Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2021-0200

Vol. **31**, No. 4, pp. 997-1019
April 2022

Random Batch Particle Methods for the Homogeneous Landau Equation

José Antonio Carrillo[1], Shi Jin[2] and Yijia Tang[3,*]

[1] *Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK.*
[2] *School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University, Shanghai, 200240, P.R. China.*
[3] *School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, 200240, P.R. China.*

Received 12 October 2021; Accepted (in revised version) 21 March 2022

Abstract. We consider in this paper random batch particle methods for efficiently solving the homogeneous Landau equation in plasma physics. The methods are stochastic variations of the particle methods proposed by Carrillo et al. [J. Comput. Phys.: X 7: 100066, 2020] using the random batch strategy. The collisions only take place inside the small but randomly selected batches so that the computational cost is reduced to $\mathcal{O}(N)$ per time step. Meanwhile, our methods can preserve the conservation of mass, momentum, energy and the decay of entropy. Several numerical examples are performed to validate our methods.

AMS subject classifications: 65C35, 65Y20, 82C40, 82D10

Key words: Homogeneous Landau equation, random batch particle method, Coulomb collision.

1 Introduction

The Fokker-Planck-Landau equation, originally derived by Landau [28], is a fundamental integro-differential equation describing the evolution of the distribution for charged particles in plasma physics [37]. It models the binary collisions between charged particles with long-range Coulomb interaction, which is the grazing limit of the Boltzmann equation [12,14,41]. Denote by $f(t,x,v)$ the mass distribution of charged particles at time t, position x with velocity v, the Fokker-Planck-Landau equation is

$$\partial_t f + v\cdot\nabla_x f + F\cdot\nabla_v f = Q(f,f) \tag{1.1}$$

*Corresponding author. *Email addresses:* carrillo@maths.ox.ac.uk (J. A. Carrillo), shijin-m@sjtu.edu.cn (S. Jin), yijia_tang@sjtu.edu.cn (Y. Tang)

with the Landau collision operator

$$Q(f,f)=\nabla_v\cdot\left(\int_{\mathbb{R}^d}A(v-v_*)(f(v_*)\nabla_v f(v)-f(v)\nabla_{v_*}f(v_*))\mathrm{d}v_*\right). \tag{1.2}$$

Eq. (1.1) is a mean-field kinetic equation. The left-hand side is the Vlasov equation modeling the transport of charged particles, where F is the acceleration due to external or self-consistent forces under the effects of electrostatic and magnetic fields. The Landau collision operator $Q(f,f)$ describes the binary collisions between charged particles of single species with long-range Coulomb interactions. The collision kernel

$$A(z)=\Lambda|z|^\gamma(|z|^2I_d-z\otimes z),\quad -d-1\le\gamma\le 1,\quad \Lambda>0,\quad d\ge 2,$$

is symmetric positive semi-definite, $A(z)=A(-z)$, $\ker A(z)=\mathbb{R}z$. Similar to the Boltzmann equation, $\gamma>0$, $\gamma=0$, $\gamma<0$ represents the hard potential, Maxwell molecules and soft potential case respectively. The Coulomb potential where $d=3$, $\gamma=-3$ is of great significance since it is relevant in physical plasma applications [12].

In the numerical aspect of Eq. (1.1), the approximation of the nonlocal quadratic Landau collision operator $Q(f,f)$ is a major difficulty. Therefore, in this paper, we only focus on the spatially homogeneous Landau equation

$$\partial_t f=Q(f,f). \tag{1.3}$$

It is well-known that Eq. (1.3) has conservation of mass, momentum and energy since $\int_{\mathbb{R}^d}Q(f,f)(1,v,|v|^2)\mathrm{d}v=\mathbf{0}$. The Boltzmann entropy

$$E(f)=\int_{\mathbb{R}^d}f\log f\mathrm{d}v$$

is dissipated through

$$\frac{\mathrm{d}E}{\mathrm{d}t}=-D=-\frac{1}{2}\iint_{\mathbb{R}^{2d}}B_{v,v_*}\cdot A(v-v_*)B_{v,v_*}ff_*\mathrm{d}v\mathrm{d}v_*\le 0.$$

Here,

$$B_{v,v_*}=\nabla_v\frac{\delta E}{\delta f}-\nabla_{v_*}\frac{\delta E_*}{\delta f_*},\quad \frac{\delta E}{\delta f}=\log f+1,$$

and $f_*=f(v_*)$, $E_*=E(f_*)$ for short. Moreover, f is the equilibrium of (1.3) if and only if f is given by the Maxwellian

$$\mathcal{M}_{\rho,u,T}=\frac{\rho}{(2\pi T)^{d/2}}\exp\left(-\frac{|v-u|^2}{2T}\right) \tag{1.4}$$

with ρ (density), u (velocity), T (temperature) determined from the conserved mass, momentum and total energy defined respectively by

$$\rho=\int_{\mathbb{R}^d}f\mathrm{d}v,\quad \rho u=\int_{\mathbb{R}^d}vf\mathrm{d}v,\quad \rho u^2+\rho dT=\int_{\mathbb{R}^d}|v|^2f\mathrm{d}v.$$

The well-posedness of the homogeneous Landau equation in the hard potential [16, 17] and Maxwell molecule [42] cases are well-understood resorting to the notion of H-solution proposed by Villani [41]. In the soft potential case, there are still lots of open questions. It has been partially resolved by probabilistic techniques [21] and entropy dissipation estimate [15].

Various numerical methods have been proposed for the homogeneous Landau equation. The entropy schemes [3, 13] are widely used to provide a decay of numerical entropy. Moreover, they can preserve the conserved quantities and the stationary states are discrete Maxwellians. There are conservative implicit schemes [30] as well. To overcome the stiffness of the collision operator and capture the fluid dynamic limit, a class of asymptotic-preserving schemes [27] are introduced. Furthermore, a stochastic Galerkin method is developed to deal with the Landau equation with uncertainties [23]. However, how to solve Eq. (1.3) efficiently remains a major concern. The complexity of evaluating the quadratic collision operator $Q(f,f)$ is of $\mathcal{O}(N^2)$ where N is the number of discrete velocity points. Some fast algorithms are investigated for reducing the cost to $\mathcal{O}(N\log N)$ including the multigrid algorithm [4] and the one combined with fast multipole expansion [29]. A Fourier spectral method is developed in [39], which is $\mathcal{O}(N\log N)$ by using the fast Fourier transform thanks to the convolutional property. Recently, an Hermite spectral method [35] is presented, where surrogate collision models are used to accelerate the computation. For the nonhomogeneous case, time splitting strategies are adopted, we refer readers to [18, 20, 23, 34, 43].

Particle methods have received profound development in the past several decades, see the review paper [11] and the references therein. In the particle methods, the solution is approximated by the linear combination of Dirac delta-functions located at the particles. The particle locations and weights are evolved in time according to the ODE systems obtained from the weak formulation of the target equation. For diffusive-type equations, it seems that the existing particle methods do not keep the gradient flow structure of the equation except for the porous medium equation [38]. In order to make sense of the entropy functional and maintain the gradient flow structure of the homogeneous Landau equation at the particle level, Carrillo et al. [5] provided two kinds of regularizations of the entropy functional following [9]. Hence, the deterministic particle methods can preserve the basic properties of the Landau equation. Carrillo et al. [8] rigorously studied the gradient flow structure under a tailored metric inspired by that to the Boltzmann equation [19]. The gradient flow structures of both equations are rigorously connected through the grazing collision limit via Γ-convergence of the gradient flows [7].

Though having the good properties, the cost of the particle method in [5] is $\mathcal{O}(N^2)$. With the help of treecode summation, it can be reduced to $\mathcal{O}(N\log N)$. Motivated by the random batch method (RBM) [24], our objective is to introduce a stochastic particle method for the homogeneous Landau equation with only $\mathcal{O}(N)$ cost. RBM is devoted to simplifying the pair-wise interactions between particles. Utilizing small but randomly selected–for each time step– batch strategy, the interactions only occur inside the small batches so that the computational cost is reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ per time step. Us-

ing a random mini-batch was popular in machine learning, known as the stochastic gradient descent method [2]. The random binary collision idea was also proposed for mean-field equations of flocking and swarming [1, 6]. Due to the simplicity and scalability, it already has a variety of applications in solving mean-field Poisson-Nernst-Planck and Poisson-Boltzmann equations [33], efficient sampling [31,32], molecular dynamics [25,36] and so on, see [26] for the recent review of RBM.

In this paper, we propose random batch particle methods for the homogeneous Landau Eq. (1.3). The random batch idea can be applied to both types of regularizations of the entropy functional introduced in [5]. It is used to reduce the cost of binary collisions. In addition, taking advantage of the rapid decay property of the mollifier, we further reduce the cost of approximation to the solution and the cost of the gradient of the first variation of the regularized entropy. Hence, the final cost is $\mathcal{O}(N)$ per time step. As we shall see, the random batch particle methods can retain the conserved quantities as well.

The rest of the paper is organized as follows. In Section 2, we briefly introduce two types of the regularized homogeneous Landau equations and the corresponding particle methods they induce. In Section 3, the random batch particle methods are introduced and we also study their conservation and dissipation properties. Numerical examples are given in Section 4, which validate the efficiency and accuracy of our random batch particle methods. The paper is concluded in Section 5.

