

COMPUTATIONAL METHODS FOR A ONE-DIMENSIONAL PLASMA MODEL WITH TRANSPORT FIELD

DUSTIN W. BREWER[†]

Advisor: Stephen Pankavich^{†,}*

ABSTRACT. The electromagnetic behavior of a collisionless plasma is described by a system of partial differential equations known as the Vlasov-Maxwell system. From a mathematical standpoint, little is known about this physically accurate three-dimensional model, but a one-dimensional toy model of the equations can be studied much more easily. Knowledge of the dynamics of solutions to this reduced system, which computer simulation can help to determine, would be useful in predicting the behavior of solutions to the unabridged Vlasov-Maxwell system. Hence, we design, construct, and implement a novel algorithm that couples efficient finite-difference methods with a particle-in-cell code. Finally, we draw conclusions regarding their accuracy and efficiency, as well as, the behavior of solutions to the one-dimensional plasma model.

1. INTRODUCTION

A plasma is a partially or completely ionized gas. Approximately 99.99% of the visible matter in the universe exists in the state of plasma, as opposed to a solid, fluid, or a gaseous state. Matter assumes a plasma phase if the average velocity of particles in a material achieves an enormous magnitude, for example a sizable fraction of the speed of light. Hence all matter, if heated to a significantly great temperature, will reside in a plasma state (Figure 1). In terms of practical use plasmas are of great interest to the energy and aerospace industries among others, as they are used in the production of electronics, plasma engines, and lasers, as well as in the operation of fusion reactors [1]. Due to their free-flowing abundance of ions and electrons, they are great conductors of electricity and widely used in solid state physics. When a collection of charged particles is of low density or the characteristic time scales of interest are sufficiently small, the plasma is deemed to be “collisionless”, as collisions between particles become infrequent. A variety of collisionless plasmas occur in nature, including the solar wind, Van Allen radiation belts, and comet tails [9].

2000 *Mathematics Subject Classification*. Primary: 65M75, 35Q83, 76M28; Secondary: 82D10, 82C22.

Key words and phrases. kinetic theory, Vlasov-Maxwell, particle-in-cell, Lax-Wendroff.

[†]Department of Mathematics, University of Texas at Arlington, Arlington, TX 76019; email: dustin.brewer@mavs.uta.edu.

*Department of Mathematics, United States Naval Academy, Annapolis, MD 21402; email: pankavic@usna.edu, sdp@uta.edu.

This research was partially supported by a Center for Undergraduate Research in Mathematics (CURM) mini-grant funded by NSF Grant DMS-063664 and independently by the National Science Foundation under NSF Grant DMS-0908413.

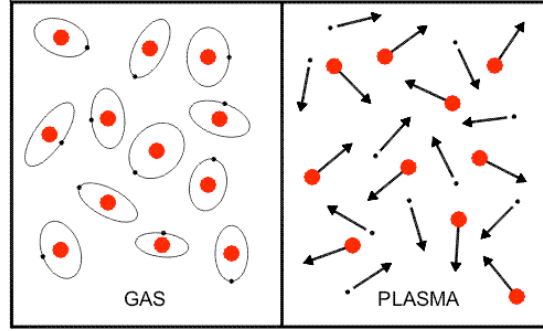


Figure 1. An example of a plasma (right) after heated significantly enough to become ionized and escape from its gaseous phase (left). After heating, sufficient energy has been applied to free electrons from their orbitals and form a plasma of ions and electrons that interact dominantly through self-consistent electric and magnetic fields.

The basic (non-relativistic) equations of collisionless plasma physics are given by a coupled system of partial differential equations (PDEs) known as the Vlasov-Maxwell system:

$$(1.1) \quad \left\{ \begin{array}{l} \partial_t f + v \cdot \nabla_x f + \left(E + \frac{v}{c} \times B \right) \cdot \nabla_v f = 0 \\ \rho(t, x) = \int f(t, x, v) dv, \quad j(t, x) = \int v f(t, x, v) dv \\ \partial_t E = c \nabla \times B - j, \quad \nabla \cdot E = \rho \\ \partial_t B = -c \nabla \times E, \quad \nabla \cdot B = 0. \end{array} \right.$$

In these equations $t \geq 0$, $x \in \mathbb{R}^3$, and $v \in \mathbb{R}^3$ represent time, space, and momentum respectively, and $f = f(t, x, v)$ is the number density - the distribution of the number of particles within a specific spatial and velocity element - of a particular species of ion in the plasma. In addition, ρ and j are the charge and current densities of the plasma, while E and B , the electric and magnetic fields, are generated by the individual charges of the ions. The constant c denotes the speed of light, and the first equation of (1.1), known as the Vlasov equation, describes the motion of ions and electrons in the system due to their free-streaming velocities and the force from the electric and magnetic fields. The distribution of ions in f gives rise to charge and current densities which, in turn, drive the fluctuations of the electric and magnetic fields. Hence, the Vlasov equation is nonlinearly coupled to Maxwell's equations which model the behavior of the fields. A complete derivation of the system and further discussion can be found in the standard references [9] and [17].

A given plasma can be composed of a variety of (say, N) species of differing charge; for example H^+ ions, electrons, and O^{2-} ions. In fact, most plasmas contain at least two species - ions and electrons. Thus, a separate Vlasov equation must be imposed for each particle density $f^\alpha(t, x, v)$ with charge e_α , indexed by $\alpha = 1, \dots, N$, and the densities of charge and current are written as sums over α . For simplicity, we will discuss and investigate monocharged plasmas only; hence $e_\alpha = 1$ and we omit the superscript α as in (1.1). Only in the verification of the computational

method and the resulting construction of a steady state will we consider a two-species plasma. In addition to ions of differing charges, a number of particle masses can be involved, but throughout we will consider the mass m of each ion in the plasma to be normalized to one.

To complete the initial-value problem on the whole space, the system (1.1) is supplemented with given, smooth initial conditions $f(0, x, v) = f_0(x, v)$, $E(0, x) = E_0(x)$, $\partial_t E(0, x) = E_1(x)$, $B(0, x) = B_0(x)$, and $\partial_t B(0, x) = B_1(x)$. Unfortunately, many questions concerning this problem remain open from a mathematical viewpoint. For example, it is unknown as to whether a smooth (continuously differentiable) particle distribution f and vector fields E and B result for all $t \geq 0$ from imposing the same smoothness properties on their initial values. While this problem is currently at the forefront of research within kinetic theory, much more is known about the system if magnetic effects are removed. That is, if one assumes that $B(t, x) \equiv 0$ or if one takes the limit as $c \rightarrow \infty$, then the resulting equations (cf. [15]) are given by the Vlasov-Poisson system which models only electrostatic effects of the plasma:

$$(1.2) \quad \begin{cases} \partial_t f + v \cdot \nabla_x f + E \cdot \nabla_v f = 0 \\ \rho(t, x) = \int f(t, x, v) dv \\ E = \nabla U, \quad \Delta U = \rho. \end{cases}$$

Now, it is well known that the field equations in (1.1) can be written as wave equations, meaning the smooth vector fields E and B satisfy the above formulation of Maxwell's equations if and only if

$$\partial_{tt} E - c^2 \Delta E = -c^2 \nabla \rho - \partial_t j \quad \text{and} \quad \partial_{tt} B - c^2 \Delta B = c \nabla \times j.$$

Thus, the field equations in the full electromagnetic problem are hyperbolic, or "wave-like" in nature. Contrastingly, the equation for the electrostatic field in (1.2) has no time derivatives and no "wave-like" behavior. Instead, its behavior is determined by a solution to Poisson's equation (i.e., $\Delta U = \rho$) which is a prototypical elliptic partial differential equation. Hence, the field equations in (1.2) tend to yield smoother solutions than those in (1.1), and this has led to the majority of the progress in the analysis of the system.

