Optimization by marker removal for $\delta f$ particle simulations

Wenjun Deng*, Guo-Yong Fu
Princeton Plasma Physics Laboratory, Princeton, NJ 08543, USA

A R T I C L E   I N F O
Article history:
Received 15 June 2013
Received in revised form 20 August 2013
Accepted 23 August 2013
Available online 30 August 2013

Keywords:
Particle simulation
Particle-in-cell
Marker particle
Particle distribution
Optimization
Arbitrary dimension

A B S T R A C T
A marker removal optimization technique is developed for $\delta f$ particle simulations. The technique uses the linear eigenmode structure in the equilibrium constant-of-motion space to construct an importance function, then removes some markers based on the importance function and adjusts the weights of the leftover markers to optimize the marker distribution function, so as to save markers and computing time. The technique can be directly applied to single-mode linear simulations. For multi-mode or nonlinear simulations, the technique can still be directly applied if there is one most unstable mode that dominates the simulation and $\delta f$ does not change too much in the nonlinear stage, otherwise special care is needed, which is discussed in detail in this paper. The technique’s effectiveness, e.g., marker saving factor, depends on how localized $\delta f$ is. The technique can be used for a phase space of arbitrary dimension, as long as the constants of motion in equilibrium can be found. In this paper, the technique is tested in a 2D bump-on-tail simulation and a 5D gyrokinetic toroidal Alfvén eigenmode (TAE) simulation and saves markers by factors of 4 and 19, respectively. The technique is not limited to particle-in-cell (PIC) simulations but could be applied to other approaches of marker particle simulations such as particle-in-wavelet (PIW) and grid-free treecode simulations.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Marker particle simulation and motivation for optimization

Marker particle simulation (usually shortened as particle simulation) is a popular tool for plasma physics studies. It uses a small set (compared to the set of actual physical particles in plasma) of computational particles for Monte Carlo sampling of the physical particle distribution function [1]. These computational particles are called marker particles (shortened as markers). Sometimes they are also called super-particles or macroparticles. Some simulations use markers to sample the whole physical distribution function $f$, which are called full-$f$ or total-$f$ simulations. Some others decompose the distribution $f$ into two parts: an equilibrium part $f_0$ that is usually time-independent and known analytically, and a perturbed part $\delta f$. And they use markers to sample only the perturbed distribution $\delta f$ and are thus called $\delta f$ simulations.

There are multiple simulation approaches that make use of marker particles. The oldest one is the particle-in-cell (PIC) approach that dates back to the 1950s [2]. The PIC approach is still very popular today and is used in a lot of currently active plasma simulation codes, such as GTC [3], GEM [4], XGC [5], AWACS [6], GYGES [7], HMGC [8], MEGA [9], M3D-K [10], etc. There are also other approaches that are developed to improve the PIC approach, such as the particle-in-wavelet (PIW) approach [11] that improves accuracy, and the grid-free treecode approach [12] that solves the PIC approach’s difficulty in revolving particle interactions within one cell.

For better resolution in phase space, i.e., lower particle noise, in particle simulations, it is wise to load markers such that the marker distribution function (MDF), denoted as $g$ in this paper, is proportional to the physical particle distribution function $f$, which is called “importance sampling” in Ref. [1]. This can usually be done straightforwardly in full-$f$ simulations. For example, if a simulation starts with a Maxwellian distribution $f|_{t=0} = f_{\text{Maxw}}$, then markers are loaded with a Maxwellian distribution $g|_{t=0} = C \cdot f_{\text{Maxw}}$, such that $f/g = 1/C$ is a constant, so as to achieve the “importance sampling”. Note that in collisionless simulations, the evolution of both $f$ and $g$ are governed by the Vlasov equation (or the gyrokinetic equation in gyrokinetic simulations), i.e., $df/dt = 0$ and $dg/dt = 0$, so $f/g$ stays constant during the whole simulation when $g$ is initially loaded with the “importance sampling”, meaning that the “importance sampling” is effective during the whole simulation. Things become different for $\delta f$ simulations, where $df/dt \neq 0$. Loading markers such that $\delta f/g$ is a constant initially does not guarantee $\delta f/g$ to stay constant during the simulation. To guarantee that $\delta f$ is well-resolved during the whole simulation, typical $\delta f$ simulations load markers in the phase space either uniformly or such that $g$ is proportional to the total distribution function $f$.  

