Journal of Computational Physics 315 (2016) 467–475

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A new hybrid-Lagrangian numerical scheme for gyrokinetic simulation of tokamak edge plasma

S. Ku [a,*], R. Hager [a], C.S. Chang [a], J.M. Kwon [b], S.E. Parker [c]

[a] Princeton Plasma Physics Laboratory, Princeton University, Princeton, NJ 08543, USA
[b] National Fusion Research Institute, Republic of Korea
[c] University of Colorado Boulder, USA

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 24 March 2016
Accepted 29 March 2016
Available online 1 April 2016

Keywords:
Plasma
Tokamak
Gyrokinetic
Lagrangian
delta-f
XGC

ABSTRACT

In order to enable kinetic simulation of non-thermal edge plasmas at a reduced computational cost, a new hybrid-Lagrangian δf scheme has been developed that utilizes the phase space grid in addition to the usual marker particles, taking advantage of the computational strengths from both sides. The new scheme splits the particle distribution function of a kinetic equation into two parts. Marker particles contain the fast space-time varying, δf, part of the distribution function and the coarse-grained phase-space grid contains the slow space-time varying part. The coarse-grained phase-space grid reduces the memory-requirement and the computing cost, while the marker particles provide scalable computing ability for the fine-grained physics. Weights of the marker particles are determined by a direct weight evolution equation instead of the differential form weight evolution equations that the conventional delta-f schemes use. The particle weight can be slowly transferred to the phase space grid, thereby reducing the growth of the particle weights. The non-Lagrangian part of the kinetic equation – e.g., collision operation, ionization, charge exchange, heat-source, radiative cooling, and others – can be operated directly on the phase space grid. Deviation of the particle distribution function on the velocity grid from a Maxwellian distribution function – driven by ionization, charge exchange and wall loss – is allowed to be arbitrarily large. The numerical scheme is implemented in the gyrokinetic particle code XGC1, which specializes in simulating the tokamak edge plasma that crosses the magnetic separatrix and is in contact with the material wall.

1. Introduction

In recent decades the δf scheme for Lagrangian particle-in-cell (PIC) simulation of the five dimensional (5D) gyrokinetic equations, of which some early works can be represented by Refs. [1–3], has rendered many successful interpretations of the turbulent plasma transport in toroidal magnetic confinement devices. An alternative method emerged later using the Eulerian phase-space grid that avoided the Monte-Carlo particle noise at the expense of the enhanced grid memory requirement [4]. The present work focuses on a further development of the Lagrangian PIC δf scheme from the "conventional" one as used by the existing gyrokinetic codes to a hybrid-Lagrangian scheme that also utilizes the Eulerian phase-space grid,

* Corresponding author.
E-mail address: sku@pppl.gov (S. Ku).

http://dx.doi.org/10.1016/j.jcp.2016.03.062
0021-9991/

in order to treat the irreversible physics phenomena on the right-hand side of the Boltzmann equation, to enable easier description of the non-Maxwellian edge plasma, and to reduce the computational cost at the same time.

When the tokamak core plasma is near Maxwellian, the conventional δf scheme reduces the statistical noise from the finite number of marker particles and allows computationally efficient turbulence simulations, since majority of the plasma is described by an analytic Maxwellian distribution function. Unlike the core plasma, however, the tokamak edge plasma can be significantly non-Maxwellian because it has a steep gradient whose scale length is similar to the banana orbit width, is in-contact with the material wall, and has a strong fluctuation level ($\sim$50%). Also, tokamak edge plasmas have strong sources and sinks from wall loss, neutral ionization and charge-exchange, radiative cooling and others, which are strong drivers of non-Maxwellian plasma. Non-Maxwellian distributions and strong sources and sinks are difficult to be described by the conventional δf scheme since it prefers small perturbation of the particle distribution function from a known Maxwellian distribution function. The original total-f (also called full-f) method can include these effects. The total-f method yields the most straightforward particle-in-cell simulation in solving kinetic equations without changing the weights from particle motions. It was successfully used to simulate the tokamak pedestal plasmas with adiabatic electron response [5]. However, the total-f method has less freedom than the δf method in using various numerical techniques such as the split weight scheme [6] and fluid-kinetic hybrid schemes [7,8]. More seriously, it is difficult to use the brute force total-f method in the edge scrape-off layer since the kinetic electron dynamics is an essential part of the scrape-off layer physics but requires a part of the electron distribution function to be prescribed as an adiabatic response in order to avoid the so called "Omega-H" instabilities [1]. In this report, we introduce a new δf scheme, called "hybrid-Lagrangian δf" scheme, which can handle the non-Maxwellian distribution function, nonlinear collisions, and strong sources and sinks without the development of the Omega-H mode instability by utilizing the velocity space grids, and which is mathematically identical to the total-f scheme.