2 The regularized homogeneous Landau equations and their particle approximations

2.1 The regularized homogeneous Landau equations

In this section, we give a brief review of the regularized homogeneous Landau equations in [5]. Rewrite Eq. (1.3) as a nonlinear continuity equation

$$\partial_t f = Q(f,f) = -\nabla_v \cdot (U(f)f), \tag{2.1}$$

with velocity field

$$U(f) = -\int_{\mathbb{R}^d} A(v-v_*)\left(\nabla_v \frac{\delta E}{\delta f} - \nabla_{v_*}\frac{\delta E_*}{\delta f_*}\right) f_* \mathrm{d}v_*.$$

As mentioned in the introduction, one can easily obtain a formal gradient flow structure and generalize to the regularized equations using this form [8].

Consider the mollifier

$$\psi_\epsilon(v) = \frac{1}{(2\pi\epsilon)^{d/2}} \exp\left(-\frac{|v|^2}{2\epsilon}\right). \tag{2.2}$$

We can take into consideration two types of regularizations. Define the regularized entropy as follows:

$$\text{type I:} \qquad E_\epsilon(f)=\int_{\mathbb{R}^d}(f*\psi_\epsilon)\log(f*\psi_\epsilon)\mathrm{d}v; \tag{2.3}$$

$$\text{type II:} \qquad \tilde{E}_\epsilon(f)=\int_{\mathbb{R}^d} f\log(f*\psi_\epsilon)\mathrm{d}v. \tag{2.4}$$

Here, the operator $*$ represents for convolution. Apparently, $f*\psi_\epsilon$ is smooth. By a simple computation, one has [9]

$$\frac{\delta E_\epsilon}{\delta f}=\psi_\varepsilon*(\log(f*\psi_\epsilon)+1),\quad \nabla\frac{\delta E_\epsilon}{\delta f}=\nabla\psi_\varepsilon*\log(f*\psi_\epsilon),$$
$$\frac{\delta \tilde{E}_\epsilon}{\delta f}=\log(f*\psi_\epsilon)+\left(\frac{f}{f*\psi_\epsilon}\right)*\psi_\varepsilon,\quad \nabla\frac{\delta \tilde{E}_\epsilon}{\delta f}=\frac{f*\nabla\psi_\epsilon}{f*\psi_\epsilon}+\left(\frac{f}{f*\psi_\epsilon}\right)*\nabla\psi_\varepsilon.$$

Then the regularized homogeneous Landau equation of type I is given by

$$\partial_t f=Q_\epsilon(f,f)=-\nabla_v\cdot(U_\epsilon(f)f), \tag{2.5}$$

where

$$U_\epsilon(f)=-\int_{\mathbb{R}^d}A(v-v_*)\left(\nabla_v\frac{\delta E_\epsilon}{\delta f}-\nabla_{v_*}\frac{\delta E_{\epsilon,*}}{\delta f_*}\right)f_*\mathrm{d}v_*.$$

Eq. (2.5) satisfies the following properties [5]:

1. Conservation of mass, momentum and energy:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d}\begin{pmatrix}1\\ v\\ |v|^2\end{pmatrix}f\mathrm{d}v=\mathbf{0}. \tag{2.6}$$

2. Dissipation of entropy: let $B^\epsilon_{v,v_*}=\nabla_v\frac{\delta E_\epsilon}{\delta f}-\nabla_{v_*}\frac{\delta E_{\epsilon,*}}{\delta f_*}$, then

$$\frac{\mathrm{d}E_\epsilon}{\mathrm{d}t}=-D_\epsilon=-\frac{1}{2}\iint_{\mathbb{R}^{2d}}B^\epsilon_{v,v_*}\cdot A(v-v_*)B^\epsilon_{v,v_*}ff_*\mathrm{d}v\mathrm{d}v_*\le 0. \tag{2.7}$$

3. The stationary state of (2.5) is characterized by a Maxwellian.

Similarly, the regularized homogeneous Landau equation of type II reads

$$\partial_t f=\tilde{Q}_\epsilon(f,f)=-\nabla_v\cdot(\tilde{U}_\epsilon(f)f), \tag{2.8}$$

with

$$\tilde{U}_\epsilon(f)=-\int_{\mathbb{R}^d}A(v-v_*)\left(\nabla_v\frac{\delta \tilde{E}_\epsilon}{\delta f}-\nabla_{v_*}\frac{\delta \tilde{E}_{\epsilon,*}}{\delta f_*}\right)f_*\mathrm{d}v_*.$$

Eq. (2.8) also shares the aforementioned conservation (2.6) and entropy dissipation (2.7) with E_ϵ being replaced by $\tilde{E}_\epsilon$.

2.2 Deterministic particle methods

In order to numerically approximate the Landau equation, Carrillo et al. [5] derived two deterministic particle methods based on the gradient flow structure of the regularized Landau equations (2.5) and (2.8). The particle methods can preserve the basic conservation and entropy decay properties of the Landau operator $Q(f,f)$. Now, we briefly introduce these methods.

2.2.1 Type I method

Approximate f by the N particle formulation

$$f^N(t,v)=\sum_{i=1}^{N} w_i\delta(v-v_i(t)). \tag{2.9}$$

Here, $v_i(t)$ and w_i are the velocity and weight of particle i, N is the total number of particles. Then, the blob solution [9] can be constructed

$$\tilde{f}^N(t,v):=f^N*\psi(t,v)=\sum_{i=1}^{N} w_i\psi_\epsilon(v-v_i(t))$$

so as to visualize the particle solution.

Plugging (2.9) into (2.5), one can obtain the evolution for the particle velocity

$$\frac{\mathrm{d}v_i(t)}{\mathrm{d}t}=U_\epsilon(f^N)(v_i(t))=-\sum_{j=1}^{N} w_j A(v_i-v_j)\left[\nabla\frac{\delta E_\epsilon^N}{\delta f}(v_i)-\nabla\frac{\delta E_\epsilon^N}{\delta f}(v_j)\right], \tag{2.10}$$

where

$$\nabla\frac{\delta E_\epsilon^N}{\delta f}(v)=\int_{\mathbb{R}^d}\nabla\psi_\epsilon(v-u)\log\tilde{f}^N(u)\mathrm{d}u. \tag{2.11}$$

Furthermore, truncate the whole velocity space by a computational domain $\Omega=[-L,L]^d$ with L large enough. Applying the second order composite mid-point quadrature rule to approximate the velocity integral in (2.11) yields

$$\nabla\frac{\delta \bar{E}_\epsilon^N}{\delta f}(v):=\sum_{l=1}^{N} h^d\nabla\psi_\epsilon(v-v_l^c)\log\tilde{f}^N(v_l^c)=\nabla\frac{\delta E_\epsilon^N}{\delta f}(v)+\mathcal{O}(h^2), \tag{2.12}$$

where the mesh size is $h=2L/n_o$ with $n_o=N^{1/d}$ being the number of particles per dimension, the quadrature nodes v_l^c, $l=1,\cdots,N$ are the centers of square mesh. Eq. (2.10) guarantees the conservation of discrete mass, momentum and energy exactly, while the discrete entropy dissipates exactly with (2.11) or up to $\mathcal{O}(h^2)$ with (2.12) [5].

2.2.2 Type II method

Analogous to type I, the evolution for the particle velocity of type II is

$$\frac{\mathrm{d}v_i}{\mathrm{d}t}=\tilde{U}_\epsilon(f^N)(v_i(t))=-\sum_{j=1}^N w_j A(v_i-v_j)\left[\nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v_i)-\nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v_j)\right], \tag{2.13}$$

where

$$\nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v)=\sum_{k=1}^N w_k\frac{\nabla\psi_\epsilon(v-v_k)}{\tilde{f}^N(v)}+\sum_{k=1}^N w_k\frac{\nabla\psi_\epsilon(v-v_k)}{\tilde{f}^N(v_k)}.$$

Since there is no convolution outside logarithmic term in the regularized entropy (2.4), this type of regularization is free of numerical quadrature in velocity. The discrete mass, momentum and energy of (2.13) are conserved, while the discrete entropy is dissipated exactly too [5].

3 Random batch particle methods for the regularized homogeneous Landau equation

In either (2.10) or (2.13), one needs to sum over all the particles v_j to evolve particle v_i, which leads to an $\mathcal{O}(N^2)$ computational cost for each time step. To significantly reduce the cost, we apply the random batch method [24] to this summation. At each time step, the N particles are randomly divided into $q=N/p$ batches $\mathcal{C}_v$, $v=1,\cdots,q$ with batch size $p\ll N$. Then particle v_i will update itself only with those particles in the same batch.

3.1 Type I RBM

Mathematically, Eq. (2.10) is changed to

$$\begin{aligned}\frac{\mathrm{d}v_i(t)}{\mathrm{d}t}&=U_\epsilon^*(f^N)(v_i(t))\\&=-\frac{N-1}{p-1}\sum_{j\neq i,j\in\mathcal{C}_v} w_j A(v_i-v_j)\left[\nabla\frac{\delta E_\epsilon^N}{\delta f}(v_i)-\nabla\frac{\delta E_\epsilon^N}{\delta f}(v_j)\right],\quad i\in\mathcal{C}_v.\end{aligned} \tag{3.1}$$

Since the interactions only take place with the batch of p particles, the computational cost is $\mathcal{O}(p^2\cdot q)=\mathcal{O}(pN)$ per time step.

