

A Comparison between the Discontinuous Galerkin Method  
and the High Resolution Wave Propagation Algorithm  
for the Full Two-Fluid Plasma Model

Bhuvana Srinivasan

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

Master of Science

University of Washington

2006

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

Bhuvana Srinivasan

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

---

Nathan Kutz

Date: \_\_\_\_\_

In presenting this thesis in partial fulfillment of the requirements for a master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purpose or by any means shall not be allowed without my written permission.

Signature\_\_\_\_\_

Date\_\_\_\_\_

University of Washington

**Abstract**

A Comparison between the Discontinuous Galerkin Method  
and the High Resolution Wave Propagation Algorithm  
for the Full Two-Fluid Plasma Model

Bhuvana Srinivasan

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

The high resolution wave propagation method and the Runge-Kutta discontinuous Galerkin method are compared for applications of several hyperbolic equation sets. For the discontinuous Galerkin algorithm, the conserved variable is defined as a linear combination of a set of basis functions and the selection of these basis functions sets the spatial order of the solution. A Runge-Kutta time integration scheme is used with this method. The high resolution wave propagation algorithm is a finite volume method that uses cell averages to define the conserved variable and it is essentially similar to a low order discontinuous Galerkin method. Both methods compute the numerical flux at the cell edges with an approximate Riemann solver. After benchmarking these numerical methods with simpler equation sets such as the stress-strain relations in homogeneous and layered media, and the Dispersive Euler equations, they are applied to the full Two-Fluid Plasma Model. The two algorithms are compared for stability, accuracy, convergence and computational expense.

## TABLE OF CONTENTS

List of Figures . . . . .	iii
List of Tables . . . . .	v
Chapter 1: Introduction . . . . .	1
1.1 Introductory Remarks . . . . .	1
1.2 Software Used . . . . .	3
Chapter 2: High Resolution Wave Propagation Method . . . . .	5
2.1 First Order Scheme . . . . .	5
2.2 High Resolution Corrections . . . . .	8
2.3 Source Term Handling . . . . .	10
Chapter 3: Discontinuous Galerkin Method . . . . .	13
3.1 The Base Scheme . . . . .	13
3.2 Selection of Basis Functions . . . . .	15
3.3 Limiters for the Discontinuous Galerkin Scheme . . . . .	16
3.4 Time Stepping Scheme . . . . .	18
Chapter 4: Explored Hyperbolic Equation Systems . . . . .	19
4.1 Stress-Strain Relations . . . . .	19
4.2 Dispersive Euler Equations . . . . .	20
Chapter 5: Full Two-Fluid Plasma Model . . . . .	22
5.1 Basic Equations . . . . .	22
5.2 The Two-Fluid Source Terms . . . . .	23
5.3 Radial Source Terms for 1D Cylindrical Coordinates . . . . .	25
Chapter 6: Applications and Results . . . . .	27
6.1 Strain Waves . . . . .	27
6.1.1 Description of Problem . . . . .	27
6.1.2 Results . . . . .	28

6.2	Electron Acoustic Waves . . . . .	39
6.2.1	Analytical Solution . . . . .	39
6.2.2	Description of Problem . . . . .	41
6.2.3	Results . . . . .	42
6.3	Radial Z-Pinch . . . . .	58
6.3.1	Description of Problem . . . . .	58
6.3.2	Results . . . . .	61
Chapter 7:	Conclusions . . . . .	74
7.1	Concluding Remarks . . . . .	74
7.2	Further Study . . . . .	76
Bibliography	. . . . .	78

## LIST OF FIGURES

Figure Number	Page
6.1 Strain Wave: Strain Vs X for both algorithms with 1000 grid points at t=40	30
6.2 Strain Wave: Strain Vs X to compare Discontinuous Galerkin method solutions with and without higher order coefficients, 1000 cells, t=40 . . . . .	31
6.3 Strain Wave: Strain Vs X for both algorithms with 300 cells at t=40 . . . . .	33
6.4 Strain Wave: $l_2$ -norm of strain Vs Grid Resolution for both algorithms . . . . .	34
6.5 Strain Wave: Computational Time Vs Grid Resolution for both algorithms . . . . .	36
6.6 Electron Acoustic Pulse: An exact solution at t=1000 for N=5000 . . . . .	43
6.7 Electron Acoustic Pulse: Initial condition for N=9 . . . . .	44
6.8 Electron Acoustic Pulse: u Vs x for 40 cells with $c_s = \sqrt{2}$ and $\omega_c = 10$ for both algorithms . . . . .	46
6.9 Electron Acoustic Pulse: u Vs x for 120 cells with $c_s = \sqrt{2}$ and $\omega_c = 10$ for both algorithms . . . . .	47
6.10 Electron Acoustic Pulse: $l_2$ -norm of velocity Vs Grid Resolution with $c_s = \sqrt{2}$ and $\omega_c = 10$ for both algorithms . . . . .	48
6.11 Electron Acoustic Pulse: Computational Time Vs Grid Resolution with $c_s = \sqrt{2}$ and $\omega_c = 10$ for both algorithms . . . . .	50
6.12 Electron Acoustic Pulse: u Vs x with 100 cells, $c_s = \sqrt{2}$ and $\omega_c = 100$ for both algorithms . . . . .	53
6.13 Electron Acoustic Pulse: u Vs x with 512 cells, $c_s = \sqrt{2}$ and $\omega_c = 100$ for wave propagation method . . . . .	54
6.14 Electron Acoustic Pulse: u Vs x with 100 and 512 cells, $c_s = \sqrt{2}$ and $\omega_c = 100$ for Wave Propagation algorithm using semi-implicit source term handling . . . . .	55
6.15 Electron Acoustic Pulse: u Vs x with 160 cells, $c_s = \sqrt{2}$ and $\omega_c = 10$ for wave propagation method using implicit source term handling . . . . .	56
6.16 Radial Z-pinch: Two-Fluid Equilibrium Initial Conditions . . . . .	60
6.17 Radial Z-pinch: Two-Fluid Equilibrium with 32 cells . . . . .	63
6.18 Radial Z-pinch: Two-Fluid Equilibrium with 100 cells . . . . .	64
6.19 Radial Z-pinch: Two-Fluid Equilibrium with 256 cells . . . . .	65
6.20 Radial Z-Pinch: $l_2$ -norm of magnetic field Vs Grid Resolution for both algorithms . . . . .	67

6.21 Radial Z-Pinch: Computational Time Vs Grid Resolution for both algorithms	69
6.22 Radial Z-pinch: Two-Fluid equilibrium with 128 cells after a long time, t=500	70
6.23 Radial Z-pinch: Two-Fluid equilibrium to compare the ability of the algorithms to maintain equilibrium for the same computational effort, when run out to a long time. . . . .	71

## LIST OF TABLES

Table Number		Page
6.1	Some computational times for the Strain Waves problem . . . . .	35
6.2	Some computational times for the <i>Dispersive Euler</i> Electron Acoustic Wave problem . . . . .	49
6.3	Some computational times for the Two-Fluid Plasma Model Radial Z-Pinch problem . . . . .	68

## Chapter 1

## INTRODUCTION

**1.1 Introductory Remarks**

This thesis constitutes a study of algorithms that could be applied to solving inhomogeneous partial differential equations containing hyperbolic homogeneous parts. There are a number of equation systems that are either hyperbolic or contain hyperbolic parts such as the Euler equations, Maxwell's equations, Stress-strain relations, etc. It is useful to investigate algorithms that provide accurate and precise solutions to such equation sets with reasonable computational effort and minimum diffusive and dispersive errors. The equation systems studied here are described by the following conservation law with source terms (also known as *balance laws*),

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}, \quad (1.1)$$

where  $\mathbf{Q}$  represents the conserved variables,  $\mathbf{F}$  represents the fluxes and  $\mathbf{S}$  represents the source terms. The number of conserved variables, flux terms and source terms is determined by the number of balance laws in the equation set. If the equation system contains  $m$  balance laws, then there are  $m$  conserved variables and source terms. Due to the spatial dependence of the flux, there are  $m$  flux terms for each dimension.

The *flux Jacobian* of a one-dimensional balance law is defined as

$$\mathbf{A} \equiv \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \quad (1.2)$$

$\mathbf{A}$  is an  $m \times m$  matrix for a set of  $m$  balance laws. If the flux Jacobian has real eigenvalues and a complete set of right eigenvectors[5, 7], that are linearly independent, then the equation system can be stated to have hyperbolic, homogenous parts. If the matrix  $\mathbf{A}$  has real *and* distinct eigenvalues, then the homogeneous part is called *strictly hyperbolic*.

Hyperbolic equations can have discontinuous solutions even if the initial conditions are smooth, and this makes the approximation of the exact solution difficult. Lower order methods do not capture such discontinuities effectively, while higher order methods tend to develop large oscillations around the sharp gradients. The key is to develop and use algorithms that would accurately capture such discontinuities as the solution evolves. Two such algorithms are compared in this thesis: the high resolution wave propagation algorithm [4] and the discontinuous Galerkin algorithm [10].

The wave propagation algorithm is a finite volume method that involves discretizing a domain into cells and keeping track of the conserved variables,  $\mathbf{Q}$ , at each time step. Cell averages are used to approximate the conserved variables as a result of which the solution need not be continuous at the cell edges. A Riemann problem is solved at each cell interface using the interface value of  $\mathbf{Q}$  (in this case the average value of the conservative variable in that cell), and this is used to compute the numerical fluxes at each of the cell edges. Using these computed values, the solution is then updated in a manner similar to the central difference scheme that is described in detail in Chapter 2.

The discontinuous Galerkin algorithm is similar, in some respects, to the wave propagation algorithm, but it is a finite element method as it uses quadrature points within each cell. While the wave propagation algorithm is of second order when the flux corrections are implemented, the discontinuous Galerkin method can be of higher spatial orders. For certain equation sets, all the physics might not be accurately captured at a lower spatial order, so higher order methods might be more advantageous. Just like the case of the wave propagation algorithm, the Riemann problem is solved at the cell interface. However, instead of using cell averages and second order corrections, the balance law in each cell is multiplied by a set of basis functions and the order of the polynomial determines the spatial order of the method. A Runge-Kutta time stepping scheme is implemented here for the solution update. This method is detailed in Chapter 3.

This thesis compares the two algorithms, the high resolution wave propagation algorithm [7, 6] and the discontinuous Galerkin algorithm [2, 10], to solve hyperbolic equations such as the stress-strain relations in homogeneous and layered media, the *Dispersive Euler* equation system and the ideal Two-Fluid plasma model. The algorithms are compared for accuracy,

stability, convergence and equally important, computational expense. It is known that the discontinuous Galerkin method can offer higher orders of accuracy, however, is the physics captured at these high orders of accuracy significantly different from that of the wave propagation method and is it worth the computational effort involved? Is the wave propagation method superior to the discontinuous Galerkin method for any applications? These questions are investigated for several hyperbolic equation sets to determine which method is superior in what context.

The wave propagation method and the discontinuous Galerkin method fall into the category of *Godunov methods* which require that the Riemann problem be solved at each cell interface in order to advance the solution. A study regarding the handling of source terms also constitutes an essential part of the comparisons. Source term handling is detailed in Chapter 2. The wave propagation method handles the source terms by performing a splitting. There are limitations associated with the source term splitting that do not appear with the discontinuous Galerkin method which is an unsplit method. Using source splitting causes phase errors in the solution with the wave propagation method when the grid resolution is low or when the oscillation frequency is high such that the Nyquist frequency is violated for proper sampling. The effects of the split and unsplit schemes are described in Chapter 6 with the results for each application. However, for systems with a single characteristic speed, it appears that the wave propagation method captures the solution almost exactly when used with a Courant number of 1 without any diffusive or dispersive errors. These algorithms are described in detail in the following chapters. Chapters 4 and 5 describe the equation systems that are used for these comparisons. The applications and results of the comparisons are detailed in Chapter 6.

## **1.2 Software Used**

The wave propagation and the discontinuous Galerkin algorithms are written using the same computing language, C++, for best comparisons. Both algorithms use the exact same framework, and the only differences lie in the calculation of the interface fluxes and in the time stepping schemes used. The algorithms call the exact same set of Riemann problem solvers.

The discontinuous Galerkin method uses the quadrature points to get the interface values to the desired order while the Wave Propagation method uses cell averages. For the Discontinuous Galerkin method, the Riemann problem is called as many times as the chosen order of the algorithm to calculate each of the coefficients of the conserved variable. For the wave propagation algorithm, the Riemann problem is only called once. This is an indication that the discontinuous Galerkin method is expected to take more computing time than the wave propagation method.

The time-stepping also varies depending on the algorithm used. The discontinuous Galerkin algorithm uses the Runge-Kutta time integration scheme while the wave propagation algorithm uses an update formula for the conserved variable that resembles the central difference formula.

`wave_step` and `rkdg_step` are the only functions in the code that contain algorithm-specific calculations. These functions have been checked to make sure that there are no additional unnecessary computations being performed. Both algorithms call the same functions for the source and flux terms. For the initializations and applications of boundary conditions, both algorithms call the exact same `qinit.cc` and `bc.cc` files.

## Chapter 2

**HIGH RESOLUTION WAVE PROPAGATION METHOD**

The high resolution wave propagation algorithm can be applied to equations that have hyperbolic parts. Such equations, as mentioned previously, are of the form:

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}, \quad (2.1)$$

This chapter describes the wave propagation algorithm and the higher order corrections that are implemented to increase the order of accuracy from first to second. The method detailed here is for one spatial dimension and it can be extended to account for multiple dimensions with the use of normal and transverse corrections in each dimension. The hyperbolic equation sets that are studied in this thesis include the stress-strain relations for homogeneous and layered media, the Dispersive Euler equation system and the five-moment ideal two-fluid plasma equations. The application of this method to the stated equation systems will be discussed in the chapters to follow. LeVeque describes this method in much greater detail in [1, 6, 7]. This chapter follows the algorithm as detailed by LeVeque and Hakim.

**2.1 First Order Scheme**

The scheme is detailed in one spatial dimension here since the applications considered are in one dimension only. It can be extended to account for multiple dimensions. In one dimension, a homogeneous hyperbolic equation is written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0, \quad (2.2)$$

where  $\mathbf{Q}$  represents the conserved variable and  $\mathbf{F}$  represents the flux in the  $X$  direction. The domain that this equation is discretized on is defined within the boundaries,  $[x_a, x_b]$ . The cells are introduced as  $I_i = [x_{i-1/2}, x_{i+1/2}]$ , for each cell interval, where  $x_{i-1/2}$  and  $x_{i+1/2}$  are the coordinates at the left and right edges of each cell. The cell center is defined as  $(x_i)$ ,

where  $x_i \equiv (x_{i-1/2} + x_{i+1/2})/2$ . Now, taking the conservation law defined in Eq. (2.2), and integrating it over cell  $I_i$  from time  $t_n$  to  $t_{n+1}$  gives the update formula

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left( [\mathcal{F}]_{i+1/2}^{n+1/2} - [\mathcal{F}]_{i-1/2}^{n+1/2} \right) \quad (2.3)$$

where  $Q_i^n$  represents the value of the conserved variable based on the cell average,

$$Q_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{Q}(x, t) dx. \quad (2.4)$$

Here  $\Delta x \equiv x_{i+1/2} - x_{i-1/2}$ ,  $\Delta t \equiv t_{n+1} - t_n$  and the *numerical flux* at each of the cell interfaces,  $[\mathcal{F}]$ , is defined as

$$[\mathcal{F}]_{i-1/2}^{n+1/2} \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \mathbf{F}(Q(x_{i-1/2}, t), x_{i-1/2}) dt. \quad (2.5)$$

It can be seen that the expression in Eq. (2.3) resembles the central difference formula. This equation is a general update formula for finite volume schemes. The numerical flux can be updated by using several different approaches and the choice of the flux update forms the basis for the various finite volume schemes. This thesis focuses on a specific finite volume method using an *F-Wave* approach that is introduced and described in [1], the *High Resolution Wave Propagation method*. This method approximates the value of the conserved variable in a given cell as the cell average. Therefore, for cells sharing a given interface, the value at that interface will be discontinuous in general. A Riemann problem needs to be solved at each cell edge to determine the numerical flux at each interface.

For the homogeneous, hyperbolic equation described by:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0, \quad (2.6)$$

the Riemann problem is an initial value problem described by the initial conditions  $\mathbf{Q}(x < 0, 0) = \mathbf{Q}_l$  and  $\mathbf{Q}(x > 0, 0) = \mathbf{Q}_r$ , where  $\mathbf{Q}_{l,r}$  are constant vectors. If a hyperbolic equation system is linear then the Riemann problem has exact solutions. If it is a nonlinear equation set, then valid solutions around  $x = 0$  (i.e. at the interface) can be obtained for short time intervals by introducing a linearization. As mentioned previously, the linear hyperbolic equation system described by Eq. (2.6) can be written in the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{Q}}{\partial x} = 0, \quad (2.7)$$

where the flux Jacobian,  $\mathbf{A}$ , is constant for the linear system described here.  $r^p$ ,  $l^p$  and  $s^p$  represent the right eigenvectors, the left eigenvectors and the eigenvalues of  $\mathbf{A}$ . This is a hyperbolic system, so it is known that all the eigenvalues of the flux Jacobian are real and the eigenvectors are assumed to be linearly independent, and in this case, orthonormal. In order to obtain the wave equations for the wave propagation method, Eq. (2.6) can be multiplied with the left eigenvectors,  $l^p$ , to obtain

$$\frac{\partial w^p}{\partial t} + s^p \frac{\partial w^p}{\partial x} = 0, \quad (2.8)$$

where  $w^p \equiv l^p \cdot \mathbf{Q}$ . To solve the Riemann problem for such linear systems exactly,  $w^p(x, t)$  needs to be determined. This is done by obtaining the initial condition for  $w_0^p(x, t)$ , where  $w_0(x) = l^p \cdot \mathbf{Q}(x, 0)$ . Once this is done,  $\mathbf{Q}(x, t) = \sum_p w^p r^p$  is solved to obtain the solutions.

The wave propagation method involves solving the Riemann problem at each cell interface and this solution is used to arrive at the following approximation to the numerical fluxes by accounting for the right- and left-going fluctuations.

$$[\mathcal{F}]_{i-1/2} = \frac{1}{2}(\mathbf{F}_i + \mathbf{F}_{i-1}) - \frac{1}{2}(\mathcal{A}^+ \Delta Q_{i-1/2} - \mathcal{A}^- \Delta Q_{i-1/2}). \quad (2.9)$$

Taking this expression and plugging it in the update formula defined by Eq. (2.3), the following equation is obtained:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} [\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}]. \quad (2.10)$$

The  $\mathcal{A}^\pm \Delta Q_{i-1/2}$  terms here are called the *fluctuations* and they are described by

$$\mathcal{A}^- \Delta Q_{i-1/2} = \sum_{p: s_{i-1/2}^p < 0} \mathcal{Z}_{i-1/2}^p + \frac{1}{2} Z_{i-1/2} \quad (2.11)$$

$$\mathcal{A}^+ \Delta Q_{i-1/2} = \sum_{p: s_{i-1/2}^p > 0} \mathcal{Z}_{i-1/2}^p + \frac{1}{2} Z_{i-1/2} \quad (2.12)$$

where

$$\mathcal{Z}_{i-1/2}^p = l_{i-1/2}^p \cdot (\mathbf{F}_i - \mathbf{F}_{i-1}) r_{i-1/2}^p \quad (2.13)$$

and

$$Z_{i-1/2} = \sum_{p: s_{i-1/2}^p = 0} \mathcal{Z}_{i-1/2}^p. \quad (2.14)$$

The above expression is the *F-Wave* method as developed by LeVeque. If using the *Q-Wave* method, then  $\mathcal{Z}_{i-1/2}^p$  will have the jump in the conserved variable,  $\mathbf{Q}$ , instead of the jump in flux,  $\mathbf{F}$ . Treating the fluctuations in Eq. (2.10), the identity,

$$\mathcal{A}^- \Delta Q_{i-1/2} + \mathcal{A}^+ \Delta Q_{i-1/2} = \sum_p \mathcal{Z}_{i-1/2}^p = \mathbf{F}_i - \mathbf{F}_{i-1} \quad (2.15)$$

follows from the definition of  $\mathcal{Z}_{i-1/2}^p$  described by Eq. (2.13) (in this case for the *F-Waves*). If using *Q-Waves*, then the  $\mathbf{F}$  is replaced with  $\mathbf{Q}$  in the flux difference expression defined by Eq. (2.15). At each cell interface, the right and left eigenvectors,  $r_{i-1/2}^p$ ,  $l_{i-1/2}^p$  and the eigenvalues  $s_{i-1/2}^p$  are computed from the flux Jacobian. The eigensystem used for linear systems is constant throughout, however, for nonlinear systems special treatments, i.e. an appropriate averaging, must be performed (such as Roe averaging[11]). If Roe averages are not used, the scheme still provides the appropriate solution and continues to remain conservative when *F-Waves* are used as long as the method chosen to obtain the eigensystem is consistent throughout. To elaborate, using F-Waves, one can consistently choose arithmetic averages or just the left or right value for the conserved variables to arrive at the values for the fluctuations, and the result will continue to remain conservative as long as the same treatment is maintained throughout the domain and through all times. With Q-Waves however, an appropriate averaging scheme such as Roe averaging must be chosen for the solution to remain conservative. If Roe averages are used then a conservative solution can be obtained using either *F-Waves* or *Q-Waves*. The method described in this section can be applied to a variety of hyperbolic problems including those that have spatially dependent conserved variables and flux functions.

## 2.2 High Resolution Corrections

The algorithm detailed in the previous section is only first order accurate and to increase the order of accuracy from first to second, certain corrections need to be performed. This is done by taking a Taylor series expansion of the conserved variables and the second order terms from this expansion are retained. The new, second order accurate algorithm is now

described by

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} [\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}] - \frac{\Delta t}{\Delta x} ([\tilde{\mathcal{F}}]_{i+1/2} - [\tilde{\mathcal{F}}]_{i-1/2}), \quad (2.16)$$

where the new term introduced here,  $[\tilde{\mathcal{F}}]_{i-1/2}$ , is the *correction flux* given by

$$[\tilde{\mathcal{F}}]_{i-1/2} = \frac{1}{2} \sum_p \text{sign}(s_{i-1/2}^p) \left( 1 - \frac{\Delta t}{\Delta x} |s_{i-1/2}^p| \right) \mathcal{Z}_{i-1/2}^p. \quad (2.17)$$

This correction increases the order of the High Resolution Wave Propagation method and makes it equivalent to the standard Lax-Wendroff algorithm. As a result of this increased accuracy to second order in Eq. (2.16), the algorithm can experience severe oscillations near discontinuities similar to the case of the Lax-Wendroff. Limiters can be applied in regions with sharp discontinuities. This limits the formal order of accuracy of the algorithm to first in these regions by replacing  $\mathcal{Z}_{i-1/2}^p$  in Eq. (2.17) by a *limited wave*  $\tilde{\mathcal{Z}}_{i-1/2}^p = \mathcal{Z}_{i-1/2}^p \phi(\theta_{i-1/2}^p)$ . Here,  $\phi(\theta)$  is the limiter function that is chosen

$$\theta_{i-1/2}^p \equiv \frac{\mathcal{Z}_{I-1/2}^p \cdot \mathcal{Z}_{i-1/2}^p}{\mathcal{Z}_{i-1/2}^p \cdot \mathcal{Z}_{i-1/2}^p} \quad (2.18)$$

with  $I = i - 1$  if  $s_{i-1/2}^p > 0$  and  $I = i + 1$  if  $s_{i-1/2}^p < 0$ . Some of the limiters that have been used with the wave propagation Algorithm include the Min-mod limiter,

$$\phi(\theta) = \text{minmod}(1, \theta), \quad (2.19)$$

the Superbee limiter,

$$\phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta)), \quad (2.20)$$

the Van Leer limiter,

$$\phi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|}, \quad (2.21)$$

and the Monotonized Centered limiter,

$$\phi(\theta) = \max(0, \min((1 + \theta)/2, 2, 2\theta)). \quad (2.22)$$

To extend these corrections to multi-dimensions, high resolution *transverse corrections* need to be performed in a similar manner to account for flow that is transverse to the coordinates. For the purposes of this thesis and the one-dimensional applications studied here, transverse corrections have been ignored.

