A PARTICLE-IN-CELL METHOD FOR THE SIMULATION OF PLASMAS BASED ON AN UNCONDITIONALLY STABLE WAVE EQUATION SOLVER

By

Eric Matthew Wolf

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Applied Mathematics - Doctor of Philosophy

2015

ABSTRACT

A PARTICLE-IN-CELL METHOD FOR THE SIMULATION OF PLASMAS BASED ON AN UNCONDITIONALLY STABLE WAVE EQUATION SOLVER

By

Eric Matthew Wolf

In this thesis, we present a particle-in-cell method for the simulation of plasmas based on an unconditionally stable solver for the second-order scalar wave equation, that is, a wave equation solver that is not subject to a Courant-Friedrichs-Lewy (CFL) stability restriction, typical of explicit methods, while maintaining a computational cost and code complexity comparable to such explicit solvers. This permits the use of a time step size many times larger than allowed by widely-used explicit methods.

TABLE OF CONTENTS

Chapter 1 Introduction **1**
1.1 Overview 4
1.2 Mathematical Models 6
1.3 Nondimensionalization and Asymptotic Analysis 8
1.3.1 Electrostatic Case 9
1.3.2 Electromagnetic Case 12

Chapter 2 Description of the wave equation solver **14**
2.1 Origin and Mitigation of the CFL Restriction 14
2.2 Derivation of Second-Order Semi-Discrete Schemes With the Method of Lines Transpose 17
2.2.1 Dispersive Semi-Discrete Scheme 18
2.2.2 Diffusive Semi-Discrete Scheme 18
2.3 Semi-Discrete Von Neumann Analysis of the Second Order Schemes 19
2.3.1 Dispersive Scheme 20
2.3.2 Diffusive Scheme 20
2.4 Solution of the Modified Helmholtz Equation 22
2.4.1 Dimensional Splitting 23
2.4.2 Integral Solution Method in 1D 24
2.4.3 Fast Numerical Evaluation of the 1D Convolution Operator 25
2.5 Fully Discrete Von Neumann Analysis 27
2.6 Artificial Dissipation in Centered Schemes 33
2.6.1 Artificial Dissipation in 1D 33
2.6.2 Artificial Dissipation in 2D 35
2.7 Derivation of a Fourth-Order Dissipative Newmark Scheme 36
2.8 Embbeded Boundary Method for Neumann Boundary Conditions 40
2.8.1 Description of the Two-Point Boundary Correction Method 42
2.8.2 Description of the Embedded Boundary Method and Proof of Convergence of the Iterative Solution in 1D 42
2.8.3 Description of the Method in 2D 48
2.9 Summary Table For Second-Order Wave Solver 51
2.10 Numerical Examples 54
2.10.1 Rectangular Cavity With Domain Decomposition 54
2.10.2 Double Circle Cavity 55
2.10.3 Periodic Slit Diffraction Grating 58
2.10.4 Bessel Mode with Neumann Boundary Conditions 59

Chapter 3 Description of the particle-in-cell method **65**
3.1 Particle Weighting Scheme 65
3.2 Method for Controlling Divergence Error 66

3.3 Particle Equations of Motion . . . 67
3.4 Fast Convolution Algorithm For Particle Sources . . . 68
3.4.1 Fast Convolution Algorithm in 1D . . . 68
3.4.1.1 Local Deposit Step in 1D . . . 69
3.4.1.2 Recursive Sweep Step in 1D . . . 71
3.4.2 Fast Convolution Algorithm in 2D . . . 72
3.4.2.1 Local Deposit Step in 2D . . . 72
3.4.2.2 Recursive Sweep Step in 2D . . . 74
3.5 Particle Boundary Conditions . . . 75
3.5.1 1D Periodic BC . . . 75
3.5.2 2D Periodic BC . . . 76
3.5.3 1D Dirichlet BC . . . 76
3.5.3.1 Analysis of the First Approach . . . 77
3.5.3.2 Analysis of the Second Approach . . . 79
3.5.3.3 Comparison of the Approaches . . . 81
3.5.4 2D Dirichlet BC . . . 82
3.6 Numerical Results . . . 82
3.6.1 Electrostatic Test Problems . . . 82
3.6.1.1 Cold Plasma Langmuir Wave . . . 83
3.6.1.2 Two Stream Instability . . . 84
3.6.1.3 Landau Damping . . . 85
3.6.1.4 Sheath Formation in a Bounded 1D Plasma . . . 88
3.6.2 Electromagnetic Test Problems . . . 89
3.6.2.1 Bennett Pinch Problem . . . 89
3.6.2.2 Mardahl Beam Problem . . . 94

Chapter 4 Conclusion . . . 102

BIBLIOGRAPHY . . . 103

Chapter 1

Introduction

Collisionless plasmas - systems of charged particles interacting through electromagnetic fields - are modelled by the Vlasov-Maxwell system of partial differential equations (PDEs), which couple Maxwell's equations, describing the evolution of the electric and magnetic fields $\mathbf{E}$ and $\mathbf{B}$, to Vlasov equations, a type of hyperbolic PDE describing the evolution of the phase-space distribution functions (DFs) f_s of the various species s of charged particles. Particle-in-cell (PIC) methods [4, 25, 55], in development and use since the 1960s and a primary tool in the computer simulation of plasmas, combine an Eulerian description of the fields with a Lagrangian description of the DFs; that is, fields are evolved on a fixed mesh, while DFs are represented by moving particles whose trajectories are characteristics of the corresponding Vlasov equation. Thus, PIC methods require a method to compute the fields on a mesh and a method to compute particle trajectories, as well as interpolation tools to provide for their coupling. This work focuses on a new method for the computation of the fields, along with associated interpolation techniques.

Under the Lorenz gauge condition, Maxwell's equations reduce to uncoupled wave equations for the scalar and vector potentials, Φ and $\mathbf{A}$. Recently, a novel method for the solution of the wave equation has been developed [10, 9, 11], based on the Method of Lines Transpose (MOLT), dimensional splitting and an efficient 1D integral solution method, which is unconditionally stable (or A-stable) - that is, it is not subject to the Courant-Friedrichs-Lewy (CFL) restriction limiting the ratio of the temporal step size to the spatial step size, typical

of widely used explicit methods. In this work, we apply this method to the uncoupled wave equations for Φ and $\mathbf{A}$ to solve Maxwell's equations with a method comparable in computational cost and complexity of code to explicit methods such as the well-known Yee scheme [56, 53], but without introducing a CFL restriction based on the speed of light as in such explicit methods. In conjunction with an appropriate description of particles, we seek to develop a PIC method that retains the simplicity of explicit finite-difference-based methods while eliminating this CFL restriction.

We demonstrate the application of our method to both electrostatic and electromagnetic problems. In the non-relativistic, zero-magnetic field limit, it is typical to make the electrostatic approximation, $\mathbf{E} = -\nabla\Phi$, $-\nabla^2\Phi = \rho/\epsilon_0$, simplifying the Vlasov-Maxwell system to the Vlasov-Poisson system. Correspondingly, in this work we consider the same non-relativistic, zero-magnetic field limit, and drop $\mathbf{A}$ and the corresponding wave equations from our model. When particle velocities are small compared to the speed of light, we argue that this model, which we term the Vlasov-Wave model due to the replacement of the Poisson equation with a wave equation, will agree approximately with the usual electrostatic Vlasov-Poisson model. We present numerical results applying our method to several standard electrostatic test problems in one and two dimensions, showing agreement with the predictions of linear theory for the electrostatic model. We apply our method to a fully electromagnetic beam pinch problem in two dimensions. It is well known that an electromagnetic PIC method must satisfy a discrete form of Gauss' law ($\nabla \cdot \mathbf{E} = \rho/\epsilon_0$) to prevent serious numerical errors related to the violation of charge conservation [37]. In this work, we obtain solutions that satisfy exactly discrete forms of Gauss' law for the electric field and the divergence-free condition for the magnetic field through a staggered grid approach, adapted from the well-known Yee grid [56], with a Poisson equation formulation for the scalar

potential. In addition to eliminating the CFL restriction, the wave solver method used in this work offers the handling of complex boundary geometry in a Cartesian grid without using a staircasing approximation [9, 54], and can be extended to higher-order accuracy [11], features that will be incorporated into our PIC method in future work.

Building on prior work on unconditionally stable ADI-FDTD schemes for Maxwell's equations [58, 57, 42, 34], unconditionally stable ADI-FDTD methods for Maxwell's equations that conserve the divergence of the discrete electric and magnetic fields were incorporated into PIC methods in [51], using the method of [20] to handle complex geometries. Further, a PIC method based on a high-order discontinuous Galerkin scheme for Maxwell's equations was developed in [27, 26], allowing for the handling of complex geometries through unstructured meshes, but which is still subject to CFL restrictions. Future work will compare our method to these approaches in the context of PIC methods.

In most PIC methods, further stability restrictions also apply, the most restrictive often being the need to resolve the electron plasma period. Our method, making use of explicit algorithms for advancing particles, is subject to such restriction. In problems where time scales much longer than the electron plasma period are of primary interest and dynamics on the scale of the electron plasma period can be safely underresolved - for instance, in certain problems in the study of ion dynamics - it is desirable to take a much larger time step than prescribed by the plasma period stability restriction. Since years ago [18, 7, 33, 24], and with a recent resurgence [13, 38], it has been sought to develop implicit PIC algorithms that are not subject to the stability restriction based on the plasma period (or the cyclotron period). An ultimate challenge for our method would be synthesis with a suitable implicit particle integration scheme to achieve a practical fully implicit method, eliminating the stability restriction based on the plasma period as well as the field-based CFL restriction. However,

that is beyond the scope of the present work.

1.1 Overview

We now describe the content of this dissertation, and point out the original contributions given in it. This dissertation builds on the implicit wave solver developed in [10, 9, 11]. The contributions of the present author to [9] include the development of an embedded boundary method for Neumann boundary conditions on a complex boundary geometry and several numerical examples, which are reproduced in this dissertation. The development of the embedded boundary method for Neumann boundary conditions is important, as it is necessary to impose perfect electric conducting (PEC) boundary conditions with our formulation of Maxwell's equations. An analysis in [28] suggests that such a method will be unstable without artificial dissipation, so the present author developed a method for artificial dissipation in centered schemes that maintains unconditional stablity. Further contributions of the present author consist of the developement of a PIC method for the simulation of plasmas based on the implicit wave solver, which is given in this dissertation. Important aspects of this PIC method are a fast convolution algorithm for particle source terms and a staggered-grid approach to enforcing divergence conditions. We now describe the structure of the present work with respect to the previous work on the implicit wave solver and the original contributions of the present author.

In Section 1.2, we describe the mathematical models used in the PIC method, and in Section 1.3, we describe the nondimensionalizations used in this work. In Section 2.1, we give a heuristic explanation of the CFL restriction in typical explicit wave solvers along with a method to mitigate this restriction, and in Section 2.2, we describe the derivation of two

second-order accurate semi-discrete schemes, describing work from [11] and [9]. In Section 2.3, we give a semi-discrete Von Neumann analysis of these schemes, which is an original contibution. In Section 2.4, we describe the fast numerical method used for an integral solution, describing work from [9].

The remainder of the dissertation is original work of the present author. In Section 2.5, we give a proof that the fully discrete scheme is unconditionally stable, in the sense of Von Neumann analysis. In Section 2.6, we describe a method for introducing artificial dissipation into centered schemes of arbitrary order accuracy while maintaining unconditional stability. In Section 2.7, we describe a fourth-order accurate Newmark method for the wave equation that possesses dissipation. In Section 2.8, we describe an embedded boundary method for the wave equation with Neumann boundary conditions on a complex boundary geometry. In Section 2.9, we give a table that summarizes the implicit wave solver (including the previous work described above). In Section 2.10, we present several numerical examples of the implicit wave equation solver.

The PIC method is described in Chapter 3. The charge and current weighting schemes used are given in Section 3.1, and a method for reducing divergence error to machine precision is given in Section 3.2. The particle equations of motion are described in Section 3.3. A fast algorithm for evaluating convolutions of particle source terms is given in Section 3.4. Some issues with imposing boundary conditions with particles are described in Section 3.5. In Section 3.6, we present several numerical examples using our PIC method. Finally, we conclude the dissertation with a brief discussion in Chapter 4.

1.2 Mathematical Models

The self-consistent evolution of a collisionless, single species plasma is described by the Vlasov-Maxwell system, given in the SI system of units by

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = 0 \tag{1.1}$$

$$\epsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{B} - \mu_0 \mathbf{J} \qquad \frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{1.2}$$

$$\nabla \cdot \mathbf{E} = \rho / \varepsilon_0 \qquad \nabla \cdot \mathbf{B} = 0 \tag{1.3}$$

$$\rho(\mathbf{x}, t) = q \int_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t)\, d\mathbf{v}, \quad \mathbf{J} = q \int_{\mathbf{v}} \mathbf{v} f(\mathbf{x}, \mathbf{v}, t)\, d\mathbf{v} \tag{1.4}$$

where $f(\mathbf{x}, \mathbf{v}, t)$ is the phase-space distribution function, q is the charge and m is the mass of a particle, $\mathbf{E}(\mathbf{x}, t)$ is the electric field, $\mathbf{B}(\mathbf{x}, t)$ is the magnetic field, $\rho(\mathbf{x}, t)$ is the charge density (and ρ_B represents a static uniform background charge distribution), $\mathbf{J}(\mathbf{x}, t)$ is the current density, ε_0 is the electric permittivity of the vacuum, and μ_0 is the magnetic permability of the vacuum. (Boldface variables are to stand for vector quantities, while nonboldface variables are to stand for scalar quantities.)

In the limit, in an appropriate sense, of $|\mathbf{v}|/c \to 0$ (where $c = 1/\sqrt{\varepsilon_0 \mu_0}$ is the speed of light in vacuum) and in the absence of magnetic fields, the Vlasov-Maxwell system reduces

to the Vlasov-Poisson system:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f - \frac{q}{m} \nabla \Phi \cdot \nabla_{\mathbf{v}} f = 0 \tag{1.5}$$

$$-\Delta \Phi = \rho/\varepsilon_0, \quad \rho(\mathbf{x}, t) = q \int_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t)\, d\mathbf{v} \tag{1.6}$$

The Vlasov-Maxwell and Vlasov-Poisson models have been widely studied, and many numerical methods have been developed to solve them, such as electrostatic and electromagnetic PIC methods [4, 25, 17, 39, 23, 27, 51], as well as Eulerian methods [22, 45, 1, 16, 15] and semi-Lagrangian methods [14, 52, 47, 48]. Instead of solving either of these systems directly, we seek to develop a semi-implicit approach for the fields based on a vector potential formulation. Under the Lorenz gauge condition,

$$\frac{1}{c^2} \frac{\partial \Phi}{\partial t} + \nabla \cdot \mathbf{A} = 0,$$

Maxwell's equations reduce to uncoupled wave equations:

$$\frac{1}{c^2} \frac{\partial^2 \Phi}{\partial t^2} - \Delta \Phi = \rho/\varepsilon_0 \tag{1.7}$$

$$\frac{1}{c^2} \frac{\partial^2 \mathbf{A}}{\partial t^2} - \Delta \mathbf{A} = \mu_0 \mathbf{J} \tag{1.8}$$

with $\mathbf{E} = -\nabla \Phi - \frac{\partial \mathbf{A}}{\partial t}$ and $\mathbf{B} = \nabla \times \mathbf{A}$ [31]. A similar potential-based approach was taken in [40], which used the Coulomb gauge condition ($\nabla \cdot \mathbf{A} = 0$), resulting in more a complicated set of coupled equations, which they solved using explicit finite difference methods. We choose to work insead in the Lorenz guage, as the resulting uncoupled wave

equations are simpler to handle numerically. We note that perfect electric conducting (PEC) boundary conditions may be imposed on the potentials by setting $\Phi = 0$, $\mathbf{A} \times \mathbf{n} = \mathbf{0}$ and $(\nabla(\mathbf{A} \cdot \mathbf{n})) \cdot \mathbf{n} = 0$ on the boundary, where $\mathbf{n}$ denotes the normal to the boundary.

In the electromagnetic case, we will solve these wave equations coupled into a particle-in-cell method to constitute a solution of the usual Vlasov-Maxwell system. In the non-relativistic, zero-magnetic field limit, we drop $\mathbf{A}$ and the corresponding wave equations and consider a quasi-electrostatic Vlasov-Wave model, which will agree with the electrostatic Vlasov-Poisson model when $|\mathbf{v}|/c << 1$. This condition frequently can be interpreted as $\omega_p L << c$ for a relevant physical length scale L, where ω_p is the (electron) plasma frequency. The resulting Vlasov-Wave system is as follows:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f - \frac{q}{m} \nabla \Phi \cdot \nabla_{\mathbf{v}} f = 0 \tag{1.9}$$

$$\frac{1}{c^2} \frac{\partial^2 \Phi}{\partial t^2} - \Delta \Phi = \rho / \varepsilon_0, \quad \rho(\mathbf{x}, t) = q \int_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \tag{1.10}$$

The purpose of this work is the numerical solution of (nondimensionalized forms of) these systems with PIC methods that avoid imposing a CFL stablity restriction related to the speed of light due to the presence of the wave equations for Φ and $\mathbf{A}$.

1.3 Nondimensionalization and Asymptotic Analysis

We provide the normalizations used in the test problems presented in this work in both electrostatic and electromagnetic cases. In the electrostatic case, we argue by formal asymp-

totic analysis and classical solution formulas that the Vlasov-Wave system agrees with the Vlasov-Poisson system in the relevant scaling limit.

1.3.1 Electrostatic Case

Here we give the normalization used for the electrostatic test problems in Section 3.6.1. Consider the following change of variables:

$$\tilde{f} = Ff, \quad \tilde{t} = Tt, \quad \tilde{\mathbf{x}} = L\mathbf{x}, \quad \tilde{\mathbf{v}} = V\mathbf{v}, \quad \tilde{\rho} = qN\rho, \quad \tilde{\Phi} = \Phi_0\Phi$$

applied to the system

$$\frac{\partial \tilde{f}}{\partial \tilde{t}} + \tilde{\mathbf{v}} \cdot \nabla_{\tilde{\mathbf{x}}} \tilde{f} - \frac{q}{m} \nabla \tilde{\Phi} \cdot \nabla_{\tilde{\mathbf{v}}} \tilde{f} = 0 \tag{1.11}$$

$$\frac{1}{c^2} \frac{\partial^2 \tilde{\Phi}}{\partial \tilde{t}^2} - \Delta \tilde{\Phi} = \tilde{\rho}/\varepsilon_0, \quad \tilde{\rho}(\tilde{\mathbf{x}}, \tilde{t}) = q \int_{\tilde{\mathbf{v}}} \tilde{f}(\tilde{\mathbf{x}}, \tilde{\mathbf{v}}, \tilde{t}) d\tilde{\mathbf{v}} \tag{1.12}$$

Assuming the scalings,

$$V = \frac{L}{T}, \quad T = \sqrt{\frac{m\varepsilon_0}{Nq^2}} = \omega_p^{-1}, \quad \Phi_0 = \frac{qNL^2}{\varepsilon_0}, \quad F = \frac{N^{\frac{2-d}{d}} (\varepsilon_0 m)^{d/2}}{q^d L^d}, \tag{1.13}$$

which are the natural scalings in the electrostatic limit, we obtain:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f - \nabla \Phi \cdot \nabla_{\mathbf{v}} f = 0 \tag{1.14}$$

$$\epsilon^2 \frac{\partial^2 \Phi}{\partial t^2} - \Delta \Phi = \rho, \quad \rho(\mathbf{x}, t) = \int_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t)\, d\mathbf{v} \tag{1.15}$$

where $\epsilon = \frac{L}{cT} = \frac{V}{c}$ (not to be confused with the electric permittivity ε_0). Note that $1/\epsilon$ is the speed of propagation of waves in the potential in this normalization, and becomes large when ϵ is small, that is, when the characteristic particle velocities are small compared to the speed of light.

Assume the following formal asymptotic expansions:

$$f = f_0 + \epsilon f_1 + \epsilon^2 f_2 + \cdots \tag{1.16}$$

$$\rho = \int_{\mathbf{v}} f d\mathbf{v} \tag{1.17}$$

$$= \int_{\mathbf{v}} f_0 d\mathbf{v} + \epsilon \int_{\mathbf{v}} f_1 d\mathbf{v} + \epsilon^2 \int_{\mathbf{v}} f_2 d\mathbf{v} + \cdots \tag{1.18}$$

$$= \rho_0 + \epsilon \rho_1 + \epsilon^2 \rho_2 + \cdots \tag{1.19}$$

$$\Phi = \Phi_0 + \epsilon \Phi_1 + \epsilon^2 \Phi_2 + \cdots \tag{1.20}$$

Collecting in orders of ϵ:

$$O(1): \qquad \frac{\partial f_0}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_0 - \nabla \Phi_0 \cdot \nabla_{\mathbf{v}} f_0 = 0 \tag{1.21}$$

$$-\Delta \Phi_0 = \rho_0, \quad \rho_0 = \int_{\mathbf{v}} f_0\, d\mathbf{v} \tag{1.22}$$

$$O(\epsilon): \qquad \frac{\partial f_1}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_1 - \nabla \Phi_0 \cdot \nabla_{\mathbf{v}} f_1 - \nabla \Phi_1 \cdot \nabla_{\mathbf{v}} f_0 = 0 \tag{1.23}$$

$$-\Delta \Phi_1 = \rho_1, \quad \rho_1 = \int_{\mathbf{v}} f_1\, d\mathbf{v} \tag{1.24}$$

$$O(\epsilon^k), \quad k \geq 2: \qquad \frac{\partial f_k}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_k - \sum_{j=0}^{k} \nabla \Phi_j \cdot \nabla_{\mathbf{v}} f_{k-j} = 0 \tag{1.25}$$

$$-\Delta \Phi_k = \rho_k - \frac{\partial^2 \Phi_{k-2}}{\partial t^2}, \quad \rho_k = \int_{\mathbf{v}} f_k\, d\mathbf{v} \tag{1.26}$$

We note that the leading order is precisely the Vlasov-Poisson system (nondimensionalized under the same scalings). This formal computation suggests that our model will agree with the electrostatic model to $O(\epsilon = V/c)$ when particle velocities are small compared to the speed of light. This can be considered as a consequence of the strong Huygens' principle in odd spatial dimensions, and of a weaker decay property that holds in even spatial dimensions, which can be deduced from classical solution formulas [21, 44]. Consider the Cauchy problem,

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \Delta u = f(x,t) \qquad (x,t) \in \mathbb{R}^d \times (0,\infty) \tag{1.27}$$

$$u(x,0) = 0 \qquad x \in \mathbb{R}^d \tag{1.28}$$

$$u_t(x,0) = 0 \qquad x \in \mathbb{R}^d \tag{1.29}$$

for $d = 2, 3$ and f sufficiently smooth with compact support, for which classical explicit solution formulas exist. We consider the case of f having compact support in $B(0,R) \times (0,T) \subset \mathbb{R}^d \times (0,\infty)$, where $B(0,R) = \{x \in \mathbb{R}^d | \sum_{j=1}^{d} x_j^2 < R^2\}$. Classical solution formulas imply that for $x \in B(0,R)$ and $t > T + 2R/cT$, we have

$$u(x,t) = O((ct)^{-1}) \qquad d = 2 \tag{1.30}$$

$$u(x,t) = 0 \qquad d = 3 \tag{1.31}$$

As a generalization of this, for any sufficiently smooth $f(\cdot,t)$ supported in $B(0,R)$ for all $t > 0$ with $f(\cdot,t) = f_T(\cdot)$ for all $t > T$, it is again easily argued that for $x \in B(0,R)$ and

$t > T + 2R/cT$, we have

$$u(x,t) = u_P^2(x) + O((ct)^{-1}) \qquad d = 2 \tag{1.32}$$

$$u(x,t) = u_P^3(x) \qquad d = 3 \tag{1.33}$$

where $u_P^d(x)$ is the classical integral solution of the Poisson equation $-\Delta u_P^d = f_T$ in dimension d.

The convergence of solutions to the Vlasov-Maxwell system to those of the Vlasov-Poisson system has been rigorously considered in works such as [2, 50]. It may be possible to apply similar techniques to rigorously study the convergence of solutions of our Vlasov-Wave model to those of the Vlasov-Poisson system, but this is outside of the scope of the present work.