Here, we explain why RBM works. Define the fluctuation of the random force on v_i by

$$\chi_i=U_\epsilon^*(f^N)-U_\epsilon(f^N).$$

Similar to [24, Lemma 3.1], we have

$$\mathbb{E}(\chi_i)=0,\quad \mathrm{Var}(\chi_i)=\mathbb{E}|\chi_i|^2=\left(\frac{1}{p-1}-\frac{1}{N-1}\right)\Lambda_i, \tag{3.2}$$

where

$$\Lambda_i = \frac{1}{N-2}\sum_{j:j\neq i}\left|(N-1)w_j A(v_i-v_j)B_{ij}^N - \sum_{k:k\neq i} w_k A(v_i-v_k)B_{ik}^N\right|^2,$$

with $B_{ij}^N = \nabla\frac{\delta E_\epsilon^N}{\delta f}(v_i) - \nabla\frac{\delta E_\epsilon^N}{\delta f}(v_j)$. The expectation is taken over random divisions. $\mathbb{E}(\chi_i)=0$ indicates that the random force $U_\epsilon^*(f^N)$ is unbiased. Λ_i is independent of the batch size p, so the variance is smaller for larger p. Since the cross-batch interactions are neglected, the approximation error is $\mathcal{O}(1)$ for a single time step. We do random reshuffling at each time step so that any two particles will have the chance to interact with each other. As the random force is unbiased, the random errors will roughly cancel out over time. It means that RBM works due to the "law of large numbers" type mechanism in time. In other words, RBM can be regarded as sampling in time with the Monte Carlo rate $\sqrt{\text{Variance}\cdot\Delta t}\sim\sqrt{\Delta t/p}$. The convergence of RBM has already been proved in [24,31,32] for regular forces. Moreover, the error bound is uniform in N so that one can choose the time step Δt and the batch size p independent of N, see [24].

Next, we show that the random batch particle method (3.1) retains the basic conservation and entropy decay properties. Define the indicator function

$$I_{ij} = \begin{cases} 1, & i,j \text{ in the same batch,} \\ 0, & i,j \text{ not in the same batch.} \end{cases}$$

Then, (3.1) can be rewritten as

$$\frac{\mathrm{d}v_i(t)}{\mathrm{d}t} = U_\epsilon^*(f^N)(v_i(t)) = -\frac{N-1}{p-1}\sum_{j=1}^N w_j I_{ij} A(v_i-v_j)B_{ij}^N. \tag{3.3}$$

The next proposition shows that (3.3) inherits the desired properties due to the symmetry of I_{ij}.

Proposition 3.1. The semi-discrete random batch particle method of type I (3.3) satisfies the following properties:

1. Conservation of mass, momentum and energy:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=1}^N w_i\phi(v_i)=0 \quad \text{for } \phi(v)=1,v,|v|^2.$$

2. Dissipation of entropy: $\frac{\mathrm{d}E_\epsilon^N}{\mathrm{d}t} = -D_\epsilon^* \leq 0$, where

$$D_\epsilon^* = \frac{N-1}{2(p-1)}\sum_{i=1}^N\sum_{j=1}^N w_i w_j I_{ij} B_{ij}^N \cdot A(v_i-v_j)B_{ij}^N.$$

Proof. 1. In fact,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=1}^N w_i\phi(v_i) &= \sum_{i=1}^N w_i\nabla\phi(v_i)\cdot U_\epsilon^*(f^N)(v_i)\\ &= -\frac{N-1}{p-1}\sum_{i=1}^N\sum_{j=1}^N w_i w_j I_{ij}\nabla\phi(v_i)\cdot A(v_i-v_j)B_{ij}^N\\ &= -\frac{N-1}{2(p-1)}\sum_{i=1}^N\sum_{j=1}^N w_i w_j I_{ij}\left(\nabla\phi(v_i)-\nabla\phi(v_j)\right)\cdot A(v_i-v_j)B_{ij}^N,\end{aligned}$$

which vanishes with $\phi(v)=1,v,|v|^2$ since $v\in\ker A(v)$.

2.

$$\begin{aligned}\frac{\mathrm{d}E_\epsilon^N}{\mathrm{d}t} &= \int_{\mathbb{R}^d}\sum_{i=1}^N w_i\nabla\psi_\epsilon(v-v_i)\cdot\left(-\frac{\mathrm{d}v_i(t)}{\mathrm{d}t}\right)\log\tilde{f}^N(v)\mathrm{d}v\\ &\quad+\int_{\mathbb{R}^d}\tilde{f}^N(v)\frac{\sum_{k=1}^N w_k\nabla\psi_\epsilon(v-v_k)\cdot\left(-\frac{\mathrm{d}v_k(t)}{\mathrm{d}t}\right)}{\tilde{f}^N(v)}\mathrm{d}v=:I_1+I_2.\end{aligned}$$

Note that

$$I_2=-\sum_{k=1}^N\int_{\mathbb{R}^d}w_k\nabla\psi_\epsilon(v-v_k)\cdot\frac{\mathrm{d}v_k(t)}{\mathrm{d}t}\mathrm{d}v=\frac{\mathrm{d}}{\mathrm{d}t}\sum_{k=1}^N w_k\int_{\mathbb{R}^d}\psi_\epsilon(v-v_k)\mathrm{d}v=0,$$

since $\int_{\mathbb{R}^d}\psi_\epsilon(v-v_k)\mathrm{d}v=1$. Besides,

$$\begin{aligned}I_1 &= \sum_{i=1}^N w_i\left(\int_{\mathbb{R}^d}-\nabla\psi_\epsilon(v-v_i)\log\tilde{f}^N(v)\mathrm{d}v\right)\cdot\frac{\mathrm{d}v_i}{\mathrm{d}t}=\sum_{i=1}^N w_i\nabla\frac{\delta E_\epsilon^N}{\delta f}(v_i)\cdot\frac{\mathrm{d}v_i}{\mathrm{d}t}\\ &= -\frac{N-1}{p-1}\sum_{i=1}^N w_i\nabla\frac{\delta E_\epsilon^N}{\delta f}(v_i)\cdot\sum_{j=1}^N w_j I_{ij}A(v_i-v_j)B_{ij}^N\\ &= -\frac{N-1}{2(p-1)}\sum_{i=1}^N\sum_{j=1}^N w_i w_j I_{ij}B_{ij}^N\cdot A(v_i-v_j)B_{ij}^N\\ &= -D_\epsilon^*\le 0.\end{aligned}$$

This completes the proof of entropy dissipation. □

Furthermore, approximating $\nabla\frac{\delta E_\epsilon^N}{\delta f}$ by $\nabla\frac{\delta\bar{E}_\epsilon^N}{\delta f}$, we can obtain the random batch version of the discrete-in-velocity particle method

$$\frac{\mathrm{d}v_i}{\mathrm{d}t}=\bar{U}_\epsilon^*(f^N)(v_i)=-\frac{N-1}{p-1}\sum_{j=1}^N w_j I_{ij}A(v_i-v_j)\bar{B}_{ij}^N, \tag{3.4}$$

where $\bar{B}_{ij}^N=\nabla\frac{\delta\bar{E}_\epsilon^N}{\delta f}(v_i)-\nabla\frac{\delta\bar{E}_\epsilon^N}{\delta f}(v_j)=B_{ij}^N+\mathcal{O}(h^2)$.

Proposition 3.2. (3.4) inherits the following properties:

1. Conservation of mass, momentum and energy:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=1}^{N} w_i \begin{pmatrix} 1 \\ v_i \\ |v_i|^2 \end{pmatrix} = \mathbf{0}.$$

2. Dissipation of entropy up to $\mathcal{O}(h^2)$:

$$\bar{E}_\epsilon^N(t) - \bar{E}_\epsilon^N(0) = -\int_0^t \bar{D}_\epsilon^* \mathrm{d}s + \mathcal{O}(h^2),$$

where

$$\bar{D}_\epsilon^* = \frac{N-1}{2(p-1)} \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j I_{ij} \bar{B}_{ij}^N \cdot A(v_i - v_j)\bar{B}_{ij}^N \geq 0.$$

Proof. The proof of conservation is the same as that in Proposition 3.1. The proof of entropy decay is the discrete version of that in Proposition 3.1, while the $\mathcal{O}(h^2)$ error is brought by the mid-point composite quadrature rule in I_2. To be specific,

$$I_2 = -\sum_{k=1}^{N} w_k \sum_{l=1}^{N} h^d \nabla\psi_\epsilon(v_l^c - v_k)\cdot\frac{\mathrm{d}v_k}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\sum_{k=1}^{N} w_k \sum_{l=1}^{N} h^d \psi_\epsilon(v_l^c - v_k) = \frac{\mathrm{d}}{\mathrm{d}t}\sum_{k=1}^{N} w_k(1+\mathcal{O}(h^2)).$$

□

3.2 Type II RBM

Like in type I, in order to update the particle velocity v_i, one needs to sum over all the particles v_j, which is time-consuming. So we can adopt the random batch idea to reduce the cost of (2.13) from $\mathcal{O}(N^2)$ to $\mathcal{O}(pN)$ per time step. That is, at each time step, randomly divide the N particles into $q = N/p$ batches $\mathcal{C}_v$. For each batch $\mathcal{C}_v$, $v = 1,\cdots,q$, evolve the particles in $\mathcal{C}_v$ through

$$\begin{aligned}\frac{\mathrm{d}v_i}{\mathrm{d}t} &= \tilde{U}_\epsilon^*(f^N)(v_i) \\ &= -\frac{N-1}{p-1}\sum_{j\neq i, j\in\mathcal{C}_v} w_j A(v_i - v_j)\left[\nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v_i) - \nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v_j)\right] \\ &= -\frac{N-1}{p-1}\sum_{j=1}^{N} w_j I_{ij} A(v_i - v_j)\tilde{B}_{ij}^N, \end{aligned} \tag{3.5}$$

with $\tilde{B}_{ij}^N = \nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v_i) - \nabla\frac{\delta\tilde{E}_\epsilon^N}{\delta f}(v_j)$.

