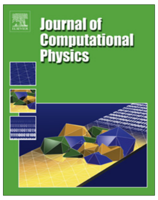
Journal of Computational Physics 242 (2013) 561–580

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp



Higher-order time integration of Coulomb collisions in a plasma using Langevin equations

A.M. Dimits [a,*], B.I. Cohen [a], R.E. Caflisch [b], M.S. Rosin [b], L.F. Ricketson [b]

[a] Lawrence Livermore National Laboratory, L-637, P.O. Box 808, Livermore, CA 94511-0808, USA
[b] Mathematics Department, University of California at Los Angeles, Los Angeles, CA 90036, USA

ARTICLE INFO

Article history:
Received 31 August 2012
Accepted 26 January 2013
Available online 8 February 2013

Keywords:
Monte-Carlo methods
Milstein method
Collision processes
Plasmas
Collisions
Computer applications

ABSTRACT

The extension of Langevin-equation Monte-Carlo algorithms for Coulomb collisions from the conventional Euler–Maruyama time integration to the next higher order of accuracy, the Milstein scheme, has been developed, implemented, and tested. This extension proceeds via a formulation of the angular scattering directly as stochastic differential equations in the fixed-frame spherical-coordinate velocity variables. Results from the numerical implementation show the expected improvement [$O(\Delta t)$ vs. $O(\Delta t^{1/2})$] in the strong convergence rate both for the speed $|\mathbf{v}|$ and angular components of the scattering. An important result is that this improved convergence is achieved for the angular component of the scattering if and only if the “area-integral” terms in the Milstein scheme are included. The resulting Milstein scheme is of value as a step towards algorithms with both improved accuracy and efficiency. These include both algorithms with improved convergence in the averages (weak convergence) and multi-time-level schemes. The latter have been shown to give a greatly reduced cost for a given overall error level when compared with conventional Monte-Carlo schemes, and their performance is improved considerably when the Milstein algorithm is used for the underlying time advance versus the Euler–Maruyama algorithm. A new method for sampling the area integrals is given which is a simplification of an earlier direct method and which retains high accuracy. This method, while being useful in its own right because of its relative simplicity, is also expected to considerably reduce the computational requirements for the direct conditional sampling of the area integrals that is needed for adaptive strong integration.

1. Introduction

In this paper, we describe the application of higher-order stochastic-differential-equation (SDE) numerical integration methods to the problem of Coulomb collisions in a dilute (classical weakly coupled) plasma.

Coulomb collisions arise and can dictate or strongly affect the behavior of many plasma systems. Examples arise in magnetic fusion (MFE) [1], inertial fusion (ICF) [2], plasma processing [3], and near-earth (or planetary) space plasma [4]. There has been a long history of study of Coulomb collisions in plasmas. In his pioneering work, Landau [5] recognized that the action of the Coulomb collisions on the velocity of a charged particle in a plasma is dominated by many small-angle collision events, i.e., that large-angle events are so unlikely as to be subdominant in their effect, and that the relevant expression to describe their effect on the plasma distribution functions is therefore a drag-diffusion (“Fokker–Planck”) term rather than a

---

* Corresponding author. Tel.: +1 925 422 0211; fax: +1 925 423 3484.
E-mail address: dimits1@llnl.gov (A.M. Dimits).

http://dx.doi.org/10.1016/j.jcp.2013.01.038

Boltzmann term. Landau's collision term gives the rate of change of the phase-space distribution function (density) $f_\alpha(\mathbf{x},\mathbf{v})$ of plasma charged-particle species α (which could be electrons or an ion species) as

$$\left.\frac{\partial f_\alpha}{\partial t}\right|_{\text{coll}} = \frac{\partial}{\partial \mathbf{v}}\cdot\left[\pi q_\alpha^2 \lambda \sum_\beta q_\beta^2 \int d^3\mathbf{v}'\left(f_\alpha \frac{\partial f'_\beta}{\partial \mathbf{v}'} - f'_\beta \frac{\partial f_\alpha}{\partial \mathbf{v}}\right)\frac{(u^2\mathbf{I}-\mathbf{uu})}{u^3}\right], \qquad (1)$$

where the sum is over the index β of the plasma charged-particle species, $\mathbf{x}$ and $\mathbf{v}$ are the position and velocity, t is the time, $f_\alpha \equiv f_\alpha(\mathbf{x},\mathbf{v})$, q_β is the charge of species β, $f'_\beta \equiv f_\beta(\mathbf{x},\mathbf{v}')$, $\mathbf{u} = \mathbf{v} - \mathbf{v}'$, $u = |\mathbf{u}|$, and λ is the "Coulomb logarithm". That Eq. (1) is in the form of a drag-diffusion equation is evident from the fact that f_α and $\partial f_\alpha/\partial\mathbf{v}$ are local in $\mathbf{x}$ and $\mathbf{v}$ and only derivatives up to the second derivative in $\mathbf{v}$ appear. Another important property of the right hand side of Eq. (1) is that it is in conservative (or "continuity") form, i.e., it is the (velocity-space) divergence of a flux. Eq. (1) contains explicit expressions for the drag and diffusion coefficients in terms of the distribution functions f_β of the plasma species, and forms the basis for (both analytical and computational) treatments of Coulomb collisions used to this day.

While analytical results are available for a wide variety of idealized problems [6], solutions often are needed for cases where a clean separation between the time scales associated with various competing effects, including Coulomb collisions, do not exist, and so these analytic results are not applicable. Unless the collisional time scales are much more rapid than the other time scales in the system, kinetic (i.e., non-hydrodynamic) phenomena may be important. In many cases where the collisional time scale for gross changes in the plasma distribution function is long, the plasma behavior is dictated by kinetic features that are localized in velocity space. Because of the diffusive nature of the Coulomb collisions, the rate at which they act on these kinetic features can become competetive with the rates of other effects. Kinetic numerical plasma simulation methods, including initial-value solutions for time-dependent problems and relaxation methods for time-independent problems, have therefore become widely used to study plasma phenomena. These simulation methods can be broadly grouped into particle-based, hybrid, and continuum-kinetic methods, and it is for the first two that our results are useful.

Particle-based kinetic plasma simulation methods include direct mesh-free methods in which the Coulomb interactions between pairs of particles are treated directly, particle-in-cell (PIC) and the related δf-PIC methods. The PIC methods can achieve great efficiency gains relative to the direct methods by mediating the Coulomb interactions through mesh quantities (densities, currents, electric and magnetic fields). Hybrid methods generally treat a plasma system as a combination of a part that is treated with a particle-based method with a part that can be treated as a fluid (i.e., is described by a small number of variables that depend on position and time, and not on the velocity). The kinetic continuum (or "Vlasov") methods evolve the distribution function (phase-space density) directly on a kinetic phase-space (i.e., position-velocity) mesh treating it essentially as a fluid in phase space. In the PIC and δf-PIC methods, the use of the mesh to mediate the interactions between the charged particles greatly reduces and alters the collisions, as represented by the system [7,8]. The collisions can be restored to physically relevant levels through the use of statistical "Monte-Carlo" (MC) methods. These utilize the fact that a drag-diffusion (partial) differential equation, such as Eq. (1), can be viewed as an equation for the expectation (mean) phase-space density of a group of "particles" whose velocities evolve according to stochastic (ordinary) differential equations, and which can be approximately solved with a discretization using a finite ensemble of such particles. For most problems of interest, this approach (combining PIC or δf methods with MC methods) is much more computationally efficient than direct pair-wise computation of the Coulomb interactions.

Various authors have developed Monte-Carlo discretizations of the Coulomb collision operator. One approach, upon which the present work based, is Langevin-equations based algorithms [9–12]. In this approach, each particle is treated as a test particle, acted on by drag and stochastic forces such that the expectation of the result of the latter is the appropriate diffusion term. The resulting operator can be made self-consistent through the inclusion of "field-particle" effects, which include (but are not limited to) enforcing momentum and energy conservation. This can be done via a cell-by-cell shift (boost) and rescaling of the particle velocities (in the case of a PIC algorithm) or via source/sink terms that manifest themselves as a change in the particle weights (in a δf-PIC method). Another approach is binary Monte-Carlo collision algorithms [13–15]. In these, the particles are scattered elastically in pairs, thereby giving quite accurate momentum and energy conservation for each scattering event. The particles are typically paired with other particles in the same spatial cell. Although the conservation is not perfect because the particles in a pair do not reside at identical spatial positions, the inaccuracy (spreading) can be controlled to be acceptably small. The scattering is done by a rotation of the relative velocity vector of the particles. These rotations are not those for individual physical Coulomb collisions at given impact parameters, but are chosen from a statistical ensemble so that, again, the expectation of the result matches the Landau operator. Coulomb collisions have also been introduced into continuum-kinetic methods [16–18].

Numerical Langevin-equations-based Coulomb collision treatments to date [9–12] have predominantly used the lowest-order temporal SDE discretization, the Euler–Maruyama method [19]. However, higher-order temporal SDE discretizations have been developed [19,20] and have found quite wide use, for example in financial applications and chemical physics. Going up in the hierarchy of temporal SDE discretization methods beyond the Euler–Maruyama method are the Milstein method, which achieves a higher order of strong convergence [19] and, beyond the Milstein scheme, class of schemes that achieve higher-order weak convergence [19]. In the present work, we extend the Langevin-equations-based treatment of Coulomb collisions to the next higher order of accuracy beyond the conventional Euler–Maruyama time integration, i.e., the Milstein scheme. The Milstein method is of interest for two reasons: (A) It is the first in a hierarchy of higher-order methods for SDE's, which also includes schemes with improved (higher-order) weak convergence. (B) When used instead of the

Euler–Maruyama scheme as the underlying building block of Giles' multi-(time-) level scheme [21], the resulting efficiency is significantly improved, i.e., the computational complexity for a given overall error is significantly reduced. Both the multilevel schemes [21] and the higher-weak-order schemes [19] have been shown to give improved computational efficiency over the Euler–Maruyama scheme for the numerical solution of SDE's, i.e., smaller error for a given computational cost or lower computational cost for a given error in the solution. A more detailed discussion of these errors and costs will be given at the end of Section 2.

The concept of strong convergence is central to understanding the properties of the Milstein scheme, but is perhaps less familiar in the plasma-physics literature than weak convergence, so a brief discussion of these, including the distinction between them is warranted. Strong convergence concerns the strong error, which can be defined as the expectation value (over paths) of the discretization error in each individual path. One specific choice of the discretization error, which we use in this paper, is the error at a specific time T. If $S(T)$ is the (possibly vector-valued) solution to a SDE evaluated at time T, and $\hat{S}_{T/h}$ is the value at time T given by a the solution of a temporal discretization of the SDE with time step h (and with the same initial condition, i.e., $\hat{S}_{0/h} = S(0)$), then the definition that will be used here of the strong error at time T in the discretized solution is [21]

$$\varepsilon_{\text{strong}}[S,T,h] = \left\langle \left\| S(T) - \hat{S}_{T/h} \right\|^2 \right\rangle^{1/2}, \tag{2}$$

where $\|A\|$ denotes some suitable norm of (the vector) A, and $\langle f \rangle$ denotes the expectation value over paths (also often denoted as $E[f]$) of any functional f of the path. Then the discretization that was used to obtain $\hat{S}_{T/h}$ is said to have $O(h^\alpha)$ strong convergence if $\varepsilon_{\text{strong}}[S,T,h] = O(h^\alpha)$ as $h \to 0$. In contrast, weak convergence concerns the weak error, which is defined as the error in the average of a path-dependent quantity over paths. While there are more general definitions [19], it is useful to consider the particular weak error at a specific time, which is analogous to the choice made in the definition of the strong error in Eq. (2):

$$\varepsilon_{\text{weak}}[S,T,h] = \langle S(T) \rangle - \left\langle \hat{S}_{T/h} \right\rangle. \tag{3}$$

The discretization that was used to obtain $\hat{S}_{T/h}$ is then said to have $O(h^\alpha)$ weak convergence if $\varepsilon_{\text{weak}}[S,T,h] = O(h^\alpha)$ as $h \to 0$.

