

A Gear-like Predictor-Corrector Method for Brownian Dynamics Simulation

Lu-Jing Hou^{1,2} and Z. L. Mišković¹

¹*Department of Applied Mathematics,*

University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

²*Institut für Experimentelle und Angewandte Physik,*

Christian-Albrechts Universität zu Kiel, D-24098 Kiel, Germany

(Dated: July 25, 2008; Received; Published)

Abstract

We introduce a Predictor-Corrector type method suitable for performing many-particle Brownian Dynamics simulations. Since the method goes over to the Gear's method for Molecular Dynamics simulation in the limit of vanishing friction, we refer to it as a Gear-like algorithm. The algorithm has been tested on a one-dimensional, stochastically damped harmonic oscillator model, showing that it can cover a wide range of friction coefficients with a high-order accuracy, excellent stability, and a very small energy drift on the long time scales.

PACS numbers: 02.70.-c, 47.57.-s, 52.40.Hf, 52.25.Vy

I. INTRODUCTION

Brownian Dynamics (BD) simulation method [1, 2, 3, 4] has been widely used in studying problems in many dispersed systems, such as polymer solutions [5], colloidal suspensions [6, 7, 8], and, more recently, complex plasmas [9, 10, 11, 12, 13], and has been particularly regarded as the "mainstay" of colloid modeling over the past two decades.

BD method may be regarded as a generalization of the usual Molecular Dynamics (MD) method, which is a viewpoint that will be held throughout the present paper. As the MD method is based on Newton's equations of motion, the BD method is based on their stochastic generalization, namely, the Langevin equation and its integral (or Kramers equation written in Hamilton form),

$$\begin{aligned}\frac{d}{dt}\mathbf{r} &= \mathbf{v} \\ \frac{d}{dt}\mathbf{v} &= -\gamma\mathbf{v} + \frac{1}{m}\mathbf{F} + \mathbf{A}(t).\end{aligned}\tag{1}$$

Here, as usual, m , \mathbf{v} and \mathbf{r} are respectively the mass, velocity and the position of a Brownian particle, and \mathbf{F} is the systematic (deterministic) force resulting from external sources and/or from the inter-particle interactions in a many-particle system of Brownian particles interacting with the ambient gas of light particles. What is different from Newton's equations is the appearance of dynamic friction, $-\gamma\mathbf{v}$, and the stochastic (Brownian) acceleration, $\mathbf{A}(t)$. As already pointed out by Langevin, those two terms represent complementary effects of the same microscopic phenomenon: numerous, frequent collisions between the Brownian particle and molecules in the surrounding medium. While friction represents an average effect of those collisions, the stochastic acceleration represents fluctuations due to the discreteness of collisions with the ambient particles, and is represented by a delta-correlated Gaussian white noise. When the system is in thermal equilibrium, the friction and the stochastic acceleration are related to the ambient temperature via fluctuation-dissipation theorem. Note that the friction coefficient γ is usually regarded as constant [4].

The Langevin equation (1) may be integrated numerically in a manner similar to that used for Newton's equations in MD simulations [4, 14, 15, 16, 17], and the resulting technique is usually called Brownian Dynamics. However, when compared to the well-established MD techniques, algorithms for conducting BD simulations are considerably less well developed because of their stochastic nature. It should be noted that, besides the term Brownian Dynamics, one finds in literature other names that are used to designate various techniques of numerical integration of

Langevin equation, such as Langevin Dynamics, or Langevin Molecular Dynamics, depending on the background, area of application, or simply preferences of various authors [15]. It is beyond the scope of this paper to discuss all such techniques and subtle differences among them in the derivation of the simulation formulae and/or in their implementations. Therefore, to be specific, we are going to stick to the definitions used by Allen and Tildesley [4], and to the strategies proposed by Ermak *et al.* [1], Allen [2], van Gunsteren and Berendsen [3], which are so far the most popular BD methods used in practice.

A straightforward method of conducting the BD simulation based on Eq. (1) was developed by Ermak *et al.* [1], in which equations of motion are integrated over a time interval t under the assumption that the systematic force \mathbf{F} remains approximately constant over that interval. Such method is often referred to as an Euler-like method in the sense that it goes over to the Euler's scheme of standard MD simulation, in the limit of an infinitesimally small γ . In spite of the low efficiency of Euler's method in the MD simulation, the Euler-like method in BD simulation seems to be very robust for very large γ , and is still widely used in practice because of its simplicity [5, 7, 8]. However, very small time steps are required in this method to achieve bearable inaccuracy and energy drift, as will be demonstrated below.

Ermak's method can be extended to higher order schemes by keeping higher-order terms in a Taylor series of the deterministic force \mathbf{F} or, equivalently in the acceleration $\mathbf{a} \equiv \mathbf{F}/m$, as follows

$$\mathbf{a}(t) = \mathbf{a}(0) + \dot{\mathbf{a}}(0)t + \frac{1}{2!}\ddot{\mathbf{a}}(0)t^2 + \frac{1}{3!}\dddot{\mathbf{a}}(0)t^3 + \cdots + \frac{1}{n!}\mathbf{a}^{(n)}(0)t^n + \cdots, \quad (2)$$

where $\mathbf{a}^{(n)}$ represents the n th-order time derivative of \mathbf{a} . For example, when integrating Eq. (1), Allen [2, 4] obtained a scheme by effectively keeping up to the second-order derivative in Eq. (2). His method is reduced to the Beeman's method for MD simulation when $\gamma \rightarrow 0$, and is therefore appropriately referred to as a Beeman-like method for BD simulation. We note that this method has a 3rd order accuracy for both the velocity and position, using the definition of Ref. [14] in which the order of an algorithm is given by the highest order of the time step in simulation formulae. Another algorithm for integrating Eq. (1), which also gained popularity in practice, was developed by van Gunsteren and Berendsen [3], by using similar assumption as Allen [2, 4]. Since their algorithm is reduced to the position Verlet algorithm for MD simulation in the limit $\gamma \rightarrow 0$, we shall refer to it as Verlet-like method for BD simulation. It was proven [2, 4] that the Verlet-like method is numerically equivalent to the Beeman-like method in position, whereas the latter has higher accuracy in velocity.

Over the years, much effort has been invested in searching for better and higher-order algorithms for BD simulation [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. For example, similar idea of truncating the Taylor expansion was adopted by Helfand [18], Iniesta and Garcia de la Torre [22], and Honeycutt [24] to obtain second-order Runge-Kutta-like methods, and more recently by Hershkovitz [25] to obtained a fourth-order Runge-Kutta-like method. However, most of those algorithms [18, 22, 24, 25, 26] are based on direct extensions of the deterministic Runge-Kutta algorithms to include some stochastic terms [24], or on solving the deterministic part of the Langevin equation using standard Runge-Kutta package [25]. Consequently, such approaches require more than one evaluation of the deterministic force per time step, which clearly reduces their efficiency for simulating problems in many-particle systems [4, 14, 15, 16, 17].

More recently, Brańka and Heyes [26] developed a series of algorithms to improve the efficiency of BD simulation based on a finite step-size expansion for stochastic differential equations (SDEs), which may be therefore also classified as second-order Runge-Kutta-like methods. It was found that such algorithms could significantly increase the efficiency of simulation [26]. However, they are mainly designed for simulating over-damped systems, such as colloidal suspensions, as they are based on the so-called position Langevin equation [26], in which the variation of the velocity [i.e., the left-hand-side of the second line in Eq. (1)] is neglected [18]. Such simplification appears to be quite adequate for simulating polymer solutions and colloidal suspensions, which are always over-damped, but may not be suitable for simulating complex plasmas, which are typically only lightly damped [9, 10, 11, 12, 13].

Currently, the most advanced BD algorithms [27, 28, 30] (to the best of our knowledge) are based on operator expansions of time propagators for Fokker-Plank or Kramers equations using Trotter decomposition technique [15, 16, 17, 21, 23, 27, 28, 29, 30, 31, 32]. When $\gamma \rightarrow 0$, these methods reduce to the class of so-called symplectic integrators for Hamiltonian dynamics, which have the unique properties of conserving the phase-space volume and being time reversible. These properties result in excellent stability properties and only very small energy drift on the long time scales [16]. However, all such methods [21, 23, 27, 28, 29, 30, 31, 32] also use the standard Runge-Kutta algorithm to solve the deterministic part of dynamics and they, too, need multiple evaluations of deterministic forces per time step. As a consequence, they can become computationally quite expensive when used in simulation of many-particle systems, which restricts their applicability to systems of limited size, thus justifying further quest for more efficient algorithms of a comparable order.

On the other hand, the higher-order Predictor-Corrector (PC) methods, apparently, have not attracted much attention in BD simulation in comparison to their popularity in MD simulation [4, 14]. A PC method for BD was proposed by Öttinger in 1996 [5], and was subsequently adopted by Hütter [6] in a simulation of colloidal suspensions. However, in their method the first step (predictor) was just a first-order Euler-like scheme, which obviously does not utilize the full spectrum of advantages offered by the PC method.

