Collision operators for partially linearized particle simulation codes

Andris M. Dimits and Bruce I. Cohen
Lawrence Livermore National Laboratory, University of California, Livermore, California 94550

(Received 19 August 1993)

Algorithms are presented for energy- and momentum-conserving like-particle Coulomb collisions in partially linearized (δf) particle simulations. They are developed and implemented in particular for gyrokinetic simulation models of a strongly magnetized plasma. The collision operators include both drag and diffusion terms, are not restricted to a single or few Fourier modes, and approximately conserve both momentum and energy locally in space in a statistical sense. The first algorithm is a many-mode generalization of a test-particle–plus–source algorithm previously proposed. The second is easier to implement and improves upon the first significantly by not requiring many time steps for good conservation. Implementations for the case for ion-ion collisions are given and conservation properties are demonstrated, both directly with non-self-consistent test simulation runs and indirectly with self-consistent runs. The computational cost of particle pushing and solving for fields depends on the relative collisionality and can result in a tripling of the total computational costs if collisions are done at each time step, but typically will be a small fraction of the total simulation cost. It is also shown that binary-collision-based algorithms are unsuitable for partially linearized simulations.

PACS number(s): 52.65.+z, 52.20.Hv, 52.35.Qz, 02.70.−c

I. INTRODUCTION

Standard and partially linearized particle simulation algorithms, in their basic form, are essentially collisionless [1]. Many processes of interest in plasma physics, for which one would like to have kinetic simulation studies, involve collisions. Our effort is directed to implementing a usable energy- and momentum-conserving like-particle collision operator for partially linearized δf gyrokinetic particle computer codes because, for a given particle number, such codes have been shown to have tremendously reduced noise compared with standard fully nonlinear particle simulation codes [2].

Electron collisions off ions can often be modeled as pitch-angle scattering because of the large ion-electron mass ratio. Momentum conservation is unimportant for these collisions, and it is straightforward to construct and implement Monte Carlo models in which the energy is conserved [3,4]. These simplifications are absent for like-particle collisions, and additional steps must be taken to guarantee that the collision model conserves energy and momentum. Two approaches have been suggested to do this. The first approach [5,6] is based on the form of the linearized Landau collision operator which separates into drag-diffusion terms that constitute a test-particle operator and source-sink terms that enforce the conservation laws. The drag and diffusion terms are handled in the standard way by randomly accelerating and displacing the particles, while the source and sink terms change the particle weights so as to conserve the energy and (canonical) momentum of the particles on a spatial scale of order a grid cell size. This approach is directly usable in linearized [6] and partially linearized [7,2] δf simulations. The second approach [8–10] is a binary scheme in which the random accelerations and displacements of spatially nearby pairs of particles are correlated so as to exactly conserve the energy and momentum of each pair. This approach is directly applicable to standard fully nonlinear particle simulations. While the formulations of the collision operators are given in their respective contexts by the authors of each approach, considerations beyond those addressed there are involved in deciding which approach is to be preferred in our partially linearized simulations.

The binary algorithm of Refs. [8–10] is formulated for standard fully nonlinear simulations only, but not for partially linearized or other δf simulations. In the latter, the previous binary collision prescriptions would result in collisions that conserve the energy and momentum of the zero-order or marker particles, but not of the physical first-order (δf) energy and momentum. We have obtained and investigated the correct linearization of the binary algorithm. The resulting algorithm is impractical for simulation purposes because the number of simulation particles or "markers" increases rapidly as a function of time.

In the work of Catto and Tsang [5] and of Xu and Rosenbluth [6], the source and sink terms that are necessary for energy and momentum conservation were never implemented. Furthermore, the gyroaverages are formulated in terms of Bessel functions involving the perpendicular Fourier mode number. This is an inconvenient choice for all but linear implementations. A much more efficient procedure is to average over a finite number of points on a circle centered at the gyrocenter [11]. This necessitates a reformulation of source-sink terms in terms of explicit gyroaverages, particle-grid depositions, and field-particle interpolations. Because the deposition and interpolation steps are computationally expensive, it is necessary to consider the implementation in some detail in order to decide on its feasibility. Also, in the previous studies [5,6], the method given for the calculation of the sources is to use analytical expression for the test-particle operator. The test-particle collisions only approximate the analytic operator in the limit of many collisions. The conservation of energy and momentum in this scheme are therefore accurate only if the collision angle per time step is sufficiently small and only over regions large enough that the relative fluctuations in their evolution due to the test-particle portion of the operator are small. These requirements are unnecessarily restrictive. After our tests of the first method showed its conservation properties to be inadequate for some cases of practical interest, an alternative method for calculating these sources was developed. In this method, the momentum and energy source fields are calculated at each time step after the test-particle scattering has been applied, directly from the momentum and energy change that was produced by the test-particle scattering. The conservation is then accurate after each time step, although still only over a spatial scale associated with a region that contains many particles. This scheme is much simpler to implement than the previous scheme and gives greatly improved energy and momentum conservation. Because of the direct detailed correlation between the test-particle velocity changes and the source terms, this scheme is also more straightforward to test and debug.

The above discussion and the general methods given in this paper are applicable to partially linearized δf simulations, whether unmagnetized, magnetized with full dynamics, gyrokinetic, or drift kinetic. The specific implementations for gyrokinetic simulations involve additional complications which are addressed in detail in this paper.

We note that the collision operator very nearly triples the running time of the simulations if the simulation is dominated by pushing particles and if the collisions are applied at each time step. This is a worst case; for most applications, we anticipate that the collision operator can be applied less often than every time step, which will reduce the incremental cost of the collisions to a fraction of the total cost of the simulation. The additional memory requirements associated with the collision algorithm are acceptable.

The plan of the paper is as follows. In order to provide some context for the subsequent sections and to avoid the need to refer to many different references, some of which are unpublished, a brief summary of the main formulas of partially linearized δf and gyrokinetic particle simulation algorithms is given in Sec. II. Section III presents an argument that shows that a binary algorithm is unsuitable for partially linearized simulations. The test-particle plus source algorithms are discussed in general terms in Sec. IV. The multimode generalization of the algorithm based on the work of Catto and Tsang [5] and Xu and Rosenbluth [6] for collisions in a gyrokinetic simulation is given in Sec. V. This algorithm consists of prescriptions for both the drag and diffusion of the particle trajectories and the equation of evolution of the particle weights for the perturbed distribution function. The improved, more direct collision algorithm is described in Sec. VI. The implementations of the collision algorithms are outlined in Sec. VII, and the results of both direct tests of the conservation laws and of self-consistent test cases are presented.

II. SUMMARY OF PARTIALLY LINEARIZED AND GYROKINETIC ALGORITHMS

Here the partially linearized gyrokinetic particle simulation model is summarized for the simple case of a magnetized electrostatic plasma in a slab magnetic field. The partially linearized gyrokinetic algorithm [7] is a method for solving the small-amplitude nonlinear gyrokinetic equations [12,13]. These equations are a reduction of the Vlasov-Poisson (Vlasov-Maxwell in the electromagnetic case) equations based on the well-known gyrokinetic ordering,

$$\frac{\delta f}{F_M} \sim \frac{q_\alpha \phi}{T_\alpha} \sim \frac{\omega}{\Omega_\alpha} \sim \frac{\rho_\alpha}{L} \sim \epsilon \ll 1 \ , \quad L \sim L_\parallel \ ,$$

where $\rho_\alpha \equiv v_{T\alpha}/\Omega_\alpha$; $\Omega_\alpha \equiv q_\alpha B/m_\alpha c$; $v_{T\alpha} \equiv \sqrt{T_\alpha/m_\alpha}$; q_α, m_α, and T_α are, respectively, the charge, mass, and temperature for species α; c is the speed of light; B is the magnetic field strength; ϕ is the electrostatic potential; ω is the frequency of the perturbation; L is a characteristic perpendicular equilibrium scale length of the system; and $L_\parallel$ is the characteristic parallel wavelength of the perturbation.

The electrostatic gyrokinetic Vlasov equation for a plasma in a uniform magnetic field is [12,13]

$$\frac{\partial \delta f}{\partial t} + v_\parallel \hat{\mathbf{b}} \cdot \frac{\partial \delta f}{\partial \mathbf{R}} - \frac{c}{B} \frac{\partial}{\partial \mathbf{R}} \cdot \left\{ \left[\frac{\partial \bar{\phi}}{\partial \mathbf{R}} \times \hat{\mathbf{b}} \right] F \right\} = -\frac{q}{T} \frac{\partial \bar{\phi}}{\partial \mathbf{R}} \cdot \hat{\mathbf{b}} F_M - \kappa \frac{c}{B} \frac{\partial \bar{\phi}}{\partial y} F_M + C(\delta f) \ , \tag{1a}$$

where $\kappa \equiv -d \ln F_M/dx$, F_M is an equilibrium Maxwellian distribution function,

$$\bar{\phi} = \frac{1}{2\pi} \int d\hat{\boldsymbol{\rho}} \, \phi(\mathbf{R}+\boldsymbol{\rho}) \ , \tag{1b}$$

$\mathbf{R} \equiv \mathbf{x} - \boldsymbol{\rho}$, $\boldsymbol{\rho} \equiv v_\perp \times \hat{\mathbf{b}}$, $\hat{\boldsymbol{\rho}}$ is a unit vector in the direction of $\boldsymbol{\rho}$, $\mathbf{x}$ is the particle position, $\mathbf{v}_\perp$ is the perpendicular velocity, $\delta f(\mathbf{R}, \mu, v_\parallel, t)$ is the gyroaveraged perturbed distribution function, $\mu \equiv v_\perp^2/2$, and $C(F)$ is a collision operator. The electrostatic potential ϕ is given by the gyrokinetic Poisson equation which, for a single ion species i, is

$$\nabla^2 \phi - \frac{\tau(\phi - \tilde{\phi})}{\lambda_D^2} = -4\pi e (\bar{n}^i - n^e) \ , \tag{1c}$$

where

$$\tilde{\phi}(\mathbf{x}) \equiv \frac{1}{2\pi} \int d\mu \, d\hat{\boldsymbol{\rho}} \, \bar{\phi}(\mathbf{x} - \boldsymbol{\rho}) \ , \tag{1d}$$

$$\bar{n}(\mathbf{x}) \equiv \int d\mu \, dv_\parallel d\hat{\boldsymbol{\rho}} \, \delta f(\mathbf{x} - \boldsymbol{\rho}, \mu, v_\parallel, t) \ , \tag{1e}$$

and $\tau \equiv T_e/T_i$, $\rho_s \equiv c_s/\Omega_i$, $c_s \equiv \sqrt{T_e/m_i}$, $k_\perp$ is the perpendicular wave number, $\lambda_D \equiv \sqrt{T_e/4\pi n_0 e^2}$ is the electron Debye length, and n_0 is the background ion number density.

