Journal of Computational Physics **162,** 483–513 (2000)
doi:10.1006/jcph.2000.6549, available online at http://www.idealibrary.com on IDEAL®

Loading and Injection of Maxwellian Distributions in Particle Simulations[1]

K. L. Cartwright,* J. P. Verboncoeur,† and C. K. Birdsall†

**Air Force Research Laboratory, Kirtland AFB, New Mexico 87117-5776; and †Electronics Research Laboratory, University of California, Berkeley, California 94720*

Received December 21, 1999; revised May 5, 2000

Existing and new particle loading and injection algorithms for particle simulations are analyzed to determine numerical accuracy and computational efficiency. Emphasis has been placed on loading and emission of Maxwellian, drifting Maxwellian, and cutoff Maxwellian velocity distributions. Once a velocity distribution has been inverted for loading or injection, time-centering of the position and velocity is necessary in order to maintain second-order accuracy. Here, the accuracy of these methods is determined and compared to three analytic test cases with spatially varying, time-dependent, and time-independent electric fields in a homogeneous magnetic field and a self-consistent crossed-field diode. The initial push is shown to be important in calculating the correct electric field at the boundary where particles are injected, in relaxing constraints on the time step, and in providing reliable field fluctuations due to particle statistics. © 2000 Academic Press

1. INTRODUCTION

We first observed injection difficulties when detailed comparisons were made between 1d and 2d diodes [1–3] when using common particle loading and injection algorithms. The phase space plots (position-velocity; Fig. 1) showed incorrect velocities with small gaps (incorrect position) in emission. The culprit was initiating the leap-frog integrator with velocity, **v**, and position, **x**, at the same time; **v** and **x** should be $\Delta t/2$ apart in time in order to obtain second-order accuracy. The result of not time-centering the velocity and position correctly was a zero-order error in each emitted particle.

For some models, this error might have a small or unnoticeable effect on the simulation. However, in models we were (and are) studying for noise and stability in cross field devices,

[1] This research was supported in part by Office of Naval Research AASERT N100014-94-1-1033, Air Force Office of Scientific Research-MURI Grant F49620-95-1-0253, and Air Force Office of Scientific Research Grants FDF 49620-96-1-0154 and F49620-98-C-0053.

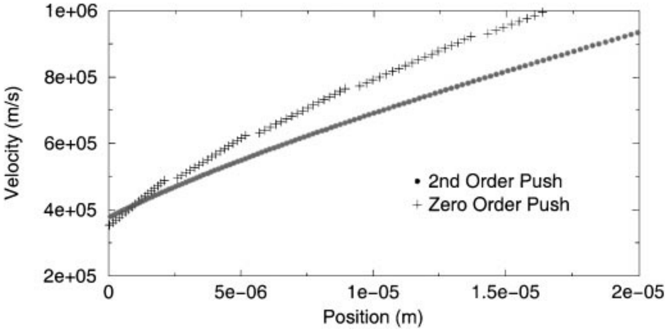

FIG. 1. The particle phase space near the cathode emitter in equilibrium with the critical current injected for the zero-order injection model and the new second-order accuracy injection method.

these errors were unacceptable. Hence, the choice was to devise a second-order accurate method to start the leap-frog integrator.

The initial conditions of the particle distribution function, $f(\mathbf{x}, \mathbf{v}, t = 0)$, where $\mathbf{x}$, $\mathbf{v}$, and t are the position, velocity, and time, respectively, are important in determining the later behavior of the system, especially for the study of transients or instabilities. The distribution function continues to be important for simulations with collisions using Monte Carlo collision (MCC) packages, but the emphasis here is long time (longer than an ion transit time) scale developments. The boundary condition of the particle distribution, $f(\mathbf{x} = \text{wall}, \mathbf{v}, t)$ (for example, thermionic, field, or secondary emission), which may be a function of time, is also important in determining the continuing behavior of the system. These considerations are especially important for modeling diodes [4], electron guns [5], crossed-field emission [4, 6, 7], multipactor breakdown [8], and any time particle emission is near the space charge limit [9]. Incorrect loading and injection can often manifest itself in nonobvious ways, such as larger than expected field fluctuations; this will be discussed in detail later.

Chapter 16 of Birdsall and Langdon [10] discusses some methods in wide use for inverting densities n($\mathbf{x}$) and f($\mathbf{v}$) as well as the particle flux, $\mathbf{\Gamma} = \mathbf{v}_\perp f(\mathbf{x} = \text{wall}, \mathbf{v}, t)$, where $\mathbf{v}_\perp$ is the velocity perpendicular to the wall. Although these boundary conditions are required for many bounded PIC codes, a comprehensive accuracy or error analysis has not appeared in the literature. The purpose of this paper is to expand upon the commonly used injection and loading schemes, adding an analysis of the accuracy of a number of techniques. This paper discusses techniques for converting model distributions (stated with $\mathbf{x}$ and $\mathbf{v}$ given at the same time) into $\mathbf{x}$ and $\mathbf{v}$ suitable for the common second-order accurate leap-frog integrator (with $\mathbf{x}$ and $\mathbf{v}$ half a time step, $\Delta t/2$, apart).

The organization of this paper is as follows. First, we will discuss and define cumulative distribution and scalings used in this paper. Second, we will review the inversion of Maxwellian distributions and fluxes with the addition of refinements to improve numerical accuracy and efficacy. Last, we will discuss several new methods for time-centering particles injected from the edge of the simulation and compare the results of these new methods in a crossed-field diode.

1.1. *Motivation*

In order to demonstrate the error, we present highlights of injection in a magnetized crossed-field diode, previously simulated and published by Verboncoeur and Birdsall [4]

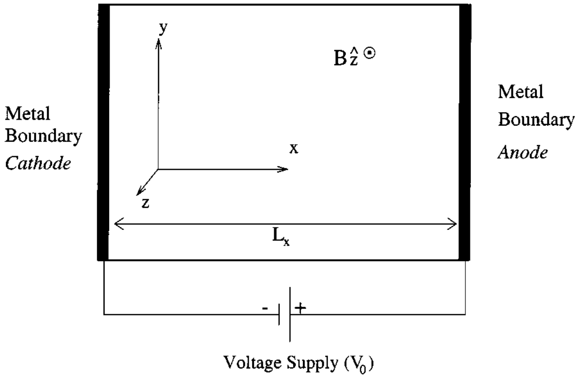

FIG. 2. Model for the three cases and self-consistent examples. The cathode is at $x = 0$.

using the particle-in-cell (PIC) code XPDP1 (X-window plasma device planar one-dimensional) [11]. A diagram of the diode is shown in Fig. 2, and the physical and numerical parameters are summarized in Table II; the complete behavior is given in Section 3.4. The theory for this diode was derived by Lau *et al.* [12].

Compared to the theoretical predictions, XPDP1 version 3.1 (which used the old initial advance for **x** and **v**) obtains a small 0.2% error in the critical current and a large 64% error in the cathode electric field and surface charge. Using the second-order initial step presented in this paper, with the same numerical parameters, results in a less than 3×10^{-5}% error in the critical current, and a 4% error in the expected cathode electric field. These errors also illustrate that not all parameters in a simulation are affected equally by the initial push. The critical current is insensitive to the initial push while the surface field was sensitive.

Phase space near the cathode, when the simulation is in equilibrium, is shown for both old (zeroth-order) and new (second-order) initial step algorithms in Fig. 1. The phase space shows the results of using the more accurate injection push; first, it removes the incorrect spacing of particles in the successive time steps. Second, the correct positions of the particles produce a space charge different than the incorrect injection; thus, the electric field that was used in the second-order injection method is different from that produced by the zero-order injection. These changes in the equilibrium state agree better with theory [12].

1.2. *Background*

In the literature, papers that discuss particle injection and loading are usually focused on reducing macroparticle induced noise in PIC simulations by careful ordering of the particles in position and velocity space, called a *quiet start*. These ordering methods are first attributed to J. A. Byers in 1970 [13] and discussed and reviewed in 1980 by Denavit and Kruer [14]. While quiet starts are not the main focus of this paper they are applicable to the inversion algorithms presented here. Quiet starts are most useful in transient or instability growth simulations when low initial noise levels are important. Indeed, the intent of a quiet start is to postpone the development of thermal fluctuations in a simulation; it cannot prevent them. The following is a brief literature review of the different methods proposed for reducing the injection and loading noise in a PIC simulation.

In 1971, Morse and Nielson [15] readjusted randomly picked velocities so that the first moment (momentum) in each cell was equal to that of the initial distribution function. This technique was expanded upon by Gitomer in the same year [16] so that both the first and the second (energy) moments were correct. These methods yield a small reduction of noise but do not have the numerical problems of the following schemes.

In 1972, Denavit [17] used a hybrid PIC model and numerical solutions of the Vlasov equation in order to reduce noise in phase space. The distribution function is reconstructed periodically, as in Vlasov solutions, by a local averaging operation in phase space. This distribution function is inverted to obtain particle velocities using the variable weights. The analysis of this variable weighting method was later published by Gitomer and Adam [18].

Gitomer and Adam [18] discussed two methods; one assigned the velocities in an ordered manner and the other varied the particles weights. The first method, ordered velocity initialization, can give rise to a multibeam instability, which was predicted by Dawson [19]. The second method, variable particle weights, leads to a multibeam instability [18] and was later found to have a nonphysical heat exchange between the particles of different numerical weight [20]. Both of these methods are still useful because these nonphysical (numerical) problems are significant only in certain parameter regimes, which are discussed in the articles.

In 1988, Lawson [21] modified the ordered velocity sequencing of Gitomer and Adam [18] for flux injection, but cooling occurs due to the particles becoming disordered, giving up energy to the electric field fluctuations.

Inverting a velocity distribution requires an understanding of the ramifications of using macroparticles. Denavit [22] discusses the effect of discrete particles and discrete particle position and velocity loading for electrostatic and Darwin (magnetoinductive) field solves. For simulations that reach a steady state, an understanding of the effects of discrete particles and fluctuations is important because quiet starts do not reduce the fluctuations in steady state. Matsuda and Okuda [23] studied numerically and analytically the discrete particle effect on drag and diffusion of particles in velocity space. Loading and injection are mentioned briefly in these papers.

1.3. *Cumulative Distribution Functions*

The cumulative distribution function maps the distribution function variable (e.g., velocity) to a uniformly distributed set of numbers, R, typically normalized between 0 and 1. The cumulative distribution, $F(v)$ is

$$R = F(v) \equiv \frac{\int_{v_{cl}}^{v} f(v')\,dv'}{\int_{v_{cl}}^{v_{cu}} f(v')\,dv'}, \tag{1}$$

where v_{cl} and v_{cu} are the lower and upper cutoffs, respectively. The method of choosing R in the cumulative distribution function may affect the results of a simulation. The two methods used to choose R are pseudorandom or subrandom sequences[2]. Cumulative distributions

[2] A random sequence should uniformly fill in the domain at a rate of $1/\sqrt{N}$ where N is the number of points. Many physical processes will fill out a distribution function at this rate. In natural processes, N may be much larger than can be used in a simulation because of the limitations in memory and speed of the computer. There are other ways to choose R in order to fill in the domain more quickly. If the number of particles to be used is known in advance, the R's can be distributed uniformly. There are also subrandom sequences in which the relative error decreases at least as fast as $1/N$. Obviously, if the simulation results depend upon the physical fluctuation scaling law of $1/\sqrt{N}$, this difference will affect the results. An ideal quiet start would maintain the same scaling

that cannot be inverted analytically will be inverted numerically. The numerical inversion consists of making a second-order table and then interpolating the cumulative distribution function between table values. The whole process is second-order accurate in table value spacing.

1.4. *Definitions and Scaling*

Due to the ubiquity of the Maxwellian–Boltzmann velocity distribution, emphasis has been on loading Maxwellian distributions and injecting Maxwellian fluxes. The methods presented here are also applicable to any separable continuous distribution function, $f(v_x) f(v_y) f(v_z)$, with continuous and finite first and second derivatives. (Relativistic velocities have the added complication that the velocity components couple; approximate decoupling is achieved when the drift is large compared to the thermal velocity.) The Maxwell–Boltzmann distribution is given by

$$f(E) \propto \exp\left(\frac{-E}{kT}\right), \tag{2}$$

where k is Boltzmann's constant, E is the kinetic energy, and T is the temperature.

A nonrelativistic anisotropic Maxwellian distribution can be broken into three separate one-dimensional distributions and then inverted independently; this separation is properly exploited whenever practicable in this paper. For this work, the definition of the ith component of thermal velocity, v_{ti}, is taken to be

$$\frac{1}{2} m v_{ti}^2 = \frac{1}{2} k T_i. \tag{3}$$

The thermal velocity of an isotropic Maxwellian is given by

$$\frac{1}{2} m v_t^2 = \frac{3}{2} kT, \quad \text{with } v_t^2 = \sum_{i=1}^{3} v_{ti}^2. \tag{4}$$

Using the 1d definitions (Eq. (3)) a Maxwellian distribution is defined as:

$$f(v) \propto e^{-v^2/2v_{ti}^2}. \tag{5}$$

With these definitions the standard deviation of the distribution is v_{ti}. The velocities in this paper are normalized by $v_{ti}\sqrt{2}$ (the normalized velocity is equal to $v/(v_{ti}\sqrt{2})$), in order to simplify the equations, unless otherwise noted.

law but reduce the proportionality constant. The scaling laws of many quiet start methods are not published; in the few published cases the scaling deviates from $1/\sqrt{N}$. Using subrandom sequences is a variation of a quiet start. Pseudorandom numbers can be replaced with a subrandom or uniform number sequence, if numerical noise suppression is desired. Bratley and Fox [24] provide a review of (and references for) some of the more notable examples of subrandom sequences, including bit-reversed, Fibonacci, and Sobol. One note of caution about using ordered subrandom sequences is that successive numbers fill in gaps left previously in the sequences. A consequence of the sequence being ordered is that the inversion must start at the beginning. For example, numbers 1 through N will fill in the space uniformly but numbers $N/2$ through N will not. There are also consequences to the spectral content of the fluctuations. Pseudorandom sequences approach a uniform spectrum in Fourier space as N increases, but subrandom sequences will have a finite number of Fourier components. The choice of sequence is a fine tuning knob that the reader may adjust for the application.