Consequently, we propose in the present paper a class of general purpose PC algorithms, which cover a wide range of friction coefficient γ with higher-order accuracy, excellent stability, and very small energy drift on the long time scales. These methods reduce to Gear's methods for MD simulation in the limit of $\gamma \rightarrow 0$ [4, 14]. We shall discuss here only Gear-like algorithms that go up to the fifth-order, but extensions to higher orders should be quite straightforward, if required. The proposed Gear-like algorithms will be tested on simulation of a simple, one-particle system, and numerical comparisons will be made mainly with the Euler-like and Beeman-like (Verlet-like) methods because all these methods are based on a Taylor expansion of the deterministic force, Eq. (2), and they all require only one force evaluation per time step. As such, all these methods are specifically designed for simulations of large many-particle systems. Further discussion of advantages of the proposed algorithms over those involving higher-order Runge-Kutta-like methods [21, 23, 27, 28, 30] will be provided latter in the text.

The paper is organized as follows. In Sec. II, we give detailed algorithms for conducting BD simulation. We next perform in Sec. III numerical tests for different methods (Euler-, Beeman- and Gear-like) by using a stochastically damped harmonic oscillator (SDHO) model [33, 34]. Discussions of possible derivatives of the proposed BD method, as well as comparisons with the high-order Runge-Kutta-like methods are given in Sec. IV, which is followed by a brief conclusion in Sec. V.

II. ALGORITHMS FOR BD SIMULATION

A. General formula

There are various ways to derive formulae for conducting BD simulation starting from Langevin equation Eq. (1). We shall follow here the strategy adopted in Refs. [1, 2, 3, 4], and more recently in Refs. [34, 35], because it is simple and straightforward, especially for readers

with some background in simulation but lacking background in SDEs. Following the technique of integrating the Langevin equation which is elaborated in Ref. [34], one realizes that Eq. (1) can be integrated exactly over a short time, based on certain assumptions about the deterministic acceleration $\mathbf{a}(t)$, thus giving updating formulae for BD simulation. The resulting formulae emphasize the fact that, under the assumptions that the stochastic acceleration in Eq. (1) is a Gaussian white noise and that γ is constant, the two dynamic variables, $\mathbf{v}(t)$ and $\mathbf{r}(t)$, are actually normally distributed random variables themselves [or Gaussian random variables (GRVs)]. Consequently, according to the *Normal linear transform theorem* [34, 35], $\mathbf{v}(t)$ and $\mathbf{r}(t)$ are completely determined by their instantaneous means (vectors) and variances (scalars), and can be expressed, respectively, as follows

$$\begin{aligned}\mathbf{v}(t) &= \text{mean}\{\mathbf{v}(t)\} + \sqrt{\text{var}\{\mathbf{v}(t)\}} \mathbf{N}_{\mathbf{v}}(0, 1) \\ \mathbf{r}(t) &= \text{mean}\{\mathbf{r}(t)\} + \sqrt{\text{var}\{\mathbf{r}(t)\}} \mathbf{N}_{\mathbf{r}}(0, 1).\end{aligned}\tag{3}$$

Here, $\mathbf{N}(0, 1)$ is a short-hand notation for a random vector [$\mathbf{N} = \{N_x, N_y, N_z\}$ in three-dimensions (3D) or $\mathbf{N} = \{N_x, N_y\}$ in two-dimensions (2D)], whose components are mutually independent, standard (or unit) normal distributions (i.e., having the mean 0 and variance 1). The subscripts \mathbf{v} and \mathbf{r} in Eq. (3) emphasize that the two random normal vectors are associated with the velocity and position, respectively, and that they are different but correlated, as will be shown below. In the following, for the sake of simplicity, we shall occasionally denote the means by angular brackets, $\langle \mathbf{v} \rangle$ and $\langle \mathbf{r} \rangle$, and use standard deviations $\sigma_v = \sqrt{\text{var}\{\mathbf{v}\}}$ and $\sigma_r = \sqrt{\text{var}\{\mathbf{r}\}}$, instead of variances.

We emphasize the significance of Eq. (3) because it provides the general updating formula for BD simulation. Namely, with Eq. (3), the process of solving the stochastic differential equation Eq. (1) and obtaining the two random variables $\mathbf{v}(t)$ and $\mathbf{r}(t)$ becomes simply a matter of determining the two sets of deterministic quantities: the means and the variances of $\mathbf{v}(t)$ and $\mathbf{r}(t)$, as well as the covariance between $\mathbf{v}(t)$ and $\mathbf{r}(t)$. Most importantly, the variances and covariances can be shown to be independent of the form of deterministic acceleration $\mathbf{a}(t)$. For variances, we find [1, 2, 4, 34, 35]:

$$\begin{aligned}\sigma_v &= \sqrt{\text{var}\{\mathbf{v}\}} = \sqrt{\frac{k_B T}{m} (1 - e^{-2\gamma t})} \\ \sigma_r &= \sqrt{\text{var}\{\mathbf{r}\}} = t \sqrt{\frac{2k_B T}{m\gamma t} \left(1 - 2\frac{1 - e^{-\gamma t}}{\gamma t} + \frac{1 - e^{-2\gamma t}}{2\gamma t} \right)},\end{aligned}\tag{4}$$

where k_B is the Boltzmann constant and T the temperature of the surrounding medium. In passing, we note that, when a Brownian particle is in equilibrium with the medium, T also defines its kinetic temperature. However, a Brownian particle may have a kinetic temperature different from T , which opens the possibility of simulating non-equilibrium processes by using BD, such as particle stopping in a medium [11].

Further, the only non-zero covariances are $\text{cov}\{v_x, x\} = \text{cov}\{v_y, y\} = \text{cov}\{v_z, z\} \equiv \text{cov}\{v, r\}$, and are given by (note, $\text{cov}\{v, r\} \equiv \langle vr \rangle - \langle v \rangle \langle r \rangle$) [34, 35]:

$$\text{cov}\{v, r\} = t \frac{k_B T}{m \gamma t} (1 - 2e^{-\gamma t} + e^{-2\gamma t}). \quad (5)$$

Since the velocity and position are not independent, but rather jointly distributed normal variables, the updating formulae Eq. (3) can be further written as [34]

$$\begin{aligned} \mathbf{r}(t) &= \langle \mathbf{r} \rangle + b_1 \mathbf{N}_1(0, 1) + b_2 \mathbf{N}_2(0, 1), \\ \mathbf{v}(t) &= \langle \mathbf{v} \rangle + \sigma_v \mathbf{N}_1(0, 1), \end{aligned} \quad (6)$$

with $b_1 = \text{cov}\{v, r\}/\sigma_v$ and $b_2 = \sqrt{\sigma_r^2 - b_1^2}$, where the unit normals $\mathbf{N}_1(0, 1)$ and $\mathbf{N}_2(0, 1)$ are now, by design, statistically (component-wise) independent of each other, and may be generated, e.g., by the Box-Muller method [36] on a computer.

It is worthwhile pointing out that the updating formulae Eq. (6) can be interpreted as a two-step algorithm: first, one calculates the means of the velocity and the position at time t and, second, one adds random increments of the velocity and position. Similar idea may also be seen in the Fokker-Planck based methods, e.g., in Ref. [28]. This notion is very important in the sense that the first step is no more than evaluation of deterministic Newton's equations with the addition of an average friction force, so that, in principle, any algorithm suitable for MD simulation (for example, Verlet, Beeman and even multi-step PC algorithms [4, 14]) may be used here. Moreover, with the second step being essentially independent of the first step, it can be always completed at the end of the time step as a correction to the previous step. More importantly, in the above two steps, the inaccuracy, or error, enters only in the first step through assumptions made regarding the deterministic acceleration $\mathbf{a}(t)$ [14], whereas in the second step, random increments of the velocity and position are added with the weights obtained from Eqs. (4,5), which are exact and independent of the assumptions employed in the first step.

Now, the only problem that remains is to determine the means of $\mathbf{v}(t)$ and $\mathbf{r}(t)$, which are closely dependent on the assumptions made about $\mathbf{a}(t)$, and which result in different simulation methods. We provide details for the Gear-like methods in the following.

B. Gear-like Predictor-Corrector method

Given the form of $\mathbf{a}(t)$, defined by the manner of how is the Taylor series in Eq. (2) truncated, $\langle \mathbf{v} \rangle$ and $\langle \mathbf{r} \rangle$ can be obtained in a number of ways by integrating Eq. (1). For simplicity, we follow here Lemons' methodology [34, 35], in which Eq. (1) is reduced to a set of deterministic (ordinary) differential equations by taking the statistical averages of both sides, giving

$$\begin{aligned}\frac{d\langle \mathbf{r} \rangle}{dt} &= \langle \mathbf{v} \rangle \\ \frac{d\langle \mathbf{v} \rangle}{dt} &= -\gamma \langle \mathbf{v} \rangle + \mathbf{a}(t),\end{aligned}\tag{7}$$

where the fact that the Brownian acceleration $\mathbf{A}(t)$ is a Gaussian white noise with the mean 0 was used. It is also important to note that the second equation in (7) may be further simplified by letting $\langle \mathbf{v} \rangle = e^{-\gamma t} \langle \mathbf{v} \rangle'$, so that

$$e^{\gamma t} \frac{d\langle \mathbf{v} \rangle'}{dt} = \mathbf{a}(t).\tag{8}$$

Eq. (7) can be solved exactly with the help of Eq. (8) and, subject to the initial conditions \mathbf{r}_0 , \mathbf{v}_0 , \mathbf{a}_0 , $\dot{\mathbf{a}}_0$, $\ddot{\mathbf{a}}_0$ and $\mathbf{a}_0^{(n)}$ at $t = 0$, it yields

$$\begin{aligned}\langle \mathbf{r} \rangle &= \mathbf{r}_0 + c_1 \mathbf{v}_0 t + c_2 \mathbf{a}_0 t^2 + c_3 \dot{\mathbf{a}}_0 t^3 + c_4 \ddot{\mathbf{a}}_0 t^4 + \dots + c_n \mathbf{a}_0^{(n-2)} t^n + \dots, \\ \langle \mathbf{v} \rangle &= c_0 \mathbf{v}_0 + c_1 \mathbf{a}_0 t + c_2 \dot{\mathbf{a}}_0 t^2 + c_3 \ddot{\mathbf{a}}_0 t^3 + c_4 \dddot{\mathbf{a}}_0 t^4 + \dots + c_n \mathbf{a}_0^{(n-1)} t^n + \dots.\end{aligned}\tag{9}$$

