Grid-Free Plasma Simulation Techniques

Andrew J. Christlieb, *Member, IEEE*, Robert Krasny, John P. Verboncoeur, *Member, IEEE*, Jerold W. Emhoff, and Iain D. Boyd

Invited Paper

***Abstract*—A common approach to modeling kinetic problems in plasma physics is to represent the plasma as a set of Lagrangian macro-particles which interact through long-range forces. In the well-known particle-in-cell (PIC) method, the particle charges are interpolated to a mesh and the fields are obtained using a fast Poisson solver. The advantage of this approach is that the electrostatic forces can be evaluated in time $O(N \log N)$, where N is the number of macro-particles, but the scheme has difficulty resolving steep gradients and handling nonconforming domains unless a sufficiently fine mesh is used. The current work describes a grid-free alternative, the boundary integral/treecode (BIT) method. Using Green's theorem, we express the solution to Poisson's equation as the sum of a volume integral and a boundary integral which are computed using particle discretizations. The treecode replaces particle–particle interactions by particle–cluster interactions which are evaluated by Taylor expansions. In addition, the Green's function is regularized and adaptive particle insertion is implemented to maintain resolution. Like PIC, the operation count is $O(N \log N)$, but BIT avoids using a regular grid, so it can potentially resolve steep gradients and handle complex domains more efficiently. We applied BIT to several bounded plasma problems including a one-dimensional (1-D) sheath in direct current (dc) discharges, 1-D virtual cathode, cold two-stream instability, two-dimensional (2-D) planar and cylindrical ion optics, and particle dynamics in a Penning–Malmberg trap. Some comparisons of BIT and PIC were performed. These results and ongoing work will be reviewed.**

***Index Terms*—Boundary integral method, Coulomb potential, grid-free, ion optics, multipole expansion, particle-in-cell method, Penning–Malmberg trap, Poisson solver, sheath formation, treecode algorithm, two-stream instability, virtual cathode.**

I. Introduction

MANY important problems in plasma physics involve bounded domains with complex geometry, e.g., spacecraft thuster plume interactions [1], plasma sensors, and semiconductor fabrication systems [2], [3]. Plasmas also arise in unbounded domains in the space environment and give rise to internal layers (e.g., solar flares). This paper reports on the development of a novel grid-free approach for simulating these type of problems.

Nonequilibrium plasmas can be described by a variety of models depending on the physical conditions. Complete information is obtained by solving the Boltzmann equation for the distribution function $f_i(t, \mathbf{x}_i, \mathbf{v}_i)$ of each plasma species i

$$\frac{\partial f_i}{\partial t} + \mathbf{v}_i \cdot \nabla_{\mathbf{x}_i} f_i + \frac{\mathbf{F}_i}{m_i} \cdot \nabla_{\mathbf{v}_i} f_i = \left\{ \frac{\partial f_i}{\partial t} \right\}_c \tag{1}$$

where $\mathbf{x}_i$ is position, $\mathbf{v}_i$ is velocity, $\mathbf{F}_i$ is the applied/self force, m_i is mass, and $\{\partial f_i/\partial t\}_c$ is a convolution integral describing particle–particle collisions [4]. Because (1) is six-dimensional plus time, it is not computationally feasible to solve it for each species. However, to make accurate predictions about the behavior of the system, a kinetic description is often necessary for at least one of the species. For example, the tail of the electron distribution function (EDF) plays a dominant role in the ionization process. Often the energetic tail is far from a Maxwellian distribution, hence, a kinetic model is necessary to accurately describe the EDF and thereby correctly describe the ionization process [5], [6].

Many options are available for the numerical solution of the Boltzmann equation (1) [7]–[15]. In the case of collisionless plasma, (1) reduces to the Vlasov–Poisson (VP) system. One well-accepted approach is solve the VP system using a Lagrangian framework (LF). In an LF, $f(t, \mathbf{x}, \mathbf{v})$ is represented as a collection of macro-particles in phase space [15]. The particles are given an initial distribution by sampling phase space at $t = 0$ and the evolution of the system is obtained by solving Newton's equations for each particle

$$\dot{\mathbf{x}}_j = \mathbf{v}_j, \quad \dot{\mathbf{v}}_j = \frac{\mathbf{F}_j}{m_j}. \tag{2}$$

In this approach, collisions can be accounted for by coupling this method to a Monte Carlo technique [15].

The electrostatic force in three dimensions is given by

$$\mathbf{F}_j = \sum_{\substack{k=0 \\ k \neq j}}^{N} \frac{q_k}{|\mathbf{x}_j - \mathbf{x}_k|} \tag{3}$$

for $j = 1, \ldots, N$. Direct summation requires $O(N^2)$ operations, but $\mathbf{F}_j$ can be evaluated in $O(N \log N)$ operations by the

Manuscript received September 25, 2005; revised January 24, 2006. This work was supported in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-05-1-0199, in part by the National Science Foundation under Grant DMS-0510162, in part by the Air Force Research Laboratory (AFRL)-Edwards Contract F013829 and Contract F013830, and in part by AFRL-Kirtland.

A. J. Christlieb and R. Krasny are with the Mathematics Department, University of Michigan, Ann Arbor, MI 48109-1043 USA (e-mail: christli@umich.edu).

J. P. Verboncoeur is with the Nuclear Engineering Department, University of California, Berkeley, CA 94720-1770 USA.

J. W. Emhoff and I. D. Boyd are with the Aerospace Engineering Department, University of Michigan, Ann Arbor, MI 48109-2140 USA.

Digital Object Identifier 10.1109/TPS.2006.871104

0093-3813/$20.00 © 2006 IEEE

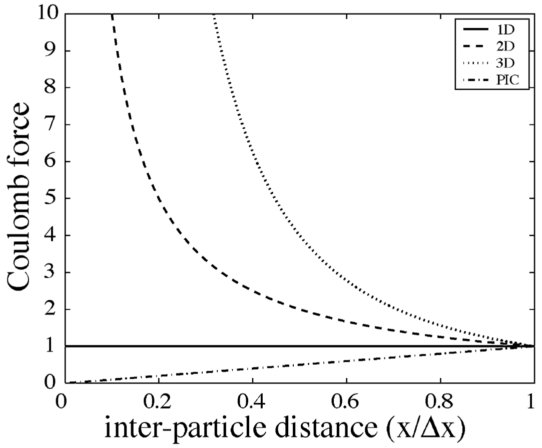


Fig. 1. Coulomb force as a function of distance between two charged particles within a mesh cell; one, two, and three dimensions compared with linear weighting PIC [15].

particle-in-cell (PIC) method. PIC involves four steps: particles are first interpolated to a mesh to give a density, a fast Poisson solver is used to compute the fields, the force is interpolated back to the particle locations, and finally the new $\mathbf{x}_j$ and $\mathbf{v}_j$ are computed [7], [8].

PIC has optimal performance when the underlying mesh is uniform. This is because on a nonuniform mesh, standard finite difference stencils become lower order accurate and more elaborate stencils are required to restore higher order accuracy. Other mesh-based effects include difficulty in resolving internal layers [16] and an incorrect description of interparticle forces within a mesh cell [15]. Moreover, not resolving interparticle forces can have a significant impact on systems where Coulomb collisions are important, as in high-density plasmas [17]. To illustrate this, Fig. 1 plots the exact Coulomb force as a function of distance between two charged particles within a mesh cell, in one, two, and three dimensions, compared to the result from a linear weighting PIC scheme. The PIC results differ considerably from the exact force. This can lead to spurious plasma heating, but the opposite occurs in a damped system, i.e., the plasma can be erroneously cooled [7]. To see why this is a problem, consider a case in which scattering due to small angle electrostatic collisions is important. In a standard PIC scheme, the interparticle force falls to zero inside a mesh cell (Fig. 1), which is unphysical. The small angle scatter is then computed incorrectly, leading to unphysical cooling of some particles.

Much effort has gone into overcoming these difficulties. Finite element PIC addresses the issues of complex domain geometry and boundary layers, but there are still problems with internal layers and resolving interparticle forces within a mesh cell [18], [19]. The particle–particle/particle-mesh (P3M) method was developed to address the issue of interparticle forces [8], but the cost approaches $O(N^2)$. Another method for handling short-range Coulomb forces sums the collisions into a single large angle event [20], [21], but resolving sharp gradients remains an issue. The current state of the art for plasma simulations combines PIC with adaptive mesh refinement (AMR) [22]–[25], but despite the success of AMR-PIC, some limitations persist. For time-dependent simulations in complex geometry, AMR requires elaborate meshing tools to generate the locally refined meshes. Moreover, in regions where coarse and fine meshes intersect, the change in resolution can lead to spurious features. For hyperbolic transport equations, this is a local defect that can be corrected, but for elliptic field equations the difficulty is nonlocal and a small error in one location can affect the entire solution. Solutions to this problem involve filtering [25], but this can introduce other artifacts.

Here, we describe an alternative grid-free approach for plasma simulations in complex domains, the boundary integral/treecode (BIT) method. Treecodes were introduced as an efficient method for computing gravitational forces in systems of point masses [26]. The key idea is to replace the particle–particle interactions by particle–cluster interactions. In the simplest case of a monopole approximation, a cluster of particles is replaced by a single particle carrying the total mass of the cluster. This reduces the work needed to compute long-range forces from $O(N^2)$ to $O(N \log N)$. To gain efficiency, the fast multipole method (FMM) uses higher order multipole expansions and cluster-cluster interactions [27], [28], while other approaches use variable order expansions and locally adapted clusters [29].

Treecodes and the FMM have been used extensively in astrophysics [30], fluid dynamics [31], and molecular dynamics [32]. Treecode simulations have also been conducted for dense plasmas in simple domains with free-space or periodic boundary conditions [17], [33], [34]. However, realistic plasma systems generally require Dirichlet, Neumann, or mixed boundary conditions on a domain with possibly complex geometry. In this case, a boundary integral method can be used to correct the free-space potential and account for the presence of a boundary while retaining the essential grid-free nature of the scheme [35], [36]. Boundary integral methods use Green's formula and recast the original partial differential equation, say Poisson's equation, as an integral equation [38], which reduces the dimensionality of the problem by one.

The grid-free BIT approach is an attractive alternative to mesh-based PIC methods. One advantage of a BIT approach is that it eliminates errors introduced by interpolation to the mesh. Moreover, BIT can naturally handle systems with nonuniform particle density and complex geometry. We have applied BIT to simulate one-dimensional (1-D) sheath formation in direct current (dc) discharges, 1-D formation of a virtual cathode, two-stream instability, two-dimensional (2-D) planar and cylindrical ion optics, and charged particle dynamics in a Penning–Malmberg trap. Here, we present preliminary timing comparisons between BIT and multigrid PIC for 2-D planar geometry. In the following sections, we outline the BIT method, discuss its advantages and disadvantages, present a summary of these applications, and conclude with a discussion of future prospects.

II. BIT

The BIT scheme combines a grid-free field solver with a boundary integral method, and we may view the approach from either of two perspectives. First, BIT is complementary to PIC in the sense that the mesh-based field solver in PIC is replaced by a grid-free field solver in BIT. The second perspective is to view the Lagrangian macro-particles as marker points on phase space curves. As the curves evolve, the marker points spread out, and new points are inserted along the curves to maintain resolution. Point insertion is performed by interpolation with respect to a Lagrangian parameter along phase space curves [29], [37]. To accommodate point insertion, the present formulation of the field equation is more general than in our previous work [35], [36]. In the following sections, we derive a Green's function method for Poisson's equation, convert the volume integral to a sum over Lagrangian macro-particles, show how multipole expansions can be used to evaluate the sum efficiently, and demonstrate how to handle boundary conditions.