### 2.3 Source Term Handling

The source terms for the wave propagation method are handled in several ways to note their effect on the solution and to make the algorithm more competitive. The balance law considered here in one dimension, as mentioned before, is

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x} = \mathbf{S}(\mathbf{Q}). \quad (2.23)$$

The first method of handling the source terms involves modifying Eq. (2.13) to use the method described in [1]

$$\mathcal{Z}_{i-1/2} = l_{i-1/2}^p \cdot ([\mathbf{F}_1]_i - [\mathbf{F}_1]_{i-1} - \Delta x [\mathbf{S}]_{i-1/2}) r_{i-1/2}^p, \quad (2.24)$$

where the value of the source term,  $[\mathbf{S}]_{i-1/2}$ , is calculated by taking an average between the cells so that an approximate value is obtained at each cell interface. In Eq. (2.24), it is noted that this method solves the Riemann problem by directly taking the source term values into account. This method works fine for multiple dimensions as well but splitting the source terms into separate components could become complicated for some balance laws. Also, as long as the source terms are damped, this scheme would work. When there are undamped oscillations in the source terms, this algorithm could become unstable because explicit schemes in general are unstable to equations with undamped oscillations. This is true for the source terms of the Two-Fluid Equations as will be discussed in Chapter 5.

The source terms can also be handled by performing a source term splitting method which involves solving the homogeneous equation separately from the ordinary differential equation (ODE). The ODE is the part that accounts for the source terms and it is described by:

$$\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{S}. \quad (2.25)$$

For purposes of achieving higher order accuracy, i.e. second order accuracy in this case, the ODE described by Eq. (2.25) is first advanced with a half time step of  $\Delta t/2$ . Then the homogeneous equation is solved over a full time step of  $\Delta t$  following which, the ODE is advanced again by another half time step of  $\Delta t/2$ . The results from each time advancement are used as initial conditions for the time steps that follow. To solve the ODE, any ODE scheme can be used and the one chosen here is that of the fourth order Runge-Kutta scheme. The source term splitting method detailed here is called *Strang splitting*. The trapezoidal method can also be used here to solve the ODE,

$$\mathbf{Q}(t + \Delta t) = \mathbf{Q}(t) + \frac{\Delta t}{2} (\mathbf{S}(\mathbf{Q}(t)) + \mathbf{S}(\mathbf{Q}(t + \Delta t))). \quad (2.26)$$

Taking a Taylor series expansion for  $\mathbf{S}(\mathbf{Q}(t + \Delta t))$ , a semi-implicit update formula for  $\mathbf{Q}$  is acquired,

$$\mathbf{Q}(t + \Delta t) = \mathbf{Q}(t) + \Delta t \left( \mathbf{I} - \frac{\Delta t}{2} \frac{\partial \mathbf{S}}{\partial \mathbf{Q}} \right)^{-1} \mathbf{S}(\mathbf{Q}(t)), \quad (2.27)$$

with  $\partial \mathbf{S}/\partial \mathbf{Q}$  being the source Jacobian and  $\mathbf{I}$  being the identity matrix. Equally good solutions can be obtained from both the ODE schemes. The advantage to using the Runge-Kutta method is that the source Jacobian does not need to be computed each time the source terms are modified or additional terms are added.

Yet another way to handle the source terms involves treating the problem implicitly by rewriting Eq. (2.23) as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x} = \mathbf{M}\mathbf{Q} \quad (2.28)$$

where  $\mathbf{M}$  is the source Jacobian. The update formula for this implicit case can be written as

$$\mathbf{Q}(t + \Delta t) = \mathbf{Q}(t) + \Delta t \mathcal{L}(\mathbf{Q}(t)) + \Delta t \mathbf{M}\mathbf{Q}(t + \Delta t) \quad (2.29)$$

where  $\mathcal{L}$  represents the flux update part of the equation. The update formula reduces to

$$\mathbf{Q}(t + \Delta t) = (\mathbf{I} - \Delta t \mathbf{M})^{-1} [\mathbf{Q}(t) + \Delta t \mathcal{L}(\mathbf{Q}(t))] \quad (2.30)$$

where the term within the square brackets is computed while calculating the flux update. The solution from the flux update needs to be multiplied with the inverted matrix,

$(\mathbf{I} - \Delta t M)^{-1}$ , at every cell to obtain the updated formula with the source terms at every time step.

This implicit source term method is explored for the electron acoustic problem detailed in Chapter 6 to attempt to handle the phase errors that are obtained from the source term splitting method when specific input parameters are used.

For linear ODEs, taking the source Jacobian can determine the solution type. If the eigenvalues of the Jacobian are real, then the solution grows or decays for positive and negative eigenvalues respectively. For imaginary eigenvalues, however, the solution is oscillatory and the frequency of these harmonics is governed by the magnitude of the eigenvalues. The real component of the eigenvalues can lead to growing or decaying oscillations.

These imaginary solutions allow for dispersive effects to be captured. Refining the grid, however, allows for more and more dispersive effects because waves of smaller and smaller wavelengths are now captured. Such effects of grid refinement make it hard to do a grid convergence study on dynamic problems when imaginary eigenvalues are present in the source Jacobian (as with the Two-Fluid Plasma Model).

If there is a hyperbolic, homogeneous part to the equation system, then disturbances propagate as waves with finite speeds in the medium. Capturing these wave-propagations can play a significant role in understanding the physics described by the equation set. The source terms need to be treated appropriately because they could significantly affect the solution.

## Chapter 3

**DISCONTINUOUS GALERKIN METHOD**

The discontinuous Galerkin method is a generalization of the high resolution wave propagation method presented in Chapter 2. The wave propagation method is of second order in space and time after applying the flux corrections. In some cases it is useful to have a higher order algorithm that will resolve the solutions and instabilities to higher orders of accuracy both spatially and temporally. The discontinuous Galerkin (DG) method does precisely that; the spatial accuracy depends on the choice of the basis functions and the temporal accuracy depends on the choice of the time integration scheme. The following sections describe this algorithm in one-dimension and it can be applied to account for multiple dimensions. This chapter follows the algorithm as detailed by Loverich, Hakim, Shumlak [10].

**3.1 The Base Scheme**

A one-dimensional balance law, as noted before, is written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S}, \quad (3.1)$$

with  $\mathbf{F}$  representing the flux in the  $x$  direction and  $\mathbf{S}$  representing the source term. The domain is defined within the boundaries,  $[x_a, x_b]$ . The cells are introduced as  $I_i = [x_{i-1/2}, x_{i+1/2}]$ , for each cell interval, where  $x_{i-1/2}$  and  $x_{i+1/2}$  are the coordinates at the left and right edges of each cell. The cell center is defined as  $(x_i)$ , where  $x_i \equiv (x_{i-1/2} + x_{i+1/2})/2$ . The wave propagation method uses cell averages within the cell to account for the conserved variable and solve the Riemann problem. The solution in each cell was piecewise-constant. The DG scheme uses basis functions to represent the conserved quantities, so it assumes that the solution in each cell is a piecewise-polynomial.

The basis-functions,  $v_r(x)$ , for  $r = 0, 1, \dots$  are locally defined within each cell, i.e.  $x \in \Omega_i = [x_{i-1/2}, x_{i+1/2}]$ . Any function can be represented with the polynomial basis

functions, and applying this idea to the conserved quantities within each cell, it is seen that

$$Q(x) = \sum_{r=0}^{\infty} Q_r v_r(x), \quad (3.2)$$

where  $Q_r$  are the expansion coefficients. The number of terms that are retained in this expansion is a representation of the spatial order of the solution. Usually only the first two or three terms are retained to make sure that the spatial order does not exceed the temporal order. The discontinuous Galerkin method requires temporal order  $\geq$  spatial order for stability. If this is violated, then the solution obtained can be subject to errors. Orthogonal basis functions are selected,

$$\int_{I_i} v_r(x) v_m(x) dx = \Delta x C_r \delta_{rm}, \quad (3.3)$$

where  $\Delta x \equiv (x_{i+1/2} - x_{i-1/2})$  is the grid spacing,  $C_r$  are normalization constants and  $\delta_{rm}$  is the Kronecker-delta symbol. Using this orthogonality equation described by Eq. (3.3), and multiplying the orthogonal basis functions,  $v_m(x)$ , with Eq. (3.2) gives,

$$Q_r = \frac{1}{C_r \Delta x} \int_{I_i} Q(x) v_r(x) dx, \quad (3.4)$$

which are the expansion coefficients that are used to project any function onto the basis function. Now applying this to the balance law described by Eq. (3.1), the equation is multiplied by  $v_r(x)$  and integrated over the cell to get

$$\frac{\partial}{\partial t} \int_{I_i} v_r(x) \mathbf{Q} dx + \int_{I_i} v_r(x) \frac{\partial \mathbf{F}}{\partial x} dx = \int_{I_i} v_r(x) \mathbf{S}. \quad (3.5)$$

Use integration by parts on the second term to get:

$$\int_{I_i} v_r(x) \frac{\partial \mathbf{F}}{\partial x} dx = v_r(x) \mathbf{F}(\mathbf{Q}(x, t)) \Big|_{x_{i-1/2}}^{x_{i+1/2}} - \int_{I_i} \frac{dv_r}{dx} \mathbf{F} dx \quad (3.6)$$

The conserved variables are then expanded into basis functions using

$$\mathbf{Q}(x, t) = \sum_{r=0}^{\infty} \mathbf{Q}_r(t) v_r(x) \quad (3.7)$$

to get

$$C_r \frac{d\mathbf{Q}_r}{dt} + \frac{\mathbf{F}_{i+1/2} v_r(x_{i+1/2}) - \mathbf{F}_{i-1/2} v_r(x_{i-1/2})}{\Delta x} - \frac{1}{\Delta x} \int_{I_i} \frac{dv_r(x)}{dx} \mathbf{F} dx = \frac{1}{\Delta x} \int_{I_i} v_r(x) \mathbf{S}. \quad (3.8)$$

for  $r = 0, 1, \dots$ , where the interface fluxes at  $x_{i\pm 1/2}$  are given by

$$\mathbf{F}_{i\pm 1/2} \equiv \mathbf{F}(\mathbf{Q}(x_{i\pm 1/2}, t)). \quad (3.9)$$

The discontinuous Galerkin method is described by Eq. (3.8) for the given balance law. Rearranging the time-dependent and spatially-dependent variables, Eq. (3.8) is written as

$$\frac{d\mathbf{Q}_r}{dt} = \mathcal{L}_r(\mathbf{Q}) \quad (3.10)$$

for  $r = 0, 1, \dots$ , where  $\mathcal{L}_r(\mathbf{Q})$  is the operator containing all the operations for the spatially dependent part of the equations (flux and source computations),

$$\mathcal{L}_r(\mathbf{Q}) = -\frac{\mathbf{F}_{i+1/2}v_r(x_{i+1/2}) - \mathbf{F}_{i-1/2}v_r(x_{i-1/2})}{\Delta x} + \frac{1}{\Delta x} \int_{I_i} \frac{dv_r(x)}{dx} \mathbf{F} dx + \frac{1}{\Delta x} \int_{I_i} v_r(x) \mathbf{S}. \quad (3.11)$$

After computing  $\mathcal{L}_r(\mathbf{Q})$ , the ODE described by Eq. (3.10) needs to be solved for the expansion coefficients,  $Q_r$  using any standard ODE solver. The Runge-Kutta time integration scheme is used here (second, third and fourth orders) which makes this DG method specifically a *Runge-Kutta Discontinuous Galerkin* (RKDG) scheme.

The RKDG method also requires the interface fluxes to solve the Riemann problem at each interface. Computing these interface fluxes is identical to what was done for the wave propagation method detailed in Eq. (2.9), Eq. (2.11) and Eq. (2.12) in Chapter 2.

### 3.2 Selection of Basis Functions

The Legendre polynomials are commonly selected as basis functions for this algorithm because they are orthogonal. The basis functions would be

$$v_r(x) = P_r(\eta(x)), \quad (3.12)$$

where the interval,  $I_i$ , needs to be mapped to lie within the interval,  $[-1, 1]$ , because the Legendre polynomials are only defined within this interval. This mapping is done using

$$\eta(x) \equiv \frac{x - x_i}{\Delta x/2} \quad (3.13)$$

for  $x_i$  and  $\Delta x$  as described in the previous section. The Legendre polynomials have a convenient property that makes them useful for the RKDG algorithm, they are orthogonal. They satisfy the property,

$$\int_{-1}^1 P_n(x)P_m(x)dx = \frac{2}{2m+1}\delta_{mn}, \quad (3.14)$$

where the coefficients  $C_r$  are found to be  $C_r = 1/(2r+1)$ . Following the mapping, the value of the basis functions can be obtained at the cell interfaces,

$$v_r(x_{i\pm 1/2}) = P_r(\pm 1) = (\pm 1)^r. \quad (3.15)$$

To handle the second and third terms on the right hand side of Eq. (3.11), Gaussian quadrature is used so that these integrals can be numerically evaluated. Applying this to a function,  $Q(x)$ , over interval  $I_i$  gives

$$\int_{I_i} Q(x)dx = \frac{1}{2} \int_{-1}^1 Q(x(\eta))d\eta = \frac{1}{2} \sum_j w_j \bar{Q}(\eta_j), \quad (3.16)$$

where  $x(\eta) = \eta\Delta x/2 + x_i$ ,  $\bar{Q}(\eta) \equiv Q(x(\eta))$  and  $w_j$  and  $\eta_j$  are weights and abscissa of the chosen Gaussian quadrature scheme. The choice of the Gaussian quadrature depends on the order of the basis polynomial used, i.e., it depends on the highest value of  $r$  selected in Eq. (3.2).

Using the Legendre polynomial basis functions in Eq. (3.11)  $\mathcal{L}_r$  is now written as

$$\mathcal{L}_r(\mathbf{Q}) = -\frac{\mathbf{F}_{i+1/2} - (-1)^r \mathbf{F}_{i-1/2}}{\Delta x} + \frac{1}{\Delta x} \int_{-1}^1 \frac{dP_r(\eta)}{d\eta} \bar{\mathbf{F}} d\eta + \frac{1}{2} \int_{-1}^1 P_r \bar{\mathbf{S}} d\eta, \quad (3.17)$$

where  $\bar{\mathbf{F}}(\eta) \equiv \mathbf{F}(Q(x(\eta), t))$  and  $\bar{\mathbf{S}}(\eta) \equiv \mathbf{S}(Q(x(\eta), t))$ . The interface fluxes are computed using the same method as the wave propagation method and the integrals in Eq. (3.17) are computed using Gaussian quadrature. This is the DG scheme.

### 3.3 Limiters for the Discontinuous Galerkin Scheme

The DG scheme can produce large oscillations in the solution with the presence of sharp gradients. Like with the case of the Wave Propagation algorithm, limiters can be applied to the DG method as well around regions of discontinuities. For the wave propagation method, the limiters are applied to the waves, for the DG method however, they are applied

to either the conserved variables or the characteristic variables depending on the limiter used. There are two types of limiters that are investigated with this algorithm. The first is the characteristics-based limiter according to which, the conserved variables are transformed to characteristic variables. For  $\mathbf{Q}_r^i$  defined as the expansion coefficients of the conserved variable in cell  $i$ ,  $a^p \equiv l^p \mathbf{Q}_1^i$ ,  $a_+^p \equiv l^p(\mathbf{Q}_0^{i+1} - \mathbf{Q}_0^i)$  and  $a_-^p \equiv l^p(\mathbf{Q}_0^i - \mathbf{Q}_0^{i-1})$ . The coefficient of the linear term is modified using the minmod limiter as

$$\mathbf{Q}_1^i = \sum_p r^p \text{mm}(a^p, a_+^p, a_-^p) \quad (3.18)$$

where  $r^p$  and  $l^p$  are the right and left eigenvectors of the flux Jacobian computed from cell averages.  $\text{mm}(a, b, c)$  is a modified min-mod function defined as

$$\text{mm}(a, b, c) = a \quad \text{if } |a| < M\Delta x^2 \quad (3.19)$$

$$= \text{m}(a, b, c) \quad \text{otherwise,} \quad (3.20)$$

where  $M$  is a constant and the function,  $\text{m}(a, b, c)$  is defined by

$$\text{m}(a, b, c) = \max(a, b, c) \quad \text{if } \text{sgn}(a) = \text{sgn}(b) = \text{sgn}(c) = + \quad (3.21)$$

$$= \min(a, b, c) \quad \text{if } \text{sgn}(a) = \text{sgn}(b) = \text{sgn}(c) = - \quad (3.22)$$

$$= 0 \quad \text{otherwise.} \quad (3.23)$$

Once the limiting is done, the solution of the characteristic variables is transformed back to that of the conserved variables. Just as with the wave propagation algorithm, the limiter limits the order of the algorithm to a lower one in regions where it is applied. If limiters are applied to the first order coefficients (i.e. for  $r = 1$ ), then all the higher-order coefficients are set to zero and the algorithm at those local points reduces to second order.

Component-based limiters can also be applied to the DG algorithm. These are faster than characteristic-based limiters because they do not require computing the Jacobian and performing an eigen-decomposition. They involve directly applying the limiters to the conserved variables without transforming them. However, they are not TVDM so oscillations can develop.

### 3.4 Time Stepping Scheme

Once the right hand side of Eq. (3.10) is solved using the methods detailed above, the solution can be advanced in time using the Runge-Kutta time stepping scheme. Usually, the temporal order of the Runge-Kutta time integration scheme chosen is at least that of the spatial order set by the basis functions and quite often it is even one higher than the spatial order. Generally, for a spatial order of  $k$ , the temporal order chosen would need to at least be  $k$ . The 3<sup>rd</sup> order Runge-Kutta time stepping scheme is

$$\mathbf{Q}^1 = \mathbf{Q}^n + \Delta t \mathcal{L}(\mathbf{Q}^n) \quad (3.24)$$

$$\mathbf{Q}^2 = \frac{3}{4} \mathbf{Q}^n + \frac{1}{4} (\mathbf{Q}^1 + \Delta t \mathcal{L}(\mathbf{Q}^1)) \quad (3.25)$$

$$\mathbf{Q}^{n+1} = \frac{1}{3} \mathbf{Q}^n + \frac{2}{3} (\mathbf{Q}^2 + \Delta t \mathcal{L}(\mathbf{Q}^2)) \quad (3.26)$$

Here,  $\mathbf{Q}^n$  represents the solution at time  $t$  and  $\mathbf{Q}^{n+1}$  represents the solution at time  $t + \Delta t$ . For the RKDG method, stability is ensured by using a CFL number of  $1/(2k - 1)$  where  $k$  is the spatial order of the scheme.

## Chapter 4

**EXPLORED HYPERBOLIC EQUATION SYSTEMS**

In order to understand the difference between the algorithms as thoroughly and completely as possible, it is useful to explore some simpler balance laws. The effects seen with the simpler equation systems can provide insight into the more complex models such as the full two-fluid model studied in Chapter 5. In one dimension, for example, the balance law is written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x} = \mathbf{S}(\mathbf{Q}). \quad (4.1)$$

The following sections describe two balance laws, the stress-strain relationship for homogeneous and layered media, and the “dispersive Euler” equation system.

**4.1 Stress-Strain Relations**

The equation system described in this section involves two simple hyperbolic equations to study the longitudinal elastic strain waves that propagate through media. The equation system is described by:

$$\epsilon_t(x, t) - u_x(x, t) = 0 \quad (4.2)$$

$$(\rho(x)u(x, t))_t - \sigma_x(\epsilon(x, t), x) = 0 \quad (4.3)$$

where  $\epsilon(x, t)$  is the strain,  $u(x, t)$  is the velocity,  $\rho(x)$  is the density of the medium, and  $\sigma(\epsilon(x, t), x)$  is the stress. When the strains are small enough, the stress-strain relations can be treated as linear equations.

$$\sigma_A(\epsilon) = K_A\epsilon, \quad \sigma_B(\epsilon) = K_B\epsilon, \quad (4.4)$$

where  $K_A$  and  $K_B$  are the bulk moduli of the two materials that form a layered medium[9]. For the homogeneous medium, the density and bulk modulus are initialized as constant

values throughout the domain. For the layered medium however, the density and stress are described by:

$$(\rho(x), \sigma(\epsilon, x)) = \begin{cases} (\rho_A, \sigma_A(\epsilon)) & \text{if } j\delta < x < (j + \alpha)\delta \text{ for some integer } j, \\ (\rho_B, \sigma_B(\epsilon)) & \text{otherwise.} \end{cases} \quad (4.5)$$

Since the relations described by Eq.(4.4) are treated for the linear case only, the bulk moduli and densities are constant values for each medium.

A solitary wave is initialized and its propagation is studied in both the homogeneous and layered media. The details of the problem including the initial and boundary conditions are described in Chapter 6.

## 4.2 Dispersive Euler Equations

Yet another model for benchmarking purposes is that of the Euler equations. The *Dispersive Euler* equations[4], are an extension of the Euler equations with the use of source terms. The reason for doing this is to create a nonlinear balance law that allows for dispersive effects in the solution. The equation system is described by,

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e + p)u \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda\rho(vB_z - wB_y) \\ \lambda\rho(wB_x - uB_z) \\ \lambda\rho(uB_y - vB_x) \\ 0 \end{pmatrix} \quad (4.6)$$

where the energy is,

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2 + w^2), \quad (4.7)$$

$\mathbf{B} = \mathbf{B}(x)$  is a spatially dependent vector that is independent of the conserved variables, and  $\lambda$  is a constant. The source Jacobian of this system is

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda B_z & -\lambda B_y & 0 \\ 0 & -\lambda B_z & 0 & \lambda B_x & 0 \\ 0 & \lambda B_y & -\lambda B_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.8)$$

and its eigenvalues are:  $0, 0, 0, \pm i\lambda B$ , where  $B$  is the magnitude of  $\mathbf{B}$ . The presence of the imaginary eigenvalue shows that this is not a growing or decaying system, instead it is an oscillatory system that would maintain the dispersive effects without decaying the oscillations that form. It is difficult to capture such dispersions accurately as refining the grid excites waves of smaller and smaller wavelengths. This model is a much simplified approximation of the two-fluid plasma model and applying it to certain problems shows properties that are very similar to that of the two-fluid model. This equation system is applied to the *electron acoustic wave* which is discussed in greater detail in Chapter 6 where the results are examined as well.

This section introduces two balance laws, namely, the stress-strain relations for elasticity and the Dispersive Euler equation system. As will be seen in the following chapter, the equations of the plasma fluid models also have hyperbolic, homogeneous parts and can be written in balance law form. Chapter 5 that follows extends the balance laws to a more complicated model that forms the basis of this research.

## Chapter 5

**FULL TWO-FLUID PLASMA MODEL**

The full two-fluid plasma model that will be discussed in this section comes from taking moments of the Boltzmann equation and treating the electrons and ions as two separate fluids. These equations that result have homogeneous, hyperbolic parts and hence, can be written in balance law form. In one-dimension, the equation is written as,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x} = \mathbf{S}(\mathbf{Q}). \quad (5.1)$$

**5.1 Basic Equations**

The equations described here are the five-moment equations that result from taking the 0<sup>th</sup> and 1<sup>st</sup> moments of the Boltzmann equation in addition to using the moment that describes the energy. These equations are then closed with an equation of state. There are five fluid equations for electrons, five for ions and six Maxwell's equations that result in the sixteen equations of the two-fluid model. Hence, the three systems of balance laws in the form defined by Eq. (5.1) are:

$$\frac{\partial \mathbf{Q}_e}{\partial t} + \frac{\partial \mathbf{F}_e(\mathbf{Q}_e)}{\partial x} = \mathbf{S}_e(\mathbf{Q}_e, \mathbf{Q}_{em}), \quad (5.2)$$

for the electron fluid equations,

$$\frac{\partial \mathbf{Q}_i}{\partial t} + \frac{\partial \mathbf{F}_i(\mathbf{Q}_i)}{\partial x} = \mathbf{S}_i(\mathbf{Q}_i, \mathbf{Q}_{em}), \quad (5.3)$$

for the ion fluid equations, and

$$\frac{\partial \mathbf{Q}_{em}}{\partial t} + \frac{\partial \mathbf{F}_{em}(\mathbf{Q}_{em})}{\partial x} = \mathbf{S}_{em}(\mathbf{Q}_e, \mathbf{Q}_i), \quad (5.4)$$

for Maxwell's equations, where subscripts *em* represent electromagnetic terms.