In the sense that the new hybrid-Lagrangian scheme uses both the marker particles and the phase space grid together, it has some similarities with the semi-Lagrangian method [9,10]. The key differences between the new hybrid-Lagrangian scheme and the semi-Lagrangian method are that (1) the semi-Lagrangian method resets the particle positions on the phase-space grid at every time step, but the new hybrid-Lagrangian scheme keeps the continuous particle motion, and that (2) the new hybrid-Lagrangian scheme stores the fast space-time varying distribution function in marker particles but the semi-Lagrangian scheme stores all the information on the phase space grid.

Another Lagrangian particle scheme that utilizes the phase space grid is the coarse-graining reset procedure [11–13] for reduction of the growing weight problem. In this procedure, the δf particles are periodically interpolated at a large time interval to the grid and the particles are reset to their original phase-space coordinates with the new δf values. The key difference between the new hybrid scheme and the coarse-graining reset procedure is that (1) the new hybrid scheme keeps the coarse-graining information on the phase-space grid while the coarse-graining reset procedure gives it back to the particles, and that (2) the new hybrid scheme evolves f_0 so that it can reduce both the mean and the variance of the δf particle weights.

In section 2, the new hybrid-Lagrangian δf scheme is presented. In section 3, we demonstrate the new hybrid-Lagrangian scheme using an ion scale turbulence in the XGC1 gyrokinetic code, without complicating the demonstration with kinetic electrons and mixed turbulence modes. Section 4 presents summary and discussions.

2. A new hybrid-Lagrangian δf scheme

2.1. The conventional δf scheme

In the conventional δf scheme that solves the isolated collisionless Vlasov equation

$$\frac{Df}{Dt} \equiv \frac{\partial f}{\partial t} + \dot{\mathbf{z}}\frac{\partial f}{\partial \mathbf{z}} = 0, \tag{1}$$

where f is the particle distribution function, $\mathbf{z}$ is the vector of phase space variables (the configuration space and velocity coordinates), and $f = f(\mathbf{z})$ is the distribution function in phase space, the division of f into two parts $f = f_0 + \delta f$ yields

$$\frac{D\delta f}{Dt} = -\frac{Df_0}{Dt}.$$

δf is then evaluated from the weight evolution equation [1–3], and f_0 can be an arbitrary pre-determined function. If f_0 is a known function and δf is a small perturbation from f_0, then the computational cost savings can be substantial.

In a realistic system, there is the Coulomb collision operator on the right-hand side and the system is described by the Boltzmann equation,

$$\frac{Df}{Dt} \equiv \frac{\partial f}{\partial t} + \dot{\mathbf{z}}\frac{\partial f}{\partial \mathbf{z}} = C(f).$$

Utilizing $f = f_0 + \delta f$, we obtain

$$\frac{D\delta f}{Dt} - C(\delta f) = -\frac{Df_0}{Dt} + C(f_0)$$

In this idealized isolated system, the equilibrium thermodynamics yields $f_0 = f_M$ with $\delta f = 0$ according to the H-theorem, where f_M is the equilibrium Maxwellian distribution function in the velocity space, with $C(f_M) = 0$. In order to describe a small deviation of the isolated system from the thermodynamic equilibrium, $f_0 = f_M$ is chosen and some $\delta f << f_M$ is allowed to develop. If for some reason f_0 is driven far away from f_M, then the physics is dominated by the large collisional process on f_0: $C(f_0) \neq 0$. The small-δf scheme, then, loses its merit. For example, when δf is chosen to describe the turbulence physics, then the collisional physics from f_0 dominates the δf turbulence physics.

In a quiescent core region of a tokamak plasma, the non-Maxwellian driver is regarded to be small. Thus, the choice of an equilibrium $f_0 = f_M$ with a small perturbative δf is justified, and used by most of the tokamak microturbulence codes describing the core region (based on a reduced five dimensional gyrokinetic equation). Representative examples of such Lagrangian codes are GEM [14], GTS [15], and GTC [2]. Since f_0 is a Maxwellian, a linearized Coulomb collision can be used, which makes the simulation much easier.

2.2. Generalization of δf scheme

In the edge region of a tokamak, unfortunately, there are multiple non-Maxwellian drivers that can put the plasma in a non-equilibrium thermodynamic state, opposing the Coulomb Maxwellian driver. They are the steep pressure gradient whose scale length is as short as the ion banana width (the non-local ion banana mixing does not allow a local thermodynamic equilibrium state), the neutral ionization and charge exchange, the particle loss to the material wall, the radiative energy loss, and others. All the simplifications used for the near-Maxwellian plasmas are not allowed any more. The Coulomb collision, which is in the velocity space, needs to be fully nonlinear while satisfying the conservation properties. f_0 may be far away from Maxwellian and evolving in time to self-organize with the sources and sinks, the Coulomb collisions, the Grad-B and curvature drifts, and the turbulence. The purpose of this paper is to report construction of a new hybrid-Lagrangian PIC scheme that utilizes the phase space grid to resolve such a non-equilibrium thermodynamics issue in the kinetic Boltzmann simulation while reducing the computational cost.