1.3.2 Electromagnetic Case

Here we give the normalization used for the electromagnetic test problem in Section 3.6.2. Consider the following change of variables:

$$\tilde{f} = Ff, \quad \tilde{t} = Tt, \quad \tilde{\mathbf{x}} = L\mathbf{x}, \quad \tilde{\mathbf{v}} = V\mathbf{v}, \tag{1.34}$$

$$\tilde{\rho} = qN\rho, \quad \tilde{J}_z = qVNJ_z, \tag{1.35}$$

$$\tilde{\Phi} = \Phi_0\Phi, \quad \tilde{\mathbf{A}} = A_0 A_z \tag{1.36}$$

applied to the system

$$\frac{\partial \tilde{f}}{\partial \tilde{t}} + \tilde{\mathbf{v}} \cdot \nabla_{\tilde{\mathbf{x}}} \tilde{f} + \frac{q}{m}\left(-\nabla_{\tilde{\mathbf{x}}}\tilde{\Phi} - \tilde{\mathbf{v}} \times (\nabla_{\tilde{\mathbf{x}}} \times (0,0,\tilde{A}_z))\right) \cdot \nabla_{\tilde{\mathbf{v}}} \tilde{f} = 0 \tag{1.37}$$

$$\frac{1}{c^2}\frac{\partial^2\tilde{\Phi}}{\partial\tilde{t}^2} - \Delta\tilde{\Phi} = \tilde{\rho}/\varepsilon_0, \qquad \tilde{\rho}(\tilde{\mathbf{x}},\tilde{t}) = q\int_{\tilde{\mathbf{v}}} \tilde{f}(\tilde{\mathbf{x}},\tilde{\mathbf{v}},\tilde{t})d\tilde{\mathbf{v}} \tag{1.38}$$

$$\frac{1}{c^2}\frac{\partial^2\tilde{A}_z}{\partial\tilde{t}^2} - \Delta\tilde{A}_z = \mu_0\tilde{J}_z, \qquad \tilde{J}_z(\tilde{\mathbf{x}},\tilde{t}) = q\int_{\tilde{\mathbf{v}}} \tilde{v}_z\tilde{f}(\tilde{\mathbf{x}},\tilde{\mathbf{v}},\tilde{t})d\tilde{\mathbf{v}} \tag{1.39}$$

Assuming the scalings,

$$V = \frac{L}{T}, \quad T = \sqrt{\frac{m\varepsilon_0}{Nq^2}} = \omega_p^{-1}, \tag{1.40}$$

$$\Phi_0 = \frac{qNL^2\varepsilon_R}{\varepsilon_0}, \quad A_0 = \frac{\mu_0 VqNL^2}{\mu_R}, \quad \varepsilon_R\mu_R = V^2/c^2, \tag{1.41}$$

$$F = \frac{N^{\frac{2-d}{d}}(\varepsilon_0 m)^{d/2}}{q^d L^d} \tag{1.42}$$

$$\tag{1.43}$$

where ε_R and μ_R are dimensionless parameters introduced to enforce the Lorenz gauge condition, we obtain:

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} f + \varepsilon_R(-\nabla\Phi + \mathbf{v}\times(\nabla\times(0,0,A_z))\cdot\nabla_{\mathbf{v}} f = 0 \tag{1.44}$$

$$\epsilon^2\frac{\partial^2\Phi}{\partial t^2} - \Delta\Phi = \rho/\varepsilon_R, \quad \rho(\mathbf{x},t) = \int_{\mathbf{v}} f(\mathbf{x},\mathbf{v},t)\,d\mathbf{v} \tag{1.45}$$

$$\epsilon^2\frac{\partial^2 A_z}{\partial t^2} - \Delta A_z = \mu_R J_z, \quad J_z(\mathbf{x},t) = \int_{\mathbf{v}} v_z f(\mathbf{x},\mathbf{v},t)\,d\mathbf{v} \tag{1.46}$$

where $\epsilon = \frac{L}{cT} = \frac{V}{c}$, as in the electrostatic case. For the Bennett pinch problem in Section 3.6.2, we choose $\varepsilon_R = 1$.

Chapter 2

Description of the wave equation solver

2.1 Origin and Mitigation of the CFL Restriction

Consider the Cauchy problem for the wave equation:

$$u_{tt} = c^2 \Delta u \tag{2.1}$$

$$u(x, 0) = g(x) \tag{2.2}$$

$$u_t(x, 0) = h(x). \tag{2.3}$$

To see the origin of the CFL restriction in a typical explicit finite difference method, consider the semi-discrete scheme obtained by substituting the centered, second-order finite difference discretization of u_{tt} into the wave equation [19]. We obtain

$$\frac{u(x, t + \Delta t) - 2u(x, t) + u(x, t - \Delta t)}{\Delta t^2} = c^2 \Delta^2 u(x, t).$$

Upon Fourier transforming in the spatial variable, we obtain

$$\hat{u}(\xi, t + \Delta t) - 2\hat{u}(\xi, t) + \hat{u}(\xi, t - \Delta t) = -c^2 \Delta t^2 |\xi|^2 \hat{u}(\xi, t) \tag{2.4}$$

where ξ is the spatial frequency. The problem is that a high-frequency perturbation of u - such as that introduced by truncation error - will be amplified by the symbol $-c^2\Delta t^2|\xi|^2$, resulting in instability. In a fully discrete method, the spatial frequencies are bounded by $2/\Delta x$ (the Nyquist frequency), meaning that $c\Delta t/\Delta x$ must be sufficiently small to prevent amplification of high frequencies. In terms of semi-discrete Von Neumann analysis, consider substituting into 2.4 a semi-discrete solution of the form $\hat{u}(\xi, n\Delta t) = \lambda^n$ for some $\lambda \in \mathbb{C}$. We obtain a quadratic equation $\lambda^2 - (2 - z^2)\lambda + 1 = 0$ and the stability condition $|\lambda| \leq 1$ for any root λ of this equation, where $z = c\Delta t|\xi|$. It is easily verified from the quadratic formula that the roots λ_i satisfy $|\lambda_i| \leq 1$ for $i = 1, 2$ if and only if $|z| \leq 2$, that is if and only if $c\Delta t \leq 2/|\xi|$. For the 1D case, inserting the Nyquist frequency $2/\Delta x$ into the upper bound gives the usual CFL stability restriction $c\Delta t \leq \Delta x$.

To avoid the CFL restriction while maintaining an explicit update equation, we modify the symbol of the Laplacian to bound it and prevent the amplification of high frequencies. Consider the one-dimensional case. We look for bounded rational approximations of ξ^2 near $\xi = 0$. One option is

$$\xi^2 \approx \frac{\alpha^2 \xi^2}{\xi^2 + \alpha^2} \approx \xi^2(1 - (\xi/\alpha)^2 + \cdots)$$

We can write this modified semi-discrete scheme in Fourier space as

$$\hat{u}(\xi, t+\Delta t) - 2\hat{u}(\xi, t) + \hat{u}(\xi, t-\Delta t) = -c^2\Delta t^2 \frac{\alpha^2\xi^2}{\xi^2+\alpha^2}\hat{u}(\xi, t)$$

Choosing $\alpha = \beta/c\Delta t$ for some properly chosen β, we can see that the modified symbol is uniformly bounded by β^2 for all Δt, so that choosing $\beta \leq 2$ will ensure stability, independently of Δt, and it turns out that $\beta = 2$ gives optimal accuracy for stable schemes of this family [9, 11]. Multiplication in Fourier space is equivalent to convolution in physical space, so transforming back into physical space gives a method based on convolution with the inverse Fourier transform of the modified symbol:

$$u(x, t+\Delta t) - 2u(x, t) + u(x, t-\Delta t) = -\beta^2 D[u](x, t)$$

where $D[f](x) = \int (\delta(x-x') - \frac{\alpha}{2}e^{-\alpha|x-x'|})f(x')\,dx' = f(x) - \frac{\alpha}{2}\int e^{-\alpha|x-x'|}f(x')\,dx'$.

As a general comment, this modification of the symbol of the Laplacian introduces error at high spatial frequencies, so that this technique may not be suitable for problems in which extremely accurate propagation of waves at high spatial frequencies (high wavenumber) is essential. However, in many problems of plasma physics, the physics is dominated by effects at low spatial frequencies, and in these problems this technique can find useful application.

A detailed discussion of numerical methods for the wave equation derived from the perspective of Fourier multipliers, including higher-order methods and the extension to multiple dimensions, can be found in [11]. In the following section, we give the derivation of two useful second-order numerical methods through alternative means, the Method of Lines Transpose,

which also facilitates the extension of the methods to bounded domains and to multiple dimensions.

2.2 Derivation of Second-Order Semi-Discrete Schemes With the Method of Lines Transpose

In the Method of Lines Transpose for the solution of time-dependent PDEs, also known as Rothe's method [49], finite difference discretizations of time derivatives are substituted into the PDE, resulting in a boundary value problem (BVP) to be solved at each time step. In recent years, the MOLT has been applied as a numerical method to solve various time-dependent PDEs, with initial focus on parabolic equations [12, 29, 30, 8, 36]. The work [10] extended this approach to the second order wave equation, and is the basis of the present work. Further work extended this approach to higher dimensions through dimensional splitting [9] and to higher-order through the Fourier multiplier approach mentioned in the previous section [11]. Based on the work in [10, 9, 11], we give here an overview of the derivation of two useful second-order schemes for the wave equation, their extension to multiple dimensions through dimensional splitting, and a fast numerical algorithm for the 1D problem. In Section 2.3, we give semi-discrete Von Neumann analyses and dispersion relations for these semi-discrete schemes, and in Section 2.5 we give a new proof that the fully discrete schemes are unconditionally stable, in the sense of Von Neumann analysis.

2.2.1 Dispersive Semi-Discrete Scheme

As in the previous section, we substitute the following second-order centered discretization into the wave equation $\frac{1}{c^2}u_{tt} - \Delta u = f(x,t)$:

$$u_{tt}^n = \frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} - \frac{\Delta t^2}{12} u_{tttt}(x,\eta).$$

Instead of proceding by the Fourier multiplier approach as mentioned previously, we apply an averaging technique with similar results. We average the Laplacian term in time

$$\Delta u^n = \frac{1}{4}\Delta\left(u^{n+1} + 2u^n + u^{n-1}\right) + O(\Delta t^2)$$

and substitute into the wave equation. Defining $\alpha = 2/(c\Delta t)$ and manipulating gives the semi-discrete scheme

$$\left(-\frac{1}{\alpha^2}\Delta + 1\right)\left(u^{n+1} + 2u^n + u^{n-1}\right) = 4u^n + \frac{4}{\alpha^2} f(x,t^n) + O(\Delta t^4).$$

We call this the dispersive semi-discrete scheme, since all terms in the semi-discrete dispersion relation are real-valued (see 2.3).

2.2.2 Diffusive Semi-Discrete Scheme

We substitute the following backward difference formula (BDF) discretization:

$$u_{tt}^{n+1} = \frac{2u^{n+1} - 5u^n + 4u^{n-1} - u^{n-2}}{\Delta t^2} - \frac{11\Delta t^2}{12} u_{tttt}(x,\eta)$$

into the wave equation $\frac{1}{c^2}u_{tt} - \Delta u = f(x,t)$.

Rearranging, defining $\alpha = \sqrt{2}/(c\Delta t)$ and dividing by α^2 gives the semi-discrete scheme

$$\left(-\frac{1}{\alpha^2}\Delta + 1\right) u^{n+1} = \frac{1}{2}\left(5u^n - 4u^{n-1} + u^{n-2}\right) + \frac{1}{\alpha^2} f(x, t^{n+1}) + O(\Delta t^4).$$

We call this the diffusive semi-discrete scheme, due to the presence of imaginary-valued terms in the semi-discrete dispersion relation (see 2.3). The (slight) dissipation of this method proved useful in initial PIC simulations aimed at capturing the steady-state potential due to a single particle, and so we use this scheme (rather than the dispersive scheme) in all PIC simulations in this work. On the other hand, the dispersive scheme (and the higher-order versions in [11]) have smaller truncation error than the diffusive scheme, and do not have an implicit source term (at time level $n+1$) that confound the use of higher-order time stepping schemes for particles. Future work will investigate the use of these time-centered schemes, making use of artificial dissipation (as in Section 2.6) when appropriate, and corresponding time-stepping schemes for particles.

2.3 Semi-Discrete Von Neumann Analysis of the Second Order Schemes

In this section, we provide semi-discrete Von Neumann stability analyses and dispersion relations for the semi-discrete schemes derived in 2. These build on similar analyses for related, but different, schemes given in [10].

2.3.1 Dispersive Scheme

Substitute ansatz $u^n = e^{i(k\cdot x - \tilde{\omega} n \Delta t)}$ into

$$\left(-\frac{1}{\alpha^2}\Delta + 1\right)\left(u^{n+1} + 2u^n + u^{n-1}\right) = 4u^n$$

Obtain a polynomial

$$\lambda^2 + 2\left(\frac{|k|^2 - \alpha^2}{|k|^2 + \alpha^2}\right)\lambda + 1 = 0$$

where $\lambda = e^{i\tilde{\omega}\Delta t}$.

We can solve to obtain $\lambda_{1/2} = \frac{\alpha \pm i|k|}{\alpha \mp i|k|}$, which gives $|\lambda_{1/2}| = 1$, meaning this scheme is non-dissipative.

Noting that $\cos(\tilde{\omega}\Delta t) = \frac{1}{2}(\lambda_1 + \lambda_2)$, and defining $z = |k|/\alpha = |k|c\Delta t/2$, we obtain

$$\begin{aligned}
\tilde{\omega} &= \frac{2}{\Delta t}\arccos\left(\sqrt{\frac{\alpha^2}{|k|^2 + \alpha^2}}\right) = \frac{2}{\Delta t}\arccos\left(\sqrt{\frac{1}{1+z^2}}\right) \\
&\approx \frac{2}{\Delta t}\left(z - z^3/3 + O(z^5)\right) \\
&\approx |k|c\left(1 - \frac{1}{12}\left(|k|c\Delta t\right)^2 + O((|k|c\Delta t)^4)\right)
\end{aligned}$$

So the phase error is $|\frac{\tilde{\omega}}{|k|c} - 1| = O((|k|c\Delta t)^2)$. Moreover, $\tilde{\omega}$ is real, owing to the non-dissipative nature of the scheme.

2.3.2 Diffusive Scheme

Substituting the ansatz $u^n = e^{i(k\cdot x - \tilde{\omega} n \Delta t)}$ into

$$\left(-\frac{1}{\alpha^2}\Delta + 1\right)u^{n+1} = \frac{1}{2}\left(5u^n - 4u^{n-1} + u^{n-2}\right),$$

we obtain a polynomial

$$\lambda^3 - 4\lambda^2 + 5\lambda - 2(1+z^2) = 0$$

where $\lambda = e^{i\tilde{\omega}\Delta t}$, $z = |k|/\alpha$. The three roots tell us about possible modes $u^n = e^{ik\cdot x}\lambda^{-n}$.

A necessary condition for stability is $\lambda \geq 1$ for all roots.

The first root,

$$\lambda_1 = \frac{4}{3} + \frac{1}{3}\left(27z^2 + 3\sqrt{3}\sqrt{27z^4+2z^2}+1\right)^{1/3} + \frac{1}{3\left(27z^2 + 3\sqrt{3}\sqrt{27z^4+2z^2}+1\right)^{1/3}},$$

corresponds to a spurious nonpropagating mode of the form $u^n = e^{ik\cdot x}\lambda_1^{-n}$. Since $\lambda_1 \geq 2$ for all $z = |k|c\Delta t/\sqrt{2} \geq 0$, the mode rapidly decays and poses no threat to stability.

The other two roots are a pair of complex conjugates:

$$\begin{aligned}\lambda_{2/3} &= -\frac{1}{6}(1 \mp i\sqrt{3})\left(27z^2 + 3\sqrt{3}\sqrt{27z^4+2z^2}+1\right)^{1/3} - \\ &\quad - \frac{1 \pm i\sqrt{3}}{6(27z^2 + 3\sqrt{3}\sqrt{27z^4+2z^2}+1)^{1/3}} + 4/3 \\ &\approx 1 \pm i\sqrt{2}z - z^2 \mp i\frac{5\sqrt{2}}{4}z^3 + 4z^4 \pm i\frac{231\sqrt{2}}{32}z^5 + O(z^6) \text{ as } z \to 0.\end{aligned}$$

We can show that

$$|\lambda_{2/3}|^2 = \frac{16}{9} + \frac{4W^4 - 16W^3 - 4W^2 - 16W + 4}{36W^2} \geq 1 \quad \text{for all } W \geq 1$$

where $W = \left(27z^2 + 3\sqrt{3}\sqrt{27z^4 + 2z^2} + 1\right)^{1/3} \geq 1$ for all $z \geq 0$.

Since $\lambda = e^{i\tilde{\omega}\Delta t}$, we have

$$\begin{aligned}
\tilde{\omega} &= \frac{1}{i\Delta t}\log(\lambda) \\
&\approx \frac{1}{i\Delta t}\log(1 + i\sqrt{2}z - z^2 - i\frac{5\sqrt{2}}{4}z^3 + 4z^4 + i\frac{231\sqrt{2}}{32}z^5 + O(z^6)) \\
&\approx |k|c\left(1 - \frac{11(|k|c\Delta t)^2}{24} - i\frac{(|k|c\Delta t)^3}{2} - \frac{15(|k|c\Delta t)^4}{16} + O((|k|c\Delta t)^5)\right)
\end{aligned}$$

So the phase error is $|\frac{\tilde{\omega}}{|k|c} - 1| = O((|k|c\Delta t)^2)$.

The presence of the imaginary third term in the expansion shows that $u^n \sim e^{-in(-i(|k|c\Delta t)^3)\Delta t)} = e^{-n(|k|c)^3\Delta t^4/2}$, causing the mode to decay. This is why we term this scheme diffusive (or dissipative).

2.4 Solution of the Modified Helmholtz Equation

As seen in the previous section, both the dispersive and diffusive semi-discrete schemes include an elliptic BVP to be solved at each time step. The resulting PDE is sometimes called the modified Helmholtz equation [30]. In contrast to the usual frequency-domain Helmholtz equation $\Delta u + \frac{\omega^2}{c^2}u = f$, the modified Helmholz equation has a nonoscillatory Green's function. The oscillation in the solution of the wave equation is supported by the presence of multiple time levels in the semi-discrete equations. Our solution strategy is to use the well-known technique of dimensional splitting [43] to reduce problems in multiple dimensions to problems in one dimension, to which we apply a fast integral solution method. The dimensionally-split integral solution naturally leads to unconditionally stable numerical

schemes with computational cost and coding complexity comparable to explicit schemes.

2.4.1 Dimensional Splitting

For smooth $u(x, y)$, we have

$$\left(-\frac{1}{\alpha^2}\left(\partial_{xx}+\partial_{yy}\right)+1\right)u=\left(-\frac{1}{\alpha^2}\partial_{xx}+1\right)\left(-\frac{1}{\alpha^2}\partial_{yy}+1\right)u-\frac{1}{\alpha^4}\partial_{xxyy}u$$

So, to approximately solve $\left(-\frac{1}{\alpha^2}\left(\partial_{xx}+\partial_{yy}\right)+1\right)u=f$ we will instead solve

$$\left(-\frac{1}{\alpha^2}\partial_{xx}+1\right)\left(-\frac{1}{\alpha^2}\partial_{yy}+1\right)u=f.$$

where we have introduced the splitting error

$$\frac{1}{\alpha^4}\partial_{xxyy}u=O(c^4\Delta t^4).$$

To solve

$$\left(-\frac{1}{\alpha^2}\partial_{xx}+1\right)\left(-\frac{1}{\alpha^2}\partial_{yy}+1\right)u=f$$

we define $w=\left(-\frac{1}{\alpha^2}\partial_{yy}+1\right)u$ and solve the following one-dimensional BVPs "line by line":

$$\left(-\frac{1}{\alpha^2}\partial_{xx}+1\right)w(x,\cdot)=f(x,\cdot)$$
$$\left(-\frac{1}{\alpha^2}\partial_{yy}+1\right)u(\cdot,y)=w(\cdot,y)$$

where appropriate boundary conditions are supplied (see [10, 11]). Since $w=u+O(c^2\Delta t^2)$, for second order accuracy it suffices to use the same boundary conditions for w as u. Future

work will investigate higher-order accurate boundary conditions. To facilitate the "line-by-line" solution, we discretize the domain with a uniform Cartesian grid.

2.4.2 Integral Solution Method in 1D

Consider the modified Helmholtz equation in one dimension, $\left(-\frac{1}{\alpha^2}\frac{d^2}{dx^2}+1\right)u = f(x)$ for $x \in \Omega = (a,b)$, with appropriate boundary conditions imposed at $x = a$ and $x = b$. We can write $u(x) = u^P(x) + u^H(x)$, where

$$u^P(x) = \frac{\alpha}{2}\int_a^b f(x')e^{-\alpha|x-x'|}\,dx'$$
$$u^H(x) = Ae^{-\alpha(x-a)} + Be^{-\alpha(b-x)}$$

are the particular and homogeneous solutions and where A and B depend on the boundary conditions imposed, as well as the values of $u^P(a)$ and $u^P(b)$ [10]. Our fast integral solver consists of a fast convolution algorithm for the evaluation of the particular solution u^P, along with appropriate algorithms for evaluating the homogeneous solution u^H, which can be viewed as boundary correction terms. For many common boundary conditions, the coefficients A and B for the boundary correction terms can be found by applying the given boundary conditions and solving a 2×2 system by hand for A and B. For instance, in the case of homogeneous Dirichlet boundary conditions $u(a) = u(b) = 0$ and defining $\gamma = e^{-\alpha(b-a)}$, we find

$$A = \frac{\gamma u^P(b) - u^P(a)}{1-\gamma^2}$$
$$B = \frac{\gamma u^P(a) - u^P(b)}{1-\gamma^2}.$$

In the case of periodic boundary conditions, we find

$$A = \frac{u^P(b)}{1-\gamma}$$
$$B = \frac{u^P(a)}{1-\gamma}.$$

With some further consideration, other boundary conditions can be derived, including outflow (absorbing) boundary conditions (for the underlying wave equation, based on one-way wave equations). For further details, see [10, 9, 11].

2.4.3 Fast Numerical Evaluation of the 1D Convolution Operator

The convolution operator giving the particular solution can be decomposed as

$$\begin{aligned} I[f](x) &= \frac{\alpha}{2}\int_{-\infty}^{\infty} f(x')e^{-\alpha|x-x'|}\,dx' \\ &= \frac{\alpha}{2}\int_{-\infty}^{x} f(x')e^{-\alpha|x-x'|}\,dx' + \frac{\alpha}{2}\int_{x}^{\infty} f(x')e^{-\alpha|x-x'|}\,dx' \\ &=: I^L[f](x) + I^R[f](x) \end{aligned}$$

for a given function f. Meanwhile, we have the recursion relations

$$I^L[f](x) = e^{-\alpha\Delta x} I^L[f](x-\Delta x) + \frac{\alpha}{2}\int_{x-\Delta x}^{x} f(x')e^{-\alpha(x-x')}\,dx'$$
$$I^R[f](x) = e^{-\alpha\Delta x} I^R[f](x+\Delta x) + \frac{\alpha}{2}\int_{x}^{x+\Delta x} f(x')e^{-\alpha(x'-x)}\,dx'.$$

Based on these observations, we outline the fast algorithm for the numerical evaluation of this convolution operator on a uniform Cartesian grid developed in [10, 9, 11]. Consider the convolution operator applied to a function f supported on the interval (a, b), with the

convolution also to be evaluated in (a, b). The interval is discretized into N equal subintervals of length $\Delta x = (b-a)/N$, with endpoints $x_1 = a$, $x_{j+1} = x_j + \Delta x$ for $j = 1, ..., N$. We denote $I_j = I[f](x_j)$, $I_j^L = I^L[f](x_j)$ and $I_j^R = I^R[f](x_j)$, as defined above. Further, we define the local integrals

$$J_j^L = \frac{\alpha}{2} \int_{x_{j-1}}^{x_j} f(x') e^{-\alpha(x_j - x')} \, dx' \quad j = 2, ..., N+1$$
$$J_j^R = \frac{\alpha}{2} \int_{x_j}^{x_{j+1}} f(x') e^{-\alpha(x' - x_j)} \, dx' \quad j = 1, ..., N.$$

Suppose we have already computed these J_j^L and J_j^R. Setting $I_1^L = I_{N+1}^R = 0$, we can then perform a recursive evaluation of the convolution integral by computing $I_{j+1}^L = e^{-\alpha \Delta x} I_j^L + J_{j+1}^L$ and $I_{N+1-j}^R = e^{-\alpha \Delta x} I_{N+1-j+1}^R + J_{N+1-j}^R$ for $j = 1, ...N$. We then sum $I_j = I_j^L + I_j^R$ for each $j = 1, ..N+1$.

The method for evaluating the local integrals J_j^L and J_j^R depends on the nature of the integrand. For the particle convolution integral, where the integrand is the sum of particle shape functions, we can analytically evaluate the local integrals. This is described in Section 3.4. For general integrands, and specifically for the terms involving u on the right hand sides of the semi-discrete schemes, we numerically evaluate the local integrals with a quadrature rule. It is important to note that any given quadrature rule may or may not deliver an accurate and stable overall scheme for the wave equation. To achieve accuracy and stability, we use a quadrature rule found by analytically integrating against a Lagrange polynomial interpolant. For a quadratic interpolant, leading to second-order accurate quadrature in

space, we obtain the following approximations

$$J_j^L \approx Pf(x_j) + Qf(x_{j-1}) + R(f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))$$
$$J_j^R \approx Pf(x_j) + Qf(x_{j+1}) + R(f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))$$

where, defining $\nu = \alpha\Delta x$ and $d = e^{-\nu}$,

$$P = 1 - \frac{1-d}{\nu}$$
$$Q = -d + \frac{1-d}{\nu}$$
$$R = \frac{1-d}{\nu^2} - \frac{1+d}{2\nu}$$

Higher-order spatial accuracy can be obtained by using higher-order accuracy quadrature rules, with the outcome of unconditional stability limiting the choice of quadrature rules. Further details can be found in [10, 9, 11]. In Section 2.5, Von Neumann analyses are carried out for the fully discrete diffuse and dispersive schemes in one and multiple dimensions, and it is shown that they are both unconditionally stable.

2.5 Fully Discrete Von Neumann Analysis

In this section, we provide fully discrete Von Neuman analyses for the two fully discrete schemes derived in 2, and show that they are unconditionally stable, in the sense of Von Neumann analysis. Combining the quadrature rules and exponential recursion, and ignoring

boundaries, we can write

$$
\begin{aligned}
I[f](x_j) = \frac{\alpha}{2}\int_{-\infty}^{\infty} f(x')e^{-\alpha|x_j-x'|}\,dx' \approx \\
\approx I_h[f_j] = Pf_j + \frac{1}{2}Q\left(f_{j+1}+f_{j-1}\right) + R\left(f_{j+1}-2f_j+f_{j-1}\right) + \\
+ \frac{1}{2}\sum_{k=1}^{\infty} e^{-\nu k}\left[P\left(f_{j+k}+f_{j-k}\right) + Q\left(f_{j+k+1}+f_{j-k-1}\right) + \right. \\
\left. + R\left(f_{j+k+1}-2f_{j+k}+f_{j+k-1}+f_{j-k+1}-2f_{j-k}+f_{j-k-1}\right)\right]
\end{aligned}
$$

with $\nu = \alpha h = \alpha\Delta x$ and P, Q, and R defined as in Section 2.

