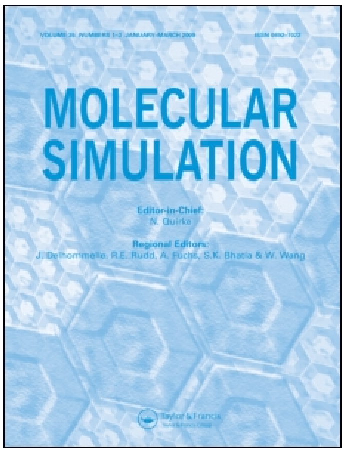

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

A Leap-frog Algorithm for Stochastic Dynamics

W. F. Van Gunsteren[a]; H. J. C. Berendsen[a]
[a] Laboratory of Physical Chemistry, University of Groningen, AG Groningen, The Netherlands

To cite this Article Van Gunsteren, W. F. and Berendsen, H. J. C.(1988) 'A Leap-frog Algorithm for Stochastic Dynamics', Molecular Simulation, 1: 3, 173 — 185

To link to this Article: DOI: 10.1080/08927028808080941

URL: http://dx.doi.org/10.1080/08927028808080941

Molecular Simulation, 1988, Vol. 1, pp. 173–185
Photocopying permitted by license only
Reprints available directly from the publisher

A LEAP-FROG ALGORITHM FOR STOCHASTIC DYNAMICS

W.F. VAN GUNSTEREN and H.J.C. BERENDSEN

Laboratory of Physical Chemistry, University of Groningen, Nijenborgh 16, 9747 AG Groningen, The Netherlands

(*Received April 1987; in final form July 1987*)

A third-order algorithm for stochastic dynamics (SD) simulations is proposed, identical to the powerful molecular dynamics leap-frog algorithm in the limit of infinitely small friction coefficient γ. It belongs to the class of SD algorithms, in which the integration time step Δt is not limited by the condition $\Delta t \ll \gamma^{-1}$, but only by the properties of the systematic force. It is shown how constraints, such as bond length or bond angle constraints, can be incorporated in the computational scheme. It is argued that the third-order Verlet-type SD algorithm proposed earlier may be simplified without loosing its third-order accuracy. The leap-frog SD algorithm is proven to be equivalent to the verlet-type SD algorithm. Both these SD algorithms are slightly more economical on computer storage than the Beeman-type SD algorithm.

KEY WORDS: Stochastic dynamics, Langevin equation, leap-frog algorithm, computer simulation, Brownian dynamics.

1 INTRODUCTION

Over the last two decades the method of computer simulation of many-particle systems, such as liquids, solutions and macromolecules, has developed into a variety of techniques that are applied to a wide range of physical and chemical systems. An impressive overview of the state of the art in the field of molecular dynamics is given in [1]. Classical *molecular dynamics* (MD) solves Newton's equations of motion

$$\dot{x}_i(t) = v_i(t) \tag{1.1}$$

$$m_i \dot{v}_i(t) = F_i(\{x_i(t)\}) \tag{1.2}$$

for a system of N particles. The index i labels particles and cartesian components ($i = 1, 2 \ldots, 3N$). The x-, y- or z-coordinate of a particle is denoted by x_i, its mass by m_i and the cartesian components of the velocity by v_i. The systematic force F_i is to be derived from a conservative force field $V(\{x_i\})$

$$F_i(t) = -\frac{\partial}{\partial x_i} V(\{x_i(t)\}) \tag{1.3}$$

This potential function V describes the interaction between the particles, which generally depends on the coordinates of all particles, denoted by $\{x_i(t)\}$.

When a large system is simulated, it is generally desired to keep the number of degrees of freedom as low as possible. When a certain subset of particles (labelled by index α) can be distinguished, of which the details of their motion is not relevant, these particles can be omitted from the detailed MD simulation. However, the forces they exert on the remaining particles (labelled by index i) must be represented as faithfully as possible. This means that correlations of such forces with positions and velocities of particle i must be incorporated in the equations of motion of particle i, while uncorrelated contributions can be represented by random forces. This brings us to the field of *stochastic dynamics* (SD).

Part of the influence of the α-particles on the i-particles can be included in the force field $V(\{x_i\})$. If the potential or force field $V(\{x_i, x_\alpha\})$ is averaged over an equilibrium ensemble of α-particles, a *potential of mean force*

$$V_m(\{x_i\}) = \langle V(\{x_i, x_\alpha\})\rangle_\alpha \tag{1.4}$$

is obtained. The force field V_m for the i-particles now includes the average interaction with the α-particles. We shall not consider here how such potentials of mean force can be obtained. They modify the force field, but do not change the equations of motion (1.1–1.2).

However, the velocity-correlated and random parts of the forces exerted by the α-particles on the i-particles do modify the equations of motion of the latter. The random forces are related to the velocity-dependent friction forces by the second fluctuation–dissipation theorem [2–5].