Define

$$\tilde{\chi}_i = \tilde{U}^*_\epsilon(f^N) - \tilde{U}_\epsilon(f^N).$$

Again, we have

$$\mathbb{E}(\tilde{\chi}_i) = 0.$$

Hence, the previous Monte Carlo interpretation still holds. Since the batches will be chosen independently at each time step, the random errors will cancel out due to the unbiased force approximation. The time averaging effect could ensure the convergence of RBM with $\sqrt{\text{Variance}\cdot\Delta t}$ error. See [24].

Similarly, the random batch particle method of regularization type II (3.5) inherits the desired properties due to the symmetry of I_{ij}, i.e., Proposition 3.1 holds correspondingly.

3.3 Algorithms

A variety of time discretizations can be applied to solve the particle systems (2.10), (2.13) or the random batch particle systems (3.4), (3.5). High order explicit schemes such as the Runge-Kutta methods or multistep methods are expected to achieve high accuracy. In this paper, for simplicity, we choose the first order forward Euler in time, which is enough to compare the random batch and original particle methods. Indeed, it is able to see the reduction of computational time with the lowest order in time. On the other hand, in numerical examples, instead of considering convergence in time step, we only take into account the convergence in particle number.

Since the structures of the particle methods are almost the same, we summarize them into a general algorithm, see Algorithm I. And the details in each particle method are listed in Tables 1-2. They are renamed as Algorithms 1 to 4 correspondingly.

Algorithm I The framework of particle method for the homogeneous Landau equation

Input Number of particles per dimension n_o, truncated length $L>0$ ($N=n_o^d$, $h=2L/n_o$, $\Omega=[-L,L]^d$). Start time t_0 and terminal time t_{end}, time step Δt. Regularizing parameter ϵ. Closeness parameter σ and batch size $p \ll N$ (for random batch cases).

Initialization Particles v_i^0 and corresponding weights w_i, $k=1,\cdots,N$.

At each time step $n=0,1,\cdots,\lfloor\frac{t_{end}-t_0}{\Delta t}\rfloor$, do the following:

1: Compute the blob solution $\tilde{f}^N$.
2: Compute the gradient of the first variation of the regularized entropy.
3: Update the particle velocities.

Output Particle velocities at t_{end}.

For Algorithms 1-2, direct computations of each step require $\mathcal{O}(N^2)$ cost per time step due to summation of N velocity points, as listed in the last column in Table 1. To save the cost, we apply RBM to the summation of step 3. As stated in Section 3.1, the cost

Table 1: The three steps in the loops of deterministic particle methods.

method	step	cost
Type I method (Algorithm 1)	1. $f_l^c = \sum_{k=1}^N w_k \psi_\epsilon(v_l^c - v_k^n), \quad \forall\, l$	$\mathcal{O}(N^2)$
	2. $F_i = \sum_{l=1}^N h^d \nabla \psi_\epsilon(v_i^n - v_l^c) \log f_l^c, \quad \forall\, i$	$\mathcal{O}(N^2)$
	3. $\frac{v_i^{n+1} - v_i^n}{\Delta t} = -\sum_{j=1}^N w_j A(v_i^n - v_j^n)(F_i - F_j), \quad \forall\, i$	$\mathcal{O}(N^2)$
Type II method (Algorithm 2)	1. $f_i = \sum_{k=1}^N w_k \psi_\epsilon(v_i^n - v_k^n), \quad \forall\, i$	$\mathcal{O}(N^2)$
	2. $F_i = \sum_{k=1}^N w_k \nabla \psi_\epsilon(v_i^n - v_k^n)\left(\frac{1}{f_i} + \frac{1}{f_k}\right), \quad \forall\, i$	$\mathcal{O}(N^2)$
	3. $\frac{v_i^{n+1} - v_i^n}{\Delta t} = -\sum_{j=1}^N w_j A(v_i^n - v_j^n)(F_i - F_j), \quad \forall\, i$	$\mathcal{O}(N^2)$

Table 2: The three steps in the loops of random batch particle methods.

method	step	cost
Type I RBM (Algorithm 3)	1. $f_l^c = \sum_{k \in \mathcal{A}_l} w_k \psi_\epsilon(v_l^c - v_k^n)$, $\mathcal{A}_l = \{k : \lvert v_l^c - v_k^n \rvert \le \sigma\}, \quad \forall\, l$	$\mathcal{O}(N)$
	2. $F_i = \sum_{l \in \mathcal{B}_i} h^d \nabla \psi_\epsilon(v_i^n - v_l^c) \log f_l^c$, $\mathcal{B}_i = \{l : \lvert v_i^n - v_l^c \rvert \le \sigma\}, \quad \forall i$	$\mathcal{O}(N)$
	3. $\frac{v_i^{n+1} - v_i^n}{\Delta t} = -\frac{N-1}{p-1} \sum_{j \in \mathcal{C}_v} w_j A(v_i^n - v_j^n)(F_i - F_j), \quad i \in \mathcal{C}_v$	$\mathcal{O}(pN)$
Type II RBM (Algorithm 4)	1. $f_i = \sum_{k \in \mathcal{A}_i} w_k \psi_\epsilon(v_i^n - v_k^n)$, $\mathcal{A}_i = \{k : \lvert v_i^n - v_k^n \rvert \le \sigma\}, \quad \forall\, i$	$\mathcal{O}(N)$
	2. $F_i = \sum_{k \in \mathcal{A}_i} w_k \nabla \psi_\epsilon(v_i^n - v_k^n)\left(\frac{1}{f_i} + \frac{1}{f_k}\right), \quad \forall\, i$	$\mathcal{O}(N)$
	3. $\frac{v_i^{n+1} - v_i^n}{\Delta t} = -\frac{N-1}{p-1} \sum_{j \in \mathcal{C}_v} w_j A(v_i^n - v_j^n)(F_i - F_j), \quad i \in \mathcal{C}_v$	$\mathcal{O}(pN)$

is $\mathcal{O}(pN)$. The error bound of RBM is uniform in N so that we can choose batch size $p = \mathcal{O}(1)$ independent of N. Besides, we make use of the rapid decay property of the mollifier ψ_ϵ to reduce the summation in steps 1 and 2. Denote σ as the velocity distance to be considered in the summation of steps 1 and 2. Then, the blob solution and the gradient of the first variation are computed using only the velocity points within σ to the target velocity point. The computational cost is again $\mathcal{O}(N)$ per time step resorting to the cell list data structure [22, Appendix F]. These give rise to Algorithms 3-4, whose detailed steps are in Table 2. In Table 2, the sets $\mathcal{A}_l$,$\mathcal{B}_i$,$\mathcal{A}_i$ are the indices of neighbors within distance σ, while $\mathcal{C}_v$ are the random batches. If we choose $p \ll N$, the overall cost of Algorithms 3-4 will scale like $\mathcal{O}(N)$.

Due to first order Euler in time, we have

Proposition 3.3. The discrete mass and momentum of the four algorithms are conserved exactly, while the discrete energy is only conserved up to $\mathcal{O}(\Delta t)$. Namely,

$$\sum_i w_i = const, \quad \frac{\sum_i w_i v_i^{n+1} - \sum_i w_i v_i^n}{\Delta t} = \mathbf{0}, \quad \frac{\sum_i w_i \left|v_i^{n+1}\right|^2 - \sum_i w_i \left|v_i^n\right|^2}{\Delta t} = \mathcal{O}(\Delta t).$$

Proof. The mass conservation holds automatically. Denote

$$R(v_i) = -\frac{N-1}{p-1}\sum_j w_j I_{ij} A(v_i - v_j)(F_i - F_j),$$

where $p = N$, $I_{ij} \equiv 1$ for Algorithms 1-2. The Euler scheme is $v_i^{n+1} = v_i^n + \Delta t R(v_i^n)$. Since

$$A(v) = A(-v), \quad \ker A(v) = \mathbb{R}v, \quad I_{ij} = I_{ji},$$

we have

$$\sum_i w_i R(v_i) = \mathbf{0}, \quad \sum_i w_i v_i \cdot R(v_i) = 0.$$

Hence,

$$\begin{aligned}
&\sum_i w_i v_i^{n+1} = \sum_i w_i v_i^n, \\
&\sum_i w_i \left|v_i^{n+1}\right|^2 = \sum_i w_i |v_i^n|^2 + \Delta t^2 \sum_i w_i |R(v_i^n)|^2.
\end{aligned}$$

This implies that the discrete momentum is conserved and the discrete energy is conserved up to $\mathcal{O}(\Delta t)$. □

Remark 3.1. Higher order time discretizations lead to higher order results in energy conservation. Take the improved Euler method

$$\begin{cases}
v_i^* = v_i^n + \Delta t R(v_i^n), \\
v_i^{n+1} = \frac{1}{2} v_i^n + \frac{1}{2} v_i^* + \frac{1}{2} \Delta t R(v_i^*),
\end{cases}$$

as an example. It follows from

$$\sum_i w_i \left|v_i^{n+1}\right|^2 = \sum_i w_i |v_i^n|^2 + \frac{1}{4}\Delta t^2 \sum_i w_i |R(v_i^*) - R(v_i^n)|^2$$

that the discrete energy is conserved up to $\mathcal{O}(\Delta t^3)$.

