

# A split control variate scheme for PIC simulations with collisions

Eric Sonnendrücker<sup>1,3</sup>, Abigail Wacher<sup>1</sup>, Roman Hatzky<sup>1</sup>, and Ralf Kleiber<sup>2</sup>

<sup>1</sup>Max Planck Institute for Plasma Physics, Boltzmannstr. 2, 85748 Garching, Germany

<sup>2</sup>Max Planck Institute for Plasma Physics, Wendelsteinstr. 1, 17491 Greifswald, Germany

<sup>3</sup>Mathematics Center, TU Munich, Boltzmannstr. 3, 85747 Garching, Germany

February 26, 2015

## Abstract

When the distribution function of plasma particles stays close to some analytically known function, statistical noise inherent to Monte Carlo simulations can be greatly reduced by introducing this function as a control variate in the computation of the velocity moments. Such a method, even though it can be naturally applied to nonlinear simulations, has originally emerged from linearised simulations and is usually called the  $\delta f$  particle-in-cell (PIC) method. In the past, the method has been extended to also handle collisions. This resulted in a two weight scheme which is known to produce a pronounced weight growth problem which rapidly makes it inefficient as a control variate method for variance reduction. In this work we analyse the weight growth problem within a simple example, which allows us to overcome its pathological behaviour. We also introduce a new split algorithm based on switching the control variate for PIC simulations with collisions. A key element of our algorithm is a new weight smoothing operator which enables us to obtain a significant noise reduction both in the presence of collisions and in the deep nonlinear phase of PIC simulations.

**Keywords:** *Vlasov-Poisson system; Fokker-Planck; Collisions; Particle In Cell; PIC; Monte Carlo; Control Variate.*

## 1 Introduction

The Particle In Cell (PIC) method is a well established method for the numerical simulation of collisional or collisionless kinetic plasmas [4]. Its main drawback compared to grid based methods is its numerical noise that is slowly decreasing (as  $1/\sqrt{N}$ ) with the number of particles. In some situations, like in gyrokinetic simulations of magnetic fusion plasmas, using a standard PIC method requires either a huge number of particles or generates noise levels above the signal level. This could be remedied by the so-called  $\delta f$  approach [10] which allows to decrease the noise by orders of magnitude by simulating with particles only the deviation from equilibrium. This method involving linearisation in its early stages can be made completely nonlinear. Aydemir [3] pointed out that this method could be interpreted as a control variate method classically used in Monte Carlo simulations. This method has been proven to be very efficient and useful for collisionless gyrokinetic simulations. However, it has been very difficult to efficiently handle collisions with this method. The two weight scheme introduced in [5] has been prone to an artificial weight growth [8] that is problematic for long time simulations as it can increase the variance and thus also the noise

drastically. Even though the coarse graining method [7] allows to alleviate the problem it is still not completely satisfying.

In this paper we motivate a new method for PIC simulations with collisions using a control variate approach. One of its features is the idea of switching the control variate between the advection and the collision step, which has been first introduced by Vernay et al. [15, 16]. We proceed by analysing the error introduced by the two weight  $\delta f$  particle in cell method for solving the collisional Vlasov equation, and observe that the pathological weight growth can be eliminated by using a well suited control variate during the collision step, an equilibrium function of the collision operator, which can be usually computed in practice. This brings us to the formulation of our algorithm based on a splitting between the advection step and the collision step, using at every collision step the actual equilibrium of the collision operator as a control variate. Any other function (close enough) can be used as a control variate during the advection phase, the algorithm then involving a switching of control variates between the advection and collision steps. In practice this method only involves the evolution of one weight with no time integration during the collision step. This completely suppresses the weight growth during the collision step, however a slight weight growth is still observed due to the advective and rescaling stages.

Moreover, the collision step should not only be free of weight growth but also smooth the weights so that the collisions actually bring back the distribution to equilibrium and thus diminish the weights at that step. This could be realised by a kernel density estimate which is akin to coarse graining. Instead, we propose a simpler procedure which is grid less and only involves pairs of close particles, thus making it completely local, which is a big asset for large parallel computations. We call this: neighbour smoothing procedure. We will show that adding this step to our algorithm allows to have an appropriate relaxation of the distribution function towards the equilibrium with a corresponding reduction of the variance of the weights and results in better statistics.

The article is organised as follows. First we introduce an abstract model for a collisional phase space transport. In our examples it will represent the Vlasov-Fokker-Planck equation, but it could also be the gyrokinetic equations with collisions. This model is split into two parts, the first one defining an advection along some characteristics and the second a collision operator that admits a known steady state. Then both parts are considered individually, the collision operator being cast into the framework of stochastic differential equations (SDE). A split algorithm for the solution of the initial problem is then introduced. After that, a new neighbour smoothing procedure is introduced to be used in addition with the previous algorithm to ensure a pointwise convergence of the distribution function to the equilibrium distribution in the case of strong collisions. A section is then devoted to the analysis of the weight growth problem that plagues standard collisional two weights PIC simulations. And finally we validate the new methods introduced in the paper on relevant test cases.

## 2 An abstract collisional Vlasov equation

Denoting by  $\mathbf{z} = (x, v)$  the phase space variables we consider a collisional Vlasov equation with a known source term of the form

$$\frac{\partial f}{\partial t} + \mathbf{A} \cdot \nabla f = -\nabla \cdot (\mathbf{A}_C f) + \frac{1}{2} \nu^2 \Delta_v f + \tilde{\mathcal{S}}(f). \quad (1)$$

where  $\nabla := (\partial_x, \partial_v)$  acts on the phase space and the diffusion operator  $\Delta_v$  acts in velocity space only.  $\tilde{\mathcal{S}}(f)$  is a general source term. This abstract model contains the Vlasov-Fokker-Planck model as well as the gyrokinetic equations with a linear collision operator.

The linearity of the collision operator assumed for simplicity in this work is not required for our algorithm. It would also be possible to use the nonlinear Landau collision operator expressed with Rosenbluth potentials.

In our particle in cell scheme we are going to treat independently the advection and collision term using a time splitting technique, the advection part being

$$\frac{\partial f}{\partial t} + \mathbf{A} \cdot \nabla f = \tilde{\mathcal{S}}(f), \quad (2)$$

and the collision part

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{A}_C f) - \frac{1}{2} \nu^2 \Delta_v f = 0. \quad (3)$$

This splitting is somewhat arbitrary since for a given equation we have some freedom in what to put into  $\mathbf{A}$  and what into  $\mathbf{A}_C$ . However a strong requirement for our method to work is that the collision part Eq. (3) admits an analytically known non trivial steady-state, which is typically a local Maxwellian.

### 3 The $\delta f$ method in the pure advection case

The standard variance reduction technique used in the PIC method, is the so-called  $\delta f$  method, where a control variate associated with a known equilibrium distribution close to the computed distribution is used [8].

Let us introduce an arbitrary function  $f^0(t, \mathbf{z})$  that is analytically known. Then define, by explicit computation

$$\mathcal{S}^0 = -\frac{\partial f^0}{\partial t} - \mathbf{A}[f] \cdot \nabla f^0. \quad (4)$$

We explicitly highlight a possible dependency of  $\mathbf{A}$  on  $f$  through *e.g.* an electric field. Then, defining  $\delta f = f - f^0$ , we get from Eq. (2)

$$\frac{\partial \delta f}{\partial t} + \mathbf{A}[f] \cdot \nabla \delta f = \tilde{\mathcal{S}} + \mathcal{S}^0 =: \mathcal{S}. \quad (5)$$

This is an advection equation with source term that can be solved numerically using a weighted particle method. The weights will evolve in time due to the source term. In this context, in order to get more flexibility in the initial phase-space positions of the particles  $\mathbf{z}_k(0)$ , instead of drawing them according to the distribution function of the plasma particles  $f$  as is usually done in PIC methods, we draw them according to an arbitrary probability density  $g$  with support larger than the support of  $f$ . The phase space positions drawn according to the chosen probability density  $g$ , will be called markers. These will be evolved in time following the characteristics of the advection equation, which are the trajectories of the actual plasma particles. A particle approximation of the distribution function can be expressed with these markers as

$$f_N(t, \mathbf{z}) = \sum_{k=1}^N c_k \delta(\mathbf{z} - \mathbf{z}_k). \quad (6)$$

The weights  $c_k$  do not evolve in time as  $f$  and  $g$  are conserved along the same characteristics, and they are determined at the initial time by the initial marker positions  $\mathbf{z}_k^0 = \mathbf{z}_k(0)$ :  $c_k = f_0(\mathbf{z}_k^0)/g_0(\mathbf{z}_k^0)$  with the definition  $f_0(\mathbf{z}_k^0) = f(0, \mathbf{z}_k^0)$  and  $g_0(\mathbf{z}_k^0) = g(0, \mathbf{z}_k^0)$ .

An additional particle approximation of  $\delta f$  can be expressed by using the same markers distributed according to  $g$

$$\delta f_N(t, \mathbf{z}) = \sum_{k=1}^N \gamma_k(t) \delta(\mathbf{z} - \mathbf{z}_k), \quad (7)$$

with the initial values of the markers given by

$$\gamma_k^0 = \frac{\delta f(0, \mathbf{z}_k^0)}{g_0(\mathbf{z}_k^0)} = \frac{f(0, \mathbf{z}_k^0) - f^0(0, \mathbf{z}_k^0)}{g_0(\mathbf{z}_k^0)}. \quad (8)$$

Then  $\delta f_N$  is a solution of Eq. (5) provided

$$\frac{d\mathbf{z}_k}{dt} = \mathbf{A}[f](t, \mathbf{z}_k(t)), \quad \frac{d\gamma_k}{dt} = \frac{\mathcal{S}(t, \mathbf{z}_k(t))}{g(t, \mathbf{z}_k(t))}. \quad (9)$$

The evolution of the phase space positions of the markers  $\mathbf{z}_k$  can be computed as in a standard PIC method. In addition, the evolution of the weights need to be computed by numerically solving the equation for  $\gamma_k$ . If no linearisation is performed this method is equivalent to a standard full- $f$  PIC method and yields a nonlinear approximation of the full distribution function  $f$  by adding  $f^0$  to the computed  $\delta f_N$ .

It has been noted by Aydemir [3] and Allfrey and Hatzky [2] that because  $\frac{d}{dt} \left( \frac{f(t, \mathbf{z}_k(t))}{g(t, \mathbf{z}_k(t))} \right) = 0$  along the characteristics in the absence of source terms (*i.e.*  $\tilde{\mathcal{S}} = 0$ ), the solution of the weight equation can be computed explicitly once the phase space trajectories are known. Indeed integrating the previous relation between  $t = 0$  and  $t_n$  and writing  $f(t, \mathbf{z}_k) = \delta f(t, \mathbf{z}_k) + f^0(t, \mathbf{z}_k)$ , we get

$$\frac{\delta f(t_n, \mathbf{z}_k^n)}{g(t_n, \mathbf{z}_k^n)} = \frac{\delta f(0, \mathbf{z}_k^0)}{g_0(\mathbf{z}_k^0)} + \frac{f^0(0, \mathbf{z}_k^0)}{g_0(\mathbf{z}_k^0)} - \frac{f^0(t_n, \mathbf{z}_k^n)}{g(t_n, \mathbf{z}_k^n)}. \quad (10)$$

Then using that  $g$  is conserved along the marker trajectories, we get  $g(t_n, \mathbf{z}_k^n) = g_0(\mathbf{z}_k^0)$ , which is explicitly known and it follows

$$\gamma_k^n = \frac{f(0, \mathbf{z}_k^0) - f^0(t_n, \mathbf{z}_k^n)}{g_0(\mathbf{z}_k^0)} = c_k - \frac{f^0(t_n, \mathbf{z}_k^n)}{g_0(\mathbf{z}_k^0)}. \quad (11)$$

This gives an explicit solution for the weight equation. The methods based on this are generally called direct  $\delta f$  methods in the literature. In cases where  $f$  is not conserved along the characteristics because of the presence of a source term  $\tilde{\mathcal{S}}$  or when Eq. (5) is linearised, the weights need to be integrated numerically using a standard ODE solver.