In the simplest case of SD, the random force is taken to have white-noise character, and no correlations between the various degrees of freedom are assumed to exist. Under these conditions the velocity-dependent frictional force becomes proportional to the instantaneous velocity of the particle involved. So, the second equation of motion is transformed into the stochastic equation:

$$m_i\dot{v}_i(t) = F_i(\{x_i(t)\}) - m_i\gamma_i v_i(t) + R_i(t) \tag{1.5}$$

The friction coefficient of a particle is denoted by γ_i and the random force by R_i.

The stochastic force $R_i(t)$ is assumed to be a stationary Gaussian random variable with zero mean and to have no correlation with prior velocities or with the systematic force:

$$\langle R_i(0)R_j(t)\rangle = 2m_i\gamma_i kT_{ref}\delta_{ij}\delta(t) \tag{1.6}$$

$$W(R_i) = [2\pi\langle R_i^2\rangle]^{-1/2}\exp\{-R_i^2/(2\langle R_i^2\rangle)\} \tag{1.7}$$

$$\langle R_i\rangle = 0, \tag{1.8}$$

$$\langle v_i(0)R_j(t)\rangle = 0, \quad t \geqq 0, \tag{1.9}$$

$$\langle F_i(0)R_j(t)\rangle = 0, \quad t \geqq 0. \tag{1.10}$$

Here $\langle\ldots\rangle$ denotes averaging over an equilibrium ensemble, k is Boltzmann's constant, T_{ref} is the reference temperature and $W(R_i)$ is the (Gaussian) probability distribution of the stochastic force. A minor correction to Eq. (1.6) has been discussed in [3–5].

The subject of the present paper is to present a new algorithm for integrating the stochastic equations (1.1, 1.5). It is the SD equivalent of the well-known leap-frog MD algorithm for integration of Eqs (1.1, 1.2). Section 2 contains a brief introduction to the most important MD and SD algorithms. In section 3 the leap-frog SD algorithm is derived. In section 4 it is shown how the computational scheme has to be changed when holonomic (bond length) constraints are introduced into the system of particles. Section 5 contains a short discussion of various aspects of the algorithm.

2 ALGORITHMS FOR MOLECULAR DYNAMICS AND STOCHASTIC DYNAMICS

Properties of various MD algorithms have extensively been discussed in [6, 7]. Here, we briefly review some basic points relevant to the SD algorithm to be derived.

Equation (1.5) differs from Eq. (1.2) only by the presence of the velocity-dependent frictional term and of the random force term. This means that an MD algorithm designed for integration of (1.1, 1.2) can be generalized to an SD algorithm for integration of Eqs (1.1, 1.5), if it belongs to the class of open integration methods [7]. The three open MD algorithms that are most widely used in molecular dynamics studies are the Verlet [8], leap-frog [9] and Beeman [10] algorithms. The three algorithms produce exactly the same trajectory and are thus equivalent. They possess the useful property of being invariant upon time reversal. This implies that the total energy, which is a constant of motion, shows essentially no drift. Although they are of third order in the integration time step Δt and so less accurate than higher-order Gear algorithms, they are much easier to use, require less storage and are more stable and efficient for large values of Δt [6, 7]. In practice, it only makes sense to use higher-order algorithms when accurate dynamics is required based on accurately known forces (no random parts) and using a relatively small time step Δt, or when the forces are highly harmonic, which makes them predictable by higher-order derivatives. Algorithms that are of lower than third order are not efficient in MD of molecular systems [7, 11]. This is because molecular potential functions V have generally positive second derivatives, requiring a third- or higher-order algorithm.

The *Verlet* algorithm [8] consists of the equation:

$$x(t_n + \Delta t) = 2x(t_n) - x(t_n - \Delta t) + m^{-1}F(t_n)(\Delta t)^2 + O[(\Delta t)^4]. \qquad (2.1)$$

Here the index n denotes the n^{th} integration step and the indices denoting particles and cartesian components have been omitted. Formula (2.1) can be derived by expanding both $x(t_n + \Delta t)$ and $x(t_n - \Delta t)$ in a Taylor series in Δt and subsequently adding up the two equations. We not that only even powers of Δt occur in Eq. (2.1); the algorithm contains no explicit velocities and the third-order terms cancel. If velocities are required, they are usually obtained by the formula

$$v(t_n) = \{x(t_n + \Delta t) - x(t_n - \Delta t)\}/(2\Delta t) + O[(\Delta t)^2] \qquad (2.2)$$

The *leap-frog* algorithm [9] consists of two equations

$$v(t_n + \Delta t/2) = v(t_n - \Delta t/2) + m^{-1}F(t_n)\Delta t + O[(\Delta t)^3] \qquad (2.3)$$

and

$$x(t_n + \Delta t) = x(t_n) + v(t_n + \Delta t/2)\Delta t + O[(\Delta t)^3] \qquad (2.4)$$

Its name is based on the fact that it uses coordinates and velocities at alternating time points t_n, $t_n + \Delta t/2$, etc. Formula (2.3) can be derived by expanding both $v(t_n + \Delta t/2)$ and $v(t_n - \Delta t/2)$ in a Taylor series in $\Delta t/2$ and subsequently subtracting the two equations. Formula (2.4) is obtained similarly by expanding both $x(t_{n+1/2} + \Delta t/2)$ and $x(t_{n+1/2} - \Delta t/2)$ in a Taylor series and subtracting the two equations. We note that the third-order terms in (2.4) do not cancel. However, it is easily proved that Eqs (2.3, 2.4) are exactly equivalent to Eq. (2.1). This means that the leap-frog algorithm is a third-order algorithm, just as Verlet's is. The leap-frog algorithm has the advantage that the velocities explicitly appear in the integration scheme, which allows for a coupling of the system to a thermal bath.