A. Potential Function

Poisson's equation is

$$\nabla^2\Phi(\mathbf{x}) = -\frac{\rho(\mathbf{x})}{\epsilon_o}, \quad a\Phi(\mathbf{z}) + b\partial_{\mathbf{n}}\Phi(\mathbf{z}) = \gamma(\mathbf{z}) \tag{4}$$

where $\Phi(\mathbf{x})$ is the electrostatic potential at $\mathbf{x} \in \Omega \setminus \partial\Omega$, $\rho(\mathbf{x})$ is the charge density, ϵ_o is the permittivity of free space, $\partial_{\mathbf{n}}$ is the normal derivative, $\mathbf{z} \in \partial\Omega$, and the constants a, b determine the type of boundary condition (Dirichlet, Neumann, or mixed). To solve the Poisson equation, we employ Green's formula

$$\int_{z_0}^{z_1} \left(h\frac{d^2}{dx^2}g - g\frac{d^2}{dx^2}h \right) dx = \left(h\frac{d}{dx}g - g\frac{d}{dx}h \right)\Bigg|_{z_0}^{z_1}$$

and

$$\int_{\Omega} (h\nabla^2 g - g\nabla^2 h) d\mathbf{x} = \oint_{\partial\Omega} (h\nabla g - g\nabla h)\cdot \mathbf{n}\, dS$$

in one and two dimensions, respectively. Note that the bold $d\mathbf{x}$ implies a 2-D integral over the coordinate $\mathbf{x} \in \Omega$. We also use the free-space Green's function for the Laplace operator, $G(\mathbf{x}|\mathbf{y})$, which satisfies

$$\nabla^2_{\mathbf{y}} G(\mathbf{x}|\mathbf{y}) = C\delta\left(||\mathbf{x}-\mathbf{y}||_2\right), \quad [\partial_{\mathbf{n}} G]_{\mathbf{y}} = C, \quad [G]_{\mathbf{y}} = 0$$

where $||\mathbf{x}-\mathbf{y}||_2 = \sqrt{\sum_i (x_i - y_i)^2}$, $[\cdot]_{\mathbf{y}}$ denotes a jump at $\mathbf{y}$, and C depends on the dimension. The free-space Green's functions in one and two dimensions are

$$G(x|y) = -\frac{|x-y|}{2} \tag{5}$$

and

$$G(\mathbf{x}|\mathbf{y}) = -\frac{\ln\left(||\mathbf{x}-\mathbf{y}||_2\right)}{2\pi} \tag{6}$$

respectively. Letting $h = \Phi$ be the solution of Poisson's equation (4) and $g = G$ the free-space Green's function, we express the potential as

$$\Phi(y) = \int_{z_0}^{z_1} -G(x|y)\frac{\rho(x)}{\epsilon_o}dx + \left[\Phi(x)\frac{d}{dx}G(x|y) - G(x|y)\frac{d}{dx}\Phi(x)\right]_{x=z_0}^{z_1} \tag{7}$$

and

$$\Phi(\mathbf{y}) = \int_{\Omega} -G(\mathbf{x}|\mathbf{y})\frac{\rho(\mathbf{x})}{\epsilon_o}d\mathbf{x} + \oint_{\partial\Omega} \left(\Phi(\mathbf{x})\nabla_{\mathbf{x}} G(\mathbf{x}|\mathbf{y}) - G(\mathbf{x}|\mathbf{y})\nabla_{\mathbf{x}}\Phi(\mathbf{x})\right)\cdot \mathbf{n}\, dS_{\mathbf{x}} \tag{8}$$

where $\{y, \mathbf{y}\} \in \Omega \setminus \partial\Omega$. Note that $\Phi(\mathbf{y}) = \Phi_P(\mathbf{y}) + \Phi_H(\mathbf{y})$, where the volume integral, $\Phi_P(\mathbf{y})$, is the particular solution of (4), and the boundary integral, $\Phi_H(\mathbf{y})$, is the homogeneous solution. For later reference, we record the particular solutions,

$$\Phi_P(y) = -\int_{z_0}^{z_1} G(x|y)\frac{\rho(x)}{\epsilon_o}dx \tag{9}$$

and

$$\Phi_P(\mathbf{y}) = -\int_{\Omega} G(\mathbf{x}|\mathbf{y})\frac{\rho(\mathbf{x})}{\epsilon_o}d\mathbf{x} \tag{10}$$

and the homogeneous solutions

$$\Phi_H(y) = \left[\Phi(x)\frac{d}{dx}G(x|y) - G(x|y)\frac{d}{dx}\Phi(x)\right]_{x=z_0}^{z_1} \tag{11}$$

and

$$\Phi_H(\mathbf{y}) = \oint_{\partial\Omega} \left(\Phi(\mathbf{x})\nabla_{\mathbf{x}} G(\mathbf{x}|\mathbf{y}) - G(\mathbf{x}|\mathbf{y})\nabla_{\mathbf{x}}\Phi(\mathbf{x})\right)\cdot \mathbf{n}\, dS_{\mathbf{x}}. \tag{12}$$

Depending on the boundary conditions, $\Phi_H(\mathbf{y})$ can be represented as either a single layer or a double layer potential. This will be discussed below.

B. Volume Integral/Particular Solution

Recall that the charge density in general is given by

$$\rho(\mathbf{x}) = \int_{\mathbb{R}^d} \sum_{i=1}^{m} q_i f_i(t, \mathbf{x}, \mathbf{v}) d\mathbf{v} \tag{13}$$

where d is the number of velocity dimensions and m is the number of species. We consider two approaches for obtaining the potential Φ_P.

Particle Approximation: In our first approach, we represent the phase space distribution f_i as a set of delta functions

$$f_i(t,\mathbf{x},\mathbf{v}) = \sum_{j=1}^{N_i} w_j \delta(\mathbf{x}-\mathbf{x}_j)\delta(\mathbf{v}-\mathbf{v}_j)$$

where w_j is the weight at $\{\mathbf{x}_j,\mathbf{v}_j\}$. Substituting this into the equation for ρ gives

$$\rho(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{N_i} q_i w_j \delta(\mathbf{x}-\mathbf{x}_j)$$

and the particular solution then reduces to

$$\Phi_P(\mathbf{y}) = -\sum_{i=1}^{m}\sum_{j=1}^{N_i} \frac{q_i w_j}{\epsilon_o} G(\mathbf{x}_j|\mathbf{y}). \tag{14}$$

This is the same as in PIC, where the phase space particles $\mathbf{x}_j$ are viewed as macro-particles.

Integral Form: In our second approach, we restrict attention to the Vlasov equation so that the collision operator in the Boltzmann equation (1) is set to zero. We make this assumption in order to focus on problems in which diffusive effects due to collisions are negligible. In the absence of such diffusive effects, the solution of the Vlasov equation can develop fine scale features and these are handled here by point insertion. Extending this approach to collisional plasmas is a topic for future work.

In this approach, we employ ideas from Lagrangian vortex methods in computational fluid dynamics [31]. The volume integral over $\mathbf{x}$ for Φ_P, given above, is recast as a volume integral over coordinates $(\mathbf{x}_o,\mathbf{v}_o)$, where

$$\mathbf{x} = \mathbf{x}(t,\mathbf{x}_o,\mathbf{v}_o), \quad \mathbf{v} = \mathbf{v}(t,\mathbf{x}_o,\mathbf{v}_o)$$

is the flow map in phase space, i.e., $(\mathbf{x}_o,\mathbf{v}_o)$ is the initial condition (location, velocity) that maps into $(\mathbf{x},\mathbf{v})$ at a later time t. The initial condition $(\mathbf{x}_o,\mathbf{v}_o)$ is a Lagrangian parameterization of phase space and this is critical to the point insertion scheme, as explained below. The volume integral over initial condition for Φ_p is then discretized using the trapezoidal rule.

Before recasting the integral for Φ_p, it helps to think about how we treat the distribution function f_i. Consider a volume Γ_t in phase space at the initial time $t = t_0$ and a later time $t = T$ as in Fig. 2. Note that the sketch is intended to represent one, two, or three dimensions in space and velocity. Using the change of variables formula, the integral of f_i over volume Γ_T can be expressed as an integral over the initial volume Γ_{t_o}

$$\iint\limits_{\Gamma_T} f_i(T,\mathbf{x},\mathbf{v})d\mathbf{x}d\mathbf{v} = \iint\limits_{\Gamma_{t_o}} f_i\left(T,\mathbf{x}(T,\mathbf{x}_o,\mathbf{v}_o),\mathbf{v}(T,\mathbf{x}_o,\mathbf{v}_o)\right) J_o d\mathbf{x}_o d\mathbf{v}_o \tag{15}$$

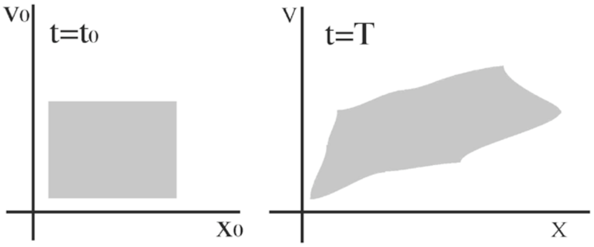


Fig. 2. Flow map in phase space. t_o: Initial time. T: Later time.

where $J_o = |\partial(\mathbf{x},\mathbf{v})/\partial(\mathbf{x}_o,\mathbf{v}_o)|$ is the Jacobian determinant of the flow map. It is well known that if the collision operator is identically zero, the fraction of f_i in a phase space volume Γ_{t_o} is preserved under the action of external fields, and, therefore, in the absence of absorbing boundaries, the volume integral of f_i over phase space must be constant for all $T \geq t_o$

$$\iint\limits_{\Gamma_T} f_i(T,\mathbf{x},\mathbf{v})d\mathbf{x}d\mathbf{v} = \iint\limits_{\Gamma_{t_o}} f_i(t_o,\mathbf{x}_o,\mathbf{v}_o)d\mathbf{x}_o d\mathbf{v}_o. \tag{16}$$

From (15) and (16), it follows that

$$f_i\left(T,\mathbf{x}(T,\mathbf{x}_o,\mathbf{v}_o),\mathbf{v}(T,\mathbf{x}_o,\mathbf{v}_o)\right) J_o = f_i(t_o,\mathbf{x}_o,\mathbf{v}_o) \tag{17}$$

which expresses the conservation of charge in phase space for the Vlasov–Poisson equation.

Now recall the particular solution in (10). Substituting the charge density from (13) into that expression gives

$$\Phi_P(\mathbf{y}) = -\sum_{i=1}^{m} q_i \iint\limits_{\Gamma_T} \frac{G(\mathbf{x}|\mathbf{y})}{\epsilon_o} f_i(T,\mathbf{x},\mathbf{v})d\mathbf{x}d\mathbf{v}.$$

Under our change of variables, the integrand is

$$\frac{G\left(\mathbf{x}(T,\mathbf{x}_o,\mathbf{v}_o)|\mathbf{y}\right)}{\epsilon_o} f_i\left(T,\mathbf{x}(T,\mathbf{x}_o,\mathbf{v}_o),\mathbf{v}(T,\mathbf{x}_o,\mathbf{v}_o)\right) J_o$$

and using (17), the particular solution becomes

$$\Phi_P(\mathbf{y}) = -\sum_{i=1}^{m} q_i \iint\limits_{\Gamma_{t_o}} \frac{G\left(\mathbf{x}(T,\mathbf{x}_o,\mathbf{v}_o)|\mathbf{y}\right)}{\epsilon_o} \times f_i(t_o,\mathbf{x}_o,\mathbf{v}_o)d\mathbf{x}_o d\mathbf{v}_o. \tag{18}$$

Assuming now an initial discretization of phase space $(\mathbf{x_j},\mathbf{v_k})$ and applying the trapezoidal rule, the integral over Γ_{t_o} for each species f_i can be approximated by

$$\sum_{\mathbf{j=0}}^{\mathbf{N}_i}\sum_{\mathbf{k=0}}^{\mathbf{M}_i} w_\mathbf{j} w_\mathbf{k} \frac{G\left(\mathbf{x}(T,\mathbf{x_j},\mathbf{v_k})|\mathbf{y}\right)}{\epsilon_o} f_i(t_o,\mathbf{x_j},\mathbf{v_k})$$

where $\mathbf{j}$ denotes a single or double sum approximating a single or double integral in space and $\mathbf{k}$ denotes a single, double, or triple sum approximating the velocity integral, and $w_\mathbf{j}$, $w_\mathbf{k}$ are

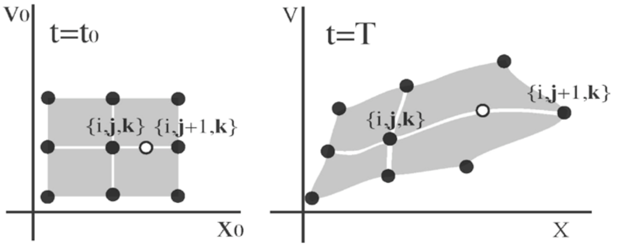


Fig. 3. Stretching of phase space and particle insertion. Black dots: Original particles. White dot: Inserted particle.

quadrature weights. Setting $\gamma_{\{i,\mathbf{j},\mathbf{k}\}} = w_{\mathbf{j}} w_{\mathbf{k}} f_i(t_o, \mathbf{x_j}, \mathbf{v_k})$, (18) is approximated by

$$\Phi_P(\mathbf{y}) = -\sum_{i=1}^{m}\sum_{\mathbf{j=0}}^{\mathbf{N}_i}\sum_{\mathbf{k=0}}^{\mathbf{M}_i} \frac{q_i\gamma_{\{i,\mathbf{j},\mathbf{k}\}}}{\epsilon_o} G\left(\mathbf{x}(T,\mathbf{x}_j,\mathbf{v}_k)|\mathbf{y}\right) \quad (19)$$

where $\gamma_{\{i,\mathbf{j},\mathbf{k}\}}$ is the phase space weight of a macro-particle. The sum in (19) resembles the sum in (14).

Due to the stretching of phase space during a simulation, it often happens that adjacent Lagrangian particles on a curve in phase space move far apart, as depicted in Fig. 3. In this case, the quadrature formula becomes inaccurate, but this can be controlled by inserting new particles to maintain resolution. Suppose that a particle needs to be inserted between $\{i,\mathbf{j},\mathbf{k}\}$ and $\{i,\mathbf{j}+1,\mathbf{k}\}$, based on some criterion, as shown in Fig. 3. Recall that the initial phase space coordinates are Lagrangian parameters. We insert a new particle at the midpoint of the interval between $\{i,\mathbf{j},\mathbf{k}\}$ and $\{i,\mathbf{j}+1,\mathbf{k}\}$ in phase space at $t = t_o$. The phase space weights γ of the new particle and the existing particles $\{i,\mathbf{j},\mathbf{k}\}$ and $\{i,\mathbf{j}+1,\mathbf{k}\}$ are determined in accord with the trapezoid rule over the initial condition. The phase space coordinates of the new particle at time T are determined using a cubic interpolating polynomial with respect to the Lagrangian parameters at particles $\{i,\mathbf{j}-1,\mathbf{k}\}$, $\{i,\mathbf{j},\mathbf{k}\}$, $\{i,\mathbf{j}+1,\mathbf{k}\}$, and $\{i,\mathbf{j}+2,\mathbf{k}\}$. The same process can be carried out to refine with respect to velocity as well. It should be noted that, in principle, intermediate times could be used to define the Lagrangian parameter for point insertion, however, that is not done here. Further details about the particle insertion scheme will be discussed elsewhere [41]. Related work is in references [29] and [37].