In a practical gyrokinetic particle codes, the angle integrations of Eqs. (1b) and (1e) are replaced by averages over a finite number (often four) points on a ring, combined with a charge-deposition or field-interpolation calculations involving spatial weighting functions [11].

The term on the right-hand side of Eq. (1a) containing κ represents the radial $\mathbf{E}\times\mathbf{B}$ advection from an equilibrium with gradients in the radial direction. In the presence of density and temperature gradients, for example, $\kappa_\alpha=\kappa_n\{1+\eta_\alpha(v^2/2v_{T\alpha}^2-\frac{3}{2})\}$, where $\eta\equiv(d\ln T_0/d\ln n_0)$ and κ_n represents the density gradient. Here $\partial/\partial x$ represents a derivative in the direction of the equilibrium gradients and $\partial/\partial y$ represents a derivative in the direction perpendicular to the magnetic field and the equilibrium gradients.

The term "partially linearized" denotes the fact that there is no parallel acceleration nonlinearity in Eq. (1a). This fact is used in the partially linearized particle simulation method by recognizing that the characteristics of Eq. (1a) preserve any spatially uniform particle distribution. We therefore use a solution of the form

$$\delta f=\sum_j w_j(t)\delta(\mathbf{R}_\perp-\mathbf{R}_{\perp j})\delta(v_\parallel-v_{\parallel j})\delta(\mu-\mu_j)\ , \tag{2}$$

where $\mathbf{R}_j$ and $v_{\parallel j}$ evolve according to the characteristic equations of Eq. (1a). In the absence of collisions, if the simulation particles j are loaded as a uniform Maxwellian then the source terms on the right-hand side of Eq. (1a) are correctly taken into account if w_j evolves according to

$$\dot{w}_j=-\frac{q}{T}\frac{\partial\bar{\phi}}{\partial\mathbf{R}}\cdot\hat{\mathbf{b}}-\kappa\frac{c}{B}\frac{\partial\bar{\phi}}{\partial y}\ .$$

The partially linearized method is a special case of the more general "δf method," which allows for different choices of particle loading and for the inclusion of nonlinearities such as the parallel acceleration nonlinearity that become important when $\delta f/F_M\ll 1$ no longer holds [2]. A generalization of the partially linearized method to include the parallel nonlinearity has also been given by Parker and Lee [14].

The remaining details of the basic particle simulation method are reviewed, for example, by Birdsall and Langdon [1], while the details specific to the interpolation, deposition and field solvers used in gyrokinetic simulations are reviewed by Lee [11].

It is the aim of this paper to obtain a momentum- and energy-conserving like-particle Monte Carlo collision operator that can be used in Eq. (1a). We use the fact that because $\delta f/F_M\ll 1$, only terms to linear order in δF need to be kept in the collision operator. Similar methods could be used to guarantee momentum and energy conservation in collision operators for nonlinear δf models where this does not hold. The main difference is that then the collision frequency will have to become a functional of the local distribution.

III. BINARY ALGORITHM

We first present some general aspects of binary collision algorithms that suggest why these may be unsuited for partially linearized particle simulations. Consider a collision between two simulation particles with initial velocities $\mathbf{v}$ and $\mathbf{v}_1$ and final velocities $\mathbf{v}'$ and $\mathbf{v}_1'$ in a three-dimensional simulation. There are six quantities (the velocity components of two particles) that can change and four constraints (the total energy and three components of the total momentum). Thus there are two free parameters that can change, often chosen as $\delta\theta$, the angle between the final and initial relative velocities, and ϕ, some azimuthal angle (e.g., the angle between the plane containing the initial and final relative velocities and the line formed by the intersection between the plane perpendicular to the initial relative velocity and some fixed plane). Consider a partially linearized code binary collision in which there are two zero-order ("marker") particles before and after the collision. There are now two additional free parameters that can change, the particle weights. Detailed conservation would demand the conservation of the total zero- and first-order momentum and energy, and the first-order particle number, i.e., a total of nine constraints. Clearly, therefore, detailed conservation, analogous to that for binary collisions in a standard code, is impossible.

It is interesting to see how conservation laws are satisfied for the linearized Boltzmann collision operator, which one would expect to describe binary collisions with sufficient accuracy for partially linearized simulations. The full Boltzmann collision operator can be written as [15]

$$C_B[f](\mathbf{v})=\int d\mathbf{v}_1 d\mathbf{v}' d\mathbf{v}_1'\{W(\mathbf{v},\mathbf{v}_1|\mathbf{v}',\mathbf{v}_1')f(\mathbf{v}')f(\mathbf{v}_1') -W(\mathbf{v}',\mathbf{v}_1'|\mathbf{v},\mathbf{v}_1)f(\mathbf{v})f(\mathbf{v}_1)\}\ , \tag{3}$$

where $W(\mathbf{v}',\mathbf{v}_1'|\mathbf{v},\mathbf{v}_1)d\mathbf{v}\,d\mathbf{v}_1 d\mathbf{v}' d\mathbf{v}_1'$ is the transition rate for collisions in which particles with initial velocities $\mathbf{v}$ and $\mathbf{v}_1$ have final velocities $\mathbf{v}'$ and $\mathbf{v}_1'$, respectively, and the arguments $\mathbf{x}$ and t, upon which $C_B[f]$ and f also depend, have been suppressed. This can be linearized to give

$$\delta C_B[f](\mathbf{v})=\int d\mathbf{v}_1 d\mathbf{v}' d\mathbf{v}_1'\{W(\mathbf{v},\mathbf{v}_1|\mathbf{v}',\mathbf{v}_1')\delta f(\mathbf{v}')f_0(\mathbf{v}_1')+f_0(\mathbf{v}')\delta f(\mathbf{v}_1')] -W(\mathbf{v}',\mathbf{v}_1'|\mathbf{v},\mathbf{v}_1)[\delta f(\mathbf{v})f_0(\mathbf{v}_1)+f_0(\mathbf{v})\delta f(\mathbf{v}_1)]\}\ .$$

We now use the ansatz that results in the standard-loading partially linearized algorithm $f_0=\sum_i\delta(\mathbf{v}-\mathbf{v}_i)$, $\delta f=\sum_i w_i(t)\delta(\mathbf{v}-\mathbf{v}_i)$, where $w_i(t)$ are the particle weights, and note that a particle does not collide with itself, to obtain

$$\delta C_B[f](\mathbf{v})=\sum_i\sum_{j(\neq i)}(w_i+w_j)\left\{\int d\mathbf{v}_1 W(\mathbf{v},\mathbf{v}_1|\mathbf{v}_i,\mathbf{v}_j)-\delta(\mathbf{v}-\mathbf{v}_i)\int d\mathbf{v}' d\mathbf{v}_1' W(\mathbf{v}',\mathbf{v}_1'|\mathbf{v}_i,\mathbf{v}_j)\right\}\ . \tag{4}$$

Now consider a gas consisting of two particles only, and formally integrate over a time interval $[t_0, t_0+\tau]$ such that

$$\int_{t_0}^{t_0+\tau} dt\, W(\mathbf{v},\overline{\mathbf{v}}|\mathbf{v}_1,\mathbf{v}_2)=\delta(\mathbf{v}-\mathbf{v}_1')\delta(\overline{\mathbf{v}}-\mathbf{v}_2') , \tag{5}$$

where $\mathbf{v}_1'$ and $\mathbf{v}_2'$ are two velocities that can be obtained from $\mathbf{v}_1$ and $\mathbf{v}_2$ by a standard energy- and momentum-conserving collision. The result is

$$\begin{aligned}\int_{t_0}^{t_0+\tau} dt\, \delta C_B[f](\mathbf{v}) &= (w_1+w_2)\{\delta(\mathbf{v}-\mathbf{v}_1')-\delta(\mathbf{v}-\mathbf{v}_1) \\ &\quad +\delta(\mathbf{v}-\mathbf{v}_2')-\delta(\mathbf{v}-\mathbf{v}_2)\} .\end{aligned} \tag{6a}$$

This result clearly conserves particle number, energy and momentum because its form is that of a number multiplied by an expression that represents a standard collision. It can be generalized to collisions between many pairs of particles by summing over colliding pairs. Adding the distribution before the collision, we obtain

$$\begin{aligned}\delta f(\mathbf{v},t_0+\tau) &= (w_1+w_2)[\delta(\mathbf{v}-\mathbf{v}_1')+\delta(\mathbf{v}-\mathbf{v}_2')] \\ &\quad -w_2\delta(\mathbf{v}-\mathbf{v}_1)-w_1\delta(\mathbf{v}-\mathbf{v}_2) .\end{aligned} \tag{6b}$$

The key feature of Eq. (6b) is that markers both at the new and old velocities are required to represent the final distribution function. The additional free parameters made available by the additional marker particles are necessary to permit the collisions both to conserve number, energy, and momentum and to permit a prescription of the collision rate. Although the marker at one of the old velocities could be eliminated by adding some multiple of Eq. (6), this would result in an unpredictable collision rate. For a fixed collision rate, the number of simulation particles must increase exponentially as a function of time. This makes the binary algorithm totally unusable for partially linearized simulations.

