t**** CONSTANTS ****\n');
            % Ion and electron masses and charges
            qI = 1;
            qE = -1;
            mE = 1/1836;
            mI = 1;
            Ex = 0*ones(nc,1);  % Applied E-Field

            %Coefficiencts
            c1 = 1e-16;    % Plasma Size / Ion Larmor Radius
            c2 = 1e-16;   % Ion Larmor Radius / Debye Length

            fprintf('%20s:\t%g\n','Electron Mass',mE);
            fprintf('%20s:\t%g\n','Ion Mass',mI);
            fprintf('%20s:\t%g\n','Electron Charge',qE);
            fprintf('%20s:\t%g\n','Ion Charge',qI);
            fprintf('%40s:\t%g\n','Plasma Size / Ion Larmor Radius',c1);
            fprintf('%40s:\t%g\n','Ion Larmor Radius / Debye Length',c2);

            % **** IONS ****
            fprintf('\n\t**** IONS ****\n');
```

```
% Left State
rhoL = 50;
pL = 50;
pxxL = 50;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Scalar Pressure',pL);
fprintf('%20s:\t%g\n','x-Pressure',pxxL);
fprintf('%20s:\t%g\n','x-Heat',qL);

% Right State
rhoR = 1;
pR = 1;
pxxR = 1;
qR = 0;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Scalar Pressure',pR);
fprintf('%20s:\t%g\n','x-Pressure',pxxR);
fprintf('%20s:\t%g\n','x-Heat',qR);

% Set values based on x-position
for i=1:nc
    q = qL;    % Heat Vector
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
        pxx = pxxL; % X-Direction Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
        pxx = pxxR;  % X-Direction Pressure
    end
    % Convert to CONSERVATIVE variables and store to U
    U(i,1) = rho;
    U(i,2) = rho*v;
    U(i,3) = 0.5*rho*v*v+1.5*p;
    U(i,4) = rho*v*v + pxx;
    U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
end
IONS = U;

% **** ELECTRONS ****
fprintf('\n\t**** ELECTRONS ****\n');
% Left State
rhoL = 50;
pL = 50;
pxxL = 50;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
```

```
        fprintf('%20s:\t%g\n','Scalar Pressure',pL);
        fprintf('%20s:\t%g\n','x-Pressure',pxxL);
        fprintf('%20s:\t%g\n','x-Heat',qL);

        % Right State
        rhoR = 1;
        pR = 1;
        pxxR = 1;
        qR = 0;
        vR = 0;
        fprintf('\tRight State:\n');
        fprintf('%20s:\t%g\n','Density',rhoR);
        fprintf('%20s:\t%g\n','Velocity',vR);
        fprintf('%20s:\t%g\n','Scalar Pressure',pR);
        fprintf('%20s:\t%g\n','x-Pressure',pxxR);
        fprintf('%20s:\t%g\n','x-Heat',qR);

        % Set values based on x-position
        for i=1:nc
            q = qL;    % Heat Vector
            v = vL;    % Velocity

            if xc(i) < 0.0  % State 1: Upstream of Shock
                rho = rhoL; % Density
                p = pL;   % Scalar Pressure
                pxx = pxxL; % X-Direction Pressure
            else            % State 2: Downstream of shock
                rho = rhoR; % Density
                p = pR;    % Scalar Pressure
                pxx = pxxR;  % X-Direction Pressure
            end
            % Convert to CONSERVATIVE variables and store to U
            U(i,1) = rho;
            U(i,2) = rho*v;
            U(i,3) = 0.5*rho*v*v+1.5*p;
            U(i,4) = rho*v*v + pxx;
            U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
        end
        ELEC = U;
    case 'testcase1'
        % Electrostatic Test Case 1
        outputStr = 'Two Fluid Simulation';
        for i=1:60  fprintf('*'); end;  fprintf('\n');
        fprintf('%40s\n','Shock Tube Simulation');
        for i=1:60  fprintf('*'); end;  fprintf('\n');
        fprintf('\n\t**** CONSTANTS ****\n');
        % Ion and electron masses and charges
        qI = 1;
        qE = -1;
        mE = 1/1836;
        mI = 1;
        Ex = 0*ones(nc,1);  % Applied E-Field

        %Coefficiencts
        c1 = 10^-1;    % Plasma Size / Ion Larmor Radius
        c2 = 100;   % Ion Larmor Radius / Debye Length

        fprintf('%20s:\t%g\n','Electron Mass',mE);
        fprintf('%20s:\t%g\n','Ion Mass',mI);
        fprintf('%20s:\t%g\n','Electron Charge',qE);
```

```
fprintf('%20s:\t%g\n','Ion Charge',qI);
fprintf('%40s:\t%g\n','Plasma Size / Ion Larmor Radius',c1);
fprintf('%40s:\t%g\n','Ion Larmor Radius / Debye Length',c2);

% **** IONS ****
fprintf('\n\t**** IONS ****\n');
% Left State
rhoL = 1;
pL = 0.5;
pxxL = 0.5;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Scalar Pressure',pL);
fprintf('%20s:\t%g\n','x-Pressure',pxxL);
fprintf('%20s:\t%g\n','x-Heat',qL);

% Right State
rhoR = .125;
pR = 0.05;
pxxR = 0.05;
qR = 0;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Scalar Pressure',pR);
fprintf('%20s:\t%g\n','x-Pressure',pxxR);
fprintf('%20s:\t%g\n','x-Heat',qR);

% Set values based on x-position
for i=1:nc
    q = qL;    % Heat Vector
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
        pxx = pxxL; % X-Direction Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
        pxx = pxxR;  % X-Direction Pressure
    end
    % Convert to CONSERVATIVE variables and store to U
    U(i,1) = rho;
    U(i,2) = rho*v;
    U(i,3) = 0.5*rho*v*v+1.5*p;
    U(i,4) = rho*v*v + pxx;
    U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
end
IONS = U;

% **** ELECTRONS ****
fprintf('\n\t**** ELECTRONS ****\n');
% Left State
rhoL = 1*mE/mI;
pL = pL;
```

```
        pxxL = pxxL;
        qL = 0;
        vL = 0;
        fprintf('\tLeft State:\n');
        fprintf('%20s:\t%g\n','Density',rhoL);
        fprintf('%20s:\t%g\n','Velocity',vL);
        fprintf('%20s:\t%g\n','Scalar Pressure',pL);
        fprintf('%20s:\t%g\n','x-Pressure',pxxL);
        fprintf('%20s:\t%g\n','x-Heat',qL);

        % Right State
        rhoR = .125*mE/mI;
        pR = pR;
        pxxR = pxxR;
        qR = 0;
        vR = 0;
        fprintf('\tRight State:\n');
        fprintf('%20s:\t%g\n','Density',rhoR);
        fprintf('%20s:\t%g\n','Velocity',vR);
        fprintf('%20s:\t%g\n','Scalar Pressure',pR);
        fprintf('%20s:\t%g\n','x-Pressure',pxxR);
        fprintf('%20s:\t%g\n','x-Heat',qR);

        % Set values based on x-position
        for i=1:nc
            q = qL;    % Heat Vector
            v = vL;    % Velocity

            if xc(i) < 0.0  % State 1: Upstream of Shock
                rho = rhoL; % Density
                p = pL;   % Scalar Pressure
                pxx = pxxL; % X-Direction Pressure
            else            % State 2: Downstream of shock
                rho = rhoR; % Density
                p = pR;    % Scalar Pressure
                pxx = pxxR;  % X-Direction Pressure
            end
            % Convert to CONSERVATIVE variables and store to U
            U(i,1) = rho;
            U(i,2) = rho*v;
            U(i,3) = 0.5*rho*v*v+1.5*p;
            U(i,4) = rho*v*v + pxx;
            U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
        end
        ELEC = U;

    case 'testcase2'
        % Electrostatic Test Case 1
        outputStr = 'Two Fluid Simulation';
        for i=1:60  fprintf('*'); end;  fprintf('\n');
        fprintf('%40s\n','Shock Tube Simulation');
        for i=1:60  fprintf('*'); end;  fprintf('\n');
        fprintf('\n\t**** CONSTANTS ****\n');
        % Ion and electron masses and charges
        qI = 1;
        qE = -1;
        mE = 1/1836;
        mI = 1;
        Ex = 0*ones(nc,1);  % Applied E-Field
```

```
%Coefficiencts
c1 = 1^-1;    % Plasma Size / Ion Larmor Radius
c2 = 100;   % Ion Larmor Radius / Debye Length

fprintf('%20s:\t%g\n','Electron Mass',mE);
fprintf('%20s:\t%g\n','Ion Mass',mI);
fprintf('%20s:\t%g\n','Electron Charge',qE);
fprintf('%20s:\t%g\n','Ion Charge',qI);
fprintf('%40s:\t%g\n','Plasma Size / Ion Larmor Radius',c1);
fprintf('%40s:\t%g\n','Ion Larmor Radius / Debye Length',c2);