4 Numerical experiments

In this section, we compare the particle methods in [5] and our random batch particle methods for the two types of regularizations for the homogeneous Landau equation. We give four classical numerical examples. It shows that our random batch particle methods have almost second order accuracy as the deterministic particle methods while the cost can be remarkably reduced to $\mathcal{O}(N)$.

Example 4.1 (2D BKW solution for Maxwell molecules)**.** Consider the case when $d=2$, $\gamma=0$, $A(z)=\frac{1}{16}(|z|^2 I_d - z\otimes z)$. Requiring the macroscopic quantities to be $\rho=1$, $u=\mathbf{0}$, $T=1$ (thus $\int_{\mathbb{R}^d} f|v|^2 \mathrm{d}v = d = 2$), the exact solution of the homogeneous Landau equation is given by

$$f^{ext}(t,v)=\frac{1}{2\pi K}\exp\left(-\frac{|v|^2}{2K}\right)\left(\frac{2K-1}{K}+\frac{1-K}{2K^2}|v|^2\right),\quad K=1-\frac{\exp(-t/8)}{2}.$$

Let $t_0=0$, $t_{end}=5$. The real distributions in $[-4,4]^2$ are shown in Fig. 1.

First of all, we test the convergence and the computational cost with respect to particle numbers. The number of particles per dimension is chosen as $n_o=40,60,80,100,120,140,160,180,200,220,240$ respectively, the total particle number is $N=n_o^2$. Set $L=8$ for Algorithms 1 and 3, $L=10$ for Algorithms 2 and 4. The truncated length L is chosen such that the particles do not escape from the computational velocity domain $\Omega=[-L,L]^2$ during the time span. We initialize the particle velocities uniformly in the support $[-4,4]^2$, the weights are given according to $f^{ext}(0,v)$. Let $\Delta t=0.01$, the default regularization parameter is $\epsilon=0.64h^{1.98}$ with $h=2L/n_o$ as in [5]. Besides, in the random batch particle methods (Algorithms 3 and 4), the closeness parameter is $\sigma=4\sqrt{\epsilon}$, the number of batches per dimension $q_o=n_o/p_o$ is fixed as 5. Measured by the relative L_2 norm between the exact solution and the blob solution, the error is defined as

$$\frac{\sqrt{\sum_{l=1}^N h^d \left|f^{ext}(t,v_l^c)-\tilde{f}^N(t,v_l^c)\right|^2}}{\sqrt{\sum_{l=1}^N h^d \left|f^{ext}(t,v_l^c)\right|^2}}.$$

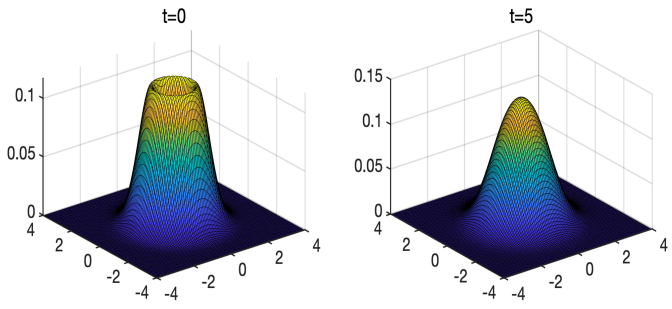

Figure 1: Left: initial distribution. Right: exact distribution at t_{end}.

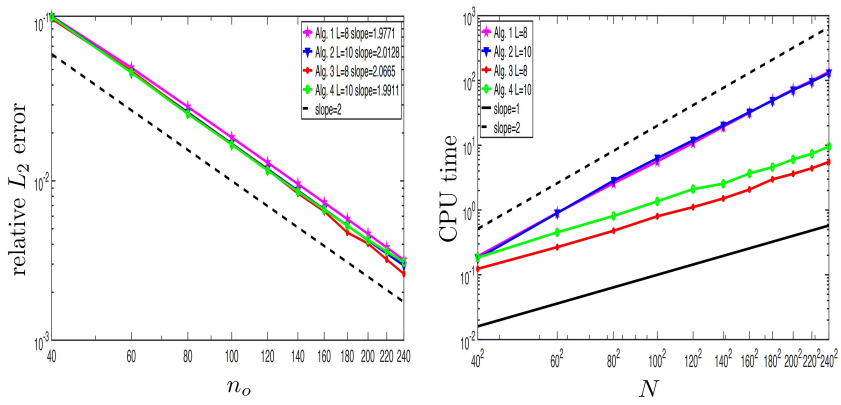

Figure 2: Convergence result for Algorithms 1 and 3 when $L=8$ and Algorithms 2 and 4 when $L=10$. Left: relative L_2 error at $t_{end}=5$ with respect to particle number per dimension n_o; Right: CPU time (in seconds) per time step with respect to particle number N.

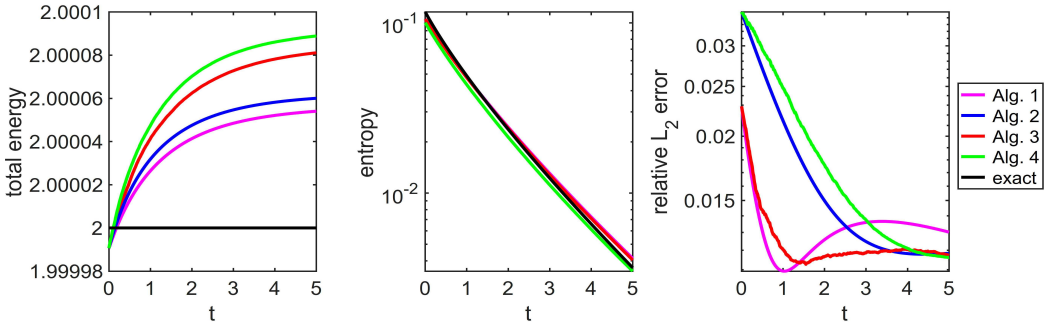

Figure 3: Time evolution of the total energy (left), entropy (middle) and the relative L_2 error (right) for Algorithms 1-4 when $n_o=120$.

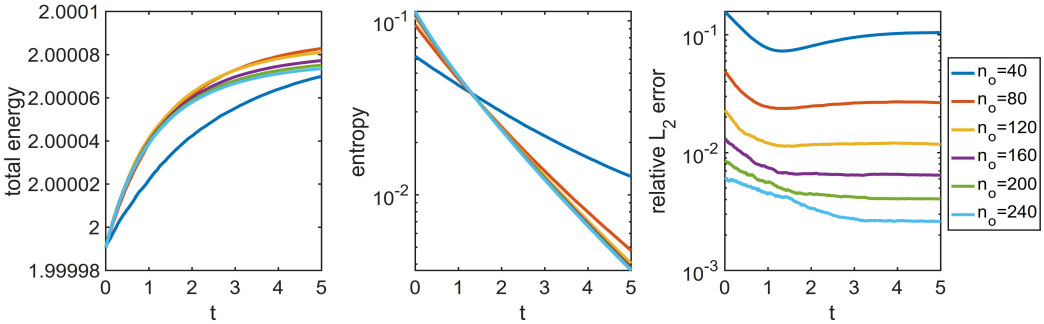

Figure 4: Time evolution of the total energy (left), entropy (middle) and the relative L_2 error (right) with respect to n_o for Algorithm 3.

It is clear from Fig. 2 (left) that all the four algorithms have second order decay in the relative L_2 error in n_o. As for the computational cost, we can conclude from Fig. 2 (right) that the costs for the original particle methods (Algorithms 1-2) are a bit lower than $\mathcal{O}(N^2)$, while the costs for the random batch particle methods (Algorithms 3-4) are $\mathcal{O}(N)$.

Next, we check the conservation and entropy dissipation properties. The total mass, determined by the initialization, remains unchanged during the evolution. The momentum is always in the order of $1e-16$, which means it is conserved within machine precision. These coincide with Proposition 3.3 and we will not present them any more. In Fig. 3, we show the time evolution for the four algorithms with fixed n_o. It is clear that the total energy is conserved up to a small error, the entropy is dissipated and the relative L_2 error is stable. In Fig. 4, we plot the time evolution for Algorithm 3 with regard to different particle numbers. Similar performance can be observed for the other algorithms.