It is interesting to note that the coefficients in Eq. (9) obey very simple recursive relations. For $n = 0$, we have $c_0 = \exp(-\gamma t)$, while for $n \geq 1$, we find

$$\begin{aligned}c_n &= \frac{1}{(n-1)! \gamma t} [1 - (n-1)! c_{n-1}], \\ \frac{d}{dt}(c_n t^n) &= c_{n-1} t^{n-1}.\end{aligned}\tag{10}$$

As expected, these coefficients reduce to those of a standard Taylor series when $\gamma \rightarrow 0$.

To construct the Gear-like method for BD simulation, one also needs higher order derivatives of the velocity (or position) [37]. In principle, that may be attempted directly, by taking successively time derivatives on both sides of the second equation in Eq. (7), but that approach turns out to be

awkward. Instead, by using the equivalent equation Eq. (8), one finds

$$\begin{aligned}
e^{-\gamma t} \frac{d\langle \mathbf{v} \rangle'}{dt} &= \mathbf{a}(t) = \mathbf{a}_0 + \dot{\mathbf{a}}_0 t + \frac{1}{2!} \ddot{\mathbf{a}}_0 t^2 + \frac{1}{3!} \dddot{\mathbf{a}}_0 t^3 + \dots \\
\frac{d}{dt} \left[e^{-\gamma t} \frac{d\langle \mathbf{v} \rangle'}{dt} \right] &= \dot{\mathbf{a}}(t) = \dot{\mathbf{a}}_0 + \ddot{\mathbf{a}}_0 t + \frac{1}{2!} \dddot{\mathbf{a}}_0 t^2 + \dots \\
\frac{d^2}{dt^2} \left[e^{-\gamma t} \frac{d\langle \mathbf{v} \rangle'}{dt} \right] &= \ddot{\mathbf{a}}(t) = \ddot{\mathbf{a}}_0 + \dddot{\mathbf{a}}_0 t + \dots \\
\frac{d^3}{dt^3} \left[e^{-\gamma t} \frac{d\langle \mathbf{v} \rangle'}{dt} \right] &= \dddot{\mathbf{a}}(t) = \dddot{\mathbf{a}}_0 + \dots
\end{aligned} \tag{11}$$

Eq. (11), together with Eq. (7), constitutes a Taylor-like series which can be used directly to construct the Gear-like PC method, as follows.

In a direct analogy with simulations in deterministic systems, the Gear-like PC methods for BD simulation also include three stages, namely, predicting, force evaluating, and correcting [4, 14, 37]. The difference is that, one has to add random displacements of the velocity and position by using Eq. (6) at the end of a time step to complete the BD simulation. The basic procedure goes as follows.

Predicting: In this stage, one has:

$$\begin{aligned}
\mathbf{r}^P &= \mathbf{r}_0 + c_1 \mathbf{v}_0 t + c_2 \mathbf{a}_0 t^2 + c_3 \dot{\mathbf{a}}_0 t^3 + c_4 \ddot{\mathbf{a}}_0 t^4 + c_5 \dddot{\mathbf{a}}_0 t^5 \\
\mathbf{v}^P &= c_0 \mathbf{v}_0 + c_1 \mathbf{a}_0 t + c_2 \dot{\mathbf{a}}_0 t^2 + c_3 \ddot{\mathbf{a}}_0 t^3 + c_4 \dddot{\mathbf{a}}_0 t^4 \\
\mathbf{a}^P &= \mathbf{a}_0 + \dot{\mathbf{a}}_0 t + \frac{1}{2!} \ddot{\mathbf{a}}_0 t^2 + \frac{1}{3!} \dddot{\mathbf{a}}_0 t^3 \\
\dot{\mathbf{a}}^P &= \dot{\mathbf{a}}_0 + \ddot{\mathbf{a}}_0 t + \frac{1}{2!} \dddot{\mathbf{a}}_0 t^2 \\
\ddot{\mathbf{a}}^P &= \ddot{\mathbf{a}}_0 + \dddot{\mathbf{a}}_0 t \\
\dddot{\mathbf{a}}^P &= \dddot{\mathbf{a}}_0,
\end{aligned} \tag{12}$$

where the superscript P is to indicate that these are quantities in the predicting stage. (For simplicity, we have dropped derivatives of $\mathbf{a}(t)$ higher than the third order, but extensions to higher orders are quite straightforward.) One may notice in Eq. (12) that we have used in Eq. (9) the means of the position and velocity instead of using Taylor series of the position and velocity, as is normally done in the Gear method for MD simulation [4, 14]. Other than that, the remaining parts of this stage (derivatives of the force) are essentially the same as those in the MD simulation.

Force evaluating: In this stage, the predicted position $\mathbf{r}^P(t)$ is used to obtain a new force, that is, new acceleration $\mathbf{a}(t)$, and a difference between the predicted acceleration $\mathbf{a}^P(t)$ and the new

acceleration $\mathbf{a}(t)$ is formed:

$$\Delta \mathbf{a} = [\mathbf{a}(t) - \mathbf{a}^P(t)]. \quad (13)$$

It can be seen that this step is exactly the same as in MD simulation [4, 14].

Correcting: In this stage, the above difference term is used to correct all predicted positions and their "derivatives", as follows:

$$\begin{aligned} \mathbf{r}^C &= \mathbf{r}^P + 2c_2\alpha_0\Delta\mathbf{R}2 \\ \mathbf{v}^C t &= \mathbf{v}^P t + c_1\alpha_1\Delta\mathbf{R}2 \\ \frac{\mathbf{a}^C t^2}{2!} &= \frac{\mathbf{a}^P t^2}{2!} + \alpha_2\Delta\mathbf{R}2 \\ \frac{\dot{\mathbf{a}}^C t^3}{3!} &= \frac{\dot{\mathbf{a}}^P t^3}{3!} + \alpha_3\Delta\mathbf{R}2 \\ \frac{\ddot{\mathbf{a}}^C t^4}{4!} &= \frac{\ddot{\mathbf{a}}^P t^4}{4!} + \alpha_4\Delta\mathbf{R}2 \\ \frac{\dddot{\mathbf{a}}^C t^5}{5!} &= \frac{\dddot{\mathbf{a}}^P t^5}{5!} + \alpha_5\Delta\mathbf{R}2 \end{aligned} \quad (14)$$

with

$$\Delta\mathbf{R}2 \equiv \frac{\Delta\mathbf{a}t^2}{2!}, \quad (15)$$

and the coefficients given in the following table:

α_i	3th-order	4th-order	5th-order
α_0	$\frac{1}{6}$	$\frac{19}{120}$	$\frac{3}{16}$
α_1	$\frac{5}{6}$	$\frac{3}{4}$	$\frac{251}{360}$
α_2	1	1	1
α_3	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{11}{18}$
α_4	0	$\frac{1}{12}$	$\frac{1}{6}$
α_5	0	0	$\frac{1}{60}$

This table is simply a reproduction of those in Refs. [4, 14, 37], and is provided here for completeness. By using parameters in different columns, one may realize 3rd-, 4th-, and 5th-order (or 4-, 5- and 6-value) [4, 14] Gear-like algorithms for BD simulation. Note that the first two lines in Eq. (14) are slightly different from those in MD simulation, in order to restore the damping effect on deterministic acceleration, and to keep the consistency with the corresponding terms in Eq. (9) [or the first two lines in Eq. (12)].

Adding random displacements: The above three stages are the same as those of Gear's algorithm for MD simulation. However, to complete the BD simulation, we have to use the updating formulae Eq. (6) to add random displacements of the velocity and position. It should be noted that now the corrected values \mathbf{r}^C and \mathbf{v}^C should be used in the places of $\langle \mathbf{r} \rangle$ and $\langle \mathbf{v} \rangle$ in Eq. (6), respectively. Also note that, to implement the simulation, one needs the initial conditions, \mathbf{r}_0 , \mathbf{v}_0 , \mathbf{a}_0 , $\dot{\mathbf{a}}_0$, $\ddot{\mathbf{a}}_0$ and $\ddot{\ddot{\mathbf{a}}}_0$, at $t = 0$. This poses a problem for the very first several steps of simulation, because $\dot{\mathbf{a}}_0$, $\ddot{\mathbf{a}}_0$ and $\ddot{\ddot{\mathbf{a}}}_0$ are undefined. The simplest way to get around this issue is to set all of them to be zero at the very first step, whereas their values will be updated during subsequent iterations. A better way would be to start the simulation by using a Runge-Kutta procedure for the first few steps [14]. However, neither of these solutions of the problem will have any effect on the results in real, many-particle simulations.