The one-dimensional Euler Equation balance laws with additional source terms which form the basis of the electron and ion fluid equations are defined by:

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho_s \\ \rho_s u_s \\ \rho_s v_s \\ \rho_s w_s \\ e_s \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho_s u_s \\ \rho_s u_s^2 + p_s \\ \rho_s u_s v_s \\ \rho_s u_s w_s \\ (e_s + p_s)u_s \end{pmatrix} = \begin{pmatrix} 0 \\ q_s n_s (E_x + v_s B_z - w_s B_y) \\ q_s n_s (E_y + w_s B_x - u_s B_z) \\ q_s n_s (E_z + u_s B_y - v_s B_x) \\ q_s n_s (E_x u_s + E_y v_s + E_z w_s) \end{pmatrix} \quad (5.5)$$

where subscript  $s$  represents the species (electrons or ions) and the energy,  $e$  is given by

$$e_s = \frac{p_s}{\gamma - 1} + \frac{1}{2} \rho_s (u_s^2 + v_s^2 + w_s^2). \quad (5.6)$$

It can be seen that the source terms of Eq. (5.5) contain the Lorentz forces on the electrons and ions. These source terms relate the fluid equations to the electromagnetic terms of Maxwell's equations. The Lorentz forces act as body forces on the electrons and ions and they have evolving electromagnetic source terms in them that can make the equation set and the solutions rather complicated with a lot of dynamics. Maxwell's equations in matrix form in one-dimension are given by

$$\frac{\partial}{\partial t} \begin{pmatrix} E_x \\ E_y \\ E_z \\ B_x \\ B_y \\ B_z \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} 0 \\ c^2 B_z \\ -c^2 B_y \\ 0 \\ -E_z \\ E_y \end{pmatrix} = \begin{pmatrix} -\frac{1}{\epsilon_0} (q_e n_e u_e + q_i n_i u_i) \\ -\frac{1}{\epsilon_0} (q_e n_e v_e + q_i n_i v_i) \\ -\frac{1}{\epsilon_0} (q_e n_e w_e + q_i n_i w_i) \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (5.7)$$

This is how the two-fluid plasma model comes about with the sixteen balance laws. The full two-fluid equations are applied to the one-dimensional Z-pinch equilibrium. Applications and results of the two-fluid plasma model are detailed in Chapter 6.

## 5.2 The Two-Fluid Source Terms

When the source terms are collected from Eq. (5.5) and Eq. (5.7), the following ODE is obtained by assuming that the advection terms disappear (spatial derivatives are assumed

to vanish):

$$\frac{d\mathbf{Q}}{dt} = \mathbf{S}. \quad (5.8)$$

$\mathbf{Q}$  and  $\mathbf{S}$  can be defined as shown below. It can be noted that  $\mathbf{Q}$  only has 9 variables as compared to the initial 16 variables. This is because the variables with source terms equal to 0, i.e. the density and the magnetic fields, have been neglected here and the energy equation has been neglected in this analysis as well. The reason for ignoring the energy equation is that it does not explicitly appear in any of the source terms. The  $\mathbf{Q}$  and  $\mathbf{S}$  for purposes of solving the ODE are given by:

$$\mathbf{Q} = \begin{bmatrix} \rho_e u_e \\ \rho_e v_e \\ \rho_e w_e \\ \rho_i u_i \\ \rho_i v_i \\ \rho_i w_i \\ E_x \\ E_y \\ E_z \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} q_e n_e (E_x + v_e B_z - w_e B_y) \\ q_e n_e (E_y + w_e B_x - u_e B_z) \\ q_e n_e (E_z + u_e B_y - v_e B_x) \\ q_i n_i (E_x + v_i B_z - w_i B_y) \\ q_i n_i (E_y + w_i B_x - u_i B_z) \\ q_i n_i (E_z + u_i B_y - v_i B_x) \\ -\frac{1}{\epsilon_0} (q_e n_e u_e + q_i n_i u_i) \\ -\frac{1}{\epsilon_0} (q_e n_e v_e + q_i n_i v_i) \\ -\frac{1}{\epsilon_0} (q_e n_e w_e + q_i n_i w_i) \end{bmatrix}. \quad (5.9)$$

Eq. (5.8) indicates that  $\mathbf{S}$  is linear in  $\mathbf{Q}$  and studying the source Jacobian  $\mathbf{M}$  can determine the solution type of the ODE in Eq. (5.8). Examining the eigenvalues of this source Jacobian, it is noted that the first three eigenvalues are  $0, \pm i\omega_p$  where  $\omega_p^2 = \omega_{pe}^2 + \omega_{pi}^2$ . The remaining six eigenvalues are all imaginary roots of a 6<sup>th</sup> order polynomial. The non-zero eigenvalues are all imaginary. The plasma frequency,  $\omega_{ps}$  is defined as

$$\omega_{ps} = \sqrt{\frac{n_s q_s^2}{\epsilon_0 m_s}}, \quad (5.10)$$

where subscript  $s$  represents each species (electrons and ions).

Therefore, it is evident that the source terms of the Two-Fluid model are not damped. Since the source Jacobian has imaginary eigenvalues only, this plasma model is not diffusive but instead is dispersive with undamped oscillations. The use of explicit schemes will be

unstable generally with the presence of these oscillations. Also, for the dynamics to be properly resolved, the frequency of the waves in the system should be significantly larger than the frequency of these oscillations in order to resolve the oscillations well. This ensures adequate time steps per oscillation. In the *F-Wave* approach detailed by LeVeque, the scheme will be extremely unstable if the source terms are directly incorporated in the Riemann solver as described in the *Source term handling* section in Chapter 2 for the wave propagation method. For this algorithm, source term splitting is the best approach to handle this. The effects of handling source terms in several ways are seen in Chapter 6, with the *electron acoustic pulse* application.

### **5.3 Radial Source Terms for 1D Cylindrical Coordinates**

For purposes of modeling the one-dimensional radial Z-pinch described in Chapter 6, it is important to understand how to treat the radial terms that arise from the Two-Fluid balance laws. In this thesis, these additional radial terms that arise from the equations are treated as source terms and they are incorporated into the source matrix described above in Eq. (5.9). However, due to the presence of radial source terms in the density and energy as well, the treatment of  $\mathbf{S}$  will involve using the full  $\mathbf{Q}$  vector of conserved variables. Using the appropriate divergence and curl terms in cylindrical coordinates gives the modified  $\mathbf{S}$

for one-dimensional radial problems:

$$\mathbf{Q} = \begin{bmatrix} \rho_e \\ \rho_e u_e \\ \rho_e v_e \\ \rho_e w_e \\ e_e \\ \rho_i \\ \rho_i u_i \\ \rho_i v_i \\ \rho_i w_i \\ e_i \\ E_r \\ E_\phi \\ E_z \\ B_r \\ B_\phi \\ B_z \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} -\frac{\rho_e u_e}{r} \\ -\frac{\rho_e u_e^2}{r} - \frac{p_e}{r} + \frac{\rho_e v_e^2}{r} + \frac{p_e}{r} + q_e n_e (E_r + v_e B_z - w_e B_\phi) \\ -2\frac{\rho_e u_e v_e}{r} + q_e n_e (E_\phi + w_e B_r - u_e B_z) \\ -\frac{\rho_e u_e w_e}{r} + q_e n_e (E_z + u_e B_\phi - v_e B_r) \\ -\frac{u_e(e_e + p_e)}{r} + q_e n_e (E_r u_e + E_\phi v_e + E_z w_e) \\ -\frac{\rho_i u_i}{r} \\ -\frac{\rho_i u_i^2}{r} - \frac{p_i}{r} + \frac{\rho_i v_i^2}{r} + \frac{p_i}{r} + q_i n_i (E_r + v_i B_z - w_i B_\phi) \\ -2\frac{\rho_i u_i v_i}{r} + q_i n_i (E_\phi + w_i B_r - u_i B_z) \\ -\frac{\rho_i u_i w_i}{r} + q_i n_i (E_z + u_i B_\phi - v_i B_r) \\ -\frac{u_i(e_i + p_i)}{r} + q_i n_i (E_r u_i + E_\phi v_i + E_z w_i) \\ -\frac{1}{\epsilon_0} (q_e n_e u_e + q_i n_i u_i) \\ -\frac{1}{\epsilon_0} (q_e n_e v_e + q_i n_i v_i) \\ \frac{c^2 B_\phi}{r} - \frac{1}{\epsilon_0} (q_e n_e w_e + q_i n_i w_i) \\ 0 \\ 0 \\ -\frac{E_\phi}{r} \end{bmatrix}. \quad (5.11)$$

For the radial source terms,  $u$  represents the radial component of the velocity,  $v$  represents the azimuthal and  $w$  represents the  $z$  component. The first five terms in  $\mathbf{Q}$  and  $\mathbf{S}$  represent electron variables, the next five represent ion variables and the last six represent electromagnetic terms. Subscripts  $e$  and  $i$  stand for electron and ion variables respectively. In this manner the radial source terms can be taken into account and applied to the radial Z-pinch problem. It is assumed that the pressure is isotropic, so  $p_{rr} = p_{\phi\phi} = p$  is assumed for both electron and ion fluids.

## Chapter 6

**APPLICATIONS AND RESULTS**

The three applications that have been studied in this thesis for purposes of comparing the high resolution wave propagation algorithm to the discontinuous Galerkin algorithm include: the stress-strain relations, the Dispersive Euler equation set, and lastly the full two-fluid plasma equations. The goal is to compare the algorithms, determine the advantages and disadvantages of each algorithm and note if there are any problem dependent issues relating to each algorithm. This will shed some light on which algorithm is most suited for applications of the two-fluid plasma equations. It is essential to analyze the algorithms with simpler equation systems to gain a more complete understanding of their applicability before applying them to complicated equation sets.

**6.1 Strain Waves**

The stress-strain equations were implemented with linear and nonlinear dependence of the stress on the strain. Only the linear equations are detailed here. The equation sets that are used to simulate the strain waves are described in Chapter 4 in Eq. (4.3). There are no source terms in this system.

$$\mathbf{Q} = \begin{bmatrix} \epsilon \\ \rho u \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -u \\ -K\epsilon \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (6.1)$$

where  $\epsilon$  is the strain,  $u$  is the velocity,  $\rho$  is the density, and  $K$  is the bulk modulus. A description of the problem and the results comparing the algorithms to each other and to LeVeque's solution in [9] are detailed in this section.

*6.1.1 Description of Problem*

The problem involves initializing a homogeneous medium and a layered medium with a strain wave. The homogeneous medium has constant density and constant bulk modulus

throughout the domain. The layered medium has two materials that are layered such that they alternate with each other at every user-specified interval in the x-direction. The layering is implemented as follows:

$$(\rho(x), \sigma(\epsilon, x)) = \left\{ \begin{array}{ll} (\rho_A, \sigma_A(\epsilon)) & \text{if } j\delta < x < (j + \alpha)\delta \text{ for some integer } j, \\ (\rho_B, \sigma_B(\epsilon)) & \text{otherwise.} \end{array} \right\}, \quad (6.2)$$

with  $\sigma$  representing the stress. For the layered medium, the simulations are performed with matched and mismatched impedances. Impedance is defined as  $Z = \sqrt{K\rho}$ , with  $K$  representing the bulk modulus and  $\rho$  denoting the density of the medium. For matched impedances, i.e.  $Z_A = Z_B$ , the results obtained differ from the mismatched impedances ( $Z_A \neq Z_B$ ) and the ability of both the algorithms to capture these differences and provide adequate resolution and accuracy is studied through a grid convergence study.

Both the conserved quantities,  $\epsilon$  and  $\rho u$  are initialized to zero. The strain wave is initialized on the left boundary using

$$u(0, t) = \left\{ \begin{array}{ll} \bar{u}(1 + \cos(\pi(t - 10)/10)) & 0 \leq t \leq 20, \\ 0 & t > 20. \end{array} \right\}, \quad (6.3)$$

where  $\bar{u}$  is a constant chosen to be 0.2 here. This resembles pulling the left edge outward for time  $0 \leq t \leq 20$  and generating a strain wave that propagates to the right. The right boundary condition used is that of zero-order extrapolation. The results are displayed in the following section.

### 6.1.2 Results

The homogeneous solution obtained is smooth and free of discontinuities, so both algorithms capture the solution rather effectively.

A better comparison lies in capturing a complicated solution, with a number of discontinuities that evolve as the solution progresses in time, and with disturbances propagating with different speeds. A layered medium for the strain waves equation system is better suited to compare the effectiveness of the algorithms in capturing complexities in the solution accurately. All the results shown in this section are for layered media with mismatched impedances allowing for waves to propagate and reflect at each layer interface.

The layering described in Eq. (6.2) is implemented to reproduce the solutions from LeVeque's paper [9]. It is noted that as the grid resolution is increased, the results obtained seem to be converging to a specific solution. Hence, a grid convergence study is performed. The simulation is run with 1000 grid points using both the wave propagation algorithm and the discontinuous Galerkin method, and these are compared to determine which one is suitable for use as the converged solution.

The wave propagation algorithm is run with CFL=0.5 to maintain stable results. For higher CFL numbers, the solution goes unstable instantly for this problem. This issue was studied by Fogarty and LeVeque in [3] where the application of a new method of limiting is introduced for such problems to obtain more stable results. For the Riemann problem to be valid within each cell, it is important to ensure that the positive- and negative-going waves within each cell do not interact with each other. If this interaction happens, then the solution can go unstable. The fastest wave speeds are  $+c$  and  $-c$  for this problem, where  $c$  is the eigenvalue of the equation system. It is possible that waves traveling with speed  $+c$  interact with waves traveling with speed  $-c$  within a cell. This is possibly the reason why, for larger CFL numbers, the solution develops significant oscillations for this problem because there are waves that are constantly propagating through and reflecting as they hit each layer. The layers change at every  $x=0.5$  in space, i.e. every  $1/100$  of the domain, so there is a high probability that positive- and negative-going waves are interacting within the cells. Hence, by reducing the CFL number, i.e. taking smaller time steps, the algorithm becomes more stable as the fastest waves do not travel far enough within each cell to interact.

The discontinuous Galerkin method is run with CFL=0.3 based on the stability requirement of this algorithm:  $\text{CFL} = 1/(2n - 1)$  where  $n$  is the spatial order. A characteristics-based limiter is used for the purposes of this comparison because despite taking a long time to run, the best solution is obtained with the use of this limiter. Figure 6.1 displays both the algorithms with 1000 grid points when run until time,  $t = 40$ . The plot is that of the strain Vs grid location. It is evident that the solutions obtained from both methods are rather similar. The solution is discontinuous at every layer in the region where the strain wave propagates. The solution from discontinuous Galerkin method is rather straight at

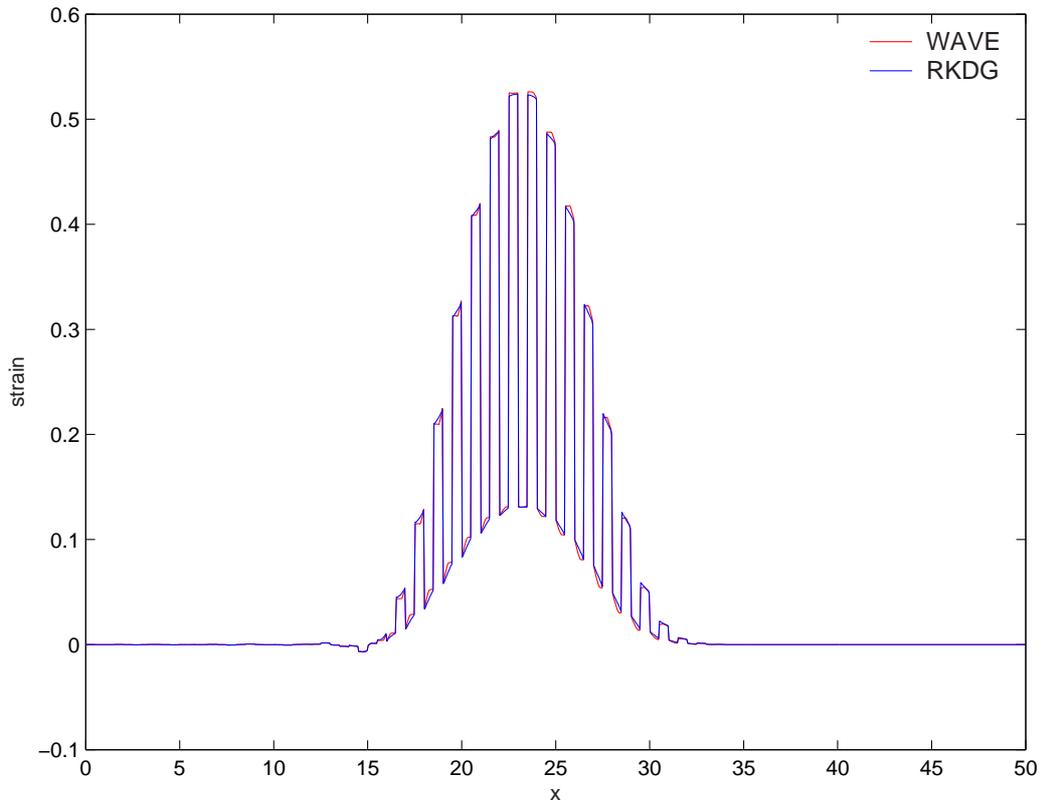


Figure 6.1: Strain Wave problem: Solutions computed for Strain Vs X to determine the result that is used for the grid convergence study as the converged solution. wave propagation algorithm with 2nd order corrections (red line) and 2nd order discontinuous Galerkin solution (blue line) are shown for 1000 grid points at time  $t=40$ . This figure is for a layered medium with mismatched impedances.

the top and bottom of each step (within the layers) and this makes it similar to LeVeque's solution. For the same resolution however, the wave propagation method has curved solutions within each layer between the jumps. The discontinuities themselves match closely when comparing the two algorithms.

Figure 6.2 compares the average values of the discontinuous Galerkin method (i.e. only the first coefficient of the conserved variable) with that of the solution constructed by taking the expansion coefficients of the basis functions into account as well. This is done by taking the appropriate Gauss-Legendre quadrature points into account within each cell and multiplying the higher order coefficients with the appropriate point,  $\eta$ , in local coordinates.

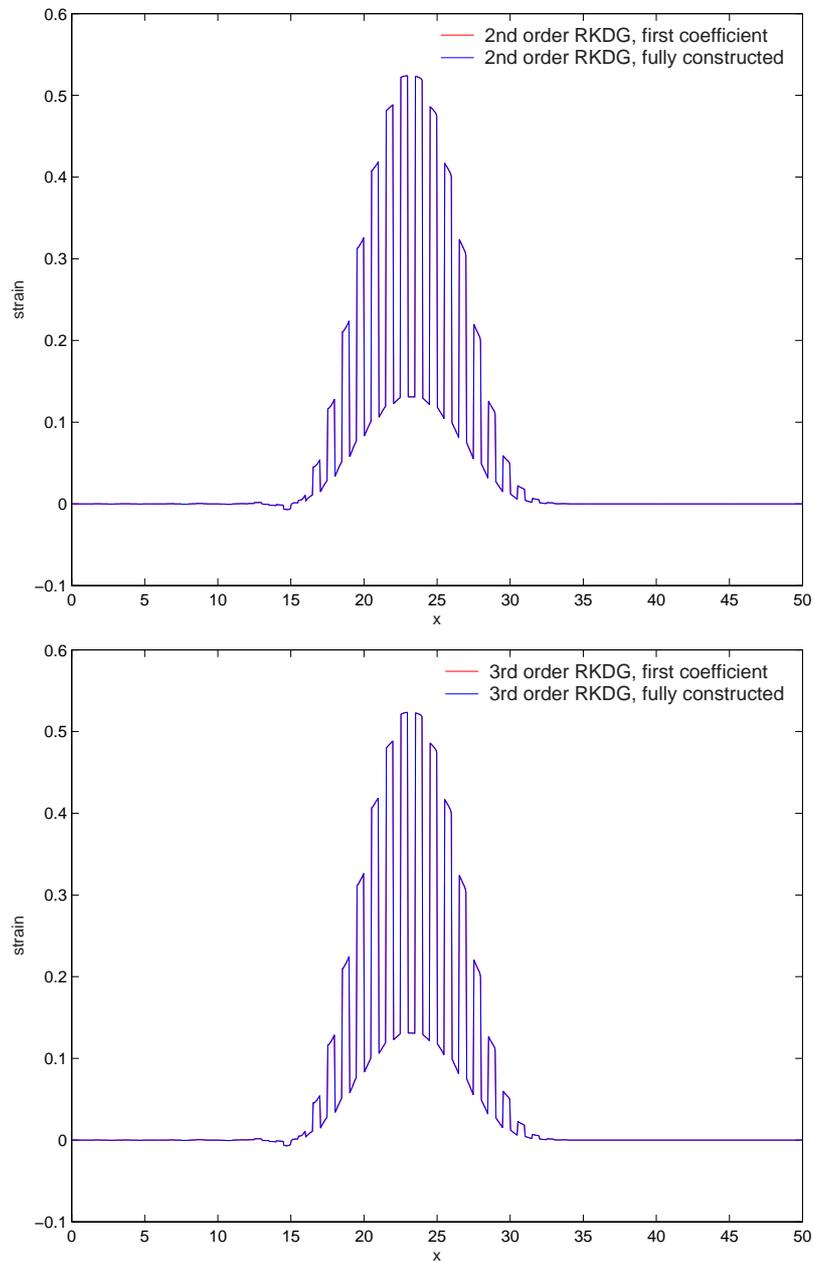


Figure 6.2: Strain Wave problem: Strain Vs X shown for the discontinuous Galerkin method. Upper panel:  $2^{nd}$  order scheme. Lower panel:  $3^{rd}$  order scheme. Red line shows the first expansion coefficient of the strain only, blue line shows the fully constructed solution using the linear term as well. Both solutions at 1000 grid points and time  $t=40$ . These figures are for layered media with mismatched impedances.

The coordinate transformations and quadrature points are described in Chapter 3. The upper panel of Figure 6.2 refers to the  $2^{nd}$  order RKDG method and the lower panel contains the comparisons for the  $3^{rd}$  order scheme. It is evident from these plots that there is very little difference between just using the first order coefficients and constructing the entire solution with the appropriate quadrature points. Therefore, it is possible to state that taking only the first order coefficients will suffice for the purposes of the comparisons performed here. The very subtle differences could be attributed to the limiters that are only applied to the first order coefficient.

The above comparisons are just performed to justify the choice of solution used as the converged result. Both solutions are extremely similar at 1000 grid points, but there are small, subtle differences. To be fair to both the schemes for the grid convergence study, the converged result used is that of 5000 grid points using the wave propagation algorithm for the study of the wave propagation scheme, and 5000 grid points using the  $3^{rd}$  order RKDG scheme with characteristics-based limiting for the RKDG convergence study. The accuracy of the methods are compared by computing the  $l_2$ -norm of each of the algorithms with various grid resolutions. A more fair and accurate comparison of the two methods follows through a grid convergence study. In the paragraphs that follow, RKDG is used to denote the Runge-Kutta discontinuous Galerkin method.

The various limiters of the wave propagation method described in Chapter 2 provide results that are either identical to each other or they vary by an insignificant amount. The two limiters of the RKDG method however, produce different results and involve varying computational effort. The characteristics-based limiter produces the best results with the drawback of larger computational effort as compared to the component-based limiter. Figure 6.3 displays the 300 grid point solution with respect to the converged solution for both algorithms.