Since little is known about the three-dimensional electromagnetic problem, it makes sense to pose it in one dimension first (i.e., one component each for space and momentum) and attempt to solve the dimensionally-reduced problem. Unfortunately, this destroys the cross product terms in the Vlasov equation of (1.1) and again magnetic effects are lost. Instead, the natural one-dimensional reduction of (1.1) is the 1D Vlasov-Poisson system:

$$(1.3) \quad \begin{cases} \partial_t f + v \partial_x f + E(t, x) \partial_v f = 0, \\ \partial_x E = \int f(t, x, v) dv. \end{cases}$$

If one wishes to maintain the full electromagnetic structure, a second velocity variable is required in the formulation, giving rise to the "one-and-one-half dimensional"

Vlasov-Maxwell system [7]:

$$(1.4) \quad \begin{cases} \partial_t f + v \partial_x f + (E_1 + v_2 B) \partial_{v_1} f + (E_2 - v_1 B) \partial_{v_2} f = 0 \\ \partial_x E_1 = \int f dv, \quad \partial_t E_1 = - \int v_1 f dv \\ \partial_t (E_2 + B) + \partial_x (E_2 + B) = - \int v_2 f dv \\ \partial_t (E_2 - B) - \partial_x (E_2 - B) = - \int v_2 f dv \end{cases}$$

where $x \in \mathbb{R}$, $v \in \mathbb{R}^2$, and the speed of light c has been normalized to one. Surprisingly, the question of the C^1 regularity of f, E , and B remains open even in this case. The noticeable difference between (1.4) and (1.2) is the introduction of electric and magnetic fields E_2 and B that are solutions to transport equations, which propagate values of the fields with speed $c = 1$. Mathematically speaking, the transport equations may create problems with intersecting characteristics due to the unbounded nature of particle velocities v that may travel faster than the propagation of signals from the electric and magnetic fields. Simply put, it is possible for $v = c = 1$ and hence the Vlasov and Maxwell characteristics can intersect in space-time. This phenomena has been referred to in a previous work [6] as “resonant transport”, and its effects on the regularity of solutions are still unknown. Thus, in order to study the behavior of such a system, but keep the problem posed in a one-dimensional setting, we consider the following nonlinear system of PDEs which couples the Vlasov equation to a transport field equation:

$$(1.5) \quad \begin{cases} \partial_t f + v \partial_x f + B(t, x) \partial_v f = 0, \\ \partial_t B + \partial_x B = \int f(t, x, v) dv \end{cases}$$

with given initial conditions $f(0, x, v) = f_0(x, v)$ and $B(0, x) = B_0(x)$. This nonlinear (as B depends upon f and vice versa) system of PDEs serves as a toy model of plasma dynamics in which the effects of the field are transported in space-time. Though it does not truly represent a magnetic field, we have used B to denote the field variable of (1.5) in order to distinguish it from the field of (1.2). Additionally, we note that the model, though not obtained directly from (1.1) and hence not completely physical, does retain the aforementioned property of resonant transport. Thus, studying the behavior of (1.5) will allow us to gain intuition about the effects of resonance within (1.4) and (1.1).

In [13] the initial-value problem corresponding to (1.5) with given f_0 and B_0 was considered, and the local-in-time well-posedness of smooth solutions was shown. As this aspect of the system has been investigated and no numerical studies of (1.5) exist in the literature, we turn our attention to the design, construction, and implementation of computational methods for approximating solutions to the system. Since analytical solutions are often so difficult to derive, numerical methods are crucial to understanding the behavior of these quantities. Therefore, the main focus of our paper is to design and construct accurate and efficient computational methods to analyze solutions of (1.5). Additionally, we wish to determine (by computational means) the impact that the hyperbolic structure of the field has on the properties of solutions, as well as, the possible influence of different boundary conditions.

This paper proceeds as follows. In the next section, we will provide a detailed description of the numerical methods that we utilize in order to compute approximate solutions to (1.5). Section 3 will contain a discussion of different boundary conditions, as well as, the validation of our numerical methods using representative test runs. In Section 4, we present numerical results involving different initial particle distributions and boundary conditions. This will allow us to draw some conclusions about the utility and feasibility of these methods and the general behavior of solutions.

2. DESCRIPTION OF COMPUTATIONAL METHODS

One of the main computational tools for simulating the behavior of collisionless plasmas is the particle method (also known as a “particle-in-cell” or “PIC” method), which combines finite-difference approximations with interpolation and averaging techniques in order to track representative “superparticles” formed by a conglomeration of thousands or millions of ions. For problems that use a kinetic (rather than a continuum) description of plasma, particle methods are often utilized to approximate solutions numerically and can be less expensive than other traditional approximation techniques for partial differential equations, such as finite-difference or finite-element methods. Other advantages of PIC methods include their relative ease to construct and their cost-effective nature in simulating higher-dimensional problems. Standard references on the subject include the books [1] and [10]. Other works regarding PIC methods include [3], [4], [5], [8], and [14].

2.1. Overview. The general structure of a particle method can be described without great difficulty. As for other numerical methods, the simulation phase space, which is (x, v) for (1.5), is discretized into grids of finite length. Therefore, we must divide the spatial domain of total length L_x into N cells, each of length Δx and indexed by $i = 1, 2, \dots, N$ so that $L_x = N \cdot \Delta x$. Similarly, we divide the velocity domain of total length L_v into M cells, each of length Δv and indexed by $j = 1, 2, \dots, M$ so that $L_v = M \cdot \Delta v$. Also, the simulation of total time-length T is decomposed into P timesteps, each of length Δt and indexed by $n = 1, 2, \dots, P$, so that $T = P \cdot \Delta t$. We assume here that L_x , L_v , T , Δx , Δv , and Δt are arbitrary, positive values known prior to performing the simulation and used to determine N , M , and P explicitly. When the discretized dimension of time is added to the particle domain, we arrive at what is referred to as the particle grid or particle mesh. The left and right ends of each cell in the spatial domain at timestep n are called grid points. Hence, there are $N + 1$ total grid points that form the spatial component of the mesh with the first grid point representing the left boundary and the last grid point representing the right boundary of the domain.

Upon constructing the particle mesh, we place representative “superparticles” within the phase space domain. These particles are not individual ions in the plasma, but rather represent a group of many charged particles and are used to track particle interactions, positions, and velocities. To begin the simulation, the particles are initialized with starting positions and velocities at time $t = 0$, with representative particles placed within cells, often at grid points or at the center of each cell. We place one at the center of each cell and thus choose the total number of representative particles to be $K = N \cdot M$. Upon computing and applying the associated fields and forces at each time, the particles will be allowed to move freely through the spatial and velocity domain. Thus, the position and velocity of the k th

particle at timestep n , denoted x_k^n and v_k^n respectively, are allowed to move off of the mesh. In fact, the velocity grid is only used to initialize the particles; after time zero no velocity grid points are needed to track the particles. Contrastingly, we will see that spatial grid points will be crucial in the computation of field values and forces on particles. Once the particle distribution is known, we use it to compute macroscopic quantities such as the charge density

$$\rho(t, x) = \int f(t, x, v) dv,$$

and then the field $B(t, x)$ by approximating the solution of the field equation of (1.5), namely

$$\partial_t B + \partial_x B = \rho.$$

Finally, we calculate the force exerted by the field and “move” the particles by changing their respective positions and velocities accordingly. Since the trajectories have now been calculated for the next timestep, the process repeats and this can be continued until the stopping time $t = T$ is reached. The particle-in-cell method combines differential approximations with particle-tracking, where the positions and velocities of the particles are used to calculate the macro-quantities in each cell; hence the name “particle-in-cell”. As we will soon describe, weighting schemes play a large role in these calculations, specifically because particle charges, positions, and velocities must be recorded “at the particle”, whereas densities, fields, and forces are indexed by static grid points.

2.2. PIC Method. With the preliminaries out of the way, the particle method can be precisely described. In what follows, we will construct the method assuming that the spatial and velocity intervals of interest begin at the origin, though any interval can be discretized in the following fashion using simple translations and dilations of the grid. For much of the construction we will follow [1], which describes a similar particle method for an electrostatic plasma governed by (1.2).

Let the initial data $f_0 \in C^1([0, L_x] \times [0, L_v])$ and $B_0 \in C^1[0, L_x]$ be given. We begin the simulation by choosing $\Delta x, \Delta v > 0$ and define for every $i = 1, \dots, N$ and $j = 1, \dots, M$ with $k = i \cdot j$,

$$x_k^0 = \frac{2i-1}{2} \Delta x,$$

$$v_k^0 = \frac{2j-1}{2} \Delta v,$$

$$q_k = f_0(x_k^0, v_k^0) \Delta x \Delta v.$$

These quantities represent the initial particle positions, initial particle velocities, and the total charge of each particle included in the simulation, respectively. Notice that the charge of a particle is a conserved quantity as it does not vary over time, and hence q_k is not time-dependent. This attribute of the particle-in-cell method guarantees that the law governing the conservation of total charge is preserved throughout the simulation. Choose $\Delta t > 0$ and define $t^n = n \cdot \Delta t$ for $n \in \mathbb{N}$. The functions that represent particle positions and velocities for other times x_k^n and v_k^n will be defined later for $n \in \mathbb{N}$. Once these are known, the approximation of the

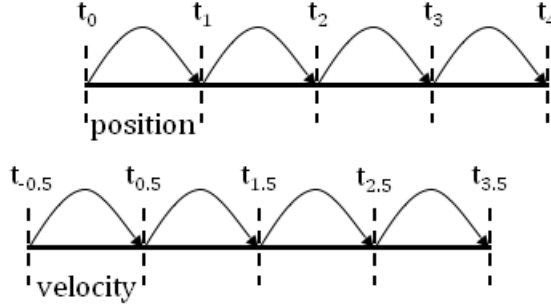


Figure 2. The leap-frog algorithm, in which values of the particle velocities and positions are advanced at opposing timesteps in order to create a more stable method.

continuous particle distribution is then given by

$$(2.1) \quad f(t^n, x, v) = \sum_{k=1}^K \frac{q_k}{\Delta x \Delta v} \hat{\delta}(x - x_k^n) \delta(v - v_k^n)$$

where δ is the Dirac delta function and $\hat{\delta}$ is the first-order weighting function defined by

$$(2.2) \quad \hat{\delta}(x) = \begin{cases} 1 - \frac{|x|}{\Delta x}, & \text{if } |x| < \Delta x \\ 0, & \text{else.} \end{cases}$$

Next, define the grid points $x_i = i \cdot \Delta x$ for $i = 0, \dots, N$. We will write

$$B_i^n = B(t^n, x_i)$$

for field values at the n th timestep and i th grid point and define the function $B^n(x)$ by linear interpolation of the grid point values B_i^n . A leap-frog scheme is utilized for the particle trajectories and first-order averaging methods are used to interpolate the field and charge density values, which we shall now discuss in greater detail.