In our case the advection scheme is only one of the split steps, the other being a collision step in which there are no characteristics for tracing back the value of  $g$ . Hence we can no more trace back  $g$  to the initial value along the characteristics. However for one marker in the advection part we still have that  $g(t_{n+1}, \mathbf{z}_k^{n+1}) = g(t_n, \mathbf{z}_k^n)$ . We can use this to find an expression of  $\gamma_k^{n+1}$  as a function of  $\gamma_k^n$ . This is easily obtained by deriving the equivalent of Eq. (10) by integration between  $t_n$  and  $t_{n+1}$  instead of 0 and  $t_n$  from which it follows that

$$\gamma_k^{n+1} = c_k - (c_k - \gamma_k^n) \frac{f^0(t_{n+1}, \mathbf{z}_k^{n+1})}{f^0(t_n, \mathbf{z}_k^n)}. \quad (12)$$

## 4 Stochastic differential equation associated to a Fokker-Planck equation

The particle method introduced in the previous section, can be extended to include the collision term by replacing the characteristic equations by a stochastic differential equation (see *e.g.* [8]).

Let  $f$  be a function depending on time and on a  $d$ -dimensional phase space variable  $\mathbf{z}$  solution of the Fokker-Planck equation

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{A}_C f) - \frac{1}{2} \nu^2 \Delta_v f = 0, \quad (13)$$

with an initial condition  $f(0, \mathbf{z}) = f_0(\mathbf{z})$ . In order to be able to interpret  $f$  as a probability density, we shall assume that it is normalised such that  $\int f_0(\mathbf{z}) d\mathbf{z} = 1$ . Because the equation is conservative, this property is then true for  $f(t, \cdot)$  at any time. Here  $\mathbf{A}_C$  denotes a vector with  $d$  components and  $\nu$  is a constant diffusion coefficient. More complex diffusion tensors could be handled also in the same framework, but we shall stick with constant diffusion for simplicity.

The Itô stochastic differential equations (SDE), see for example [13] for an introduction, provide a framework where the solution of a SDE can be used like characteristics to define the solution of Eq. (13). The SDE associated to Eq. (13) in the Itô formalism reads

$$d\mathbf{Z} = \mathbf{A}_C(t, \mathbf{Z}) dt + \nu d\mathbf{W}_v, \quad (14)$$

$$\mathbf{Z}(s) = \mathbf{z}. \quad (15)$$

Here  $s$  is an arbitrary initial time and  $\mathbf{W}_v$  is a  $d/2$ -dimensional Wiener process in velocity as usual in the theory of SDEs. In accordance with our physical problem, we consider only diffusion in velocity space.

The Itô formula provides an equivalent to the total derivative in the classical case. This reads in our case

$$df(t, \mathbf{Z}(t)) = \left( \frac{\partial f}{\partial t} + \mathbf{A}_C \cdot \nabla f + \frac{1}{2} \nu^2 \Delta_v f \right) (t, \mathbf{Z}(t)) dt + \nu \nabla f(t, \mathbf{Z}(t)) \cdot d\mathbf{W}_v. \quad (16)$$

However, because this involves random variables and a Wiener process, it cannot be used to travel at wish forward or backward in time like characteristics. Taking the expectation, enables to get rid of the Wiener process due to one of its properties, then we get the conservation of expectation along the path of a stochastic differential equation, which enables us to link the SDE and a partial differential equation. However, the PDEs involved are different in the forward, initial value problem, and backward, terminal value problem. One is the adjoint of the other. We are interested in the initial value problem, for which the Fokker-Planck equation, also known as Kolmogorov forward equation in the stochastic literature, associated to SDE Eqs. (14)–(15) is the PDE Eq. (13) we are interested in. In other words, the probability density evolved by markers satisfying Eqs. (14)–(15) is the solution of Eq. (13). This gives us the framework needed for defining a particle method to find an approximate solution of Eq. (13).

## 5 The stochastic Particle In Cell (PIC) method

In order to make clear that random variables which are functions are different mathematical objects than standard variables, in the following random variables will be denoted by capital letters and standard variables with small letters (*e.g.*  $\Gamma, \gamma$ ), as is standard in the mathematical literature.

A particle method, as introduced in Section 3, using an approximation of a function as a sum of Dirac masses can also be expressed with the tools of probability theory. It consists in approximating the initial distribution function of particles, which is normalised such that it defines a probability density, by a large number of realisations of the random variable  $\mathbf{Z}(0)$  drawn with the initial phase space density. Then, as the density is conserved along the characteristics the evolution in time of the phase space density is sampled by the random variable  $\mathbf{Z}(t)$ , where  $\mathbf{Z}(t)$  is a solution of the deterministic equation of characteristics with initial conditions  $\mathbf{Z}(0)$ . The physical quantities of interest at any time can be defined as expectations of functions of  $\mathbf{Z}(t)$ . For any function  $\psi$  defined on the phase space  $\mathbb{R}^d$ , by the law of large numbers an approximation of the expectations is given by the average of the samples:

$$\mathbb{E}(\psi(\mathbf{Z}(t))) \approx \frac{1}{N} \sum_{k=1}^N \psi(\mathbf{z}_k(t)), \quad (17)$$

where  $\mathbf{z}_k(0)$ ,  $k = 1, \dots, N$  are  $N$  realisations of the initial random variable  $\mathbf{Z}(0)$  of density  $f_0$ . Then  $\mathbf{z}_k(t)$ ,  $k = 1, \dots, N$  the solutions of the characteristic equations with initial conditions  $\mathbf{z}_k(0)$ , are realisations of  $\mathbf{Z}(t)$  of density  $f(t, \mathbf{z})$ .

The error of this approximation, in a probabilistic sense, is given by the central limit theorem and depends on the variance of the random variables. This error is known as numerical noise. Knowing this, in order to reduce the error for a given number of particles, the strategy which is classical in Monte Carlo approximation is to reduce the variance. Many ways of doing that like importance sampling, control variates, antithetic variates and others are known from statistics.

As the stochastic differential equations provide a replacement for the characteristics for advection-diffusion equations, the PIC method can be applied in the same way in this context.

We now need to extend the weighted PIC method, necessary to define the  $\delta f$  method as a tool to reduce noise in a PIC method, to the stochastic setting. We therefore need a new definition of the weights as a new random variable, denoted by  $\Gamma$ , which will enable us to express any probability density with respect to the marker distribution in the phase space extended by a new variable  $\gamma$ . In the  $\delta f$  PIC context this has been done in [6, 5, 8].

More concretely, consider that we want to solve Eq. (13) with a weighted PIC method. Then we need to define the SDE for which Eq. (13) is the Fokker-Planck equation. These are Eqs. (14)–(15). But now in order to get more flexibility in the initial phase-space positions of the particles  $\mathbf{Z}(0)$ , we draw them according to the probability density  $\bar{g}_0$  on the extended phase space  $(\mathbf{z}, \gamma) \in \mathbb{R}^d \times [0, +\infty)$  that we are free to fix instead of  $f_0$ . The idea is related to the technique of importance sampling, see *e.g.* [11]. To connect  $\bar{g}_0$  with  $f_0$  we require that

$$\int_0^{+\infty} \gamma \bar{g}_0(\mathbf{z}, \gamma) d\gamma = f_0(\mathbf{z}). \quad (18)$$

Then we can compute the expectations with respect to  $f_0$  that we need as expectations with respect to  $\bar{g}_0$  using

$$\mathbb{E}_{f_0}(\psi(\mathbf{Z})) = \int \psi(\mathbf{z}) f_0(\mathbf{z}) d\mathbf{z} = \int \int_0^{+\infty} \gamma \psi(\mathbf{z}) \bar{g}_0(\mathbf{z}, \gamma) d\gamma d\mathbf{z} = \mathbb{E}_{\bar{g}_0}(\Gamma \psi(\mathbf{Z})). \quad (19)$$

Note that an approximation of  $\mathbb{E}_{\bar{g}_0}(\Gamma \psi(\mathbf{Z}))$  by a finite number of samples yields the natural form of a weighted PIC method:

$$\mathbb{E}_{\bar{g}_0}(\Gamma \psi(\mathbf{Z})) \approx \frac{1}{N} \sum_{k=1}^N \gamma_k \psi(\mathbf{z}_k). \quad (20)$$

In this framework the factor  $1/N$  is taken out of the weights compared to the definition of  $f_N$  in Section 3. In order to define a weighted PIC method, knowing  $f_0$  we then need a procedure to define the initial extended marker distribution  $\bar{g}_0$  such that Eq. (18) is satisfied. A natural way to do this is to pick arbitrarily the phase space marker distribution  $g_0(\mathbf{z}) = \int_0^{+\infty} \bar{g}_0(\mathbf{z}, \gamma) d\gamma$ . It could follow a uniform distribution for example, or any other distribution that makes sense for the problem at hand. Then the classical weighted PIC method is obtained by taking  $\bar{g}_0(\mathbf{z}, \gamma) = \delta(\gamma - \gamma(\mathbf{z}))g_0(\mathbf{z})$ . Plugging this expression into Eq. (18) we get

$$\int_0^{+\infty} \gamma \bar{g}_0(\mathbf{z}, \gamma) d\gamma = \gamma(\mathbf{z})g_0(\mathbf{z}) = f_0(\mathbf{z}). \quad (21)$$

Thus, it follows that the condition Eq. (18) is satisfied provided  $\gamma(\mathbf{z}) = f_0(\mathbf{z})/g_0(\mathbf{z})$ . This is the classical expression for importance sampling.

Finally, in order to initialise our weighted PIC method, we draw the initial particle phase space positions  $\mathbf{z}_k(0)$  according to the phase space marker density  $g_0$ . Then the weights are defined by  $\gamma_k^0 = f_0(\mathbf{z}_k(0))/g_0(\mathbf{z}_k(0))$ .

Now if we are not merely interested in a weighted PIC method, but in a control variate PIC method, we need to describe two distribution functions, the one of the actual particles  $f$  and also the control variate  $f^0$  with the same marker distribution. This will eventually yield the weights for  $\delta f$  as the difference between the weights for  $f$  and for  $f^0$ . Hence we consider a marker distribution  $\bar{g}(z, \gamma, \gamma_2)$  which depends on two weights,  $\gamma$  for representing  $f$  and  $\gamma_2$  for  $f^0$ . As  $f$  and  $f^0$  are probability densities as well as  $\bar{g}$ , they are positive, and so also the weights  $\gamma$  and  $\gamma_2$ . On the other hand  $\gamma_1 = \gamma - \gamma_2$  does represent  $\delta f$  which is not a probability density and can take positive and negative values.