% **** IONS ****
fprintf('\n\t**** IONS ****\n');
% Left State
rhoL = 1;
pL = 0.5;
pxxL = 0.5;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Scalar Pressure',pL);
fprintf('%20s:\t%g\n','x-Pressure',pxxL);
fprintf('%20s:\t%g\n','x-Heat',qL);

% Right State
rhoR = .125;
pR = 0.05;
pxxR = 0.05;
qR = 0;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Scalar Pressure',pR);
fprintf('%20s:\t%g\n','x-Pressure',pxxR);
fprintf('%20s:\t%g\n','x-Heat',qR);

% Set values based on x-position
for i=1:nc
    q = qL;    % Heat Vector
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
        pxx = pxxL; % X-Direction Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
        pxx = pxxR;  % X-Direction Pressure
    end
    % Convert to CONSERVATIVE variables and store to U
    U(i,1) = rho;
    U(i,2) = rho*v;
    U(i,3) = 0.5*rho*v*v+1.5*p;
    U(i,4) = rho*v*v + pxx;
    U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
end
```

```
        IONS = U;

        % **** ELECTRONS ****
        fprintf('\n\t**** ELECTRONS ****\n');
        % Left State
        rhoL = 1*mE/mI;
        pL = pL;
        pxxL = pxxL;
        qL = 0;
        vL = 0;
        fprintf('\tLeft State:\n');
        fprintf('%20s:\t%g\n','Density',rhoL);
        fprintf('%20s:\t%g\n','Velocity',vL);
        fprintf('%20s:\t%g\n','Scalar Pressure',pL);
        fprintf('%20s:\t%g\n','x-Pressure',pxxL);
        fprintf('%20s:\t%g\n','x-Heat',qL);

        % Right State
        rhoR = .125*mE/mI;
        pR = pR;
        pxxR = pxxR;
        qR = 0;
        vR = 0;
        fprintf('\tRight State:\n');
        fprintf('%20s:\t%g\n','Density',rhoR);
        fprintf('%20s:\t%g\n','Velocity',vR);
        fprintf('%20s:\t%g\n','Scalar Pressure',pR);
        fprintf('%20s:\t%g\n','x-Pressure',pxxR);
        fprintf('%20s:\t%g\n','x-Heat',qR);

        % Set values based on x-position
        for i=1:nc
            q = qL;    % Heat Vector
            v = vL;    % Velocity

            if xc(i) < 0.0  % State 1: Upstream of Shock
                rho = rhoL; % Density
                p = pL;   % Scalar Pressure
                pxx = pxxL; % X-Direction Pressure
            else            % State 2: Downstream of shock
                rho = rhoR; % Density
                p = pR;    % Scalar Pressure
                pxx = pxxR;  % X-Direction Pressure
            end
            % Convert to CONSERVATIVE variables and store to U
            U(i,1) = rho;
            U(i,2) = rho*v;
            U(i,3) = 0.5*rho*v*v+1.5*p;
            U(i,4) = rho*v*v + pxx;
            U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
        end
        ELEC = U;

    case 'testcase2'
        % Electrostatic Test Case 1
        outputStr = 'Two Fluid Simulation';
        for i=1:60  fprintf('*'); end;  fprintf('\n');
        fprintf('%40s\n','Shock Tube Simulation');
        for i=1:60  fprintf('*'); end;  fprintf('\n');
        fprintf('\n\t**** CONSTANTS ****\n');
```

```
% Ion and electron masses and charges
qI = 1;
qE = -1;
mE = 1/1836;
mI = 1;
Ex = 0*ones(nc,1);  % Applied E-Field

%Coefficienccts
c1 = 1^-1;    % Plasma Size / Ion Larmor Radius
c2 = 100;   % Ion Larmor Radius / Debye Length

fprintf('%20s:\t%g\n','Electron Mass',mE);
fprintf('%20s:\t%g\n','Ion Mass',mI);
fprintf('%20s:\t%g\n','Electron Charge',qE);
fprintf('%20s:\t%g\n','Ion Charge',qI);
fprintf('%40s:\t%g\n','Plasma Size / Ion Larmor Radius',c1);
fprintf('%40s:\t%g\n','Ion Larmor Radius / Debye Length',c2);

% **** IONS ****
fprintf('\n\t**** IONS ****\n');
% Left State
rhoL = 1;
pL = 0.5;
pxxL = 0.5;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Scalar Pressure',pL);
fprintf('%20s:\t%g\n','x-Pressure',pxxL);
fprintf('%20s:\t%g\n','x-Heat',qL);

% Right State
rhoR = .125;
pR = 0.05;
pxxR = 0.05;
qR = 0;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Scalar Pressure',pR);
fprintf('%20s:\t%g\n','x-Pressure',pxxR);
fprintf('%20s:\t%g\n','x-Heat',qR);

% Set values based on x-position
for i=1:nc
    q = qL;    % Heat Vector
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
        pxx = pxxL; % X-Direction Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
        pxx = pxxR;  % X-Direction Pressure
    end
```

```
        % Convert to CONSERVATIVE variables and store to U
        U(i,1) = rho;
        U(i,2) = rho*v;
        U(i,3) = 0.5*rho*v*v+1.5*p;
        U(i,4) = rho*v*v + pxx;
        U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
    end
    IONS = U;

    % **** ELECTRONS ****
    fprintf('\n\t**** ELECTRONS ****\n');
    % Left State
    rhoL = 1*mE/mI;
    pL = pL;
    pxxL = pxxL;
    qL = 0;
    vL = 0;
    fprintf('\tLeft State:\n');
    fprintf('%20s:\t%g\n','Density',rhoL);
    fprintf('%20s:\t%g\n','Velocity',vL);
    fprintf('%20s:\t%g\n','Scalar Pressure',pL);
    fprintf('%20s:\t%g\n','x-Pressure',pxxL);
    fprintf('%20s:\t%g\n','x-Heat',qL);

    % Right State
    rhoR = .125*mE/mI;
    pR = pR;
    pxxR = pxxR;
    qR = 0;
    vR = 0;
    fprintf('\tRight State:\n');
    fprintf('%20s:\t%g\n','Density',rhoR);
    fprintf('%20s:\t%g\n','Velocity',vR);
    fprintf('%20s:\t%g\n','Scalar Pressure',pR);
    fprintf('%20s:\t%g\n','x-Pressure',pxxR);
    fprintf('%20s:\t%g\n','x-Heat',qR);

    % Set values based on x-position
    for i=1:nc
        q = qL;    % Heat Vector
        v = vL;    % Velocity

        if xc(i) < 0.0  % State 1: Upstream of Shock
            rho = rhoL; % Density
            p = pL;   % Scalar Pressure
            pxx = pxxL; % X-Direction Pressure
        else            % State 2: Downstream of shock
            rho = rhoR; % Density
            p = pR;    % Scalar Pressure
            pxx = pxxR;  % X-Direction Pressure
        end
        % Convert to CONSERVATIVE variables and store to U
        U(i,1) = rho;
        U(i,2) = rho*v;
        U(i,3) = 0.5*rho*v*v+1.5*p;
        U(i,4) = rho*v*v + pxx;
        U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
    end
    ELEC = U;
case 'testcase3'
```

```
            outputStr = 'Two Fluid Simulation';
for i=1:60  fprintf('*'); end;  fprintf('\n');
fprintf('%40s\n','Shock Tube Simulation');
for i=1:60  fprintf('*'); end;  fprintf('\n');
fprintf('\n\t**** CONSTANTS ****\n');
% Ion and electron masses and charges
qI = 1;
qE = -1;
mE = 1/1836;
mI = 1;
Ex = 0*ones(nc,1);  % Applied E-Field

%Coefficiencts
c1 = .1^-1;    % Plasma Size / Ion Larmor Radius
c2 = 100;   % Ion Larmor Radius / Debye Length

fprintf('%20s:\t%g\n','Electron Mass',mE);
fprintf('%20s:\t%g\n','Ion Mass',mI);
fprintf('%20s:\t%g\n','Electron Charge',qE);
fprintf('%20s:\t%g\n','Ion Charge',qI);
fprintf('%40s:\t%g\n','Plasma Size / Ion Larmor Radius',c1);
fprintf('%40s:\t%g\n','Ion Larmor Radius / Debye Length',c2);

% **** IONS ****
fprintf('\n\t**** IONS ****\n');
% Left State
rhoL = 1;
pL = 0.5;
pxxL = 0.5;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Scalar Pressure',pL);
fprintf('%20s:\t%g\n','x-Pressure',pxxL);
fprintf('%20s:\t%g\n','x-Heat',qL);

% Right State
rhoR = .125;
pR = 0.05;
pxxR = 0.05;
qR = 0;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Scalar Pressure',pR);
fprintf('%20s:\t%g\n','x-Pressure',pxxR);
fprintf('%20s:\t%g\n','x-Heat',qR);