The *Beeman* algorithm [10] looks very different.

$$x(t_n + \Delta t) = x(t_n) + v(t_n)\,\Delta t + 2/3m^{-1}F(t_n)\,(\Delta t)^2 - 1/6m^{-1}F(t_n - \Delta t)\,(\Delta t)^2 + \mathrm{O}[(\Delta t)^4] \qquad (2.5)$$

and

$$v(t_n + \Delta t) = v(t_n) + 1/3m^{-1}F(t_n + \Delta t)\Delta t + 5/6m^{-1}F(t_n)\Delta t - 1/6m^{-1}F(t_n - \Delta t)\Delta t + \mathrm{O}[(\Delta t)^3] \qquad (2.6)$$

Yet it is exactly equivalent to the Verlet and leap-frog schemes [7, 12, 13]. Since the Beeman algorithm is more complicated and requires more storage than the other two equivalent algorithms, there sem to be no good reasons for its use.

When integrating the stochastic equations (1.1, 1.5), the same type of algorithms can be used as when solving (1.1, 1.2) in ordinary molecular dynamics. The relation between the time step Δt and the accuracy obtained by the algorithm is determined entirely by the form of the systematic force F in Eqs (1.2) and (1.5) and not by the friction term or the random force. The two latter terms can directly be integrated, since they do not depend on the coordinates 814].

The earliest SD algorithms were all of second order, and so of low efficiency. Therefore we will not discuss them here. In recent years two third-order SD algorithms have been proposed independently [11, 15]. Allen [15] has given the SD equivalent of the Beeman algorithm. The SD equivalent of the Verlet algorithm has been derived in [11]. In the following section we will derive the SD equivalent of the leap-frog algorithm.

3 A LEAP-FROG ALGORITHM FOR STOCHASTIC DYNAMICS

In this section we derive a SD algorithm that is identical to the MD leap-frog algorithm in the limit of zero friction ($\gamma = 0$).

Solving the linear, inhomogeneous first-order differential equation (1.5) one finds

$$v(t) = v(t_n)\exp[-\gamma(t - t_n)] + m^{-1}\exp[-\gamma(t - t_n)]\int_{t_n}^{t}\exp[-\gamma(t_n - t')] \times \{F(t') + R(t')\}\,\mathrm{d}t', \qquad (3.1)$$

where indices denoting particles and cartesian components have been omitted. Since the stochastic properties of $R(t')$ are known (Eqs (1.6–1.10)), the integral over $R(t')$ can be obtained directly. The integral over $F(t')$ is obtained by expanding $F(t')$ in a power series

$$F(t) = F(t_n) + F'(t_n)\,(t - t_n) + \mathrm{O}[(t - t_n)^2 \qquad (3.2)$$

The derivative of the systematic force with respect to time at $t = t_n$ is denoted by $F(t_n)$. Since the leap-frog scheme is a third-order algorithm, an expansion of $F(t)$ up to second order suffices. Expansion (3.2) is valid only in the limit of small friction coefficients γ. This can be understood by noting that the systematic force $F(\{x(t)\})$ depends on the coordinates that must satisfy the Langevin equation (1.5); this equation includes apart from the systematic force two extra terms, the frictional force and the random force, which will influence the dependence of $x(t)$ on time. In the high friction limit the latter terms will dominate in Eq. (1.5), in which case inclusion of the derivative of the systematic force would make no sense. Therefore, we will omit from the formulae below all terms beyond third order in Δt in the coordinates and beyond second order in the velocities.

Performing the integration in Eq. (3.1) we find

$$\begin{aligned} v(t) &= v(t_n) \exp[-\gamma(t-t_n)] + (m\gamma)^{-1} F(t_n)\,[1 - \exp[-\gamma(t-t_n)]] \\ &\quad + (m\gamma^2)^{-1} F'(t_n)\,[\gamma(t-t_n) - [1 - \exp[-\gamma(t-t_n)]]] \\ &\quad + m^{-1}\exp[-\gamma(t-t_n)] \int_{t_n}^{t} \exp[-\gamma(t_n - t')] R(t')\mathrm{d}t' \\ &\quad + \mathrm{O}[(t-t_n)^3] \end{aligned} \tag{3.3}$$

