Development of Electromagnetic Solvers for Use with The Two-Fluid Plasma Algorithm

Andree Susanto

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

Master of Science in Aeronautics and Astronautics

University of Washington

2009

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

Andree Susanto

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

Committee Members:

Uri Shumlak

Brian Nelson

Date:

Signature____________________________

Date_______________________________

University of Washington

Abstract

Development of Electromagnetic Solvers for Use with
The Two-Fluid Plasma Algorithm

Andree Susanto

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

The two-fluid plasma model describes a plasma as a group of ions and electrons. The plasma fluid is modelled using the Euler equations of gas dynamics, reacting to source terms produced by the electromagnetic fields. Two techniques to advance the electromagnetic fields are then proposed. The first one is the perfectly hyperbolic Maxwell's equations. This technique is convenient, because it preserves the hyperbolic structure of the five-moment two-fluid plasma model. The second one uses potential formulation of the electric and magnetic fields. The potential formulation is a second order system of wave equations, which means that it has to be solved outside the main equation system. In this paper, results are presented and the two techniques are compared for stability, accuracy and divergence error.

TABLE OF CONTENTS

Page

List of Figures . . . iii

List of Tables . . . v

Chapter 1: Introduction . . . 1
- 1.1 Introductory Remarks . . . 1
- 1.2 Software Used . . . 1

Chapter 2: Two Fluid System for Plasma . . . 3
- 2.1 Euler Equations to Model Plasma Fluid . . . 3
- 2.2 Electromagnetic Models . . . 4
  - 2.2.1 Perfectly Hyperbolic Maxwell's Equations . . . 5
  - 2.2.2 Electromagnetic Potential Formulation . . . 6

Chapter 3: Numerical Methods . . . 8
- 3.1 Wave Propagation Method . . . 8
  - 3.1.1 First Order Godunov Update . . . 8
  - 3.1.2 Second-Order Corrections . . . 10
  - 3.1.3 Transverse Corrections . . . 11
- 3.2 Laplacian Operator . . . 12
  - 3.2.1 WARPX Implementation of the Laplacian Operator . . . 13
- 3.3 Curl Operator . . . 18
- 3.4 Divergence Operator . . . 19
  - 3.4.1 Divergence of a Curl in a Uniform Cartesian Coordinates . . . 19
- 3.5 Gradient Operator . . . 21
- 3.6 Poisson Solver . . . 21
  - 3.6.1 WARPX Implementation of the Poisson's Solver . . . 21

Chapter 4: Electromagnetic Source Terms Test Cases . . . 24
- 4.1 Grid Definition . . . 24

4.2 Source Terms Handling 26
4.3 Open Boundary Conditions 26
4.3.1 Copy Boundary Conditions 27
4.3.2 Second Order Mur Radiation Boundary Conditions 27
4.4 Current Source Terms 27
4.4.1 Divergence Error Comparison for the Current Problem 33
4.5 Steady State Charge Density Source Term Test 34
4.5.1 Divergence Error Comparison for the Steady State Charge Density Test Case 38
4.6 Time-Dependent Linearly Growing Charge Density Source Term Test Case . 48
4.6.1 Divergence Error Comparison for the Time Dependent Charge Problem 51
4.7 The Effects of Grid distortion on the Solution 59
4.7.1 Grid Distortion on Divergence Error 59
4.8 Conclusions For the Electromagnetic Source Term Testing 64

Chapter 5: Preliminary Plasma Test Cases in Cartesian Grid 67
5.1 Implementation of Potential Solvers into the Two-Fluid Plasma System . . . 67
5.1.1 Lorenz Gauge Implementation 68
5.1.2 Coulomb Gauge Implementation 68
5.2 1D Two-Fluid Plasma Shock Problem 69
5.2.1 Divergence Error Comparison for Two-Fluid Plasma Shock Problem . 70
5.3 Magnetic Reconnection 75
5.3.1 Divergence Error Comparison for Magnetic Reconnection Problem . . 79
5.4 Conclusion of the Cartesian Plasma Simulation Test Cases 82

Chapter 6: Conclusions and Possible Future Work 92
6.1 Concluding Remarks 92
6.2 Possible Future Work 93

Bibliography 95

LIST OF FIGURES

Figure Number Page

3.1 9 Point Trapezoidal Stencil 14

3.2 9 Point Uniform Cartesian Stencil 20

4.1 Distorted grid for $\alpha = 4$ 25

4.2 Current Test Case 29

4.3 B_x on skewed grid PHM 30

4.4 B_x on skewed grid potential 31

4.5 B_x on skewed grid steady state 32

4.6 l_2-norm magnetic field PHM skewed grid 35

4.7 l_2-norm magnetic field Potential skewed grid 36

4.8 l_2-norm error γ analysis 37

4.9 E_x on skewed grid PHM 39

4.10 E_x on skewed grid Lorenz 40

4.11 E_x on skewed grid solved using the Coulomb gauge Formulation 41

4.12 Differences in Magnitude of the Electric Field Between Transient and Steady State Solutions 42

4.13 Steady State Divergence Error PHM 44

4.14 Steady State Divergence Error Lorenz Gauge 45

4.15 Steady State Divergence Error Coulomb Gauge 46

4.16 Steady State Divergence Error 47

4.17 l_2-norm error χ analysis steady state charge 49

4.18 Electric Field in the x-direction PHM Solution 52

4.19 Electric Field in the x-direction Lorenz Solution 53

4.20 Electric Field in the x-direction Coulomb Solution 54

4.21 Divergence Error PHM System 56

4.22 Divergence Error Lorenz Gauge 57

4.23 Divergence Error Coulomb Gauge 58

4.24 Divergence Error for Different Values of χ for the time-dependent charge density problem 60

4.25 Zoomed-in Skewed grids 61
4.26 Grid Effects for Current Problem 63
4.27 Grid Effects for Steady State Charge Problem 65

5.1 Electron Mass Density Two Fluid MHD Shock Problem 71
5.2 Ion Mass Density Two Fluid MHD Shock Problem 72
5.3 MHD Shock Div E Error 74
5.4 Momentum of the Reconnection Problem for PHM $\chi = 0$ 77
5.5 Momentum of the Reconnection Problem for PHMaxwell $\chi = c_0$ 78
5.6 Momentum of the Reconnection Problem for PHMaxwell $\chi = 2c_0$ 83
5.7 Electron Momentum of the Reconnection Problem for PHM $\chi = 4c_0$ 84
5.8 Ion Momentum of the Reconnection Problem for PHM $\chi = 4c_0$ 85
5.9 Momentum of the Reconnection Problem for Lorenz solver 86
5.10 Momentum of the Reconnection Problem for Coulomb solver 87
5.11 Comparison of Reconnected Flux 88
5.12 Reconnection Div E Error 89
5.13 Spatial Variation of Divergence of **E** Error 90
5.14 Reconnection Div B Error 91

LIST OF TABLES

Table Number Page

4.1 The slope of the l_2-norm of the Divergence E Error on The Skewed Grid . . . 55

ACKNOWLEDGMENTS

The author would like to express gratitude to, first of all, Professor Uri Shumlak for his guidance and expertise in plasma physics, and numerical modelling. A thank you also goes out to Professor Brian Nelson for his willingness to be part of the author's reading committee. The author would like to thank Dr. Ammar Hakim for substantial programming and numerics help, and also for letting us use the code, WAPRX, whose foundation he has laid down from scratch. The computational fluid dynamics group is forever indebted to him.

The author would like to thank Professor Dana Dabiri for introducing him to the world of research. If it weren't for him, the author would not even consider being a graduate student. The author would also like to extend his gratitude to other lab dwellers, Robert Clifton Lilly, Weston Lowrie, Eder Marinho Sousa, and Bhuvana Srinivasan. This thesis is as much mine as it is theirs. And of course, the author can't forget his family, his parents, Emi and Freddy Susanto, and his brothers, Anthonius and Eric Susanto. Also many thanks to the Lai Family and its acquintances. Thank you so much for sticking with him until the very end. Lastly, but as importantly, the author wishes to thank his God, for the metaphysical, and spiritual strength that have been given him to pursue this work.

Chapter 1
INTRODUCTION

1.1 Introductory Remarks

The Computational Plasma group at the University of Washington has been developing numerical method for the two-fluid plasma algorithm. Physical, numerical and comparison studies of the model have been conducted by the group's researchers in the past [1, 2, 3].

The main idea of the two-fluid plasma system is to model the plasma as a fluid consisting of two species (the ions and the electrons). The evolution of each species is described by the Euler equations for gas dynamics with electromagnetic source terms. The electromagnetic sources can be obtained by advancing the Maxwell's equations either directly, or by expressing them in potential formulations.

The Euler's equations and the Maxwell's equations are first order hyperbolic systems of equations, and can be written in the following balance law

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S}. \tag{1.1}$$

The potential formulation, however, as described in Sec. 2.2.2, is a second order system, and conventional numerical method for first order hyperbolic system cannot be directly applied. For first order systems, LeVeque's wave propagation method [4] is used. For the potential formulation, numerical techniques are developed.

The results are then compared in terms of convergence, divergence error, and computational expenses.

1.2 Software Used

The simulations within this study are conducted using the WARPX (Washington Approximate Riemann Problem Solver Version X) code, which is a framework specialized for computational physics.

WARPX was originally written with hyperbolic balance laws in mind, similar to CLAWPACK [5], but the framework has since been extended to also include finite element, and elliptic solvers, capabilities which are beyond CLAWPACK offering. The framework has been written in C++, to output solutions into the HDF5 format [6]. WARPX also takes advantage of multi-processor architecture through MPI implementations, such as MPICH2 [7] and OpenMPI [8]. Linear systems of equations, such as the second derivative operator in the Poisson's equations, are solved using PETSc [9], which itself requires its own set of dependencies.

Of all the external packages that might be linked to PETSc, WARPX is compatible with SuperLU, which is a library of direct solvers for "large, sparse, nonsymmetric systems of linear equations", [10] and also Hypre, "a library of high performance preconditioners that features parallel multigrid methods for both structured and unstructured grid problems" [11].

Parallel direct solver can be accomplished using SuperLU_DIST [12], but as of the time of this writing, it is not yet a viable solution. While it does work in parallel, the scaling is poor, and for some problems, it is actually slower in parallel than in serial. The possibility of using parallel direct solver in WARPX is still currently being investigated.

Chapter 2
TWO FLUID SYSTEM FOR PLASMA

2.1 Euler Equations to Model Plasma Fluid

The two-fluid plasma system is derived by taking moments of the Vlasov equation, which can be written as

$$\frac{\partial f_s}{\partial t} + \mathbf{v}_s \cdot \nabla f_s + \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v}_s \times \mathbf{B})\nabla_v f_s = 0. \tag{2.1}$$

E and **B** here denote the electric and magnetic fields respectively, q_s and m_s are the charge and mass of plasma species, and s denotes each species. The distribution function is represented here by f_s. For Euler equations, the fluid variables are defined by multiplying the distribution functions by the m_s and $m_s\mathbf{v}_s$ to obtain mass density and momentum, which can be written as follows:

$$\rho_s = m_s \int_{-\infty}^{\infty} f_s \, d\mathbf{v} \tag{2.2}$$

$$\mathbf{u}_s = \int_{-\infty}^{\infty} \mathbf{v}_s f_s \, d\mathbf{v} \tag{2.3}$$

Here, $\mathbf{v}_s$ is the phase-space velocity, whereas $\mathbf{u}_s$ is the average fluid velocity. The conservation equations for mass and momentum can then be obtained by multiplying the Vlasov equation by the m_s and $m_s\mathbf{v_s}$. The resulting fluid equations can then be written as:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \mathbf{u_s}) = 0 \tag{2.4}$$

$$\frac{\partial(\rho_s \mathbf{u_s})}{\partial t} + \nabla \cdot (\bar{\bar{p}}_s + \rho_s \mathbf{u_s}\mathbf{u_s}) = \frac{\rho_s q_s}{m_s}(\mathbf{E} + \mathbf{u_s} \times \mathbf{B}) \tag{2.5}$$

As is the case with any moment models, when new moments are taken, new flux terms appear. Some way to close the system is then needed, and for the Euler equations, another moment is taken by multiplying the Vlasov equation with the square of the velocity, and integrating it over the whole phase-space domain.

The aforementioned step introduces total energy, which can be written as:

$$\varepsilon_s = \frac{1}{2} m_s \int_{-\infty}^{\infty} \mathbf{v}_s \cdot \mathbf{v}_s f \, d\mathbf{v} \tag{2.6}$$

whose conservation equation can be written as follows:

$$\frac{\partial \varepsilon_s}{\partial t} + \nabla \cdot (p_s \mathbf{u_s} + \varepsilon_s \mathbf{u_s}) = \frac{\rho_s q_s}{m_s} \mathbf{u_s} \cdot \mathbf{E} \tag{2.7}$$

Here, p_s represents the scalar pressure, and it is related to the total energy through the following closure relation:

$$\varepsilon_s = \frac{3}{2} p_s + \frac{1}{2} m_s n_s \mathbf{u_s} \cdot \mathbf{u_s} \tag{2.8}$$

where n_s is the number density.

Thus, Eq. 2.4, Eq. 2.5, and Eq. 2.7, with the closure relation as described in Eq. 2.8, and coupled with the electromagnetic fields as described in Sec. 2.2, fully describe the ideal two-fluid five-moment plasma system.

Higher order moment equations such as the ten moment, and the thirteen moment equations can also be used to describe the fluid parts of the plasma system [13, 14]. The feasibility of extending the two-fluid plasma model to include the more physically complete higher order moment equations is currently being investigated [15].

The plasma simulations conducted in Sec. 5, however, will only employ the five-moment model, as it is not yet clear at this point, if the ten moment model has readily been implemented to simulate two-fluid plasma.

2.2 Electromagnetic Models

The electromagnetic fields for the two-fluid plasma system are advanced using Maxwell's equations:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2.9}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \tag{2.10}$$

$$\nabla \cdot \mathbf{E} = \frac{\rho_c}{\epsilon_0} \tag{2.11}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{2.12}$$

Eq. 2.9 and Eq. 2.10 are usually referred to as the curl equations, whereas Eq. 2.11 and

Eq. 2.12 are the divergence constraints. With 8 equations and 6 unknowns, the Maxwell system is overdetermined. However, it has been mathematically shown that, if the constraints are initially satisfied, then they will always be satisfied [16]. The coupling to plasma happens through the fluid source terms, as present in Eq. 2.10 and Eq. 2.11. In terms of the fluid variables, the source terms are given as

$$\mathbf{J} = \sum_{\alpha} q_\alpha n_\alpha \mathbf{u}_\alpha \tag{2.13}$$

$$\rho_c = \sum_{\alpha} q_\alpha n_\alpha \tag{2.14}$$

The summation is carried out over all species within the plasma (ion and electron for the two-fluid plasma system). Here, q_α denotes the charge of a species, and $\mathbf{u}_\alpha$ is the plasma velocity. α denotes the index of a species (ion or electron). For plasma simulations in which source terms for Maxwell's equations are present, charge conservation can easily be violated, and this can lead to a very large numerical divergence error that may develop over time.

Therefore, developing electromagnetic solvers for plasma simulation is of great importance. Towards this end, two methods are investigated. The first one is the perfectly hyperbolic Maxwell's equations, and the second is by formulating the electromagnetic fields in terms of their potentials. Brief discussions of these methods are given in Sec. 2.2.1 and Sec. 2.2.2.

2.2.1 Perfectly Hyperbolic Maxwell's Equations

Perfectly Hyperbolic Maxwell (PHM) equations can be written as follows [16]

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} + \gamma \nabla \psi = 0 \tag{2.15}$$

$$\frac{\partial \mathbf{E}}{\partial t} - c^2 \nabla \times \mathbf{B} + \chi c^2 \nabla \phi = -\frac{\mathbf{J}}{\epsilon_0} \tag{2.16}$$

$$\frac{\partial \phi}{\partial t} + \chi \nabla \cdot \mathbf{E} = \chi \frac{\rho_c}{\epsilon_0} \tag{2.17}$$

$$\frac{\partial \psi}{\partial t} + \gamma c^2 \nabla \cdot \mathbf{B} = 0 \tag{2.18}$$

This formulation introduces two new variables, ϕ and ψ, which are the error correction potentials. These variables couple the divergence constraints to the main system that is now no longer overdetermined. The correction speeds, χ and γ, are the speeds by which the divergence error is convected out of the domain. Observing Eq. 2.17 and Eq. 2.18, it can be seen that PHM approaches regular Maxwell's equations as these correction speeds approach infinity. Preliminary observations by Hakim [1] have shown that a good value for χ and γ in a plasma simulation is about equal to or two times the speed of light.

The Perfectly Hyperbolic Maxwell's equations are a system of hyperbolic equations, which can be conveniently solved numerically by using methods such as the wave propagation method [4] and the discontinuous Galerkin method [17].

2.2.2 Electromagnetic Potential Formulation

Another approach to minimize numerical divergence error is by formulating the electric and the magnetic fields in terms of their potentials

$$\mathbf{E} = -\nabla\Phi - \frac{\partial \mathbf{A}}{\partial t} \tag{2.19}$$

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{2.20}$$

The advance equations can be written as follows

$$\nabla^2 \Phi = -\frac{\rho_c}{\epsilon_0} \tag{2.21}$$

$$\frac{\partial^2 \mathbf{A}}{\partial t^2} - c^2 \nabla^2 \mathbf{A} = \frac{\mathbf{J}}{\epsilon_0} - \nabla \left(\frac{\partial \Phi}{\partial t} \right) \tag{2.22}$$

if the Coulomb gauge condition,

$$\nabla \cdot \mathbf{A} = 0, \tag{2.23}$$

is assumed.

The right hand side of Eq. 2.22 represents the divergence-free displacement current. The $\frac{\partial \Phi}{\partial t}$ term can be derived by combining Eq. 2.21, with conservation of charge, from which the following relation can be derived

$$\nabla \cdot \frac{\mathbf{J}}{\epsilon_0} = \nabla^2 \left(\frac{\partial \Phi}{\partial t} \right) = -\frac{\partial}{\partial t} \frac{\rho_c}{\epsilon_0} \tag{2.24}$$

The Coulomb gauge preserves the electric and magnetic field divergence constraints by expressing them in terms of scalar and vector potentials. Charge conservation is also directly included in the calculation through Eq. 2.24. After solving for the $\frac{\partial \Phi}{\partial t}$ term, which is the solution to Poisson's equation, Eq. 2.24, the source term for Eq. 2.22 can be computed, and is mathematically divergence-free, which leads to a divergence-free vector potential solution as required by Eq. 2.23.

The potentials can be modelled by the second-order wave equations, which can be written as

$$\frac{\partial^2 \mathbf{A}}{\partial t^2} - c^2 \nabla^2 \mathbf{A} = \frac{\mathbf{J}}{\epsilon_0} \tag{2.25}$$

$$\frac{\partial^2 \Phi}{\partial t^2} - c^2 \nabla^2 \Phi = \frac{c^2 \rho_c}{\epsilon_0} \tag{2.26}$$

if the Lorenz gauge condition,

$$\nabla \cdot \mathbf{A} = -\frac{1}{c^2} \frac{\partial \Phi}{\partial t}, \tag{2.27}$$

is assumed. By expressing the electromagnetic fields in terms of their potentials, one explicitly includes and advances the magnetic and the electric field divergence constraints (Eq. 2.11 and Eq. 2.12).

Both of these formulations are second-order systems of differential equations, and cannot be solved directly by hyperbolic solvers such as the wave propagation and the discontinuous Galerkin methods. Finite volume numerical methods to solve these systems are developed, and described in chapter 3.

Chapter 3
NUMERICAL METHODS

3.1 Wave Propagation Method

Maxwell's equations form a hyperbolic system of equations. For this class of problems, various numerical techniques are available, one of which is the wave propagation method [4].

Within this section, the algorithms for the wave propagation method in general two-dimensional quadrilateral geometry is briefly described. There are extensive literatures detailing numerical algorithms for the wave propagation method in a uniform Cartesian grid [1, 4, 3], so such efforts are not duplicated here.

The wave propagation method can only be directly applied to the homogeneous part of the balance law (which is the left hand side of Eq. 1.1). The source terms are treated through another method, which is to be discussed in Sec. 4.2.

3.1.1 First Order Godunov Update

The homogeneous part of the hyperbolic balance law is given by the left hand side of Eq. 1.1. To advance the solutions, one first needs to apply a first order Godunov update to the solutions, which requires approximating the Riemann problem, which, in its most fundamental sense, is a piecewise constant having a single discontinuity.

LeVeque [4] proposes two ways to solve the Riemann problem across an interface in general geometry. The first one is to modify the eigensystem of the Riemann problem so that it will work with arbitrary normal directions. The second one is solving the Riemann problem by first rotating the data so that it will align with the normal and tangential coordinate systems of the interface the Riemann problem is to be solved, and then solving the regular Riemann problem across the interface. The second method is suitable for a framework that already has a fairly complicated Cartesian Riemann solver implemented, and this is the method that is used for this project.

In his book, LeVeque [4] formulates the update formula through a fluctuation differencing form. The formula adopted here is a direct application of the divergence theorem in which actual volumes and surface area vectors are used instead of ratios of computational to physical quantities as with the fluctuation differencing form.

The general update formula for finite volume method can be written as [4]

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{|C_{ij}|}\left(h_{i+\frac{1}{2},j}\breve{F}_{i+\frac{1}{2},j} - h_{i-\frac{1}{2},j}\breve{F}_{i-\frac{1}{2},j} + h_{i,j+\frac{1}{2}}\breve{G}_{i,j+\frac{1}{2}} - h_{i,j-\frac{1}{2}}\breve{G}_{i,j-\frac{1}{2}}\right) \tag{3.1}$$

where $\breve{F}$ and $\breve{G}$ are fluxes normal the interfaces at which Riemann problem is being solved, $|C_{ij}|$ is the volume of cell i, j, and h the length of the sides. The half-indices indicate locations in between two adjacent cells. For examples, indices $i, j + \frac{1}{2}$ indicate an interface that is located between cells i, j and $i, j + 1$.

To obtain fluxes normal to the cell interfaces, the Cartesian data are first rotated into the local coordinate system. Then, the fluctuations are computed using the regular un-modified Riemann solver.

For example, consider cell i, j and $i + 1, j$. To rotate the global Cartesian data to the local $i+\frac{1}{2}, j$, in two dimensions, the following rotation matrix is applied to a vector solution:

$$\mathcal{R}_{i+\frac{1}{2},j} = \begin{bmatrix} n^{x}_{i+\frac{1}{2},j} & n^{y}_{i+\frac{1}{2},j} \\ -n^{y}_{i+\frac{1}{2},j} & n^{x}_{i+\frac{1}{2},j} \end{bmatrix} \tag{3.2}$$

In WARPX, the following commands are executed to rotate the right and the left data from the global to the local coordinate systems

```
// rotate right data to local coordinate system
_eqnSet.rotateToLocalFrame(norm, qoldItr.data(), &qrLocal[0]);
// rotate left data to local coordinate system
_eqnSet.rotateToLocalFrame(norm, qoldItr1.data(), &qlLocal[0]);
```

Using the rotated data, the rotated waves, $\breve{\mathcal{W}}_{i-\frac{1}{2},j}$, can then be computed using the usual eigen decomposition [4]. The resulting waves then have to be re-rotated back to Cartesian coordinate system, from which the fluctuations can be computed

$$\breve{\mathcal{A}}^{\pm}_{i-\frac{1}{2},j}\Delta Q_{i-\frac{1}{2},j} = \sum_{p=1}^{m}\left(\breve{s}^{p}_{i-\frac{1}{2},j}\right)^{\pm}\breve{\mathcal{W}}^{p}_{i-\frac{1}{2},j} \tag{3.3}$$

where m is the number of waves present in the equation system, $(\breve{s}^p_{i-\frac{1}{2},j})^{\pm}$ is the p^{th} eigenvalue of $\breve{\mathcal{A}}^{\pm}_{i-\frac{1}{2},j}$, and $\breve{\mathcal{W}}^p_{i-\frac{1}{2},j}$ is the p^{th} wave of the system.

In WARPX, the fluctuations are computed through the following code fragment

```
for (unsigned m=0; m<meqn; ++m)
{
  amdqxItr[m] = 0.0;
  apdqxItr[m] = 0.0;
  for (unsigned mw=0; mw<mwave; ++mw)
  {
    if (sxItr[mw] < 0.0)
      amdqxItr[m] += sxItr[mw]*_waveax->operator()(i,m,mw);
    else
      apdqxItr[m] += sxItr[mw]*_waveax->operator()(i,m,mw);
  }
}
```

The full Godunov update equation for the wave propagation method can then be written as follows

$$\begin{aligned} Q^{n+1}_{ij} = Q^n_{ij} - \frac{\Delta t}{|C_{ij}|}(&h_{i-\frac{1}{2},j}\mathcal{R}^{-1}_{i-\frac{1}{2},j}\breve{\mathcal{A}}^{+}_{i-\frac{1}{2},j}\Delta Q_{i-\frac{1}{2},j} + h_{i+\frac{1}{2},j}\mathcal{R}^{-1}_{i+\frac{1}{2},j}\breve{\mathcal{A}}^{-}_{i+\frac{1}{2},j}\Delta Q_{i+\frac{1}{2},j} \\ &+h_{i,j-\frac{1}{2}}\mathcal{R}^{-1}_{i,j-\frac{1}{2}}\breve{\mathcal{B}}^{+}_{i,j-\frac{1}{2}}\Delta Q_{i,j-\frac{1}{2}} + h_{i,j+\frac{1}{2}}\mathcal{R}^{-1}_{i,j+\frac{1}{2}}\breve{\mathcal{B}}^{-}_{i,j+\frac{1}{2}}\Delta Q_{i,j+\frac{1}{2}}) \end{aligned} \tag{3.4}$$

which is first-order accurate.

3.1.2 Second-Order Corrections

The Godunov update method, as described in Sec. 3.1.1, is only first order accurate. To improve the order of accuracy to second order, second order correction terms can be applied to the update formula (Eq. 3.4), resulting in the following

$$\begin{aligned} Q^{n+1}_{ij} = Q^n_{ij} - \frac{\Delta t}{|C_{ij}|}(&h_{i-\frac{1}{2},j}\mathcal{R}^{-1}_{i-\frac{1}{2},j}\breve{\mathcal{A}}^{+}_{i-\frac{1}{2},j}\Delta Q_{i-\frac{1}{2},j} + h_{i+\frac{1}{2},j}\mathcal{R}^{-1}_{i+\frac{1}{2},j}\breve{\mathcal{A}}^{-}_{i+\frac{1}{2},j}\Delta Q_{i+\frac{1}{2},j} \\ &+h_{i,j-\frac{1}{2}}\mathcal{R}^{-1}_{i,j-\frac{1}{2}}\breve{\mathcal{B}}^{+}_{i,j-\frac{1}{2}}\Delta Q_{i,j-\frac{1}{2}} + h_{i,j+\frac{1}{2}}\mathcal{R}^{-1}_{i,j+\frac{1}{2}}\breve{\mathcal{B}}^{-}_{i,j+\frac{1}{2}}\Delta Q_{i,j+\frac{1}{2}}) \\ -\frac{\Delta t}{|C_{ij}|}(&h_{i+\frac{1}{2},j}\breve{\mathcal{F}}^2_{i+\frac{1}{2},j} - h_{i-\frac{1}{2},j}\breve{\mathcal{F}}^2_{i-\frac{1}{2},j} + h_{i,j+\frac{1}{2}}\breve{\mathcal{G}}^2_{i,j+\frac{1}{2}} - h_{i,j-\frac{1}{2}}\breve{\mathcal{G}}^2_{i,j-\frac{1}{2}}) \end{aligned} \tag{3.5}$$

where

$$\breve{\mathcal{F}}^2_{i-\frac{1}{2},j} = \frac{1}{2}\mathcal{R}^{-1}_{i-\frac{1}{2},j}\sum_{p=1}^{m}|\breve{s}^p_{i-\frac{1}{2},j}|\left(1 - \frac{\Delta t}{|C_{i-\frac{1}{2},j}|}|\breve{s}^p_{i-\frac{1}{2},j}|\right)\hat{\mathcal{W}}^p_{i-\frac{1}{2},j} \tag{3.6}$$

$\breve{\mathcal{G}}^2_{i,j-\frac{1}{2}}$ can then be constructed in a similar manner. The following WARPX code fragment is responsible for computing the second order flux correction term

```
for (unsigned m=0; m<meqn; ++m)
{
  fsxItr[m] = 0.0;
  for (unsigned mw=0; mw<mwave; ++mw)
  {
    sabs = fabs(sxItr[mw]);
    corr = 0.5*sabs*(1.0 - sabs*dt*areaVol)*_waveax->operator()(i,m,mw);
    fsxItr[m] += corr; // compute second order correction
  }
}
```

Here, $\hat{\mathcal{W}}$ is the limited wave, which is the same as $\breve{\mathcal{W}}$ if no limiter is applied. When the limiter is applied, the following formula is applied:

$$\hat{\mathcal{W}}^p_{i-\frac{1}{2},j} = \breve{\mathcal{W}}^p_{i-\frac{1}{2},j}\varphi\left(\theta^p_{i-\frac{1}{2},j}\right) \tag{3.7}$$

where $\varphi(\theta^p_{i-\frac{1}{2},j})$ is a suitable limiter function, and

$$\theta^p_{i-\frac{1}{2},j} \equiv \frac{\breve{\mathcal{W}}^p_{I-\frac{1}{2},j}\breve{\mathcal{W}}^p_{i-\frac{1}{2},j}}{\breve{\mathcal{W}}^p_{i-\frac{1}{2},j}\breve{\mathcal{W}}^p_{i-\frac{1}{2},j}} \tag{3.8}$$

where $I = i - 1$ for $\breve{s}^p_{i-\frac{1}{2},j} > 0$, and $I = i + 1$ for $\breve{s}^p_{i-\frac{1}{2},j} < 0$. WARPX currently supports the following limiters: monotonized-centered, superbee, minmod and van Leer. Please refer to [4] for more detailed discussions about these limiters.

For all of the plasma simulations conducted in Sec. 5, van Leer is used, whereas, monotonized-centered is used for electromagnetic simulations conducted in Sec. 4.

3.1.3 Transverse Corrections

Up to this point, after including the correction terms for fluxes, the method is second order accurate. However, the method can only be run up to a CFL number of $\frac{1}{2}$. For an explicit scheme, being able to run the simulations with a CFL number of up to one is desirable. Towards this end, the wave propagation method can be modified to also include the transverse correction terms.

To accomplish this, first, the fluctuations in global Cartesian coordinate system, $\mathcal{A}\Delta Q$, is rotated to the local coordinate system in the transverse direction. For example, if one is working on an x-sweep and the current cell is i, j, then the local coordinate systems will be those associated with the $i, j - \frac{1}{2}$ and the $i, j + \frac{1}{2}$ interfaces. Each of the fluctuations (left and right) is then split into up-going and down-going transverse fluctuations, $\mathcal{B}^{\pm}\mathcal{A}\Delta Q$. To split the normal fluctuations, the transverse going waves are computed by solving Riemann problem in the transverse direction. The resulting waves are then rotated back into the global coordinate system, and multiplied with the transverse eigenvalues.

