Journal of Computational Physics 228 (2009) 1391–1403

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Small-angle Coulomb collision model for particle-in-cell simulations

Don S. Lemons [a,*], Dan Winske [b], William Daughton [b], Brian Albright [b]

[a] *Bethel College, North Newton, KS 67117, United States*
[b] *Los Alamos National Laboratory, Los Alamos, NM 87545, United States*

ARTICLE INFO

Article history:
Received 30 July 2008
Received in revised form 21 October 2008
Accepted 22 October 2008
Available online 5 November 2008

Keywords:
Collision algorithm
Particle-in-cell simulation
Plasma simulations
Numerical methods
Coulomb collisions in plasmas
Stochastic differential equations

ABSTRACT

We construct and investigate a set of stochastic differential equations that incorporate the physics of velocity-dependent small-angle Coulomb collisions among the plasma particles in a particle-in-cell simulation. Each particle is scattered stochastically from all the other particles in a simulation cell modeled as one or more Maxwellians. Total energy and momentum are conserved by linear transformation of the velocity increments. In two test simulations the proposed "particle-moment" collision algorithm performs well with time steps as large as 10% of the relaxation time – far larger than a particle-pairing collision algorithm, in which pairs of particles are scattered from one another, requires to achieve the same accuracy.

1. Introduction

Particle-in-cell (PIC) codes represent forces among spatially separate particles by field quantities collected on a spatial grid. Such codes efficiently model low-density plasma interactions but ignore collisions, that is, the effect of forces among particles closer than the distance between grid points. When collisions are important the usual PIC code algorithm needs to be supplemented with a collision model. Binary collisions between charged particles are important in many applications, such as interpenetrating and stagnating laser-produced plasmas [1]; dense plasmas that slow down, deposit energy, and equilibrate temperatures [2]; burning plasmas that produce neutrons [3]; and dusty plasmas [4].

Because a pair of interacting particles can easily be made to conserve momentum and energy, the collision model most widely adopted in PIC codes elastically scatters pairs of computational particles from one another [5]. Because each particle could scatter from every other particle in the cell the scatterings represented are only a random sample of possible ones. Cumulative scattering through large angles [6] and particle division and recombination [7] have improved particle-pairing algorithms. Particle-pairing algorithms have also been vectorized [8] and generalized to allow for weighted particles [9]. More recently, several particle-pairing algorithms have been compared [10] and a new method to speed up such algorithms by representing the plasma as a combination of Maxwellian distributions and discrete particles has been introduced [11]. The particle-pairing method has been successfully employed in large-scale gyrokinetic and delta-f simulations of cross-field transport in magnetic fusion systems [12].

Particle-pairing is the most commonly used collision model in plasma simulations, but alternative, faster, collision algorithms have been developed. One of these, the multi-fluid method, treats each species or component as a separate fluid that couples to every other fluid via plasma slowing down and equilibration rates [13–15]. More recently, Sherlock has developed

[*] Corresponding author.
E-mail address: lemons.don@gmail.com (D.S. Lemons).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.10.025

a collision algorithm for modeling electron–ion collisions in hybrid codes where the ions are treated as particles and the electrons as a fluid [16].

Our contribution has developed from the so-called "collision-field" method in which each particle is scattered only once each time step from field quantities that represent the collisional effect of other species (inter-species collisions) within the cell [17]. Collisions among particles of the same species (intra-species collisions) were modeled with solutions to a Langevin equation. Re-scaling the post-collision particle velocities conserved energy and momentum. The collision-field method is no noisier than particle-pairing since both methods execute order N collisions each time step where N is the number of particles per cell. The collision-field method also allows for larger time steps. In its earliest implementation [17], the collision-field method employed a velocity independent collision rate.

We propose a conceptually simple generalization of the collision-field method that incorporates velocity-dependent collision rates and treats intra- and inter-species collisions with the same algorithm. In order to calculate velocity-dependent collision rates describing the scattering of a single test particle from an arbitrary distribution of field particles, a multi-dimensional integral over velocity space must, in principle, be performed. Manheimer et al. reduced this integration to one requiring a single integration by assuming a spherically symmetric velocity distribution [18]. While interesting, non-Maxwellian, locally isotropic, particle distributions are a special case.

Instead we assume that the field particles compose one or a small number of drifting *isotropic* Maxwellians. In this case the integrations can be completed in terms of standard functions and result in the velocity space diffusion coefficients first calculated by Chandrasekhar 65 years ago [19] and collected in Lyman Spitzer's classic text [20]. In particular, we incorporate Spitzer's diffusion coefficients into parameters that characterize a set of stochastic differential equations. These stochastic differential equations are completely equivalent to a Fokker–Planck equation [21]. Solutions to the stochastic differential equations are the heart of our proposed collision algorithm. We refer to it as the "particle-moment" method.

2. Collision algorithm

After the electromagnetic fields interpolated from the grid advance each particle in phase space the collision algorithm advances each particle in velocity space. The application of the particle-moment collision model discussed in this paper can be summarized with the following series of steps:

1. Choose one particle among the particles in a cell to be the test particle; the remaining particles are its complementary field particles. Calculate the average particle velocity $\underline{v}_f$ and effective temperature T_f of each species of complementary field particle having the same mass and charge.
2. Choose one species of complementary field particle. Shift to a local velocity frame in which the average velocity of that species vanishes and, consequently, the test particle has a velocity $\underline{v}_t - \underline{v}_f$ where $\underline{v}_t$ and $\underline{v}_f$ are lab frame velocities. Define polar velocity coordinates of each test particle with respect to the axis pointing in the direction of the difference vector $\underline{v}_t - \underline{v}_f$.
3. Scatter the test particle in its local frame. Collisions rotate and stretch or compress each velocity difference vector $\underline{v}_t - \underline{v}_f$. This transformation is determined by finite difference solutions to stochastic differential equations that incorporate Spitzer's velocity space diffusion coefficients. The latter are a function of the field particle species effective temperature T_f, mass m_f, and charge q_f.
4. Allow each particle to be the test particle in turn. Complete steps 1–3 for each.
5. Shift the scattered velocity of each particle back to the lab frame and calculate the lab frame velocity increment Δv_i for each particle i caused by collisions with that species.
6. When more than one species is present repeat steps 1–5 for each species and find the net lab frame velocity increment Δv_i for each particle i caused by collisions with all other particles.
7. Linearly shift all Δv_i in such a way that the total momentum and energy of all particles is conserved.

Each step of this algorithm is explained in one or more of the following sub-sections.

2.1. From lab frame to local particle coordinates

One of the $N + 1$ particles in a cell is chosen as the test particle and the remaining N compose its complementary field particles. Here we assume that all N field particles have the same mass m_f and charge q_f. The extension to more than one field particle species is straightforward.

