Coarse-graining phase space in δf particle-in-cell simulations

Yang Chen and Scott E. Parker
University of Colorado at Boulder, Boulder, Colorado 80309, USA

(Received 2 March 2007; accepted 30 May 2007; published online 1 August 2007)

A numerical scheme for periodically coarse-graining the distribution in the phase space for δf particle-in-cell (PIC) simulation is presented. δf is periodically deposited on a five-dimensional phase-space grid, then reevaluated at the particle position using interpolation. Any discontinuity of δf in time arising from this coarse-graining procedure is reduced by resetting only a small fraction of the particle weight given by the interpolated value. An estimate of the numerical diffusion due to this smoothing procedure is provided in the limit of large particle number. Using three-dimensional toroidal ion-temperature-gradient driven turbulence as an example, the numerical scheme is demonstrated to effectively suppress the long-term increase of the particle weights, while keeping the turbulent flux unchanged. © *2007 American Institute of Physics*. [DOI: 10.1063/1.2751603]

I. INTRODUCTION

The δf method particle-in-cell (PIC) simulation is an effective tool for the study of plasma turbulence and transport.[1–3] In the δf method, the total particle distribution function is separated into a known equilibrium part, f_0, and a perturbation, δf. The particle weight is defined to be proportional to δf at the particle phase-space location. The weight is treated as an independent variable, evolved in time similar to particle space and velocity coordinates. In steady-state turbulence, the perturbation δf is usually small, $\delta f/f_0 \ll 1$. However, the evolution of the particle weights is unbounded due to the diffusive particle motion in a turbulent field. Since the discrete particle noise is proportional to the root-mean-square (rms) value of the particle weights, a growing average weight leads to increased noise. Typically, one is interested in measuring the steady-state particle and/or energy flux. It is necessary to carefully monitor the noise level to ensure that the noise-induced diffusive flux is much smaller than the turbulent transport level.[4] The algebraic growth of the weights may pose a problem for the δf PIC method, in cases in which a long steady state is needed to accurately determine the transport level.

The fact that the particle weights continue to grow algebraically in time in a saturated turbulence simulation is well-known (the so-called "the growing weight problem"), and previous studies[5,6] have shown that this is a necessary consequence of the dissipationless PIC simulation algorithm. See, in particular, Ref. 7 and references therein. In a collisionless plasma, the particle equation of motion is Hamiltonian. As a consequence, the distribution is constant along the particle trajectories. The perturbed distribution δf is therefore simply given as the difference between the distribution at the particle's initial location and the equilibrium distribution at the particle's current location,

$$\delta f = f(\mathbf{x}(0), \mathbf{v}(0), t=0) - f_0(\mathbf{x}(t), \mathbf{v}(t)). \tag{1}$$

In collisionless simulations, the weight is defined as[8]

$$w = \delta f/g, \tag{2}$$

with g the loaded particle distribution. Equation (1) then ensures that for bounded plasmas, particle weights cannot increase indefinitely. However, Eq. (1) is rarely used directly in δf simulations. The equilibrium distribution f_0 is frequently approximated by one that is not rigorously an equilibrium distribution. For instance, f_0 is often taken to be Maxwellian with density and temperature dependent on the minor radius of a tokamak. Even if a rigorous equilibrium distribution is used, approximations are frequently made in the equation of motion. Thus Eq. (1) leads to a change of the particle's weight in time even if there is no perturbation, an undesirable feature. [See Angelino *et al.*[9] for more discussions on using Eq. (1) directly.] Due to this reason, the weight equation is cast into the form of an evolution equation, which is numerically integrated along the perturbed particle trajectories. Not surprisingly, when the fluctuations are turbulent with broadband frequency and wave-number spectra, the average magnitude of weights characteristically displays secular growth. Phase-space filamentation also occurs as a result of accumulated wave-particle interaction, leading to fine scales in the distribution function that is no longer resolved by the finite number of particles. Since this process is only limited by particle collisions, the simulation of steady-state turbulence without collisions can be questioned. The fact that small but finite dissipation is needed for stationary turbulence is a physical result that manifests itself in collisionless δf PIC simulations as a secular growth of the weights.

The implementation of collisions in the δf method usually involves a random change to the particle velocity coordinates, i.e., a Monte Carlo scatter. In simulations with collisions, particle weights are frequently observed to grow faster than collisionless simulations.[10,11] This is counterintuitive, since collisions limit the process of phase-space filamentation and make a true steady-state turbulence possible. This apparent paradox is resolved by the fact that in the usual implementation of collisions, the weights can no longer be viewed as proportional to δf, instead the weights at a given phase-space location have a distribution. The relation between particle weights and δf is[12]

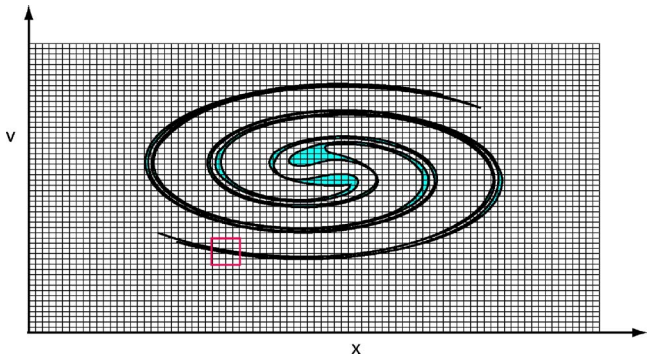

FIG. 1. (Color online) Figures 1 and 2 attempt to illustrate the method in one dimension (x,v). This figure shows a filamented phase-space structure. The red box shown in the lower left containing 16 phase-space cells is blown up and shown in Fig. 2.

$$\delta f(\mathbf{z}) = \int wF(\mathbf{z},w)dw, \tag{3}$$

where $\mathbf{z}$ denotes the usual five-dimensional (5D) phase-space location. $F(\mathbf{z},w)$ is the marker particle distribution in the extended phase space, with the particle weight as the sixth dimension. Although δf reaches a steady level and is smooth across small scales, the distribution of weights across a small scale can continue to broaden in time.[10] That is, the rms value of the weights still increases in time.