Figure 6.3 shows that the Wave Propagation algorithm is diffusive for lower grid resolutions. The RKDG method however, is still quite competitive even at the lower resolutions. Therefore, for a thorough comparison, it is useful to run the WAVE algorithm at a higher grid resolution until its solution is comparable to that of the RKDG at 300 cells. Then, the computational effort involved is compared to determine which algorithm performs better to

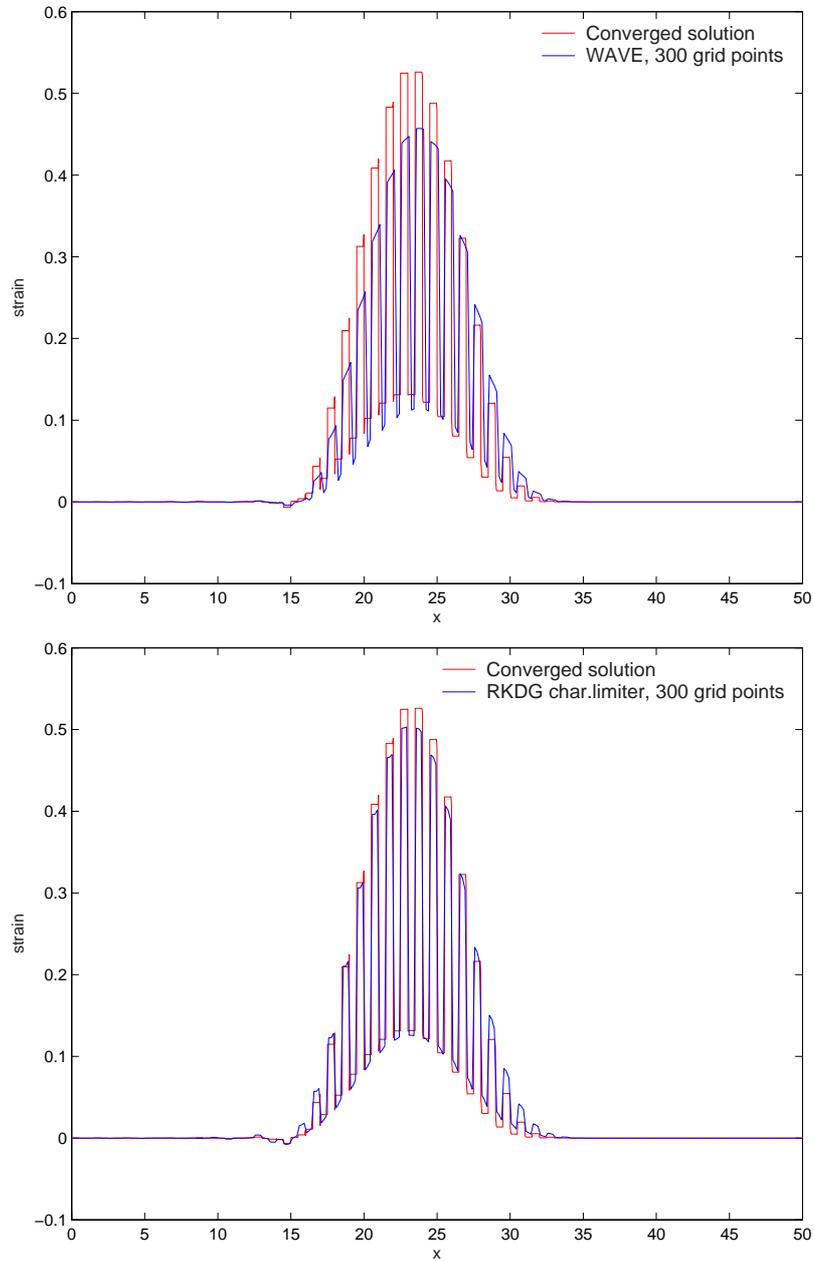


Figure 6.3: Strain Wave problem: Strain Vs X solutions computed using wave propagation scheme (upper panel) and the RKDG 2nd order scheme with characteristics based limiters (lower panel) on 300 cells as shown by the blue line. The red line is the converged solution at 5000 cells. These figures are for layered media with mismatched impedances.

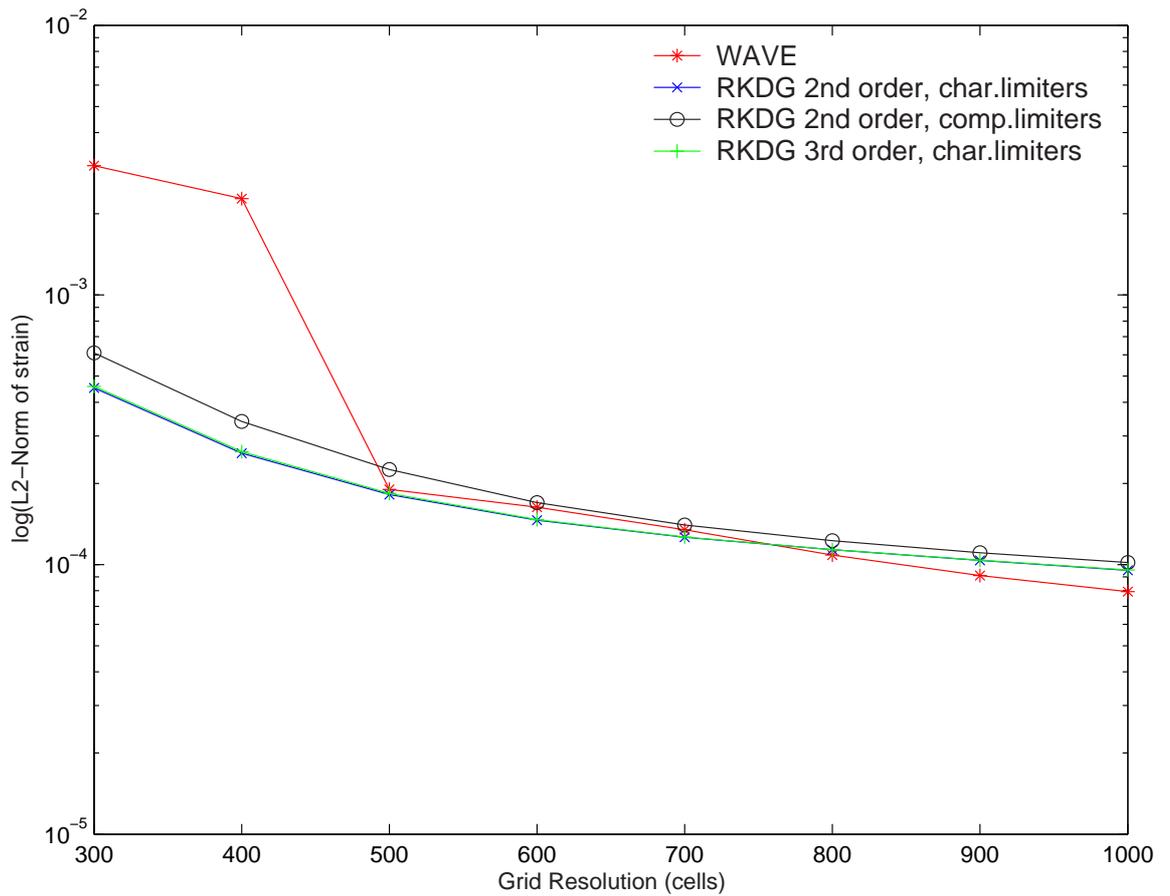


Figure 6.4: Strain Wave problem:  $l_2$ -norm of strain Vs Grid Resolution for the wave propagation Scheme (red),  $2^{nd}$  order RKDG with characteristics based limiters (blue),  $2^{nd}$  order RKDG with component based limiters (black), and  $3^{rd}$  order RKDG (green) for the strain at time  $t=40$  based on the converged result with 5000 cells.

Algorithm	CFL	Order	Time(sec)
J=1000 cells			
WAVE	0.5	2	19
RKDG with characteristic limiters	0.3	2	221
RKDG with component-based limiters	0.3	2	131.5
RKDG with characteristic limiters	0.2	3	426.5
J=800 cells			
WAVE	0.5	2	11.6
RKDG with characteristic limiters	0.3	2	140.7
RKDG with component-based limiters	0.3	2	84.2
RKDG with characteristic limiters	0.2	3	273.2
J=500 cells			
WAVE	0.5	2	4.71
RKDG with characteristic limiters	0.3	2	53.9
RKDG with component-based limiters	0.3	2	32.3
RKDG with characteristic limiters	0.2	3	107.8
J=300 cells			
WAVE	0.5	2	1.75
RKDG with characteristic limiters	0.3	2	19.5
RKDG with component-based limiters	0.3	2	12.3
RKDG with characteristic limiters	0.2	3	38.4

Table 6.1: Strain Waves problem: Some computational times for the wave propagation algorithm and the RKDG algorithm with both characteristic-based and component-based limiters for several grid resolutions.

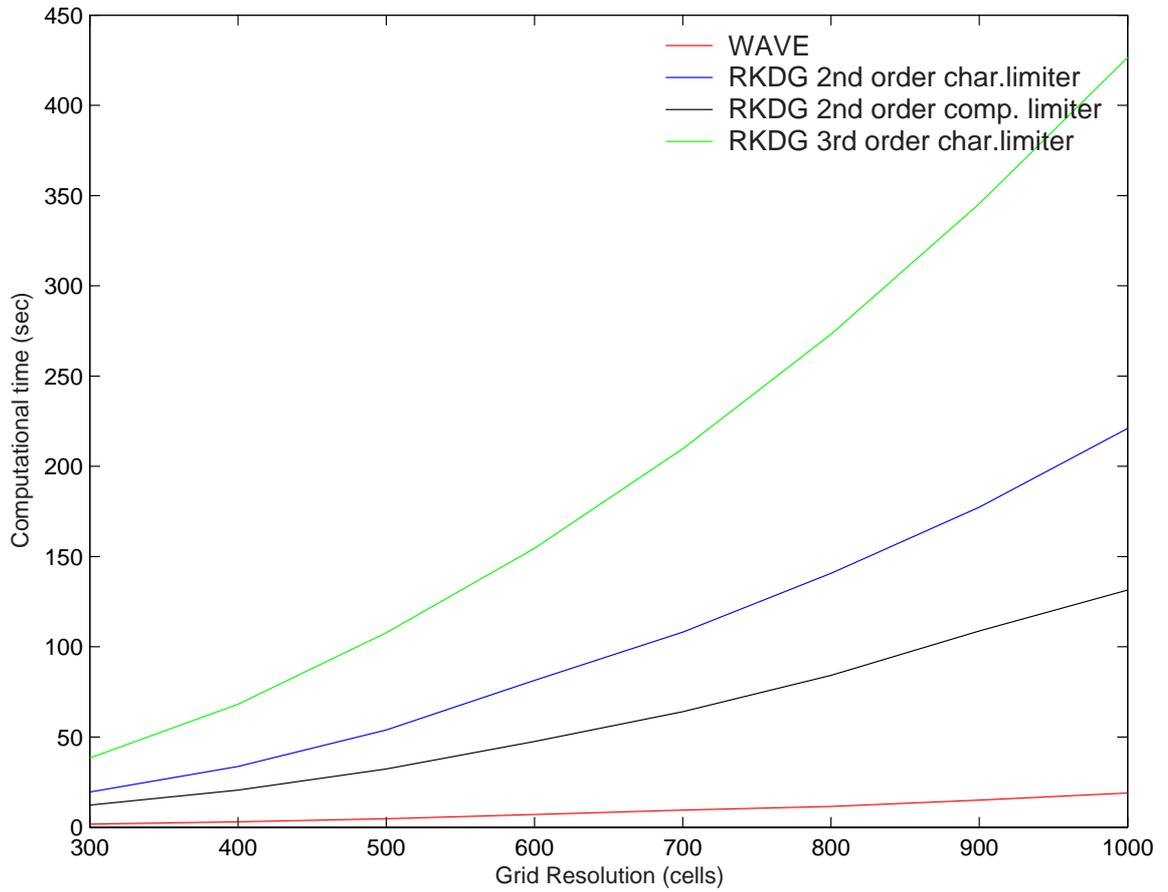


Figure 6.5: Strain waves problem: Computational time(sec) Vs Grid Resolution for the wave propagation method (red),  $2^{nd}$  order RKDG with characteristics-based limiters (blue),  $2^{nd}$  order RKDG with component-based limiters (black), and  $3^{rd}$  order RKDG with characteristics-based limiters (green). It is clear that the wave propagation method takes very little computational effort, followed by the  $2^{nd}$  order RKDG scheme with component-based limiting, then comes the  $2^{nd}$  order RKDG scheme with characteristics-based limiting, and finally the  $3^{rd}$  order RKDG method with its large computational effort.

achieve roughly the same level of accuracy. This is detailed in the paragraphs that follow.

Both algorithms are run with a number of grid resolutions and they are compared to the converged results using an  $l_2$ -norm to quantify the convergence. The  $l_2$ -norm is described by,

$$|\Delta y| = \frac{1}{n} \sqrt{\sum_{x=1}^n (y_{calc} - y_{converged})^2} \quad (6.4)$$

where  $y_{calc}$  is the obtained solution,  $y_{converged}$  is the converged solution and the solution is evaluated at every point in  $x$ . Linear interpolation is performed to determine the value of the converged solution at the desired locations. For this problem the values of the strain are used to determine the  $l_2$ -norm for a number of grid resolutions. Figure 6.4 shows that the  $l_2$ -norm decreases as the grid resolution is increased, i.e. both algorithms show a convergence. Initially, at lower grid resolutions (on the order of 300-400), the RKDG method outperforms the wave propagation method. The Wave Propagation algorithm is rather diffusive at these resolutions while the RKDG method does not deviate significantly from the converged result. However, as the grid resolution is increased, the Wave scheme rapidly catches up with the RKDG method at a grid resolution of about 500 and seems to perform well for resolutions higher than this. The wave propagation algorithm is compared to the RKDG method with component-based limiters at the point where they have the same accuracy in Figure 6.4 (i.e. where the red line crosses the black line). It is seen that even with the exact same accuracy at a grid resolution of about 500, the wave propagation algorithm takes only 1/9 the computational time of the  $2^{nd}$  order RKDG algorithm and about 1/25 the computational time of the  $3^{rd}$  order RKDG scheme, making it the preferred method for this equation system. It would be useful to study the computing time for the RKDG method at 300 grid points as compared to the WAVE scheme at 500 to get a better picture of the effectiveness of the methods. Table 6.1 displays the computational effort of both algorithms for several grid resolutions. The point to note is that, in addition to the stability and accuracy of the algorithms, the computational effort involved is much higher for the RKDG than it is for the Wave method. At 300 cells, the  $2^{nd}$  order RKDG method takes about 20seconds with the characteristics based limiters, and about 12seconds with the

component based limiters. At 500 cells, the WAVE scheme takes about 5*seconds* to provide a reasonable solution. Also, it is seen that the 3<sup>rd</sup> order RKDG method is not superior to the 2<sup>nd</sup> order scheme with characteristics-based limiting. The blue and green lines in Figure 6.4 overlap. The wave propagation method seems to achieve the same accuracy as the 3<sup>rd</sup> order RKDG method at a grid resolution of about 750 following which, the Wave Propagation scheme provides better accuracy. Beyond a grid resolution of about 500 cells, the wave propagation scheme performs very well both in terms of accuracy and computational effort (and better than the RKDG method with component-based limiting). If accuracy is the biggest requirement and computational time is not an issue, then the RKDG 2<sup>nd</sup> order method with characteristic-based limiting is the best option for grid resolutions lower than about 750 for this problem. This accuracy does not differ much from that of the wave propagation algorithm if the grid resolution is higher than 500 cells. Therefore, for an overall accurate solution with an efficient use of computing time, the wave propagation algorithm performs better for grid resolutions of over 500.

For the RKDG algorithm, the characteristic-based limiting can be made to be as competitive or even better than component-based limiting in terms of computational time. This is done by applying a problem specific characteristics-based limiter routine than using a generally supplied characteristics-based limiter routine that is applicable for all problems. To elaborate, characteristics-based limiters involve splitting the higher order coefficients and the coefficients of the forward and backward differences of the averages to compute the waves. These waves are used to determine the conserved quantities that are limited using the modified min-mod limiter described in Chapter 3. When used in a general routine, the Riemann problem solver needs to be called several times to split each of these coefficients and this can increase the computation time. Hence, it is computationally effective to write a separate problem-specific routine that calculates the coefficients and the waves for each application and saves the computation time involved in each additional call to the Riemann problem solver. In this way, the use of the characteristic-based limiters can now be justified for all problems over the use of component based limiters due to its increased accuracy in addition to its comparable computational effort. For the problems in the following sections, this local problem-specific characteristics-based limiting is applied and it is evident that the

characteristics-based limiters are superior to the component-based ones.

It is noted from Table 6.1 that the wave propagation algorithm takes less than 1/10 the time to provide a solution as compared to the 2<sup>nd</sup> order RKDG method and it takes about 1/25 time as compared to the 3<sup>rd</sup> order RKDG algorithm for the same grid resolution at the same user-specified time. Hence, for this problem, it is more computationally effective to run the wave propagation method at a higher grid resolution and obtain an accurate, stable result in much less time as compared to an almost equally accurate RKDG solution with a much lower grid resolution. Figure 6.5 clearly displays the computational effort. This problem only has one significant speed for disturbances to propagate, and with such problems the WAVE algorithm provides good results. In fact, if the WAVE scheme can be run with a CFL of 1.0 for problems with only one characteristic speed, the solution propagates almost exactly without any diffusive or dispersive errors.

## 6.2 *Electron Acoustic Waves*

The *Dispersive Euler* equation system is used for the electron acoustic wave problem. The equation set used is described in Chapter 4 in Eq. (4.6). The Euler equations are taken and are modified to contain source terms that would allow for dispersions. These dispersions lead to a solution for the electron acoustic pulse. The source terms that are used are equivalent to having an electron cyclotron frequency. The reason for using this equation system instead of the two-fluid model here is to be able to analyze a simpler system of equations that would provide exactly the same solution as the full two-fluid model. Once the algorithms are analyzed and compared using simpler models, they are then applied to the full two-fluid model. The advantage of the electron acoustic wave problem is that it has an exact solution.

### 6.2.1 *Analytical Solution*

The electron acoustic pulse problem has an exact solution and the derivations leading to this exact solution for the dispersion relation using the *Dispersive Euler* model are displayed here. As noted earlier, the Euler equations are modified to contain dispersive source terms which is the reason these equations are referred to as *Dispersive Euler* by Hakim [4]. In

non-conservative form, the continuity, momentum and energy equations are,

$$\frac{\partial \rho}{\partial t} + \rho \frac{\partial u}{\partial x} + u \frac{\partial \rho}{\partial x} = 0 \quad (6.5)$$

$$\rho \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} \right) = -\frac{\partial p}{\partial x} + \rho v \frac{q}{m} B_z \quad (6.6)$$

$$\rho \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} \right) = -\rho u \frac{q}{m} B_z \quad (6.7)$$

$$\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} = -\gamma p \frac{\partial u}{\partial x}, \quad (6.8)$$

where  $p$  is the pressure,  $q$  is the charge,  $m$  is the mass and  $B_z$  is any spatially dependent vector which in this case is chosen to be the magnetic field in the  $z$ -direction. In this manner, the cyclotron frequency,  $\omega_c = \frac{q}{m} B_z$  can be introduced into the source terms of these Euler equations.  $B_x = B_y = 0$  is assumed for simplicity. These equations are linearized such that  $\rho = \rho_0 + \rho_1$ ,  $p = p_0 + p_1$  and  $u_0 = v_0 = 0$ , where subscript 0 denotes the equilibrium state terms and subscript 1 stands for the perturbations. After introducing the linear terms the products of the perturbations are neglected as they are assumed to be very small and the equilibrium terms are canceled appropriately to get the linear system,

$$\frac{\partial \rho_1}{\partial t} = -\rho_0 \frac{\partial u_1}{\partial x} \quad (6.9)$$

$$\rho_0 \frac{\partial u_1}{\partial t} = -\frac{\partial p_1}{\partial x} + \rho_0 v_1 \frac{q}{m} B_z \quad (6.10)$$

$$\rho_0 \frac{\partial v_1}{\partial t} = -\rho_0 u_1 \frac{q}{m} B_z \quad (6.11)$$

$$\frac{\partial p_1}{\partial t} = -\gamma p_0 \frac{\partial u_1}{\partial x}. \quad (6.12)$$

Assume that the perturbations are of the form described by the Fourier transform,

$$f(x, t) = \sum_{n=0}^{\infty} f_n e^{i(k_n x + \omega_n t)} \quad (6.13)$$

where  $k_n$  represents the wave number and  $\omega_n$  stands for the frequency for each term of the expansion. Applying the Fourier transform to the linear equations,

$$i\omega_n \rho_1 = -ik_n u_1 \rho_0 \quad (6.14)$$

$$i\omega_n u_1 \rho_0 = -ik_n p_1 + \lambda v_1 b_z \quad (6.15)$$

$$i\omega_n v_1 \rho_0 = -\rho_0 \lambda u_1 b_z \quad (6.16)$$

$$i\omega_n p_1 = i\gamma p_0 k_n u_1 \quad (6.17)$$

are obtained. Solving the equation system,

$$\begin{bmatrix} i\omega_n\rho_0 & -\rho_0\frac{q}{m}B_z & ik_n \\ \rho_0\frac{q}{m}B_z & i\omega_n\rho_0 & 0 \\ i\gamma p_0 k_n & 0 & i\omega_n \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ p_1 \end{bmatrix} = 0 \quad (6.18)$$

by setting the determinant of the matrix equal to 0, the following dispersion relation is obtained:

$$\omega_n = \pm (k_n^2 c_{s0}^2 + \omega_c^2)^{1/2} \quad (6.19)$$

where  $\omega_c$  is the cyclotron frequency described previously and  $c_{s0}$  is the speed of sound that is given by  $\sqrt{\gamma p_0/\rho_0}$ . It is noted that the dispersion relation is not linear in  $k$  due to the second term in Eq. (6.19) and instead is quadratic, leading to the dispersions.

### 6.2.2 Description of Problem

The problem involves initiating a square pulse that is approximated with a Fourier expansion. This is done for the velocity, such that  $u = \sum_{n=0}^{\infty} u_1(x, t)$  with  $u$  being a function of  $x$  and  $t$ .  $p_0$  and  $\rho_0$  are initialized as constant values with

$$u_1(x, t) = U_0 \sum_{n=0}^N \frac{i}{2n+1} e^{ik_n x} e^{i\omega_n t} \quad (6.20)$$

such that  $k_n = 2\pi(2n+1)$  is defined for the function approximation. It is seen that as  $N \rightarrow \infty$ , Eq. (6.20) represents a step function for the interval  $[0, 1]$ . Therefore, at each Fourier component,  $n$ ,

$$u_i^{(n)} \equiv iU_0/(2n+1)e^{i(k_n x + \omega_n t)} \quad (6.21)$$

$$\rho_1^{(n)} = -\frac{k_n \rho_0}{\omega_n} u_1^{(n)} \quad (6.22)$$

$$v_1^{(n)} = -i\frac{qB_z}{m\omega_n} u_1^{(n)} \quad (6.23)$$

$$p_1^{(n)} = -\frac{\gamma k_n p_0}{\omega_n} u_1^{(n)}, \quad (6.24)$$

are defined such that when each of these variables is summed over  $n = 0 \dots N$ , the exact solution to the linear problem is obtained.  $B_z$  can be spatially dependent in these equations.

The exact solution for  $u(x, t)$  is given by

$$u(x, t) = - \sum_{n=0}^{\infty} \frac{U_0}{2n+1} \sin(2\pi(2n+1)x + \omega_n t) \quad (6.25)$$

for the step function approximation and this is used to analyze the numerically obtained solutions. The  $\omega_n$  used is from Eq. (6.19).

Figure 6.7 shows the initial condition for  $N = 9$  and Figure 6.6 shows the exact analytical solution for  $t = 1000$ ,  $\gamma = 2$ ,  $q/m = 10$ ,  $\rho_0 = p_0 = 1$ ,  $B_z = 1$ , and  $N = 5000$ , such that  $c_s = \sqrt{2}$  and  $\omega_c = 10$ .  $x$  is discretized on a domain  $[0, 1]$  with 512 cells. It is evident that there are a large number of oscillations and sharp gradients in the exact analytical solution that could be mistaken for numerical dispersion. This problem has quite complex solutions that need to be properly resolved by the algorithms used. This is a good test problem to assess how the algorithms resolve the small scale features in comparison to the exact solution and at what computational expense. These studies are performed in the following section where the limitations of both algorithms are considered.

### 6.2.3 Results

Figure 6.7 shows the initial condition when  $N = 9$  Fourier modes are taken into account. As mentioned earlier, the parameters used are  $\gamma = 2$ ,  $q/m = 10$ ,  $B_z = 1$ ,  $\rho_0 = p_0 = 1$ ,  $c_s = \sqrt{2}$  and  $\omega_c = 10$  with the 9 Fourier modes. The wave propagation scheme is compared to the RKDG scheme for these same parameters. The interest is in the linear regime here; to ensure a linear solution,  $U_0 = 10^{-8}$  is used. Based on Eq. (6.21-6.24), the initial conditions for all the other variables are given by the exact solutions of these variables at time  $t = 0$  after adding these calculated perturbed quantities to the equilibrium values. The CFL numbers used here are 1 for the wave propagation method and  $1/3$  for the  $2^{nd}$  order RKDG scheme. The temporal order of the RKDG method is  $3^{rd}$ . All the results that are compared, are for time  $t = 3$ . The analytical solution in all the results is for all the same parameters described in the problem.