The average field particle velocity $\underline{v}_f$ in the lab frame is given by

$$\underline{v}_f = \frac{\sum_{j=1}^{N} \underline{v}_{fj}}{N}. \tag{1}$$

Here $\underline{v}_{fj}$ is the velocity of the jth field particle. Likewise, the effective temperature of the field particles T_f is given by

$$\frac{3NT_f}{2} = \sum_{j=1}^{N} \frac{m_f(\underline{v}_{fj} - \underline{v}_f)^2}{2}. \tag{2}$$

The field particle velocity $\underline{v}_f$ and effective temperature T_f characterize the drifting Maxwellian that is used to determine velocity space diffusion coefficients.

Every particle can be a test particle and each test particle has its own set of complementary field particles. We describe the effect of scattering a test particle with mass m_t, charge q_t, and velocity $\underline{v}_t$ in terms of a velocity difference vector:

$$\underline{\omega} = \underline{v}_t - \underline{v}_f, \tag{3}$$

where $\underline{v}_f$ is the average field particle velocity. Scattering a test particle from its complementary field particles shifts both the orientation and the magnitude of $\underline{\omega}$, that is, scattering a test particle shifts the velocity difference vector from $\underline{\omega}$ to $\underline{\omega} + d\underline{\omega}$. We describe $d\underline{\omega}$ in terms of changes in polar coordinates (θ, ϕ, ω) referenced to an axis lying along and pointing in the direction of $\underline{\omega} = \underline{v}_t - \underline{v}_f$. Pre-scattering test particles have polar coordinates $(0, 0, \omega)$ while post-scattering test particles have polar coordinates $(d\theta, d\phi, \omega + d\omega)$. Fig. 1 illustrates these coordinates and their increments.

2.2. Stochastic differential equations

Scattering is a random process. Thus, the differentials, $d\theta$, $d\phi$, and $d\omega$ that describe the rotation and stretching or compressing of the test particle velocity difference vector $\underline{\omega} = \underline{v}_t - \underline{v}_f$ are differentials of random variables θ, ϕ, and ω. Consequently, the time evolution of θ, ϕ, and ω is governed by a set of stochastic differential equations:

$$d\theta = \sqrt{2\gamma dt} N_\theta(0,1), \tag{4a}$$

$$d\phi = 2\pi U_\phi(0,1), \tag{4b}$$

and

$$d\omega = -\beta\omega dt + \sqrt{\delta^2 dt} N_\omega(0,1). \tag{4c}$$

where the γ, β, and δ^2 characterize the process. Here $N_\theta(0,1)$ and $N_\omega(0,1)$ are temporarily uncorrelated stochastically independent normal random variables with mean zero and variance one and $U_\phi(0,1)$ is a temporally uncorrelated uniform random variable between zero and one. This notation is familiar from texts on stochastic processes (see, e.g., Refs. [21,22]).

2.3. Collision rates

According to Spitzer [20] the rates at which a test particle with initial speed ω_o collisionally interacts with a field particle species whose average velocity vanishes are described by

$$\left(\frac{d}{dt}\langle v_z\rangle\right)_o = -l_f^2 A_D \left(1 + \frac{m_t}{m_f}\right) G(l_f\omega_o), \tag{5a}$$

$$\left(\frac{d}{dt}\left\{\langle v_z^2\rangle - \langle v_z\rangle^2\right\}\right)_o = \frac{A_D}{\omega_o} G(l_f\omega_o), \tag{5b}$$

and

$$\left(\frac{d}{dt}\left\{\langle v_\perp^2\rangle - \langle v_\perp\rangle^2\right\}\right)_o = \frac{A_D}{\omega_o}[\Phi(l_f\omega_o) - G(l_f\omega_o)]. \tag{5c}$$

where brackets, e.g., $\langle v_z\rangle$, indicate expectation values. Furthermore, $v_z = \underline{\omega_o} \cdot \hat{e}_z$, $\underline{v_\perp} = \underline{\omega_o} - v_z\hat{e}_z$, $\hat{e}_z = \underline{\omega_o}/\omega_o$, $l_f^2 = m_f/2T$,

$$\Phi(x) = (2/\sqrt{\pi}) \int_0^x e^{-y^2} dy, \tag{6a}$$

$$G(x) = [\Phi(x) - x\Phi'(x)]/2x^2, \tag{6b}$$

$\Phi'(x) = d\Phi(x)/dx$, $A_D = 8\pi n_f q_t^2 q_f^2 \ln\Lambda/m_t^2$, and $\ln\Lambda$ is the Coulomb logarithm.

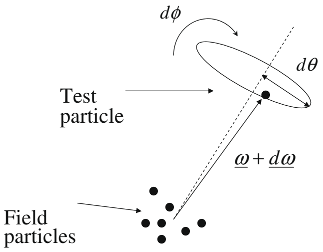

Fig. 1. Velocity space spherical polar coordinates that describe differential changes in the velocity difference vector $\underline{\omega} = \underline{v}_t - \underline{v}_f$.

2.4. Characterizing parameters defined

Here we describe some of the calculations that lead to defining the parameters γ, β, and δ^2, of the stochastic process (4) that incorporate the diffusion coefficients (5). Note that in a frame at rest with respect to the average velocity of the pre-scattered field particles, the test particle velocity $\underline{v}_t$ Cartesian component speeds v_z and $v_\perp = \sqrt{v_x^2 + v_y^2}$ are related to the components of $\underline{\omega}$ by

$$v_z = \omega_z, \tag{7a}$$

and

$$v_\perp = \sqrt{\omega_x^2 + \omega_y^2}. \tag{7b}$$

In terms of polar coordinates

$$\begin{aligned} d\omega_z &= d[\omega\cos\theta] = (\omega + d\omega)\cos(\theta + d\theta) - \omega\cos(\theta) = (\omega + d\omega)\cos(d\theta) - \omega\cos(0) \\ &= (\omega + d\omega)(1 - d\theta^2/2 + \cdots) - \omega = -\omega d\theta^2/2 + d\omega - d\omega d\theta^2/2 + \cdots, \end{aligned} \tag{8}$$

since $\theta(t = 0) = 0$. Given (4a) and (4c), Eq. (8) becomes

$$d\omega_z = -\omega\gamma dt N_\theta^2(0,1) - \omega\beta dt + \sqrt{\delta^2 dt} N_\omega(0,1), \tag{9}$$

where we have dropped terms smaller than order dt. One novelty of taking derivatives of stochastic variables is that the term $(d\theta)^2$ is the same order as dt. Taking the expectation value of both sides of (9) and specializing to the initial condition $\omega = \omega_o$ produces an initial rate

$$\left(\frac{d\langle\omega_z\rangle}{dt}\right)_o = -\omega_o(\gamma_o + \beta_o), \tag{10}$$

since $\langle N_\theta(0,1)\rangle = 0$ and $\left\langle N_\omega^2(0,1)\right\rangle = 1$. Note that since ω_o is a sure, i.e., non-stochastic, variable

$$\langle\omega_o[\gamma(\omega_o) + \beta(\omega_o)]\rangle = \omega_o(\gamma_o + \beta_o). \tag{11}$$