IV. TEST-PARTICLE PLUS SOURCE/SINK ALGORITHMS

For the reason stated in the preceding section, we develop collision operators based on the test-particle plus source approach. Here these algorithms are discussed in general terms.

The Landau collision operator can be written in the form

$$C(f)=\overline{C}[f]f ,$$

where $\overline{C}[f]g$ denotes a linear operator, which is a linear functional of f, acting on g. In addition, $\overline{C}$ has the form of a drag-diffusion operator. The functional dependence of the drag and diffusion coefficients in $\overline{C}[f]$ on f makes these coefficients difficult to compute when f evolves. This difficulty is greatly reduced in partially linearized codes, because in the small-δf ordering used there, the operator can be linearized. The linearization is valid provided $\nu \ll \omega L/\lambda$, where ν is the collision frequency, ω is the characteristic timescale of the fluctuations under study, L is the radial equilibrium length scale, and λ is the radial scale of the fluctuations. As an example, for electrostatic drift-wave problems, $\omega L/\lambda$ is of order the ion gyrofrequency. Carrying out the linearization about a Maxwellian F_M, we have [5,6]

$$\begin{aligned}\delta(\overline{C}[f]f) &= \overline{C}[F_M]\delta f+\overline{C}[\delta f]F_M \\ &= C_{\mathrm{TP}}(\delta f)+p(\delta f)F_M .\end{aligned} \tag{7}$$

Here C_{TP} is a test-particle drag-diffusion operator in velocity space, with drag and diffusion coefficients that do not vary with time and position, $p(\delta f)\equiv(\overline{C}[\delta f]F_M)/F_M$, and $p(\delta f)F_M$ is a source that guarantees local energy and momentum conservation. The source $p(\delta f)$ can be expanded in functions orthogonal with respect to the (one-, two-, or three-dimensional) weighting function F_M. Catto and Tsang [5] and Xu and Rosenbluth [6] note that number conservation is automatic for C_{TP} and that to obtain momentum and energy conservation, only the first two terms of $p(\delta f)$ need to be kept. Thus we can take

$$p(\delta f)=\mathbf{v}\cdot\mathbf{p}(\mathbf{x})+\lambda(\mathbf{x})(v^2/2v_T^2-\tfrac{3}{2}) , \tag{8a}$$

where

$$\mathbf{p}(\mathbf{x})=-\frac{1}{n_0 v_T^2}\int d\mathbf{v}\,\mathbf{v}C_{\mathrm{TP}}(\delta f) , \tag{8b}$$

$$\lambda(\mathbf{x})=-\frac{1}{3n_0 v_T^2}\int d\mathbf{v}\,v^2C_{\mathrm{TP}}(\delta f) . \tag{8c}$$

The two possible approaches mentioned in the Introduction amount to particular choices of C_{TP} in Eqs. (8). In the approach of Catto and Tsang [5] and Xu and Rosenbluth [6], which results in what will here be called "algorithm I," the continuous analytical velocity-space drag-diffusion operator expression for C_{TP} is used. In a particle code, the test-particle collisions are actually implemented with a Monte Carlo model in which the particle velocities are changed stochastically. The stochastic velocity kicks approximate the continuous analytical operator only in the limit of many kicks. Thus the conservation of energy and momentum are accurate only in the limit of many kicks. In our approach, which will be termed "algorithm II," and which is developed in detail for gyrokinetic simulations in Sec. VI, the actual velocity kicks that cause the velocity-space drag and diffusion in the Monte Carlo model are used directly to calculate the sources. We can formally integrate Eqs. (8) over the time interval during which the collision operator is applied once in the particle code and define $\Delta_c\delta f$ to be the change in δf after one application of the collision operator and $\Delta_{C\mathrm{TP}}\delta f$ to be that portion due to the test-particle scattering (kicks applied to the velocities of the marker particles). We then have

$$\Delta_c\delta f=\Delta_{C\mathrm{TP}}\delta f+\Delta p F_M , \tag{9a}$$

where

$$\Delta p(\mathbf{x},\mathbf{v})=\mathbf{v}\cdot\Delta\mathbf{p}(\mathbf{x})+\Delta\lambda(\mathbf{x})(v^2/2v_T^2-\tfrac{3}{2}) \tag{9b}$$

and

$$\Delta \mathbf{p}(\mathbf{x}) = -\frac{1}{n_0 v_T^2} \int d\mathbf{v}\, \mathbf{v}\, \Delta_{CTP} \delta f \,, \tag{9c}$$

$$\Delta \lambda(\mathbf{x}) = -\frac{1}{3 n_0 v_T^2} \int d\mathbf{v}\, v^2 \Delta_{CTP} \delta f \,. \tag{9d}$$

In this approach the sources are correlated with the velocity kicks and conservation of energy and momentum is imposed for each collision time step.

V. ALGORITHM I BASED ON THE APPROACH OF XU AND ROSENBLUTH

Gyroaveraging in the first of the test-particle plus source schemes, which gives a multimode version of the operator derived by Catto and Tsang [5] and Xu and Rosenbluth [6], is discussed in this section.

The linearized Landau collision operator is given by Eq. (7) with C_{TP} and $p(\delta f)$ given by [6]

$$C_{\text{TP}}(\delta f) = \frac{\partial}{\partial v_\perp^2}\left[\nu_{s\perp} v^2 \delta f + \frac{1}{2}\frac{\partial}{\partial v_\perp^2}(\nu_\perp v^4 \delta f) + \frac{1}{2}\frac{\partial}{\partial v_\|}(\nu_{\|\perp} v^3 \delta f)\right] + \frac{\partial}{\partial v_\|}\left[\nu_{s\|} v_\| \delta f + \frac{1}{2}\frac{\partial}{\partial v_\|}(\nu_\| v^2 \delta f) + \frac{\partial}{\partial v_\perp^2}(\nu_{\|\perp} v^3 \delta f)\right] + \frac{1}{v_\perp}\frac{\partial}{\partial \varphi}\left[\frac{1}{v_\perp}\frac{\partial}{\partial \varphi}(G\delta f)\right] \tag{10}$$

and

$$p(\delta f) = \frac{1}{n_0 v_T^2}\left[\mathbf{v}\cdot \int d\mathbf{v}\, \mathbf{v} F \delta f + \frac{2}{3}\left[\frac{v^2}{v_T^2} - \frac{3}{2}\right]\int d\mathbf{v} (v^2 F - 3G - H)\delta f\right] , \tag{11}$$

where $\nu_{s\|}$, $\nu_{s\perp}$, $\nu_\|$, $\nu_{\|\perp}$ F, G, and H are functions of $v_\|$ and $v_\perp$ defined in Ref. [6]. Equation (10) has been expressed as the divergence of the collisional velocity-space flux. If δf is a Maxwellian with zero mean velocity and the same temperature as that for which the collisional drag and diffusion coefficients is evaluated, then each of the groups of terms in the square brackets in Eq. (10) sums to zero. First, δf is expressed in terms of the gyroaveraged distribution function g, accurate to first order in the gyrokinetic smallness parameter [13], via

$$\delta f = h - \frac{q}{T} F_M \phi \,, \tag{12a}$$

where

$$h = g + \frac{q}{T} F_M \bar{\phi} \,, \tag{12b}$$

and $\bar{\phi}$ is the gyroaveraged potential. The $(q/T)F_M\phi$ term in Eq. (12a) does not contribute to either term in the collision operator. Next, the spatial position is expressed in terms of the lowest-order gyrocenter position, $\mathbf{R} \equiv \mathbf{x} - \boldsymbol{\rho}$, where $\boldsymbol{\rho} \equiv \hat{\mathbf{b}} \times \mathbf{v}/\Omega$. This requires the following transformation formulas for the partial derivatives:

$$\left.\frac{\partial}{\partial v_\perp^2}\right|_x = \left.\frac{\partial}{\partial v_\perp^2}\right|_R - \frac{\boldsymbol{\rho}}{2 v_\perp^2}\cdot\frac{\partial}{\partial \mathbf{R}} \,, \tag{13a}$$

$$\left.\frac{\partial}{\partial \varphi}\right|_x = \left.\frac{\partial}{\partial \varphi}\right|_R - (\boldsymbol{\rho}\times\hat{\mathbf{b}})\cdot\frac{\partial}{\partial \mathbf{R}} \,. \tag{13b}$$