Figure 6.8 compares the wave propagation scheme to the RKDG scheme with both of its limiters, and both schemes are compared to the analytical solution. It is seen that even at a resolution of 40 cells, the wave propagation algorithm performs rather well. The RKDG

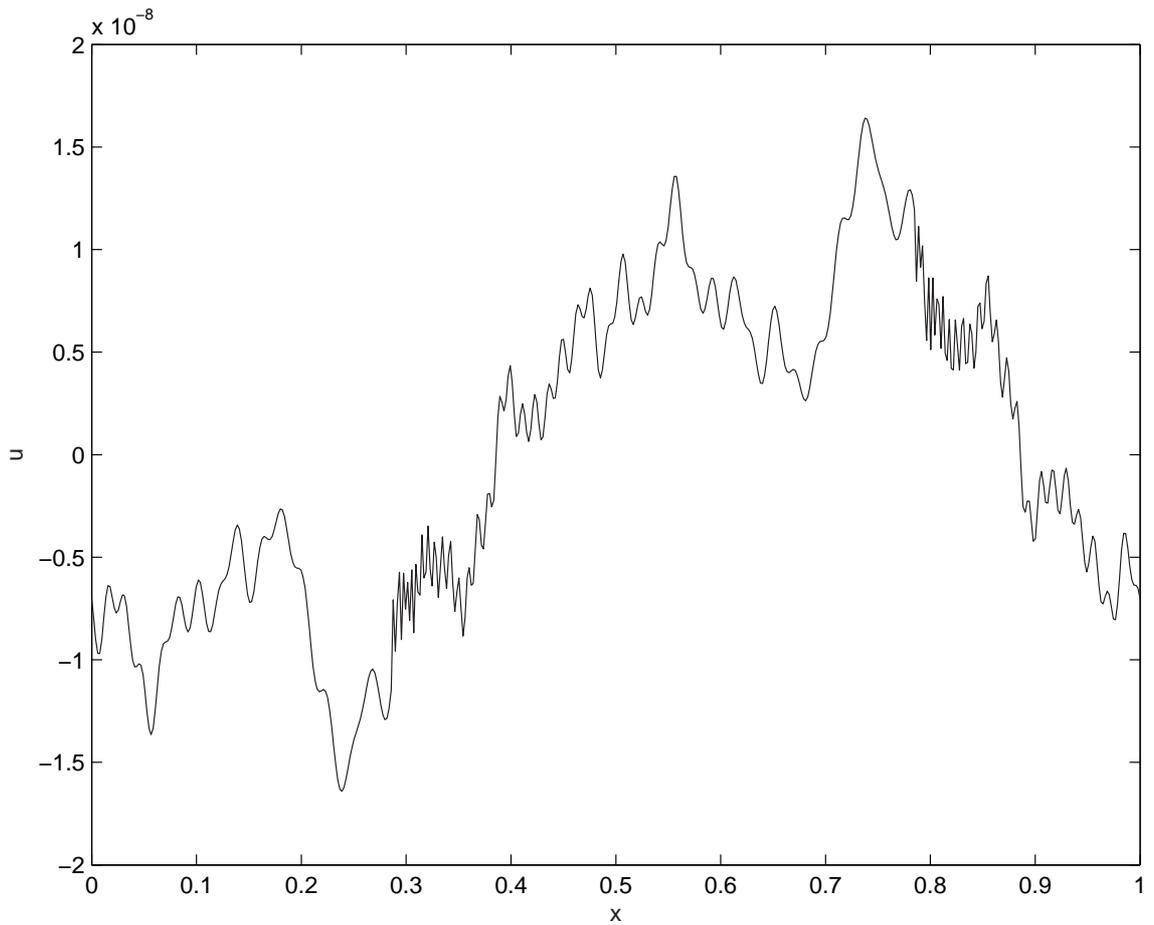


Figure 6.6: Electron Acoustics problem: The exact solution for  $N=5000$  at  $t=1000$ ,  $u$  Vs  $x$  with  $c_s = \sqrt{2}$  and  $\omega_c = 10$ . This is an example of an exact solution using Eq. (6.25). Notice that the solution is rather complicated with a number of small wavelength dispersions that need to be well resolved.

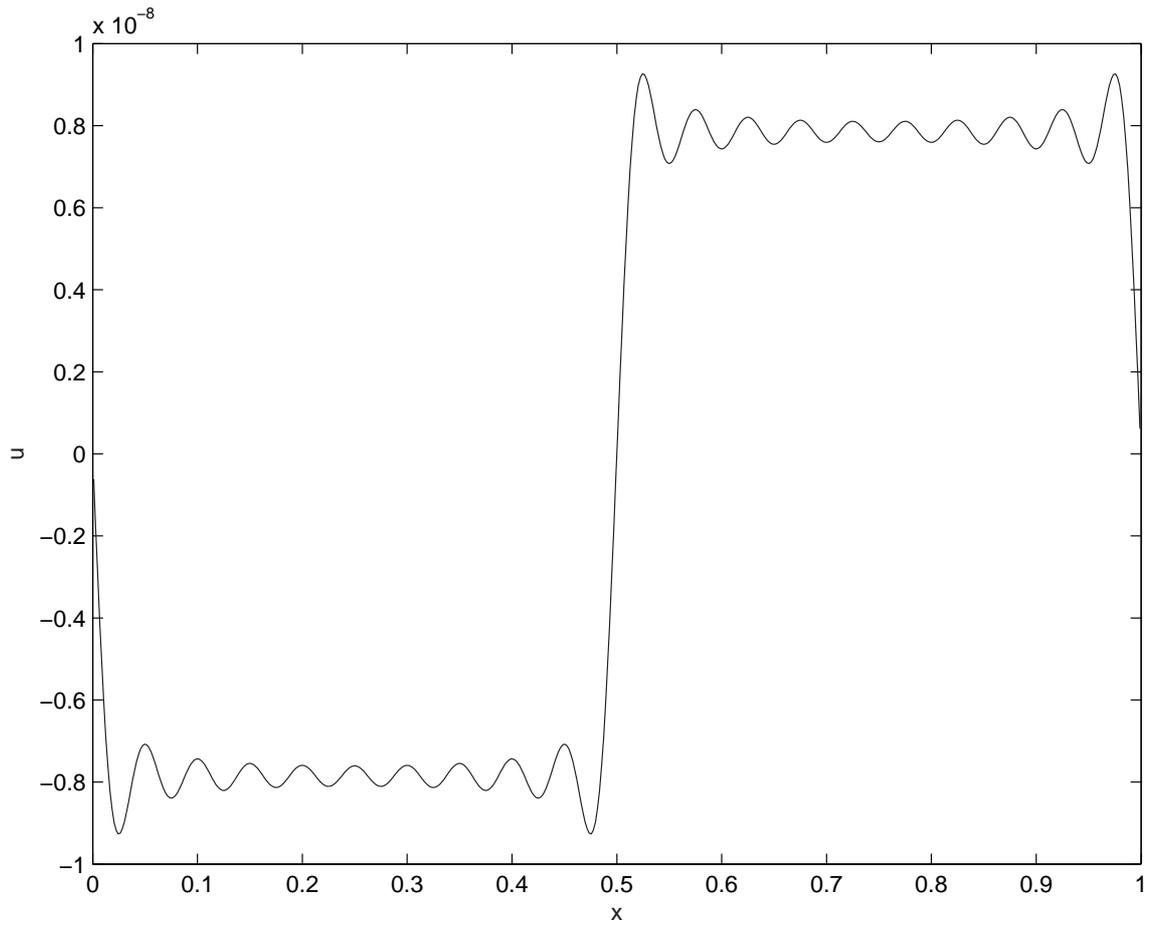


Figure 6.7: Electron Acoustics problem: Initial condition for  $N=9$  that is used to approximate the step function. This initial condition is used for all the results obtained from using both the algorithms in this section.  $u$  Vs  $x$  with  $c_s = \sqrt{2}$  and  $\omega_c = 10$ .

however, is quite diffusive and does not capture some of the smaller wavelength features at all. Also, the results from both RKDG limiters are quite comparable for this problem.

Figure 6.9 shows the WAVE and the RKDG schemes as compared to the analytical solution for 120 grid points. It is noted that using a CFL of 1 for the wave propagation scheme eliminates the errors introduced by diffusion and dispersion to a large extent in this problem. The wave propagation solution is almost that of the analytical solution, whereas the RKDG solution is diffusive at the points with smaller wavelength possibly due to the presence of limiters along with the smaller time steps that are taken to maintain the CFL stability condition of this scheme. Using a  $3^{rd}$  spatial order RKDG scheme would resolve these diffusive issues, however it would be even more computationally intensive.

The solutions from both the algorithms are compared to the analytical solution by taking an  $l_2$ -norm of the velocities for a number of grid resolutions. This offers a way to quantify the accuracy of both algorithms. Figure 6.10 displays the  $l_2$ -norms of both the algorithms as the grid resolution is increased. It is very clear that the wave propagation method is superior to the RKDG scheme for this problem for the given parameters. This could be attributed to the fact that if there is only one significant wave speed in the system, the wave propagation algorithm can be run with  $CFL = 1$  to give almost an exact result. This is the case with the electron acoustic pulse problem using the Dispersive Euler model.

Table 6.2 displays the computational time taken for both the Wave algorithm and the RKDG scheme with characteristics based and component-based limiters for several grid resolutions. Since the computational times with the use of both RKDG limiters are comparable, the  $3^{rd}$  order RKDG method is only explored with characteristics-based limiting. It is seen that, just like with the case of the strain waves problem, the wave propagation algorithm takes about 1/10 the time to provide results to higher order accuracy as compared to the  $2^{nd}$  order RKDG method and about 1/30 the time of the  $3^{rd}$  order RKDG method to provide results with an accuracy that is slightly lower. Comparing the wave propagation scheme to the  $3^{rd}$  order RKDG scheme at a grid resolution of about 80 cells, it is seen from Figure 6.10 that at this grid resolution, both methods have the same accuracy. Comparing the times for these two methods at this resolution, the wave propagation method takes 1 sec. to run while the  $3^{rd}$  order RKDG method takes 28 sec. Hence, this makes

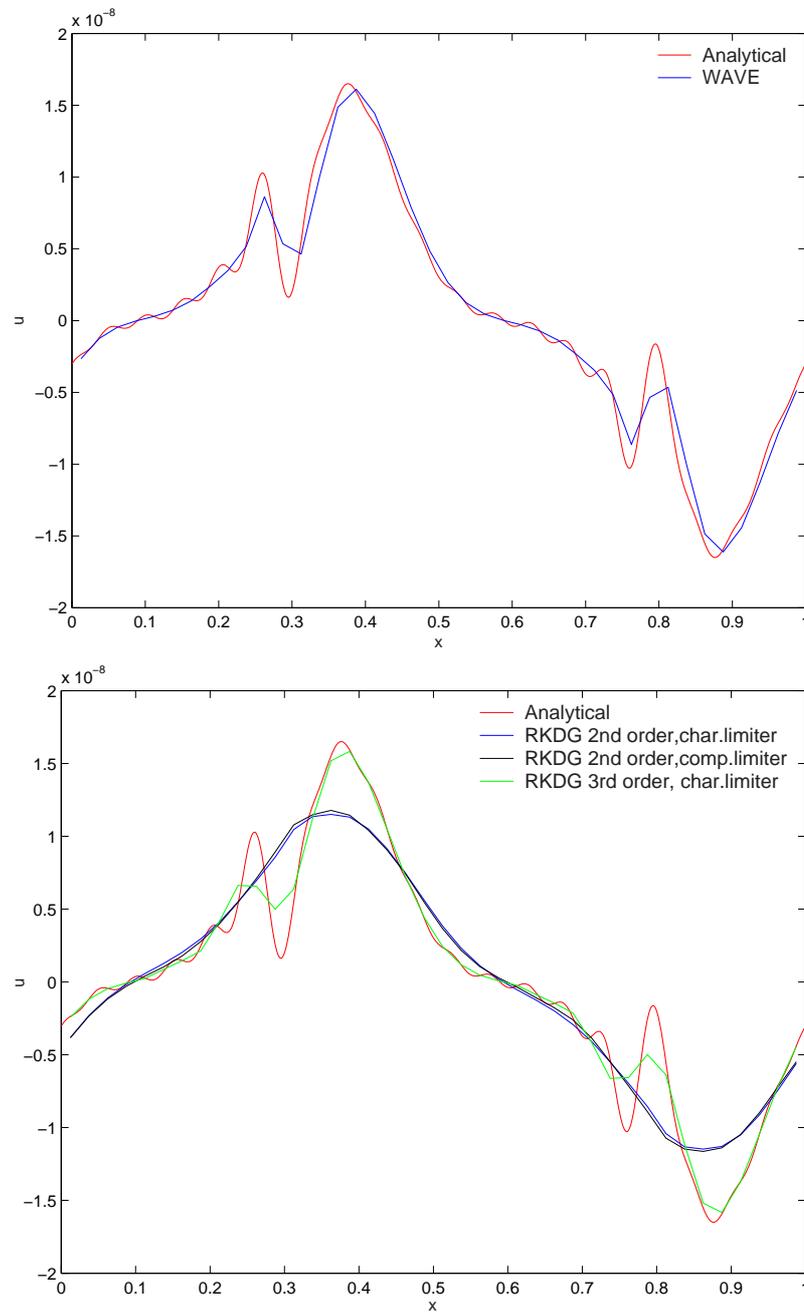


Figure 6.8: Electron Acoustics problem: Plots are of  $u$  Vs  $x$  for 40 cells with  $c_s = \sqrt{2}$  and  $\omega_c = 10$ . The upper panel shows the WAVE algorithm (blue) and the analytical solution (red) and the lower panel shows the RKDG algorithm with its characteristics-based limiters (blue) and component-based limiters (black) along with the analytical solution (red).

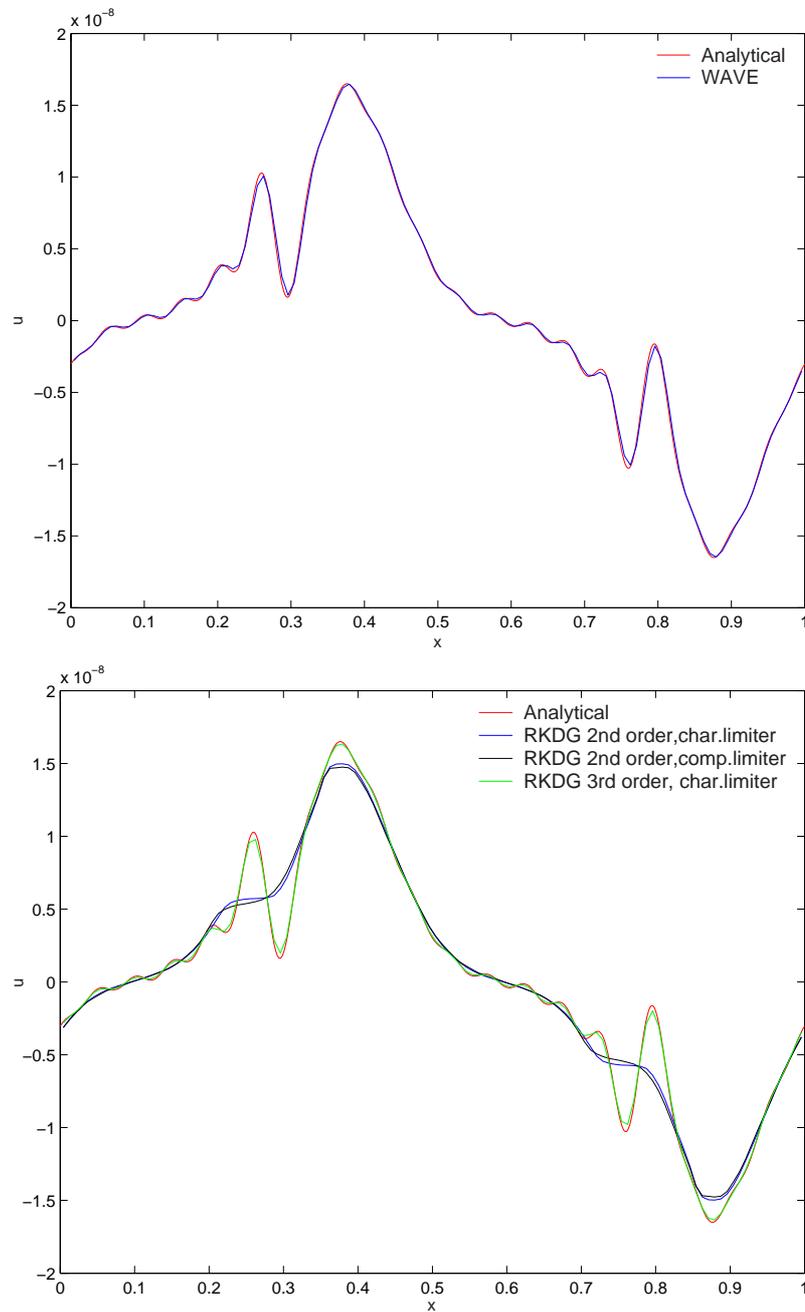


Figure 6.9: Electron Acoustics problem: Plots are of  $u$  Vs  $x$  for 120 cells with  $c_s = \sqrt{2}$  and  $\omega_c = 10$ . The upper panel shows the WAVE algorithm (blue) and the analytical solution (red) and the lower panel shows the RKDG algorithm with its characteristics-based limiters (blue) and component-based limiters (black) along with the analytical solution (red).

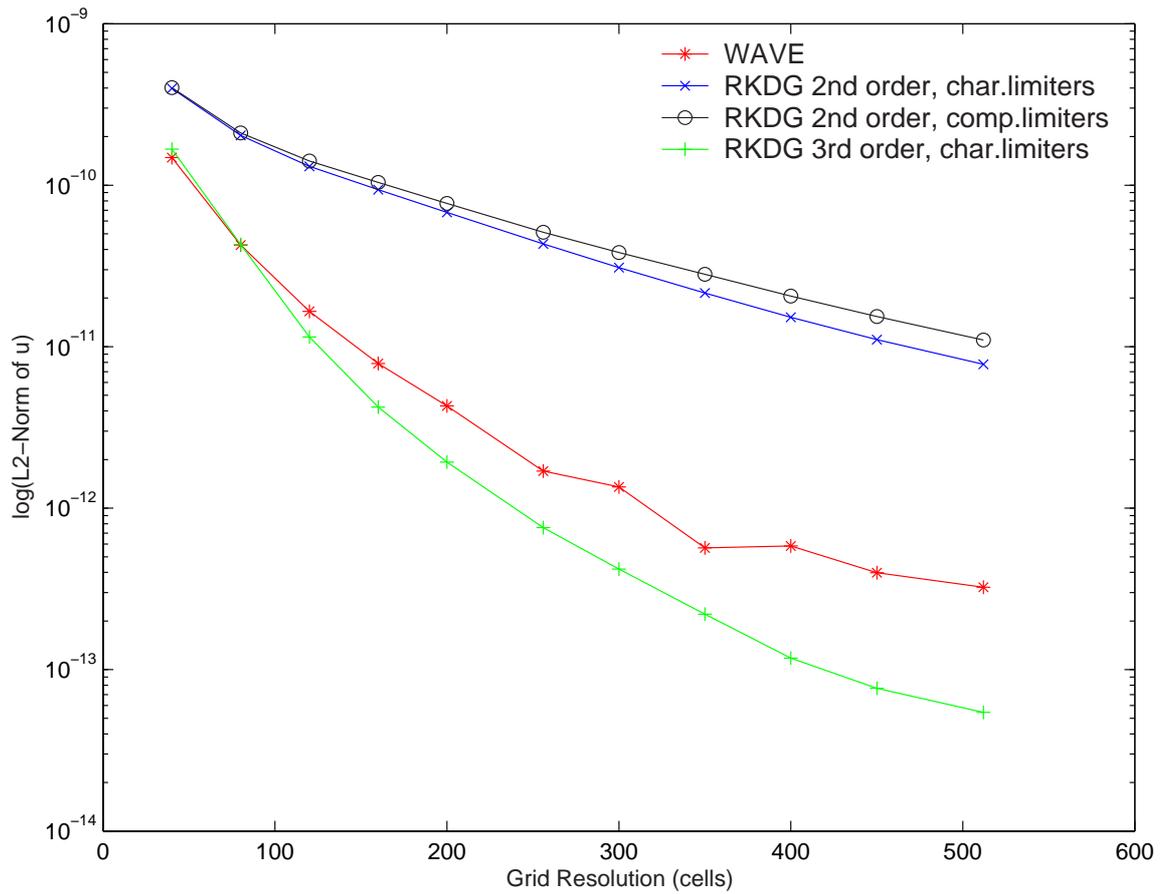


Figure 6.10: Electron Acoustics problem:  $l_2$ -norm of velocity Vs Grid Resolution for the wave propagation method (red),  $2^{nd}$  order RKDG with characteristics-based limiters (blue),  $2^{nd}$  order RKDG with component-based limiters (black), and  $3^{rd}$  order RKDG with characteristics-based limiters (green) as compared to the exact analytical solution with  $c_s = \sqrt{2}$  and  $\omega_c = 10$ .

Algorithm	CFL	Order	Time(sec)
J=512 cells			
WAVE	1.0	2	39.8
RKDG with characteristic limiters	0.3	2	432.1
RKDG with component-based limiters	0.3	2	437.7
RKDG with characteristic limiters	0.2	3	1129.4
J=256 cells			
WAVE	1.0	2	15.5
RKDG with characteristic limiters	0.3	2	109.3
RKDG with component-based limiters	0.3	2	110.4
RKDG with characteristic limiters	0.2	3	282.5
J=120 cells			
WAVE	1.0	2	2.21
RKDG with characteristic limiters	0.3	2	24.1
RKDG with component-based limiters	0.3	2	24.6
RKDG with characteristic limiters	0.2	3	62.7
J=40 cells			
WAVE	1.0	2	0.24
RKDG with characteristic limiters	0.3	2	2.82
RKDG with component-based limiters	0.3	2	2.85
RKDG with characteristic limiters	0.2	3	7.2

Table 6.2: Dispersive Euler Electron Acoustic Wave problem: Some computational times for the wave propagation algorithm and the RKDG algorithm with both characteristic-based and component-based limiters for several grid resolutions.

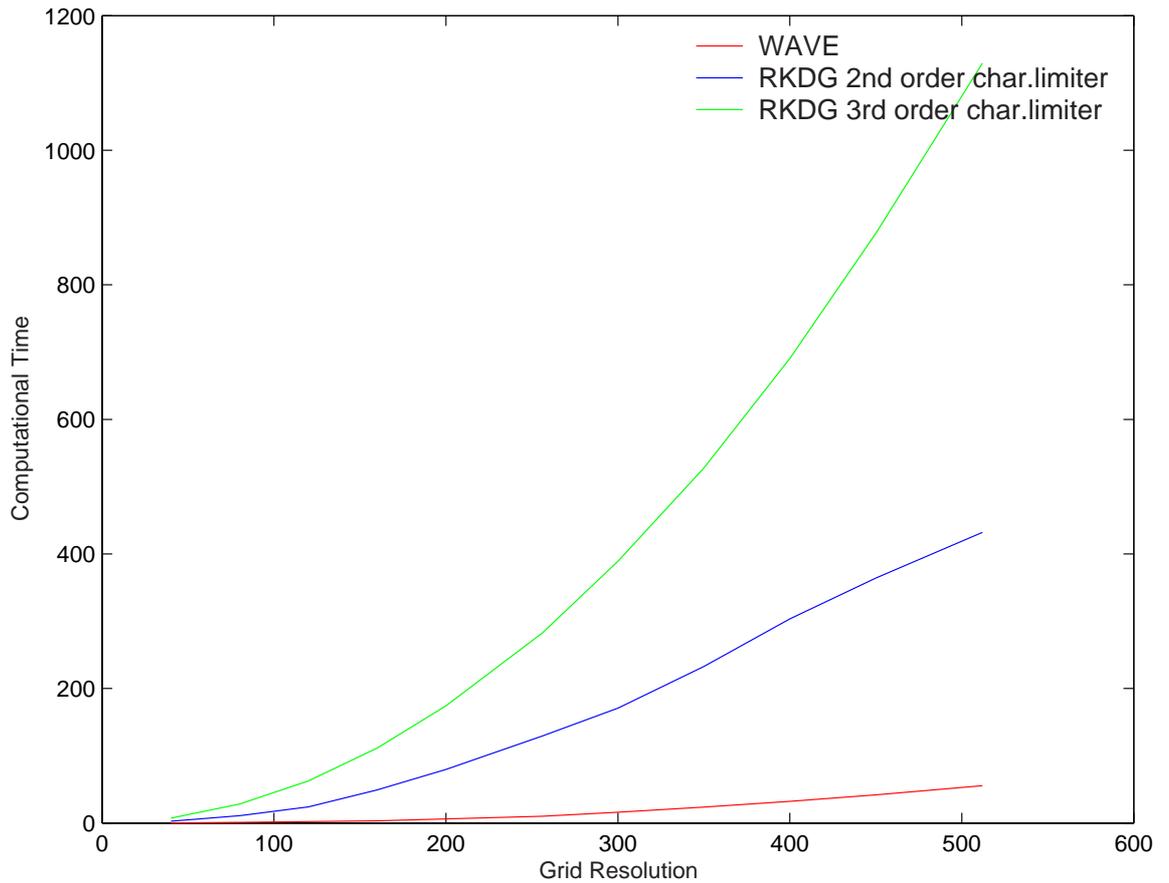


Figure 6.11: Electron Acoustics problem: Computational time(sec) Vs Grid Resolution for the wave propagation method (red),  $2^{nd}$  order RKDG with characteristics-based limiters (blue), and  $3^{rd}$  order RKDG with characteristics-based limiters (green) for  $c_s = \sqrt{2}$  and  $\omega_c = 10$ . It is noted that the  $3^{rd}$  order RKDG method takes much higher computational effort while the wave propagation algorithm takes very little.

the wave propagation algorithm more computationally effective. The two limiters applied to the RKDG scheme are comparable in this case in terms of computational effort. As discussed previously with the strain waves problem, the use of characteristics-based limiters can be justified for all problems over the use of component-based limiters if the limiting is implemented appropriately for each problem. In this case, the characteristics-based limiters take less time and provide higher accuracy as compared to the component-based limiters.