Integrating the velocity over half a time step from t_n to $t_{n+1/2} = t_n + \Delta t/2$ we find

$$\begin{aligned} v(t_n + \Delta t/2) &= v(t_n)\exp(-\gamma\Delta t/2) + (m\gamma)^{-1}F(t_n)\,[1 - \exp(-\gamma\Delta t/2)] \\ &\quad + (m\gamma^2)^{-1}F'(t_n)\,[\gamma\Delta t/2 - [1 - \exp(-\gamma\Delta t/2)]] \\ &\quad + m^{-1}\exp[-\gamma\Delta t/2] \int_{t_n}^{t_n+\Delta t/2} \exp[-\gamma(t_n - t')] \\ &\quad \times R(t')\mathrm{d}t' + \mathrm{O}[(\Delta t)^3] \end{aligned} \tag{3.4}$$

For the fourth term, which is a random variable, we use the notation:

$$V_n(\Delta t/2) = m^{-1}\exp[-\gamma\Delta t/2] \int_{t_n}^{t_n+\Delta t/2} \exp[-\gamma(t_n - t')]R(t')\mathrm{d}t' \tag{3.5}$$

If we write Eq. (3.4) with $-\Delta t$ instead of Δt, multiply the obtained equation by exp $[-\gamma\Delta t]$ and substract it from Eq. (3.4), the term involving $v(t_n)$ is eliminated and we find

$$\begin{aligned} v(t_{n+1/2}) &= v(t_{n-1/2})\exp(-\gamma\Delta t) + m^{-1}F(t_n)\Delta t(\gamma\Delta t)^{-1}\,[1 - \exp(-\gamma\Delta t)] \\ &\quad + V_n(\Delta t/2) - \exp(-\gamma\Delta t)\, V_n(-\Delta t/2) + \mathrm{O}[(\Delta t)^3] \end{aligned} \tag{3.6}$$

where the term involving $F'(t_n)$ has been omitted, since it is of third order in Δt. When $\gamma \rightarrow 0$, this formula reduces to Eq. (2.3), which is the velocity formula of the leap-frog algorithm.

The positions are obtained by integrating Eq. (1.1) from time

$$t_{n+1/2} = t_n + \Delta t/2$$

$$x(t) = x(t_{n+1/2}) + \int_{t_{n+1/2}}^{t} v(t')\,\mathrm{d}t' \tag{3.7}$$

If we use in Eq. (3.7) formula (3.3) for $v(t')$ with t_n replaced by $t_{n+1/2}$, perform the integrations and reduce the double integral over $R(t')$ using partial integration, we find for $t = t_{n+1/2} + \Delta t/2$:

$$\begin{aligned} x(t_{n+1/2} + \Delta t/2) \;=\; & x(t_{n+1/2}) + v(t_{n+1/2})\gamma^{-1}\,[1 - \exp(-\,\gamma\Delta t/2)] + (m\gamma^2)^{-1}F(t_{n+1/2}) \\ & \times\,[\gamma\Delta t/2 - [1 - \exp[-\,\gamma\Delta t/2)] + (m\gamma^3)^{-1}F'(t_{n+1/2}) \\ & \times\,[1/2(\gamma\Delta t/2)^2 - \{\gamma\Delta t/2 - [1 - \exp(-\,\gamma\Delta t/2)]\}] + (m\gamma)^{-1} \\ & \times \int_{t_{n+1/2}}^{t_{n+1/2}+\Delta t/2} [1 - \exp[-\,\gamma(t_{n+1/2} + \Delta t/2 - t')]]R(t')\,\mathrm{d}t' \\ & +\, \mathrm{O}[(\Delta t)^4] \end{aligned} \qquad (3.8)$$

For the fifth term in Eq. (3.8), which is a random variable, we use the notation:

$$X_{n+1/2}(\Delta t/2) \;=\; (m\gamma)^{-1}\int_{t_{n+1/2}}^{t_{n+1/2}+\Delta t/2} [1 - \exp[-\,\gamma(t_{n+1/2} + \Delta t/2 - t')]]R(t')\,\mathrm{d}t' \qquad (3.9)$$

If we write Eq. (3.8) with $-\Delta t$ instead of Δt and subtract it from Eq. (3.8), the term involving $x(t_{n+1/2})$ is eliminated and we find

$$\begin{aligned} x(t_{n+1}) \;=\; & x(t_n) + v(t_{n+1/2})\,\Delta t\,(\gamma\Delta t)^{-1}\,[\exp(+\,\gamma\Delta t/2) - \exp(-\,\gamma\Delta t/2)] \\ & +\, X_{n+1/2}(\Delta t/2) - X_{n+1/2}(-\,\Delta t/2) + \mathrm{O}[(\Delta t)^3] \end{aligned} \qquad (3.10)$$

where the terms involving $F(t_{n+1/2})$ and $F'(t_{n+1/2})$ have been omitted since they are of third order in Δt. When $\gamma \to 0$, this formula reduces to Eq. (2.4), which is the coordinate formula of the leap-frog algorithm. Although Eq. (3.10) seems to be of order $(\Delta t)^3$, the SD leap-frog algorithm derived above is actually of order $(\Delta t)^4$ in the positions. This can be understood using the same line of argument as in section 2. In section 5 the equivalence of (3.6, 3.10) with a third-order stochastic Verlet algorithm will be shown.