We may generalize the Boltzmann equation by adding the source/sink term S on the right-hand side:

$$\frac{Df}{Dt} \equiv \frac{\partial f}{\partial t} + \dot{\mathbf{z}}\frac{\partial f}{\partial \mathbf{z}} = C(f) + S(f), \tag{2}$$

where $C(f)$ is the Fokker–Planck Coulomb collision operator, and $S(f)$ represents the sources and sinks of density, energy and momentum. The left-hand side (LHS) of this system, the Vlasov equation, conserves phase space volume and is time reversible.

In a generalized Lagrangian δf scheme, the marker particles describe a perturbed distribution function ($\delta f = f_p$) driven by f_0 that is expressed as function of the phase space variables. Hence, the distribution function can be written as

$$f = f_0 + f_p,$$

and the generalized Boltzmann equation becomes

$$\frac{Df_p}{Dt} = -\frac{Df_0}{Dt} + S^*(f), \tag{3}$$

where $S^*(f) = C(f) + S(f)$. We call this equation a generalized total-δf Boltzmann equation since it contains the source/sink term $S(f)$ and mathematically identical to the total/full-f equation. The Coulomb collision operator C must be nonlinear in a non-Maxwellian plasma. It has been shown by Ku et al. [16] using an analytic f_0 that Eq. (3) yields the same self-organized quasi-equilibrium state as Eq. (2) does, even when the marker particle numbers are much smaller in the total-δf scheme than in the total-f scheme.

The new hybrid-Lagrangian scheme presented in the following subsections solves the total-δf Boltzmann equation, Eq. (3) by putting the right hand on the 5D continuum grid. One issue in any δf scheme is the growth of the weight in the long time simulation [17]. A method to mitigate the weight growth in f_p is discussed at the end of next subsections in the hybrid-Lagrangian scheme. In the next subsection, we construct a direct weight evolution method as a pre-requite for a successful numerical application of the new hybrid-Lagrangian scheme.

2.3. A direct weight evolution method for the δf scheme

In order to describe the effect of $S^*(f)$ on the particle distribution function f_p, we use a two-weight scheme [18–20]. For simplicity, we choose the initial f to be f_0, i.e. $f_p(t=0) = 0$. The normalization weight w_0 is defined as $w_0 g \equiv f_0(\mathbf{z}_{t=0}, t=0)$, where $\mathbf{z}_{t=0}$ is the initial positions and g is the distribution function of the marker particles. f_p is the marker particle distribution function with the weight w_1,

$$f_p = w_1 w_0 g = w_1 f_0(\mathbf{z}_{t=0}, t=0). \tag{4}$$

For the weight evolution equation, we take the total derivative of Eq. (4), and obtain the usual differential form weight evolution equation,

$$\frac{dw_1}{dt} = \frac{1}{w_0 g}\left(-\frac{Df_0}{Dt} + S^*(f)\right). \tag{5}$$

Note that g is constant along a collisionless and sourceless/sinkless Vlasov particle trajectory since the phase space volume is conserved, and so is w_0.

Time integration of Eq. (5) gives simple difference form of weight change,

$$\begin{aligned}\Delta w_1 &= -\frac{\Delta f_0}{w_0 g} + \frac{1}{w_0 g}\int S^*(f)dt \\ &\simeq -\frac{\Delta f_0}{w_0 g} + \frac{\Delta t}{w_0 g} S^*(f),\end{aligned} \tag{6}$$

where Δ represents finite difference along the particle trajectory, and the Euler method can be applied to obtain the time integration of $S^*(f)$. We call this a direct weight evolution in the difference form. Key advantage of using the difference form is that it can avoid numerical derivative of f_0. This is important especially when we use the interpolation function on the phase space grid for f_0, to be described in the next section 2.4. Another advantage is that the error from time integration of Df_0/Dt can be avoided. This may allow a longer time simulation of weight evolution. Note that if the ∇B and curvature drifts in Df_0/Dt on the right-hand side of Eq. (5) is neglected, as many conventional δf codes do, Df_0/Dt is not an exact total derivative. Hence, these codes have not been able to utilize the direct weight evolution method.