* Corresponding author.
E-mail addresses: weng@wdeng.info (W. Deng), fu@pppl.gov (G.-Y. Fu).
URL: http://wdeng.info (W. Deng).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.08.019
Either loading method usually causes markers to spread all over the phase space.

Energetic particle physics is crucial to ITER and other fusion devices. Instabilities excited by energetic particles are being broadly studied by PIC simulations, such as toroidal Alfvén eigenmode (TAE) [13–22], reversed shear Alfvén eigenmode (RSAE) [23–25], β-induced Alfvén eigenmode (BAE) [26–31], etc. These modes usually exhibit localized δf structures, i.e., δf is nearly zero in a large portion of the phase space. The localized δf structures are usually but not always near the wave–particle resonance regions. Maintaining a large number of markers in the δf ≈ 0 regions is a waste of computing time. Reducing marker numbers in these regions to optimize the MDF can save a lot of computing time.

1.2. Candidate optimization techniques

There are at least two categories of techniques for optimizing the MDF during the simulation: marker increase and marker reduction. In the first category, the simulation initially loads a small amount of markers. Then during the simulation when δf is known, in the regions where δf is localized, every one (or a few) marker is split into multiple markers to increase resolution. Or alternatively, when δf is known, the initial set of markers is removed and a new set of markers in a larger amount is loaded based on the calculated δf, to get better resolution. The reloading technique, also called phase space remapping, has been systematically developed for PIC simulations in 2D [32] and 4D [33] phase spaces. In the second category, the simulation initially loads a large amount of markers, then merges or removes markers in the δf ≈ 0 regions.

The first category is a process of upsampling from a low-resolution sample and is difficult in general. The splitting technique is especially difficult because:

1. it can easily introduce a non-conservation problem if not treated carefully [34], bringing in systematic error;
2. the children markers split from one parent marker are correlated; the statistical uncertainty of a Monte Carlo simulation goes ∼ 1/√N where N is the number of independent sampling points; the splitting technique does not increase independent markers (unless after the decorrelation time, which could be very long) and therefore it is hard to reduce statistical uncertainty.

The second category is a process of downsampling from a high-resolution sample and is relatively easier than the first category. For the merging technique, finding nearby markers to merge is not very easy, especially in high-dimensional phase space. In this work, a marker removal technique is developed.

This paper firstly presents the algorithm of the marker removal optimization technique in Section 2. Then the technique is applied to a 2D bump-on-tail simulation in Section 3 and a 5D gyrokinetic TAE simulation in Section 4. Further possible extensions of the technique are discussed in Section 5. Finally, a summary is given in Section 6.

2. Algorithm of marker removal optimization technique

Our marker removal optimization technique can be applied to a phase space of arbitrary dimension, as long as the constants of motion in equilibrium can be found. In such a phase space I’, the volume element dI’ can be written as:

\[ dI' = J d\mathbf{z}_n d\mathbf{x}_n, \]

where J denotes the Jacobian, \( \mathbf{z}_n \) denotes the constants of motion, and \( \mathbf{x}_n \) denotes the non-constant coordinates.

The marker removal is performed only once in the early linear stage, when the dominant linear eigenmode just forms. The removal procedure consists of two marker loops.

The first marker loop is to calculate an importance function. Before normalization, the importance function is defined as:

\[ i(z_j) = \frac{\int |\delta f(z_j, s)| d\mathbf{z}_n}{\int g(z_j, \mathbf{z}_n) d\mathbf{z}_n}, \]

where g(z_j, \( \mathbf{z}_n \)) is the MDF. Making a grid on \( z_j \), the importance function can be approximated by:

\[ i(z_j) \approx \frac{\sum_j |w_j| S(z_{ij} - z_j)}{\sum_j S(z_{ij} - z_j)} \]

where \( w_j \) is the perturbed weight representing δf/g for each marker, \( z_{ij} \) is the constants of motion for each marker, S( ) is the shape function for marker-grid interpolation, and \( \sum_j \) sums over all markers. Note that this approximation may overestimate i(z_j) because:

\[ i(z_j) \approx \frac{\int \sum_j |w_j| S(z_{ij} - z_j) S(n_j - n)| d\mathbf{z}_n}{\int \sum_j S(z_{ij} - z_j) S(n_j - n) d\mathbf{z}_n} \]

This overestimation should be small and is not important because the accuracy of the importance function is not important as will be discussed in Section 5.1. The importance function is normalized as:

\[ I(z_j) = \frac{i(z_j)}{\max(i(z_j))}. \]

This normalization makes the function I(z_j) in the range of [0, 1].