2. LOADING A MAXWELLIAN VELOCITY DISTRIBUTION

The cumulative distribution function for a Maxwellian with both an upper, v_{cu}, and a lower, v_{cl}, velocity cutoff in one dimension is

$$F(v) = R = \frac{\int_{v_{cl}}^{v} e^{-v'^2} dv'}{\int_{v_{cl}}^{v_{cu}} e^{-v'^2}\, dv'} = \frac{\mathrm{Erf}(v) - \mathrm{Erf}(v_{cl})}{\mathrm{Erf}(v_{cu}) - \mathrm{Erf}(v_{cl})}, \tag{6}$$

where Erf is the error function. Solving for v:

$$v = \mathrm{Erf}^{-1}(R\,\mathrm{Erf}(v_{cu}) + (1 - R)\mathrm{Erf}(v_{cl})). \tag{7}$$

With present day digital computers it is faster to tabulate Eq. (6) and then interpolate within the table than to calculate Eq. (7) for each particle. A table of uniform probability [25] was calculated using the bisection method (the guaranteed convergence of the bisection method outweighed the increased convergence of methods such as Newton and secant).

A Maxwellian distribution can also be inverted using the Box–Muller method [26] and is presented here because it is easily modified for upper cutoff Maxwellian distributions. This method transforms a uniformly distributed pseudorandom number to a Maxwellian distribution. First, pick two pseudorandom numbers, ν_1 and ν_2, with $0 < \nu_1, \nu_2 \leq 1$. Because of the requirement that ν_1 and ν_2 be uncorrelated, this method is not well suited for use with subrandom sequences. The sum of the squares, $R^2 = \nu_1^2 + \nu_2^2$, is formed. If $R^2 > 1$, then reject ν_1 and ν_2, otherwise accept them. The Box–Muller transformation returns two normal deviates,

$$v_1 = \nu_1\sqrt{-\frac{\ln(R^2)}{R^2}} \quad \text{and} \quad v_2 = \nu_2\sqrt{-\frac{\ln(R^2)}{R^2}}. \tag{8}$$

This is equivalent to inverting $vf(v)$ (see Section 3) and then randomly choosing an angle to obtain two velocities. To show that this is the correct transformation between uniform deviates and the distribution function, take the determinant of the Jacobian of the transformation:

$$\frac{\partial(\nu_1, \nu_2)}{\partial(v_1, v_2)} = \begin{vmatrix} \frac{\partial\nu_1}{\partial v_1} & \frac{\partial\nu_1}{\partial v_2} \\ \frac{\partial\nu_2}{\partial v_1} & \frac{\partial\nu_2}{\partial v_2} \end{vmatrix}. \tag{9}$$

If the Jacobian is a product of a function of v_1 and v_2 alone, then each velocity is independently distributed according to each function in the product. The Jacobian for the Box–Muller transform, Eq. (8), is

$$\frac{\partial(\nu_1, \nu_2)}{\partial(v_1, v_2)} = -\exp\left(-v_1^2\right)\exp\left(-v_2^2\right). \tag{10}$$

The domain of the pseudorandom number can be scaled so that that it transforms to a cutoff Maxwellian distribution. Since the Box–Muller transformation is equivalent to inverting $vf(v)$ and then randomly choosing an angle (Section 3), it can be solved for v with an upper cutoff and then an angle can be chosen at random to obtain two velocities. Let

$R_{cu} = e^{v_{cu}^2}$, where v_{cu} is the upper velocity cutoff. The transformation for an upper cutoff Maxwellian is:

$$\begin{aligned} v_1 &= \frac{\nu_1}{R}\sqrt{v_{cu}^2 - \ln(R^2 + (1 - R^2)R_{cu})}, \quad \text{and} \\ v_2 &= \frac{\nu_2}{R}\sqrt{v_{cu}^2 - \ln(R^2 + (1 - R^2)R_{cu})}. \end{aligned} \tag{11}$$

This method is not easily expanded for a lower cutoff; hence, the transformation for a lower cutoff must be done numerically.

When the desired distribution is a Maxwellian with a nonrelativistic drift component, the drift, $\mathbf{v}_0$, can be added linearly, $\mathbf{v}' = \mathbf{v}_0 + \mathbf{v}$. For the relativistic case, the addition is nonlinear; see, for example, Smith [27, p. 117].

2.1. *Time-Centering of Loaded Particles*

Many PIC codes use a leap-frog integrator in order to integrate the equations of motion for the particles. In the leap-frog algorithm, the velocities and positions are offset by half a time step, often called time-centering. In Section 2.7 of Birdsall and Langdon [10] a second-order method for time-centering particles at $t = 0$ is presented for uniform plasma with the electron having an initial position of the form $x_i(t = 0) = x_{i0} + x_{i1}\cos(k_s x_{i0})$ with zero velocity; $x_{i1}\cos(k_s x_{i0})$ is a perturbation of equally spaced positions (x_{i0}). This method is first-order when $d\mathbf{a}/dt \neq 0$, when $\mathbf{B} \neq 0$, or when $\mathbf{v}_0 \neq 0$, as shown below. This section will present a generalized second-order method for time-centering particles. In this paper the velocities have been arbitrarily (but computationally efficient) chosen to lag the positions at the end of a time step. Time-centering will also be discussed for particle injection in Section 3.1, applying the same formalism.

To formulate a numerical PIC injection or loading method, one can begin with the Lorentz equation of motion,

$$m\ddot{\mathbf{x}}(t) = q\mathbf{E}(\mathbf{x}(t), t) + q\mathbf{v}(t) \times \mathbf{B}(\mathbf{x}(t), t), \tag{12}$$

which can be rewritten as

$$\ddot{\mathbf{x}}(t) = \mathbf{e}(\mathbf{x}(t), t) + \Omega(\mathbf{x}(t), t)\mathbf{v}(t) \times \hat{\mathbf{b}}(\mathbf{x}(t), t), \tag{13}$$

where $\mathbf{e} = q\mathbf{E}/m$, $\Omega = q|\mathbf{B}|/m$, and $\hat{\mathbf{b}} = \mathbf{B}/|\mathbf{B}|$. Continuous quantities will be denoted by ( ), e.g., $\mathbf{E}(t)$ is the electric field as a continuous function of time.

Before discussing the half-step needed for loading, a review of the standard leap-frog integrator with the magnetic term centered by averaging [10] is presented;

$$\begin{aligned} \mathbf{x}_{n+1} - \mathbf{x}_n &= \mathbf{v}_{n+1/2}\Delta t, \quad \text{and} \\ \mathbf{v}_{n+1/2} - \mathbf{v}_{n-1/2} &= \mathbf{e}_n\Delta t + \tan(\Omega_n\Delta t/2)\big(\mathbf{v}_{n-1/2} + \mathbf{v}_{n+1/2}\big) \times \hat{\mathbf{b}}_n, \end{aligned} \tag{14}$$

where n indicates the time after n time steps, $t_n = n\Delta t$. Loading is usually done at $t = 0$ $(n = 0)$, but to maintain generality so that we can use these results for injecting particles from a boundary (Section 3.1) we will center particles at time t_n (time step n). Discrete quantities are denoted by subscripts, e.g., $\mathbf{E}_n$ is the electric field at time step n. To find the order of accuracy these equations may be combined and Taylor expanded around t_n. The acceleration truncation error, $\mathcal{E}$, will be defined as the continuous acceleration equation, Eq. (13), minus the discrete acceleration equation, Eq. (14), at time, t_n:

$$\begin{aligned}\mathcal{E} &= \ddot{\mathbf{x}}(t_n) - (\mathbf{e}(\mathbf{x}(t_n), t_n) + \Omega(\mathbf{x}(t_n), t_n)\mathbf{v}(t_n) \times \hat{\mathbf{b}}(\mathbf{x}(t_n), t_n)) \\ &\quad - \left(\frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{\Delta t^2} - \left(\mathbf{e}_n + \frac{\tan(\Omega_n \Delta t/2)(\mathbf{x}_{n+1} - \mathbf{x}_{n-1}) \times \hat{\mathbf{b}}_n}{\Delta t^2}\right)\right).\end{aligned} \tag{15}$$

Assuming that the discrete and continuous fields agree at integer time steps, the error may be simplified at the nth time step to:

$$\begin{aligned}\mathcal{E} &= \ddot{\mathbf{x}}(t_n) - \frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{\Delta t^2} \\ &\quad + \left(\frac{\tan(\Omega_n \Delta t/2)(\mathbf{x}_{n+1} - \mathbf{x}_{n-1})}{\Delta t^2} - \Omega(\mathbf{x}(t_n), t_n)\dot{\mathbf{x}}(t_n)\right) \times \hat{\mathbf{b}}_n.\end{aligned} \tag{16}$$

Taylor expanding in time around t_n and explicitly writing the second-order error term, we obtain

$$\mathcal{E} = -\frac{1}{12}[d_{tttt}\mathbf{x}(t) - \Omega(\mathbf{x}(t), t)(\Omega^2(\mathbf{x}(t), t)\dot{\mathbf{x}}(t) + 2d_{ttt}\mathbf{x}(t)) \times \hat{\mathbf{b}}_n]\Delta t^2 + \mathcal{O}(\Delta t^4), \tag{17}$$

where $d_{t\times n}$ is shorthand for $(d/dt)^n$ and $\mathcal{O}(\Delta t^n)$ is a vector where the lowest order component is order n. Equation (17) shows that the leap-frog integration is globally second-order accurate. From Eq. (17) and the velocity update of Eq. (14) one might expect that the velocity would have a local truncation error of $\mathcal{O}(\Delta t^3)$; however, the velocity is only $\mathcal{O}(\Delta t^2)$. As shown here by Taylor expanding the position update of Eq. (14) and showing the second-order term explicitly:

$$\mathbf{v}_{n+1/2} = \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} = \mathbf{v}((n + 1/2)\Delta t) + \frac{1}{24}d_{ttt}\mathbf{x}(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3). \tag{18}$$

It will be shown that the second-order loading (and later the injection) method will have the same $\mathcal{O}(\Delta t^2)$ local truncation error as shown in Eq. (18). The position, as expected, has a local truncation of $\mathcal{O}(\Delta t^4)$, which can be shown from Eqs. (17) and (14). It would be incorrect to assume that a fourth-order position injection and any second-order velocity injection would comprise a second-order global method. In this case, the second-order error in Eq. (18) cancels when it is differenced with $\mathbf{v}_{n-1/2}$ to form the acceleration:

$$\begin{aligned}\mathbf{a}_n &= \frac{\mathbf{v}_{n+1/2} - \mathbf{v}_{n-1/2}}{\Delta t} \\ &= \frac{\mathbf{v}((n+1/2)\Delta t) + \frac{1}{24}d_{ttt}\mathbf{x}(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3) - \left(\mathbf{v}((n-1/2)\Delta t) + \frac{1}{24}d_{ttt}\mathbf{x}(t_n)\Delta t^2 + \mathcal{O}(\Delta t^3)\right)}{\Delta t} \\ &= \frac{\mathbf{v}((n + 1/2)\Delta t) - \mathbf{v}((n - 1/2)\Delta t)}{\Delta t} + \mathcal{O}(\Delta t^2).\end{aligned} \tag{19}$$

This shows how the second-order truncation error in velocity cancels when combined to form the acceleration. This seemingly trivial cancellation is the essence of the difficulty in forming a second-order truncation error.

The proper way to obtain the truncation error of a loading or injection method is to substitute the position and velocity into Eq. (16), the leap-frog error equation:

$$\begin{aligned}\mathcal{E} &= \ddot{\mathbf{x}}(t_n) - \frac{\mathbf{x}_{n+1} - \mathbf{x}_n - \mathbf{v}_{n-1/2}\Delta t}{\Delta t^2} \\ &\quad + \left(\frac{\tan(\Omega_n \Delta t/2)\big(\mathbf{x}_{n+1} - \mathbf{x}_n + \mathbf{v}_{n-1/2}\Delta t\big)}{\Delta t^2} - \Omega\left(\mathbf{x}(t_n), t_n\right)\dot{\mathbf{x}}(t_n)\right) \times \hat{\mathbf{b}}_n. \end{aligned} \tag{20}$$

Here $\mathbf{x}_n$ is the position at time step n and $\mathbf{v}_{n-1/2}$ is the velocity at time step $n - 1/2$, the first position and velocity after time-centering. For Eq. (20) to be second-order, the local truncation error in position of the time-centered particle, $\mathbf{x}_n$, must be at least fourth-order, as in standard leap-frog. It is important that the position have the same truncation error as leap-frog because the particle position, x_n, is used in the field solve at time step n. For Eq. (20) to be second-order accurate, the velocity of the time-centered particle, $\mathbf{v}_{n-1/2}$, must have the same second-order local truncation error as leap-frog (Eq. (18)) because the truncation error cancels as the particle integrator advances (Eq. (19)). Also, it is necessary to have the velocity error second-order, in order to have accurate current collection for an electromagnetic field solve. In the analysis of the following methods the truncation error will be kept through second-order in the Lorentz equation error (Eq. (20)) and third-order in $\mathbf{v}_{n-1/2}$.

We will analyze two loading methods. The first loading method presented is from Birdsall and Langdon [10] with an added magnetic field. The second method is a second-order method for general time-dependent fields.