% Set values based on x-position
for i=1:nc
    q = qL;    % Heat Vector
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
        pxx = pxxL; % X-Direction Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
        pxx = pxxR;  % X-Direction Pressure
    end
    % Convert to CONSERVATIVE variables and store to U
    U(i,1) = rho;
    U(i,2) = rho*v;
    U(i,3) = 0.5*rho*v*v+1.5*p;
    U(i,4) = rho*v*v + pxx;
    U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
end
IONS = U;

% **** ELECTRONS ****
fprintf('\n\t**** ELECTRONS ****\n');
% Left State
rhoL = 1*mE/mI;
pL = pL;
pxxL = pxxL;
qL = 0;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Scalar Pressure',pL);
fprintf('%20s:\t%g\n','x-Pressure',pxxL);
fprintf('%20s:\t%g\n','x-Heat',qL);

% Right State
rhoR = .125*mE/mI;
pR = pR;
pxxR = pxxR;
qR = 0;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Scalar Pressure',pR);
fprintf('%20s:\t%g\n','x-Pressure',pxxR);
fprintf('%20s:\t%g\n','x-Heat',qR);

% Set values based on x-position
for i=1:nc
    q = qL;    % Heat Vector
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
        pxx = pxxL; % X-Direction Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
        pxx = pxxR;  % X-Direction Pressure
    end
    % Convert to CONSERVATIVE variables and store to U
    U(i,1) = rho;
    U(i,2) = rho*v;
    U(i,3) = 0.5*rho*v*v+1.5*p;
```

```
                U(i,4) = rho*v*v + pxx;
                U(i,5) = (0.5*rho*v*v + 1.5*p)*v + pxx*v + q;
            end
            ELEC = U;

    end % Ends Switch

    % Package all scalars into a single CONSTANT vector
    CON(1) = nc;            % Number of Cells
    CON(2) = NumGhostCells; % Number of Ghost Cells
    CON(3) = dx;            % Spatial derivative step size
    CON(4) = first;         % Beginning of Domain
    CON(5) = last;          % End of Domain
    CON(6) = qI;            % Ion Charge
    CON(7) = qE;            % Electron Charge
    CON(8) = mI;            % Ion Mass
    CON(9) = mE;            % Electron Mass
    CON(10) = c1;           % Plasma Size / Ion Larmor Radius
    CON(11) = c2;           % Ion Larmor Radius / Debye Length
end % Ends Initial Conditions
```

C.1.3 BoundaryConditions.m

```
function [OUT] =BoundaryConditions(Uc, first, last)
% Name:
%   BoundaryConditions
% Description:
%   Sets Boundary Conditions by manipulating ghost cells.
% Algorithms :
%
% Variables :
%   Uc      [in]    : Variables for Current Time (Matrix)
%   first   [in]    : First real domain cell (scalar)
%   last    [in]    : Last real domain cell (Scalar)
%   OUT     [out]   : Variables with updated BC's (Matrix)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%

% Manipulate Ghost Cells (2 on each side)

    % Left Side - Van Neummann
    for i = 1:first-1
        Uc(i,:) = Uc(first,:);
    end

    % Right Side - Van Neummann
    for i = last+1:last+(first-1)
        Uc(i,:) = Uc(last,:);
    end

    % Return Uc
    OUT = Uc;
```

C.1.4 AdvanceToPoint.m

```
function [t, IONS, ELEC, Ex] = AdvanceToPoint( IONS, ELEC, AppliedEx, ...
    xc, CON, dt, t, tend, CFLmax, CFLset, tfinal )
% Name:
%   AdvanceToPoint
% Description:
%   Advances the solution to the 5/13 moment 1-D equations to a certain
%   point in time. This method uses an UNSPLIT method where the solution
%   is integrated in time using Heun's Method. Solutions are computed at
%   a fixed time-step, and then the CFL number is compared to a set value
%   and adjusted until it is within a certain tolerance.
% Algorithms :
%   Unsplit Methods
%   Heun's Method
% Variables :
%   IONS    [in/out]: Conservative Variables for Ions (Matrix)
%   ELEC    [in/out]: Conservative Variables for Electrons (Matrix)
%   xc      [in]    : X-positions for the center of each cell (Vector)
%   dt      [in]    : Time derivative step size (Scalar)
%   t       [in/out]: Current Time (Scalar)
%   tend    [in]    : Stopping Point for "AdvanceToPoint" (Scalar)
%   CFLmax  [in]    : Maximum CFL value allowed (usually 1) (Scalar)
%   CFLset  [in]    : Desired CFL value (~average CFL) (Scalar)
%   tfinal  [in]    : Simulation final time(used for waitbar) (Scalar)
%   Ex      [out]   : Electric Field Strength (Matrix)
%   CON     [in]    : Constants (Vector)
%   AppliedEx - Deprecated.
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   July 1st, 2011 - Modified to use an unsplit method for source terms.
%

% Pull Constants
nc = CON(1);
dx = CON(3);
first = CON(4);
last = CON(5);
qI = CON(6);            % Ion Charge
qE = CON(7);            % Electron Charge
mI = CON(8);            % Ion Mass
mE = CON(9);            % Electron Mass
c1 = CON(10);           % Plasma Size / Ion Larmor Radius
c2 = CON(11);           % Ion Larmor Radius / Debye Length

% Temp ****
Ex = zeros(nc,1);

% Create Waitbar Object
persistent h;   % Declare Persistent Variable
str = sprintf('Running : Currently Advancing to = %f...',tend);
    if tend == 0
        h = waitbar(t/tfinal,str);
    else
        % Updated message
        waitbar(t/tfinal,h,str);
    end

% Run Simulation until time reaches desired point
small = 1e-9;
Cmax = 0;
```

```
while abs(t-tend) > small
    CFLnow = 10*CFLmax;    % Forces a do-while loop
    % Advances Solution a single time step
    while(CFLnow > CFLmax) % or
        CFLnow = 0.0;  % Clear Previous

        % Call Time Integrator
        [IONStemp, ELECtemp, Ex, CFLnow] = HuenStep(IONS, ELEC, CON, dt);

        if( CFLnow > CFLmax)
            % If the current CFL number exceeds max, then decrease step
            % size and reject current time step.
            dt = dt*CFLset/CFLnow;
        else
            t = t + dt;
            dt = dt*CFLset/CFLnow;

            if( t+dt > tend )
                % If current time step would go beyond desired ending
                % point, decrease step size to match.
                dt = tend-(t);
            end
        end

    end % Ends : Advances Solution a single time step

    % Update Progress Bar
    waitbar(t/tfinal,h);
    drawnow;

    % Save Solution and continue
    IONS = IONStemp;
    ELEC = ELECtemp;
    IONS = BoundaryConditions(IONS, first, last);
    ELEC = BoundaryConditions(ELEC, first, last);

% When end time is reached, exit condition is met.
end % Ends : Advances Solution in Small Increments

if abs(t-tfinal) < small % Close Waitbar on final
    close(h);   % Waitbar
end

end     % AdvanceToPoint

function [oIONS, oELEC, oEx, CFL] = HuenStep(IONS, ELEC, CON, dt)
% Name:
%   HeunStep
% Description:
%   2nd Order time integration of flux and source terms
% Algorithms :
%   Huen's Method (aka Modified/Corrected Euler Method)
% Variables :
%   IONS    [in]    : Ion Conservative Variables for Current Time (Matrix)
%   ELEC    [in]    : Electron Conservative Variables for Current Time (Matrix)
%   CON     [in]    : Constants (Vector)
%   dt      [in]    : Time derivative step size (Scalar)
%   oIONS   [out]   : Conservative Variables for Updated Time (Matrix)
%   oELEC   [out]   : Conservative Variables for Updated Time (Matrix)
%   oEx     [out]   : Electric Field Strength for Updated Time (Matrix)
%   CFL      [out]    : Calculated CFL number (scalar)
% Functions :
%   Gudunov
%   BoundaryConditions
%   Esolve
%   Sfunc
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   July 1st, 2011 - Document Creation
%   Vectorized code should allow this function to work for both 5 and 13
%   Moment cases.
%

    % Pull Constants
    nc = CON(1);
    dx = CON(3);
    first = CON(4);
    last = CON(5);
    qI = CON(6);            % Ion Charge
    qE = CON(7);            % Electron Charge
    mI = CON(8);            % Ion Mass
    mE = CON(9);            % Electron Mass
    c1 = CON(10);           % Plasma Size / Ion Larmor Radius
    c2 = CON(11);           % Ion Larmor Radius / Debye Length

    % Update Fluids and Sources Together using
    % Heun's Method (2nd Order Runge-Kutta Method)

    % Predictor Step (Euler Method)
        % Update Fluid Variables
        [IONStemp, CFLion] = Gudunov(IONS, dx, dt, 0, 'vanleer');
        [ELECtemp, CFLele] = Gudunov(ELEC, dx, dt, 0, 'vanleer');

        % Update Output Variables from Fluid Sources
        oIONS = IONS + 0.5*IONStemp;
        oELEC = ELEC + 0.5*ELECtemp;