2.4. The phase space grid for f_0 and S^*

In the new hybrid-Lagrangian scheme, we describe f_0 on a phase space grid. The velocity space grid is coarse-grained. We decompose f_0 into f_a and f_g, where f_a is an analytic function. Hence,

$$f = f_0 + f_p = f_a + f_g + f_p.$$

For the main ion and electron species, we may choose f_a to be a Maxwellian. f_g represents the derivation of the background distribution function from f_a on the coarse-grained velocity space grid at each configuration space grid point. f_g can arise from ionization and charge exchange interaction with neutral particles, and from the slow time varying part of f_p. f_a can be chosen to be stationary or updated from f_g while keep the relation $\Delta f_a = -\Delta f_g$.

If we want to capture the slow time varying part of f_p, the following operation is performed at every time step:

$$f_g(t+\Delta t) = f_g(t) + \alpha f_p, \tag{7}$$

where $\alpha \ll 1$ is an arbitrary coefficient which can vary in time and phase space position. If we keep performing the above operation over many time steps, f_g can capture the slow time variation part of f_p. From Eqs. (7) and (3), Df_p/Dt has the time-decay term accordingly as

$$\frac{Df_p}{Dt} = -\frac{\alpha}{\Delta t} f_p - \frac{Df_a}{Dt} + S^*(f) = -\frac{\alpha}{\Delta t} f_p + \left[\frac{Df_p}{Dt}\right]_{\alpha=0}. \tag{8}$$

When f reaches a stationary solution without a drive from turbulence, f_p decays to zero and f_g contains the deviation of f from f_a.

The decay of w_1 can be evaluated in the following way. From Eq. (6), the change of w_1 becomes

$$\Delta w_1 = -\frac{f_g(t+\Delta t) - f_g(t)}{w_0 g} + [\Delta w_1]_{\alpha=0} \tag{9}$$

Since $[f_g(t+\Delta t) - f_g(t)]/[w_0 g]$ is originated from αf_p, according to Eq. (7), we get

$$\begin{aligned}\Delta w_1 &= -\alpha \frac{f_p}{w_0 g} + [\Delta w_1]_{\alpha=0} \\ &= -\alpha w_1 + [\Delta w_1]_{\alpha=0}\end{aligned} \tag{10}$$

This can mitigate the growing weight problem in a long time simulation.

As stated above, the continuum grid can be used for the evaluation of $S^*(f)$. One important example is the nonlinear Fokker–Planck collision on the velocity space grid [21,22]. In a tokamak edge plasma, the distribution function can be non-Maxwellian and the use of a nonlinear collision operator becomes necessary. In this scheme, we can calculate the nonlinear Fokker–Planck collision operator on the velocity space grid, and the $\partial f/\partial t$ from collisions on the grid is transferred back to particle weights. In the rare cases that a grid point does not have any corresponding particles, the collisional $\partial f/\partial t$ is kept in f_g: another advantage of the new scheme. Other physics processes such as neutral ionization, neutral charge exchange, radiation cooling, heating, and momentum injection are implemented in the same way.

3. Demonstration with Ion Temperature Gradient turbulence

To demonstrate and verify the new hybrid-Lagrangian scheme, we use the Ion Temperature Gradient (ITG) turbulence in a concentric circular geometry with the adiabatic electron response in a 5D gyrokinetic code XGC1. Pure ITG turbulence in a

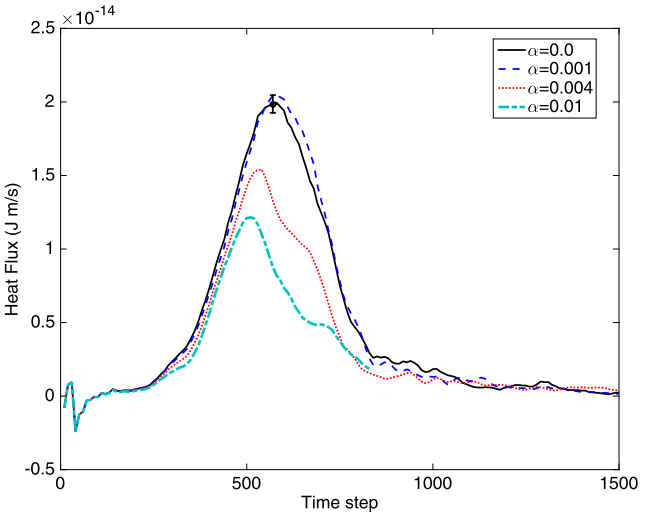

Fig. 1. Heat flux per particle from the XGC1 ITG (Ion Temperature Gradient) turbulence simulations with $\alpha = 0$, 0.001, 0.004 and 0.01. A standard $1/\sqrt{N_g}$ error bar is shown together. When $\alpha = 0.004$ and $\alpha = 0.01$, a reduction in the heat flux and the turbulence is observed.

circular geometry plasma is a well benchmarked case even between a total-f and a conventional δf case [16], and is a good target for the verification of a new scheme. On the other hand, the edge plasma for which the present scheme is developed can only be simulated by XGC1 within the gyrokinetic framework, and a wide code-to-code verification is not available. We note here, however, that the core plasma example chosen here for the verification and demonstration purpose is not far from Maxwellian, hence the gain in the computational capability and cost from the new hybrid scheme will not be as large as for the edge plasma.