The second marker loop uses I(z_j) to identify the importance of each marker and determine whether each marker should be removed. For each marker indexed \( j \),

1. its importance \( I_j \) is calculated directly by the importance function:
2. then a uniform [0, 1] random number called “dice” is generated;
3. if dice > I_j, remove this marker; otherwise, keep this marker and scale up its weights: \( w_j \rightarrow w_j/l_j, p_j \rightarrow p_j/l_j \).

Here \( p_j \) is the total weight for each marker. Note that the above procedure applies to particle simulations with two weights: the perturbed weight \( w = \delta f/g \)
and the total weight
\[ p = f/g. \]

Some particle simulations use a single weight \( w_s = \delta f/f \) based on loading markers such that g and f are proportional. This marker removal technique cannot be directly applied to such a single-weight scheme, because g and f are not proportional anymore after marker removal.

This marker removal technique makes the MDF to have the same shape as \( \int |\delta f| g d\mathbf{z}_n \), and only statistical error is introduced to the physical distribution function, as well as mass, momentum and energy. In the limit of an infinite number of markers, the...
physical distribution function and its moments are conserved exactly.

Because the importance function and the MDF after marker removal are functions of the constants of motion in equilibrium, the MDF in the linear stage is time independent. As a result, the optimization effect is kept during the linear stage. In the nonlinear stage, the optimization is still effective if the importance function does not change too much from the linear stage.

Note that when there are multiple species in the system, each species has its own importance function and the above procedure needs to be performed for each species.

The next two sections will present numerical results of the marker removal optimization technique in 2D and 5D phase spaces.

3. Numerical results in 2D electrostatic bump-on-tail simulation

To test the marker removal optimization technique, we start with a simple 2D electrostatic bump-on-tail simulation. This section first presents the PIC simulation model in Section 3.1, then introduces the bump-on-tail instability and compares the analytic solution and PIC simulation result in Section 3.2. Next, a rigorous convergence test is introduced for evaluation of the effectiveness of the optimization technique and is demonstrated on the bump-on-tail simulation in Section 3.3. Finally, in Section 3.4, the marker removal technique is tested on the bump-on-tail simulation and its effectiveness is evaluated through the rigorous convergence test.

3.1. Simulation model

The simulation model mainly consists of an electrostatic δf Vlasov equation and a Poisson equation:

\[ \frac{d}{dt} \delta f_\alpha(x, v, t) = - \frac{d}{dv} E(x, t) \delta f_\alpha(v), \]

\[ \partial_t E(x, t) = \sum_\alpha Z_\alpha \int_{-\infty}^{\infty} \delta f_\alpha(x, v, t) dv, \]

where the subscript \( \alpha \) indicates the particle species; \( Z \) is the particle charge; \( m \) is the particle mass; \( \delta f \) is the perturbed distribution function and is a function of the position \( x \), the velocity \( v \) and the time \( t \); \( f_0 \) is the equilibrium distribution function and is assumed to be independent on \( x \) and \( t \); and \( E \) is the electrostatic field. Here we use a two-weight scheme similar to that in Ref. [35]. The marker evolution equations then write:

\[ \frac{dx_\alpha}{dt} = v_\alpha, \]

\[ \frac{dv_\alpha}{dt} = \frac{Z_\alpha}{m_\alpha} E(x_\alpha), \]

\[ \frac{dp_\alpha}{dt} = 0, \]

\[ \frac{d\omega_\alpha}{dt} = -(p_\alpha - w_\alpha) \frac{Z_\alpha}{m_\alpha} E(x_\alpha) \frac{\partial f_0}{f_0}, \]

where the subscript \( \omega \) indicates the marker index; and \( w \) and \( p \) are the perturbed and total weights given by Eqs. (8) and (7), respectively.

All quantities in the above equations are already normalized as follows: time is normalized by \( 1/\omega_{pe} \); length is normalized by electron Debye length \( \lambda_{De} \); mass is normalized by electron mass \( m_e \); charge is normalized by proton charge \( e \); temperature is normalized by electron temperature \( T_e \); density is normalized by electron equilibrium density \( n_{eq} \); and electric field is normalized by \( e/\left(T_e m_e \right) \).