The above are the basic procedures for the Gear-like PC method for BD simulation. It should be restated that these formulae and the simulation stages are quite similar to those in the Gear method for MD simulation of Newton's equations [4, 14], except for the expressions for the velocity and position in Eq. (12) and Eq. (14), and for the addition of random displacement at the end of every time step. When $\gamma \rightarrow 0$, the Gear-like method goes over to the Gear method for MD simulation. In the following we shall show that the performance of the proposed Gear-like method also shares many features with its counterpart in MD simulations.

III. TESTING THE ALGORITHM

In this section, we present some simple computational examples as testing cases for our Gear-like algorithm, and for the previously designed Euler-like and Beeman-like (Verlet-like) algorithms. For simplicity, we shall occasionally denote the Euler-like, Beeman-like, Verlet-like and Gear-like methods by EL, BL, VL, and GL, respectively.

We employ the model of a one-dimensional (1D) SDHO [33, 34], in which Eq. (1) may be simplified to

$$\begin{aligned} \frac{d}{dt}v &= -\gamma v - \omega_0^2 r + A(t), \\ \frac{d}{dt}r &= v. \end{aligned} \tag{16}$$

This model has been studied extensively and its solutions are well known [33, 34]. We have listed Chandrasekhar's results in the Appendix A to facilitate subsequent discussion.

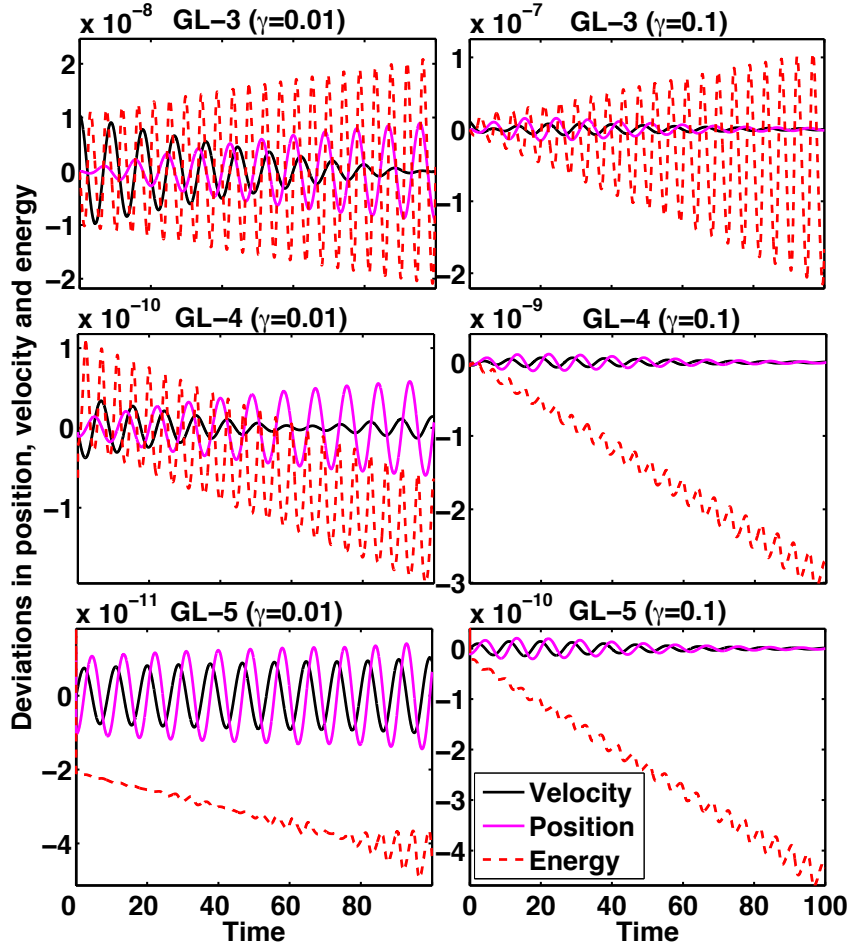


FIG. 1: (Color online) Deviations in the position, velocity and energy from their respective exact solutions within the first 100 time units. The left column is for $\gamma = 0.01$ and the right one for $\gamma = 0.1$. Here, GL-3, GL-4 and GL-5 denote the 3rd-, 4th-, and 5th-order Gear-like methods, respectively. The time step is fixed at 0.01. The (red) dashed lines, (blue) solid lines, and (black) solid lines show the deviations of energy, velocity and position, respectively. Note that the energy deviation is defined relative to the exact values by $E(t)/E_e(t) - 1$.

According to our discussion above, especially following Eq. (6), the task of a BD simulation is simply to predict the position $\mathbf{r}(t)$ and velocity $\mathbf{v}(t)$ of a Brownian particle at time t , given the set of initial conditions at time 0. As previously pointed out, $\mathbf{r}(t)$ and $\mathbf{v}(t)$ are normally distributed random vectors, whose variances and covariances are exact (at the level of Langevin equation), whereas their means are calculated numerically according to certain approximation schemes, such as EL, BL, VL or GL methods. From this point of view, the errors come only from evaluation of the means. In other words, the accuracy in calculating the means will decide the main performance of a BD simulation, so we begin our tests by considering the means.

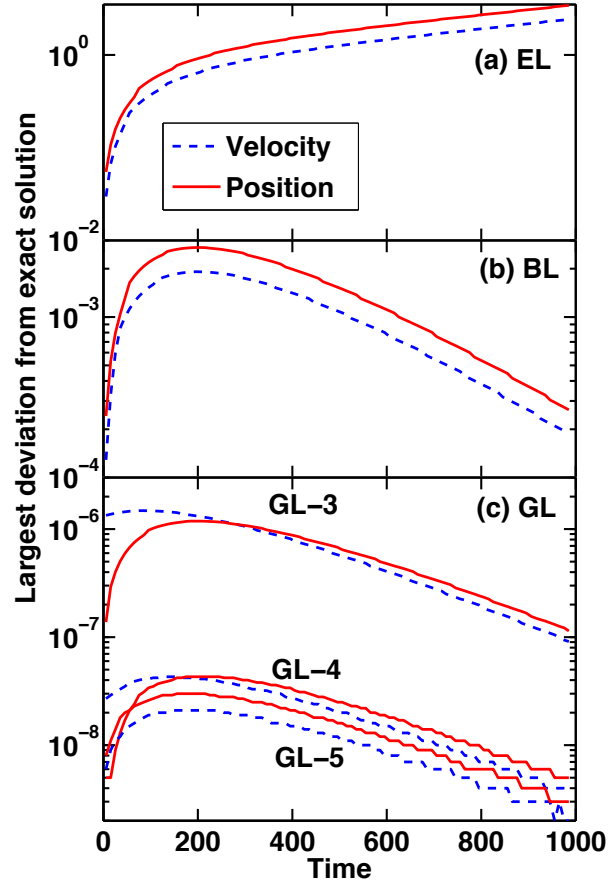


FIG. 2: (Color online) Amplitudes of time-dependent deviations (or the largest deviations) in the position and velocity within 1000 time units for different methods. Here, EL and BL denote the Euler-like and Beeman-like methods, respectively. $\gamma = 0.01$ and the time step of 0.05 are used in these tests.

As usual [4, 14], we firstly carry out simulations over certain periods and record, as a measure of accuracy, the deviations of r and v from their exact solutions without the inclusion of random displacements. We shall also judge the behavior of simulations by monitoring the quantity

$$E(t) = v^2(t) + \omega_0^2 r^2(t), \quad (17)$$

which is proportional to the total energy of the Brownian particle. Note that, because of the existence of damping, this energy is no longer a conserved quantity, but rather decreases exponentially with the factor $\exp(-2\gamma t)$. This may be seen from the exact solutions for the mean position and velocity listed in the Appendix C. In order to judge the energy conservation performance of numerical methods, we normalize $E(t)$ by its exact counterpart $E_e(t)$, which is simply obtained from Eq. (17) by substituting the exact values for the position and velocity. The resultant ratio $E(t)/E_e(t)$ should be a conserved quantity with the expected value of unity.

Without loss of generality, in all ensuing simulations we set $k_B T = 1$, $m = 1$ and $\omega_0 = \sqrt{2}/2$, and choose the initial conditions to be $r_0 = 1$ and $v_0 = 0$. These parameters are also used in the exact results based on Appendix A.

The results of simulations for deviations in the position, velocity and energy from their exact solutions are shown in Fig. 1 for the 3rd-, 4th-, and 5th-order Gear-like methods (denoted as GL-3, GL-4 and GL-5, respectively) within the first 100 time units. Note that the period of the damped oscillator is approximately 8.9 for both $\gamma = 0.01$ and $\gamma = 0.1$, and that the highest amplitude of velocity is approximately $\sqrt{2}/2$. The oscillatory patterns of the deviations in position and velocity resemble approximately those of the exact solutions, but with much smaller amplitudes. The patterns of the energy deviations have a higher frequency (doubled) and seldom oscillate around 0; rather, their oscillating centers drift in an approximately linear manner (in a semi-logarithmic plot). The slope of this line is usually defined as the *energy drift*, which is the main criterion testing the energy conservation in an algorithm [4, 14]. Of course, one would expect the slope to be as small as possible. Another criterion for the energy conservation is the amplitude of oscillations in the energy deviation, which is called the *noise* of the energy drift. Needless to say, the *noise* should be small, too, in accurate simulations. It is interesting to notice that some of the slopes in Fig. 1 are negative, indicating that the energy is damped. This is quite different from the usual situation in the MD simulation where damping is absent by definition, and where the slopes of *energy drift* are always found positive. This, and other aspects of *energy drift* will be further discussed below.