        % Generate Predicted values for next step
        IONS_tilde = IONStemp + IONS;
        ELEC_tilde = ELECtemp + ELEC;

            % Calculate Electric Field (Used to update source terms)
            Ex = Esolve(ELEC_tilde, IONS_tilde, CON);

        % Add Source Term Contributions
        IONS_tilde = IONS_tilde + dt*Sfunc(IONS, Ex, CON, 'ions');
        ELEC_tilde = ELEC_tilde + dt*Sfunc(ELEC, Ex, CON, 'electrons');

        % Apply Boundary Conditions
        [ELEC_tilde] =BoundaryConditions(ELEC_tilde, first, last);
        [IONS_tilde] =BoundaryConditions(IONS_tilde, first, last);

    % Corrector Step (Trapezoidal Method)
        % Correct Fluid Variables
        [IONStemp, CFLion] = Gudunov(IONS_tilde, dx, dt, 0, 'vanleer');
        [ELECtemp, CFLele] = Gudunov(ELEC_tilde, dx, dt, 0, 'vanleer');

        % Correct Output Variables
        oIONS = oIONS + 0.5*IONStemp;
        oELEC = oELEC + 0.5*ELECtemp;

            % Calculate Electric Field
            Ex_tilde = Esolve(oELEC, oIONS, CON);

        % Add Source Term Contributions
        oIONS = oIONS + dt*Sfunc(IONS_tilde, Ex_tilde, CON, 'ions');
        oELEC = oELEC + dt*Sfunc(ELEC_tilde, Ex_tilde, CON, 'electrons');

    % Calculate Plasma/Upper Hybrid Frequencies
        % Calculate Number Density
        nE = ELEC(:,1)/mE;      % Electron Number Density
        nI = IONS(:,1)/mI;      % Ion Number Density



    % Calculate Maximum Wavespeeds for stability

        eps0 = (c1*c2^2)^(-1);  % Normalized eps0

        % Plasma Frequencies
        Wpe = sqrt( max(abs(nE*qE^2/(eps0*mE))) );
        Wpi = sqrt( max(abs(nI*qI^2/(eps0*mI))) );

        CFLwpe = 10*dt*Wpe; % Multiplier is used to ensure nyquist sampling
        CFLwpi = 10*dt*Wpi; % rate is sufficient to capture plasma waves.
                            % 10x multiplier suggested by (B. Srinivasan).

    % Outputs
        % MAX CFL
        CFL = max([abs(CFLion), abs(CFLele), abs(CFLwpe), abs(CFLwpi)]);

        % Ex
        oEx = Ex_tilde;

end % Ends HeunStep

% Electric Field Solver
function [Ex] = Esolve(ELEC, IONS, CON)
% Name:
%   Esolve
% Description:
%   Uses finite difference method to solve poisson's equation for the
%   voltage (phi). Then takes the gradient to get the E-field
% Algorithms :
%   Finite Difference Method - Kutz
%   LU-Decomposition (Matlab)
%   Gradient (Matlab)
% Variables :
%   IONS    [in/out]: Conservative Variables for Ions (Matrix)
%   ELEC    [in/out]: Conservative Variables for Electrons (Matrix)
%   Ex      [out]   : Electric Field Strength (Matrix)
%   CON     [in]    : Constants (Vector)
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   July 1st, 2011 - Modified to use an unsplit method for source terms.
%   Boundary Conditions are set by this specific problem.
%
```

```
% Pull Coefficients
dx = CON(3);
first = CON(4);
last = CON(5);
qI = CON(6);            % Ion Charge
qE = CON(7);            % Electron Charge
mI = CON(8);            % Ion Mass
mE = CON(9);            % Electron Mass
c1 = CON(10);           % Plasma Size / Ion Larmor Radius
c2 = CON(11);           % Ion Larmor Radius / Debye Length

% Calculate Number Density
nE = ELEC(:,1)/mE;     % Electron Number Density
nI = IONS(:,1)/mI;     % Ion Number Density

% Calculate Charge Density
rho_charge = (qI*nI + qE*nE);

    persistent num;
    persistent matA;
    persistent setFlag;
    if isempty(setFlag)
        num = length(rho_charge);
        e = ones(num*num,1);
        n = num + 2;
        % 2nd Order Centered Differencing
            matA = spdiags([e -2*e e], -1:1, n, n);

            % Clear Top & Bottom Rows
            matA(1,:) = 0;
            matA(n,:) = 0;

            % Apply Neumann BC
            matA(1,1) = -2;
            matA(1,3) = 2;

            % Dirchelet BC
            matA(n,n) = 1;

        setFlag = 1;    % if = 1, Initialized
    end

    % RHS including source term information
    f = [0; dx^2*c1*c2^2*rho_charge; 0];

    tmp = matA\f;
    phi = tmp(2:end-1,1);

    % Calculate Electric Field
        Ex = zeros(size(phi));
        for i=2:num-1
            Ex(i,1) = ( - phi(i-1,1) + phi(i+1,1) )/(2*dx);
        end
        Ex(1,1) = Ex(2,1);
        Ex(num,1) = Ex(num-1,1);

end % Ends Esolve

% --------         13 - Moment        ------ (Sfunc)
```

```
function [OUT] = Sfunc(IN, Ex, CON, species)
% Name:
%   Sfunc
% Description:
%   Calculates source terms to be included in time-integration.
% Algorithms :
%
% Variables :
%   IN      [in]    : Conservative Variables (Matrix)
%   Ex      [in]    : Electric Field Strength (Matrix)
%   CON     [in]    : Constants (Vector)
%   species [in]    : Switch : 'ions' or otherwise
%   OUT     [out]   : Source Contributions (Matrix)
% Comments :
%   Shaun Gilliam
%   July 1st, 2011 - Document Creation
%

    % Update Variables based on source terms
    % **** using vectorized format
    nc = CON(1);

    switch lower(species)
        case 'ions'
            q = CON(6);            % Ion Charge
            m = CON(8);            % Ion Mass
        otherwise
            q = CON(7);            % Electron Charge
            m = CON(9);            % Electron Mass
    end % Ends Switch

    c1 = CON(10);           % Plasma Size / Ion Larmor Radius
    c2 = CON(11);           % Ion Larmor Radius / Debye Length

    % Convert to Primitive Variables
    V = zeros(size(IN));
    V = ConsToPrim(IN);

    % Grab Vectors
    n = IN(:,1)/m;     % Number Density
    vel = V(:,2);
    press = V(:,3);
    pxx = V(:,4);

    % Source Terms
    S1 = zeros(size(n));
    S2 = c1*q*n.*Ex;
    S3 = c1*q*n.*Ex.*vel;
    S4 = 2*c1*q*n.*Ex.*vel;
    S5 = 0.5*c1*(3*q/m*Ex.*press + 2*q/m*Ex.*pxx + 3*q*Ex.*n.*vel.*vel);

    OUT = [S1,S2,S3,S4,S5];   % Output Source Terms

end % Ends Sfunc
```

C.1.5 Output.m

```
function Output(IONS, ELEC, Ex, xc, CON, t, ...
            CFLmax, CFLset, filename, tag)
% This function outputs a Matlab Binary File with the saved simulation
% steps.

% Create File Name
savefile = strcat(filename, num2str(tag),'.dat');

try
    save(savefile,'-mat','IONS', 'ELEC', 'Ex', 'xc', ...
        'CON', 't', 'CFLmax', 'CFLset');
    fprintf('Save succeeded at t = %f...\n',t);
catch
    % Do nothing
    fprintf('Save failed at t = %f...\n',t);
end
```

C.1.6 Gudunov.m

```
function [Unew, CFL] = Gudunov(Uold, dx, dt, c_, setlim)
% Name:
%   Gudunov
% Description:
%   Updates solution using an extension of the Gudunov scheme to second
%   order accuracy. Function can also handle Heun's Method updates. Flux
%   Limiter is VanLeer.
% Algorithms :
%   (Toro)
%   (LeVeque)
% Variables :
%   Uold    [in]    : Conserved Variables for current time (Matrix)
%   dx      [in]    : Spatial derivative step size (Scalar)
%   dt      [in]    : Time derivative step size (Scalar)
%   setlim  [in]    : Determines which Limiter is activated.
%                       Current Options :
%                           'VanLeer' : Van Leer Slope Limiter
%                           'Const'   : No reconstruction (default)
%   Unew    [out]   : Conserved Variables for updated time (Matrix)
%   CFL     [out]   : CFL number based on current timestep (Scalar)
% Functions :
%   ConsToPrim
%   VanLeer
%   FORCE
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   Temporatily calculates first and last, but that will need to eventually
%   be pulled from initial conditions.
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format. Original Code Can bee seen below.
%

% *****
% TEMPORARY ONLY !!!!
last = length(Uold) - 2;
first = 3;
% *****

% Set Max Speed
maxCx = 0;
% Preallocate for speed
V = zeros(size(Uold));