2.3. Newton’s Equations and the Leap-Frog Algorithm. When performing a simulation, the calculations of the velocity and position of each particle at a new timestep cannot be simultaneously performed since the two quantities are not simultaneously known. In order to correct for this, the calculations must be offset from each other so that the position can be calculated from the velocity at a previous timestep and vice versa. A well-known finite-difference method of this type is commonly referred to as the “leap-frog” method. To initialize this process, we first use the force on the particles calculated from their charge and initial positions to find the velocity at a half timestep earlier; that is, we calculate $v_k^{1/2}$. Beginning with this value, we may alternatively advance the particle positions and velocities in future timesteps while maintaining this structure. Hence, the velocity of each particle will be offset from its position by a half timestep and the algorithm can calculate each value at the same time. This scheme is said to be time-centered since the values of each are determined based on the value of their counterpart in the center of each timestep (see Figure 2). Additionally, leap-frog possess other

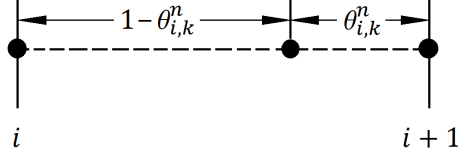


Figure 3. The particle weighting which allows the simulation to store the effects of the ions at grid points so that the field can be computed at these points. Here, $\theta_{i,k}^n$ is defined by (2.4).

nice properties such as second-order accuracy and time-reversibility. To initiate the leap-frog scheme, we first use initial field values and particle positions to define

$$v_k^{1/2} = v_k^0 - B_0(x_k^0) \cdot \frac{\Delta t}{2}.$$

In order to find the velocity and position of each particle at the next timestep, we simply apply standard, one-sided finite-difference approximations to solve Newton's equations of motion. From these equations, we know that $F = ma = m \frac{dv}{dt}$, where F represents a force, and since we have normalized the mass to one, the equation is reduced to $\frac{dv}{dt} = F$. From here, a forward-difference approximation is utilized. Thus, assume for some $n \in \mathbb{N}$ that the positions and velocities x_k^n and $v_k^{n-1/2}$ are known for all $k = 1, \dots, K$, and that B_i^{n+1} is known for all $i = 0, \dots, N$. After a linear interpolation of the field values at grid points, we obtain the function $B^{n+1}(x)$ and employ the approximation

$$\frac{v_k^{n+1/2} - v_k^{n-1/2}}{\Delta t} = B^{n+1}(x_k^n).$$

Here, the force is determined by values of the discretized field, thus in Newton's equation $F = B^{n+1}(x_k^n)$. Solving the above equation for $v_k^{n+1/2}$, the velocity at the next timestep, we find

$$v_k^{n+1/2} = v_k^{n-1/2} + \Delta t \cdot B^{n+1}(x_k^n).$$

Now that we know $v_k^{n+1/2}$, we can invoke the evolution equation for the position of particles, namely $\frac{dx}{dt} = v$. Using the same explicit approximation of the derivative, we find

$$\frac{x_k^{n+1} - x_k^n}{\Delta t} = v_k^{n+1/2}$$

and solving as before for the approximation at the next timestep, this becomes

$$x_k^{n+1} = x_k^n + \Delta t \cdot v_k^{n+1/2}.$$

At the time of initialization, the field B_k^0 is known *a priori* for all $x \in \mathbb{R}$ and hence leap-frog can be utilized without an initial field computation. This crucial portion of the process is called the "particle-mover" for obvious reasons.

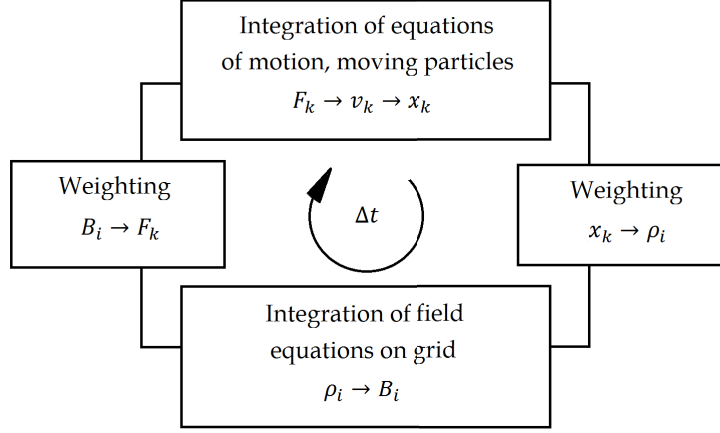


Figure 4. A flow chart of the processes involved in one timestep of the particle-in-cell method. Such an algorithm is also discussed in [1].

2.4. Weighting and Interpolation. Now that we have a value for the position and velocity of each particle, we need to calculate the field that is created by the movement of their associated charges. This is where one of the key components of the particle-in-cell method comes into play; namely weighting or averaging. Rather than computing the force imposed on each particle from the field of every other particle in the simulation, the fields will be calculated at grid points on the boundary of each cell. The first step in this process is to calculate the charge density at each grid point. If x_k^n and $v_k^{n-1/2}$ have been computed by the particle-mover, we can now compute the density at the n th timestep and hence the field at timestep $n + 1$. We define for $n \in \mathbb{N}$ and all $i = 0, \dots, N$,

$$(2.3) \quad \rho_i^n = \int f(t^n, x_i, v) dv.$$

As we cannot compute the integral in (2.3) exactly due to the discretization of the domain, we use a summation of discrete elements and a first-order weighting scheme to perform the interpolation of the value of ρ at grid points. This integral can be computed by weighting the effect that the charge of the ions (indexed by particle) has at each of their closest grid points. Thus, we divide the charge of each particle into two pieces, with one piece affecting the spatial grid point to the left of the particle and the other piece associated to the grid point located directly to the right. As we have chosen to implement a first-order weighting scheme (using $\hat{\delta}$), the level or degree to which the charge is split then depends linearly upon the proximity to either grid point. Hence, we compute

$$\rho_i^n = \sum_{k=1}^K \frac{q_k}{\Delta x} [\theta_{i,k}^n + (1 - \theta_{i-1,k}^n)]$$

where the proportion of charge $\theta_{i,k}^n \in [0, 1]$ is given by the first-order weighting function

$$(2.4) \quad \theta_{i,k}^n = \hat{\delta}(x_i - x_k^n)$$

and $\hat{\delta}$ is defined by (2.2). This allows us to average the charge over all particles and determine the density at spatial grid points (see Figure 3).

Though it is not immediately clear at present, it will be advantageous for us later to utilize the function

$$(2.5) \quad \psi(t, x) = \int (1 + v) f(t, x, v) dv.$$

Further, we will denote the value of this function at grid points by $\psi_i^n = \psi(t^n, x_i)$. Then, a similar weighting scheme can be utilized to compute these values from q_k and v_k^n , namely

$$\psi_i^n = \sum_{k=1}^K \frac{q_k}{\Delta x} (1 + v_k^n) [\theta_{i,k}^n + (1 - \theta_{i-1,k}^n)].$$

Once ρ_i^n and ψ_i^n have been computed, we define B_i^{n+1} using a finite-difference method to solve the field equation of (1.5), which depends upon knowledge of these quantities. At this stage of the process, many options are available to compute the field values. As we will discuss in greater detail within the next section, the method we have chosen utilizes an explicit, higher-order finite-difference scheme to compute B_i^{n+1} for $i = 0, \dots, N$.

Finally, after the values of the field have been determined at grid points by B_i^{n+1} , we must use them to determine the force on each particle at timestep $(n + 1)$ so that the particles can then be moved. However, particle quantities such as position and momentum need not reside at gridpoints and hence we need a similar weighting procedure to generate forces “at the particle” from field values at meshpoints. As previously mentioned, this is done by linear interpolation, or more precisely by using

$$F_k^{n+1} = \sum_{i=1}^N \left(\theta_{i,k}^n \cdot B_i^{n+1} + (1 - \theta_{i,k}^n) \cdot B_{i+1}^{n+1} \right)$$

where $\theta_{i,k}^n \in [0, 1]$ is determined by the proximity of the particle’s position to the i th spatial grid point using (2.4). The process may then continue by advancing the particle positions and velocities, x_k^n and $v_k^{n-1/2}$, to the next timestep, using the particle-mover to compute $v_k^{n+1/2}$ and then x_k^{n+1} . A visualization of the sequence of calculations for each timestep is shown in Figure 4.