3.1. XGC1

The 5D gyrokinetic code, XGC1 includes the X-point geometry and the magnetic separatrix surface for tokamak edge plasmas. It normally uses the experimental magnetic field and tokamak boundary geometry of divertor and limiter. The simulation domain can be extended to the whole plasma including the magnetic axis. To handle the magnetic axis and the X-point, XGC1 uses a cylindrical coordinate system instead of a magnetic coordinate system in the particle equation of motion. Unstructured field-following triangular mesh is used for turbulent field solver. Kinetic equations for the main ions, impurities, and electrons are solved. Electrostatic ITG turbulence is used for this study, but XGC1 normally solves for wider range of turbulence involving the kinetic electron dynamics. A fully nonlinear Fokker–Plank–Landau collision operator is calculated on the 2D velocity space grid.

XGC1 is designed for high performance computers (HPC), and utilizes MPI, OpenMP, GPU computing with OpenACC and CUDA, and vectorization. XGC1 shows good weak and strong scalings to the maximal capability of the leadership HPCs.

3.2. α factor

The α factor is an small coefficient that determines the conversion rate of f_p to f_g according to Eq. (7). When a too large fraction of f_p from particles is converted into f_g on the grid, a numerical smoothing of δf_p can occur from the interpolation to corse-grained phase space grid. This can dissipate turbulence energy. Fig. 1 shows the heat fluxes with various α values. The simulation size is relatively small, and the characteristic system size is $a/\rho_i = 100$, where a is the minor radius and ρ_i is the ion gyro radius. In these simulations, 3×10^5 real space grid points, each of which has 32 $(v_\perp) \times 31$ $(v_\parallel)$ velocity space grid points, are used. The total number of marker particles are 4×10^8, which corresponds to $N_g = 1300$ particles per real space grid node or 1.5 particles per velocity space grid node. A simple statistical error bar $1/\sqrt{N_g}$ is shown together in Fig. 1. The temperature and density profiles used are simple hyperbolic tangent functions with maximal gradient of $R/L_T = 8$, and $R/L_n = 2$, where R is the major radius, L_T is the temperature gradient scale length, and L_n is the density gradient scale length. The adiabatic electron temperature is set to be the same as the ion temperature. The simulation shown in Fig. 1 is non-flux driven. Thus, the heat flux decays eventually as the free energy from the initial profile is exhausted. In Fig. 1, the heat flux grows as turbulence develops, and decays after it reaches the maximum due to the nonlinear effect, and decays further as the temperature profile relaxes. The maximum heat flux and the time integrated heat flux depends on the α factor. When α is 0.004 or 0.01, a clear reduction of turbulence and heat flux is observed, and this is a sign of turbulence damping from the numerical dissipation of the particle-grid interpolation. When $\alpha = 0.001$, the heat flux is acceptably close to the $\alpha = 0$ case.

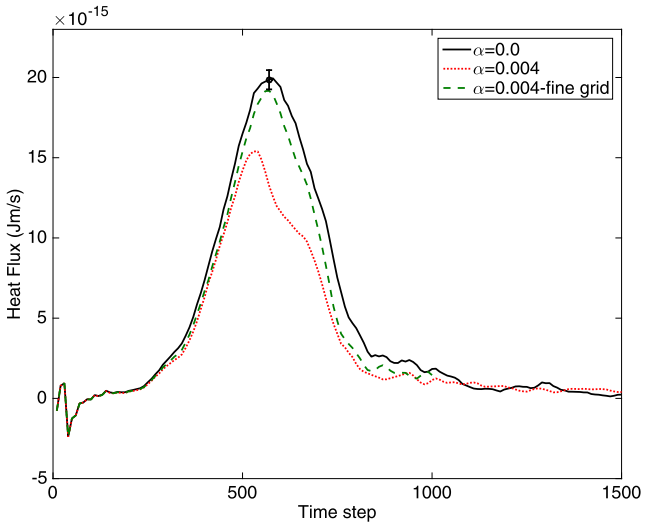

Fig. 2. Heat flux per particle from the XGC1 ITG turbulence simulations with $\alpha = 0$, $\alpha = 0.004$ on a coarse grained velocity grid (32×31), and $\alpha = 0.004$ on a finer grid (62×61). A standard $1/\sqrt{N_g}$ error bar is shown together. Finer grid restores the heat flux close to the $\alpha = 0$ case.