This example verifies that the random batch particle methods can preserve the conservation and entropy dissipation properties of the homogeneous Landau equation as expected. They have second order accuracy like the deterministic particle methods, while benefit from the $\mathcal{O}(N)$ cost.

Example 4.2 (3D BKW solution for Maxwell molecules)**.** Consider the case when $d=3$, $\gamma=0$, $A(z)=\frac{1}{24}(|z|^2 I_d - z\otimes z)$. Requiring the macroscopic quantities to be $\rho=1$, $u=\mathbf{0}$, $T=1$ (thus $\int_{\mathbb{R}^d} f|v|^2 \mathrm{d}v = d = 3$), the exact solution of the homogeneous Landau equation is given by

$$f^{ext}(t,v)=\frac{1}{(2\pi K)^{3/2}}\exp\left(-\frac{|v|^2}{2K}\right)\left(\frac{5K-3}{2K}+\frac{1-K}{2K^2}|v|^2\right),\quad K=1-\exp(-t/6).$$

Let $t_0=5.5$, $t_{end}=6$. As in Example 4.1, we test the convergence and the computational cost. The number of particles per dimension is chosen as $n_o=30,40,50,60,70,80,90$ respectively, then the total particle number is $N=n_o^3$. Initialize the particles uniformly in $[-4,4]^3$ and take $L=8$ to ensure that the particles always stay in Ω. Take $\Delta t=0.01$, $h=2L/n_o$, $\epsilon=0.64h^{1.98}$, $\sigma=4\sqrt{\epsilon}$. For the random batch cases, choose the number of batches per dimension to be $q_o=2$ or 5, correspondingly, the batch size $p=(n_o/q_o)^d$ is $N/8$ or $N/125$. The results are given in Fig. 5. From Fig. 5 (left), one can observe roughly second order convergence rate in n_o and the orders are lower for regularization type II (Algorithms 2 and 4) compared to regularization type I (Algorithms 1 and 3). For the random batch cases (Algorithms 3-4), the errors for larger q_o (red and green dashed lines) are bigger than those for smaller q_o (red and green solid lines), since the noise level of RBM is lower for smaller p (larger q_o). From Fig. 5 (right), we see the costs for the original particle methods (magenta and blue solid lines) are a bit lower than $\mathcal{O}(N^2)$. Meanwhile, the random batch methods with $q_o=2$ (red and green solid lines) are also very expensive since $p=N/8$ does not satisfy $p\ll N$, so pN scales like N^2. While for $q_o=5$ (red and green dashed lines), i.e., $p=N/125$, the cost is $\mathcal{O}(N)$. This comparison tells that the choice of batch size p is a compromise between complexity and accuracy, one would sacrifice somewhat from accuracy for efficiency.

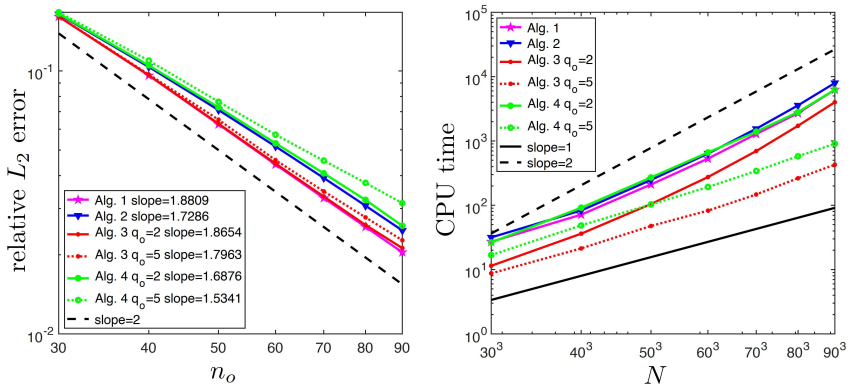

Figure 5: Convergence results for Algorithms 1-4 when $L=8$. Left: relative L_2 error at $t_{end}=6$ with respect to particle number per dimension n_o; Right: CPU time (in seconds) per time step with respect to particle number N.

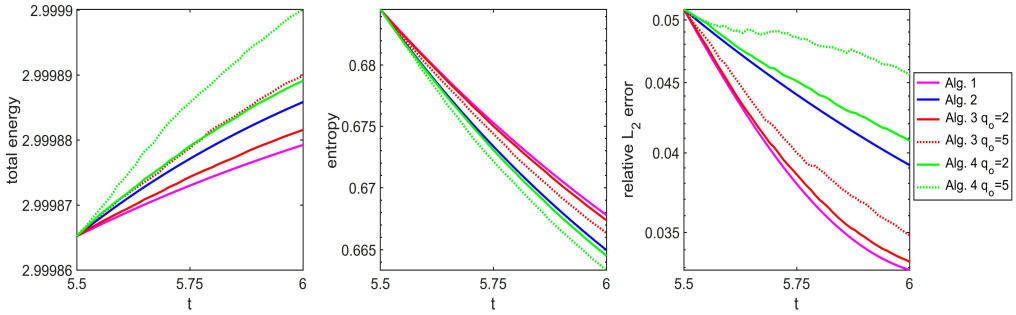

Figure 6: Time evolution of the total energy (left), entropy (middle) and the relative L_2 error (right) for Algorithms 1-4 when $n_o=70$.

Next, we show the performance of the four algorithms. Fig. 6 shows the conserved total energy, the dissipated entropy and the relative L_2 error. The behavior is similar to that in Example 4.1.

Fig. 7 depicts the error to snapshots of the solutions $f(:,\frac{n_o}{2},\frac{n_o}{2})$ at $t=5.5,5.75,6$ respectively when $n_o=70$. The curves in the left subfigure are overlapped due to the same initialization. And the error is smaller for larger p (smaller q_o), see the middle and right subfigures. We also plot snapshots of the solutions versus different n_o at $t=5.5,5.75,6$ in Fig. 8 computed by Algorithm 3 with $q_o=5$, where we can observe a better match as n_o increases.

We can conclude from the example that the random batch particle methods are applicable to high dimensional problems. It is very efficient if the required precision is not high.

Example 4.3 (2D anisotropic solution with Coulomb potential)**.** Consider the case when $d=2$, $\gamma=-3$. The collision kernel is $A(z)=\frac{1}{16}\frac{1}{|z|^3}(|z|^2 I_d - z\otimes z)$. The initial condition is

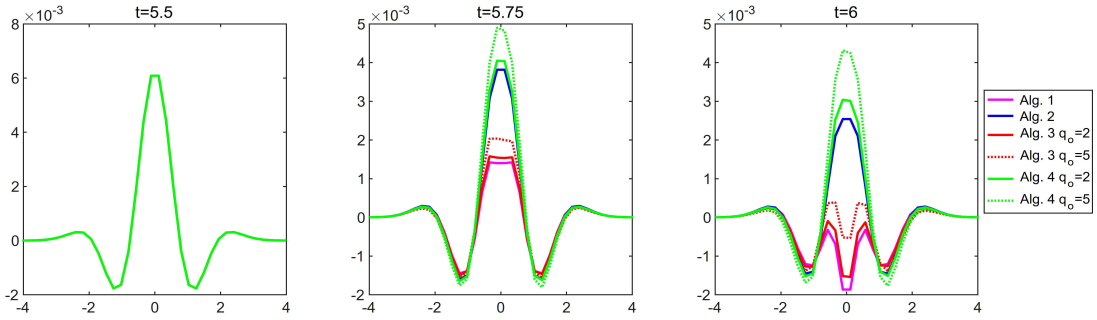

Figure 7: Error profiles at $t=5.5$ (left), $t=5.75$ (middle) and $t=6$ (right) for Algorithms 1-4 when $n_o=70$.

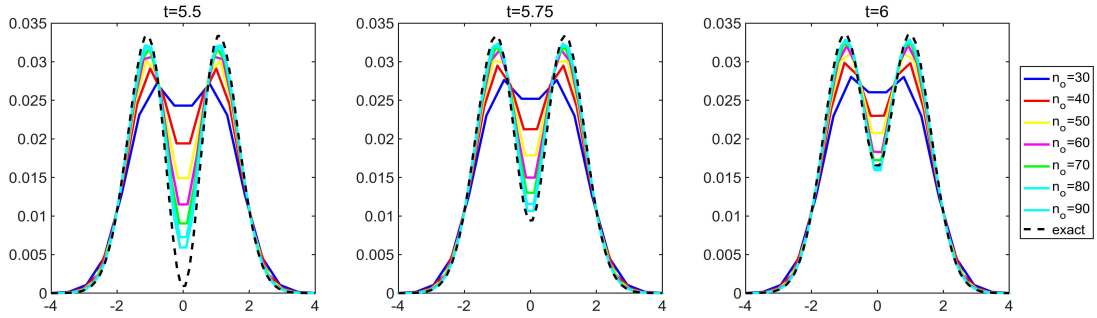

Figure 8: Snapshots $f(:,\frac{n_o}{2},\frac{n_o}{2})$ at $t=5.5$ (left), $t=5.75$ (middle) and $t=6$ (right) for Algorithm 3 with $q_o=5$ with respect to different n_o.

chosen to be bi-Maxwellian

$$f(0,v)=\frac{1}{4\pi}\left\{\exp\left(-\frac{(v-u_1)^2}{2}\right)+\exp\left(-\frac{(v-u_2)^2}{2}\right)\right\},\quad u_1=(-2,1),\quad u_2=(0,-1).$$

For this example, we do not have the exact solution. Therefore, we use $n_o=200$ as the reference solution and test $n_o=40,60,80,100,120,140,160,180$ for the four algorithms. Let $t_0=0$, $t_{end}=20$, $\Delta t=0.1$, $L=10$, $q_o=5$, the results are shown in Fig. 9. Once again, the convergence rate is second order, the cost for the original and the random batch particle methods is $\mathcal{O}(N^2)$ and $\mathcal{O}(N)$ respectively. Fig. 10 shows the error of snapshots $f(:,\frac{n_o}{2})$ and $f(\frac{n_o}{2},:)$ with different n_o, where the magnitude of error profile decreases as n_o increases.