2.5. Finite-Difference Methods and Lax-Wendroff. Though the particle method is utilized to approximate the number density that satisfies the Vlasov equation of (1.5), we must still compute the values of B in the field equation. To do so we utilize a traditional finite-difference method. An introductory summary of such methods can be found in [11]. Assuming the values of B_i^n are known for every $i = 0, \dots, N$ and some $n \in \mathbb{N}$, we attempt to find the values of B at timestep $(n + 1)$, namely B_i^{n+1} . Applying the most elementary of these approximations, the one-sided forward-difference method, to the space and time derivatives in the field equation of (1.5):

$$\partial_t B + \partial_x B = \rho,$$

we arrive at

$$\frac{B_i^{n+1} - B_i^n}{\Delta t} + \frac{B_{i+1}^n - B_i^n}{\Delta x} = \rho_i^n.$$

Therefore, the value of B_i^{n+1} depends upon values of B^n at exactly two other grid points, B_i^n and B_{i+1}^n . Since we are looking to find the value for B at the next timestep, we solve for B_i^{n+1} which yields

$$B_i^{n+1} = \left(1 + \frac{\Delta t}{\Delta x}\right) B_i^n - \frac{\Delta t}{\Delta x} B_{i+1}^n + \Delta t \rho_i^n.$$

The forward-difference method is easy to implement since it is explicit in time; that is, it calculates values for timestep $(n + 1)$ from known values via a finite-difference approximation at the n th timestep. Unfortunately, this method is known to be numerically unstable. Additionally, it is well known that the error term of this approximation is only first order. To find a more accurate and stable approximation scheme, we look to a higher-order method.

What we desire is a method that incorporates the ease and expense of an explicit forward-difference method, yet possesses better accuracy and stability properties. Intuitively, since the error terms are calculated from the difference of the approximations using a Taylor series, it would make sense to derive a higher-order approximation starting from the first few terms of the Taylor expansion, thereby reducing the error term. The so-called Lax-Wendroff method does just this. It begins with a Taylor series expansion over one timestep and converts each of the terms in the approximation to a different, but equivalent, form that contains no time derivatives. Using an idea similar to this, we expand the field B in time to find

$$(2.6) \quad B(t + \Delta t, x) = B(t, x) + \Delta t \cdot \partial_t B(t, x) + \frac{(\Delta t)^2}{2} \partial_{tt} B(t, x) + O(\Delta t^3).$$

From the field equation, we know

$$\partial_t B = \rho - \partial_x B.$$

Hence, we can remove time derivatives of B , and therefore the need to know values of B at later timesteps, by replacing them with spatial derivatives of B and values of ρ . For the second-order term in the expansion, we need to find the second time derivative of B , and again the field equation yields

$$\partial_{tt} B = \partial_t \rho - \partial_t (\partial_x B) = \partial_t \rho - \partial_x (\rho - \partial_x B) = \partial_t \rho - \partial_x \rho + \partial_{xx} B.$$

If we assume the particle distribution is periodic in v or possesses compact v -support, we can use the Vlasov equation of (1.5) to represent $\partial_t \rho - \partial_x \rho$ without time derivatives as

$$\begin{aligned} \partial_t \rho - \partial_x \rho &= \int (\partial_t f - \partial_x f) dv \\ &= \int (\partial_t f + v \partial_x f - (1 + v) \partial_x f) dv \\ &= - \int \partial_v (B f) dv - \int (1 + v) \partial_x f dv \\ &= - \partial_x \int (1 + v) f dv \\ &= - \partial_x \psi. \end{aligned}$$

Here $\int \partial_v f dv = 0$ due to the assumption of compact velocity support or periodicity in v . Recall ψ was defined earlier in (2.5), and we can now see that this function is useful in representing $\partial_{tt}B$ without time derivatives as

$$\partial_{tt}B = -\partial_x\psi + \partial_{xx}B.$$

Using these terms in the Taylor series expansion (2.6), we find

$$(2.7) \quad \begin{aligned} B(t + \Delta t, x) &= B(t, x) + \Delta t(\rho(t, x) - \partial_x B(t, x)) \\ &+ \frac{(\Delta t)^2}{2} (\partial_{xx}B(t, x) - \partial_x\psi(t, x)) + O(\Delta t)^3 \end{aligned}$$

Notice that the truncation of this approximation is accurate up to order $(\Delta t)^3$. From this expression, we can apply central-difference approximations to the spatial derivatives. In general, such an approximation can be derived by subtracting the Taylor series expansions for an arbitrary smooth function $\mu(x)$ about the points $x + \Delta x$ and $x - \Delta x$, with $\Delta x > 0$. Doing this yields

$$\mu(x + \Delta x) - \mu(x - \Delta x) = 2\Delta x\mu'(x) + \frac{2}{3!}(\Delta x)^3\mu'''(x) + \dots$$

or $\mu'(x) \approx \frac{\mu(x+\Delta x) - \mu(x-\Delta x)}{2\Delta x}$ with an error term of order $(\Delta x)^2$, assuming μ''' is bounded. A second-derivative central-difference approximation can also be derived from the Taylor series by adding the function values of μ at $x + \Delta x$ to those at $x - \Delta x$ to find

$$\mu(x + \Delta x) + \mu(x - \Delta x) = 2\mu(x) + (\Delta x)^2\mu''(x) + \frac{2}{4!}(\Delta x)^4\mu^{(4)}(x) + \dots$$

or $\mu''(x) \approx \frac{\mu(x+\Delta x) - 2\mu(x) + \mu(x-\Delta x)}{(\Delta x)^2}$ with an error term of order $(\Delta x)^2$, assuming $\mu^{(4)}$ is bounded. Discretizing these difference formulas and applying them to the $\partial_x B$, $\partial_x\psi$, and $\partial_{xx}B$ terms, we have

$$(2.8) \quad \begin{aligned} \partial_x B(t^n, x_i) &\approx \frac{B_{i+1}^n - B_{i-1}^n}{2\Delta x} \\ \partial_x \psi(t^n, x_i) &\approx \frac{\psi_{i+1}^n - \psi_{i-1}^n}{2\Delta x} \\ \partial_{xx} B(t^n, x_i) &\approx \frac{B_{i+1}^n - 2B_i^n + B_{i-1}^n}{(\Delta x)^2}. \end{aligned}$$

Assuming the bounds on higher derivatives above, the approximations in (2.8) are accurate up to order $(\Delta x)^2$.

Finally, we use the approximations (2.8) in (2.7) evaluated at $t = t^n$ and $x = x_i$ to find our explicit, second-order finite-difference approximation of the field

$$\begin{aligned} B_i^{n+1} &= B_i^n + \Delta t \left(\rho_i^n - \frac{B_{i+1}^n - B_{i-1}^n}{2\Delta x} \right) \\ &+ \frac{(\Delta t)^2}{2} \left(\frac{B_{i+1}^n - 2B_i^n + B_{i-1}^n}{(\Delta x)^2} - \frac{\psi_{i+1}^n - \psi_{i-1}^n}{2\Delta x} \right) \end{aligned}$$

or grouping terms,

$$(2.9) \quad \begin{aligned} B_i^{n+1} &= \left(1 - \left(\frac{\Delta t}{\Delta x} \right)^2 \right) B_i^n + \Delta t \rho_i^n - \frac{(\Delta t)^2}{4\Delta x} (\psi_{i+1}^n - \psi_{i-1}^n) \\ &- \frac{\Delta t}{2\Delta x} \left[\left(1 - \frac{\Delta t}{\Delta x} \right) B_{i+1}^n - \left(1 + \frac{\Delta t}{\Delta x} \right) B_{i-1}^n \right]. \end{aligned}$$

This is the finite-difference equation in its final form that is implemented in the program to compute the field at the next timestep, for every $i = 1, \dots, N - 1$ and $n = 0, \dots, P - 1$. The only stipulation to ensure the stability of this method is a well-known statement called the Courant-Friedrichs-Lewy (or CFL, cf. [12]) condition which states that we must choose Δt and Δx in order to satisfy $\frac{\nu \Delta t}{\Delta x} \leq 1$, where ν is the characteristic speed of the signals generated by the field $B(t, x)$. In our case, ν is the speed of light, which has been normalized to 1. Since the field equation tells us that such characteristics move to the right with speed $\nu = 1$, we need only choose $\Delta t \leq \Delta x$ to ensure stability of the method.

Hence, in our particle code, we employ a variant of the Lax-Wendroff scheme: an explicit one-step method which, when solved for at timestep $(n + 1)$, is given by (2.9). As shown above this method increases the order of accuracy (to order $(\Delta x)^2$) of the one-sided, explicit method and does not require the extra computations that are needed to utilize an implicit method. In this vein, we now turn our attention to gauging the accuracy of the entire particle-in-cell method used to model the system.

2.6. Accuracy and a Conserved Quantity. Since the particle method utilizes a number of mathematical approximations for the differential equations, it can be difficult to determine a precise order of accuracy analytically, though some error analysis of (1.2) has been performed [18], [8], [2]. Often a good heuristic for measuring accuracy is the extent to which the PIC method preserves the value of the conserved energy. We note, however, that in some cases secular changes in such conserved quantities can be observed during PIC simulations, for example due to self-heating [1], [10]. In our case, we can derive a conserved, energy-like quantity for (1.5). Though this quantity technically does not represent the energy of a physical system, we will still refer to it as the energy for (1.5). As we will see this quantity is an invariant of the system and is similar to the conserved energy of (1.1) in that it contains a velocity moment (though first-order instead of second) in the kinetic portion and the square of the field in the potential portion.