2.1.1. Half time step push for particle loading. First we will analyze the method of time-centering presented in Sections 2–7 of Birdsall and Langdon [10] in which the velocity is pushed back a half-step. This method may be written in the form

$$\mathbf{v}_{n-1/2} - \mathbf{v}_n = -\mathbf{e}\frac{\Delta t}{2} - \tan(\Omega \Delta t/4)\big(\mathbf{v}_{n-1/2} + \mathbf{v}_n\big) \times \hat{\mathbf{b}}. \tag{21}$$

Substituting Eq. (21) into the truncation error equation, Eq. (20), then Taylor expanding and applying the chain rule we obtain a first-order accurate method for general time-dependent fields. Due to the complexity and length of the analytic expression for the truncation error for this method with time-dependent fields, the expression is not shown here. The local truncation error for the half-step velocity, $\mathcal{E}_v$, is second-order.

The above method is first order in general; however, it is second-order under limited conditions. For an illustration of when this method is second-order we will choose the magnetic field to be zero; then the truncation error of the above method can be written as

$$\mathcal{E} = -\frac{1}{6}\left(\mathbf{v}(t_n) \cdot \nabla\mathbf{e}(t_n) + \frac{\partial \mathbf{e}(t_n)}{\partial t}\right)\Delta t + \mathcal{O}(\Delta t^2). \tag{22}$$

This truncation error (Eq. (22)) will be second-order if the velocity ($\mathbf{v}(t_n)$) or $\nabla\mathbf{e}(t_n)$ is zero or $\nabla\mathbf{e}(t_n)$ is perpendicular to $\mathbf{v}(t_n)$ and the derivative of the electric field in time is zero. These assumptions are true for the loading of the perturbation in Birdsall and Langdon (Sections 2–7); thus their loading scheme is second-order accurate for that special case.

The velocity error for zero magnetic field is

$$\mathcal{E}_v = \mathbf{v}((n-1/2)\Delta t) - \mathbf{v}_{n-1/2} = -\frac{1}{24}d_{ttt}\mathbf{x}(t_n)\Delta t^2 - \frac{1}{6}\left(\mathbf{v}(t_n)\cdot\nabla\mathbf{e}(t_n) + \frac{\partial\mathbf{e}(t_n)}{\partial t}\right)\Delta t^2. \tag{23}$$

Comparing Eq. (23) with the error in the leap-frog velocity, Eq. (18), we see the second term on the right does not cancel, leaving the velocity with a second order local error term, which is globally first order in the general case (per Eq. (22)).

2.1.2. General second order time-centering for particle loading. The particle injection method discussed later (Section 3.1) is a generalization of a second-order loading method where all the particle velocities and positions are known at time t_n. (The injection method shown later is for particles that are injected at an arbitrary time between time steps.) To save space we will not derive the specific results here and the more general results later, but simplify the more general results from Section 3.1. For the second-order injection method past time step field values are used. However, for the loading previous field values are not available; instead the time derivative of the fields must be given as an initial condition to at least first-order. Therefore, the fields needed for the half step push (a simplification of Eq. (49), with $f = 0$ and the time index on the fields shifted from $n-1$ to n) are

$$\begin{aligned}
\Omega_v &= \Omega(\mathbf{x}_n, n) - \frac{3}{4}\frac{\partial\Omega(\mathbf{x}_n, n)}{\partial t}\Delta t - \frac{1}{4}\mathbf{v}_n\cdot\nabla\Omega(\mathbf{x}_n, n)\Delta t,\\
\hat{\mathbf{b}}_v &= \hat{\mathbf{b}}(\mathbf{x}_n, n) - \frac{3}{4}\frac{\hat{\mathbf{b}}(\mathbf{x}_n, n)}{\partial t}\Delta t - \frac{1}{4}\mathbf{v}_n\cdot\nabla\hat{\mathbf{b}}(\mathbf{x}_n, n),\\
\mathbf{e}_v &= \mathbf{e}(\mathbf{x}_n, n) - \frac{3}{4}\frac{\mathbf{e}(\mathbf{x}_n, n)}{\partial t}\Delta t - \frac{1}{4}\mathbf{v}_n\cdot\nabla\mathbf{e}(\mathbf{x}_n, n)\Delta t.
\end{aligned} \tag{24}$$

The velocity half-step for the load (a simplification of Eq. (51)) is

$$\begin{aligned}
\mathbf{v}' - \mathbf{v}_n &= -\frac{1}{2}\mathbf{e}_v\Delta t - \tan(\Omega_v\Delta t/4)(\mathbf{v}' + \mathbf{v}_n)\times\hat{\mathbf{b}}_v,\\
\mathbf{v}_{n-1/2} - \mathbf{v}' &= \frac{\Delta t^2}{24}\Bigg(\mathbf{e}\times\hat{\mathbf{b}}\Omega/\Delta t + (\hat{\mathbf{b}}(\hat{\mathbf{b}}\cdot\mathbf{v}_n) - \mathbf{v}_n)\Omega^2\\
&\quad + \mathbf{v}_n\cdot(\nabla\mathbf{e} + (\nabla\Omega\mathbf{v}_n)\times\hat{\mathbf{b}} + \Omega\mathbf{v}_n\times\nabla\hat{\mathbf{b}})\\
&\quad + \frac{\partial\mathbf{e}(\mathbf{x}_n, n)}{\partial t} + \Omega\mathbf{v}_n\times\frac{\partial\hat{\mathbf{b}}(\mathbf{x}_n, n)}{\partial t} + \frac{\partial\Omega(\mathbf{x}_n, n)}{\partial t}\mathbf{v}_n\times\hat{\mathbf{b}}\Bigg).
\end{aligned} \tag{25}$$

This simplified version of Eq. (51) is obtained because x_n does not need to be changed to be time-centered and we have used the relation:

$$\mathbf{v}_n = \frac{\mathbf{v}_{n+1/2} + \mathbf{v}_{n-1/2}}{2} + \mathcal{O}(\Delta t^2). \tag{26}$$

The local truncation error for the half-step velocity, $\mathcal{E}_v$, has $d_{ttt}\mathbf{x}(t_n)/24$ as a second-order term, which properly cancels in the leap-frog integrator. The global error of the leap-frog integrator with this time centering is second-order.

3. INJECTION OF MAXWELLIAN FLUX

The flux of a distribution is used to inject particles from a boundary in the system. The boundary may represent a physical interface or a computational box to reduce the problem to a practical size. Since the emission will only occur for velocities moving away from the boundary, the coordinate system is chosen so that the positive velocities are moving away from the boundary. A point that is often confused is whether a Maxwellian flux or a drifting Maxwellian flux model should be applied. The electron flux from a hot (thermionic) cathode is a nondrifting Maxwellian flux, which has an average velocity, $v_{\text{ave}} = \int_0^\infty v f(v)\, dv$. To model electrons that came from a Maxwellian source and were accelerated (i.e., by an electric field), a Maxwellian flux with no drift should be inverted (as inverting the distribution at the source) and then a velocity corresponding to the acceleration should be added (relativistically or nonrelativistically); a drifting Maxwellian flux should not be used.

A drifting Maxwellian flux is used to inject a thermalized drifting plasma. The drifting plasma could be a beam that has thermalized due to Coulomb collisions, normally a long time (distance) for beams. Also it could be used when the simulation frame is moving through a resting plasma, for example, a space tether or spacecraft in the rest frame of the space tether or spacecraft. The computational boundary emitting the flux represents an infinite plasma.

The normal to the emitting surface defines the direction of the flux at the surface; the other two directions are not fluxes but distributions (as treated in Section 2). For example, in Fig. 2 the emission flux direction normal to the surface is in the x_1 direction; in the transverse directions, x_2 and x_3, the distribution is inverted (Section 2). The direction of the velocity of the inverted flux is normal to the wall with a magnitude denoted by v.

A Maxwellian flux without a drift at the emitting surface can be inverted in closed form, with or without cutoffs. The cumulative distribution function is

$$R' = F(v) = \frac{\int_{v_{cl}}^{v} v' e^{-v'^2}\, dv'}{\int_{v_{cl}}^{v_{cu}} v' e^{-v'^2}\, dv'}, \tag{27}$$

where R' is a uniformly distributed pseudorandom number, $0 \leq R' \leq 1$. Equation (27) can be integrated and then solved for velocity,

$$v = \sqrt{v_{cl}^2 + v_{cu}^2 - \ln\big(R' \exp\big(v_{cl}^2\big) + (1 - R') \exp\big(v_{cu}^2\big)\big)}. \tag{28}$$

For a distribution without cutoffs, this reduces to

$$v = \sqrt{-\ln(1 - R')} = \sqrt{-\ln(R)}, \tag{29}$$

where $R = 1 - R'$ is a uniformly distributed pseudorandom number, $0 < R \leq 1$; $R = 0$ is removed from the domain because an infinite velocity is not easily representable by a computer.

For a Maxwellian flux with a drift, the velocity cannot be written in closed form. Instead, the flux distribution must be inverted numerically. The cumulative distribution for a

Maxwellian flux with drift and upper and lower cutoffs is

$$R = F(v) = \frac{\int_{v_{cl}}^{v} v' e^{-(v'-v_0)^2}\, dv'}{\int_{v_{cl}}^{v_{cu}} v' e^{-(v'-v_0)^2}\, dv'}. \tag{30}$$

This can be integrated:

$$R = F(v) = \frac{e^{(v_{cl}-v_0)^2} - e^{(v-v_0)^2} + \sqrt{\pi} v_0 \operatorname{Erf}(v - v_0) - \sqrt{\pi} v_0 \operatorname{Erf}(v_{cl} - v_0)}{e^{(v_{cl}-v_0)^2} - e^{(v_{cu}-v_0)^2} + \sqrt{\pi} v_0 \operatorname{Erf}(v_{cu} - v_0) - \sqrt{\pi} v_0 \operatorname{Erf}(v_{cl} - v_0)}. \tag{31}$$

For $v_{cu} \to \infty$ and $v_{cl} \to 0$,

$$R = F(v) = \frac{e^{-v_0^2} - e^{-(v-v_0)^2} + \sqrt{\pi}\, v_0(\operatorname{Erf}(v - v_0) + \operatorname{Erf}(v_0))}{e^{-v_0^2} + \sqrt{\pi}\, v_0(1 + \operatorname{Erf}(v_0))}. \tag{32}$$

The inversion of these equations to obtain $v = F^{-1}(f)$ must be done numerically. The numerical inversion was the same as outlined in Section 2. The integration for an arbitrary distribution can also be done numerically. The effects of discretizing the distribution are analyzed in Appendix A.

For a Maxwellian flux in which the drift is much larger than the thermal velocity, this method can be approximated by a drifting Maxwellian as done in Section 2. A drifting Maxwellian is easily inverted using the Box–Muller transform, whereas for a drifting Maxwellian flux the inversion must be done numerically; therefore, it is advantageous to use a drifting Maxwellian wherever appropriate. Most of the error (Maxwellian flux minus drifted Maxwellian) occurs at $v_0 \pm 1/\sqrt{2}$ as shown in Fig. 3 with $v_0 = 1, 3, 6,$ and 10. The largest positive error occurs at

$$v_{\text{pos}} = v_0 + \frac{1 - \sqrt{1 + 2e^{2v_0^2}\pi(1 + \operatorname{Erf}(v_0))^2}}{2e^{v_0^2}\sqrt{\pi}\,(1 + \operatorname{Erf}(v_0))}, \tag{33}$$

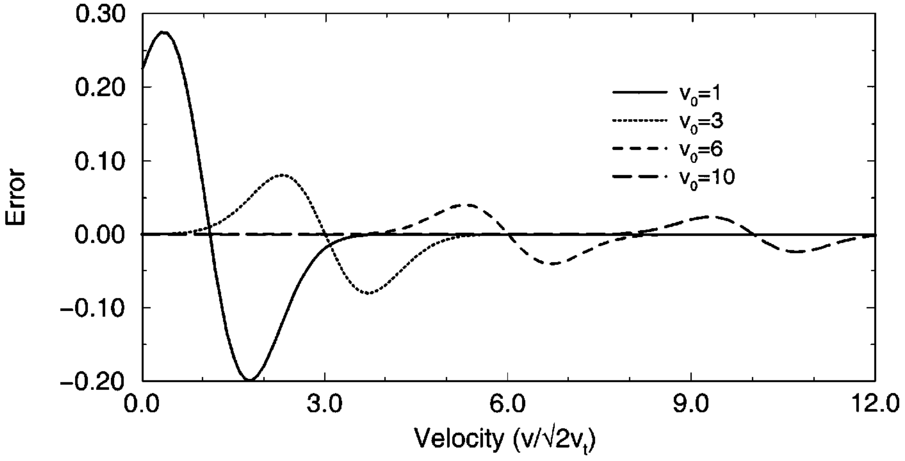

FIG. 3. The absolute error, Maxwellian flux minus the flux of a drifting Maxwellian, as a result of using a drifting Maxwellian instead of a drifting Maxwellian flux for $v_0 = 1, 3, 6,$ and 10 (v_0 normalized by $\sqrt{2}v_{ti}$).