Finally, the collision operator is gyroaveraged, i.e., averaged with respect to φ with $\mathbf{R}$ held fixed. The various contributions will be considered separately. In gyroaveraging the $C_{\text{TP}}(g)$ term, classical transport contributions arise from the $\partial/\partial\mathbf{R}$ terms in Eqs. (13). The result is

$$\langle C_{\text{TP}}(g)\rangle = \frac{\partial}{\partial v_\perp^2}(\nu_{s\perp} v^2 g) + \frac{\partial}{\partial v_\|}(\nu_{s\|} v_\| g) + \frac{1}{2}\frac{\partial^2}{\partial (v_\perp^2)^2}(\nu_\perp v^4 g) + \frac{1}{2}\frac{\partial^2}{\partial v_\|^2}(\nu_\| v^2 g) + \frac{\partial^2}{\partial v_\perp^2 \partial v_\|}(\nu_{\|\perp} v^3 g) + \frac{1}{2 v_\perp^2}\rho^2 \nabla_R^2 \left[\left[\frac{v^4 \nu_\perp}{8 v_\perp^2} + G\right] g\right] ,$$

where $\partial/\partial v_\perp^2$ is now evaluated at fixed $\mathbf{R}$. This is a test-particle drag-diffusion operator acting on g, which is the quantity that is evaluated along the particle characteristics. This piece can therefore be modeled using standard Monte Carlo techniques [6,16]. The gyroaverage of $C_{\text{TP}}(F_M q\bar{\phi}/T)$ represents the effect of collisions on the portion of the distribution function that is due to the polarization drift. Inserting $F_M\bar{\phi}$ into Eq. (10) and noting that (a) the components of the collisional phase-space flux, of which $C_{\text{TP}}(F_M)$ is a divergence, vanish and (b) $\partial\bar{\phi}/\partial v_\| = 0$ gives

$$C_{\text{TP}}(F_M\bar{\phi}) = F_M \left[-\nu_{s\perp} v^2 \frac{\partial\bar{\phi}}{\partial v_\perp^2} + \frac{1}{2}\nu_\perp v^4 \frac{\partial^2\bar{\phi}}{\partial (v_\perp^2)^2} + \frac{G}{v_\perp^2}\frac{\partial^2\bar{\phi}}{\partial\varphi^2}\right] , \tag{14}$$

where $\partial/\partial v_\perp^2$ and $\partial/\partial\varphi$ are evaluated at constant $\mathbf{x}$. Again, transforming the spatial coordinate from $\mathbf{x}$ to $\mathbf{R}$, using Eqs. (13) and gyroaveraging gives

$$\langle C_{\text{TP}}(F_M\bar{\phi})\rangle = F_M \left[-\nu_{s\perp} v^2 \frac{\partial\bar{\phi}}{\partial v_\perp^2} + \frac{1}{2}\nu_\perp v^4 \frac{\partial^2\bar{\phi}}{\partial (v_\perp^2)^2} + \left[\frac{v^4 \nu_\perp}{8 v_\perp^4} + \frac{G}{v_\perp^2}\right]\frac{1}{2}\rho^2 \nabla_R^2 \bar{\phi}\right] , \tag{15}$$

where $\partial/\partial v_\perp^2$ is again now evaluated at fixed $\mathbf{R}$. The method for evaluating each of these terms is as follows. For the first term, we have

$$\frac{\partial\bar{\phi}}{\partial v_\perp^2}=\frac{1}{2v_\perp\Omega}\frac{1}{2\pi}\oint d\varphi\,\hat{\boldsymbol{\rho}}\cdot\nabla_\perp\phi(\mathbf{R}+\boldsymbol{\rho})\ .$$

This can be evaluated approximately by replacing the integral by a sum over a finite number of values of φ. Interpolation is used to obtain $\nabla_\perp\phi(\mathbf{R}+\boldsymbol{\rho})$ from the values of $\nabla_\perp\phi$ given on the grid (as for the standard electric field calculation). In order to simplify the second term, note that $\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\rho}}+(\hat{\boldsymbol{\rho}}\times\hat{\mathbf{b}})(\hat{\boldsymbol{\rho}}\times\hat{\mathbf{b}})=\mathbf{I}-\hat{\mathbf{b}}\hat{\mathbf{b}}$, $\hat{\mathbf{b}}\times\hat{\boldsymbol{\rho}}=d\hat{\boldsymbol{\rho}}/d\varphi$, and integrate by parts to obtain

$$\frac{\partial^2\bar{\phi}}{\partial(v_\perp^2)^2}=\frac{1}{4v_\perp^3}\left[-\frac{1}{v_\perp^2}\frac{\partial\bar{\phi}}{\partial v_\perp^2}+\frac{1}{4v_\perp^4}\rho^2\nabla_R^2\bar{\phi}\right]\ .$$

Because the Laplacian commutes with the gyroaveraging, $\nabla_R^2\bar{\phi}$ can be evaluated by gyroaveraging over values of $\nabla^2\phi$ obtained by interpolation from values known on the grid, having evaluated $\nabla^2\phi$ on the grid using grid operations. This also takes care of the last term in Eq. (15). The result is a gyrocenter quantity that contributes a source at the gyrocenter position. We note that our Eq. (15) in the single-mode limit does not agree with the corresponding expressions in Eq. (7) of Ref. [6].

The contribution of Eq. (11) can be calculated by noting that

$$\begin{aligned}\langle p(\delta f)\rangle(\mathbf{R})={}&v_\parallel\langle p_\parallel(h)\rangle+\langle\mathbf{v}_\perp\cdot\mathbf{p}_\perp(h)\rangle\\&+(v^2/v_T^2-\tfrac{3}{2})\langle\lambda(h)\rangle\ ,\end{aligned}\tag{16a}$$

where

$$p_\parallel(h;\mathbf{x})\equiv\frac{1}{n_0v_T^2}\int d^3v'v'_\parallel F'h(\mathbf{x}-\boldsymbol{\rho}')\ ,\tag{16b}$$

$$\mathbf{p}_\perp(h;\mathbf{x})\equiv\frac{1}{n_0v_T^2}\int d^3v'\mathbf{v}'_\perp F'h(\mathbf{x}-\boldsymbol{\rho}')\ ,\tag{16c}$$

$$\lambda(h;\mathbf{x})\equiv\frac{2}{3n_0v_T^2}\int d^3v'(v'^2F'-3G'-H')h(\mathbf{x}-\boldsymbol{\rho}')\tag{16d}$$

are $\mathbf{x}$-space fields.

The quantities in Eqs. (16b)–(16d) are computed by noting that h is a gyroaveraged quantity and so can be represented by adding the polarization contribution $(q/T)F_M\bar{\phi}(\mathbf{x}-\boldsymbol{\rho}')$ to the particle weights. They are then multiplied by the respective functions of velocity to form the quantities in the integrands in Eqs. (16b)–(16d), distributed over a finite number of values of $\boldsymbol{\rho}'$ on a ring, and deposited to the grid. The resulting grid quantities are then interpolated onto a finite number of points on the ring around each gyrocenter, multiplied by the appropriate functions of $v_\parallel$ and $\mathbf{v}_\perp$ in Eq. (16a), evaluated at the values for the respective gyrocenter, and summed to produce a source that is a gyrocenter quantity.

We note that a formal procedure for gyroaveraging the collision operator has been given in terms of Lie transforms by Brizard [17]. To the order with which we are interested, Brizard's results amount to the above procedure.

VI. COLLISION ALGORITHM II: A MORE DIRECT APPROACH

In the collision algorithm detailed in the preceding section, the prescription for $\langle p(\delta f)\rangle$ and $\langle C_{\mathrm{TP}}(F_Mq\bar{\phi}/T)\rangle$ should be accurate in the limit that the particle distribution is adequately resolved with a sufficient number of marker particles and after a sufficient number of collisions, each of sufficiently small collision angle, to ensure good statistics.

In practice, as will be seen in the test cases given in Sec. VII, this approach has two problems. First, for strong collisionality, the requirement of sufficiently many collisions can necessitate many collisions per gyrokinetic time step. Relevant test cases will be shown in which one collision per gyrokinetic time step gives poor conservation. Second, very good grid resolution may be necessary to ensure that the source term of Eq. (15) results in reasonable energy and momentum conservation. This is because these terms are not in a form that is explicitly energy and momentum conserving when combined with $C_{\mathrm{TP}}(F_M\bar{\phi})$ and discretized. The test-particle portion, given by Eq. (15), contains second derivatives of $\bar{\phi}$, while the polarization response portion of Eqs. (16) is in integral form. Finally, the form. derivation, and implementation of Eq. (15) are sufficiently complicated to make it difficult to verify that the final implementation is correct. These difficulties motivated the development of the second, more direct implementation of the collision algorithm. The details of implementing this operator in a gyrokinetic simulation are presented in this section. This operator involves fewer computations and requires only that there be sufficient numbers of marker particles to resolve the velocity distribution. As already stated, the drag and diffusion of the marker particles are produced in the same way as for the first algorithm. However, the calculations of the $\langle p(\delta f)\rangle$ and $\langle C_{\mathrm{TP}}(F_Mq\bar{\phi}/T)\rangle$ source terms in the second algorithm differ.

The $\langle p(\delta f)\rangle$ momentum- and energy-conserving source term can be calculated from the appropriately gyroaveraged and weighted deflections in the momenta $\Delta\mathbf{v}_i$ and changes in the kinetic energy $\Delta(v_i^2)$ of the marker particles due to collisions. To be specific, the change in the particle weights for momentum and energy conservation $\Delta w_i^{\mathrm{cons}}$ is given directly by

$$\Delta w_i^{\mathrm{cons}}=-\langle\mathbf{v}_i\cdot\Delta\mathbf{p}\rangle-\langle[(v_i^2/v_T^2)-\tfrac{3}{2}]\Delta\lambda\rangle\ ,\tag{17}$$

where

$$\Delta\mathbf{p}=(1/n_0v_T^2)\sum_i(w_i+q\bar{\phi}^i/T)\sum_j\Delta\mathbf{v}_{ij}W(\mathbf{R}_i+\rho_i\hat{\boldsymbol{\rho}}_j-\mathbf{x}_g)\ ,\tag{18a}$$

$$\Delta\lambda=(1/3n_0v_T^2)\sum_i(w_i+q\bar{\phi}^i/T)\sum_j\Delta v_{ij}^2W(\mathbf{R}_i+\rho_i\hat{\boldsymbol{\rho}}_j-\mathbf{x}_g)\tag{18b}$$

are collected on the spatial grid from the marker particle deflections due to collisions at the gyro-orbit points. Here i is the gyrocenter index, j is the gyro-orbit point index which in our codes runs from 1 to 4, $\bar{\phi}^i$ is the gyroaveraged potential seen by the ith gyrocenter, W is a spatial weighting function of finite size that specifies the algorithm by which particle data are deposited onto the simulation grid, $\mathbf{R}_i$ and ρ_i are, respectively, the position and gyroradius of the ith gyrocenter, $\hat{\rho}_j$ is the unit vector of the jth orbit point, and $\mathbf{x}_g$ is the grid point position. The right-hand side of Eq. (17) is similarly evaluated as a four-point gyroaverage by interpolating from the values of $\Delta\mathbf{p}$ and $\Delta\lambda$ on the grid to the particle position $\mathbf{R}+\boldsymbol{\rho}$. The fact that the marker particle deflections $\Delta\mathbf{v}_i$ are *not* saved at the four angles around the gyro-orbit, but are immediately used to increment the marker particle velocities, the perpendicular gyrocenter positions, and the momentum and energy change fields, is exploited to allow the order of the four gyroangles to be scrambled randomly in computing the gyroaverages to destroy any spurious anisotropy.