Numerical dissipation depends on the resolution of the velocity space grid. In Fig. 2, the heat flux from a finer (62×61) velocity space grid is compared with the results from the original 32×31 velocity space grid. When $\alpha = 0.004$ with the original grid, it shows reduction of the turbulence and heat flux level from the $\alpha = 0$ result. Increasing the grid resolution gives reduction of the numerical dissipation, and it restores the heat flux to an acceptable level even at $\alpha = 0.004$ at (62×61) velocity space grid.

In a rough estimation of the optimal α,

$$\alpha_{\text{opt}} \sim C(\Delta \mathbf{z}) \frac{\Delta t}{\tau_{\text{phy}}},$$

where Δt is the time step size of the particle-grid conversion operation, and τ_{phy} is the time scale of the physics phenomena (turbulence correlation time), and $C(\Delta \mathbf{z})$ is a function of order unity depending on the size of the velocity space grid, $\Delta \mathbf{z}$. Since τ_{phy} is the only internal quantity, which is similar among many simulations of a given tokamak, a common α-value can be used over many simulations as long as it is determined conservatively.

Numerical dissipation by α factor is also observed in the flux-driven simulations in which the central heat source supports the turbulent heat flux and the temperature gradient. Fig. 3 shows the heat fluxes and the temperature gradient scale length in the flux-driven simulations with various α values. Heat and cooling are applied to near axis and edge region, respectively. In the flux-driven plasma, simulations reach a quasi-steady state with the final heat flux being almost saturated. If we examine the temperature gradient scale length, the $\alpha = 0$ and $\alpha = 0.001$ cases converge to a similar value. However, $\alpha = 0.01$ gives higher gradient (in other words, less time integrated heat flux) compared to the $\alpha = 0$ case. This tendency is similar to that observed in the non-flux driven simulations of Fig. 1.

Fig. 4 shows the mean of $(w_1 w_0 g)^2$ in the velocity space cells with $\alpha = 0$ and $\alpha = 0.001$ after 1500 time steps. When $\alpha = 0.001$, the particle noise variance is reduced by factor of 4 compared to $\alpha = 0$ case. Even though $\alpha = 0.001$ appears to be a small number, $\alpha \times$ [No. of time steps] $= 1.5$, and the cumulative effect is significant enough.

3.3. Advantage in the computational cost in the ion temperature gradient turbulence of the core plasma

Discussions presented in this subsection will be confined to the simplified core plasma simulation of the ion temperature gradient turbulence without kinetic electrons, in which the plasma is not far from Maxwellian. A more significant advantage of the new scheme is in the non-Maxwellian edge plasma in contact with the wall, with the kinetic electron motions subcycled, but such a study will be presented in a future report.

Even in the case of the simple ion turbulence in the near Maxwellian core plasma, the gain in the computational cost is realized from the usage of the α factor, thus a lower marker particle number for the same noise quality in the new hybrid-Lagrangian scheme. As shown in the previous subsection with Fig. 4, the verified ITG turbulence with a small $\alpha =$ 0.001 yields factor of 4 reduction in the particle noise variance. This corresponds to factor of 2 reduction in the particle noise. To achieve factor of 2 reduction of statistical particle noise by increasing number in particles, we need 4 times more particles. Table 1 shows the computational cost of (A) total computing time in the main loop, (B) adjustment overhead for f_g and f_p using Eqs. (7) and (10), (C) direct weight evolution using Eq. (6), (D) main loop w/o (B) or (C), and (E) main loop with 4 times more particles w/o (B) and (C). The new hybrid scheme shows increase in the computational cost by 55% as