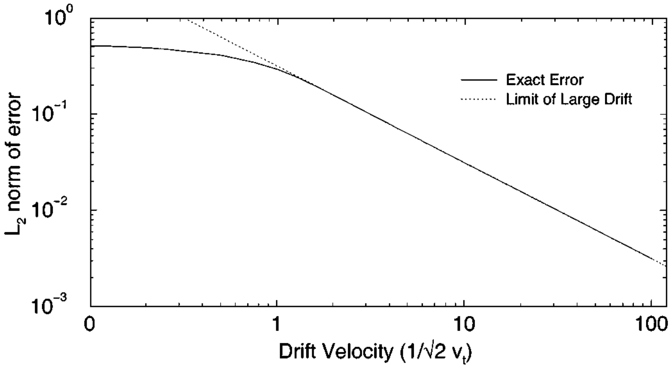

FIG. 4. The integrated absolute error squared (L_2 norm) as a result of using a drifting Maxwellian instead of a drifting Maxwellian flux.

and the largest negative error occurs at

$$v_{\text{neg}} = v_0 + \frac{1 + \sqrt{1 + 2e^{2v_0^2}\pi(1 + \text{Erf}(v_0))^2}}{2e^{v_0^2}\sqrt{\pi}\,(1 + \text{Erf}(v_0))}. \tag{34}$$

An analytic expression for the integrated RMS (L_2 norm) error is:

$$\|\text{error}\|_{L_2} = \frac{a}{2\left(a + v_0 e^{v_0^2}\right)}\sqrt{-8ae^{v_0^2} - 4v_0 + b\left(4a^2 + e^{2v_0^2}\right)}, \tag{35}$$

where

$$a = \frac{1}{\sqrt{\pi}(1 + \text{Erf}(v_0))} \quad \text{and} \quad b = \sqrt{2\pi}(1 + \text{Erf}(\sqrt{2}v_0)). \tag{36}$$

In the limit $v_0 \gg 1$ the $\|\text{error}\|_{L_2} = 1/(2\sqrt[4]{2\pi}v_0)$. The integrated RMS error is shown as a function of v_0 in Fig. 4. Also, a Maxwellian with a relativistic drift and nonrelativistic thermal velocity may replace a drifting Maxwellian flux through the use of the relativistic velocity addition.

3.1. *Time-Centering Injected Particles*

Time-centering of velocity, **v**, and position, **x**, when loading particles is common; however, time-centering injected particles emitted from the boundaries is not. Figure 5 conveys the problem graphically. The authors have found two papers that have some discussion of time-centering at the boundaries. Lawson [28] states the need for particles to be

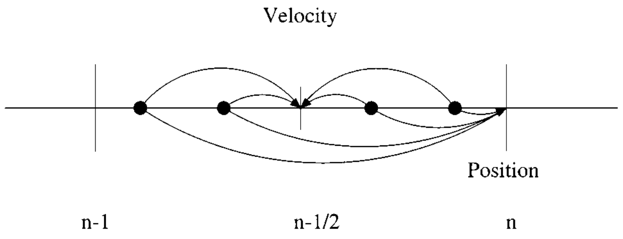

FIG. 5. To inject a particle in a time-centered manner in the interval $(n-1)\Delta t < t \leq n\Delta t$, the particle needs to be advanced from $\mathbf{x}(t)$ and $\mathbf{v}(t)$ to $\mathbf{x}(t = n\Delta t)$ and $\mathbf{v}(t = (n - \frac{1}{2})\Delta t)$.

time-centered, but does not provide a method. Schwager and Birdsall [29] use an energy-conserving reflection algorithm which time-centers the particle positions and velocities (due to Lawson), but is not second-order accurate. Many simulations are not sensitive to the shape of the emitted distribution function. However, it has been observed recently that in some cases time-centering is critical, such as in cross–field diodes and virtual cathode diodes [2, 3]. Inaccurate emission of a distribution function can be thought of as changing the shape of the emitted distribution function, which varies during a time step since the error is a function of when during the time step the particle is injected. The difference between a non-time-centered distribution and a time-centered distribution is at most the change in the velocity and position (the placement of the emitter) of a leap-frog advance. In other word, if the position $(\mathbf{x}_{n-f})$ and velocity $(\mathbf{v}_{n-f})$ are used in the leap-frog advance (Eq. (14)) as $(\mathbf{x}_n)$ and $(\mathbf{v}_{n-1/2})$ the change in position to $(\mathbf{x}_{n+1})$ and $(\mathbf{v}_{n+1/2})$ bounds the changes that any of the time-centering methods in this paper generate during the time-centering procedure. If this difference in distribution function is unimportant to the results of the simulation the accuracy of the emission can be disregarded.

In this section, a second-order accurate method is derived for particle injection at an arbitrary time with time-dependent, inhomogeneous fields. Derivation of the methods and an accuracy analysis are in Section 3.2. The performance (computer speed) is discussed in Section 3.3. Comparison to analytic test problems and a self-consistent simulation are in Section 3.4.

3.2. *Injection Methods*

To formulate a numerical PIC injection method, begin with the Lorentz equation of motion, Eq. (12), just as in the time-centering of loaded particles (Section 2.1). It will be shown that the second-order injection method will have the same $\mathcal{O}(\Delta t^2)$ local truncation error in $(\mathbf{v}_{n-1/2})$ (as shown in Eq. (18)) so that this error cancels when the particle integrator advances (as shown in Eq. (19)). The position, as expected, has a local truncation of $\mathcal{O}(\Delta t^4)$, which can be shown from Eqs. (17) and (14). It is incorrect to assume that a fourth-order position injection and any second-order velocity injection would comprise a second-order global method because of the cancellation shown in Eq. (19). The order of the injection method is obtained by substituting the injection velocity and position algorithm into Eq. (20). For Eq. (20) to be second-order, the local truncation error in position of the injected particle, $\mathbf{x}_n$, is fourth-order, as in standard leap-frog. It is important that the position has the same truncation error as leap-frog because the particle position, x_n, is used in the field solve at time step n. Also, the velocity of the injected particle, $\mathbf{v}_{n-1/2}$, must have the same second-order local truncation error as leap-frog in order to cancel the second-order error of Eq. (18) when substituted into Eq. (19). It is necessary to have the velocity error second-order in order to have accurate current collection for an electromagnetic field solve. In the analysis of the following methods the truncation error will be kept through second-order in the Lorentz equation error (Eq. (20)) fourth-order in $\mathbf{x}_n$ and third-order in $\mathbf{v}_{n-1/2}$. In many applications a general second-order method is not needed because requirements are relaxed; for example, uniform fields or time-independent fields.

The five injection methods to be presented were chosen for the following reasons. The method used in the plasma device codes [11] illustrates the inaccuracy of injection when the problem was observed, which will be referred to as the “simple” injection method.

The simple Boris[3] push with fractional time step is a simple but much improved first-order method. The next three methods are second-order for special cases. The modified Boris push with fractional time step is second-order for time-independent uniform fields. The fractional Boris push with field gradients is second-order for time-independent fields. For simulation parameters near these special cases the first-order error will be small. The last method, using field values from the previous time step, is second-order for the general case. Other second-order special cases may be constructed; for example, past field values can be used with the modified Boris push if a problem has time-dependent uniform fields.

3.2.1. Historical analysis of simple injection method. The first method to be examined was implemented in the suite of plasma device codes from Berkeley [11] around 1988:

$$\begin{aligned} \mathbf{x}_n - \mathbf{x}_{n-f} &= \mathbf{v}_{n-f}\, f\,\Delta t, \quad \text{and} \\ \mathbf{v}_{n-\frac{1}{2}} - \mathbf{v}_{n-f} &= \mathbf{e}(\mathbf{x}_{n-f}, n-1)\left(f - \frac{1}{2}\right)\Delta t. \end{aligned} \tag{37}$$

Here $\mathbf{x}_{n-f}$ and $\mathbf{v}_{n-f}$ are the position and velocity, respectively, at the fractional time step of injection, $n - f$, where $f = \delta t/\Delta t, 0 \leq f < 1$. The algorithm that is used to pick f makes the current as continuous as possible by injecting particles uniformly in time during a time step. Accumulated current that is a noninteger number of computer particles is carried over to the next time step. To obtain the truncation error, substitute Eq. (37) into Eq. (20), Taylor expand around the nth time step, and apply the chain rule:

$$\mathcal{E} = -\frac{1}{2}((f^2 + 2f - 1)\Omega_n \mathbf{v}_n \times \hat{\mathbf{b}}_n + f^2\mathbf{e}_n) + \mathcal{O}(\Delta t). \tag{38}$$

Equation (38) shows that this method is a zeroth-order accurate method. The position is second-order accurate:

$$\mathcal{E}_x = x(t_n) - x_n = -\frac{1}{2} f^2(\mathbf{e}_n + \Omega_n \mathbf{v}_n \times \hat{\mathbf{b}}_n)\Delta t^2 + \mathcal{O}(\Delta t^3). \tag{39}$$

The velocity is first-order accurate:

$$\mathcal{E}_v = v((n - 1/2)\Delta t) - v_{n-1/2} = -\left(f - \frac{1}{2}\right)\Omega_n \mathbf{v}_n \times \hat{\mathbf{b}}_n \Delta t + \mathcal{O}(\Delta t^2). \tag{40}$$

This method is a poor choice because the lowest-order error in the Lorentz equation is independent of Δt. One consequence of an error term independent of Δt is that it renders a commonly used error checking technique useless. This nonrigorous method for checking error for a given problem is to measure the result as a function of Δt. If the injection error dominates the truncation error, the zeroth-order error is hidden and changing Δt will not change the error even if it is unacceptable.

[3] Both Buneman's [30] and Boris's [31] push centers the magnetic term by averaging the velocity, thus obtaining the same results. However, they obtained the results by two different implementations. Boris's implementation has become the standard method and hence we refer to the method as the Boris push.

3.2.2. Simple Boris push. This method is a generalization of the Boris push using fractional time steps. The method depends only on the field values at time step $n-1$, of which the first (zeroth-order) method (called the simple injection method, Section (3.2.1)) is a subset:

$$\begin{aligned} \mathbf{v}_{n-1/2} - \mathbf{v}_{n-f} &= \mathbf{e}c_1\Delta t + \tan(\Omega c_2 \Delta t/2)\left(\mathbf{v}_{n-1/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}, \\ \mathbf{x}_n - \mathbf{x}_{n-f} &= c_3 \mathbf{v}_{n-f/2}\Delta t, \quad \text{and} \\ \mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}c_4\Delta t + \tan(\Omega c_5 \Delta t/2)\left(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}. \end{aligned} \tag{41}$$

Here the parameters $(c_1, \ldots, c_5)$ are independent of Δt, and $\mathbf{v}_{n-f/2}$ is a fictitious (not accurate) velocity at the fractional time step $n - f/2$, and $\Omega = \Omega(\mathbf{x}_{n-f}, n-1)$, $\mathbf{b} = \mathbf{b}(\mathbf{x}_{n-f}, n-1)$, and $\mathbf{e} = \mathbf{e}(\mathbf{x}_{n-f}, n-1)$. Substituting Eq. (41) into the injection truncation error equation, Eq. (20) and then Taylor expanding, applying the chain rule, and choosing the five parameters to cancel the error coefficients in the expansions results in a first-order accurate method for general time-dependent fields. Due to the complexity and length of the analytic expression for the truncation error for this method with time-dependent fields, the expression is not shown here. The truncation error for the post-injection position, $\mathcal{E}_x$, is third-order, and the truncation error for the post-injection velocity, $\mathcal{E}_v$, is second-order. Table III shows the truncation error for the three test cases that will be described later.

The method with $c_1 = c_2 = f - 1/2$, $c_3 = f$, and $c_4 = c_5 = f/2$ is rewritten as

$$\begin{aligned} \mathbf{v}_{n-1/2} - \mathbf{v}_{n-f} &= \mathbf{e}\left(f - \frac{1}{2}\right)\Delta t + \tan\left(\Omega\left(f - \frac{1}{2}\right)\Delta t/2\right)\left(\mathbf{v}_{n-1/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}, \\ \mathbf{x}_n - \mathbf{x}_{n-f} &= f\mathbf{v}_{n-f/2}\Delta t, \quad \text{and} \\ \mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}\frac{f\Delta t}{2} + \tan(\Omega f \Delta t/4)\left(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}. \end{aligned} \tag{42}$$

Thus, from Eq. (42) it is seen that this method is almost as simple as standard leap-frog and meets the minimum requirement that the error decreases with decreasing Δt.

3.2.3. Modified fractional time step Boris push. In this section, the fractional Boris push is modified to make it second-order accurate for uniform fields. To make an injection method second-order accurate, the velocity push must have an added term that is independent of the fractional time step, f, that an injected particle is pushed, but which depends on the time step for the push, Δt. This term is the lowest order truncation term from Eq. (18) $(d_{ttt}\mathbf{x}(t)\Delta t^2/24)$. As shown in Section 3.2, Eq. (19), the second-order velocity error cancels in the left hand side of the leap-frog velocity update, Eq. (14), because of symmetry. On the right hand side of Eq. (14), the second-order velocity error contributes only to the second-order acceleration error. The entire truncation term $(d_{ttt}\mathbf{x}(t)\Delta t^2/24)$ will be taken into account later; however, for this method only field terms independent of time and position will be used to construct this term. Using the normalized Lorentz equation (Eq. (13)), $d_{ttt}\mathbf{x}(t)$ for a constant field is $d_{tt}\mathbf{x}(t) \times \mathbf{B}$; this can be calculated efficiently using $(\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}\Omega/\Delta t$. Alternatively, $d_{tt}\mathbf{x}(t) \times \mathbf{B}$ can be calculated using $\mathbf{e} \times \hat{\mathbf{b}}\Omega + (\hat{\mathbf{b}}(\hat{\mathbf{b}} \cdot \overline{\mathbf{v}}) - \overline{\mathbf{v}})\Omega^2$; however, this is numerically less efficient because $\overline{\mathbf{v}}$ is not known.

The position advance, unlike the velocity push, has no cancellation of error terms due to symmetry. There is a neglected $d_{ttt}\mathbf{x}(t)$ term which is taken into account by a term $(\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}\Omega/\Delta t$, similar to the velocity update.