In contrast to PIC simulations, the average fluctuation amplitude of the particle distribution is seen to saturate in Eulerian simulations,[13] as a consequence of the numerical dissipation introduced by the finite-difference schemes. It is a remarkable fact that Eulerian codes and PIC codes, despite the vast algorithmic difference between them, agree so well in benchmark studies.[14,15] This fact alone suggests that for nearly collisionless turbulence, the details of dissipation (physical or numerical) might not be important in determining the physical quantities we are most interested in, e.g., particle and energy transport, a possibility that has been explored at length by Krommes and Hu.[6] Krommes later suggested a numerical procedure, called the thermostated δf method, for extrapolating the steady-state fluxes from simulations in which the quantity I (defined in Sec. II) is prescribed.[7] It should be noted that very early work by Denavit had explored full-f hybrid Vlasov-particle schemes in one-dimensional particle simulations.[16] Such a scheme uses a phase-space grid in both space and velocity that is suggestive of a conventional Vlasov simulation, while the distribution is evolved along the particle trajectories as is done in PIC simulations. Other numerical schemes along this line have been developed.[17–19] Of particular interest to the present work is that of Brunner *et al.*,[10] who developed a collisional model that solved the growing weight problem by averaging particles in a phase-space grid and modifying the equilibrium distribution accordingly.

In this paper, we present an algorithm that coarse-grains phase space by periodically resetting the particle weights. The basic idea is to deposit δf on a phase-space grid and reset δf for a given particle according to the phase-space grid value. This periodic resetting causes averaging of particle

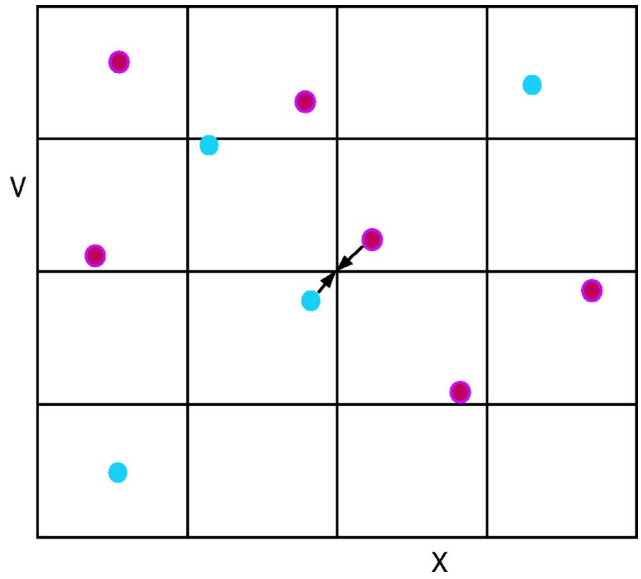

FIG. 2. (Color online) When using nearest-grid-point interpolation, the simplest situation occurs when two particles are near the same phase-space grid cell. In this situation, the interaction is simply reassigning w to the average weight of the two particles. If nearby particles have large opposite value, this averaging procedure will reduce the rms value of the particle weights.

weights for particles that come close enough together in phase space. Figures 1 and 2 attempt to illustrate the method in one dimension (x,v). Figure 1 shows a filamented phase-space structure. The red box shown in the lower left containing 16 phase-space cells is blown up and shown in Fig. 2. When using nearest-grid-point interpolation, the simplest situation occurs when two particles are near the same phase-space grid cell. In this situation, the interaction is simply reassigning w to the average weight of the two particles. If nearby particles have large opposite value, this averaging procedure will reduce the rms value of the particle weights.

Ideally, given a set of parameters ϵ_i, one for each of the five dimensions (three spatial and two in velocity), we would like to average the weights of two particles whose coordinates are within ϵ_i of each other in each dimension. In practice, this requires $\sim N \times N$ comparisons of the particle coordinates for N particles, a computationally prohibitive procedure. A more efficient approach is to use a 5D phase-space grid. The distribution function is first deposited on the phase-space grid. This grid-based representation of the distribution is then used to reset the particle weights, with the particle spatial and velocity coordinates unchanged. This grid-based approach is reminiscent of the particle-continuum method.[19] The difference is that here particle coordinates are not reset, and uniform loading in velocity is not used. In this respect, the coarse-graining scheme presented here can be viewed as a generalization of the weight spreading reduction procedure developed by Brunner *et al.*[10] for collisional plasma simulations. The coarse-graining scheme is, however, not intended specifically for reducing the collisional spreading of particle weights. It can be viewed as introducing numerical dissipation in the otherwise dissipationless δf PIC simulations. By adjusting the amount of dissipation, we are able to study the effect of small dissipation on turbulence

simulations.[6] Similar numerical study has been carried out by Candy and Waltz in the context of continuum simulations.[13] Although the same set of gyrokinetic-Maxwell equations is used in both the PIC simulation and the Eulerian simulation, dissipation arises from different sources: in Eulerian simulations of Candy and Waltz,[13] dissipation arises from the upwind finite-difference scheme, whereas in the PIC algorithm presented here it is artificially created. The simulations in this paper are carried out for the Ion-Temperature-Gradient (ITG) Driven turbulence using the Cyclone Base Case parameters.[20] The simulation results show that numerical dissipation can indeed be introduced into PIC simulations such that the continual growth of particle weights for a turbulent stationary state is suppressed, while the turbulent transport itself is not affected by the numerical dissipation.

II. WEIGHT EVOLUTION

The particle weight evolves according to[21]

$$\frac{dw}{dt} = \kappa v_{\mathrm{Ex}} + \frac{\dot{\epsilon}}{T_0}, \tag{4}$$

where $\kappa = \kappa_n - (1.5 - \epsilon/T_0)\kappa_T$, $\kappa_n = -1/n_0\, dn_0/dr$, $\kappa_T = -1/T_0\, dT_0/dr$, $\dot{\epsilon} = q(v_\parallel \mathbf{b} + \mathbf{v}_D)\cdot\langle \mathbf{E}\rangle$ is the rate of change of particle kinetic energy, and $\mathbf{b}$ is the unit vector along the equilibrium magnetic field. $\mathbf{v}_E = \langle \mathbf{E}\rangle \times \mathbf{b}/B$ is the $E\times B$ drift velocity. In this paper, we restrict ourselves to electrostatic simulations. We also note that a collisional term could be added to the weight equation, in addition to random scattering of particle trajectories.[22]

A proper measure of the magnitude of particle weights is given by the quantity

$$I = \frac{1}{2N}\sum_j w_j^2 = \frac{1}{2}\langle w^2\rangle, \tag{5}$$

where N is the number of particles in the simulation. Using Eq. (4), the rate of change of I is given by

$$\frac{dI}{dt} = \frac{1}{N}\sum_j \kappa v_{\mathrm{Ex}} w_j + \frac{1}{N}\sum_j \frac{\dot{\epsilon}_j}{T_0} w_j. \tag{6}$$