3.2. Electron bump-on-tail instability

Here we keep using the normalization in Section 3.1 and make the following additional assumptions: ions are immobile and they only provide a fixed background; electrons have a bump-on-tail distribution:

\[ f_{be}(v) = \frac{1}{\sqrt{2\pi}} \left( 1 - n_b \right) \exp \left( -\frac{v^2}{2} \right) \]

\[ + \frac{n_b}{v_{th,b}} \exp \left[ \frac{(v - v_{th})^2}{2v_{th,b}^2} \right], \]

where \( n_b \) represents the electron density of the beam, \( v_{th,b} \) represents the beam velocity, and \( v_{th,b} \) represents the thermal velocity of the beam. The linear dispersion relation for this scenario is:

\[ 0 = 1 + \frac{1}{k^2} \left( 1 + \zeta_0 Z(Z_\alpha) \right) + \frac{n_b}{T_0} \left( 1 + \zeta_0 Z(Z_b) \right), \]

\[ \zeta = \frac{\omega}{\sqrt{2k}}, \]

\[ \zeta_b = \frac{\omega}{\sqrt{2v_{th,b}}}, \]

where \( Z(\zeta) \) is the plasma dispersion function. This dispersion equation can be solved numerically using Muller’s method [36]. Specifically, in this work we use these parameters: \( n_b = 0.1 \), \( v_{th,b} = 5 \), \( v_{th,b} = 0.1 \), \( k = 0.401 \). Muller’s method gives the real frequency and growth rate as: \( \omega = 1.4674 \), \( \gamma = 0.029120 \). The expression for \( \delta f \) is:

\[ \delta f_\alpha(x, v, t) = \left[ 1 - n_b \right] \frac{v}{\sqrt{2\pi}} \exp \left( -\frac{v^2}{2} \right) \]

\[ + \frac{n_b}{v_{th,b}} \frac{v - v_{th}}{\sqrt{2v_{th,b}^2}} \frac{1}{\sqrt{2\pi}} \exp \left[ \frac{(v - v_{th})^2}{2v_{th,b}^2} \right] \]

\[ \times \left[ \frac{i}{\omega - ku} E_0 \exp^{i(kx - \omega t) + \text{complex conjugate}} \right], \]

where \( E_0 \) is the initial electric field amplitude. The structure of \( \delta f \) in phase space for \( t = 2\pi j/\omega \) with \( j \) being an arbitrary integer is shown in Fig. 1(a).

The PIC simulations of the bump-on-tail instability are performed by the code PIC1D-PEStc [37]. The simulation domain in \( x \) covers one wavelength \( \lambda = 2\pi/k \) with 192 grid points and periodic boundary condition (grid size \( \Delta x = 2\pi/0.401/192 = 0.0816 \). The Poisson equation is solved by a Fourier solver, and only the lowest mode (with the mode number \( k = 0.401 \)) is kept. The markers are loaded uniformly in \( x \) and in the velocity range of \([-8, 8]\). The simulation result in the linear stage gives \( \omega = 1.47 \), \( \gamma = 0.0287 \), and \( \delta f \) structure as shown in Fig. 1(b). The PIC simulation result agrees well with the semi-analytic (Muller’s method) solution.

3.3. Rigorous convergence test for marker particle simulations

To evaluate how good the marker removal optimization technique is, a robust method for numerical error measurement is needed. A marker particle simulation is essentially a Monte Carlo simulation [1], whose results, e.g., linear growth rate, nonlinear...
saturation level, etc., are random variables with certain expectations and standard deviations. The expectations should be equal to the physical results, and the standard deviations usually go as $\sim 1/\sqrt{N}$ due to the central limit theorem, where $N$ is the number of sampling points, i.e., the number of markers in a particle simulation. In this case, the simulation results converge to the physical results as $N \to \infty$.

Because the statistical uncertainty is random for every run, it is not revealed by a traditional single-scan convergence test, which performs the simulation once for each marker number. For example, Fig. 2(a) gives a traditional single-scan convergence test, showing the effect of the marker number on the linear growth rate of the bump-on-tail instability by PIC1D-PETSc simulation. Due to the random fluctuation in each run, this convergence test has three issues:

1. the convergence trend is not very clear;
2. the inverse-square-root relation between the statistical uncertainty and the number of markers is not shown;
3. it is not a consistent test as redoing the test with markers loaded with the same distribution but with a different random number sequence may give a very different image; Fig. 2(b) and (c) give another two possible results of such redoing.