In order to give an example of strong convergence, Fig. 1 shows plots of a normalized speed v and the component of the velocity unit vector along a given axis μ versus the normalized time t from various computations of one underlying sample trajectory. The particular equation used (Eq. (13) or Eq. (23)) will be derived and discussed in detail later in this paper. The simulations are run to an end time t_{end}=1.0. Each trajectory is calculated using the Milstein scheme, with set of time step values $\Delta t = t_{\text{end}} M^{-j}$, with $M = 3$, and $j = 0, 1, 2, \ldots 10$. While the end points from the coarser calculations show some scatter, the finer computations (those with smaller time steps) approach a well defined end point at $t = t_{\text{end}}$. This figure illustrates that, even for stochastic trajectories, computations can be carried out with different values of the time step for a given underlying trajectory, and the values of the variables at any given time point (including the end time point) can converge to well defined values. The strong convergence studies that will be shown later in this paper examine differences taken between variable values from computations for the same underlying trajectory, but which use different values of the time step.

There have been some partial implementations of the Milstein method for Coulomb collisions [11,12], but in these works, strong-convergence tests were not undertaken and no significant effect of the additional Milstein terms was observed. A second-order weak method has also been implemented for Coulomb collisions as a part of a neoclassical Monte-Carlo transport code for the study of transport in stellarators [22,23], although the improved weak convergence for the particular collisional

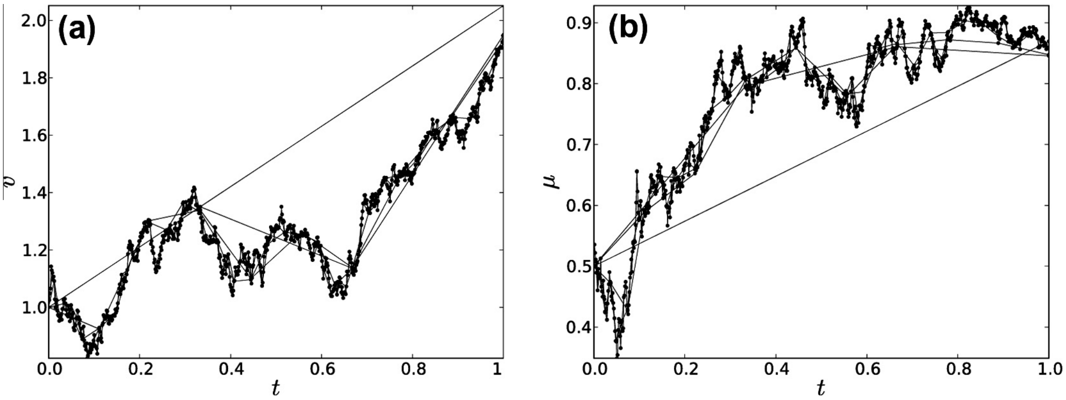


Fig. 1. (a) Speed v (normalized as explained later after Eqs. (20)–(22)), and (b) component of the velocity unit vector along a given axis μ vs. normalized time for numerical solutions of Eq. (17), for one underlying trajectory, using the Milstein algorithm with different values of the time step. The line segments connect the points, which are represented by the dots, along each discretized trajectory.

implementation was not documented. There is therefore a need for studies such as the present one that focus on the details of the implementation and on the improved rates of convergence for the higher order schemes in the context of the Coulomb-collision problem.

The organization of the remainder of this paper is as follows. In Section 2, the Euler–Maruyama and Milstein schemes for a (vector) system of SDE's are briefly summarized and reviewed. The discussion of our approach for the Milstein treatment of the Coulomb-collision problem begins in Section 3 where we develop the SDE's (Langevin equations) for the velocity expressed with the direction referenced to a fixed ("laboratory") frame that are consistent with the test-particle Maxwellian-background reduction of Eq. (1). This differs from the approach of [11], where the Langevin equations are formulated with respect to the current velocity direction, as is discussed in detail in Section 3 (and in Appendix A.) The Milstein discretization of the collisional Langevin equations is also given explicitly in Section 3. While this follows somewhat routinely from the Langevin equations, as discussed in Section 2, its implementation requires the sampling of particular (non-Gaussian) random numbers known as area integrals. In Section 4 we discuss the area integrals and develop a new and highly efficient and accurate method for their sampling. In Section 4, we also address the "compounding" of the random numbers (including the area integrals) that is needed both for strong-scaling studies and for the use of the Milstein scheme within a multi-level SDE algorithm [21]. The results of our Euler–Maruyama and Milstein implementations, including the scalings of the errors with the time step, are shown and discussed in Section 5. A summary and discussion of our work is given in Section 6.

2. Euler–Maruyama and Milstein schemes

Higher order methods can be formulated and understood in terms of iterative Taylor expansions of the formal finite-time integral solution [19]. The Euler–Maruyama and Milstein methods are the first two in a hierarchy of successive integration methods for SDE's of increasing order in the computational time step Δt. Consider the formal discrete SDE for a vector $\mathbf{Y}(t)$ with components $Y^i(t)$

$$dY^i(t) = a^i(t, \mathbf{Y}(t))dt + b^i(t, \mathbf{Y}(t))dW^i(t). \tag{4}$$

Here, each $W^i(t)$ is a Wiener process, i.e., a stochastic process with centered Gaussian increments which are independent for non-overlapping time intervals and which have variance

$$\left\langle \left[W^i(t_2) - W^i(t_1)\right]^2 \right\rangle = |t_2 - t_1|. \tag{5}$$

As such, Eq. (4) is not completely specified, but can be made so by considering it as a limit of a time-discrete equation in which all time-dependent quantities are given at times $\tau_k = t_0 + k\delta t$, where k is an integer time index and δt is a positive time increment. One such specification, known as the "Ito interpretation" (or specification), can be stated as the limit as $\delta t \to 0$ for the vector $\mathbf{y}_k = \mathbf{Y}(\tau_k)$, the components of which we denote as y^i_k, and which evolve according to

$$\begin{aligned} &y^i_{k+1} = y^i_k + \delta y^i_k, \quad \text{where} \\ &\delta y^i_k = a^i(\tau_k, \mathbf{y}_k)\delta t + b^i(\tau_k, \mathbf{y}_k)\delta W^i_k. \end{aligned} \tag{6}$$

Here, each δW^i_k is a centered normal random numbers with variance δt, and is independent of δW^j_l unless $i = j$ and $k = l$. The key point in Eq. (6) is that in the evaluation of b^i, the value of $\mathbf{Y}$ at the start of the time increment is used. Other specifications can be used. In the Stratonovich interpretation, for example, the b term in Eq. (6) is replaced by

$$b^i\left(\tau_k, \frac{1}{2}\left[\mathbf{y}_k + \mathbf{y}_{k+1}\right]\right)\delta W^i_k.$$

This distinction is important in that it affects the particular placement of the derivatives (drag terms) in the equation for the expectation value of the probability density of $\mathbf{Y}$.

Then $dY^i(t)$ can be defined by its integral. A simple version of such a definition is

$$Y^i(t_0 + \Delta t) - Y^i(t_0) = \int_{t_0}^{t_0+\Delta t} dY^i(t) = \lim_{N\to\infty} \sum_{k=0}^{N-1} \delta y^i_k,$$

where δy^i_k is given by Eq. (6), with $\delta t = \Delta t/N$.

The temporal discretizations to be formulated and used for computation can then be given by taking Δt to be the (finite) computational time step. The discretized SDE attempts to approximate $\mathbf{Y}_m = \mathbf{Y}(t_m)$, the components of which we denote as Y^i_m, and where $t_m = t_0 + m\Delta t$. The Euler–Maruyama and Milstein discretizations of Eq. (4) can be obtained as truncations of iterated expansions of the formal solution of Eq. (6).

The Euler–Maruyama scheme is

$$Y^i_M = Y^i_{\mathrm{Eul},M} = Y^i_0 + \sum_{m=1}^{M} \Delta_{\mathrm{Eul}} Y^i_{m-1}, \quad \text{where}$$
$$\Delta_{\mathrm{Eul}} Y^i_m = a^i(t_m, \mathbf{Y}_m)\Delta t + b^i(t_m, \mathbf{Y}_m)\Delta W^i_m,$$

where each ΔW^i_m is a centered Gaussian random number with variance Δt, and is independent of ΔW^j_n unless $i = j$ and $m = n$. The Euler–Maruyama scheme has the associated strong and weak errors [19]

$$\Delta_{\mathrm{Eul}} Y^i_m - \Delta Y^i_m = \begin{cases} O(\Delta t) & -\text{strong}, \\ O(\Delta t^2) & -\text{weak}. \end{cases}$$

If this scheme is run for some number M time steps over a time duration $T = M\Delta t \lesssim \|\mathbf{b}\|^2 \big/ \|\mathbf{a}\|^2$, then the resulting errors are (from the proof of Theorem 10.6.3 and from Theorem 14.6.1 of Ref. [19])

$$Y^i_{\mathrm{Eul},M} - Y^i_M = \begin{cases} O\left(\sqrt{T\Delta t}\right) & -\text{strong}, \\ O(T\Delta t) & -\text{weak}. \end{cases}$$

The Milstein (first-order "strong" in $\triangle t$) scheme is

$$Y^i_M = Y^i_{\mathrm{Milst},M} = Y^i_0 + \sum_{m=1}^{M} \Delta_{\mathrm{Milst}} Y^i_{m-1}, \quad \text{where} \tag{7}$$
$$\Delta_{\mathrm{Milst}} Y^i_m = a^i(t_m, \mathbf{Y}_m)\Delta t + b^i(t_m, \mathbf{Y}_m)\Delta W^i_m + b^i_{,j}(t_m, \mathbf{Y}_m) b^j(t_m, \mathbf{Y}_m) \int_0^{\Delta t} dW^i(t_{m-1} + s) \int_0^s dW^j(t_{m-1} + \eta).$$

The last term in Eq. (7) contains the double "area integral," which can be defined as

$$A_{ij,m} = \int_0^{\Delta t} dW^i(t_m + s) \int_0^s dW^j(t_m + \eta) = \lim_{N\to\infty} \sum_{k=1}^{N} \delta W^i_{mN+k-1} \sum_{l=1}^{k-1} \delta W^j_{mN+l-1},$$

and which are highly non-Gaussian random numbers. Note that because it is formed from increments of the fine Wiener increments, e.g., δW^i_l, and these also must satisfy $\Delta W^i_m = \sum_{k=1}^{N} \delta W^i_{mN+k-1}$, any sampling method for these integrals must respect the correlations between $A_{ij,m}$, and ΔW^i_m and ΔW^j_m. The sampling of these area integrals will be addressed in detail in Section 4.

The Milstein scheme has the associated strong and weak errors [19]

$$\Delta_{\mathrm{Milst}} Y^i_m - \Delta Y^i_m = \begin{cases} O(\Delta t^{3/2}) & -\text{strong}, \\ O(\Delta t^2) & -\text{weak}. \end{cases}$$

Note that the improvement in the error scaling for the Milstein scheme is for the strong error only.