To determine the time evolution of the weights, we define the stochastic processes  $\Gamma(t)$  and  $\Gamma_2(t)$  such that  $\mathbf{Z}(t)$ , along with  $\Gamma(t)$  and  $\Gamma_2(t)$  can be used to approximate expectations relative to  $f(t)$  and  $f^0(t)$  respectively. Therefore, we introduce the following extended phase-space marker equation

$$\frac{\partial \bar{g}}{\partial t} + \nabla_z \cdot (\mathbf{A}_C \bar{g}) - \frac{1}{2} \nu^2 \Delta_v \bar{g} + \frac{\partial}{\partial \gamma} (\lambda \bar{g}) + \frac{\partial}{\partial \gamma_2} (\lambda_2 \bar{g}) = 0, \quad (22)$$

where  $\lambda$  and  $\lambda_2$  are functions of  $\gamma, \gamma_2$  and  $\mathbf{z}$  that need to be defined. The SDE for which Eq. (22) is the Fokker-Planck equation reads

$$d\mathbf{Z}(t) = \mathbf{A}_C dt + \nu d\mathbf{W}_v(t), \quad d\Gamma = \lambda dt \quad d\Gamma_2 = \lambda_2 dt. \quad (23)$$

This is the SDE associated to our initial equation supplemented with a SDE (which is in reality an ODE depending on random variables) for the weights.

In order to be able to compute, for all positive times  $t$ , expectations with respect to  $f(t, \mathbf{z})$  and  $f^0(t, \mathbf{z})$  using expectations with respect to  $\bar{g}(t, \mathbf{z}, \gamma, \gamma_2)$ , we need as for Eq. (18) that

$$\int_0^{+\infty} \int_0^{+\infty} \gamma \bar{g}(t, \mathbf{z}, \gamma, \gamma_2) d\gamma d\gamma_2 = f(t, \mathbf{z}), \quad \int_0^{+\infty} \int_0^{+\infty} \gamma_2 \bar{g}(t, \mathbf{z}, \gamma, \gamma_2) d\gamma d\gamma_2 = f^0(t, \mathbf{z}). \quad (24)$$

These conditions determine  $\lambda$  and  $\lambda_2$ . Indeed, multiplying Eq. (22) by  $\gamma$  and integrating with respect to  $\gamma$  and  $\gamma_2$ , we get, using Eq. (24)

$$\frac{\partial f}{\partial t} + \nabla_z \cdot (\mathbf{A}_C f) - \frac{1}{2} \nu^2 \Delta_v f - \int_0^{+\infty} \int_0^{+\infty} \lambda \bar{g} d\gamma d\gamma_2 = 0. \quad (25)$$

Comparing this equation with Eq. (13) satisfied by  $f$ , we get that  $\lambda = 0$ , which means that the weights do not change in time in this case. This is the same as in the deterministic case for the pure advection equation. These constants weights, corresponding to the  $c_k$  in the pure advection case are then given at any time by

$$\Gamma(t) = \Gamma(0) = \frac{f_0(\mathbf{Z}(0))}{g_0(\mathbf{Z}(0))}. \quad (26)$$

In the same way, multiplying Eq. (22) by  $\gamma_2$ , integrating with respect to the two weights and assuming  $\bar{g}$  vanishes for  $\gamma_2 = 0$  and infinity, we get

$$\int_0^{+\infty} \int_0^{+\infty} \lambda_2 \bar{g} d\gamma d\gamma_2 = \frac{\partial f^0}{\partial t} + \nabla_z \cdot (\mathbf{A}_C f^0) - \frac{1}{2} \nu^2 \Delta_v f^0 =: \mathcal{S}. \quad (27)$$

Taking  $\lambda_2(t, \gamma, \gamma_2, \mathbf{z}) = \gamma_2 \mathcal{S}(t, \mathbf{z}) / f^0(t, \mathbf{z})$  solves this equation using the second equality of (24). Moreover, the weight  $\Gamma_1$  for  $\delta f$  can be defined as  $\Gamma_1 = \Gamma - \Gamma_2$ , and as  $\Gamma$  is constant in time, we get the evolution equations for the weights, which complement the equations of motion of the particles:

$$d\Gamma_1(t) = -\Gamma_2(t) \frac{\mathcal{S}(t, \mathbf{Z}(t))}{f^0(t, \mathbf{Z}(t))} dt, \quad d\Gamma_2(t) = \Gamma_2(t) \frac{\mathcal{S}(t, \mathbf{Z}(t))}{f^0(t, \mathbf{Z}(t))} dt. \quad (28)$$

These general equations are the same as the two weight scheme described in [5, 8]. They are complemented with the initial conditions

$$\Gamma_1(0) = \frac{f_0(\mathbf{Z}(0)) - f^0(\mathbf{Z}(0))}{g_0(\mathbf{Z}(0))}, \quad \Gamma_2(0) = \frac{f^0(\mathbf{Z}(0))}{g_0(\mathbf{Z}(0))}. \quad (29)$$

The particle equations of motion Eqs. (14)–(15) along with these SDEs defining the weights can be integrated numerically using a standard SDE solver, like the Euler-Maruyama method, or a higher order stochastic Runge-Kutta method [9].

As we will see in the sequel, within our splitting procedure, a much better procedure is to always use as  $f^0$  for the collisional part an equilibrium  $f_{\text{eq}}$  of Eq. (3) since then  $\mathcal{S} = 0$  as can be seen from its definition in Eq. (27) and  $\Gamma_1, \Gamma_2$  are constant and consequently the weight growth problem is absent. This also has the advantage of avoiding a numerical integration of the weight equations as was already noticed by Vernay [15, 16]. As he also noticed, a simple rescaling procedure still allows to use an arbitrary control variate for the advection step. As we will observe later, this only involves a moderate weight growth which saturates and does not lead to infinitely large weights like the previous method with an integration of the weight equation.

## 6 The split algorithm for coupled advection and collisions

As we have seen in the previous sections, in the pure advection case the evolution of the weights can be computed analytically for all choices of control variates. On the other hand, in the collisional case, the weights remain constant when an equilibrium of the collision operator is used as a control variate. In order to couple these we shall define a split algorithm treating the two parts separately. Then if the control variates are switched between the advection and the collision step and also when the local Maxwellian is an equilibrium of the collision operator and changes from one time step to the next, a scaling phase will be necessary before the collision step. The correct weighting of the marker density in order to simulate a given distribution is given by Eq. (18). So if we want



to switch the representation of the weights from a function  $f^1$  to another function  $f^2$ , we simply need to use

$$f^2(\mathbf{z}) = \int_0^{+\infty} \gamma \frac{f^2(\mathbf{z})}{f^1(\mathbf{z})} \bar{g}(\mathbf{z}, \gamma) d\gamma, \quad \text{as} \quad f^1(\mathbf{z}) = \int_0^{+\infty} \gamma \bar{g}(\mathbf{z}, \gamma) d\gamma, \quad (30)$$

so that the weights associated to  $f^2$  are obtained from the weights associated to  $f^1$  by a multiplication by  $f^2(\mathbf{z}_k)/f^1(\mathbf{z}_k)$ .

We can now express our algorithm. We will consider here only the direct  $\delta f$  method assuming the source term  $\tilde{S} = 0$ . When this is not the case, the direct integration of the weight equation in the advection part just needs to be replaced by a numerical integration. Our algorithm then reads as follows:

1. Initialisation:

- Pick any appropriate control variate  $f^0$  for the advection step.
- Load initial markers  $\mathbf{z}_k^0$  in phase space according to distribution  $g_0(\mathbf{z})$ .
- Define

$$c_k^* = \frac{f_0(\mathbf{z}_k^0)}{g_0(\mathbf{z}_k^0)}, \quad \gamma_{2,k}^* = \frac{f^0(\mathbf{z}_k^0)}{g_0(\mathbf{z}_k^0)}. \quad (31)$$

- Rescale  $c_k^*$  and  $\gamma_{2,k}^*$  so that their sum is exactly one. This will enable exact particle number conservation, if the rescaling is performed at each time step (see Eq. (39)) before the weights are needed in an assignment procedure.

$$c_k = c_k^* / \sum_{l=1}^N c_l^*, \quad \gamma_{2,k}^0 = \gamma_{2,k}^* / \sum_{l=1}^N \gamma_{2,l}^*, \quad \gamma_{1,k}^0 = c_k - \gamma_{2,k}^0. \quad (32)$$

2. Time iteration from  $t_n$  to  $t_{n+1}$ . Given  $\mathbf{z}_k^n, \gamma_{1,k}^n$ ,

- Advection step:
  - Advance markers numerically solving the characteristics Eq. (9)

$$\frac{d\mathbf{z}_k}{dt} = \mathbf{A}[f](t, \mathbf{z}_k(t)) \quad (33)$$

using any numerical solver on a step  $\Delta t$  to get  $\mathbf{z}_k^*$ . We use a fourth order Runge-Kutta method.

- Advance weights using Eq. (12)

$$\gamma_{1,k}^* = c_k - (c_k - \gamma_{1,k}^n) \frac{f^0(t_{n+1}, \mathbf{z}_k^*)}{f^0(t_n, \mathbf{z}_k^n)}. \quad (34)$$

- Collision step:
  - Compute parameters needed to define the equilibrium distribution  $f_{\text{eq}}(t, \mathbf{z})$  of the collision operator. These would be mean velocity  $u(x)$  and temperature  $T(x)$  if it is a local Maxwellian. Rescale control variate weights in order to use the equilibrium of the collision operator as a control variate.

$$\gamma_{2,k}^{**} = \gamma_{2,k}^* \frac{f_{\text{eq}}(\mathbf{z}_k^*)}{f^0(\mathbf{z}_k^*)}. \quad (35)$$

- Advance velocities by solving the Stochastic Differential Equation associated to the collision operator (14)–(15). In the case when this SDE defines the Ornstein-Uhlenbeck process

$$dV = -\mu(V - u) dt + D dW, \quad (36)$$

where  $D$  and  $\mu$  are related to the temperature  $T$  of its equilibrium distribution by  $T = \sqrt{D^2/(2\mu)}$ , we use the known transition probability of the Ornstein-Uhlenbeck process, *i.e.*  $v_k^{**}$  is drawn randomly from the normal distribution (Gaussian)  $\mathcal{N}(m, \sigma^2)$  with

$$m = u + (v_k^* - u)e^{-2\mu\Delta t}, \quad \sigma^2 = T(1 - e^{-2\mu\Delta t}). \quad (37)$$

In the general case we would solve the SDE numerically using the Euler-Maruyama or a higher-order numerical method for SDEs.

- Positions ( $x_k$ ) and weights ( $\gamma_{1,k}$ ) are not modified during this step.
- Rescale control variate weights  $\gamma_2$  to the initial control variate:

$$\gamma_{2,k}^{***} = \gamma_{2,k}^{**} \frac{f^0(\mathbf{z}_k^{**})}{f_{\text{eq}}(\mathbf{z}_k^{**})}. \quad (38)$$

- Rescale the  $\gamma_2$  weights so that their sum is exactly one and compute  $\gamma_1$  accordingly:

$$\gamma_{2,k}^{n+1} = \gamma_{2,k}^{***} / \sum_{l=1}^N \gamma_{2,l}^{***}, \quad \gamma_{1,k}^{n+1} = c_k - \gamma_{2,k}^{n+1}. \quad (39)$$

Note that the weight rescaling step in Eq. (39) is essential. It is generally considered a better estimator in the context of importance sampling even though the rescaling introduces a small bias. For Particle-In-Cell algorithm this step makes the method conservative. It is essential to avoid *e.g.* the development of a spurious electric field when the split algorithm is applied to Eq. (50).

The idea to switch control variates between advection and collision steps and to use the local Maxwellian during the collision step to avoid a numerical weight integration was first introduced to our knowledge by Vernay [15, 16].