Using the discrete convolution operator I_h, and ignoring sources and boundaries, we can write the diffusive version of the fully discrete scheme as

$$u_j^{n+1} = \frac{1}{2}I_h[5u_j^n - 4u_j^{n-1} + u_j^{n-2}]$$

and the dispersive scheme as

$$u_j^{n+1} + 2u_j^n + u_j^{n-1} = 4I_h[u_j^n]$$

Defining $I_{h,x}$ and $I_{h,y}$ as the discrete convolution operators acting in the x- and y-directions, and again ignoring sources and boundaries, we can write the diffusive version of the 2D fully discrete scheme as

$$u_{jk}^{n+1} = \frac{1}{2}I_{h,x}[I_{h,y}[5u_{jk}^n - 4u_{jk}^{n-1} + u_{jk}^{n-2}]]$$

and the 2D fully dispersive scheme as

$$u_{jk}^{n+1} + 2u_{jk}^{n} + u_{jk}^{n-1} = 4I_{h,x}[I_{h,y}[u_{jk}^{n}]]$$

We will refer to these as free-space schemes. We now state a stability theorem based on the Von Neumann analysis of the schemes.

Theorem. *The fully discrete dispersive and diffusive free-space schemes in one dimension are unconditionally stable. In higher dimensions, the corresponding dimensionally-split schemes are also unconditionally stable.*

To prove the stability theorem, we consider some properties of the discrete covolution operator.

Lemma. *Define the amplification factor* $A(\tilde{k}, \nu) = e^{-i\tilde{k}j\Delta x} I_h[e^{i\tilde{k}j\Delta x}]$. *Then* $A(\tilde{k}, \nu)$ *is well-defined (independent of* j*), and the following hold.*

- *If* $\nu > 0$ *and* $0 < |\tilde{k}\Delta x| \leq \pi$, *then* $0 < A(\tilde{k}, \nu) < 1$.
- *If* $\nu > 0$, *then* $A(0, \nu) = 1$.
- *For any* $0 < |\tilde{k}\Delta x| \leq \pi$, $\lim_{\nu \to 0^+} A(\tilde{k}, \nu) = 0$.
- *For any* $0 < |\tilde{k}\Delta x| \leq \pi$, $\lim_{\nu \to \infty} A(\tilde{k}, \nu) = 1$.

Proof of the Lemma. We calculate:

$$
\begin{aligned}
A(\tilde{k},\nu) =& e^{-i\tilde{k}j\Delta x} I_h[e^{i\tilde{k}j\Delta x}] \\
=& P\left(1+\sum_{k=1}^{\infty} e^{-k\nu}\cos(k\tilde{k}\Delta x)\right) + \\
& + Q\left(e^{\nu}\sum_{k=1}^{\infty} e^{-k\nu}\cos(k\tilde{k}\Delta x)\right) + \\
& + 2R(\cos(\tilde{k}\Delta x)-1)\left(1+\sum_{k=1}^{\infty} e^{-k\nu}\cos(k\tilde{k}\Delta x)\right) \\
=& 1+T,
\end{aligned}
$$

$$
T = \frac{\frac{d^2+d}{\nu}(\cos(\tilde{k}\Delta x)-1)^2 - 2\frac{1-d}{\nu^2}(d\cos(\tilde{k}\Delta x)-1)(\cos(\tilde{k}\Delta x)-1)}{d^2 - 2\cos(\tilde{k}\Delta x)d + 1}
$$

with $d = e^{-\nu}$.

If $\tilde{k}\Delta x = 0$ and $\nu > 0$, then $T = 0$. If $0 < |\tilde{k}\Delta x| \leq \pi$, some calculus shows $-1 < T < 0$ for any $\nu > 0$, and $\lim_{\nu\to 0+} T = -1$, and $\lim_{\nu\to\infty} T = 0$. □

Proof of the Theorem. We first prove that each one-dimensional free-space fully discrete scheme is unconditionally stable, then describe the extension of the result to multiple dimensions for the dimensionally-split schemes.

Dispersive Scheme

In this section, we prove that the one-dimensional free-space fully discrete dispersive scheme is unconditionally stable and non-dissipative. Consider applying the fully discrete dispersive scheme to $u_j^n = e^{i(\tilde{k}j\Delta x - \tilde{\omega}n\Delta t)}$. We obtain a polynomial

$$
\lambda^2 + (2-4z)\lambda + 1 = 0
$$

where $\lambda = e^{i\tilde{\omega}\Delta t}$ and $z = A(\tilde{k}, \nu)$. Note that the lemma implies $0 \leq z \leq 1$ for all $\tilde{k}$ and $\Delta x, \Delta t > 0$. We can solve to obtain the roots $\lambda_{1/2} = (2z-1) \pm i\sqrt{(1-z)z}$. We can show $|\lambda_{1/2}| = 1$ for all $0 \leq z \leq 1$, so the fully discrete dispersive scheme is unconditionally stable, and non-dissipative. This proves the theorem in the case of the one dimensional dispersive scheme.

Diffusive Scheme

Consider applying the fully discrete diffusive scheme to $u_j^n = e^{i(\tilde{k}j\Delta x - \tilde{\omega}n\Delta t)}$. We obtain a polynomial

$$\lambda^3 - 4\lambda^2 + 5\lambda - z = 0$$

where $\lambda = e^{i\tilde{\omega}\Delta t}$ and $z = 2/A(\tilde{k}, \nu)$. Note that the lemma implies $z \geq 2$ for all $\tilde{k}$ and $\Delta x, \Delta t > 0$. The condition for stability is $|\lambda| \geq 1$ for all roots of the polynomial, which we verify below.

The first root corresponds to a spurious mode:

$$\lambda_1 = \frac{4}{3} + \frac{\sqrt[3]{3\sqrt{3}\sqrt{27z^2 - 104z + 100} + 27z - 52}}{3\sqrt[3]{2}} + \\ + \frac{\sqrt[3]{2}}{3\sqrt[3]{3\sqrt{3}\sqrt{27z^2 - 104z + 100} + 27z - 52}}$$

We can show that

$$\lambda_1 = \frac{W^2 + 4W + 1}{3W} \geq 2 \quad \text{for all } W \geq 1$$

where $W = \frac{\sqrt[3]{3\sqrt{3}\sqrt{27z^2 - 104z + 100} + 27z - 52}}{\sqrt[3]{2}} \geq 1$ for $z \geq 2$.

The other roots are pair of complex conjugates:

$$\lambda_{2/3} = \frac{4}{3} - (1 \mp i\sqrt{3}) \frac{\sqrt[3]{3\sqrt{3}\sqrt{27z^2 - 104z + 100} + 27z - 52}}{6\sqrt[3]{2}} - \\ - (1 \pm i\sqrt{3}) \frac{\sqrt[3]{2}}{6\sqrt[3]{3\sqrt{3}\sqrt{27z^2 - 104z + 100} + 27z - 52}}$$

We can show that

$$|\lambda_{2/3}|^2 = \frac{4W^4 - 16W^3 + 60W^2 - 16W + 4}{36W^2} \geq 1 \quad \text{for all } W \geq 1$$

where $W = \frac{\sqrt[3]{3\sqrt{3}\sqrt{27z^2-104z+100}+27z-52}}{\sqrt[3]{2}} \geq 1$ for $z \geq 2$.

This proves the theorem in the case of the one dimensional diffusive scheme.

Extension to Higher Dimensions

When applying the dimensionally-split two-dimensional schemes to $u^n_{jk} = e^{i(\tilde{k}_x j\Delta x + \tilde{k}_y k\Delta y - \tilde{\omega} n\Delta t)}$, we obtain the same Von Neumann polynomials as in the 1D case, and basically the same stability analysis can be repeated. This can be easily extended to dimensionally-split schemes in any dimension.

□

This Von Neumann analysis does not take into consideration the effects of boundary conditions, and in principle, certain numerical boundary conditions could result in instability. In the test problems presented in [10, 9, 11] as well as the present work, the stability of the method seems robust under a variety of numerical boundary conditions. A stability analysis for some 1D fully discrete schemes (slightly different from those presented here)

with numerical Dirichlet and Neumann boundary conditions was carried out in [10], showing unconditional stability in those schemes. A similar stability analysis could also be carried out for the schemes considered in this work to study the stability of these schemes under the inclusion of numerical boundary conditions.

2.6 Artificial Dissipation in Centered Schemes

As discussed in 2.8, it is necessary to include some artificial dissipation in the numerical scheme to maintain stability with embedded boundary methods for Neumann boundary conditions, necessary for the implemenation of PEC boundary conditions with complex geometries. The diffusive scheme is dissipative, but has an implicit source term (at time level $n+1$), presenting a difficulty in coupling with particles in the fully electromagnetic case. This seems to restrict the method based on the diffusive solver to use first-order accurate, explicit forward Euler time stepping or second-order accurate, implicit Crank-Nicholson time stepping for particles, the second option requiring iteration of the field and particle updates. Hence, we seek to find schemes that maintain explicit source terms but also include some dissipation. The second-order dispersive scheme given above, and the higher-order accurate schemes given in [11], have explicit source terms (at time level n), but are non-dissipative. In this section, we give a method for adding tunable artificial dissipation terms into these second- and fourth-order accurate centered schemes.

2.6.1 Artificial Dissipation in 1D

We give modified forms of the centered versions of the implicit wave solver with tunable artificial dissipation that retain the property of unconditional stability. We restrict our

attention to the 1D schemes here, but the same proofs should apply in the higher-dimensional case. We let ϵ denote a small artificial dissipation parameter, and $\mathcal{D}[u] = u - \mathcal{L}^{-1}[u] = u(x) - \frac{\alpha}{2}\int_{-\infty}^{\infty} e^{-\alpha|x-x'|}u(x')\,dx'$ be defined as usual.

Modifiying the schemes given in [11], we have the second order scheme with dissipation,

$$u^{n+1} - 2u^n + u^{n-1} = -\beta^2\mathcal{D}[u^n] + \epsilon\mathcal{D}^2[u^{n-1}], \tag{2.5}$$

and the fourth order scheme with dissipation,

$$u^{n+1} - 2u^n + u^{n-1} = -\beta^2\mathcal{D}[u^n] - (\beta^2 + \frac{\beta^4}{12})\mathcal{D}^2[u^n] + \epsilon\mathcal{D}^3[u^{n-1}]. \tag{2.6}$$

We now prove the unconditional stability of these schemes for prescribed values of β. As in [11], we pass to the high-frequency limit.

Second Order Scheme

We obtain the Von Neumann polynomial $\rho^2 - (2-\beta^2)\rho + (1-\epsilon)$. We can check that the roots of this polynomial will be complex if $0 < \beta \leq \sqrt{2+2\sqrt{1-\epsilon}}$, and that in this case the roots satisfy

$$|\rho|^2 = \frac{1}{4}\left((2-\beta^2)^2 + 4(1-\epsilon) - (2-\beta^2)\right) \tag{2.7}$$

$$= 1 - \epsilon < 1 \tag{2.8}$$

which shows both the stability and dissipative nature of the second order scheme.

Fourth Order Scheme

We obtain the Von Neumann polynomial $\rho^2 - (2 - 2\beta^2 + \beta^4/12)\rho + (1-\epsilon)$. We can check that the roots of this polynomial will be complex if $0 < \beta \leq \sqrt{12\left(2 - \sqrt{1 - \frac{1}{6}\left(1 + \sqrt{1+\epsilon}\right)}\right)}$, and that in this case the roots satisfy

$$|\rho|^2 = \frac{1}{4}\left((2 - 2\beta^2 + \beta^4/12)^2 + 4(1-\epsilon) - (2 - 2\beta^2 + \beta^4/12)\right) \tag{2.9}$$

$$= 1 - \epsilon < 1 \tag{2.10}$$

which shows both the stability and dissipative nature of the fourth order scheme. □

We note that in each case, the maximum allowed value of β is slightly smaller than what is allowed by the corresponding scheme without dissipation. A more detailed analysis shows that the effective damping rate is $\left(\frac{k^2}{k^2+\alpha^2}\right)^2 \epsilon$ for the second order scheme, and $\left(\frac{k^2}{k^2+\alpha^2}\right)^3 \epsilon$ for the fourth order scheme, meaning that high frequencies are more rapidly damped than low frequencies.

2.6.2 Artificial Dissipation in 2D

Again modifiying the schemes given in [11], we have the second order scheme with dissipation,

$$u^{n+1} - 2u^n + u^{n-1} = -\beta^2 \mathcal{C}[u^n] + \epsilon \mathcal{C}^2[u^{n-1}], \tag{2.11}$$

and the fourth order scheme with dissipation,

$$u^{n+1} - 2u^n + u^{n-1} = -\beta^2 \mathcal{C}[u^n] - \beta^2 \mathcal{D}\mathcal{C}[u^n] - \frac{\beta^4}{12}\mathcal{C}^2[u^n] + \epsilon \mathcal{C}^3[u^{n-1}]. \tag{2.12}$$

where now $\mathcal{D} = 1 - \mathcal{L}_x^{-1}\mathcal{L}_y^{-1}$ and $\mathcal{C} = \mathcal{L}_y^{-1}\mathcal{D}_x + \mathcal{L}_x^{-1}\mathcal{D}_y$. For further details on these operators, see [11]. Numerical experiments indicate that the 2D schemes with artificial dissipation are indeed unconditionally stable with the same maximum value of β as with the 1D schemes.

2.7 Derivation of a Fourth-Order Dissipative Newmark Scheme

As an alternative to the artificial dissipation terms for centered schemes given in Section 2.6, we now give the derivation of a fourth-order accurate, dissipative, and (apparently) unconditionally stable method that we call a Newmark scheme, due to the introduction of an auxiliary variable $v = u_t$.

Consider a smooth solution $u(x,t)$ to (2.1). We can form the Taylor expansion

$$u(x, t+\Delta t) = \sum_{k=0}^{p-1} \frac{\partial^k u}{\partial t^k}(x,t)\frac{\Delta t^k}{k!} + \frac{\partial^p u}{\partial t^p}(x, \eta(x,t))\frac{\Delta t^p}{p!}$$

Then to form a p-th order accurate approximation to $u(x, t+\Delta t)$, we must form $(p-k)$-th order approximations to $\frac{\partial^k u}{\partial t^k}(x,t)$, for $k = 0, ..., p-1$.

Alternatively, we could keep the odd time derivatives in (2.7), and consider the Taylor

expansion of $v = u_t$, using the fact that v solve the wave equation $v_{tt} - c^2\Delta v = f_t(x,t)$:

$$\begin{aligned}
v(x,t+\Delta t) &= \sum_{k=0}^{\infty} \frac{\partial^k v}{\partial t^k}(x,t)\frac{\Delta t^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{\partial^{2k} v}{\partial t^{2k}}(x,t)\frac{\Delta t^{2k}}{(2k)!} + \\
&+ \sum_{k=1}^{\infty} \frac{\partial^{2k} u}{\partial t^{2k}}(x,t)\frac{\Delta t^{2k-1}}{(2k-1)!} \\
&= v(x,t) + \sum_{k=1}^{\infty} \left[c^{2k}\Delta^k v(x,t) + \sum_{j=1}^{k} c^{2(k-j)}\Delta^{k-j}\frac{\partial^{2j-1}}{\partial t^{2j-1}} f(x,t) \right] \frac{\Delta t^{2k}}{(2k)!} + \\
&+ \sum_{k=1}^{\infty} \left[c^{2k}\Delta^k u(x,t) + \sum_{j=1}^{k} c^{2(k-j)}\Delta^{k-j}\frac{\partial^{2(j-1)}}{\partial t^{2(j-1)}} f(x,t) \right] \frac{\Delta t^{2k-1}}{(2k-1)!}.
\end{aligned}$$

Writing out the first few terms of the Taylor expansions explicitly:

$$\begin{aligned}
u(x,t+\Delta t) &= u(x,t) + \Delta t v(x,t) + \frac{\Delta t^2}{2}\left(c^2\Delta u + f\right)(x,t) + \\
&+ \frac{\Delta t^3}{6}(c^2\Delta v + f_t)(x,t) + O(\Delta t^4) \\
v(x,t+\Delta t) &= v(x,t) + \Delta t\left(c^2\Delta u + f\right)(x,t) + \frac{\Delta t^2}{2}(c^2\Delta v + f_t)(x,t) + \\
&+ \frac{\Delta t^3}{6}\left(c^4\Delta^2 u + c^2\Delta f + c^2 f_{tt}\right)(x,t) + O(\Delta t^4)
\end{aligned}$$

In our target application (particle simulation of plasmas), the source term represents the contribution of particles, and must be handled with some care. For now, we ignore the source

term:

$$u(x,t+\Delta t) = u(x,t) + \Delta t v(x,t) + \frac{\Delta t^2}{2} c^2 \Delta u(x,t) + \frac{\Delta t^3}{6} c^2 \Delta v(x,t) + O(\Delta t^4)$$

$$v(x,t+\Delta t) = v(x,t) + \Delta t c^2 \Delta u(x,t) + \frac{\Delta t^2}{2} c^2 \Delta v(x,t) + \frac{\Delta t^3}{6} c^4 \Delta^2 u(x,t) + O(\Delta t^4)$$

In order to obtain an A-stable scheme, we replace the Laplacian operator and its powers, which are unbounded, with bounded approximating operators. We consider the one-dimensional case, with higher-dimensions to be handled through dimensional splitting. We use the identity from [11]:

$$\left(\frac{\partial_{xx}}{\alpha^2}\right)^m = (-1)^m \sum_{p=m}^{\infty} \binom{p-1}{m-1} \mathcal{D}^p$$

where

$$\mathcal{D}[u](x) = u(x) - \frac{\alpha}{2} \int_{-\infty}^{\infty} e^{-\alpha|x-y|} u(y)\, dy$$

We write out the first few terms of the approximations we need:

$$\frac{\partial_{xx}}{\alpha^2} = -\mathcal{D} - \mathcal{D}^2 + \cdots$$
$$\left(\frac{\partial_{xx}}{\alpha^2}\right)^2 = \mathcal{D}^2 + 2\mathcal{D}^3 + \cdots$$

Inserting:

$$\begin{aligned} u(x,t+\Delta t) &= u(x,t) + \Delta t v(x,t) + \frac{\Delta t^2}{2} c^2\alpha^2(-\mathcal{D}-\mathcal{D}^2)u(x,t) + \\ &+ \frac{\Delta t^3}{6} c^2\alpha^2(-\mathcal{D}-\mathcal{D}^2)v(x,t) + O(\Delta t^4) \\ v(x,t+\Delta t) &= v(x,t) + \Delta t c^2\alpha^2(-\mathcal{D}-\mathcal{D}^2)u(x,t) + \frac{\Delta t^2}{2} c^2\alpha^2(-\mathcal{D}-\mathcal{D}^2)v(x,t) + \\ &+ \frac{\Delta t^3}{6} c^4\alpha^4(\mathcal{D}^2+2\mathcal{D}^3)u(x,t) + O(\Delta t^4) \end{aligned}$$

Choosing $\alpha = \beta/c\Delta t$:

$$\begin{aligned} u(x,t+\Delta t) &= u(x,t) + \Delta t v(x,t) + \frac{\beta^2}{2}(-\mathcal{D}-\mathcal{D}^2)u(x,t) + \\ &+ \frac{\Delta t\beta^2}{6}(-\mathcal{D}-\mathcal{D}^2)v(x,t) + O(\Delta t^4) \\ v(x,t+\Delta t) &= v(x,t) + \frac{\beta^2}{\Delta t}(-\mathcal{D}-\mathcal{D}^2)u(x,t) + \frac{\beta^2}{2}(-\mathcal{D}-\mathcal{D}^2)v(x,t) + \\ &+ \frac{\beta^4}{6\Delta t}(\mathcal{D}^2+2\mathcal{D}^3)u(x,t) + O(\Delta t^4) \end{aligned}$$

Numerical experiments indicate that the following scheme, including select higher-order terms, maintains unconditional stability for $0 < \beta \leq 1$, with a small amount of dissipation:

$$
\begin{aligned}
u^{n+1} &= u^n + \Delta t v^n - \frac{\beta^2}{2}\left(\mathcal{D}u^n + \mathcal{D}^2 u^n\right) - \\
&\quad - \frac{\beta^2 \Delta t}{6}\left(\mathcal{D}v^n + \mathcal{D}^3 v^n\right) + \frac{\beta^4}{24}\mathcal{D}^2 u^n \\
v^{n+1} &= v^n - \frac{\beta^2}{\Delta t}\left(\mathcal{D}u^n + \mathcal{D}^2 u^n\right) - \frac{\beta^2}{2}\left(\mathcal{D}v^n + \mathcal{D}^2 v^n + \mathcal{D}^3 v^n\right) + \\
&\quad + \frac{\beta^4}{6\Delta t}\left(\mathcal{D}^2 u^n + 2\mathcal{D}^3 u^n\right) + \frac{\beta^4}{24}\mathcal{D}^2 v^n
\end{aligned}
\tag{2.13}
$$

This scheme possesses certain disadvantages compared to the centered schemes with artificial dissipation. Namely, it requires greater storage and computational requirements (requiring a larger number of convolutions), and the level of dissipation is not obviously tunable. On the other hand, it may be possible to further modify the coefficients of the higher-order terms to allow for tuning of the level of dissipation, and further, in electromagnetic PIC simulations, we need to compute the time derivative $\mathbf{A}_t$ in order to calculate the electric field, which would be computed via finite differences in the case of centered schemes, but would be given automatically by the Newmark method to fourth-order accuracy.

2.8 Embbeded Boundary Method for Neumann Boundary Conditions

In implementing Neumann boundary conditions for boundary geometries conforming to grid lines, such as a rectangular domain, we can directly impose a two-point boundary correction. One way to extend this method to a general polygonal domain would be to use multiple overset grids, each aligned with a boundary segment, which communicate with the interior grid through interpolation on a ghost cell region, though we do not pursue that approach in

this work.

For curved boundaries, an alternative approach is that of an embedded boundary method, which involves determining the Dirichlet values at the endpoints of each x- and y-sweep lines that result in the approximate satisfaction of the Neumann boundary condition (in effect, constructing an approximate Neumann-to-Dirichlet map). We present the implementation of an embedded boundary method for Neumann boundary conditions for the implicit wave solver on a curved boundary geometry. The approach taken here follows the work in [28], which proposes an embedded boundary method for Neumann boundary conditions with a finite difference method for the wave equation. The analysis in that work suggests that, on the continuous level, the modified equations and boundary conditions resulting from typical truncation error terms possess unstable boundary layer solutions, so that the addition of a dissipative term is necessary to achieve a stable method. This is consistent with our experience in the implementation described here, with the embedded boundary method becoming unstable when applied to the non-dissipative dispersive solver, but remaining stable for the diffusive solver, which is dissipative.

In the following, we briefly describe the two-point boundary correction method for a grid-aligned rectangular boundary. We then describe the embedded boundary method for a 1D problem. This method requires an iterative procedure, which we show, in the setting of the 1D problem, to be a convergent contraction mapping with a rate of convergence that depends on the CFL number. We describe the implementation of the embedded boundary method in 2D, and finally give numerical results.

2.8.1 Description of the Two-Point Boundary Correction Method

In a rectangular domain where the boundaries conform to grid lines, it is straightforward to impose the two-point boundary correction terms in a line-by-line fashion, since in this case, the grid lines are not coupled through the normal derivative. As this is a simple extension of the 1D boundary correction algorithm, we do not elaborate further.

2.8.2 Description of the Embedded Boundary Method and Proof of Convergence of the Iterative Solution in 1D

We consider the situation of a one-dimensional domain $\{x_B < x\}$ with a single boundary point not aligned with the grid points, as displayed in Figure 2.1. We have grid points $x_0, x_1, ...$ with uniform grid spacing $x_{i+1} - x_i = \Delta x$, boundary location x_B, and ghost point location $x_G = x_0$. We define interior points to be any grid points lying within the domain (including the boundary), and exterior points to be any grid points lying outside of the domain. We define a ghost point to be any exterior point for which at least one of the neighboring points $x_{i\pm1}$ is an interior point. We neglect the right boundary in the present analysis for simplicity, though it can be extended to the case with both boundaries. We consider applying the diffusive version of the wave solver, having calculated the convolution integral $I(x)$, and now needing to find the value of the coefficient A such that the solution $u(x)$ at the next time step is given by

$$u(x) = I(x) + Ae^{-\alpha(x-x_G)}, \quad x \geq x_0$$

Given the value of the convolution integral and the solution at the ghost point, $I_G =$

$I(x_G)$ and $u_G = u(x_G)$, respectively, the coefficient may be computed as $A = u_G - I_G$. We now describe the procedure for determining the value of the solution at the ghost point, u_G, that leads to a solution consistent with homogeneous Neumann boundary conditions to second-order accuracy. We construct a quadratic interpolant using the boundary condition and interior interpolation points $x_I = x_B + \Delta s_I$ and $x_{II} = x_B + 2\Delta s_I$, lying in between grid points x_m and x_{m+1}, and x_n and x_{n+1}, respectively. The interpolation distance will be chosen such that $\Delta x < \Delta s_I < (3/2)\Delta x$. We define the distances $\xi_G = x_B - x_G$, $\xi_I = x_I - x_B = \Delta s_I$, and $\xi_{II} = x_{II} - x_B = 2\Delta s_I$, and construct a quadratic Hermite-Birkhoff [5] interpolant $P(\xi)$ by imposing the conditions $P'(0) = 0$, $P(\xi_I) = u_I$ and $P(\xi_{II}) = u_{II}$. We then obtain the following second-order approximation to the ghost point value, given by

$$\begin{aligned} u_G = P(\xi_G) + O(\Delta x^2) &= u_I \frac{\xi_{II}^2 - \xi_G^2}{\xi_{II}^2 - \xi_I^2} + u_{II} \frac{\xi_G^2 - \xi_I^2}{\xi_{II}^2 - \xi_I^2} + O(\Delta x^2) \\ &= \gamma_I u_I + \gamma_{II} u_{II}. \end{aligned}$$

As the coefficients $\gamma_I = \frac{\xi_{II}^2 - \xi_G^2}{\xi_{II}^2 - \xi_I^2} > 0$ and $\gamma_{II} = \frac{\xi_G^2 - \xi_I^2}{\xi_{II}^2 - \xi_I^2} < 0$ are $O(1)$, we only need supply second-order accurate approximations to u_I and u_{II} to maintain overall second-order accuracy. (The coefficients would be $O(1/\Delta x)$ in the case of nonhomogeneous Neumann boundary conditions, which would require third-order accurate approximations to u_I and u_{II}. For simplicity, we consider only the case of homogeneous Neumann boundary conditions in the present work.) Such approximations may be obtained through linear interpolation, giving

$$u_I = \sigma_I u_m + (1 - \sigma_I) u_{m+1}$$

$$u_{II} = \sigma_{II} u_n + (1 - \sigma_{II}) u_{n+1}$$

where $\sigma_I = \frac{x_{m+1} - x_I}{\Delta x}$, $\sigma_{II} = \frac{x_{n+1} - x_{II}}{\Delta x}$, and $u_j = u(x_j)$ are the values of the function at the uniform gridpoints for $j = m, m+1, n, n+1$.