The amplitudes of the time-dependent deviations in position and velocity are shown in Fig. 2 for an extended time period of 1000 time units. One can easily see large differences in the amplitudes among different methods, with the patterns of the Euler-like and Beeman-like (Verlet-like) methods looking similar to those of the Gear-like method, but exhibiting orders of magnitude larger amplitudes than the Gear-like method. This is to be expected as, generally, all Gear-like methods have much better accuracy than Euler-like and Beeman-like (Verlet-like) methods. It should be furthermore pointed out that, except for deviations of the Euler-like method, which grow monotonously, all other deviations pass through broad maxima around the time of 200 time units. This indicates that the Euler-like method is not stable, while the others are. This illustrates yet another aspect in which BD algorithms differ from those of MD simulation, where the deviations invariably grow with time [4]. Namely, it is obvious that the finite damping γ actually stabilizes numerical algorithms (except for the Euler-like method). This is not surprising at all, because previous studies [15, 17, 38] have demonstrated a possibility of stabilizing the MD

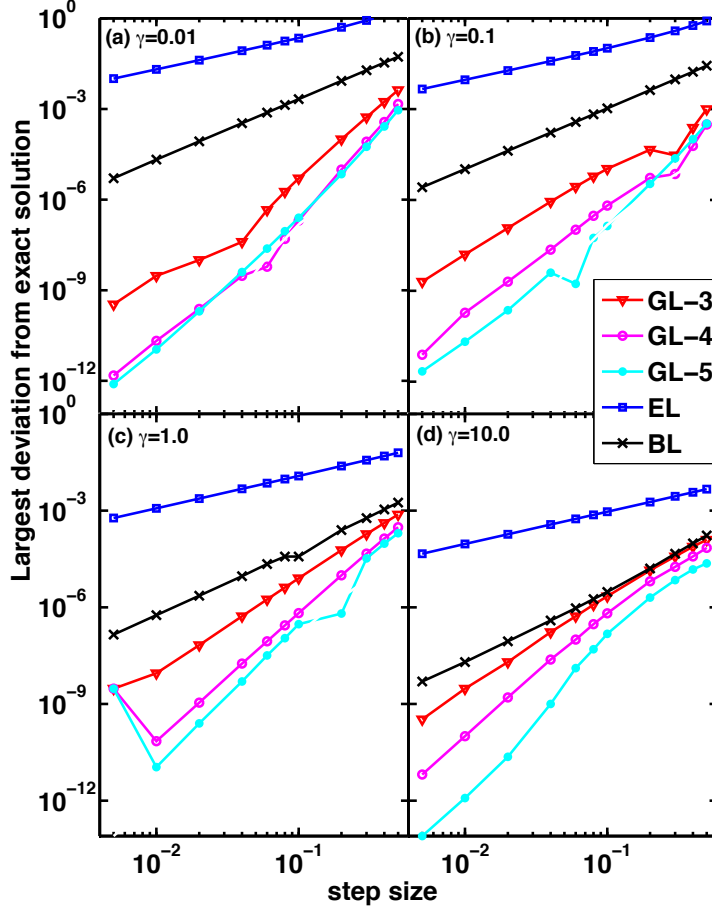


FIG. 3: (Color online) The largest deviation in the position from exact solution vs. the time step size in the first 20 time units, for different methods and different γ s.

simulation by introducing a small, but finite damping. A similar idea is also well-known in the computational fluid dynamics where, for example, a small artificial viscosity is often introduced in fluid simulations to stabilize the computation [39].

Next, we make some comparisons involving different time steps, as that is always the key issue in both the MD and BD simulations. Fig. 3 shows the largest deviations (defined by the largest deviation in the first 20 units) in position versus the size δt of the time step for different methods and different friction coefficients γ . This figure is plotted in a double logarithmic scale, and the curves are nearly straight lines. The slope of these lines is called the *apparent order* [14], illustrating the dependence of error on the time step size. Namely, if the error is found to be proportional to δt^p , then the exponent p is the *apparent order*. It is found that for very small γ , for example $\gamma = 0.01$, as shown in Fig. 3(a), the *apparent orders* are $p_{EL} \approx 1$, $p_{BL} \approx 2$, $p_{GL-3} \approx 3.5$, $p_{GL-4} \approx 4.2$ and $p_{GL-5} \approx 4.6$, respectively for the Euler-like, Beeman-like, 3rd-, 4th-, and 5th-order Gear-like methods. These values are very close to those from the MD simulations where

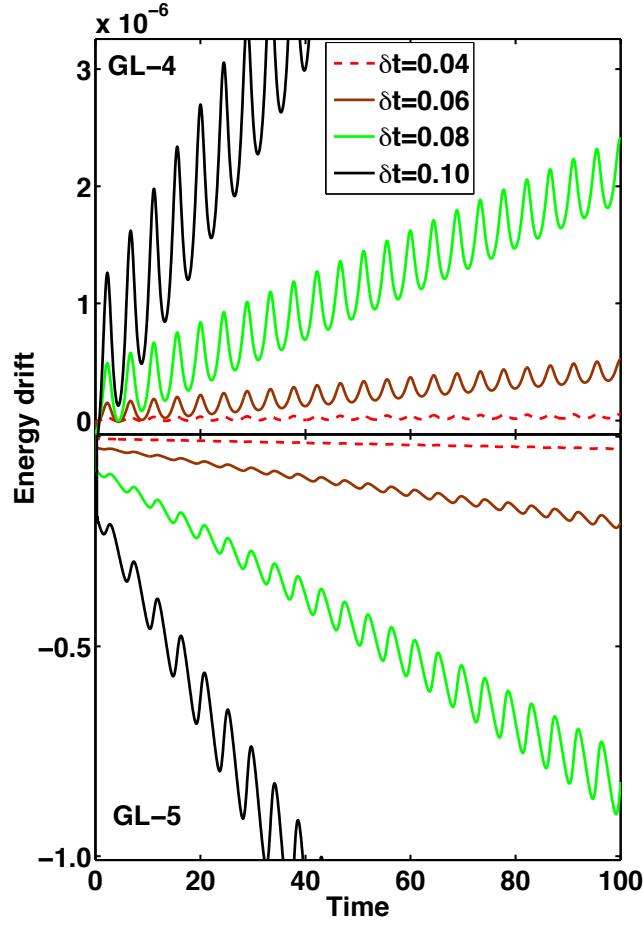


FIG. 4: (Color online) Time-dependent energy drift in GL-4 and GL-5 for different time step sizes δt , with $\gamma = 0.01$ being fixed.

damping is absent [14]. When γ increases, the absolute value of the error for Euler-like and Beeman-like methods decreases, while p_{EL} and p_{BL} remain almost unchanged. On the other hand, p_{GL-3} , p_{GL-4} and p_{GL-5} slightly decrease with increasing γ . For example, for $\gamma = 1$, as shown in Fig. 3(c), Gear-like methods have the worst performance in accuracy: the *apparent orders* become now $p_{GL-3} \approx 3.1$, $p_{GL-4} \approx 3.9$ and $p_{GL-5} \approx 4.0$), but the amplitudes of their errors are still much smaller than those of the Beeman-like and Euler-like methods.

We further provide a more detailed analysis of the energy conservation performances. We have already discussed the *energy drift* and *noise* for Gear-like methods in Fig. 1. In particular, negative energy drifts were noticed there for the 4th- and 5th-order Gear-like methods. However, this is not the case for all time step sizes, as illustrated in Fig. 4, showing the energy drifts in the 4th- and 5th-order methods for different time step-sizes. We see that, when the step-size increases, the energy drift for GL-4 becomes increasingly positive, while that of GL-5 grows increasingly