When the SD leap-frog algorithm Eqs (3.6, 3.10) are used, it must be noted that $V_n(-\Delta t/2)$ is correlated with $X_{n-1/2}(\Delta t/2)$, since they are different integrals of $R(t)$ over the time interval $(t_{n-1/2}, t_n)$

$$V_n(-\,\Delta t/2) \;=\; m^{-1}\int_{t_{n-1/2}}^{t_{n-1/2}+\Delta t/2} (-)\exp[-\,\gamma(t_{n-1/2} - t')]R(t')\,\mathrm{d}t' \qquad (3.11)$$

and

$$X_{n-1/2}(\Delta t/2) \;=\; (m\gamma)^{-1}\int_{t_{n-1/2}}^{t_{n-1/2}+\Delta t/2} [1 - \exp[-\,\gamma(t_{n-1/2} + \Delta t/2 - t')]]\;R(t')\,\mathrm{d}t' \qquad (3.12)$$

The same observation holds for the other pair of random variables $X_{n+1/2}(-\Delta t/2)$ and $V_n(\Delta t/2)$ occurring in Eqs (3.6, 3.10). These are different integrals of $R(t)$ over the time interval $(t_n, t_{n+1/2})$:

$$X_{n+1/2}(-\,\Delta t/2) \;=\; (m\gamma)^{-1}\int_{t_n}^{t_n+\Delta t/2} (-)\,[1 - \exp[-\,\gamma(t_n - t')]]R(t')\,\mathrm{d}t' \qquad (3.14)$$

and

$$V_n(\Delta t/2) \;=\; m^{-1}\int_{t_n}^{t_n+\Delta t/2} \exp[-\,\gamma(t_n + \Delta t/2 - t')]R(t')\,\mathrm{d}t' \qquad (3.15)$$

In both cases the pair of random variables obeys a bivariate Gaussian distribution [16]. The parameters of these two bivariate distributions and the way of sampling from them will be subsequently discussed.

The bivariate Gaussian distribution for the first pair of random variables (3.11, 3.12) reads

$$\begin{aligned} W(X_{n-1/2}(\Delta t/2),\ V_n\,(-\ \Delta t/2)) \;=\; & [4\pi^2\sigma_1^2\sigma_2^2\,[1\;-\;s^2]]^{-1/2}\exp\{-\;[\sigma_2^2X_{n-1/2}^2(\Delta t/2) \\ & -\;2\sigma_1\sigma_2 sX_{n-1/2}(\Delta t/2)V_n(-\;\Delta t/2) \\ & +\;\sigma_1^2V_n^2\,(-\;\Delta t/2)]/[2\sigma_1^2\sigma_2^2\,[1\;-\;s^2]]\} \qquad (3.16) \end{aligned}$$

The parameters σ_1, σ_2 and s of this distribution can be determined by evaluating the quantities $\langle X_{n-1/2}^2(\Delta t/2)\rangle$, $\langle V_n^2(-\ \Delta t/2)\rangle$ and $\langle X_{n-1/2}\,(\Delta t/2)V_n(-\ \Delta t/2)\rangle$ using (3.11, 3.12) and the property (1.6)

$$\langle R(t)R(t')\rangle \;=\; 2m\gamma kT_{ref}\,\delta(t\;-\;t') \qquad (3.17)$$

We find

$$\sigma_1^2 \;=\; \langle X_{n-1/2}^2(\Delta t/2)\rangle \;=\; \frac{kT_{ref}}{m\gamma^2}\,C(\gamma\Delta t/2), \qquad (3.18)$$

$$\sigma_2^2 \;=\; \langle V_n^2(-\ \Delta t/2)\rangle \;=\; \frac{kT_{ref}}{m}\,[\exp\,(+\gamma\Delta t)\;-\;1], \qquad (3.19)$$

$$s\sigma_1\sigma_2 \;=\; \langle X_{n-1/2}(\Delta t/2)\,V_n\,(-\ \Delta t/2)\rangle \;=\; \frac{kT_{ref}}{m\gamma}\,D(\gamma\Delta t/2), \qquad (3.20)$$

$$\sigma_2^2[1\;-\;s^2] \;=\; \sigma_2^2\;-\;(s\sigma_1\sigma_2)^2/\sigma_1^2 \;=\; \frac{kT_{ref}}{m}\,B(\gamma\Delta t/2)/C(\gamma\Delta t/2) \qquad (3.21)$$

and

$$s\sigma_2/\sigma_1 \;=\; (s\sigma_1\sigma_2)/\sigma_1^2 \;=\; \gamma D(\gamma\Delta t/2)/C(\gamma\Delta t/2), \qquad (3.22)$$