% **********************************************************
% Step 1: Convert from Conservative to Primitive variables
    % from: Cell Before 1st domain cell
    % to: Cell After last domain cell

    V = ConsToPrim(Uold);

% **********************************************************
% Step 2: Boundary Extrapolated Values step

% Preallocate for speed
delta = zeros(size(Uold));  % Difference between values

%c_ = 1; % Euler Methods
```

```
%c_ = 0; % Heun Method
Unew = c_*Uold + zeros(size(Uold));    % Set next step = current step

    % Calculate Differences between values.
    % from: Cell Before 1st domain cell
    % to: Cell After last domain cell
    switch lower(setlim)
        case 'vanleer'
            delta(first-1:last+1,:) = VanLeer(V(first-1:last+1,:) - ...
                V(first-2:last,:),V(first:last+2,:) - V(first-1:last+1,:));
        otherwise
            delta = zeros(size(Uold));
    end

    % from: Cell Before 1st domain cell
    % to: last domain cell
    VL = zeros(size(V));    % Preallocate
    VR = VL;
    % Left and Right States
    VL(first-1:last,:) = V(first-1:last,:) + 0.5*delta(first-1:last,:);
    VR(first-1:last,:) = V(first:last+1,:) - 0.5*delta(first:last+1,:);

% ***********************************************************
% Step 3: Riemann Problem step

    % Solve Riemann Problem using FORCE flux
    F = zeros(size(Uold));
    [maxCx, F(first-1:last,:)] = FORCE(VL(first-1:last,:), ...
        VR(first-1:last,:), dx, dt);

% ***********************************************************
% Step 4: Update Solution step
    Unew(first-1:last,:) = Unew(first-1:last,:) - ...
        dt/dx*F(first-1:last,:);
    Unew(first:last+1,:) = Unew(first:last+1,:) + ...
        dt/dx*F(first-1:last,:);

    % Return CFL based on current dt, etc.
    CFL = dt*maxCx/dx;

end % Ends Gudunov

function [OUT] = VanLeer(d1,d2)
% Name:
%   VanLeer
% Description:
%   Limits SLOPE using Van Leer limiter
% Algorithms :
%   (Toro)
%   (LeVeque)
% Variables : (Scalars)
%   d1      [in]    : Difference Between current cell and left cell
%   d2      [in]    : Difference Between current cell and right cell
%   OUT     [out]   : VanLeer limited slope
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format. Original Code Can bee seen below.
%
eps = 1.0e-10;
w1 = abs(d1) + eps;
w2 = abs(d2) + eps;

% Van Leer
OUT = ( (w1.*d2 + w2.*d1)./(w1+w2) );

end

% ***************************************************************
% ***                    Original Method                      ***
% ***************************************************************
% Changing method to vectorized format makes the code more difficult
% to read. The original method here (deprecated) is more readable.
%

% function [Unew, CFL] = Gudunov(Uold, dx, dt, setlim)
% *****
% TEMPORARY ONLY !!!!
% last = length(Uold) - 2;
% first = 3;
% *****
%
% Set Max Variables
% maxCx = 0;
% Preallocate for speed
% V = zeros(size(Uold));
%
% ************************************************************
% Step 1: Convert from Conservative to Primitive variables
%     from: Cell Before 1st domain cell
%     to: Cell After last domain cell
% for i=first-1:last+1
%     V(i,:) = ConsToPrim(Uold(i,:));
% end
%
% ************************************************************
% Step 2: Boundary Extrapolated Values step
%
% Preallocate for speed
% delta = zeros(size(Uold));  % Difference between values
%
% Unew = Uold;    % Set next step = current step
%     from: Cell Before 1st domain cell
%     to: Cell After last domain cell
% for i=first-1:last+1
%     switch lower(setlim)
%         case 'vanleer'
%             Uses the VanLeer limiter
%             ( used during 2nd order accuracy extenstion step )
%
%             for j=1:3 % For each equation
%                 delta(i,j) = VanLeer(V(i,j)-V(i-1,j),V(i+1,j)-V(i,j));
%             end
%
%         otherwise
%             Uses constant values
%             ( used for 1st order and Data Reconstruction Step )
%
%             for j=1:3 % For each equation
%                 delta(i,j) = 0;
%             end
%     end
% end
%
%     from: Cell Before 1st domain cell
%     to: last domain cell
% for i=first-1:last
%     Preallocate for speed
%     VL = zeros(3);  % Left Riemann State
%     VR = zeros(3);  % Right Riemann State
%
%     for j = 1:3 % For each equation
%         Reconstructed Riemann States
%         VL(j) = V(i,j) + 0.5*delta(i,j);
%         VR(j) = V(i+1,j) - 0.5*delta(i+1,j);
%     end
%
% *************************************************************
% Step 3: Riemann Problem step
%
%     Solve Riemann Problem using FORCE flux
%     [Cmax, F] = FORCE(VL, VR, dx, dt);
%
% *************************************************************
% Step 4: Update Solution step
%
%     Gudunov Scheme
%         This clever method from Torrilhon attributes flux to
%         each cell without needing to know the next flux value
%         explicitly.
%     for j=1:3 % For each equation
%         Current Cell
%         Unew(i,j) = Unew(i,j) - dt/dx*F(j);
%         Downstream Cell
%         Unew(i+1,j) = Unew(i+1,j) + dt/dx*F(j);
%     end
%
%     maxCx = max([maxCx, Cmax]);
%
% end
%     Return CFL based on current dt, etc.
%     CFL = dt*maxCx/dx;
%
% end % Ends Gudunov
%
% function [OUT] = VanLeer(d1, d2)
%
% eps = 1.0e-10;      % Small Value to avoid divide by zero
% w1 = abs(d1)+eps;   % Used to find weighted averages
% w2 = abs(d2)+eps;
%
% Van Leer (with weighted average)
% OUT = ( (w1*d2+w2*d1)/(w1+w2) );
%
% end % Ends VanLeer
```

C.1.7 ConsToPrim.m

```
% --------         13 - Moment        ------
function [P] = ConsToPrim(U)
% Name:
%   ConsToPrim
% Description:
%   Converts from Conservative to Primitive equations for the 5 moment
%   model
% Algorithms :
%
% Variables :
%   U       [in]    : Conservative Variables (Vector)
%   P       [out]   : Primitive Variables (Vector)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Adapted for vector input/output
%

% Converts from Conservative variables to primitive ones
  rho = U(:,1);
  v = U(:,2)./rho;
  p = (2*U(:,3)-rho.*v.*v)/3;
  pxx = U(:,4) - rho.*v.*v;
  q = U(:,5) - U(:,3).*v - pxx.*v;

  P(:,1) = rho;
  P(:,2) = v;
  P(:,3) = p;
  P(:,4) = pxx;
  P(:,5) = q;
```

C.1.8 PrimToCons.m

```
% --------         13 - Moment        ------
function [U] = PrimToCons(P)
% Name:
%   PrimToCons
% Description:
%   Converts from Primitive to Conservative equations.
% Algorithms :
%
% Variables :
%   U       [out]   : Conservative Variables (Vector)
%   P       [in]    : Primitive Variables (Vector)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Adapted for vector input/output
%

  % Preallocate for speed
  U = zeros(size(P));

% Converts from Primitive to Conservative Variables
  rho = P(:,1);
  v = P(:,2);
  p = P(:,3);
  pxx = P(:,4);
  q = P(:,5);

  U(:,1) = rho;
  U(:,2) = rho.*v;
  U(:,3) = 0.5*rho.*v.*v+1.5*p;
  U(:,4) = rho.*v.*v + pxx;
  U(:,5) = (0.5*rho.*v.*v + 1.5*p).*v + pxx.*v + q;
```

C.1.9 FORCE.m

```
% --------        13 - Moment        ------ (CharSpeeds, IntercellFlux)
function [Cmax, F] = FORCE( VL, VR, dx, dt )
% Name:
%   FORCE
% Description:
%   Uses FORCE method to calculate the fluxes for
%   the 1D Hyperbolic Equations
% Algorithms :
%   (Toro)
%   (LeVeque)
% Variables :
%   VL      [in]    : Primitive Variables of Left Interface (Matrix)
%   VR      [in]    : Primitive Variables of Right Interface (Matrix)
%   F       [out]   : Force Flux (Matrix)
%   Cmax    [out]   : Max wave speed (scalar)
% Functions :
%   CharSpeeds
%   PrimToCons
%   IntercellFlux
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format.
%

% To Calculate Numerical Flux
% ************************************************************
% Step 1: Compute the Speeds (Sound, Left, Right, and Max)
aR = CharSpeeds(VR);
aL = CharSpeeds(VL);

uL = VL(:,2);
uR = VR(:,2);
Cmax = max([abs(0.5*(uL+uR)+max([aR,aL]))]);

% ************************************************************
% Step 2: Intercell Fluxes and Conserved Variables