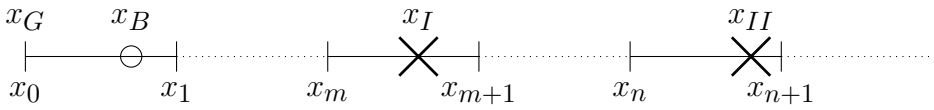


Figure 2.1: Boundary geometry in 1D.

Hence, to determine the ghost point value u_G that leads to a solution consistent with homogeneous Neumann boundary conditions, we should solve the following system of equations.

$$u_G = \gamma_I(\sigma_I u_m + (1 - \sigma_I) u_{m+1}) + \gamma_{II}(\sigma_{II} u_n + (1 - \sigma_{II}) u_{n+1})$$

$$u_m = I_m + (u_G - I_G) e^{-\alpha(x_m - x_0)}$$

$$u_{m+1} = I_{m+1} + (u_G - I_G) e^{-\alpha(x_{m+1} - x_0)}$$

$$u_n = I_n + (u_G - I_G) e^{-\alpha(x_n - x_0)}$$

$$u_{n+1} = I_{n+1} + (u_G - I_G) e^{-\alpha(x_{n+1} - x_0)}$$

with γ_I, γ_{II}, σ_I and σ_{II} defined as above, and where $I_j = I(x_j)$ is the convolution integral evaluated at uniform grid points for $j = m, m+1, n, n+1$. This system can be

solved formally for u_G, giving

$$\begin{aligned}
u_G = &\Big[\gamma_I \left(\sigma_I(I_m - I_G e^{-\alpha(x_m - x_G)}) + (1-\sigma_I)(I_{m+1} - I_G e^{-\alpha(x_{m+1} - x_G)})\right) + \\
&\gamma_{II} \left(\sigma_{II}(I_n - I_G e^{-\alpha(x_n - x_G)}) + (1-\sigma_{II})(I_{n+1} - I_G e^{-\alpha(x_{n+1} - x_G)})\right)\Big] \div \\
&\Big[1 - \gamma_I \left(\sigma_I e^{-\alpha(x_m - x_G)} + (1-\sigma_I) e^{-\alpha(x_{m+1} - x_G)}\right) - \\
&\gamma_{II} \left(\sigma_{II} e^{-\alpha(x_n - x_G)} + (1-\sigma_{II}) e^{-\alpha(x_{n+1} - x_G)}\right)\Big]
\end{aligned}$$

To show that this solution formula is well-defined, we argue that

$$\begin{aligned}
0 < K := &\gamma_I \left(\sigma_I e^{-\alpha(x_m - x_G)} + (1-\sigma_I) e^{-\alpha(x_{m+1} - x_G)}\right) + \\
&+ \gamma_{II} \left(\sigma_{II} e^{-\alpha(x_n - x_G)} + (1-\sigma_{II}) e^{-\alpha(x_{n+1} - x_G)}\right) < 1
\end{aligned}$$

for the relevant values of m, n and ξ_G, ξ_I, and ξ_{II}. We define $d = e^{-\alpha \Delta x}$, and noting that $0 < d < 1$, $m < n$, $\xi_G < \xi_I = \Delta s_I < \xi_{II} = 2\Delta s_I$, $\gamma_I > 0$, $\gamma_{II} < 0$, $0 \leq \sigma_I \leq 1$, and $0 \leq \sigma_{II} \leq 1$, we obtain

$$\begin{aligned}
K &= \gamma_I d^m \left[\sigma_I + (1-\sigma_I)d\right] + \gamma_{II} d^n \left[\sigma_{II} + (1-\sigma_{II})d\right] \\
&\leq \gamma_I d^m + \gamma_{II} d^{n+1} \\
&= \frac{d^m \xi_{II}^2 - d^{n+1}\xi_I^2 + \xi_G^2 (d^{n+1} - d^m)}{\xi_{II}^2 - \xi_I^2} \\
&\leq \frac{d^m \xi_{II}^2 - d^{n+1}\xi_I^2}{\xi_{II}^2 - \xi_I^2} \\
&= \frac{4\Delta s_I^2 d^m - d^{n+1}\Delta s_I^2}{4\Delta s_I^2 - \Delta s_I^2} = \frac{4d^m - d^{n+1}}{3}
\end{aligned}$$

Now, since $\Delta x < \Delta s_I < (3/2)\Delta x$, we can see that it is the case that either $m = 1$ and $n = 2$ or 3, or that $m = 2$ and $n = 3$. It is then a matter of some simple calculus to check that that the functions $f_{m,n}(x) = (4x^m - x^{n+1})/3$ satisfy $f_{m,n}(x) < 1$ for $0 < x < 1$ and the given combinations of m and n. This proves that $K < 1$ for the relevant values of the parameters, so that the solution for u_G is well-defined. We note, however, that K approaches 1 as d approaches 1, that is, as the CFL number becomes large. Thus, we may expect an ill-conditioned system when the CFL number is very large.

To obtain the lower bound on K, we observe

$$\begin{aligned}
K &= \gamma_I d^m \left[\sigma_I + (1-\sigma_I)d\right] + \gamma_{II} d^n \left[\sigma_{II} + (1-\sigma_{II})d\right] \\
&\geq \gamma_I d^{m+1} + \gamma_{II} d^n \\
&= \frac{d^{m+1}(\xi_{II}^2 - \xi_G^2) + d^n(\xi_G^2 - \xi_I^2)}{\xi_{II}^2 - \xi_I^2} \\
&= \frac{d^{m+1}(4\xi_I^2 - \xi_G^2) + d^n(\xi_G^2 - \xi_I^2)}{\xi_{II}^2 - \xi_I^2} \\
&= \frac{(d^{m+1} - d^n)(\xi_I^2 - \xi_G^2) + 3d^{m+1}\xi_I^2}{\xi_{II}^2 - \xi_I^2} > 0
\end{aligned}$$

In the two-dimensional case, the line-by-line solution method couples the ghost point values, and a general explicit solution formula is impossible to write down. In principle, one may write out and directly solve a linear system to obtain the ghost point values. Instead, we propose an iterative solution method that avoids the formation of a matrix. We now describe this iterative solution method and prove its convergence in the context of the one-dimensional problem described above.

Suppose we have the convolution integral evaluated at the gridpoints, I_j, and a k-th

iterate for the ghost point value, u_G^k. Then we may obtain the next iterate by the formulas

$$
\begin{aligned}
u_m^{k+1} &= I_m + (u_G^k - I_G)e^{-\alpha(x_m - x_0)} \\
u_{m+1}^{k+1} &= I_{m+1} + (u_G^k - I_G)e^{-\alpha(x_{m+1} - x_0)} \\
u_n^{k+1} &= I_n + (u_G^k - I_G)e^{-\alpha(x_n - x_0)} \\
u_{n+1}^{k+1} &= I_{n+1} + (u_G^k - I_G)e^{-\alpha(x_{n+1} - x_0)} \\
u_G^{k+1} &= \gamma_I(\sigma_I u_m^{k+1} + (1-\sigma_I)u_{m+1}^{k+1}) + \gamma_{II}(\sigma_{II} u_n^{k+1} + (1-\sigma_{II})u_{n+1}^{k+1})
\end{aligned}
$$

where quantities are defined as above. Now, to prove the convergence of the interation, we show it is contractive. Taking the difference of two iterates, we have

$$
\begin{aligned}
|u_G^{k+1} - u_G^k| &= |\gamma_I(\sigma_I(u_m^{k+1} - u_m^k) + (1-\sigma_I)(u_{m+1}^{k+1} - u_{m+1}^k)) + \\
&\qquad \gamma_{II}(\sigma_{II}(u_n^{k+1} - u_n^k) + (1-\sigma_{II})(u_{n+1}^{k+1} - u_{n+1}^k))| \\
&= |\gamma_I \left(\sigma_I(u_G^k - u_G^{k-1})e^{-\alpha(x_m - x_G)} + (1-\sigma_I)(u_G^k - u_G^{k-1})e^{-\alpha(x_{m+1} - x_G)}\right) + \\
&\quad + \gamma_{II} \left(\sigma_{II}(u_G^k - u_G^{k-1})e^{-\alpha(x_n - x_G)} + (1-\sigma_{II})(u_G^k - u_G^{k-1})e^{-\alpha(x_{n+1} - x_G)}\right)| \\
&\leq K|u_G^k - u_G^{k-1}|
\end{aligned}
$$

where $0 < K < 1$ as defined above. Hence, the Contraction Mapping Theorem implies that the iteration converges to a unique fixed point (which is the solution given in the formula above). We note again that K approaches 1 as the CFL number becomes large, so that the rate of convergence will become slower for larger CFL numbers.

2.8.3 Description of the Method in 2D

We now describe the implementation of the embedded Neumann boundary condition in the 2D case. We consider the situation displayed in Figure 2.2, in which we need to determine the value of our unknown u_G at the ghost point location (x_G, y_G). In the 2D case, we define a ghost point to be any exterior point (x_i, y_j) for which at least one of the neighboring points $(x_{i\pm1}, y_j)$ or $(x_i, y_{j\pm1})$ is an interior point. Similarly to the 1D case, we will construct a quadratic Hermite-Birkhoff boundary interpolant $P(\xi)$ along the direction normal to the boundary, which intersects the boundary curve Γ at location (x_B, y_B), and supply the interior interpolation point values u_I and u_{II}, at points (x_I, y_I) and (x_{II}, y_{II}), respectively, by further interpolation from interior grid points. These points are selected along the normal, in analogy to the 1D case, such that $\xi_I = |(x_I, y_I) - (x_B, y_B)| = \Delta s_I$ and $\xi_{II} = |(x_{II}, y_{II}) - (x_B, y_B)| = 2\Delta s_I$, where we will typically take $\Delta s_I = \sqrt{2}\Delta x$. We construct a quadratic Hermite-Birkhoff interpolant $P(\xi)$ by imposing the conditions $P'(0) = 0$, $P(\xi_I) = u_I$ and $P(\xi_{II}) = u_{II}$. Defining further the distance from the boundary to the ghost point $\xi_G = |(x_G, y_G) - (x_B, y_B)|$, we obtain, as in the 1D case, the following second-order approximation to the ghost point value, given by

$$\begin{aligned} u_G = P(\xi_G) + O(\Delta x^2) &= u_I \frac{\xi_{II}^2 - \xi_G^2}{\xi_{II}^2 - \xi_I^2} + u_{II} \frac{\xi_G^2 - \xi_I^2}{\xi_{II}^2 - \xi_I^2} + O(\Delta x^2) \\ &= \gamma_I u_I + \gamma_{II} u_{II}. \end{aligned}$$

where the coefficients are defined as $\gamma_I = \frac{\xi_{II}^2 - \xi_G^2}{\xi_{II}^2 - \xi_I^2} > 0$ and $\gamma_{II} = \frac{\xi_G^2 - \xi_I^2}{\xi_{II}^2 - \xi_I^2} < 0$. In the 2D case, we find approximations to u_I and u_{II} through bilinear interpolation. This is in contrast

to [28], who find the intersection of the normal with grid lines, then interpolate along the grid lines. We have also implemented a second-order accurate version of this approach and compared to the bilinear interpolation scheme proposed here. We have found that the two schemes behave similarly, however the bilinear interpolation scheme is slightly more accurate and simpler to code, not requiring to handle separate cases of intersection with horizontal and vertical grid lines. The bilinear interpolation scheme is standard, but we give it here for completeness. If the interpolation point u_I lies in a cell with corners (x_i, y_j), (x_{i+1}, y_j), (x_{i+1}, y_{j+1}) and (x_i, y_{j+1}), then we have the following approximation for u_I:

$$u_I = w_1 u_{i,j} + w_2 u_{i+1,j} + w_3 u_{i+1,j+1} + w_4 u_{i,j+1}$$

where $w_1 = \frac{(x_{i+1}-x_I)(y_{j+1}-y_I)}{\Delta x \Delta y}$, $w_2 = \frac{(x_I-x_i)(y_{j+1}-y_I)}{\Delta x \Delta y}$, $w_3 = \frac{(x_I-x_i)(y_I-y_j)}{\Delta x \Delta y}$ and $w_4 = \frac{(x_{i+1}-x_I)(y_I-y_j)}{\Delta x \Delta y}$. With this interpolation scheme established, we now outline the algorithm for the 2D dimensionally-split wave solver.

The above interpolation procedure applies regardless of the variety of the wave solver that it is used with, provided that the wave solver has sufficient dissipation to maintain stability. We now describe the rest of the embedded boundary algorithm in the context of the diffusive wave solver, though it it may be similarly applied to the the dispersive schemes with artificial dissipation or the dissipative Newmark scheme described above. In analogy to the iteration presented in the 1D case, the 2D iterative algorithm proceeds by using the interpolation scheme to provide values at the ghost points, which in turn provide new values for the boundary correction coefficients, which are then used to update the values at the interior grid points, comprising one full iteration. It should be noted that not all interior

grid points need be updated in the iteration, only those near the boundary that lie within the boundary interpolation stencils.

Using values from previous time steps, the initial guess for the interior grid points in the boundary interpolation stencils may be given by extrapolation in time, as $u^{n+1,0} = 2u^n - u^{n-1}$ (linear extrapolation) or $u^{n+1,0} = 3u^n - 3u^{n-1} + u^{n-2}$ (quadratic extrapolation). Either extrapolated initial guess provides a modest reduction in the number of iterations required versus a zero initial guess, with only a slight further reduction in the number of iterations going from linear to quadratic extrapolation. An effective stopping criterion for iteration is $|u^{n+1,l+1} - u^{n+1,l}|_\infty < \epsilon$, where ϵ is some chosen tolerance, which may be chosen to be quite small, as the iteration is a fixed point interation. In the numerical example, we choose a tolerance of 10^{-15}, and we achieve convergence in less than 40 iterations at a CFL number of 2.

In applying the diffusive version of the wave solver, we assume we have previous time steps u^n, u^{n-1} and u^{n-2}. We have to solve the modified Helmholtz equation with homogeneous Neumann boundary conditions,

$$\left(1 - \frac{1}{\alpha^2}\nabla^2\right) u^{n+1} = \frac{1}{2}\left(5u^n - 4u^{n-1} + u^{n-2}\right) \text{ in } \Omega$$

$$\frac{\partial u}{\partial n} = 0 \text{ on } \Gamma = \partial\Omega$$

where $\alpha = \frac{\sqrt{2}}{c\Delta t}$. We apply dimensional splitting to find

$$\left(1 - \frac{1}{\alpha^2}\nabla^2\right) u^{n+1} = \left(1 - \frac{1}{\alpha^2}\partial_{xx}\right)\left(1 - \frac{1}{\alpha^2}\partial_{yy}\right) u^{n+1} + O\left(\frac{1}{\alpha^4}\right)$$

so we define $w = \left(1 - \frac{1}{\alpha^2}\partial_{yy}\right) u$, and noting that $w = u + O\left((c\Delta t)^2\right)$ so that $\frac{\partial w}{\partial n} = \frac{\partial u}{\partial n} + O\left((c\Delta t)^2\right)$, we obtain the following approximate system

$$\left(1 - \frac{1}{\alpha^2}\partial_{xx}\right) w^{n+1} = \frac{1}{2}\left(5u^n - 4u^{n-1} + u^{n-2}\right) \text{ in } \Omega$$

$$\frac{\partial w}{\partial n}^{n+1} = 0 \text{ on } \Gamma = \partial\Omega$$

$$\left(1 - \frac{1}{\alpha^2}\partial_{yy}\right) u^{n+1} = w^{n+1} \text{ in } \Omega$$

$$\frac{\partial u}{\partial n}^{n+1} = 0 \text{ on } \Gamma = \partial\Omega$$

We now suppose our domain is embedded in a uniform Cartesian grid, with horizontal grid lines corresponding to $y = y_k$, $1 \leq k \leq N_y$ and vertical grid lines corresponding to $x = x_j$, $1 \leq j \leq N_x$. The embedded boundary algorithm will be applied when calculating the intermediate variable w^{n+1} in horizontal line sweeps as well as the solution variable u^{n+1} in vertical line sweeps. The iterations for these two variables are separate; first, the iterative procedure is applied to w to convergence, and then this value of w is used to compute u, and the iterative procedure is applied to u to convergence. However, in each iteration, the grid lines are coupled through the interpolation scheme, so that all grid lines must be iterated together. The overall iterative algorithm is described in 1, with details specified for the iteration on w. The iteration on u is very similar, and so we omit the details.

2.9 Summary Table For Second-Order Wave Solver

Algorithm 1 Application of Neumann Boundary Conditions in 2 dimensions

1. **(Initialization of ghost points)** Perform the interpolation scheme described above to obtain the values of u^n, u^{n-1}, and u^{n-2} at the ghost points, which are the endpoints of the horizontal and vertical grid lines.

2. **(Evaluation of particular solution)** For each horizontal line $y = y_k$, for $1 \leq k \leq N_y$, with ghost (end) points $x = a_k$ and b_k find the particular solution $w_{p,k}^{n+1}$ for the intermediate variable $w_k^{n+1}(x)$ by evaluating the discrete convolution operator

$$w_{p,k}^{n+1}(x_j) = \frac{\alpha}{4} \int_{a_k}^{b_k} [5u^n - 4u^{n-1} + u^{n-2}](x', y_k) e^{-\alpha|x_j - x'|}\, dx'$$

   for each grid point x_j in the horizontal line, including the ghost points.

3. **(Boundary correction initialization)** For each horizontal line $y = y_k$, set the initial guess for the intermediate variable via extrapolation, $w_k^{n+1,0} = 3w_k^n - 3w_k^{n-1} + w_k^{n-2}$, on the interior points within the boundary interpolation stencil.

4. **(Boundary correction iteration)** For each horizontal line $y = y_k$, perform the interpolation scheme using the interior values of $w_{p,k}^{n+1,l}$ to find the ghost point values. Using these ghost point values, apply the boundary correction on each line to obtain the updated intermediate variable,

$$w_k^{n+1,l+1}(x_j) = w_{p,k}^{n+1}(x_j) + A_k e^{-\alpha(x_j - a_k)} + B_k e^{-\alpha(b_k - x_j)}$$

   for the values of x_j lying within the boundary interpolation stencil, where $A_k = \frac{w_k^{n+1,l}(a_k) - w_{p,k}^{n+1}(a_k) - \mu_k\left(w_k^{n+1,l}(b_k) - w_{p,k}^{n+1}(b_k)\right)}{1-\mu_k^2}$, $B_k = \frac{w_k^{n+1,l}(b_k) - w_{p,k}^{n+1}(b_k) - \mu_k\left(w_k^{n+1,l}(a_k) - w_{p,k}^{n+1}(a_k)\right)}{1-\mu_k^2}$, $\mu_k = e^{-\alpha(b_k - a_k)}$. Check for convergence, and if converged, store the intermediate variable w^{n+1}.

5. Repeat this process for the vertical line sweeps, using the intermediate variable w^{n+1} to calculate the particular solution for u^{n+1}, then apply the bounday correction interation.

Modified Helmholtz Equation	Dispersive Scheme, $\alpha = \frac{2}{c\Delta t}$: $\left(-\frac{1}{\alpha^2}\nabla^2 + 1\right)\left(u^{n+1} + 2u^n + u^{n-1}\right) = 4u^n + \frac{4}{\alpha^2}f(x,t^n)$ Diffusive Scheme, $\alpha = \frac{\sqrt{2}}{c\Delta t}$: $\left(-\frac{1}{\alpha^2}\nabla^2 + 1\right)u^{n+1} = \frac{1}{2}\left(5u^n - 4u^{n-1} + u^{n-2}\right) + \frac{1}{\alpha^2}f(x,t^{n+1})$
Dimensionally Split Modified Helmholtz Equation	$\left(-\frac{1}{\alpha^2}\nabla^2 + 1\right)u = f \Rightarrow$ $\left(-\frac{1}{\alpha^2}\frac{\partial^2}{\partial x^2} + 1\right)\left(-\frac{1}{\alpha^2}\frac{\partial^2}{\partial y^2} + 1\right)u = f \Rightarrow$ $\left(-\frac{1}{\alpha^2}\frac{\partial^2}{\partial x^2} + 1\right)w = f, \quad \left(-\frac{1}{\alpha^2}\frac{\partial^2}{\partial y^2} + 1\right)u = w$
1D Integral Solution	$\left(-\frac{1}{\alpha^2}\frac{d^2}{dx^2} + 1\right)u = f$ on $(a,b) \Rightarrow$ $u(x) = \frac{\alpha}{2}\int_a^b f(x')e^{-\alpha\lvert x-x'\rvert}\,dx' + Ae^{-\alpha(x-a)} + Be^{-\alpha(b-x)}$ $= I[f](x) + Ae^{-\alpha(x-a)} + Be^{-\alpha(b-x)}$
1D BC Correction Coefficients Dirichlet: $u(a) = u_a,\ u(b) = u_b$ Neumann: $u'(a) = v_a,\ u'(b) = v_b$ Periodic: $u(a) = u(b),\ u'(a) = u'(b)$	$A = \frac{(u_a - I_a) - \mu(u_b - I_b)}{1-\mu^2},\ B = \frac{(u_b - I_b) - \mu(u_a - I_a)}{1-\mu^2}$ $A = \frac{\mu(v_b + \alpha I_b) - (v_a - \alpha I_a)}{\alpha\left(1-\mu^2\right)},\ B = \frac{(v_b + \alpha I_b) - \mu(v_a - \alpha I_a)}{\alpha\left(1-\mu^2\right)}$ $A = \frac{I_b}{1-\mu},\ B = \frac{I_a}{1-\mu}$ $I_a = I[f](a),\ I_b = I[f](b),\ \mu = e^{-\alpha(b-a)}$

Fast Convolution Algorithm	$a = x_0 < x_1 < \cdots < x_N = b,$
	$x_{j+1} - x_j + \Delta x,\ j = 0, ..., N-1$
	$I_j = I[f](x_j) = \frac{\alpha}{2} \int_a^b f(x') e^{-\alpha \lvert x_j - x' \rvert}\, dx' = I_j^L + I_j^R$
	$I_j^L = \frac{\alpha}{2} \int_a^{x_j} f(x') e^{-\alpha \lvert x_j - x' \rvert}\, dx',\ I_j^R = \frac{\alpha}{2} \int_{x_j}^b f(x') e^{-\alpha \lvert x_j - x' \rvert}\, dx'$
	$I_0^L = 0,\ I_j^L = e^{-\alpha \Delta x} I_{j-1}^L + J_j^L,$
	$J_j^L = \frac{\alpha}{2} \int_{x_{j-1}}^{x_j} f(x') e^{-\alpha \lvert x_j - x' \rvert}\, dx',\ j = 1, ..., N$
	$I_N^R = 0,\ I_j^R = e^{-\alpha \Delta x} I_{j+1}^R + J_j^R,$
	$J_j^R = \frac{\alpha}{2} \int_{x_j}^{x_{j+1}} f(x') e^{-\alpha \lvert x_j - x' \rvert}\, dx',\ j = N-1, ..., 0$
Second Order Quadrature	$J_j^R = Pf(x_j) + Qf(x_{j+1}) + R(f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))$
	$J_j^L = Pf(x_j) + Qf(x_{j-1}) + R(f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))$
	$P = 1 - \frac{1-d}{\nu},\ Q = -d + \frac{1-d}{\nu},\ R = \frac{1-d}{\nu^2} - \frac{1+d}{2\nu}$
	$\nu = \alpha \Delta x,\ d = e^{-\nu}$

2.10 Numerical Examples

2.10.1 Rectangular Cavity With Domain Decomposition

In this section, we demonstrate the second order convergence of our proposed method, including domain decomposition, for a simple rectangular cavity problem with homogeneous Dirichlet and Neumann boundary conditions. A rectangular domain $\Omega = [-L_x/2, L_x/2] \times [-L_y/2, L_y/2]$ is divided into four subdomains with new artificial boundaries along $x = 0$ and $y = 0$, as in Figure 2.3. Due to the Cartesian geometry of this example, the domain decomposition algorithm we use follows directly from the 1D algorithm we have presented. A

more general domain decomposition algorithm will be implemented on complex subdomains, and is a subject of future investigation.

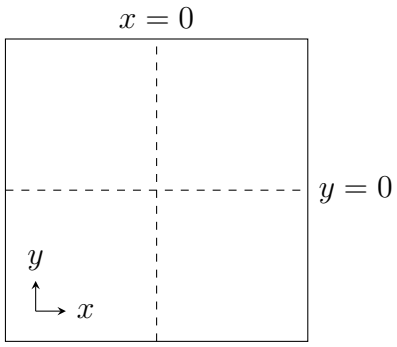


Figure 2.3: Rectangular cavity with domain decomposition.

As initial conditions, we choose

$$u(x, y, 0) = \begin{cases} \cos\left(\frac{(2m+1)\pi x}{L_x}\right) \cos\left(\frac{(2n+1)\pi y}{L_y}\right) & \text{Dirichlet case} \\ \sin\left(\frac{(2m+1)\pi x}{L_x}\right) \sin\left(\frac{(2n+1)\pi y}{L_y}\right) & \text{Neumann case} \end{cases}$$

and

$$u_t(x, y, 0) = 0$$

for $(x, y) \in \Omega$, m and n integers. Exact solutions are well-known in each case. The results of refinement studies are listed in Tables 2.1 and 2.2. The error is the maximum discrete L^2 error (computed against the exact solution) over all time steps.