Running this scheme for M time steps over a time duration $T = M\Delta t \lesssim \|\mathbf{b}\|^2 \big/ \|\mathbf{a}\|^2$ yields the resulting errors (again from the proof of Theorem 10.6.3 and from Theorem 14.6.1 of Ref. [19])

$$Y^i_{\mathrm{Milst},M} - Y^i_M = \begin{cases} O\left(\sqrt{T}\Delta t\right) & -\text{strong}, \\ O(T\Delta t) & -\text{weak}. \end{cases}$$

The Milstein method is of interest for two reasons. It is the first in a hierarchy of higher-order methods for SDE's. This hierarchy includes schemes with improved (higher-order) weak convergence. Also, when used instead of the Euler–Maruyama scheme as the underlying building block of Giles' multi-(time-) level scheme [21], the resulting efficiency is significantly improved, i.e., the computational complexity for a given overall error is significantly reduced.

Giles [21] gave a set of arguments, including a "complexity theorem," which compares the expected asymptotic computational cost between single-level and multi-level Euler and Milstein schemes. Specifically, he argued that for a given RMS error ϵ, so that the mean squared error (MSE) is ϵ^2, for MC integration up to a given time, the computational complexity (number of operations) scales dominantly as

- $C = O\ (\epsilon^{-3})$ for the single-level Euler–Maruyama and Milstein schemes,
- $C = O\ (\epsilon^{-2}[\log \epsilon]^2)$ for the Euler–Maruyama-based multi-level scheme,
- $C = O\ (\epsilon^{-2})$ for the Milstein – or any higher-strong-order algorithm-based multi-level scheme.

Note that Giles argues [21] that going beyond the Milstein algorithm in a multi-level scheme (i.e., using an algorithm with higher strong or weak order than the Milstein algorithm) does not result in any further improvement in the complexity vs. error beyond the Milstein-based multilevel scheme.

Because of prior work in the plasma-physics literature on higher-order weak schemes, it is of interest to also apply Giles' reasoning to a single-level scheme of arbitrary weak order. In order to achieve a MSE of order ϵ^2 optimally in a single-level

MC scheme, both the square of the bias error and the variance should be of order ϵ^2. For an $O(\Delta t^n)$ weak MC scheme, the bias error at a fixed end time versus Δt scales asymptotically as Δt^n. Thus, the number of time steps scales as $\epsilon^{1/n}$. The single-realization variance is asymptotically independent of Δt, i.e., it scales as $O(\Delta t^0)$. Therefore the number of realizations (particles) needed to reduce the variance to $O(\epsilon^2)$ is of order$O(\epsilon^{-2})$. The complexity for a single-level calculation the product of the number of time steps and particles, i.e.,

- $C = O(\epsilon^{-(2+1/n)})$ for the single-level, $O(\Delta t^n)$ weak MC scheme.

This is consistent with Giles' results for the single-level Euler–Maruyama and Milstein schemes. Thus, for example, a single-level MC calculation using a second-order weak scheme will have computational complexity that scales asymptotically as $\epsilon^{-2.5}$. The multi-level schemes, including the Euler–Maruyama based scheme, are therefore expected to result in significant efficiency improvements over single-level Monte-Carlo schemes, and the Milstein-based multi-level scheme has the potential to be optimal among the various schemes.

We have implemented a 2-dimensional Multi-level Milstein scheme, using a representation and area-integral sampling method developed in this paper, and preliminary results show the expected scaling of computational time with MSE [32].

3. Langevin equations and the Euler–Maruyama and Milstein schemes for the Coulomb-scattering test-particle problem

In this section, we develop the SDE's (Langevin equations) for the velocity expressed with the direction referenced to a fixed ("laboratory") frame, and which are consistent with the test-particle, isotropic-Maxwellian-background reduction of Eq. (1). The Milstein discretization of the collisional Langevin equations will also given explicitly. Our formulation differs from that in a widely used class of schemes [9,11,12], in which the angular-evolution steps are formulated in a frame aligned with the velocity at the start of the step.

The drag and diffusion coefficients for test-particle Coulomb scattering in an isotropic Maxwellian distribution of field particles, and which therefore correspond to the drag and diffusion coefficients that are effectively present in Eq. (1) were calculated explicitly in terms of known standard functions (the error function) by Chandrasekhar [24] (for the similar case of gravitational interactions), Spitzer [25] and Trubnikov [6]. Rosenbluth et al. [26] transformed Eq. (1) to spherical velocity coordinates which are the speed $v, \mu = \cos\theta$, where θ is the angle with respect to some reference axial direction, and the azimuthal angle φ. They also gave an elegant expression of the drag and diffusion coefficients in terms of potential functions that can be evaluated via a simple elliptic (Poisson-) equation solution. Trubnikov's evaluation of the drag and diffusion coefficients in Eq. (1) for the isotropic Maxwellian field-particle case [6] was through the evaluation of these potential functions.

The isotropy of the drag and diffusion coefficients is a useful simplifying feature of the present problem, which can be viewed as a special case of the results of Ref. [26]. The test-particle equation can be obtained from Eqs. (19) and (31) of Ref. [26] by fixing the field particle distribution in the potential functions g and h, effectively considering the field particles to be a separate species that does not evolve, even when the field particles represent the same physical species as the test particles. The main result of Ref. [26] (Eq. (31) of Ref. [26]) is for the case of azimuthally symmetric test- and field-particle distributions. The corresponding result for general (non-axisymmetric) test-particle distribution, but still an azimuthally symmetric field-particle distribution, is also easily obtained from Ref. [26] by keeping the S^{33} term given in Eq. (30) of Ref. [26] in the evaluation of the terms in Eq. (22) of Ref. [26]. Doing so, and taking the field particles to be isotropic, so that all of the partial derivatives of the potential functions g and h of Ref. [26] with respect to μ are zero, gives

$$\frac{1}{\Gamma_{tf}}\left(\frac{\partial f_t}{\partial t}\right)_c = -\frac{1}{v^2}\frac{\partial}{\partial v}\left[\left(v^2\frac{\partial h}{\partial v}+\frac{\partial g}{\partial v}\right)f_t\right]+\frac{1}{2v^2}\frac{\partial^2}{\partial v^2}\left(v^2\frac{\partial^2 g}{\partial v^2}f_t\right)+\frac{1}{2v^3}\frac{\partial g}{\partial v}\left\{\frac{\partial}{\partial\mu}\left[(1-\mu^2)\frac{\partial f_t}{\partial\mu}\right]+\frac{1}{(1-\mu^2)}\frac{\partial^2 f_t}{\partial\phi^2}\right\}. \tag{8}$$

Here,

$$\Gamma_{tf} = \frac{4\pi q_t^2 q_f^2 \lambda}{m_t^2},$$

where the subscripts t and f refer to test- and field-particle quantities, q_t and q_f are the test and field-particle masses m_t is the test-particle mass, and λ is the Coulomb logarithm. The results of Ref. [26] can easily be generalized, if needed, to allow for azimuthal (ϕ) dependence in the potential functions g and h. These can be expressed using Trubnikov's normalizations

$$g(v) = -8\pi\psi_f(v),$$
$$h(v) = -4\pi\left(1+\frac{m_t}{m_f}\right)\varphi_f(v),$$

where

$$\psi_f(v) = -\frac{1}{8\pi}\int d\mathbf{v}' f_f |\mathbf{v}-\mathbf{v}'|,$$
$$\varphi_f(v) = -\frac{1}{4\pi}\int d\mathbf{v}' f_f |\mathbf{v}-\mathbf{v}'|^{-1}.$$

The functions ψ_f and φ_f then satisfy the simple Poisson equations

$$\nabla_v^2\psi_f = \varphi_f,$$
$$\nabla_v^2\varphi_f = f_f.$$

They are evaluated by Trubnikov for the Maxwellian case in Ref. [6] (Eqs. (17.5), (17.9), (17.22), (17.23), (15.9) therein). The results are

$$\begin{aligned} g(v) &= n_f v\left[\frac{\Phi}{x^2} + (\Phi - G)\right] \\ &= \frac{1}{2} n_f \sqrt{2} v_f \left[\Phi\left(2x + \frac{1}{x}\right) + \Phi'\right], \\ h(v) &= \frac{1}{2}\left(1 + \frac{m_t}{m_f}\right)\frac{1}{v^2}\frac{\partial}{\partial v}\left(v^2\frac{\partial g}{\partial v}\right) \\ &= n_f\left(1 + \frac{m_t}{m_f}\right)\frac{\Phi}{v}. \end{aligned}$$

Here, $\Phi(x)$ is the standard error function and G is the Chandrasekhar function

$$G(x) = \frac{\Phi(x) - x\Phi'(x)}{2x^2}.$$

Also, n_f is the field-particle density, m_t and m_f the test and field-particle masses,

$$x = l_f v,$$
$$l_f = \left(\sqrt{2} v_f\right)^{-1},$$
$$v_f = \sqrt{\frac{T_f}{m_f}}$$

and T_f is the field-particle temperature. The drag force F_d and the v and angular diffusion coefficients D_v and D_a then follow as

$$F_d(v) = -\Gamma_{tf}\left(\frac{\partial h}{\partial v} + \frac{1}{v^2}\frac{\partial g}{\partial v}\right) = -\frac{A_D}{2v^2}\left\{\left[\left(1 + \frac{m_t}{m_f}\right)2l_f^2 v^2 + 1\right]G(l_f v) - \Phi(l_f v)\right\}, \tag{9}$$

$$D_v(v) = \frac{\Gamma_{tf}}{2}\frac{\partial^2 g}{\partial v^2} = \frac{A_D}{2v}G(l_f v), \tag{10}$$

$$D_a(v) = \frac{\Gamma_{tf}}{2v^3}\frac{\partial g}{\partial v} = \frac{A_D}{4v^3}[\Phi(l_f v) - G(l_f v)], \tag{11}$$

where

$$A_D = 2n_f\Gamma_{tf} = \frac{8\pi n_f q_t^2 q_f^2 \lambda}{m_t^2}.$$

As discussed in Appendix A, these results agree with those of Ref. [11], which are based on the results of Refs. [24,25].