## 7 Weight smoothing

Especially during the collision step, particles with different weights are mixed in neighbouring regions of phase space. Due to this the statistics are not as good as they would be with smoothly varying weights. This explains in particular why the  $L^2$  error for Ornstein-Uhlenbeck (see Figure 1) does not decrease as the distribution function gets closer to the control variate which is the equilibrium distribution. In order to keep smoothly varying weights after a stochastic collision step each individual marker needs to get information from close-by weights. This can be done by resampling the different probability densities from time to time, which is usually done in statistics via a so-called kernel density estimate, directly computing the approximate probability density at the particle positions, which can be fairly expensive. Another, generally cheaper option is to sample the probability densities on a fixed grid and then interpolate at the particle positions. Such a procedure was introduced as a coarse graining technique by Brunner, Valeo and Krommes [5] and by Chen and Parker [7]. This procedure can be very inaccurate if the grid is too fine and so the local statistics not good enough, and very diffusive if the grid is too coarse. Moreover, it introduces

a bias linked to the choice of the grid. To alleviate this Chen and Parker proposed a blending procedure, including only a fraction of this coarse grained value in the weight update.

We propose here another procedure, which is grid less and more probabilistic in nature. Moreover, it has the advantage of not needing to compute the numerical probability densities, which is error prone in regions where the particle statistics are not very good and more expensive. The principle consists in randomly pairing particles and replacing the two corresponding weights by an average, weighted by a function of the distance in phase space of the two particles. Of course, the way of computing the distance and the weight has an influence on the diffusiveness of the method. Moreover, a purely random pairing of the particles would be very inefficient as most of them would not interact because they are too far apart. For this reason we use a procedure for pairing only particles which are close enough. This is done as follows.

First we consider only particles that are in the same cell in configuration space, then if the velocity space is 1D we sort the particles according to their velocity and each particle is paired with its neighbour in velocity space. Then for two particles labeled by  $k_1$  and  $k_2$  that have been paired, the weights are smoothed according to the following formula

$$\gamma_{k_1}^{\text{new}} = (1 - e^{-\frac{(v_{k_1} - v_{k_2})^2}{2h_v}}) \gamma_{k_1}^{\text{old}} + e^{-\frac{(v_{k_1} - v_{k_2})^2}{2h_v}} \frac{\gamma_{k_1}^{\text{old}} + \gamma_{k_2}^{\text{old}}}{2} \quad (40)$$

$$\gamma_{k_2}^{\text{new}} = (1 - e^{-\frac{(v_{k_1} - v_{k_2})^2}{2h_v}}) \gamma_{k_2}^{\text{old}} + e^{-\frac{(v_{k_1} - v_{k_2})^2}{2h_v}} \frac{\gamma_{k_1}^{\text{old}} + \gamma_{k_2}^{\text{old}}}{2} \quad (41)$$

where  $h_v$  is a factor determining how fast the influence of particles should decrease with respect to their distance.

We call this procedure neighbour smoothing. In practice it is not essential to get the exact nearest neighbour. An approximate statistically unbiased nearest neighbour is good enough. In particular if the markers in one physical cell are distributed over several processors each having an equivalent particle distribution we use the local nearest neighbour on the processor. This makes the procedure computationally local and quite efficient especially for parallel programming.

If the velocity space has two dimensions, instead of sorting the particles according to their velocity, which is no longer possible, we group the particles within each cell in configuration space by a quad tree procedure in velocity space. This works by first setting a maximum number of particles per quad tree box. We took ten in our simulations. Then we subdivide recursively the two dimensional computational box in four sub-boxes until the number of particles in each box is less than the set maximum number. We then apply the above neighbour smoothing procedure to the pairs of consecutive particles in each box. The same idea could be applied using octrees in a three-dimensional velocity space.

By picking different pairs in each quad tree box, this procedure can be applied several times at each time step or what we did in our simulations only at some given interval of time steps. This needs to be tuned depending on the smoothing that is needed making sure not to make the method too diffusive.

**Remark 1.** *In our 1D simulations, we also tried out the quad tree smoothing by pairing the particles corresponding to their phase space position, but this yields very comparable results to the velocity smoothing procedure only. Therefore, we discarded it as it is more expensive, especially in higher dimensions. This can be easily understood because the particles move around in the cells in physical space due to the advection step and are thus paired with different particles at each step, which automatically implies a smoothing in configuration space also.*

**Remark 2.** We designed the smoothing procedure as a complement to the stochastic diffusion operator for the weighted PIC method, which has the effect of randomly mixing particle velocities. This destroys the smooth variation of the weights according to their phase-space position. Our smoothing operator has the effect of restoring this smooth variation as we will see in the numerical results.

Note that the smooth variation of the weights can also be destroyed by non linear effects, like phase mixing. In such cases our smoothing procedure can also be very helpful for non collisional simulations.

**Remark 3.** Disregarding the influence of the distance, i.e. setting  $h_v = +\infty$  the smoothing formula is obtained by minimising  $(\gamma_{k_1}^{\text{new}})^2 + (\gamma_{k_2}^{\text{new}})^2$  under the constraint that  $\gamma_{k_1}^{\text{new}} + \gamma_{k_2}^{\text{new}} = \gamma_{k_1}^{\text{old}} + \gamma_{k_2}^{\text{old}}$ . By taking three close particles we could also perform the minimisation under the additional constraint that the total momentum of the three particles is conserved, and by taking four particles we could also add the constraint of kinetic energy conservation. However, as the Lenard-Berstein collision operator we are considering in this paper is neither momentum nor kinetic energy conserving, we did not implement this.

## 8 The weight growth problem

As discussed by Kleiber et al. in [8], even in the simplest case of a 1D linear Fokker-Planck equation in velocity space the variance of the weights grows severely in time for the classical two weight scheme, i.e. using Eq. (28) with a general control variate  $f^0$  that is not an equilibrium of the collision operator, with integration of a weight equation.

This translates in a larger error in the approximation of the distribution function, to the point that after some time the error of the  $\delta f$  method becomes way larger than the error of the standard full- $f$  PIC method (see Figure 1). The solution introduced in [8] was to use an adaptive control variate method that falls back to the standard PIC method when the weight distribution is such that the numerical control variate becomes too decorrelated.

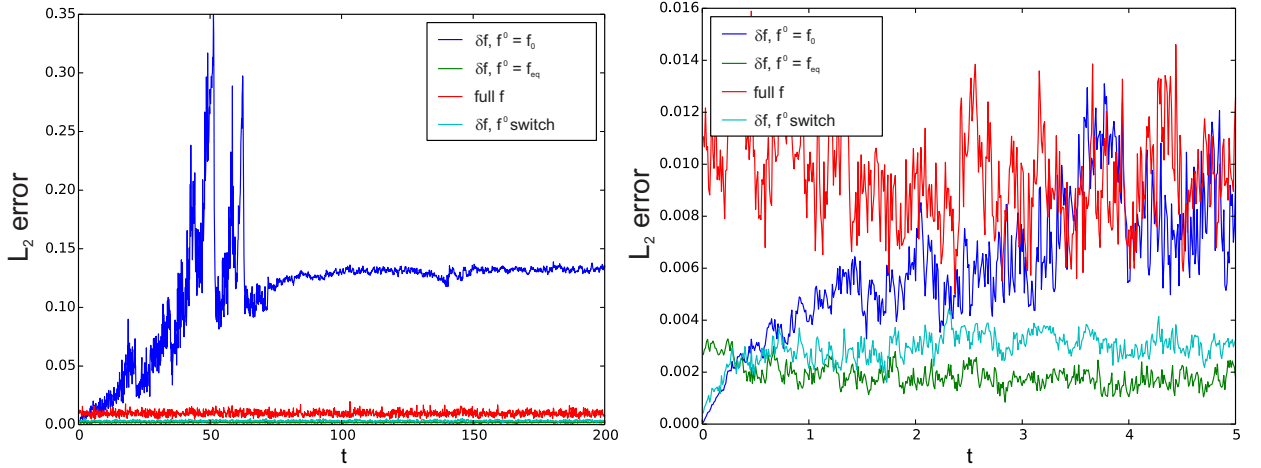


Figure 1:  $L^2$  error on the distribution function for the standard PIC method (red curve), the  $\delta f$  method with  $f_0$  as a control variate (blue curve), the  $\delta f$  method with  $f_{eq}$  as a control variate (green curve), and the  $\delta f$  method with rescaling (cyan curve). Shown are runs up to time 200 on the left and a zoom up to time 5 on the right.

There is a weight growth inherent to the fact that two probability densities, here  $f$  and  $f^0$  are transported with the same markers. This is a classical problem in large deviation methods in statistics, see [14] for example, or also [1] in the context of the Boltzmann equation. This becomes pathological for collisions with the classical two weight scheme.

The problem is removed when using the equilibrium function of the collision operator as a control variate for the collision step and using our simple rescaling procedure when another control variate is more efficient. For the simple Ornstein-Uhlenbeck process, the effectiveness of this idea on the evolution of the error is illustrated in Figure 1, where we see that the error is constant in time when  $f^0 = f_{\text{eq}}$ , which is defined below.

To further investigate the pathological behaviour of the two weight scheme we can analytically compute the expected value  $\mathbb{E}(\log \Gamma_2(t))$  in the case of the Ornstein-Uhlenbeck process for which we know analytically the evolution of the probability density, we assume  $f = g$  in our computations even though they could also be performed in a more general setting.

We consider the following Ornstein-Uhlenbeck process

$$dV(t) = -\mu V dt + D dW(t), \quad (42)$$

the evolution of its density is governed by the 1D Fokker-Planck equation

$$\frac{\partial f}{\partial t} - \mu \frac{\partial(vf)}{\partial v} - \frac{1}{2} D^2 \frac{\partial^2 f}{\partial v^2} = 0. \quad (43)$$

Denoting by  $K(t) = \frac{D^2}{2\mu}(1 - e^{-2\mu t})$ , an analytical solution is

$$f(t, v) = \frac{1}{\sqrt{2\pi(K(t) + \sigma^2 e^{-2\mu t})}} e^{-\frac{v^2}{2(K(t) + \sigma^2 e^{-2\mu t})}}, \quad (44)$$

which for  $t \rightarrow +\infty$  gives the stationary solution

$$f_{\text{eq}} = \sqrt{\frac{\mu}{\pi D^2}} e^{-\frac{\mu v^2}{D^2}}. \quad (45)$$

In this case the evolution of  $\Gamma_2$  is governed by

$$\frac{d\Gamma_2}{dt} = -\frac{\Gamma_2}{f^0} \left[ \mu \frac{\partial(vf^0)}{\partial v} + \frac{1}{2} D^2 \frac{\partial^2 f^0}{\partial v^2} \right]_{v=V}. \quad (46)$$

Taking  $f^0 = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{v^2}{2\sigma^2}}$  this becomes

$$\frac{d}{dt} \log \Gamma_2(t, V) = G(V) = \left( \mu - \frac{D^2}{2\sigma^2} \right) \left( \frac{V^2}{\sigma^2} - 1 \right) \quad (47)$$

so that

$$\frac{d}{dt} \mathbb{E}(\log \Gamma_2(t)) = \mathbb{E}(G(V)) = \int_{-\infty}^{\infty} G(v) f(t, v) dv. \quad (48)$$

After some straightforward computations, assuming that  $f^0 = f_0$  so that the initial weights are one, which implies  $\mathbb{E}(\log \Gamma_2(0)) = 0$ , we obtain a closed formula for the expected value of  $\Gamma_2$ :

$$\mathbb{E} \left( \log \Gamma_2(t) \right) = -\frac{1}{2\mu^2} \left( \mu - \frac{D^2}{2\sigma^2} \right)^2 (\exp(-2\mu t) + 2\mu t - 1). \quad (49)$$

We can see from this formula that

$$\lim_{t \rightarrow \infty} \mathbb{E}(\log \Gamma_2(t)) = -\infty.$$

This is an analytic way to express the weight spreading which is typically observed in numerical simulations of the two weight  $\delta f$  PIC method for collisions. We have analytically both  $\mathbb{E}(\Gamma_2(t)) = 1$  for all times and  $\mathbb{E}(\log \Gamma_2(t)) \rightarrow -\infty$ . In practice this means that the time evolution will create an decreasing number of increasingly large  $\Gamma_2$  weights and the others will tend to zero.