The change in the particle weight $\Delta w_i^{\rm pol}$ due to the collisionally induced change deriving from $q\langle C_{\rm TP}(F_M\bar{\phi})\rangle/T$ can be directly calculated from the change in $\bar{\phi}$ calculated before and after the collisional deflection of the marker particles. This is much less cumbersome than the procedure presented in Sec. III. Consider the perturbed ion velocity distribution function after a collision as a sum of gyrocenter and polarization pieces, $\delta f_2 = g_2 + q(\bar{\phi}_2-\phi)F_M/T$, which is related to the corresponding quantities before the collision by the equation

$$\delta f_2 = g_1 + q(\bar{\phi}_1-\phi)F_M/T + \Delta\delta f_c \ , \tag{19}$$

where g_1 and g_2 are the gyrocenter velocity distribution functions before and after the collision, and $\Delta\delta f_c$ is the collisional change due to $\langle p(\delta f)\rangle$ and the collisional deflections of the marker particles. Thus the contribution to g_2-g_1 from just the collisional change in $\bar{\phi}$ is deduced from

$$g_2 - g_1 = -q(\bar{\phi}_2-\bar{\phi}_1)F_M/T + \Delta\delta f_c \ .$$

Hence the contribution just from the change in $\bar{\phi}$ is

$$\Delta w_i^{\rm pol} = -q(\bar{\phi}_2-\bar{\phi}_1)/T \ . \tag{20}$$

The quantity $\bar{\phi}_2-\bar{\phi}_1$ represents the change in the gyroaveraged potential seen by a particle due to the collisional change in its gyroorbit, in the absence of any change in the potential itself. Note that because each collision event does not change the *particle* position, there is no $\phi_2-\phi_1$ term. Also, F_M is represented by the marker particle distribution, which is preserved by the collisions.

Because the sign of Eq. (20) is somewhat counterintuitive and it is not obvious that this result is equivalent to $C_{\rm TP}(F_M\bar{\phi})$ as used in algorithm I, a direct derivation is given here. The δf realization of $F_M\bar{\phi}$ can be written as

$$F_M\bar{\phi} \rightarrow \sum_i \bar{\phi}(X_i)\delta(X-X_i) \ ,$$

where $X\equiv(\mathbf{R},\mu,v_\parallel)$ is a five-vector consisting of the gyrocenter phase-space coordinates and X_i is its value for particle i. The test-particle scattering operator, by definition, changes the phase space coordinates without changing the particle weight. The change due to a collision is therefore

$$\Delta C_{\rm TP}(F_M\bar{\phi}) = \sum_i \bar{\phi}(X_i)[\delta(X-(X_i+\Delta X_i))-\delta(X-X_i)] \ .$$

This can be written as

$$\begin{aligned}\Delta_{C{\rm TP}}(F_M\bar{\phi}) &= \sum_i \bar{\phi}(X_i+\Delta X_i)\delta(X-(X_i+\Delta X_i)) - \sum_i \bar{\phi}(X_i)\delta(X-X_i) + \sum_i[\bar{\phi}(X_i)-\bar{\phi}(X_i+\Delta X_i)]\delta(X-(X_i+\Delta X_i)) \\ &= \bar{\phi}(X)\left[\sum_i \delta(X-(X_i+\Delta X_i)) - \sum_i \delta(X-X_i)\right] + \sum_i[\bar{\phi}(X_i)-\bar{\phi}(X_i+\Delta X_i)]\delta(X-(X_i+\Delta X_i)) \ .\end{aligned} \tag{21}$$

The first term in square brackets on the right-hand side of Eq. (21) contains the difference between two realizations of F_M and is therefore neglected. This gives the result

$$\Delta_{C{\rm TP}}(F_M\bar{\phi}) \rightarrow \sum_i[\bar{\phi}(X_i)-\bar{\phi}(X_i+\Delta X_i)]\delta(X-(X_i+\Delta X_i)) \ ,$$

which is equivalent to Eq. (20).

The computation of $\bar{\phi}$ just involves a four-point gyroaverage at $\mathbf{R}+\boldsymbol{\rho}$. Overall, the second collision algorithm is simpler and more direct, and should perform better than the first algorithm presented in Sec. III.

VII. CODE IMPLEMENTATION AND TEST CASES

A. Collision algorithm I

The implementation of the drag-diffusion collision operator breaks naturally into three parts: $\langle C_{\rm TP}(g)\rangle$, $\langle C_{\rm TP}(F_M q\bar{\phi}/T)\rangle$, and $\langle p(\delta f)\rangle$. The first part $\langle C_{\rm TP}(g)\rangle$ is the gyroaveraged drag-diffusion operator acting on the test-particle velocity components evaluated along the characteristics. The second and third parts contribute source and sink terms to the equation of evolution for the particle weight used in representing g, i.e.,

$$g = \sum_i w_i(t)\delta(\mathbf{v}-\mathbf{v}_i)\delta(\mathbf{R}-\mathbf{R}_i) \ , \tag{22}$$

where $F_M=\sum_i\delta(\mathbf{v}-\mathbf{v}_i)\delta(\mathbf{R}-\mathbf{R}_i)$. The evolution equation for the particle weight $w_i(t)$ is

$$\frac{d}{dt}w_i(t) = F_M^{-1}\langle C_{\rm TP}(F_M q\bar{\phi}/T)\rangle + \langle p(\delta f)\rangle \ , \tag{23}$$

where Eqs. (15) and (16) are used for the right-hand side of Eq. (23) and dw_i/dt is calculated along the particle trajectories as influenced by $\langle C_{\rm TP}(g)\rangle$.

1. Drag-diffusion of particle trajectories

The drag-diffusion scattering of test particles using Monte Carlo methods is well established [6,16]. The implementation of Rognlien and Cutler provides a prescription for small-angle scattering of a three-dimensional velocity vector due to collisions with a fixed Maxwellian background. This method is readily adapted to a four-point gyroaveraged collision operator as follows. Four gyrophase angles ($N\pi/2$, $N=0,1,2,3$) are selected with which $\mathbf{x}=\mathbf{R}+\boldsymbol{\rho}$ and $\mathbf{v}=(v_x,v_y,v_z)$ are calculated from $\mathbf{v}_\perp$ and $v_\parallel$. The drag-diffusion operator described in detail in Ref. [16] is then applied to $\mathbf{v}$ four times per time step at the four gyroangles using a time step of $\Delta t/4$ in the drag-diffusion coefficients. From the scattered velocity vector, scattered values of $\mathbf{v}_\perp$ and $v_\parallel$ are reconstructed and the gyrocenter position must be modified to reflect the scattering of $\mathbf{v}_\perp$, viz., $\Delta\mathbf{R}=\Delta\mathbf{v}\times\hat{\mathbf{b}}/\Omega$. The drag-diffusion computation is a serial operation on the particle list and has been vectorized easily. The error function and its derivative used repeatedly in the collision operator are evaluated using a standard rational approximation, tabulated in arrays, and linearly interpolated when needed.

2. Polarization contribution to collision operator

The first term on the right-hand side of Eq. (23) is associated with the polarization contribution to the collision operator. Its evaluation based on Eq. (15) and the equations that immediately follow is straightforward. The increment to the particle weight $\Delta w_i = F_M^{-1}\langle C_{\mathrm{TP}}(F_M q\bar{\phi}/T)\rangle\Delta t$ is conveniently evaluated when the particle trajectories are advanced using the approximations

$$\frac{\partial\bar{\phi}}{\partial v_\perp}\approx -\tfrac{1}{4}[E_x(\mathbf{x}+\rho\hat{\mathbf{x}})-E_x(\mathbf{x}-\rho\hat{\mathbf{x}}) + E_y(\mathbf{x}+\rho\hat{\mathbf{y}})-E_y(\mathbf{x}-\rho\hat{\mathbf{y}})] \tag{24}$$

and

$$\nabla_R^2\bar{\phi}\approx\tfrac{1}{4}[\nabla_{R\perp}^2\phi(\mathbf{x}+\rho\hat{\mathbf{x}})+\nabla_{R\perp}^2\phi(\mathbf{x}-\rho\hat{\mathbf{x}}) + \nabla_{R\perp}^2\phi(\mathbf{x}+\rho\hat{\mathbf{y}})+\nabla_{R\perp}^2\phi(\mathbf{x}-\rho\hat{\mathbf{y}})] \ . \tag{25}$$

The Laplacian of the potential in Eq. (25) is calculated on the spatial grid, stored when Poisson's equation is solved, and then interpolated to the particle positions at the four gyrophase positions when the force assignment is computed. The electric field components in Eq. (24) also are calculated and stored when the gyroaveraged force assignment is computed. The computation of the contribution to Eq. (23) from Eq. (15) is a serial computation over the particle list and has been vectorized easily.