Also note that in (10) and (11), we adopt the notation $\gamma_o = \gamma(\omega_o)$ and $\beta_o = \beta(\omega_o)$. Eqs. (5a) and (7a) are consistent with (10) only if

$$\left(1 + \frac{m_t}{m_f}\right) l_f^2 A_D G(l_f\omega_o) = \omega_o(\gamma_o + \beta_o). \tag{12}$$

Similar calculations show that, as expected, $d\langle\omega_x\rangle = d\langle\omega_y\rangle = 0$ and, with somewhat greater effort,

$$\left(\frac{d\langle(\omega_z - \langle\omega_z\rangle)^2\rangle}{dt}\right)_o = \delta_o^2, \tag{13}$$

and

$$\left(\frac{d\langle(\omega_x - \langle\omega_x\rangle)^2 + (\omega_y - \langle\omega_y\rangle)^2\rangle}{dt}\right)_o = 2\gamma_o\omega_o^2. \tag{14}$$

Eqs. (13) and (14) are consistent with the velocity diffusion rates (5b) and (5c) only if

$$\frac{A_D}{\omega_o} G(l_f\omega_o) = \delta_o^2, \tag{15}$$

and

$$\frac{A_D}{\omega_o}[\Phi(l_f\omega_o) - G(l_f\omega_o)] = 2\gamma_o\omega_o^2. \tag{16}$$

The three requirements (12), (15), and (16) can be solved for the three functions $\gamma(\omega)$, $\beta(\omega)$, and $\delta^2(\omega)$ that define the stochastic process (4). Doing so, we find that

$$\gamma(\omega) = \frac{A_D}{2\omega^3}[\Phi(l_f\omega) - G(l_f\omega)], \tag{17a}$$

$$\beta(\omega) = \frac{A_D}{2\omega^3}\left[G(l_f\omega)\left\{\left(1 + \frac{m_t}{m_f}\right)2\omega^2 l_f^2 + 1\right\} - \Phi(l_f\omega)\right], \tag{17b}$$

and

$$\delta^2(\omega) = \frac{A_D G(l_f\omega)}{\omega}. \tag{17c}$$

In what follows we will also need the derivative

$$\frac{d\delta^2}{d\omega} = \frac{-3A_D G(l_f\omega) + A_D l_f \omega \Phi'(l_f w)}{\omega^2}. \tag{17d}$$

2.5. Finite difference equations

A simple Euler update solution of (4) replaces differential quantities with finite differences (for example, $d\theta \to \Delta\theta$ and $dt \to \Delta t$) and evaluates all variables at the beginning of the time step Δt. An update of (4) that also includes the next higher order term, the so-called Milstein term [22], in the Taylor series expansion is

$$\theta(t+\Delta t) = \theta(t) + \sqrt{2\gamma\Delta t}N_{\theta,\Delta t}(0,1), \tag{18a}$$

$$\phi(t+\Delta t) = \phi(t) + 2\pi U_{\phi,\Delta t}(0,1), \tag{18b}$$

and

$$\omega(t+\Delta t) = \omega(t) - \beta\omega(t)\Delta t + \sqrt{\delta^2\Delta t}N_{\omega,\Delta t}(0,1) + \frac{\delta^{2'}}{4}[N_{\omega,\Delta t}(0,1)^2 - 1]\Delta t, \tag{18c}$$

where $\delta^{2'} = d\delta^2/d\omega$, and all time-dependent quantities γ, β, δ^2, and $\delta^{2'}$ on the right hand side of (18) are evaluated at time t.

Because the Milstein term $(\delta^{2'}/4)[N_{\omega,\Delta t}(0,1)^2 - 1]\Delta t$, in (18c) has mean zero and standard deviation $\delta^{2'}\Delta t/(2\sqrt{2})$ it is smaller by a factor of $\sqrt{\Delta t}$ than the stochastic term $\sqrt{\delta^2\Delta t}N_{\omega,\Delta t}(0,1)$ with mean zero and standard deviation $\sqrt{\delta^2\Delta t}$. Thus, the Milstein term may be dropped if only leading order accuracy is required. However, including the Milstein term increases the convergence order of the Euler update equation from 0.5 to 1.0 [22]. In the numerical tests reported in Section 3, we include the Milstein term although we see no discernible differences when we leave it out.

The size of the time step Δt in the finite difference equations (18) is limited by the requirement that terms left out of the Taylor series expansion that generates (18) remain small compared to those included. Kloeden and Platen [22] identify deterministic terms of order Δt^2 and stochastic terms whose standard deviation is of order $\Delta t^{3/2}$ that have been dropped in generating (18). Because each of the dropped terms has coefficients that are functions of the variable ω, a precise bound on Δt is, in general, problem dependent. Here we note that, apart from terms and bounded functions of order one, the bound on Δt scales as $\nu_1\Delta t < 1$ where ν_1 the self-relaxation collision frequency $\nu_1 = A_D l_f^3$ used in Section 3.

In fact, some of the functions $\gamma(\omega)$, $\beta(\omega)$, $\delta^2(\omega)$, and $\delta^{2'}(\omega)$ diverge in the limit $\omega \to 0$. In particular, to leading order in small ω, $\gamma(\omega) \approx (2/3\sqrt{\pi})(A_D l_f/\omega^2)$, $\beta(\omega) \approx -(3/3\sqrt{\pi})(A_D l_f/\omega^2)$, $\delta^2(\omega) \approx (2/3\sqrt{\pi})A_D l_f$, and $\delta^{2'}(\omega) \approx 0$. For this reason, the finite difference equations (18c) will boost small-ω particles to an unphysically large value of $\omega(t + \Delta t)$. We avoid this problem by noting that when ω is very small the deterministic term dominates the stochastic term in the stochastic differential equation (4c) from which (18c) or any alternative to it must be derived. Therefore, when ω is sufficiently small we drop the stochastic term from (4c) and so generate a deterministic equation of motion

$$d\omega = \frac{2A_D l_f}{3\sqrt{\pi}}\frac{dt}{\omega}, \tag{19}$$

whose solution is

$$\omega(t+\Delta t) = \sqrt{\omega^2(t) + \frac{4}{3\sqrt{\pi}}A_D l_f \Delta t}. \tag{20}$$

Comparing (20) to the Euler finite difference solution (18c) reveals the meaning of "sufficiently small". Solution (20) differs from the small-ω limit of the deterministic part of (18c) only when

$$\omega^2 \leqslant \frac{4}{3\sqrt{\pi}}A_D l_f \Delta t \tag{21}$$

where $\omega^2 = \omega^2(t)$. In terms of the multi-fluid collision frequency $\nu_o = A_D l_f^3/2$ inequality (21) becomes

$$(\omega l_f)^2 \leqslant \frac{8}{3\sqrt{\pi}}(\nu_1\Delta t). \tag{22}$$

Inequality (21) or, equivalently, (22) defines the regime in which (20) should replace (18c).

Eq. (18a) may also boost the polar angle θ to unphysically large, if normally distributed, values when ω is small, yet, because θ is an angle, this causes no conceptual or numerical problems.