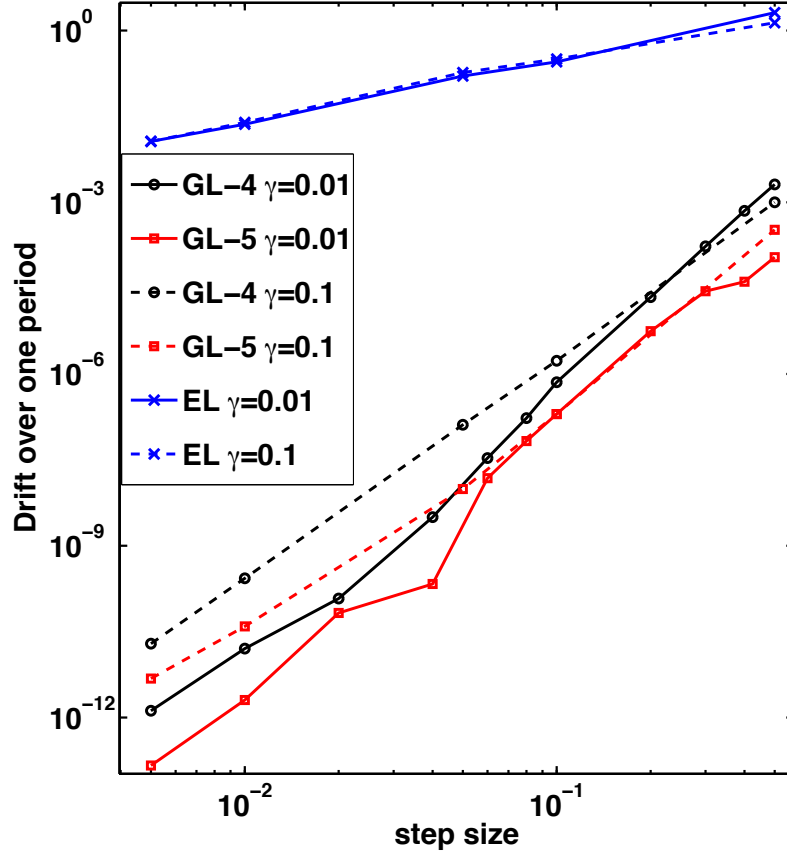


FIG. 5: (Color online) Energy drift in one period of oscillation vs. the time step size for different methods and for $\gamma = 0.01$ and 0.1 .

negative, until a very large time step is used (e.g., $\delta t = 0.6$, not shown in the figure). On the other hand, the noise in the energy drift is always seen to increase with δt .

However, for GL-3, the trends in the energy drift are seen in Fig. 1 to be quite different, especially for smaller γ . Namely, the oscillations in the energy deviation seem rather symmetrical about zero, with the amplitude of the noise being at least two orders of magnitude larger than the drift, as can be seen in the top left panel of Fig. 1. It is found that the Beeman-like method exhibits similar behavior as GL-3, which makes it very hard to do the least-squares fitting and linear regression analysis [14]. On the other hand, for large γ , e.g. $\gamma > 1$, the whole energy of the system is practically damped to zero in a few time units, so that it does not make much sense to discuss the energy drift in such cases. Therefore, in the following we shall omit the energy drift analysis for the BL and GL-3 methods altogether, as well as for other methods in the cases of large γ .

Figure 5 shows the *energy drift* over one period of oscillations versus the time step size [14] for

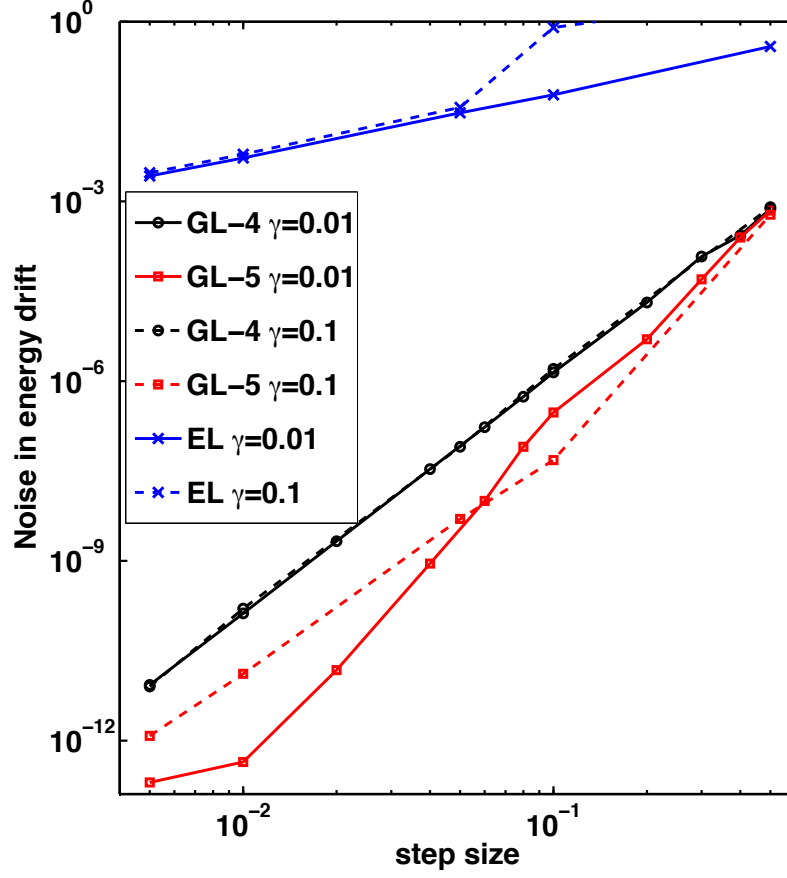


FIG. 6: (Color online) Noise in the energy drift (defined by the average amplitude of oscillations in the first 100 time units) vs. the time step size, for different methods and for $\gamma = 0.01$ and 0.1 .

the EL, GL-4 and GL-5 methods with two different friction coefficients γ . Similarly, Fig. 6 shows the noise in the energy deviation (defined by the averaged amplitudes of oscillations in the first 100 time units) versus the time step size, for these same methods with the same friction coefficients γ . Since those figures are also plotted in a double logarithmic scale, we see that both the *energy drift* and the *noise* increase almost linearly with the time step size, and that the slopes of these curves are very close to the corresponding *apparent orders*, shown in Fig. 3. Moreover, the energy drifts for $\gamma = 0.1$ in the GL-4 and GL-5 methods seem worse off than the corresponding drifts for $\gamma = 0.01$, but the magnitudes of the *energy drift* and *noise* are still very small.

In the above, we have discussed the short-time performances of various BD methods in terms of accuracy and energy drift, without considering random displacements in the position and velocity. In the following, we perform several long-time tests including the random terms, in order to test the statistical properties of the simulation results.

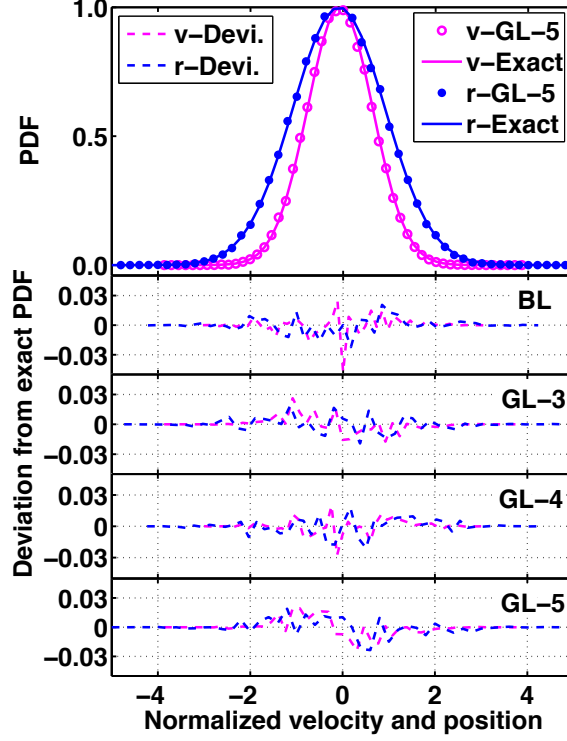


FIG. 7: (Color online) Solid lines show the exact probability distribution functions (PDFs) for velocity and position given by Eq. (A2), while symbols (circles and dots) show the statistics of 80,000 samples taken from 800,000 time units of simulation using the G-5 method. (Note that the PDFs have been scaled up, and their peaks are now unity.) Lower panels show the deviations of PDFs obtained in simulations from the exact ones, for the BL, GL-3, GL-4, and GL-5 methods. $\gamma = 0.01$ and $\delta t = 0.1$ are used in these simulations.

In the long time limit, the oscillator approaches thermal equilibrium with its environment, so that both its velocity and position should become normally distributed. This may be easily seen from their exact solutions, Eq. (A1), or more clearly from their long time limits, Eq. (A2), which are plotted in Fig. 7, along with the statistical ensembles obtained from the simulation data. In this figure, solid lines show the probability distribution functions (PDFs) of the velocity and position given by Eq. (A2), whereas symbols (circles and dots) show ensembles of 80,000 sample points taken from 800,000 time units of simulation. (Note that the first 2000 time units are used to allow the oscillator to reach an equilibrium, so that data from this period are not included in the ensembles shown in Fig. 7). The dashed lines display deviations between the simulation data and the exact PDFs. We used $\gamma = 0.01$ and $\delta t = 0.1$ in simulations, allowing us to compare the results from the BL, GL-3, GL-4 and GL-5 methods, whereas the EL method is ruled out because it is not stable under these conditions. Since there are essentially no visible differences in the distribution