where [11]

$$C(\gamma\Delta t/2) \;=\; \gamma\Delta t\;-\;3\;+\;4\exp\,(-\;\gamma\Delta t/2)\;-\;\exp\,(-\;\gamma\Delta t), \qquad (3.23)$$

$$D(\gamma\Delta t/2) \;=\; 2\;-\;\exp\,(+\gamma\Delta t/2)\;-\;\exp\,(-\;\gamma\Delta t/2) \qquad (3.24)$$

and

$$B(\gamma\Delta t/2) \;=\; \gamma\Delta t[\exp\,(+\gamma\Delta t)\;-\;1]\;-\;4\,[\exp\,(+\gamma\Delta t/2)\;-\;1]^2. \qquad (3.25)$$

Sampling in a bivariate distribution such as (3.16) is generally performed by first sampling one variable from its distribution, and subsequently sampling the other variable from its conditional distribution, given the value obtained for the first variable [16]. The distribution for $X_{n-1/2}(\Delta t/2)$, irrespective the value of $V_n(-\Delta t/2)$, is:

$$W(X_{n-1/2}(\Delta t/2)) \;=\; [2\pi\sigma_1^2]^{-1/2}\exp\,\{-\;X_{n-1/2}^2(\Delta t/2)/(2\sigma_1^2)\} \qquad (3.26)$$

and the conditional distribution for $V_n(-\Delta t/2)$, given a specific value of $X_{n-1/2}(\Delta t/2)$, reads [16]

$$W(V_n(-\Delta t/2)|X_{n-1/2}(\Delta t/2)) = [2\pi\sigma_2^2[1-s^2]]^{-1/2}\exp\{-[V_n(-\Delta t/2) - s\sigma_2/\sigma_1 X_{n-1/2}(\Delta t/2)]^2/[2\sigma_2^2[1-s^2]]\} \quad (3.27)$$

Correspondingly, the bivariate Gaussian distribution for the second pair of random variables (3.13, 3.14) reads

$$W(V_n(\Delta t/2), X_{n+1/2}(-\Delta t/2)) = [4\pi^2\rho_1^2\rho_2^2[1-r^2]]^{-1/2}\exp\{-[\rho_2^2 V_n^2(\Delta t/2) - 2\rho_1\rho_2 r V_n(\Delta t/2)\, X_{n+1/2}(-\Delta t/2) + \rho_1^2 X_{n+1/2}(-\Delta t/2)]/[2\rho_1^2\rho_2^2[1-r^2]]\} \quad (3.28)$$

The parametersρ_1, ρ_2 and r of this distribution can be determined by evaluating the quantities $\langle V_n^2(\Delta t/2)\rangle$, $\langle X_{n+1/2}^2(-\Delta t/2)\rangle$ and $\langle V_n(\Delta t/2)\, X_{n+1/2}(-\Delta t/2)\rangle$ using (3.13, 3.14) and the property (3.17). We find

$$\rho_1^2 = \langle V_n^2(\Delta t/2)\rangle = \frac{kT_{ref}}{m}(-)\,[\exp(-\gamma\Delta t) - 1], \quad (3.29)$$

$$\rho_2^2 = \langle X_{n+1/2}^2(-\Delta t/2)\rangle = \frac{kT_{ref}}{m\gamma^2}(-)\,C(-\gamma\Delta t/2), \quad (3.30)$$

$$r\rho_1\rho_2 = \langle V_n(\Delta t/2)\, X_{n+1/2}(-\Delta t/2)\rangle = \frac{kT_{ref}}{m\gamma}(-)\,D(-\gamma\Delta t/2), \quad (3.31)$$

$$\rho_2^2[1-r^2] = \rho_2^2 - (r\rho_1\rho_2)^2/\rho_1^2 = \frac{kT_{ref}}{m\gamma^2}(-)\,B(-\gamma\Delta t/2)/[\exp(-\gamma\Delta t) - 1] \quad (3.32)$$

and

$$r\rho_2/\rho_1 = (r\rho_1\rho_2)/\rho_1^2 = \gamma^{-1}D(-\gamma\Delta t/2)/[\exp(-\gamma\Delta t) - 1]. \quad (3.33)$$

The distribution for $V_n(\Delta t/2)$, irrespective the value of $X_{n+1/2}(-\Delta t/2)$, is

$$W(V_n(\Delta t/2)) = [2\pi\rho_1^2]^{-1/2}\exp\{-V_n^2(\Delta t/2)/(2\rho_1^2)\} \quad (3.34)$$

and the conditional distribution for $X_{n+1/2}(-\Delta t/2)$, given a specific value of $V_n(\Delta t/2)$, reads

$$W(X_{n+1/2}(-\Delta t/2)|V_n(\Delta t/2)) = [2\pi\rho_2^2[1-r^2]]^{-1/2}\exp\{-[X_{n+1/2}(-\Delta t/2) - r\rho_2/\rho_1 V_n(\Delta t/2)]^2/[2\rho_2^2[1-r^2]]\}. \quad (3.35)$$

Now we may summarize the *computational scheme* of the *leap-frog SD algorithm*.