The transverse correction term can then be added to the flux second order correction term (Eq. 3.6). For example, the fluctuations, $\mathcal{B}^{\pm}\mathcal{A}^{\pm}\Delta Q_{i-\frac{1}{2},j}$, affect 4 neighboring correction fluxes, $\breve{\mathcal{G}}^2_{i,j+\frac{1}{2}}$, $\breve{\mathcal{G}}^2_{i-1,j+\frac{1}{2}}$, $\breve{\mathcal{G}}^2_{i,j-\frac{1}{2}}$, and $\breve{\mathcal{G}}^2_{i-1,j-\frac{1}{2}}$.

For example, the fluctuation, $\mathcal{B}^{+}\mathcal{A}^{+}\Delta Q_{i-\frac{1}{2},j}$, affects the flux $\breve{\mathcal{G}}^2_{i,j+\frac{1}{2}}$ in the following manner

$$\breve{\mathcal{G}}^2_{i,j+\frac{1}{2}} := \breve{\mathcal{G}}^2_{i,j+\frac{1}{2}} - \frac{\Delta t}{2|C_{ij}|}\mathcal{R}^{-1}_{i,j+\frac{1}{2}} h_{i-\frac{1}{2},j}\breve{\mathcal{B}}^{+}\breve{\mathcal{A}}^{+}\Delta Q_{i-\frac{1}{2},j} \tag{3.9}$$

The other fluxes are then updated in a similar manner, which, in total (after including the modifications due to the up and down going fluctuations, $\mathcal{B}^{\pm}\Delta Q_{i,j-\frac{1}{2}}$, as well), modify 6 of the neighboring cells. With transverse correction terms, the wave propagation method is now stable up to a CFL number of 1.

The transverse fluctuations are computed by executing the following code pieces in WARPX

```
eqnSet.riemannt(0, xl, xc, &qlLocal[0].data(), &qrLocal[0].data(),
  amdqxItr.data(), _bmamdq, _bpamdq, apdqxItr.data(), _bmapdq, _bpapdq);
```

which are done in local coordinate systems.

3.2 Laplacian Operator

Expanding the Laplacian operator to $\nabla\cdot\nabla u$, and applying the divergence theorem of calculus yields

$$\int\limits_{V} \nabla^2 u \, dV = \int\limits_{\partial V} \nabla u \cdot d\mathbf{S}, \tag{3.10}$$

where V represents the cell volume of interest, and ∂V denotes the surface area surrounding V. We can then take the cell-averaged values of the Laplacian, and in discretized space, Eq.

3.10 can be written as

$$\nabla^2 u = \frac{1}{V} \sum \nabla u \cdot \hat{n} \Delta \mathbf{S} \tag{3.11}$$

The sum is taken over all interfaces of a cell volume, with normal area vectors pointing outwards. Most of the numerical efforts to approximate the solution actually go into approximating the gradients, $\nabla u \cdot \hat{n}$, at the cell interfaces [18].

To illustrate the point, consider the following 2-dimensional 9-point stencil of structured quadrilateral grids (Fig. 3.1). To approximate the gradients at the cell interfaces, namely, at points A, B, C and D, one needs to also approximate the values of u at points E, F, G, H, I, J, K, and L as well.

This is done by applying linear interpolation between the solution at the cell center (the cell center averaged solution), and the neighboring solution. For example, the solution at point E is approximated through linear weighted averaging the solutions at cell center 0 and cell center 1. The solution at point E can then be approximated as

$$u_E = u_0 \frac{d(E,1)}{d(0,1)} + u1 \frac{d(E,0)}{d(0,1)} \tag{3.12}$$

The gradient at point A can then be approximated as follows:

$$\nabla u \cdot \hat{n}|_A = \frac{u_I - u_E}{d(E,I)} \tag{3.13}$$

where $d(E,I)$ is the distance between point E and point I. Gradient operators at points B, C and D can be then constructed in a similar manner.

The full Laplacian operator can then be written as follows:

$$\nabla^2 u = \frac{1}{V}(\nabla u \cdot \hat{n}|_A S_A + \nabla u \cdot \hat{n}|_B S_B + \nabla u \cdot \hat{n}|_C S_C + \nabla u \cdot \hat{n}|_D S_D) \tag{3.14}$$

S_A, S_B, S_C, and S_D are lengths of sides A, B, C and D respectively.

This Laplacian operator is similar to the one proposed by Faille [18], and provides close to second order of accuracy for reasonably smooth quadrilateral grid. The method also collapses to the 5-point stencil Laplacian in regular Cartesian domain.

3.2.1 WARPX Implementation of the Laplacian Operator

The Laplacian operator can be expressed as an $A\mathbf{x} = \mathbf{b}$ problem. Now, the vector, $\mathbf{x}$ is actually known, and matrix A is the Laplacian operating matrix. To find the Laplacian of

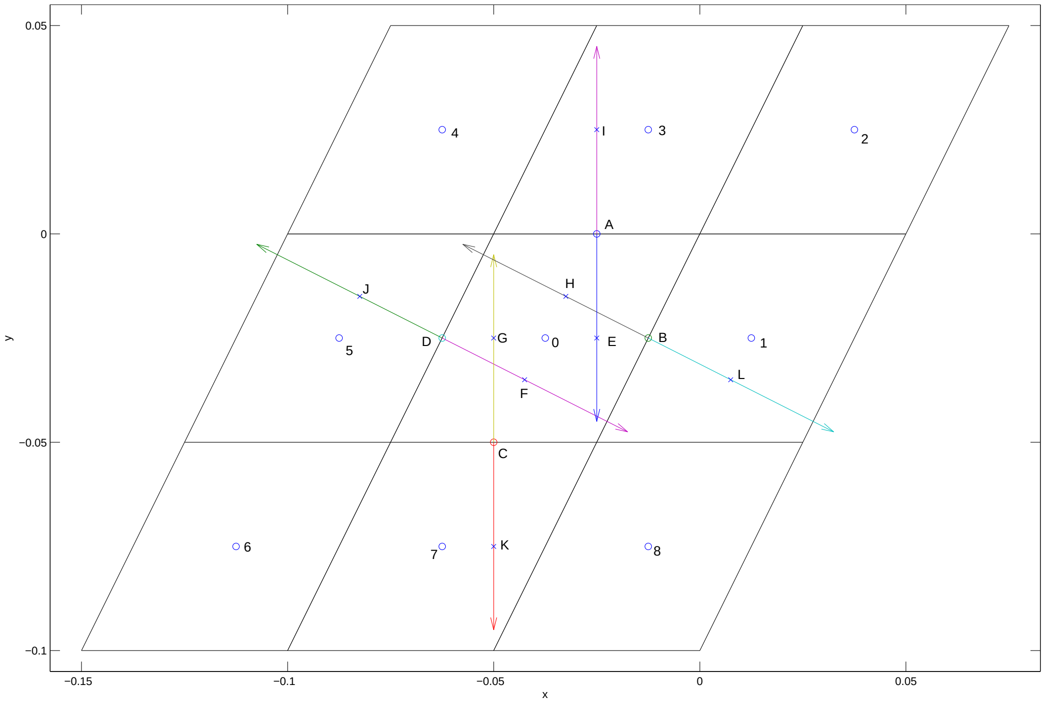

Figure 3.1: An example of a 9-point trapezoidal stencil on a structured quadrilateral 2-dimensional grid. Applying the divergence theorem involves approximating the solutions at cell interfaces, which are represented by points A, B, C and D, which are the midpoints of the cell faces. To approximate the quantities of interest at, for example, point A, the solutions need to first be approximated through linear approximation at points E and I.

x, which is the vector, b, matrix multiplication is performed.

In WARPX, the following pieces of the code are executed

```
for (int k = laplacian.lower(0); k<laplacian.upper(0); ++k)
{
  for (int l = laplacian.lower(1); l<laplacian.upper(1); ++l)
  {
    A0 = 0.0; A1 = 0.0; A2 = 0.0;
    A3 = 0.0; A4 = 0.0; A5 = 0.0;
    A6 = 0.0; A7 = 0.0; A8 = 0.0;

    laplacian.setItr(laplacianItr, k, l);

    solutions.setItr(solutionsItr0, k  , l  );
    solutions.setItr(solutionsItr1, k+1, l  );
    solutions.setItr(solutionsItr2, k+1, l+1);
    solutions.setItr(solutionsItr3, k  , l+1);
    solutions.setItr(solutionsItr4, k-1, l+1);
    solutions.setItr(solutionsItr5, k-1, l  );
    solutions.setItr(solutionsItr6, k-1, l-1);
    solutions.setItr(solutionsItr7, k  , l-1);
    solutions.setItr(solutionsItr8, k+1, l-1);

    computeTopFlux(geo, intersects, A0, A1, A2, A3, A4, A5, k, l);
    computeRightFlux(geo, intersects, A0, A1, A2, A3, A7, A8, k, l);
    computeBottomFlux(geo, intersects, A0, A1, A5, A6, A7, A8, k, l);
    computeLeftFlux(geo, intersects, A0, A3, A4, A5, A6, A7, k, l);

    laplacianItr[_laplace_index] =
      A0*solutionsItr0[_solution_index] +
      A1*solutionsItr1[_solution_index] +
      A2*solutionsItr2[_solution_index] +
      A3*solutionsItr3[_solution_index] +
      A4*solutionsItr4[_solution_index] +
      A5*solutionsItr5[_solution_index] +
      A6*solutionsItr6[_solution_index] +
      A7*solutionsItr7[_solution_index] +
      A8*solutionsItr8[_solution_index];
  }
}
```

After coefficients of A are computed, it multiplies vector $\mathbf{x}$. The numbering scheme shown in the aforementioned code fragments follows the one described in Fig. 3.1.

The matrix coefficients, A_0 - A_8, are computed using the method described in Sec. 3.2, through the functions *computeTopFlux*, *computeRightFlux*, *computeBottomFlux*, and *computeLeftFlux*. The *computeTopFlux* function modifies all of the coefficients that are relevant to the computation of the flux term for the top cell face, namely A_0, A_1, A_2, A_3, A_4, A_5. Coefficients associated with the other cell faces are computed in a similar manner.

The following pieces of code details implementation for the *computeTopFlux* function in WARPX

```
intersects.setIndex(local_col_index  , local_row_index  );
intersects.getUpperIntersection(1, t_inter_0);

intersects.setIndex(local_col_index  , local_row_index+1);
intersects.getLowerIntersection(1, b_inter_3);

dx_temp = b_inter_3[0] - t_inter_0[0];
dy_temp = b_inter_3[1] - t_inter_0[1];
distance0i = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

if (t_inter_0[0] > cell0[0])
{
  A5 += 0;

  dx_temp = cell1[0] - cell0[0];
  dy_temp = cell1[1] - cell0[1];
  distance0p = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  //dx_temp = intersectsItr0[_tIntersect_x] - cell0[0];
  //dy_temp = intersectsItr0[_tIntersect_y] - cell0[1];

  dx_temp = t_inter_0[0] - cell0[0];
  dy_temp = t_inter_0[1] - cell0[1];

  distance0 = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  A1 += -(distance0/distance0p)/(distance0i*geo.getCellVolume())*top_area;
}
else
{
  A1 += 0;

  dx_temp = cell0[0] - cell5[0];
  dy_temp = cell0[1] - cell5[1];
  distance0p = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  dx_temp = cell0[0] - t_inter_0[0];
  dy_temp = cell0[1] - t_inter_0[1];
  distance0 = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  A5 += -(distance0/distance0p)/(distance0i*geo.getCellVolume())*top_area;
}

if (b_inter_3[0] > cell3[0])
{
  A4 += 0;

  dx_temp = cell2[0] - cell3[0];
  dy_temp = cell2[1] - cell3[1];
  distance3p = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  dx_temp = b_inter_3[0] - cell3[0];
  dy_temp = b_inter_3[1] - cell3[1];
  distance3 = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  A2 += (distance3/distance3p)/(distance0i*geo.getCellVolume())*top_area;
}
else
{
  A2 += 0;

  dx_temp = cell3[0] - cell4[0];
  dy_temp = cell3[1] - cell4[1];
  distance3p = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  dx_temp = cell3[0] - b_inter_3[0];
  dy_temp = cell3[1] - b_inter_3[1];
  distance3 = sqrt(dx_temp*dx_temp + dy_temp*dy_temp);

  A4 += (distance3/distance3p)/(distance0i*geo.getCellVolume())*top_area;
}

A0 += -(1-distance0/distance0p)/(distance0i*geo.getCellVolume())*top_area;
A3 += (1-distance3/distance3p)/(distance0i*geo.getCellVolume())*top_area;
```

The if statements within the *computeTopFlux* function are used to determine the location of point E (see Fig. 3.1) is to the right or to the left of point 0. If it is to the right of point 0, then to approximate the solution at point E, the solution at point 1 is needed, so A_1 has to be modified. Similarly, if it is to the right of point 0, then A_5 would need to be modified, because then, the solution would require interpolation between point 0 and 5.

Other derivative operators described in Sec. 3.3, Sec. 3.4, and Sec. 3.5 employ similar techniques for interpolation and flux calculation, so code fragments for these operators will not be provided because their WARPX implementations are similar to the implementation of the Laplacian operator described in this Section.

3.3 Curl Operator

The divergence theorem can also be applied to a cross product so that the curl operator in finite volume can be derived and written as

$$\int\limits_V \nabla \times \mathbf{F}\, dV = \int\limits_{\partial V} d\mathbf{S} \times \mathbf{F} \tag{3.15}$$

Once again, the numerical approximation now comes down mostly to approximating the values of **F** at the cell interfaces. In the current implementation, the face values are interpolated in a similar manner as in the case of Laplacian operator.

For example, to approximate the value of **F** at point A (see Fig. 3.1), one first needs to approximate the solution at points E and I. Through linear interpolation of these two values, the solution at point A can be approximated as follows

$$\mathbf{F}_A = \mathbf{F}_E \frac{d(A, I)}{d(E, I)} + \mathbf{F}_I \frac{d(A, E)}{d(E, I)} \tag{3.16}$$

The values of $\mathbf{F}_B$, $\mathbf{F}_C$, and $\mathbf{F}_D$ can then be approximated in a similar manner.

In discrete space, Eq. 3.15 can be written as

$$\begin{aligned} \nabla \times \mathbf{F} &= \frac{1}{V} \sum \mathbf{n} \times \mathbf{F} \Delta S \\ &= \frac{1}{V} (\mathbf{n}_A \times \mathbf{F}_A + \mathbf{n}_B \times \mathbf{F}_B + \mathbf{n}_C \times \mathbf{F}_C + \mathbf{n}_D \times \mathbf{F}_D) \end{aligned} \tag{3.17}$$

The × operator here is the just the regular cross product operator, and can be written as (in two dimensions)

$$\mathbf{n} \times \mathbf{F} = (n_y F_z)\hat{x} - (n_x F_z)\hat{y} + (n_x F_y - n_y F_x)\hat{z} \tag{3.18}$$

In a uniform Cartesian grid, the curl operator also collapses to a centered second order finite difference curl operator, which would lead to a numerically divergence-less vector field (see Sec. 3.22 for more details).

3.4 Divergence Operator

The divergence operator is derived by a direct application of divergence theorem:

$$\int_V \nabla \cdot \mathbf{F}\, dV = \int_{\partial V} \mathbf{F} \cdot \mathbf{n}\, dS, \tag{3.19}$$

or in discrete space, it can be written as

$$\nabla \cdot \mathbf{F} = \frac{1}{V} \sum \mathbf{F} \cdot \mathbf{n} \Delta S \tag{3.20}$$

As is the case with the curl operator, the values of F the cell interfaces can be approximated through linear interpolation, using the solution at the cell in question, and its neighbors (see Eq. 3.16). This finite volume divergence operator also collapses to the centered 2^{nd} order divergence operator for a uniform Cartesian grid. The combination of the divergence operator and the curl operator in a uniform Cartesian grid produces a numerically divergence-free vector field, without having to stagger the solutions.

3.4.1 Divergence of a Curl in a Uniform Cartesian Coordinates

Consider the stencil shown in Fig. 3.2, which is a uniform Cartesian grid. For such grid, the finite volume operators for divergence, Eq. 3.20, and curl, Eq. 3.17, reduce to second order centered finite difference operators.

The curl of a vector, **A**, can be written as

$$\begin{aligned} \nabla \times \mathbf{A} = (\frac{Az_{i,j+1} - Az_{i,j-1}}{\Delta y})\hat{x} - (\frac{Az_{i+1,j} - Az_{i-1,j}}{\Delta x})\hat{y} + \\ (\frac{Ay_{i+1,j} - Ay_{i-1,j}}{\Delta x} - \frac{Ax_{i,j+1} - Ax_{i,j-1}}{\Delta y})\hat{z} \end{aligned} \tag{3.21}$$

Taking the divergence of Eq. 3.21, the following can be obtained

$$\begin{aligned} \nabla \cdot (\nabla \times \mathbf{A}) = \frac{\partial(\frac{Az_{i,j+1} - Az_{i,j-1}}{\Delta y})}{\partial x} - \frac{\partial(\frac{Az_{i+1,j} - Az_{i-1,j}}{\Delta x})}{\partial y} = \\ (\frac{Az_{i+1,j+1} - Az_{i-1,j+1} - Az_{i+1,j-1} + Az_{i-1,j-1}}{\Delta y \Delta x}) - \\ (\frac{Az_{i+1,j+1} - Az_{i+1,j-1} - Az_{i-1,j+1} + Az_{i-1,j-1}}{\Delta y \Delta x}) \end{aligned} \tag{3.22}$$

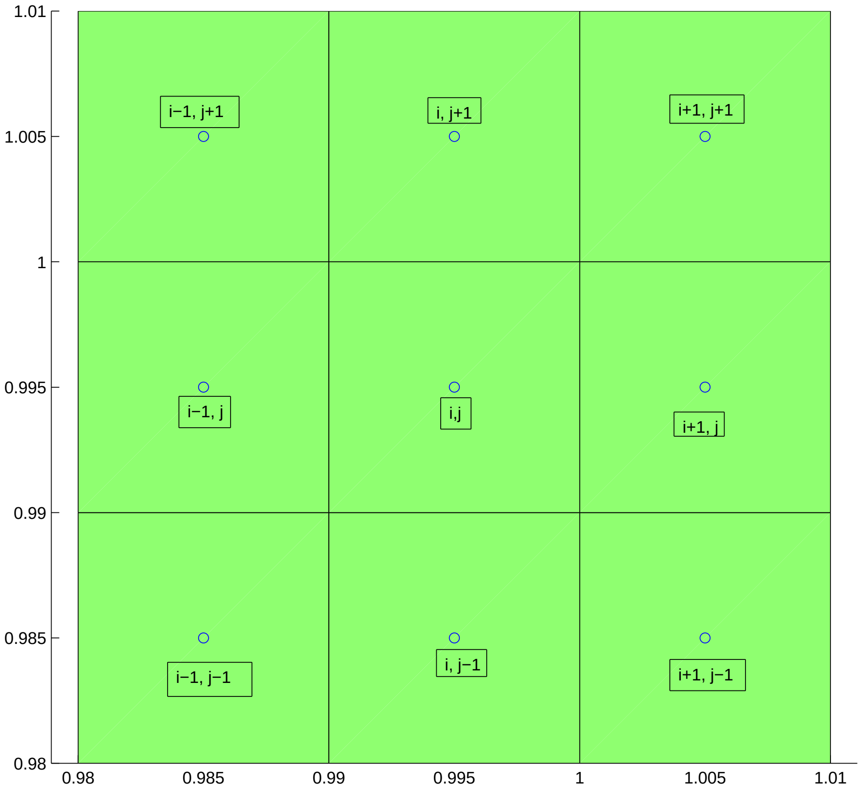

Figure 3.2: A 9-point uniform Cartesian stencil.

As can be seen from Eq. 3.22, the combination of the operators reduces to zero numerically, which is consistent mathematically with corresponding vector identities.

It is important to once again emphasize that, by using these operators, the magnetic field divergence constraint will necessarily be satisfied in a uniform Cartesian grid, without the complication of a staggered grid, which is the usual method to maintain the divergence constraint.

3.5 Gradient Operator

The fundamental theorem of Calculus applied to the gradient operator yields

$$\int\limits_{V} \nabla\Phi \, dV = \int\limits_{\partial V} \Phi\mathbf{n} \, dS \tag{3.23}$$

Once again, the operator directly requires the values of the solutions at the cell interfaces, and one would then need to approximate the values of the solutions at cell faces. The approximations are obtained through linear interpolation as written in Eq. 3.16. This operator also reduces to the centered second order finite difference approximation in the uniform Cartesian grid.

3.6 Poisson Solver

The Poisson's equations are solved by inverting the Laplacian operator as described in Sec. 3.2. Because the operator is perfectly linear, it can be written as a system of linear equations

$$A\mathbf{x} = \mathbf{b} \tag{3.24}$$

where A is the operator matrix, $\mathbf{x}$ is the solution to be solved, and $\mathbf{b}$ is the source term.

In the current solver, for 2-dimensional problems, $\mathbf{x}$ and $\mathbf{b}$ are stacked in a row-major order. The coefficients of matrix A are determined through grid coupling of a cell and its neighboring cells. A particular cell is coupled to at most eight of its neighbors, and a single row of matrix A has at most 9 elements, which makes A a large but sparse matrix. Through PETSc, several solvers can be employed to solve such a system of equations. For the purpose of this project, an external package, HYPRE, provides the best compromise of speed and accuracy, and it scales better than direct solvers such as SUPERLU. For more discussions about PETSc and its supported external packages, readers are invited to consult the PETSc manual (Ref. [19, 9]).

3.6.1 WARPX Implementation of the Poisson's Solver

The implementation of Poisson's solver is similar with that of the Laplacian operator. The problem is, once again, expressed as an $A\mathbf{x} = \mathbf{b}$ problem. To solve for $\mathbf{x}$, matrix A needs to be first inverted, and, as previously mentioned in Sec. 3.6, an external package PETSc is used for this purpose.

To fill in the coefficients of matrix A, similar techniques and implementations as described in Sec. 3.2.1 are used. The $MatSetValues$ command, which is a PETSc function, is used to set the coefficients of matrix A, which is a PETSc object. The following code fragment illustrates this implementations

```
MatSetValues(A,1,&l,1,&temp_col,&A0_P,INSERT_VALUES);CHKERRQ(ierr);
// put the coefficient A3 into the matrix (top stencil)
if (apordering[l-rstart]+_XCells < n)
{
  temp_col = apordering[l-rstart] + _XCells;
  AOApplicationToPetsc(ao, 1, &temp_col);

  MatSetValues(A,1,&l,1,&temp_col,&A3_P,INSERT_VALUES);CHKERRQ(ierr);

  // i+columns-1 (A4)
  if (((apordering[l-rstart]+_XCells)%_XCells)!=0)
  {
    temp_col = apordering[l-rstart]+_XCells-1;
    AOApplicationToPetsc(ao, 1, &temp_col);
    MatSetValues(A,1,&l,1,&temp_col,&A4_P,INSERT_VALUES);CHKERRQ(ierr);
  }

  // i+columns+1 (A2)
  if (((apordering[l-rstart]+_XCells+1)%_XCells)!=0)
  {
    temp_col = apordering[l-rstart]+_XCells+1;
    AOApplicationToPetsc(ao, 1, &temp_col);
    MatSetValues(A,1,&l,1,&temp_col,&A2_P,INSERT_VALUES);CHKERRQ(ierr);
  }
}

// put the coefficient A1 into the matrix (right stencil)
if (((apordering[l-rstart]+1)%_XCells)!=0)
{
  temp_col = apordering[l-rstart] + 1;
  AOApplicationToPetsc(ao, 1, &temp_col);

  MatSetValues(A,1,&l,1,&temp_col,&A1_P,INSERT_VALUES);CHKERRQ(ierr);
}

// put the coefficient A7 into the matrix (bottom stencil)
```

```
if (apordering[l-rstart]-_XCells >= 0)
{
  temp_col = apordering[l-rstart] - _XCells;
  AOApplicationToPetsc(ao, 1, &temp_col);

  MatSetValues(A,1,&l,1,&temp_col,&A7_P,INSERT_VALUES);CHKERRQ(ierr);

  // i-columns-1 (A6)
  if (((apordering[l-rstart]-_XCells)%_XCells)!=0)
  {
    temp_col = apordering[l-rstart]-_XCells-1;
    AOApplicationToPetsc(ao, 1, &temp_col);
    MatSetValues(A,1,&l,1,&temp_col,&A6_P,INSERT_VALUES);CHKERRQ(ierr);
  }

  // i-columns+1 (A8)
  if (((apordering[l-rstart]-_XCells+1)%_XCells)!=0)
  {
    temp_col = apordering[l-rstart]-_XCells+1;
    AOApplicationToPetsc(ao, 1, &temp_col);
    MatSetValues(A,1,&l,1,&temp_col,&A8_P,INSERT_VALUES);CHKERRQ(ierr);
  }
}

// put the coefficient A5 into the matrix (left stencil)
if (((apordering[l-rstart])%_XCells)!=0)
{
  temp_col = apordering[l-rstart] - 1;
  AOApplicationToPetsc(ao, 1, &temp_col);

  MatSetValues(A,1,&l,1,&temp_col,&A5_P,INSERT_VALUES);CHKERRQ(ierr);
}
```

The *apordering* variable stacks the multidimensional solution in a vector using the row major order. It is also a PETSc object whose function is to ensure contiguous data structure when the domain is decomposed into several pieces. This PETSc feature contains mapping information between the natural application ordering, which might not necessarily result in contiguous ordering in parallel, and the contiguous PETSc ordering. Please consult Sec. 2.3.1 of the PETSc manual, Ref. [19], for more information about application ordering.

Chapter 4

ELECTROMAGNETIC SOURCE TERMS TEST CASES

The goal of this project is to develop electromagnetic solvers that self-consistently include the divergence constraints (Eq. 2.11 and 2.12) into the update equations so that numerical, non-physical divergence error can be reduced.

The eventual goal is then to couple the proposed electromagnetic solvers with the two-fluid plasma model as described in Chapter 2. The coupling between the Euler equations and the electromagnetic system happens through the source terms. Therefore, it is very important that the source terms be accurately handled.

4.1 Grid Definition

The algorithms presented in Chapter 3 are valid for structured general quadrilateral grid. The following grid definition is used to assess the capability of the algorithms in handling general geometry

$$\begin{aligned} x_p &= x_c + \alpha \Delta x \cos\left[\pi b_1 (x_c - c_1)\right] \cos\left[2\pi b_2 (y_c - c_2)\right] \\ y_p &= y_c + \alpha \Delta y \cos\left[2\pi b_1 (x_c - c_1)\right] \cos\left[\pi b_2 (y_c - c_2)\right] \end{aligned} \tag{4.1}$$

where b_1, c_1, b_2,and c_2 are variables that depend on the size and location of the domain, to ensure that the mesh is appropriately skewed, and that the boundaries are rectangular. Δx and Δy are the spacings in the computational domain, which is a uniform Cartesian grid. Likewise, x_c and y_c are the x and y coordinates in the computational domain.

Here, α, is a parameter that defines the skewness of the mesh. A grid with $\alpha = 4$ is shown in Fig. 4.1, which is the grid that is used for the simulations done in this Chapter. Analysis on how the amount of grid distortion affects the solution is presented in Sec. 4.7.

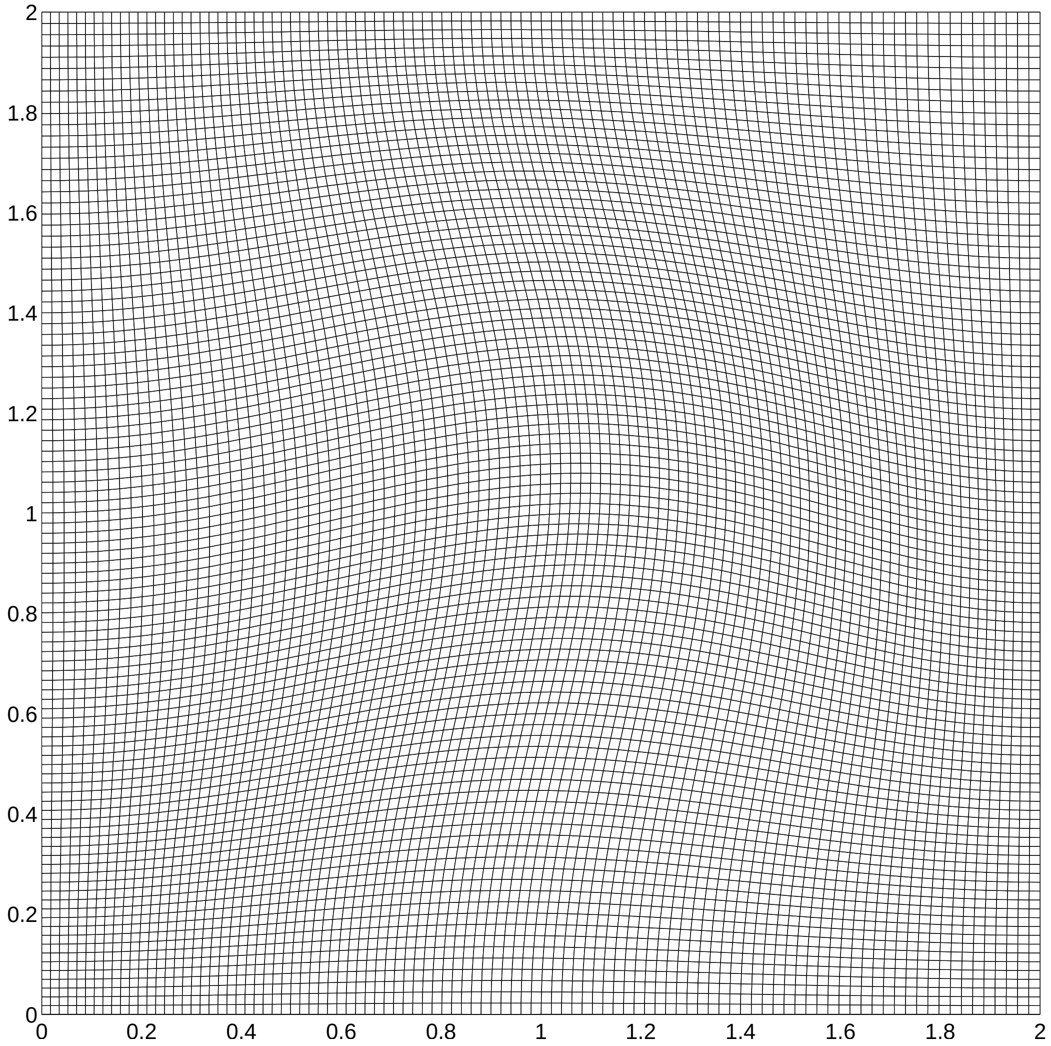

Figure 4.1: Skewed mesh with twist applied in the middle. The grid shown in this figure has $\alpha = 4$. Unless stated otherwise, this is the grid that is used for the majority of simulations conducted within this Chapter.

4.2 Source Terms Handling

The wave propagation method, which handles the Perfectly Hyperbolic Maxwell system, employs a source splitting technique. The homogeneous part is first solved using first order Godunov's method and second order flux correction

$$\frac{\partial Q}{\partial t} + \nabla \cdot F = 0 \tag{4.2}$$

Then, the source terms are updated separately by solving a system of ordinary differential equations.

$$\frac{d\mathbf{Q}}{dt} = \mathbf{s} \tag{4.3}$$

Any standard ODE solver can be then used to solve this system. For the purpose of this project, a fourth order Runge-Kutta scheme is used.