A second-order method for constant fields that incorporates the previously neglected velocity truncation term $(d_{ttt}\mathbf{x}(t)\Delta t^2/24)$ is:

$$\begin{aligned}
\mathbf{v}' - \mathbf{v}_{n-f} &= \mathbf{e}c_1\Delta t + \tan(\Omega c_2 \Delta t/2)(\mathbf{v}' + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}, \\
\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f} &= \mathbf{e}\Delta t + \tan(\Omega\Delta t/2)(\mathbf{v}_{n+1-f} + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}, \\
\mathbf{v}_{n-1/2} - \mathbf{v}' &= b_1(\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}\Omega\Delta t, \\
\mathbf{x}_n - \mathbf{x}_{n-f} &= c_3\mathbf{v}_{n-f/2}\Delta t + b_2\left(\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}\Omega\Delta t^2, \quad \text{and} \\
\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}c_4\Delta t + \tan(\Omega c_5 \Delta t/2)\left(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}.
\end{aligned} \tag{43}$$

Here the parameters $(c_1, \ldots, c_5)$, Ω, $\mathbf{b}$, $\mathbf{v}_{n-f/2}$, and $\mathbf{e}$ are defined as in Section 3.2.2 and parameters (b_1 and b_2) are independent of Δt. The velocity, $\mathbf{v}'$, is the final velocity in the fractional time step Boris push method (Section 3.2.2) and is now modified explicitly (all the other velocity equations of Eq. (43) are implicit) by the truncation term $(d_{ttt}\mathbf{x}(t)\Delta t^2/24)$ discussed previously. Again substituting Eq. (43) into Eq. (20) and canceling all possible terms, it is found that this method is first-order for general time-dependent fields, but is second-order for spatial uniform fields. The truncation error for the injected position, $\mathcal{E}_x$, is third-order for general time-dependent fields and fourth-order for constant fields. The truncation error for the injected velocity, $\mathcal{E}_v$, is second-order for general fields, but for uniform fields the truncation coefficient is $d_{ttt}\mathbf{x}(t_n)/24$, the same as for leap-frog (Eq. (14)). This results in a cancellation when going from the injection to the leap-frog integrator. The method is rewritten as

$$\begin{aligned}
\mathbf{v}' - \mathbf{v}_{n-f} &= \mathbf{e}\left(f - \frac{1}{2}\right)\Delta t + \tan\left(\Omega\left(f - \frac{1}{2}\right)\Delta t/2\right)(\mathbf{v}' + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}, \\
\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f} &= \mathbf{e}\Delta t + \tan(\Omega\Delta t/2)(\mathbf{v}_{n+1-f} + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}, \\
\mathbf{v}_{n-1/2} - \mathbf{v}' &= \frac{1}{24}(\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}\Omega\Delta t, \\
\mathbf{x}_n - \mathbf{x}_{n-f} &= f\mathbf{v}_{n-f/2}\Delta t + \frac{f^2}{12}\left(\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}\Omega\Delta t^2, \\
\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}\frac{f\Delta t}{2} + \tan(\Omega f\Delta t/4)\left(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}.
\end{aligned} \tag{44}$$

This method is more complex than the fractional time step Boris push.

3.2.4. Field gradient fractional time step Boris push. This method achieves second-order accuracy for time-independent fields by using the spatial derivatives of **B** and **E** that are calculated and used on the injection boundary. This method is based on the modified fractional time step Boris push of Section 3.2.3. First, the fields that are used are modified by the derivatives with different constants for the velocity and position integration:

$$\begin{aligned}
\Omega_v &= \Omega(\mathbf{x}_{n-f}, n-1) + d_1\mathbf{v}_{n-f} \cdot \nabla\Omega(\mathbf{x}_{n-f}, n-1)\Delta t, \\
\hat{\mathbf{b}}_v &= \hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1) + d_2\mathbf{v}_{n-f} \cdot \nabla\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1)\Delta t, \\
\mathbf{e}_v &= \mathbf{e}(\mathbf{x}_{n-f}, n-1) + d_3\mathbf{v}_{n-f} \cdot \nabla\mathbf{e}(\mathbf{x}_{n-f}, n-1)\Delta t, \\
\Omega_x &= \Omega(\mathbf{x}_{n-f}, n-1) + d_4\mathbf{v}_{n-f} \cdot \nabla\Omega(\mathbf{x}_{n-f}, n-1)\Delta t, \\
\hat{\mathbf{b}}_x &= \hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1) + d_5\mathbf{v}_{n-f} \cdot \nabla\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1)\Delta t, \quad \text{and} \\
\mathbf{e}_x &= \mathbf{e}(\mathbf{x}_{n-f}, n-1) + d_6\mathbf{v}_{n-f} \cdot \nabla\mathbf{e}(\mathbf{x}_{n-f}, n-1)\Delta t.
\end{aligned} \tag{45}$$

Second, for the velocity push, the truncation term $d_{ttt}\mathbf{x}(t_n)$ has the additional terms for time-independent fields denoted by:

$$\begin{aligned}\zeta_x = {} & \Omega(\mathbf{x}(t_n))d_{tt}\mathbf{x}(t_n) \times \hat{\mathbf{b}}(\mathbf{x}(t_n)) + \mathbf{v}(t_n) \cdot (\nabla\mathbf{e}(\mathbf{x}(t_n)) \\ & + \nabla\Omega(\mathbf{x}(t_n))\mathbf{v}(t_n) \times \hat{\mathbf{b}} + \Omega(\mathbf{x}(t_n))\mathbf{v}(t_n) \times \nabla\hat{\mathbf{b}}(\mathbf{x}(t_n))).\end{aligned} \tag{46}$$

Time-independent second-order solutions do exist without the new term, ζ_x, in the truncation term, $d_{ttt}\mathbf{x}(t_n)/24$, in the velocity push; however, the coefficients have a functional form $1/(f - 1/2)$ and are singular at $f = 1/2$; therefore, ζ_x will be used. Incorporating ζ_x and the fields modified by the derivatives of the fields into the previous injection method (Section 3.2.3) yields:

$$\begin{aligned}\mathbf{v}' - \mathbf{v}_{n-f} &= \mathbf{e}_v c_1 \Delta t + \tan(\Omega_v c_2 \Delta t/2)(\mathbf{v}' + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}_v, \\ \mathbf{v}_{n+1-f} - \mathbf{v}_{n-f} &= \mathbf{e}\Delta t + \tan(\Omega\Delta t/2)(\mathbf{v}_{n+1-f} + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}, \\ \mathbf{v}_{n-1/2} - \mathbf{v}' &= b_1(\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}\Omega\Delta t \\ &\quad + d_7\mathbf{v}_{n-f} \cdot (\nabla\mathbf{e} + (\nabla\Omega\mathbf{v}_{n-f}) \times \hat{\mathbf{b}} + \Omega\mathbf{v}_{n-f} \times \nabla\hat{\mathbf{b}})\Delta t^2, \\ \mathbf{x}_n - \mathbf{x}_{n-f} &= c_3\mathbf{v}_{n-f/2}\Delta t + b_2\left(\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}_x\Omega\Delta t^2, \quad \text{and} \\ \mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}_x c_4 \Delta t + \tan(\Omega_x c_5 \Delta t/2)\left(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}_x.\end{aligned} \tag{47}$$

Here the parameters (b_1 and b_2), ($c_1, \ldots, c_5$), $\mathbf{v}_{n-f/2}$, $\mathbf{v}'$, Ω, $\mathbf{b}$, and $\mathbf{e}$ are the same as defined in Section 3.2.3 and the parameters ($d_1, \ldots, d_7$) are independent of Δt. Again substituting Eq. (47) into Eq. (20) and canceling all possible terms by choosing parameters results in a second-order method for all time-independent field configurations, but a first-order method for time-dependent fields. The truncation error for the post-injected position, $\mathcal{E}_x$, is third-order for time-dependent fields and fourth-order for time-independent fields. The truncation error for the post-injected velocity, $\mathcal{E}_v$, is second-order for time-dependent fields, and the second-order term is $d_{ttt}\mathbf{x}(t_n)/24$ for time-independent field configurations. This method (with $d_7 = 1/24$) can be rewritten as:

$$\begin{aligned}\mathbf{v}' - \mathbf{v}_{n-f} &= \mathbf{e}_v\left(f - \frac{1}{2}\right)\Delta t + \tan\left(\Omega_v\left(f - \frac{1}{2}\right)\Delta t/2\right)(\mathbf{v}' + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}_v, \\ \mathbf{v}_{n+1-f} - \mathbf{v}_{n-f} &= \mathbf{e}\Delta t + \tan(\Omega\Delta t/2)(\mathbf{v}_{n+1-f} + \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}, \\ \mathbf{v}_{n-1/2} - \mathbf{v}' &= \frac{1}{24}(\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f}) \times \hat{\mathbf{b}}\Omega\Delta t \\ &\quad + \frac{1}{24}\mathbf{v}_{n-f} \cdot \left(\nabla\mathbf{e} + (\nabla\Omega\mathbf{v}_{n-f}) \times \hat{\mathbf{b}} + \Omega\mathbf{v}_{n-f} \times \nabla\hat{\mathbf{b}}\right)\Delta t^2, \\ \mathbf{x}_n - \mathbf{x}_{n-f} &= \mathbf{v}_{n-f/2} f\Delta t + \frac{f^2}{12}\left(\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}_x\Omega\Delta t^2, \quad \text{and} \\ \mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}_x\frac{f\Delta t}{2} + \tan(\Omega_x f\Delta t/4)\left(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\right) \times \hat{\mathbf{b}}_x,\end{aligned} \tag{48}$$

using $d_1 = d_2 = d_3 = (f - 1/2)/2$ and $d_4 = d_5 = d_6 = f/3$ in Eq. 45. Without a field gradient this case reduces to the modified fractional time step Boris push (Section 3.2.3).

3.2.5. General second-order method. This is a method that is second-order for general spatially and temporally varying fields. This method requires field values at a previous time as well as spatial derivatives of **B** and **E**. Defining the field values to be used in this method to be a combination of old, new, and spatial derivatives results in:

$$
\begin{aligned}
\Omega_v &= (1-t_1)\Omega(\mathbf{x}_{n-f}, n-1) + t_1\Omega(\mathbf{x}_{n-f}, n-2) + d_1\mathbf{v}_{n-f}\cdot\nabla\Omega(\mathbf{x}_{n-f}, n-1)\Delta t,\\
\hat{\mathbf{b}}_v &= (1-t_2)\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1) + t_2\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-2) + d_2\mathbf{v}_{n-f}\cdot\nabla\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1)\Delta t,\\
\mathbf{e}_v &= (1-t_3)\mathbf{e}(\mathbf{x}_{n-f}, n-1) + t_3\mathbf{e}(\mathbf{x}_{n-f}, n-2) + d_3\mathbf{v}_{n-f}\cdot\nabla\mathbf{e}(\mathbf{x}_{n-f}, n-1)\Delta t,\\
\Omega_x &= (1-t_4)\Omega(\mathbf{x}_{n-f}, n-1) + t_4\Omega(\mathbf{x}_{n-f}, n-2) + d_4\mathbf{v}_{n-f}\cdot\nabla\Omega(\mathbf{x}_{n-f}, n-1)\Delta t,\\
\hat{\mathbf{b}}_x &= (1-t_5)\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1) + t_5\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-2) + d_5\mathbf{v}_{n-f}\cdot\nabla\hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-1)\Delta t, \quad \text{and}\\
\mathbf{e}_x &= (1-t_6)\mathbf{e}(\mathbf{x}_{n-f}, n-1) + t_6\mathbf{e}(\mathbf{x}_{n-f}, n-2) + d_6\mathbf{v}_{n-f}\cdot\nabla\mathbf{e}(\mathbf{x}_{n-f}, n-1)\Delta t.
\end{aligned}
\tag{49}
$$

For the velocity injection push, the $d_{ttt}\mathbf{x}(t_n)$ truncation term for time-dependent fields contains the same terms, ζ_x (Eq. (46)), as for time-independent fields in addition to the following terms:

$$
\zeta_t = \partial_t\mathbf{e}(\mathbf{x}(t), t) + \partial_t\Omega(\mathbf{x}(t), t)\mathbf{v}(t)\times\hat{\mathbf{b}}(\mathbf{x}(t), t) + \Omega(\mathbf{x}(t), t)\mathbf{v}(t)\times\partial_t\hat{\mathbf{b}}(\mathbf{x}(t), t). \tag{50}
$$

Therefore $d_{ttt}\mathbf{x}(t_n) = d_{tt}\mathbf{x}(t)\times\mathbf{B} + \zeta_x + \zeta_t$ for general fields. Incorporating this into the previous injection method given in Eq. (47) yields:

$$
\begin{aligned}
\mathbf{v}' - \mathbf{v}_{n-f} &= \mathbf{e}_v c_1\Delta t + \tan(\Omega_v c_2\Delta t/2)(\mathbf{v}' + \mathbf{v}_{n-f})\times\hat{\mathbf{b}}_v,\\
\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f} &= \mathbf{e}\Delta t + \tan(\Omega\Delta t/2)(\mathbf{v}_{n+1-f} + \mathbf{v}_{n-f})\times\hat{\mathbf{b}},\\
\mathbf{v}_{n-1/2} - \mathbf{v}' &= b_1(\mathbf{v}_{n+1-f} - \mathbf{v}_{n-f})\times\hat{\mathbf{b}}\Omega\Delta t + d_7\mathbf{v}_{n-f}\cdot\big(\nabla\mathbf{e} + (\nabla\Omega\mathbf{v}_{n-f})\\
&\quad\times\hat{\mathbf{b}} + \Omega\mathbf{v}_{n-f}\times\nabla\hat{\mathbf{b}}\big)\Delta t^2 + t_7\big(\mathbf{e} - \mathbf{e}(\mathbf{x}_{n-f}, n-2)\\
&\quad + \Omega\mathbf{v}_{n-f}\times(\hat{\mathbf{b}} - \hat{\mathbf{b}}(\mathbf{x}_{n-f}, n-2))\\
&\quad + (\Omega - \Omega(\mathbf{x}_{n-f}, n-2))\mathbf{v}_{n-f}\times\hat{\mathbf{b}}\big)\Delta t,\\
\mathbf{x}_n - \mathbf{x}_{n-f} &= c_3\mathbf{v}_{n-f/2}\Delta t + b_2\big(\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f}\big)\times\hat{\mathbf{b}}_x\Omega\Delta t^2, \quad \text{and}\\
\mathbf{v}_{n-f/2} - \mathbf{v}_{n-f} &= \mathbf{e}_x c_4\Delta t + \tan(\Omega_x c_5\Delta t/2)\big(\mathbf{v}_{n-f/2} + \mathbf{v}_{n-f}\big)\times\hat{\mathbf{b}}_x.
\end{aligned}
\tag{51}
$$

Here the parameters (b_1 and b_2), ($c_1, \ldots, c_5$), ($d_1, \ldots, d_7$), $\mathbf{v}_{n-f/2}$, $\mathbf{v}'$, Ω, $\mathbf{b}$, and $\mathbf{e}$ are as defined in Section 3.2.4, and the parameters ($t_1, \ldots, t_7$) are independent of Δt. Substituting Eq. (51) into Eq. (20) and canceling all possible terms by choosing the $t_1, \ldots, t_7$ parameters results in a second-order method for arbitrary fields. The truncation error for the post-injected position, $\mathcal{E}_x$, is fourth-order for general fields. The truncation error for the post-injected velocity, $\mathcal{E}_v$, has $d_{ttt}\mathbf{x}(t_n)/24$ as a second-order term. For this method to be $\mathcal{O}(\Delta t^2)$ accurate, $t_1 = t_2 = t_3 = (2f-3)/4$, $t_4 = t_5 = t_6 = 2f/3 - 1$ in Eq. (49), and $t_7 = 1/24$ in Eq. (51). Since the only difference between Eqs. (51) and (48) is the term with the t_7 coefficient added to the velocity update, this method will not be rewritten. With time-independent fields this method reduces to the field gradient fractional time step method in Section 3.2.4.