The specific method we will employ is quite simple: the simulation value of the energy is computed at each timestep and measured against its initial value to see how well it is conserved throughout. Specifically, this quantity is found by computing the Hamiltonian of the system (1.5). To derive a formula for this, we must start by making a few assumptions. Namely, we assume that the particle distribution tends to zero at infinity; i.e., $f(t, x, v) \rightarrow 0$ as $|x| \rightarrow \infty$ or $|v| \rightarrow \infty$ for every $t > 0$. In addition, we assume that a similar statement holds for the field; i.e., $B(t, x) \rightarrow 0$ as $|x| \rightarrow \infty$ for every $t > 0$. Assumptions of compact support or periodicity (which we shall implement later) for these functions would be sufficient as well.

Next, we derive the corresponding conservation law. To this end, we define

$$\mathcal{E}(t) = \frac{1}{2} \int B(t, x)^2 dx - \iint v f(t, x, v) dv dx.$$

Taking the time derivative, integrating by parts, and using (1.5) with the aforementioned boundary or support assumption, it is a straightforward calculation to show that $\mathcal{E}'(t) = 0$. Hence, $\mathcal{E}(t)$ is constant and therefore we write

$$\mathcal{E}(t) = \mathcal{E} := \frac{1}{2} \int B_0(x)^2 dx - \iint v f_0(x, v) dv dx$$

for every $t \geq 0$. Here, the first term of \mathcal{E} represents the potential energy in the system, while the second represents a corresponding kinetic energy term. Notice that each possesses an integral that must be approximated numerically. Thus, instead of tracking \mathcal{E} throughout the particle simulation, we define a discretized analogue which we deem the simulation energy, namely

$$(2.10) \quad \bar{\mathcal{E}}(t) = \frac{1}{2} \sum_{i=1}^N (B_i^n)^2 \cdot \Delta x - \sum_{k=1}^K v_k^n \cdot q_k$$

where $t = n \cdot \Delta t$. Here, v_k^n is computed by stepping forward in time by $\Delta t/2$ from the value of $v_k^{n-1/2}$ in the same way that we initially push back v_k^0 . Within the simulations of the next section, we will use this quantity to provide some measure as to the effectiveness of our computations. For instance, if we find that $\bar{\mathcal{E}}$ has changed by 3% at the stopping time, then we can generally assume that our computation of the solution is within 3% of its true value. In addition to this, we will verify the spatial accuracy of the simulations in a later section by computing a derived steady state.

3. BOUNDARY CONDITIONS AND PARTICLE METHOD TESTING

In order to construct and implement any numerical method for this problem, the initial state of the plasma (via its initial particle distribution f_0 and initial field B_0) must be known in advance. Additionally, the aforementioned finite-difference methods require boundary conditions in order to solve the corresponding equations near the boundary of the spatial grid. Thus, we now discuss the variety of options available for the boundary data.

3.1. Boundary Conditions. Since it is important for the equations to model a variety of situations, and the simulations require knowledge of the behavior of particles at the boundaries of the spatial domain (recall that the velocity domain is unconstrained throughout the simulation), we consider many different boundary conditions (BCs). The most often used of these in such simulations are:

- (1) Periodic boundary conditions
- (2) Absorbing boundary conditions
- (3) Reflective boundary conditions.

A physical picture of these conditions is displayed in Figure 5. The first variety of boundary condition models the motion of the plasma when moving along a periodic structure or within a periodic medium. When a particle in the simulation reaches the spatial boundary, its position is reinitialized to emanate from the opposing boundary wall by the exact amount from which it escaped the domain. Similarly, spatial periodicity is imposed on the field. Often the mathematical assumption of periodicity is made in order to study this phenomena in different geometries such as crystals or torii, which possess the interesting property of repeating their inherent structure.

The second condition, namely absorption of the charge at the boundary, models the situation in which particles that leave a specific spatial region become separated from the remaining plasma. When a particle in the simulation reaches the boundary, its charge is identically set to zero, thereby “absorbing” its ionic attributes at the boundary wall. Once a particle exits through one side of the mesh boundary, it no longer affects the remaining particles, thereby leading to a dissipation of the

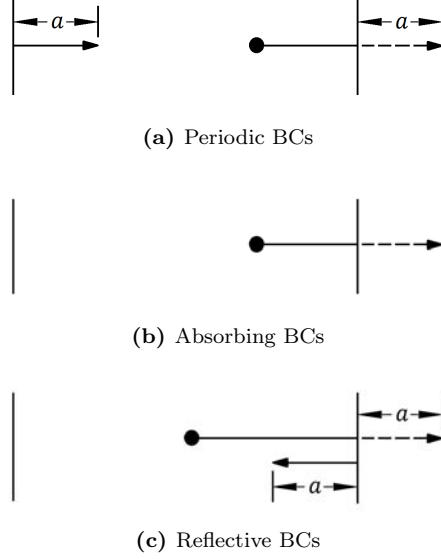


Figure 5. Graphical representations of different boundary conditions for the plasma described by (1.5). Vertical lines represent spatial boundaries, solid horizontal lines represent particle trajectories within the spatial domain, and dashed horizontal lines represent these trajectories outside of the domain. The distance a denotes the effect that the particle paths encounter upon reaching the boundary.

energy. Thus, we naturally expect $\bar{\mathcal{E}}$ to decrease significantly during any simulation that utilizes absorbing boundary conditions. Additionally, Dirichlet boundary conditions are utilized for the field.

The third condition, in which particles are reflected (and therefore not allowed to escape the simulation), is perhaps the most complicated boundary condition. In this scenario, we assume that a barrier exists at the spatial boundary that reflects particles back into the domain of interest. We are careful to note here that the type of reflection assumed in the simulation is that of a perfectly elastic collision. More specifically, it is assumed that upon colliding with the barrier at the boundary, a particle's inherent properties remain unchanged other than by altering its direction; that is, we say that no exchange of momentum takes place when the particle comes in contact with the boundary. Instead, it is perfectly reflected back into the domain with a velocity of equal magnitude but opposite direction. Again, Dirichlet boundary conditions are implemented for the field computations. Since (1.5) is a hyperbolic system, we must also choose the initial data for B to be compatible with the imposed boundary conditions. Hence, we shall choose the initial field $B_0(x)$ later to satisfy either periodicity or the Dirichlet boundary conditions, as well.

3.2. Verification and Testing. In order to test the previously described particle method using a known steady state solution, we follow the explicit example of [14]. We utilize two species of ions: positive charges with density f^+ and electrons with density f^- . In this situation the steady charge density is determined using the sum

Δx	$\sup_i B_i^n - B(x_i) $	$\sup_i B_i^n $
0.04	3.20×10^{-3}	0.858
0.02	7.99×10^{-4}	0.858
0.01	2.00×10^{-4}	0.858

Table 1. Error of the field for steady state solution. Here B_i^n is the computed field and $B(x_i)$ is the known steady state solution evaluated at spatial grid points. The error values in the table are identical over a minimum of 1000 timesteps for a spatial grid of $[-1, 1]$

of the particle distributions by

$$\rho(x) = \int (f^+(x, v) - f^-(x, v)) dv.$$

We let the electron distribution be defined as a function of the Hamiltonian

$$f^-(x, v) = \mathcal{F}\left(\frac{1}{2}|v|^2 + U(x)\right),$$

where

$$\mathcal{F}(e) = \begin{cases} -e & \text{if } e \leq 0 \\ 0 & \text{if } e > 0 \end{cases}$$

and

$$U(x) = -\frac{1}{2}(1 - x^2)^3 \chi_{[-1,1]}(x)$$

is the potential for the steady field $B(x)$. Here,

$$\chi_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

is the indicator function for the set A . From $U(x)$ the time-independent field is calculated by

$$B(x) = U'(x) = 3x(1 - x^2)^2 \chi_{[-1,1]}(x).$$

Additionally, the steady charge density can be found by calculating the derivative of the field

$$\rho(x) = B'(x) = 3(1 - x^2)(1 - 5x^2) \chi_{[-1,1]}(x).$$

Therefore, the distribution of positive charge is determined by ρ and f^- as

$$\int f^+(x, v) dv = \rho(x) + \int f^-(x, v) dv$$

whence we may take

$$f^+(x, v) = \left(3(1 - x^2)(1 - 5x^2) + \frac{2}{3}(1 - x^2)^{\frac{3}{2}}\right) \chi_{[-1,1]}(x) \delta(v)$$

where δ is the Dirac delta function.