The potential formulation incorporates the source terms directly into the system. The model equation for the potential formulation is the 2^{nd} order wave equation with a source. It can be written as

$$\frac{\partial^2 u}{\partial t^2} - c^2 \nabla^2 u = S \tag{4.4}$$

where c denotes the wave speed (or the speed of light in the case of Maxwell's equations). The update formula is then

$$u^{n+1} = (S + c^2 \nabla^2 u)\Delta t^2 + 2u^n - u^{n-1} \tag{4.5}$$

which calculates cell-averaged solution of u for the next time step. The update formula is a three step process because it involves solutions from three time levels.

At $t = 0$, the data at time level -1 are not available, and thus, have to be extrapolated using the initial time derivative of the solution. In the case the initial time derivative is zero, then $u^{-1} = u^0$.

4.3 Open Boundary Conditions

This section describes two possible methods to truncate the infinite physical domain of the problem. The goal is to come up with a method to suppress artificial reflections off the computational boundaries so that solutions that agree well with what would be obtained in an infinite domain can be obtained.

4.3.1 Copy Boundary Conditions

The copy boundary conditions approximate the open boundaries by copying out the values of the outer most cells to the ghost cells to be used for computations. While this method is simple, and should result in loss of some information in multidimensional problems, the approximation has been found to be sufficient for most hyperbolic systems of equations (Sec. 21.8.5 in Ref. [4]), solved using the wave propagation method.

4.3.2 Second Order Mur Radiation Boundary Conditions

While copy boundary conditions work well with the wave propagation method, Maxwell's equations expressed in their potential formulation, which is a system of second order wave equations, require imposing source-free one way wave equations at the boundaries to approximate open infinite boundaries.

For two-dimensional rectangular boundaries, the following boundary conditions for the four sides of the domain are obtained (Ref. [20])

$$\frac{1}{c}\frac{\partial^2 u}{\partial x \partial t} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \frac{1}{2}\frac{\partial^2 u}{\partial y^2} \tag{4.6}$$

$$\frac{1}{c}\frac{\partial^2 u}{\partial x \partial t} = -\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} + \frac{1}{2}\frac{\partial^2 u}{\partial y^2} \tag{4.7}$$

$$\frac{1}{c}\frac{\partial^2 u}{\partial y \partial t} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \frac{1}{2}\frac{\partial^2 u}{\partial x^2} \tag{4.8}$$

$$\frac{1}{c}\frac{\partial^2 u}{\partial y \partial t} = -\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \frac{1}{2}\frac{\partial^2 u}{\partial x^2} \tag{4.9}$$

for the left, right, bottom and top boundaries respectively.

This particular implementation assumes rectangular boundaries. It is not yet clear if this implementation can be readily be extended to non-Cartesian boundaries, a topic of which is still an active area of research (Ref. [21]).

4.4 Current Source Terms

The current source term appears in the Ampere's circuital law (Eq. 2.10), and acts as a source term for the electric field. The electric field, then, in turns, produces magnetic field through Faraday's law (Eq. 2.9).

The test is conducted on the skewed grid with α set to 4 (see Eq. 4.1). This grid is as shown in Fig. 4.1. The current is initialized in the middle of the domain, and is defined by its magnitude and its radius (a magnitude of one, and a radius of 0.075 are chosen). It has a circular shape, is shown in Fig. 4.2, and points in the z-direction. Mathematically, the current can be written as

$$J_z = \begin{cases} 1 & r < 0.075 \\ 0 & r \geq \ 0.075 \end{cases} \tag{4.10}$$

where

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{4.11}$$

where, x_c and y_c are the x and y locations at which the current pulse is centered, and r_c is the radius of the pulse.

The Coulomb and the Lorenz gauge formulations will produce the same solutions because the scalar potential is always zero for both cases, due to the conservation of charge (Eq. 4.18), which dictates that a solenoidal current source will result in a zero time-derivative for the charge density. Because the scalar potential is set to zero initially, then it will remain zero at all times, thus the Coulomb and the Lorenz solutions will have the same solutions.

Open boundary conditions are implemented for this problem. The copy boundary conditions, as described in Sec. 4.3.1, are used for PHM, and second order Mur radiation boundary conditions, as described in Sec. 4.3.2, are employed for the potential based solvers.

The potential and the PHM solvers produce qualitatively similar solutions (see Fig. 4.3 and Fig. 4.4 for comparisons). However, there is no known analytical solution for the problem. The steady state solution can then be approximated by solving the elliptic equation (see Fig. 4.5)

$$\nabla^2 \mathbf{A} = -\mu_0 \mathbf{J} \tag{4.12}$$

From the figures, it can be seen that the solutions start from the middle, and propagate outward. Eventually, the solutions approach the steady state solutions at later times.

To quantify the solutions produced by the proposed different solvers, the divergence error for each solver is computed and compared in Sec. 4.4.1.

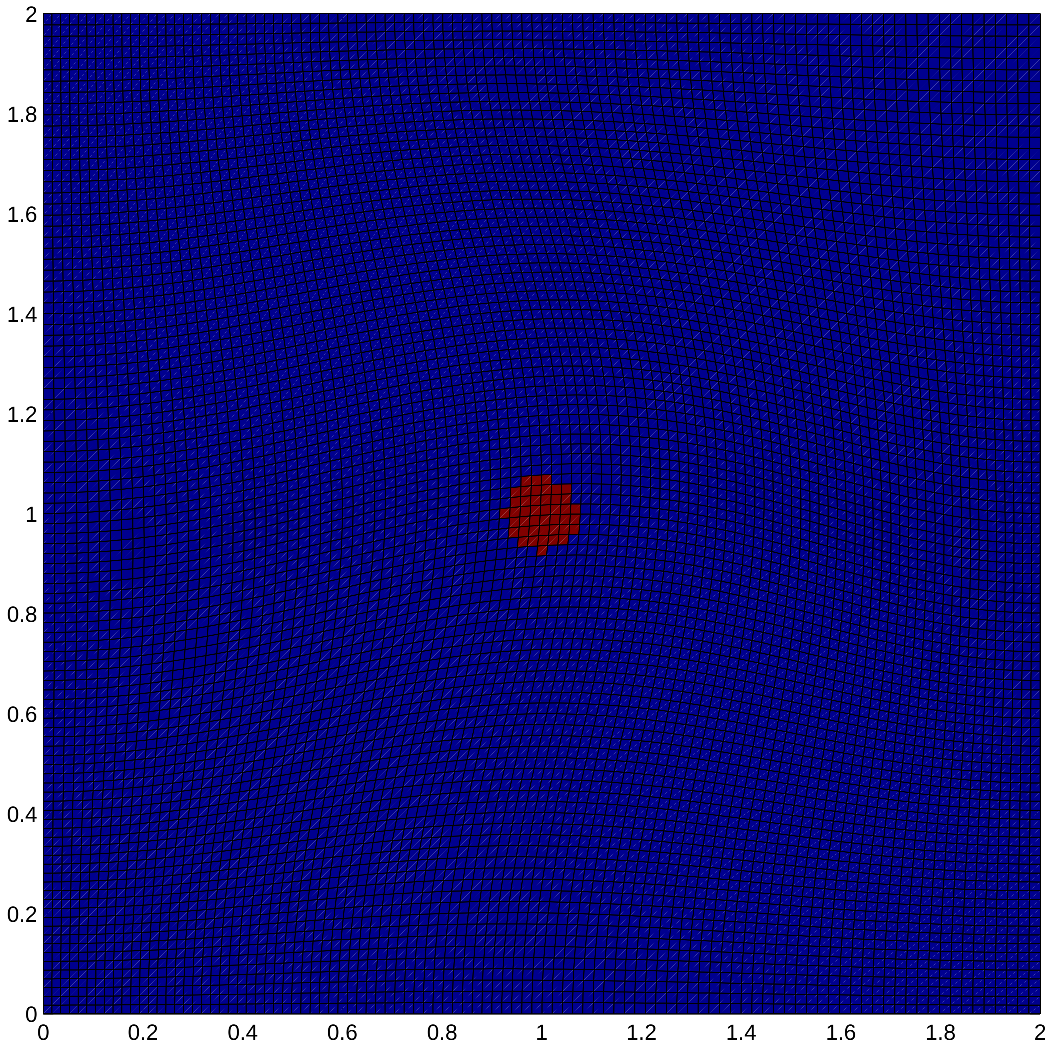

Figure 4.2: Current is initialized in the middle of the domain. The current is set to zero outside of the circular pulse of some radius, which, in this case, is 0.075.

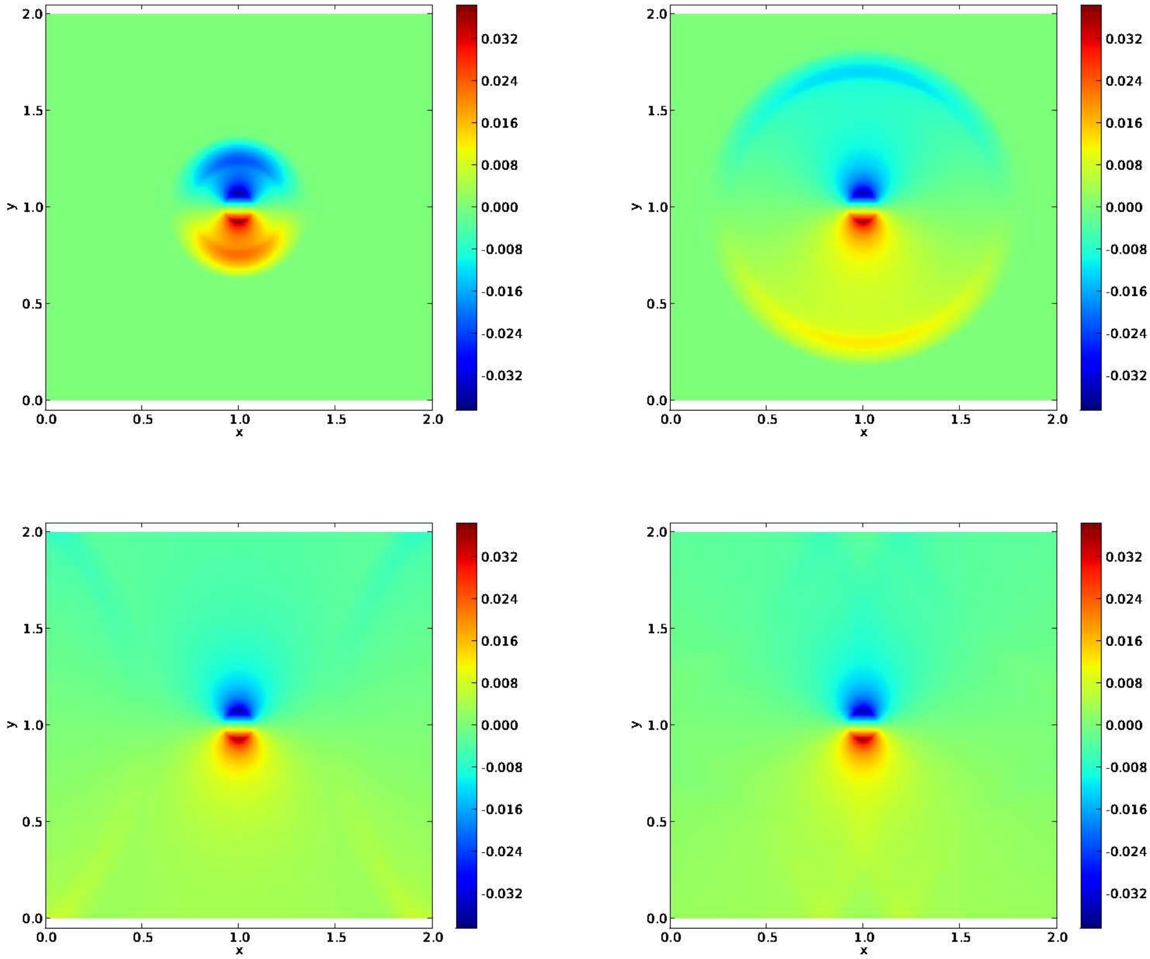

Figure 4.3: The evolution of the magnetic field in the x direction solved on the skewed grid using the PHM solver with $\gamma = 1$. Going from left to right, top to bottom, the evolution is captured at t = 0.3, t = 0.75, t = 1.5 and t = 2.1. While the transient wave seems to have left the domain at t = 1.5, some reflections can be observed coming off the boundaries. For this problem, the open boundaries are approximated by copying out the solutions of the outermost cells to the ghost cells.

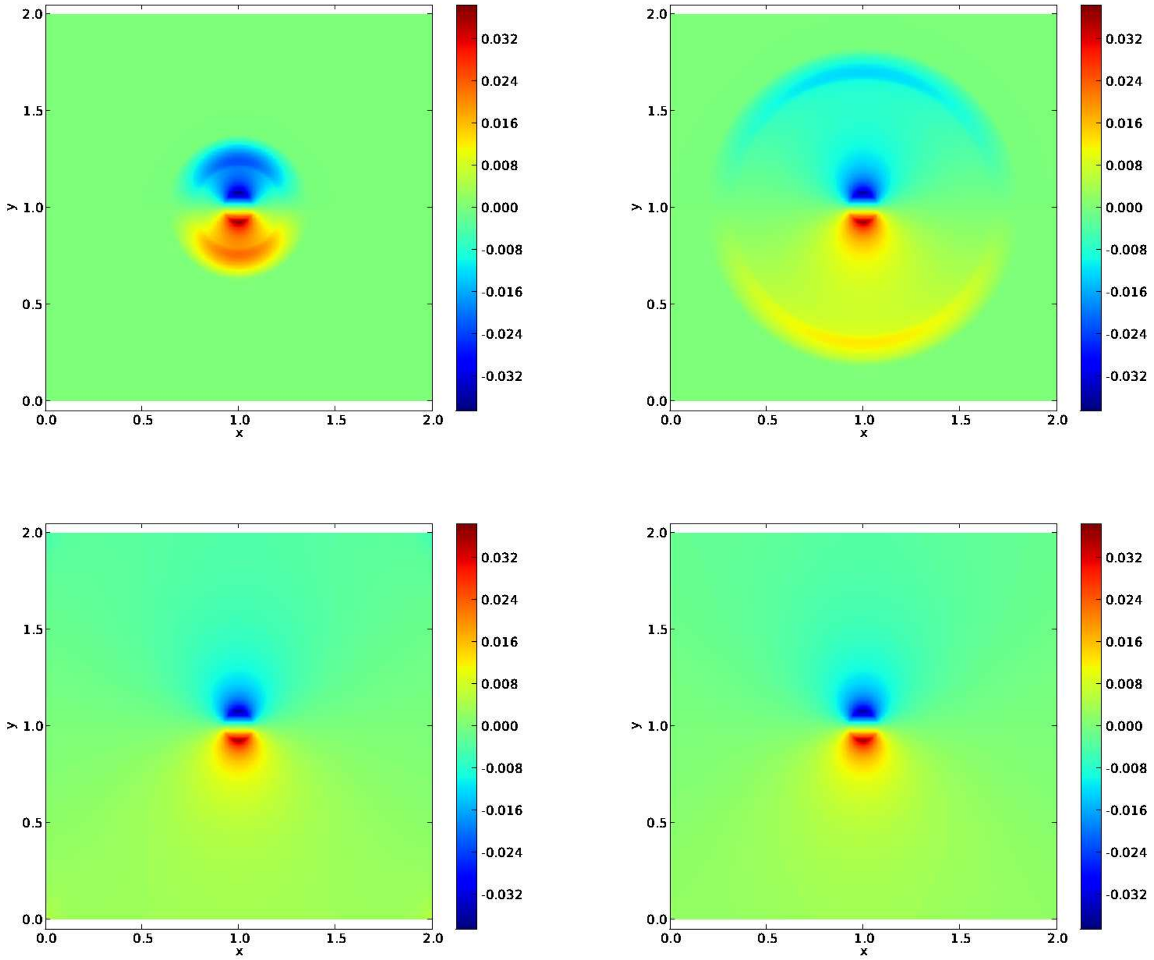

Figure 4.4: The evolution of the magnetic field in the x direction solved on the skewed grid using the potential solver. Going from left to right, top to bottom, the evolution is captured at $t = 0.3$, $t = 0.75$, $t = 1.5$ and $t = 2.1$. Second order Mur open boundary formulation is implemented, and it seems to work well. At $t = 1.5$, no visible boundary reflections are observed.

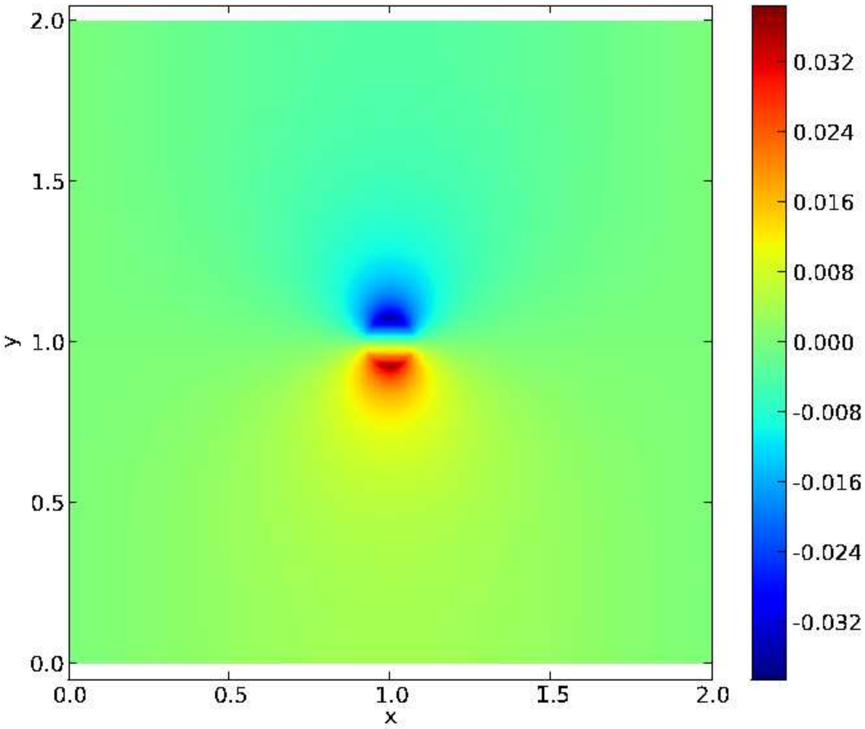

Figure 4.5: The magnetic field in the x direction solved on the $\alpha = 4$ skewed grid. Elliptic steady state solution. The steady state solution looks very similar with the solutions of the PHM and Lorenz gauge formulation solvers.

4.4.1 Divergence Error Comparison for the Current Problem

The divergence error for the magnetic field calculation is computed. To compare how well each method preserves the divergence constraint, the l_2-norm of the error is computed using the following formula

$$\|\nabla \cdot \mathbf{B}\|_2 = \sqrt{\sum (\nabla \cdot \mathbf{B}\ \Delta V)^2} \tag{4.13}$$

where ΔV is the volume of a cell. The summation in Eq. 4.13 is carried throughout the domain, which results in a single number for each time-step.

Fig. 4.6 and Fig. 4.7 show the l_2-norm of the divergence of the magnetic field (Eq. 4.13) on the skewed grid, solved using the PHM and the potential solvers respectively. From the plots, one can observe how the divergence error reduces as the grid resolution is increased.

The PHM exhibits some oscillations before the error settles down. The potential solution initially increases with time, then it overshoots. The error then decreases, until finally it plateaus. The potential formulation has divergence error about 3 to 4 times smaller than the divergence error of the PHM system.

When the resolution is doubled, the error of the PHM solution reduces by a factor of about 2 (see Fig. 4.6), implying about first order convergence of the solution. While it is expected that the method is second order accurate, several factors might have contributed to the reduced order of accuracy of the results.

One possible explanation is artificial reflection. As described in Sec. 4.3.1, copy boundary condition is a simple way to approximate open boundary condition, and while it works really well given its simplicity, using it may lead to a potential reflection problem.

Another possible reason for the reduced order of accuracy is the way source terms are treated for the wave propagation method. To make the method officially second order accurate, Strang splitting is required, whereas, Godunov source splitting is only formally first order accurate.

For the potential solver, the error decreases by a factor of about 2.8 (see Fig. 4.7) when the resolution is doubled. As mentioned, the potential formulation cannot be solved directly by the wave propagation method because it is a system of second order wave equations, and finite volume methods dealing with the derivative operators are developed (as described in

Chapter 3).

While these methods are expected to be second order accurate, it is not clear how well they would behave when dealing with non-smooth solutions, as is the case with the current density test case, with a discontinuous source term. This might have contributed to the slightly reduced order of accuracy of the solution.

4.4.1.1 The Effects of Varying γ on Divergence Error for the Current Test Case

To isolate the effects of varying values of γ, the divergence error of the magnetic field using different γ values are compared. The resolution is fixed at 200-by-200 in both the x and the y directions. The result is shown in Fig. 4.8.

As can be seen from the plot, increasing the γ value does not significantly affect the solution. Going from no correction to a value of γ of one, however, proves beneficial. With no correction, the solution increases in a linear fashion until around t = 0.5 seconds, then the growth rate starts to decrease. At this point, the error is about five to six times larger than the corrected solutions. However, among the corrected solutions, the values of γ do not seem to affect the solution by much.

Early in the simulation, the solutions behave as expected, with smaller error for higher γ. As time passes, there are times when the error for each value of γ is greater than the two others, and there does not seem to be any recognizable pattern for the error as a function of γ.

4.5 Steady State Charge Density Source Term Test

The current source term test case can also be re-used to test how the proposed algorithms handle charge density. Instead of the current, the charge density is initialized. Similar to Eq. 4.10, the charge density can be written as

$$\rho_c = \begin{cases} 1 & r < r_c \\ 0 & r \geq \ 0.075 \end{cases} \tag{4.14}$$

where

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{4.15}$$

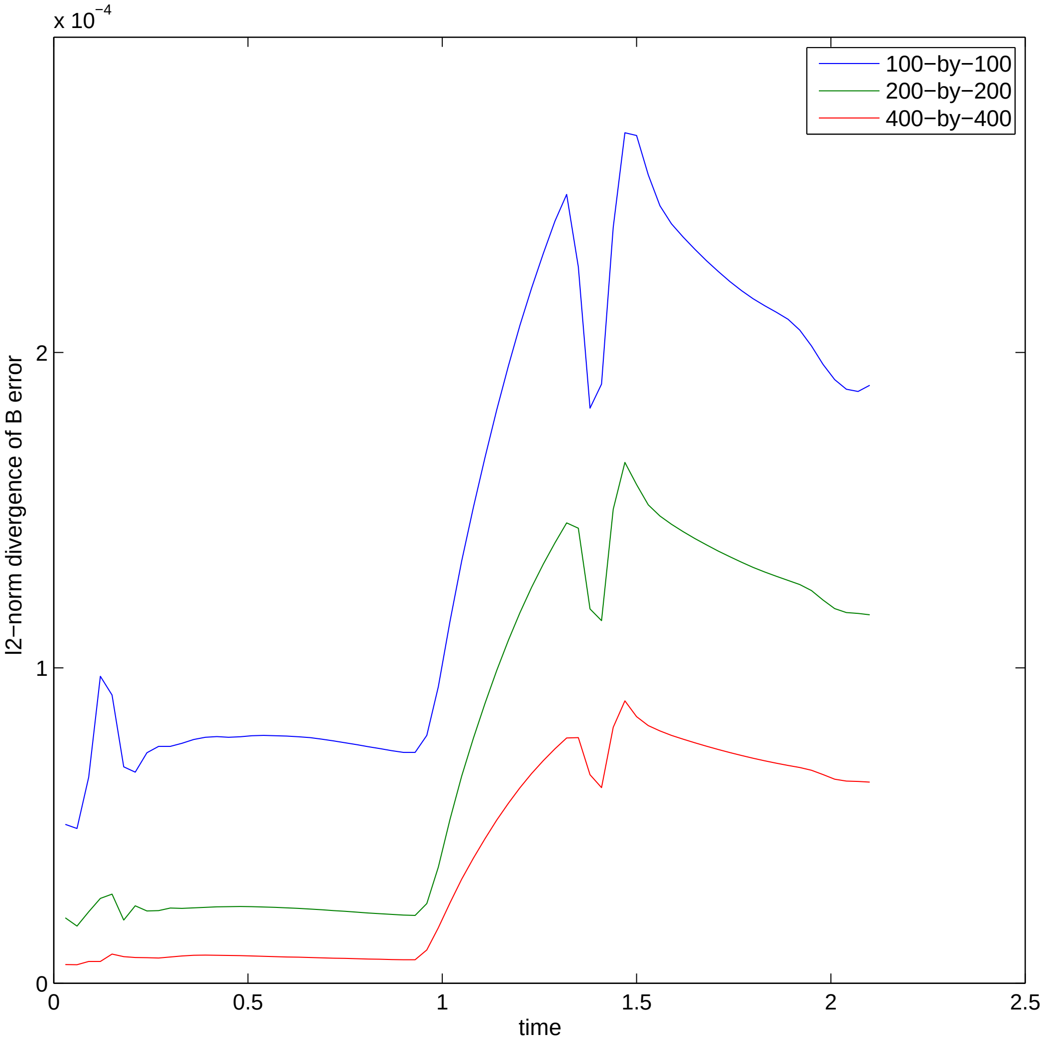

Figure 4.6: l_2-norm of the divergence error of the magnetic field as a function of time for the PHM solution on a skewed grid. The errors for three different grid resolution are given. The error, in general, seems to reduce by a factor of about 2 when the resolution is doubled.

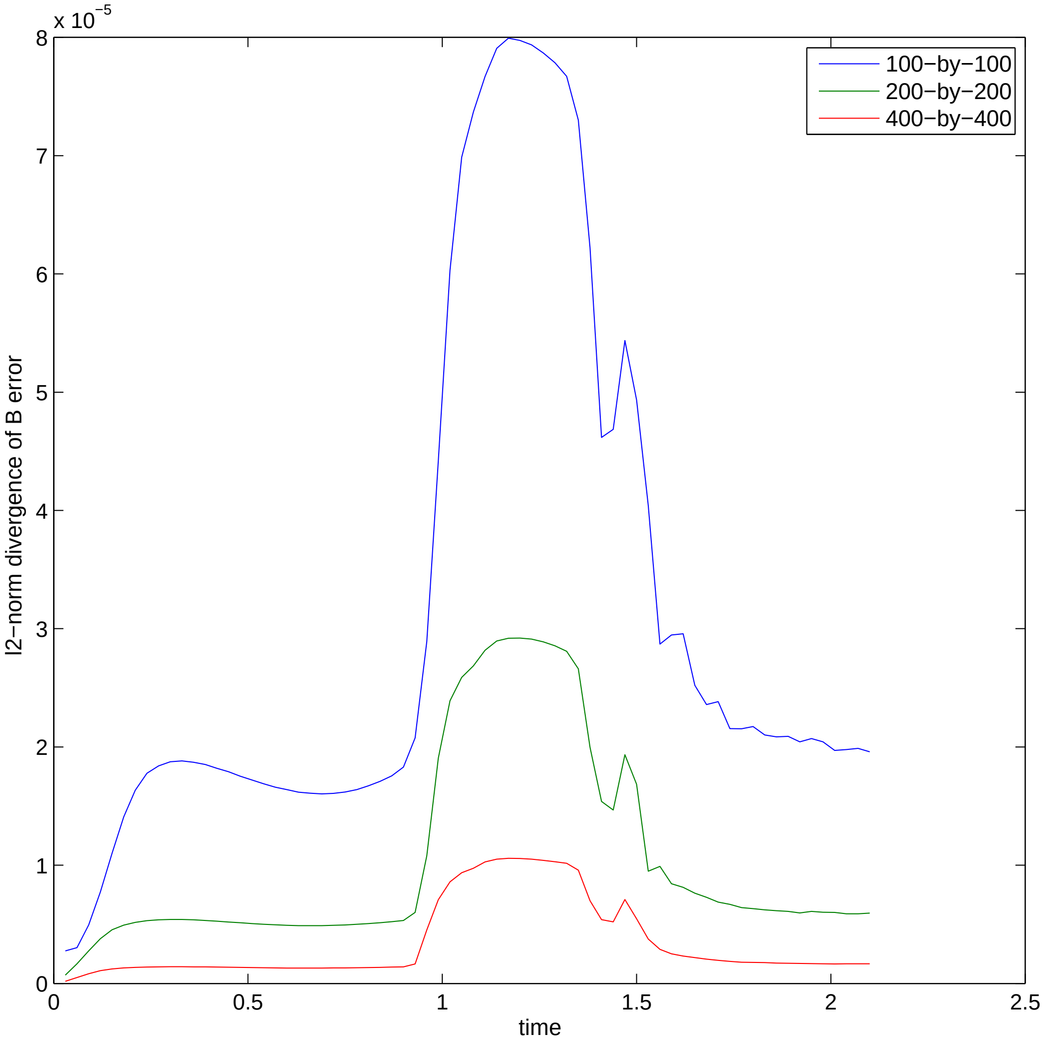

Figure 4.7: l_2-norm of the divergence error of the magnetic field as a function of time for the potential solution on a skewed grid. The error here is smaller than the error of the PHM solver. The error also reduces by a factor of about 2.8 when the resolution is doubled, implying close to a second order accuracy.

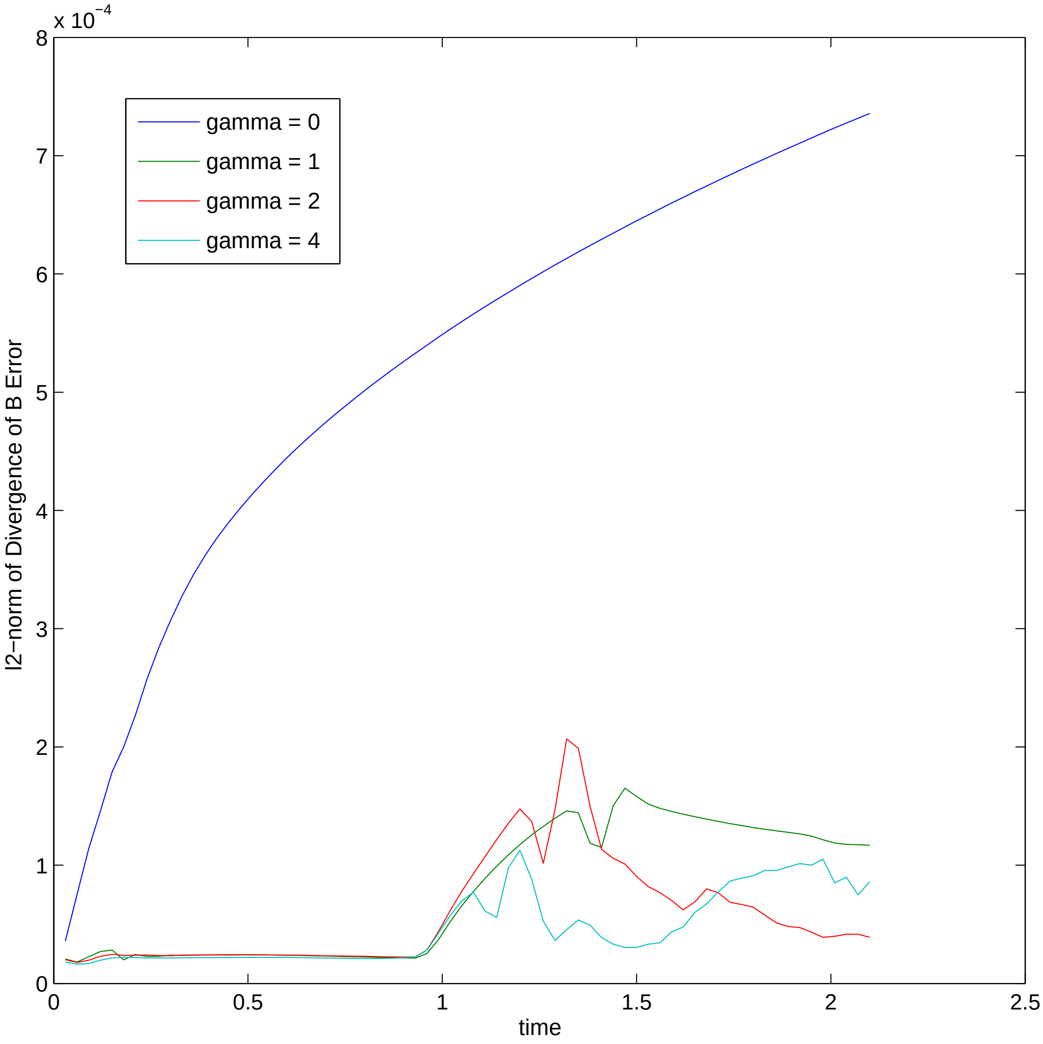

Figure 4.8: The l_2-norm error of the divergence error of the magnetic field when using different values of γ for PHM. The solution with no correction has significantly larger error than the other solutions. However, increasing the value of γ above 1 does not seem to significantly reduce the error.

where, once again, x_c and y_c are the x and y locations at which the charge density pulse is centered, and r_c is the radius of the pulse.