3.3. *Computer Performance of Injection*

The more accurate pushes are computationally more expensive. An operation count for the methods described here is given in Table I, in three spatial and velocity coordinates; standard leap-frog is given as well. On contemporary computers the tangent function is by far the most expensive part of the push. If the tangent is used, then the ratio of the speed

TABLE I
Operation Count for the Different Methods Discussed in the Text

Method	Operation count		
	Addition	Multiply	Tan
Simple push	7	6	0
Fractional time step Boris push	43	56	2
Modified Boris push	89	105	3
Boris push with field gradients	110	129	3
Second-order method	123	155	3
Leap-frog	21	22	1

per each push is roughly proportional to the number of tangent function calls. Because of the expense of the tangent call, a small angle approximation is almost always used for time changing magnetic fields (if the magnetic field is time-independent then $\tan(\Omega\delta t/2)$ can be calculated once and stored on a grid for spatially changing fields). If the small angle approximation is used, the ratio of the multiplication operation count will roughly be the ratio of the computation speed for each push. Therefore the fractional time step Boris push, modified Boris push, Boris push with field gradients, and general second-order method are about 2.5, 4.7, 5.9, and 7 times as expensive as leap-frog. Due to different coding methods and compiler optimizations for implementing a push, a more accurate ratio is hard to estimate. For example, for the leap-frog push there is generally not a function call overhead for each particle; however, for our implementation, each injection push does have a function call overhead. Also, more information is used in the higher order pushes, past time field values, and derivatives of fields; the retrieval time from the main memory of the computer may be a significant part of the time needed to calculate the injection push. However, due to the increase in accuracy the time step could be increased by up to an order of magnitude with the same error; see Figs. 9 through 11. The increase of the time step Δt is still limited by numerical instabilities; for example, in explicit leap-frog the time step is still limited by $\omega_p\Delta t < 2$ where ω_p is the plasma frequency. The increased expense of calculating a higher order injection push is negated by a large savings in pushing particles in the plasma bulk with a larger time step, assuming the bulk has many more particles than are being injected each time step (it is difficult to construct a case where this is not true). For single species diode simulation which will have the fewest number of bulk particles to injected particles; the ratio can be as small as one hundred.

3.4. *Results*

The test cases analyzed here are special cases of the crossed-field diode [4, 12], illustrated in Fig. 5. For clarity, examples will be given in one dimension (XPDP1 [11]); however, the general second-order method is implemented in XOOPIC [32] and XPDP2 [33, 34] in 2D. A uniform time-independent external magnetic field, B, is imposed parallel to the cathode surface along z. Under a constant imposed voltage, V_0, on the diode, the Hull field [35], B_H, is defined as the minimum field for magnetic insulation, such that one electron leaving the cathode with velocity v_x would just graze the anode,$B_H = (2mV_0/|e|L_x^2 + (mv_x/|e|L_x)^2)^{1/2}$, where m is the mass and e is the charge of an electron. The imposed

TABLE II
A Summary of Parameters Used in the Three Test Cases

	Case 1	Case 2	Case 3
	Constant E	Gradient of E	Oscillatory E
$mv_{0x}^2/(2q)$	0.5 eV		
$mv_{0y}^2/(2q)$	0.5 eV		
B$\hat{z}$	337 G ($\|\Omega\| = 5.92617 \times 10^9$ rad/s)		
Δt	5×10^{-12} sec ($\|\Omega\|\Delta t = 0.0296$)		
E_0	-10^6 V/m	-9.391×10^4 V/m	-10^5 V/m
E_1	0	-4.49×10^9 V/m^2	0
ϵ	0	22.4824	0
ω	1	4.84580	1
$\tilde{E}$	0	0	1

magnetic field for all the cases is 1.5×10^{-2}% below the Hull field; hence, electrons are collected by the anode. For emission below the limiting current, J_c [36], all the current injected at the cathode propagates to the anode. This current is also known as the critical current, the largest current where a steady state cycloidal flow exists. The definition of critical current is also valid for solutions with fields larger than the Hull field where no particle current is transmitted. The three imposed field test cases are idealized limits of this diode, summarized in Table II. The self-consistent case, summarized in Table IV, is a simulation near the limiting current slightly below the Hull field, then comparing with theory [12] the L_2 norm of the density for different injection methods and number of cells.

Case 1 approximates the initial condition ($t = 0$) in the gap. Initially, the gap is empty and there is a large vacuum electric field with no gradient. Case 2 approximates the steady state behavior of the gap for currents below the critical current. At steady state the electric field is smaller than the vacuum case, and the electric field gradient is large near the cathode. Both of these cases will be approximated by time-independent fields to simplify the problem. Case 3, a time-dependent example, approximates the "oscillatory steady state" of a gap with a current above the critical current. The fields are not self-consistent but oscillate harmonically to make the analysis tractable. The frequency and amplitude of the electric field oscillation for the third case were obtained from a self-consistent PIC simulation with twice the critical current injected. Efficacy of injection methods will be compared. After the injection push, the particles are pushed with leap-frog as usual.

The lack of an unmagnetized example is not an oversight; because an unmagnetized push is a subset of the magnetized push, it is not done. These methods can be greatly simplified for the unmagnetized case; $\mathbf{v}_{n+1-f}$ is not needed for any of the pushes.

For the three test cases an analytic solution can be obtained. The equations of motion describing the idealized cases are

$$\begin{aligned} \ddot{x}(t) &= \frac{q}{m}(E(x,t) + \dot{y}(t)B_0) \quad \text{and} \\ \ddot{y}(t) &= -\frac{q}{m}\dot{x}(t)B_0, \end{aligned} \tag{52}$$

where $E(x,t) = E_0(1 + \tilde{E}\cos(\omega_0' t)) + E_1 x$, and q and m are the charge and mass of the particle, respectively. In terms of the dimensionless variables, $\tilde{x} = \Omega^2 m/(qE_0)x$ and $\tilde{t} = \Omega t$

with $\Omega = qB_0/m$ (signed cyclotron frequency) results in

$$\begin{aligned}\ddot{\tilde{x}}(\tilde{t}) &= 1 + \tilde{E}\cos(\omega_0\tilde{t}) + \epsilon\tilde{x}(\tilde{t}) + \dot{\tilde{y}}(\tilde{t})\\ \ddot{\tilde{y}}(\tilde{t}) &= -\dot{\tilde{x}}(\tilde{t}),\end{aligned} \tag{53}$$

where $\omega_0 = \omega_0'/\Omega$ and $\epsilon = (q/m)E_1/\Omega^2$. The solution may be written in closed form for this equation with the initial conditions $x(0) = 0$, $v_x(0) = \dot{x}(0) = v_{x0}$, and $v_y(0) = \dot{y}(0) = v_{y0}$.

$$\begin{aligned}\tilde{x}(\tilde{t}) = {} & \big\{\big[(v_{y0}+1)\big(\omega^2-\omega_0^2\big) + \omega^2\tilde{E}\big]\cos(\omega\tilde{t}) - \omega^2\tilde{E}\cos(\omega_0\tilde{t})\\ & + \big(\omega^2-\omega_0^2\big)[v_{x0}\omega\sin(\omega\tilde{t}) - (v_{y0}+1)]\big\}\big/\big(\omega^2\big(\omega^2-\omega_0^2\big)\big)\\ \tilde{y}(\tilde{t}) = {} & \big\{\omega\omega_0\big(\omega^2-\omega_0^2\big)\big[-v_{x0} + (v_{y0}(\omega^2-1)-1)\tilde{t} + v_{x0}\cos(\omega\tilde{t})\big]\\ & + \omega_0\big(\tilde{E}\omega^2 + \big(\omega^2-\omega_0^2\big)(v_{y0}+1)\big)\sin(\omega\tilde{t})\\ & - \tilde{E}\omega^3\sin(\omega_0\tilde{t})\big\}\big/\big(\omega^3\omega_0\big(\omega^2-\omega_0^2\big)\big)\\ \tilde{v}_x(\tilde{t}) = {} & \big\{v_{x0}\omega\big(\omega^2-\omega_0^2\big)\cos(\omega\tilde{t}) + \big[(v_{y0}+1)\big(\omega^2-\omega_0^2\big) + \omega^2\tilde{E}\big]\sin(\omega\tilde{t})\\ & - \tilde{E}\omega\omega_0\sin(\omega_0\tilde{t})\big\}\big/\big(\omega\big(\omega^2-\omega_0^2\big)\big)\\ \tilde{v}_y(\tilde{t}) = {} & \big\{\big[(v_{y0}+1)\big(\omega^2-\omega_0^2\big) + \omega^2\tilde{E}\big]\cos(\omega\tilde{t}) - \omega^2\tilde{E}\cos(\omega_0\tilde{t})\\ & + \big(\omega^2-\omega_0^2\big)\big[v_{y0}(\omega^2-1) - 1 - v_{x0}\omega\sin(\omega\tilde{t})\big]\big\}\big/\big(\omega^2\big(\omega^2-\omega_0^2\big)\big),\end{aligned} \tag{54}$$

where $\omega^2 = 1 - \epsilon$. $\tilde{x}(\tilde{t})$, $\tilde{v}_x(\tilde{t})$, and $\tilde{v}_y(\tilde{t})$ will be compared to the numerical calculation. Figures 6 through 8 compare the accuracy of the first push. The error is normalized by the Δx, Δv_x, or Δv_y of leap-frog push; this effectively slides the vertical axes without changing the shape of the graph. In other words, the normalization is reasonable but not unique. A different normalization would change the magnitude but not the shape of the error. The

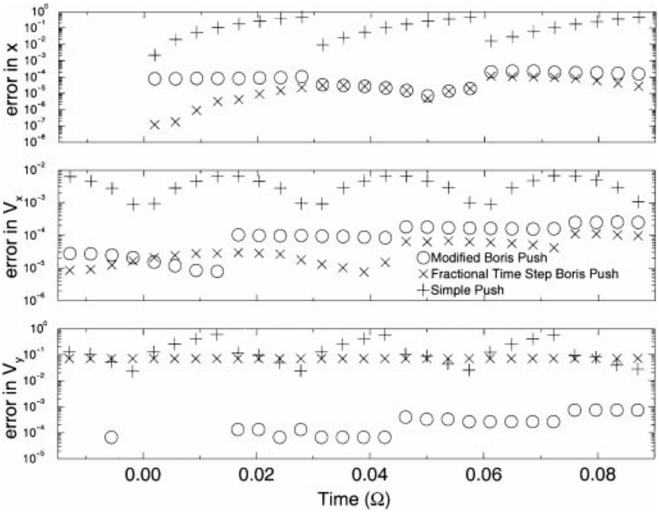

FIG. 6. The fractional error in the injected particle's position and velocity with injection methods discussed in the text for Case 1 (constant electric field). The Boris push with field gradients and the second-order method are equivalent to the modified Boris push for this case; therefore, only the modified Boris push is shown.