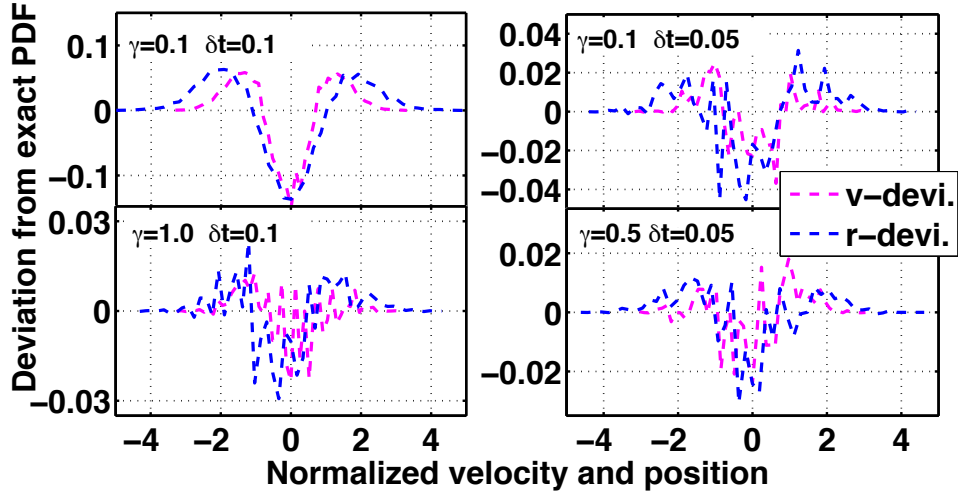


FIG. 8: (Color online) Deviations of PDFs for velocity and position for the EL method, with different time step sizes and damping coefficients. Note that the PDFs have been scaled up, and their peaks are now unity, as is shown in Fig. 7.

functions generated by these four methods, we only show the PDF from the GL-5 method as an example, but display all deviations. We see that the deviations of all GL methods lie approximately within the range of $(-3\%, 3\%)$, which is the statistical error for 80,000 samples according to our experience. [We use the Box-Muller method to take 80,000 samples, with the PDF given by Eq. (A2), and compare the resultant PDF with the exact PDF. This gives a deviation in the range of $(-3\%, 3\%)$.] The deviation from the BL method, as shown in the figure, is a bit larger: within the range of $(-4\%, 4\%)$. With a smaller time step δt and/or larger γ , all the above methods (EL, GL-3, GL-4, and GL-5) exhibit basically the same performance in the long time limit, and all their deviations are well within the range of the statistical error.

Generally, a larger γ would allow a larger time step δt to be used for the same accuracy in the long time limit, and this is true for all methods including the EL method. For example, this method requires typically $\gamma > 1$, along with the step size of about $\delta t = 0.1$, enabling it to produce deviations in the PDF comparable to the other methods, as is shown in Fig. 8.

IV. DISCUSSION

A. Possible simplification

We have provided detailed derivation of a class of Gear-like PC methods for performing BD simulation, and we tested them by using the SDHO model. In particular, we have compared

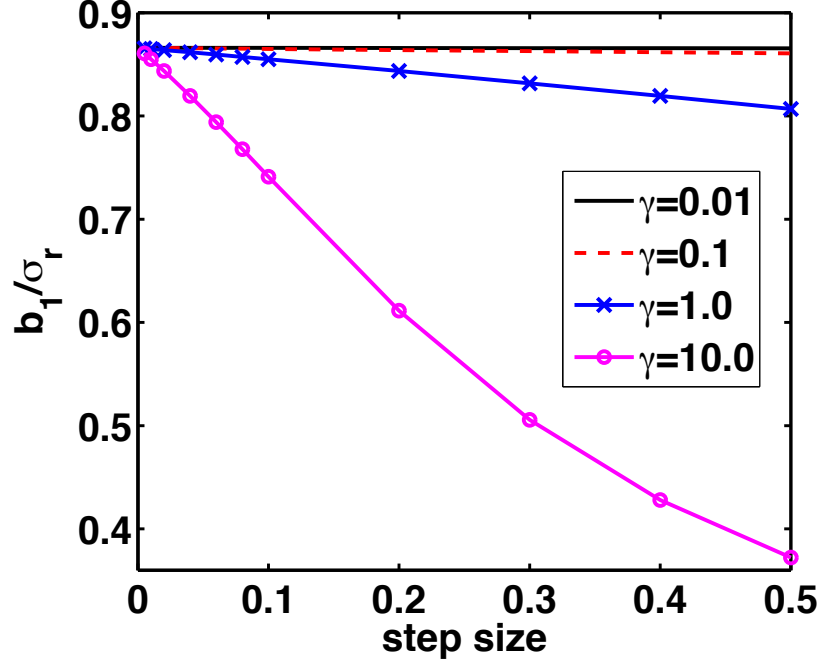


FIG. 9: (Color online) The correlation coefficient between the velocity and position vs. the time step size, for $\gamma = 0.01, 0.1, 1.0$ and 10.0 , respectively.

the performances of different BD methods, namely, the Euler-like, Beeman-like (Verlet-like) and Gear-like methods, and discussed several similarities and differences between the BD and MD simulations.

However, it may be sometimes computationally too demanding to implement a full BD simulation scheme, so we discuss briefly here some possible derivatives of the BD method.

One possible simplification that might be made to the above BD simulation procedure is to neglect the correlation between the velocity and position, i.e., to neglect their covariance. This may be done by simply setting $b_1 = 0$ in Eq. (6), as many researchers actually do in their simulations. Therefore, we have also performed simulations without the covariance, and the results turned out essentially the same as those taking full account of the covariance. Physically, it is not easy to see why such a simplification should work at all. In fact, Fig. 9, showing the correlation coefficient b_1/σ_r versus the time step size for different γ , implies that the correlation between the velocity and position is not negligible at all. Therefore, it seems that monitoring the accuracy, energy conservation, and statistical properties of the ensembles of data from numerical results is not sufficient to fully evaluate performance of an algorithm. Further simulations with many-particle systems are currently being carried out to test if this simplification would affect any other physical

quantities of such systems.

Yet another possible simplification to the above procedure is to apply a finite difference scheme for the deterministic force directly to Langevin equation (1), and numerically solve the resultant formulae like in MD simulation, but with the inclusion of corrections from the Langevin terms at the end of each time step. It can be proved that this simplification is equivalent to using linear approximations to the coefficients c_n in the Taylor-like series of Eq. (9), for example, by letting

$$\begin{aligned}
c_0 &\approx 1 - \gamma t, \\
c_1 &\approx 1 - \frac{1}{2}\gamma t, \\
c_2 &\approx \frac{1}{2} - \frac{1}{6}\gamma t, \\
c_3 &\approx \frac{1}{6} - \frac{1}{24}\gamma t \\
&\dots
\end{aligned} \tag{18}$$

Clearly, when $\gamma t \ll 1$, the above expressions would provide very good approximations to their real values given by Eq. (9). However, even if $\gamma t = 0.01$, the use of this approximation would produce an error of magnitude around 10^{-4} , which is still much larger than the accuracy that the Gear-like methods can achieve.

B. Comparison with High-order Runge-Kutta-like methods

We provide here a brief discussion of the strengths and weaknesses of our algorithm in comparison with some of the most recent developments in the area of high-order algorithms for BD simulation, emphasizing that they typically involve Runge-Kutta methods [25, 27, 28],

One should note that the algorithms based on decomposition of the exponential operator for time propagation provide superior stability of simulation, especially in the limit of low damping, owing to their unique properties of conserving the phase-space volume and being time reversible when simulating Hamiltonian dynamics problems. However, according to Bussi and Parrinello [31], while such algorithms are derived with the aim of producing accurate trajectories up to a given order, they usually break down when a high damping is applied, which is the situation where time reversibility is irrelevant. Moreover, the design of such algorithms is not focused on the correctness of the ensemble generated in simulation [31], which may become a serious shortcoming when simulating, e.g., equilibrium structure of a many-particle system interacting

with a bath. On the other hand, our method has demonstrated excellent performance in the long time limit, where equilibrium distribution functions were reproduced within the statistical error, both for the present single-particle model system, and for the many-particle applications in dusty plasmas [11, 13].

An additional advantage of the methods based on the time propagator with Fokker-Planck operator lies in the fact that various decomposition techniques make analysis quite transparent and allow for systematic strategies in developing higher-order schemes, the highest so far being the fourth-order numerical integrators developed by Drozdov and Brey [27], and by Forbert and Chin [28]. However, it is precisely the decomposition of the exponentiated Fokker-Planck operator that seems to hinder further extensions to orders higher than fourth because of excessive growth in the complexity of such techniques with increasing order. On the other hand, simplicity of our algorithm makes extensions to orders higher than fifth quite straightforward, therefore offering an attractive pay-off for their lack of transparency.

As mentioned in the Introduction, the most serious limitation of algorithms involving high-order Runge-Kutta-like methods, such as [25, 27, 28], may arise when systems with large numbers of Brownian particles are simulated, where the need for force evaluations including all inter-particle interactions becomes the most critical issue in any simulation. For example, the method of HersHKovitz [25], when used in a 1D simulation, requires four evaluations of force, four GRVs, and one evaluation of the force derivative per time step. Moreover, the method of Drozdov and Brey [27], and the K4a method of Forbert and Chin [28] require three evaluations of force, one of the force derivative, and four GRVs for the former method while the latter method requires even eight GRVs. The K4b and K4c methods of Forbert and Chin [28] require eight force evaluations and three GRVs per step. Yet the accuracy of these methods goes only up to the fourth order, whereas our method achieves easily the fifth-order accuracy with only one force evaluation and three GRVs per time step, which presents its greatest advantage for many-particle simulations.