The first term can be related to the particle and energy fluxes,

$$\frac{dI}{dt} = (\kappa_n - 1.5\kappa_T)\frac{\Gamma_P}{n_0} + \frac{\kappa_T}{n_0 T_0}\Gamma_e + \frac{1}{N}\sum_j \frac{\dot{\epsilon}_j}{T_0} w_j, \tag{7}$$

with

$$\Gamma_p = \frac{n_0}{N}\sum_j w_j v_{\mathrm{Ex}}, \tag{8}$$

$$\Gamma_e = \frac{n_0}{N}\sum_j w_j \epsilon_j v_{\mathrm{Ex}}. \tag{9}$$

Assuming the third term on the right-hand side (RHS) of Eq. (7) is small, which is true in general (see Sec. IV for a numerical demonstration), Eq. (7) states that in a stationary state with nonzero transport, the average level of particle weights, measured by I, will keep increasing. This is paradoxical.[6] In the presence of collisions, the weights are no longer proportional to δf [see Eq. (3)], and a growing level of particle weights poses no paradox. Nevertheless, the growing weights lead to increasing discrete particle noise, which eventually limits the length of time that a simulation is valid. The particle-continuum algorithm[19] can be viewed as introducing numerical dissipation into the simulation so a dissipative term then appears on the RHS of Eq. (7),

$$\frac{dI}{dt} = (\kappa_n - 1.5\kappa_T)\frac{\Gamma_P}{n_0} + \frac{\kappa_T}{n_0 T_0}\Gamma_e + \frac{1}{N}\sum_j \frac{\dot{\epsilon}_j}{T_0} w_j - D, \tag{10}$$

where D is the dissipative term. This dissipative term then balances the flux term, allowing for rms value of the weights to achieve a steady state.

III. COARSE-GRAINING δf

The particle-continuum method described in Ref. 19 was simple to implement for the case of uniform loading. In this case, the phase-space volume associated with each particle is a constant, and the particle weight is simply δf at the particle location. However, we have found a uniform load impractical for 5D simulations. Assume particles are initially loaded uniformly within the cylinder $v_\perp \le v_c$, $|v_\parallel| \le v_c$, in velocity space at each spatial coordinate. The initial particle density, $\delta n(\mathbf{x}) = \int \delta f(\mathbf{x},\mathbf{v}) d\mathbf{v}$, is then computed by integrating δf over this cylinder in velocity space. However, this domain of integration will deform in time, because $v_\perp$ and $v_\parallel$ are not constants of particle motion. For instance, a passing particle initially loaded with $v_\perp = v_c$ at the outer midplane has a perpendicular velocity $v_\perp > v_c$ at the location where it passes the inner midplane, causing at this location deformation of the populated velocity domain from the initial cylinder. This deformation of the integration domain is insignificant only if v_c is sufficiently large such that the deformed domain still covers the important velocity domain, say $v \le 3v_T$ [$v_T = (T/m)^{1/2}$ being the thermal velocity] everywhere in real space. Numerically we have found that, for the Cyclone Base Case (Sec. IV) ITG problem, it is necessary to load particles uniformly in a large velocity domain, $-4v_T < v < 4v_T$, to have a converged linear growth rate. This loading puts too many particles in the high velocity region, where there are actually few physical particles. Uniform loading turns out not to be an efficient sampling of the physical distribution.

A. Coarse-graining δf using a phase-space grid

Numerically the perturbed distribution is represented as

$$\widetilde{\delta f} = \frac{V}{N}\sum_p \frac{w_p}{J}\delta(x - x_p)\delta(y - y_p)\delta(z - z_p)\delta(E - E_p) \times \delta(\lambda - \lambda_p), \tag{11}$$

where (x,y,z) are spatial coordinates. $E = mv^2/2$ and $\lambda = v_\parallel/v$ is the pitch angle variable. The Jacobian is defined through

$$dV 2\pi v_\perp dv_\perp dv_\parallel = J dx dy dz dE d\lambda. \tag{12}$$

To coarse-grain δf, we need to define a smooth function in the 5D phase space. We do this by replacing each of the δ functions with an interpolation function,

$$\widetilde{\delta f}_g = \frac{V}{N}\sum_p \frac{w_p}{J}\frac{1}{\triangle x \triangle y \triangle z \triangle E \triangle \lambda} S[(x_i - x_p)/\triangle x] \times S[(y_j - y_p)/\triangle y] S[(z_k - z_p)/\triangle z] \cdot S[(E_l - E_p)/\triangle E] S[(\lambda_m - \lambda_p)/\triangle \lambda]. \quad (13)$$

Here, $(x_i, y_j, z_k, E_l, \lambda_m)$ labels the 5D grid point. The function $S(x)$ is the particle shape function. Both linear interpolation (LI) and nearest-grid-point (NGP) can be used for $S(x)$. For linear interpolation,

$$S_{\mathrm{LI}}(x) = \begin{cases} 1 - |x| & \text{for } |x| \le 1 \\ 0 & \text{for } |x| > 1. \end{cases} \quad (14)$$

For the nearest-grid-point scheme,

$$S_{\mathrm{NGP}}(x) = \begin{cases} 1 & \text{for } |x| \le 0.5 \\ 0 & \text{for } |x| > 0.5. \end{cases} \quad (15)$$

Now $\widetilde{\delta f}_g$ is defined on the 5D grid. Using the same particle shape function, a new function $\widetilde{\delta f}'$, defined everywhere in the phase space, can be obtained from $\widetilde{\delta f}_g$. $\widetilde{\delta f}'$ is smoother than the distribution of particle weights in the 5D phase space. For instance, if two particles are located close to each other in phase space, but have opposite weights with equal magnitude, they will contribute little to $\widetilde{\delta f}'$.

To assign a new weight to a particle, we also need the marker particle density at the particle's phase-space location. The marker density $\tilde{g}$ is represented as

$$\tilde{g} = \frac{V}{N}\sum_p \frac{1}{J}\delta(x - x_p)\delta(y - y_p)\delta(z - z_p)\delta(E - E_p)\delta(\lambda - \lambda_p). \quad (16)$$

The smooth function $\tilde{g}'$ is calculated in exactly the same way as $\widetilde{\delta f}'$ is calculated. First, $\tilde{g}$ is deposited on the 5D phase-space grid, then $\tilde{g}'$ is computed at the particle position using the same shape function as that used for $\widetilde{\delta f}'$. The new particle weight is then reset to be

$$w_p' = \frac{\widetilde{\delta f}'(p)}{\tilde{g}'(p)}, \quad (17)$$

where $\widetilde{\delta f}'(p)$ is the value of $\widetilde{\delta f}'$ evaluated at the particle's 5D phase-space location.