Using these functions, the method was implemented for several choices of Δx , Δv , and Δt with periodic boundary conditions in order to ensure convergence to the correct steady state solution. Table 1 summarizes the results of these runs, listing the error found by calculating the difference between the known steady field solution and the computed field at every timestep. Figure 6 displays a comparison between

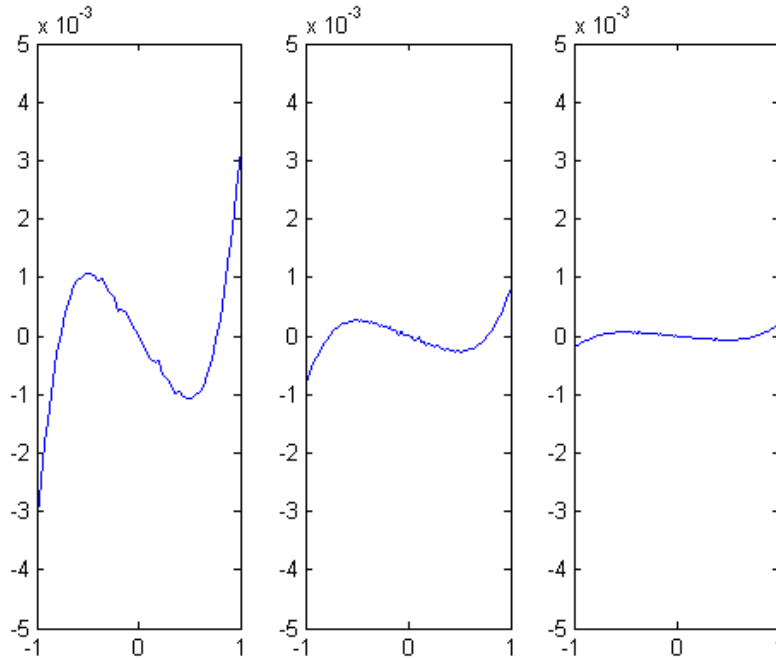


Figure 6. The difference between computed field values and the steady state field for (left) $\Delta x = 0.04$, (center) $\Delta x = 0.02$, and (right) $\Delta x = 0.01$, respectively. The horizontal axis ranges over $x \in [-1, 1]$ and the maximum values of this difference correspond to the stated error values of Table 1. As in Table 1, the computed fields do not change after a minimum of 1000 timesteps for a grid of length $L = 2$.

the computed field values and the known steady-state field. We expect that as the mesh is refined (and the value of Δx decreases), the corresponding error should also decrease at a suitable rate for each time. Evaluating each of the error values in Table 1, we see that this is the case. As the values of spacings are halved, the error in computing the field values effectively decreases by a factor of 4. Thus, the method converges at a quadratic rate, as demonstrated by the approximation of derivatives in the finite-difference method. In the next section, we will compare and contrast results from simulations with differing boundary conditions and discuss the resulting changes in behavior.

4. NUMERICAL RESULTS

Upon verifying the method, we conducted several preliminary simulations. The particle code was implemented for different choices of initial data and the boundary conditions discussed in Section 3.1. In this section, we will present results from these simulations that display the inherent differences amongst their behavior and depend strongly on the BCs of choice. Throughout the simulations, one similarity must endure - since the particles are all of like charge, the generated force is repulsion and the particle distribution must expand quickly. One specific choice of initial

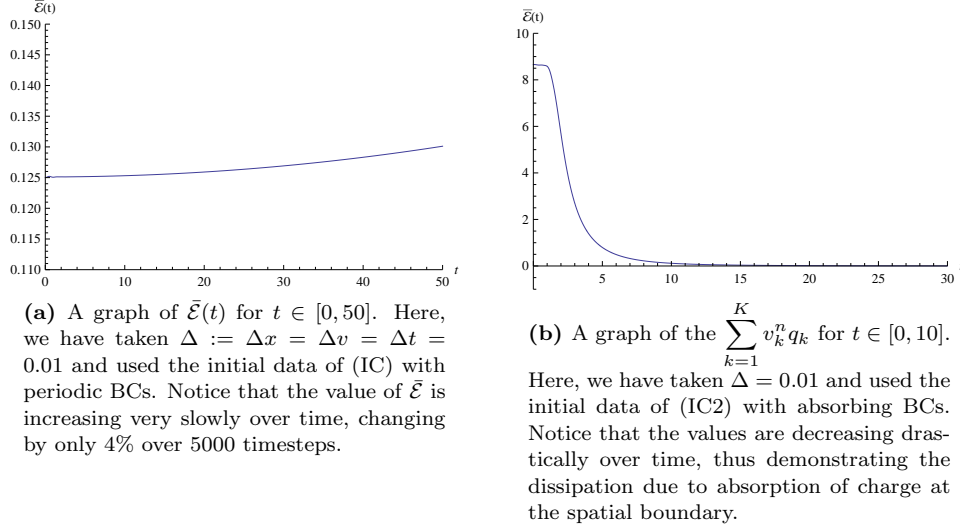


Figure 7. The change in the energy over time for periodic and absorbing BCs

data represents a smooth, compactly-supported distribution of ions in the system that quickly decays to zero. The initial field is taken to be sinusoidal in order to represent the wave-like behavior of such physical phenomena. We utilize the phase space $(x, v) \in [-1, 1] \times [-1, 1]$ and define the functions

$$(IC) \quad \begin{cases} f_0(x, v) = (1 - 4x^2)^2(1 - 4v^2)^2 \chi_{[-\frac{1}{2}, \frac{1}{2}]}(x) \chi_{[-\frac{1}{2}, \frac{1}{2}]}(v) \\ B_0(x) = \frac{1}{2} \sin(2\pi x) \chi_{[-1, 1]}(x). \end{cases}$$

With these initial conditions, the energy is easily calculated by subtracting

$$\iint v f_0(x, v) dv dx = 0$$

from the square norm of the field

$$\frac{1}{2} \int B_0(x)^2 dx = \frac{1}{2} \cdot \frac{1}{4} = 0.125.$$

Here, the kinetic portion is zero due to the even symmetry of the initial charge distribution, and we find that $\bar{\mathcal{E}}(0) = 0.125$.

Beginning with the test runs of the previous section, a simulation of (1.5) assuming initial conditions (IC) and periodic boundary conditions was conducted for $\Delta = \Delta t = \Delta x = \Delta v = 0.01$. Figure 7a shows $\bar{\mathcal{E}}$ in the computational domain for times $0 \leq t \leq 50$. As previously mentioned, the deviation of this quantity from its initial (and conserved) value is seen as a good measure of the accuracy of the method over time. In this case, the values range between 0.12500, initially, and 0.13011, at time $t = 50$. In order to measure this difference, we compute the relative change in values of $\bar{\mathcal{E}}$ over time. Table 2 summarizes the results of these runs, listing the values of the relative change as the mesh values are decreased. This

Δ	$t = 2$	$t = 4$	$t = 6$	$t = 8$	$t = 10$
0.10	1.36×10^{-2}	3.15×10^{-2}	6.10×10^{-2}	1.02×10^{-1}	1.58×10^{-1}
0.05	5.86×10^{-3}	1.04×10^{-2}	1.81×10^{-2}	2.89×10^{-2}	4.27×10^{-2}
0.025	2.62×10^{-3}	3.73×10^{-3}	5.70×10^{-3}	8.46×10^{-3}	1.20×10^{-2}
0.0125	1.22×10^{-3}	1.48×10^{-3}	1.98×10^{-3}	2.68×10^{-3}	3.58×10^{-3}

Table 2. The relative error of the simulation energy $\mathcal{R}_{\mathcal{E}}(t)$. Here, we take $\Delta = \Delta t = \Delta x = \Delta v$ and use periodic boundary conditions. The initial value $\bar{\mathcal{E}}(0) = 0.125$ is identical to \mathcal{E} , and hence $\mathcal{R}_{\mathcal{E}}(0) = 0$, for every choice of mesh or grid size.

quantity is found by calculating the difference between the known conserved value and the computed simulation value at every timestep and dividing by $\bar{\mathcal{E}}$, i.e.

$$\mathcal{R}_{\mathcal{E}}(t) = \frac{\bar{\mathcal{E}}(t) - \bar{\mathcal{E}}(0)}{\bar{\mathcal{E}}(0)}.$$

For instance, from Figure 7a we can see that the relative change for $\Delta x = \Delta v = \Delta t = 0.01$ at $t = 50$ is computed to be

$$\mathcal{R}_{\mathcal{E}}(50) = \frac{\bar{\mathcal{E}}(50) - \bar{\mathcal{E}}(0)}{\bar{\mathcal{E}}(0)} = \frac{0.13011 - 0.125}{0.125} \approx 0.0409 = 4.09\%.$$

Thus, we generally expect our computations, and hence our approximate particle distribution and field, to be within 4.1% of their actual value. Notice that the error begins to creep into the calculations more drastically at time $t = 50$, which requires 5000 timesteps to reach. If we limit the computation to a stopping time of $t = 10$ with the same data and mesh values, the approximation is seemingly closer to the actual value since we find a simulation energy of $\bar{\mathcal{E}}(10) = 0.12531$ and thus $\mathcal{R}_{\mathcal{E}}(10) = 2.48 \times 10^{-3} \approx 0.25\%$. Hence, the method conserves the simulation energy quite well (within 99.75%) even after 1000 timesteps with a relatively large mesh spacing. Graphs of the computed particle distribution $f(t, x, v)$ at times $t = 1, 2, 4, 6$ are included in Figure 8. From the figure, we can see that the mass of charge expands quickly with particles being pushed to the right (i.e. both spatial and velocity values are increased) by their initial velocities and the force of the field. Upon reaching the boundary particles begin to wrap around the periodic domain, resulting in greater interaction amongst the charges and the development of long, string-like structures.