To obtain the appropriate SDE's, one can proceed directly from Eqs. (8)–(11). Multiplying Eq. (8) by $2\pi v^2$, writing $\hat{f}_t = 2\pi v^2 f_t$, and taking the derivatives to the outside gives

$$\left(\frac{\partial \hat{f}_t}{\partial t}\right)_c = -\frac{\partial}{\partial v}\left[F_d(v)\hat{f}_t\right] + \frac{\partial^2}{\partial v^2}\left[D_v(v)\hat{f}_t\right] + \frac{\partial}{\partial \mu}\left[2D_a(v)\mu\hat{f}_t\right] + \frac{\partial^2}{\partial \mu^2}\left[D_a(v)(1-\mu^2)\hat{f}_t\right] + \frac{\partial^2}{\partial \phi^2}\left[\frac{D_a(v)}{(1-\mu^2)}\hat{f}_t\right]. \tag{12}$$

Eq. (12) is in the form of the Fokker–Planck equation for a 3-dimensional SDE system with drag and diffusion coefficients that depend on the variables in the Ito-calculus. It is also the Fokker–Planck equation corresponding to forward explicit discretizations of this SDE, such as the Euler–Maruyama or Milstein schemes. The resulting SDE's are

$$dv(t) = F_d(v)dt + \sqrt{2D_v(v)}dW_v(t), \tag{13}$$

$$d\mu(t) = -2D_a(v)\mu dt + \sqrt{2D_a(v)(1-\mu^2)}dW_\mu(t), \tag{14}$$

$$d\phi(t) = \sqrt{\frac{2D_a(v)}{(1-\mu^2)}}dW_\phi(t). \tag{15}$$

We choose to normalize the velocity to v_f, i.e., use $\hat{v} = v/v_f$ as the velocity variable. This normalization is perhaps not the most convenient for the argument of the error function and associated functions, but gives the 1D field-particle Maxwellian a variance of 1. That is

$$F_{\mathrm{Mf}}(\hat{v}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\hat{v}^2}{2}\right).$$

Define a standard thermal field-particle collision rate ν_f by

$$\nu_f = \frac{4\pi n_f q_t^2 q_f^2 \lambda}{m_t^2 v_f^3} = \frac{A_D}{2 v_f^3} \tag{16}$$

and normalize the time by this rate, i.e., $\hat{t} = \nu_f t$. Then the resulting dimensionless equations are

$$d\hat{v}(\hat{t}) = \hat{F}_d(\hat{v}) d\hat{t} + \sqrt{2\hat{D}_v(\hat{v})} dW_v(\hat{t}), \tag{17}$$

$$d\mu(\hat{t}) = -2\hat{D}_a(\hat{v})\mu d\hat{t} + \sqrt{2\hat{D}_a(\hat{v})(1-\mu^2)} dW_\mu(\hat{t}), \tag{18}$$

$$d\phi(\hat{t}) = \sqrt{\frac{2\hat{D}_a(\hat{v})}{(1-\mu^2)}} dW_\phi(\hat{t}), \tag{19}$$

with the normalized (dimensionless) drag and diffusion coefficients

$$\hat{D}_v(\hat{v}) = \frac{D_v(v)}{\nu_f v_f^2} = \frac{1}{\hat{v}} G\left(\frac{\hat{v}}{\sqrt{2}}\right), \tag{20}$$

$$\hat{D}_a(\hat{v}) = \frac{D_a(v)}{\nu_f} = \frac{1}{2\hat{v}^3}\left[\Phi\left(\frac{\hat{v}}{\sqrt{2}}\right) - G\left(\frac{\hat{v}}{\sqrt{2}}\right)\right]. \tag{21}$$

$$\hat{F}_d(\hat{v}) = \frac{F_d(v)}{\nu_f v_f} = -\left(1 + \frac{m_t}{m_f}\right)\hat{v}\hat{D}_v(\hat{v}) + 2\hat{v}\hat{D}_a(\hat{v}). \tag{22}$$

In the remainder of this paper, we work in the normalized variables, and the carets are suppressed.

The Euler and Millstein schemes are next elaborated for the Langevin Coulomb collision operator. From Eqs. (17)–(19), we can obtain the corresponding Euler and Milstein schemes by iterative Taylor expansion and retention of terms up to $O(\sqrt{\Delta t})$ for Euler or $O(\Delta t)$ for Milstein:

$$\Delta v = F_{d0}\Delta t + \sqrt{2D_{v0}}\Delta W_v + \kappa_{\mathrm{M}} D'_{v0}\frac{1}{2}(\Delta W_v^2 - \Delta t), \tag{23}$$

$$\Delta\mu = -2D_{a0}\mu_0\Delta t + \sqrt{2D_{a0}(1-\mu_0^2)}\Delta W_\mu, +\kappa_{\mathrm{M}}\left[-2D_{a0}\mu_0\frac{1}{2}(\Delta W_\mu^2 - \Delta t) + \sqrt{\frac{D_{v0}}{D_{a0}}}\sqrt{(1-\mu_0^2)}D'_{a0}A_{v\mu}\right], \tag{24}$$

$$\Delta\phi = \sqrt{\frac{2D_a(v)}{1-\mu_0^2}}\Delta W_\phi + \kappa_{\mathrm{M}}\left[\sqrt{\frac{D_{v0}}{D_{a0}}}\frac{D'_{a0}}{\sqrt{1-\mu_0^2}}A_{v\phi} + \frac{2D_{a0}\mu_0}{1-\mu_0^2}A_{\mu\phi}\right], \tag{25}$$

where the coefficient $\kappa_{\mathrm{M}} = 0$ for Euler, and 1 for Milstein, and

$$\begin{aligned}
&\Delta t = t_{i+1} - t_i,\\
&F_{d0} = F_d(v_0),\\
&v_0 = v(t_i),\\
&\mu_0 = \mu(t_i),\\
&D_{v0} = D_v(v_0),\\
&D'_{v0} = D'_v(v_0),\\
&D_{a0} = D_a(v_0),\\
&D'_{a0} = D'_a(v_0),
\end{aligned} \tag{26}$$

$$\Delta W_j = \int_{t_i}^{t_{i+1}} dW_j(s),$$

$$A_{jl} = \int_{t_i}^{t_{i+1}} dW_l(s) \int_{t_i}^{s} dW_j(\xi)$$

for $j, l = v, \mu, \phi$. Note that here, we have put the component indices, e.g., j in ΔW_j as subscripts instead of as superscripts as was done in Section 2, and that this does not cause a notational conflict because there is no time-index subscript in Eqs. (26).

4. Sampling and compounding area integrals

As noted in Section 2, multidimensional Milstein cases generally require sampling of the area integrals. These are highly non-Gaussian random numbers, and their sampling needs to be done consistently with the associated Wiener displacements.

Several methods have been developed for this sampling. One class of approaches is based on discrete representations [19,27]. Kloeden and Platen [19, Section 5.8] consider the calculation of the area integrals via the truncation of a series that results from a Fourier-series representation of the Wiener process, and showed that the mean square error in the resulting samples scales as the inverse of the number of terms kept. Gaines and Lyons [27] developed a quadrature method based on the summation of products of the Wiener-process increments, and showed that this method also had a similar scaling of the mean square error with the number of terms. Thus, these discrete methods have an unfavorable scaling of computational effort versus accuracy. The method of Ref. [27], however, has the advantage that it lends itself to an adaptive algorithm (which was a key focus of that work), and both of the discrete approaches generalize in a straightforward way to integrals of higher dimension which arise in schemes of higher strong order than the Milstein scheme.

For the standard two-dimensional area integrals that arise in the Milstein scheme, there is a body of theoretical work [28] on the joint and conditional PDF's of the area integral and the Wiener displacements that enables direct sampling methods. One direct method for the sampling has been developed [29], which uses a 2-dimensional extension of a method due to Marsaglia (see [29,30] and references therein) applied to the joint PDF. This method is direct and accurate and, unlike for the discrete approaches, the computational effort is only very weakly dependent on the desired accuracy, but has perhaps found somewhat limited use because 2-dimensional extension of Marsaglia's method is somewhat involved.

Here, we propose and develop an alternative direct method, based on sampling the *conditional density* (instead of the joint density) of the area integral given the Wiener increments. The appeal of this method is that once the Wiener increments (which are normal random numbers) are obtained, the remaining conditional PDF of the area integral consistent with the Wiener-increment values is a one-dimensional function of the area integral, and the sampling can be done with a standard transformation method. We use an additional approximation, the accuracy of which we will quantify, to greatly simplify the sampling. While small, the errors associated with this approximation, can easily be removed, if needed, through either a rejection method [30] or by use of an accurate numerical tabulation of the PDF and a numerical transformation-method-based sampling routine. Our method gives very good accuracy with a marginal cost for each sample comparable to sampling an additional Gaussian random number.

In this section, we will work with normalized Wiener increments and area integrals to remove the explicit appearance of the numerical time step Δt. Thus, we write $\Delta W_i(\Delta t) = \sqrt{\Delta t}\Delta\hat{W}_i$ for $i = 1, 2$, where

$$\Delta\hat{W}_i \equiv \Delta W_i(1) = \int_0^1 dW_i(\xi)$$

and $A_{12}(\Delta t)$, e.g., as it arises in Eqs. (7), (24) or (25) is normalized as $A_{12}(\Delta t) = \Delta t\hat{A}_{12}$, where

$$\hat{A}_{12} \equiv A_{12}(1) = \int_0^1 W_1(\xi) dW_2(\xi).$$

The carets will be removed in the remainder of this section.

4.1. Joint PDF of area integral and Gaussian displacements

The basic probability density function of interest is the joint probability density of ΔW_1, ΔW_2, and A_{12} which will be denoted as $p_{JA}(A_{12}, \Delta W_1, \Delta W_2)$. This is related to the conditional probability density of A_{12} given ΔW_1 and ΔW_2, $p_{cA}(A_{12}|\Delta W_1, \Delta W_2)$ by

$$p_{JA}(A_{12}, \Delta W_1, \Delta W_2) = p_{cA}(A_{12}|\Delta W_1, \Delta W_2) p_{\Delta W}(\Delta W_1) p_{\Delta W}(\Delta W_2),$$

where $p_{\Delta W}$ is the probability density of the Wiener-process increments. The fundamental body of work that underlies the sampling of these quantities [28] deals with the antisymmetric "Levy area," which for the two-dimensional case can be defined as the area included by the curve $(W_1(\xi), W_2(\xi))$, $\xi \in [0, 1]$ and its chord [28], i.e.,

$$L_{12} \equiv \frac{1}{2}\int_0^1 [W_1(\xi)dW_2(\xi) - W_2(\xi)dW_1(\xi)]$$

and is related to the area integrals by

$$L_{12} \equiv A_{12} - \frac{1}{2}\Delta W_1 \Delta W_2 = \frac{1}{2}\Delta W_1 \Delta W_2 - A_{21}. \tag{27}$$

Thus, for example,

$$p_{cA}(A_{12}|\Delta W_1, \Delta W_2) = p_{cL}\left(A_{12} - \frac{1}{2}\Delta W_1 \Delta W_2 \middle| \Delta W_1, \Delta W_2\right),$$

where $p_{cL}(L_{12}|\Delta W_1, \Delta W_2)$ is the conditional probability density of L_{12} given ΔW_1 and ΔW_2.

Levy [28] showed that $p_{cL}(L_{12}|\Delta W_1, \Delta W_2)$ depends on ΔW_1 and ΔW_2 only through the combination $R \equiv \sqrt{\Delta W_1^2 + \Delta W_2^2}$, i.e., $p_{cL}(L_{12}|\Delta W_1, \Delta W_2) = p_{cL}(L_{12}|R)$, and its characteristic function is given by

$$\phi_{cL}(k|R) = \langle \exp(-ikL)\rangle|_R = \frac{k/2}{\sinh(k/2)} \exp\left\{\frac{R^2}{2}\left[1 - \frac{(k/2)\cosh(k/2)}{\sinh(k/2)}\right]\right\} \tag{28}$$

and that the corresponding (1-point unconditional) characteristic function of L is

$$\phi_L(k) \equiv \langle \exp(-ikL)\rangle = \frac{1}{\cosh(k/2)}.$$

Plots of $\phi_{cL}(k|R)$ are shown in Fig. 2.

The PDF's can be obtained easily from the their characteristic functions by direct (numerical) Fourier integration, using an integration path on the real axis, as the resulting integrands are nonsingular, smooth, bounded and approach zero rapidly for large $|k|$. (Also, note that it is straightforward to use discrete Fourier transforms on a sufficiently large interval and sufficiently many points to evaluate such integrals over a wide range of L.)