[UL] = PrimToCons(VL);  % Conserved Quantities
[UR] = PrimToCons(VR);
[FL] = IntercellFlux(VL); % Intercell Fluxes
[FR] = IntercellFlux(VR);

% ************************************************************
% Step 3: Flux

% Preallocate for speed
F = zeros(size(VL));
FLF = F;
U = F;
FRI = F;

FLF = 0.5*(FL+FR) + 0.5*dx/dt*(UL-UR);
U = 0.5*(UL+UR) + 0.5*dt/dx*(FL-FR);

Vtmp = ConsToPrim(U);
FRI = IntercellFlux(Vtmp);
```

```
F = 0.5*(FLF+FRI);

end % Ends FORCE function

% ****************************************************************
% ***                       CharSpeeds                         ***
% ****************************************************************

function [Cmax] = CharSpeeds(V)
% Name:
%   CharSpeeds
% Description:
%   Calculates Characteristic Speeds for the flux calculation.
%   (1-D; based on PRIMITIVE variables).
% Algorithms :
%   C = 100*Vthi
% Variables :
%   V           [in]    : Primitive Variables of Current Cell (Matrix)
%   Cmax        [out]   : Maximum Wave speed for HLL flux calc (Scalar)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   March 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format.
%

% Unpack Primtive variables
rho = V(:,1);     % Density
u = V(:,2);       % Velocity
p = V(:,3);       % Scalar Pressure

% Calculate Values
theta = p./rho;  % Temperature (based on ideal gas law)

% Normalized speed of light is a constant times the 2 dimensional
% ion thermal velocity.
% Cmax = 100*sqrt(max(abs(2*theta)));

% Factor times the 2-D ion/electron thermal velocity.
Cmax = 5*sqrt(max(abs(2*theta)));

% % Characteristic Speed for Single-Fluid Simulations :
%     % Singular Closure
%     Cmax = 5*sqrt(max(abs(theta)));
%     % Realizable Closure
%     Cmax = 30*sqrt(max(abs(theta)));

end % Ends CharSpeed Function

% ****************************************************************
% ***                     IntercellFlux                        ***
% ****************************************************************

function [F] = IntercellFlux(V)
% Name:
```

```
%   IntercellFlux
% Description:
%   Calculates the intercell flux for the 1-D Hyperbolic
%   Equations based on PRIMITIVE variables.
% Algorithms :
%
% Variables :
%   V           [in]    : Primitive Variables of Current Cell (Matrix)
%   Cmax        [out]   : Maximum Wave speed for HLL flux calc (Scalar)
%   F           [out]   : Flux (Matrix)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   March 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format.
%

% Calculates flux quantities from primitive variables

% ****
% Get Closure Scheme from Global Variable
global SetClosure;      % Global Switch
    arg = SetClosure;   % Local Argument
% ****

% Calculates flux quantities from primitive variables

% Unpack Primtive variables
rho =   V(:,1);   % Density
u =     V(:,2);   % Velocity
p =     V(:,3);   % Scalar Pressure
pxx =   V(:,4);   % X-Component Pressure
q =     V(:,5);   % Heat Vector

% Calculate Intermediate Variables
theta = p./rho;  % Temperature (ideal gas relation)
sigma = pxx - p;% Shear Stress

% Non-Dimensional Terms
bigQ = q./(rho.*theta.^(3/2).*sqrt(1+sigma./p).*(1+0.25*sigma./p));

switch lower(arg)
  case 'real'
      % Realizable closure scheme
      D = 3*(1+bigQ.^2.*(22+bigQ.^2)./(32-bigQ.^2));
  otherwise
      % Defaults to Singular Closure Scheme
      D = 3*(1+0.5*bigQ.^2);
end

% Closure Terms
mxxx = q.*(1+sigma./p)./(1+0.25*sigma./p);
rxx = (rho.*D.*theta.^2)./3.*(1+sigma./p).*(5+2*sigma./p);

% Flux Calculation
F = zeros(size(V));   % Preallocate for speed.
```

```
F1 = rho.*u;
F2 = rho.*u.*u+pxx;
F3 = (0.5*rho.*u.*u + 1.5*p).*u + pxx.*u + q;
F4 = rho.*u.*u.*u + 3*pxx.*u + mxxx;
F5 = (0.5*rho.*u.*u + 1.5*p).*u.*u + (5/2)*pxx.*u.*u + 2*q.*u + mxxx.*u ...
    + 0.5*rxx;

F = [F1,F2,F3,F4,F5];

end % Ends IntercellFlux Function
```

C.1.10 Plots.m

```
function SinglePlot
% Thesis : Numerical Results Plots
% Quick Example of how to pull information and plot.
%

clc; clear all; close all;

% FIGURE 1 : Strong Shock Tube with Singular Closure
    % c1 = c2 = 1e-16
    % 50 : 1
    % 4000 pts (-1,2)
    filename = 'data/SingFluid13_sing_lo_';
    tag = 10;
    savefile = strcat(filename, num2str(tag),'.dat');
    try
        load(savefile,'-mat','IONS', 'ELEC', 'Ex', 'xc', ...
            'CON', 't', 'CFLmax', 'CFLset');
        % Convert to Primitive
        figure();
        Vout = ConsToPrim(IONS);
        plot(xc,Vout(:,1),'b-',xc,Vout(:,3),'r-','LineWidth',1,'LineSmoothing','on' );
        legend('\rho','p');
        xlabel('x');
        ylabel('Pressure & Density')
        str = sprintf('Pressure & Density at Time = %f',t);
        title(str);

        % Annotation
        str1(1) = {'13-Moment'};
        str1(2) = {'Single-Fluid'};
        str1(3) = {'Singular Closure'};
        str1(4) = {'Strong Shock'};
        text(0,40,str1)
    catch
        fprintf('Loading Failed...\n');
    end
```

C.2 5-Moment Electrostatic Simulation

To switch to 5-Moment code, replace functions beginning with the tag “13-Moment” with their 5-Moment equivalents. Only these new functions are listed here.

C.2.1 InitialConditions.m (5-Moment)

```
% --------        5 - Moment       ------

% ************************************************************
% ***                  INITIAL CONDITIONS                  ***
% ************************************************************

function [IONS, ELEC, Ex, xc, CON, outputStr] = ...
    InitialConditions(nc, xl, xr, type)
% Name:
%   InitialConditions
% Description:
%   Sets the initial conditions based on the type passed.
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   July 1st, 2011 - Updated for Two-Fluid Model
%

% Set Number of Ghost Cells (Don't change unless necessary.)
NumGhostCells = 2; % (On Each Side)

nc = nc + NumGhostCells*2;  % Updated Number of Cells
nf = nc+1;                  % Number of faces
xf = linspace(xl,xr,nf);% Vector of face x-positions
dx = mean(diff(xf));        % Spatial resolution

first = NumGhostCells + 1;  % Beginning of Domain
last = nc - NumGhostCells;  % End of Domain

% Vector of center positions
xc = xl+dx/2:dx:xr-dx/2;
outputStr = '';
% For each simulation type,
    switch lower(type)
        otherwise
            outputStr = []; % outputStr def'd as matrix
            for i=1:60  fprintf('*'); end;  fprintf('\n');
            fprintf('%40s\n','Shock Tube Simulation');
            for i=1:60  fprintf('*'); end;  fprintf('\n');
            fprintf('\n\t**** CONSTANTS ****\n');
            % Ion and electron masses and charges
            qI = 1;
            qE = -1;
            mE = 1/1836;
            mI = 1;
            Ex = 0*ones(nc,1);  % Applied E-Field
                    % Applied E-Field option has been dropped in this
                    % version.

            %Coefficiencts
            c1 = 10^-1;    % Plasma Size / Ion Larmor Radius
            c2 = 100;   % Ion Larmor Radius / Debye Length

            fprintf('%20s:\t%g\n','Electron Mass',mE);
            fprintf('%20s:\t%g\n','Ion Mass',mI);
            fprintf('%20s:\t%g\n','Electron Charge',qE);
            fprintf('%20s:\t%g\n','Ion Charge',qI);
            fprintf('%40s:\t%g\n','Plasma Size / Ion Larmor Radius',c1);
            fprintf('%40s:\t%g\n','Ion Larmor Radius / Debye Length',c2);
```

```
% **** IONS ****
fprintf('\n\t**** IONS ****\n');
% Left State
rhoL = 1;
pL = 0.5;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Pressure',pL);

% Right State
rhoR = .125;
pR = 0.05;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Pressure',pR);

% Set values based on x-position
for i=1:nc
    v = vL;    % Velocity

    if xc(i) < 0.0  % State 1: Upstream of Shock
        rho = rhoL; % Density
        p = pL;   % Scalar Pressure
    else            % State 2: Downstream of shock
        rho = rhoR; % Density
        p = pR;    % Scalar Pressure
    end
    % Convert to CONSERVATIVE variables and store to U
    U(i,1) = rho;
    U(i,2) = rho*v;
    U(i,3) = 0.5*rho*v*v+1.5*p;
end
IONS = U;