3. $\langle p(\delta f)\rangle$ contribution to collision operator

The expression for $\langle p(\delta f)\rangle$ and the discussion following Eq. (16) provide a prescription for calculating the remaining contribution to the right-hand side of Eq. (23). These source/sink terms involve the evaluation of an integral over the perturbed part of the distribution function in Eqs. (11) and (16). The perpendicular velocities $\mathbf{v}_\perp$ needed in Eq. (16c) are calculated from $v_\perp$ and the four gyrophase angles at the positions $\mathbf{R}+\boldsymbol{\rho}$. The error function and its derivative required in Eq. (16) are functions of the particle energy and are evaluated by linear interpolation on precomputed values stored in an array. The moments $p_\parallel$, $\mathbf{p}_\perp$, and λ are computed like the charge density, i.e., as sums over the particles, averaged over the four gyrophase-angle positions, and linearly interpolated onto the spatial grid. We have taken care to make the moment calculations efficient by vectorizing them using the method of Heron and Adam [18]. This is the most computationally intensive component of the collision algorithm. The calculation of the $\langle p(\delta f)\rangle$ contribution to the collisions is completed by evaluating the right-hand side of Eq. (16a) by linearly interpolating the moments back to the four gyrophase-angle positions and averaging. This last step is best done when the gyroaveraged forces are calculated. We note that in addition to the increased number of computations involved with the depositions of four more moments of the ion distribution function, there is the increased memory requirement of the additional four grid arrays for the moments and several computer vector-length arrays for the Heron and Adam vectorization.

B. Collision algorithm II

As described in Sec. IV, the second algorithm uses the same marker-particle drag-diffusion collision operator as the first algorithm. The order of the four gyroangles is randomly scrambled. While collisionally deflecting the marker-particle trajectories in the second algorithm, the momentum deflections and energy changes are used to collect the four additional moments of the ion distribution function needed for the $\langle p(\delta f)\rangle$ calculation. The change in the gyrocenter weights to account for the change due to $\langle C_{\mathrm{TP}}(F_M q\bar{\phi}/T)\rangle$ are calculated directly from the changes in $\bar{\phi}$ due to the collisional changes in the marker particles. It has been found to be preferable to use the velocities before the collision instead of those after the collision in the calculation of $\Delta w_i^{\mathrm{cons}}$ in Eq. (17). This introduces order $\sqrt{\nu_0^{ii}\Delta t/N_s}$ relative errors into the source term, where Δt is the time step and N_s is the number of particles in the structure under consideration. These average to zero over many time steps and are not problematic. The use of the precollision velocities avoids secular second-order correlations that arise if the velocities after the collisions are used in the conserving sources of Eqs. (8). These correlations result in a secular change in the energy of order $\nu_0^{ii}\Delta t/N_{pg}$ per collision operation, where N_{pg} is the number of particles per grid cell, and can be problematic over many time steps.

The costliest part of the collision calculation on a vector computer is the accumulation of the four additional moments of the ion distribution function determined from the collisional momentum and energy changes. This is vectorized following the Heron and Adam method [18]. To improve efficiency, on time steps when there are no collisions only the charge density is collected using the minimum amount of grid arrays, index arrays, and other auxiliary arrays required by the Heron-Adam method.

C. Test cases and code timing

It is important to be able to diagnose any numerical implementation of the above collision operator in order to verify that the collision rates are correct and that the particle number, energy, and momentum are indeed conserved. Several tests are possible.

The collision rates are inherent to the test-particle portion of the operator. The implementation of this portion by Rognlien and Cutler [16] has previously been tested and shown to give the correct rates of relaxation of the parallel momentum, the energy, and the correct diffusion coefficient for the gyrocenters. In order to verify that this operator has been imported correctly, tests of the relaxation of temperature anisotropy and of the perpendicular diffusion of the marker particles are presented in Sec. VII C 1.

The source terms in Eqs. (14) and (16) are constructed so as to conserve the first-order (δf) particle number, momentum, and energy statistically on a spatial scale of order the grid cell size. The conservation becomes exact only in the limit where (i) the grid cell size is much smaller than the spatial scale of the region in which the conservation is being measured, (ii) there are infinitely many particles in this region, (iii) the particles are distributed in velocity space so that they accurately sample the Maxwellian, and (iv) the test-particle operator is sufficiently accurate that it drives the simulation particle distribution towards a Maxwellian with the correct specified temperature $T_i \equiv m_i v_T^2$. In the case of collision algorithm I, many time steps are also required. Both direct non-self-consistent and indirect self-consistent tests have been carried out and are described below.

1. Tests of the drag and diffusion of the marker particles

To test the drag diffusion of the particle trajectories in our collision operator, we simulated the collisional relaxation of a weak temperature anisotropy with the self-consistent electric field suppressed. Temperature relaxation of a weak anisotropy obeys the equation

$$\Delta T(t) = [T_{i\perp}(0) - T_{i\parallel}(0)]\exp(-\alpha \nu_0^{ii} t) , \tag{26}$$

where $\nu_0^{ii} = 4\pi q^4 n_i \ln\Lambda / m_i^2 v_T^3$ is the basic collision frequency, $v_T = \sqrt{T_i / m_i}$ is the ion thermal speed, and $\alpha = 0.23$ [19]. Figure 1 presents the temporal relaxation of a temperature anisotropy from a simulation. There is a significant reduction in the fluctuations with four times as many particles, and good agreement with Eq. (26) is obtained. Good accuracy is obtained in this and the other test cases for $\nu_0^{ii} \Delta t \leq 0.025 - 0.05$.

The classical collisional diffusion of test particles across the magnetic field is used as a second test for the drag-diffusion operator. The self-consistent electric field is again suppressed. A set of test particles is loaded initially at a common location in x. The spatial diffusion in space should be described by

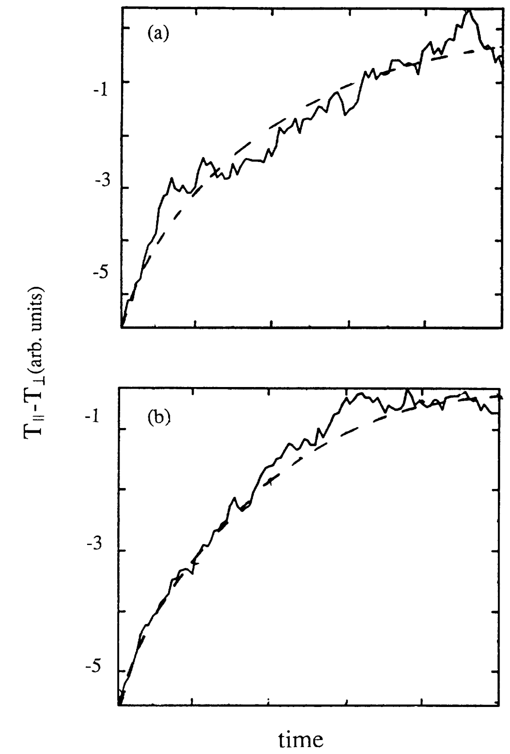

FIG. 1. Temporal relaxation of a weak temperature anisotropy due to collisions. The theoretical expectation is shown with a dashed line. Simulations with (a) 16 particles per cell and (b) 64 particles per cell.

$$\lim_{t \to \infty} \frac{\langle [x(t) - x(0)]^2 \rangle}{t} = O(1) \frac{\rho_i^2}{\tau_{ii}} , \tag{27}$$

where $\tau_{ii} = \nu_0^{ii} / 3\pi^{1/2}$. A simulation result for this is shown in Fig. 2. The agreement with Eq. (27) improves as the number of collision events grows in time until the particle displacements begin to approach the system dimension.

2. Momentum and energy conservation

The simplest and most direct tests of the first-order conservation can be made as follows. A large number of

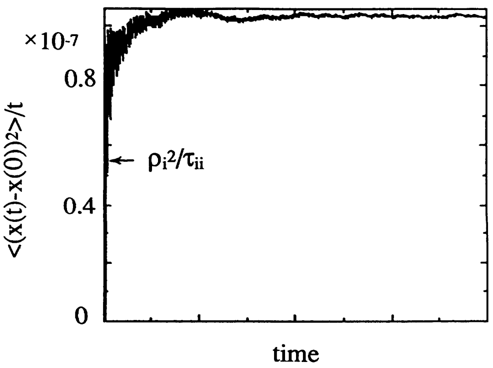

FIG. 2. Spatial diffusion coefficient, ratio of the variance in the displacement in x to the elapsed time, as a function of time.

marker particles are loaded from a uniform Maxwellian distribution. A direction in phase space is chosen, depending on which conservation law is to be tested. The weights of the particles are initialized to be equal to some constant in a narrow interval about a single value of the chosen phase-space coordinate, and zero outside this interval. This initialization is independent of the other phase-space directions. If it were not for the source-sink terms, the zero-order drag and diffusion acting on those particles would cause the value of the first-order moment associated with the chosen phase-space direction to evolve. When the source terms are kept, however, the evolution of the moment should be small. The other first-order moments, whose zero-order counterparts do not evolve, should also not evolve. Such tests have been carried out for the energy, the parallel momentum, and the canonical perpendicular momenta, which to lowest order in the gyroradius correspond to the gyrocenter positions.

Results from these tests are shown in Figs. 3–6. Figures 3 and 4 show, respectively for algorithms I and II, time histories of the first-order energy and momentum, with (solid curves) and without (dashed curves) the conserving source terms. The parameters used were $N_p=4096$ particles, grid size $\Delta_x=\Delta_y=\rho$, number of grid cells $N_x\times N_y=64\times 64$, $v_0^{ii}\Delta t=0.011$, where Δt is the time step. Collisions were done every time step. The particle weights were initially set to zero except for those with parallel velocities $1.4v_T<v_\parallel<1.6v_T$, which have particle weights set initially to 0.01. The electrostatic potential was set to zero for these cases. The energy and parallel momentum are both initially larger than what they would be if those particles with nonzero weights and those with zero weights both had the same isotropic Maxwellian distribution as the marker particles. The test-particle collisions (i.e., collision operator without the source terms) therefore cause the energy and parallel momentum to decrease, as shown by the dashed curves.