2.10.2 Double Circle Cavity

In this example, we solve the wave equation with homogeneous Dirichlet boundary conditions on a 2D domain Ω which is, as in Figure 2.4, the union of two overlapping disks, with centers

	CFL 0.5		CFL 2		CFL 10	
Δx	L^2 error	L^2 order	L^2 error	L^2 order	L^2 error	L^2 order
1/40	8.4801×10^{-4}	–	5.8919×10^{-3}	–	1.0692×10^{-1}	–
1/80	1.9018×10^{-4}	2.15671	1.4578×10^{-3}	2.01499	3.3924×10^{-2}	1.65620
1/160	4.4811×10^{-5}	2.08544	3.6061×10^{-4}	2.01523	8.8683×10^{-3}	1.93555
1/320	1.0861×10^{-5}	2.04468	8.9550×10^{-5}	2.00965	2.2080×10^{-3}	2.00592
1/640	2.6726×10^{-6}	2.02284	2.2300×10^{-5}	2.00535	5.4638×10^{-4}	2.01476

Table 2.1: Refinement study for rectangular cavity with Dirichlet BC using domain decomposition. Here, $c = 1$, $m = n = 0$, and $L_x = L_y = 1$.

	CFL 0.5		CFL 2		CFL 10	
Δx	L^2 error	L^2 order	L^2 error	L^2 order	L^2 error	L^2 order
1/40	8.8928×10^{-4}	–	6.1862×10^{-3}	–	1.1147×10^{-1}	–
1/80	1.9486×10^{-4}	2.1902	1.4942×10^{-3}	2.0497	3.4771×10^{-2}	1.6807
1/160	4.5367×10^{-5}	2.1027	3.6511×10^{-4}	2.0329	8.9791×10^{-3}	1.9532
1/320	1.0929×10^{-5}	2.0535	9.0110×10^{-5}	2.0185	2.2217×10^{-3}	2.0148
1/640	2.6809×10^{-6}	2.0273	2.2370×10^{-5}	2.0098	5.4800×10^{-4}	2.0192

Table 2.2: Refinement study for rectangular cavity with Neumann BC using domain decomposition. Here, $c = 1$, $m = n = 0$, and $L_x = L_y = 1$.

$P_1 = (-\gamma, 0)$ and $P_2 = (\gamma, 0)$, respectively, and each with radius R:

$$\Omega = \{(x, y) : |\,(x, y) - P_1| < R\} \cup \{(x, y) : |\,(x, y) - P_2| < R\}$$

where $|\,(x, y)\,| = \sqrt{x^2 + y^2}$ is the usual Euclidean vector norm, and $\gamma < R$.

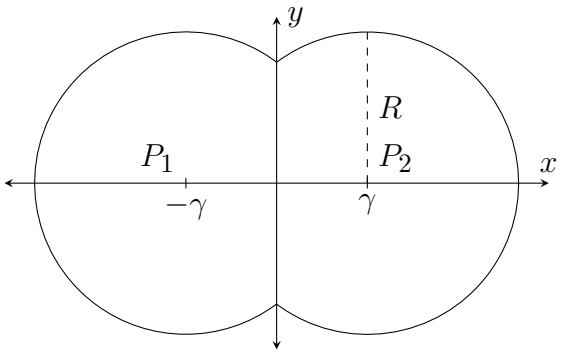


Figure 2.4: Double circle geometry.

This geometry is of interest due to, for example, its similarity to that of the radio frequency (RF) cavities used in the design of linear particle accelerators, and presents numerical difficulties due to the curvature of, and presence of corners in, the boundary. Our method avoids the staircase approximation used in typical finite difference methods to handle curved boundaries, which reduces accuracy to first order and may introduce spurious numerical diffraction.

As initial conditions, we choose

$$u(x,y,0) = \begin{cases} -\cos^6\left(\frac{\pi}{2}\left(\frac{|(x,y)-P_1|}{0.8\gamma}\right)^2\right) & |(x,y)-P_1| < 0.8\gamma \\ \cos^6\left(\frac{\pi}{2}\left(\frac{|(x,y)-P_2|}{0.8\gamma}\right)^2\right) & |(x,y)-P_2| < 0.8\gamma \\ 0 & \text{otherwise} \end{cases}$$

and

$$u_t(x,y,0) = 0$$

for $(x,y) \in \Omega$. Selected snapshots of the evolution are given in Figure 2.5, and the results of a refinement study are given in Table 2.3. The discrete L^2 error was computed against a well-refined numerical reference solution ($\Delta x = 2.1875 \times 10^{-4}$); the error displayed in the table is the maximum over time steps with $t \in [0.28, 0.29]$. For this example, $R = 0.3$, $\gamma = 0.2$, $c = 1$, and the CFL is 2.

Δx	Δy	Δt	L^2 error	L^2 order
7.0000×10^{-3}	4.3333×10^{-3}	8.6667×10^{-3}	6.1437×10^{-3}	–
3.5000×10^{-3}	2.1667×10^{-3}	4.3333×10^{-3}	1.6829×10^{-3}	1.8681
1.7500×10^{-3}	1.0833×10^{-3}	2.1667×10^{-3}	4.3595×10^{-4}	1.9488
8.7500×10^{-4}	5.4167×10^{-4}	1.0833×10^{-3}	1.0515×10^{-4}	2.0517

Table 2.3: Refinement study for the double circle cavity with Dirichlet BC. For the numerical reference solution, $\Delta x = 2.1875 \times 10^{-4}$, $\Delta y = 1.3542 \times 10^{-4}$, and $\Delta t = 2.7083 \times 10^{-4}$.

2.10.3 Periodic Slit Diffraction Grating

In this example, we apply our method to model an infinite, periodic diffraction grating under an incident plane wave. Diffraction gratings are periodic structures used in optics to separate different wavelengths of light, much like a prism. The high resolution that can be achieved with diffraction gratings makes them useful in spectroscopy, for example, in the determination of atomic and molecular spectra. Our numerical experiment, depicted in Figure 2.6, demonstrates the use of our method with multiple boundary conditions and nontrivial geometry in a single simulation to capture complex wave phenomena.

An idealized slit diffraction grating consists of a reflecting screen of vanishing thickness, with open slits of aperture width a, spaced distance d apart, measured from the end of one slit to the beginning of the next (that is, the periodicity of the grating is d). We impose an incident plane wave of the form $u_{inc}(x, y, t) = \cos(\omega t + ky)$, where $k = 2\pi/a$ and $\omega = k/c$, where c is the wave speed. Periodic BCs at $x = \pm d/2$ (determining the periodicity of the grating), and homogeneous Dirichlet BCs are imposed at the screen. We also test outflow boundary conditions in multiple dimensions, which are imposed at $y = \pm L_y/2$.

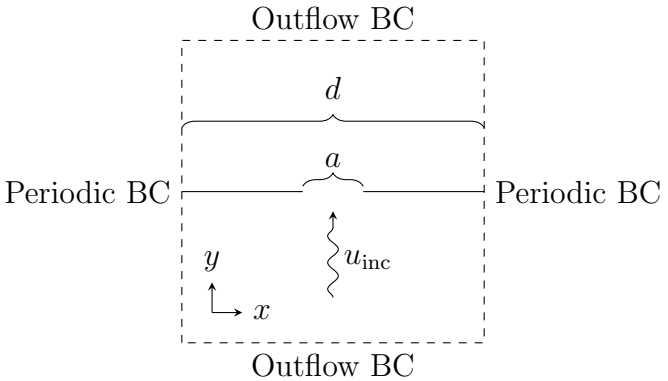


Figure 2.6: Periodic slit diffraction grating geometry

In Figure 2.7, we observe the time evolution of the incident plane wave passing through

the aperture, and the resulting interference patterns as the diffracted wave propagates across the periodic boundaries. The outflow boundary conditions allow the waves to propagate outside the domain. While a rigorous analysis of the efficacy of our outflow BCs is the subject of future work, the results look quite reasonable, as no spurious reflections are seen at the artificial boundaries.

2.10.4 Bessel Mode with Neumann Boundary Conditions

Here we present a numerical example of the embedded boundary method for homogeneous Neumann boundary conditions given in Section 2.8. We apply the method to a circular domain, for which analytical solutions exist. We consider a radially-symmetric Bessel mode with homogeneous Neumann boundary conditions, with an analytic solution given by

$$u(r,t) = J_0\left(Z_0 \frac{r}{R}\right) \cos\left(Z_0 \frac{ct}{R}\right), \tag{2.14}$$

where J_0 is the Bessel function of the first kind of order 0, $r = \sqrt{x^2+y^2}$, R is the radius of the domain, and $Z_0 \approx 3.8317$ is the smallest nonzero root of J_0' (so that $\frac{\partial u}{\partial \mathbf{n}}(R,t) = \frac{\partial u}{\partial r}(R,t) = \frac{Z_0}{R} J_0'(Z_0) \cos\left(Z_0 \frac{ct}{R}\right) = 0$). In this example, we take radius $R = \pi/2$ and wave speed $c = 1$. An example of the embedded boundary grid used is given in Figure 2.8. We perform a refinement study with a fixed CFL number of 2, with the results in Figure 2.9 indicating the expected second-order convergence. We set the iteration tolerance to 10^{-15}, and we see convergence of the boundary correction iteration in fewer than 40 iterations. We note some oscillation of the L^∞ error about the line giving second-order accuracy, which we believe to be due to the grid points moving with respect to the boundary through the

refinement, causing some variation in the maximum error.

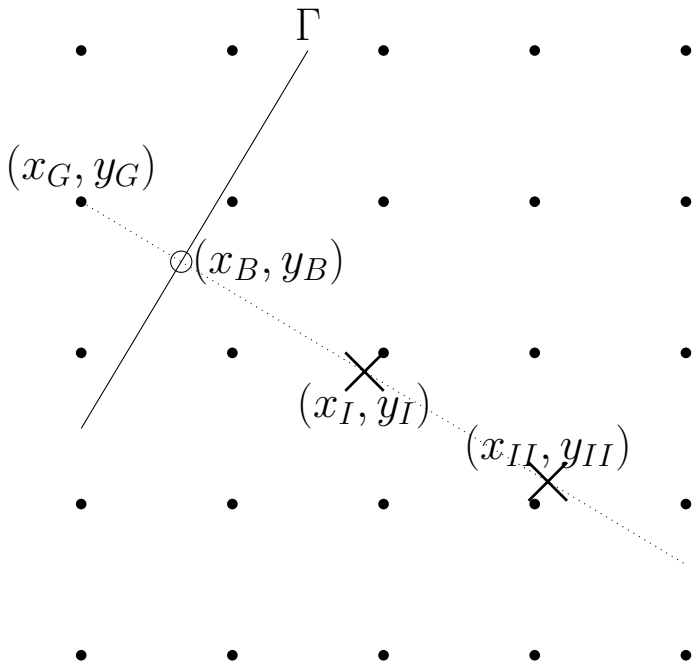
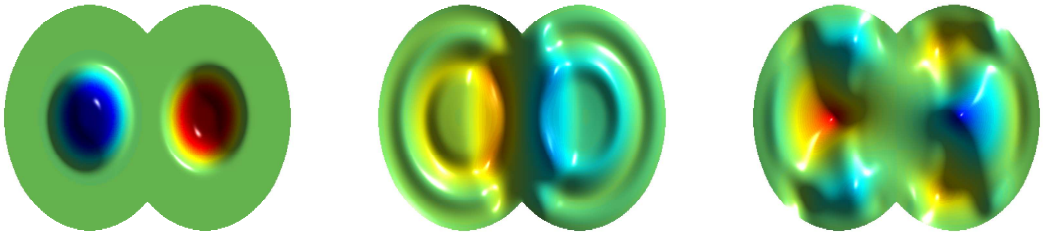
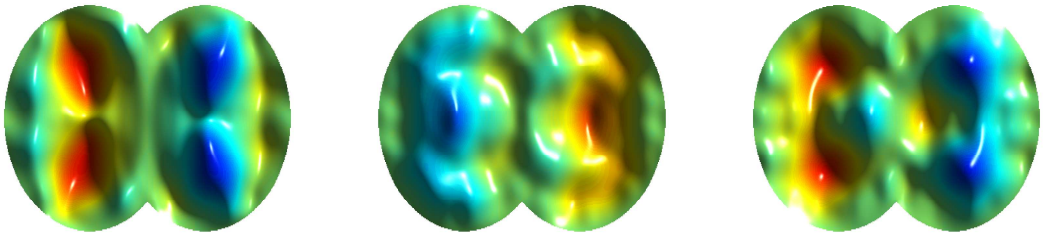


Figure 2.2: Boundary geometry in 2D.

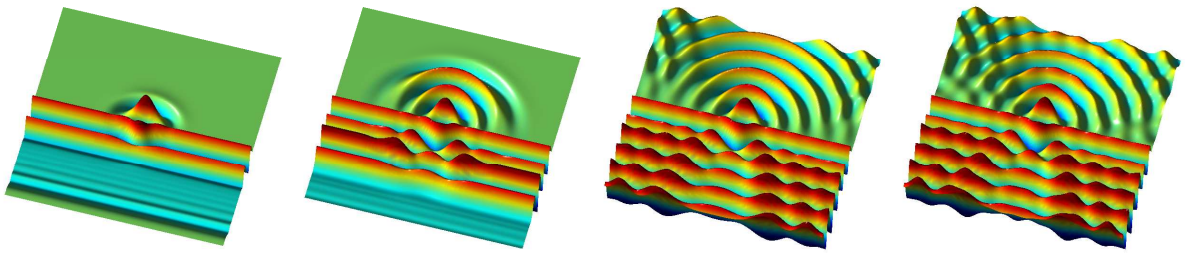


(a) $t = 0$ (b) $t = 0.257250$ (c) $t = 0.504000$



(d) $t = 1.097250$ (e) $t = 1.506750$ (f) $t = 1.753500$

Figure 2.5: Evolution of the double circle cavity problem.



(a) $t = 0.31$ (b) $t = 0.51$ (c) $t = 1.01$ (d) $t = 2.01$

Figure 2.7: Evolution of the slit diffraction grating problem, with aperture width $a = 0.1$, grating periodicity $L_y = d = 1$, and wave speed $c = 1$. The CFL is fixed at 2.

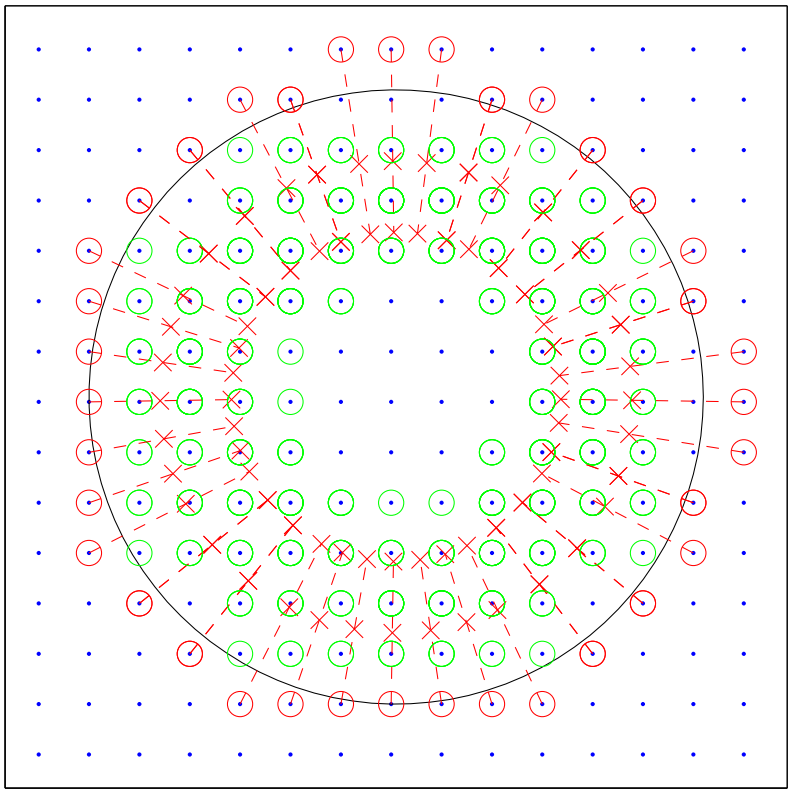


Figure 2.8: An example of the embedded boundary grid used. The red circled exterior grid points are the endpoints where a value is to be calculated via the interpolation procedure. The red crosses are the points where values are imposed on quadratic boundary interpolant along the normal direction (red dashed line). Values for the bilinear interpolants are supplied from the green circled interior grid points.

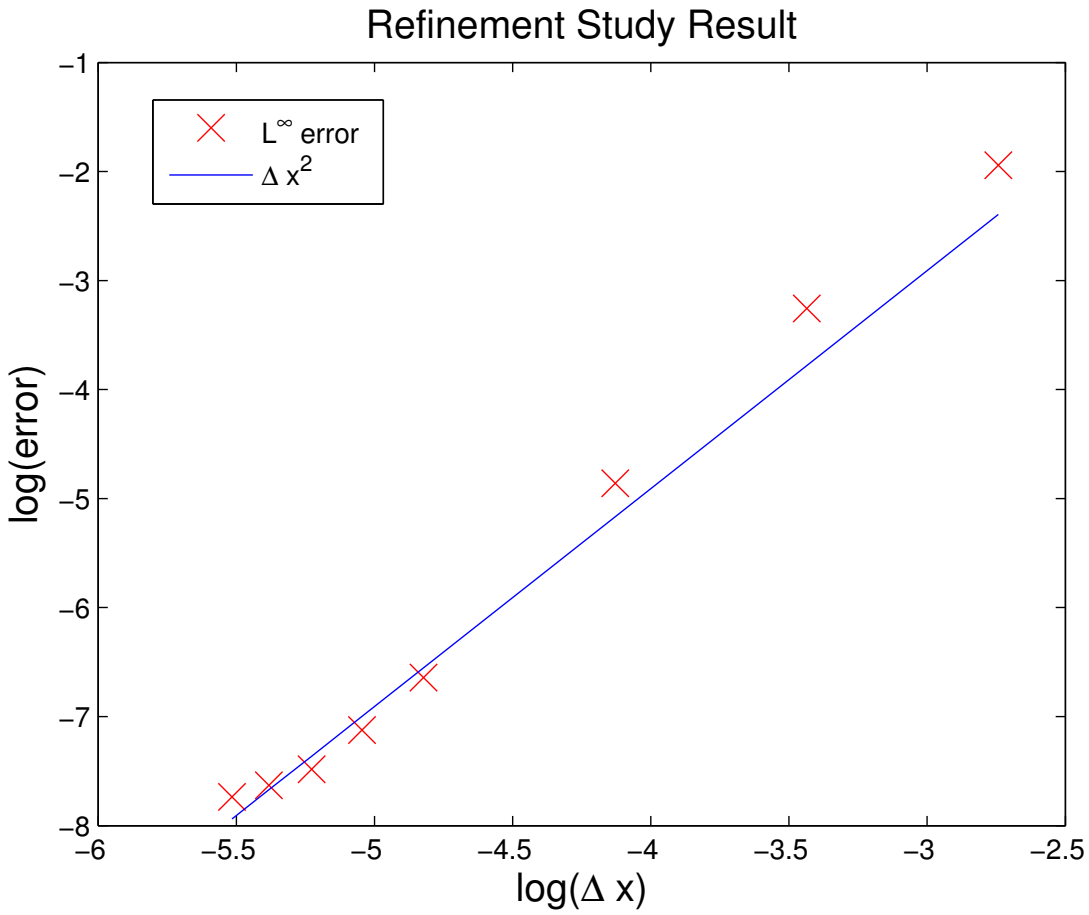


Figure 2.9: Refinement study for the Bessel mode in a circular domain with fixed CFL number of 2. Using quadratic boundary interpolant with bilinear interior interpolant.

Chapter 3

Description of the particle-in-cell method

3.1 Particle Weighting Scheme

In our PIC methods, the charge and current densities are represented as the sum of particle shape functions:

$$\rho(\mathbf{x},t) = \sum_{i=1}^{N_p} q_i S\left(\frac{\mathbf{x}-\mathbf{x}_{p,i}(t)}{\Delta x}\right) \tag{3.1}$$

$$\mathbf{J}(\mathbf{x},t) = \sum_{i=1}^{N_p} q_i \mathbf{v}_{p,i} S\left(\frac{\mathbf{x}-\mathbf{x}_{p,i}(t)}{\Delta x}\right) \tag{3.2}$$

where N_p is the number of particles, $\mathbf{x}_{p,i}(t)$ and $\mathbf{v}_{p,i}(t)$ are the position and velocity, respectively, of particle i at time t, and $S(\mathbf{x})$ is a particle shape function. It should be emphasized that these are not physical particles, but rather macro- or superparticles that represent a discretization of the PDF [4]. We analytically evaluate the particle convolution integrals corresponding to these source terms with the algorithms described below.

3.2 Method for Controlling Divergence Error

It is well known that electromagnetic PIC methods which do not satisfy a discrete form of Gauss' law through their field solvers and charge and current weighting schemes will suffer severe numerical errors related to charge conservation [37]. We seek to develop staggered grid approaches to computing potentials and fields that satisfy discrete analogues of Gauss' law $\nabla \cdot \mathbf{E} = \rho/\epsilon_0$ and the identity $\nabla \cdot \mathbf{B} = 0$. The divergence-free condition for $\mathbf{B}$ will be easily satisfied in general, as the magnetic field will be calculated as a finite difference curl of the vector potential. In order to numerically enforce Gauss' law, we seek to perform an elliptic divergence correction. Future work will consider the alternative of hyperbolic divergence cleaning [41].

We now give the mathematical underpinning of our elliptic divergence correction technique. The electric field is calculated as $\mathbf{E} = -\nabla\Phi - \frac{\partial \mathbf{A}}{\partial t}$. Then Gauss' law may be rewritten as:

$$\rho/\epsilon_0 = \nabla \cdot \mathbf{E} \tag{3.3}$$

$$= \nabla \cdot (-\nabla\Phi - \frac{\partial \mathbf{A}}{\partial t}) \tag{3.4}$$

$$= -\Delta\Phi - \frac{\partial (\nabla \cdot \mathbf{A})}{\partial t}. \tag{3.5}$$

Thus, the scalar potential satisfies the Poisson equation

$$-\Delta\Phi = \rho/\epsilon_0 + \frac{\partial (\nabla \cdot \mathbf{A})}{\partial t} \tag{3.6}$$

Our method is based on the observation that if this Poisson equation is suitably dis-

cretized and solved on a staggered grid to provide the scalar potential used in calculating the electric field, then the electric field will automatically satisfy a discrete form of Gauss' law. While the exact form of the staggered grid will depend on which components of the current density and the electric and magnetic fields are retained in a given model, our method guarantees exact discrete divergence relations independently of the charge and current weighting schemes used, of the nature of the solver used for $\mathbf{A}$, and of the gauge condition specified. Details about the specific staggered grids used are given in 3.6.2, along with proofs of the exact discrete divergence relations. It should be noted that this procedure is an analogue of the elliptic divergence correction techniques presented in [32, 37], and also bears similarity to the method given in [40] to enforce the divergence relations when evolving the potentials in the Coulomb gauge. To impose outflow boundary conditions on the scalar potential, we solve an auxiliary wave equation for Φ using our wave solver with outflow boundary conditions, which then supplies the boundary values for Φ in the Poisson solve.

3.3 Particle Equations of Motion

In our PIC methods, the approximation of the evolution of the Vlasov equation amounts to the integration of the equations of motion of the particles:

$$\frac{d\mathbf{x}_{p,i}}{dt} = \mathbf{v}_{p,i}(t) \tag{3.7}$$

$$\frac{d\mathbf{v}_{p,i}}{dt} = \mathbf{a}_{p,i}(t) \tag{3.8}$$

where $\mathbf{a}_{p,i}(t)$ is the acceleration of particle i at time t. To evolve the particle equations of

motion, we obtain particle accelerations through the usual interpolation of fields from grid points to particle locations [4], and we use standard numerical methods, such as the explicit leapfrog method for electrostatic problems and the forward Euler method for electromagnetic problems (the implicit source term in the diffusive wave solver causes some difficulty in achieving higher-order accuracy in the integration of the particle equations of motion - for this reason the dispersive scheme and higher-order centered schemes given in [11] will be pursued in future work - but the accuracy achieved is sufficient for the numerical results in this work). Fields are calculated as finite differences of potentials on the grid. As these aspects are standard and not the focus of the present work, we do not elaborate further.

3.4 Fast Convolution Algorithm For Particle Sources

3.4.1 Fast Convolution Algorithm in 1D

We now describe the algorithm used for the fast exact evaluation of the convolution of charge and current density source terms due to particles. It has two main steps. There is a local deposit step and then a recursive sweep step. This basic structure is the same in all dimensions. For definiteness, we describe the application of the algorithm to linear particle shapes in one and two dimensions. However, it may be generalized to any separable particle shapes with compact support in any dimension. Note that this includes many widely used particle shapes in PIC algorithms, namely typical spline-based particle shapes and (suitably cut-off) Gaussian particle shapes. For the case of the charge density integral, the particle shape function S_p below is replaced by $q_p S_p$ and its contribution summed to the charge density integral, and for the case of the current density integral, S_p is replaced by $\mathbf{v}_p q_p S_p$ and its contribution summed to the current density integral.

3.4.1.1 Local Deposit Step in 1D

Consider particle p located in the cell $[x_m, x_{m+1}]$ in a uniform grid with cell length Δx. Let $S_p(x)$ be the shape function of the particle. Assume that the support of S_p has length $2r\Delta x$ for some integer r. The local deposit step then consists of analytically evaluating the integrals

$$J_{j+1}^{L,p} = \alpha \int_{x_j}^{x_{j+1}} S_p(x') e^{-\alpha(x_{j+1}-x')}\, dx' \tag{3.9}$$

$$J_{j}^{R,p} = \alpha \int_{x_j}^{x_{j+1}} S_p(x') e^{-\alpha(x'-x_j)}\, dx' \tag{3.10}$$

for $j = m - r, ..., m, ..., m + r$ and for each particle p, and summing their values on the grid.

For linear particle shapes (corresponding to $r = 1$), we have $S_p(x) = S(\frac{x-x_p}{\Delta x})$, where

$$S(x) = \begin{cases} 1 - |x| & |x| < 1, \\ 0 & |x| \geq 1 \end{cases} \tag{3.11}$$

Let $a = (x_p - x_m)/\Delta x$, where x_p is the location of the particle. For simplicity, let $x_p = 0$ and $\Delta x = 1$. For linear particle shapes, we then have the situation displayed in Figure 3.1.