C. Treecode

Evaluating the particular solution in (14) and (19) is an example of N-body problem and the central processing unit (CPU) time is an important issue [42]. We employ a treecode to reduce the operation count from $O(N^2)$ to $O(N \log N)$. In a treecode, the particles are divided into a hierarchy of clusters and the particle–particle interactions are replaced by particle–cluster interactions, which are evaluated using multipole expansions. Barnes and Hut [26] used monopole approximations and a divide-and-conquer evaluation strategy, while Greengard and Rokhlin [27] used higher order spherical harmonics expansions and a more sophisticated evaluation procedure. Treecodes have been very successful in particle simulations and there is ongoing interest in optimizing their performance [43]–[51]. Our approach follows [29], which extends the Barnes–Hut treecode [26] in various ways.

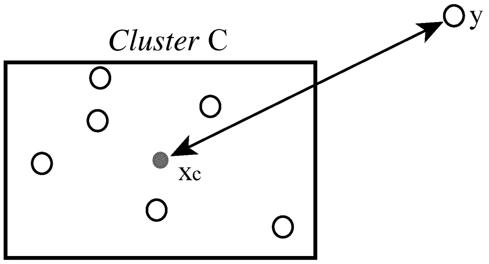


Fig. 4. Particle–cluster interaction: particle $\mathbf{y}$, cluster C, cluster center $\mathbf{x}_c$.

Particle–Cluster Interactions: The particular solution can be cast in the form

$$\Phi_P(\mathbf{y}) = -\sum_{i=1}^{m} \frac{q_i}{\epsilon_o}\Phi_i(\mathbf{y}), \quad \Phi_i(\mathbf{y}) = \sum_{j=1}^{\mathbf{N_i}} w_j G(\mathbf{x}_j|\mathbf{y}). \quad (20)$$

The simplest procedure for evaluating Φ_i is direct summation, but this requires $O(N^2)$ operations which is prohibitively expensive when N is large. The key idea in a treecode is to replace the particle–particle interactions in (20) by suitably chosen particle–cluster interactions.

The potential $\Phi_i(\mathbf{y})$ is first expressed as

$$\Phi_i(\mathbf{y}) = \sum_C \sum_{j\in C} w_j G(\mathbf{x}_j|\mathbf{y}) = \sum_C \Phi_i(\mathbf{y}, C) \quad (21)$$

where

$$\Phi_i(\mathbf{y}, C) = \sum_{j\in C} w_j G(\mathbf{x}_j|\mathbf{y}) \quad (22)$$

is the potential due to the interaction between particle $\mathbf{y}$ and a cluster of particles $C = \{\mathbf{x}_j | \mathbf{x}_j \in C\}$ (see Fig. 4). The procedure for choosing the clusters will be explained below; for now, it is enough to assume that the clusters in (21) are nonoverlapping and their union is the entire particle distribution.

Next, perform a Taylor expansion of the Green's function about the cluster center $\mathbf{x}_c$

$$\begin{aligned}\Phi_i(\mathbf{y}, C) &= \sum_{j\in C} w_j G\left(\mathbf{x}_c + (\mathbf{x}_j - \mathbf{x}_c)|\mathbf{y}\right)\\ &\approx \sum_{j\in C} w_j \sum_{l=0}^{p} \frac{1}{l!}\partial_{\mathbf{x}}^l G(\mathbf{x}_c|\mathbf{y})(\mathbf{x}_j - \mathbf{x}_c)^l\\ &= \sum_{l=0}^{p} \frac{1}{l!}\partial_{\mathbf{x}}^l G(\mathbf{x}_c|\mathbf{y}) \sum_{j\in C} w_j (\mathbf{x}_j - \mathbf{x}_c)^l\\ &= \sum_{l=0}^{p} T_l(\mathbf{x}_c, \mathbf{y}) M_l(C) \end{aligned} \quad (23)$$

where p is the order of approximation, $T_l(\mathbf{x}_c, \mathbf{y})$ is the lth Taylor coefficient of the Green's function, and $M_l(C)$ is the lth moment of the cluster. Note that Cartesian multi-index notation is used, e.g., in two dimensions if we set $\mathbf{x} = (x_1, x_2)$ and $l = (l_1, l_2)$, then $\mathbf{x}^l = x_1^{l_1} x_2^{l_2}$, and so forth. The error in the approximation is $O((r/R)^p)$, where $r = \max_j \|\mathbf{x}_j - \mathbf{x}_c\|_2$ is

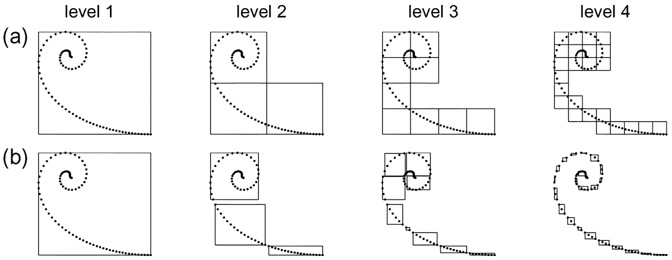


Fig. 5. Examples of tree structure. (a) Standard. (b) Adaptive [29].

the cluster radius and $R = \|\mathbf{x}_c - \mathbf{y}\|_2$ is the particle–cluster distance. The speedup occurs because the cluster moments are independent of the particle $\mathbf{y}$, and the Taylor coefficients are independent of the number of particles in the cluster C. The Taylor expansion in (23) is a Cartesian analog of the classical spherical harmonics multipole expansion [52]. A recurrence relation can be used to efficiently compute the Taylor coefficients to any desired order [29].

Tree Structure: The set of all clusters has a hierarchical tree structure. In a standard treecode, the clusters on each level of the tree are uniform cubes obtained from bisecting the previous generation of clusters in each coordinate direction. This is shown in Fig. 5(a) for a set of particles on a spiral curve. However, in many cases, it is beneficial to shrink the boxes as shown in Fig. 5(b). The nonuniform clusters in the second scheme are well-adapted to the particle distribution and this can lead to a savings in CPU time.

Evaluation Strategy: The potential Φ_i is evaluated using a recursive divide-and-conquer strategy [26]. As indicated in (21), the potential Φ_i at a given point $\mathbf{y}$ is expressed as a sum of particle–cluster interactions $\Phi_i(\mathbf{y}, C)$ for suitably chosen clusters C. The treecode has three options for evaluating each term $\Phi_i(\mathbf{y}, C)$: Taylor approximation, direct summation, or descending the tree. The present simulations used the standard multipole acceptance criterion (MAC), i.e., the Taylor expansion is used only if $r/R \leq \theta$, where r is the radius of the cluster, R is the distance between the particle and the cluster, and θ is a user-specified parameter [26], [45]. If the MAC is not satisfied, the code considers the interactions between particle $\mathbf{y}$ and the children of cluster C. If the cluster C has no children, then it is a leaf of the tree and $\Phi_i(\mathbf{y}, C)$ is evaluated by direct summation. The procedure is sketched in Fig. 6.

Electrostatic Force: The method described above can be adapted to compute the electrostatic force on a particle $\mathbf{x}_k$ in time $O(N \log N)$. Substituting (23) for the particle–cluster interaction into (20) for the particular solution, using the relation

$$\mathbf{E}_i(\mathbf{x}_k) = -\nabla_{\mathbf{y}} \Phi_i(\mathbf{y})|_{\mathbf{y}=\mathbf{x}_k}$$

for the electric field at the particle $\mathbf{x}_k$, and noting that $\nabla_{\mathbf{y}} G(\mathbf{x}|\mathbf{y}) = -\nabla_{\mathbf{x}} G(\mathbf{x}|\mathbf{y})$, the electric field is

$$\mathbf{E}_i(\mathbf{x}_k) \approx \sum_C \sum_{\substack{j \in C \\ j \neq k}} \sum_{l=0}^{p} \left(\nabla_{\mathbf{x}_c} T_l(\mathbf{x}_c, \mathbf{x}_k)\right) M_l(C).$$

Then the electrostatic force on $\mathbf{x}_k$ due to species i is $q_k w_k \mathbf{E}_i$, where $q_k w_k$ is the total charge on particle k.

```
function $\mathbf{ComputePotential}(\mathbf{y}, C)$
  is the Taylor approximation sufficiently accurate?
    if yes, compute $\Phi_i(\mathbf{y}, C)$ by Taylor approximation
    if no, does $C$ have any children?
      if yes, call $\mathbf{ComputePotential}(\mathbf{y}, C')$ for each
        child $C'$ of $C$
      if no, compute $\Phi_i(\mathbf{y}, C)$ by direct summation
```

Fig. 6. Procedure for evaluating a particle–cluster interaction $\Phi_i(\mathbf{y}, C)$.

Regularizing the Green's Function: In two and three dimensions, the interparticle force $\mathbf{F}_j$ in the equations of motion (2) becomes singular as the distance between the particles tends to zero, placing a constraint on the maximum allowable time step. Our approach to overcoming this problem is to regularize the Green's function [29], [37]. In two dimensions, we use

$$G_\delta(x|y) = -\frac{\ln\left(\|\mathbf{x} - \mathbf{y}\|_2^2 + \delta^2\right)}{4\pi}$$

where δ is a smoothing parameter, so that the maximum force is proportional to $1/\delta$. The appropriate value of δ could be determined by physical properties of the system, e.g., in a neutral plasma, one could use the Debye shielding length to choose δ. However, in the work presented here, δ was treated as a numerical parameter.

D. Boundary Integral/Homogeneous Solution

Having completed our discussion of the particular solution, next we describe a boundary integral method for the homogeneous solution; for an alternative finite element formulation, see [38]. Recall that the potential function Φ satisfies the Poisson equation (4) subject to Dirichlet, Neumann, or mixed boundary conditions depending on the application. The homogeneous solution is given in (11) or (12) and at each point on the boundary either Φ, $\nabla\Phi \cdot \mathbf{n}$, or $a\Phi + b\nabla\Phi \cdot \mathbf{n}$ is specified. As an example, consider the case of Dirichlet boundary conditions in which $\Phi = \alpha$ is known but $\nabla\Phi \cdot \mathbf{n} = \beta$ is unknown. We require the boundary condition to be satisfied in the sense that

$$\lim_{\mathbf{y} \to \mathbf{z}} \Phi(\mathbf{y}) = \alpha(\mathbf{z}), \quad \text{for} \quad \mathbf{z} \in \partial\Omega. \tag{24}$$

A similar relation holds as well for the other possible boundary conditions.

For our Dirichlet example, this yields an equation for β; a 2 × 2 linear system in one dimension

$$\begin{aligned} &\lim_{y \to z_{\{0,1\}}} \left(\beta(z_1) G(z_1|y) - \beta(z_0) G(z_0|y)\right) \\ &= \left(-\frac{1}{2}\alpha(z_1) - \frac{1}{2}\alpha(z_0)\right) - \lim_{y \to z_{\{0,1\}}} \left(\alpha(y) - \Phi_P(y)\right) \end{aligned} \tag{25}$$

and an integral equation of the first kind in higher dimensions

$$\begin{aligned} \oint_{\partial\Omega} \beta(\mathbf{x}) G(\mathbf{x}|\mathbf{z}) dS_{\mathbf{x}} = -\frac{\alpha(\mathbf{z})}{2} + \Phi_P(\mathbf{z}) \\ + \oint_{\partial\Omega} \alpha(\mathbf{x}) \nabla_{\mathbf{x}} G(\mathbf{x}|\mathbf{z}) \cdot \mathbf{n}\, dS_{\mathbf{x}} \end{aligned} \tag{26}$$

where $\alpha(\mathbf{z})$ on the right side of (26) is divided by 2 to account for the singular nature of the limit as $\mathbf{y}$ approaches the boundary [53].