By contrast, a more rigorous convergence test is to perform the simulation multiple times as a sample, with markers loaded with different random number sequences, which can be achieved simply by changing the random seeds, then use the sample statistics to get a clearer error estimate as shown in Fig. 3. Here we assume that the distributions of the linear growth rate and other macroscopic simulation results are Gaussian without rigorous mathematical proof. From statistics we know that, if from a Gaussian random variable $X$ we take a sample $\{X_j, j = 1 \cdots N\}$ which is of size $N$, the mean of the sample $\overline{X}$ is also a Gaussian random variable with the same expectation as $X$, i.e., $\mathbb{E}(\overline{X}) = \mathbb{E}(X)$, and a standard deviation $\sigma(\overline{X}) = \sigma(X)/\sqrt{N}$, where $\sigma(X)$ is the standard deviation of $X$. According to Cochran’s theorem [38], the quantity $\sqrt{N - 1} s(X)/\sigma(X)$ has a chi distribution with $N - 1$ degrees of freedom, where $s(X)$ is the standard deviation of the sample. Based on these properties, we can estimate the expectation and the statistical uncertainty of the linear growth rate and other simulation results. Here we take the sample size to be 20, i.e., 20 simulations are performed with different random seeds for every choice of the marker number. The sample means as estimations of the expectations of the linear growth rate are shown in Fig. 3(a). The error bars give the intervals of $\overline{\gamma} \pm 2 s(\gamma)/\sqrt{20}$ and therefore are $\sim 95\%$ confidence intervals, where $\overline{\gamma}$ and $s(\gamma)$ are the mean and the standard deviation of the sample growth rate, respectively. It can be seen that these data points have approximately the same $\gamma$ values within error bars. This indicates that a finite marker number does not introduce apparent systematic error. The sample standard deviations as estimations of the standard deviations of the linear growth rate are shown in Fig. 3(b) (normalized by the estimated expectation). The error bars of the standard deviation give the intervals of $[s(\gamma)/\overline{\gamma}](100\% \pm 30\%)$ and therefore are $\sim 93\%$ confidence

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{\textit{δf linear structure in (x, v) phase space. (a) Semi-analytic (Muller’s method) solution. (b) Simulation by PIC1D-PETSc.}}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Traditional single-scan convergence tests of linear growth rate of bump-on-tail instability vs number of markers. (a)-(c) give three possible test results when markers are loaded with the same distribution but with different random number sequences.}
\end{figure}
normal importance function becomes:

\[
\text{i}(v) = \frac{\int \delta f(x, v) \, dx}{\int f(x, v) \, dx} \approx \frac{\sum |w_j| S(v_j - v)}{\sum S(v_j - v)}.
\]

(19)

The normalized importance function becomes:

\[
I(v) = \frac{i(v)}{\max_i(|i(v)|)}.
\]

(20)

3.4. Effectiveness of marker removal optimization technique

Here the marker removal technique is applied to the electron bump-on-tail simulation described in Section 3.2 and is evaluated by the rigorous convergence test described in Section 3.3.

In the \((x, v)\) 2D phase space, \(v\) is a constant of motion in equilibrium and \(x\) is a non-constant coordinate. Therefore, the importance function before normalization becomes:

\[
\text{i}(v) = \frac{\int \delta f(x, v) \, dx}{\int f(x, v) \, dx} \approx \frac{\sum |w_j| S(v_j - v)}{\sum S(v_j - v)}.
\]

The normalized importance function becomes:

\[
I(v) = \frac{i(v)}{\max_i(|i(v)|)}.
\]

(20)

In our bump-on-tail simulation, a grid in \(v\) with 128 grid points (grid size \(\Delta v = 16/(128 - 1) = 0.126\)) is used for calculation of the importance function. The time evolution of the electric field energy is shown in Fig. 4. As described in Section 2, the marker removal technique is applied in the early linear stage when the linear eigenmode just forms. The MDF before and after the marker removal are shown in Fig. 5. It can be seen that the originally uniform MDF becomes concentrated near the peaks and valleys of \(d\ell\) after the marker removal.

The rigorous convergence test method described in Section 3.3 is used to test two common macroscopic quantities in simulation results, linear growth rate and nonlinear saturation level, so as to measure the optimization effect of the marker removal technique. The convergence results are shown in Figs. 6 and 7. It can be seen from the upper panels of both figures that “optimized” data points are vertically very close to the “regular” ones, indicating that the marker removal technique does not introduce apparent systematic error. The lower panels of these two figures show that the marker removal technique reduces the statistical uncertainty by a factor of \(\sim 2\), indicating that retaining the same statistical uncertainty, the marker removal technique can save markers by a factor of \(\sim 4\) due to the inverse-square-root relation.