The desired integrals are then easily evaluated for linear particle shapes:

$$J_m^{L,p} = \alpha \int_{-1}^{-a} (1 + x')e^{-\alpha(-a-x')}\, dx' \tag{3.12}$$

$$= (((1-a)\alpha - 1) + e^{\alpha a}e^{-\alpha})/\alpha \tag{3.13}$$

$$J_{m+1}^{L,p} = \alpha \int_{-a}^{1-a} (1 - |x'|)e^{-\alpha(1-a-x')}\, dx' \tag{3.14}$$

$$= (e^{-\alpha}((a-1)\alpha + 1) + a\alpha + 1 - 2e^{-\alpha}e^{\alpha a})/\alpha \tag{3.15}$$

$$J_{m+2}^{L,p} = \alpha \int_{1-a}^{1} (1 - x')e^{-\alpha(2-a-x')}\, dx' \tag{3.16}$$

$$= (e^{-\alpha}(-a\alpha - 1) + e^{-\alpha}e^{\alpha a})/\alpha \tag{3.17}$$

$$J_{m-1}^{R,p} = \alpha \int_{-1}^{-a} (1 + x')e^{-\alpha(x'-(-1-a)}\, dx' \tag{3.18}$$

$$= (e^{-\alpha}((a-1)\alpha - 1) + e^{-\alpha a})/\alpha \tag{3.19}$$

$$J_m^{R,p} = \alpha \int_{-a}^{1-a} (1 - |x'|)e^{-\alpha(x'-(-a)}\, dx' \tag{3.20}$$

$$= ((1-a)\alpha + 1 + e^{-\alpha}(-a\alpha + 1) - 2e^{-\alpha a})/\alpha \tag{3.21}$$

$$J_{m+1}^{R,p} = \alpha \int_{1-a}^{1} (1 - x')e^{-\alpha(x'-(1-a))}\, dx' \tag{3.22}$$

$$= (a\alpha - 1 + e^{-\alpha a})/\alpha \tag{3.23}$$

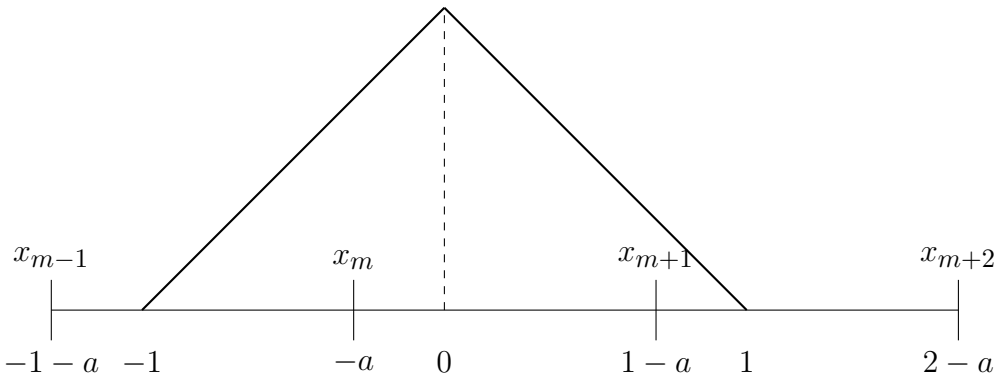


Figure 3.1: $S(x) = 1 - |x|, |x| < 1, S(x) = 0, |x| \geq 1$

Note that just one evaluation of an exponential function is required per particle (namely $e^{\alpha a}$). To account for arbitrary Δx, we make the substitution $\alpha \leftarrow \alpha \Delta x = \nu$.

To obtain the total local deposit, we simply sum the particle contributions on to the grid. Let N_p be the total number of particles. The algorithm for the local deposit step is given by:

Initialize $J_k^L = J_k^R = 0$ for all k.

for $p = 1 : N_p$ **do**

Particle p located in cell $[x_m, x_{m+1}]$.

for $j = m - r : m + r$ **do**

Compute $J_{j+1}^{L,p}$, $J_j^{R,p}$

Deposit $J_{j+1}^L = J_{j+1}^L + J_{j+1}^{L,p}$, $J_j^R = J_j^R + J_j^{R,p}$

end for

end for

Note that the local deposit step costs $O(N_p)$ operations.

3.4.1.2 Recursive Sweep Step in 1D

Once we have performed the local deposit step, we complete the evaluation of the particle integral with a recursive sweep step. Suppose we have N gridpoints, $x_1, ..., x_N$. The algorithm for the recursive sweep step is given by:

Initialize $I_1^L = I_N^R = 0$

for $j = 1 : N - 1$ **do**

$I_{j+1}^L = J_{j+1}^L + e^{-\nu} I_j^L$

$I_{N-j}^R = J_{N-j}^R + e^{-\nu} I_{N-j+1}^R$

end for

$$I = I^L + I^R$$

Note that the recursive sweep step costs $O(N)$ operations.

3.4.2 Fast Convolution Algorithm in 2D

For a separable particle shape $S(x, y) = S_x(x)S_y(y)$, we have

$$I_x[I_y[S]](x, y) = I_x[S_x](x) \cdot I_y[S_y](y) \tag{3.24}$$

$$= \Big(I_x^L[S_x](x) + I_x^R[S_x](x)\Big) \cdot \Big(I_y^D[S_y](y) + I_y^U[S_y](y)\Big) \tag{3.25}$$

$$= I_x^L[S_x](x) \cdot I_y^D[S_y](y) + I_x^L[S_x](x) \cdot I_y^U[S_y](y) + \tag{3.26}$$

$$+ I_x^R[S_x](x) \cdot I_y^D[S_y](y) + I_x^R[S_x](x) \cdot I_y^U[S_y](y) \tag{3.27}$$

This is suggestive of how we will build the 2D algorithm.

3.4.2.1 Local Deposit Step in 2D

Consider particle p centered at $(x_p, y_p) \in [x_m, x_{m+1}] \times [y_n, y_{n+1}]$, with separable particle shape $S_p(x, y) = S(\frac{x-x_p}{\Delta x})S(\frac{y-y_p}{\Delta y})$ where

$$S(x) = \begin{cases} 1 - |x| & |x| < 1, \\ 0 & |x| \geq 1 \end{cases} \tag{3.28}$$

The support of the particle shape is shown in Figure 3.2.

In the local deposit step, we form a tensor product on the grid as suggested by the above decomposition. Note that a total of 12 local integrals must be evaluated for each particle, then summed onto the grid as a tensor product.

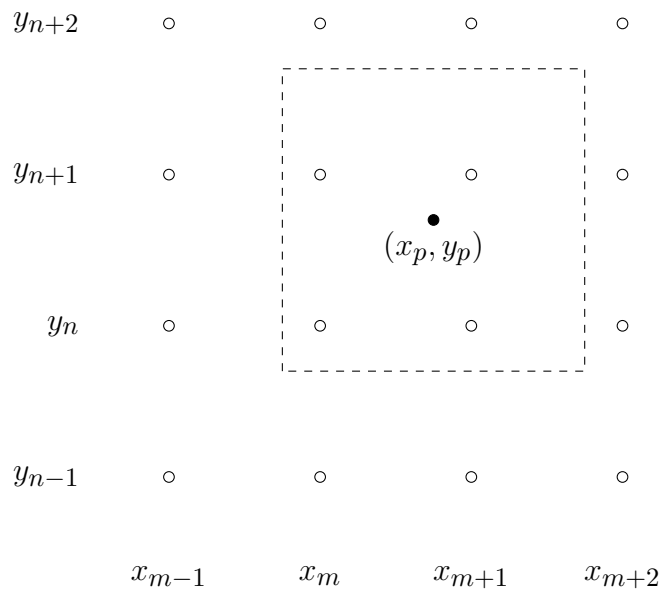


Figure 3.2: The support of a linear particle shape $Sp(x, y)$ in 2D.

Initialize $J^{LU}_{j,k} = J^{LD}_{j,k} = J^{RU}_{j,k} = J^{RD}_{j,k} = 0$ for all j, k.

for $p = 1 : N_p$ **do**

Particle p located in cell $[x_m, x_{m+1}] \times [y_n, y_{n+1}]$.

for $j = m - r : m + r$, $k = n - r : n + r$ **do**

Compute $J^{L,p}_{j+1}$, $J^{R,p}_{j}$, $J^{D,p}_{k+1}$, $J^{U,p}_{k}$

Deposit:

$$J^{LD}_{j+1,k+1} = J^{LD}_{j+1,k+1} + J^{L,p}_{j+1} \cdot J^{D,p}_{k+1}$$
$$J^{LU}_{j+1,k} = J^{LU}_{j+1,k} + J^{L,p}_{j+1} \cdot J^{U,p}_{k}$$
$$J^{RD}_{j,k+1} = J^{RD}_{j,k+1} + J^{R,p}_{j} \cdot J^{D,p}_{k+1}$$
$$J^{RU}_{j,k} = J^{RU}_{j,k} + J^{R,p}_{j} \cdot J^{U,p}_{k}$$

end for

end for

3.4.2.2 Recursive Sweep Step in 2D

The recursive sweep step is similar to the 1D case, and is given below.

for $k = 1 : N_y$ **do**

Initialize $I^{LD}_{1,k} = I^{LU}_{1,k} = I^{RD}_{N_x,k} = I^{RU}_{N_x,k} = 0.$

for $j = 1 : N_x - 1$ **do**

$$I^{LD}_{j+1,k} = J^{LD}_{j+1,k} + e^{-\nu} I^{LD}_{j,k}$$
$$I^{LU}_{j+1,k} = J^{LU}_{j+1,k} + e^{-\nu} I^{LU}_{j,k}$$
$$I^{RD}_{N_x-j,k} = J^{RD}_{N_x-j,k} + e^{-\nu} I^{RD}_{N_x-j+1,k}$$
$$I^{RU}_{N_x-j,k} = J^{RU}_{N_x-j,k} + e^{-\nu} I^{RU}_{N_x-j+1,k}$$

end for

end for

$J^D = I^{LD} + I^{RD}$

$J^U = I^{LU} + I^{RU}$

for $j = 1 : N_x$ **do**

Initialize $I^D_{j,1} = I^U_{j,N_y} = 0.$

for $k = 1 : N_y - 1$ **do**

$$I^D_{j,k+1} = J^D_{j,k+1} + e^{-\nu} I^D_{j,k}$$
$$I^U_{j,N_y-k} = J^U_{j,N_y-k} + e^{-\nu} I^U_{j,N_y-k+1}$$

end for

end for

$I = I^D + I^U$

Both 1D and 2D overall algorithms cost $O(N_p + N)$ operations, where N is the total number of gridpoints. Since in a typical PIC simulation, $N_p >> N$, the cost of the overall

algorithm is dominated by the local deposit step.

3.5 Particle Boundary Conditions

In dealing with boundaries, two types of considerations must be made. First, we must determine what to do in the integration of boundary particles, for which the support of the shape function extends outside of the domain. This will be dependent upon the type of boundary condition. For periodic boundary conditions, we can simply extend the particle shape function periodically and proceed to integrate. For Dirichlet boundary conditions, we extend the integration domain to include ghost points just beyond the boundary to which the boundary particles are weighted.

Second, we must ensure that the particle convolution integral is consistent with the boundary conditions on the wave function. This is easily handled through the usual boundary correction terms in one-dimension, and can be extended to the dimensionally-split multidimensional case.

3.5.1 1D Periodic BC

In the case of 1D periodic boundary conditions, we extend the shape function of the boundary particle as in Figure 3.3, and the local deposit step is carried out for this extended shape function. The recursive sweep step and the final boundary correction step are carried out as usual.

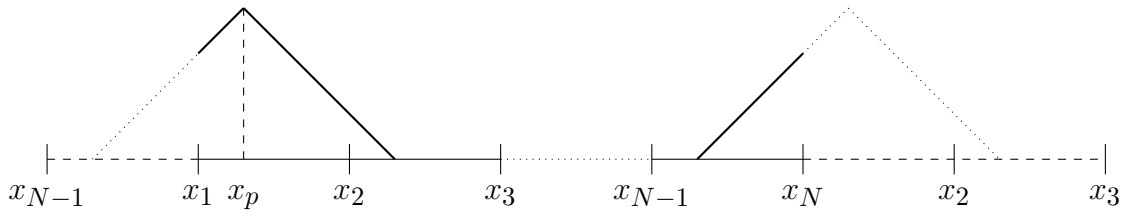


Figure 3.3: Periodic extension of shape function for boundary particle.

3.5.2 2D Periodic BC

Periodic boundary conditions are easily imposed in higher dimensions by similarly periodically extending the particle shape functions of boundary particles, peforming the local deposit step accordingly and the recursive sweep step as usual.

3.5.3 1D Dirichlet BC

In imposing homogeneous Dirichlet boundary conditions with boundary particles, we consider two possible approaches. The first approach simply extends the integration domain to include appropriate ghost cells extending just past the boundary to include the entire support of the particles. A boundary correction term of the form $Ae^{-\alpha x}$ (at the left boundary $x = 0$) to impose the boundary condition. Considering the left boundary only, we obtain a total potential of the form $\Phi(x) = \Phi_p(x) - \Phi_p(0)e^{-\alpha x}$, where Φ_p is the potential associated with the particle. The second approach places appropriate image particles on the opposite side of the boundary, while appropriately extending the integration domain with ghost cells to include the supports of the boundary particles and their image particles. Considering again the left boundary only, we obtain a total potential of the form $\Phi(x) = \Phi_p(x) + \Phi_{\text{img}}(x)$, where Φ_p is again the potential associated with the particle and Φ_{img} is the potential associated with the image particle.

We can verify that both approaches can impose the homogeneous Dirichlet boundary condition $\Phi(0) = 0$, but the potential and the fields associated with the approaches are not identical. Due to the difficulties that arise near corners or other geometric singularities of the boundary, it is preferable to avoid the use of image particles. We describe the above approaches for linear particle shapes, but the analysis can be extended to other particle shapes.

3.5.3.1 Analysis of the First Approach

Consider a grid $x_1 = 0$, $x_{j+1} = x_j + \Delta x$, $j = 1, 2, ...$ Consider a single boundary particle centered in the first cell at $x_p \in [x_1, x_2] = [0, \Delta x]$. The shape function of the particle is

$$S_p(x) = \begin{cases} 1 - |x - x_p|/\Delta x & |x - x_p| < \Delta x \\ \\ 0 & |x - x_p| \geq \Delta x \end{cases} \tag{3.29}$$

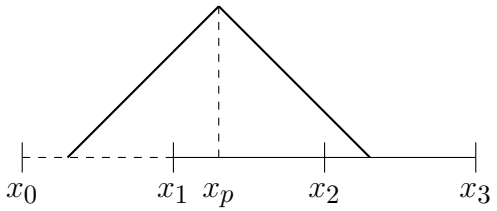


Figure 3.4: Particle shape function $S_p(x)$ of a boundary particle with ghost cell $[x_0, x_1] = [-\Delta x, 0]$.

The potential associated with the particle is

$$\Phi_p(x) = I[S_p](x) \tag{3.30}$$

$$= \alpha \int_{x_p-\Delta x}^{x_p+\Delta x} e^{-\alpha|x-y|} S_p(y)\, dy \tag{3.31}$$

$$= \alpha \int_{x_p-\Delta x}^{x} e^{-\alpha(x-y)} S_p(y)\, dy + \tag{3.32}$$

$$+ \alpha \int_{x}^{x_p+\Delta x} e^{-\alpha(y-x)} S_p(y)\, dy \tag{3.33}$$

$$= I_L[S_p](x) + I_R[S_p](x) \tag{3.34}$$

The field associated with the particle is

$$E_p(x) = -\frac{d}{dx}\Phi_p(x) \tag{3.35}$$

$$= \alpha I_L[S_p](x) - \alpha I_R[S_p](x) \qquad |x - x_p| < \Delta x \tag{3.36}$$

$$= \alpha I_L[S_p](x) \qquad x \geq x_p + \Delta x \tag{3.37}$$

To impose a homogeneous Dirichlet boundary condition at $x = 0$ with a correction term $Ae^{-\alpha x}$, we choose $A = -\Phi_p(0)$. Thus, the total potential in the first approach is given by

$$\Phi(x) = \Phi_p(x) - \Phi_p(0)e^{-\alpha x} \tag{3.38}$$

The total field for the first approach is given by

$$E(x) = -\frac{d}{dx}\Phi(x) \tag{3.39}$$

$$= \alpha I_L[S_p](x) - \alpha I_R[S_p](x) + \alpha\Phi_p(0)e^{-\alpha x} \qquad |x - x_p| < \Delta x \tag{3.40}$$

$$= \alpha I_L[S_p](x) + \alpha\Phi_p(0)e^{-\alpha x} \qquad x \geq x_p + \Delta x \tag{3.41}$$

$$= E_p(x) + \alpha\Phi_p(0)e^{-\alpha x} \tag{3.42}$$

3.5.3.2 Analysis of the Second Approach

Consider a single boundary particle, as before. In the second approach, we place an image particle centered at $-x_p$. The shape function of the image particle is

$$S_{\text{img}}(x) = \begin{cases} -1 + |x + x_p|/\Delta x & |x + x_p| < \Delta x \\ 0 & |x + x_p| \geq \Delta x \end{cases} \tag{3.43}$$

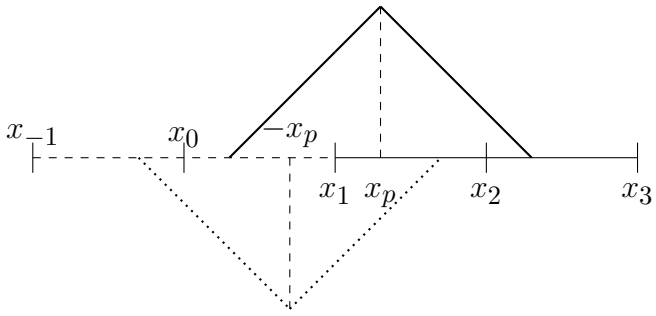


Figure 3.5: Particle shape functions $S_p(x)$ (solid) and $S_{\text{img}}(x)$ (dotted) of a boundary particle and its image particle, with ghost cells $[x_{-1}, x_0] = [-2\Delta x, -\Delta x]$ and $[x_0, x_1] = [-\Delta x, 0]$.

The potential associated with the image particle is

$$\Phi_{\text{img}}(x) = I[S_{\text{img}}](x) \tag{3.44}$$

$$= \alpha \int_{-x_p-\Delta x}^{-x_p+\Delta x} e^{-\alpha|x-y|} S_{\text{img}}(y)\, dy \tag{3.45}$$

It is easy to see through symmetry that $\Phi_{\text{img}}(0) = -\Phi_p(0)$. If we attempt to apply the usual boundary correction term, the total potential $\Phi(x) = \Phi_p(x) + \Phi_{\text{img}}(x) + Ae^{-\alpha x}$ must then satisfy $A = 0$. So the total potential is

$$\Phi(x) = \Phi_p(x) + \Phi_{\text{img}}(x) \tag{3.46}$$

and satisfies the boundary condition $\Phi(0) = 0$.

The field associated with the image particle is

$$E_{\text{img}}(x) = -\frac{d}{dx}\Phi_{\text{img}}(x) \tag{3.47}$$

$$= \alpha I_L[S_{\text{img}}](x) - \alpha I_R[S_{\text{img}}](x) \qquad |x + x_p| < \Delta x \tag{3.48}$$

$$= \alpha I_L[S_{\text{img}}](x) \qquad x > -x_p + \Delta x \tag{3.49}$$

The total field is then $E(x) = E_p(x) + E_{\text{img}}(x)$.

3.5.3.3 Comparison of the Approaches

It is clear that the difference in the potential between the approaches is

$$\delta\Phi(x) = \Phi_{\text{img}}(x) + \Phi_p(0)e^{-\alpha x} \tag{3.50}$$

and the difference in the field between the approaches is

$$\delta E(x) = E_{\text{img}}(x) - \alpha\Phi_p(0)e^{-\alpha x} \tag{3.51}$$

We have analytic formulas for these expressions we can use for the comparison. We can compute that

$$\Phi_{\text{img}}(x) = \begin{cases} -\frac{2e^{-\alpha(x+x_p)}}{\alpha\Delta x}\left(\cosh(\alpha\Delta x) - 1\right) & x \geq -x_p + \Delta x \\ -2\left[1 - (x + x_p)|/\Delta x - \frac{e^{-\alpha(x+x_p)}}{\alpha\Delta x} + \frac{e^{-\alpha\Delta x}}{\alpha\Delta x}\cosh(\alpha(x + x_p))\right] & 0 \leq x < -x_p + \Delta x \end{cases} \tag{3.52}$$

On the other hand,

$$\Phi_p(0)e^{-\alpha x} = 2\left[1 - x_p/\Delta x - \frac{e^{-\alpha x_p}}{\alpha\Delta x} + \frac{e^{-\alpha\Delta x}}{\alpha\Delta x}\cosh(\alpha x_p)\right]e^{-\alpha x} \tag{3.53}$$

From these formulas, we can show that

$$\delta\Phi(\Delta x) = 2e^{-\alpha\Delta x}\left[1 - \frac{x_p}{\Delta x} + \frac{1}{\alpha\Delta x}\sinh(\alpha(\Delta x - x_p))\right] \tag{3.54}$$

$$= O((\alpha\Delta x)^2) \tag{3.55}$$

and that

$$\delta E(\Delta x) = -\alpha\delta\Phi(\Delta x) \tag{3.56}$$

$$= O(\alpha^3\Delta x^2) \tag{3.57}$$

3.5.4 2D Dirichlet BC

If we use the first approach described above, we will extend the integration domain with ghost cells beyond the boundary to include the support of all particles. Then the boundary condition will be imposed with the usual boundary correction term.

3.6 Numerical Results

3.6.1 Electrostatic Test Problems

We first consider three standard periodic electrostatic test problems in 1D and 2D, then a 1D bounded plasma problem, the simulation of sheath formation. In the first three test

problems, electrons are loaded from a perturbed initial distribution of the form

$$f_e(x, v, t = 0) = f_e(v) \left(1 + \epsilon \sin\left(\frac{2\pi x}{L_x}\right)\right) \tag{3.58}$$

where L_x is the length of the domain, ϵ is the amplitude of perturbation, and $f_e(v)$ is the initial velocity distribution. In the 2D case, simulations are taken to be uniform in the y-direction. We normalize quantities according to the nondimensionalization presented in the Section 1.3.1. In particular, we normalize time quantities to the inverse plasma frequency ω_p^{-1}. We will consider a periodic domain with a uniform neutralizing background charge, and further we set the speed of light $c = 100$. In these problems, we see good performance even at large CFLs, since the physics is dominated by the low frequency spatial modes.

3.6.1.1 Cold Plasma Langmuir Wave

We consider a cold plasma Langmuir wave [4] with $f_e(v) = \delta(v)$. Electrons are perturbed away from a uniformly distributed, motionless state against a static, uniform neutralizing background charge distribution. The resulting separation of charge produces cold plasma oscillation. In the 1D simulation, we set $L_x = 2\pi$ and $\epsilon = 0.1$. We use a 100 cell grid and take $\Delta t = 0.1$, and we use $N_p = 3600$ particles. In the 2D simulation, we set $L_x = L_y = 2\pi$ and the perturbation strength $\epsilon = 0.1$. We use a 100×100 grid and again take $\Delta t = 0.1$, and we use $N_p = 360000$ particles. In both 1D and 2D cases, the CFL number used is $c\Delta t/\Delta x \approx 159$, much larger than what would be allowed by an explicit method. The oscillation in the potential energy is plotted and compared to the prediction of linear theory in Figure 3.6; we see that the plasma frequency is accurately reproduced.

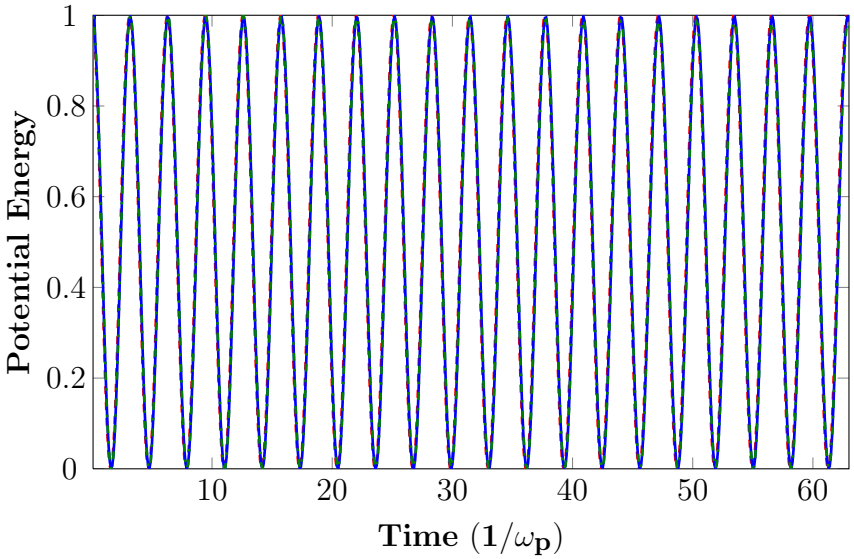


Figure 3.6: Potential energy in cold plasma oscillation. Green is the 1D numerical result, blue is the 2D numerical result, and red is the prediction of linear theory. We see the plasma frequency is accurately reproduced in our simulations.

3.6.1.2 Two Stream Instability

We consider the two stream instability with $f_e(v) = \delta(v - v_{\text{beam}}) + \delta(v + v_{\text{beam}})$. Two counterstreaming beams of electrons are perturbed away from a uniformly distributed state against a static, uniform neutralizing background charge distribution. The beams interact and "roll up" in phase space, causing some of the particles' kinetic energy to be transformed into potential energy stored in the electric field. According to the dispersion relation for the two stream instability from linear theory [4], we have

$$\omega^4 - 2\omega^2(\omega_p^2 + k^2 v_{\text{beam}}^2) + k^2 v_{\text{beam}}^2(k^2 v_{\text{beam}}^2 - 2\omega_p^2) = 0 \tag{3.59}$$

which gives the greatest growth rate, $\gamma \approx 0.3535$, for $k \approx 3.06$. We therefore scale the domain to this value of k, and take $L_x = 2\pi/3.06$. We take the beam velocity $v_{\text{beam}} = 1$ and the perturbation strength $\epsilon = 0.0005$. In our 1D simulation, we use a 100 cell grid

with $\Delta t = 0.1$, and we use $N_p = 1000$ particles. In our 2D simulation, we use a 100×100 grid with $\Delta t = 0.1$, and we use $N_p = 1000000$ particles. This results in a CFL number of $c\Delta t/\Delta x \approx 68$. We run the simulations for 1000 time steps. The growth of the $k = 3.06$ mode of the electric field is shown in Figure 3.7 for the 1D and 2D cases, and agrees with the rate from linear theory. In the nonlinear saturation stage, we see a slight discrepancy between the 1D and 2D results, probably due to the accumulation of numerical error. We also show selected phase space plots in Figure 3.8, where we see the expected "rolling up" of the two beams. Resolution is limited by the number of particles.