While (25) can be directly solved for the unknowns, (26) will be solved by a collocation method. First, discretize the boundary integrals into panels $\partial\Omega_j$ for $j = 1, \dots, M$

$$\sum_{j=1}^{M} \int_{\partial\Omega_j} \beta(\mathbf{x}) G(\mathbf{x}|\mathbf{z}) dS_{\mathbf{x}} = -\frac{\alpha(\mathbf{z})}{2} + \Phi_P(\mathbf{z}) + \sum_{j=1}^{M} \int_{\partial\Omega_j} \alpha(\mathbf{x}) \nabla_{\mathbf{x}} G(\mathbf{x}|\mathbf{z}) \cdot \mathbf{n}\, dS_{\mathbf{x}}. \quad (27)$$

Choosing $\mathbf{z} = \mathbf{z}_i$ to be the center of the ith panel and letting $\alpha = \alpha_j$ and $\beta = \beta_j$ be constant along each panel $\partial\Omega_j$ (for example), we obtain

$$\sum_{j=1}^{M} \beta_j \sum_{l} w_l G(\mathbf{x}_{jl}|\mathbf{z}_i) = -\frac{\alpha(\mathbf{z}_i)}{2} + \Phi_P(\mathbf{z}_i) + \sum_{j=1}^{M} \alpha_j \sum_{l} w_l \nabla_{\mathbf{x}} G(\mathbf{x}_{jl}|\mathbf{z}_i) \cdot \mathbf{n}_j, \quad (28)$$

where $\mathbf{x}_{jl}$ are Gaussian quadrature points on the jth panel and w_l are the corresponding weights. This yields a linear system for the β_j

$$A\beta = \kappa$$

where

$$a_{ij} = \sum_{l} w_l G(\mathbf{x}_{jl}|\mathbf{z}_i)$$

and

$$\kappa_i = -\frac{\alpha(\mathbf{z}_i)}{2} + \Phi_P(\mathbf{z}_i) + \sum_{j=1}^{M} \alpha_j \sum_{l} w_l \nabla_{\mathbf{x}} G(\mathbf{x}_{jl}|\mathbf{z}_i) \cdot \mathbf{n}_j.$$

This formulation worked well in our preliminary work [36]. For geometrically complex domains, the linear system becomes challenging to solve and an efficient method such as preconditioned generalized minimal residual (GMRES) is required. Due to the special form of the matrix elements a_{ij}, the matrix-vector multiplication in each step of GMRES can be computed efficiently using the treecode. Hence, the matrix A never needs to be explicitly constructed and stored, making geometrically complex domains more tractable.

This approach can also be used for problems with Dirichlet and Neumann conditions on different portions of the boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. In this case, we represent Φ_H as a combination of single and double layer potentials

$$\Phi_H(\mathbf{y}) = \int_{\partial\Omega_N} \alpha(\mathbf{x}) G(\mathbf{x}|\mathbf{y}) dS_{\mathbf{x}} + \int_{\partial\Omega_D} \beta(\mathbf{x}) \nabla_{\mathbf{x}} G(\mathbf{x}|\mathbf{y}) \cdot \mathbf{n} dS_{\mathbf{x}}$$

where α, β are unknown functions. This eliminates the need to compute a boundary integral for each κ_i. We found through numerical experiments that the resulting matrix A is well-conditioned when a double layer potential is used for the Dirichlet boundary and a single layer potential is used for the Neumann boundary. Mixed boundary conditions can be handled by going back to the original formulation using (12) for Φ_H.

III. Applications

In this section, we give an overview of the problems to which the BIT approach has been applied. We divide the discussion into two sections based on the problem dimension.

A. 1-D Problems

We used BIT to simulate several benchmark 1-D problems in plasma physics including sheath formation, virtual cathode formation, and the two-stream instability. In the latter case, we also used particle insertion.

Sheath Formation: The 1-D sheath domain is bounded by two parallel metal plates a distance $z_1 - z_0 = L$ apart, each held at a potential of zero volts. The domain initially contains a neutral plasma with equal electron and ion densities. The electrons are lighter and more mobile than the ions, and a fraction of the electrons near the wall impact the wall on a timescale much faster than the ion transit time. The reduced electron density near the wall leaves a net positive charge in a narrow region between the wall and the main body of the plasma. This gives rise to a potential barrier (sheath) at the walls, and if any of the remaining electrons are to escape, they must have a large enough velocity to overcome the sheath potential. Eventually the system reaches a steady state in which the electrons are electrostatically confined to the interior of the domain.

We looked at this application in order to demonstrate that BIT can reporoduce the undriven sheath in a decaying plasma as an initial benchmark. This sheath decay includes some kinetic elements, since the tail is lost first and the Coulomb collision rate at these low densities is insufficient to maintain a Maxwell–Boltzmann distribution, so kinetic effects are important, and hence this is best represented by a kinetic model. This type of problem, decay of an undriven bounded plasma and resulting sheath formation, has been studied by PIC at some length [54]–[56], and is widely used as a benchmark for other models including fluid models [57], [58].

In this simulation, the electrons are represented as mobile particles, while the ions, being much heavier, are immobile. The Poisson equation takes the form

$$\nabla^2 \Phi = \frac{e}{\varepsilon_0} \left[\sum_{j=1}^{N_e} w_j \delta(x - x_j) - n_i \right] \quad (29)$$

where e is the electron charge, N_e is the number of electrons in the simulation, and n_i is the constant background ion density, chosen such that $n_i L = w_j N_e$. The electrons are initially distributed uniformly in space with a random thermal velocity sampled from a Maxwellian distribution.

The potential is the sum of three terms

$$\Phi(y) = \Phi_e(y) + \Phi_i(y) + \Phi_H(y) \quad (30)$$

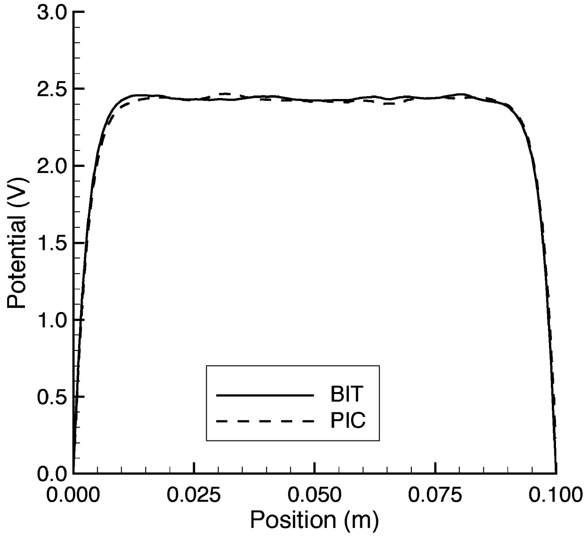


Fig. 7. Potential Φ as a function of position across domain for BIT and PIC.

where Φ_e is due to the electrons, Φ_i is due to the background ion density, and Φ_H is the homogeneous solution needed to enforce the boundary conditions

$$\left.\begin{array}{lcl} \frac{d^2}{dx^2}\Phi_e &=& \frac{e}{\varepsilon_0}\sum_{i=1}^{N_e} w_i\delta(x-x_i) \\ \frac{d^2}{dx^2}\Phi_i &=& -\frac{e}{\varepsilon_0}n_i \\ \frac{d^2}{dx^2}\Phi_H &=& 0 \\ \Phi_H|_{\{z_0,z_1\}} &=& -[\Phi_e+\Phi_i]_{\{z_0,z_1\}} \end{array}\right\}. \tag{31}$$

The term Φ_e is given by (14), Φ_i is obtained by integrating twice, and Φ_H is found by substituting (25) into (11)

$$\left.\begin{array}{lcl} \Phi_e(y) &=& \frac{e}{\varepsilon_0}\sum_{i=1}^{N_e} w_i G(x_i|y) \\ \Phi_i(y) &=& -\frac{e}{\varepsilon_0}\frac{1}{2}n_i y^2 \\ \Phi_H(y) &=& G(z_0|y)\beta(z_0) - G(z_1|y)\beta(z_1) \end{array}\right\} \tag{32}$$

where we have also imposed the boundary conditions $\Phi(z_0) = \Phi(z_1) = 0$ in (11). With these boundary conditions, the limits in (25) give

$$\left.\begin{array}{l} \beta(z_0) = -G(z_0|z_1)^{-1}\left[\Phi_e(z_1)+\Phi_i(z_1)\right] \\ \beta(z_1) = G(z_1|z_0)^{-1}\left[\Phi_e(z_0)+\Phi_i(z_0)\right] \end{array}\right\}. \tag{33}$$

BIT results will be compared with PIC results from [59]. Note that the 1-D Green's function in (7) is piecewise linear and so the first order Taylor approximation is exact; this implies that BIT computes exact forces and potentials in one dimension, and gives identical results to direct summation. In the work presented here, the BIT simulation was initialized using the same parameter set as in the PIC simulation so that the only difference was in the field solver. In both simulations, the domain length is $L = 0.1$ m, the number density of each species is 10^{13} m^{-3}, and the electron temperature is 1 eV. The initial number of electrons is 4000 in both simulations.

Fig. 7 plots the potential Φ as a function of position across the domain for BIT and PIC. There is no significant difference between the two methods, either in magnitude or sheath thickness. The calculated matrix sheath thickness for this domain is 5.4 mm [60], comparable to the computed results of approximately 10 mm for both BIT and PIC.

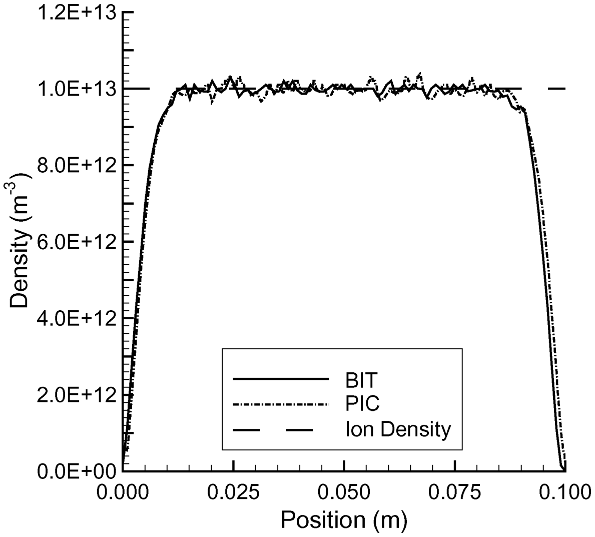


Fig. 8. Electron density for BIT and PIC. Ion density is also shown.

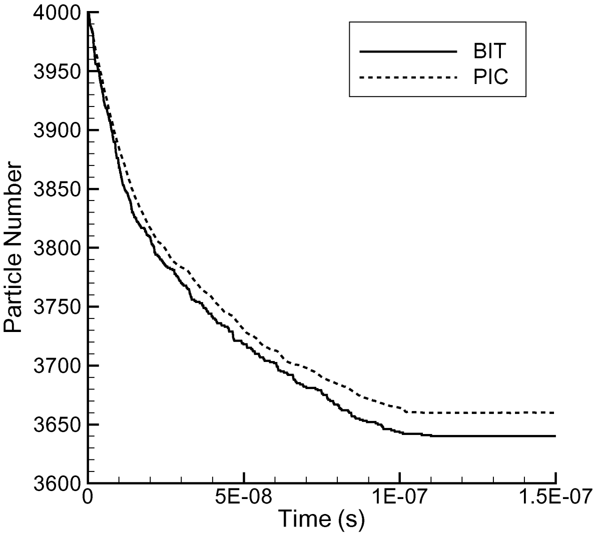


Fig. 9. Number of electrons as a function of time for BIT and PIC.

Fig. 8 plots the electron and ion densities for BIT and PIC. As for the potential, the computed particle densities for the two methods are almost the same.

Fig. 9 plots the number of electrons in the simulation as a function of time. Again the agreement between the BIT and PIC results is very good. Both have an initial depletion rate that matches the theoretical result. The particle number drops slightly faster for BIT, resulting in approximately 20 fewer particles after convergence, but this has negligible effect on the potential and the particle densities. To examine whether the additional loss in BIT arises from the initial seed of the Maxwellian velocity distribution, many additional runs were done, each using a different seed in the random number generator. All of the results were within 1–3 particles of the BIT results in Fig. 9. Note that this does not imply that BIT and PIC will agree in a full discharge simulation. One explanation for the discrepancy in the number of particles is that BIT includes short range Coulomb collisions which are neglected in PIC.

Hence, we expect that the tail in BIT will be repopulated more quickly than in PIC. Since the tail is the only part that can escape the potential well, we then expect BIT to have a higher loss rate and hence a slightly lower number of particles than PIC.

Virtual Cathode: Virtual cathodes (VC) are basic structures that arise in many plasma applications, e.g., they have been used to explain anomalous acceleration of positive ions in a vacuum diode [61], and have also been proposed as a confinement mechanism for controlled fusion [63]. The formation of a VC is understood in general terms, but predicting the details is difficult because the process is highly nonlinear. Such systems are often simulated using PIC [64], but it is not clear that this is optimal since sharp gradients in the plasma properties occur near the VC.

In previous work, we simulated the formation of a virtual cathode in one dimension using a BIT algorithm [35]. Two infinite parallel flat plates have applied voltages. The gap between the plates is initially a vacuum, and one of the plates is heated to the point where electrons are emitted and a current flows across the gap. For a sufficiently high emission rate, the maximum current density between the plates approaches a limiting value $J_{\rm CL}$ predicted by the Child–Langmuir law. When this happens, a virtual cathode is formed somewhere between the two plates, and the emitted electrons turn around and strike the emitting plate. As the electrons repel one another, the potential minimum rises, thereby allowing the emitted electrons to again start crossing the gap, and the entire sequence repeats. We considered the classical formulation in which the electrodes are held at ground and the emitted electrons have an initial velocity [39].