Equation (17) ensures that w' is a weighted average of w of particles in neighboring cells. This yields some desirable properties. If particle weights are all constant, then the resetting leaves the weights unchanged. In PIC simulation, the number of particles is often chosen such that the average number of particles per phase space cell is below one. In such a situation, it frequently happens that there is only one particle in nearby cells. In such a case, Eq. (17) also leaves the weight unchanged.

In Sec. IV, the GEM code[11] will be used to demonstrate the coarse-graining technique. GEM has domain decomposition both along the magnetic field, i.e., the z direction, and in the radial direction. In this case, the coarse-graining scheme using linear interpolation requires communication of a large data size ($\sim N_x N_y N_e N_\lambda$, N_x is the number of grid points in x, etc.) between neighboring processors. On the other hand, using nearest-grid-point interpolation is equivalent to simply averaging the particle weights if more than one particle resides in one of the 5D grid cells, and does not require such data communication. In the examples presented in the following section, the NGP scheme is used. We note that if NGP is used for interpolation in all dimensions, the coarse-graining scheme presented here is identical to the scheme presented by Brunner *et al.*,[10] except a refinement we use to spread the discontinuity arising from resetting the particle weights over many time steps [see Eq. (25) below].

Despite the added computational cost, the linear interpolation scheme has the following attractive features compared with the nearest-grid-point scheme. First, linear interpolation is smoother. The above constructed $\widetilde{\delta f}'$ is a continuous function across the phase-space cell boundaries. Secondly, using the same grid sizes for coarse-graining, a particle sees many more other particles available for the weighted average. Roughly speaking, the number of particles that participate in the averaging procedure with LI is increased by 3^D-fold from that with NGP (D being the phase-space dimensionality). Given the high dimensionality of the turbulence problem, the average number of particles per phase-space cell cannot be expected to much exceed unity. In such a case, the LI scheme is much more effective.

The above formulation of the coarse-graining procedure, with a smooth function $\widetilde{\delta f}'$ constructed on the entire phase space, allows other possibilities for coarse-graining the distribution function. For instance, it might be possible to first compute the Fourier transform of $\widetilde{\delta f}'$ in one dimension (e.g., the toroidal dimension), then reconstruct the distribution using only a selected number of Fourier components. The coarse-graining scheme presented here only addresses the coarse-graining of the particle weights. The marker particle distribution in the phase space is not changed, since particle phase-space coordinates $(\mathbf{x}, \mathbf{v})$ are not changed. If the simulation is indefinitely extended, the marker particle distribution can in principle deviate from the assumed equilibrium form significantly, due to accumulated numerical error. In this case, it can be desirable to reset the particle coordinates (in a sense, to *reload* particles), in addition to resetting the particle weights, in order to provide long-term control of the marker particle distribution g. The particle-continuum[19] algorithm, in which particles are relocated during the resetting step, can be viewed as an example of coarse-graining schemes in which particle phase-space coordinates are changed.

B. Numerical dissipation due to coarse-graining

It is clear that averaging nearby particle weights introduces numerical dissipation in the simulation. The amount of dissipation can be estimated by examining the coarse-graining procedure in the infinite particle number limit. In this limit, the marker distribution is a Maxwellian distribution, uniform in space. If the grid size in velocity is small, the variation of the Maxwellian distribution across the grid can be ignored. In the one-dimensional limit, the new weight on the grid point x_i is then given by

$$w'(x_i) = \int \frac{w(x)}{\triangle x} S_{1D}\left(\frac{|x - x_i|}{\triangle x}\right) dx, \tag{18}$$

for linear interpolation,

$$S_{1D}(x) = S_{LI}(x). \tag{19}$$

Assuming the particle weights are given by

$$w(x) = \exp(ikx), \tag{20}$$

Eq. (18) can be evaluated to give

$$w'(x_i) = \exp(ikx_i)\left[1 - \frac{1}{12}(k \triangle x)^2\right]. \tag{21}$$

Assuming the weight $w(x,y,z,E,\lambda)$ is the product of this Fourier form in each dimension in the 5D space, the new weight is simply the product of the above 1D form in each dimension. If coarse-graining is performed every N_s steps, the equivalent dissipative term introduced to the kinetic equation for δf can be expressed as

$$\left(\frac{\partial \delta f}{\partial t}\right)_{Res} = \frac{(\triangle x)^2}{12 N_s \triangle t} \frac{\partial^2 \delta f}{\partial x^2} \tag{22}$$

in each dimension. To estimate the dissipation rate due to coarse-graining, a characteristic scale length of the perturbed distribution δf in each dimension is needed. In the real space this is provided by the typical wave number. For ITG modes, $k \triangle x \sim 0.5$ can be used in the plane perpendicular to the field line (the x-y plane), while $k_\parallel \sim 1/qR$ can be used for the parallel direction. For the velocity space, the frequency of coarse-graining should be chosen such that

$$\frac{(\triangle \lambda)^2}{12 N_s \triangle t} \ll \nu \tag{23}$$

if collisions are important. If the physical collision frequency is too small to be important, $\triangle\lambda$ and N_s can be chosen to yield an effective dissipation rate larger than the physical collision rate, as long as the results are insensitive to these parameters. As mentioned above, the average number of particles per 5D cell is not much above unity. In such simulations, it frequently happens that a particle is the only one in nearby cells. The particle weight is unchanged in this case. Thus no dissipation is introduced in this region of the phase space. In general, the above estimate should be viewed as the upper limit of numerical dissipation introduced by coarse-graining.

Using NGP, the corresponding result is

$$w'(x_i) = \exp(ikx_i)\left[1 - \frac{1}{24}(k \triangle x)^2\right]. \tag{24}$$

Since the phase-space grids are fixed, repeating the cell averaging does not further reduce the particle weights. Further smoothing occurs only after particles have evolved such that either the weights inside the same cell have different values, or different particles have entered the same cell.