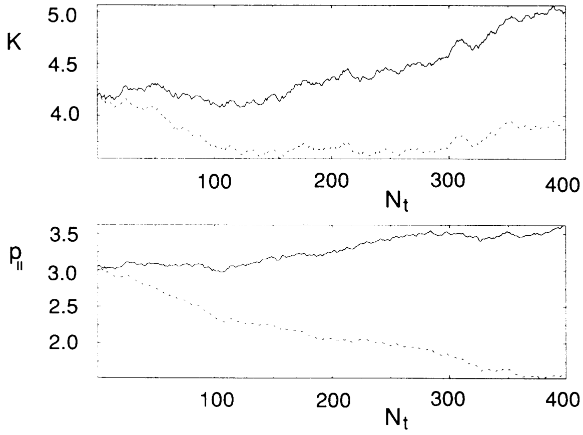

FIG. 3. Time histories of the first-order energy and momentum in arbitrary units, with (solid curves) and without (dashed curves) the conserving source terms for algorithm I. The parameters and initialization are described in the text. The horizontal axis is the number of time steps. The total time is $v_0^{ii}t=4.35$.

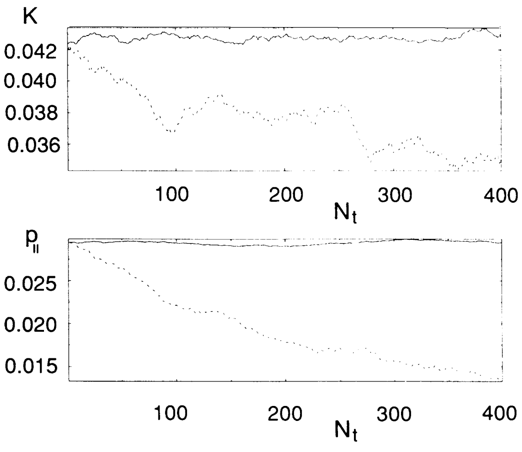

FIG. 4. Same as Fig. 4, but for algorithm II.

The differences in the dashed curves between Figs. 3 and 4 are due to the fact that the order of application of the test-particle collisions with respect to the gyrophase is different in the two cases. With the source terms kept, both quantities should be conserved. By comparing Figs. 3 and 4 it is seen that algorithm II results in greatly improved energy and momentum conservation. In particular, algorithm II significantly reduces energy and momentum fluctuations at all time scales, including those shorter than the collision time. This property is particularly advantageous for resolving waves or structures with short spatial scales which, because of poorer particle statistics, would have large energy and momentum fluctuations if algorithm I were used.

The test for perpendicular momentum conservation in a gyrokinetic code is that to first order in the gyroradius, there should be no diffusion of the first-order gyrocenter density. This can be seen by expanding the gyrocenter density to first order in the gyroradius,

$$\begin{aligned} n_{\rm gc}(\mathbf{x}) &\equiv \int d\mu\, dv_\parallel g(\mathbf{x},\mu,v_\parallel) \\ &\simeq \int d\mu\, dv_\parallel [g(\mathbf{x}-\boldsymbol{\rho},\mu,v_\parallel)+\boldsymbol{\rho}\cdot\boldsymbol{\nabla}_\perp g(\mathbf{x}-\rho,\mu,v_\parallel)] \\ &= n(\mathbf{x})+\boldsymbol{\nabla}_\perp\cdot\Omega^{-1}\hat{\mathbf{b}}\times\int d\mu\, dv_\parallel \mathbf{v}_\perp g(\mathbf{x}-\boldsymbol{\rho},\mu,v_\parallel)\ . \end{aligned} \tag{28}$$

For this test, we take $\phi=0$. The integral in the last form in Eq. (28) is the perpendicular momentum field. Thus if the density and momentum are conserved locally by the collision operator, then to first order in $\boldsymbol{\rho}$, so is $n_{\rm gc}(\mathbf{x})$. To test this conservation law, the particles are loaded with initial weights that have a dependence on x that has a (Gaussian) bell shape with adjustable width. The evolution of $n_{\rm gc}$ is then compared with and without the conserving source terms. The results are shown in Fig. 5. Each plot shows 41 curves of ion gyrocenter density at different times, separated by $v_0^{ii}\Delta t=0.56$. The initial loading of the weights was of the form $w_i=\epsilon\exp[-(x_i-x_c)^2/\Delta^2]$, where x_i is the radial position of the gyrocenter, and with $\epsilon=0.01$, $x_c=8\rho_s$, and $\Delta=3\rho_s$. In Fig. 5(b), the first curve is among the highest, and the gyrocenter test-particle diffusion causes the peak

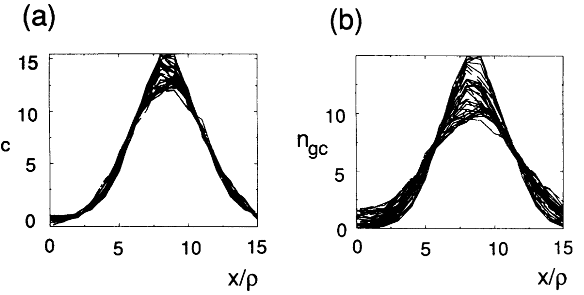

FIG. 5. Curve of ion gyrocenter density at different times, separated by $\nu_0^{ii}\Delta t=0.56$, for algorithm II. The weights are initialized with a Gaussian profile in x, and ϕ is set to zero. Other details are given in the text. The curves in (a) include the conserving source terms, while those in (b) do not. The units for the vertical scale are arbitrary, but are the same for (a) and (b).

gyrocenter density to decrease, in the absence of the conserving source terms. In Fig. 5(a), in which the source terms were kept, the gyrocenter diffusion is greatly reduced, indicating good conservation of perpendicular momentum. Because the nondiffusion result of Eq. (28) is accurate only to first order in ρ, some residual gyrocenter diffusion is expected in Fig. 5(a).

Finally, a test of the conservation of the perpendicular momentum for the polarization density terms is shown for algorithm II in Fig. 6. This test is suggested by making the replacement $g \to g+qF_M\bar{\phi}/T$ in Eq. (28). The weights are initialized to zero, but an electrostatic potential of the form $e\phi/T_e=\epsilon\exp[-(x-x_c)^2/\Delta^2]$, with $\epsilon=0.01$, $x_c=8\rho_s$, and $\Delta=3\rho_s$, is imposed for all time. The other parameters were as for Fig. 5, except that the time between successive curves and the total time have been doubled. Plotted are curves of the gyrocenter density including the $q\bar{\phi}/T$ term in the weights. The adiabatic response $-q\phi/T$ is not included, because its average does not evolve for a fixed ϕ. Again, Fig. 6(a) shows the gyrocenter density with the conserving source terms included, while Fig. 6(b) shows the result without the source terms. The effect of Δw_i^{pol} in Eq. (20) in this case is simply to maintain the sum $w_i+q\bar{\phi}/T$ constant, so that this test is almost equivalent to that of Fig. 5.

Similar results were obtained with identical parameters (including particle number) for algorithm I. It is worth remarking again, however, that the complexity of the polarization terms in algorithm I allows many more possibilities for errors in th derivation and implementation of the various terms involved than for algorithm II. A single test of the type discussed here is therefore less sufficient as a test of the correctness of the implementation of algorithm I than of algorithm II.

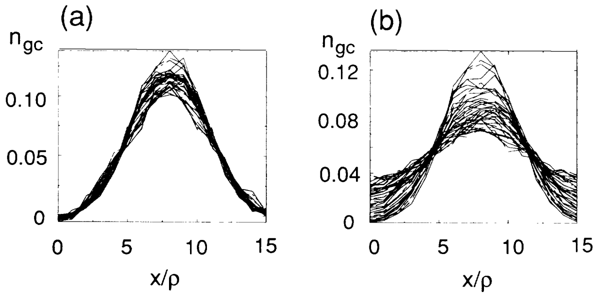

FIG. 6. Curves of ion gyrocenter density at different times, separated by $\nu_0^{ii}\Delta t=1.12$, for algorithm II. The weights are initialized to be zero. The potential is fixed with a Gaussian profile in x. Other details are given in the text. The curves in (a) include the conserving source terms, while those in (b) do not.

3. Ion-temperature-gradient instability with collisions

As a test of the collision algorithm in the presence of self-consistent fields we have simulated the slab ion-temperature-gradient (ITG) instability [20] in the long-wavelength limit and in the presence of ion-ion collisions. The instability threshold value of η_i in the limit $k_\perp\rho_i \ll 1$ is reduced by collisions from $\eta_i=2$ in the collisionless limit to a value of $\eta_i=\frac{2}{3}$ in the limit that the parallel wavelength is long compared to the ion collisional mean free path [21,22]. If η_i, the ratio of the density to temperature scale lengths, is fixed at a value close to its value at marginal stability, then the growth rates of given modes are mostly expected to increase with increasing collisionality. These trends can be viewed as being due to a lowering of the effective ratio of specific heats as the compression passes from being one dimensional in the collisionless limit to three dimensional in the collisional limit. Furthermore, the increase in growth rate is expected to depend critically on the conservation of parallel momentum. In the absence of the conservation of parallel momentum, the collisions would induce a parallel drag. This would make the ion response in the collisional limit dissipative rather than inertial and result in decreased growth rates, although the marginally stable value of η_i would not necessarily be altered by the drag. Thus the ITG mode near marginal stability provides a test of the conservation laws, especially the conservation of parallel momentum.