4. Numerical results in 5D gyrokinetic simulation

In this section, the marker removal optimization technique is applied to a TAE simulation by the kinetic/MHD hybrid code M3D-K [10]. The marker removal technique has been implemented in the kinetic part that solves energetic particles in the 5D gyrokinetic phase space.

In the 5D gyrokinetic phase space, the phase space volume element is:

\[
\delta \mathbf{v} = \delta \mu \delta \ell \, dt \, d\xi \, d\ell, \quad \text{where} \; \delta \mathbf{v} \text{ is the Jacobian and } \mathbf{E}, \mu, P, \zeta, \ell \text{ are gyro-center energy, magnetic moment, toroidal angular momentum, toroidal angle and length along projected poloidal orbit, respectively. The length along projected poloidal orbit } \ell \text{ is the same as the one defined in Ref. [39].}
\]

The reason to choose this special coordinate system for marker removal is to separate constants of motion and non-constant coordinates. In this case, the constants of motion are \(z_k = (\mathbf{E}, \mu, P, \ell)\) and the non-constant coordinates are \(z_{\ell} = (\zeta, \ell)\). The unnormalized importance function in the 5D phase space is:

\[
i(\mathbf{E}, \mu, P) = \frac{\int \delta f(\mathbf{v}) \, d\xi \, d\ell}{\int g \delta \mathbf{v} \, d\xi \, d\ell} \approx \frac{\sum |w_j| S(\mathbf{E}_j - \mathbf{E}) S(\mu_j - \mu) S(P_{\ell,j} - P_{\ell})}{\sum S(\mathbf{E}_j - \mathbf{E}) S(\mu_j - \mu) S(P_{\ell,j} - P_{\ell})}.
\]

(21)

The normalized importance function is:

\[
I(\mathbf{E}, \mu, P) = \frac{i(\mathbf{E}, \mu, P)}{\max_{\mathbf{E}, \mu, P} [i(\mathbf{E}, \mu, P)]}.
\]

(22)
A simulation is set up for an $n = 1$ TAE excited by energetic particles in a tokamak with circular poloidal cross-section. The plasma parameters are the same as those of the base line case described in Section II B in Ref. [15]. The energetic particle distribution function is the same as the one given in Section II A in Ref. [15] except that collision is suppressed here. The mode structures in the linear and nonlinear stages are similar as can be seen from Fig. 8. The time history of the MHD perturbation energy is shown in Fig. 9. The third nonlinear peak is marked in this figure as this is used in the convergence tests for measuring the marker saving effect in the nonlinear stage.

As described in Section 2, the marker removal is performed in the early linear stage when the linear eigenmode just forms. The importance function is calculated on a $60 \times 64 \times 68$ grid in the $(\mu, E, P_t)$ constant-of-motion space. Fig. 10 shows the change of MDF by the removal in one $E-P_t$ slice ($\mu = 0$) of the constant-of-motion space. In this simulation, initially the markers are loaded uniformly in $(R, Z, \zeta, v_\parallel, v_\perp)$ coordinates. As we look at the system in the constant-of-motion space, the coordinate transformation Jacobian $J_G$ makes the MDF concentrated at the bottom of the phase space slice as shown in Fig. 10(a). Meanwhile, the $\delta f$ structure located mainly on the left side of the phase space slice as shown in Fig. 10(b). After marker removal, the markers become concentrated in a narrow band on the left side of the phase space slice as shown in Fig. 10(c).

The rigorous convergence test method is applied to linear growth rate, nonlinear saturation level, and also the amplitude of the third nonlinear peak. The results are given in Figs. 11–13, respectively. From these figures, it is again verified that numerical error introduced by finite marker number is mainly statistical. It can be also seen that the marker removal technique does not introduce apparent systematic error. The marker removal technique saves markers by factors of 19, 34 and 24 with respect to linear growth rate, nonlinear saturation level, and third nonlinear peak, respectively. The overall saving factor is the minimum of the saving factors with respect to all macroscopic quantities according to Liebig’s Law of the Minimum. Therefore, the overall saving factor for this simulation is no more than 19.
Fig. 8. TAE mode structures in our simulation: (a) linear stage; (b) nonlinear stage.

Fig. 9. Time history of MHD perturbation energy in our TAE simulation. Marker removal is performed at the early linear stage when the linear eigenmode just forms.