For this nonlinear oscillator, we compared PIC, BIT, and direct summation (DS). As noted in the section on sheath formation, DS and BIT agree to machine round off, so the main reason for the DS calculation is for timing comparisons. For the BIT and DS runs, the grid-free solution to the Poisson's equation is identical to (30), except that n_i is identically zero. The PIC routine employed here used standard area-weighting. The injected current was varied by adjusting the charge/mass ratio of the particles. With a small injected current, the solution converged to a steady state and all three methods gave the same result to within plotting accuracy. The mesh size for the PIC simulation was determined by decreasing the value until the solution converged to the nonoscillatory steady state. This mesh spacing was then used for the remainder of the PIC simulations.

With a higher injected current, a VC forms and the solution converges to a time-periodic state with nonuniform spatial structure. Fig. 10 shows the time trace of the particle density (n/n_0) at three locations x/L between the plates; just past injection, near the VC, and downstream of the VC. Fig. 10 reveals significant differences between the BIT/DS and PIC results. First, the oscillation period is 5% greater for PIC than for BIT/DS, producing a significant phase lag. Fig. 11, shows the particle distribution in phase space at a given time. The plot shows that some of the emitted particles turn back and approach the emitter plate. The PIC results have irregular voids, while the BIT/DS results have a smooth particle distribution.

All three methods used the same particle push and particle injection/deletion procedures. The only difference between the three routines was the force evaluation kernel. In all simulation results presented, the time step used was that of the time converged DS method. The criterion for the time step was that the phase space point wise L_2 norm must be less then a user set tolerance

$$\left\| DS(\Delta t) - DS\left(\frac{\Delta t}{2}\right) \right\|_2 < \theta$$

at any observed time between t_o and t_f. This guarantees pointwise convergence in phase space. The observed differences in

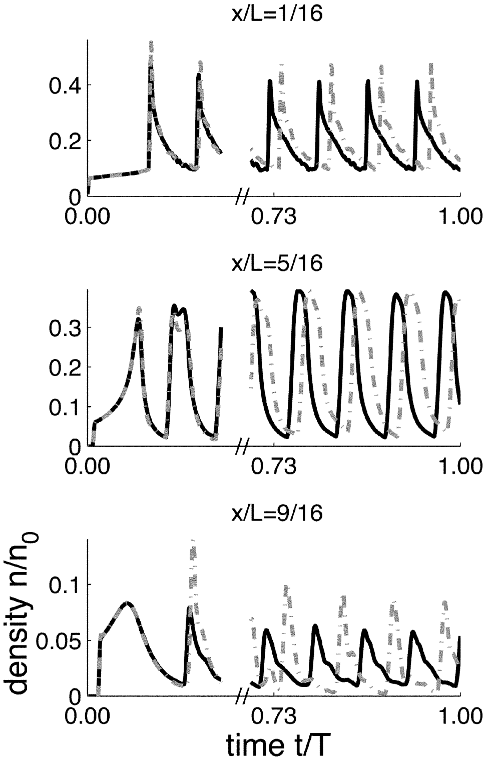


Fig. 10. Time trace in the formation of a virtual cathode at three spatial locations across the channel, $x/L = 1/16, 5/16, 9/16$. Solid line is BIT/DS and the dashed line is PIC.

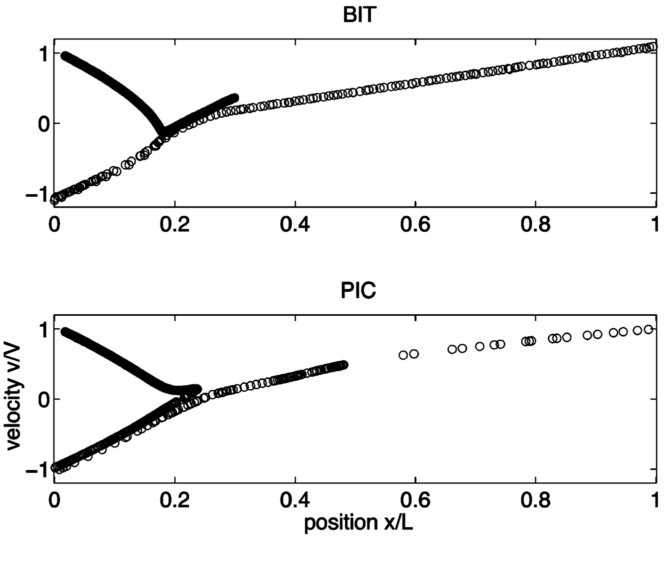


Fig. 11. Phase space plot at time $t/T = 0.5$. Irregular voids appear for PIC, while BIT/DS has a smooth particle distribution.

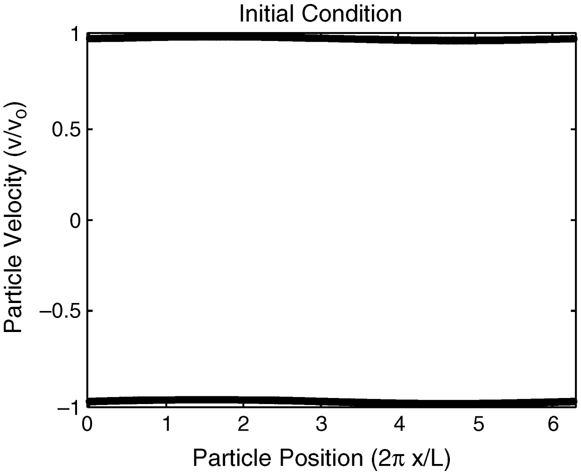


Fig. 12. Two-stream instability; initial electron distribution in phase space.

the simulation results, shown in Figs. 10 and 11, can be attributed to the error in interparticle force within a mesh cell of a PIC simulation, as shown in Fig. 1. These preliminary runs were performed using Matlab; PIC required 7 h, BIT 12 h, and DS seven days. Further work is needed to optimize the run time of the BIT code, but the results suggest that BIT is capable of resolving small-scale features better than a mesh-based PIC code.

Two-Stream Instability: This example concerns a neutral plasma with equal numbers of ions and electrons in a 1-D periodic domain. As above, the electrons evolve dynamically and the ions are immobile. Consider an electron distribution function of the form $f(x,v,t_0) = f(x,-v,t_0) = g(|v|)$, i.e., independent of position x and an even function of velocity v. This is an unstable equilibrium solution of the Vlasov–Poisson system. A perturbation in velocity gives the electrons a spatial variation in momentum. This leads to nonuniformity in the electron spatial distribution, thereby generating an electric field which feeds back into the velocity perturbation, causing phase space to fold up on itself.

As with the virtual cathode, we focus on a cold two-stream instability in the sense that

$$g(|v|) = \begin{cases} \frac{1}{2}, & \text{if } v = \pm v_o \\ 0, & \text{otherwise} \end{cases}$$

where v_o is the initial unperturbed speed of the electrons. The initial velocity perturbation has the form

$$v_p(x) = a \sin\left(2\pi \frac{x - z_0}{z_1 - z_0}\right)$$

where a is the perturbation amplitude. Fig. 12 shows the initial electron distribution in phase space. The electrons are modeled as particles while the ions are treated as a constant background density n_i. The potential of the domain is then described by

$$\nabla^2 \Phi = \frac{e}{\varepsilon_0}\left[\sum_{j=1}^{N_e} w_j \delta(x - x_j) - n_i\right]$$

with periodic boundary conditions

$$\Phi(z_0) = \Phi(z_1), \frac{d\Phi(z_0)}{dx} = \frac{d\Phi(z_1)}{dx}$$

where e is the charge on an electron and N_e is the number of electrons in the simulation.

As in the sheath formation problem, the particular solution Φ_P is the sum of Φ_e and Φ_i from (32). The boundary conditions determine the homogenous solution given by (25), i.e., $\Phi(z_0) = \Phi(z_1)$, $d\Phi(z_0)/dx = d\Phi(z_1)/dx$ gives $\alpha(z_0) = \alpha(z_1)$, $\beta(z_0) = \beta(z_1)$. Thus, (25) gives

$$\beta(z_1)G(z_1|z_0) = \left(-\frac{1}{2}\alpha(z_1) - \frac{3}{2}\alpha(z_0)\right) + \Phi_P(z_0)$$

and

$$-\beta(z_0)G(z_0|z_1) = \left(-\frac{3}{2}\alpha(z_1) - \frac{1}{2}\alpha(z_0)\right) + \Phi_P(z_1).$$

Using the periodic boundary conditions and noting that $G(z_1|z_0) = G(z_0|z_1)$, we obtain

$$\alpha(z_1) = \frac{1}{4}\left(\Phi_P(z_0) + \Phi_P(z_1)\right)$$

and

$$\beta(z_1) = \frac{1}{2G(z_1|z_0)}\left(\Phi_P(z_0) - \Phi_P(z_1)\right).$$

The initial ion and electron density is 4×10^6 cm^{-3}. All the phase space plots are normalized to the initial electron velocity $v = c$. Time stepping was performed using the fourth-order Runge–Kutta (RK4) method and the results were considered time accurate when the difference between runs with Δt and $\Delta t/2$ was less then 5%. Spatial (i.e., phase space) convergence was achieved by systematically increasing the number of simulation points N_e. Fig. 13, top frame, shows a time converged result for $N_e = 3200$ after 64 000 time steps with $\Delta t = (z_1 - z_0)/kc$, where $k = 16 \times 10^4$. The electron distribution function is concentrated on a rolled-up curve in phase space. Fig. 13, middle and bottom frame, shows that the simulation loses spatial accuracy later in time when adjacent particles become separated.

To overcome the loss of resolution that inevitably occurs using a fixed number of particles, we implemented a particle insertion scheme as discussed briefly in Section II-B. In these results a simple distance criterion is used, i.e., a new particle is inserted between particles i and $i+1$ when

$$\sqrt{(x_i - x_{i+1})^2 + (v_i - v_{i+1})^2} \geq \frac{3}{2} \cdot S$$

where S is the initial particle spacing. Fig. 14 shows the result obtained using particle insertion, starting with 200 points and using the same time step as in Fig. 13. This simulation was checked for convergence both in time as well as in space by computing with twice as many initial points. The results indicate that particle insertion is an efficient means for maintaining accuracy as the solution evolves in time. A detailed analysis of these results, as well as extension to the warm two-stream instability, are topics for future work.

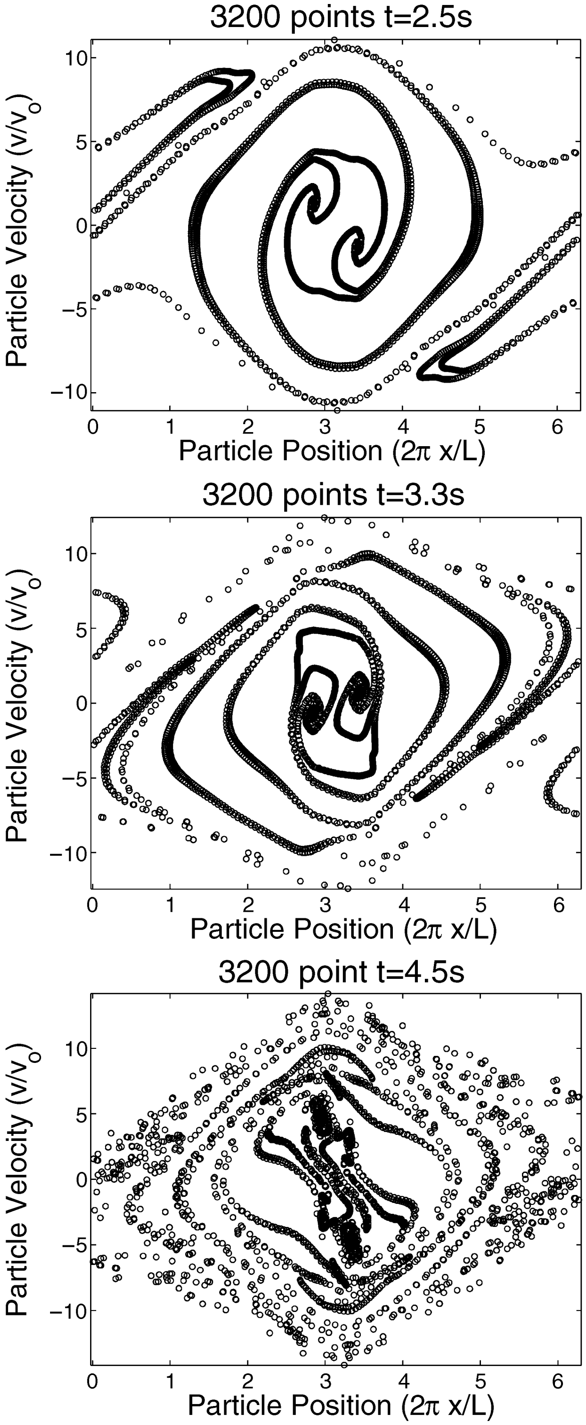


Fig. 13. Two-stream instability; phase space plot with $N_e = 3200$. Top frame $t = 2.5$. Middle frame $t = 3.3$. Bottom frame $t = 4.5$.