A similar set of simulations with the identical initial particle distribution and field, but with reflective boundary conditions, was also conducted. Notice that $\bar{\mathcal{E}}$ for this system cannot be conserved as the assumptions of our computation of the Hamiltonian in the previous section are no longer valid. That is, the change in velocities of particles once they encounter the boundary of the computational domain will affect the kinetic energy-like term. Hence, additional time-dependent boundary terms must appear in this quantity and it will not be completely conserved over time. Graphs of the computed particle distribution $f(t, x, v)$ at times $t = 1, 2, 4, 6$ for the reflective case are displayed in Figure 9. Again, particles that encounter the boundary are propelled back into the computational domain. Notice,

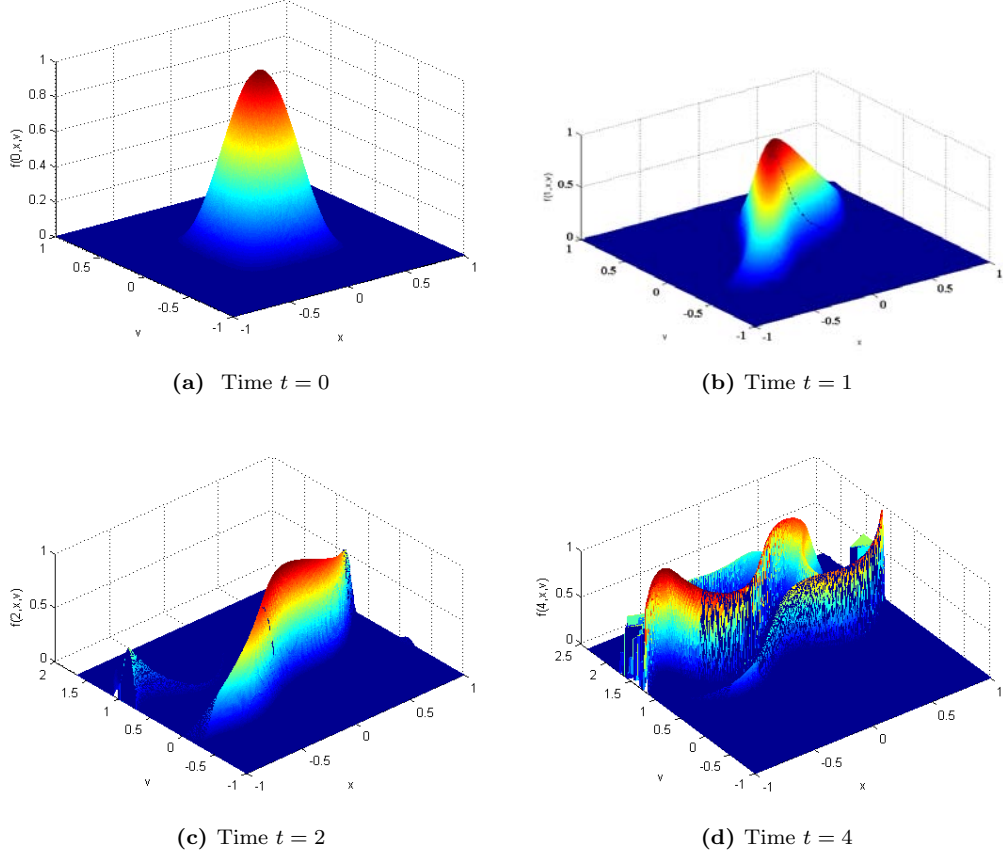


Figure 8. The particle distribution with periodic BCs using (IC) at times $t = 0, 1, 2, 4$. Notice that the values of the velocities of particles (y -axis) may increase outside of the original domain of $[-1, 1]$, but spatial values (x -axis) do not.

however, at time $t = 2$ a group of particles that attempts to leave the region is pushed back into the domain with identical spatial values but negative velocities, in contrast to those of the periodic case that maintain their velocity values but are spatially transported to the other end of the x interval.

Finally, we implemented the method to generate simulations with absorbing boundary conditions and the following initial data:

$$(IC2) \quad \begin{cases} f_0(x, v) = (1 - 4x^2)^2(1 - 4|v - 0.3|^2)^2 \chi_{[-\frac{1}{2}, \frac{1}{2}]}(x) \chi_{[-\frac{1}{2}, \frac{1}{2}]}(|v - 0.3|) \\ B_0(x) = \frac{1}{2} \sin(2\pi x) \chi_{[-1, 1]}(x). \end{cases}$$

This set of data was specifically chosen so that the particle distribution would not be symmetric in v , thereby ensuring that the kinetic portion of \mathcal{E} is not identically zero. As demonstrated in Figure 10, the particles that begin inside the domain disperse after a very short time period. Those that hit the boundary are absorbed and the

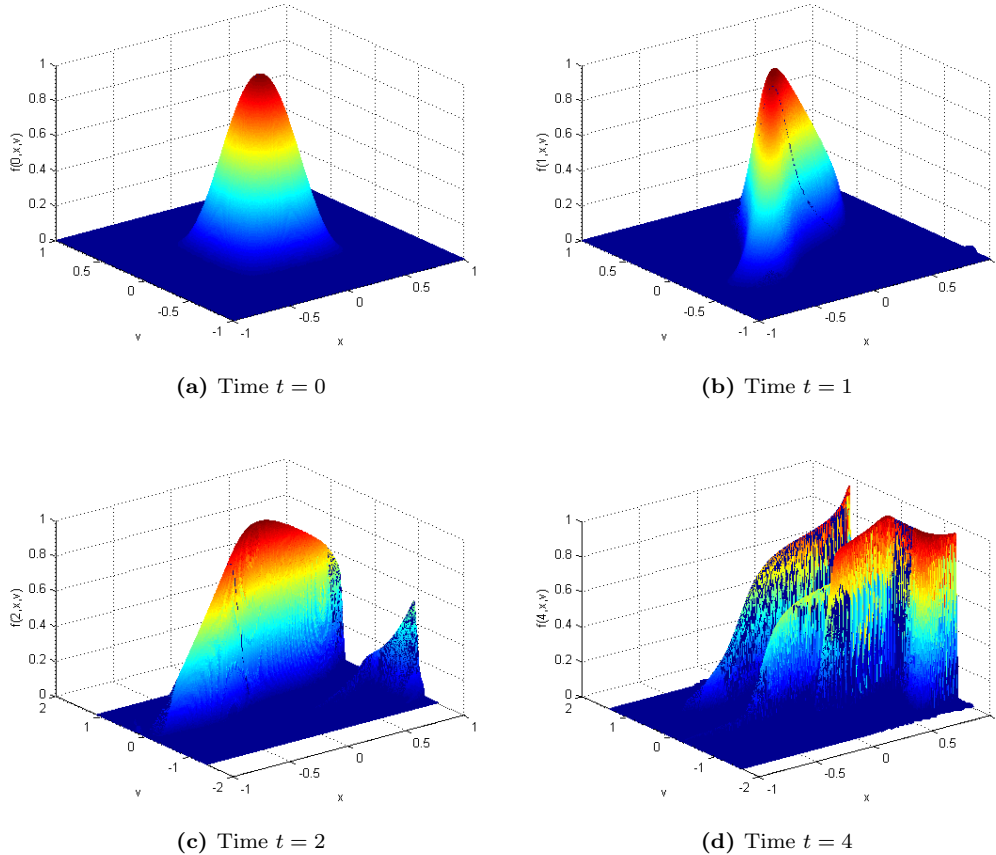


Figure 9. The particle distribution with reflective BCs using (IC2) at times $t = 0, 1, 2, 4$

charge in the system quickly dissipates. Particles that begin with small velocities and are initially positioned far from the boundary remain in the simulation but carry very little charge. Eventually, however, the force of the imposed field must push them far enough to the right that they interact with the boundary and are absorbed. After time $t = 5$, no charged particles remain, and the system ceases to change. Similar to Figure 7a, a graph of $\mathcal{E}(t)$ is displayed in Figure 7b. Notice that this quantity quickly decays as the particles are absorbed at the boundaries, and hence kinetic energy is lost in the system. As in the case of reflective boundary conditions, \mathcal{E} cannot be conserved as the absorbing boundary alters the computation of the Hamiltonian in Section 3.