% **** ELECTRONS ****
fprintf('\n\t**** ELECTRONS ****\n');
% Left State
rhoL = 1*mE/mI;
pL = 0.5;
vL = 0;
fprintf('\tLeft State:\n');
fprintf('%20s:\t%g\n','Density',rhoL);
fprintf('%20s:\t%g\n','Velocity',vL);
fprintf('%20s:\t%g\n','Pressure',pL);

% Right State
rhoR = .125*mE/mI;
pR = 0.05;
vR = 0;
fprintf('\tRight State:\n');
fprintf('%20s:\t%g\n','Density',rhoR);
fprintf('%20s:\t%g\n','Velocity',vR);
fprintf('%20s:\t%g\n','Pressure',pR);

% Set values based on x-position
for i=1:nc
```

```
                v = vL;    % Velocity

                if xc(i) < 0.0  % State 1: Upstream of Shock
                    rho = rhoL; % Density
                    p = pL;   % Scalar Pressure
                else            % State 2: Downstream of shock
                    rho = rhoR; % Density
                    p = pR;    % Scalar Pressure
                end
                % Convert to CONSERVATIVE variables and store to U
                U(i,1) = rho;
                U(i,2) = rho*v;
                U(i,3) = 0.5*rho*v*v+1.5*p;
            end
            ELEC = U;

    end % Ends Switch

    % Package all scalars into a single CONSTANT vector
    CON(1) = nc;            % Number of Cells
    CON(2) = NumGhostCells; % Number of Ghost Cells
    CON(3) = dx;            % Spatial derivative step size
    CON(4) = first;         % Beginning of Domain
    CON(5) = last;          % End of Domain
    CON(6) = qI;            % Ion Charge
    CON(7) = qE;            % Electron Charge
    CON(8) = mI;            % Ion Mass
    CON(9) = mE;            % Electron Mass
    CON(10) = c1;           % Plasma Size / Ion Larmor Radius
    CON(11) = c2;           % Ion Larmor Radius / Debye Length
end
```

C.2.2 Sfunc - part of AdvanceToPoint.m (5-Moment)

```
function [OUT] = Sfunc(IN, Ex, CON, species)
% Name:
%   Sfunc
% Description:
%   Calculates source terms to be included in time-integration.
% Algorithms :
%
% Variables :
%   IN      [in]    : Conservative Variables (Matrix)
%   Ex      [in]    : Electric Field Strength (Matrix)
%   CON     [in]    : Constants (Vector)
%   species [in]    : Switch : 'ions' or otherwise
%   OUT     [out]   : Source Contributions (Matrix)
% Comments :
%   Shaun Gilliam
%   July 1st, 2011 - Document Creation
%

    % Update Variables based on source terms
    % **** using vectorized format
    nc = CON(1);

    switch lower(species)
        case 'ions'
            q = CON(6);            % Ion Charge
            m = CON(8);            % Ion Mass
        otherwise
            q = CON(7);            % Electron Charge
            m = CON(9);            % Electron Mass
    end % Ends Switch

    c1 = CON(10);           % Plasma Size / Ion Larmor Radius
    c2 = CON(11);           % Ion Larmor Radius / Debye Length

    % Convert to Primitive Variables
    V = zeros(size(IN));
    V = ConsToPrim(IN);

    % Grab Vectors
    n = IN(:,1)/m;      % Number Density
    vel = V(:,2);

    % Source Terms
    S1 = zeros(size(n));
    S2 = c1*q*n.*Ex;
    S3 = c1*q*n.*Ex.*vel;

    OUT = [S1,S2,S3];   % Output Source Terms

end
```

C.2.3 ConsToPrim.m (5-Moment)

```
% --------         5 - Moment        ------
function [P] = ConsToPrim(U)
% Name:
%   ConsToPrim
% Description:
%   Converts from Conservative to Primitive equations for the 5 moment
%   model
% Algorithms :
%
% Variables :
%   U       [in]    : Conservative Variables (Vector)
%   P       [out]   : Primitive Variables (Vector)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Adapted for vector input/output
%

% Preallocate for speed
  P = zeros(size(U));

% Converts from Conservative variables to primitive ones
  rho = U(:,1);
  v = U(:,2)./rho;
  p = (2*U(:,3)-rho.*v.*v)/3;

  P(:,1) = rho;
  P(:,2) = v;
  P(:,3) = p;
```

C.2.4 PrimToCons.m (5-Moment)

```
% --------         5 - Moment        ------
function [U] = PrimToCons(P)
% Name:
%   PrimToCons
% Description:
%   Converts from Primitive to Conservative equations for the 5 moment
%   model
% Algorithms :
%
% Variables :
%   U       [out]   : Conservative Variables (Vector)
%   P       [in]    : Primitive Variables (Vector)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Adapted for vector input/output
%

% Preallocate for speed
  U = zeros(size(P));

% Converts from Primitive to Conservative Variables
  rho = P(:,1);
  v = P(:,2);
  p = P(:,3);

  gamma = 5/3;
  energy = 0.5*rho.*v.*v + p/(gamma-1);

  U(:,1) = rho;
  U(:,2) = rho.*v;
  U(:,3) = energy;
```

C.2.5 FORCE.m (5-Moment)

```
% --------         5 - Moment        ------ (CharSpeeds, IntercellFlux)
function [Cmax, F] = FORCE( VL, VR, dx, dt )
% Name:
%   FORCE
% Description:
%   Uses FORCE method to calculate the fluxes for
%   the 1D Hyperbolic Equations
% Algorithms :
%   (Toro)
%   (LeVeque)
% Variables :
%   VL      [in]    : Primitive Variables of Left Interface (Matrix)
%   VR      [in]    : Primitive Variables of Right Interface (Matrix)
%   F       [out]   : Force Flux (Matrix)
%   Cmax    [out]   : Max wave speed (scalar)
% Functions :
%   CharSpeeds
%   PrimToCons
%   IntercellFlux
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   April 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format.
%

% To Calculate Numerical Flux
% ***********************************************************
% Step 1: Compute the Speeds (Sound, Left, Right, and Max)
aR = CharSpeeds(VR);
aL = CharSpeeds(VL);

uL = VL(:,2);
uR = VR(:,2);
Cmax = max([abs(0.5*(uL+uR)+max([aR,aL]))]);

% ***********************************************************
% Step 2: Intercell Fluxes and Conserved Variables

[UL] = PrimToCons(VL);  % Conserved Quantities
[UR] = PrimToCons(VR);
[FL] = IntercellFlux(VL); % Intercell Fluxes
[FR] = IntercellFlux(VR);

% ***********************************************************
% Step 3: Flux

% Preallocate for speed
F = zeros(size(VL));
FLF = F;
U = F;
FRI = F;

FLF = 0.5*(FL+FR) + 0.5*dx/dt*(UL-UR);
U = 0.5*(UL+UR) + 0.5*dt/dx*(FL-FR);

Vtmp = ConsToPrim(U);
FRI = IntercellFlux(Vtmp);
```

```
F = 0.5*(FLF+FRI);

end % Ends FORCE function

% ****************************************************************
% ***                       CharSpeeds                         ***
% ****************************************************************

function [Cmax] = CharSpeeds(V)
% Name:
%   CharSpeeds
% Description:
%   Calculates Characteristic Speeds for the flux calculation.
%   (1-D; based on PRIMITIVE variables).
% Algorithms :
%   C = 100*Vthi
% Variables :
%   V           [in]    : Primitive Variables of Current Cell (Matrix)
%   Cmax        [out]   : Maximum Wave speed for HLL flux calc (Scalar)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   March 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format.
%

% Unpack Primtive variables
rho = V(:,1);     % Density
u = V(:,2);       % Velocity
p = V(:,3);       % Scalar Pressure

% Calculate Values
theta = p./rho;  % Temperature (based on ideal gas law)

% Normalized speed of light is a constant times the 2 dimensional
% ion thermal velocity.
Cmax = 5*sqrt(max(abs(2*theta)));

end % Ends CharSpeed Function

% ****************************************************************
% ***                    IntercellFlux                         ***
% ****************************************************************

function [F] = IntercellFlux(V)
% Name:
%   IntercellFlux
% Description:
%   Calculates the intercell flux for the 1-D Hyperbolic
%   Equations based on PRIMITIVE variables.
% Algorithms :
%
% Variables :
%   V           [in]    : Primitive Variables of Current Cell (Matrix)
%   Cmax        [out]   : Maximum Wave speed for HLL flux calc (Scalar)
%   F           [out]   : Flux (Matrix)
% Functions :
%
% Restrictions :
%
% Comments :
%   Shaun Gilliam
%   March 26, 2011 - Document Creation
%   June 20, 2011 - Updated method to take advantage of improved speed of
%                   vectorized format.
%

% Calculates flux quantities from primitive variables

% Unpack Primtive variables
rho =   V(:,1);   % Density
u =     V(:,2);   % Velocity
p =     V(:,3);   % Scalar Pressure

gamma = 5/3;
energy = 0.5*rho.*u.*u + p/(gamma-1);

% Flux Calculation
F = zeros(size(V));   % Preallocate for speed.