Physically, initially setting only the charge density but not the electric field violates the electric field (Eq. 2.11) divergence constraint. Thus, it is expected that an electromagnetic solver based just on the two curl Maxwell's equations (Eq. 2.9 and Eq. 2.10) to perform poorly. In fact, without an initialization of the electric field, the charge density source term does not matter, because it is never directly included in the system. Thus, the resulting electric field is always zero, which is incompatible with the presence of the charge density.

The PHM and the potential-based solvers should perform better, because the charge density is explicitly included in the equation systems. Similar to the previous finding, the solutions should start as transient solutions, with an error wave propagating from the middle of the domain. They should then eventually approach the steady state solutions.

As can be seen from Fig. 4.9 and Fig. 4.10, the solutions produced by the PHM and Lorenz-gauge based solvers propagate outward from the middle of the domain, and, at some later time, seem to approach steady state. These solutions look at least qualitatively similar with the solution produced by the Coulomb-gauge based solver, which represents the steady state solution (see Fig. 4.11).

To confirm, the l_2-norm of the difference in the magnitude of the electric field between the transient solutions (produced by the PHM and the solver based on the Lorenz gauge formulation) is computed, and it is found that the transient solutions do approach the steady state solutions (see Fig. 4.12). More detail analysis on this finding can be found in Sec. 4.5.1.

4.5.1 Divergence Error Comparison for the Steady State Charge Density Test Case

The l_2-norm for the divergence error of the electric field is computed by applying the following formula

$$\|\nabla \cdot \mathbf{E} - \frac{\rho_c}{\epsilon_0}\|_2 = \sqrt{\sum([\nabla \cdot \mathbf{E} - \frac{\rho_c}{\epsilon_0}]\ \Delta V)^2} \qquad (4.16)$$

where ΔV is the volume of a cell. The summation in Eq. 4.16 is carried throughout the domain, which results in a single number for each time step.

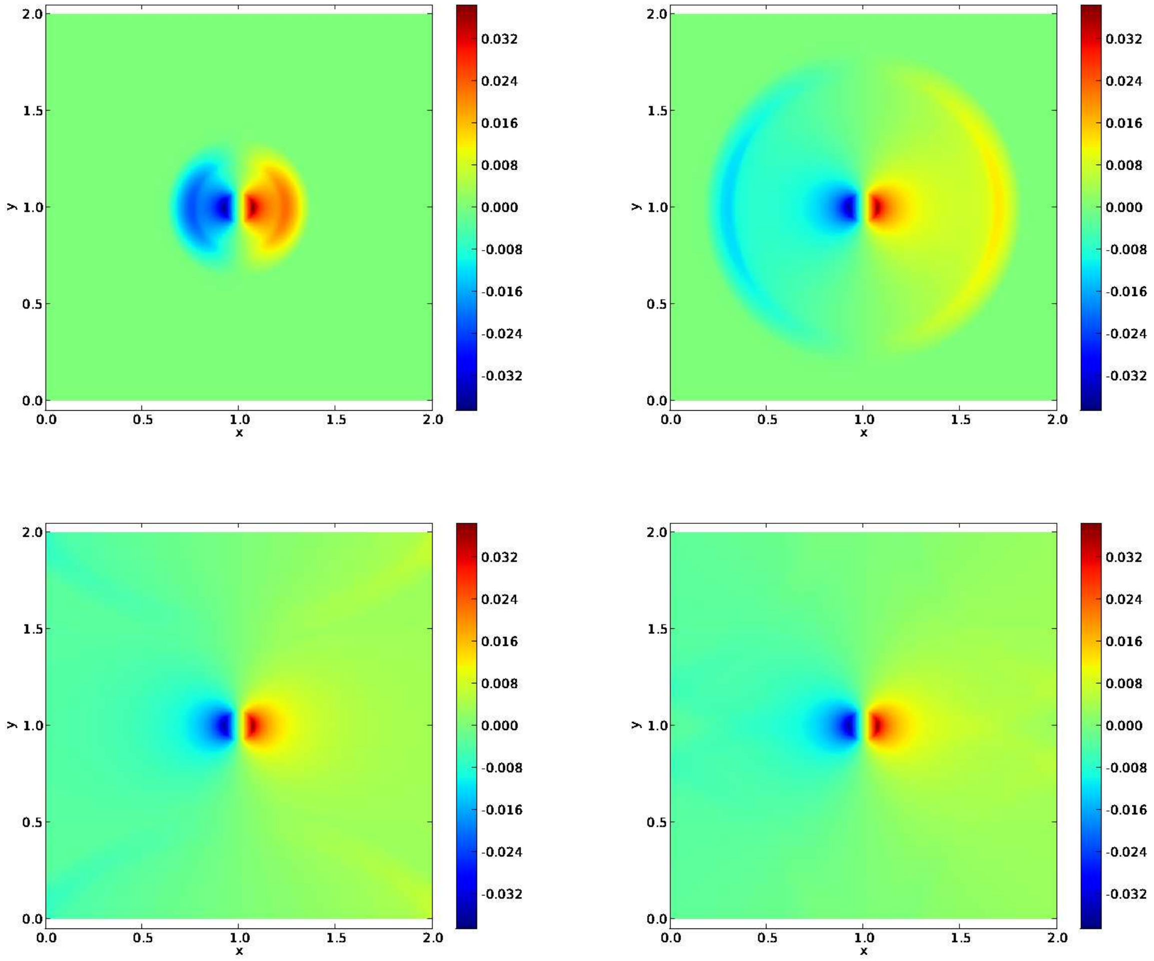

Figure 4.9: The evolution of the electric field in the x direction solved on the skewed mesh using PHM with $\chi = 1$. Going from left to right, top to bottom, the evolution is captured at t = 0.3, t = 0.75, t = 1.5 and t = 2.1. Similarly to the solution to the current density test case, reflections off the boundaries are visible at t = 1.5, when the transient wave has supposedly left the domain.

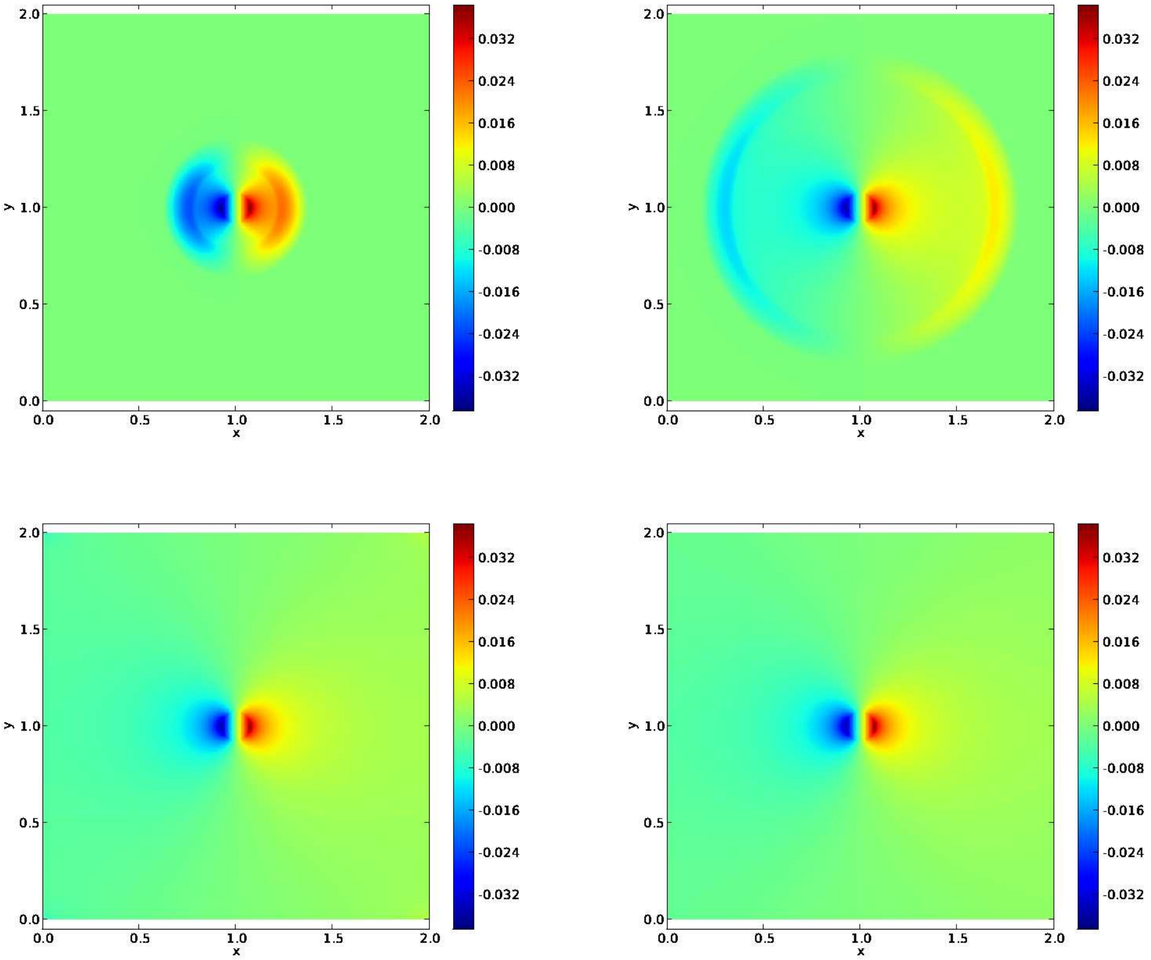

Figure 4.10: The evolution of the skewed field in the x direction solved on the skewed grid using the Lorenz gauge formulation. Going from left to right, top to bottom, the evolution is captured at t = 0.3, t = 0.75, t = 1.5 and t = 2.1. Second order Mur open boundary formulation is implemented, and it seems to work well. At t = 1.5, no visible boundary reflections are observed.

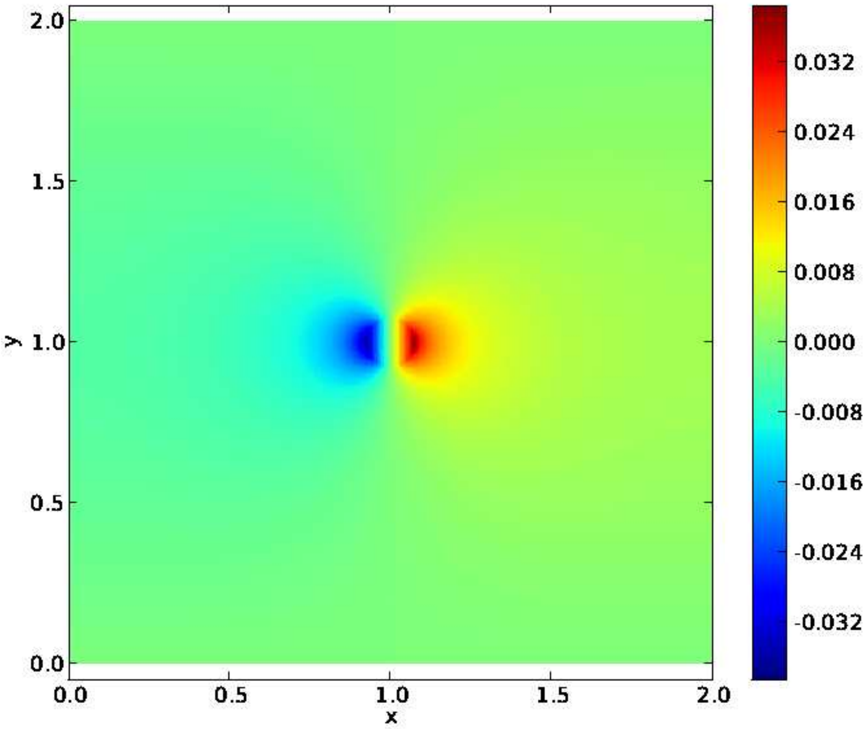

Figure 4.11: The steady solution of the electric field for the steady state charge density problem, solved on the skewed grid using Coulomb gauge formulation. The solution is the electrostatic steady state solution to the Poisson's equation of the scalar potential (Eq. 2.21)

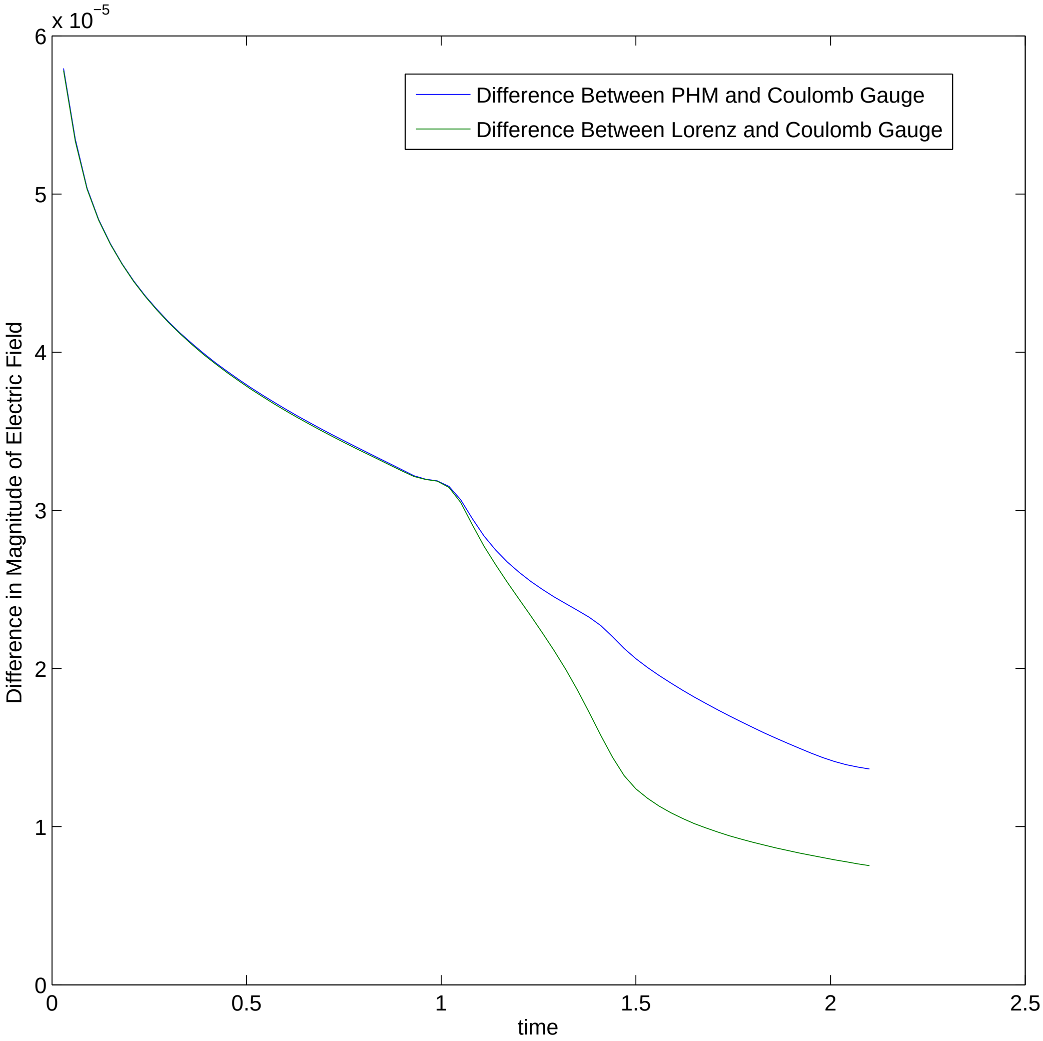

Figure 4.12: Comparison of the l_2-norm of the magnitude of the electric field of the numerical steady state (Coulomb gauge), and the transient (Lorenz gauge and PHM) solutions. Before $t = 1$, the two transient solutions seem to be about identical. For t greater than 1, the Lorenz gauge solution is approaching the Coulomb gauge solution faster, which is consistent with the previous analysis of the divergence error.

The normalization employed here is slightly different compared than the one used by Munz for this same problem [16]. However, Munz normalization can only be applied directly to the PHM system, and for the purpose of this study, Eq. 4.16 is the more proper way to normalize the error so that direct comparisons among the various solvers can be made.

Comparing the divergence error of the electric field, it can be seen that the error decreases by a factor of about 2, when the resolution is increased by about the same factor. Once again both the PHM and the potential solvers fail to achieve second order of accuracy. As described in Sec. 4.4.1, factors such as the accuracy of the boundary conditions, the smoothness of the solutions, source term treatment, and any combinations thereof might have contributed to the less than expected order of accuracy.

In both the PHM, and the Lorenz-gauge test cases, the errors eventually asymptote to single values, indicating that the solution approaches steady state (see Fig. 4.13 and Fig. 4.14).

This is especially apparent for the solver based on the Lorenz gauge formulation (Fig. 4.16), where the divergence error seems to approach the steady state error of the Coulomb gauge based solver, which is consistent with the previous comparison between the transient and the steady state solutions (see Fig. 4.12).

The l_2-norm of the difference between the electric field magnitude solutions is compared, and it is found that the Lorenz gauge solution approaches faster to the Coulomb gauge solution. The two lines are initially really close to each other. At $t \approx 1$, however, the two lines start to diverge, with the Lorenz gauge solution approaching faster to the Coulomb gauge solution.

These differences can be attributed to the accuracy of the boundary conditions. At $t \approx 1$, the error wave hits the boundaries, and, as can be observed from the bottom left plots of Fig. 4.3, some reflections are still visible in the PHM solutions, even after the error wave seems to have left the domain. The reflections are less visible in the potential solution, as can be seen from Fig. 4.4.

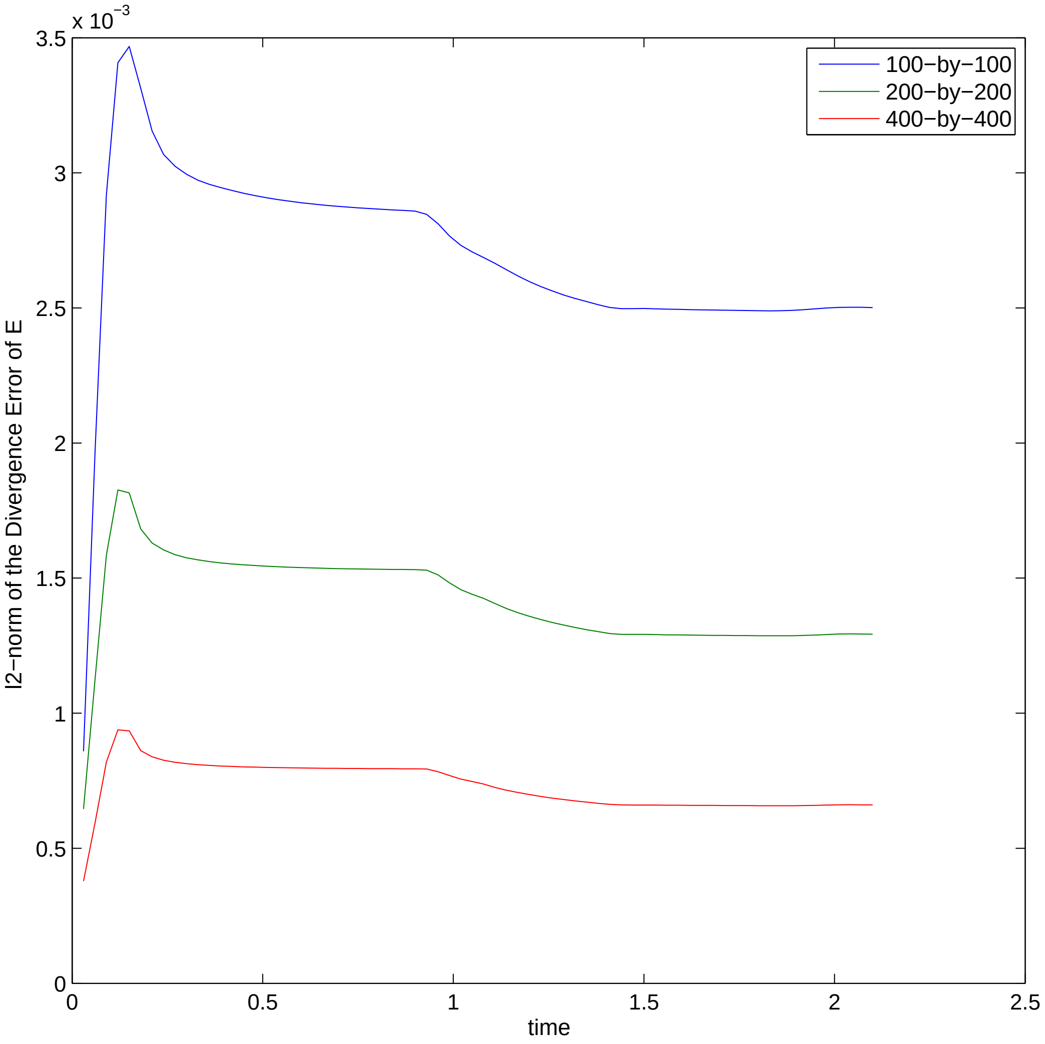

Figure 4.13: l_2-norm of the electric field divergence error for the PHM solver for the steady state charge density problem on a skewed grid. The errors for three different grid resolution are given. The error, in general, seems to reduce by a factor of about 2 when the resolution is doubled.

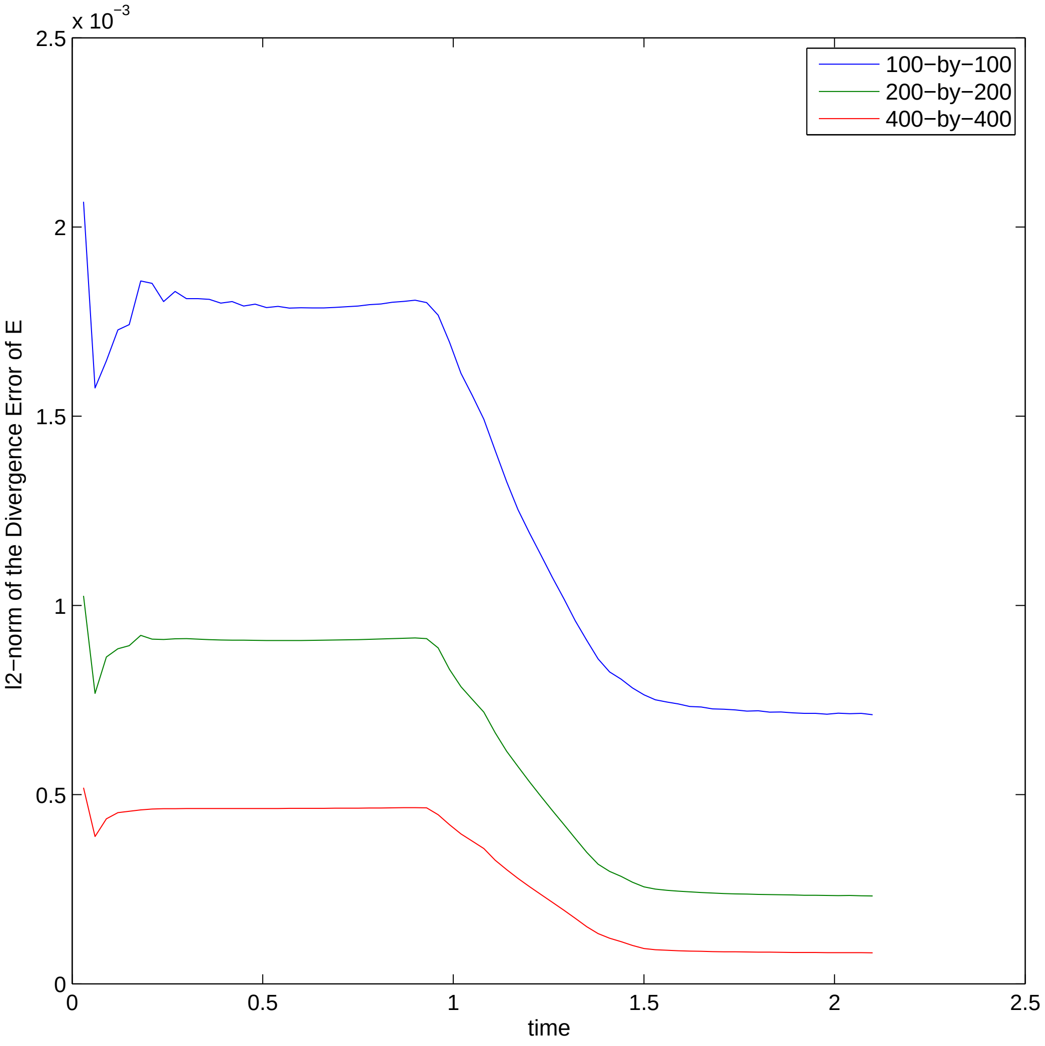

Figure 4.14: l_2-norm of the electric field divergence error for the Lorenz gauge-solver solver for the steady state charge density problem. The errors for three different grid resolution are given. The error, in general, seems to reduce by a factor of about 2.8 when the resolution is doubled.

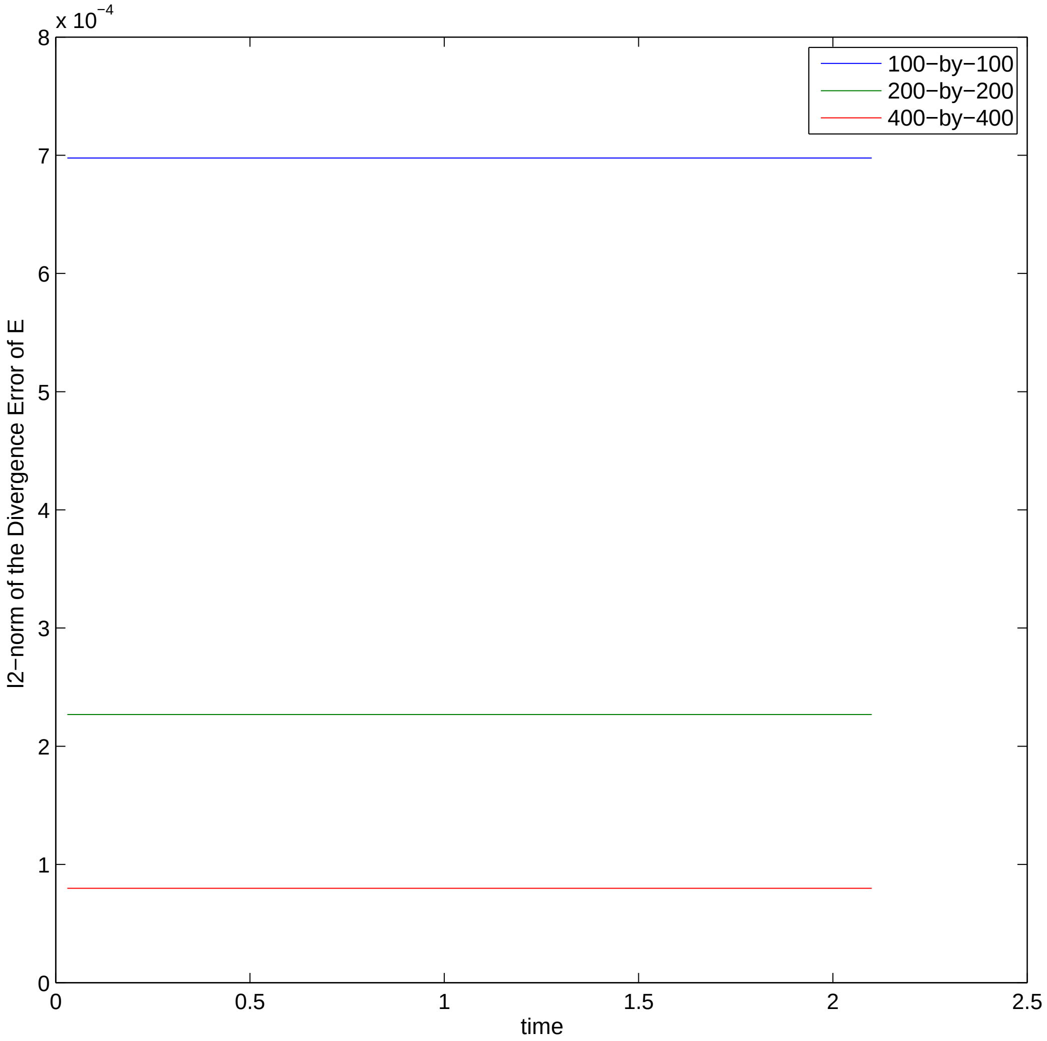

Figure 4.15: l_2-norm of the electric field divergence error for the Coulomb gauge-solver solver for the steady state charge density problem.