Regardless of the boundary conditions utilized, the particle distribution seems to develop filaments, or localized “string-like” pieces rather quickly (see Fig 11). These structures occur along curves in phase space at which the derivatives of $f(t, x, v)$ grow large. In each case this seems to occur even before additional particle interaction arises from the imposed boundary conditions and leads one to believe that it is possible for discontinuities in phase space to occur even from smooth

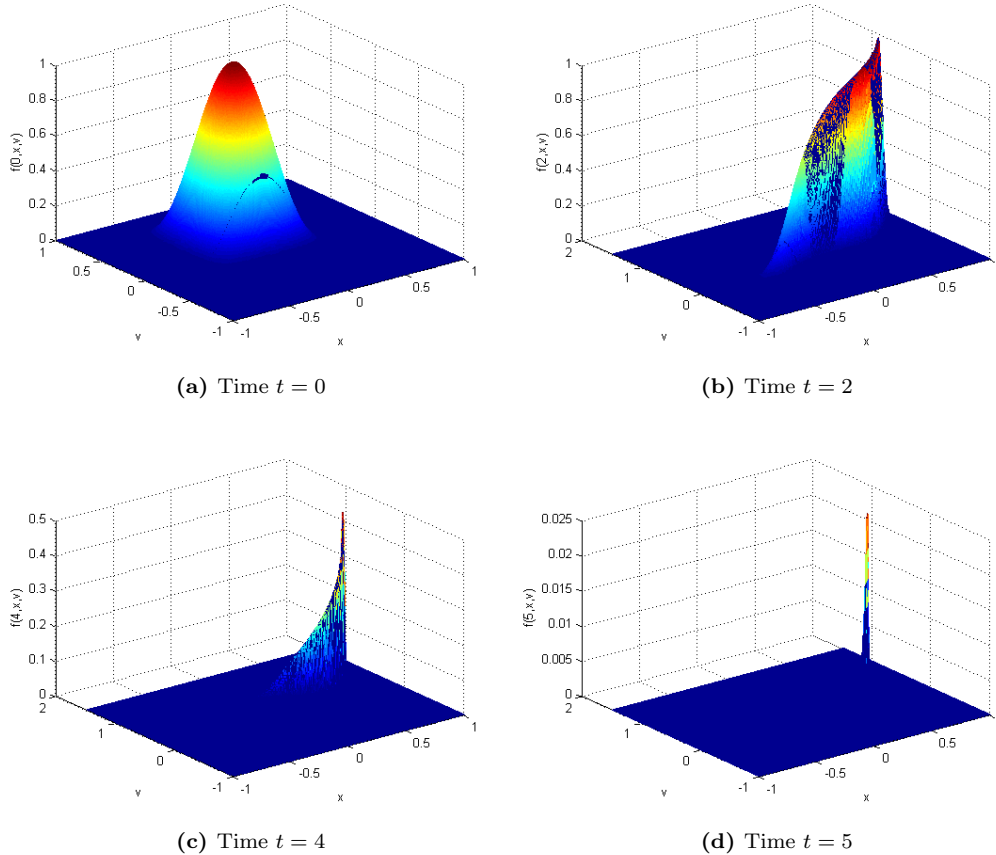


Figure 10. The particle distribution with absorbing BCs using (IC2) at times $t = 0, 2, 4, 5$

initial data such as (IC) and (IC2). As one can see from Fig 11, the two-dimensional support of the distribution function f appears to focus into a one-dimensional set. Such an occurrence of discontinuities for (1.5) would exist in sharp contrast to simulations of (1.2) in which solutions remain smooth if they begin with a smooth initial distribution (cf. [14]). A proof of this regularity result has even been obtained for the much more difficult three-dimensional analogue of (1.2) in [16]. Such a question regarding the loss of regularity of solutions in the case of a transport field, i.e. for (1.5), or for the full Vlasov-Maxwell system (1.1), remains unanswered. However, some limited progress has been made in [13] for (1.5). It is our hope that our method and results will both lead to the development of new intuition and serve as the impetus for further study of the properties of solutions to kinetic equations when coupled to a transported or, more generally an electromagnetic, field.

REFERENCES

- [1] BIRDSALL, C., AND LANGDON, A. *Plasma Physics Via Computer Simulation*. Institute of Physics, Series in Plasma Physics. Taylor and Francis, New York, NY, 1991.

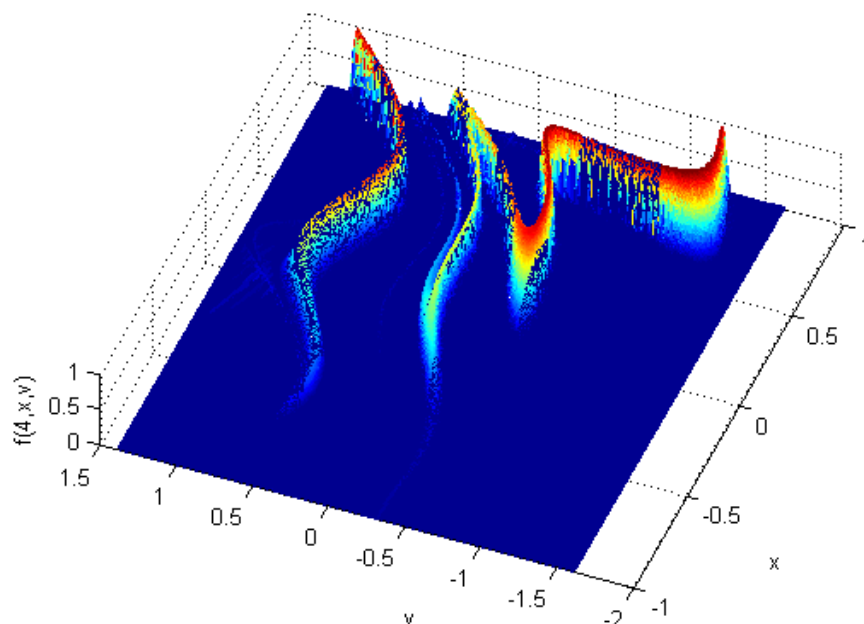


Figure 11. Another view of filaments forming from the initially smooth particle distribution (aerial view of Figure 9d). Notice that the (x, v) support of the distribution function becomes localized and the function develops sharp fronts. The two-dimensional support of the distribution function f appears to focus into a one-dimensional set.

- [2] BOSTAN, M. Analysis of a particle method for the one-dimensional Vlasov-Maxwell system. *Numer. Methods Partial Differential Equations* 25, 4 (2009), 757–782.
- [3] COTTET, G.-H., AND RAVIART, P.-A. Particle methods for the one-dimensional Vlasov-Poisson equations. *SIAM J. Numer. Anal.* 21, 1 (1984), 52–76.
- [4] COTTET, G.-H., AND RAVIART, P.-A. On particle-in-cell methods for the Vlasov-Poisson equations. *Transport Theory Statist. Phys.* 15, 1-2 (1986), 1–31.
- [5] DEGOND, P., DELUZET, F., AND NAVORET, L. An asymptotically stable particle-in-cell (PIC) scheme for collisionless plasma simulations near quasineutrality. *C. R. Math. Acad. Sci. Paris* 343, 9 (2006), 613–618.
- [6] GERARD, P., AND PALLARD, C. A mean-field toy model for resonant transport. *Kin. Rel. Mod.* 3, 2 (2010), 299–309.
- [7] GLASSEY, R., AND SCHAEFFER, J. On the “one and one-half dimensional” relativistic Vlasov-Maxwell system. *Math. Methods Appl. Sci.* 13, 2 (1990), 169–179.
- [8] GLASSEY, R., AND SCHAEFFER, J. Convergence of a particle method for the relativistic Vlasov-Maxwell system. *SIAM J. Numer. Anal.* 28, 1 (1991), 1–25.
- [9] GLASSEY, R. T. *The Cauchy problem in kinetic theory*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.
- [10] HOCKNEY, R., AND EASTWOOD, J. *Computer Simulation using Particles*. Institute of Physics, 1988.
- [11] KINCAID, D., AND CHENEY, W. *Numerical analysis*, second ed. Brooks/Cole Publishing Co., Pacific Grove, CA, 1996. Mathematics of scientific computing.
- [12] LEVEQUE, R. J. *Numerical methods for conservation laws*, second ed. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 1992.

D. BREWER

- [13] NGUYEN, C., ANDERSON, J., AND PANKAVICH, S. A one-dimensional kinetic model of plasma dynamics with a transport field. *submitted to SIAM Journal on Mathematical Analysis* (2010).
- [14] PANKAVICH, S. A particle method for a collisionless plasma with infinite mass. *submitted to Math. Comp. Sim.* (2010).
- [15] SCHAEFFER, J. The classical limit of the relativistic Vlasov-Maxwell system. *Comm. Math. Phys.* *104*, 3 (1986), 403–421.
- [16] SCHAEFFER, J. Global existence of smooth solutions to the Vlasov-Poisson system in three dimensions. *Comm. Partial Differential Equations* *16*, 8-9 (1991), 1313–1335.
- [17] VAN KAMPEN, N., AND FELDERHOF, B. *Theoretical Methods in Plasma Physics*. Wiley, New York, NY, 1967.
- [18] VICTORY, JR., H. D., TUCKER, G., AND GANGULY, K. The convergence analysis of fully discretized particle methods for solving Vlasov-Poisson systems. *SIAM J. Numer. Anal.* *28*, 4 (1991), 955–989.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF TEXAS AT ARLINGTON, 411 S NEDDERMAN DRIVE, ARLINGTON, TEXAS 76019

E-mail address: `dustin.brewer@mavs.uta.edu`