Coarse-graining is usually done with $N_s \gg 1$. If N_s is too large, using Eq. (17) directly could introduce an apparent discontinuity in the simulation. Discontinuity can be smoothed out over time by using the following lag average formula:

$$w'' = (1 - \delta)w + \delta w', \tag{25}$$

with $\delta \ll 1$. Coarse-graining can then be done more frequently without introducing large dissipation to the simulation. In this case, the equivalent diffusion term is, with LI,

$$\left(\frac{\partial \delta f}{\partial t}\right)_{Res} = \frac{\delta(\triangle x)^2}{12 N_s \triangle t} \frac{\partial^2 \delta f}{\partial x^2}. \tag{26}$$

C. Conservation of particle number, momentum, and energy in coarse-graining

We now consider the conservation of particle number, momentum, and energy in the coarse-graining of particle weights. By construction, the coarse-graining scheme conserves particle number, momentum, and energy in the limit of a fine phase-space grid and infinite particle number. In practice, small changes are introduced. We first note that the changes in particle number, momentum, and energy are small and statistically fluctuating. We also note that any nonconserving changes to δf are proportional to the resetting parameter δ [Eq. (25)], which is typically small. If NGP is used for coarse-graining, the information flow among particles is limited to within a cell. Conservation of particle number, momentum, and energy requires the following equations to be satisfied:

$$\begin{aligned} \sum_j w_j'' &= \sum_j w_j, \\ \sum_j w_j'' v_{\parallel,j} &= \sum_j w_j v_{\parallel,j}, \\ \sum_j w_j'' v_j^2 &= \sum_j w_j v_j^2. \end{aligned} \tag{27}$$

We can simply extend the method used in Ref. 10 to account for these conservation laws. We note, however, that using Eq. (25) with NGP, the particle number is automatically conserved within the cell.

The linear interpolation scheme causes fractions of the particle weights to be distributed to nearby cells. Enforcing conservation laws then involves information across the entire phase-space dimension along which LI is used. If LI is used along all dimensions, conservation laws should be enforced only globally. However, any combination of LI and NGP can be used. For instance, one can use LI in velocity space and NGP in real space, in which case conservation laws can be enforced for each spatial cell. However, the fitting procedure of Brunner *et al.*[10] cannot be used. In general, the dependence of the weights on the velocity contains important kinetic information and cannot be fitted with a simple velocity dependence. This is a problem currently being studied within the context of developing alternative collisional algorithms in full-f PIC simulation. It has been suggested that the particles within each spatial cell be partitioned into three groups, with a constant adjustment added to all the particle

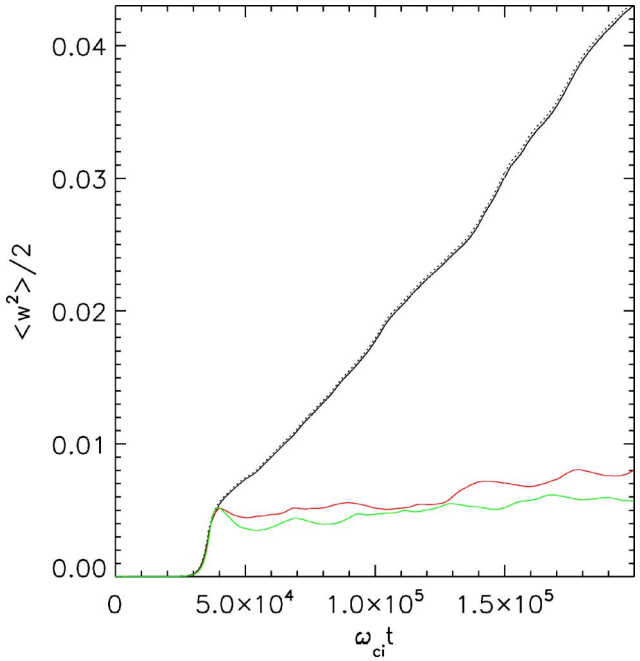

FIG. 3. (Color online) Evolution of $I=1/2\langle w^2\rangle$ for three runs with $N_s=10$. (a) Black solid: $\delta=0$. (b) Red: $\delta=0.05$. (c) Green: $\delta=0.1$. The black dashed line is the integral of the flux term in Eq. (10).

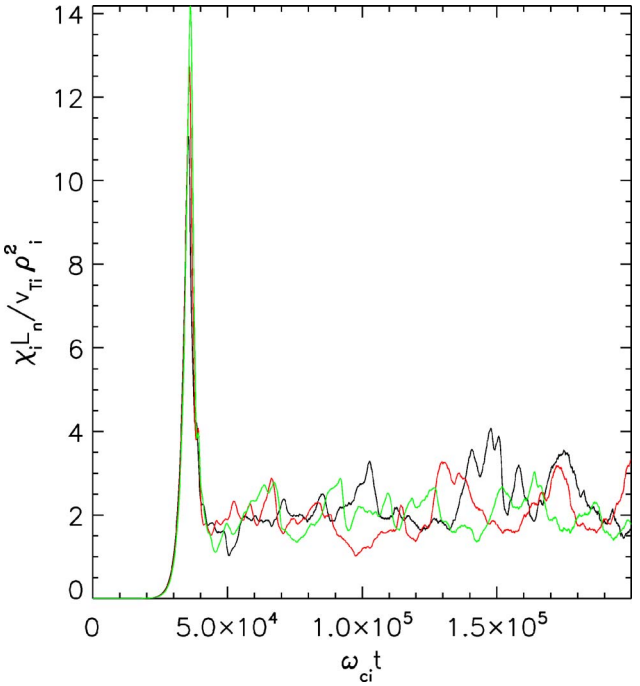

FIG. 4. (Color online) Evolution of ion heat diffusivity for the three runs of Fig. 3.

weights inside a group. The conservation laws then determine these adjustments.[23] This is a topic of current research and will not be pursued further here. We note, again, that the violation of these conservation laws fluctuates in time and decreases with increasing particle number. As long as the violation does not spuriously accumulate in the long term, the simulation is valid. In the simulations below, we do not attempt to enforce conservation laws using Eq. (27). We note that typically one neglects the parallel nonlinearity in ITG turbulence simulations, so energy is not rigorously conserved.[13]

IV. DEMONSTRATION USING THE CYCLONE BASE CASE

We now demonstrate the weight coarse-graining scheme for ITG turbulent transport using the Cyclone Base Case[24,20] as an example. The Cyclone base case is a circular H-mode type plasma and has become a standard parameter set for benchmarking purposes. Electrons are assumed to be adiabatic. The simulation is done in conventional flux-tube geometry. Time is measured in the unit of ω_{ci}^{-1} and energy in the unit of T_i, the ion temperature at the center of the simulation box. For details of the simulation code, we refer the reader to Ref. 21. The Cyclone Base Case has the following parameters: $R/L_{Ti}=6.9$, $R/L_n=2.2$, $q_0=1.4$, $\hat{s}=r_0q'/q_0=0.78$, and $r_0/R=0.18$. Simulations are done for a flux-tube size of $l_x=64\rho_i$, $l_y=64\rho_i$, and $l_z=2\pi q_0R_0=8796\rho_i$. The number of grid cells in each dimension is $N_x=64$, $N_y=64$, and $N_z=32$. This spatial grid is also used for coarse-graining the particle weights. For resetting the weights, the velocity space $(0,E_{\max})\times(-1,1)$, in the direction of energy E and pitch variable λ, respectively, is divided into $(N_E\times N_\lambda)$ grids, so that $\triangle E=E_{\max}/N_E$ and $\triangle\lambda=2/N_\lambda$.