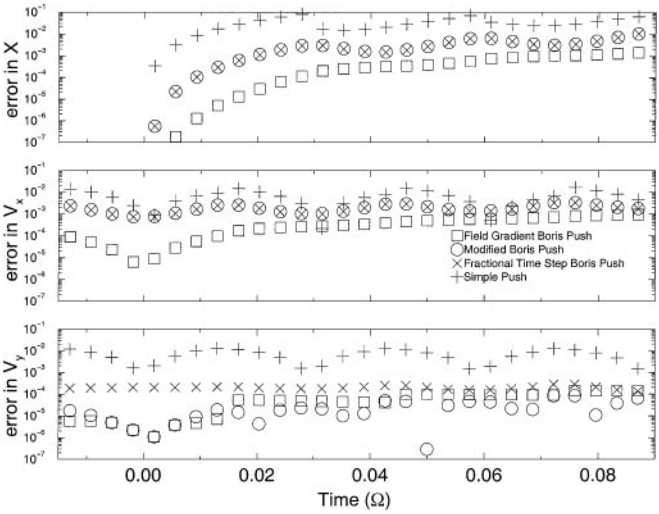

FIG. 7. The fractional error in the injected particle's position and velocity with injection methods discussed in the text for Case 2 (gradient of electric field). For this case the Boris push with field gradients is equivalent to the second-order method; therefore, the second-order method is not shown.

horizontal axis is the time elapsed since the particle has been injected from the cathode. Note that the error in v_y is dominated by the $d_{ttt}\mathbf{x}(t)\Delta t^2/24$ term in the noncorrected pushes. Figures 9 through 11 show the root mean squared truncation error averaged over the fractional time step,

$$\langle||\mathcal{E}||\rangle_f = \int_0^1 ||\mathcal{E}||\,df, \tag{55}$$

for the different cases as a function of Δt. The lowest-order term in the RMS error is shown in Table III. The leap-frog error is for a full time step. These figures were made by keeping

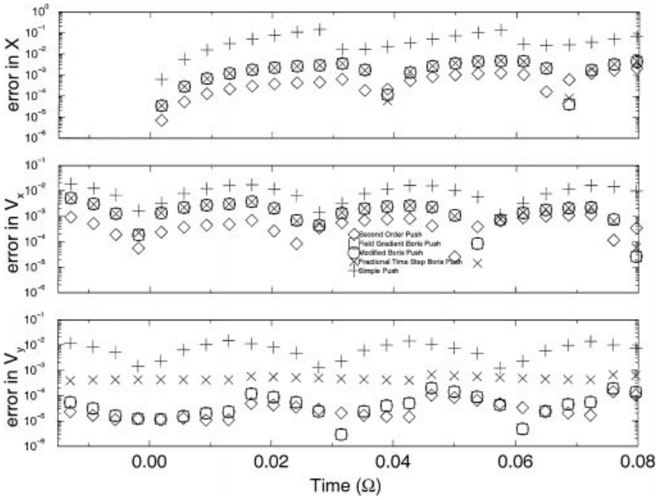

FIG. 8. The fractional error in the injected particle's position and velocity with injection methods discussed in the text for Case 3 (oscillatory electric field). The Boris push with field gradients and the modified Boris push are equivalent for this case, the symbols overlap.

TABLE III

A Summary of the Order of the Injection Methods Shown in the Text for Different Cases, the Constant Can Be Found in Appendix B

	Case 1	Case 2	Case 3
Simple push	g_{11}	g_{12}	g_{13}
Simple Boris push	$g_{21}\Delta t$	$g_{22}\Delta t$	$g_{23}\Delta t$
Modified Boris push	$g_{31}\Delta t^2$	$g_{32}\Delta t$	$g_{33}\Delta t$
Boris push with field gradients	$g_{31}\Delta t^2$	$g_{42}\Delta t^2$	$g_{43}\Delta t$
Second-order push	$g_{31}\Delta t^2$	$g_{42}\Delta t^2$	$g_{53}\Delta t^2$
Leap-frog	$g_{61}\Delta t^2$	$g_{62}\Delta t^2$	$g_{63}\Delta t^2$

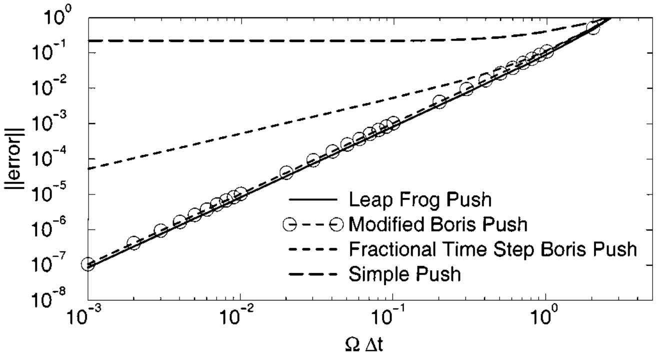

FIG. 9. The error, $\|\mathcal{E}\|_2$, in the injection method averaged over the injection time during a time step with injection methods discussed in the text for Case 1 (constant electric field). The Boris push with field gradients and the second-order method are equivalent to the modified Boris push for this case; therefore, only the modified Boris push is shown.

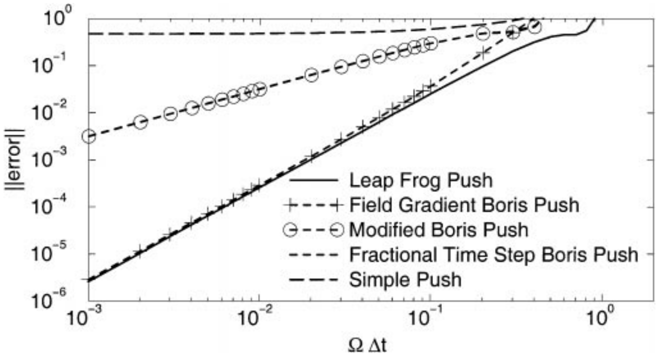

FIG. 10. The error, $\|\mathcal{E}\|_2$, in the injection method averaged over the injection time during a time step with injection methods discussed in the text for Case 2 (gradient of electric field). The modified Boris push and the fractional time step Boris push curves overlay. For this case the Boris push with field gradients is equivalent to the second-order method; therefore, the second-order method is not shown.

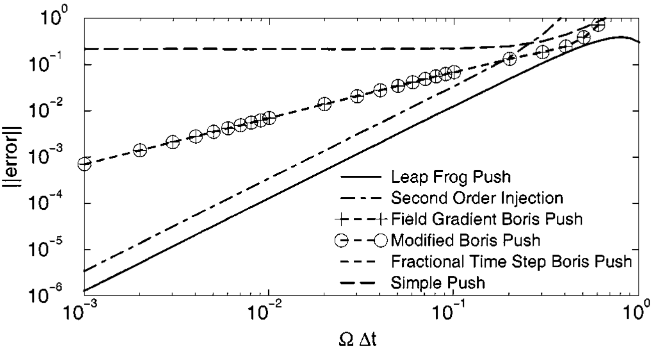

FIG. 11. The error, $\|\mathcal{E}\|_2$, in the injection method averaged over the injection time during a time step with injection methods discussed in the text for Case 3 (oscillatory electric field). The Boris push with field gradients and the modified Boris push are equivalent for this case which overlay the fractional time step Boris push.

one higher truncation error than the lowest order term; because of this, these figures may be inaccurate as $\Omega\Delta t$ approaches one.

The reason that injection push error is substantially larger than the leap-frog push in Case 3 as compared to Cases 1 and 2 is that the time-dependent field is being extrapolated to a future point which incurs a larger error.

For models with cathode-field characteristics similar to one of the cases presented here using the truncation error can determine whether the injection push error is negligible. For example, assume for the moment that a fractional time step Boris push has been implemented. Using the electric field given in Case 2, $E_0 = 9.391 \times 10^4\ V/m$, the gradient of the electric field which would result in one hundred times the error over implementing the field gradient injection push is

$$g_{22}\Delta t \geq (100) g_{42} \Delta t^2, \tag{56}$$

where g_{22} and g_{42} are defined in Appendix B. By reducing Δt this relationship will be satisfied for some Δt; however, because of the other constraints in the bulk plasma requiring Δt to be 1×10^{-12} s, then Eq. (56) is satisfied if the normalized electric field gradient is between -2.25 ($4.00 \times 10^{11}\ V/m^2$) and -0.14 ($2.52 \times 10^{10}\ V/m^2$). For all other values of the electric field gradient it is more than one hundred times more accurate to use the field gradient push.

The above analysis is only the push truncation error and does not take into account the feedback of the coupling of the particles to the fields. The injection push is important when space charge is important and the error propagates through the system due to the error in the fields as well as the particles directly.

When a lower order or a non-time-centered push is used, the effects of the discrete time steps can be severe. For the examples shown here, there are gaps in particle positions between the time steps; if the field was chosen so that it decelerated the injected particles, particle positions from different time steps would have overlapped. This is a source of “noise” when these particles are weighted to the grid.

For the self-consistent simulation (all the parameters are shown in Table IV) the externally imposed magnetic field of 336G is 0.371% below the Hull field and the injected current of 16705 A/m^2 is 2.50% below the critical current. Therefore, all the current injected at the cathode is collected by the anode.

TABLE IV
A Summary of Parameters Used in the Self-Consistent Case

Physical parameters	$mv_{0x}^2/(2q)$	0.5 eV
	$B\hat{z}$	336 G ($\lvert\Omega\rvert = 5.9086 \times 10^9$ rad/s)
	V_0	-10^4 V
	L	0.01 m
	J	16705 A/m^2
Physical scaling parameters	$B_H\hat{z}$	337.25 G
	J_c	17133.81 A/m^2
Numerical parameter	Δx	1.22070×10^{-6} m

The convergence in the cell size Δx is limited by the cold beam nonphysical instability [10] proportional to Δx, whereas the accuracy of the field solve is Δx^2. This instability only occurs for a cold beam; we are using this cold example because we have a closed form for the solution. To reduce the effect of the cold beam instability, 8192 cells are used in the simulation of the crossed-field diode. This adds a fixed error in the simulation based on the grid error from the instability rather than the truncation error of the method. The L_2 norm of relative error $\|n_{PIC}/n_t - 1\|_2$ is shown in Fig. 12 as a function of the time step in the simulation. This figure shows the error decreasing for the second-order (the general second-order method and the field gradient fractional time step Boris push because the problem is time-independent) injection method as Δt^2, first-order (modified fractional time step Boris push and fractional time step Boris push) injection methods Δt, and the zeroth-order (simple injection push) injection method not decreasing. Since the size of the cell Δx is not changing there is a constant error with respect to changing the time step Δt. The L_2 norm of the relative error due to the grid is approximated by the L_2 norm of the simulation run with $\Omega\Delta t = 9.2 \times 10^{-4}$ to be about 4.74×10^{-4}. A fit to error of the second-order methods and first-order methods with the constant error (4.74×10^{-4}) from the grid is also shown in Fig. 12 and is labeled second order fit and first order fit, respectively.

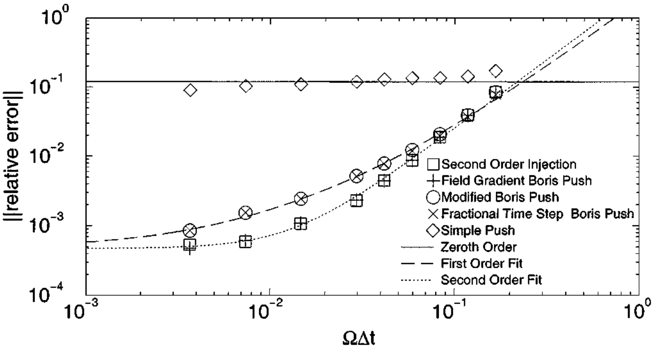

FIG. 12. The L_2 norm of the relative error ($\|n_{PIC}/n_t - 1\|_2$) in the diode for injection method given in the text for the self-consistent case. The fractional time step Boris push and the modified Boris push data points overlay each other in the plot because they are both first order. The second-order injection and the Boris push with field gradients data points overlay each other. Both of these methods are second order because this simulation has reached a steady state. For smaller time steps Δt the error is dominated by the cell size Δx error which is fixed for all the data. The fits to the data take into account the error associated with the cell size.

4. CONCLUSIONS

Giving attention to detail, the authors have found methods that are second-order accurate for the inversion of loaded and injected distributions of particles. The inversion process is limited to Maxwellian distributions and fluxes, both full distributions and arbitrary cutoff at a lower and upper velocity. The second-order injection method starts for an arbitrary position and velocity at the same time and results in a position and velocity half a time step apart, with an error consistent with the standard leap-frog integrator.

A number of common cases demonstrate sensitivity to low-order injection schemes. We have shown that using a higher order method is far more effective in obtaining a more accurate result than decreasing the time step with a lower order method. Use of these methods does not significantly increase the run time of the simulation, unless the number of particles injected is comparable to the total number of particles in the simulation.

APPENDIX A

Effects of Discretizing Distribution and Fluxes

One of the consequences of inverting the distribution function with a finite number of points is that the Maxwellian has an effective upper cutoff. The maximum size of the array is determined by the desired resolution of the tail of the distribution. For example, if $v_{\max}$ is the largest velocity that can be chosen, the probability that a particle lies beyond $v_{\max}$ is $\int_{v_{\max}}^{\infty} f(v)\,dv$ (or $\int_{v_{\max}}^{\infty} v f(v)\,dv$ for a flux). Increasing the number of bins used beyond where the probability of a single bin is less than the probability beyond $v_{\max}$ is ineffective because the fraction neglected is much larger than the error made due to the numerical method. The velocity index is constructed by choosing a pseudorandom number and multiplying by the number of bins. The velocity is computed by a linear interpolation between the array values indicated by the index.[4]

To calculate the number of bins needed, several different measures of the error have been used to quantify the error. The *local moment error* of the *n*th velocity moment is the error in each bin, which can be written as

$$\mathrm{LME} = \int_{v}^{v+\Delta v} v^n (f_{\mathrm{bin}} - f_{\mathrm{anal}})\,dv, \tag{A.1}$$

where f_{bin} is the normalized distribution formed by the linear interpolation of the array, f_{anal} is the desired normalized distribution, Δv is the width of the bin, and n is the velocity moment. The difference in moments integrated over the entire distribution is:

$$\text{moment error} = \sum_{\mathrm{bin}} \int_{v}^{v+\Delta v} v^n (f_{\mathrm{bin}} - f_{\mathrm{anal}})\,dv = \int_{v_{cl}}^{v_{cu}} v^n (f_{\mathrm{bin}} - f_{\mathrm{anal}})\,dv. \tag{A.2}$$

The L_2 norm of the difference of the moments in one bin will also be used for a measure

[4] Consider calculating a new pseudorandom number for interpolation because high bits in many random number algorithms are more “random” than low bits. In other words, use R_1 to choose the bin and then R_2 to interpolate between bins.