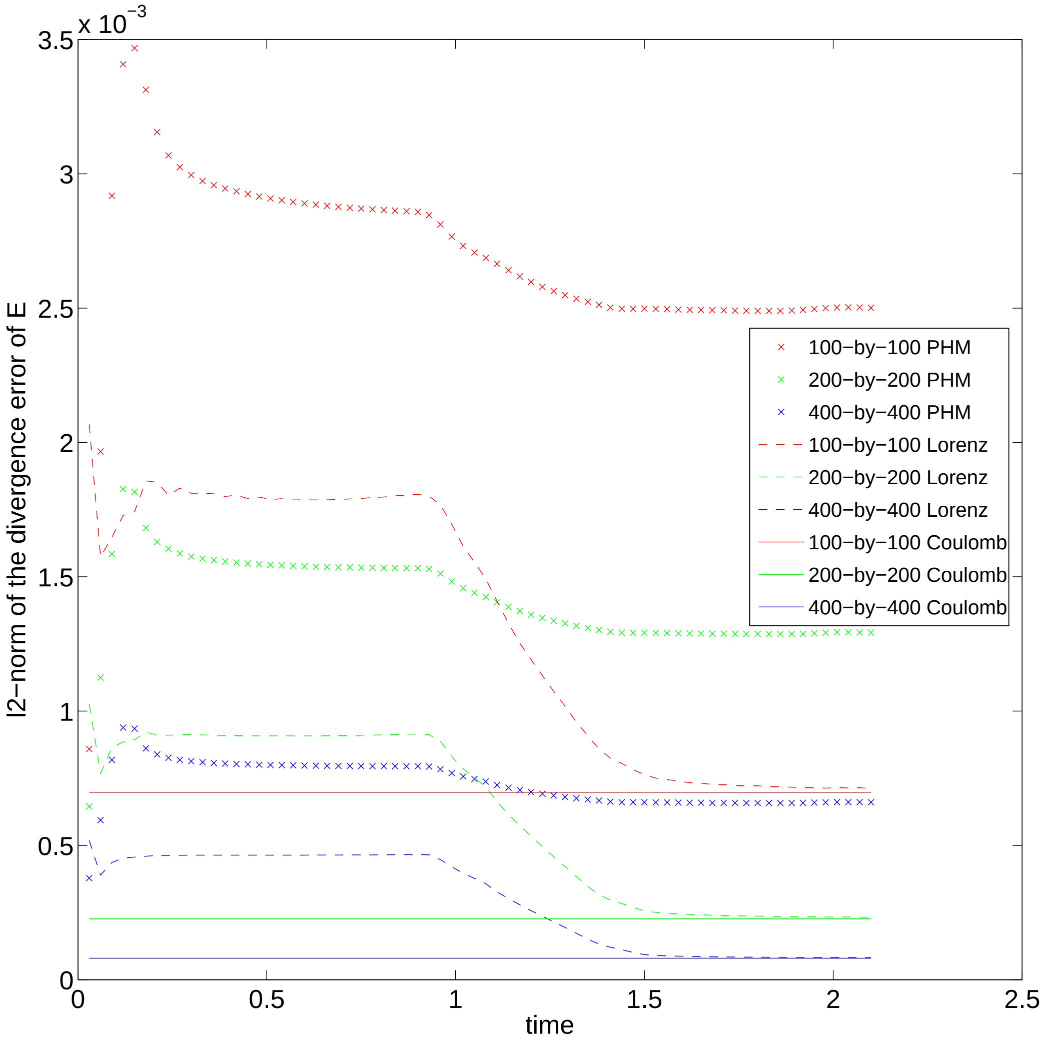

Figure 4.16: Comparison of the l_2-norm of the electric field divergence error for the steady state charge density problem. This figure shows that the potential based solvers have smaller divergence error than PHM. The Lorenz gauge based solver has error that is approaching that of the Coulomb gauge based solver, implying that the steady state solution of the Lorenz gauge formulation approaches that of the Coulomb gauge.

4.5.1.1 The Effects of Varying χ Values on Divergence Error for the Steady State Charge Density Problem

To isolate the effects of the correction potential speed on the electric field divergence error, tests with different values of χ on fixed resolutions (which, in this case is 200-by-200) are conducted. The χ values tested are one, two and four. The value of zero is not tested, in which case the charge density will be completely ignored, and the electric field will be identically zero at all times. The l_2-norm of the divergence error of the electric field is presented in Fig. 4.17.

Observing Fig. 4.17, it can be seen that the solutions behave similarly when using different values of χ. The error grows early on the simulation, and then slowly settles down to a steady-state value. All test cases settle down to approximately have about the same steady state error value. However, it can clearly be seen that the steady state error value is reached faster for higher value of χ, implying that higher correction potential speed results in the error wave being convected out of the domain faster, which also means that the electric field solution approaches steady state faster using a higher value of χ.

It is interesting to note that, while the structures of the solutions for the steady state charge problem are similar to the current density problem, the effects of the correction potential on the divergence error are different. While the divergence error of the magnetic field does not seem to be greatly affected by γ, the divergence error of the electric field behaves differently when χ is varied.

4.6 Time-Dependent Linearly Growing Charge Density Source Term Test Case

The two-fluid plasma model describes plasma as a fluid comprising of two species (ions and electrons). The charge density (see Eq. 2.14) for the two-fluid plasma can then be written as

$$\rho_c = q_e n_e + q_i n_i \tag{4.17}$$

This definition for charge density implies that a plasma does not necessarily possess charge neutrality. Compared to the commonly used MHD model, this is one of the distinctive features of the two-fluid plasma model, which allows for charge separation. Charge

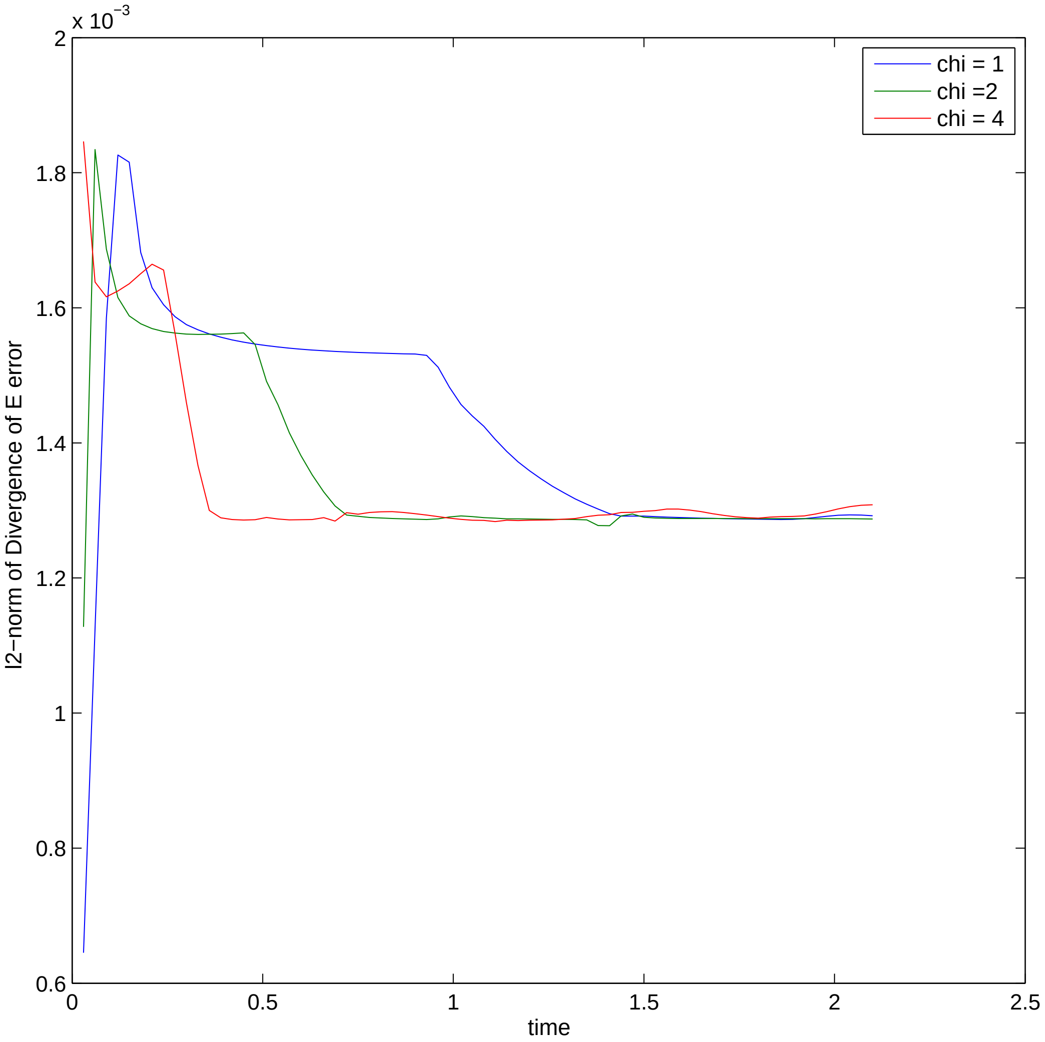

Figure 4.17: The l_2-norm of the divergence error of the electric field using different values of χ for the steady state charge density problem. Changing the value of χ does not significantly reduce the divergence error, but higher χ seems to lead to a faster steady state solution.

separation, however, can cause problems if it is not properly taken care of. As noted in Sec. 2.2, the Gauss' law (Eq. 2.11), which is the only part of Maxwell's equations that includes charge density, is treated only as a mathematical constraint, and is therefore not explicitly a part of the update formula, which only includes the curl parts of Maxwell's equations.

Mathematically Gauss' law will remain satisfied, provided that Eq. 2.11 and charge conservation, which can be written as

$$\frac{\partial \rho_c}{\partial t} + \nabla \cdot \mathbf{J} = 0 \tag{4.18}$$

are initially satisfied.

For plasma simulations, the charge and the current densities source terms are obtained from the variables of the Euler equations (Sec. 2.1), and there is no guarantee that the numerical method will necessarily preserve charge conservation throughout the simulation.

To observe how well the algorithm preserves the divergence constraint of the electric field, even when the charge conservation is not satisfied, a test case, based on Munz's work [16] is conducted. An artificial charge density, which linearly grows in time, is set in the middle of the domain (similar to the setup of the current problem, see Fig. 4.2).

The current is enforced to be zero at all times, hence, emphasizing the violation of the charge conservation. With current being set to zero, traditional Maxwell solver, which only advances the two curl equations, will produce zero for the solutions of the electromagnetic fields, clearly neglecting the existence of charge density that should give rise to an electric field.

The PHM system, however, is expected to provide corrections to the modified Gauss's law (Eq. 2.17), through the correction potential, ϕ. In potential formulation, the charge density is a direct source term to the scalar potential, Φ, whose gradients are parts of the electric field.

Thus, one can then surmise that the solutions produced by the modified Maxwell system (be it the potential formulation, or the PHM) should satisfy the divergence constraints better. At the very least, with these solvers, the artificial charge density should produce self-consistent electric field.

The domain is a [0,2] x [0,2] rectangular domain. For this problem, the charge density is set to be growing linearly with time within a circle of radius 0.075, and zero outside of the circle, or, namely,

$$\rho_c(t) = \begin{cases} t & r < 0.075 \\ 0 & r \geq \ 0.075 \end{cases} \tag{4.19}$$

where

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{4.20}$$

where, once again, x_c and y_c are the x and y locations at which the charge density pulse is centered, and r_c is the radius of the pulse.

Fig. 4.18, Fig. 4.19, and Fig. 4.20 show the magnitude of the electric field produced by the PH Maxwell system, the Lorenz gauge and the Coulomb gauge formulations respectively, solved within the skewed grid as shown in Fig. 4.1 at $t = 0.75$s.

It is important to note that the solutions produced by the Coulomb gauge formulation are a result of solving the electrostatic Poisson equation (Eq. 2.21). The solution is dependent only on the source terms, and is not a function of the solution of the previous time step. This is in contrary to the other two solvers, which depend on the solutions of the previous time step.

The instantaneous nature of the solution of the Poisson's equation is clearly apparent from Fig. 4.20. The solution is already in steady state, even from the beginning of the simulation.

Observing the solutions produced by the other two solvers (Fig. 4.18 and Fig. 4.19), however, shows the wave-like behavior of the underlying hyperbolic nature of the PHM and the Lorenz gauge based systems. In this case, the solutions start in the middle, and propagate throughout the domain.

It is important to note that, the Lorenz gauge and PHM solutions do eventually approach the steady state solution (the bottom right plots of Fig. 4.18 and Fig. 4.19).

4.6.1 Divergence Error Comparison for the Time Dependent Charge Problem

The divergence error is computed for each cell, and the l_2-norm of the divergence error of the electric field is computed using the method as given by Eq. 4.16.

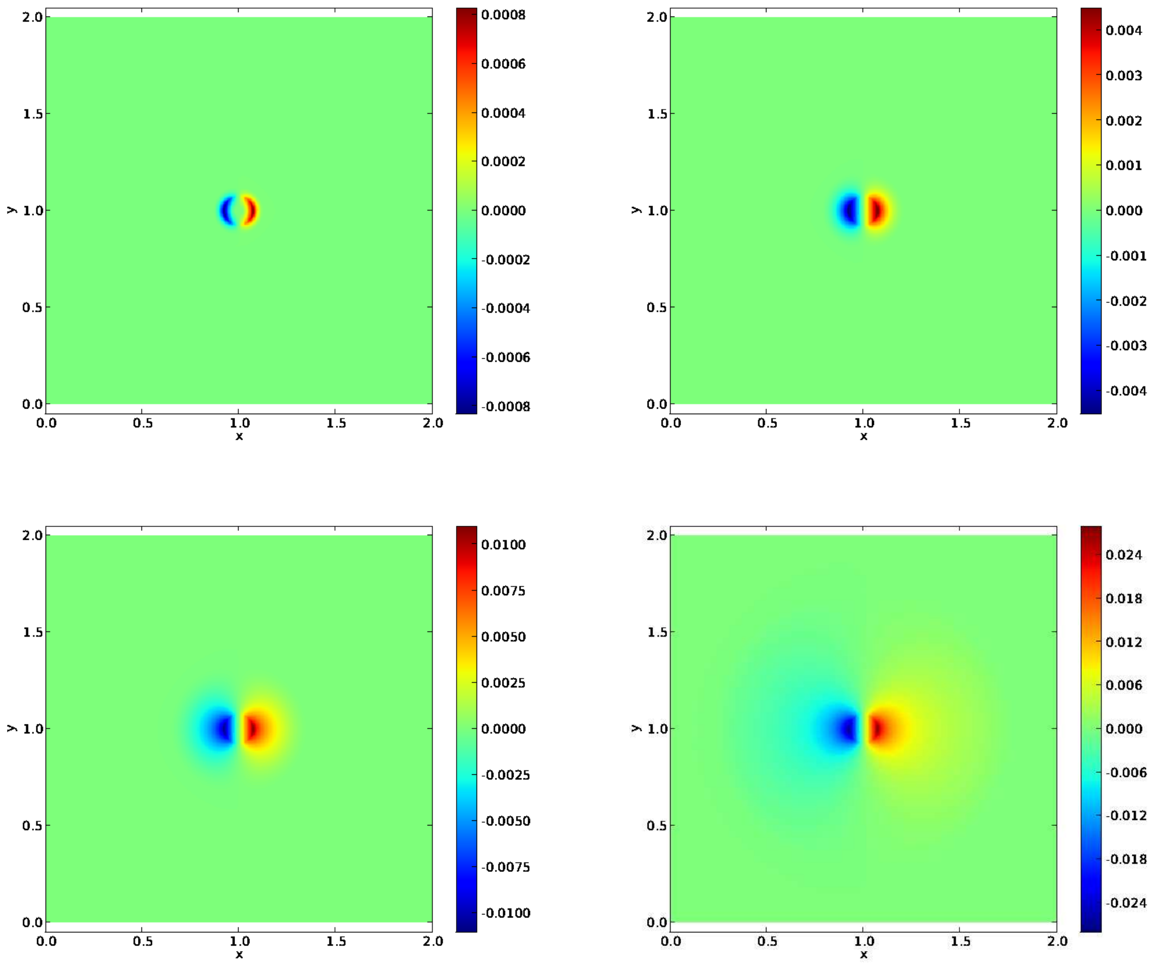

Figure 4.18: The evolution of the electric field in the x direction for the time-dependent charge density problem on a skewed grid using PHM with χ =1 . Shown here are solutions at $t = 0.06$, $t = 0.15$, $t = 0.3$, and $t = 0.75$. The solution starts in the middle of domain, and propagates outward.

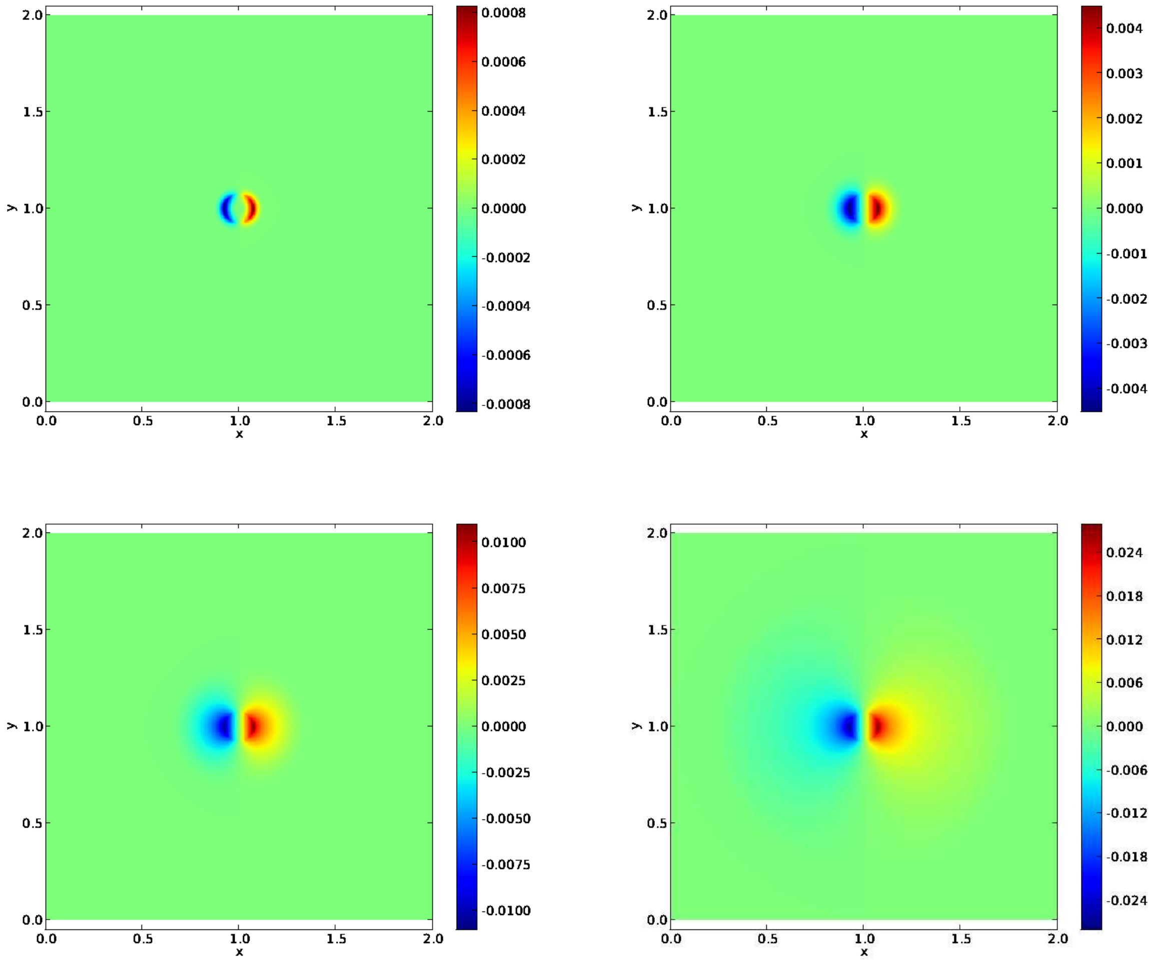

Figure 4.19: The evolution of the electric field in the x direction for the time-dependent charge density problem on a skewed grid using Lorenz gauge. Shown here are solutions at $t = 0.06$, $t = 0.15$, $t = 0.3$, and $t = 0.75$. The solution starts in the middle of domain, and propagates outward.

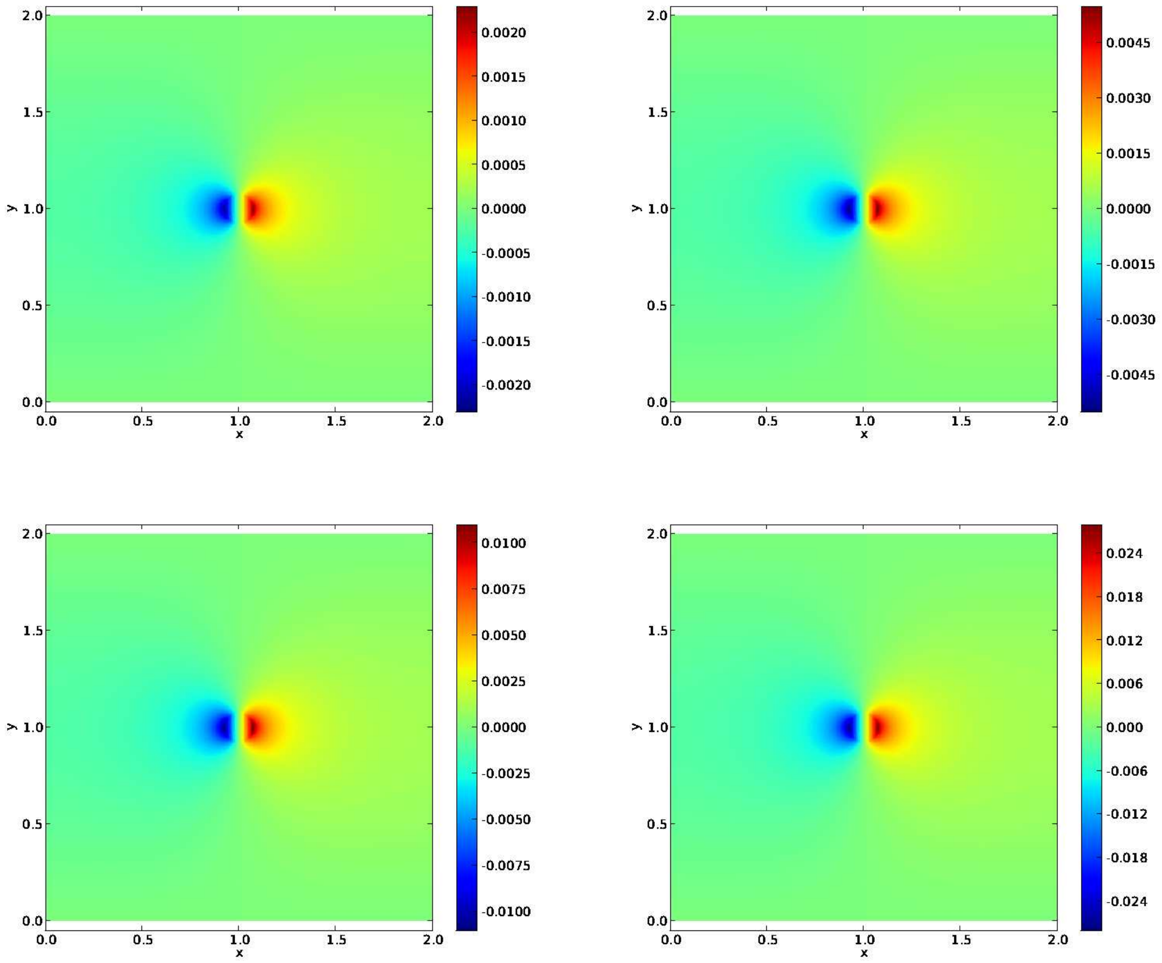

Figure 4.20: Electric Field Magnitude Solution From the Coulomb Gauge Formulation on skewed grid. Unlike the others solutions, which show wave-like structure, the Coulomb gauge formulation produces instantaneous solution for each time step.

The simulations are run at resolutions of 100-by-100, 200-by-200, and 400-by-400. Fig. 4.21, Fig. 4.22, and Fig. 4.23 show how the divergence error changes as a function of time for the aforementioned three different resolutions.

As can be seen from the plots, the divergence errors of the solvers increase almost linearly with time, which is expected. Charge density grows linearly with time, and it is expected that the divergence error growth rate is about linear.

The slopes of the l_2-norm of the divergence of the electric field vs. time plots are tabulated in Table 4.1. From the table, it can be observed that as the resolution is doubled, the errors decrease by a factor of about 2.8 for the potential solvers. For PHM, the l_2-norm error decreases by a factor of around 2 when the resolution is doubled.

Table 4.1: The slope of the l_2-norm of the Divergence E Error on The Skewed Grid

Method	100-by-100	200-by-200	400-by-400
PHM	2.6×10^{-3}	1.3×10^{-3}	6.78×10^{-4}
Coulomb Gauge	6.98×10^{-4}	2.27×10^{-4}	7.99×10^{-5}
Lorenz Gauge	6.3×10^{-4}	1.97×10^{-4}	6.90×10^{-5}

4.6.1.1 The Effects of Varying χ Values on Divergence Error for the Time Dependent Charge Density Problem

The effects of varying the correction potential speed on divergence error of the electric field for the time dependent charge density problem are observed. Similar to the previous tests described in Sec. 4.5.1.1 and Sec. 4.4.1.1.

The results are shown on Fig. 4.24. From the plot, it can be seen that the error is approaching the same behavior. It is also apparent from the figure that the higher the value of chi is, the more linear the error vs. time trend becomes, as it is evident by the correlation coefficient values of these plots: 0.9992 for $\chi = 1$, 0.9998 for $\chi = 2$, and 1 for $\chi = 4$.

While, the error does not necessarily decrease as the value of chi is increased, higher chi value does tend to linearize the divergence error vs. time results. Linear divergence error vs. time result is expected because charge density grows linearly over time (Eq. 4.20).

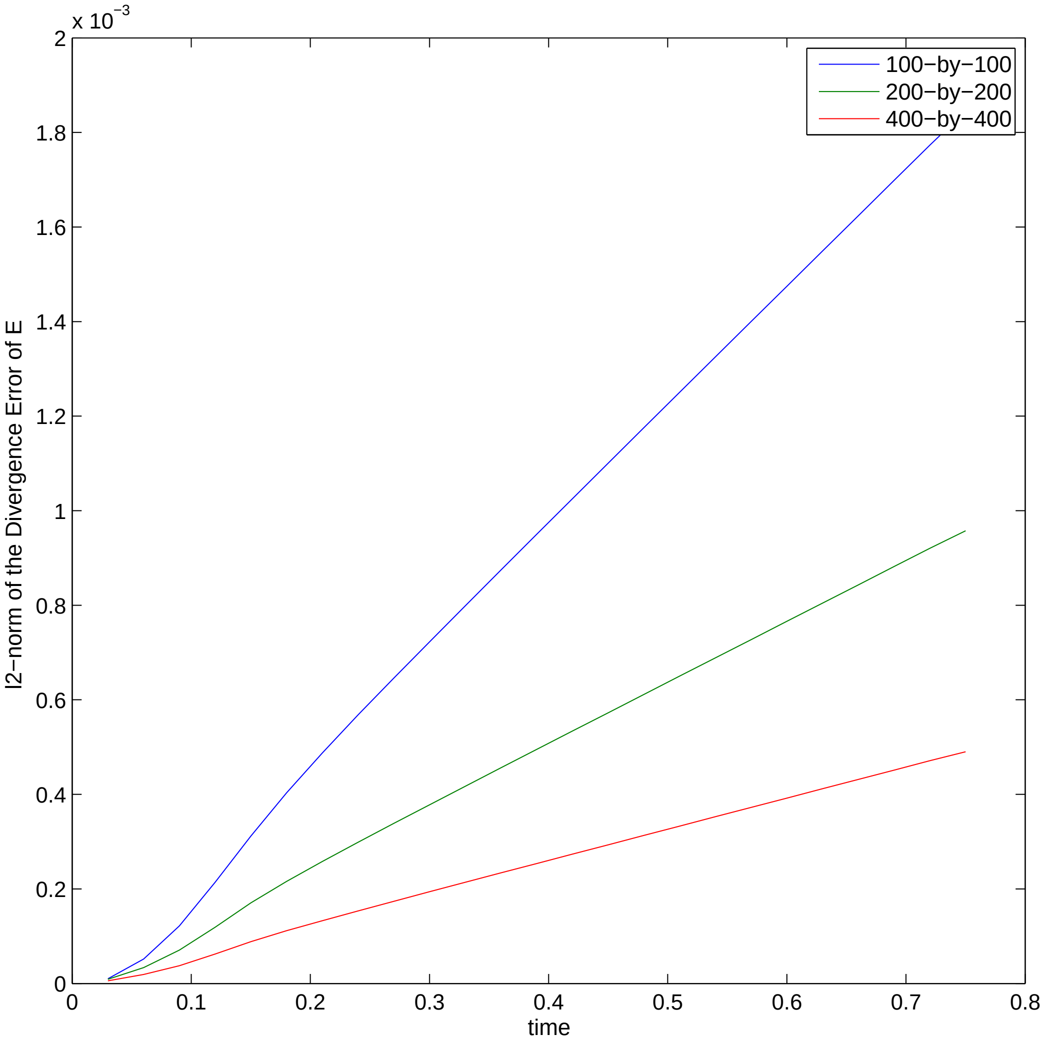

Figure 4.21: Divergence error for PHM System Formulation on different resolutions on skewed grid. The error increases almost linearly. Early in the simulation, the error increases non-linearly, but, after $t = 0.2$, the error starts to increase almost linearly, which is the expected behavior.

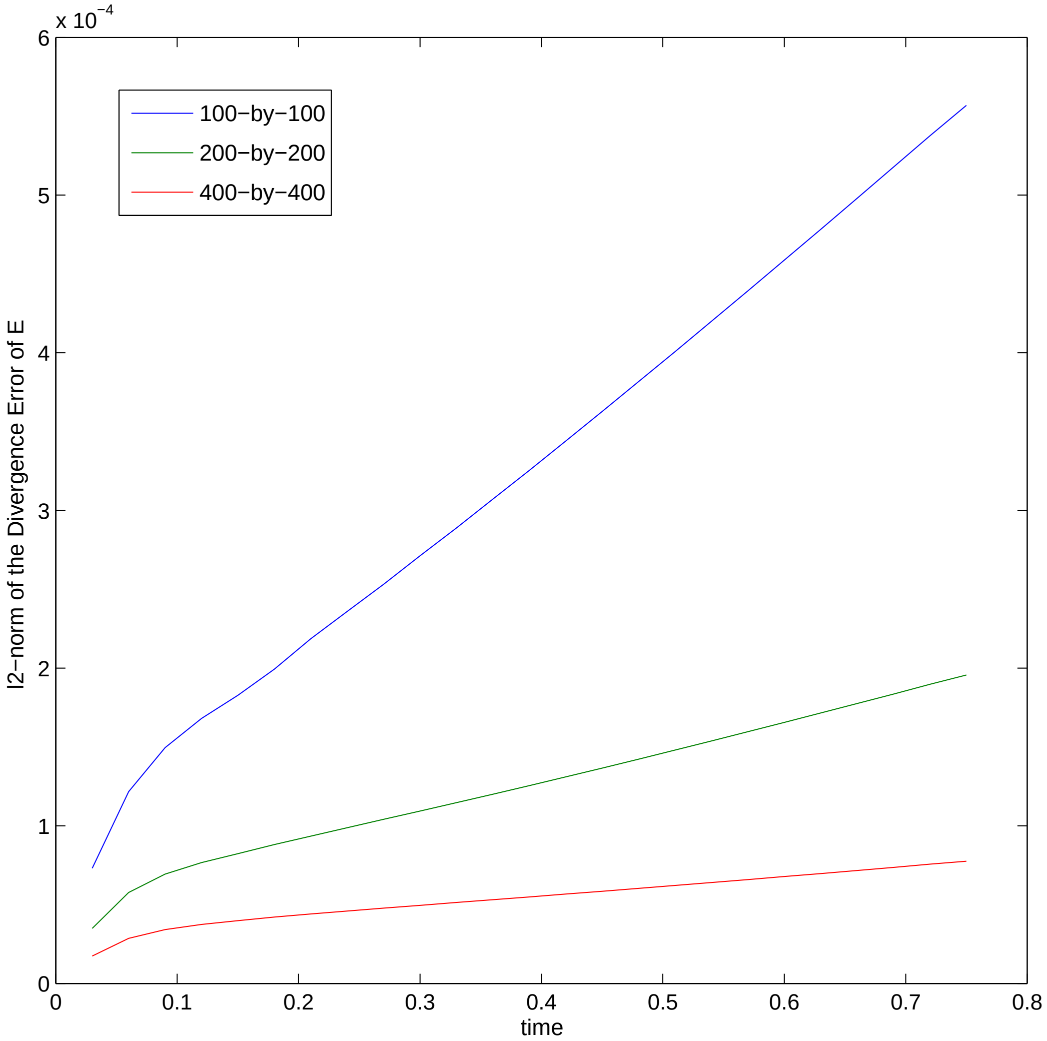

Figure 4.22: Divergence error for Lorenz Gauge Formulation on different resolutions on skewed grid. The error initially increases somehow non-linearly, but around $t = 0.2$, it starts to grow linearly. Because the charge density increases linearly with time, it should be expected that the divergence error also increases linearly.