The corresponding conditional and unconditional PDF's for L are then given by

$$p_{cL}(L|\Delta W_1, \Delta W_2) = p_{cL}(L|R),$$

where

$$\begin{bmatrix} p_{cL}(L|R) \\ p_L(L) \end{bmatrix} = \frac{1}{2\pi}\int \begin{bmatrix} \phi_{cL}(k|R) \\ \phi_L(k) \end{bmatrix} \exp(-ikL)dk. \tag{29}$$

This gives

$$p_L(L) = \frac{1}{\cosh(\pi L)}.$$

The joint PDF $p_{JL}(L, R)$ is given by

$$p_{JL}(L, R) = p_{cL}(L|R)p_R(R),$$

where $p_R(R)$ is the PDF of R

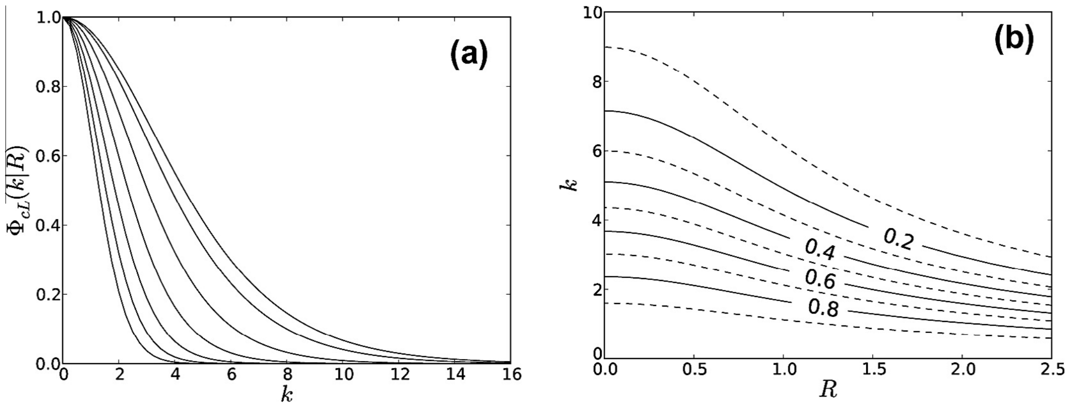


Fig. 2. (a) Line plot $\phi_{cL}(k|R)$ vs. k for $R = 0.0, 0.5, 1.0, \ldots, 3.0$. The wider curves are for smaller R, and the narrower curves are for larger R. (b) Contour plot of $\phi_{cL}(k|R)$ vs. R and k.

$$p_R(R) = R\exp\left(\frac{-R^2}{2}\right).$$

The results for $p_{cL}(L|R)$ and $p_{JL}(L,R)$ are not available in as simple a form as for $p_L(L)$, but can be obtained to any desired accuracy by direct numerical evaluation of Eq. (29). Plots of these functions are shown in Fig. 3.

As discussed above, the quantity of interest for the multidimensional Milstein schemes is the area integral A, rather than L. It is therefore of interest to calculate and plot various 1- and 2-dimensional joint PDF's involving A that can be compared with empirical PDF's resulting from any number generators used to generate A, even though the joint PDF $p_{JA}(A_{12},\Delta W_1,\Delta W_2)$ is fundamentally 3-dimensional.

One such PDF of interest is the joint PDF $p_{JA}(A_{12},R)$, which can be obtained by putting

$$\Delta W_1 = R\cos\theta,$$
$$\Delta W_2 = R\sin\theta$$

and averaging over the angle θ.

The characteristic function of the conditional PDF of A given R is

$$\phi_{cA}(k|R) \equiv \langle\exp(-ikA)\rangle|_R = J_0(kR^2/4)\frac{k/2}{\sinh(k/2)}\exp\left\{\frac{R^2}{2}\left[1-\frac{(k/2)\cosh(k/2)}{\sinh(k/2)}\right]\right\}, \tag{30}$$

where J_0 denotes the Bessel function of order zero. Again, this can be used to calculate $p_{JA}(A_{12},R)$ via direct numerical Fourier inversion. The result is shown in Fig. 4.

Finally, this relation can be used to obtain the (unconditional) characteristic function of A, by integrating over R.

$$\phi_A(k) \equiv \langle\exp(-ikA)\rangle = \frac{1}{\sqrt{\cosh(k)}}.$$

The results for $\phi_L(k)$ and $\phi_A(k)$ can be Fourier inverted to give the corresponding PDF's. For $p_L(L)$ we have

$$p_L(L) = \frac{1}{\cosh(\pi L)}.$$

$p_A(A)$ can be expressed in terms of Gamma functions of complex argument, but the direct numerical Fourier inversion is easy and accurate and, combined with asymptotic analysis of the Fourier integral representation, is more useful than the Gamma-function representation.

4.2. Approximations and simple numerical generator for A

$p_{cL}(L|R)$ has an exact closed-form analytical expression for $R = 0$,

$$p_{0L}(L) \equiv p_{cL}(L|R=0) = \frac{\pi}{2}\frac{1}{\cosh^2(L/2)}. \tag{31}$$

For large R, an approximate saddle-point evaluation gives

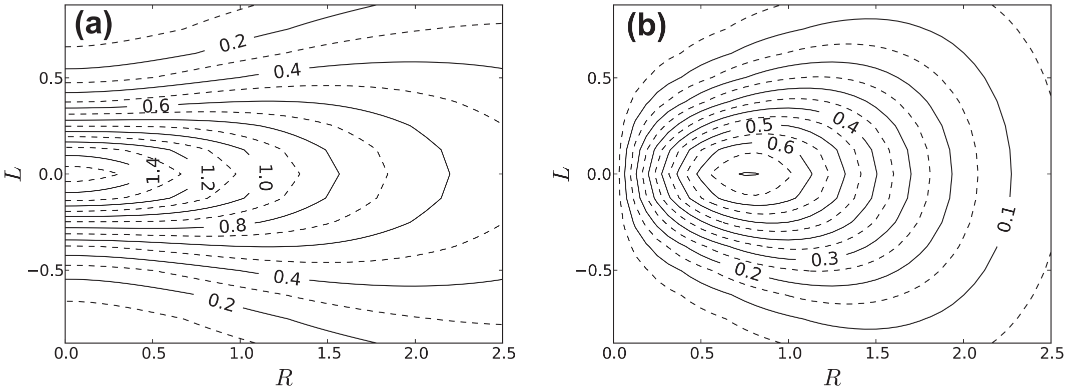


Fig. 3. Contour plots of (a) the conditional PDF $p_{cL}(L|R)$ and (b) the joint PDF $p_{JL}(L,R)$ vs. R and L from semianalytical calculations (i.e., numerical Fourier inversion of the analytically calculated characteristic function from Eq. (28)).

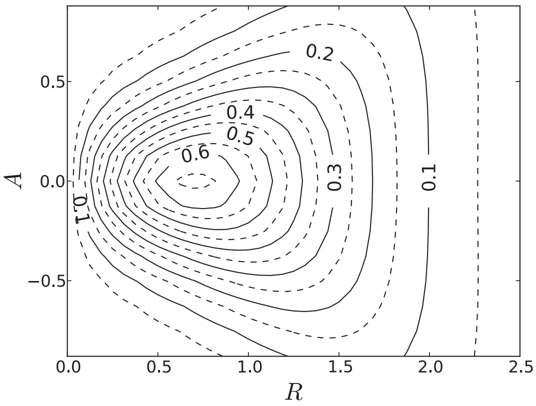


Fig. 4. Contour plot of the joint PDF $p_{JA}(A,R)$ vs. R and A from semianalytical calculation using Eq. (30).

$$p_{cL}(L|R) \sim \sqrt{\frac{1}{\pi}\left(\frac{6}{1+R^2}\right)}\exp\left[-\left(\frac{6}{1+R^2}\right)L^2\right]. \tag{32}$$

This is compared with a direct numerical evaluation of $p_{cL}(L|R)$ in Fig. 5 where it is seen that the large-R approximation gives good agreement with the numerical result for $R \gtrsim 2.5$, while the agreement becomes less satisfactory for smaller R.

In devising a sampling algorithm it is useful to note that the region of greatest probability of the joint PDF $p_{JL}(L,R)$ is heavily influenced by the PDF of R, for which most of the probability lies in the region $R \lesssim 3$. A useful approximation can be found based on the above result for $p_{0L}(L) = p_{cL}(L|R=0)$. If we define a function $s(R) \equiv p_{cL}(L=0|R)$, then a useful approximation is

$$p_{cL}(L|R) \approx p_{\mathrm{can}L}(L|R) = s(R)p_{0L}(s(R)L). \tag{33}$$

A comparison between $p_{\mathrm{can}\,L}(L|R)$ and $p_{cL}(L|R)$ using 1D plots is shown in Fig. 6.

A more global view can be obtained from 2D plots of the errors in the (R,L) plane. The relative error $p_{cL}(L|R)/p_{\mathrm{can}\,L}(L|R) - 1$ and absolute error in the resulting joint PDF $p_{\mathrm{Jan}\,L}(L,R) \equiv p_{\mathrm{can}\,L}(L|R)p_R(R)$, $p_{\mathrm{Jan}\,L}(L,R) - p_{JL}(L,R)$ are shown in Fig. 7. To put the absolute error in $p_{\mathrm{Jan}\,L}(L,R)$ shown in the second frame of Fig. 7 in perspective, note that the maximum value of $p_{JL}(L,R)$ is approximately 0.7, as shown in the second frame of Fig. 3.

A particularly useful aspect of Eq. (33) is that the inverse of its integral (distribution function), which is the function needed for sampling L can be obtained analytically in closed form. The resulting sampling function is

$$L_R(R) = \frac{s(R)}{2\pi}\log\left(\frac{r}{1-r}\right), \tag{34}$$

where r a random number uniformly distribute between 0 and 1. Furthermore, a good analytical fit to $s(R)$ is

$$s(R) \approx s_a(R) = [1 + (1.13663 \times R)^{2.27}]^{1/2.27}.$$

A comparison of $s_a(R)$ with $s(R)$ is shown in Fig. 8. $s_a(R)$ can be used instead of $s(R)$ in Eq. (34).

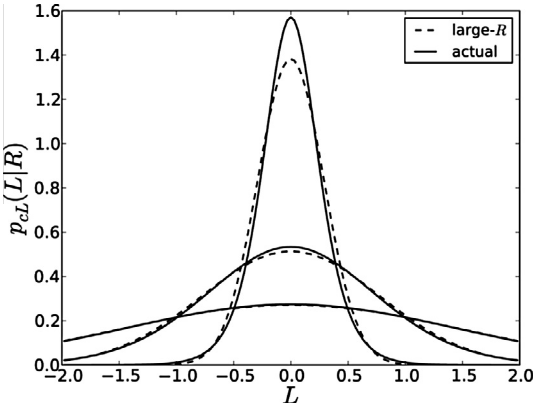


Fig. 5. Comparison of $p_{cL}(L|R)$ with large-R approximation for $R = 0$ (narrow, tall curves), 2.5, and 5 (wide, low curves).

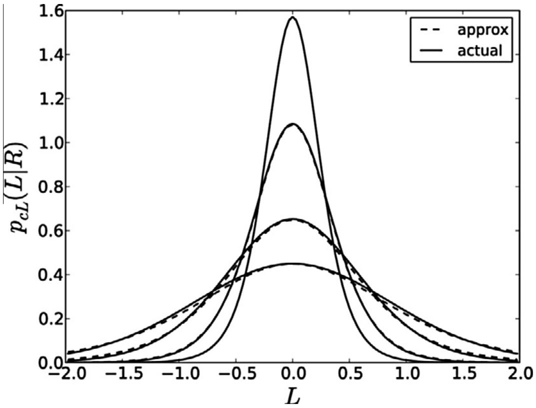


Fig. 6. Comparison between $p_{\mathrm{can}\,L}(L|R)$ and $p_L(L|R)$ for $R = 0$ (narrow, tall curves), 1.0, 2.0, and 3.0 (wide, low curves).