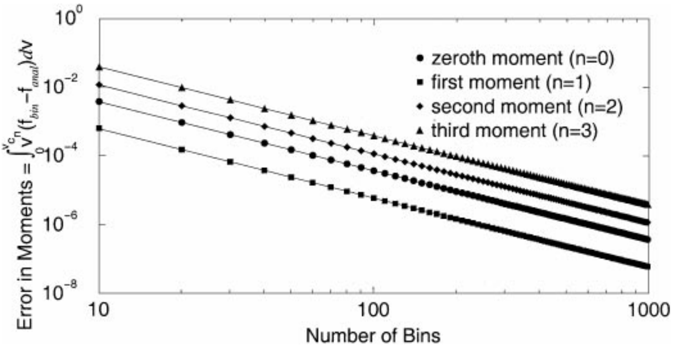

FIG. 13. The absolute error in the zeroth, first, second, and third moments as a function of the number of bins for a Maxwellian flux with $v_0 = 1$ and $v_c = 3$ (normalized by $\sqrt{2}v_{ti}$).

of the error which will be called the *local squared moment error*:

$$\text{LSME} = \sqrt{\int_v^{v+\Delta v} v^{2n}(f_{\text{bin}} - f_{\text{anal}})^2\, dv}. \tag{A.3}$$

The tolerance, T, is defined as the L_2 norm over the entire distribution:

$$\text{T} = \sqrt{\sum_{\text{bin}} \int_v^{v+\Delta v} v^{2n}(f_{\text{bin}} - f_{\text{anal}})^2\, dv} = \sqrt{\int_{v_cl}^{v_{cu}} v^{2n}(f_{\text{bin}} - f_{\text{anal}})^2\, dv}. \tag{A.4}$$

These measures of merit can be calculated numerically. Examples of Maxwellian flux with $v_0 = 1$ and $v_c = 3$ are shown in Figs. 13 and 14. The LME shows the integrated error in each bin, which mostly cancels out. In contrast the LSME does not have errors that cancel. The error in the moments of the distribution is plotted as a function of the number of bins used in Fig. 13. This can be compared to the tolerance as shown in Fig. 14. The error in the moments and the tolerance decrease as (number of bins)$^{-2}$, since the numerical inversion, both the tabulation and the interpolation, is second-order accurate. Hence for a Maxwellian distribution, for which the tail does not need to be resolved, a modest number of grids can give adequate accuracy. An easily analyzed expression for accuracy has not

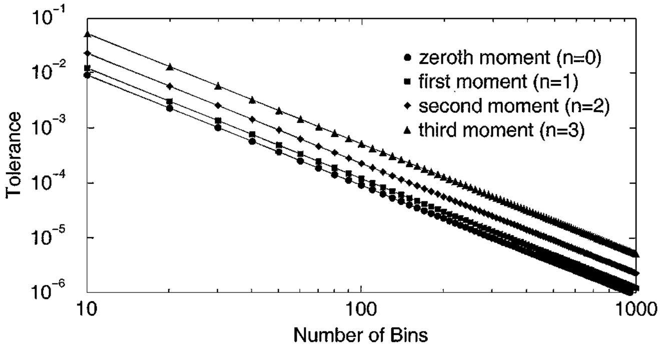

FIG. 14. The tolerance for the zeroth, first, second, and third moments as a function of the number of bins for a Maxwellian flux with $v_0 = 1$ and $v_c = 3$ (normalized by $\sqrt{2}v_{ti}$).

been found but minimal requirements are not difficult to meet. For a Maxwellian flux the corresponding graphs to Figs. 13 and 14 would have the same slope; however, the error is slightly larger for the same moment and number of bins. A few hundred points with interpolation should handle any Maxwellian-like (continuous with finite first and second derivatives) distribution.

APPENDIX B

Truncation Error Constants

This appendix list the constants for the truncation error for the methods and cases in the body of the article; for definitions of g's see Table III.

Case 1.

$$\begin{aligned}
g_{11} &= \left(\sqrt{3+13v_{x0}^2+11v_{y0}+13v_{y0}^2}\right)\Big/(2n_1\sqrt{15}),\\
g_{21} &= \frac{1}{24}\sqrt{\frac{23}{14}},\\
g_{31} &= \frac{1}{288n_1\sqrt{35}}\sqrt{31713+1631v_{x0}^2+12042v_{y0}+1631v_{y0}^2},\\
g_{61} &= 1/(12n_1), \quad \text{and}\\
n_1 &= \sqrt{v_{x0}^2+\left(1+v_{y0}^2\right)^2}.
\end{aligned} \tag{B.1}$$

Case 2.

$$\begin{aligned}
g_{12} &= \sqrt{3+13v_{x0}^2+11v_{y0}+13v_{y0}^2}/(2\sqrt{15}n_2),\\
g_{22} &= \sqrt{115+8\epsilon(104\epsilon-73)v_{x0}+115(1+v_{y0})^2}/(24\sqrt{70}n_2),\\
g_{32} &= \sqrt{\frac{13}{70}\frac{|\epsilon v_{x0}|}{3n_2}},\\
g_{42} &= \big(5\epsilon^2\big(38929v_{x0}^2+48528(1+v_{y0})^2\big)+9\big(31713+1631v_{x0}^2\\
&\quad +v_{y0}(12042+1631v_{y0})\big)-6\epsilon(14689v_{x0}-6(1+v_{y0})\\
&\quad \times(13518+1877v_{y0}))\big)^{1/2}/(864\sqrt{35}n_2),\\
g_{62} &= \sqrt{\epsilon^2v_{x0}^2+(1+\epsilon(1+v_{y0}))^2}/(12n_2), \quad \text{and}\\
n_2 &= \sqrt{v_{x0}^2+\left(1+v_{y0}^2\right)^2}.
\end{aligned} \tag{B.2}$$

Case 3.

$$g_{13} = \sqrt{6+3\tilde{E}^2+26v_{x0}^2+22v_{y0}+26v_{y0}}/(2n_3\sqrt{30}),$$

$$g_{23} = \sqrt{((230+16\epsilon(104\epsilon-73))v_{x0}^2+230(1+v_{y0})^2+\tilde{E}^2\big(115+1984\omega_0^2\big))}/(48n_3\sqrt{35}),$$

$$g_{33} = \sqrt{26\epsilon^2v_{x0}^2+31\tilde{E}^2\omega_0^2}/(6n_3\sqrt{35}),$$

$$
\begin{aligned}
g_{43} &= \sqrt{\frac{31}{35}}\frac{|\tilde{E}\omega_0|}{6},\\
g_{53} &= \bigl(10\epsilon^2\bigl(14264\tilde{E}^2 + 38929v_{x0}^2 + 48528(1+v_{y0})^2\bigr)\\
&\quad + 18\bigl(31713 + 1631v_{x0}^2 + v_{y0}(12042 + 1631v_{y0})\bigr)\\
&\quad + \tilde{E}^2\bigl(285417 - 633751\omega_0^2 + 1288224\omega_0^4\bigr)\\
&\quad - 12\epsilon\bigl(14689v_{x0}^2 - 6(1+v_{y0})(13518 + 1877v_{y0})\\
&\quad + 18\tilde{E}^2\bigl(4382\omega_0^2 - 2253\bigr)\bigr)\bigr)^{1/2}\big/(864n\sqrt{70}),\\
g_{63} &= \bigl(2 + \epsilon^2\bigl(\tilde{E}^2 + 2\bigl(v_{x0}^2 + (1+v_{y0})^2\bigr)\bigr)\\
&\quad + 2\epsilon\bigl(2 + \tilde{E}^2 + 2v_{y0} - \tilde{E}^2\omega_0^2\bigr) + \tilde{E}^2\bigl(1 - \omega_0^2 + \omega_0^4\bigr)\bigr)^{1/2}\big/(12n\sqrt{2}), \quad \text{and}\\
n_3 &= \sqrt{\frac{\tilde{E}^2}{2} + v_{x0}^2 + \bigl(1 + v_{y0}^2\bigr)^2}.
\end{aligned}
\tag{B.3}
$$

REFERENCES

1. V. P. Gopinath, J. P. Verboncoeur, and C. K. Birdsall, Simulation of transmitted current in a cylindrical cross-field diode, in *Proc. 22nd IEEE Int. Conf. Plasma Sci., Madison, WI, June 1995* (IEEE Service Center, Piscataway, NJ).
2. K. L. Cartwright, J. P. Verboncoeur, V. P. Gopinath, and C. K. Birdsall, Transverse asymmetry in a crossed–field diode, in *Proceedings of the First International Workshop on Crossed–Field Devices, Ann Arbor, MI, August 1995* (University of Michigan, Ann Arbor, MI).
3. V. P. Gopinath, J. P. Verboncoeur, and C. K. Birdsall, Similarity of stability characteristics of planar and coaxial crossed-field diodes, *Phys. Plasma* **3**, 2766 (1996).
4. J. P. Verboncoeur and C. K. Birdsall, Rapid current transition in a crossed-field diode, *Phys. Plasma* **3**, 712 (1996).
5. J. P. Verboncoeur, P. J. Christenson, and H. B. Smith, Simulation of noise in a traveling wave tube, in *Bulletin of the American Physical Society Seattle, WA, November 1999* (Division of Plasma Physics, American Physical Society), Vol. 44, p. 254.
6. K. L. Cartwright, P. J. Christenson, J. P. Verboncoeur, and C. K. Birdsall, Virtual cathode oscillations in space-charge limited electron flow, in *International Conference On Crossed Field Devices and Applications, Boston, MA, June 17–19, 1998* (Northeastern University).
7. K. L. Cartwright, P. J. Christenson, J. P. Verboncoeur, and C. K. Birdsall, Effects of voltage fluctuations at the cathode on Brillouin flow, in *International Conference on Crossed Field Devices and Applications, Boston, MA, June 17–19, 1998* (Northeastern University).
8. V. P. Gopinath, J. P. Verboncoeur, and C. K. Birdsall, Multipactor electron discharge physics using an improved secondary emission model, *Phys. Plasma* **5**, 1535 (1998).
9. K. L. Cartwright, *Theory and Simulation of Oscillations on Near-Steady State in Crossed-Field Electron Flow and the Resulting Transport* (Ph.D. thesis, University of California, Berkeley, 1999).
10. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (Hilger, New York, 1991).
11. J. P. Verboncoeur, M. V. Alves, V. Vahedi, and C. K. Birdsall, Simultaneous potential and circuit solution for 1d bounded plasma particle simulation codes, *J. Comput. Phys.* **104**, 321 (1993).
12. Y. Y. Lau, P. J. Christenson, and D. Chernin, Limiting current in a crossed-field gap, *Phys. Fluids B* **5**, 4486 (1993).
13. J. A. Byers and M. Grewal, Perpendicularly propagating plasma cyclotron instabilities simulated with a one-dimensional computer model, *Phys. Fluids* **13**, 1819 (1970).

14. J. Denavit and W. L. Kruer, How to get started in particle simulation, *Comments Plasma Phys. Control Fusion* **6**, 35 (1980).

15. R. L. Morse and C. W. Nielson, Numerical simulation of the Weibel instability in one and two dimensions, *Phys. Fluids* **14**, 830 (1971).

16. S. J. Gitomer, Comments on numerical simulation of the Weibel instability in one and two dimensions, *Phys. Fluids* **14**, 1591 (1971).

17. J. Denavit, Numerical simulation of plasmas with periodic smoothing in phase space, *J. Comput. Phys.* **9**, 75 (1972).

18. S. J. Gitomer and J. C. Adam, Multibeam instability in a Maxwellian simulation plasma, *Phys. Fluids* **19**, 719 (1976).

19. J. M. Dawson, Plasma oscillations of a large number of electron beams, *Phys. Rev.* **118**, 381 (1960).

20. W. S. Lawson and P. C. Gray, Heat flow between species in one-dimensional particles plasma simulations, *J. Comput. Phys.* **95**, 195 (1991).

21. W. S. Lawson, Artificial cooling due to quiet injection in bounded plasma particle simulations, *J. Comput. Phys.* **77**, 330 (1988).

22. J. Denavit, Discrete particle effects in whistler simulation, *J. Comput. Phys.* **15**, 449 (1974).

23. Y. Matsuda and H. Okuda, Collisions in multi-dimensional plasma simulations, *Phys. Fluids* **18**, 1740 (1975).

24. P. Bratley and B. L. Fox, Implementing Sobol's quasirandom sequence generator, *ACM Trans. Math. Software* **14**, 88 (1988).

25. G. Marsaglia and B. Narasimhan, Simulating interpolation search, *Comput. Math. Appl.* **26**(8), 31 (1993).

26. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods* (Methuen, London, 1964).

27. J. H. Smith, *Introduction to Special Relativity* (Stipes, Champaign, IL, 1965).

28. W. S. Lawson, Particle simulation of bounded 1d plasma systems, *J. Comput. Phys.* **80**, 253 (1989).

29. L. A. Schwager and C. K. Birdsall, Collector and source sheaths of a finite ion temperature plasma, *Phys. Fluids B* **2**, 1057 (1990).

30. O. Buneman, Time reversible difference procedures, *J. Comput. Phys.* **1**, 517 (1967).

31. J. P. Boris, Relativistic plasma simulation-optimization of hybrid code, in *Proc. Fourth Conf. Num. Sim. Plasmas, Wash. D.C., November 1970.*

32. J. P. Verboncoeur, A. B. Langdon, and N. T. Gladd, An object-oriented electromagnetic PIC code, *Comput. Phys. Commun.* **87**, 199 (1995).

33. V. Vahedi, C. K. Birdsall, M. A. Lieberman, G. DiPeso, and T. D. Rognlien, Verification of frequency scaling laws for capacitive radio-frequency discharges using two-dimensional simulations, *Phys. Fluids B* **5**, 2719 (1993).

34. V. Vahedi and G. DiPeso, Simultaneous potential and circuit solution for two-dimensional bounded plasma simulation codes, *J. Comput. Phys.* **131**, 149 (1997).

35. A. W. Hull, The effect of a uniform magnetic field on the motion of electrons between coaxial cylinder, *Phys. Rev.* **18**, 31 (1921).

36. P. J. Christenson and Y. Y. Lau, Transition to turbulence in a crossed-field gap, *Phys. Plasma* **1**, 3725 (1994).