2.6. From local to lab frame coordinates

Recall that scattering stretches or compresses the magnitude of the local velocity difference vector $\underline{\omega}$ from ω to $\omega_{new} = \omega + \Delta\omega$ and rotates $\underline{\omega}$ through the azimuthal and longitudinal angles $\Delta\theta$ and $\Delta\phi$. Thus, the net result of scattering is to transform $\underline{\omega}$ from the Cartesian vector $(0, 0, \omega)$ to $(\omega_{new}\sin\Delta\theta\cos\Delta\phi, \omega_{new}\sin\Delta\theta\sin\Delta\phi, \omega_{new}\cos\Delta\theta)$.

In order to shift this vector back to the lab velocity frame we must reverse the order of operations that earlier took it from the lab frame to its local velocity frame. The desired operations are: (1) rotate $\underline{\omega}_{new}$ azimuthally through θ, (2) rotate the result longitudinally through ϕ, and (3) add $\underline{v}_f$ to the result. These transformations are accomplished by [23]

$$\underline{v}(t+\Delta t) = \underline{v}_f + \begin{pmatrix} \cos\theta\cos\phi & -\sin\phi & \sin\theta\cos\phi \\ \cos\theta\sin\phi & \cos\phi & \sin\theta\sin\phi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \omega_{new}\sin\Delta\theta\cos\Delta\phi \\ \omega_{new}\sin\Delta\theta\sin\Delta\phi \\ \omega_{new}\cos\Delta\theta \end{pmatrix}. \tag{23}$$

Consequently, the lab frame velocity shift of the ith particle caused by collisions is described by

$$\Delta\underline{v}_i = \underline{v}_i(t+\Delta t) - \underline{v}_i(t). \tag{24}$$

2.7. Conservation of momentum and energy

While the expectation value of the total momentum and kinetic energy constructed from the set of lab frame velocity shifts $\Delta\underline{v}_i$, $i = 1,\ldots,N+1$, is conserved, the total momentum and kinetic energy actually realized by one set of random variables may not be conserved. One way to force conservation of total momentum and kinetic energy each time step, and so avoid a noise induced cooling or heating instability [24], is to linearly transform the lab frame post-collision velocities $\underline{v}_i(t+\Delta t)$ where $i = 1,\ldots,N+1$ in such a way as to achieve the desired effect. Such was the method of Jones et al. [17] and Manheimer et al. [18].

However, we found that linearly transforming the velocities $\underline{v}_i(t+\Delta t)$ distorts the distribution function in ways that linearly transforming the velocity increments $\Delta\underline{v}_i$ does not. For example, if collisions accelerated a few particles on the tails of the distribution to much higher velocities, conserving energy by reducing the length of the tails would be more appropriate than linearly shifting the entire distribution. Therefore, we look for a linear transformation of the velocity increments $\Delta\underline{v}_i \to \Delta\underline{v}'_i$ where

$$\Delta\underline{v}'_i = \alpha(\Delta\underline{v}_i - \underline{\beta}), \tag{25}$$

and the random variables α and $\underline{\beta}$ are chosen in order to conserve momentum and energy, that is, so that

$$\sum_{i=1}^{N+1} m_i(\underline{v}_i + \Delta\underline{v}'_i) = \sum_{i=1}^{N+1} m_i\underline{v}_i, \tag{26}$$

and

$$\sum_{i=1}^{N+1} \frac{m_i}{2}(\underline{v}_i + \Delta\underline{v}'_i)^2 = \sum_{i=1}^{N+1} \frac{m_i}{2}\underline{v}_i^2, \tag{27}$$

where $\underline{v}_i = \underline{v}_i(t)$. The solution to Eqs. (25)–(27) is

$$\alpha = \frac{-2\sum_{i=1}^{N+1} m_i\underline{v}_i \cdot \left(\Delta\underline{v}_i - \frac{\sum_{i=1}^{N+1} m_i\Delta\underline{v}_i}{\sum_{i=1}^{N+1} m_i}\right)}{\sum_{i=1}^{N+1} m_i\left(\Delta\underline{v}_i - \frac{\sum_{i=1}^{N+1} m_i\Delta\underline{v}_i}{\sum_{i=1}^{N+1} m_i}\right)^2}, \tag{28}$$

and

$$\beta = \frac{\sum_{i=1}^{N+1} m_i\Delta\underline{v}_i}{\sum_{i=1}^{N+1} m_i}. \tag{29}$$

Eqs. (28) and (29) define the linear transformation $\Delta\underline{v}_i \to \Delta\underline{v}'_i$ that preserves total momentum and kinetic energy between each step of the recursion.

3. Test problems

Here we report on two velocity space simulations that implement the particle-moment collision algorithm: one in which a single component plasma with a finite uniform velocity distribution relaxes to a Maxwellian and another in which two equal density, equal mass, Maxwellian components with different initial temperatures relax to a single Maxwellian. We study the effect of varying the time step size Δt and the number of computational particles N on each simulation. Time steps

Δt and times t are normalized to a self-collision relaxation rate $\nu_1 = A_D l_f^3$ associated with the first component if there is more than one component. Thus,

$$\nu_1 = \frac{\pi\sqrt{8}e^4 Z_1^2 n_1 \ln \Lambda}{\sqrt{m_1}T_1^{3/2}}. \tag{30}$$

We have also performed simulations of plasma components with different masses and mean velocities but do not report on these here.

3.1. Relaxation of uniform cube in velocity space

Test problem 1 simulates the relaxation of a single plasma component from an initial uniform cube in three-dimensional velocity space to a Maxwellian [17]. This process cannot be described in terms of simple functional forms. We implement it here because historically this process has quantified the numerical collision rate in PIC codes [25]. The uniform cube is extended in each velocity space direction so that $-3^{1/2} \leqslant v_j \leqslant 3^{1/2}$ where $j = x, y, z$. Given these limits $T_x = T_y = T_z = 1$ where $T_x = \langle (v_x - \langle v_x \rangle)^2 \rangle$, etc. and the bracket indicates an average over all particles. For example, the average $\langle v_x \rangle = \sum f(v_x) v_x$ where $f(v_x)$ is the fraction of particles in the velocity bin surrounding the v_x velocity and the sum is over all velocity bins in v_x space. We use 500 simulation particles to represent the distribution and a normalized time step of $\Delta t = 0.025$.

The average of the initial, that is, $t = 0$, velocity space distributions, $f(v) = (1/3)[f(v_x) + f(v_y) + f(v_z)]$, is shown in panel (a) of Fig. 2. The smooth solid curve is the Maxwellian distribution to which this distribution relaxes. The distributions $f(v)$ at various normalized times, (b) $t = 1$, (c) $t = 2$, and (d) $t = 3$, are shown in these panels. Already by $t = 1$, a Maxwellian distribution has been achieved.