Figure 3 shows the evolution of the quantity I in time for three different simulations: (a) ion weights are not coarse-grained; (b) weights are coarse-grained with $N_s=10$, $\delta=0.05$; and (c) weights are coarse-grained with $N_s=10$, $\delta=0.1$. The evolution of ion heat diffusivity for the three cases is shown in Fig. 4. For these simulations, $E_{\max}=10T_i$, $N_E=20$, and $N_\lambda=24$. The number of particles per spatial cell is $N_p=128$. Since $N_p\ll N_E N_\lambda$, resetting is done to individual particles "rarely" in an average sense, yet the rms value of the particle weights saturates and is very effectively reduced at late times. The ion heat diffusivity χ_i, averaged over the time window $0.5\times10^5<\omega_{ci}t<2\times10^5$, is 2.28, 2.03, and 2.04 for cases (a), (b), and (c), respectively. The variation in χ is very small, and completely within the statistical error given the fluctuation in time of the ion heat diffusivity.

Let us examine the case without coarse-graining. The quantity I increases linearly with time throughout the saturated state, as can be seen from Fig. 3, case (a). This is anticipated based on Eq. (10), given that a nonzero steady-state transport is observed and no coarse-graining is done [$D=0$ in Eq. (10)]. It is apparent that as simulation proceeds, eventually the discrete particle noise effect will become important, invalidating the simulation.[4] Also shown in Fig. 3 (in dashed line) is the time integral of the heat flux terms on the RHS of Eq. (10) (the Γ_p term is zero due to the assumed adiabatic electron response). The good agreement between the two curves proves the well-known result that the third term on the RHS of Eq. (10) (describing Landau damping) can be neglected in ITG turbulence.[5,6]

With coarse-graining, the quantity I saturates at a value dependent on the coarse-graining parameter δ. Larger δ leads to lower I. However, the saturated I is not inversely proportional to δ, as can be seen from Fig. 3. This can be explained

as follows. The quantity I consists of contributions from both large-scale and small-scale components in the distribution. Thus we can write, schematically,

$$I = I_L + I_S. \tag{28}$$

The large-scale contribution I_L can be viewed as little affected by the coarse-graining procedure. The small-scale contribution I_S, on the other hand, depends strongly on the coarse-graining parameters. Hence the total I changes in disproportion to δ.

In the presence of a physical collision operator, the quantity I will also saturate if the collision operator is implemented as an additional term in the weight equation. One might ask, "What is the effective collisionality introduced from the coarse-graining scheme?" Assuming that the coarse-graining algorithm can be cast in the same form as the collision operator, one can then estimate an equivalent physical collision frequency that would lead to the same saturated value of I. Let us illustrate this with the following collision operator:

$$C(\delta f) = -\frac{\partial}{\partial v}\left(\nu v + D_v \frac{\partial}{\partial v}\delta f\right). \tag{29}$$

The rate of change of I due to this collision operator is then[7]

$$\left(\frac{dI}{dt}\right)_c = -D_v\left\langle\left(\frac{\partial w}{\partial v}\right)^2\right\rangle \approx -\nu v_{\mathrm{Ti}}^2 \frac{\langle w^2\rangle}{(\triangle v)^2}, \tag{30}$$

where $D_v = v_{\mathrm{Ti}}^2 \nu$ and a characteristic velocity scale length $\triangle v$ is used to estimate the derivative. The equivalent collision frequency is obtained by setting this change in I equal to the flux term in Eq. (10),

$$\nu = \kappa_T \frac{\Gamma_e}{n_0 T_0}\left(\frac{\triangle v}{v_{\mathrm{Ti}}}\right)^2 \frac{1}{\langle w^2\rangle}. \tag{31}$$

We now approximate the velocity scale Δv by the velocity grid scale used in the simulation because the coarse-graining scheme removes scales shorter than this. The velocity grid size corresponding to the energy grid size at the thermal velocity is $\triangle v/v_{\mathrm{Ti}} = \triangle E/T_0 = 0.5$, and, for the run with $\delta = 0.1$ (the green line in Fig. 3), use $\Gamma_e \approx 3\times 10^{-5} n_0 v_{\mathrm{Ti}} T_0$, $\langle w^2\rangle = 0.012$, we obtain from Eq. (31) that $\nu \approx 0.0006 v_{\mathrm{Ti}}/L_T$. This estimated collisionality is very weak, consistent with the fact that the energy flux changes very little with or without coarse-graining. In a stationary state, D equals the flux term in Eq. (10), and if less dissipation is used, the rms value of the weights will increase. Assuming the velocity space grid is unchanged, and one reduces the frequency of coarse-graining, the weights will increase. In Eq. (30), this corresponds to the RHS being constant, decreasing ν and increasing $\langle w^2\rangle$.

To further examine the effect of coarse-graining, we examine the change in the fluctuation spectrum due to coarse-graining. We first note an important difference between PIC and continuum simulations. In PIC simulations, the field quantities are defined on a fixed set of spatial grids. In solving the Poisson equation, high-wave-number components of the electric potential that are not well resolved by the spatial grids are filtered. The distribution function, on the other hand, is evolved along characteristics. Fine-scale structures in both configuration space and velocity space are continuously generated, due to kinetic processes, including Landau damping and nonlinear wave-particle interactions. On the other hand, the field quantities contain only the resolved Fourier modes. Because there are typically a low number of particles per cell used in simulations (tens to hundreds), it is difficult to observe these fine structures uncontaminated by discrete particle noise. Nevertheless, we can observe the generation of high k_y fluctuations in the ion density. To show this, we first compute the 2D Fourier transform of the ion density,

$$\delta n(k_x, k_y, z_k) = \sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1} \delta n_{i,j,k} \exp(-\imath k_x x_i - \imath k_y y_j), \tag{32}$$

where $\imath = \sqrt{-1}$, and $\delta n_{i,j,k}$ is the perturbed ion density at the grid (i,j,k). Then we define

$$\rho^2(k_y, z_k) = \frac{1}{N_x}\sum_{k_x} |\delta n(k_x, k_y, z_k)|^2. \tag{33}$$