5. Possible extensions of the marker removal optimization technique

5.1. Flexibility of importance function

In this marker removal technique, the importance function is only for controlling what fraction of markers is to be kept and it does not directly enter the simulation system. In the limit of an infinite number of markers, no matter what importance function is used, the simulation result do not change. Therefore, the accuracy of the importance function is not crucial. As a result, the grid in constant-of-motion space for calculation of the importance function does not need to have such a small grid size that resolves all fine structures in $\delta f$ since the grid size does not directly contribute to the numerical error of the simulation. Also, the shape function for calculation of the importance function does not need to be of high order. A simple piecewise linear function is good enough. Furthermore, the choice of importance function is flexible and not unique. The importance function Eq. (2) may lead to an aggressive marker removal, which may bring in quite large statistical uncertainty to a single run. Although the rigorous con-

Fig. 10. In one $\xi - P_i$ slice ($\mu = 0$) of the constant-of-motion space in our TAE simulation: (a) MDF ($\int g \delta f d\xi d\ell |_{\mu=0}$) before removal; (b) perturbed distribution ($\int \delta f g \delta f d\xi d\ell |_{\mu=0}$) before removal; (c) MDF after removal.

Fig. 11. Convergence test of linear growth rate of TAE comparing the cases without and with marker removal optimization.
IHere are two examples of such less aggressive functions:
silver removal may be more practical. Such a less aggressive removal
it need to perform the simulation too many times. A less aggressive
test for every system scenario as it needs to perform the simulation too many times. A less aggressive
removal can be achieved by adjusting the normalized importance function. Here are two examples of such less aggressive functions:
\begin{align}
I_{at}(\mathbf{z}_c) &= \left| I(\mathbf{z}_c) \right|^{\xi} \quad 0 < \xi < 1, \\
I_{at}(\mathbf{z}_c) &= \frac{\tanh \left[ \eta \left( I(\mathbf{z}_c) - 1/2 \right) \right]}{\tanh \left( \eta/2 \right)} \quad \eta > 0, \tag{23}
\end{align}
where \( I(\mathbf{z}_c) \) is the original normalized importance function given by Eq. (5). One can simply use one of these functions to calculate the importance of each marker instead of the original normalized importance function to achieve a less aggressive removal effect. The aggressiveness can be adjusted by the parameters \( \xi \) and \( \eta \). The optimal values of \( \xi \) and \( \eta \) need further investigation to be determined.

5.2. Applicability in multi-mode and nonlinear simulations

Our marker removal optimization technique is good for single-mode linear simulations (or the linear stage of nonlinear simulations), because in this case the normalized importance function does not change over time in the linear stage, so the optimization effect, i.e., the “importance sampling”, is sustained. In multi-mode linear simulations, usually there is one most unstable mode which dominates the simulation, in which case the technique is still good. In the nonlinear stage, the technique is still good if the \( \delta f \) structures do not change or move too much; or if the \( \delta f \) structures move, but the markers also move along. This is why the rigorous convergence tests show that the technique is still good in the nonlinear stage for both the 2D bump-on-tail simulation and the 3D TAE simulation.

If there are multiple modes that are comparably unstable and are all important in the simulation, or \( \delta f \) changes too much and the markers do not follow along in the nonlinear stage, then the marker removal technique is not recommended for the very first run; but if a second run is needed for whatever reason, the technique is useful by constructing an importance function from the previous run, which will be described in Section 5.3.

5.3. Construct importance function from known properties

When some properties of the simulation are known before the simulation is performed, the importance function may be constructed from those known properties instead of from the simulated linear \( \delta f \). For example, if the linear \( \delta f \) can be calculated analytically, then the importance function can also be calculated before the simulation. In this case, the marker removal can be performed at the very beginning of the simulation instead of at the early linear stage. Or alternatively, the markers can be initially loaded to distribute as the importance function in the constant-of-motion space and no removal needs to be performed. This alternative way needs to find the mapping of particle orbits between regular phase space coordinates, which are simply the position and the velocity, and constant-of-motion space coordinates. This is solved in tokamak geometry [39], but may be unsolved and difficult in other geometries. In a geometry where the mapping is difficult to find, the marker removal is a good replacement.