This example shows that the random batch particle methods can be applied to the case where the collision kernel is singular.

Example 4.4 (3D Rosenbluth problem with Coulomb potential)**.** Consider the case when $d=3$, $\gamma=-3$. The collision kernel is $A(z)=\frac{1}{4\pi}\frac{1}{|z|^3}(|z|^2I_d-z\otimes z)$. The initial condition is given by

$$f(0,v)=\frac{1}{S^2}\exp\left(-S\frac{(|v|-\mu)^2}{\mu^2}\right),\quad \mu=0.3,\quad S=10.$$

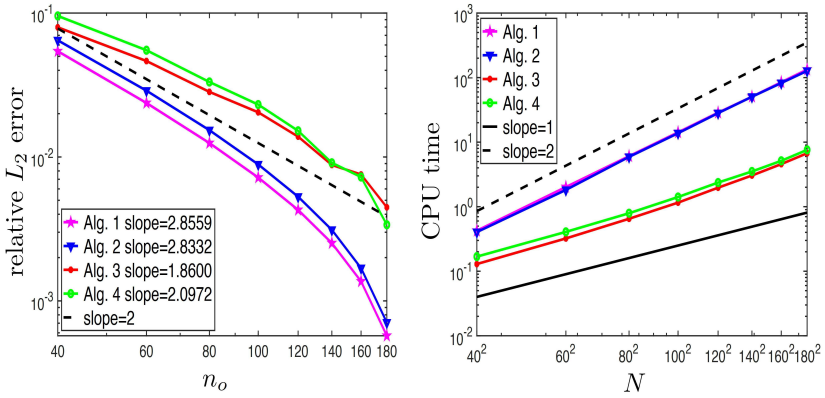

Figure 9: Convergence results for Algorithms 1-4 when $L=10$. Left: relative L_2 error at $t_{end}=20$ with respect to particle number per dimension n_o; Right: CPU time (in seconds) per time step with respect to particle number N.

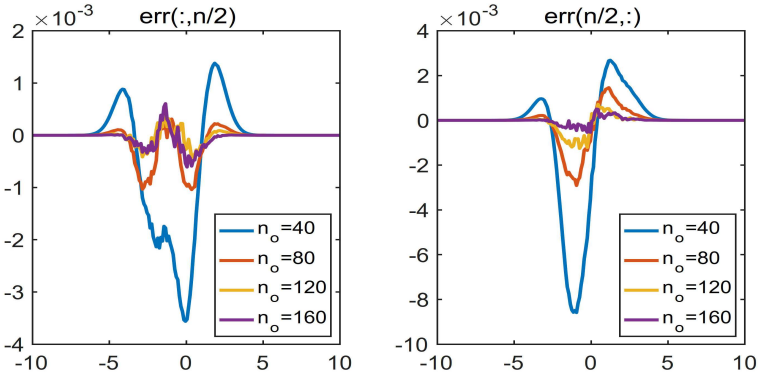

Figure 10: Error profiles of snapshots $f(:,\frac{n_o}{2})$ (left) and $f(\frac{n_o}{2},:)$ (right) for Algorithm 3 at $t_{end}=20$.

Let $\Delta t=0.2$, $L=1$, $q_o=4$. The cross sections $f(v_x,0,0)$ of the four algorithms at $t=0,5,10$ are shown in Fig. 11. As time goes by, it occurs to Algorithm 1 that the particles collide for large n_o. As the Coulomb kernel $A(z)$ is singular, the computation would break down once two particles collide. This can be overcome with a smaller time step. Since the homogeneous Landau equation is of diffusive type, $\Delta t=\mathcal{O}(\Delta v^2)$, $\Delta v=h=2L/n_o$ which is time-consuming. However, when using the random batch particle methods, the probability of two particles being close all the time is sufficiently small due to random reshuffling at each time step. So one can use a relatively bigger time step. Hence, in order to get the long time behavior of the Landau equation, it is preferable to use Algorithms 3 and 4. The cost is $\mathcal{O}(N)$ for Algorithms 3-4 as shown in Fig. 12. In Fig. 13 (left), the cross sections of the distribution function at times $t=0,9,36,81,144,225,900$ by Algorithm 3 are depicted. The results are in good agreement with those given in [3,39]. The profiles

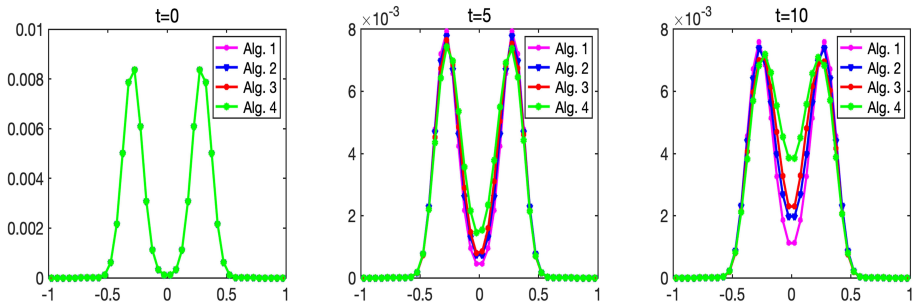

Figure 11: Cross section $f(v_x,0,0)$ at $t=0$ (left), $t=5$ (middle) and $t=10$ (right) for Algorithms 1-4 when $n_o=40$.

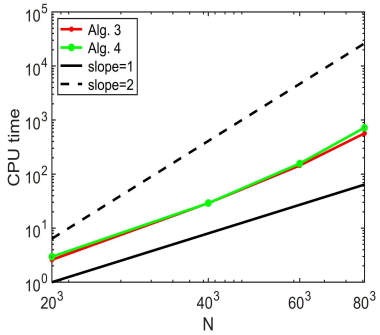

Figure 12: CPU time (in seconds) for one step for Algorithms 3-4 with respect to N.

approach the Maxwellian (1.4) with

$$\rho=\frac{2\pi\mu^3}{S^2}\left[\left(\frac{1}{2S}+1\right)\sqrt{\frac{\pi}{S}}\operatorname{erfc}(-\sqrt{S})+\frac{1}{S}\exp(-S)\right],\quad u=\mathbf{0},$$
$$T=\frac{1}{3\rho}\frac{2\pi\mu^5}{S^2}\left[\left(1+\frac{3}{S}+\frac{3}{4S^2}\right)\sqrt{\frac{\pi}{S}}\operatorname{erfc}(-\sqrt{S})+\left(\frac{1}{S}+\frac{5}{2S^2}\right)\exp(-S)\right]$$

in time as expected. As can be seen in Fig. 13, the energy and entropy reach the steady states as well.

The last example demonstrates that the random batch particle methods can efficiently approximate the equilibrium as well as the dynamics of the homogeneous Landau equation.

5 Conclusion

In this paper, we introduced random batch implementations of the particle methods for the homogeneous Landau equation proposed in [5]. For the collision term, at each time,

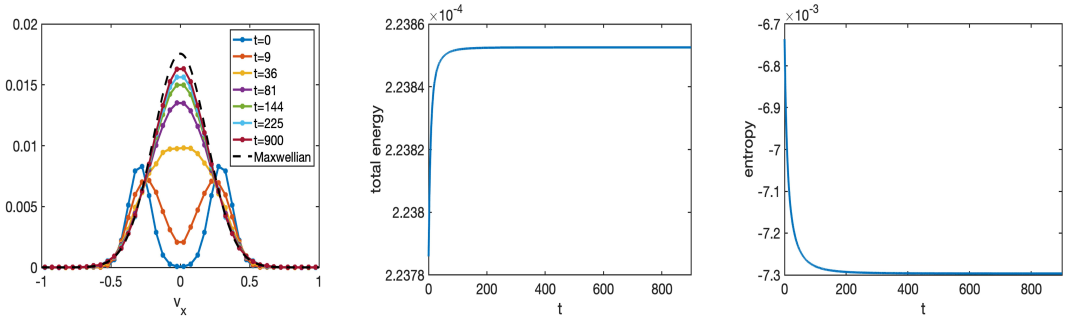

Figure 13: Results for Algorithm 3 when $n_o = 40$. Left: Cross section $f(v_x,0,0)$ of the distribution function at different times. The real equilibrium state is shown in the black dashed line. Middle: Time evolution of the total energy. Right: Time evolution of the entropy.

we randomly group the N-particles into small batches and each particle collides only with particles in the same batch. We also utilize the rapid decay property of the mollifier, hence the overall computational cost of our methods is $\mathcal{O}(N)$, instead of $\mathcal{O}(N^2)$. The conservation and entropy dissipation properties of these methods are also proved and numerical experiments verify the desired performance and theoretical results.