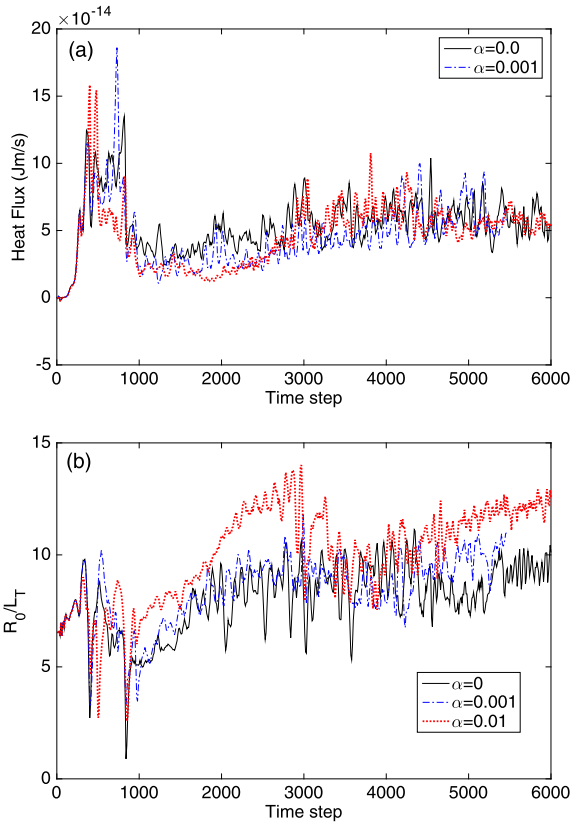

Fig. 3. Heat flux per particle (a) and temperature gradient scale length (b) from the XGC1 flux-driven ITG turbulence simulations with $\alpha = 0$, 0.001, and 0.01. When $\alpha = 0.001$ the gradient scale length converges to that of $\alpha = 0$. $\alpha = 0.01$ gives higher R_0/L_T than $\alpha = 0$ does due to reduction of turbulence. We note here that the heat flux always self-organize with the heating source in a flux-driven simulation, as shown in (a), by adjusting the temperature gradient.

the overhead in reducing the particle noise by factor of 2 [comparison between (D) and (A)]. Comparing with the equivalent simulation with 4 times more particles, (E), without the new scheme, the effective reduction of computational cost is about 56% in this simple case if we want to keep the same noise quality.

In the subcycled kinetic electron simulation of XGC1, the relative cost of (B) and (C) is fairly small since subcycled electrons do not require particle-grid operation at every electron time step. On the other hand, the electron particle push time increases by roughly the number of marker particles. For example, if we turn on the kinetic electrons in Table 1 case, the time (A) is only 18% higher than the time (D).

The new hybrid-Lagrangian scheme also saves compute memory for the same noise quality. Since each particle requires 5 gyrokinetic phase variables, w_1, and $w_0 g$, $7N_p$ variables, where N_p is the number of particles, are required when we do not use the new scheme. There are two types of additional memory required for the new scheme. One is $f_0(\mathbf{z}_{t-\Delta t}, t-\Delta t)$ for each particle to get Δf_0 for Eq. (6), where $\mathbf{z}_{t-\Delta t}$ is the phase space position of a particle at the previous time step. The other is the memory for phase space grid for f_g and f. The phase space grid for f is required when $S^*(f)$ is evaluated. With this scheme, $8N_p + 2N_v$ variables are required, where N_v is the number of phase space grid. Considering that $N_p/N_v = 1.5$ in the ITG simulation, only 33% additional memory is required to store particles and phase space grid for the new hybrid scheme. Comparing with the 4 times more particles, the new scheme reduces the memory requirement by 67% effectively.

4. Summary and discussion

A new hybrid-Lagrangian scheme for gyrokinetic simulation of tokamak edge plasma is developed and implemented in XGC1. The new scheme uses the combination of particle-in-cell and continuum grid method. Lagrangian particle push is used for the Vlasov dynamics on the left hand side of the Boltzmann equation. The irreversible right hand side physics of the Boltzmann equation is handled using the continuum gird method. The scheme can slowly convert particle weights into the coarse-grained phase space grid in order to mitigate the growing weight issue. Hence, the slow time-varying physics resides mostly on the phase space grid, while the fast time varying function remains in particles. The optimal conversion rate α depends on the physics to be resolved and the size of phase space grid. Once the α value is determined conservatively, many

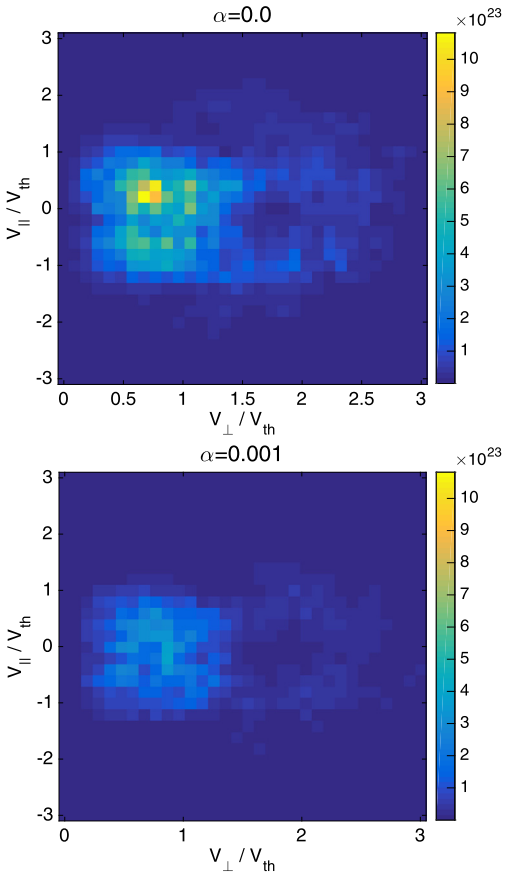

Fig. 4. Mean of $(w_1 w_0 g)^2$ for marker particles, obtained on the velocity space grid with $\alpha = 0$ (above) and $\alpha = 0.001$ after 1500 time steps (below). Factor of approximately 4 reduction is observed with $\alpha = 0.001$ compared to $\alpha = 0$.