Panels (a) and (b) of Fig. 3 show, respectively, histories of the kurtosis and entropy of the particles in this simulation. The kurtosis is defined by $K = \langle (v - \langle v \rangle)^4 \rangle / \langle (v - \langle v \rangle)^2 \rangle^2$ and thus is very sensitive to particles within the tails of the distribution. The initial cubic velocity distribution has a kurtosis $K = 1.6$ and evolves in panel (a) toward a Maxwellian with $K = 3$. We define the entropy with $S(t) = -\sum f_i \ln f_i$ where the sum is over velocity bins and f_i is, as above, the average distribution in one bin and plot in panel (b) the entropy difference $\Delta S = S(t) - S(0)$. Since $S(0) = \ln\left(2\sqrt{3}\right)$ and $S(\infty) = 0.5$, $\Delta S(t)$ begins at 0 and approaches 0.18. The entropy difference ΔS increases quite rapidly to $\Delta S \sim 0.12$ near $t \sim 0.5$ and thereafter more slowly.

Fig. 4, panels (a)–(d), show the effect of using different time steps Δt. In particular, panel (a) shows histories of the kurtosis on the interval $0 \leqslant t \leqslant 4$ for $\Delta t = 0.0025$ (dotted line), $\Delta t = 0.025$ (solid line), and $\Delta t = 0.25$ (dashed line). Panels (b), (c), and (d) show histograms of the distribution function at $t = 1$ for the three time steps. The smallest time step simulation, with $\Delta t = 0.0025$, does produce an equilibrium value of its kurtosis more quickly than the others, but differences among the three histograms at $t = 1$ are, apart from noise, not significant. If one's purpose is to keep the particle distributions near Maxwellian, an economically large time step, say, with $\Delta t = 0.25$, appears sufficient.

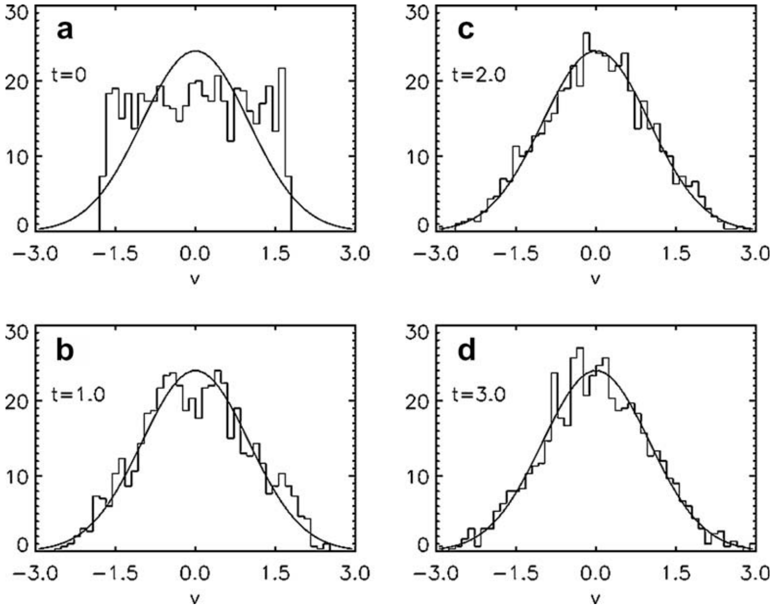

Fig. 2. Test case 1. Relaxation of uniform velocity space cube with $\nu_1 \Delta t = 0.025$ and $N = 500$. The average distribution (histogram) at: (a) $\nu_1 t = 0$, (b) $\nu_1 t = 1$, (c) $\nu_1 t = 2$, and (d) $\nu_1 t = 3$ superimposed on equilibrium Maxwellians (smooth curves).

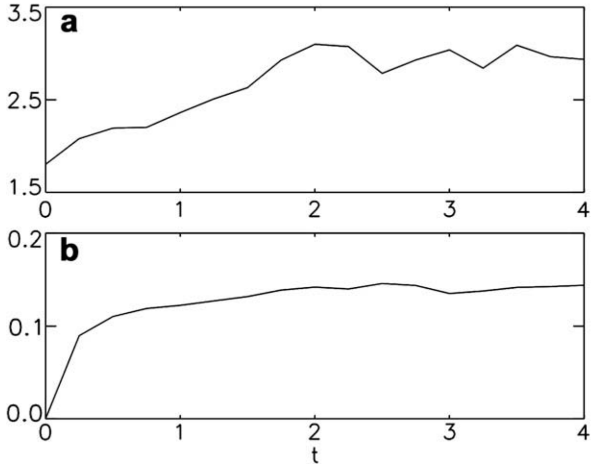

Fig. 3. Test case 1. Time histories of: (a) the kurtosis $K(t)$ and (b) the entropy increase $S(t) - S(0)$.

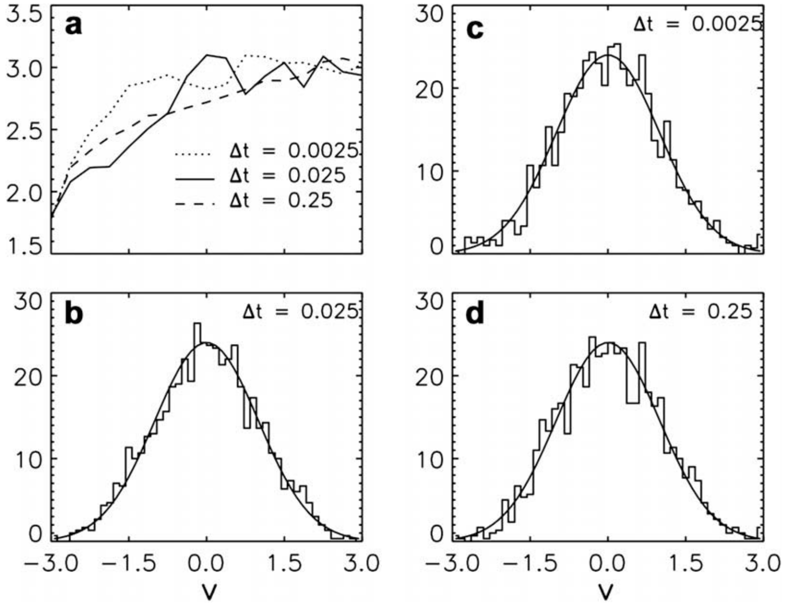

Fig. 4. Test case 1 with different time steps $\nu_1\Delta t$. Time histories of (a) the kurtosis with normalized time step $\nu_1\Delta t = 0.025$ (solid curve), $\nu_1\Delta t = 0.0025$ (dotted curve) and $\nu_1\Delta t = 0.25$ (dashed curve) and average distributions (histograms) at $\nu_1 t = 1$ for (b) $\nu_1\Delta t = 0.025$, (c) $\nu_1\Delta t = 0.0025$, and (d) $\nu_1\Delta t = 0.25$ superimposed on equilibrium Maxwellians (smooth curves).