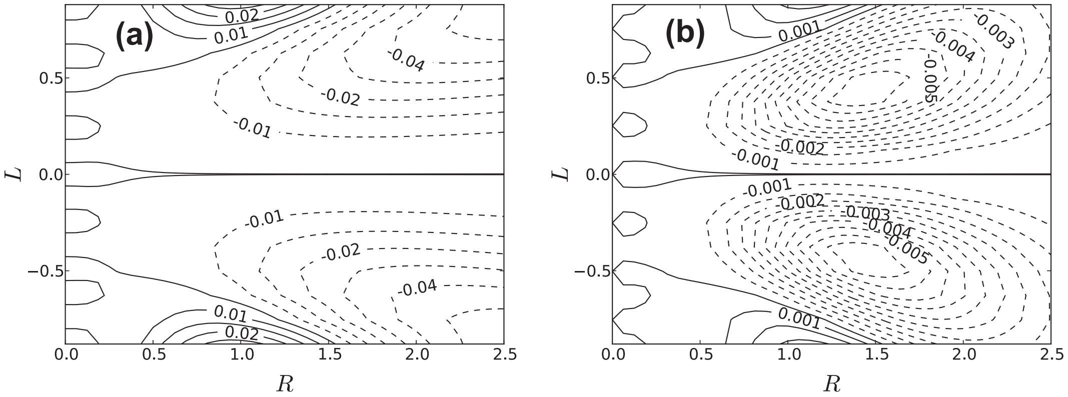


Fig. 7. Contour plots of (a) relative error in $p_{\mathrm{can}\,L}(L|R)$ and (b) absolute error in $p_{\mathrm{Jan}\,L}(L,R)$ vs. R and L.

The analytical expressions are useful for initial implementations and checks. For numerical sampling, rather than using the analytical formulas directly, it is much more computationally efficient to form tables from suitable analytical expressions or numerical computations, and to use interpolation from the resulting tables.

A property of $p_{\mathrm{can}\,L}(L|R)$ that is suggested by Fig. 6 and is clear from the asymptotic expressions Eqs. (31) and (32) is that for large $L, p_{\mathrm{can}\,L}(L|R)$ decreases with L as slowly or more slowly than $p_L(L|R)$. Because of this, our sampling method can be made more accurate if needed by using a rejection method [30] with $\beta p_{\mathrm{can}\,L}(L|R)$, where $\beta \approx 1$, as a comparison function. From

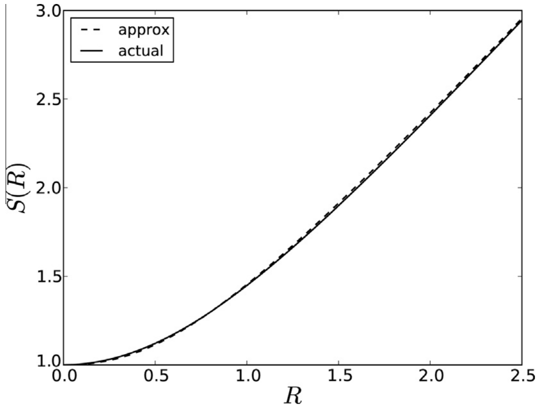


Fig. 8. Comparison of $s_a(R)$ (analytical approximation) with (actual) $s(R)$.

Figs. 6 and 7 it is evident that the fraction of rejected samples in such a method will be small, so that the refined sampling method will not be significantly more computationally expensive than the unrefined method.

4.3. Results from numerical generators

A numerical generator based on Eq. (34) for the triplets $(\Delta W_1, \Delta W_2, L_{12})$ and hence, via Eq. (27), for the triplets $(\Delta W_1, \Delta W_2, A_{12})$ has been written. The contour plot of the resulting empirical $p_{JL}(L,R)$ from the numerical generator, for 2×10^5 points in (R,L) space, is shown in Fig. 9. This plot shows excellent agreement with the near-exact semi-analytical result of Fig. 3(b), apart from the expected statistical fluctuations in Fig. 9 due to the finite number of samples. A more quantitative view of the excellent level of agreement is given by the 1-dimensional plots of empirical and semi-analytical curves for $p_L(L)$ in Fig. 11.

The empirically generated $p_{JA}(A,R)$ is shown in Fig. 10. Again, apart from the expected statistical fluctuations, this plot shows excellent agreement with the corresponding semi-analytical result shown in Fig. 4. The 1-dimensional plots of empirical and semi-analytical curves for $p_A(A)$ in Fig. 11 also show the excellent agreement.

4.4. Compounding of area integrals

In order to carry out error scaling studies involving discretizations using different time steps of the same underlying stochastic path (solution of a stochastic ODE), the ODE integrator must use random numbers at the coarser time levels that are suitably compounded from those used at the finer time levels. For the Gaussian random displacements, this compounding is simply addition of the displacements followed by a suitable normalization. Thus, if $W_l(t)$ is a Wiener process associated with random vector component l, $\delta_j W_l \equiv \int_{t_{j-1}}^{t_j} dW_l(s)$ is the jth Gaussian displacement over the time interval of length δt, $[t_{j-1}, t_j]$, where $t_j = t_{j-1} + \delta t$, and $\Delta W_l \equiv \int_0^{\Delta t} dW_l(s)$, where $\Delta t = n\delta t$, then a suitable compounding for use in the ODE discretization is simply

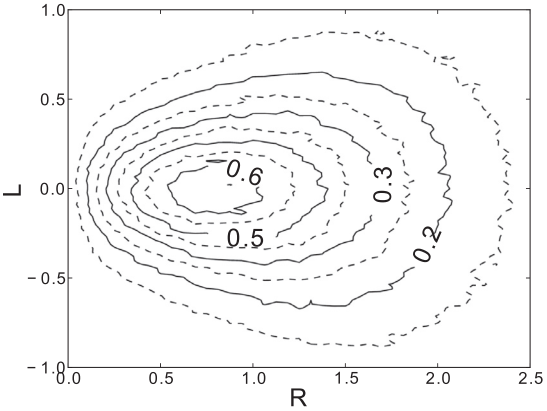


Fig. 9. Plot of $p_{JL}(L,R)$ vs. R and L for numbers produced by the numerical generator.

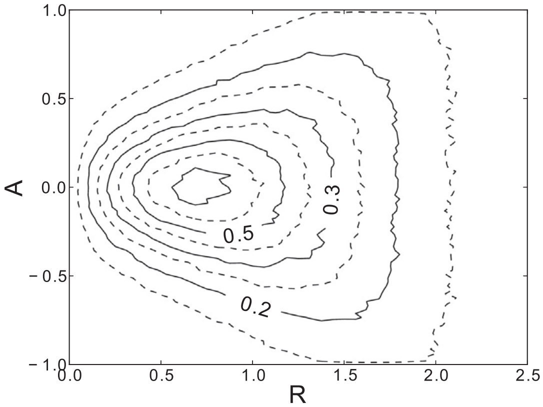


Fig. 10. Plot of $p_{JA}(A,R)$ vs. R and A for numbers produced by the numerical generator.

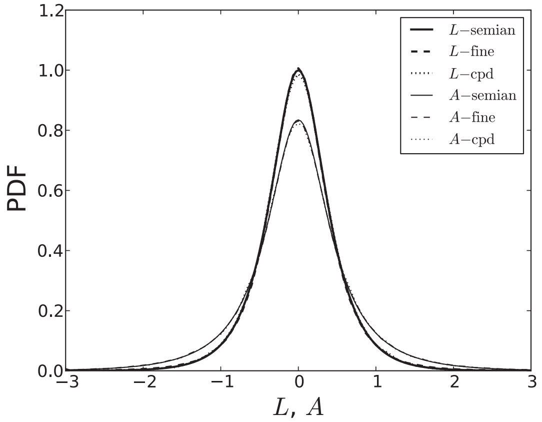


Fig. 11. $p_L(L)$ and $p_A(A)$, from semianalytical calculations, from the Levy-area/area-integral generator directly, and from compounded sets of numbers. There are three curves each for $p_L(L)$ (upper set of curves) and $p_A(A)$, (lower set of curves).

$$\Delta W_l = \sum_{j=1}^{n} \delta_j W_l.$$

An alternative is to use displacements with unit variance at each time level. Thus, for example,

$$\delta_j W_l = \sqrt{\delta t}\,\delta_j \hat{W}_l,$$
$$\Delta W_l = \sqrt{\Delta t}\,\Delta \hat{W}_l.$$

Then it follows that

$$\Delta \hat{W}_l = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \delta_j \hat{W}_l.$$

The corresponding compounding of the area integrals follows straightforwardly. If

$$\delta A_{12,j} = \int_{t_{j-1}}^{t_j} dW_1(\eta) \int_{t_{j-1}}^{\eta} dW_2(\xi)$$

and

$$\Delta A_{12} = \int_0^{\Delta t} dW_1(\eta) \int_0^{\eta} dW_2(\xi),$$

then

$$\Delta A_{12} = \sum_{i=1}^{n} \left(\delta_i W_1 \sum_{j=1}^{i-1} \delta_j W_2 + \delta A_{12,i} \right).$$

This is illustrated in Fig. 12. Note that when the compounding is implemented as per this formula, the number of operations scales as n, not as n^2.

In the alternative normalization, if

$$\delta \hat{A}_{12,j} = \frac{1}{\delta t} \delta A_{12,j},$$
$$\Delta \hat{A}_{12} = \frac{1}{\Delta t} \Delta A_{12},$$

then

$$\Delta \hat{A}_{12} = \frac{1}{n} \sum_{i=1}^{n} \left(\delta_i \hat{W}_1 \sum_{j=1}^{i-1} \delta_j \hat{W}_2 + \delta \hat{A}_{12,i} \right).$$

This compounding has been implemented. Fig. 11 shows the 1D PDF's $p_L(L)$ and $p_A(A)$ from the semi-analytical calculations, from the number generator directly, and from compounded sets of numbers. Here again, 9×10^4 sets of numbers are produced by the generator. The compounding in the results shown combines numbers from sets of 5 timesteps.

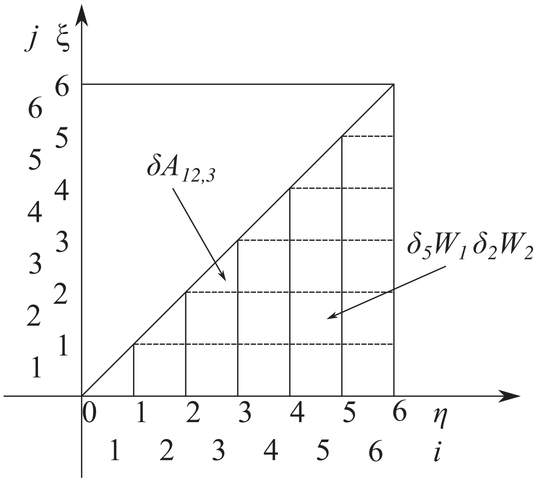


Fig. 12. Composition of area integral for $n = 6$.

Fig. 13 shows 2D PDF's $p_{JL}(L,R)$ and $p_{JA}(A,R)$, from the compounded sets of numbers. Although the 2D plots are somewhat noisy (they are densities produced from 4×10^4 points, these plots show good agreement with Figs. 3, 9, 4, and 10.

5. Results from numerical implementation

A program based on the methods discussed in the previous sections has been written to solve the two-dimensional version of the Coulomb-collision Langevin equations, Eqs. (17) and (18) via the Euler–Maruyama and Milstein schemes, as given by Eqs. (23) and (24).