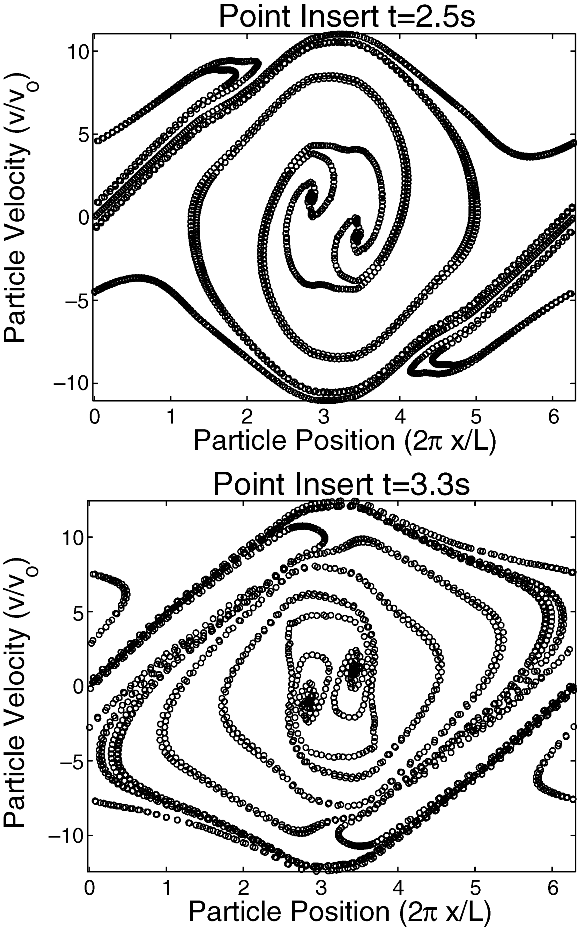


Fig. 14. Two-stream instability; phase space plot using point insertion, initial $N_e = 200$. Top frame $t = 2.5$. Bottom frame $t = 3.3$.

B. 2-D Problems

In addition to the 1-D problems discussed above, we also applied the BIT approach to several 2-D problems, including planar and cylindrical ion optics, and particle dynamics in a Penning–Malmberg trap. We performed a detailed timing comparison in 2-D planar geometry between BIT and multigrid PIC. We start with the timing comparison and then discuss the applications listed above.

Timing Results: In this section, we compare the efficiency of BIT and PIC simulations for a 2-D test case, focusing on the question: for a given level of accuracy, which method is faster? In this test, N particles were randomly placed in a square domain with grounded boundary and the electric field was computed at each particle location using BIT, PIC, and direct sum.

Note that if the homogeneous solution in (12) is computed exactly, then direct sum yields the exact solution of the test problem. The Green's function for a square domain can be expressed as an infinite series, but we chose to satisfy the boundary condition using the free-space Green's function. This requires solving a boundary integral equation for the homogeneous solution. We expressed the homogeneous solution as a single layer potential and subdivided the boundary into panels as discussed in Section II-D. In this example, the terms in the matrix, $a_{ij} = \int G(x_i|x_j)dS_{x_j}$, can be integrated exactly and the panel strengths are determined to within roundoff error. The number of panels in the direct sum solution is chosen so that the error is less than 1%. The resulting boundary integral direct sum result is taken as the exact solution.

The error is measured by the expression

$$\text{error} = \frac{1}{N}\sum_{i=1}^{N} \left\| \mathbf{E}_i^{ds} - \mathbf{E}_i^{\text{appr}} \right\|_2$$

where $\mathbf{E}_i^{ds}$ is the direct sum electric field at particle i and $\mathbf{E}_i^{\text{appr}}$ is the BIT or PIC electric field approximation. Table I shows timing results versus number of particles for DS, PIC-MG, and BIT. PIC-M denotes a PIC solution on an $M \times M$ grid

TABLE I
CPU TIME (S) FOR DS, PIC-M, BIT-T (SEE TEXT FOR DEFINITIONS)

N	DS	PIC-128	PIC-256	PIC-512	BIT-1	BIT-2
150K	792	0.50	4.53	32.4	19	22
300K	2995	0.52	4.54	32.7	38	45
600K	11633	0.55	4.55	32.9	78	90

TABLE II
ERROR FOR PIC-M, BIT-T (SEE TEXT FOR DEFINITIONS)

N	DS	PIC-128	PIC-256	PIC-512	BIT-1	BIT-2
150K	-	14%	10%	7%	8%	$< 1\%$
300K	-	13%	8%	6%	7%	$< 1\%$
600K	-	14%	10%	8%	6%	$< 1\%$

using a multigrid solver [62]. BIT-T denotes a boundary integral/treecode solution using a Taylor expansion of order T (BIT-1 used MAC parameter $\theta = 0.9$ and BIT-2 used $\theta = 0.45$). Table II shows the corresponding errors.

A few observations can be made. 1) As a function of N, DS is $O(N^2)$, PIC-MG is almost independent of N over this range, and BIT displays $O(N)$ behavior although theoretically it is $O(N \log N)$. 2) For fixed N, the time required for PIC increases by a factor of 8 when the mesh is refined by a factor of 2 in each direction. Alternatively, the time required for BIT increases by a fraction less than one when going from first to second order Taylor expansion. In terms of accuracy, the PIC-512 and BIT-1 results have comparable errors, while the BIT-2 results have much smaller errors. Note that PIC would require an exceedingly fine mesh to achieve the same accuracy as BIT-2, with a correspondingly large increase in the CPU time (estimated to be at least 2000 s). Note that in Table II, the PIC results display first-order convergence; this is due to the choice of interpolation and the effect of differencing the potential to obtain the electric field.

The timing results imply that for problems where strong coulomb interactions matter, an extremely fine spatial discretization may be required to resolve these effects in a PIC simulation. In summary, for the choice of parameters used in this strong coulomb interaction test, PIC is more efficient for 10% accuracy and BIT is more efficient for 1% accuracy. This implies that for a large class of problems, where sharp gradients in plasma properties develop in localized regions, it may be ideal to think about a hybrid PIC-BIT algorithm. In such an algorithm, BIT would only be applied to small patches of the simulation domain in order to correct the PIC calculation.

Planar Optics: Ion thrusters for spacecraft propulsion operate by electrostatically accelerating ions through a set of ion optics. The optics consist of a screen grid and an accelerator grid, each of which typically has many thousands of hexagonally-arranged apertures. Ion optics simulations usually focus on a single aperture and the immediate vicinity upstream and downstream. Fig. 15 shows the geometry and boundary conditions in the present example. Ions are introduced at the upstream edge of the domain at a discharge potential of about 2000 V. The screen grid has a potential 25 V below the discharge potential and the accelerator grid has a negative potential on the order of −200 V. Ions are accelerated electrostatically by the drop in potential, providing thrust. Downstream of the optics, the beam is neutralized by an electron source, making the downstream potential approximately zero, although the present BIT simulations included only the ions.

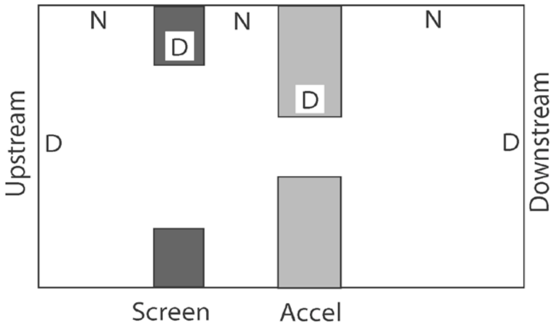


Fig. 15. Geometry for planar ion optics simulation. D: Dirichlet. N: Neumann.

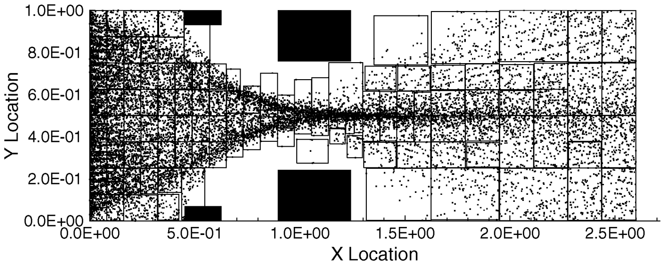


Fig. 16. Snapshot of particles in a planar ion optics simulation. Also shown are the leaf clusters in the treecode, assuming a maximum of 200 particles per leaf.

In the present simulations, ions were injected at a rate of 10 per iteration, each with a weight corresponding to number density $2.3 \cdot 10^{13}\ \mathrm{m}^{-3}$. This low density was used because the absence of neutralizing electrons causes the space-charge limit to be reached at much lower density than in typical ion thruster operation. The timestep was $\Delta t = 10^{-9}$ s and the particles were injected with Maxwellian velocities centered about 2000 m/s. The boundaries were modeled using a combination of single and double layer potentials, with panels uniformly distributed on each segment. The line integrals of the Green's function and its normal derivative were integrated exactly over each panel—our previous work used quadrature [36], but exact integration was found to mitigate the effect of the kernel singularities in evaluating particle-boundary interactions. The treecode used a fourth-order Taylor expansion and the MAC parameter was $\theta = 0.1$.

Fig. 16 shows a snapshot of the computed particle locations. Also shown are the leaf clusters in the treecode, assuming a maximum of 200 particles per leaf. The clusters conform to the particle locations; no effort is expended on empty regions, regions with low particle density have fewer clusters, and regions with high particle density have many clusters. A detailed analysis of the results is in preparation.

Cylindrical Optics: In this case, BIT is applied to a more complex problem than in the previous test cases. One problem that arises is the inclusion of an electron population for the direct summation and BIT. The PIC potential solver models a Boltzmann electron fluid at the mesh nodes, but BIT has no mesh for such a simulation and there is no known Green's function for the fully nonlinear problem. The free-space Green's function for the axisymmetric case is

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{\pi L}\int_0^{\frac{\pi}{2}} \frac{d\theta}{(1-m\sin^2\theta)^{\frac{1}{2}}} = \frac{K(m)}{\pi\sqrt{L}} \qquad (34)$$

where $\mathbf{x}, \mathbf{y}$ have cylindrical coordinates (r_x, z_x), (r_y, z_y), $L^2 = (r_x + r_y)^2 + (z_x - z_y)^2$, $m = 4r_x r_y/L^2$, and $K(m)$ is the complete elliptic integral of the first kind. The present axisymmetric treecode is relatively inefficient compared to the planar version due to the lack of a recurrence relation for the derivatives of the axisymmetric Green's function. As a result, in this example, the treecode CPU time is at least an order of magnitude greater than the PIC CPU time. The treecode could have been made more efficient by using a lookup table for the elliptic function evaluations or a different form of the Green's function [65]. The problem does not arise in three dimensions because an efficient recurrence relation is known for that case [29].

The simulation domain radius was 10^{-3} m, the upstream length was 2×10^{-3} m, the downstream length was 4×10^{-3} m, the screen grid was 0.4 mm thick with a 1.6-mm-diameter aperture, and the accel grid was 0.8 mm thick with a 1.0-mm-diameter aperture. As for planar optics, the boundary was modeled as a combination of single and double layer potentials. The boundary matrix A was constructed using an eight point Gaussian quadrature rule. To control errors at sharp corners, a Chebyshev panel spacing was used along boundary segments. The total number of panels was 512. The upstream domain boundary potential was 1800 V, the screen grid was 1775 V, the accel grid was -210 V, the downstream boundary was 22 V (all Dirichlet), and the other upper boundaries were homogeneous Neumann. The smoothing parameter value was $\delta = 4\times10^{-5}$ m. The treecode used fourth order Taylor expansions, the MAC parameter was $\theta = 0.16$, and the leaves had no more than eight particles per lowest level cluster.

The comparison done here is between direct summation and PIC in order to provide a baseline comparison. Because the differences are fairly large in this case, a comparison using the treecode would appear exactly the same due to its closeness to direct summation. The direct summation comparison to PIC imparts all the relevant information.

Fig. 17 plots the average relative difference in the particle force between PIC and direct summation and the average electric field across the domain. In the central region, the imposed electric field dominates, giving a small relative difference. In the neutral regions of the domain upstream and downstream of the ion optics, the electric field is small on average. PIC computes very small forces in these regions, while direct summation still sees large interparticle forces. The average PIC force magnitude in the upstream region is on the order of 10^{-17} N, while the average direct sum force is approximately 10^{-16} N. A detailed analysis of the results is in preparation.

Penning–Malmberg Trap: A Penning–Malmberg trap is a grounded conducting cylinder used for confining an electron plasma [66], [67]. The electrons are confined by applying a magnetic field along the cylinder axis and holding the cylinder end caps at constant voltage, so that the electrons bounce back and forth along the magnetic field lines (see Fig. 18). For example in [67], the end caps were held at 50 V and the magnetic field strength was $B \sim 1$ T. Under these conditions, the time required for an electron to complete one bounce is much smaller than the characteristic $\mathbf{E}\times\mathbf{B}$ time scale. In this case, the plasma is well described by a 2-D particle model in which the electrons behave like line charges being convected with velocity $v = \mathbf{E}\times\mathbf{B}/B^2$ [68]. The system has been investigated experimentally and computationally, and many interesting phenomena were revealed including metastable crystalline states and complex dynamics [66], [67].