If the control variate  $f^0$  is taken to be an equilibrium solution of the collision operator, then the right hand side of the SDE determining  $\Gamma_2$  vanishes and consequently  $\Gamma_2$  is constant in time. In this case, as  $f^0$  is a steady state solution of the original equation, the three distributions  $g$ ,  $f$  and  $f^0$  are transported with the same stochastic differential equation and no extended phase space is necessary,  $\Gamma_2$  being determined at the initial time, like  $\Gamma$ , the weight for  $f$ .

In order to illustrate this weight growth issue, we solved numerically the Fokker-Planck equation with the initial condition as the centered standard normal distribution with variance one, *i.e.*  $f_0(v) = 1/\sqrt{2\pi}e^{-v^2/2}$ . We took  $D = 2$ ,  $\mu = 1.5$ , so that  $f_{\text{eq}}$  has variance  $D^2/(2\mu) = 4/3$ . Like in the analytical calculation, as a control variate we took  $f^0 = f_0$  the initial condition, as is often done in  $\delta f$  PIC codes. The markers are initialised with the initial distribution, *i.e.*  $g(t=0) = f_0$ , so that the marker distribution is equal to the particle distribution in a standard PIC code. The Ornstein-Uhlenbeck process was solved using the known analytical transition probability to update the velocity  $v_k^n$  at time  $t_n$  to  $v_k^{n+1}$  at time  $t_{n+1}$  and then the weights were updated using a left rectangle rule on Eq. (47), so that  $\gamma_{2,k}^{n+1} = \gamma_{2,k}^n \exp(G(v_k^n)\Delta t)$ . A more accurate quadrature rule has no influence on the qualitative behaviour of the weights we want to investigate here.

The long time evolution of the sample estimation  $\frac{1}{N} \sum_{k=0}^N \gamma_{2,k}(t)$  of  $\mathbb{E}(\Gamma_2(t))$  is displayed in Figure 2, where we see, that it saturates at zero after a long time even though theoretically it should always be one. This can be explained by the fact that the probability of having a non zero weight tends to zero with time, and in a finite sample there will quickly be too few markers with non zero weight left in order to give a good approximation of the expected value. We do not display  $\mathbb{E}(\log \Gamma_2(t))$  which is very close to the theoretical value given in Eq. (49) and decreases linearly in time to  $-\infty$ . This translates into the long term evolution of the  $L^2$  error on  $f$  represented in Figure 1, with a saturation after some time. Note that this corresponds to an analytical integration of the stochastic process. A numerical integration with an Euler-Maruyama scheme yields an even larger error that converges towards this value for small enough time steps. On the other hand, we also see in Figure 2 that computing the evolution of the weights for an arbitrary control variate using a rescaling between the control variate and the equilibrium function gives the correct behaviour even for long times with the sample expectation always staying close to one.

The evolution of the weights is displayed in Figure 3. In the top row, corresponding to the classical two weight scheme, we observe that some weights corresponding to small velocities grow exponentially, which leads to the fast increase of the error that we observed in Figure 1. That is, if  $\mu - D^2/(2\sigma^2) \neq 0$  (which corresponds to the case when  $f^0$  is not equal to the stationary solution), the method will result in weight spreading with error growth as a result. On the other hand, we see in the bottom row of Figure 3 that the weights stay bounded with the rescaling procedure.

When the full distribution function solution of the Vlasov-Fokker-Planck equation stays close enough to the equilibrium of the collision operator which is the local Maxwellian, this should be used as a control variate, in order to remove the intrinsic weight growth associated with distributions not evolved with the same transition probability. Else, when some other function, typically the initial condition is closer to the exact value of the solution, the rescaling method can be used to avoid the generation of very large weights.

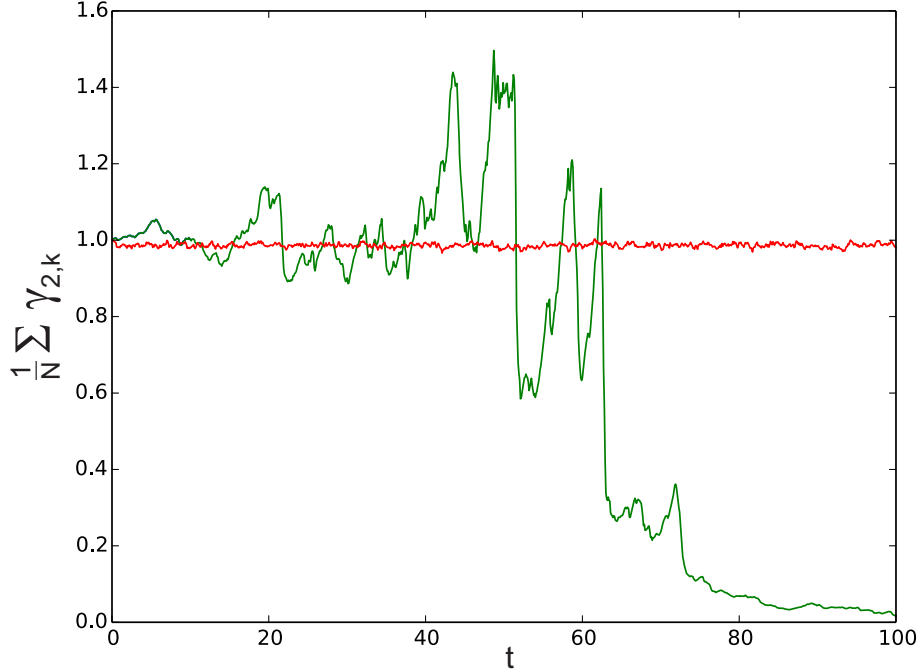


Figure 2: Evolution of sample expected value of  $\Gamma_2$  in time. With  $f_0$  as control variate with the two weight scheme (green curve) and with rescaling (red curve).

## 9 Numerical validation

### 9.1 The 1D linear Fokker-Planck equation

In the following we again use the same model as in the previous section *i.e.* the Ornstein-Uhlenbeck process (42) associated to the Fokker-Planck equation (43) with the same initial condition as in Section 8. We also take  $D = 2$ . The only difference is that here  $\mu = 1$  is used, so that the initial distribution is further away from the equilibrium distribution which has now the variance  $D^2/(2\mu) = 2$  which makes the problem harder for the control variate scheme.

The marker distribution is taken equal to the particle distribution, *i.e.*  $g$  is initialised with  $g = f_0$ . The time step is  $\Delta t = 0.1$ .  $N = 1000$  markers are used in the simulation. In order not to be dependent on different samplings, we start all the comparison runs with the same seed of the random generator, so that the runs are exactly reproducible with the same parameters.

We first check that the neighbour smoothing has the desirable property of bringing the distribution function as close as desired to the equilibrium, *i.e.* that the  $\Gamma_1$  weights converge to zero. Note that this smoothing procedure conserves exactly the mean, which will get closer to zero when increasing the number of particles but will not be exactly zero. So if we want the  $\Gamma_1$  weights to converge to zero we need to make sure that the mean is zero, which is done in our algorithm at each time step by the projection given by Eq. (39). In Figure 4 we represent the evolution of the  $L^2$  error for different smoothing frequencies. With no smoothing the error stays approximately constant and with increased smoothing frequency it decreases going to zero when the equilibrium is reached as expected.

In order to understand the effects of the smoothing procedure on the weights, we display the weights for each marker velocity in Figure 5, with and without smoothing. Without smoothing

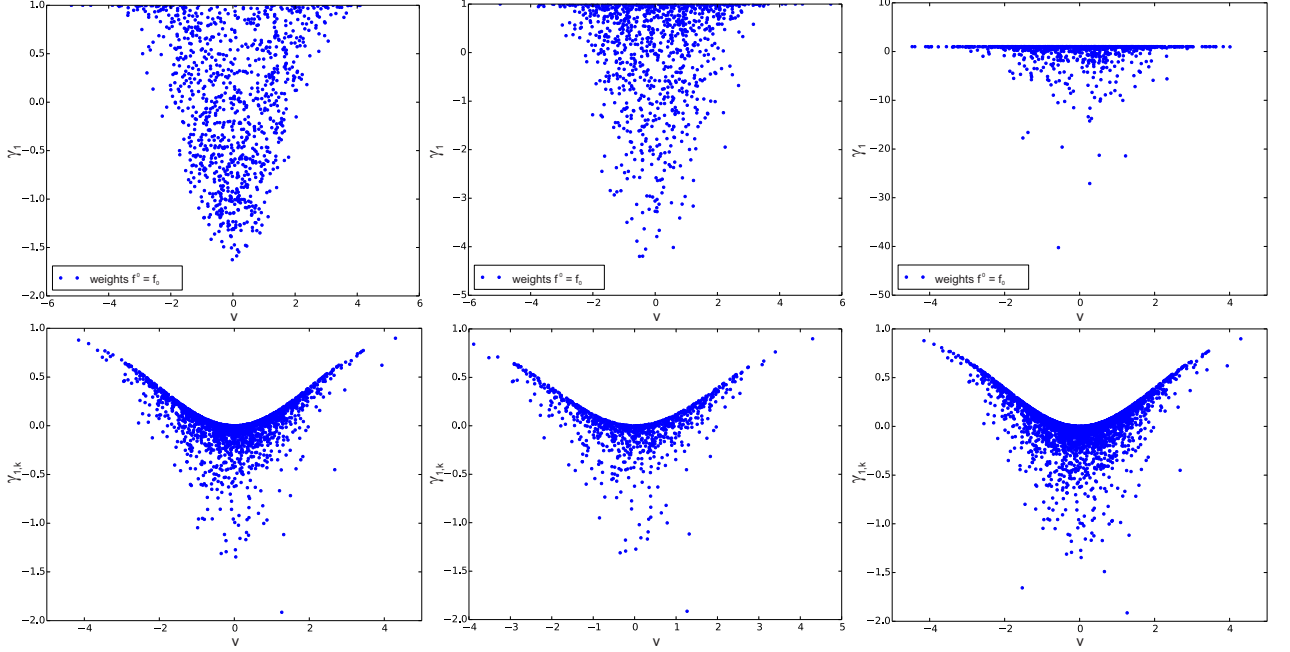


Figure 3: Position of  $\gamma_1$  weights with respect to velocity after 10 (left), 20 (middle) and 50 (right) time steps with  $f_0$  as control variate with the two weight scheme (top) and rescaling (bottom).

the weights stay constant and are just moved around in velocity space with their markers. This allows to converge statistically towards the correct equilibrium, but the noise stays important even when the equilibrium is reached. On the other hand, when smoothing is turned on (at each time step here), the variance of the weights decreases and we see in Figure 6 that the distribution of the weights stays close to the  $\delta f(v)/g(v)$  curve, which is obtained using a kernel density estimate for both distributions. This is not the case without smoothing.