3.2. Two-temperature relaxation

Test problem 2 simulates the relaxation of two, equal mass, $m_1 = m_2 = m$, equal density, $n_1 = n_2 = n$, plasma components with different initial temperatures, T_{1o} and $T_{2o} = 4T_{1o}$. If both components remain Maxwellian, their temperatures, T_1 and T_2, evolve according to the fluid equations

$$\frac{dT_1}{dt} = -\mu(T_1 - T_2), \tag{31a}$$

and

$$\frac{dT_2}{dt} = -\mu(T_2 - T_1). \tag{31b}$$

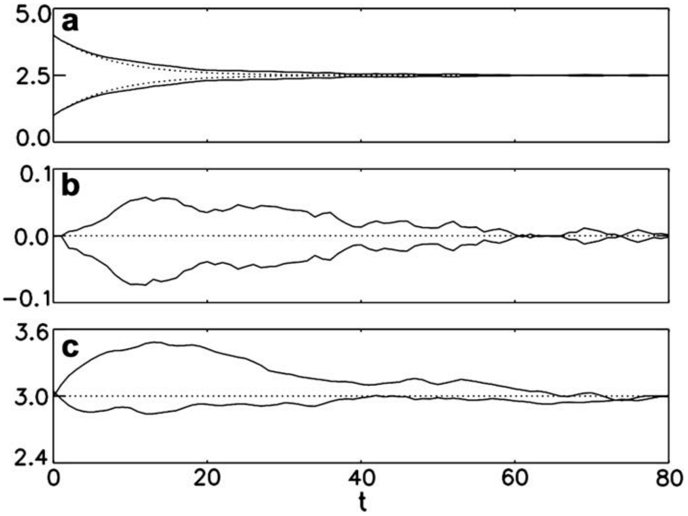

Fig. 5. Test case 2. Two-temperature relaxation with $n_2/n_1 = 1$, $m_2/m_1 = 1$, and $T_2/T_1 = 4$. Time histories of: (a) the temperatures of the two species (solid curves) and fluid theory temperatures (dotted curves), (b) particle temperature minus fluid temperature over fluid temperature for each species (solid curves), and (c) kurtosis of the two species (solid curves) and fluid theory kurtosis (dotted line).

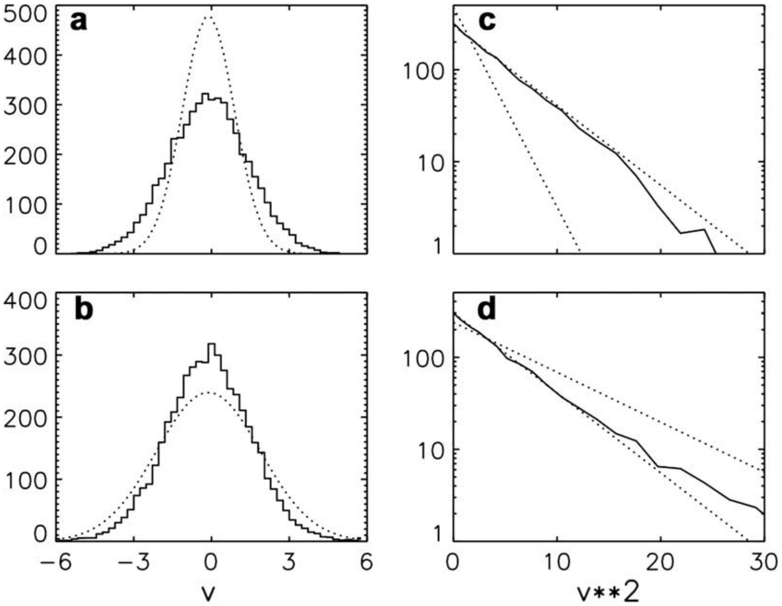

Fig. 6. Test case 2. Velocity distributions. Upper panels (a) and (b) refer to the colder component while lower panels (b) and (c) refer to the hotter component. Left panels (a) and (b) show the distribution functions at $t = 20$ histograms (solid) and initial Maxwellians (dotted), while right panels (c) and (d) are plots of $\log f(v)$ versus v^2 at $t = 20$ (solid) and initial and final Maxwellians (dotted).

These equations conserve total energy, $T_1 + T_2$, and imply a relaxation time $(2\mu)^{-1}$ given by [20,26]

$$(2\mu)^{-1} = \frac{3\sqrt{m}(T_1 + T_2)^{3/2}}{16\sqrt{2\pi} n Z^4 e^4 \ln \Lambda}. \tag{32}$$

Therefore, the normalized relaxation time $\nu_1/2\mu = 7.4$.

Figs. 5 and 6 describe simulations with 5000 particles representing each component and a time step $\Delta t = 0.25$. These particular numbers reproduce the simulation described in Fig. 1 of Rambo and Procassini [26] and by others [7,17]. The unit of time used in Rambo and Procassini, $t = 1$ ps, corresponds to our normalized time $t = 17$.

Fig. 5, panel (a), shows time histories of the temperatures of each component as determined by the particle-moment collision algorithm (solid curves) and the fluid theory described by (31) (dotted curves). In order to better show the small

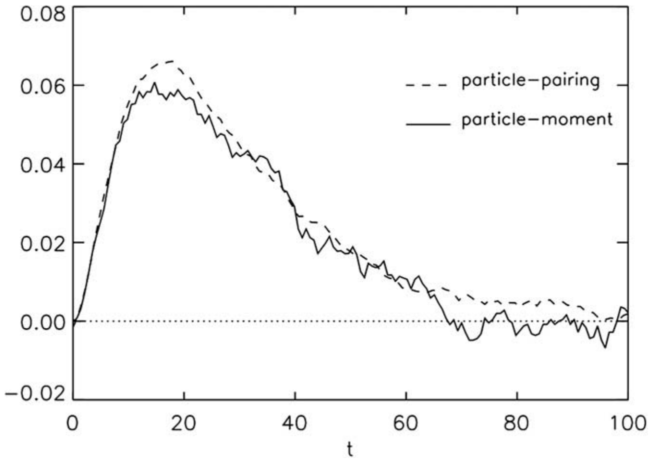

Fig. 7. Test case 2 with time step $\Delta t = 0.025$. Time histories of the particle temperature of the hotter species minus the fluid temperature of the hotter species divided by the fluid temperature of the hotter species. Particle-moment collision algorithm (solid curve) and particle-pairing collision algorithm (dashed curve) histories are shown.

difference between particle and fluid results we plot, in panel (b), the difference in the temperatures predicted by the two models divided by the fluid temperature. Note that the fluid and particle temperatures differ by at most about 5% between $t = 15$ and 35 and by less elsewhere. Presumably, the particle results are more accurate because they include small deviations from Maxwellian distributions caused both by collisions between particles from different components and by self-collisions within each component. Because energy is conserved, the sum of the two temperatures, $T_1 + T_2$, is a constant. For this reason the temperature histories in panel (b) are anti-symmetric. Panel (b) shows the kurtosis of each component as a function of time.