For the evaluation of the values of $D_v(v)$ and $D_a(v)$ of Eqs. (20) and (21) at the particle speed v, we use cubic-spline interpolation between accurately computed, tabulated values of these quantities at a set of (equally spaced) nodes. For the coefficients in the Milstein scheme we will also need the quantities $D'_v(v)$ and $D'_a(v)$, and these are also easily available from the spline-interpolation routines. While some simple analytical fits are available for the drag and diffusion coefficients of Eqs. (9)–(11) (see, e.g., Refs. [10,12]), we use table-interpolation-based routines because they are at least equally efficient and more accurate than inline numerical evaluation of the analytical fits. The tables themselves need to be precalculated only once, and this can be done with almost any desired degree of accuracy and nodal density. Furthermore, the computations involved in an optimized spline interpolation using equally spaced nodes are as efficient as the evaluation of the analytical fits, and more efficient than the evaluation of improved fits that use non-integer powers. For the Milstein-algorithm computations, we use the sampling and compounding methods described and demonstrated in the previous section.

The values of v from two sample trajectories, evolved with the Milstein algorithm and with a variety of time step values, were shown in Fig. 1. In the remainder of this section, we show other results from this implementation. We will first show a

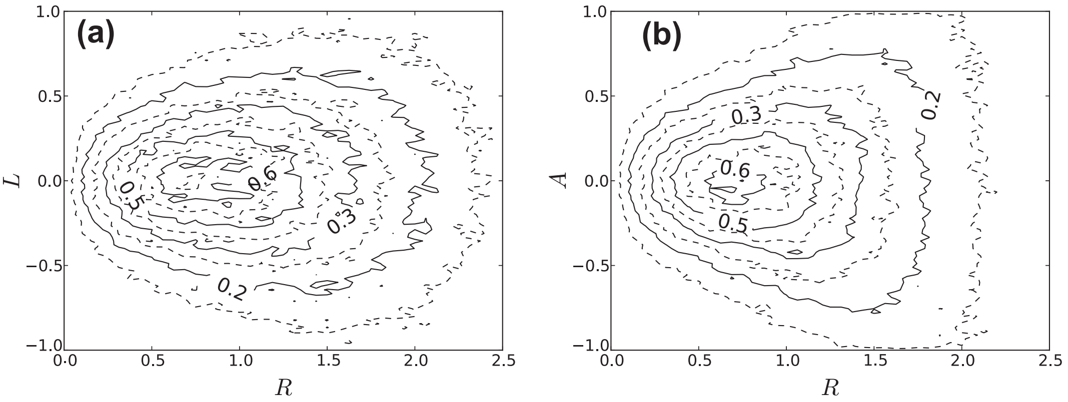


Fig. 13. 2D PDF's $p_{JL}(L,R)$ and $p_{JA}(A,R)$, from the compounded sets of numbers.

check that the correct averaged phase-space density is obtained in a simple and important test situation, and then that the Milstein algorithm gives the expected improvement in the scaling of the strong error over the Euler–Maruyama algorithm.

Fig. 14 shows the phase-space density ensemble of 10^5 particles at $t = 0$ and 4 evolved from an isotropic Maxwellian with temperature equal to that of the background (which appears in the calculation of $D_a(v),\ D_v(v)$ and $F_d(v)$), with the Milstein and Euler–Maruyama schemes. The initial condition is a set of 10^5 samples from a three-dimensional isotropic Maxwellian distribution with temperature equal to the field-particle temperature. As is seen from Fig. 1, each trajectory undergoes significant evolution over times of order 1. Nevertheless, no discernible evolution of the phase space density beyond statistical fluctuations is seen in Fig. 14. This test is consistent with the correctness of the basic model equations and (both Euler–Maruyama and Milstein) discretizations. Furthermore the lack of evolution of the v dependence of the phase space density is confirmation that the calculations (tabulation and interpolation) of the diffusion and drag coefficients $D_a(v),\ D_v(v)$ and $F_d(v)$, are accurate at least in that the relevant Einstein relations [31] (i.e., the relations between these quantities required to yield the correct equilibrium phase-space density) are well satisfied. Further tests of the average behavior of large groups of particles, and which quantify the accuracy of calculations of specific observables (or "payoffs" [19,21], as they are referred to in the financial mathematics literature) are underway and will be reported as a part of work on weak error properties and multi-level schemes based upon the present work [32].

Results on strong convergence for the two-dimensional $v - \mu$ Langevin Coulomb collision operator will now be shown and discussed. As described in the introduction (Eq. (2) and the associated discussion), the strong convergence studies that will be shown here examine differences for the same underlying trajectory between variable values, in particular the end-point values, from computations done with different time-step values. The mean-squared or RMS values of these differences averaged over an ensemble of trajectories give a measure of the strong convergence rate and are of direct importance for the use of the Milstein and Euler–Maruyama algorithms in Giles' multilevel-Monte-Carlo schemes [21].

Fig. 15 shows estimates of the strong errors in v in the solution versus Δt. These estimates are averages over a small ensemble of 100 realizations (trajectories) of the absolute values of the difference between the end point value for a given realization and time step, and the end-point value for the same realization run with the finest time step in the factor 3^j sequence of runs. There are three sets of points on each of the plots: Euler–Maruyama, "diagonal Milstein" (i.e., Milstein with

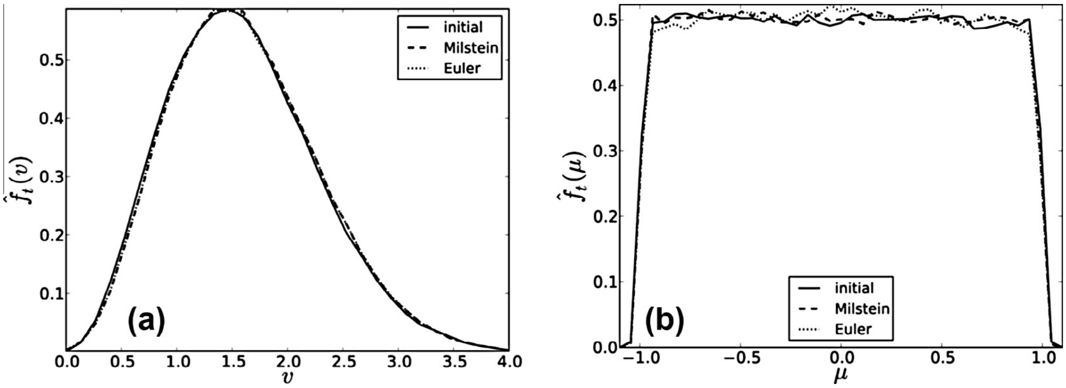


Fig. 14. Phase-space density versus (a) v and (b) μ at $t = 0$ and 4 for an ensemble of 10^5 particles evolved with the Milstein and Euler–Maruyama schemes using a time step value of 4×3^{-4}. The initial condition is a set of 10^5 samples from a three-dimensional isotropic Maxwellian distribution with temperature equal to the field-particle temperature.

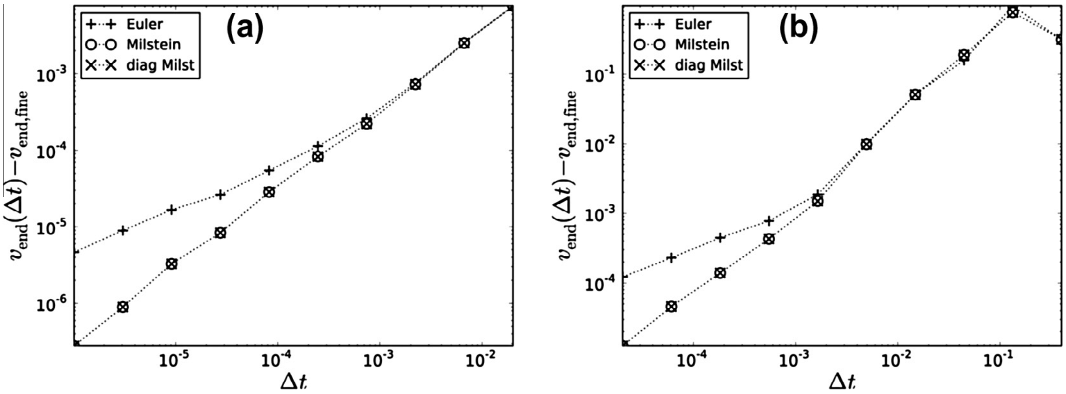


Fig. 15. RMS error in solution for v vs. time step for Euler, diagonal Milstein – i.e., without area integral term, and full Milstein, for (a) $t_{\rm end} = 0.02$ and (b) $t_{\rm end} = 0.4$.

only diagonal terms and no area integral term), and full Milstein with the area integral term included. These plots show the expected $O(\Delta t^{1/2})$ scaling of the error for the Euler–Maruyama scheme and $O(\Delta t)$ for both the diagonal and full Milstein schemes. The diagonal-Milstein and full-Milstein results are identical, which is to be expected because the evolution of v given by Eq. (23) is independent of μ and is unaffected by the area-integral term in Eq. (24).

Fig. 16 shows estimates of the strong errors in μ in the solution versus Δt, defined analogously to those for v and for the same computational run as in Fig. 15. Again, each plot has a set of points for each of the Euler–Maruyama, diagonal Milstein, and full Milstein schemes. The scaling of this error is $O(\Delta t^{1/2})$ for both the Euler–Maruyama and diagonal Milstein schemes, and $O(\Delta t)$ for full Milstein scheme. These results show (a) that the full Milstein scheme achieves the expected improvement in the scaling of the strong error in μ and (b) while the diagonal Milstein term in Eq. (24) has an effect on the μ evolution, it is insufficient to give improved strong error scaling. To achieve the benefit of the Milstein scheme for the μ evolution, the area-integral term in Eq. (24) must also be kept.

6. Summary and discussion

We have developed, implemented, and demonstrated the improved convergence of an extension of Langevin-equation Monte-Carlo algorithms for Coulomb collisions from the conventional Euler–Maruyama time integration to the next higher order of accuracy, the Milstein scheme. Results from the numerical implementation show the expected improvement [$O(\Delta t)$ vs. $O(\Delta t^{1/2})$] in the strong convergence rate both for the speed ($|\mathbf{v}|$) and angular components of the scattering. An important result is that this improved convergence is achieved for the angular component of the scattering if and only if the "area-integral" terms in the Milstein scheme are included.

The resulting Milstein scheme is of value both as a step towards algorithms with improved accuracy and efficiency either directly through algorithms with improved convergence in the averages (weak convergence) or as a building block for multi-time-level schemes [21]. The latter have been shown to give greatly reduced cost for a given overall error level compared with conventional Monte-Carlo schemes, and their performance is improved considerably when the Milstein algorithm is used for the underlying time advance versus the Euler–Maruyama algorithm.

A key aspect of the extension is that it proceeds via a formulation of the angular scattering directly as SDE's in the fixed-frame spherical-coordinate velocity variables. We have attempted a Milstein extension of the quite widely used class of schemes [9,11,12] in which the angular step is formulated in a frame aligned with the velocity at the start of the step. This extension is partially successful in that improved strong convergence for the speed v is easily obtained. However, we have not succeeded in obtaining improved strong convergence for the angular component(s) of the scattering. We have traced this difficulty to the fact that, while the Milstein extension of a single velocity step is straightforward, the aggregation of such steps also involves the rotation of the coordinate system between applications of the velocity steps. Various choices are possible for the choice of orientation of the unit vectors/axes perpendicular to the velocity direction. Examples of such choices include choosing one of the perpendicular unit vectors to be (a) perpendicular to the velocity direction and a fixed plane or (b), in the $\nabla\theta$ direction where θ is the polar angle with respect to a fixed frame or (c) rotating the system of three orthogonal unit vectors as a rigid body about a single axis. None of these choices yields a scaling of the error in the angular component of the scattering that is faster than that for the Euler–Maruyama scheme.