## 9.2 The 1D Vlasov Fokker-Planck equation

We consider in this section the following Vlasov Fokker-Planck equation in a two-dimensional phase space, periodic of period one in  $x$

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E \frac{\partial f}{\partial v} = \nu \left( \mu \frac{\partial(vf)}{\partial v} + \frac{D^2}{2} \frac{\partial^2 f}{\partial v^2} \right), \quad (50)$$

where the electric field derives from a potential  $E = -\frac{\partial \phi}{\partial x}$  which is the solution of the Poisson equation

$$-\lambda^2 \frac{\partial^2 \phi}{\partial x^2} = \rho = 1 - \int f dv. \quad (51)$$

The initial condition is of perturbed bump-on-tail type

$$f_0(x, v) = (1 + \epsilon \cos(2\pi x)) \frac{1}{1+a} \left( \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} + \frac{a}{\sqrt{2\pi}\sigma} e^{-\frac{(v-v_0)^2}{2\sigma^2}} \right), \quad (52)$$

with  $\epsilon = 10^{-3}$ ,  $a = 0.04$ ,  $v_0 = 4$ ,  $\sigma = 0.5$ . The parameter  $\lambda$  in the Poisson equation is a normalisation parameter related to the Debye length and is taken to be  $\lambda = 15$ . The time step is



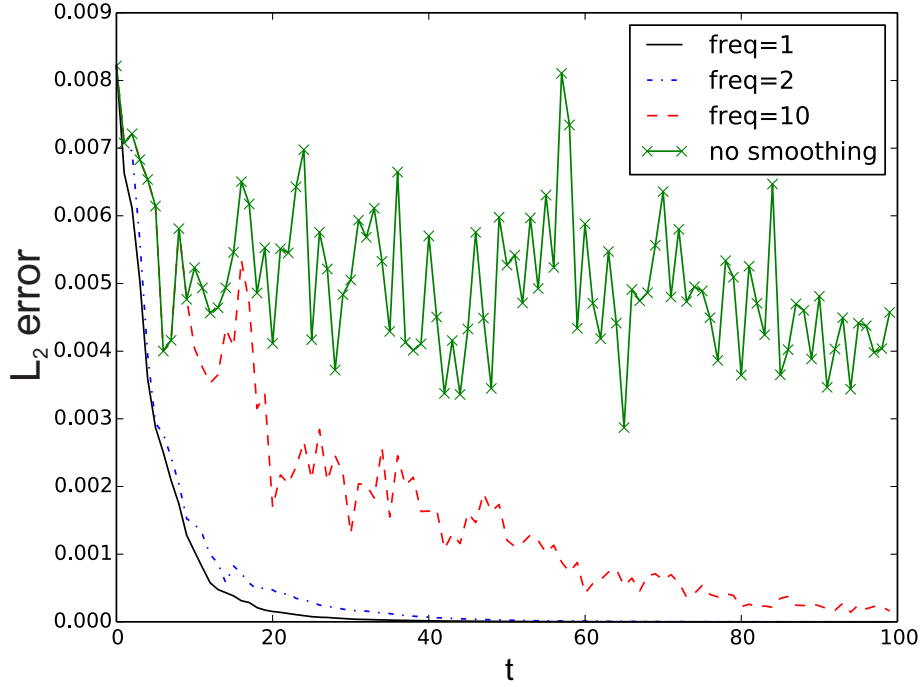


Figure 4:  $L^2$  error on the distribution function for neighbour smoothing with different smoothing frequencies, **freq** stands for the number of time steps between two applications of the smoothing mechanism.

taken as  $\Delta t = 0.01$ . The number of cells in  $x$  is 32 and the total number of particles is 32 000, so that there are around 1000 particles in each cell.

Without collisions, the fundamental mode is linearly unstable with a growth rate of 0.86377.

### 9.2.1 No collisions

We consider here the problem without collisions, that is  $\nu = 0$ . Our problem is then the standard bump-on-tail problem for the Vlasov equation and we can qualitatively compare our results with [12]. As is standard for  $\delta f$  simulations without collisions we use here the initial condition  $f_0$  as a control variate. No smoothing is performed.

We first check in the linear phase of the simulation that the growth rate on the left part of Figure 7 corresponds to the analytical value. On the right part of Figure 7 we see that first the weights grow according to the evolution of the distribution function and then oscillate around a constant value with a light growth on a long time scale.

Then, in Figure 8 we monitor the long time evolution of the potential energy, the total energy and the total momentum. The potential energy keeps oscillating after the initial phase at an almost constant value, with some noise appearing at later times. The total energy and momentum are conserved quantities of the Vlasov-Poisson equations. In our simulation they are conserved to around 1% for the total energy and 0.4% for the total momentum.

### 9.2.2 Strong collisions

Here we take  $\nu = 1$ ,  $D = \sqrt{2}$  and  $\mu = 1$  which corresponds to a centered equilibrium Maxwellian of variance one of the collision operator. As the distribution function relaxes quickly towards the

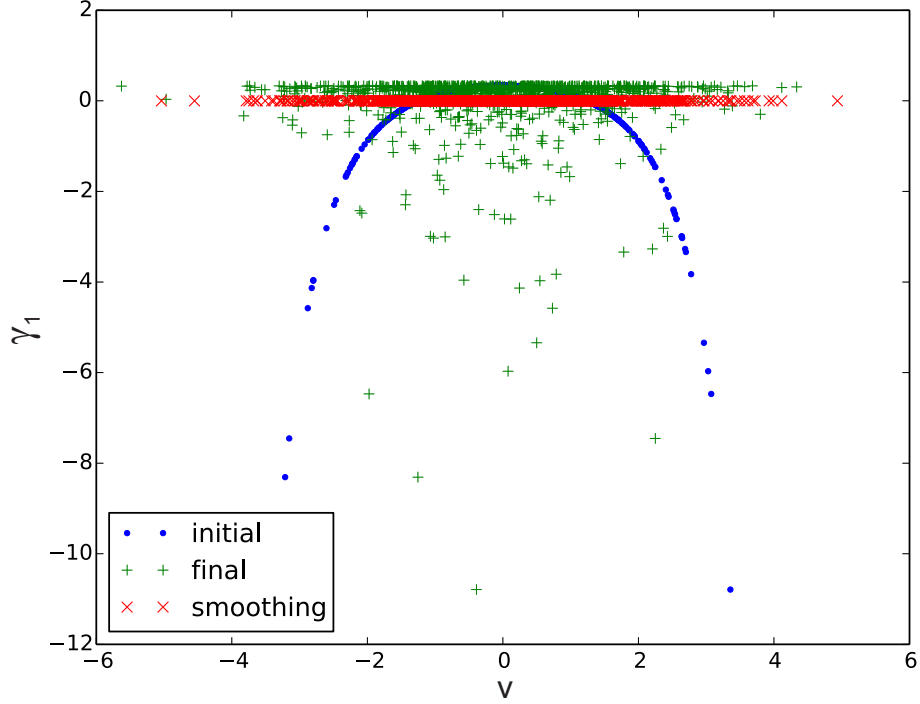


Figure 5: Value of the weights with respect to the velocity with and without smoothing

equilibrium, we use  $f_{\text{eq}}$  of the Ornstein-Uhlenbeck process as a control variate without switching.

We compare the solution with smoothing at every time step and with no smoothing. In Figure 9 we observe that the amplitude of the Fourier modes oscillates around a constant value without smoothing, whereas with smoothing they are strongly damped as is expected from a collision operator. Note that in this case the bump-on-tail instability is suppressed because of the strong collisions.

In Figure 10 we represent the weights  $\gamma_1$  of the particles with respect to the particle position at time ten. We observe that the weights are randomly distributed all over phase space when no smoothing is applied, whereas they are smoothly distributed when our smoothing procedure is applied. This results in the evolution of the sum of the  $\gamma_1$  squared weights seen in Figure 11, which shows a slow weight growth without the smoothing procedure and a damping of all the weights to zero with the smoothing procedure, as is the correct behaviour with the collision operator.

In Figure 12 we verify that the kinetic energy is drawn to 0.5 which is half the value of the variance of the equilibrium distribution and the total momentum approaches zero, which is the mean of the equilibrium distribution.

### 9.2.3 Weak collisions

Here we take  $D = \sqrt{2}$ ,  $\mu = 1$  and  $\nu = 10^{-2}$  which also corresponds to a centered equilibrium Maxwellian of variance one of the collision operator but with a collision frequency a hundred times weaker than in the previous subsection. In this case it appears best to switch the control variate between the initial condition for the advective part and the equilibrium function for the collisions part. We have performed convergence tests here for the number of particles to highlight how the weight smoothing technique can help reducing the number of needed particles in the simulation.

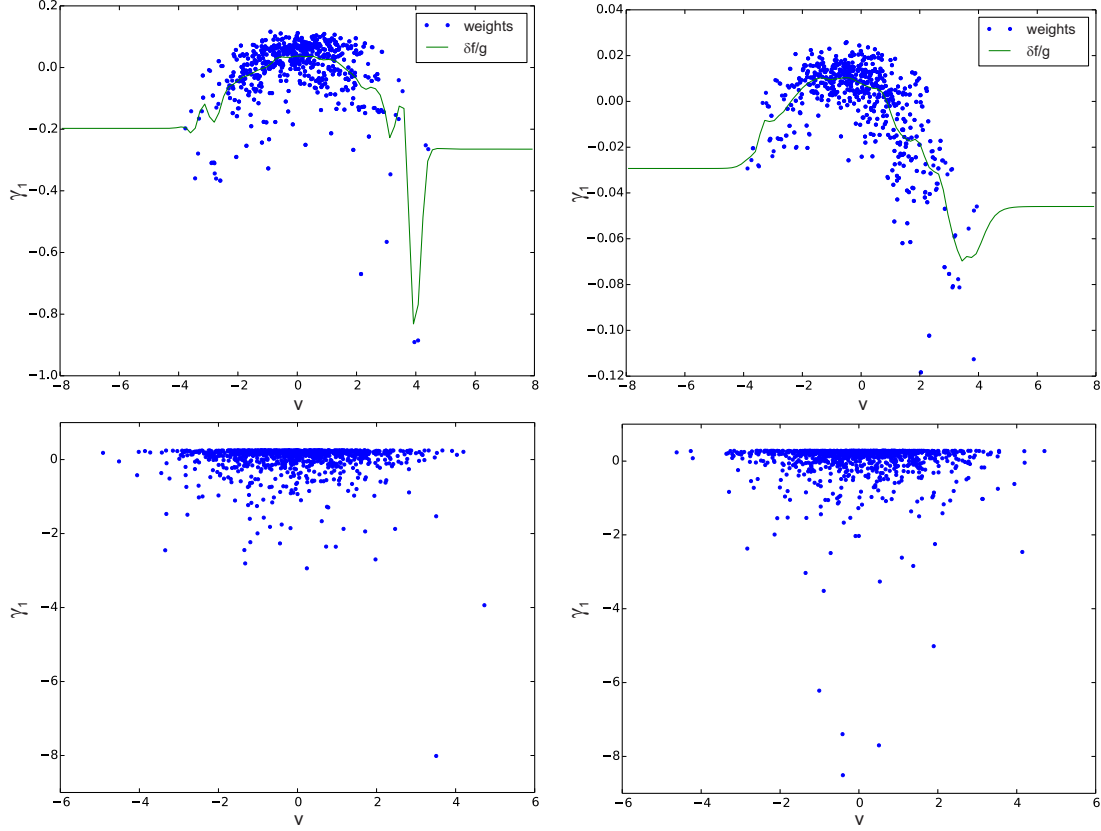


Figure 6: Value of  $\gamma_1$  weights with respect to velocity after 10 (left) and 20 (right) time steps. Top row: using  $f^0 = f_{eq}$  and smoothing ( $\delta f/g$  is also plotted). Bottom row: using  $f^0 = f_{eq}$  but without smoothing.