1. Assume that $x(t_n)$, $v(t_{n-1/2})$ and $X_{n-1/2}(\Delta t/2)$ are known. When only $x(t_n)$ is known, $v(t_{n-1/2})$ may be sampled from a Maxwellian distribution at temperature T_{ref} and $X_{n-1/2}(\Delta t/2)$ is sampled from a Gaussian distribution with zero mean and width (3.18).
2. Evaluate $F(t_n)$ from the potential $V(\{x(t_n)\})$, using Eq. (1.3).
3. Sample Y_v from a Gaussian distribution with zero mean and width (3.21) and calculate

$$V_n(-\Delta t/2) = X_{n-1/2}(\Delta t/2)\,\gamma D(\gamma\Delta t/2)/C(\gamma\Delta t/2) + Y_v. \quad (3.36)$$

Sample $V_n(\Delta t/2)$ from a Gaussian distribution with zero mean and width (3.29) and calculate the velocities $v(t_{n+1/2})$ using Eq. (3.6).

4. Sample Y_x from a Gaussian distribution with zero mean and width (3.32) and calculate

$$X_{n+1/2}(-\Delta t/2) = V_n(\Delta t/2)\gamma^{-1} D(-\gamma\Delta t/2)/[\exp(-\gamma\Delta t) - 1] + Y_x. \quad (3.37)$$

Sample $X_{n+1/2}(\Delta t/2)$ from a Gaussian distribution with zero mean and width (3.18) and calculate the positions $x(t_{n+1})$ using Eq. (3.10).

When computing the coefficients in (3.6), (3.10), (3.18), (3.21), (3.29), (3.32), (3.36) and (3.37) for small values of $\gamma\Delta t$ the numerical accuracy should be watched. For example, the leading term in the expansion of $B(\gamma\Delta t/2)$ in a power series in $\gamma\Delta t/2$ is of fourth order. So for small $\gamma\Delta t$ formula (3.25) should not be used for evaluating $B(\gamma\Delta t/2)$. Instead, the coefficients should be obtained from a power series expansion in $\gamma\Delta t/2$. These expansions are given in the Appendix.

4 A LEAP-FROG ALGORITHM FOR STOCHASTIC DYNAMICS WITH CONSTAINTS

In [17] we discussed the various ways in which holonomic, scleronomous constraints can be conserved when integrating the equations of motion of a set of particles. The most simple method turned out to be the procedure called SHAKE [18]. In this section we show how SHAKE can be incorporated in the leap-frog SD algorithm given in the previous section.

The procedure SHAKE and its application have been reviewed in [19]. Its essential feature is that after each integration time step Δt the constraints are satisfied by adding displacement vectors to the position vectors of the particles that result from a non-constraint time step. The displacement vectors are determined such that the constraints are satisfied at the final positions. We will denote the application of SHAKE by

$$\text{SHAKE}\ (x(t_n), x'(t_{n+1}), x(t_{n+1})). \quad (4.1)$$

This means that the positions $x'(t_{n+1})$ that result from the non-constraint time step will be reset to give the constrained positions $x(t_{n+1})$. The direction of the displacement vectors $(x(t_{n+1}) - x'(t_{n+1}))$ is determined by the reference positions $x(t_n)$; that is, for each individual distance constraint, the displacement of the pair of particles involved is parallel to the vector from one particle to the other in the reference configuration [19].

When the procedure SHAKE is used to impose constraints on the system, the *computational scheme* of the *leap-frog SD algorithm including constraints* looks as follows:

1. Assume that $x(t_n)$, $v(t_{n-1/2})$ and $X_{n-1/2}(\Delta t/2)$ are known, that the positions $x(t_n)$ satisfy the constraints and that the velocities $v(t_{n-1/2})$ do not contain components along the the constraint directions, that is, they also satisfy the constraints.

2. Evaluate $F(t_n)$ from the potential $V(\{x(t_n)\})$, in which the interaction terms acting merely along the constrained degrees of freedom are omitted. For example, when constraining bond lengths, the term in the potential V representing the covalent bond interaction is omitted, when computing $F(t_n)$ using Eq. (1.3).
3. Compute the non-constrained velocities $v'(t_{n+1/2})$ from (3.6), as specified in step (3) of the computational scheme for the non-constraint algorithm given in the previous section. Make these velocities satisfy the constraints by the following steps:
   (a) Compute
   $$x'(t_{n+1}) = x(t_n) + v'(t_{n+1/2})\,\Delta t(\gamma\Delta t)^{-1}\,[\exp(\gamma\Delta t/2) - \exp(-\gamma\Delta t/2)] \tag{4.2}$$
   (b) Perform
   $$\text{SHAKE}\,(x(t_n),\, x'(t_{n+1}),\, x''(t_{n+1})) \tag{4.3}$$
   (c) Obtain the shaken velocities from
   $$v(t_{n+1/2}) = [x''(t_{n+1}) - x(t_n)]/[\Delta t(\gamma\Delta t)^{-1}\,[\exp(\gamma\Delta t/2) - \exp(-\gamma\Delta t/2)]] \tag{4.4}$$
4. Compute the quantities $X_{n+1/2}(-\Delta t/2)$ and $X_{n+1/2}(\Delta t/2)$ as is specified in step (4) of the computational scheme for the non-constraint algorithm given in the previous section. Compute the non-constrained positions $x'''(t_{n+1})$ from
   $$x'''(t_{n+1}) = x''(t_{n+1}) + X_{n+1/2}(\Delta t/2) - X_{n+1/2}(-\Delta t/2) \tag{4.5}$$
   and make them satisfy the constraints by performing
   $$\text{SHAKE}\,(x(t_n),\, x'''(t_{n+1}),\, x(t_{n+1})). \tag{4.6}$$