With the  $2^{nd}$  spatial order RKDG scheme taking such large computing time for a solution that is not better than the wave propagation algorithm, it is possible to say that the wave propagation method performs better. However, if increased accuracy is desired for this problem, and computational effort is not an issue, then the  $3^{rd}$  order RKDG method is the best option. It is noted from Figure 6.10 that the  $3^{rd}$  order RKDG method gives the most accurate results as compared to both the  $2^{nd}$  order RKDG and the wave propagation schemes but Figure 6.11 shows that it is very computationally intensive. Therefore, it can be concluded that for overall performance in terms of high accuracy and efficient computational time, the wave propagation method is superior for this equation system and this is evident from Table 6.2. However, if high accuracy is the biggest requirement with computational time not being a deciding factor, the  $3^{rd}$  order RKDG scheme is the best option.

Based on the information noted above, it is evident that the wave propagation scheme outperforms the RKDG method significantly for this problem as well. It provides a better solution and uses only 1/10 of the computational time of the RKDG method. There is an exception to this behavior though. If the problem is run with  $\omega_c = 100$  instead of 10, the Wave algorithm encounters phase errors that the RKDG does not at lower grid resolutions. This can be seen in Figure 6.12. The reason for these errors is that in general, there are larger source terms from  $\omega_c$  as compared to the other parameters in the advection part. Hence, the source term handling becomes significant in this case. The wave propagation method uses the source term splitting detailed in Chapter 2 and this splitting is the reason for the phase errors. A typical time scale can be described by  $\tau_c = 2\pi/\omega_c$  which represents an oscillation time for the solution. A typical time for information to propagate is described by  $\tau_s = \Delta x/c$  where  $c$  is a wave propagation speed in the model. Now, for the oscillation to be well resolved,  $\tau_c$  must be greater than  $\tau_s$  by a few magnitudes. For the case of the wave

propagation method with 100 cells, this requirement is violated because  $\tau_c = 0.063$  while  $\tau_s = 0.071$  leading to the phase error in Figure 6.12 because the oscillations are not well resolved. The RKDG method is superior in this regard as it provides a solution that is more comparable to the exact solution for the same order of accuracy. For the wave propagation scheme, there are two ways around this phase error. One is to increase the grid resolution significantly, and the other is to take smaller time steps (i.e. lower CFL number). Taking smaller time steps however, makes the solution diffusive and this is not desirable.

It can be seen from Figure 6.13 that as the grid resolution is increased with higher  $\omega_c$ , the wave propagation algorithm overcomes the phase shift error. The solution in Figure 6.13 is for 512 cells and it is seen that the phase shift errors are almost gone at this resolution. For lower grid resolutions and higher  $\omega_c$ , the RKDG method is superior. It does not resolve the smaller wavelengths as well at  $2^{nd}$  order but increasing the spatial order to  $3^{rd}$  makes the solution a little better. The  $3^{rd}$  order method does take about twice as long to run as the  $2^{nd}$  order method, and the solution is not significantly better in terms of accuracy as noted from the lower panel of Figure 6.12. It can be argued that a slightly diffusive solution (RKDG) is preferable to a phase shifted solution. In general, the RKDG method handles problems with large source terms without introducing additional errors even when running at lower grid resolutions.

Several source term handling methods are explored with this problem that include the semi-implicit source term handling described by Eq.(2.27) and the implicit source term handling described by Eq.(2.30). The semi-implicit source term handling is first explored here. This is a split scheme as well, but instead of using the Runge-Kutta time stepping method, the trapezoidal method is used. The results obtained by applying this source term handling method are shown in plots of Figure 6.14 for both 160 cells and 512 cells. For the same parameters as Figures 6.12 and 6.13, i.e. with  $c_s = \sqrt{2}$  and  $\omega_c = 100$ , the semi-implicit scheme takes slightly more computational time and it displays phase errors similar to the Runge-Kutta time stepping scheme. The phase errors here, however, seem to be more severe than earlier. Increasing the grid resolution reduces these phase errors just like with the case of the Runge-Kutta scheme. For 160 cells the semi-implicit scheme takes 1 second more computing time than the Runge-Kutta time stepping scheme and produces

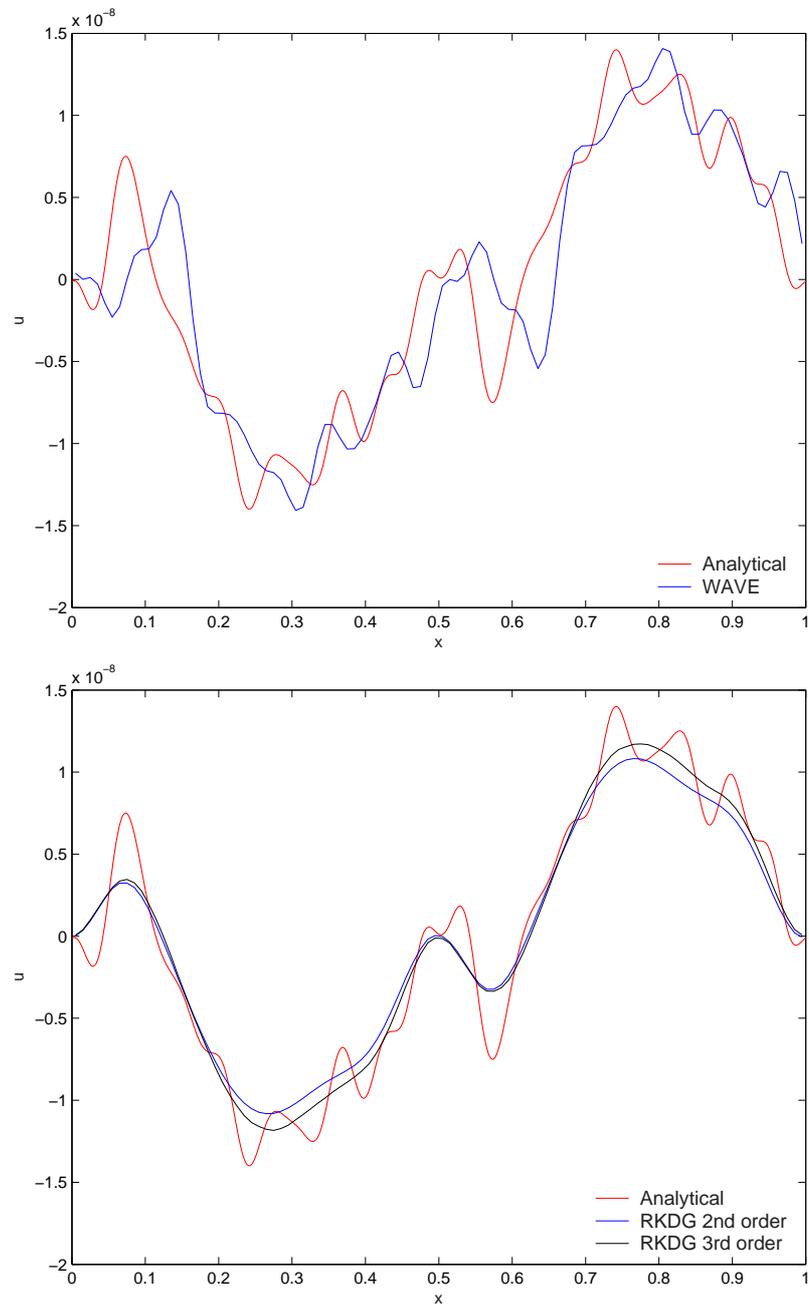


Figure 6.12: Electron Acoustics problem:  $u$  Vs  $x$  for 100 cells with  $c_s = \sqrt{2}$  and  $\omega_c = 100$ . Upper panel: wave propagation scheme (blue) as compared to the analytical solution (red) where the phase shift is noted, Lower panel: RKDG  $2^{nd}$  order (blue) and  $3^{rd}$  order (black) with characteristics-based limiters as compared to the analytical solution (red).

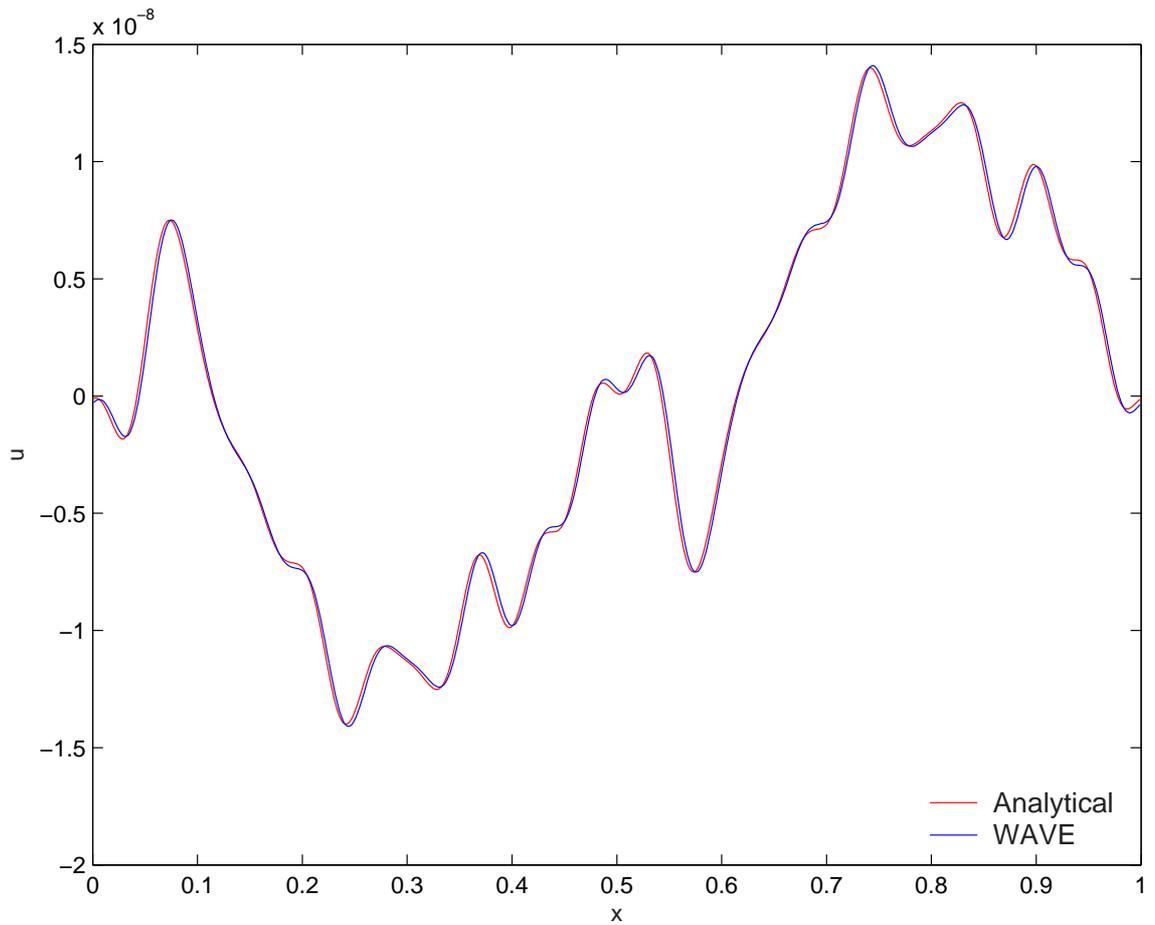


Figure 6.13: Electron Acoustics problem:  $u$  Vs  $x$  for the Wave Propagation method with source term splitting (blue) as compared to the analytical solution (red) for  $c_s = \sqrt{2}$  and  $\omega_c = 100$  with 512 cells. A slight phase error is still seen here, but increasing the grid resolution even further will fix this.

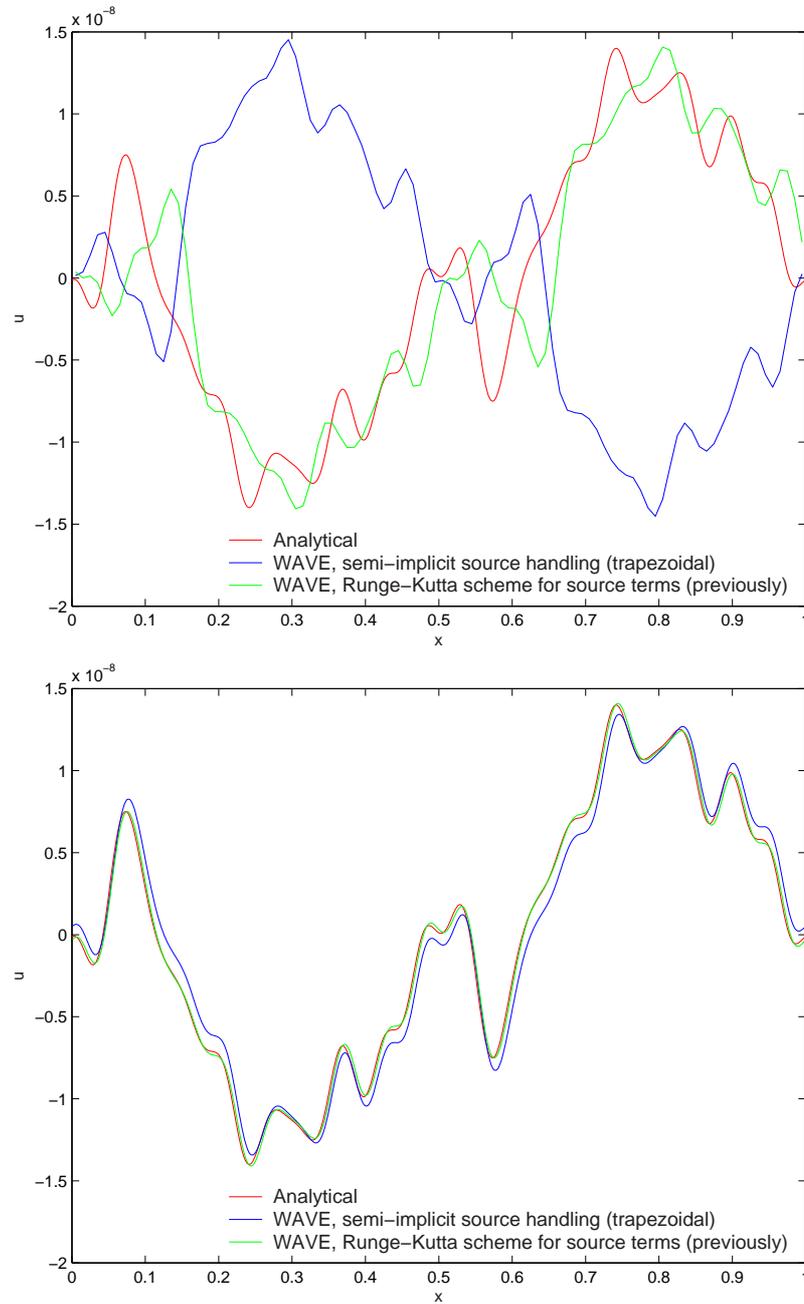


Figure 6.14: Electron Acoustics problem:  $u$  Vs  $x$  for 100 and 512 cells with  $c_s = \sqrt{2}$  and  $\omega_c = 100$  using the semi-implicit source term handling for the wave propagation Algorithm. Upper panel: Semi-implicit source term handling (blue) as compared to the analytical solution (red) and Runge-Kutta scheme source handling implemented previously (green) with 100 cells. Large phase shift noted here; Lower panel: Semi-implicit source term handling (blue) as compared to the analytical solution (red) and Runge-Kutta scheme source handling implemented previously (green) with 512 cells.

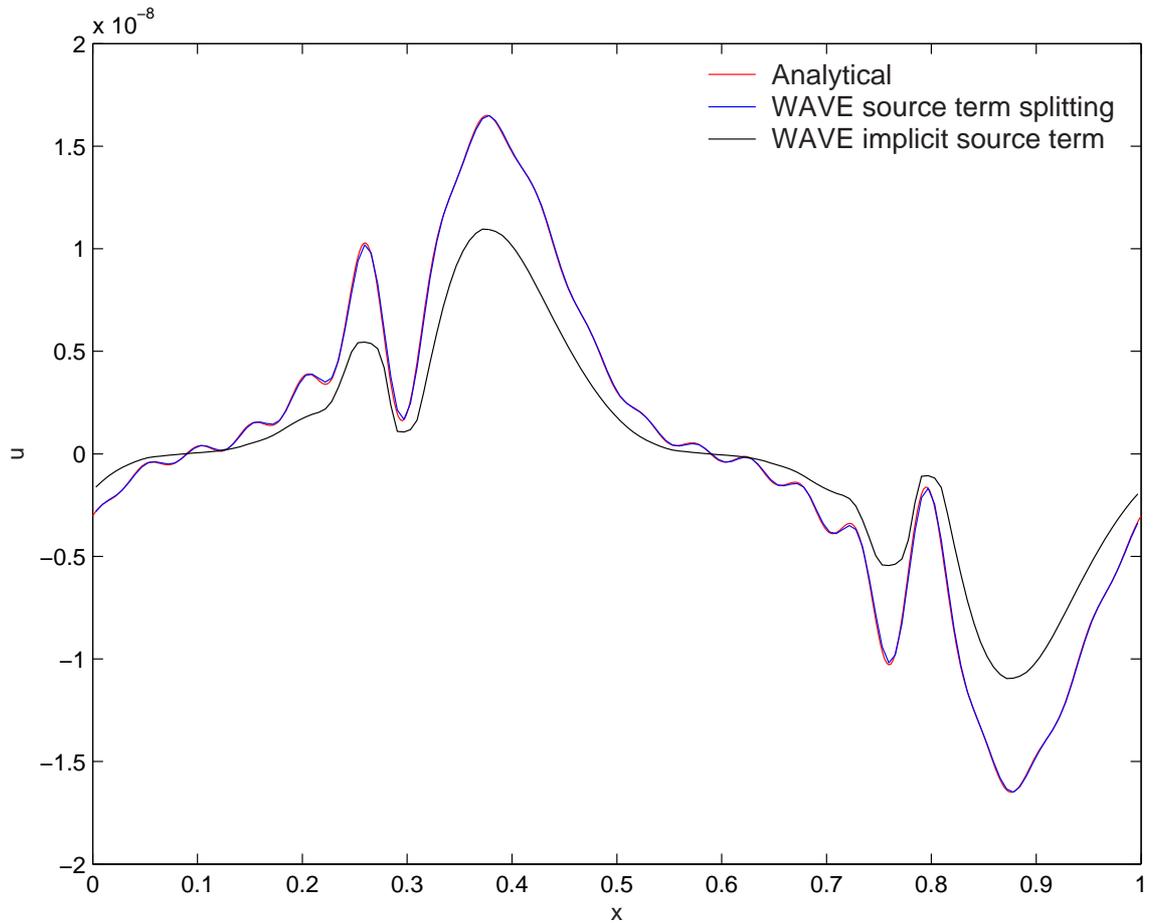


Figure 6.15: Electron Acoustics problem:  $u$  Vs  $x$  for the Wave Propagation method with implicit source term handling (black), and source term splitting (blue) as compared to the analytical solution (red) for 160 cells with  $c_s = \sqrt{2}$  and  $\omega_c = 10$ . It is seen that the implicit source term handling is rather diffusive and it is not preferable to use this.

larger errors. Therefore, the Runge-Kutta time stepping scheme is used for future handling of source terms with the wave propagation method.

Increasing the grid resolution of the Wave Propagation scheme to a little higher than the 512 cells shown in Figure 6.13 can provide results very close to the analytical solution with still less computational effort than running the 3<sup>rd</sup> or even the 2<sup>nd</sup> order RKDG method with just 256 cells. Depending on the problem parameters, i.e. the magnitude of the source terms, the problem could be run accordingly with the 3<sup>rd</sup> order RKDG scheme at a lower grid resolution or the wave propagation scheme at a higher grid resolution.

Exploring the implicit source term handling method for the wave propagation scheme, as discussed in Chapter 2, Figure 6.15 indicates the result of the implicit method. This is an unsplit scheme and it is explored to check if using an unsplit scheme provides a better solution by eliminating the phase errors that occur when the source term splitting is applied. In this case, it is evident that the implicit source term handling method provides a diffusive solution. An analytical evaluation is performed to the implicit equation described in Eq.(2.29) to verify this diffusive nature. A Taylor series expansion is taken and the solution arrived at is described by:

$$\begin{aligned} \mathbf{Q}(t + \Delta t) = & \mathbf{Q}(t) - \Delta t \mathcal{L}(\mathbf{Q}(t)) + \Delta t \mathbf{M} \mathbf{Q}(t) - \Delta t^2 \mathbf{M} \mathcal{L}(\mathbf{Q}(t)) \\ & - \Delta t^2 \mathbf{M}^2 \mathbf{Q}(t) + \Delta t^3 \mathbf{M}^2 \mathcal{L}(\mathbf{Q}(t)) \dots \end{aligned} \quad (6.26)$$

where  $\mathcal{L}$  represents the flux update part of the equation and  $\mathbf{M}$  represents the source Jacobian. It is noted that an  $\mathbf{M}^2$  term is present.  $\mathbf{M}$  contains imaginary eigenvalues for this equation system. So  $\mathbf{M}^2$  contains negative, real values. This indicates a decaying system leading to the diffusive nature of the solution. It is seen that the wave propagation method with implicit source term handling is extremely diffusive, more so than the RKDG method; while the source term splitting introduces phase errors when the source terms are significant. The implicit handling of the source terms is not recommended for this application. The phase error problem is easily fixed by increasing the grid resolution. Therefore, it can be concluded that although source term handling could be an issue with the wave propagation algorithm with certain parameters, this is easily fixed by increasing the grid resolution. Hence, this leaves the Wave Propagation method still numerically and computationally

more effective than the RKDG algorithm.

### 6.3 Radial Z-Pinch

The goal of this thesis is to determine which algorithm is best suited for the application of the full two-fluid plasma model. The two-fluid plasma model contains 16 equations making it quite a complex system. This means a lot of information could be propagating through the domain at a given time and capturing all that could be a challenge. This section involves a comparison of the two algorithms for an application of the Full Two-Fluid Plasma Model - the Radial Z-Pinch Equilibrium problem.