It should be mentioned, however, that the main weakness of our algorithm lies in the fact that it can handle only simple models of particle interactions with a bath, characterized with constant friction (diffusion) coefficients and with fluctuations driven by the Gaussian white noise. On the other hand, the methods based on the time propagator decomposition techniques [27, 28] are capable of solving rather general SDEs, e.g., multivariable Langevin equation with configuration-dependent diffusion coefficients, or Kramers equation with colored noise.

V. CONCLUDING REMARKS

In summary, a Gear-Like Predictor-Corrector algorithm was proposed for BD simulation. It has been tested by using a 1D stochastically-damped-harmonic-oscillator model in terms of its accuracy, energy conservation, and the long-time statistical properties, and the results were compared with those obtained by using the Euler-like and Beeman-like (Verlet-like) methods. It was found that the present method exhibits much better performance in all the above tests. At the same time, when compared to the recent high-order Runge-Kutta-like methods [25, 27, 28], our algorithm promises superior efficiency for simulations in many-particle systems with constant friction (diffusion) coefficient and with Gaussian white noise, as only one force evaluation is needed per time step.

We note that our Gear-like algorithm has already been used successfully for simulations of both equilibrium and nonequilibrium phenomena in strongly-coupled dusty plasmas, involving up to $\sim 10^4$ interacting dust particles [11, 12]. More simulations with many-particle Yukawa systems are being carried out to further test the performance of our method, and the results will be reported elsewhere [13].

Acknowledgments

This work was supported by NSERC and PREA. L.J.H. is now at Christian-Albrechts Universität zu Kiel, and he gratefully acknowledges supports from Alexander von Humboldt Foundation and from Prof. Alexander Piel.

-
- [1] D. L. Ermak, J. Chem. Phys. **62**, 4189 (1975); **62**, 4197 (1975); D. L. Ermak, and J. A. McCammon, J. Chem. Phys. **69**, 1352 (1978); D. L. Ermak, and H. Buckholz, J. Comput. Phys. **35**, 169 (1980).
 - [2] M. P. Allen, Mol. Phys. **40**, 1073 (1980); **47**, 599 (1982).
 - [3] W. F. van Gunsteren, and H. J. C. Berendsen, Mol. Phys. **45**, 637 (1982).
 - [4] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, New York, 1989).
 - [5] H. C. Öttinger, *Stochastic Processes in Polymeric Fluids* (Springer, Berlin 1996).

- [6] Markus Hütter, *Brownian Dynamics Simulation of Stable and Coagulating Colloids in Aqueous Suspensions* (PhD Thesis, No. 13107, ETH Zürich, 1999).
- [7] Bo Liu and Burkhard Dünweg, J. Chem. Phys. **118**, 8061 (2003).
- [8] J. C. Chen and A. S. Kim, Adv. Colloid Interface Sci. **112**, 159 (2004).
- [9] X. H. Zheng and J. C. Earnshaw, Phys. Rev. Lett. **75**, 4214 (1995).
- [10] G. P. Hoffmann, and H. Löwen, J. Phys.: Condens. Matter **12**, 7359 (2000)
- [11] L. J. Hou, Z. L. Mišković, K. Jiang and Y. N. Wang, Phys. Rev. Lett. **96**, 255005 (2006); L. J. Hou, and Z. L. Mišković, Phys. Rev. E **77**, 046401 (2008).
- [12] L. J. Hou and A. Piel, Phys. Plasmas, **15**, 073707 (2008).
- [13] L. J. Hou, and Z. L. Mišković, to be published.
- [14] H. J. C. Berendsen and W. F. van Gunsteren, in *Molecular-Dynamics Simulation of Statistical-Mechanical Systems* (eds. G. Ciccotti and W. G. Hoover, North-Holland Physics Publishing, 1986).
- [15] B. Dünweg, " Langevin methods", in *Computer simulations of surfaces and interfaces* (eds. B. Dünweg, D. P. Landau, and A. I. Milchev, Kluwer Academic Publishers, Dordrecht / Boston / London, 2003).
- [16] B. Leimkuhler, and S. Reich, *Simulating Hamiltonian Dynamics* (Cambridge University Press, 2005).
- [17] B. Dünweg, and A. J. C. Ladd, "Lattice Boltzmann simulations of soft matter systems", arXiv:0803.2826v2 [cond-mat.soft].
- [18] E. Helfand, J. Chem. Phys. **69**, 1010 (1978).
- [19] A. Greiner, W. Strittmatter, and J. Honerkamp, J. Stat. Phys. **51**, 95 (1988).
- [20] R. Mannella and V. Palleschi, Phys. Rev. A **40**, 3381 (1989).
- [21] S. A. Chin, Nucl. Phys. B **9**, 498 (1989).
- [22] A. Iniesta and J. García de la Torre, J. Chem. Phys. **92**, 2015 (1990).
- [23] S. A. Chin, Phys. Rev. A **42**, 6991 (1990).
- [24] R. L. Honeycutt, Phys. Rev. A **45**, 600 (1992).
- [25] E. Hershkovitz, J. Chem. Phys. **108**, 9253 (1998).
- [26] A. C. Brańka, and D. M. Heyes, Phys. Rev. E **58**, 2611 (1998); **60**, 2381 (1999); D. M. Heyes, and A. C. Brańka, Mol. Phys. **98**, 1949 (2000).
- [27] A. N. Drozdov and J. J. Brey, Phys. Rev. E **57**, 1284 (1998).
- [28] H. A. Forbert and S. A. Chin, Phys. Rev. E **63**, 016703 (2000).
- [29] A. Ricci, G. Ciccotti, Mol. Phys. **101**, 1927 (2003).

- [30] S. A. Chin, Phys. Rev. E **73**, 026705 (2006).
- [31] G. Bussi and M. Parrinello, Phys. Rev. E **75**, 056707 (2007).
- [32] F. Thalmann, and J. Farago, J. Chem. Phys. **127**, 124109 (2007).
- [33] S. Chandrasekhar, Rev. Mod. Phys. **15**, 1 (1943).
- [34] D. S. Lemons, *An Introduction to Stochastic Processes in Physics* (The Johns Hopkins University Press, Baltimore, 2002).
- [35] D. S. Lemons and D. L. Kaufman, IEEE Trans. Plasma Sci. **27**, 1288 (1999).
- [36] G. E. P. Box, and M. E. Muller, Ann. Math. Stat. **29**, 610 (1958).
- [37] W. C. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice Hall, Englewood Cliffs, N.J., 1971).
- [38] J. A. Izaguirre, D. P. Catarello, J. M. Wozniak, and R. D. Skeel, J. Chem. Phys. **114**, 2090 (2001).
- [39] J. R. Anderson, *Computational Fluid Dynamics: The basics with applications* (McGraw-Hill, New York, 1995).

APPENDIX A: STOCHASTIC DAMPED HARMONIC OSCILLATOR

The problem of a stochastic damped harmonic oscillator was first worked out in detail by Chandrasekhar in 1943 [33]. We provide here the list of his final results to facilitate the discussion in the main text.

$$\begin{aligned}
\text{mean}\{r(t)\} &= r_0 e^{-\gamma t/2} \left(\cosh \frac{\omega_1 t}{2} + \frac{\gamma}{\omega_1} \sinh \frac{\omega_1 t}{2} \right) + \frac{2v_0}{\omega_1} e^{-\gamma t/2} \sinh \frac{\omega_1 t}{2}, \\
\text{mean}\{v(t)\} &= v_0 e^{-\gamma t/2} \left(\cosh \frac{\omega_1 t}{2} - \frac{\gamma}{\omega_1} \sinh \frac{\omega_1 t}{2} \right) - \frac{2x_0 \omega_0}{\omega_1} e^{-\gamma t/2} \sinh \frac{\omega_1 t}{2}, \\
\text{var}\{r(t)\} &= \frac{k_B T}{m \omega_0^2} \left[1 - e^{-\gamma t} \left(1 + 2 \frac{\gamma^2}{\omega_1^2} \sinh^2 \frac{\omega_1 t}{2} + \frac{\gamma}{\omega_1} \sinh \omega_1 t \right) \right], \\
\text{var}\{v(t)\} &= \frac{k_B T}{m} \left[1 - e^{-\gamma t} \left(1 + 2 \frac{\gamma^2}{\omega_1^2} \sinh^2 \frac{\omega_1 t}{2} - \frac{\gamma}{\omega_1} \sinh \omega_1 t \right) \right], \\
\text{cov}\{r(t)v(t)\} &= \frac{4\gamma k_B T}{m \omega_1^2} e^{-\gamma t} \sinh^2 \frac{\omega_1 t}{2},
\end{aligned} \tag{A1}$$

where $\omega_1 = \sqrt{\gamma^2 - 4\omega_0^2}$. It should be noted that the above equations apply only when $\gamma \geq 2\omega_0$, i.e., in the over-damped case, while in the weakly-damped case, i.e., when $\gamma < 2\omega_0$, one needs to simply redefine ω_1 as $\omega_1 = \sqrt{4\omega_0^2 - \gamma^2}$, and replace \cosh and \sinh with \cos and \sin , respectively.

One can easily see from the above expressions that, when $\gamma t \rightarrow \infty$, the solutions become:

$$\begin{aligned} r(t) &= \sqrt{\frac{k_B T}{m\omega_0^2}} N_r(0, 1) \\ v(t) &= \sqrt{\frac{k_B T}{m}} N_v(0, 1), \end{aligned} \tag{A2}$$