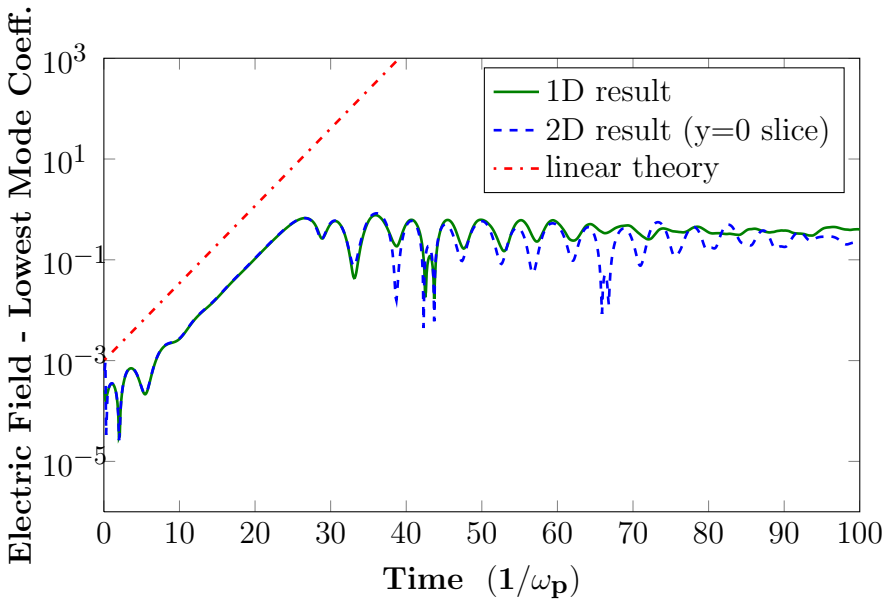
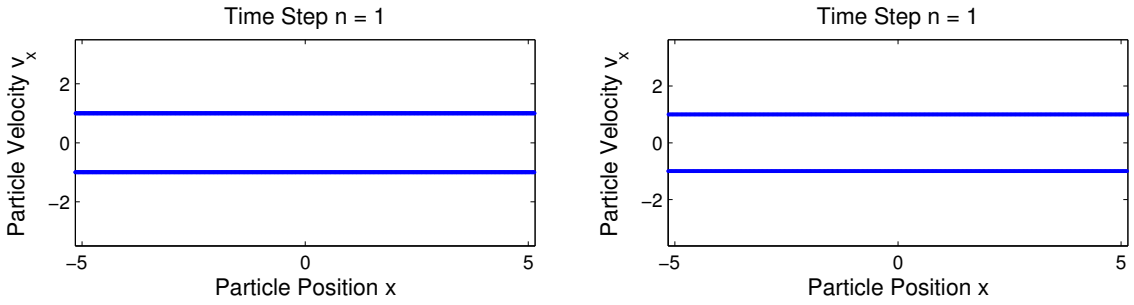


Figure 3.7: Growth of the mode with maximum growth rate in the two stream instability, corresponding to $k = 3.06$. Green is the 1D numerical result, blue is the 2D numerical result (measured along the central $y = 0$ slice), and red is the prediction of linear theory. We see the correct growth rate is reproduced in our simulations.

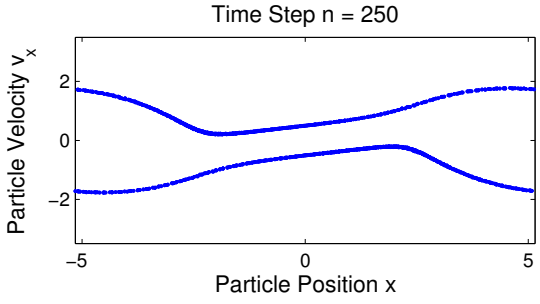
3.6.1.3 Landau Damping

We consider Landau damping of Langmuir waves in a warm plasma, with $f_e(v)$ taken to be Maxwellian. Warm electrons, following a Maxwellian velocity distribution, are perturbed

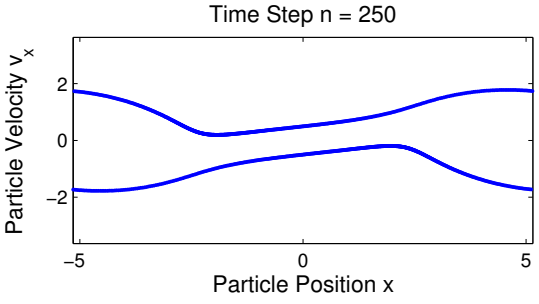


(a)

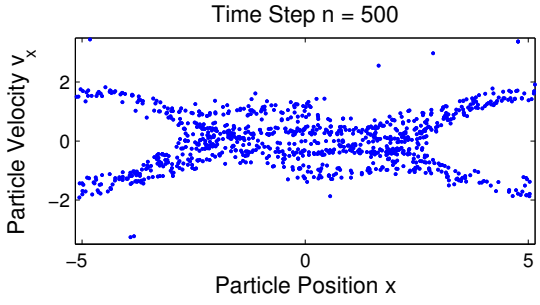
(b)



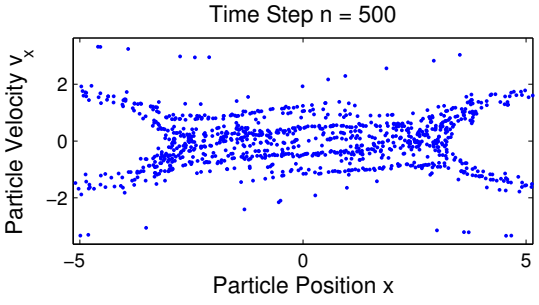
(c)



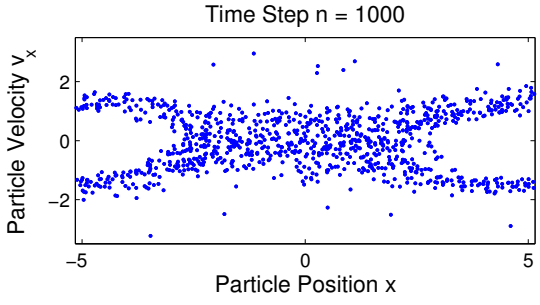
(d)



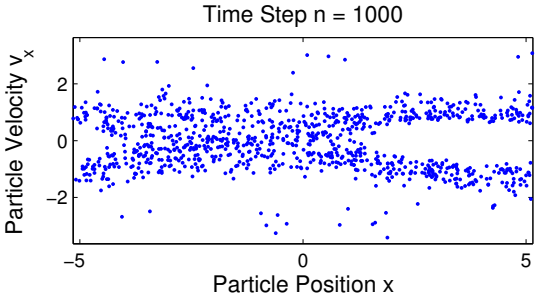
(e)



(f)



(g)



(h)

Figure 3.8: We see selected particle phase space plots for the two stream instability problem. The left column is from the 1D simulation, while the right column is from the 2D simulation, following a fixed slice of particles initialized along the line $y = -L_y/2$.

away from a unifom distribution against a static, uniform neutralizing background charge distribution. Potential energy from the electric field is transformed into kinetic energy of particles. The dispersion relation from linear theory in this case gives a decay rate of $\gamma \approx 0.154$ for the $k = 0.5$ mode [4]. We take $L_x = 4\pi$, electron thermal velocity $v_{\text{therm}} = 1$ and perturbation strength $\epsilon = 0.1$. In our 1D simulation, we use a 100 cell grid and take $\Delta t = 0.1$, and we use $N_p = 1000000$ particles. In our 2D simulation, we use a 100×100 grid and take $\Delta t = 0.1$, and we use $N_p = 9000000$ particles. We run the simulations for 300 time steps. The decay of the $k = 0.5$ mode of the electric field in the 1D and 2D simulations is shown in Figure 3.9, and agrees with the rate from linear theory. As in the two stream instability example, there is a discrepancy between the 1D and 2D results at later times, again likely due to the accumulation of numerical errors.

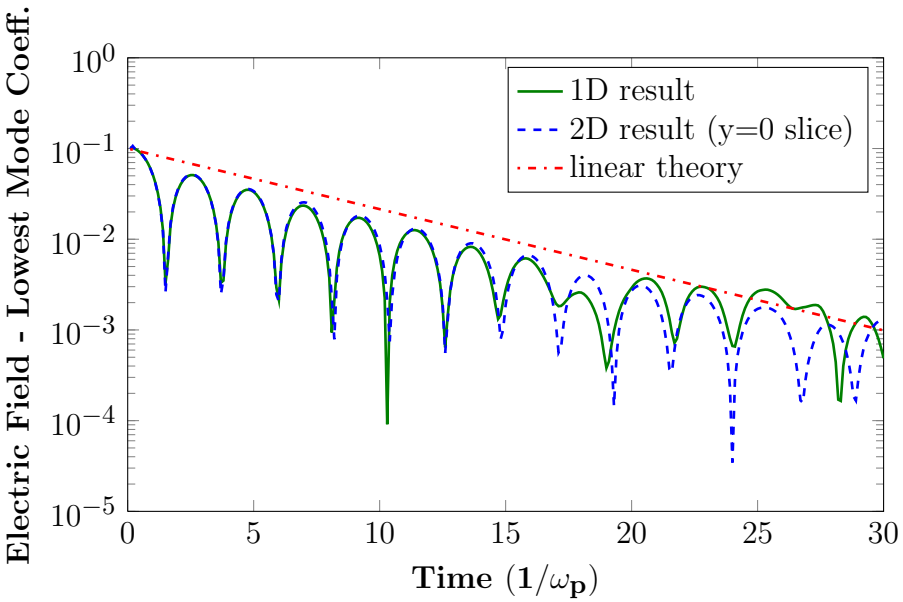


Figure 3.9: Landau damping of the lowest mode, corresponding to $k = 0.5$. Green is the 1D numerical result, blue is the 2D numerical result (measured along the central $y = 0$ slice), and red is the prediction of linear theory. We see that the correct decay rate is reproduced in our simulations

3.6.1.4 Sheath Formation in a Bounded 1D Plasma

We present the simulation of sheath formation in a bounded 1D plasma, following the model described in [46]. In contrast to the previous problems, this simulation incorporates both mobile electrons and ions. Electrons and ions are initialized from Maxwellian distributions and uniformly spatially distributed in a bounded domain. The left boundary is a symmetry plane, and so we impose Neumann boundary conditions on the potential, and reflux boundary conditions on particles, as in [46]. The right boundary is a conductor that collects charged particles. When particles hit the right boundary, they are removed from the simulation. Since electrons have a higher average velocity than ions, they have a greater flux on the collector and become depleted near the right boundary, where the difference between ion and electron densities leads to a collector sheath region, where the potential changes from the interior value to the wall value, which has the effect of repelling electrons away from the right wall. Electrons and ions are replenished by a particle source region near the left boundary, where electrons and ions are injected uniformly in the region from a Maxwellian distribution at a fixed rate per time step. We take $m_i/m_e = 100$, $T_{src,i}/T_{src,e} = 1$, and we set $v_{th,e} = 1$, and set $L_x = 20$ (in Debye lengths). We use a 100 cell grid and take $\Delta t = 0.1$, which gives a CFL number of 50. We run our simulation for 8000 time steps, up to 3.6 thermal-ion transit times. In Figure 3.10 we see the result of the simulation. In Figure 3.10a, we see the profile of the potential, which has the right qualitative features, including a collector sheath region that is several Debye lengths wide. In Figure 3.10b, we see the net electron and ion counts, along with the injection rate. The difference between the electron and ion counts reflects the difference between electron and ion densities in the collector sheath region.

3.6.2 Electromagnetic Test Problems

3.6.2.1 Bennett Pinch Problem

We present the application of our PIC method to the Bennett pinch [3], an effect related to the magnetic confinement of a beam of charged particles. A beam of charged particles induces a solenoidal magnetic field around the beam. Particles near the edge of the beam move orthogonally to these field lines at the beam drift velocity, causing the particles to be accelerated towards the center of the beam, in effect confining particles in the beam. An appropriate choice of parameters leads to a stationary steady state, uniform along the axis of the beam. A well-known magnetohydrodynamic (MHD) model of a stationary steady state gives explicit formulas for the beam density and the magnetic field [6], and provides a basis for the validation of our numerical method. Moreover, it is a first step toward applying our method to more physically interesting beam instability problems in three dimensions.

Our PIC simulation of the Bennett pinch is two-dimensional in physical space, and three-dimensional in velocity space. The particle beam is considered uniform along the axis of the beam, which we take to be the z-direction, which reduces the physical dimensions to two. Electrons drift in the z-direction with a uniform average beam drift velocity v_b, and this motion induces a confining magnetic field with only x- and y-components. A stationary ion background distribution enforces quasineutrality in the beam, with any separation of charge producing an electric field with only x- and y-components acting as a restoring force. The electrons are assumed to follow a Maxwellian distribution with thermal velocity v_{th}. We take $v_b/v_{th} = 100$ and $c/v_{th} = 1000$, where c is the speed of light. The ions are considered cold ($T_i = 0$).

Since the beam drift velocity is taken to be much larger than the (transverse) thermal

velocity, and further, the transverse velocities follow a Maxwellian distribution and so should not generate any net currents, we neglect the x- and y-components of the current density (and so also of $\mathbf{A}$). In the true solution, $\frac{\partial \Phi}{\partial z} = 0$ and $\frac{\partial A_z}{\partial t} = 0$, so we neglect $E_z = -\frac{\partial \Phi}{\partial z} - \frac{\partial A_z}{\partial t}$. Hence, we actually only solve two wave equation, one for A_z, obtaining only transverse magnetic field components, $B_x = \frac{\partial A_z}{\partial y}$ and $B_y = -\frac{\partial A_z}{\partial x}$, and one for Φ, obtaining only transverse electric field components $E_x = -\frac{\partial \Phi}{\partial x}$ and $E_y = -\frac{\partial \Phi}{\partial y}$. Thus, the Poisson equation satisfied by the scalar potential is $-\Delta\Phi = \rho/\epsilon_0$. We discretize our domain with a staggered grid, one cell of which is shown in Figure 3.11.

The scalar potential is calculated from the standard 5-point finite difference Laplacian, and satisfies the equation

$$-\frac{\Phi^{i+1,j} + \Phi^{i,j+1} + \Phi^{i-1,j} + \Phi^{i,j-1} - 4\Phi^{i,j}}{\Delta x^2} = \rho^{i,j}/\epsilon_0. \tag{3.60}$$

The electric and magnetic fields are calculated on the staggered grid by finite differences as

$$E_x^{i+1/2,j} = -\frac{\Phi^{i+1,j} - \Phi^{i,j}}{\Delta x} \tag{3.61}$$

$$E_y^{i,j+1/2} = -\frac{\Phi^{i,j+1} - \Phi^{i,j}}{\Delta x} \tag{3.62}$$

$$B_x^{i,j+1/2} = \frac{A_z^{i,j+1} - A_z^{i,j}}{\Delta x} \tag{3.63}$$

$$B_y^{i+1/2,j} = -\frac{A_z^{i+1,j} - A_z^{i,j}}{\Delta x}. \tag{3.64}$$

The electric field then satisifies the following discrete analogue of Gauss' law:

$$[\nabla \cdot \mathbf{E}]^{i,j} = \frac{E_x^{i+1/2,j} - E_x^{i-1/2,j}}{\Delta x} + \frac{E_y^{i,j+1/2} - E_y^{i,j-1/2}}{\Delta x} \tag{3.65}$$

$$= \frac{1}{\Delta x}\left(\left(-\frac{\Phi^{i+1,j} - \Phi^{i,j}}{\Delta x}\right) - \left(-\frac{\Phi^{i,j} - \Phi^{i-1,j}}{\Delta x}\right) + \right. \tag{3.66}$$

$$\left. + \left(-\frac{\Phi^{i,j+1} - \Phi^{i,j}}{\Delta x}\right) - \left(-\frac{\Phi^{i,j} - \Phi^{i,j-1}}{\Delta x}\right)\right) \tag{3.67}$$

$$= -\frac{\Phi^{i+1,j} + \Phi^{i,j+1} + \Phi^{i-1,j} + \Phi^{i,j-1} - 4\Phi^{i,j}}{\Delta x^2} \tag{3.68}$$

$$= \rho^{i,j}/\epsilon_0. \tag{3.69}$$

The magnetic field satisfies the following discrete analogue of the divergence free condition:

$$[\nabla \cdot \mathbf{B}]^{i+1/2,j+1/2} = \frac{B_x^{i+1,j+1/2} - B_x^{i,j+1/2}}{\Delta x} + \frac{B_y^{i+1/2,j+1} - B_y^{i+1/2,j}}{\Delta x} \tag{3.70}$$

$$= \frac{1}{\Delta x}\left(\left(\frac{A_z^{i+1,j+1} - A_z^{i+1,j}}{\Delta x}\right) - \left(\frac{A_z^{i,j+1} - A_z^{i,j}}{\Delta x}\right) + \right. \tag{3.71}$$

$$\left. + \left(-\frac{A_z^{i+1,j+1} - A_z^{i,j+1}}{\Delta x}\right) - \left(-\frac{A_z^{i+1,j} - A_z^{i,j}}{\Delta x}\right)\right) \tag{3.72}$$

$$= 0. \tag{3.73}$$

All computational boundaries in this problem are outflow boundaries. In order to supply the finite difference Poisson solver with suitable boundary values, the wave solver is applied with outflow boundaries conditions to evolve the wave potential Φ_W alongside the Poisson potential Φ. The boundary values from Φ_W are then supplied to the Poisson solver to use in calculating Φ. Once the wave solver reaches steady state, Φ_W and Φ differ only by 0.1%

relative error, however, the wave potential gives a discrete electric field with divergence error on the order of 10^{-3}, while the Poisson potential gives a discrete electric field with divergence error on the order of machine epsilon 10^{-16}.

Like in the other test problems, we use the diffusive version of wave solver. Particle velocities in all three directions are updated with the nonrelativistic Boris push [4]. Particles are initialized according to the MHD steady state (according to the theoretical spatial density profile and the corresponding Maxwellian distribution in velocity space) and held fixed while the field solver is stepped to an approximate steady state, after which the particle push is turned on. The simulation is run to a final time of $R_b/(2v_{th})$ (plus startup time), where R_b is an effective beam radius and v_{th} is the thermal velocity, at which time there would be substantial spreading of the beam in absence of the confinement effect. We choose R_b such that 99% of the particles in the theoretical beam are within this radius. In loading particles, the beam is cut off at radius R_b (no particles are loaded outside of this radius). The computational domain is taken to be a $4R_b \times 4R_b$ square centered on the beam axis. The computational domain is truncated with outflow boundary conditions. Particles exiting the boundary of the computational domain are reinjected into the beam to maintain constant total current. However, since most particles should be confined within the beam, such boundary crossings should be rare.

Numerical results for the Bennett pinch are given in Figure 3.12. In order to resolve large gradients near the center of the beam, we use a 500-cell by 500-cell grid and a CFL number of $c\Delta t/\Delta x = 3$ (except in Figure 3.12c as noted) and 500,000 electron particles. Final numerical solutions are shown at the final time, after 334 start up time steps and 20,834 PIC time steps. (The final time is approx. 22 plasma periods, and the diameter of the beam is approx. 280 Debye lengths.) In Figure 3.12a, we see good agreement between

the numerical electron density and MHD theory. The inset zoomed portion shows a slight discrepency at the peak of the beam, due to statistical fluctuation caused by the finite number of particles. In Figure 3.12b, we see the time histories of the potential energy, calculated as $\sum_j \frac{\Delta x \Delta y}{2}\left[\frac{1}{\mu_R}(B_{x,j}^2 + B_{y,j}^2) + \varepsilon_R(E_{x,j}^2 + E_{y,j}^2)\right]$ where the sum is over grid points j (with ε_R and μ_R defined as in Section 1.3.2), and the kinetic energy, calculated as $\sum_i \frac{1}{2} m_i (v_{x,i}^2 + v_{y,i}^2 + (v_{x,i} - v_b)^2)$ where the sum is over electron particles i. We see good energy conservation, despite the slight dissipation of the diffusive scheme. The initial spike in the potential energy is the result of transient waves, arising due to the beam turning on, and flowing out of the domain as the solution is stepped to a steady state. In Figure 3.12c, we see the result of refinement in Δt, keeping Δx fixed, showing a profile of the azimuthal magnetic field B_θ along the central $y = 0$ slice for CFL numbers of 3, 10 and 20, along with MHD theory. We observe approximate second-order convergence in Δt, as expected (a more robust convergence study is confounded by the slow convergence in particle number in PIC methods). Outside of the beam radius $R_b = 1$, there is error associated with the finite cut-off radius of the beam (the theoretical beam density decays only algebraically). In Figure 3.12d, we see the numerical error of the azimuthal magnetic field B_θ (with a CFL number of of 3) normalized by the peak value of the magnetic field, and we see that there is a geometric pattern to the numerical error, characteristic of the dimensionally-split method. In addition to this splitting error, the total error is contributed to by errors associated with the spatial quadrature and the finite differences used to calculate the magnetic field (likely contributing to the large error at the center of the beam due to large gradients there) and with the finite beam cut-off radius and the outflow boundary condition (contributing most strongly near the boundary of the computational domain). These results show that our method can indeed simulate a basic electromagnetic plasma phenomenon with a CFL number larger than what

is allowed by typical explicit schemes. The CFL number used in this problem is limited by the accuracy of the second-order wave solver. A higher-order wave solver, such as those in [11], would allow for a larger usable time step size, and will be the subject of further investigation.

3.6.2.2 Mardahl Beam Problem

We apply our method to the beam problem proposed in [37] as a diagnostic for the effects of divergence error. Particles are injected into the domain, which is a box with PEC walls, from the left wall, travel across the domain and are removed from the simulation as they hit the right wall. Parameters are chosen such that the beam should pass through the domain unperturbed.

In the Mardahl beam problem, we have currents in the plane of simulation only, and so we retain the x- and y-components of the vector potential, A_x and A_y, along with the scalar potential Φ in the model. We retain the electric field components $E_x = -\frac{\partial \Phi}{\partial x} - \frac{\partial A_x}{\partial t}$ and $E_y = -\frac{\partial \Phi}{\partial y} - \frac{\partial A_y}{\partial t}$ and the magnetic field component $B_z = \frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}$. The Poisson equation satisfied by the scalar potential is

$$-\Delta \Phi = \rho/\epsilon_0 + \frac{\partial}{\partial t}\left(\frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y}\right) \tag{3.74}$$

We discretize our domain with a staggered grid, one cell of which is shown in Figure 3.13.