We have also developed a new method for sampling the area integrals which is a simplification of an earlier direct method by Gaines and Lyons [29] and which retains high accuracy. This method is useful in its own right because of its relative simplicity compared with the method of Ref. [29]. We expect our method to be useful also for other applications that benefit from Milstein and higher-order strong schemes, including chemical physics and financial modeling. Our method is also expected to considerably reduce the computational requirements for the sampling of the area integrals that is needed for adaptive strong integration. For adaptive integration (as opposed to the error scaling tests reported here), the random

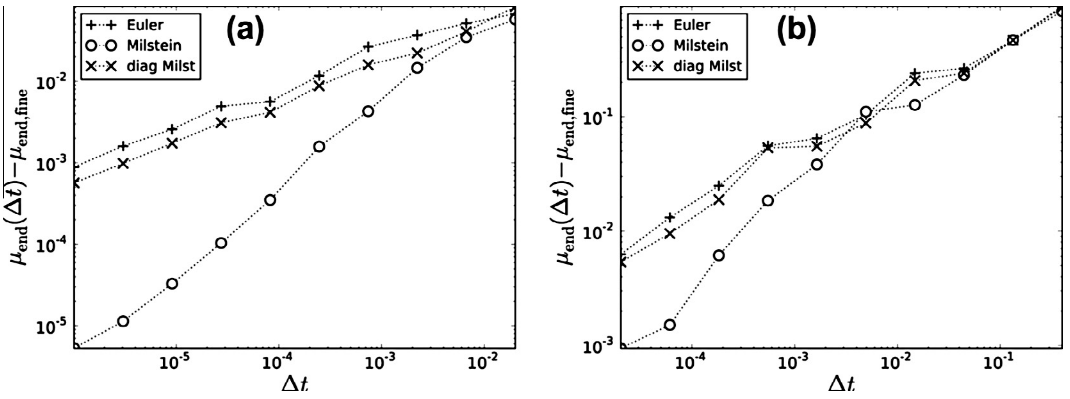


Fig. 16. RMS error in solution for μ vs. time step for Euler, diagonal Milstein, and full Milstein, for (a) $t_{\mathrm{end}} = 0.02$ and (b) $t_{\mathrm{end}} = 0.4$.

numbers must be conditionally sampled at a finer (time-step) level, given the corresponding numbers at the coarser level. For the Gaussian displacements, this conditional sampling is straightforward. For a multi-dimensional Milstein integrator, one needs to sample the finer triplets of the two Gaussian Wiener displacements and area integral given the triplets associated with the coarser time step. A method for doing this based on quadrature formulas has been given by Gaines and Lyons [27], but a direct method is also possible and is expected to be much cheaper computationally for a given level of accuracy. Such a direct sampling based on the full conditional PDF of Eqs. (28) and (29) involves a 4-dimensional sampling function. If the conditional sampling is instead based on our approximate form given in Eq. (34), the dimensionality of the conditional sampling function is reduced to 3 [33]. This reduction from 4 to 3 dimensions represents a significant reduction in the computational memory requirement for an accurate tabulation of the conditional sampling function. Finally, we note that in a system of SDE's such as that of Eq. (7) of dimensionality D, there are in general $D(D-1)/2$ distinct Levy areas, and these are not independent of each other. Gaines and Lyons [29] outline a method for generating the Levy areas for the case of general D given a direct procedure for the $D = 2$ (single-Levy-area) case. This method, additionally, requires the generation of a random $D \times D$ rotation matrix, i.e., element of the orthogonal group O_D. Straightforward methods for doing the latter are given, for example, in Ref. [34].

Acknowledgments

Work performed for US DOE by LLNL under Contract DE-AC52–07NA27344, and by UCLA under Grant DE-FG02-05ER25710.

Appendix A. Consistency of Eqs. (9)–(12) with the work of Lemons et al. [11]

Here, some of the details of the comparison of Eqs. (9)–(12) with work of Lemons et al. [11] are given. Eqs. (4a) and (4c) of Ref. [11] are the (stochastic differential) equations of motion for the test particle, expressed in terms of its speed v and polar angle θ with respect to a polar axis oriented in the current direction of the velocity:

$$dv(t) = -\beta(v)v\,dt + \sqrt{\delta^2}\,dW_v(t), \tag{35}$$

$$d\theta(t) = \sqrt{2\gamma(v)}\,dW_\theta(t). \tag{36}$$

A slightly different normalization of the quantities is used here than in Ref. [11], with $W_v(t)$ and $W_\theta(t)$ being independent Wiener processes with the normalization as given by Eq. (5). Eqs. (17b) and (17c) of Lemons et al. give respectively the drag and diffusion coefficients for the particle speed (magnitude of the velocity), (17a) is the equation for the angular diffusion coefficient, and

$$\beta(v) = \frac{A_D}{2v^3}\left\{\left[\left(1+\frac{m_t}{m_f}\right)2l_f^2v^2+1\right]G(l_f v) - \Phi(l_f v)\right\}, \tag{37}$$

$$\delta^2(v) = \frac{A_D G(l_f v)}{v},$$

$$\gamma(v) = \frac{A_D}{2v^3}[\Phi(l_f v) - G(l_f v)]. \tag{38}$$

A derivation of Eq. (37) that is perhaps slightly more direct than, but is equivalent to, that of Lemons et. al. is the following. We can express the change $\delta v = v(t+\delta t) - v(t)$ in the particle speed v over a time δt in terms of the change $\delta\mathbf{v} = \mathbf{v}(t+\delta t) - \mathbf{v}(t)$ in its velocity $\mathbf{v}$ up to second order as

$$\delta v = \sqrt{(\mathbf{v}+\delta\mathbf{v})^2} - v \approx \hat{\mathbf{v}}\cdot\delta\mathbf{v} + \frac{1}{2v}\delta\mathbf{v}\cdot(\mathbf{I}-\hat{\mathbf{v}}\hat{\mathbf{v}})\cdot\delta\mathbf{v},$$

where $\hat{\mathbf{v}} \equiv \mathbf{v}/v$ is the unit vector in the direction of $\hat{\mathbf{v}}$ and $\mathbf{I}$ is the 3-dimensional unit tensor. It follows that

$$\langle\delta v\rangle = \hat{\mathbf{v}}\cdot\langle\delta\mathbf{v}\rangle + \frac{1}{2v}(\mathbf{I}-\hat{\mathbf{v}}\hat{\mathbf{v}}):\langle\delta\mathbf{v}\delta\mathbf{v}\rangle \tag{39}$$

$$= \langle\delta v_\parallel\rangle + \frac{1}{2v}\langle\delta v_\perp^2\rangle. \tag{40}$$

Eq. (37) then follows from Eqs. (5a) and (5c) of Lemons et al. and Eq. (40).

Note that in the statement of the algorithm of Ref. [11], θ represents the polar angle with respect to an axis that is oriented in the direction of the test particle velocity at the beginning of each time step. Eq. (36) is therefore not the SDE corresponding to the polar-angle diffusion term in Eq. (8).

References

[1] F.L. Hinton, R.D. Hazeltine, Rev. Mod. Phys. 48 (1976) 239–308.

[2] John D. Lindl, Peter Amendt, Richard L. Berger, S. Gail Glendinning, Siegfried H. Glenzer, et al, Phys. Plasmas 11 (2004) 339, http://dx.doi.org/10.1063/1.1578638.
[3] Michael A. Lieberman, Allan J. Lichtenberg, Principles of Plasma Discharges and Materials Processing, John Wiley & Sons, 2005.
[4] V.K. Jordanova, L.M. Kistler, J.U. Kozyra, G.V. Khazanov, A.F. Nagy, J. Geophys. Res. 101 (1996) 111–126, http://dx.doi.org/10.1029/95JA02000.
[5] L.D. Landau, Phys. Z. Sowjet 10 (1936) 154;
L.D. Landau, JETP 7 (1937) 203.
[6] B.A. Trubnikov, in: M.A. Leontovich (Ed.), Reviews of Plasma Physics, vol. 1, Consultants Bureau, New York, 1965, p. 105.
[7] Charles K. Birdsall, A. Bruce Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, 1985.
[8] Roger W. Hockney, James W. Eastwood, Computer Simulation using Particles, Taylor and Francis, 1988.
[9] W.M. Manheimer et al, J. Comput. Phys. 138 (1997) 563–584.
[10] M. Sherlock, J. Comput. Phys. 227 (2008) 2286–2292.
[11] D.S. Lemons, D. Winske, W. Daughton, B. Albright, J. Comput. Phys. 228 (2009) 1391–1403.
[12] B.I. Cohen et al, IEEE Trans. Plasma Sci. 38 (2010).
[13] T. Takizuke, H. Abe, J. Comput. Phys. 25 (1977) 205–219.
[14] K. Nanbu, Phys. Rev. E55 (1997) 4642.
[15] A.M. Dimits, C.M. Wang, R.E. Caflisch, B.I. Cohen, Y. Huang, J. Comput. Phys. 228 (2009) 4881.
[16] J. Killeen, G.D. Kerbel, M.G. McCoy, A.A. Mirin, Computational Methods for Kinetic Models of Magnetically Confined Plasmas, Springer-Verlag, New York, 1986.
[17] Z. Xiong, R.H. Cohen, T.D. Rognlien, X.Q. Xu, J. Comput. Phys. 227 (2008).
[18] I.G. Abel, M. Barnes, S.C. Cowley, W. Dorland, A.A. Schekochihin, Phys. Plasmas 15 (2008) 122509.
[19] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, Berlin, 1992.
[20] G.N. Milstein, Numerical Integration of Stochastic Differential Equations, Kluwer Academic, Dordrecht, 1995.
[21] M.B. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, in: A. Keller, S. Heinrich, H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer-Verlag, 2007, p. 343.
[22] S. Painter, Comput. Phys. Commun. 77 (1993) 342–356.
[23] S.A. Dettrick et al, Aust. J. Phys. 52 (1999) 715–732.
[24] S. Chandrasekhar, Principles of Stellar Dynamics, University of Chicago Press, 1942. pages 63, 72, 73.
[25] L. Spitzer, Physics of Fully Ionized Gases, John Wiley and Sons, 1962. pages 128, 129.
[26] M.N. Rosenbluth, W.M. MacDonald, D.L. Judd, Phys. Rev. 107 (1957) 1.
[27] J.G. Gaines, T.J. Lyons, SIAM J. Appl. Math. 57 (1997) 1455.
[28] P. Levy, Proc. Second Berkeley Symp. Math. Stat. and Prob., vol. 2, University of California Press, Berkeley, CA, 1951.
[29] J.G. Gaines, T.J. Lyons, SIAM J. Appl. Math. 54 (1994) 1132.
[30] D.E. Knuth, Seminumerical Algorithms, The Art of Computer Programming, vol. 2, Addison-Wesley, Reading, MA, 1973.
[31] A. Einstein, Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen, Annalen der Physik 322 (1905) 549–560.
[32] M.S Rosin, L. Ricketson, A.M. Dimits, B.I. Cohen, R.E. Caflisch, in preparation.
[33] A.M. Dimits, B.I. Cohen, R.E. Caflisch, L. Ricketson, M.S. Rosin, in preparation.
[34] F. Mezzadri, Not. Am. math. Soc. 54 (2007) 592–604.