The four panels of Fig. 6 show average distributions at normalized time $t = 20$ – almost three relaxation times into the simulation. The top panels, (a) and (c), describe the initially colder distribution; the bottom panels, (b) and (d), describe the initially hotter distribution. The histograms on the left, in panels (a) and (b), appear by $t = 20$ to reproduce equilibrium Maxwellians, but the associated ln(f) versus v^2 plots on the right, in panels (c) and (d), show how these deviate from Maxwellians in their tails. Since the collision rate varies as v^{-3}, the tails relax more slowly than the core of the velocity distribution. Recall that, as shown in Fig. 5, panel (c), the initially hotter species maintains a relatively large kurtosis for a relatively long period of time.

Fig. 7 shows the temperature of the hotter component as determined by the particle-moment collision algorithm minus the hotter component temperature predicted by fluid theory divided by the latter for a simulation with $\Delta t = 0.025$ and $N = 20{,}000$ (solid curve). Fig. 7 also shows (dashed curve) this quantity as determined by the particle-pairing collision algorithm of Takizuka and Abe [5] implement in the VPIC particle-in-cell code running the same problem with more particles and

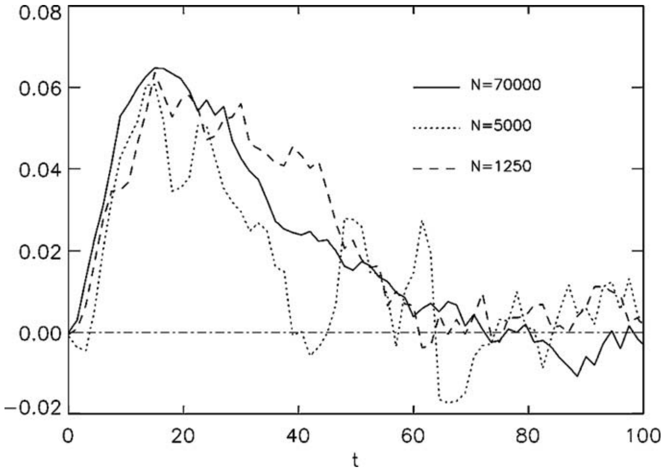

Fig. 8. Test case 2 with $\Delta t = 0.25$ and different particle numbers N. Time histories of the particle temperature of the hotter species minus the fluid temperature of the hotter species divided by the fluid temperature of the hotter species computed with the particle-moment collision algorithm. Curves are for $N = 70{,}000$ (solid), $N = 5000$ (dotted), and $N = 1250$ (dashed).

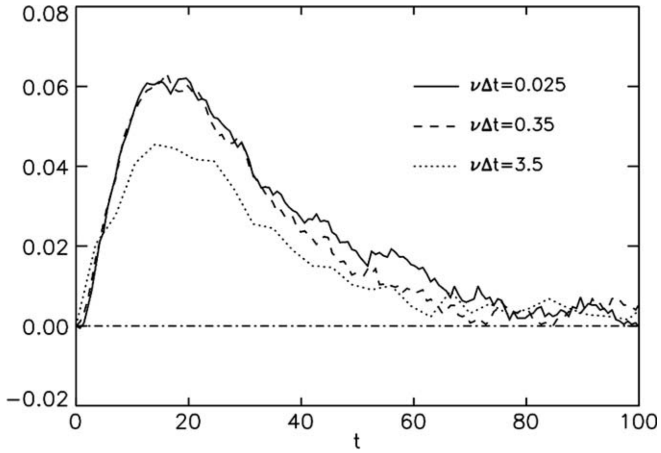

Fig. 9. Test case 2 with $N = 20{,}000$ and different time steps $\nu_1 \Delta t$ where $\nu = \nu_1$. Time histories of hotter temperature normalized to the hotter fluid temperature as computed by the particle-moment collision algorithm. Curves are for $\nu\Delta t = 0.025$ (solid), $\nu\Delta t = 0.35$ (dashed), and $\nu\Delta t = 3.5$ (dotted).

the same time step $\Delta t = 0.025$. (See the Appendix for a description of VPIC and details of this simulation.) Here we note that the particle-moment and particle-pairing algorithms produce, within noise levels, the same results.

Figs. 8 and 9 show how particle-moment collision algorithm simulations vary with component particle number N and time step Δt. Here we continue to plot the hotter particle temperature minus the hotter fluid temperature divided by the latter. Fig. 8 shows the history of this quantity when $\Delta t = 0.025$ and $N = 20{,}000$ (20,000 particles representing each component – solid curve), $N = 5000$ (reproduced from Fig. 5, panel (b) – dotted curve), and $N = 1250$ (dashed curve). Apart from the expected difference in fluctuation level these histories manifest the same trend.

Fig. 9 shows three time histories of the relative difference between the hot particle temperature and the fluid theory temperature with $N = 5000$ and $\Delta t = 0.025$ (solid curve), $\Delta t = 0.35$ (dashed curve), and $\Delta t = 3.5$ (dotted curve). Recall that $t = 7.4$ is the relaxation time. Not until the largest time step, comprising almost 50% of a relaxation time, does the temperature as calculated with the particle-moment collision algorithm deviate by as much as 20% from its converged small time step value. Note that when the particle-moment collision algorithm begins to loose integrity it does so by moving the temperature closer to that predicted by fluid theory. In contrast, according to results discussed in the Appendix, the implementation of the particle-pairing collision algorithm on VPIC running the same problem loses integrity by a similar amount, but in the opposite direction, with a time step of between 1% and 5% of a relaxation time.

4. Conclusions

We have introduced a new small-angle scattering Coulomb collision algorithm, the particle-moment method, based on the original formulation of Chandreskar [19] and Spitzer [20]. In this method, we treat the self-interaction of a cloud of N charged particles in each computational cell by representing them as a Maxwellian velocity distribution (or a series of such distributions if they the cloud deviates significantly from a single-humped velocity distribution) and considering each particle in the cloud as a test particle that interacts with the remaining particles. The particle-moment collision algorithm retains the velocity-dependent collision cross-section of the Coulomb interaction and can be made energy and momentum conserving by linearly shifting the particle velocity increments each time step.

Numerical simulations of two test problems in which plasma components relax to an equilibrium state via the particle-moment collisional algorithm converge to each other, in the limit of large particle number N and small time step Δt, and to simulations of the same interaction modeled with a particle-pairing collision algorithm. The particle-moment collision algorithm maintains integrity with time steps up to 10–50% of the relaxation time while the particle-pairing collision algorithm loses integrity in the same degree with time steps of approximately 1–5% of the relaxation time. The relative efficiency of the particle-moment algorithm is, in this case, expected. After all, the particle-moment algorithm requires N calculations each time step in order that each of N particles collisionally interact with all other particles. In contrast, the particle-pairing algorithm requires approximately $N^2/2$ calculations in order that each particle collisionally interact with all other particles. Of course, the relative effectiveness and accuracy of any collision method is, to some extent, problem dependent. Since the particle-moment collision algorithm models collisions between individual test particles and Maxwellian field particles, we suspect it may be most accurate when the plasma or plasma components are close to equilibrium. When there are significant deviations from Maxwellian distributions (e.g., bump-on-tail), a straightforward generalization, as suggested by Manheimer et al. [17] and others, is to split the distribution into several sub-distributions and model each sub-distribution as a Maxwellian. Yet more complex situations may require particle-pairing.