In Figure 13, we see that the instability can still be observed on the Fourier modes with a slight damping of the fundamental mode in the linear phase and lower saturated amplitude in the nonlinear phase. We also observe that the Fourier modes are better converged when smoothing is used, problems being apparent without smoothing when too few particles are used. The same conclusion holds for the potential energy displayed in Figure 14, where the qualitative behaviour is identical for all particle numbers shown when smoothing is performed while there are problems for low particle numbers without smoothing. Finally in Figure 15, we show the weight growth. Even though there is no pathological weight growth in our method, the weights grow slightly in time and this is essentially independent of the number of particles used. This problem is inherent to the fact that two different distributions are evolved with the same markers. This weight growth is suppressed with our weight smoothing procedure. After a small growth during the linear instability phase due to the fact that the distribution function departs from the initial condition which is used as a control variate, the weight growth stops when the distribution function reaches a dynamical almost steady state and the weights will decrease when going slowly to the Maxwell equilibrium. However, the amount of smoothing needs to be carefully tuned to avoid too much diffusion. This is done by ensuring that the growth rate of the initial instability is reproduced. The more particles are present in a simulation, the less smoothing is needed. We tune the amount of diffusion by changing the frequency of application of the weight smoothing procedure. This explains, the different jitter that is seen in Figure 15. The sum of the squared weights being decreased each time the smoothing

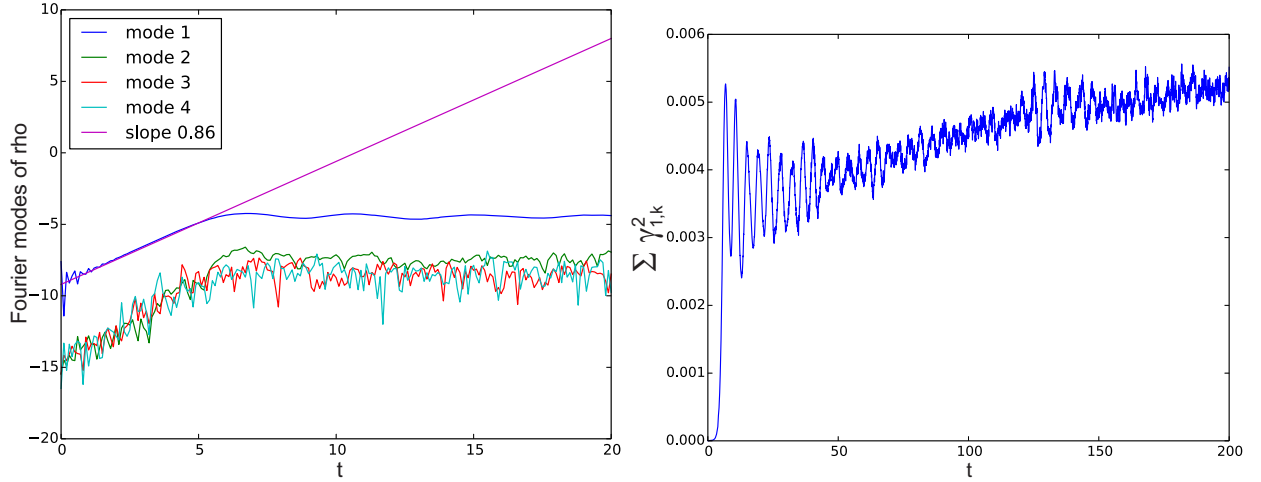


Figure 7: Bump on tail problem with no collisions. Evolution of Fourier modes (left) and sum of squared weights (right).

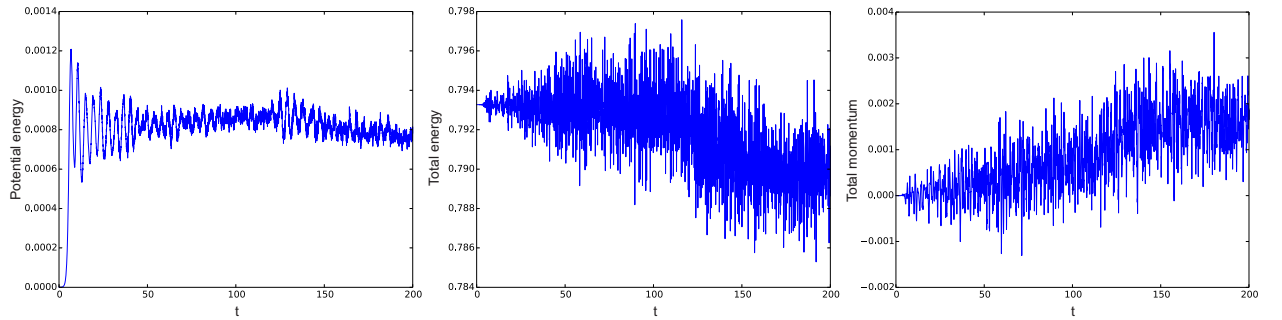


Figure 8: Bump on tail problem with no collisions. Long time evolution of the potential energy (left), the total energy (middle) and the total momentum (right).

procedure is applied. A smoother but costlier alternative would be to change the  $h_v$  parameter in Formulas (40)–(41).

## 10 Conclusion

We analysed the weight growth problem of the two weight scheme used in collisional  $\delta f$  PIC method in the simple case of an Ornstein-Uhlenbeck process. We could prove that the expected value of the log of the weights tends to minus infinity when time grows for the analytical solution of the process. Because of this, the probability that a weight will be larger than any strictly positive constant tends to zero. And as on the other hand the expected value stays constant at one, this means that there will be a decreasing number of extremely large weights compensating all the others that are close to zero. This completely destroys the statistics and makes the method inefficient for large time simulations. In addition, our analysis showed that the weights keep constant, when an equilibrium function of the collision operator is used as a control variate. This motivates the use of the equilibrium as a control variate in the case of collisions. If some other control variate is more efficient for the advection part, a simple rescaling procedure enables to switch between the two control variates. Although this procedure removes the weight growth problem of the two

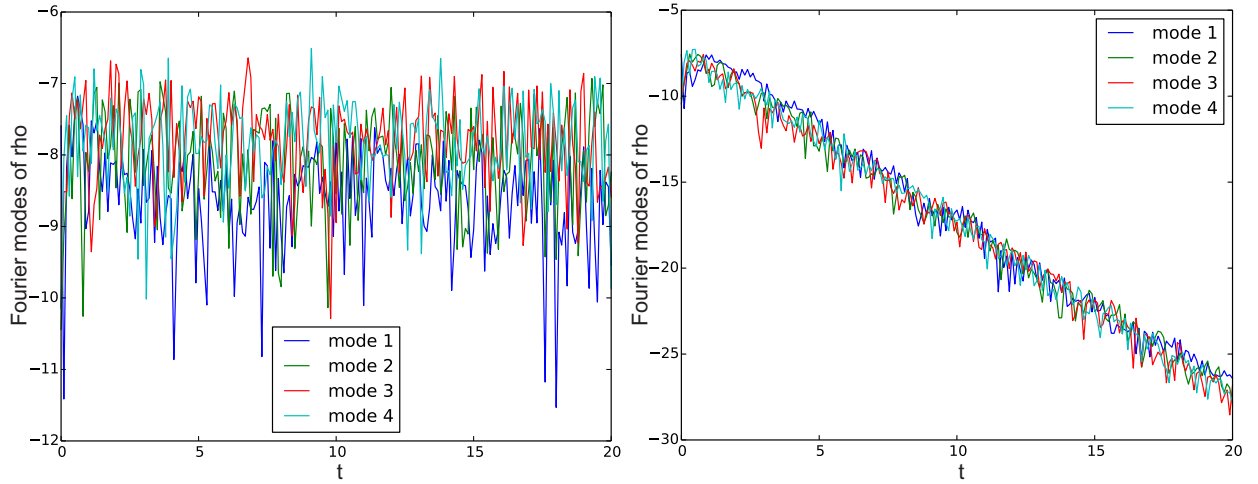


Figure 9: Strong collisions. Evolution of Fourier modes without smoothing (left) and with smoothing (right).

weight scheme, it does not decrease the individual weights when the distribution function tends to equilibrium albeit in each large enough bin their average goes to zero.

In order to get the weights to converge each to zero in this case, instead of only their average, we introduced a neighbour smoothing procedure, enabling neighbouring markers to average their weights. Thanks to this additional idea, each single weight indeed tends to zero due to the collision operator, which is the desired behaviour. We verified the efficiency of our new algorithm in the 1D velocity space first and then for a 1D Vlasov-Fokker-Planck problem. This new collision procedure enables us now to heat or cool a plasma by choosing appropriate parameter values for the Ornstein-Uhlenbeck process and also to use the local Maxwellian as an adaptive control variate in a collision scheme that conserves mass, momentum and energy. Even though the smoothing operator has been designed to be used with collisions it can also be used to smooth weights that have been mixed over time due to nonlinear effects in collisionless simulations, with the effect of decreasing the noise by reducing the variance.

First tests of the new scheme implemented by Alberto Bottino in the NEMORB gyrokinetic code show very promising results, but this deserves an exhaustive evaluation which will be addressed in a forthcoming paper.

## Acknowledgements

The ONEDIM program was used as starting point for most of the simulations. We would like to thank A. Könies as one of the developers for providing us access to the program. In addition, we would like to thank A. Bottino for discussions and proof reading of the manuscript.

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement number 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

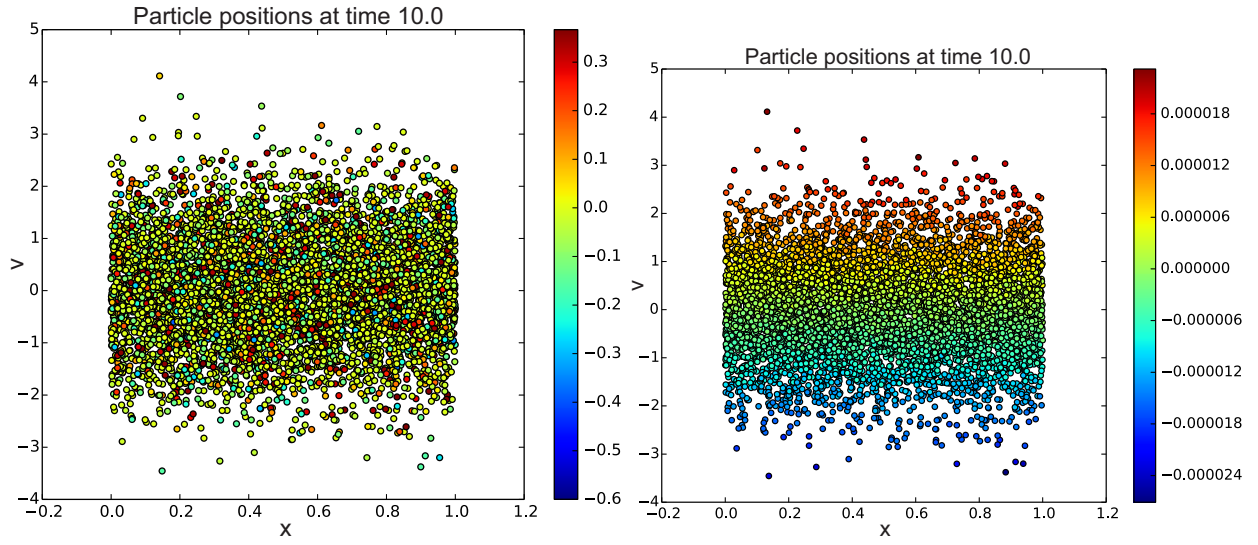


Figure 10: Strong collisions. Particle positions at time ten coloured by weight without smoothing (left) and with smoothing (right).