Table 1
Computational cost of **(A)** main loop, **(B)** adjusting f_g and f_p using Eq. (7) and Eq. (10), **(C)** direct weight evolution using Eq. (6), **(D)** main loop w/o (B) or (C), and **(E)** main loop with 4 times more particles w/o (B) and (C).

	Simulation with adiabatic electrons	CPU-sec per step	Percentage to (D)	Percentage to (E)
(A)	Main loop with (B) and (C)	6.54×10^3	155%	44%
(B)	Adjusting f_g and f_p using Eq. (7) and Eq. (10)	1.65×10^3	39%	11%
(C)	Direct weight evolution using Eq. (6)	6.7×10^2	16%	4.5%
(D)	Main loop w/o (B) or (C)	4.22×10^3	100%	29%
(E)	Main loop of 4X more particles w/o (B) and (C)	1.47×10^4	351%	100%

similar plasmas can use a common α. The scheme reduces the statistical noise of a non-Maxwellian plasma simulation, and relaxes the growing weight problem, while avoiding the Omega-H instability.

The new scheme takes advantage of the good features from both particle and continuum schemes. Since the rapidly time varying fine structure is handled by the particles, the simulation is less sensitive to the Courant condition than in the brute force continuum method and possesses a higher velocity resolution. Since majority of the non-Maxwellian plasma resides on the continuum grid, the Monte-Carlo noise is reduced. Since the particles scale well on HPCs and the coarse-grained continuum grid reduces the memory requirement, the new scheme satisfies both the scalability and the low memory requirement on HPCs.

Acknowledgements

This work was supported by the U.S. Department of Energy under Contract No. DE-AC02-09CH11466 and Grant No. DE-SC000801. This work used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] W. Lee, J. Comput. Phys. 72 (1987) 243.
[2] Z. Lin, T. Hahm, W. Lee, W. Tang, R. White, Science 281 (1998) 556.
[3] S. Parker, C. Kim, Y. Chen, Phys. Plasmas 6 (1999) 1709.
[4] F. Jenko, W. Dorland, M. Kotschenreuther, B. Rogers, Phys. Plasmas 7 (2000) 1904.
[5] C. Chang, S. Ku, P. Diamond, Z. Lin, S. Parker, T. Hahm, N. Samatova, Phys. Plasmas 16 (2009) 056108.
[6] Y. Chen, S. Parker, J. Comput. Phys. 220 (2007) 839.
[7] Z. Lin, L. Chen, Phys. Plasmas 8 (2001) 1447.
[8] Y. Chen, S. Parker, J. Lang, G.-Y. Fu, Phys. Plasmas 17 (2010) 102504.
[9] E. Sonnendrückera, J. Rochea, P. Bertrandb, A. Ghizzob, J. Comput. Phys. 149 (1999) 201.
[10] X. Garbet, Y. Sarazin, V. Grandgirard, G. Dif-Pradalier, et al., Nucl. Fusion 47 (2007) 1206.
[11] J. Denavit, J. Comput. Phys. 9 (1972) 75.
[12] S. Vadlamani, S.E. Parker, Y. Chen, C. Kim, Comput. Phys. Commun. 164 (2004) 209.
[13] Y. Chen, S. Parker, Phys. Plasmas 14 (2007) 082301.
[14] Y. Chen, S. Parker, Phys. Plasmas 16 (2009) 052305.
[15] W. Wang, T. Hahm, W. Lee, G. Rewoldt, J. Manickam, W. Tang, Phys. Plasmas 14 (2007) 072306.
[16] S. Ku, C. Chang, P. Diamond, Nucl. Fusion 49 (2009) 115021.
[17] J. Krommes, G. Hu, Phys. Plasmas 1 (1994) 3211.
[18] G. Hu, J. Krommes, Phys. Plasmas 1 (1994) 863.
[19] Y. Chen, R. White, Phys. Plasmas 10 (1997) 3591.
[20] W.X. Wang, N. Nakajima, et al., Plasma Phys. Controll. Fusion 41 (1999) 1091, http://stacks.iop.org/0741-3335/41/i=9/a=303.
[21] E. Yoon, C. Chang, Phys. Plasmas 21 (2014) 032503.
[22] R. Hager, E. Yoon, S. Ku, et al., J. Comput. Phys. 315 (2016) 644–660.