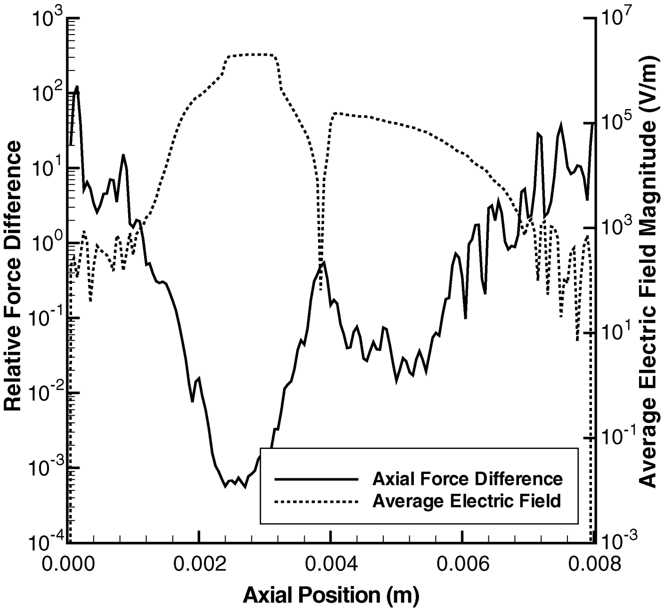


Fig. 17. Relative z force difference between PIC and direct summation.

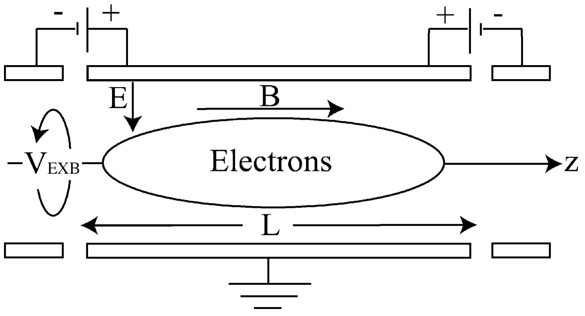


Fig. 18. Sketch of a Penning–Malmberg trap for confining an electron plasma.

In our simulations, the applied magnetic field $\mathbf{B}$ is a specified constant and the electric field $\mathbf{E}$ is computed using BIT. In addition, the particle insertion scheme discussed in Section II-B was used to maintain resolution of the electron density. The insertion criterion compares the distance between hypothetical particles inserted using linear interpolation and cubic interpolation. If the distance is greater than 5% of the initial particle spacing, a new particle is inserted. The time integration method used is RK4. The BIT field evaluation used an eighth order Taylor expansion and the MAC parameter was $\theta = 0.5$. The Dirichlet boundary condition on the cylinder wall was imposed using a single layer potential and the terms in the boundary matrix A were evaluated using eight–point Gaussian quadrature. Refinement tests were carried out in space and time.

Here, we present a simulation of wave breaking in which the cylinder parameters, size, magnetic field strength, initial electron density profile, etc., were chosen as in the experiment [69].

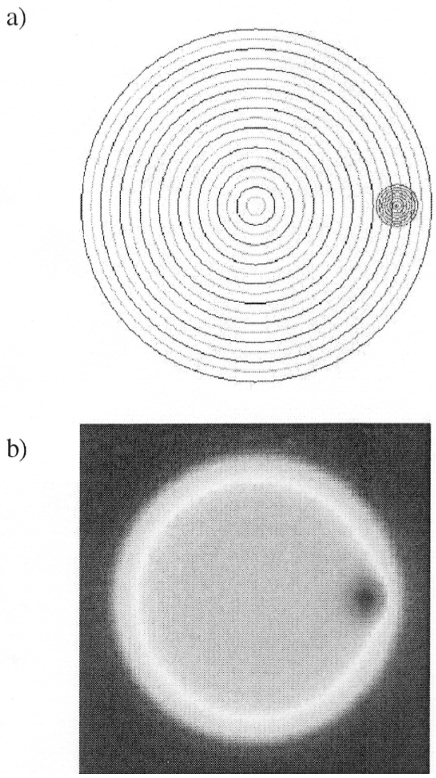


Fig. 19. Initial condition for wave breaking simulation. (a) Particle curves. (b) Electron density. The small dark gray patch is four times as dense as the large background patch.

A small dot of radius is 0.1035 cm and high density $n_{\mathrm{dot}} = 1.0 \times 10^7\ \mathrm{cm}^{-3}$ is superimposed on a large background electron disk of radius 0.8 cm and low density $n_{\mathrm{disk}} = 1.8 \times 10^6\ \mathrm{cm}^{-3}$. Fig. 19 shows the initial condition with (a) curves on which the particles were placed, and (b) electron density mapped onto a mesh. The dot rotates about the center of the disk; Fig. 20 shows the solution after one rotation and Fig. 21 shows the solution after four rotations. As the dot rotates, it entrains material from the disk and causes a wave to form on the disk boundary. The results are in excellent agreement with experiment [69]. Note that the filamentation in Fig. 21(a) gives rise to a diffuse region when mapped to a mesh, as in Fig. 21(b). Since the experimental results are obtained by crashing the electrons onto a cathode ray tube and capturing the image with a charge-coupled device camera, the resulting spatial blurring may be similar to that of mapping the particles onto a mesh. Hence, the diffuse region seen in the experiment may be due to filamentation beyond the experimental resolution. A detailed analysis of the results is in preparation [41].

IV. Future Work

Our short term goal is to apply BIT and particle insertion to simulate the warm two-stream instability. In addition, we are extending our 2-D field solvers to three dimensions, as well as optimizing the treecode approach for solving the linear system associated with the homogeneous solution. In the future, we plan to incorporate a mesh-free DSMC code currently under development to permit the mesh-free simulation of collisional plasmas. We plan to begin the development of a nonstatistical collision operator which will accommodate particle insertion. An additional goal is to incorporate cluster–cluster approximations similar in spirit to the fast multipole method [27]. Finally, we plan

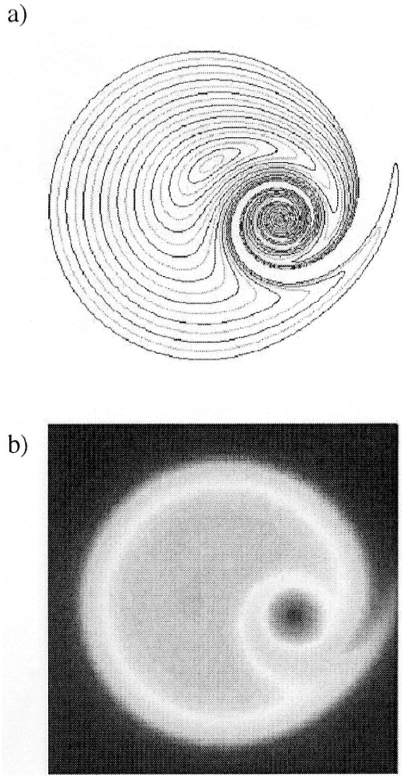


Fig. 20. Wave breaking, one rotation. (a) Particle curves. (b) Electron density. Color indicates density (red is high; blue is low). The small dark gray patch is four times as dense as the large background patch.

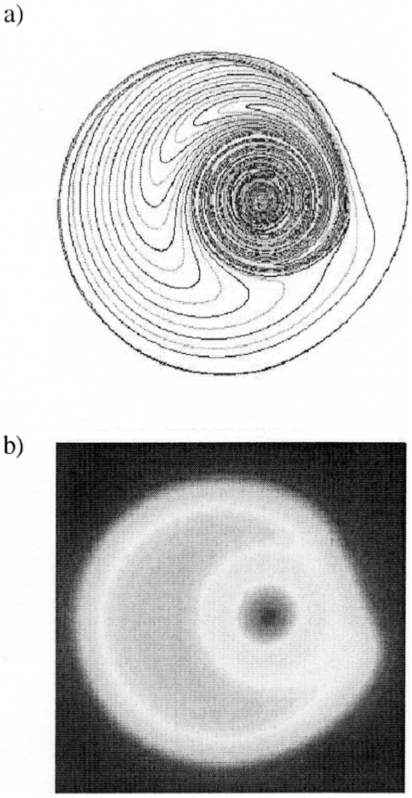


Fig. 21. Wave breaking, four rotations. (a) Particle curves. (b) Electron density. Color indicates density (red is high; blue is low). The small dark gray patch is four times as dense as the large background patch.

to investigate whether this grid-free approach can be extended to problems with time-varying magnetic fields.

V. Conclusion

We have applied a grid-free boundary integral/treecode (BIT) field solver to several bounded plasma problems and obtained comparable or better results than traditional mesh-based methods. We demonstrated that BIT exhibits comparable timing for a given accuracy and is capable of handling complex geometry, as well as a mixture of boundary conditions. Further validation is underway. We believe that BIT offers an attractive alternative to mesh-based approaches for electrostatic problems. The extension of these grid-free methods to plasma problems involving time-varying magnetic fields is still a significant challenge.

References

[1] I. D. Boyd and A. Ketsdever, "Interactions between spacecraft and thruster plumes," *J. Spacecraft Roc.*, vol. 38, no. 3, p. 380, 2001.

[2] W. Denig and M. Leardini. Space environmental sensors for the defense meteorological satellite program. [Online]. Available: http://www.afrl-horizons.com/Briefs/Aug04/VS0308.html

[3] G. Lapenta and J. U. Brackbill, "Simulation of dust particle dynamics for electrode design in plasma discharges," *Plasma Sources Sci. Technol.*, vol. 6, pp. 61–69, 1997.

[4] S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases*. Cambridge, U.K.: Cambridge Univ. Press, 1952.

[5] T. J. Sommerer, W. N. G. Hitchon, and J. E. Lawler, "Electron heating mechanisms in He RF discharges: a self-consistent kinetic calculation," *Phys. Rev. Lett.*, vol. 69, pp. 2361–2364, 1989.

[6] G. J. Parker, W. N. G. Hitchon, and J. E. Lawler, "Self-consistent kinetic model of an entire DC discharge," *Phys. Lett. A*, vol. 174, pp. 308–311, 1993.

[7] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*. Bristol, U.K.: IOP Publishing, 1991.

[8] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. Bristol, U.K.: IOP Publishing, 1988.

[9] G. G. M. Coppa, G. Lapenta, G. Dellapiana, F. Donato, and V. Riccardo, "Blob method for kinetic plasma simulation with variable-size particles," *J. Comp. Phys.*, vol. 127, pp. 268–284, 1996.

[10] G. J. Parker and W. N. G. Hitchon, "Convected scheme simulations of the electron distribution function in a positive column plasma," *Jpn. J. Appl. Phys.*, vol. 36, no. 7B, pp. 4799–4807, 1997.

[11] D. Hewett, "Grid and particle hydrodynamics," *J. Comput. Phys.*, vol. 144, pp. 358–378, 1998.

[12] A. J. Christlieb, W. N. G. Hitchon, and E. R. Keiter, "A computational investigation of the effects of varying discharge geometry for an inductively coupled plasma," *IEEE Trans. Plasma Sci.*, vol. 28, no. 6, pp. 2214–2231, Dec. 2000.

[13] D. Hewett, "Fragmentation, merging, and internal dynamics for pic simulations with finite size particles," *J. Comput. Phys.*, vol. 189, pp. 390–426, 2003.

[14] V. Kolobov and R. Arslanbekov, "Four dimensional Fokker-Planck solver for electron kinetics in collisional gas discharge plasmas," *Comp. Phys. Comm.*, vol. 164, pp. 195–201, 2004.

[15] J. P. Verboncoeur, "Particle simulation of plasmas: review and advances," *Plasma Phys. Control. Fusion*, vol. 47, pp. A231–A260, 2005.

[16] N. Hershkowitz, G. L. Payne, C. Chan, and J. R. DeKock, "Weak double layers," *Plasma Phys.*, vol. 23, pp. 903–925, 1981.

[17] P. Gibbon, F. N. Beg, E. L. Clark, R. G. Evans, and M. Zepf, "Tree-code simulations of proton acceleration from laser-irradiated wire targets," *Phys. Plasmas*, vol. 11, no. 8, pp. 4032–4040, 2004.

[18] A. C. J. Paes, N. M. Abe, V. A. Serrao, and A. Passaro, "Simulations of plasmas with electrostatic PIC models using the finite element method," *Brazilian J. Phys.*, vol. 33, pp. 411–417, 2003.

[19] J. W. Eastwood, "Particle simulation methods in plasma physics," *Comput. Phys. Commun.*, vol. 43, pp. 89–106, 1986.

[20] K. Nanbu, "Theory of cumulative small-angle collisions in plasmas," *Phys. Rev. E*, vol. 55, pp. 4642–4652, 1997.

[21] K. Nanbu and S. Yonemura, "Weighted particle in cell Coulomb collision simulations based on the theory of a cumulative scattering angle," *J. Comput. Phys.*, vol. 145, pp. 639–654, 1998.

[22] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *J. Comput. Phys.*, vol. 53, pp. 484–512, 1984.

[23] M. J. Berger and A. Jameson, "Automatic adaptive grid refinement for the Euler equations," *AIAA J.*, vol. 23, pp. 561–568, 1985.

[24] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *J. Comput. Phys.*, vol. 82, pp. 64–84, 1989.

[25] J. L. Vay, P. Colella, P. McCorquodale, B. Van Straalen, A. Friedman, and D. P. Grote, "Mesh refinement for particle-in-cell plasma simulations: Applications to and benefits for heavy ion fusion," *Laser Particle Beams*, vol. 20, pp. 569–575, 2002.

[26] J. Barnes and P. Hut, "A hierarchical $O(N \log N)$ force-calculation algorithm," *Nature*, vol. 324, pp. 446–449, 1986.