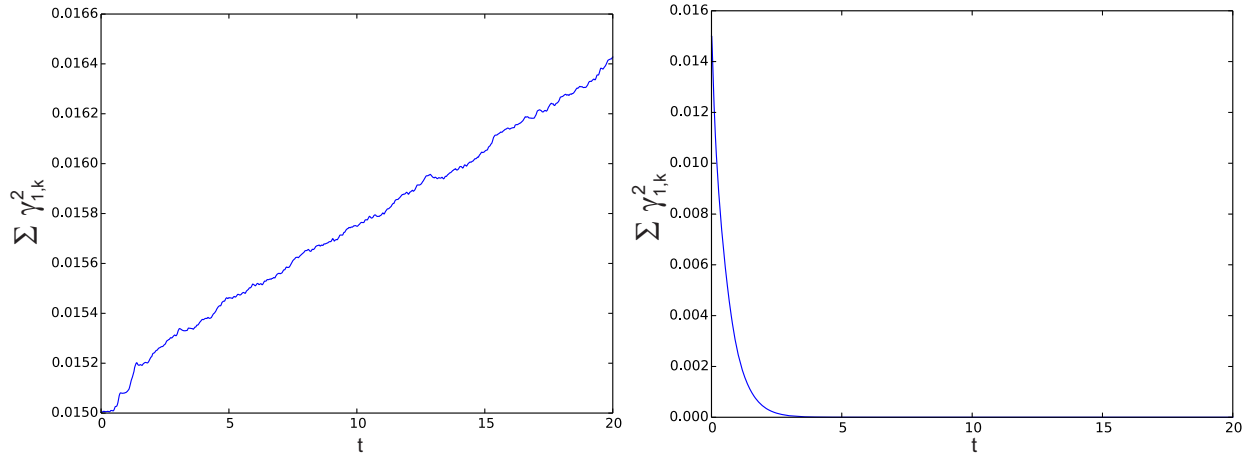


Figure 11: Strong collisions. Evolution of weights without smoothing (left) and with smoothing (right).

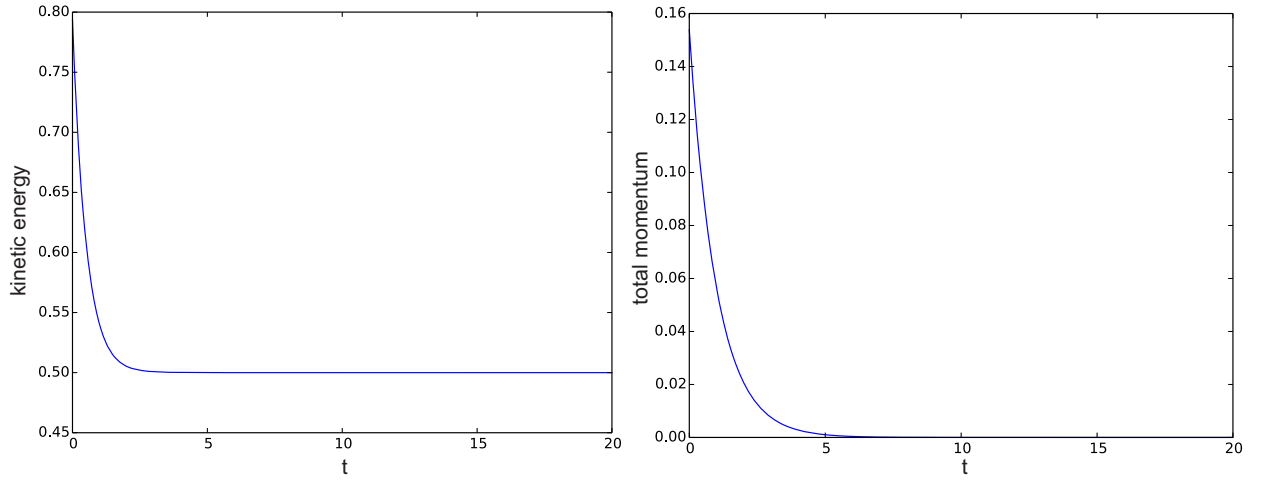


Figure 12: Strong collisions. Evolution of kinetic energy (left) and total momentum (right) with smoothing.

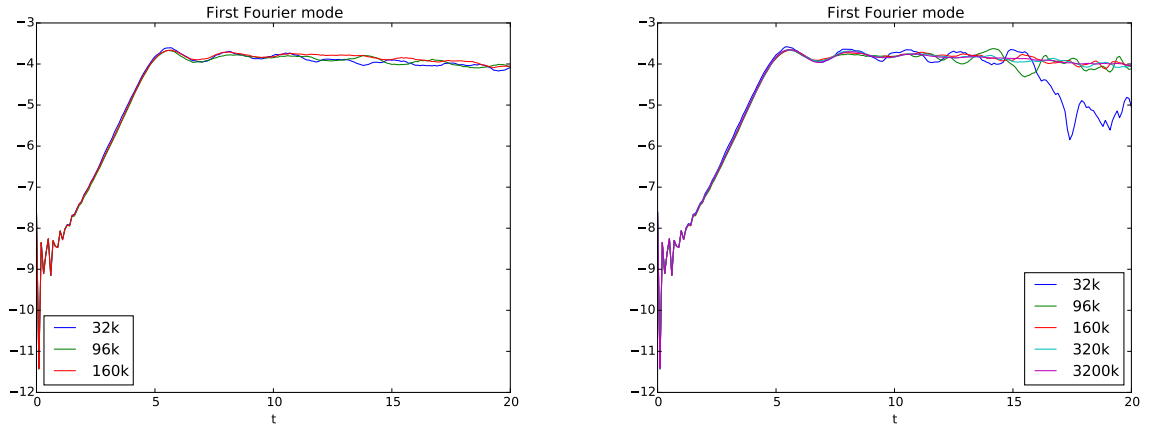


Figure 13: Weak collisions. Evolution of Fourier modes with smoothing (left) and without smoothing (right).

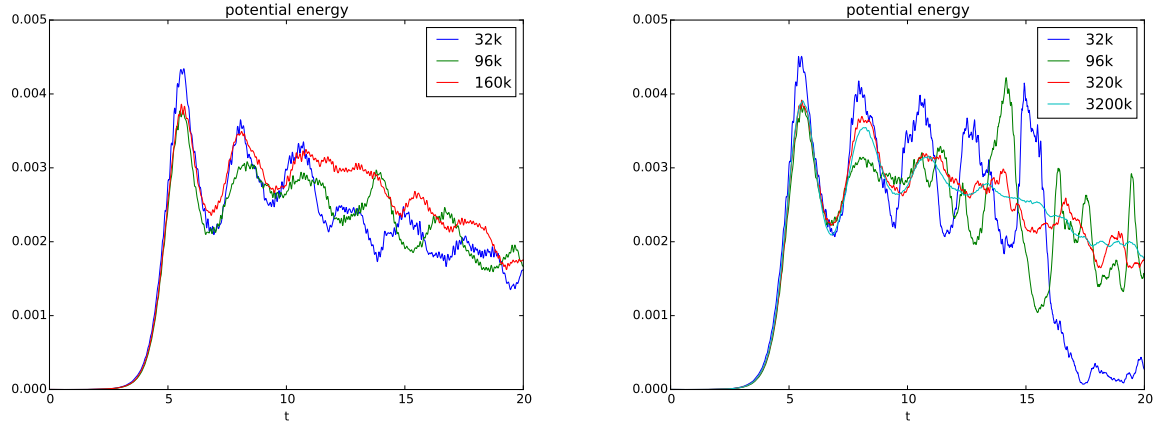


Figure 14: Weak collisions. Evolution of potential energy with smoothing (left) and without smoothing (right).

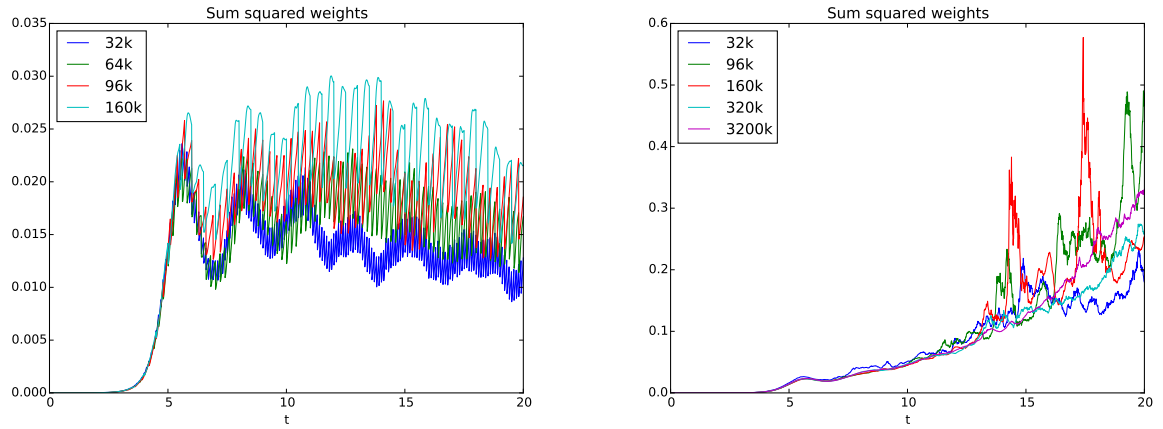


Figure 15: Weak collisions. Evolution of squared weights with smoothing (left) and without smoothing (right).



## References

- [1] Husain A. Al-Mohssen and Nicolas G. Hadjiconstantinou. Low-variance direct Monte Carlo simulations using importance weights. *ESAIM: Math. Model. Numer. Anal.*, 44:1069–1083, 2010.
- [2] Simon J. Allfrey and Roman Hatzky. A revised  $\delta f$  algorithm for nonlinear PIC simulation. *Computer physics communications*, 154(2):98–104, 2003.
- [3] Ahmet Y. Aydemir. A unified monte carlo interpretation of particle simulations and applications to non-neutral plasmas. *Physics of Plasmas*, 1(4):822–831, 1994.
- [4] Charles K. Birdsall and A. Bruce Langdon. *Plasma physics via computer simulation*. CRC Press, 2004.
- [5] Stephan Brunner, Ernest Valeo, and John A. Krommes. Collisional delta-f scheme with evolving background for transport time scale simulations. *Physics of Plasmas*, 6(12):4504, 1999.
- [6] Y. Chen and R. B. White. Collisional  $\delta f$  method. *Physics of Plasmas*, 4:3591, 1997.
- [7] Yang Chen and Scott E. Parker. Coarse-graining phase space in  $\delta f$  particle-in-cell simulations. *Physics of Plasmas*, 14(8):082301, 2007.
- [8] Ralf Kleiber, Roman Hatzky, Axel Könies, Karla Kauffmann, and Per Helander. An improved control-variate scheme for particle-in-cell simulations with collisions. *Comput. Phy. Comm.*, 182:1005–1012, 2011.
- [9] Peter E. Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer, 1992.
- [10] Mike Kotschenreuther. Numerical simulation. *Bull. Am. Phys. Soc*, 33:2107–2108, 1988.
- [11] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, 2002.
- [12] Takashi Nakamura and Takashi Yabe. Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov–Poisson equation in phase space. *Computer Physics Communications*, 120(2):122–154, 1999.
- [13] Bernt Øksendal. *Stochastic differential equations*. Universitext. Springer Berlin, Heidelberg, fifth edition, 2000.
- [14] Gerardo Rubino, Bruno Tuffin, et al. *Rare event simulation using Monte Carlo methods*. Wiley Online Library, 2009.
- [15] T. Vernay, S. Brunner, L. Villard, B.F. McMillan, S. Joliet, T.M. Tran, A. Bottino, and J.P. Graves. Neoclassical equilibria as starting point for global gyrokinetic microturbulence simulations. *Physics of Plasmas*, 17(12):122301, 2010.
- [16] Thibaut Vernay. *Collisions in global gyrokinetic simulations of tokamak plasmas using the  $\delta f$  Particle-In-Cell approach: neoclassical physics and turbulent transport*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2012.