Figure 5 shows the evolution of $\rho(k_y, z_k)$ at $z_k = l_z/2$, the outer midplane, for $k_y\rho_i = 0.196$ (a), 0.294 (b), 1.08 (c), and 1.18 (d), respectively. This run is identical to the case without coarse-graining in Fig. 3, except that the number of particles

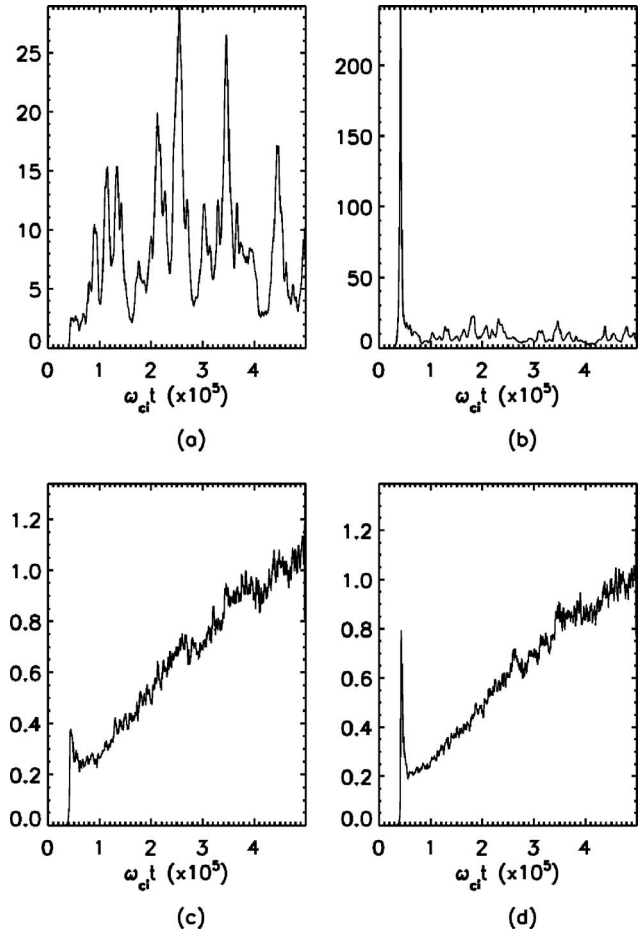

FIG. 5. Evolution of ion spectral density for four Fourier components. (a) $k_y\rho_i = 0.196$, (b) $k_y\rho_i = 0.294$, (c) $k_y\rho_i = 1.08$, and (d) $k_y\rho_i = 1.18$. Without coarse-graining.

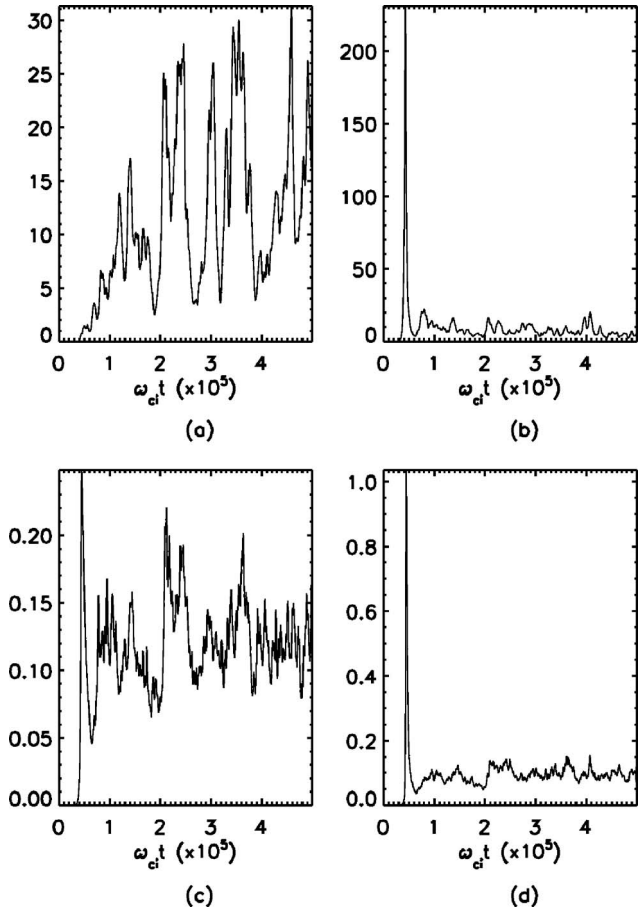

FIG. 6. Similar to Fig. 5, but with particle weights coarse-graining.

is doubled, the simulation is run much longer, and Fourier components of the electric potential with $k_x\rho_i > 1$ or $k_y\rho_i > 1$ are filtered using a hyper Gaussian filter.[25] Results of another run identical to that of Fig. 5, except that weights are coarse-grained with $N_s = 10$ and $\delta = 0.05$, are shown in Fig. 6. It is clear from this comparison that the low-wave-number components of the turbulent ion density [(a) and (b) of Figs. 5 and 6] are little affected by coarse-graining. These two components are the dominant modes in the fluctuation spectrum, and are responsible for most of the steady-state transport. On the other hand, the high wave components of (c) and (d) in Figs. 5 and 6 differ from each other. Without coarse-graining, these components keep increasing throughout the simulation, whereas in the case with coarse-graining they saturate at relatively low levels. We note, however, that even though the high k_y components of Fig. 5 increase secularly, the amplitude of these modes at the end of the simulation is still low compared with the dominant low k_y components. This is indeed why the ion heat flux (not shown for the run of Fig. 5, but see the case with no coarse-graining in Fig. 4) remains at about the same level toward the end of the simulation, although the rms value of the weights increases about eight times from $\omega_{ci}t = 5 \times 10^4$ to $\omega_{ci}t = 5 \times 10^5$.

V. DISCUSSION

We have presented a coarse-graining procedure for suppressing the secular growth of the particle weights in δf PIC simulations. We periodically coarse-grain by depositing particle weights on a phase-space grid and resetting δf by interpolating from the phase-space grid. This has the effect of averaging nearby (in phase-space) particle weights. We demonstrate with "Cyclone Base Case" parameters that nearest-grid-point coarse-graining works quite well. Ideally, as mentioned in the Introduction, one would like to be able to specify a set of closeness parameters in phase space, for instance δ_E and δ_λ in velocity space and $\delta\mathbf{x}$ in configuration space, and average the particle weights whenever the particles' energy, pitch angle, and position coordinates are detected to be within δ_E, δ_λ, and $\delta\mathbf{x}$ from each other. The problem with this idea is that detecting nearby particles is too computationally demanding. The coarse-graining scheme presented here is facilitated by a 5D phase-space grid that ensures $O(N)$ scaling with particle number. We demonstrated the efficacy of this scheme for ITG turbulence, and we are already using the coarse-graining method in more complicated kinetic electron turbulence simulations.