Sometimes a simulation needs to be re-performed because a higher resolution is needed, or output of more data is needed, or some minor mistakes were made in the previous simulation. If the previous simulation is still trustable, then it can be used to construct the importance function for the next simulation:
\begin{equation}
I_{next}(\mathbf{z}_c) = \max_i I_{prev}(\mathbf{z}_i, t), \tag{25}
\end{equation}
where \( I_{prev}(\mathbf{z}_i, t) \) is the normalized importance function for the previous run:
\begin{equation}
I_{prev}(\mathbf{z}_i, t) = \frac{I_{prev}(\mathbf{z}_i, t)}{\max_{\mathbf{z}_c} I_{prev}(\mathbf{z}_i, t)}, \tag{26}
\end{equation}
\begin{equation}
i_{prev}(\mathbf{z}_c, t) = \int \frac{\delta f(\mathbf{z}_c, t, \mathbf{z}_n) \delta d\mathbf{z}_n}{\int g(\mathbf{z}_c, \mathbf{z}_n, t) \delta d\mathbf{z}_n} \approx \frac{\sum_j w_j(t) S(\mathbf{z}_{cj}(t) - \mathbf{z}_c)}{\sum_j S(\mathbf{z}_{cj}(t) - \mathbf{z}_c)}. \tag{27}
\end{equation}
where $\delta f, g, w_j$ and $z_j$ are taken from the previous run. Note that the importance function for the previous run is time dependent and it needs to be calculated over the whole time domain of the previous run to feed Eq. (25) for the maximum operator over time. 

5.4. Automatic selection of marker removal time

In the simulations described in Sections 3 and 4, the marker removal times are specified manually and are determined based on regular simulations without marker removal. For a simulation of a different problem, the marker removal time can also be programmed to be automatically detected. Whether the linear eigenmode has formed can be detected by the stability of instantaneous growth rate and/or the stability of the normalized mode structure. Once the linear eigenmode has formed, the marker removal can and should be performed.

5.5. Applicability in simulations of microturbulence

PIC simulation is widely used to study microturbulence. It is recommended to follow the discussions in Sections 5.2 and 5.3 when applying the marker removal optimization technique to microturbulence simulations. Note that microturbulence may occupy a large portion of the phase space, so the marker saving factor may be small for such an application.

5.6. Applicability in full-f simulations

It is frequently asked whether the marker removal optimization technique can be used in full-f simulations. In full-f simulations, the “importance sampling” can be simply achieved by loading markers such that $g / f$ is a constant, as suggested in Ref. [1] and discussed in Section 1. When $f$ is a simple function, e.g., Maxwellian, it is usually easy to directly load markers such that $g$ is proportional to $f$. When $f$ is a complicated function, direct loading may be difficult. In this case, one can first load markers arbitrarily (the simplest choice would be uniform loading), then use such a simple importance function $z = f / g$, normalize it, and then perform marker removal to achieve the “importance sampling”.

5.7. Extension to other approaches of marker particle simulation

Although we applied the marker removal technique to particle-in-cell simulations in Sections 3 and 4, the technique does not directly touch the “cell” part. Therefore, the marker removal optimization technique can be directly applied to other approaches of marker particle simulation such as the PIW simulation [11] and the grid-free treecode simulation [12].

5.8. Extension to include collisions

The marker removal technique deals only with collisionless particle simulations in this paper. For simulations with collisions, modification of the technique may or may not be needed and might depend on the collision model. Further investigation is needed on this matter and is not concluded in this paper.

6. Summary

A marker removal optimization technique has been developed for $\delta f$ particle simulations. The technique uses the linear eigenmode structure in the equilibrium constant-of-motion space to construct an importance function by Eqs. (2) and (5), then removes some markers based on the importance function and adjusts the weights of the leftover markers to optimize the MDF. The technique is tested on a 2D bump-on-tail simulation and a 5D TAE simulation and can save markers by factors of 4 and 19, respectively. The marker saving factor depends on how localized the $\delta f$ structure is. The technique can be directly applied to single-mode linear simulations. For multi-mode simulations, if there is one most unstable mode that dominates the simulation, the technique can still be directly applied. For nonlinear simulations, if $\delta f$ does not change too much or the MDF follows the change of $\delta f$, the technique can still be directly applied. If there are multiple modes that are comparably unstable and are all important in the simulation, or $\delta f$ changes too much in the nonlinear stage and the MDF does not follow, the technique can be applied but the importance function needs to be constructed from other sources, such as a previous run or analytic theories. The technique is not limited to PIC simulations but could be applied to other approaches of marker particle simulations such as PIW and treecode simulations.

Acknowledgments

This manuscript has been authored by Princeton University under Contract Number DE-AC02-09CH11466 with the US Department of Energy (DOE). The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. This work has also been supported by the US DOE SciDAC Center for Nonlinear Simulation of Energetic Particles in Burning Plasmas (CSEP). Simulations have been performed using the Hopper supercomputer at NERSC. The authors acknowledge useful discussions with C.S. Chang, W.W. Lee, P. Porazik and B. Wang.

References