#### 6.3.1 Description of Problem

Chapter 5 describes the equation system and the radial source terms used for this one-dimensional radial Z-pinch problem. The use of Magnetohydrodynamic (MHD) equilibrium is not expected to maintain equilibrium with the two-fluid plasma model; it turns out to be diffusive. Hence, a true two-fluid equilibrium derived by Loverich and Hakim is used. The equilibrium used is defined by

$$\begin{aligned}
 p_e &= j_0^2 \left( \frac{1}{4} r^2 - 12r^4 + \frac{512}{3} r^6 \right) \\
 p_{back} &= p_e \text{ at } r=a \\
 p_0 &= \frac{p_{back}}{1 - \alpha} \\
 p &= p_0 - p_e \\
 P_e &= \frac{p}{2} \\
 j_{ze} &= j_0(1 - 64r^2) \\
 P_i &= \frac{p}{2} \\
 B_\phi &= j_0(0.5r - 16r^3)
 \end{aligned}$$

for the variables within the pinch which in this case is of radius  $a$ . Subscript  $e$  denotes electrons and  $i$  stands for ions.  $\alpha$  determines the amount of initial pressure at the pinch radius,  $p_0$ , and consequently helps set the background pressure to prevent errors due to

negative pressure or density anywhere in the domain.  $j_{ze}$  represents the current in the pinch flowing in the  $z$ -direction. Similarly, the parameters outside the pinch are described by,

$$\begin{aligned}
 p_e &= j_0^2 \left( \frac{1}{4}a^2 - 12a^4 + \frac{512}{3}a^6 \right) \\
 p_0 &= \frac{p_e}{1 - \alpha} \\
 p &= p_0 - p_e \\
 P_e &= \frac{p}{2} \\
 j_{ze} &= 0 \\
 P_i &= \frac{p}{2} \\
 B_\phi &= j_0 (0.5a - 16a^3) a \frac{1}{r}
 \end{aligned}$$

with  $r$  being the position outside the pinch radius. Using the expressions described above and based on the location within the domain, the conserved variables are initialized as:

$$\rho_e = m_e \frac{p}{p_0} \quad (6.27)$$

$$m_{ez} = \frac{j_{ze} m_e}{q_e} \quad (6.28)$$

$$e_e = \frac{P_e}{\gamma - 1} + 0.5 \frac{j_{ze}^2 m_e^2}{q_e^2 \rho_e} \quad (6.29)$$

$$\rho_i = m_i \frac{p}{p_0} \quad (6.30)$$

$$e_i = \frac{P_i}{\gamma - 1} \quad (6.31)$$

$m_{ez}$  is the  $z$ -direction electron momentum and  $B_\phi$  sets the azimuthal magnetic field of the Z-pinch. All other conserved variables are initialized to zero. Figure 6.16 shows plots of the initial conditions chosen.

Axis boundary conditions are used on the left edge and conducting wall boundaries are assumed on the right edge. In applying the boundary conditions, care needs to be taken to handle the higher order coefficients of the RKDG appropriately. Because it is a polynomial approximation, treating the higher order coefficients at the boundaries in exactly the same manner as the first coefficient could lead to incorrect solutions. Therefore, the higher order

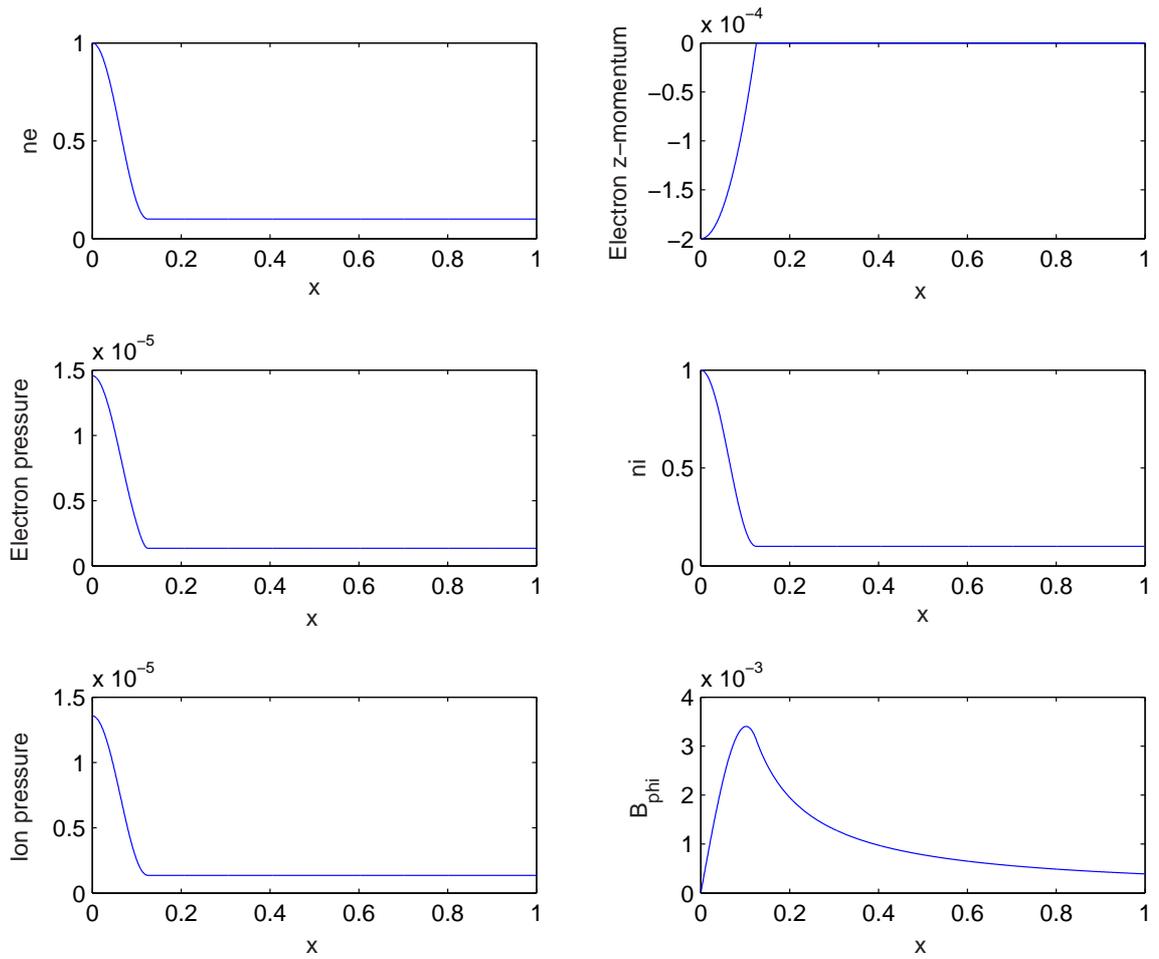


Figure 6.16: Z-pinch Two Fluid Equilibrium initial conditions for electron density, z-direction momentum and pressure; ion density and pressure; and azimuthal magnetic field.

terms are treated such that all the odd order coefficients have a flipped sign from the even order coefficients i.e.,

Scalar and Axial terms:

$$\mathbf{Q}(\text{ghost cell}) = \mathbf{Q}(\text{boundary}) \quad \text{for even order coefficients} \quad (6.32)$$

$$\mathbf{Q}(\text{ghost cell}) = -\mathbf{Q}(\text{boundary}) \quad \text{for odd order coefficients} \quad (6.33)$$

Radial and Azimuthal terms:

$$\mathbf{Q}(\text{ghost cell}) = -\mathbf{Q}(\text{boundary}) \quad \text{for even order coefficients} \quad (6.34)$$

$$\mathbf{Q}(\text{ghost cell}) = \mathbf{Q}(\text{boundary}) \quad \text{for odd order coefficients,} \quad (6.35)$$

where  $\mathbf{Q}(\text{ghost cell})$  refers to the ghost cell values while  $\mathbf{Q}(\text{boundary})$  refers to the value at the axial edge of the domain. This is how Legendre polynomials are to be treated at the boundaries. Similar implementation is done for the conducting wall boundary on the right edge of the domain. This boundary requires that the fluid normal velocities, the radial magnetic field and the tangential electric fields be negated while the rest of the variables are copied over. With these initializations and boundary conditions, the comparisons of both the algorithms are detailed in the following section to determine which algorithm is better suited for this equation system and which one holds equilibrium better.

### 6.3.2 Results

The wave propagation scheme is compared to the 2<sup>nd</sup> and 3<sup>rd</sup> order RKDG method with characteristics-based and component-based limiting. It is expected as in the previous problems that the characteristics-based limiters dominate over the component-based ones for the RKDG scheme. This will be shown in the following paragraphs once the  $l_2$ -norm is computed. Therefore, only the RKDG characteristics-based limiters are compared here. Figure 6.17 shows the Wave and RKDG algorithms with a grid resolution of 32 cells as compared to the initial condition. This figure compares the algorithms based on the electron density and the azimuthal magnetic fields. The results shown with these two variables are sufficient for the comparisons as the other conserved variables vary in a similar manner. The initial conditions chosen for purposes of the comparisons, and to compute the errors,

are shown in Figure 6.16 with 5000 cells. The algorithms are compared for their abilities to maintain equilibrium. The results shown are for simulations run until a characteristic transit time of  $t = 10$  on a domain going from  $x = 0$  to  $x = 1 = 7r_{Li}$  with the left being the axis and the right being the conducting wall. The parameters used here are  $\gamma = 1.6667$ , speed of light  $c_0 = 1.0$ , ion and electron charge-to-mass ratios of  $q_i/m_i = 10$ ,  $q_e/m_e = 500$ , ion-to-electron mass ratio of  $m_i/m_e = 50$ , ion Larmor radius-to-domain length of  $r_{Li}/x_0 = 1/7$ , and ion skin depth-to-domain length of  $\delta_i/x_0 = 1/10$ .

It is evident from Figure 6.17 that at a low grid resolution of 32 cells, the wave propagation scheme is rather diffusive, while the RKDG scheme performs really well at holding equilibrium. The reason for this diffusive behavior of the wave propagation scheme could be attributed to the slower characteristic speeds in the system getting diffused away. Increasing the CFL number of the wave propagation scheme from 0.9 to 1.0 leads to oscillations in the axis that destroy the solution. Decreasing the CFL number even lower makes the solution even more diffusive and this is not desired. The RKDG, being an unsplit scheme, seems to be superior. It appears that the  $2^{nd}$  and  $3^{rd}$  order RKDG schemes do not differ significantly as a result of which it can be concluded that the use of the  $2^{nd}$  order RKDG scheme is the best choice at this resolution based on the closeness of the solution to the initial condition and the computational effort involved. At a grid resolution of 100, the algorithms still differ from each other rather significantly. The wave propagation algorithm is diffusive even at this resolution as compared to the  $2^{nd}$  and  $3^{rd}$  order RKDG schemes. Even the RKDG scheme differs slightly from the initial condition at this resolution. This can be seen from Figure 6.18. Increasing the grid resolution to 256 provides the results shown in Figure 6.19. With 256 cells, all the algorithms appear to have a solution pretty close to the initial equilibrium condition. Figure 6.20 provides a comparison of all the algorithms based on the  $l_2$ -norm described previously. For this problem, the  $l_2$ -norm is computed for the magnetic field,  $B_{phi}$ , at numerous grid resolutions. It is noted that the  $3^{rd}$  order RKDG scheme performs the best of all the algorithms but not significantly better than the  $2^{nd}$  order RKDG method. For low resolutions, the wave propagation method performs poorly and in fact, it continues to perform poorly until the resolution is about 200 cells. The accuracy of the wave propagation scheme with 200 cells is comparable to that of the RKDG

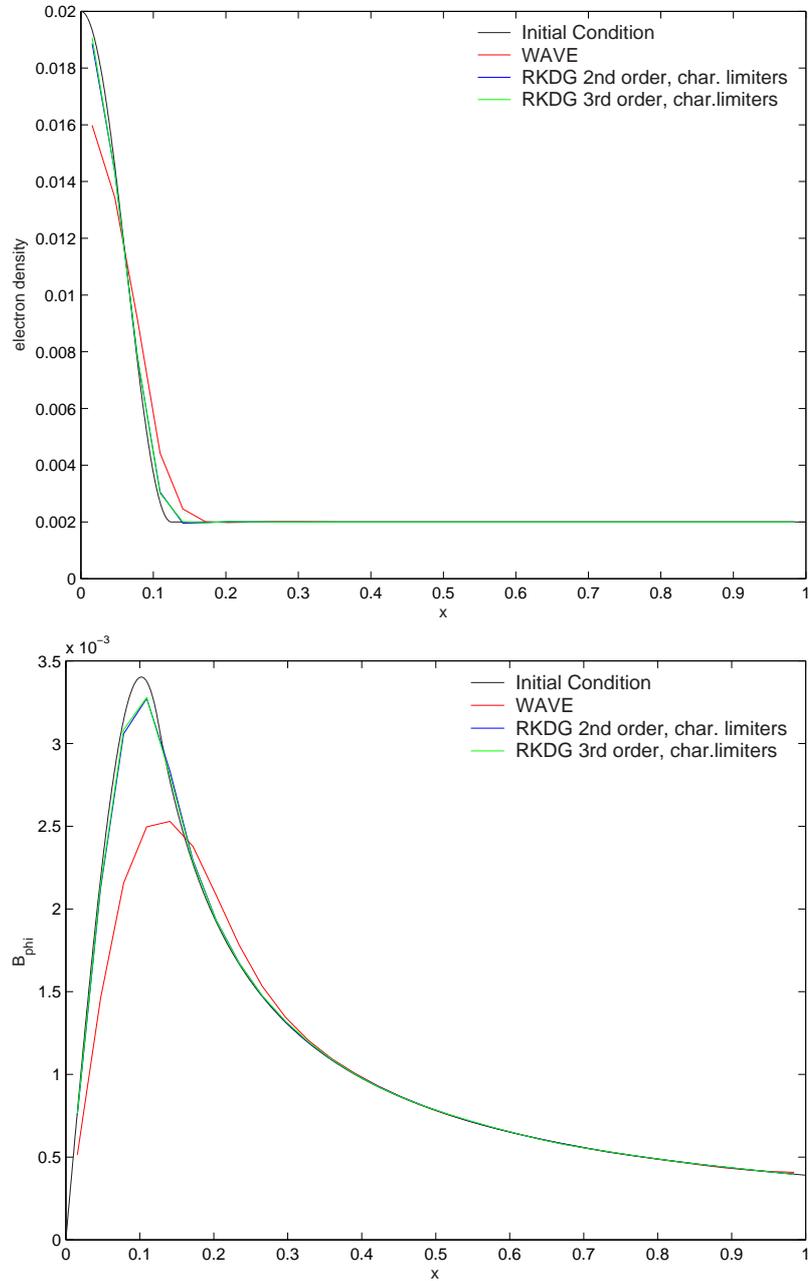


Figure 6.17: Z-pinch Two Fluid Equilibrium with 32 cells for the Wave Propagation algorithm (red), the 2<sup>nd</sup> order RKDG scheme with characteristic-based limiting (blue) and the 3<sup>rd</sup> order RKDG scheme with characteristic-based limiting (green) as compared to the initial condition (black) for the electron density (upper panel), and azimuthal magnetic field (lower panel).

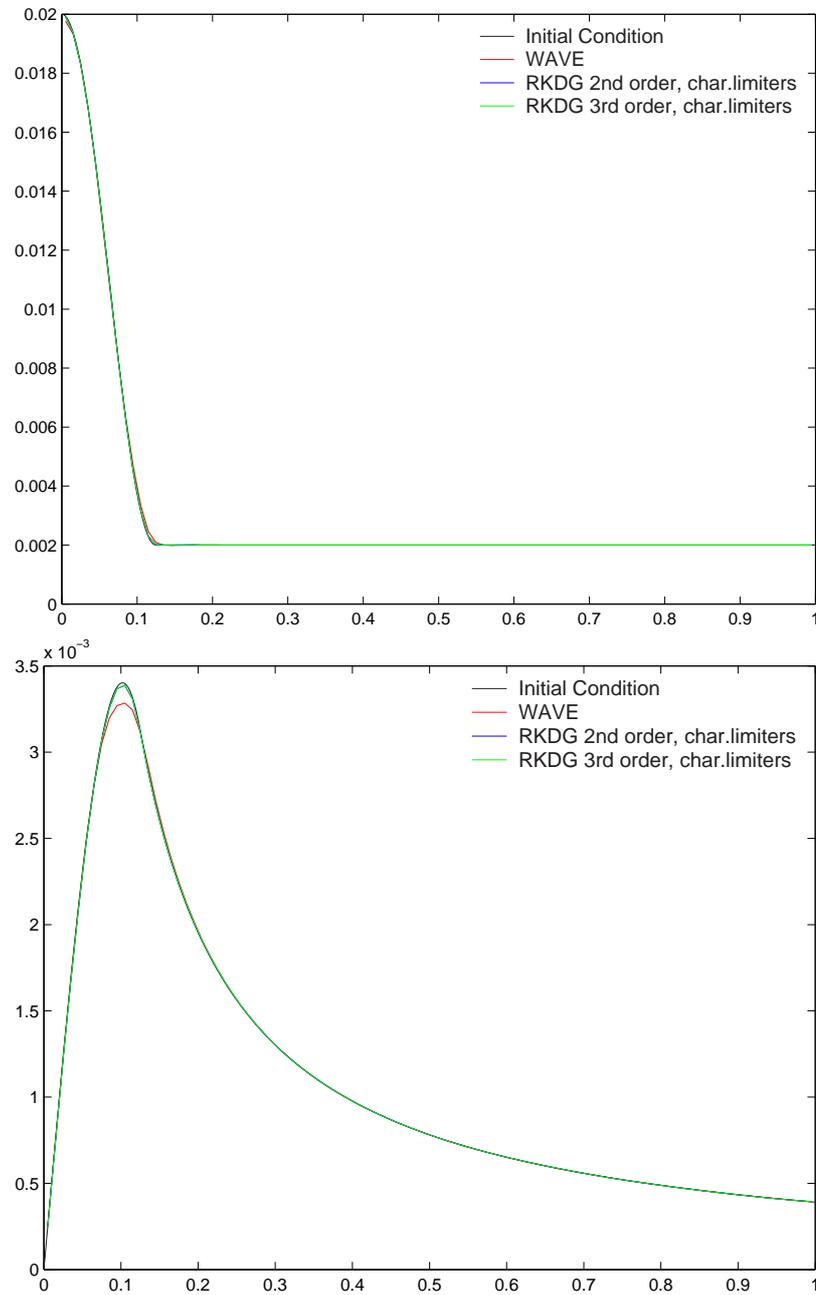


Figure 6.18: Z-pinch Two Fluid Equilibrium with 100 cells for the Wave Propagation algorithm (red), the 2<sup>nd</sup> order RKDG scheme with characteristic-based limiting (blue) and the 3<sup>rd</sup> order RKDG scheme with characteristic-based limiting (green) as compared to the initial condition (black) for the electron density (upper panel), and azimuthal magnetic field (lower panel).

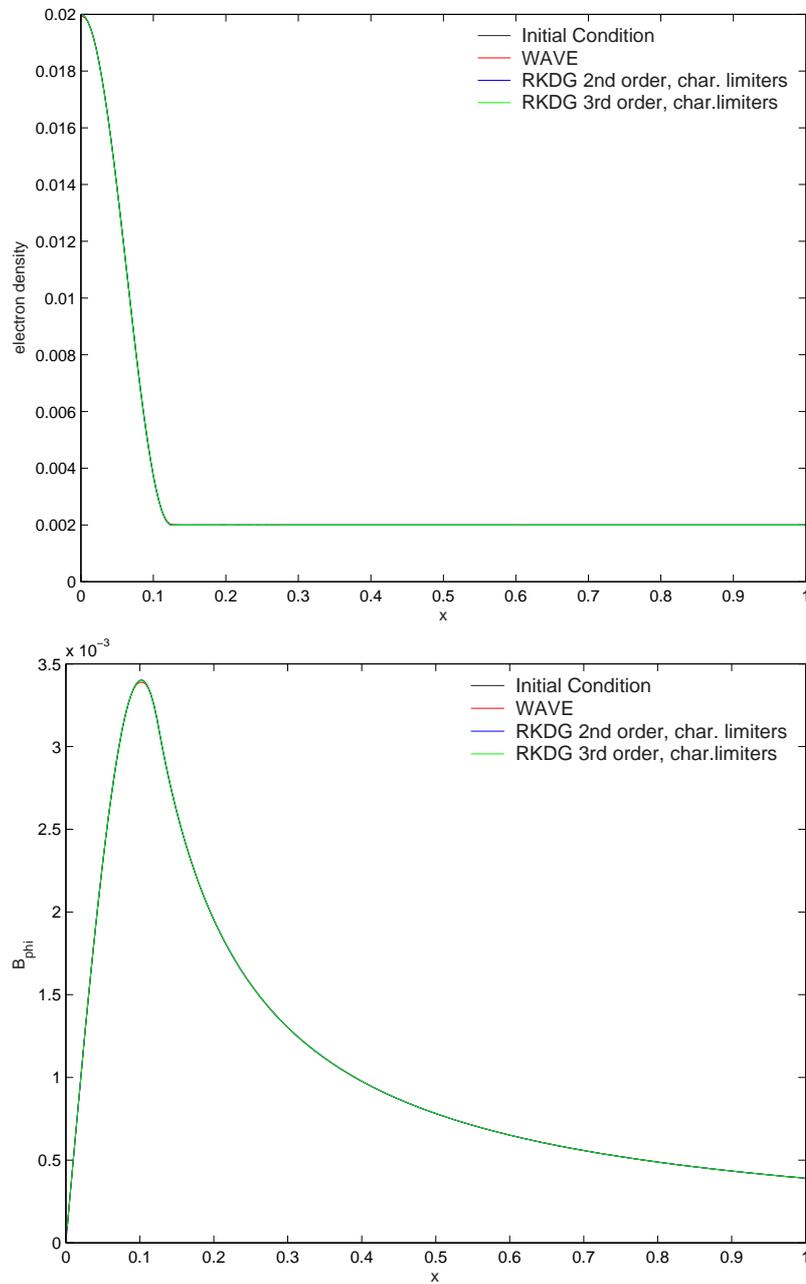


Figure 6.19: Z-pinch Two Fluid Equilibrium with 256 cells for the Wave Propagation algorithm (red), the 2<sup>nd</sup> order RKDG scheme with characteristic-based limiting (blue) and the 3<sup>rd</sup> order RKDG scheme with characteristic-based limiting (green) as compared to the initial condition (black) for the electron density (upper panel), and azimuthal magnetic field (lower panel).

schemes (both  $2^{nd}$  and  $3^{rd}$  order) with only 64 cells. A look at the computational efforts will determine which algorithm is superior. Table 6.3 displays the CPU times for each of the algorithms using several grid resolutions and this is better seen in Figure 6.21. Comparing the computational time of the wave propagation scheme at 200 cells to the  $2^{nd}$  order RKDG method with characteristics-based limiting at 64 cells, it is seen that the RKDG method takes less than half the computational effort to provide a solution with the same accuracy. Therefore, for these chosen parameters, the RKDG algorithm provides better results for the two-fluid plasma equation system.

Upon testing both the algorithms with several different values for  $q_e$  and  $q_i$  results in the comparisons being consistent to that presented with  $q_i = -q_e = 10$  in terms of both accuracy and computational effort. Increasing the electron mass,  $m_e$  to  $1/25$  and comparing the results from the two algorithm leads to the consistent comparisons as well. It does not appear to take longer to run the simulation to the same characteristic transit time,  $t_{end} = 10$ . In fact it is noted that the solutions (especially the RKDG) are so stable that it would take an incredibly long time to develop physical instabilities in this equilibrium. Both the algorithms are initialized with the same equilibrium and are run out to a very long time to see which one performs better at holding equilibrium. Figure 6.22 displays the result of running the wave propagation and the  $2^{nd}$  order RKDG scheme with 128 grid points to time  $t_{end} = 500$ . It is evident that, at the same grid resolution, the RKDG method is superior to the wave propagation algorithm at maintaining equilibrium. In fact the RKDG method holds equilibrium almost exactly as compared to the initial conditions while the wave propagation method is very diffusive. A good comparison would involve running the Wave Propagation and the  $2^{nd}$  order RKDG schemes with the same computational effort and comparing the results obtained for both of them in this case. To do this, both algorithms are run out to the same final time,  $t = 500$  but the wave propagation algorithm is run with 256 cells while the RKDG is run with 128. The results of these comparisons are shown in Figure 6.23 for the electron density and azimuthal magnetic field plots. At these resolutions, the RKDG method requires 11000s of CPU time and the Wave Propagation method requires 10000s. Comparing the equilibria, it is concluded that the RKDG method performs better for purposes of maintaining equilibrium using the two-fluid plasma system.