The scheme presented here is in many respects a generalization of the weight spreading reduction scheme developed by Brunner *et al.*[10] A comment on the coarse-graining scheme in relation to Krommes' thermostated δf (Ref. 7) is in order. Both schemes aim at introducing small numerical dissipation into the simulation to control the long-term weight growth. The thermostated δf scheme is easier to implement. It amounts to adding a Krook-like term, $-\nu w$, in the weight evolution equation. (There are important differences between the thermostated δf scheme and a simple Krook collision model, see Krommes.[7]) The coarse-graining scheme, by contrast, introduces an approximately equivalent diffusion term in the kinetic equation. Sorting particles into the 5D phase-space cells and computing the weighted averaging using linear interpolation (Sec. III) require substantially more computing time and memory. On the positive side, the weight resetting scheme is most effective for removing small-scale fluctuations in the distribution function. Since small-scale structures are more subject to particle noise, a scheme that naturally targets those structures is preferred to a scheme that treats distribution structures at different scales on equal footing. This is particularly important when physical collisions are important. As mentioned in the Introduction, the Monte Carlo collisional algorithm has the undesirable feature of broadening the particle weight distribution in a given phase-space cell, even though the distribution δf reaches steady state. Depending on the size of the grids used, the coarse-graining scheme will replace this broad distribution of weights with a narrower one. In the limit of zero grid size for coarse-graining, the original weight distribution is replaced by a δ function, with the physical distribution δf unaffected at all. As mentioned in Sec. III, in the presence of collisions one needs to choose the coarse-graining parameters, N_s and δ, such that the numerical dissipation is smaller than the physical collisional effect. Since collision frequency depends strongly on the particle velocity, one can choose a velocity-dependent $N(v)$ and $\delta(v)$ and nonuniform velocity space grids to ensure that coarse-graining is done more frequently in phase-space regions where collisional spreading of the weights is most severe.

The simulation presented in Sec. IV shows that for ITG turbulence with adiabatic electrons, the simulation without coarse-graining can proceed to very long time without any apparent effect due to particle noise. A way to control the weight growth becomes urgent only recently, when electrons are treated fully kinetically, e.g., in the simulation of Electron-Temperature-Gradient (ETG) driven turbulence.[4] Our experience with kinetic simulation of electromagnetic modes [finite-β modified ITG modes, Kinetic Ballooning Modes (KBM), and microtearing modes] also shows that when magnetic perturbation becomes important, the electron weights tend to grow faster than the ion weights in the saturated state. One way to understand this is to notice that electrons move much faster along the field line than ions. In a turbulent field where the field line itself is perturbed, the fast motion naturally leads to more filamentation in the distribution function. In such cases, the coarse-graining scheme should be very useful for removing the fine structures in the distribution and for controlling the long-term particle weights growth. In this paper, we limit our goal to presenting the coarse-graining scheme and demonstrating it with a familiar example. The study of kinetic electron physics using the resetting scheme will be reported in the future.

ACKNOWLEDGMENTS

We are grateful for support from the U.S. Department of Energy, Office of Fusion Energy Sciences. This work is part of the DOE Scientific Discovery through Advanced Computing, the Center for Gyrokinetic Particle Simulation, and the Center for Plasma Edge Simulation.

[1] M. Kotschenreuther, Bull. Am. Phys. Soc. **34**, 2107 (1988).
[2] A. Dimits and W. Lee, J. Comput. Phys. **107**, 309 (1993).
[3] S. Parker, W. Lee, and R. Santoro, Phys. Rev. Lett. **71**, 2042 (1993).
[4] W. M. Nevins, G. W. Hammett, A. M. Dimits, W. Dorland, and D. E. Shumaker, Phys. Plasmas **12**, 122305 (2005).
[5] W. W. Lee, Phys. Fluids **26**, 556 (1983).
[6] J. Krommes and G. Hu, Phys. Plasmas **1**, 3211 (1994).
[7] J. A. Krommes, Phys. Plasmas **6**, 1477 (1999).
[8] S. Parker and W. W. Lee, Phys. Fluids B **5**, 77 (1993).
[9] P. Angelino, A. Bottino, R. Hatzky, S. Jolliet, O. Sauter, T. M. Tran, and L. Villard, Phys. Plasmas **13**, 052304 (2006).
[10] S. Brunner, E. Valeo, and J. A. Krommes, Phys. Plasmas **6**, 4504 (1999).
[11] Y. Chen and S. E. Parker, J. Comput. Phys. **220**, 839 (2007).
[12] Y. Chen and R. B. White, Phys. Plasmas **10**, 3591 (1997).
[13] J. Candy and R. E. Waltz, Phys. Plasmas **13**, 032310 (2006).
[14] W. M. Nevins, J. Candy, S. Cowley, T. Dannert, A. Dimits, W. Dorland, C. Estrada-Mila, G. W. Hammett, F. Jenko, M. J. Pueschel, and D. E. Shumaker, Phys. Plasmas **13**, 122306 (2006).
[15] J. Candy and W. Dorland (private communication). The benchmark exercise was done in 2003 for the Cyclone Base Case with kinetic electrons, zero β.
[16] J. Denavit, J. Comput. Phys. **9**, 75 (1972).
[17] D. Nunn, J. Comput. Phys. **108**, 180 (1993).
[18] O. Batishchev, Bull. Am. Phys. Soc. **46**, 196 (2001).
[19] S. Vadlamani, S. E. Parker, Y. Chen, and C. Kim, Comput. Phys. Commun. **164**, 209 (2004).
[20] A. Dimits, G. Bateman, M. Beer *et al.*, Phys. Plasmas **7**, 969 (2000).
[21] Y. Chen and S. E. Parker, J. Comput. Phys. **189**, 463 (2003).
[22] A. M. Dimits and B. Cohen, Phys. Rev. E **49**, 709 (1994).
[23] C. S. Chang (private communication).
[24] S. Parker, C. Kim, and Y. Chen, Phys. Plasmas **6**, 1709 (1999).
[25] C. C. Kim and S. E. Parker, J. Comput. Phys. **161**, 589 (2000).