Denoting by $D_{\Delta t}$ a (linear) finite difference discretization of the time derivative operator $\partial/\partial t$, the scalar potential satisfies the equation

$$-\frac{\Phi^{i+1,j}+\Phi^{i,j+1}+\Phi^{i-1,j}+\Phi^{i,j-1}-4\Phi^{i,j}}{\Delta x^2} = \rho^{i,j}/\epsilon_0 + \tag{3.75}$$

$$D_{\Delta t}\left(\frac{A_x^{i+1/2,j}-A_x^{i-1/2,j}}{\Delta x}+\frac{A_y^{i,j+1/2}-A_y^{i,j-1/2}}{\Delta x}\right). \tag{3.76}$$

The electric and magnetic fields are calculated on the staggered grid by finite differences as

$$E_x^{i+1/2,j} = -\frac{\Phi^{i+1,j}-\Phi^{i,j}}{\Delta x} - D_{\Delta t}(A_x^{i+1/2,j}) \tag{3.77}$$

$$E_y^{i,j+1/2} = -\frac{\Phi^{i,j+1}-\Phi^{i,j}}{\Delta y} - D_{\Delta t}(A_y^{i,j+1/2}) \tag{3.78}$$

$$B_z^{i+1/2,j+1/2} = \frac{A_y^{i+1,j}-A_y^{i,j}}{\Delta x} - \frac{A_x^{i,j+1}-A_x^{i,j}}{\Delta y} \tag{3.79}$$

The electric field then satisifies the following discrete analogue of Gauss' law:

$$[\nabla \cdot \mathbf{E}]^{i,j} = \frac{E_x^{i+1/2,j} - E_x^{i-1/2,j}}{\Delta x} + \frac{E_y^{i,j+1/2} - E_y^{i,j-1/2}}{\Delta y} \tag{3.80}$$

$$= \frac{1}{\Delta x}\left(\left(-\frac{\Phi^{i+1,j} - \Phi^{i,j}}{\Delta x} - D_{\Delta t}(A_x^{i+1/2,j})\right) - \left(-\frac{\Phi^{i,j} - \Phi^{i-1,j}}{\Delta x} - D_{\Delta t}(A_x^{i-1/2,j})\right) + \right. \tag{3.81}$$

$$\left. + \left(-\frac{\Phi^{i,j+1} - \Phi^{i,j}}{\Delta x} - D_{\Delta t}(A_y^{i,j+1/2})\right) - \left(-\frac{\Phi^{i,j} - \Phi^{i,j-1}}{\Delta x} - D_{\Delta t}(A_y^{i,j-1/2})\right)\right) \tag{3.82}$$

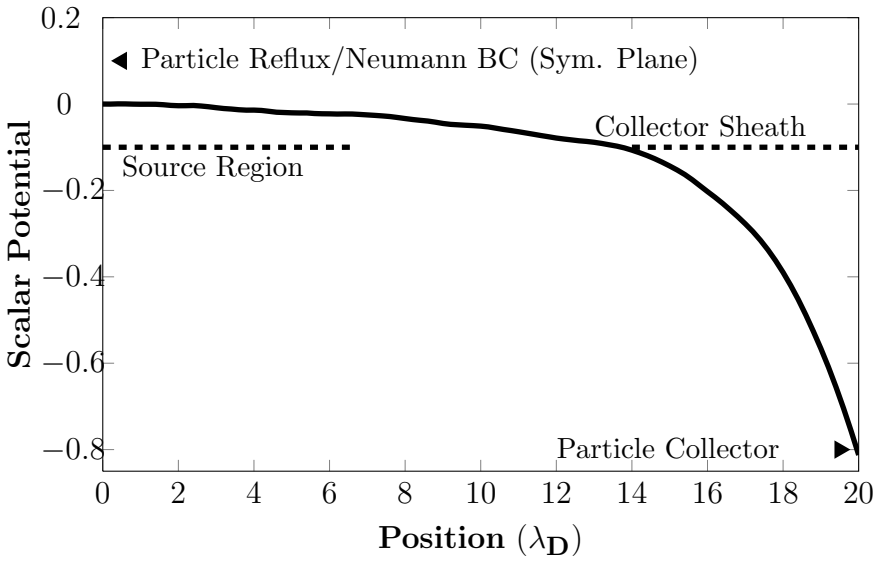
$$= -\frac{\Phi^{i+1,j} + \Phi^{i,j+1} + \Phi^{i-1,j} + \Phi^{i,j-1} - 4\Phi^{i,j}}{\Delta x^2} - \tag{3.83}$$

$$D_{\Delta t}\left(\frac{A_x^{i+1/2,j} - A_x^{i-1/2,j}}{\Delta x} + \frac{A_y^{i,j+1/2} - A_x^{i,j-1/2}}{\Delta x}\right) \tag{3.84}$$

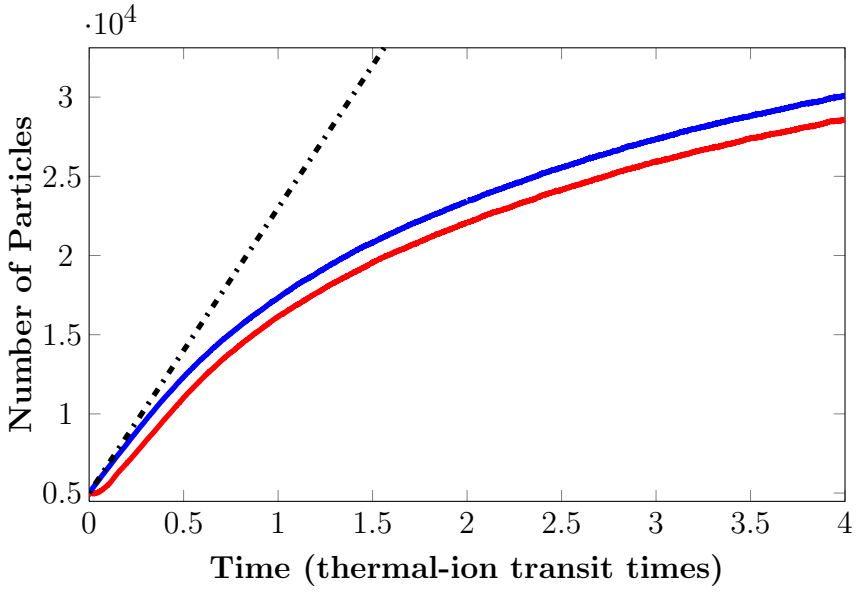
$$= \rho^{i,j}/\epsilon_0. \tag{3.85}$$

where we have used the linearity of $D_{\Delta t}$.

Numerical results for this problem are given in Figure 3.14, using a 64×64 grid and a CFL of 1. We see the expected distortion of the beam in the case when the divergence error is not controlled through the elliptic correction, whereas the beam passes through the domain unperturbed, as expected, when using the Poisson-based potential.



(a)



(b)

Figure 3.10: In 3.10a, we see the scalar potential profile at $t = 3.6$ thermal-ion transit times. In 3.10b, we see the simulation electron and ion count, the red and blue curves respectively, along with the injection rate, the black dashed line.

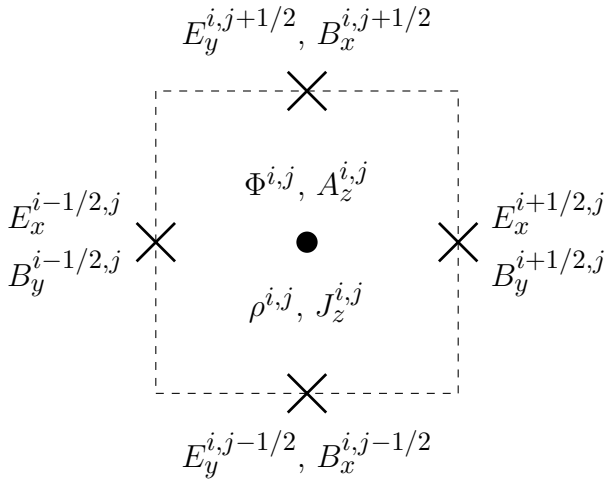


Figure 3.11: The staggered grid used for the Bennett pinch problem.

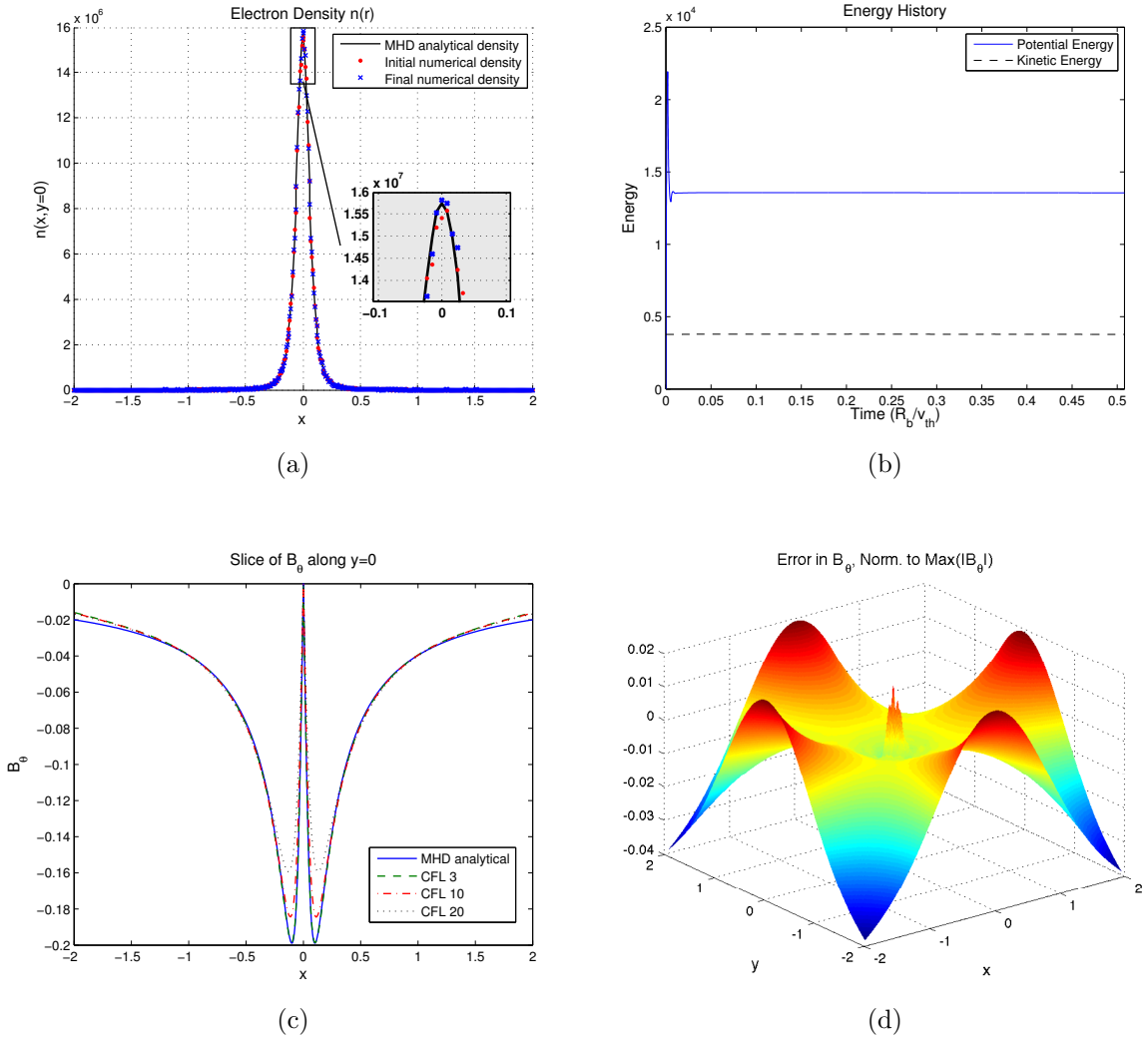


Figure 3.12: The figure shows 3.12a electron density, 3.12b and potential energies, 3.12c magnetic field at various CFL numbers, and 3.12d the relative error in the azimuthal magnetic field B_θ (normalized to the maximum value of the magnetic field). Results are with a CFL number of 3, except as noted in 3.12c. Position units are in terms of effective beam radius R_b.

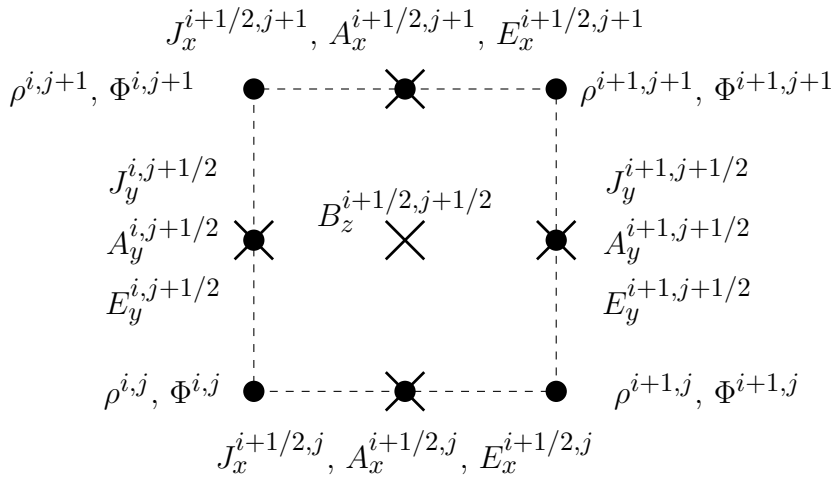
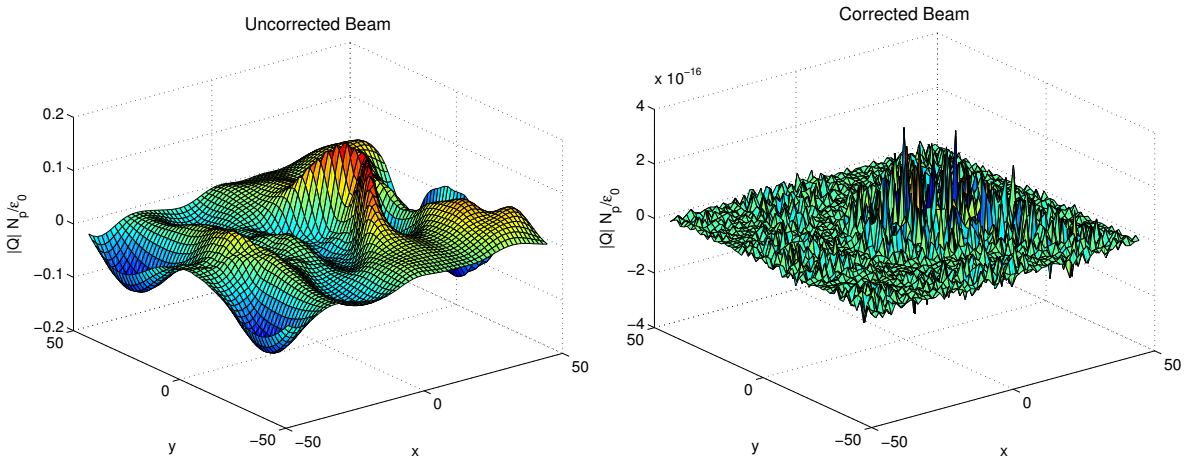
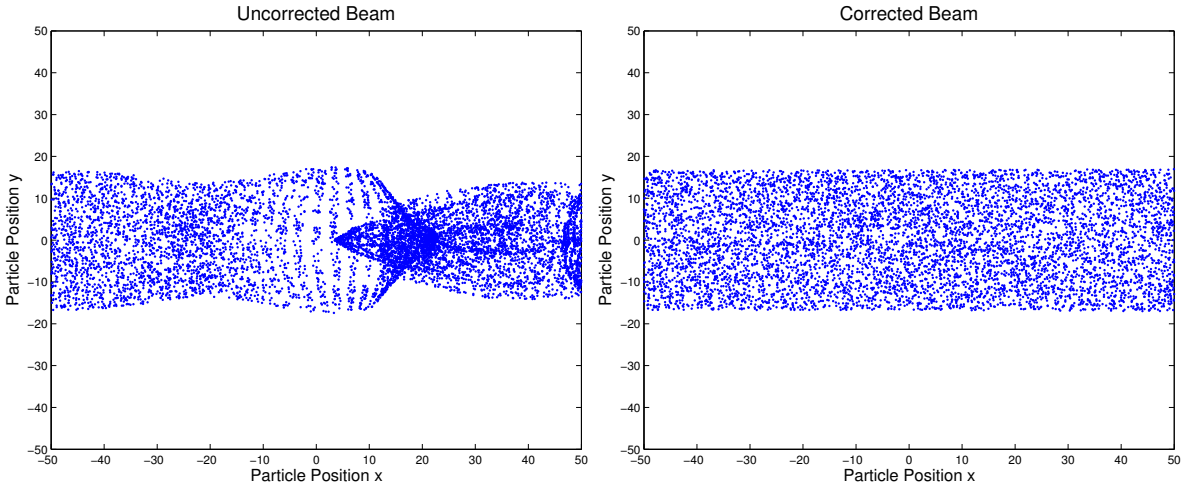


Figure 3.13: The staggered grid used for the Mardahl beam problem.



(a) (b)



(c) (d)

Figure 3.14: The figure shows the divergence error in the electric fields and the final beam distribution calculated from a wave equation potential 3.14a, 3.14c and in the Poisson equation potential 3.14b, 3.14d.

Chapter 4

Conclusion

In this work, we have described a PIC method that uses an unconditionally stable wave equation solver to eliminate the CFL restriction on the ratio of the time step size to the spatial step size, typical of explicit methods, while retaining computational cost and code complexity comparable to such explicit methods. Our numerical results show that we can apply our method to problems of plasma physics using a time step size larger than what would be allowed by a typical explicit field solver. We have seen that the usable time step size can be limited by the numerical accuracy of the method when there are large gradients (high-frequency content) in the solution. In future work, we will investigate the use of higher-order methods, such as given in [11], in our PIC method in order to increase the maximum usable time step size, and we will make use of the implicit wave solvers ability to handle complex boundary geometries without the use of a staircasing approximation. A further course of action will be to implement a boundary integral treecode (BIT) solution to solve the modified Helmholtz equations in the semi-discrete schemes, such as in [35], rather than use dimensional splitting.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] TD Arber and RGL Vann, *A critical comparison of eulerian-grid-based vlasov solvers*, Journal of computational physics **180** (2002), no. 1, 339–357.

[2] Kiyoshi Asano and Seiji Ukai, *On the vlasov-poisson limit of the vlasov-maxwell equation*, Studies in Mathematics and Its Applications **18** (1986), 369–383.

[3] Willard H Bennett, *Magnetically self-focussing streams*, Physical Review **45** (1934), no. 12, 890.

[4] Charles Kennedy Birdsall and Allan Bruce Langdon, *Plasma physics via computer simulaition*, CRC Press, 2005.

[5] George David Birkhoff, *General mean value and remainder theorems with applications to mechanical differentiation and quadrature*, Transactions of the American Mathematical Society **7** (1906), no. 1, 107–136.

[6] José A Bittencourt, *Fundamentals of plasma physics*, Springer, 2004.

[7] JU Brackbill and DW Forslund, *An implicit method for electromagnetic plasma simulation in two dimensions*, Journal of Computational Physics **46** (1982), no. 2, 271–308.

[8] Oscar P Bruno and Mark Lyon, *High-order unconditionally stable fc-ad solvers for general smooth domains i. basic elements*, Journal of Computational Physics **229** (2010), no. 6, 2009–2033.

[9] M. Causley, Y. Güçlü, E. Wolf, and A. Christlieb, *A Fast, Unconditionally stable solver for the wave equation based on the Method of Lines Transpose*, **(in preparation)**.

[10] Matthew Causley, Andrew Christlieb, Benjamin Ong, and Lee Van Groningen, *Method of lines transpose: An implicit solution to the wave equation*, Mathematics of Computation (2014).

[11] Matthew F Causley and Andrew J Christlieb, *Higher order a-stable schemes for the wave equation using a successive convolution approach*, SIAM Journal on Numerical Analysis **52** (2014), no. 1, 220–235.

[12] Roman Chapko, Rainer Kress, et al., *Rothes method for the heat equation and boundary integral equations*, Journal of Integral Equations and Applications **9** (1997), no. 1, 47–69.

[13] Guangye Chen, Luis Chacón, and Daniel C Barnes, *An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm*, Journal of Computational Physics **230** (2011), no. 18, 7018–7036.

[14] Chio-Zong Cheng and Georg Knorr, *The integration of the vlasov equation in configuration space*, Journal of Computational Physics **22** (1976), no. 3, 330–351.

[15] Yingda Cheng, Andrew J Christlieb, and Xinghui Zhong, *Energy-conserving discontinuous galerkin methods for the vlasov–maxwell system*, Journal of Computational Physics **279** (2014), 145–173.

[16] Yingda Cheng, Irene M Gamba, Fengyan Li, and Philip J Morrison, *Discontinuous galerkin methods for the vlasov–maxwell equations*, SIAM Journal on Numerical Analysis **52** (2014), no. 2, 1017–1049.

[17] Andrew J Christlieb, Robert Krasny, John P Verboncoeur, Jerold W Emhoff, and Iain D Boyd, *Grid-free plasma simulation techniques*, Plasma Science, IEEE Transactions on **34** (2006), no. 2, 149–165.

[18] Bruce I Cohen, A Bruce Langdon, and Alex Friedman, *Implicit time integration for plasma simulation*, Journal of Computational Physics **46** (1982), no. 1, 15–38.

[19] Gary Cohen, *Higher-order numerical methods for transient wave equations*, Springer, 2002.

[20] Supriyo Dey and Raj Mittra, *A locally conformal finite-difference time-domain (fdtd) algorithm for modeling three-dimensional perfectly conducting objects*, Microwave and Guided Wave Letters, IEEE **7** (1997), no. 9, 273–275.

[21] Lawrence C. Evans, *Partial differential equations*, 2nd ed ed., Graduate studies in mathematics, vol. v. 19, American Mathematical Society, Providence, R.I., 2010.

[22] Francis Filbet and Eric Sonnendrücker, *Comparison of eulerian vlasov solvers*, Computer Physics Communications **150** (2003), no. 3, 247–266.

[23] Matthew R Gibbons and Dennis W Hewett, *The darwin direct implicit particle-in-cell (dadipic) method for simulation of low frequency plasma phenomena*, Journal of Computational Physics **120** (1995), no. 2, 231–247.

[24] Dennis W Hewett and A Bruce Langdon, *Electromagnetic direct implicit plasma simulation*, Journal of Computational Physics **72** (1987), no. 1, 121–155.

[25] Roger W Hockney and James W Eastwood, *Computer simulation using particles*, CRC Press, 1988.

[26] GB Jacobs and Jan S Hesthaven, *Implicit–explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning*, Computer Physics Communications **180** (2009), no. 10, 1760–1767.

[27] GB Jacobs and JS Hesthaven, *High-order nodal discontinuous galerkin particle-in-cell method on unstructured grids*, Journal of Computational Physics **214** (2006), no. 1, 96–121.

[28] Heinz-Otto Kreiss, N Anders Petersson, and Jacob Yström, *Difference approximations of the neumann problem for the second order wave equation*, SIAM Journal on Numerical Analysis **42** (2004), no. 3, 1292–1323.

[29] Mary Catherine A Kropinski and Bryan D Quaife, *Fast integral equation methods for rothe's method applied to the isotropic heat equation*, Computers & Mathematics with Applications **61** (2011), no. 9, 2436–2446.

[30] _____, *Fast integral equation methods for the modified helmholtz equation*, Journal of Computational Physics **230** (2011), no. 2, 425–434.

[31] Lev Davidovich Landau and Evgenii Mikhailovich Lifshits, *The classical theory of fields*, vol. 2, Butterworth-Heinemann, 1975.

[32] A Bruce Langdon, *On enforcing gauss' law in electromagnetic particle-in-cell codes*, Computer Physics Communications **70** (1992), no. 3, 447–450.

[33] A Bruce Langdon, Bruce I Cohen, and Alex Friedman, *Direct implicit large time-step particle simulation of plasmas*, Journal of Computational Physics **51** (1983), no. 1, 107–138.

[34] Jongwoo Lee and Bengt Fornberg, *Some unconditionally stable time stepping methods for the 3d maxwell's equations*, Journal of Computational and Applied Mathematics **166** (2004), no. 2, 497–523.

[35] Peijun Li, Hans Johnston, and Robert Krasny, *A cartesian treecode for screened coulomb interactions*, Journal of Computational Physics **228** (2009), no. 10, 3858–3868.

[36] Mark Lyon and Oscar P Bruno, *High-order unconditionally stable fc-ad solvers for general smooth domains ii. elliptic, parabolic and hyperbolic pdes; theoretical considerations*, Journal of Computational Physics **229** (2010), no. 9, 3358–3381.

[37] PJ Mardahl and JP Verboncoeur, *Charge conservation in electromagnetic pic codes; spectral comparison of boris/dadi and langdon-marder methods*, Computer physics communications **106** (1997), no. 3, 219–229.

[38] Stefano Markidis and Giovanni Lapenta, *The energy conserving particle-in-cell method*, Journal of Computational Physics **230** (2011), no. 18, 7037–7052.

[39] Martin Masek and Paul Gibbon, *Mesh-free magnetoinductive plasma model*, Plasma Science, IEEE Transactions on **38** (2010), no. 9, 2377–2382.

[40] RL Morse and CW Nielson, *Numerical simulation of the weibel instability in one and two dimensions*, Physics of Fluids (1958-1988) **14** (1971), no. 4, 830–840.

[41] C-D Munz, Pascal Omnes, Rudolf Schneider, Eric Sonnendrücker, and Ursula Voss, *Divergence correction techniques for maxwell solvers based on a hyperbolic model*, Journal of Computational Physics **161** (2000), no. 2, 484–511.

[42] Takefumi Namiki, *3-d adi-fdtd method-unconditionally stable time-domain algorithm for solving full vector maxwell's equations*, Microwave Theory and Techniques, IEEE Transactions on **48** (2000), no. 10, 1743–1748.

[43] Donald W Peaceman and Henry H Rachford, Jr, *The numerical solution of parabolic and elliptic differential equations*, Journal of the Society for Industrial & Applied Mathematics **3** (1955), no. 1, 28–41.

[44] SV Petropavlovsky and SV Tsynkov, *Quasi-lacunae of maxwell's equations*, SIAM Journal on Applied Mathematics **71** (2011), no. 4, 1109–1122.

[45] E Pohn, Magdi Shoucri, and G Kamelander, *Eulerian vlasov codes*, Computer physics communications **166** (2005), no. 2, 81–93.

[46] RJ Procassini, CK Birdsall, and EC Morse, *A fully kinetic, self-consistent particle simulation model of the collisionless plasma–sheath region*, Physics of Fluids B: Plasma Physics (1989-1993) **2** (1990), no. 12, 3191–3205.

[47] Jing-Mei Qiu and Andrew Christlieb, *A conservative high order semi-lagrangian weno method for the vlasov equation*, Journal of Computational Physics **229** (2010), no. 4, 1130–1149.

[48] James A Rossmanith and David C Seal, *A positivity-preserving high-order semi-lagrangian discontinuous galerkin scheme for the vlasov–poisson equations*, Journal of Computational Physics **230** (2011), no. 16, 6203–6232.

[49] Erich Rothe, *Zweidimensionale parabolische randwertaufgaben als grenzfall eindimensionaler randwertaufgaben*, Mathematische Annalen **102** (1930), no. 1, 650–670.

[50] Jack Schaeffer, *The classical limit of the relativistic vlasov-maxwell system*, Communications in mathematical physics **104** (1986), no. 3, 403–421.

[51] David N Smithe, John R Cary, and Johan A Carlsson, *Divergence preservation in the adi algorithms for electromagnetics*, Journal of Computational Physics **228** (2009), no. 19, 7289–7299.

[52] Eric Sonnendrücker, Jean Roche, Pierre Bertrand, and Alain Ghizzo, *The semi-lagrangian method for the numerical resolution of the vlasov equation*, Journal of computational physics **149** (1999), no. 2, 201–220.

[53] Alen Taflove and Susan C Hagness, *Computational electrodynamics: the fdtd method*, Artech House Boston, London (2000).

[54] John P Verboncoeur, *Aliasing of electromagnetic fields in stair step boundaries*, Computer physics communications **164** (2004), no. 1, 344–352.

[55] ———, *Particle simulation of plasmas: review and advances*, Plasma Physics and Controlled Fusion **47** (2005), no. 5A, A231.

[56] Kane S Yee, *Numerical solution of initial boundary value problems involving maxwell's equations*, IEEE Trans. Antennas Propag **14** (1966), no. 3, 302–307.

[57] Fenghua Zhen, Zhizhang Chen, and Jiazong Zhang, *Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method*, Microwave Theory and Techniques, IEEE Transactions on **48** (2000), no. 9, 1550–1558.

[58] Fenghua Zheng, Zhizhang Chen, and Jiazong Zhang, *A finite-difference time-domain method without the courant stability conditions*, Microwave and Guided Wave Letters, IEEE **9** (1999), no. 11, 441–443.