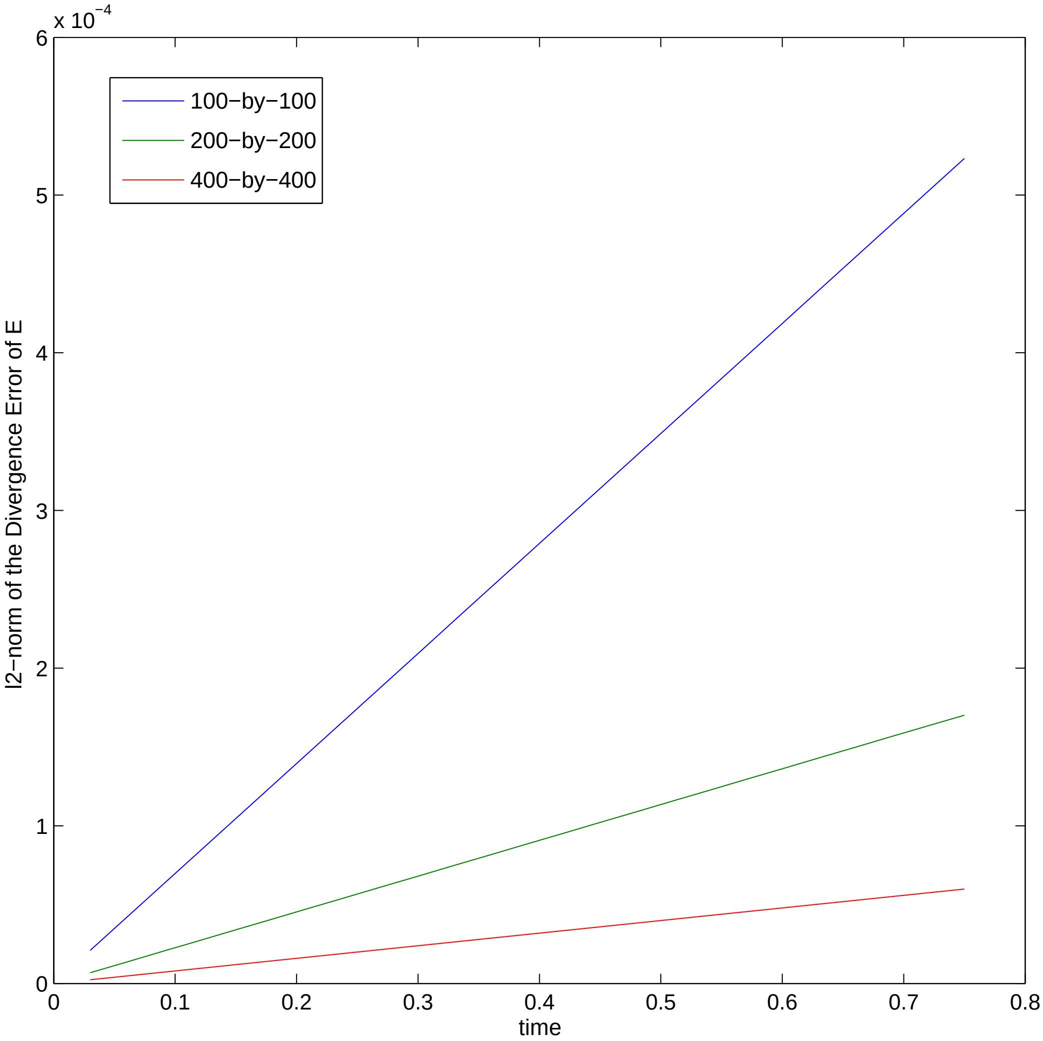

Figure 4.23: Divergence error for Coulomb Gauge Formulation on different resolutions on skewed grid. The Coulomb gauge formulation produces steady state solution for each time step, which results in a perfectly linear error vs. time graph, which is expected because the charge density increases linearly with time.

This is consistent with the finding reported in Sec. 4.5.1.1, where it is found that increasing χ does not directly reduce the divergence error, but rather, the solution approaches faster to its appropriate behavior. In this case, the error behaves linearly in time faster with higher χ values, whereas in the steady state case, the solution approaches steady state faster for higher χ.

4.7 The Effects of Grid distortion on the Solution

The effects of grid distortion on the solution are investigated. As mentioned in Sec. 4.1, the α parameter in Eq. 4.1 controls the skewness of the grid, while keeping the boundaries rectangular. Fig. 4.25) shows different grids with varying values of α. It can be seen from the figure that larger α results in more distorted grid.

4.7.1 Grid Distortion on Divergence Error

4.7.1.1 Analysis on the Current Problem

Figure 4.26 shows how varying the values of α affects the divergence error of the magnetic field. The tested values of alpha are 0, 4, 10, 20, and 30.

The l_2-norm is computed for each value of α as a function of time for comparison. As the plots suggest, the divergence error for the solutions based on the PHM solver do not seem to suffer much from increasing grid distortion. The potential-based solutions, however, are more sensitive to the grid deformation, and behave as expected, with larger divergence error for a greater value of α. Another way to highlight the differences in the divergence error for the potential solver is through the error bumps that the error experiences early on the simulation. The times at which these bumps in error occur are different for different values of α, and it is interesting to note that this bump in error occurs at a later time for a larger value of α.

The sensitivity of the potential solver to grid deformation is expected. While the wave propagation method employed to solve the PHM equation system, several derivative operators, as described in Chapter 3, have to be applied.

With PHM, the magnetic field can be advanced directly, because it is a first order hyper-

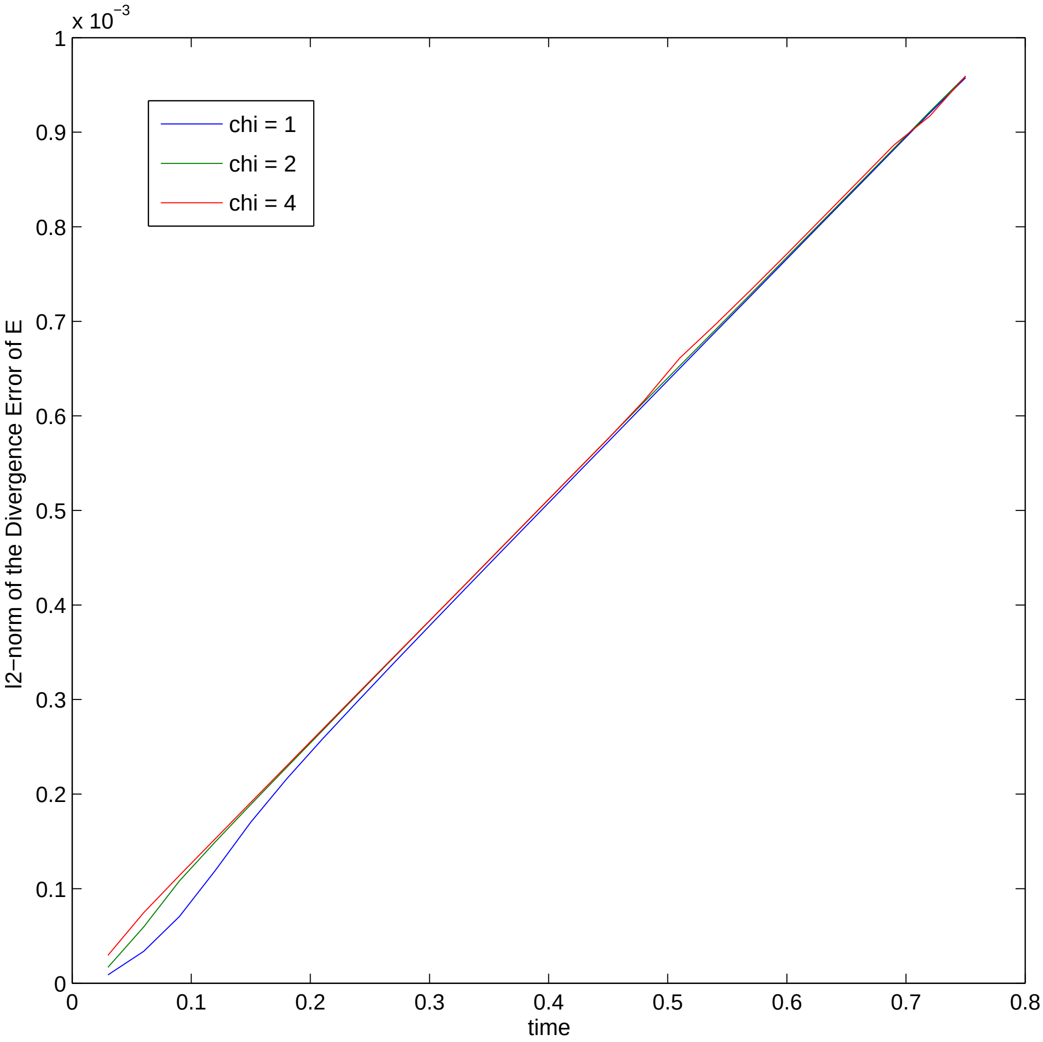

Figure 4.24: Divergence error for different values of χ on the skewed grid for the time-dependent charge density problem. As the value of χ increases, the line becomes more linear, approaching the expected behavior.

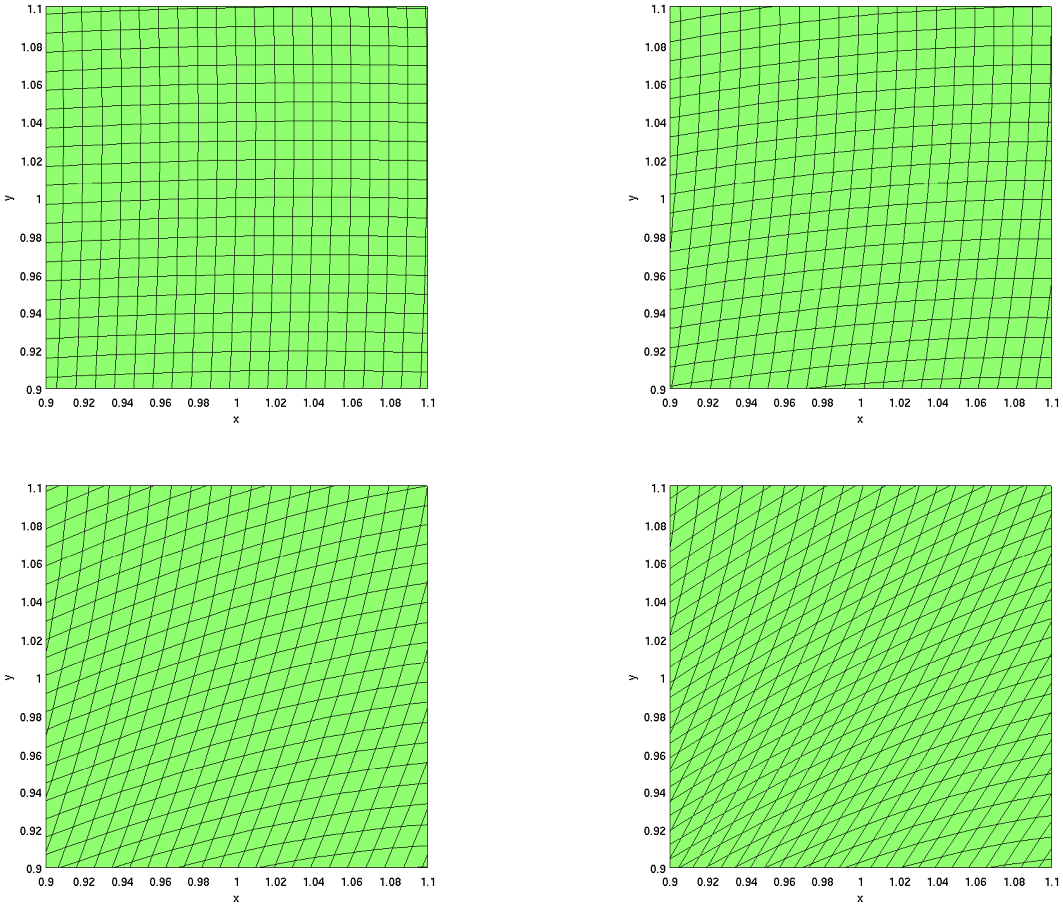

Figure 4.25: Zoomed in Skewed grid with twist applied in the middle. Upper left shows the middle of the domain when α is set to 4, upper right for α set to 10, lower left for α set to 20, and lower right for α set to 30. As the value of α increases, the grid becomes more distorted.

bolic equation system, which can be solved directly using the wave propagation method. In the potential formulation, the magnetic field is expressed as the curl of the vector potential (Eq. 2.20), which is itself the solution of the second order wave equations (Eq. 2.22 and Eq. 2.25).

It is possible that this extra layer of operation causes the solutions based on the potential formulation to be more sensitive to grid distortion, as apparent from Fig. 4.26. These derivative operators rely mostly on finding approximations of the solutions on the cell interfaces, which are obtained through linear interpolation, which may experience a loss in accuracy as the grid becomes more skewed.

The wave propagation method, however, employs several correction phases (such limiters and flux correction terms as described in Sec. 3.1.2) to ensure second order accuracy, which is a possible reason why the PHM-based solver does not suffer as much from grid deformation as the potential solver.

4.7.1.2 Analysis on the Steady State Charge Density Problem

Fig. 4.27 shows the divergence error of the electric field for the steady state charge density problem. For PHM, once again, different values of α do not affect the results in an expected manner. There does not seem to be a straight forward relationship between the results and the skewness of the grid. All the same, all test cases produce solutions with similar behavior. The divergence error first increases rapidly, and then decreases until it reaches steady state.

Observing the divergence error of the Lorenz gauge formulation solution, it can be seen that the error plot acts differently than the PHM solution. While the difference in error across the different values of α is small, the error behaves as expected in that, α increases, the error also increases. This finding is consistent with the finding of Sec. 4.7.1.1, in which the potential system is again behaving in the more expected manner with respect to grid skewness, once again confirming the degradation of the linear interpolation-based derivative operators as the grid becomes more skewed.

Similar behavior can also be observed from the Coulomb gauge formulation. The di-

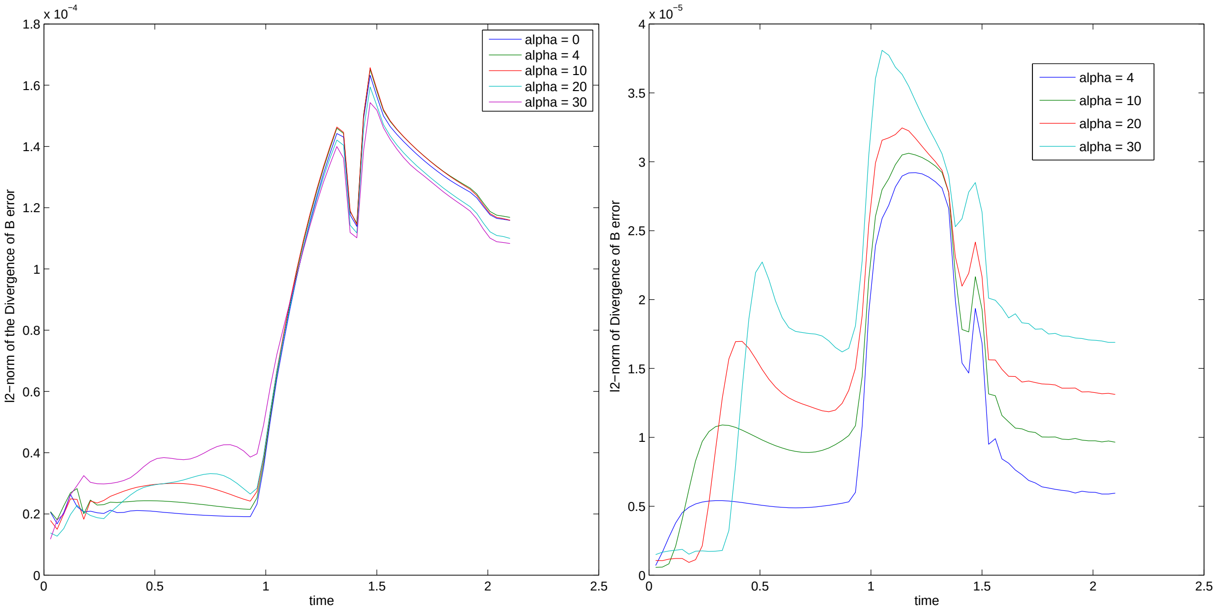

Figure 4.26: Grid effects for current density problems. Left: PHM solver, right: potential solver. The potential solver shows the expected effects of grid deformation, where the divergence error gets larger as the grid becomes more deformed. The same cannot be said for PHM, where grid deformation does not seem to have much of an effect to the divergence error.

vergence error for Coulomb gauge based solver increases as the grid becomes more skewed, even though an anomaly is observed when setting α to 30, where the error is actually lower than when α is set to either 20 or 25 (see Fig. 4.27).

4.8 Conclusions For the Electromagnetic Source Term Testing

In this section, the capability of the solvers to handle source terms is assessed. From the analysis, it is found that, at least for the problems considered within this section, the potential solvers have been found to produce solutions with smaller divergence error.

The steady state charge and current density problems reveal that both solutions produced by the Lorenz gauge and the PHM based solvers eventually approach steady state. The Coulomb gauge solution, however, is, by definition, in steady state, and thus, has the smallest divergence error (see Fig. 4.16). It is then used as a reference for comparisons with the other two solvers (PHM and Lorenz gauge), in which it is found that both solutions do indeed approach the Coulomb gauge solution, with the Lorenz gauge solution approaching it faster than the PHM solution.

While the PHM and the potential equation systems produce qualitatively similar looking solutions, upon further observation of Fig. 4.9, Fig. 4.10, and Fig. 4.11, the PHM solution has been observed to produce more artificial reflections, which can be attributed to the formulation of the boundary conditions. Copy boundary conditions are used for PHM, whereas the second order Mur one way wave equation is employed to model open potential boundaries.

Comparing the errors, the solutions of the potential-based solvers have smaller divergence error. Increasing the values of χ and γ does not seem to decrease the error. While the electric field solution does approach steady state faster when a higher value of χ is used (see Fig. 4.17), γ does not seem to affect the solutions in a similar way (see Fig. 4.8). While going from 0 to 1 provides significant correction, the differences in error for γ above 1 are negligible, and might be attributed to numerical noises.

For the time-dependent problem, once again the Coulomb gauge solution is used as a reference. Even though the charge is linearly increasing with time, the Coulomb gauge solution represents a steady state solution for each time-step. The error then should also

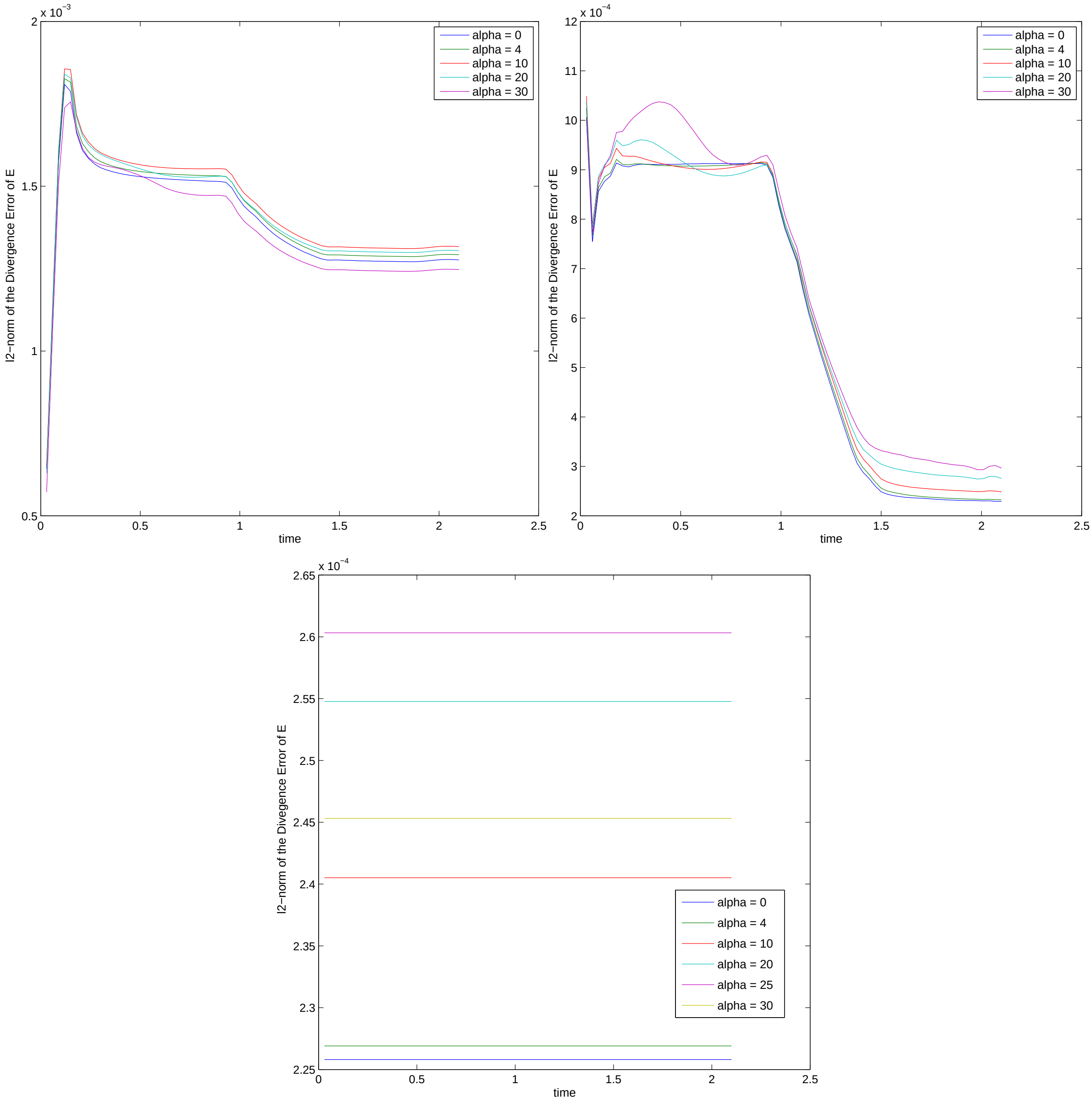

Figure 4.27: Grid Effects for Steady State Charge Problems. Upper Left: PHM solver, upper right: potential solver, bottom: Coulomb gauge solution. Once again the potential based solver behaves as expected, in which the error gets larger as the grid becomes more skewed. PHM once again shows an unexpected behavior, where the error does not seem to follow any specific pattern with respect to grid deformation. The Coulomb gauge solver behaves similarly with the Lorenz gauge solver, but somehow the error is smaller for $\alpha = 30$, than for $\alpha = 20$ and $\alpha = 25$. The PHM errors, however, are still larger than the MP solutions

be increasing linearly as can be confirmed after observation of Fig. 4.23. The PHM and the Lorenz gauge solutions both initially produce non-linear divergence errors vs. time, but later on, these errors become more linear, once again approaching the behavior of the Coulomb gauge solution (see Fig. 4.21 and Fig. 4.22).

Observing the effects of χ also shows that the behavior of the electric field solution approaches that of the Coulomb gauge solver as χ is increased (see Fig. 4.24), with the error vs. time graph becoming more linear for higher χ.

Finally, the effects of grid distortions on the solutions are examined. It is found that the potential formulation is more sensitive to grid deformation than PHM, which may be due to the higher order derivative operators that have to be applied to the solutions before the magnetic and the electric fields can be obtained.

Chapter 5
PRELIMINARY PLASMA TEST CASES IN CARTESIAN GRID

In this chapter, the two-fluid MHD shock (Sec. 5.2), and the magnetic reconnection problems (Sec. 5.3) on Cartesian grid are tested. The results across different solvers are tested on divergence error and computational time. For the magnetic reconnection problem, reconnected magnetic flux of the solution will also be compared and analyzed. The implementation of the potential solvers to the two-fluid system will also be described within this chapter (Sec. 5.1).

5.1 Implementation of Potential Solvers into the Two-Fluid Plasma System

Hakim, Loverich and Shumlak [1] have conducted preliminary two-fluid plasma simulations in a uniform Cartesian grid. PHM solver (as described in Sec. 2.2.1) is used to advance the electric and magnetic fields. In this section, potential solvers are also used to advance the electromagnetic fields, and the results will be compared with the two-fluid results using PHM system as the electromagnetic solver.

While the PHM system does not require a special treatment, both potential solvers involve solving second order partial differential equations (see Sec. 2.2.2), which means that they have to be solved outside of the main hyperbolic system, which is solved using the wave-propagation method. So the scalar and vector potentials are solved using the finite volume operators as described in chapter 3. The resulting vector and scalar potentials are converted into their electric and magnetic field equivalents (Eq. 2.19 and Eq. 2.20), which are then used in the source terms for the momentum and energy equations of the Euler system (Eq. 2.5 and Eq. 2.7).

The electromagnetic potentials and the fluid variables will have to be staggered in time, especially since the electric field requires an approximation for the time derivative of the vector potential (2.19). The currents are assumed to be located at half time steps, whereas the charge density is stored at whole time steps. By staggering the variables in time, a second order accurate central differencing in time can be used.

5.1.1 Lorenz Gauge Implementation

The potential solver using the Lorenz gauge first computes the source terms to the potential formulations, which are the currents and the charge density (Eq. 2.25 and Eq. 2.26). The Laplacian of the vector and the scalar potentials are then computed, using the method described in Sec. 3.2. The potentials are advanced, using the update formula as given in Eq. 4.5. The electric and the magnetic fields are then computed, and used as source terms for the fluid equations.

5.1.2 Coulomb Gauge Implementation

The numerical implementation of the Coulomb gauge potential formulation is more involved than that of the Lorenz gauge formulation, because it requires solving two Poisson equations for every time-step. The full Coulomb gauge implementation can be described as the following

1. Compute the displacement current term, $\frac{\partial \Phi}{\partial t}$ at time $n+\frac{1}{2}$ by using $\mathbf{J}^{n+\frac{1}{2}}$ by solving the reformulated charge conservation equation (Eq. 2.24)
$$\nabla^2 \left(\frac{\partial \Phi}{\partial t}^{n+\frac{1}{2}} \right) = \nabla \cdot \frac{\mathbf{J}^{n+\frac{1}{2}}}{\epsilon_0} \tag{5.1}$$

2. Using $\frac{\partial \Phi}{\partial t}^{n+\frac{1}{2}}$ and $\mathbf{J}^{n+\frac{1}{2}}$, advance $\mathbf{A}^{n+\frac{1}{2}}$ to $\mathbf{A}^{n+\frac{3}{2}}$ using the update equation for the vector potential (Eq. 2.22).

3. Because the wave propagation method employs a source splitting technique (as described in Sec. 4.2), Q, which are fluid variables are first homogeneously advanced to an intermediate step Q^*, which gives ρ_s^{n+1} and $\rho \mathbf{u}_s^*$.

4. Using ρ_s^{n+1}, ρ_c^{n+1} is computed, and the scalar potential, Φ, can be obtained by solving

the Gauss' law

$$\nabla^2 \Phi^{n+1} = -\frac{\rho_c^{n+1}}{\epsilon_0} \tag{5.2}$$

5. Now, the electric field, $\mathbf{E}^{n+1}$, can be computed (Eq. 2.19), using $\mathbf{A}^{n+\frac{1}{2}}$, $\mathbf{A}^{n+\frac{3}{2}}$, and Φ^{n+1}. And the magnetic field, $\mathbf{B}^{n+\frac{3}{2}}$ can be expressed as the curl of the vector potential (as given in Eq. 2.20)

$$\mathbf{E}^{n+1} = -\nabla \Phi^{n+1} - \frac{\mathbf{A}^{n+\frac{3}{2}} - \mathbf{A}^{n+\frac{1}{2}}}{\Delta t} \tag{5.3}$$

$$\mathbf{B}^{n+\frac{3}{2}} = \nabla \times \mathbf{A}^{n+\frac{3}{2}} \tag{5.4}$$

6. Lastly, using $\mathbf{B}^{n+\frac{3}{2}}$ and $\mathbf{E}^{n+1}$, the momentum, $\rho \mathbf{u}_s^*$, and also the energy, ε_s^*, can be advanced to $\rho \mathbf{u}_s^{n+\frac{3}{2}}$ and ε_s^{n+1}, by solving the ODE for the source terms associated with these variables

$$\frac{d(\rho_s \mathbf{u}_s^*)}{dt} = \frac{\rho_s q_s}{m_s} \left(\mathbf{E}^{n+1} + \mathbf{u_s} \times \mathbf{B}^{n+\frac{3}{2}} \right) \tag{5.5}$$

$$\frac{d(\varepsilon_s^*)}{dt} = \frac{\rho_s q_s}{m_s} \mathbf{u_s} \cdot \mathbf{E}^{n+1} \tag{5.6}$$

5.2 1D Two-Fluid Plasma Shock Problem

The Brio-Wu shock tube problem is a problem commonly used for benchmarking Magnetohydrodynamics (MHD) codes [22]. Ref. [23] generalized the system to the two-fluid plasma system, to incorporate multiple species.

For the purpose of testing, the discontinuities are located at the middle of the domain, which spans from 0 to 1. The problem is defined by the magnitudes of the right and the left states of the variables. The variables that have discontinuities are the mass densities, pressures and the transverse magnetic field. The initial conditions for the two-fluid MHD shock problem are

$$\begin{bmatrix} \rho_e \\ p_e \\ \rho_i \\ p_i \\ B_x \\ B_z \end{bmatrix}_{left} = \begin{bmatrix} 1.0\frac{me}{mi} \\ 5\times10^{-5} \\ 1.0 \\ 5\times10^{-5} \\ 0.75 \\ 1.0 \end{bmatrix}, \begin{bmatrix} \rho_e \\ p_e \\ \rho_i \\ p_i \\ B_x \\ B_z \end{bmatrix}_{right} = \begin{bmatrix} 0.125\frac{me}{mi} \\ 5\times10^{-6} \\ 0.125 \\ 5\times10^{-6} \\ 0.75 \\ -1.0 \end{bmatrix} \tag{5.7}$$

All of the other variables (such as momenta and the electric fields) are initially set to zero. The magnetic field in the x-direction is set constant for all times, trivially satisfying the magnetic field divergence constraint (Eq. 2.12). There is no guarantee, however, that the Gauss's law (Eq. 2.11), is always satisfied.