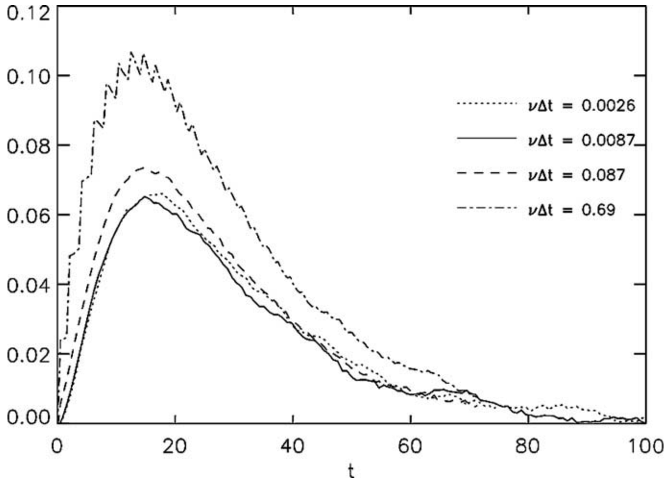

Fig. 10. Test case 2 as computed in VPIC using the particle-pairing algorithm with different time steps $\nu_1\Delta t$ where $\nu = \nu_1$. Time histories of hotter temperature normalized to the hotter fluid temperature. Curves are for: $\nu\Delta t = 0.0026$ (dotted curve), $\nu\Delta t = 0.0087$ (solid curve), $\nu\Delta t = 0.087$ (dashed curve), and $\nu\Delta t = 0.69$ (dash-dot curve).

Acknowledgments

This work is supported at Los Alamos by the NW Supporting Research Program. The authors acknowledge helpful conversations with Bruce Cohen of Lawrence Livermore National Laboratory.

Appendix. Particle-pairing collision algorithm in VPIC

VPIC is an explicit, fully relativistic, three-dimensional, particle-in-cell (PIC) simulation code carefully designed to achieve optimum performance on modern computing clusters [27]. Essentially, all high-performance PIC codes rely heavily on particle sorting to ensure data locality and to optimize cache performance. The VPIC code employs a highly efficient order N in-place counting sort [28] to order the particle array on the cell index. In addition to improving the overall performance, this sorting produces a key array that permits rapid access of the range of particle indices within each computational cell, a crucial piece of information required in particle-pairing collision algorithms.

For these reasons, the design of the VPIC code permits a highly efficient implementation of the particle-pairing algorithm [5]. However, the computational cost of performing this collision operation remains quite high and the overall performance of a large-scale simulation would be negatively impacted if the collision operation was performed every PIC time step. In order to evaluate how often the collision step may be sub-cycled while still retaining good accuracy, we simulate the temperature equilibration test problem described in Section 3.2.

The VPIC calculations were performed with $3 \times 3 \times 3$ cells, 6000 particles per cell and periodic boundary conditions. Subcycling of the particle-pairing algorithm was varied from every 30 PIC time steps ($\nu_1\Delta t = 0.0026$) up to 8000 PIC time steps ($\nu_1\Delta t = 0.69$). As illustrated in Fig. 10, the accuracy of the particle-pairing scheme is visibly degraded when the sub-cycling of the collision operation exceeds a few percent of the relaxation time. Thus, in weakly collisional regimes, it is possible to aggressively sub-cycle while retaining high accuracy, but in more collisional regimes the need to resolve the relaxation time scale will require the collision operation to be performed frequently. In these regimes, the particle-moment collision algorithm proposed here has a distinct advantage over particle-pairing.

References

[1] R.A. Bosch, R.L. Berger, B.H. Failor, N.D. Delamater, G. Charatis, R.L. Kauffman, Phys. Fluids B 4 (1992) 979.
[2] L.S. Brown, D.L. Preston, R.L. Singleton, Phys. Rep. 410 (2005) 237.
[3] F. Vidal, J.P. Matte, M. Casanova, O. Larroche, Phys. Rev. E 52 (1995) 4568.
[4] C. Castaldo, U. de Angelis, V.N. Tsytovich, Phys. Rev. Lett. 96 (2006) 075004.
[5] T. Takizuka, H. Abe, J. Comput. Phys. 25 (1977) 205.
[6] K. Nanbu, S. Yonemura, J. Comput. Phys. 145 (1998) 639.
[7] D.J. Larson, J. Comput. Phys. 188 (2003) 123.
[8] S. Ma, R.D. Sydora, J.M. Dawson, Comput. Phys. Commun. 77 (1993) 190.
[9] R.H. Miller, M.R. Combi, Geophys. Res. Lett. 21 (1994) 1735.
[10] C. Wang, T. Lin, R. Caflisch, B.I. Cohen, A.M. Dimits, J. Comput. Phys. 227 (2008) 4308.
[11] R. Caflisch, C.M. Wang, B.I. Cohen, A.M. Dimits, Multiscale Model. Simul. (2008) 865.
[12] S.E. Parker, Y. Chen, W. Wan, B.I. Cohen, W.M. Nevins, Phys. Plasmas 11 (2004) 2594.
[13] R.L. Berger, J.R. Albritton, C.J. Randall, E.A. Williams, W.L. Kruer, A. Langdon, C.J. Hanna, Phys. Fluids B 3 (1991) 3.

[14] P.W. Rambo, J. Denavit, J. Comput. Phys. 98 (1992) 317.
[15] P.W. Rambo, J. Denavit, Phys. Plasmas 1 (1994) 4050.
[16] M. Sherlock, J. Comput. Phys. 227 (2008) 2286–2292.
[17] M.E. Jones, D.S. Lemons, R.J. Mason, V.A. Thomas, D. Winske, J. Comput. Phys. 123 (1996) 169.
[18] W.M. Manheimer, M. Lampe, G. Joyce, J. Comput. Phys. 138 (1993) 563.
[19] S. Chandrasekhar, Ap. J. 97 (1943) 255.
[20] L. Spitzer Jr., Physics of Fully Ionized Gases, Wiley, New York, 1962. pp. 120–154.
[21] D.S. Lemons, An Introduction to Stochastic Processes in Physics, Johns Hopkins, Baltimore, 2002. pp. 101–102.
[22] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer, Berlin, 1992. pp. XVII–XVIII.
[23] F.W. Byron, Jr., R.W. Fuller, Mathematics of Classical and Quantum Physics, Addison-Wesley, Reading, MA, 1969, p. 5ff.
[24] D.S. Lemons, J. Lackman, Phys. Rev. E 52 (1995) 6855.
[25] J.M. Dawson, Phys. Fluids 7 (1964) 419.
[26] P. Rambo, R.J. Procassini, Phys. Plasmas 2 (1993) 3130.
[27] K.J. Bowers, B.J. Albright, L. Yin, B. Bergen, T.J.T. Kwan, Phys. Plasmas 15 (2008) 055703, 1–7.
[28] K.J. Bowers, J. Comput. Phys. 173 (2001) 393–411.