The random batch particle methods can be extended to deal with the homogeneous Landau equation for binary collisions of multi-species. It might be a promising way to efficiently solve the Fokker-Planck-Landau equation (1.1) integrating the particle-in-cell method [40] for the Vlasov equation and our random batch particle methods, so as to describe complex phenomena including Landau damping, two-stream instability, etc [10, 18, 20]. This is left for our future study.

Acknowledgments

JAC was supported by the Advanced Grant Nonlocal-CPD (Nonlocal PDEs for Complex Particle Dynamics: Phase Transitions, Patterns and Synchronization) of the European Research Council Executive Agency (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 883363). S. Jin's research was partly supported by the NSFC grant No.12031013 and the Strategic Priority Research Program of Chinese Academy of Sciences, XDA25010401.

References

[1] G. Albi and L. Pareschi, Binary interaction algorithms for the simulation of flocking and swarming dynamics, Multiscale Model. Simul., 11 (2013), 1-29.
[2] L. Bottou, Large-scale machine learning with stochastic gradient descent, in Proceedings of COMPSTAT'2010, 177-186, Physica-Verlag HD, Heidelberg, 2010.
[3] C. Buet and S. Cordier, Conservative and entropy decaying numerical scheme for the isotropic FokkerPlanckLandau equation, J. Comput. Phys., 145 (1998), 228-245.

[4] C. Buet, S. Cordier, P. Degond and M. Lemou, Fast algorithms for numerical, conservative, and entropy approximations of the FokkerPlanckLandau equation, J. Comput. Phys., 133 (1997), 310–322.
[5] J. A. Carrillo, J. Hu, L. Wang and J. Wu, A particle method for the homogeneous Landau equation, J. Comput. Phys.: X, 7 (2020), 100066.
[6] J. A. Carrillo, L. Pareschi and M. Zanella, Particle based gPC methods for mean-field models of swarming with uncertainty, Commun. Comput. Phys., 25 (2019), 508-531.
[7] J. A. Carrillo, M. G. Delgadino and J. Wu, Boltzmann to Landau from the gradient flow perspective, Nonlinear Anal., 219 (2022), 112824.
[8] J. A. Carrillo, M. G. Delgadino, L. Desvillettes and J. Wu, The Landau equation as a gradient flow, arXiv preprint arXiv: 2007.08591, 2020.
[9] J. A. Carrillo, K. Craig and F. S. Patacchini, A blob method for diffusion, Calc. Var. Partial Differ. Equ., 58 (2019), 1-53.
[10] F. F. Chen, Introduction to Plasma Physics and Controlled Fusion, Vol. 1, Springer, 1984.
[11] A. Chertock, A practical guide to deterministic particle methods, Handbook of Numerical Methods for Hyperbolic Problems, 18 (2017), 177-202.
[12] P. Degond and B. L. Desreux, The Fokker-Planck asymptotics of the Boltzmann collision operator in the Coulomb case, Math. Models Methods Appl. Sci., 2 (1992), 167-182.
[13] P. Degond and B. L. Desreux, An entropy scheme for the Fokker-Planck collision operator of plasma kinetic theory, Numer. Math., 68 (1994), 239-262.
[14] L. Desvillettes, On asymptotics of the Boltzmann equation when the collisions become grazing, Transp. Theory Stat. Phys., 21 (1992), 259-276.
[15] L. Desvillettes, Entropy dissipation estimates for the Landau equation in the Coulomb case and applications, J. Funct. Anal., 269 (2015), 1359-1403.
[16] L. Desvillettes and C. Villani, On the spatially homogeneous Landau equation for hard potentials. Part I: Existence, uniqueness ans smoothness, Comm. Partial Differential Equations, 25 (2000), 179-259.
[17] L. Desvillettes and C. Villani, On the spatially homogeneous Landau equation for hard potentials. Part II: H-theorem and applications, Comm. Partial Differential Equations, 25 (2000), 261-298.
[18] G. Dimarco, Q. Li, L. Pareschi and B. Yan, Numerical methods for plasma physics in collisional regimes, J. Plasma Phys., 81 (2015), 305810106.
[19] M. Erbar, A gradient flow approach to the Boltzmann equation, arXiv preprint arXiv: 1603.00540v1, 2016.
[20] F. Filbet and L. Pareschi, A numerical method for the accurate solution of the FokkerPlanck-Landau equation in the nonhomogeneous case, J. Comput. Phys., 179 (2002), 1-26.
[21] N. Fournier and H. Guérin, Well-posedness of the spatially homogeneous Landau equation for soft potentials, J. Funct. Anal., 256 (2009), 2542-2560.
[22] D. Frenkel and B. Smit, Understanding molecular simulation: from algorithms to applications, Vol. 1, Elsevier, 2001.
[23] J. Hu, S. Jin and R. Shu, A stochastic Galerkin method for the FokkerPlanckLandau equation with random uncertainties, in Theory, Numerics and Applications of Hyperbolic Problems II, 1-19, Springer, Cham, 2018.
[24] S. Jin, L. Li and J.-G. Liu, Random Batch Methods (RBM) for interacting particle systems, J. Comput. Phys., 400 (2020), 108877.
[25] S. Jin, L. Li, Z. Xu and Y. Zhao, A random batch Ewald method for particle systems with Coulomb interactions, SIAM J. Sci. Comput., 43 (2021), B937-B960.

[26] S. Jin and L. Li, Random batch methods for classical and quantum interacting particle systems and statistical sampling, in Active Particles, II, to appear.
[27] S. Jin and B. Yan, A class of asymptotic-preserving schemes for the FokkerPlanckLandau equation, J. Comput. Phys., 230 (2011), 6420-6437.
[28] L. D. Landau, Kinetic equation for the case of Coulomb interaction, Phys. Z. Sowjet., 154 (1936).
[29] M. Lemou, Multipole expansions for the Fokker-Planck-Landau operator, Numer. Math., 78 (1998), 597-618.
[30] M. Lemou and L. Mieussens, Implicit schemes for the Fokker–Planck–Landau equation, SIAM J. Sci. Comput., 27 (2005), 809-830.
[31] L. Li, Y. Li, J.-G. Liu, Z. Liu and J. Lu, A stochastic version of Stein variational gradient descent for efficient sampling, Comm. App. Math. Comp. Sci., 15 (2020), 37-63.
[32] L. Li, Z. Xu and Y. Zhao, A random batch Monte Carlo method for many-body systems with singular kernels, SIAM J. Sci. Comput., 42 (2020), A1486-A1509.
[33] L. Li, J.-G. Liu and Y. Tang, Some random batch particle methods for the Poisson-Nernst-Planck and Poisson-Boltzmann equations, arXiv preprint arXiv: 2004.05614, 2020.
[34] R. Li, Y. Ren and Y. Wang, Hermite spectral method for Fokker-Planck-Landau equation modeling collisional plasma, J. Comput. Phys., 434 (2021), 110235.
[35] R. Li, Y. Wang and Y. Wang, Approximation to singular quadratic collision model in Fokker–Planck–Landau equation, SIAM J. Sci. Comput., 42 (2020), B792-B815.
[36] J. Liang, P. Tan, Y. Zhao, L. Li, S. Jin, L. Hong and Z. Xu, Superscalability of the random batch Ewald method, J. Chem. Phys., 156 (2022), 014114.
[37] E. M. Lifshitz and L. P. Pitaevskiĭ, Course of Theoretical Physics ["Landau-Lifshits"], Vol. 10, Pergamon International Library of Science, Technology, Engineering and Social Studies. Pergamon Press, Oxford-Elmsford, N. Y., 1981.
[38] P.-L. Lions and S. Mas-Gallic, Une méthode particulaire déterministe pour des équations diffusives non linéaires, Comptes Rendus de l'Académie des Sciences Series I Mathematics, 332 (2001), 369-376.
[39] L. Pareschi, G. Russo and G. Toscani, Fast spectral methods for the FokkerPlanckLandau collision operator, J. Comput. Phys., 165 (2000), 216-236.
[40] D. Tskhakaya, K. Matyash, R. Schneider and F. Taccogna, The particle-in-cell method, Contrib. Plasma Phys., 47 (2007), 563–594.
[41] C. Villani, On a new class of weak solutions to the spatially homogeneous Boltzmann and Landau equations, Arch. Ration. Mech. Anal., 143 (1998), 273-307.
[42] C. Villani, On the spatially homogeneous Landau equation for Maxwellian molecules, Math. Models Methods Appl. Sci., 8 (1998), 957-983.
[43] C. Zhang and I. M. Gamba, A conservative scheme for Vlasov Poisson Landau modeling collisional plasmas, J. Comput. Phys., 340 (2017), 470-497.