The solutions produced by the three Maxwell solvers are shown in Fig. 5.1 and Fig. 5.2, which show the electron and ion mass densities. The solutions, as evident from these plots, the differences between these solvers are minimal, implying that they are all capable of handling discontinuities in the solutions. These results also agree with the two-fluid solutions of Ref. [1] and Ref. [23].

5.2.1 Divergence Error Comparison for Two-Fluid Plasma Shock Problem

The MHD Shock problem is a one-dimensional problem. Because the magnetic field in the x-direction is held constant for all times, its divergence is always zero, and Eq. 2.12 is trivially satisfied for all times.

The l_2-norm of the divergence error for the electric field is computed by using Eq. 4.16. The PHM and the potential solvers are used to advance the electromagnetic part of the two-fluid plasma system, and the results are shown and compared in Fig. 5.3. As can be seen from the plot, the peak of the error happens early in the simulation. The error then does not seem to increase over time, but rather, oscillates around a certain value. From the plot, one can then infer that the error produced by the PHM solutions is lower compared to the error of the potential solvers. The Coulomb solution is about 3 times lower compared to the Lorenz solution, whereas the value of χ does not significantly affect the solution. In

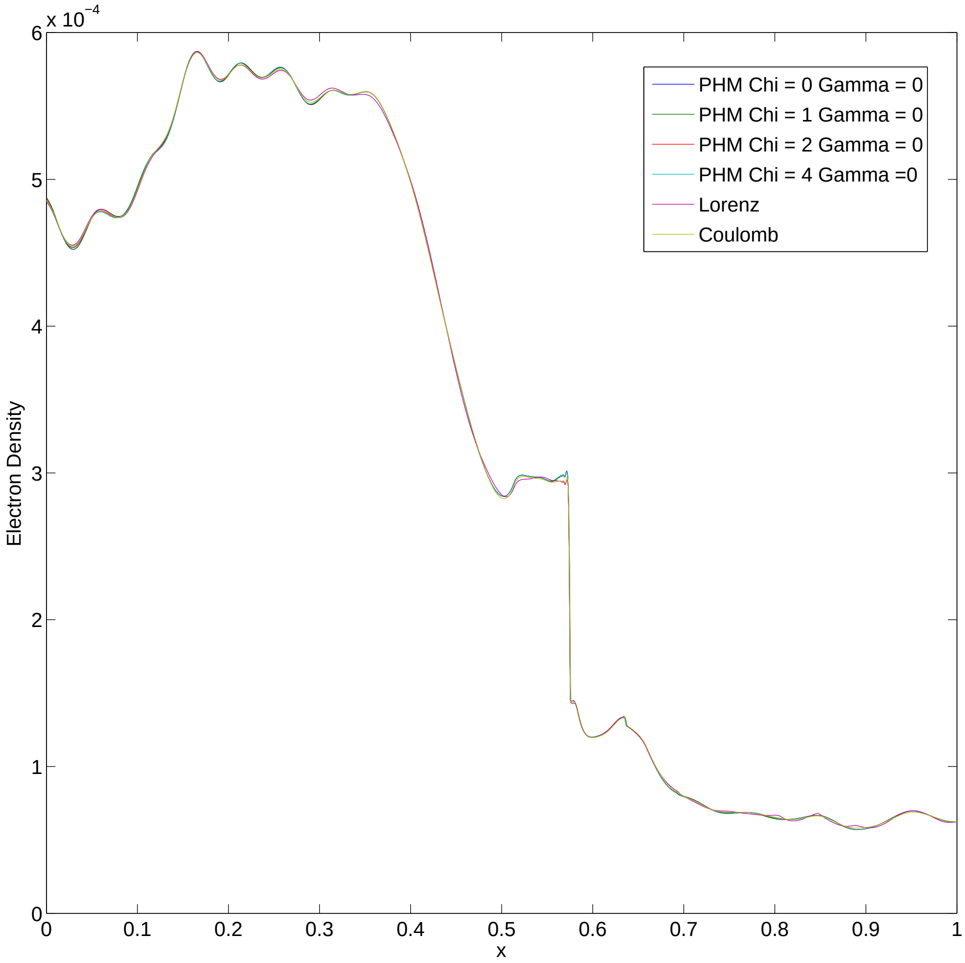

Figure 5.1: Electron mass density solutions for the two-fluid MHD Shock problem. The electron mass density looks similar for all solvers, and does not seem to be significantly affected by χ.

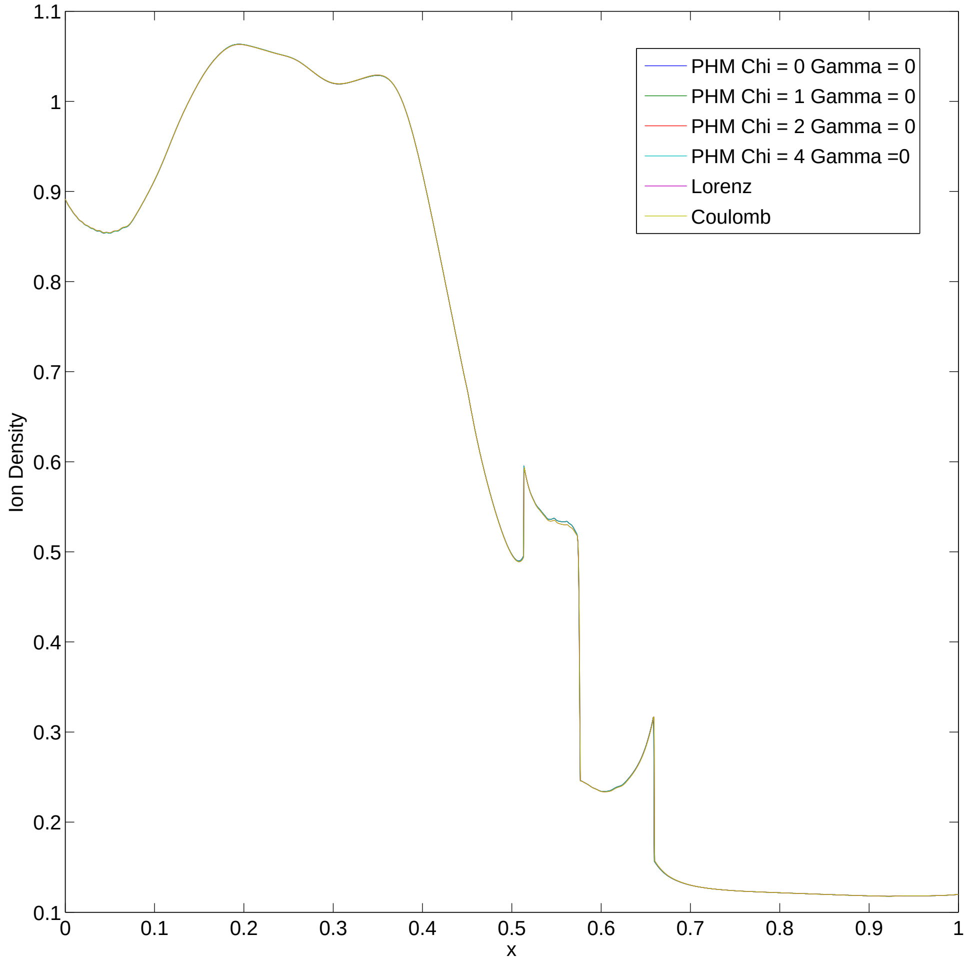

Figure 5.2: Ion mass density solutions for the two-fluid MHD Shock problem. The electron mass density looks similar for all solvers, and does not seem to be significantly affected by χ.

fact, the electric field divergence errors for the solutions using PHM with $\chi = c_0$, $\chi = 2c_0$, and $\chi = 4c_0$ are almost identical after t around 2 light transit times.

Overall, as can be seen in Fig. 5.3, the solutions of the PHM solvers have lower divergence error compared to the solutions of the potential solvers, which is in contrary to the results reported in Chapter 4, which finds that the potential based solvers have smaller divergence error compared to the PHM based solver.

The computational times for the simulations are computed and compared. The domain size considered is 5,000 grid cells. A Courant number of 0.8 is used to ensure satisfaction of CFL stability criterion, and this corresponds to a time step of 1.6×10^{-4} light transit times. The simulation is run to 10 light transit times, which requires 62,500 time steps.

For a simulation run of 10 light transit times, the Lorenz-gauge based solver requires about 250 seconds of CPU times to finish, whereas the PHM solver requires 380 seconds when χ is set to zero, and about 450 seconds when it is set to c_0. For higher values of χ, the time required to finish the simulation is higher because the CFL condition becomes more restrictive. The times required to finish the simulations are about 2800 seconds, and 5800 seconds for $\chi = 2c_0$ and $4c_0$ respectively.

For The Coulomb gauge, the solution requires 4,000 seconds. The excessive time requirement for the Coulomb-gauge based solver stems from the fact that, for each time step, two Poisson's equations need to be solved. The MHD Shock problem then requires solving 125,000 Poisson's equations throughout the whole simulation, which has proven to be computationally expensive. Despite this, however, the Coulomb gauge solver still requires less time than PHM with higher values of χ, which implies that the improvement in divergence error associated with using higher values of χ is not worth the extra computational effort required to ensure stability.

Thus, for this particular problem, the PHM based solver with $\chi = c_0$ seems to be giving the better balance of computational efficiency and accuracy. The Coulomb gauge solver does give comparable solutions with the PHM system with χ set to c_0 (Fig. 5.2, Fig. 5.1, and Fig. 5.3), but it requires solving Poisson's equations for each time step.

The one-dimensional Poisson's equations in Cartesian grid is a tridiagonal system of equations. Currently, the Poisson's operator is solved using a general sparse matrix solver

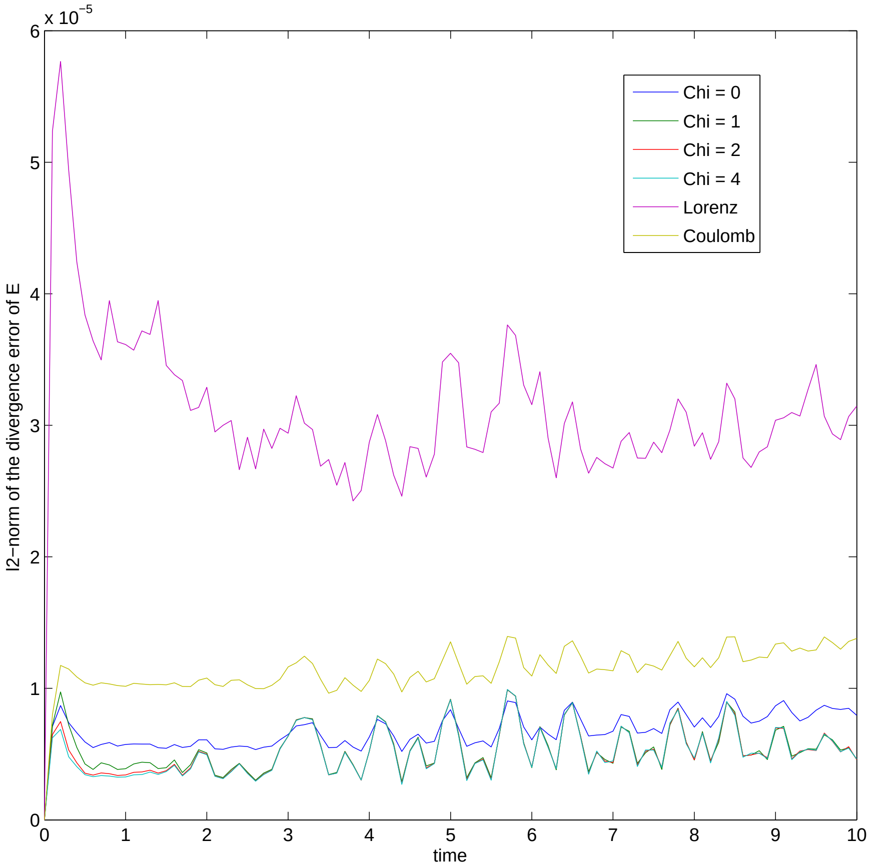

Figure 5.3: Comparison of l_2-norm of the Divergence Error of the Electric Field for the MHD Shock problem. The PHM produces solutions with lower divergence error than the potential based solvers. The error correction speed, χ, does not significantly affect the solution, provided that it is set to be at least equal to the speed of light.

from PETSc. Because the tridiagonal system is probably the simplest of linear sparse matrices that can be solved, there might not be any need to employ a general linear solver, and instead, a specialized tridiagonal matrix operator solver, which would probably be a lot more efficient, might suffice. Investigating how different linear solvers in PETSc affect the performance would be a topic worth pursuing for future studies.

5.3 Magnetic Reconnection

The magnetic reconnection problem, modelled after the GEM challenge [24], is tested with the two-fluid codes on a uniform Cartesian grid. The GEM challenge test case has been performed using several traditional plasma models such as Ideal MHD [25], Hall MHD with anisotropic pressure [26], Hall MHD [25], particle [27], and hybrid models [26].

There was, however, no simulation conducted with the two-fluid model as described in chapter 2 under the original GEM Challenge rubric. Loverich [2] and Hakim et al. [1] developed two-fluid plasma solvers, and conducted the GEM Challenge tests. It was found that the two-fluid solvers were able to capture the physics required for fast collisionless magnetic reconnection. The two-fluid solutions were also found to produce reconnected flux history that was well in agreement with the published results [24]. The two-fluid solvers used by Loverich [28] and Hakim [1] employed PHM to advance the electromagnetic fields. In this section, the potential-based solvers are also used to obtain the electromagnetic solutions. The solutions are then compared with the PHM solutions in terms of divergence error and reconnected flux.

The problem is initialized by the Harris current sheet configuration [29]. The initial magnetic field is given by

$$\mathbf{B}(y) = B_0 \tanh\left(\frac{y}{\lambda}\right)\hat{x}, \tag{5.8}$$

which is the equilibrium magnetic field of the Harris configuration. In potential form, this translates to

$$\mathbf{A}(y) = \lambda B_0 \ln\left(\cosh\left(\frac{y}{\lambda}\right)\right)\hat{z}. \tag{5.9}$$

Initially, only the electrons carry current

$$\mathbf{J}_e = -\frac{B_0}{\lambda}\mathrm{sech}^2\left(\frac{y}{\lambda}\right)\hat{z}. \tag{5.10}$$

The remaining equilibrium initial conditions are the number densities, which are the same for ions and electrons, and the pressure, for which $p_i = 5p_e$.

The number density can be written as

$$n(y) = n_0(0.2 + \mathrm{sech}^2(\frac{y}{\lambda})) \tag{5.11}$$

And the electron pressure is

$$p_e(y) = \frac{B_0}{12} n(y) \tag{5.12}$$

To initiate reconnection, the following perturbation is applied to the magnetic potential

$$\delta \mathbf{A}(x, y) = \psi_0 \cos(\frac{2\pi x}{L_x}) \sin(\frac{2\pi y}{L_y}) \hat{z} \tag{5.13}$$

This perturbation will initiate a magnetic field in the transverse direction, and also ensure that the resulting magnetic field is divergence-free, because the perturbation is applied to the vector potential, whose curl is divergence-free.

The simulation is conducted on a two-dimensional domain with a size of $[\frac{-L_x}{2}, \frac{L_x}{2}]$x $[\frac{-L_y}{2}, \frac{L_y}{2}]$, following the parameters defined for the GEM Challenge problem [24]: $L_x = 8\pi$, $L_y = 4\pi$, $B_0 = 0.1$, $\psi_0 = \frac{B_0}{10}$, and $\lambda = 0.5$.

The magnetic reconnected flux, used as a means of comparisons across different solvers, is defined as follows

$$\Delta \Phi(t) = \frac{1}{2L_x} \int_{\frac{-L_x}{2}}^{\frac{L_x}{2}} |B_y(x, y = 0, t)| dx \tag{5.14}$$

which is a measure of the net magnetic field in the y-direction, taken at the middle of the domain ($y = 0$).

In addition to reconnected flux, the total ion and the electron momenta for each solver are shown in Fig. 5.4 – Fig. 5.10 for comparison.

While the two potential solutions are qualitatively similar, significant differences can be observed from the PHM test cases, which have χ of c_0, $2c_0$ and $4c_0$. With no correction to the electric field, the solutions match the two-fluid solutions provided in [1] and [2]. No magnetic island is produced in the middle of the domain, and the solution also features the complex flow structure observed by Hakim et al. [1]. In all these simulations, $\gamma = c_0$.

The values of χ, however, seem to affect the island dynamics. When $\chi = c_0$, an island is observed in the middle of the domain, as can be seen from Fig. 5.5. It does not show up in a PHM simulation when χ is set to $2c_0$ (Fig. 5.5). When $\chi = 4c_0$, the island shows up, but late in the simulation, it merges with one of the side magnetic sites (Fig. 5.7 and Fig. 5.8).

It is also important to note that all simulations that produce the island directly include charge density, ρ_c, in the update formula, which means that charge separation may play an

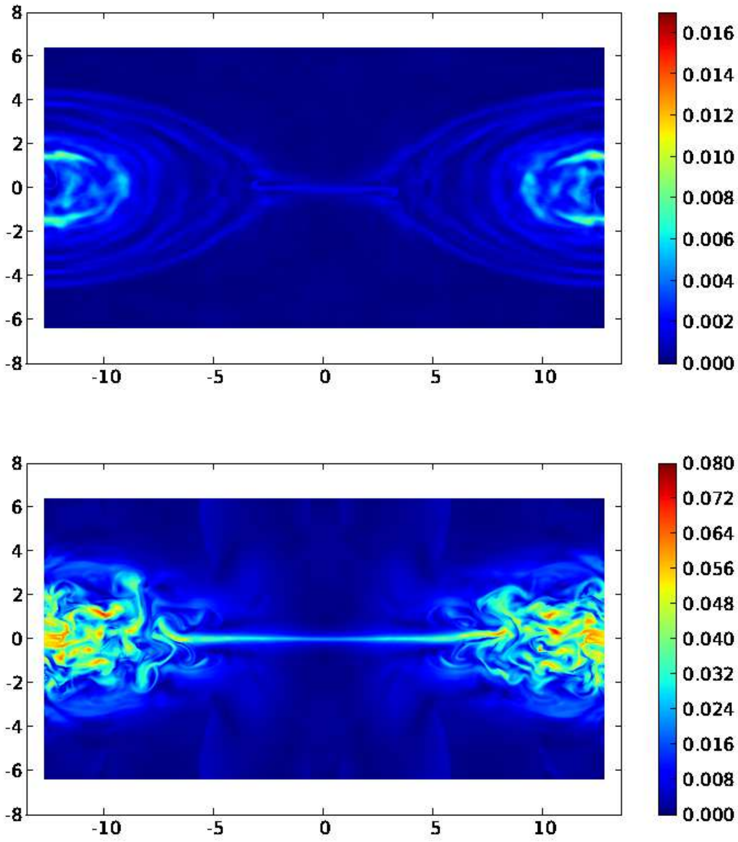

Figure 5.4: Total electron (top) and ion (bottom) momenta for the reconnection problem using PHM solver with χ set to zero. The solution features complex flow also observed by Hakim [1]. With no correction applied to the electric field, there is no magnetic island in the middle of the domain.

important role in the reconnection dynamics, and warrants further future studies.

Besides the magnetic island, the solutions look qualitatively similar across different solvers, and the electromagnetic solvers do not seem to affect the ability of the two-fluid plasma system to capture the complex physics of reconnection.

The reconnected flux is then computed (Eq. 5.14) and results are shown in Fig. 5.11. The time scale is the ion cyclotron frequency. As can be seen from the plot, the reconnected flux is initially very similar among the different solvers. For all solvers, the reconnected flux

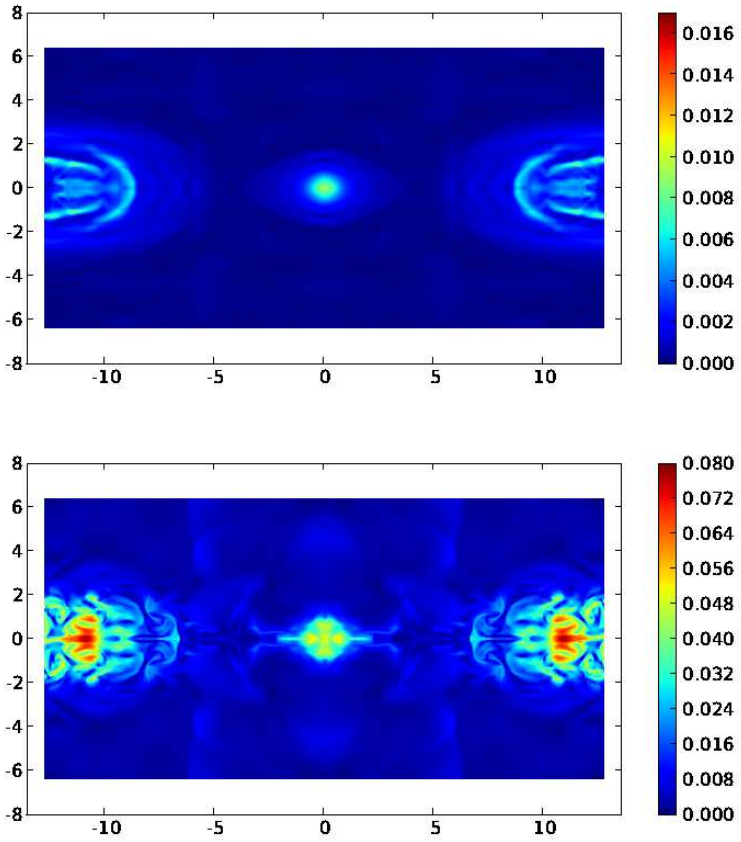

Figure 5.5: Total electron (top) and ion (bottom) momenta for the reconnection problem using PHM solver when $\chi = c_0$. This time, the solution produces secondary magnetic island in the middle of the domain.

starts to increase rapidly after about t = 15, during the onset of reconnection. From t = 15 to t = 20, the flux seems to be increasing almost linearly, with the slopes of flux vs. time for all solvers being almost equal.

The results start to diverge around t = 20, which corresponds to the time when the secondary island in the middle of the domain starts to appear. The evolution of the reconnected flux is slightly different when the island is present ($\chi = c_0$, Lorenz gauge, and Coulomb gauge), than when no island is present, but they all finish with about the same value, consistent with the finding of Loverich [28]. The test case with χ set to 4 is unique in that, the island does appear, but merges to one of the side lobes (Fig. 5.7, and Fig. 5.8). This is also reflected in the results, in which the reconnected flux increases greatly, and then decreases back to almost the normal value after the island disappears.

The discussion of the presence of secondary magnetic island during the magnetic reconnection process has been ongoing. Kuznetsova [26] observed the presence of the secondary magnetic island in the full particle simulation, whereas no such island was found in the hybrid simulation.

Similar findings can also be found in [30], which shows that no island is present for the Hall-MHD simulation, even though it does appear when current dependent resistivity is included in the system. The island also develops for the kinetic simulation. It is important to note, however, that the kinetic simulation performed by Kuznetsova et al. (as described in [31]) employs a charge conservation correction method developed by Langdon [32], implying that Gauss' law (Eq. 2.11) has somehow been incorporated into the solver.

It is not clear however, if the island should be physically present in the solutions. Loverich [28] reported that the increased dissipation of the third order discontinuous Galerkin eliminated the island. In the case of the second order solutions, however, it was instead found that increased dissipation actually led to the production of the island, similar to the finding of Hesse et al. [30], where the Hall-MHD solution with current dependent resistivity has the island.

5.3.1 Divergence Error Comparison for Magnetic Reconnection Problem

The divergence error of the electric and the magnetic field for the solutions of the reconnection problem are compared across the different solvers. The error is quantified using the l_2-norm formula as given in Eq. 4.13 and Eq. 4.16 for the magnetic and electric fields respectively.

For the results involving PHM solver, γ, the error correction potential for the magnetic field, is set to be equal to the speed of light. Test cases have been performed where γ is set to zero or greater than the speed of light, but the simulations crashed. Therefore, all of the PHM results shown in this section are obtained using a γ value that is equal to the speed of light. It is possible that the time-step needs to be reduced further to ensure stability with high values of γ. However, time-steps with CFL number of 0.1 and 0.05 have been tried, and the simulations keep crashing about half-way.

The values of χ, however, are varied from 0 to 4 times the speed of light. It is important to note that, in order for the simulation to be stable with correction potential greater than the speed of light, the time-step has to be reduced further. Using χc_0 as the restricting speed, which is one of the eigenvalues of the PHM equation system, a CFL number of 0.1 has to be used for cases where χ is set to $2c_0$ and $4c_0$.

Any value above 0.1 leads to instability for the two aforementioned values of χ. On the contrary, when χ is set to either 0 or c_0, a CFL number of 0.4 is enough to ensure stability. This implies that, using a χ of one, the simulation is about 8 times computationally less demanding compared to a χ value of $2c_0$, and about 16 times compared to a χ value of $4c_0$.

Observing the plot for the electric field divergence error vs. time, it can be seen that the PHM-based solver with χ set to one, two and four has a comparable solution with the Coulomb gauge based solver. The Lorenz gauge based solver has larger error compared the Coulomb solver and the PHM solver when $\chi = c_0$, but compared with the PHM solver with no correction for the electric field, the Lorenz solver is a little better, especially after the magnetic reconnection initiates (at t larger than 15).

It can also be seen from the results that the error seems to plateau after the formation of the secondary magnetic island (around t = 20 seconds), whereas the divergence error seems to keep increasing for the PHM solution with no χ correction term, for which no such island is observed (Fig. 5.4).

Interestingly, the magnetic island is not present in the solution when $\chi = 2c_0$ (Fig. 5.6), and while the island does show up for $\chi = 4c_0$ (the evolution of which is given in Fig. 5.7, and Fig. 5.8), it only does so temporarily before it merges to one of the side magnetic lobes.

But even without the presence of the island, the divergence error of the electric field when χ is set to 2 is comparable with other solutions that do produce the island, implying that satisfying the electric field divergence constraint may not be related to the production of the island.

However, it is important to note that the magnetic island is present only in simulations with the inclusion of Gauss' law. While satisfying the electric field divergence error does not seem to have an effect on the magnetic island, the presence of the magnetic island does seem to be highly dependent on the inclusion of charge separation.

To confirm, the spatial distribution of the divergence error of the electric field is shown in Fig. 5.13. The plot shows that, the spatial distribution is similar for PHM solutions with $\chi = c_0$, which has the island, and $\chi = 2c_0$, which has no island, and the Coulomb gauge solution. The divergence error is concentrated around the side magnetic sites for these cases, confirming that the magnetic island does not seem to have an effect on the divergence error.

For Lorenz gauge, however, there seems to be an equal amount of error between the side and the middle magnetic islands, which seems to imply that the magnetic island contributes to the divergence error. A possible explanation for this is the lack of charge conservation correction step for the Lorenz gauge formulation. While charge density is explicitly included in the Lorenz gauge formulation, charge conservation is never explicityly enforced, this could lead to larger divergence error as compared with the Coulomb gauge.

For Coulomb gauge, charge conservation is enforced through Eq. 2.24, and for PHM, the correction potential, ϕ, is directly correcting the electric field. This might explain why the divergence error is in general larger for Lorenz gauge solution, thus, also the divergence error around the secondary magnetic island.

Besides the Gauss' law, it is also important that the magnetic field satisfies the divergence constraint (Eq. 2.12). Finite volume simulations conducted with the PHM solver when γ is set to zero crash about half-way, just when the reconnection is about to initiate [28].

Fig. 5.14 shows a comparison between the two-fluid solvers tested here. It is evident from the plot (the bottom of Fig. 5.14) that the divergence error for the magnetic field is solely due to machine precision error. As described in Sec. 3.4.1, the magnetic field, $\mathbf{B}$, which is described as the curl of the vector potential, $\mathbf{A}$ (Eq. 2.20), is mathematically and numerically divergence-free in a uniform Cartesian grid (Eq. 3.22).

And even though χ is not the correction potential speed to correct the magnetic field, changing its values affects the divergence error of the magnetic field as well. When χ is set to one, the magnetic field has smaller divergence error compared to the solution when χ is set to zero. Increasing the value of χ above the speed of light does not seem to have the same effect as going from no correction to any correction. In fact, the divergence error of the magnetic field is slightly larger for $\chi = 2c_0$ and $\chi = 4c_0$ than when $\chi = c_0$.

5.4 Conclusion of the Cartesian Plasma Simulation Test Cases

Within this chapter, two-fluid plasma simulations are conducted in a uniform Cartesian grid. The two-fluid generalization of the Brio-Wu MHD Shock problem [22] and the GEM magnetic reconnection challenge are tested.

The results obtained from the different solvers are compared, and it was found that, for the two-fluid shock problem, all tested solvers produce qualitatively similar results, as can be observed from Fig. 5.1 and Fig. 5.2). The divergence error of the electric field are also computed and compared, and the Lorenz gauge solver is the one with the largest divergence error. The values of χ, however, do not seem to significantly affect the divergence error of the solution in this case, and the PHM based solvers seem to perform better than the potential based solvers.

Similar analysis is also done with the magnetic reconnection, which is the more challenging problem of the two. Reconnected flux (Eq. 5.14) is computed and compared, and all solvers capture the reconnection physics quite well. The solutions with the presence of secondary magnetic island reconnect more flux than the ones without.

It is interesting to note that all simulations in which the secondary island is present include the charge density (Eq. 2.11) either directly or indirectly. The divergence constraint equation for the electric field is approximated in the PHM through Eq. 2.17, whereas the update equation for the scalar potential, Eq. 2.21 and Eq. 2.26 represent the Gauss' law. The magnetic island, then, seems to be highly dependent on charge separation.

However, no magnetic island is present in the PHM simulation with χ set to $2c_0$. And while the magnetic island affects the reconnected flux, the divergence error does not seem to be significantly affected by the presence of the magnetic island. While the PHM solution with no electric field correction does indeed have larger divergence error, when $\chi = 2c_0$, the divergence error is comparable with the divergence error of the solution that has the island.