F1 = rho.*u;
F2 = rho.*u.*u+p;
F3 = (energy+p).*u;

F = [F1,F2,F3];

end % Ends IntercellFlux Function
```

Appendix D

DETERMINING CHARACTERISTIC SPEEDS WITH MATHEMATICA

As an example, the code used to generate the plot for the characteristic speeds of the singular closure scheme is presented here. Changing to the realizable closure scheme only requires changing the calculation of "*bigD*".

Characteristic Speeds of the Pearson - IV Singular Closure Scheme

Calculate Jacobian A (u) :

In[1]:=
```
Remove["Global`*"]
JacobianMatrix[f_List ? VectorQ, x_List] := Outer[D, f, x] /; Equal @@ (Dimensions /@ {f, x});
```

$$\theta = \frac{P}{\rho};$$

$$\sigma = Pxx - P;$$

$$mxxx = q * \frac{1 + \sigma / P}{1 + \frac{1}{4}\,\sigma / P};$$

$$bigQ = \frac{q}{\rho\,\theta^{3/2}\,\sqrt{1 + \sigma / P}\,\left(1 + \frac{1}{4}\,\sigma / P\right)};$$

$$bigD = 3\left(1 + \frac{1}{2}\,bigQ^2\right);$$

$$Rxx = \frac{\rho\,bigD\,\theta^2}{3}\,(1 + \sigma / P)\,(5 + 2\,\sigma / P);$$

$$U = Flatten\left[\begin{pmatrix} \rho \\ \rho\,u \\ \left(\frac{1}{2}\,\rho\,u^2 + \frac{3}{2}\,P\right) \\ \rho\,u^2 + Pxx \\ \left(\frac{1}{2}\,\rho\,u^2 + \frac{3}{2}\,P\right)\,u + Pxx\,u\ +\ q \end{pmatrix}\right];$$

$$F = Flatten\left[\begin{pmatrix} \rho\,u \\ \rho\,u^2 + Pxx \\ \left(\frac{1}{2}\,\rho\,u^2 + \frac{3}{2}\,P\right)\,u + Pxx\,u\ +\ q \\ \rho\,u^3 + 3\,Pxx\,u\ +\ mxxx \\ \left(\frac{1}{2}\,\rho\,u^2 + \frac{3}{2}\,P\right)\,u^2 + \frac{5}{2}\,Pxx\,u^2 + 2\,q\,u\ +\ mxxx\,u\ +\ \frac{1}{2}\,Rxx \end{pmatrix}\right];$$

$$V = Flatten\left[\begin{pmatrix} \rho \\ u \\ P \\ Pxx \\ q \end{pmatrix}\right];$$

```
jacFoV = FullSimplify[JacobianMatrix[F, V]];
jacUoV = FullSimplify[JacobianMatrix[U, V]];
jacVoU = FullSimplify[Inverse[jacUoV]];
jacFoU = FullSimplify[jacFoV.jacVoU];
jacFoU // MatrixForm
```

Out[16]//MatrixForm=

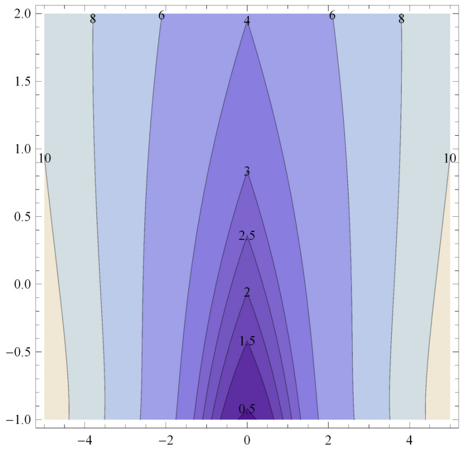
$$\left(\begin{matrix} 0 \\ 0 \\ 0 \\ -\frac{(3\,P-Pxx)\,u\,\left(-4\,q\,u\,\rho+(3\,P+Pxx)\,\left(Pxx-u^2\,\rho\right)\right)}{(3\,P+Pxx)^2\,\rho} \\ -\frac{Pxx\,(3\,P+Pxx)^3\,(3\,P+2\,Pxx)-(3\,P+Pxx)\,u\,\left(4\,\left(9\,P^2+12\,P\,Pxx+5\,Pxx^2\right)\,q+(3\,P+Pxx)\,\left(9\,P^2+12\,P\,Pxx+7\,Pxx^2\right)\,u\right)\,\rho+u^2\,\left(8\,(3\,P+5\,Pxx)\,q^2+24\,Pxx\,(3\,P+Pxx)\,q\,u-3\,(P-Pxx)\,(3\,P+Pxx)^2\,u^2\right)\,\rho^2}{2\,(3\,P+Pxx)^3\,\rho^2} \end{matrix}\right.\quad (3\,P$$

Calculate Maximum Characteristic Speed C_{max}

```
In[19]:= Cmax[x_, y_] := Module[{a = y + 1, b = x, c = 1, d = 1, e = 0, tempJac, tempEigs},
           tempJac = jacFoU /. {Pxx → a, q → b, P → c, ρ → d, u → e};
           tempEigs = N[Eigenvalues[tempJac]];
           Max[Abs[Re[tempEigs]]]
          ]

         ContourPlot[Cmax[x, y], {x, -5, 5}, {y, -1, 2},
          Contours → {.5, 1.5, 2, 2.5, 3, 4, 6, 8, 10, 15, 20, 40, 100, 500}, ColorFunctionScaling → True, ContourLabels → True]
```

Out[20]=



BIBLIOGRAPHY

[1] H. Grad, On the Kinetic Theory of Rarefied Gases. Comm. Pure Appl. Math. 2 Issue 4, p331-407 (1949).

[2] M. Torrilhon, Hyperbolic Moment Equations in Kinetic Gas Theory Based on Multi-Variate Pearson-IV-Distributions, Commun. Comput. Phys., 7 (2010), p. 639-673.

[3] U. Shumlak, Approximate Riemann solver for the two-fluid plasma model, Journ. of Comp. Phys., 187 (2003), p. 620-638.

[4] Y. Nagahara, The PDF and CF of Pearson type IV distributions and the ML estimation of the parameters, Stat. & Prob. Lett. 43 (1999), p. 251-264.

[5] J. Heinrich, A Guide to the Pearson Type IV Distribution, CDF Memo 6820, www-cdf.final.gov/physics/notes/cdf6820_pearson4.pdf, (2004).

[6] K. Pearson, Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material, Phil. Trans. R. Soc. Lond. A (1895) 186, p. 343-414.

[7] J. G. Papastavridis, Tensor Calculus and Analytical Dynamics, CRC Press, Boca Raton, FL (1999).

[8] D. Lovelock, H. Rund, Tensors, Differential Forms, and Variational Principles, John Wiley & Sons, New York (1975).

[9] J. G. Papastavridis, Tensor Calculus and Analytical Dynamics, CRC Press, Boca Raton, FL (1999).

[10] G. T. Mase, R. E. Smeler, G. E. Mase, Continuum Mechanics for Engineers Third Edition, CRC Press, Boca Raton, Fl (2010).

[11] S. I. Braginskii, Transport Processes in a Plasma, Revs. of Plasma Phys. Vol 1 (1965), p. 205-311.

[12] P. L. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, Journ. of Comp. Phys., 43 (1981), p357-372.

[13] S. L. Brown, Approximate Riemann Solvers For Moment Models of Dilute Gases, Dissertation, University of Washington, (1996).

[14] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics Third Edition, Springer, New York (2009).

[15] R. J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, New York (2002).

[16] J. Loverich, A Finite Volume Algorithm for the Two-Fluid Plasma System, 16th AIAA Comp. Fluid Dyn. Conf., June 23-26 (2003).

[17] J. N. Kutz, AMATH 581 Practical Scientific Computing, Dept. of App. Math., Version 1.1, (2005).

[18] J. J. Leader, Numerical Analysis and Scientific Computation, Pearson Assison Wesley, Boston (2004).

[19] B. Srinivasan, Numerical Methods for 3-Dimensional Magnetic Confinement Configurations using Two-Fluid Plasma Equations, Dissertation, University of Washington, (2010).

[20] R. J. Goldston, P. H. Rutherford, Introduction to Plasma Physics, Taylor & Francis Group, New York (1995).

[21] A. Hakim, J. Loverich, and U. Shumlak, A high resolution wave propagation scheme for ideal Two-Fluid plasma equations, Journ. of Comp. Phys., 219 (2006), p. 418-2006.

[22] M. Brio, C. C. Wu, An Upwind Differencing Scheme for the Equations of Ideal Magnetohydrodynamics, Journ. of Comp. Phys., 75 (1988), p.400-422.