Hassam *et al.* [21] and Chang and Callen [22] have obtained expressions for the effect of ion-ion collisions on the threshold and the growth rates for intermediate collisionality, in addition to the values of these in the limits of weak and strong collisionality. They find that the threshold η_i decreases monotonically as a function of collision frequency. Hassam *et al.* [21] used a number-, momentum-, and energy-conserving Krook-model collison operator, while Chang and Callen used a collison operator consisting of a Lorentz operator plus a momentum-conserving source term. Their analyses can be used to obtain a dispersion relation for ITG modes in the limit of weak collisionality, i.e., long mean-free path compared to the parallel wavelength.

Chang and Callen obtain a dispersion relation for the ITG instability, Eq. (55) of their paper [22], for $k_\perp\rho_i \ll 1$ and $\omega \ll \omega_{*i}$, where $\omega_{*i}=cT_i\mathbf{b}\times\nabla\ln(n_0)\cdot\mathbf{k}_\perp/(eB)$. This dispersion relation depends on a number of coefficients which in turn depend on the plasma dispersion function evaluated as a function of the mode frequency and ion collision frequency. If we expand the plasma dispersion function to first order in its argument, valid when the complex mode frequency and collision frequency are small compared to the product of the parallel wave number and the ion thermal speed (adiabatic limit), then Eq. (55) of Chang and Callen is reduced to

$$\omega \approx i[0.732(\eta_i - 2)k_{\parallel}v_{Ti} + 0.186\nu_0^{ii}] \ . \tag{29}$$

This corrects an algebraic error in Eq. (61) of Chang and Callen [22]. Note that in the collisionless limit, a kinetic analysis would give $\omega = i0.62(\eta_i - 2)k_{\parallel}v_{Ti}$ [23].

In Fig. 7, we display solutions of the linear dispersion relation for $\eta_i = 2.5$, and data from two-dimensional gyrokinetic δf simulations of the ITG instability using algorithm II. The growth rates plotted have been normalized to the collisionless growth rate to focus on the influence of collisions. Note that the collisionless growth rate observed in simulation was $0.25k_{\parallel}v_{Ti}$, while kinetic theory gives $0.31k_{\parallel}v_{Ti}$ omitting gyroradius effects and corrections from finite differencing. The simulation results were obtained using 4160 particles, with a box size $L_x \times L_y = 32\rho_i \times 32\rho_i$, and with $\rho_i/L_n = 0.05$. In the simulations we restricted the Fourier representation to only a single spatial mode with $k_x = 0$, $k_y\rho_i = 0.2$, and $k_{\parallel}/k_y = 0.01$. Note that $k_{\parallel}v_{Ti}/\omega_{*i} = 0.2$ here, and ω/ω_{*i} has been assumed small in obtaining Eq. (29). We undertook simulations of the ITG instability with and without the inclusion of the $\langle p(\delta f)\rangle$ source terms in the collision operator. We note in Fig. 7 that the agreement with the corrected Chang-Callen theory is fairly good only when the $\langle p(\delta f)\rangle$ source terms are included. For the small value of $k_y\rho_i = 0.2$ used, the polarization effects embodied in $\langle CF_M q\phi/T\rangle$ were not observed to have any significant influence on the growth rates. For these simulations the second collision algorithm was used.

An additional test of the collision algorithm and the Chang-Callen theory was provided by simulating a system with $\eta_i = 1.66$, $\rho_i/L_n = 0.1$, and the same single-mode restriction as used in the preceding. With $\nu_0^{ii}/k_{\parallel}v_{Ti} = 10$, the value of $\eta_i = 1.66$ is predicted by the Chang-Callen theory to exceed the threshold value $\eta_i = 1.1$; and we observe a weakly growing ITG instability in simulations with a frequency whose real part is $\mathrm{Re}(\omega/k_{\parallel}v_{Ti}) \approx -0.5$. With no collisions and otherwise the same parameters, no ITG instability is observed (recall that the collisionless threshold is $\eta_i = 2$ at $k_y\rho_i = 0$). Both these results and those displayed in Fig. 7 are in general agreement with the Chang-Callen theory. We were able to obtain similar results using algorithm I, provided that the number of particles was increased by a factor of 4.

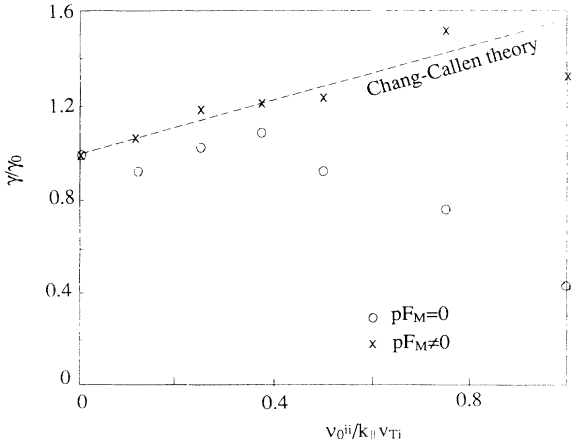

FIG. 7. Growth rates normalized to the collisionless growth rate as a function of relative collisionality $\nu_0^{ii}/k_{\parallel}v_{Ti}$ for ion temperature gradient instability with $\eta_i = 2.5$, $k_{\perp}\rho_i = 0.2$, and $k_{\parallel}/k_{\perp} = 0.01$. The dashed line is the prediction of Chang and Callen's theory as given in Eq. (29).

4. Code timings

The calculation of ion-ion collisions in a δf gyrokinetic simulation using the algorithm presented here is computationally intensive. The code has been fully vectorized for use on vector supercomputers to improve its efficiency. In the limit that particle pushing dominates the computations, the running time with collisions performed at every time step is approximately tripled. This represents a worst case. For a typical application, e.g., the simulation of drift-type instabilities of interest in tokamak research, the ion-ion collision frequency is much less than the diamagnetic drift frequency in most situations of interest. For these applications, the collision operator can be invoked less often than every time step. For example, if the ions are collided every eighth time step, the collisions add 20% to the running time of the simulation. The most expensive part of the additional calculations required for the δf collision algorithm is the accumulation of four additional particle moments. To ensure good optimization on a vector supercomputer, using a prescription for vectorizing all the accumulations such as is given by Heron and Adam [18] is very important. Furthermore, one must be careful to modify the Heron-Adam procedure to accumulate only the charge density (for electrostatic simulations) and not to manipulate those arrays associated with the additional moments on timesteps when collisions are not done.

Finally, we note that because the implementation of a number-, energy-, and momentum-conserving Krook operator, such as the one used in the analytical calculations of Hassam *et al.* [21], would require accumulation of the density, momentum, and energy moments needed for the conservation terms, the computational expense of such an operator would be comparable to that of the operators implemented here. This is despite the physics that is lost by such a Krook model due to the replacement of the test-particle drag-diffusion operator by a damping term.

ACKNOWLEDGMENTS

We are grateful to R. Cohen, T. Kaiser, T. Rognlien, A. J. Brizard, X. Xu, Z. Chang, and J. Callen for useful discussions and assistance. This work was performed for the U.S. Department of Energy under Contract No. W-7405-Eng-48 at the Lawrence Livermore National Laboratory and contributes to the Numerical Tokamak Project, which is an activity supported jointly by the U.S. Department of Energy's Office of Fusion Energy and the Office of Scientific Computing as part of the High Performance Computing and Communications Program.

[1] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).

[2] (a) M. Kotschenreuther, Bull. Am. Phys. Soc. **34**, 2107 (1988); (b) *Proceedings of the 14th International Conference on the Numerical Simulation of Plasmas* (SAIC-NRL, Annapolis, MD, 1991); (c) (private communication).

[3] R. Shanny, J. M. Dawson, and J. M. Greene, Phys. Fluids **10**, 1281 (1967).

[4] A. Boozer and G. Kuo-Petravic, Phys. Fluids **24**, 851 (1981).

[5] P. J. Catto and K. T. Tsang, Phys. Fluids **20**, 396 (1977).

[6] X. Q. Xu and M. N. Rosenbluth, Phys. Fluids B **3**, 627 (1991).

[7] A. M. Dimits and W. W. Lee, in *Proceedings of the 12th Conference on the Numerical Simulation of Plasmas* (LLNL, Livermore, CA, 1987); J. Comput. Phys. **107**, 309 (1993).

[8] T. Takizuka and H. Abe, J. Comput. Phys. **8**, 19 (1977).

[9] R. J. Procassini, C. K. Birdsall, and B. I. Cohen, Nucl. Fusion **30**, 2329 (1990).

[10] S. Ma, R. D. Sydora, and J. M. Dawson, Comput. Phys. Commun. **77**, 190 (1993).

[11] W. W. Lee, J. Comput. Phys. **72**, 243 (1987).

[12] E. A. Frieman and Liu Chen, Phys. Fluids **25**, 502 (1982).

[13] W. W. Lee, Phys. Fluids **26**, 556 (1983).

[14] S. E. Parker and W.W. Lee, Phys. Fluids B **5**, 77 (1993).

[15] See, for example, F. Reif, *Fundamentals of Statistical and Thermal Physics* (McGraw-Hill, New York, 1965).

[16] T. D. Rognlien and T. A. Cutler, Nucl. Fusion **20**, 1003 (1980).

[17] A. J. Brizard (private communication).

[18] A. Heron and J. C. Adam, J. Comput. Phys. **85**, 284 (1989).

[19] B. A. Trubnikov, *Reviews of Plasma Physics* (Consultant's Bureau, New York, 1965), Vol. 1, p. 105.

[20] L. I. Rudakov and R. Z. Sagdeev, Dokl. Akad. Nauk. SSSR **138**, 581 (1961) [Sov. Phys.—Dokl. **6**, 415 (1961)].

[21] A. B. Hassam, T. M. Antonsen, Jr., J. F. Drake, and P. N. Guzdar, Phys. Fluids B **2**, 1822 (1990).

[22] Z. Chang and J. D. Callen, Phys. Fluids B **4**, 1182 (1992).

[23] G. W. Hammett and F. W. Perkins, Phys. Rev. Lett. **64**, 3019 (1990).