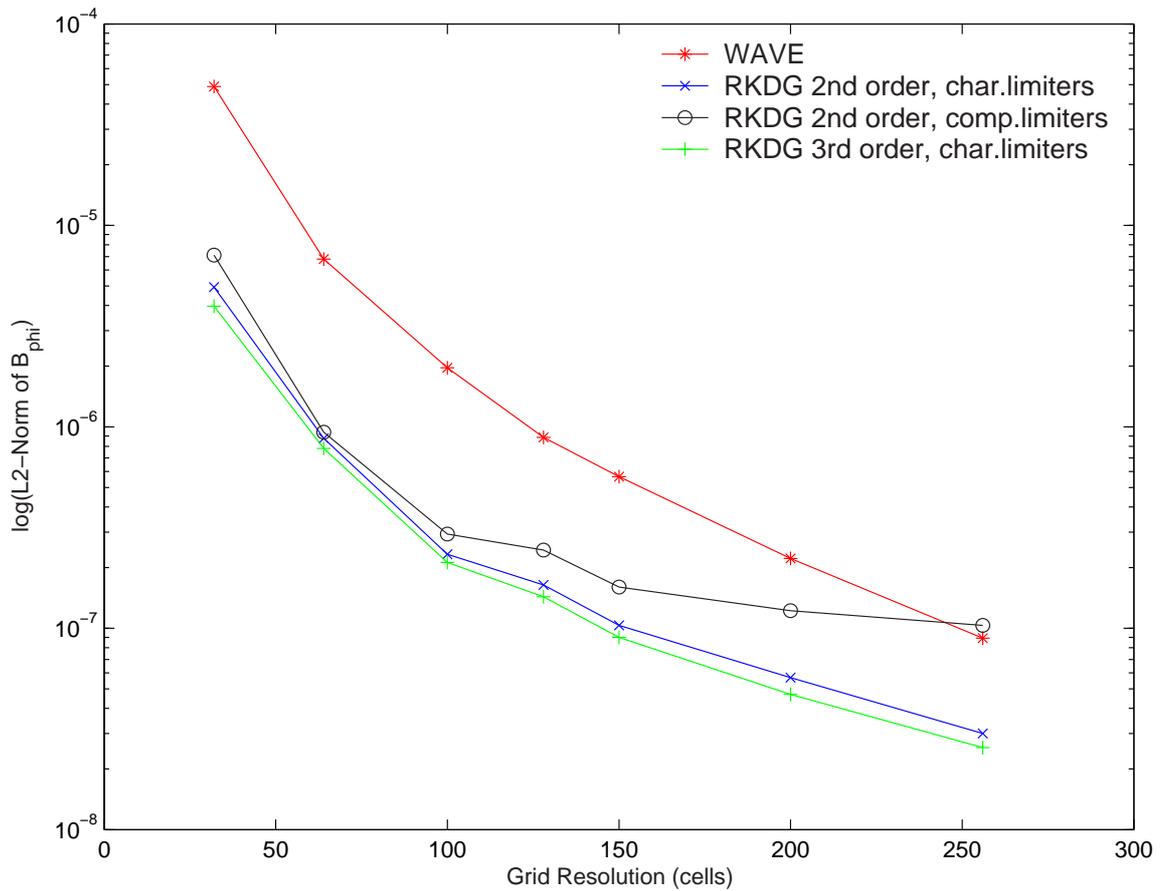


Figure 6.20: Radial Z-pinch problem:  $l_2$ -norm of  $B_{\phi}$  Vs Grid Resolution for the wave propagation method (red),  $2^{nd}$  order RKDG with characteristics-based limiters (blue),  $2^{nd}$  order RKDG with component-based limiters (black), and  $3^{rd}$  order RKDG with characteristics-based limiters (green) as compared to the initial condition with 5000 cells with  $\gamma = 1.6667$ ,  $c_0 = 1.0$ ,  $q_i = -q_e = 10$ ,  $m_i = 1.0$  and  $m_e = 1/50$ . It is seen that the RKDG algorithm is far superior to the wave propagation scheme in terms of accuracy.

Algorithm	CFL	Order	Time(sec)
J=256 cells			
WAVE	0.9	2	200
RKDG with characteristic limiters	0.3	2	925
RKDG with component-based limiters	0.3	2	1003
RKDG with characteristic limiters	0.2	3	
J=128 cells			
WAVE	0.9	2	50.5
RKDG with characteristic limiters	0.3	2	232.8
RKDG with component-based limiters	0.3	2	254.6
RKDG with characteristic limiters	0.2	3	538.4
J=64 cells			
WAVE	0.9	2	12.9
RKDG with characteristic limiters	0.3	2	59.7
RKDG with component-based limiters	0.3	2	64.6
RKDG with characteristic limiters	0.2	3	137.1
J=32 cells			
WAVE	0.9	2	3.35
RKDG with characteristic limiters	0.3	2	15.5
RKDG with component-based limiters	0.3	2	16.8
RKDG with characteristic limiters	0.2	3	36.1

Table 6.3: Two-Fluid Plasma Model Radial Z-Pinch problem: Some computational times for the wave propagation algorithm and the RKDG algorithm with both characteristic-based and component-based limiters for several grid resolutions.

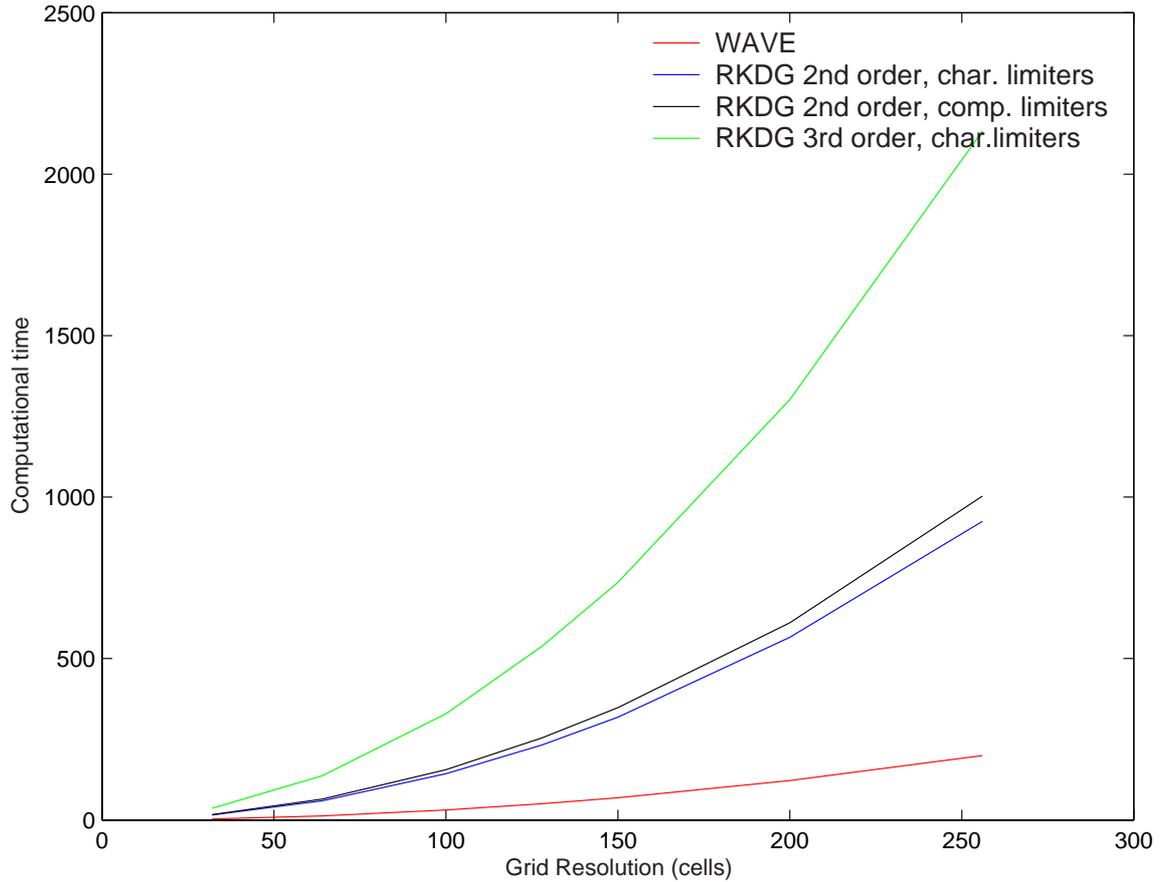


Figure 6.21: Radial Z-Pinch problem: Computational time(sec) Vs Grid Resolution for the wave propagation method (red),  $2^{nd}$  order RKDG with characteristics-based limiters (blue), and  $3^{rd}$  order RKDG with characteristics-based limiters (green) for  $\gamma = 1.6667$ ,  $c_0 = 1.0$ ,  $q_i = -q_e = 10$ ,  $m_i = 1.0$  and  $m_e = 1/50$ . It is noted that the  $3^{rd}$  order RKDG method takes much higher computational effort while the wave propagation algorithm takes very little.

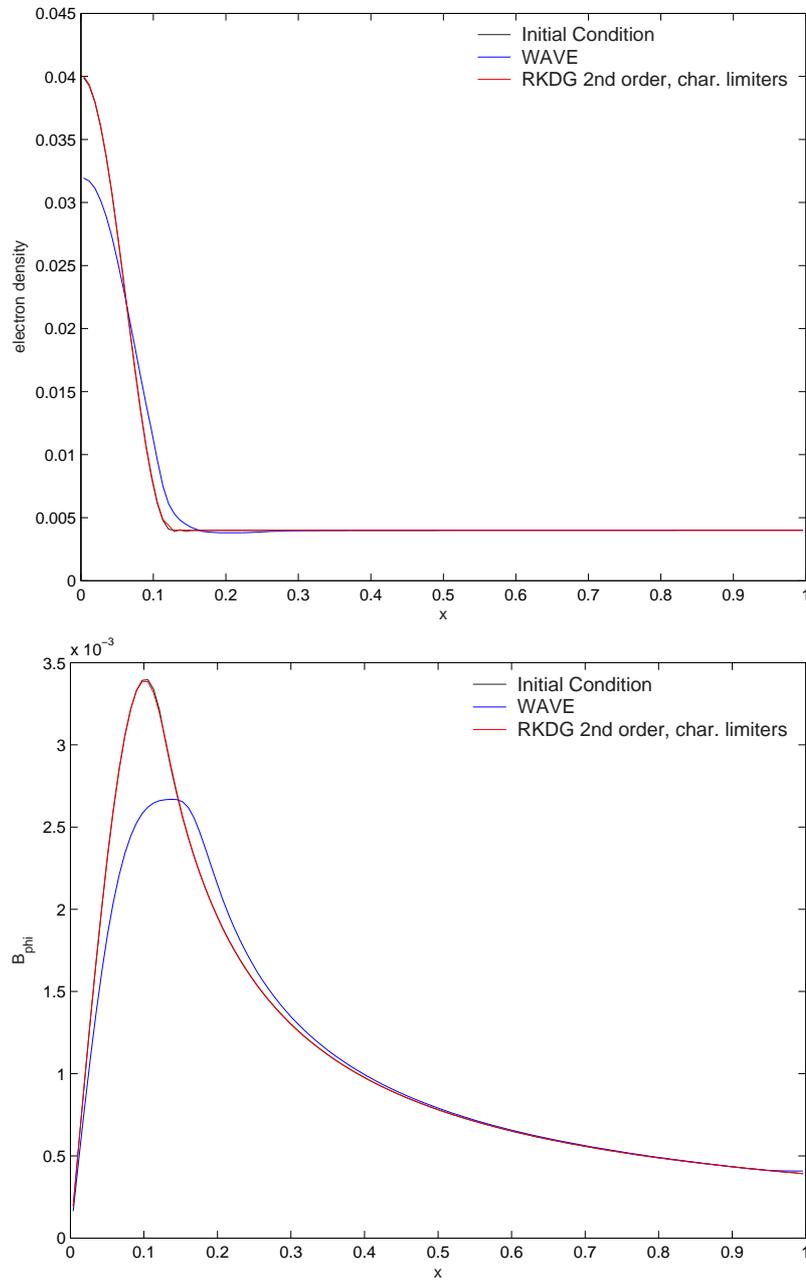


Figure 6.22: Z-pinch Two Fluid equilibrium with 128 cells for the Wave Propagation algorithm (red), the 2<sup>nd</sup> order RKDG scheme with characteristic-based limiting (blue) run out to a long time  $t_{end} = 500$  as compared to the initial condition (black) for the electron density (upper panel), and azimuthal magnetic field (lower panel).

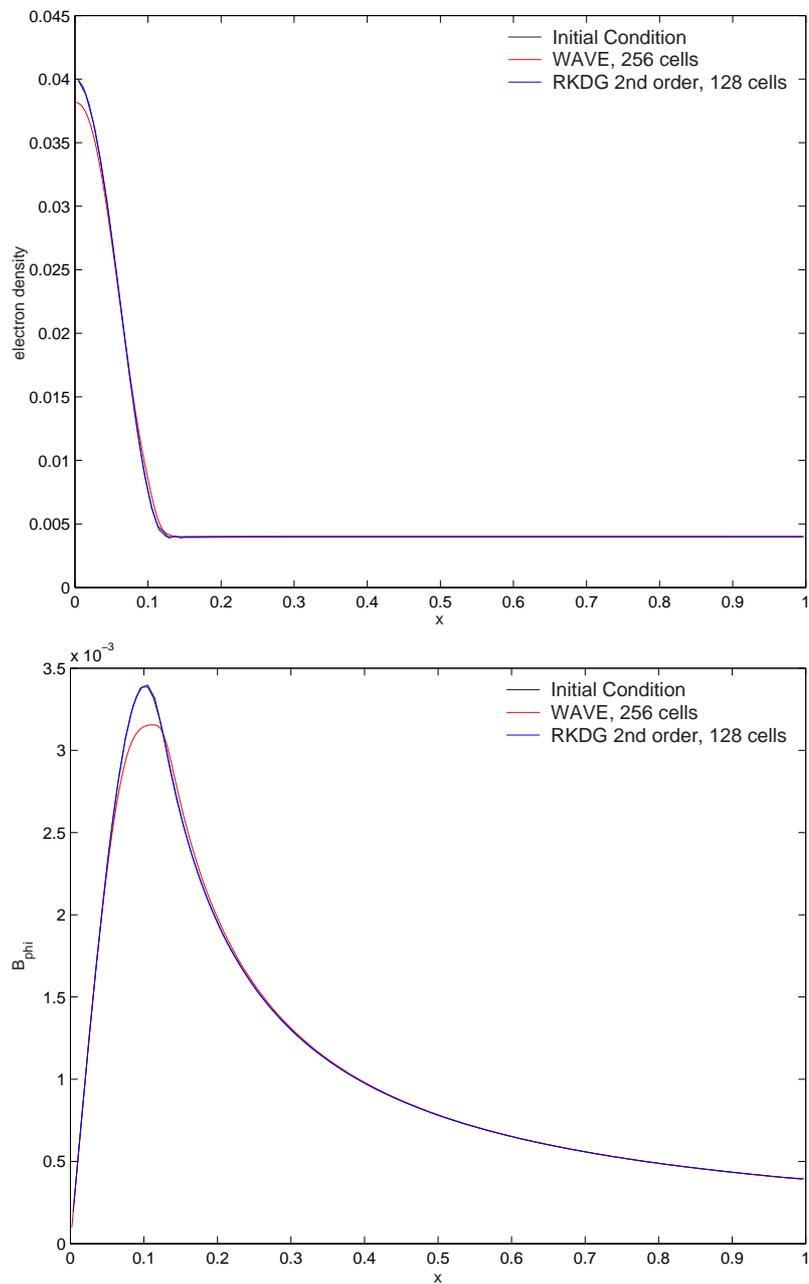


Figure 6.23: Z-pinch Two Fluid equilibrium with 256 cells for the Wave Propagation algorithm (red) and 128 cells for the 2<sup>nd</sup> order RKDG scheme with characteristic-based limiting (blue) run out to a long time  $tend = 500$  as compared to the initial condition (black) for the electron density (upper panel), and azimuthal magnetic field (lower panel). At these chosen resolutions, both algorithms have comparable computational efforts.

The full two-fluid plasma model has imaginary eigenvalues for the source terms. Performing a source splitting for the wave propagation scheme is the reason the equilibrium decays over time. Bale, LeVeque, Rossmanith and Mitran [1] study this problem and discuss the treatment of the source terms. They determine that the source term handling through the process of Strang splitting that has been applied to the wave propagation method might not work well for solutions close to a steady-state. An alternate solution is provided according to which the algorithm takes into account only those waves that contain the information propagating relative to the steady-state. There is a process of distribution of the eigen-decomposition of the source terms into the neighboring cells based on the sign of the corresponding eigenvalues. This, however, is not applicable to the case of the two-fluid plasma model. The source Jacobian for this equation system has all imaginary eigenvalues, so this unsplit scheme is not suitable here. Finding a suitable unsplit scheme for this equation system is a good topic for future study.

The wave propagation scheme is superior for equation systems with one characteristic speed. The equilibrium problem has been applied here to the two-fluid plasma model. For the case of dynamic problems, however, when there are many speeds present, such as with the full two-fluid plasma model, the wave propagation scheme might not accurately capture the information propagating with speeds much lower than the fastest characteristic speed. The slower speeds are diffused away while the faster ones are well captured.

The wave propagation scheme captures the solution almost exactly when it is run with CFL=1 and there is only one characteristic speed in the system as with the case of both the strain wave and the dispersive Euler problems. For the two-fluid model, however, there are many characteristic speeds in the system and running the problem with CFL=1 causes the slower speeds to get diffused away because the CFL=1 requirement cannot be maintained for all the characteristic speeds in the system. The RKDG algorithm appears to be rather stable and accurate for this equation system. It captures the information in the system reasonably well even at lower grid resolutions. Therefore, it is possible to conclude that for radial Z-pinch equilibrium with the two-fluid plasma model, the RKDG algorithm is superior in maintaining equilibrium in terms of accuracy and computational effort.

A study of dynamic applications of the full two-fluid plasma model would be an im-

portant topic for further study as well. The drawbacks of the wave propagation algorithm for steady-state problems might not appear in dynamic problems. The use of split source schemes for the wave propagation algorithm might work better for dynamic problems. Also, with information propagating within the domain, the algorithms might compare in a totally different light versus steady-state problems.

## Chapter 7

## CONCLUSIONS

**7.1 Concluding Remarks**

The wave propagation method and the Runge-Kutta Discontinuous Galerkin (RKDG) method are compared. Applications of these methods to equation systems containing only one characteristic speed such as the Strain Wave and the *Dispersive* Euler models indicate that the wave propagation algorithm is superior in terms of both desired accuracy and computational expense. When the wave propagation method is run with a Courant number of 1 for such simpler models that have only one characteristic speed of disturbance propagation, the solution obtained is very close to the analytical or converged solution without significant dispersive or diffusive errors. However, the wave propagation algorithm is subject to phase errors when tested with the electron acoustic pulse problem using the Dispersive Euler equation set. This is because the parameters chosen, specifically the cyclotron frequencies, were too high for the grid resolutions and the time for information to propagate through the domain was not sufficient to be able to resolve the oscillation frequency. The wave propagation algorithm applies source term splitting and this is the main reason that these phase errors occur. These phase errors are also observed with the semi-implicit source term handling method using the trapezoidal method instead of the Runge-Kutta time stepping scheme. With the semi-implicit scheme, the errors are more severe and the computational effort is slightly higher. Upon applying implicit source term handling to the wave propagation scheme it is seen that the phase errors are avoided but the solution now is extremely diffusive even at higher grid resolutions. Hence, the use of both the semi-implicit and the implicit source term handling methods could not be justified over just a higher grid resolution Runge-Kutta source term splitting based on both the accuracy and the computational effort involved. These phase errors do not occur with the RKDG method which is an un-split scheme and this would be preferable so that accurate solutions are obtained. However,

increasing the grid resolution resolves these issues for the wave propagation method and provides a better solution than the  $2^{nd}$  order RKDG method which appears to be rather diffusive even at high grid resolutions. The  $3^{rd}$  order RKDG method captures the solution rather accurately but it is very computationally intensive. Increasing the grid resolution of the wave propagation method to overcome these errors takes less computational effort than the RKDG method with lower grid resolution and provides a better solution.

For the Two-Fluid Plasma Model however, there is a notable difference in the performance of the algorithms as compared to their applications to simpler equation systems. The two-fluid equation system contains 16 equations and it is a rather complicated system with information propagating at a number of different speeds. The speed of sound and the speed of light play significant roles in resolving the information in the system and the large difference between these speeds requires an appropriate algorithm to capture the information accurately. For this model, the RKDG method provides a more accurate, well resolved and computationally efficient solution. The wave propagation method is rather diffusive at lower grid resolutions for this model and does not do a good job at maintaining equilibrium in the radial Z-pinch. Near equilibrium the speed of information propagating is much less than the characteristic speed. The Courant number chosen is applied to the fastest characteristic wave speed to ensure stability. As a result, the waves traveling at slower characteristic speeds get diffused away since the Courant condition cannot be maintained for information propagating at all different speeds, and this leads to a diffusive solution. In addition, this diffusive behavior of the wave propagation method could be attributed to the source term handling (the use of split source terms) which could introduce errors in the solution. This is a problem with equilibrium applications and has been studied by Bale, LeVeque, Rossmanith and Mitran in [1]. For problems close to steady state, the hyperbolic fluxes should balance the electromagnetic source terms because it is the discrepancy between these terms that is decomposed into propagating waves. Using Strang splitting for the source terms could introduce errors in the solution. These errors might upset the balance between the source terms and the hyperbolic flux gradients, causing the solution to become diffusive over time. There are unsplit schemes available to resolve these source term splitting issues, but they are not applicable to the two-fluid plasma model because the source Jacobian in

this equation system has imaginary eigenvalues. The RKDG method performs well and it is sufficient to use the  $2^{nd}$  order method which provides accurate and computationally effective results. Hence, to maintain equilibrium in a radial Z-pinch using the full two-fluid plasma model, the RKDG method is the superior of the two algorithms.

## 7.2 Further Study

There is potential for further study that could involve modifying the algorithms to make them more competitive. For the wave propagation scheme, it is possible to use a combined method with implicit source term handling and source term splitting such that they can be applied independently in the domain based on the times for the wave propagation and the oscillation frequency. In this manner the phase errors can be eliminated and there would be proper sampling of all the frequencies in the domain.

It is also possible to extend these comparisons to multi-dimensions and see if the performance of the algorithms is consistent in 2D and 3D domains. Loverich and Hakim have already implemented multiple dimension applications of the discontinuous Galerkin [10] and the wave propagation [4] methods for the Two-Fluid Plasma Model. Performing the comparisons in multi-dimensions could give a good idea of how the algorithms compare with effects of transverse propagation.

Implementing vacuum boundaries is yet another area to be explored to allow for realistic boundary conditions and problem specifications without worrying about negative pressures and densities in the domain. Also, the relaxation methods detailed by LeVeque and Pelanti in [8] could be further studied for applicability to the full two-fluid plasma model. It would require additional equations and might seem to make the system rather complicated initially, but it could result in better solutions.

To verify the comparisons of the Z-pinch equilibrium problem, it would be useful to perform the comparisons with a simpler equation system equilibrium problem. An equilibrium can be derived for the Dispersive Euler equation system and the algorithm comparisons can be applied here as well to make sure the results are consistent. Dynamic applications of the two-fluid plasma model would make a good study for these comparisons as well because the performance of the algorithms in equilibrium might not be consistent with their perfor-

mance in dynamics problems. Lastly, the unsplit schemes for the wave propagation method could be studied in greater detail to see if there is a possible way to apply them to a system where the source Jacobian has imaginary eigenvalues. Doing this could be the solution to resolving the diffusive nature of the wave propagation algorithm for equilibrium problems.

## BIBLIOGRAPHY

- [1] D. Bale, R. J. LeVeque, S. Mitran, and J. A. Rossmannith. A wave-propagation method for conservation laws and balance laws with spatially varying flux functions. *SIAM Journal of Scientific Computing*, 24:955–978, 2002.
- [2] B. Cockburn and C. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173–261, 2001.
- [3] T.R. Fogarty and R.J. LeVeque. High-resolution finite-volume methods for acoustic waves in periodic or random media. *Journal of Acoustical Society of America*, 106(1):17–28, 1999.
- [4] A. Hakim, J. Loverich, and U. Shumlak. A high resolution wave propagation scheme for ideal Two-Fluid plasma equations. *Journal of Computational Physics*, 2006. Submitted for publication.
- [5] A.G. Kulikovskii, N.V. Pogorelov, and A.Y. Semenov. *Mathematical Aspects of Numerical Solutions of Hyperbolic Systems*. Chapman and Hall/CRC, 2001.
- [6] J.O. Langseth and R.J. LeVeque. A wave propagation method for three-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 165:126–166, 2000.
- [7] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [8] R.J. LeVeque and M. Pelanti. A class of approximate Riemann solvers and their relation to relaxation schemes. *Journal of Computational Physics*, 172:572–591, 2001.
- [9] R.J. LeVeque and D.H. Yong. Solitary waves in layered nonlinear media. *SIAM Journal of Applied Math*, 63:1539–1560, 2002.
- [10] J. Loverich, A. Hakim, and U. Shumlak. A discontinuous Galerkin method for ideal two-fluid plasma equations. *Journal of Computational Physics*, submitted.
- [11] P. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.