[27] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, pp. 325–348, 1987.

[28] H. Cheng, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," *J. Comput. Phys.*, vol. 155, pp. 468–498, 1999.

[29] K. Lindsay and R. Krasny, "A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow," *J. Comput. Phys.*, vol. 172, pp. 879–907, 2001.

[30] S. J. Aarseth, *Gravitatoinal N-Body Simulations*. Cambridge, U.K.: Cambridge Unive. Press, 2003.

[31] G.-H. Cottet and P. D. Koumoutsakos, *Vortex Methods Theory and Practice*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[32] T. Schlick, *Molecular Modeling and Simulation. An Interdisciplinary Guide*. New York: Springer-Verlag, 2002.

[33] S. Pfalzner and P. Gibbon, *Many-Body Tree Methods in Physics*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[34] ——, "Direct calculation of inverse-bremsstrahlung absorption in strongly coupled, nonlinearly driven laser plasmas," *Phys. Rev. E*, vol. 57, pp. 4698–4705, 1998.

[35] A. J. Christlieb, R. Krasny, and J. P. Verboncoeur, "Efficient particle simulation of a virtual cathode using a grid-free treecode Poisson solver," *IEEE Trans. Plasma Sci.*, vol. 32, no. 2, pp. 384–389, Apr. 2004.

[36] ——, "A treecode algorithm for simulating electron dynamics in a Penning–Malmberg trap," *Comput. Phys. Commun.*, vol. 164, pp. 306–310, 2004.

[37] R. Krasny, "Computation of vortex sheet roll-up in the Trefftz plane," *J. Fluid Mech.*, vol. 184, pp. 123–155, 1987.

[38] J. T. Katsikadelis, *Boundary Elements: Theory and Applications*. New York: Elsevier , 2002.

[39] C. K. Birdsall and W. B. Bridges, *Electron Dynamics of Diode Regions*. New York: Academic, 1966.

[40] J. P. Verboncoeur and C. K. Birdsall, "Rapid current transition in a crossed-field diode," *Phys. Plasmas*, vol. 3, pp. 712–713, 1996.

[41] A. J. Christlieb and R. Krasny, "Particle dynamics in a Penning–Malmberg trap," , submitted for publication.

[42] L. Greengard, "Fast algorithms for classical physics," *Science*, vol. 265, pp. 909–914, 1994.

[43] L. van Dommelen and E. A. Rundensteiner, "Fast, adaptive summation of point forces in the two-dimensional Poisson equation," *J. Comput. Phys.*, vol. 83, pp. 126–147, 1989.

[44] C. R. Anderson, "An implementation of the fast multipole method without multipoles," *SIAM J. Sci. Stat. Comput.*, vol. 13, pp. 923–947, 1992.

[45] J. K. Salmon and M. S. Warren, "Skeletons from the treecode closet," *J. Comput. Phys.*, vol. 111, pp. 136–155, 1994.

[46] D. W. Elliott and J. A. Board Jr., "Fast Fourier transform accelerated multipole algorithm," *SIAM J. Sci. Comput.*, vol. 17, pp. 398–415, 1996.

[47] J. H. Strickland and R. S. Baty, "A pragmatic overview of fast multipole methods," *Lect. Appl. Math.*, vol. 32, pp. 807–830, 1996.

[48] H. Cheng, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," *J. Comput. Phys.*, vol. 155, pp. 468–498, 1999.

[49] J. Makino, "Yet another fast multipole method without multipoles—Pseudoparticle multipole method," *J. Comput. Phys.*, vol. 151, pp. 910–920, 1999.

[50] W. Dehnen, "A hierarchical $O(N)$ force calculation algorithm," *J. Comput. Phys.*, vol. 179, pp. 27–42, 2002.

[51] L. Ying, G. Biros, and D. Zorin, "A kernel-independent adaptive fast multipole algorithm in two and three dimensions," *J. Comput. Phys.*, vol. 196, pp. 591–626, 2004.

[52] J. D. Jackson, *Classical Electrodynamics*. New York: Wiley, 1975.

[53] G. B. Folland, *Introduction to Partial Differential Equations. Second Edition*. Princeton, NJ: Princeton Univ. Press, 1995.

[54] S. J. Hana and J. K. Lee, "Kinetic simulations of the transient sheath in plasma ion- implantation," *Jpn. J. Appl. Phys.*, vol. 31, no. 8, pp. 2570–2579, 1992.

[55] S. C. Paek, S. R. Yoon, and Y. S. Cho, "Analysis of capacitively coupled RF glow-discharges by particle simulations," *J. Korean Phys. Soc.*, vol. 26, no. 5, pp. 466–471, 1993.

[56] K. Koga, H. Naitou, and Y. Kawai, "Observation of local structures in asymmetric ion sheath," *J. Phys. Soc. Jpn.*, vol. 65, no. 5, pp. 1578–1584, 1999.

[57] R. Mares, M. Vicher, and R. Hrach, "PIC and fluid model of plasma sheath," *Czechoslovak J. Phys.*, vol. 52, pp. 705–709, 2002.

[58] E. Faudot, S. Heuraux, and L. Colas, "Modeling of DC electric fields induced by RF sheath in front of ICRF antenna," *Czech. J. Phys.*, vol. 53, pp. 911–923, 2003.

[59] M. A. Lieberman and A. J. Lichtenberg, *Principles of Plasma Discharges and Materials Processing*. New York: Wiley-Interscience, 1994.

[60] J. R. Roth, *Industrial Plasma Engineering: Volume 1: Principles*. Philadelphia, PA: IOP Publishing, 2001.

[61] S. A. Barengolts, N. Y. Kazarinov, G. A. Mesyats, E. A. Perelshtein, and V. F. Shevtsov, "A simplified model of the formation of a deep potential well in a vacuum diode," *IEEE Trans. Plasma Sci.*, vol. 31, no. 5, pp. 847–851, Oct. 2003.

[62] W. L. Briggs, *A Multigrid Tutorial*. Philadelphia, PA: SIAM Books, 1987.

[63] D. C. Barnes, R. A. Nebel, and L. Turner, "Production and application of dense Penning trap plasmas," *Phys. Fluids B*, vol. 5, pp. 3651–3660, 1993.

[64] J. Park, R. A. Nebel, W. G. Rellergert, and M. D. Sekora, "Experimental studies of electrostatic confinement on the intense neutron source-electron device," *Phys. Plasmas*, vol. 10, pp. 3841–3849, 2003.

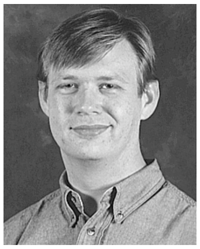
[65] J. H. Strickland and D. E. Amos, "Fast solver for systems of axisymmetric ring vortices," *AIAA J.*, vol. 30, pp. 737–746, 1992.

[66] D. A. Schecter, D. H. E. Dubin, K. S. Fine, and C. F. Driscoll, *Phys. Fluids*, vol. 11, pp. 905–914, 1999.

[67] C. F. Driscoll, D. Z. Jin, D. A. Schecter, and D. H. E. Dubin, *Physica C*, vol. 369, pp. 21–27, 2002.

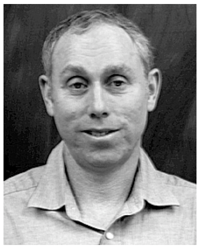
[68] F. F. Chen, *Plasma Physics and Controlled Fusion*. New York: Plenum, 1984.

[69] F. F. Durkin and F. F. Fajans, "Experimental dynamics of a vortex within a vortex," *Phys. Rev. Lett.*, vol. 85, pp. 4052–4055, 2000.



Andrew J. Christlieb (M'04) received the B.S. degree in mathematics, engineering mathematics, and electrical engineering from the University of Michigan-Dearborn, in 1996. He received the M.S. degree in applied mathematics and the Ph.D. degree from an interdisciplinary National Science Foundation (NSF)-sponsored program, Mathematics and Computers in Engineering from the University of Wisconsin-Madison, in 1998 and 2001, respectively. His thesis advisor was Prof. W. N. G. Hitchon in the Electrical and Computer Engineering Department, University of Wisconsin-Madison. His thesis was joint work between applied mathematics and electrical engineering, and focused on developing efficient numerical methods for long mean free path particle transport.

In the summer of 1999, he worked as a graduate student intern under the supervision of Dr. J. Foster in the Electric Propulsion Group at the NASA John H. Glenn Research Center. While there, he worked on updating particle-in-cell codes for the simulation of ion optics and developed data acquisition systems for ion thrusters. In 2001–2002, he was a Postdoctoral Research Fellow with Professor I. Boyd in the Aerospace Engineering Department, University of Michigan-Ann Arbor, where he worked on numerical simulations of micro-scale lifting bodies. In 2002, he joined the Mathematics Department, University of Michigan-Ann Arbor, as an Assistant Professor. His research interests include efficient Boltzmann solvers (including particle-based approaches), scientific computing, and modeling of dynamical systems such as plasmas and micro/nano-scale fluidics. He is currently working on developing grid-free particle methods for plasma dynamics. Applications of interest include space propulsion systems, plasma processing, and micro/nano-based technologies.



Robert Krasny received the B.S. and M.A. degrees in mathematics from the State University of New York, Stonybrook, and the Ph.D. degree in applied mathematics from the University of California, Berkeley.

He is currently a Professor of Mathematics at the University of Michigan, Ann Arbor. Prior to this, he held a National Science Foundation Postdoctoral Research Fellowship at the Courant Institute of Mathematical Sciences, New York University, and he also worked as a scientific programmer in the Reactor Safety Division at Brookhaven National Laboratory. His research involves developing numerical algorithms and using them to investigate mathematical and physical properties of complex physical systems. He has developed Lagrangian particle methods for vortex sheet roll-up, formation of vortex rings, and chaotic dynamics in vortex cores. His recent work focuses on improving the capability of treecode algorithms for long-range particle interactions.

Dr. Krasny was an invited speaker at the International Congress of Mathematicians (Kyoto, Japan, 1990).

John P. Verboncoeur (M'96) received the B.S. degree (with high honors) in engineering science at the University of Florida, Gainesville, in 1986, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1987 and 1992, respectively, holding the DOE-administered Magnetic Fusion Energy Technology Fellowship.

As a Postdoc at the University of California, Berkeley, and then Lawrence Livermore National Laboratory (LLNL), Livermore, CA, he did pioneering work on bounded plasma models and applying object-oriented technology to computational plasma physics. He was appointed Associate Professor in Residence in 2001. He currently leads the Computational Engineering Science program at the University of California, Berkeley, in its third year with about 40 students. His research interests are in theoretical and computational plasma physics, broadly defined to include electromagnetics, vacuum electronics, beam optics, plasma discharges, fusion energy, and numerical methods. He is the author/coauthor of the Berkeley suite of particle-in-cell Monte Carlo collision (PIC-MCC) codes, including XPDP1 and XOOPIC. He has authored/coauthored over 30 journal articles, and has taught ten international workshops and mini-courses on plasma simulation. He has also worked on a number of nonacademic projects including the Strategic Air Command Executive Support System, the U.S. Postal Service Mail Forwarding System, and the TRW Credit Data Consumer Report System, in addition to developing and licensing a number commercial hardware and software tools.

Jerold W. Emhoff received the B.S.E. (*summa cum laude*), M.S.E., and Ph.D. degrees in aerospace engineering from the University of Michigan, Ann Arbor, in 2000, 2002, and 2005, respectively.

He worked as a Graduate Student Research Assistant under Dr. I. Boyd on the modeling of ion thruster optics. He worked closely in this research with the NASA Glenn Research Center's Electric Propulsion Group. He also explored the use of the treecode algorithm applied to plasma physics with Dr. A. Christlieb. He is currently a Research Scientist at the University of Michigan, Ann Arbor, and has accepted a position as a Postdoctoral Researcher at the Applied Physics Laboratory, Johns Hopkins University, Baltimore, MD. His primary field of interest is in the modeling of plasmas using computer simulation, and the further development of techniques for this purpose. He has developed a particle-in-cell model for ion optics erosion and life prediction, and has also worked toward a treecode-based ion optics model. Applications of interest are electric propulsion such as ion, Hall, or pulsed-plasma thrusters, as well as space or fusion plasmas.

Iain D. Boyd received the B.S. degree in mathematics and the Ph.D. degree in aeronautics and astronautics from the University of Southampton, Southampton, U.K., in 1985 and 1988, respectively.

He worked for four years as a Contractor at NASA Ames Research Center in the area of aerothermodynamics. He was a faculty member in Mechanical and Aerospace Engineering Department, Cornell University, Ithaca, NY, for six years, and has been in the Department of Aerospace Engineering at the University of Michigan, Ann Arbor, for six years where he is a Full Professor. His research interests involve the development of physical models and numerical algorithms using particle methods with applications to a variety of nonequilibrium gas and plasma dynamic systems. He has authored over 100 journal articles and over 150 conference papers. He is an Associate Editor of the *Journal of Spacecraft and Rockets*.

Dr. Boyd is an associate fellow of the American Institute of Aeronautics and Astronautics (AIAA). He the recipient of the 1998 AIAA Lawrence Sperry Award and the 1997 AIAA Electric Propulsion Best Paper Award.