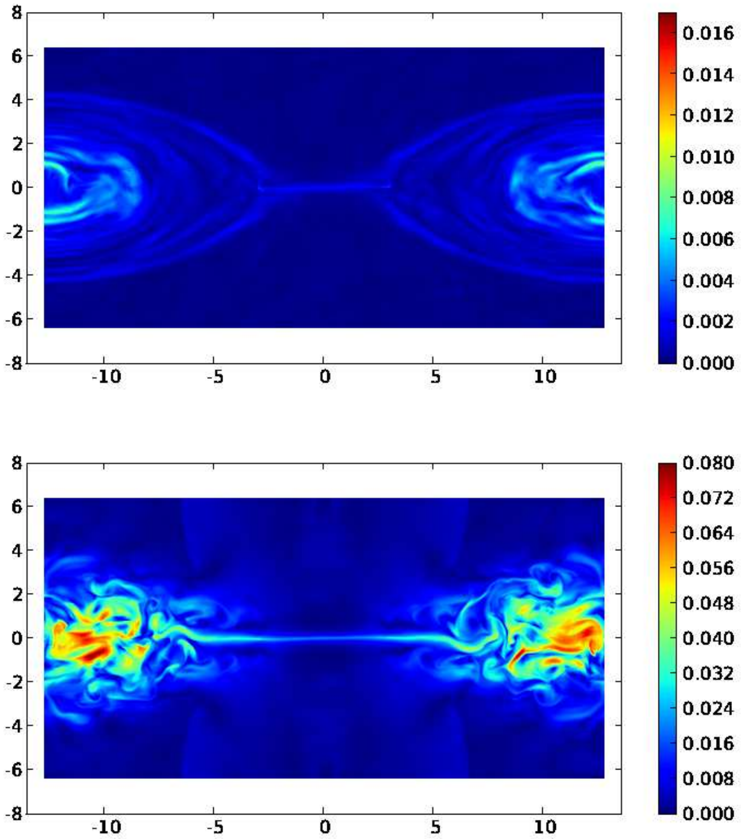

Figure 5.6: Total electron (top) and ion (bottom) momenta for the reconnection problem using PHM solver with χ set to $2c_0$. The solutions here look similar with the ones produced by the PHM simulation with no electric field correction term. No island is observed in the middle of the domain, even though charge separation exists and is included in the system (through χ)

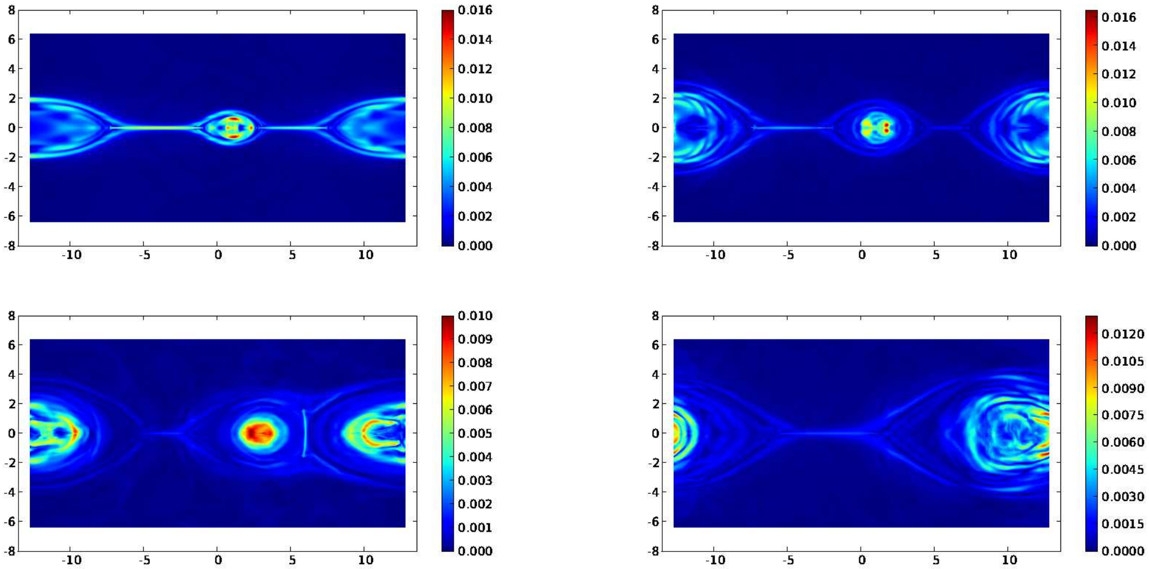

Figure 5.7: Evolution of the total electron momentum of the solution using PHM with χ set to $4c_0$. Going from left to right, top to bottom, the times shown here are $t = 20$, $t = 24$, $t = 32$, and $t = 40$. The island shows up about halfway the simulation, and then merges to one of the side magnetic lobes.

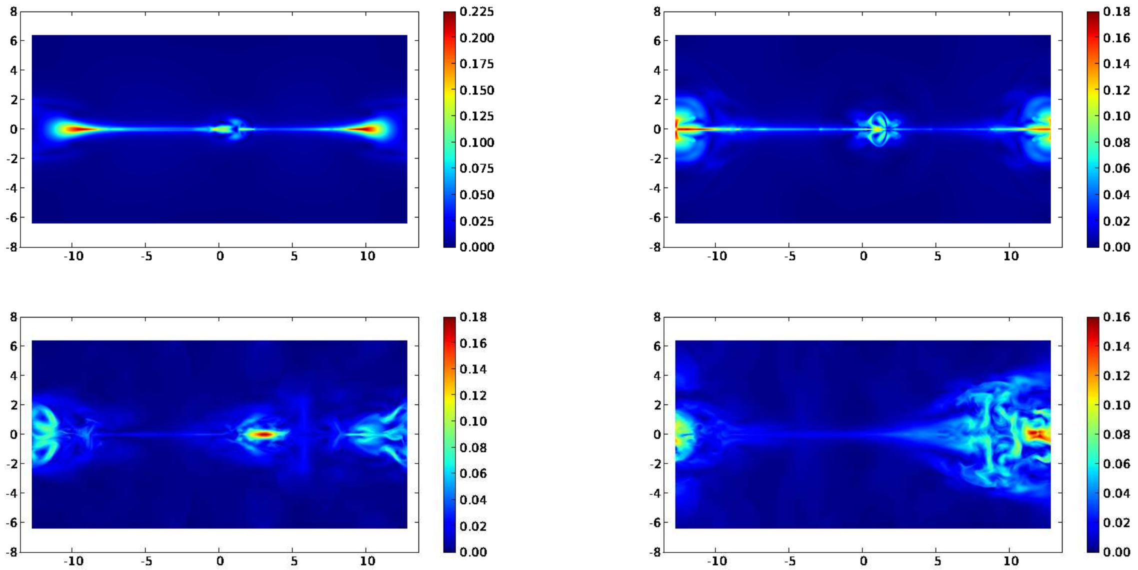

Figure 5.8: Evolution of the total ion momentum of the solution using PHM with χ set to $4c_0$. Going from left to right, top to bottom, the times shown here are $t = 20$, $t = 24$, $t = 32$, and $t = 40$. The island shows up about halfway the simulation, and then merges to one of the side magnetic lobes.

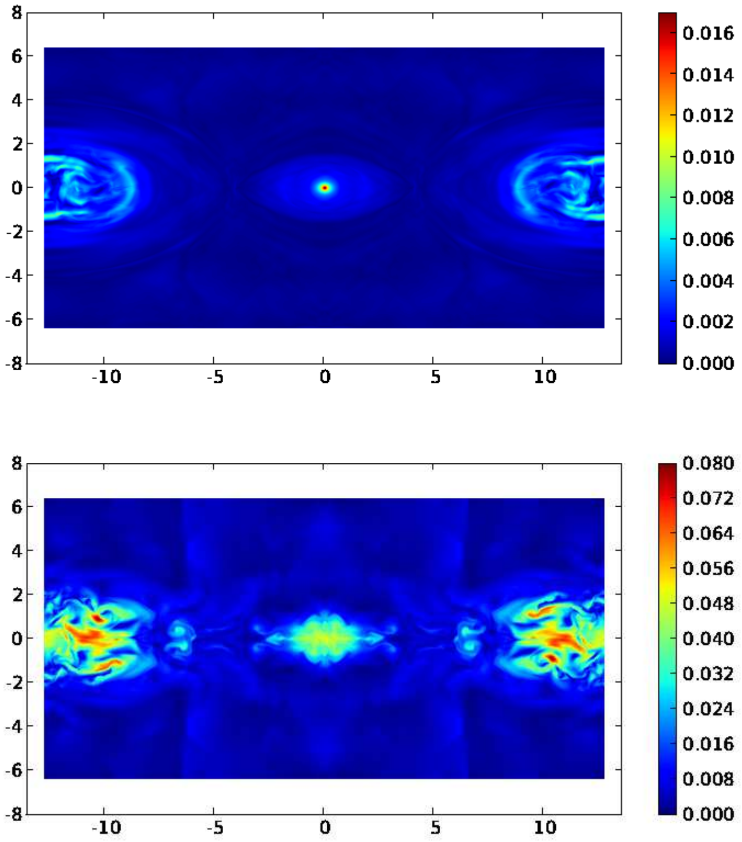

Figure 5.9: Total electron (top) and ion (bottom) momenta for the reconnection problem using Lorenz solver. Once again the secondary island is observed in the middle of the domain. The Lorenz gauge formulation advances the scalar potential through a second order wave equation with the charge density as its source term. This, once again, confirms that charge density plays a role in the production of the magnetic island.

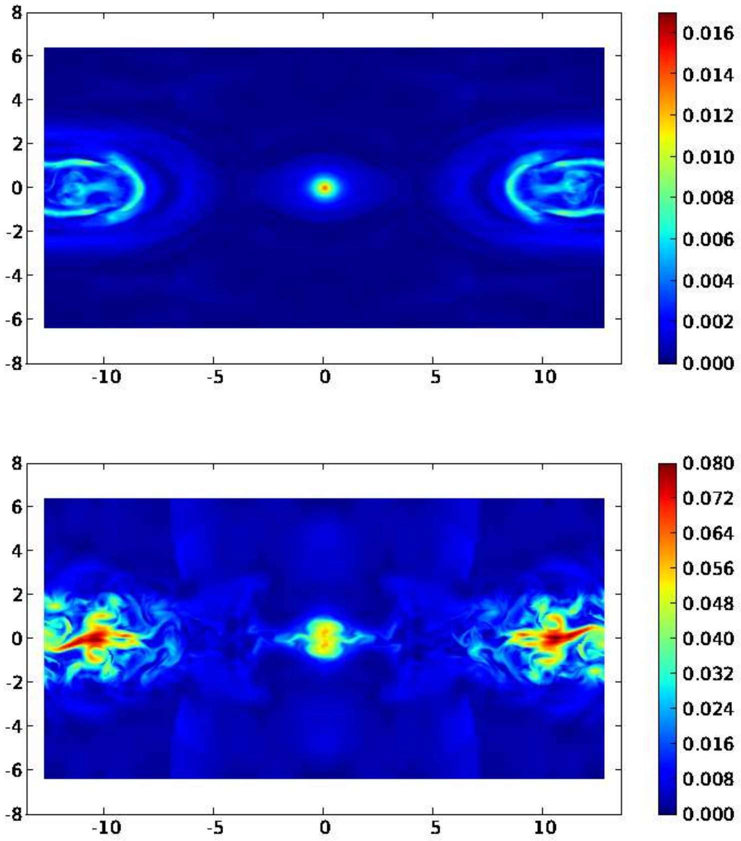

Figure 5.10: Total electron (top) and ion (bottom) momenta for the reconnection problem using Coulomb solver. The update equation for the scalar potential in the Coulomb gauge formulation is a Poisson's equation with the charge density as the source term, to which the production of the secondary island may be related

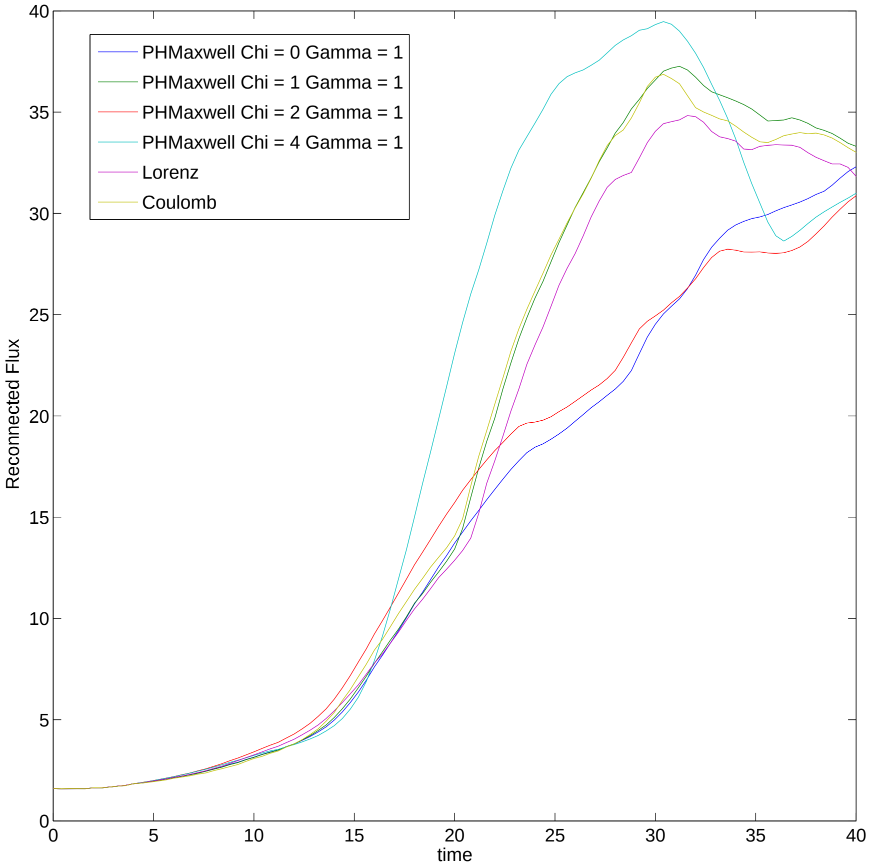

Figure 5.11: Comparison of reconnected flux vs. time of the solutions produced by the different electromagnetic solvers for the two-fluid plasma system. The solutions with the secondary island (PHM with $\chi = c_0$, Lorenz, Coulomb gauge solutions) seem to reconnect more flux. When $\chi = 4c_0$, the flux increases, and then decreases when the island is in the process of merging to one of the lobes. In the end, the flux approaches that of the solutions with no island once the island has fully merged with one of the side lobes.

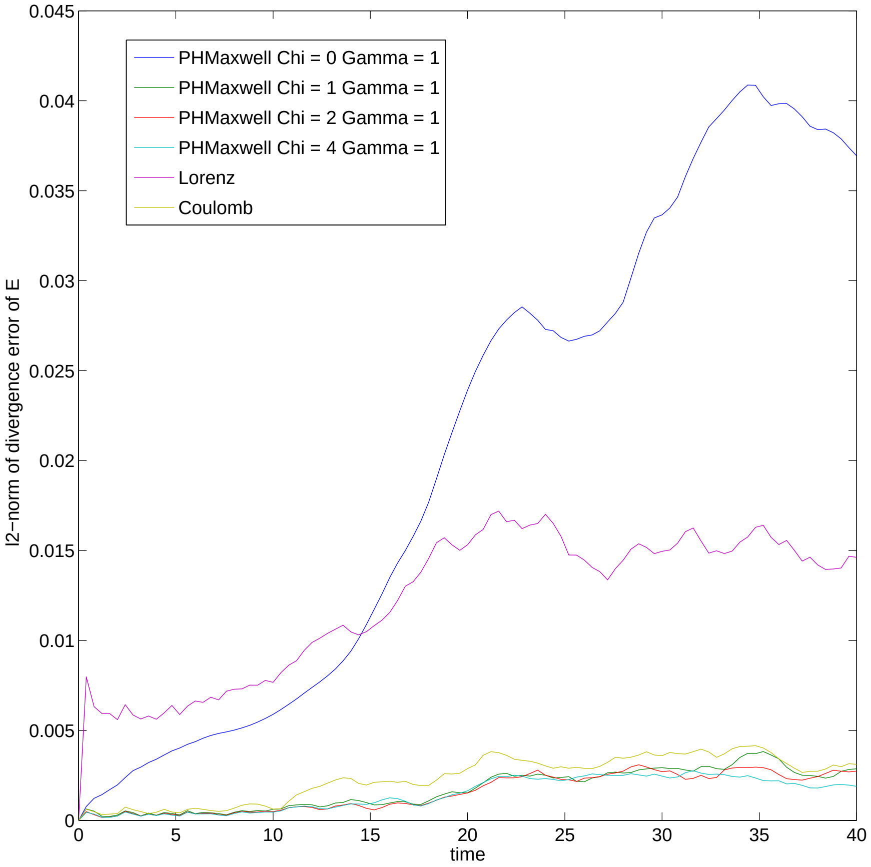

Figure 5.12: Comparison of l_2-norm of the divergence error of the electric field for the GEM Challenge Reconnection problem. The higher values of χ here do not seem to significantly reduce the divergence error of the electric field. The magnetic island does not seem to affect the divergence error. When $\chi = 0$, the divergence error is higher, but the divergence error is comparable between $\chi = c_0$, and $\chi = 2c_0$.

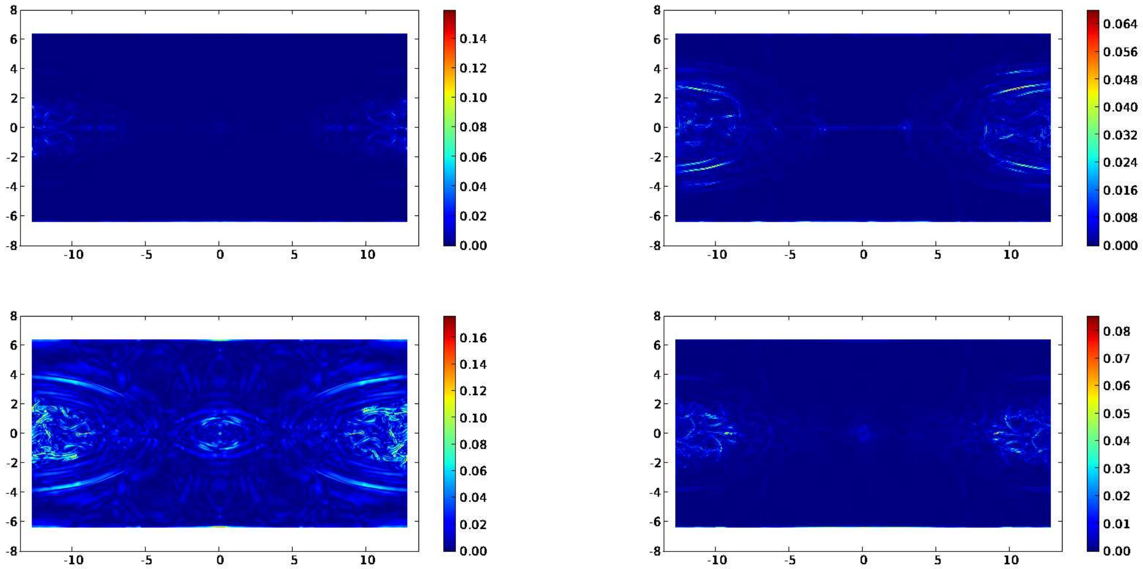

Figure 5.13: The spatial variation of the divergence of **E** error for the tested electromagnetic solvers. Going from left to right, and top to bottom, the solvers used are: PHM with $\chi = c_0$, PHM with $\chi = 2c_0$, potential solver using Lorenz gauge formulation, and potential solver using Coulomb gauge formulation. While PHM solution with $\chi = 2c_0$ has no island, its distribution of the divergence error of **E** is similar with the Coulomb gauge and the ohter PHM solutions. For all cases, the error seems to be concentrating on the sides rather than in the middle.

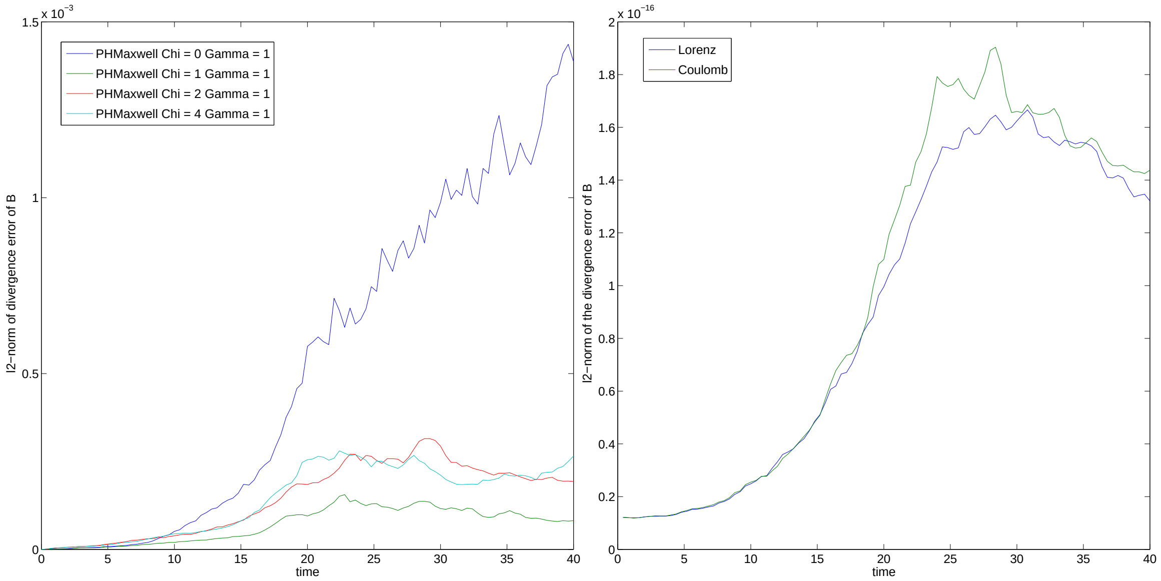

Figure 5.14: Comparison of l_2-norm of the divergence error of the magnetic field for the GEM Challenge Reconnection problem. The left plot shows the error for the PHM solvers, whereas the right one shows the error for the potential solvers. The divergence error of the magnetic field is down to the machine accuracy for the potential solvers. With PHM, χ does not really affect the divergence error. For all test cases, $\gamma = c_0$. Different values of γ lead to instability and the simulations crash in before they finish.

Chapter 6
CONCLUSIONS AND POSSIBLE FUTURE WORK

6.1 Concluding Remarks

Electromagnetic solvers for use with the two-fluid plasma system are proposed. These electromagnetic solvers directly incorporate the divergence constraints of the Maxwell's equations. The first proposed method is the perfectly hyperbolic Maxwell's equations, which deal with the divergence error by convecting it out of the domain. The second one is by expressing the electromagnetic fields in terms of potentials.

Purely electromagnetic cases and plasma tests are conducted. The potential solvers are found to produce solutions with lower divergence error for the purely electromagnetic test cases, whereas the PHM system performs better for plasma. This is likely due to the wave propagation method that is used to solve the PHM system. Shocks may develop in plasma simulations, and the wave propagation method that is used to solve the PHM system is found to be more suitable to handle problems with discontinuities.

Both the PHM and the potential solvers are also found to perform better than just the regular Maxwell's equations. However, increasing the values of the correction potential speeds, χ, and γ, does not seem to significantly affect the solutions.

The solvers are also tried to solve the GEM magnetic reconnection challenge. It is found that, in some cases, magnetic island is present in the middle of the domain, but in some others, it is not. After observing the magnitude and the spatial variation of the divergence error, it can be seen that the divergence error does not seem to be related to the formation of this magnetic island. However, all of the solutions that produce the secondary magnetic island have Gauss' law somewhat incorporated into the electromagnetic solver, which implies that the production of the secondary island may be related to the inclusion of the charge density in the evolution equations.

6.2 Possible Future Work

This work is far from being done, and while some simulations have been attempted, they involve far less variables and lack the complexities of realistic plasma experiments or physical phenomena. The algorithms developed in this paper are developed for two dimensions, whereas most plasma devices are three dimensional devices. Therefore, one of the next logical developmental steps will be to extend the algorithms presented here to three dimensions.

The extension to 3D for the wave propagation method is straight forward. The nature of the Riemann problem for each interface will remain the same, and if it is solved by rotating the solutions to the local coordinate systems, then the method can be easily extended to three dimensions. The Riemann problem will still be solved on each interface by finding the waves that travel between the two adjacent cells. However, compared to 2D, the rotation processes will now involve more operations, and a significant hit on performance is expected when compared to solving the wave propagation method on Cartesian grid. Coming up with an efficient way to do rotations in 3D is a subject worth pursuing for future studies.

Another possible extension to the current work will be to implement the Darwin approximation for Vlasov-Maxwell system [33] here. As mentioned in Sec. 4.6.1.1, solvers based on the full Maxwell's equations are restricted by the speed of light, which sets the the CFL stability limit. This limitation can be very restrictive for a plasma, because the only other speed to restrict the stability constraint will be the speed of sound, which is an eigenvalue of the Euler equations.

The speed of sound should be much slower than the speed of light, and restricting the stability limit based on the speed of light might not be computationally efficient, especially for low-frequency, collisionless phenomena, for which high frequency physics are irrelevant. The problem with the Darwin approximation is that, it requires solving 8 elliptic equations for each time step, and it is not clear if being able to use much larger time-steps than in Maxwell's system will provide a net gain in computational expenses. To analyze how the Darwin model compares to the regular two-fluid system in terms of accuracy and computational time will be one of the possible topics for further studies.

Lastly, while realistic plasma simulations require complicated domain and boundary conditions, plasma simulations tested in this paper are all done in Cartesian coordinate system. Generating grid and boundary conditions for realistic plasma simulations so that the two-fluid plasma system and the electromagnetic solvers developed here can be applied will be the eventual goal of this project.

BIBLIOGRAPHY

[1] A. Hakim, J. Loverich, and U. Shumlak. A high resolution wave propagation scheme for ideal Two-Fluid plasma equations. *Journal of Computational Physics*, 219:418–442, 2006.

[2] J. Loverich and U. Shumlak. A discontinuous galerkin method for the full two-fluid plasma model. *Computer Physics Communications*, 169:251–255, 2005.

[3] B. Srinivasan. A comparison between the discontinuous galerkin method and the high resolution wave propagation algorithm for the full two-fluid plasma model. Master's thesis, University of Washington, 2006.

[4] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[5] R. J. LeVeque. Clawpack (Conservation Laws Package) Website. http://www.amath.washington.edu/ claw/clawpack.org.

[6] The HDF Group. HDF5 library documentation, 2008. http://www.hdfgroup.org/HDF5/doc/index.html.

[7] W. Gropp, E. Lusk, D. Ashton, P. Balaji, D. Buntinas, R. Butler, A. Chan, D. Goodell, J. Krishna, G. Mercier, R. Ross, R. Thakur, B. ToonenW. Gropp, E. Lusk, D. Ashton, P. Balaji, D. Buntinas, R. Butler, A. Chan, D. Goodell, J. Krishna, G. Mercier, R. Ross, R. Thakur, and B. Toonen. MPICH2 user's guide. Technical Report Version 1.0.8, Argonne National Laboratory, 2008.

[8] The Open MPI Development Team. Open MPI: Open source high performance computing, 2004. http://www.open-mpi.org/.

[9] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[10] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999.

[11] T. Manteuffel, S. McCormick, J. Ruge, M. Brezine, M. Griebel, D. Keyes, R. Lazarov, J. Xu, L. Zikatanov, and S. Schaffer. hypre user's manual. Technical Report Version 2.0.0, Lawrence Livermore National Laboratory, 2006.

[12] Xiaoye S. Li and James W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Mathematical Software*, 29(2):110–140, June 2003.

[13] M. Torrilhon and H. Struchtrup. Regularized 13-moment equations: shock structure calculations and comparison to burnett models. *Journal of Fluid Mechanics*, 513:171198, 2004.

[14] C. D. Levermore and W. J. Morokoff. The gaussian moment closure for gas dynamics. *SIAM Journal on Applied Mathematics*, 59(1):72–96, 1998.

[15] A. Hakim. Extended mhd modelling with the ten-moment equations. *Journal of Fusion Energy*, 27:36–43, 2008.

[16] C.-D. Munz, P. Ommes, and R. Schneider. A three-dimensional finite-volume solver for the maxwell equations with divergence cleaning on unstructured meshes. *Computer Physics Communications*, 130:83–117, 2000.

[17] B. Cockburn and C. Shu. Runge-Kutta discontinous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173–261, 2001.

[18] I. Faille. A control volume method to solve an elliptic equation on a two-dimensional irregular mesh. *Computer Methods in Applied Mechanics and Engineering*, 100:275–290, 1992.

[19] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[20] S. M. Rao. *Time Domain Electromagnetics*. Academic Press, 1999.

[21] C. Aberle. Algorithm for solving colocated electromagnetic fields including sources. Master's thesis, University of Washington, 2004.

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

[23] U. Shumlak and J. Loverich. Approximate riemann solver for the two-fluid plasma model. *Journal of Computational Physics*, 187(2):620–638, 2003.

[24] J. Birn, J. F. Drake, M. A. Shay, B. N. Rogers, R. E. Denton, M. Hesse, M. Kuznetsova, Z. W. Ma, A. Bhattacharjee, A. Otto, and P. L. Pritchett. Geospace environmental modelling (gem) magnetic reconnection challenge. *Journal of Geophysical Research*, 106(A3):3715–3719, 2001.

[25] A. Otto. Geospace environment modeling (gem) magnetic reconnection challenge: Mhd and hall mhd constant and current dependent resistivity models. *Journal of Geophysical Research*, 106(A3):3751–3757, 2001.

[26] M.M. Kuznetsova, M. Hesse, and D. Winske. Collisionless reconnection supported by nongyrotropic pressure effects in hybrid and particle simulations. *Journal of Geophysical Research*, 106(A3):3799–3810, 2001.

[27] P. Pritchett. Geospace environment modeling magnetic reconnection challenge: simulations with a full particle electromagnetic code. *Journal of Geophysical Research*, 106(A3):3783–3798, 2001.

[28] J. Loverich. *A Discontinuous Galerkin Method for the Two-Fluid Plasma System and Its Application to the Z-Pinch*. PhD thesis, University of Washington, December 2005.

[29] E. G. Harris. On a plasma sheet separating regions of oppositely directed magnetic field. *Nuovo cimento*, 25:385, 1962.

[30] M. Hesse, J. Birn, and M. Kuznetsova. Collisionless magnetic reconnection: electron processes and transport modeling. *Journal of Geophysical Research*, 106(A3):3721–3735, 2001.

[31] M. Hesse, K. Schindler, J. Birn, and M. Kuznetsova. The diffusion region in collisionless magnetic reconnection. *Physics of Plasmas*, 6(5):1781–1795, 1999.

[32] A. B. Langdon. On enforcing gauss's law in electromagnetic particle-in-cell codes. *Computer Physics Communications*, 70:447–450, 1992.

[33] N. Besse, N. J. Mauser, , and E. Sonnendrucker. Numerical approximation of self-consistent vlasov models for low-frequency electromagnetic phenomena. *International Journal of Applied Mathematics and Computer Science*, 17:361–374, 2007.

[34] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation*. McGraw-Hill Book Company, 1985.

[35] 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.

[36] D. A. Calhoun, C. Helzel, and R. J. LeVeque. Logically rectangular grid and finite volume methods for pdes in circular and spherical domains. *SIAM Review*, 50:723–752, 2008.

[37] M. A. Shay, J. F. Drake, B. N. Rogers, and R. E. Denton. Alfvenic collisionless magnetic reconnection and the hall term. *Journal of Geophysical Research*, 106(A3):3759–3772, 2001.

[38] Z. Ma and A. Bhattacharjee. Hall magnetohydrodynamic reconnection: the geospace environment modeling challenge. *Journal of Geophysical Research*, 106(A3):3773–3782, 2001.

[39] W. Daughton, J. Scudder, and H. Karimabadi. Fully kinetic simulations of undriven magnetic reconnection with open boundary conditions. *Physics of Plasmas*, 13(072101), 2006.

[40] Ammar Hakim, Bhuvana Srinivasan, Robert Lilly, Andree Susanto, Eder Sousa, and Uri Shumlak. WarpX Wiki, 2008. http://warpx.org/wiki.