When the initial positions $x'(t_0)$ and velocities $v'(t_{-1/2})$ do not satisfy the constraints, they may be shaken in the following way. The positions $x'(t_0)$ can be made satisfy the constraints by taking the reference positions x'', equal to $x'(t_0)$, and performing

$$\text{SHAKE}\,(x'',\, x'(t_0),\, x(t_0)) \tag{4.7}$$

The resulting positions $x(t_0)$ are the shaken initial positions. When $x(t_0)$ satisfy the constraints, the velocities $v'(t_{-1/2})$ can be shaken by the procedure given in steps (3a–c) of the computational scheme, applied to $x(t_0)$ and $v'(t_{-1/2})$ with $-\Delta t$ instead of Δt.

5 DISCUSSION

In [12] Allen has shown that the Beeman SD algorithm, which he had derived earlier [15], is equivalent to the Verlet SD algorithm proposed by us in ref. [11]. Here, we will show that the leap-frog SD algorithm of section 3 is also equivalent to the Verlet SD algorithm of ref. [11].

The Verlet SD algorithm of ref. [11] is slightly more complex than necessary. The final formula (2.6) in [11] does contain a term involving $F'(t_n)$, which is of fourth order in Δt. However, in section 2 it has been argued that the positions need be determined only up to third order inclusive, as in the Verlet MD algorithm. Therefore, the Verlet SD algorithm of ref. [11] can be simplified. The terms involving $F'(t_n)$ in formulae (2.6) and (2.22) and all steps involving $F(t_n)$ in the computational schemes may be omitted. This also means that the systematic forces $F(t_{n-1})$ do not need to be stored.

When making these simplifications, the formula for the *Verlet SD algorithm* reads

$$x(t_{n+1}) = x(t_n)[1 + \exp(-\gamma\Delta t)] - x(t_{n-1})\exp(-\gamma\Delta t) + m^{-1}F(t_n)(\Delta t)^2(\gamma\Delta t)^{-1} \times [1 - \exp(-\gamma\Delta t)] + X_n(\Delta t) + \exp(-\gamma\Delta t)X_n(-\Delta t) + \mathrm{O}[(\Delta t)^4], \tag{5.1}$$

with

$$X_n(\Delta t) = (m\gamma)^{-1}\int_{t_n}^{t_n+\Delta t}[1 - \exp[-\gamma(t_n + \Delta t - t')]]\,R(t')\,\mathrm{d}t' \tag{5.2}$$

in accordance with the definition used previously in (3.9). The velocities, which are external to the operation of the algorithm, may be obtained from

$$v(t_n) = [x(t_{n+1}) - x(t_{n-1}) + X_n(-\Delta t) - X_n(\Delta t)](\Delta t)^{-1}(\gamma\Delta t)/[\exp(+\gamma\Delta t) - \exp(-\gamma\Delta t)]. \tag{5.3}$$

The equivalence of the leap-frog SD formulas (3.6, 3.10) with the Verlet SD formula (5.1) can be proved as follows. We start from Eq. (3.10) and substitute $v(t_{n+1/2})$ using Eq. (3.6). In the resulting formula $v(t_{n-1/2})$ is substituted by the expression obtained by writing in Eq. (3.10) $(n - 1)$ instead of n. The result is

$$\begin{aligned} x(t_{n+1}) = {} & x(t_n)[1 + \exp(-\gamma\Delta t)] - x(t_{n-1})\exp(-\gamma\Delta t) + m^{-1}F(t_n)(\Delta t)^2(\gamma\Delta t)^{-1} \\ & \times [1 - \exp(-\gamma\Delta t)]\cdot\{[\exp(\gamma\Delta t/2) - \exp(-\gamma\Delta t/2)]/(\gamma\Delta t)\} \\ & + X_{n+12}(\Delta t/2) - X_{n+1/2}(-\Delta t/2) + [V_n(\Delta t/2) - \exp(-\gamma\Delta t) \\ & \times V_n(-\Delta t/2)][\exp(\gamma\Delta t/2) - \exp(-\gamma\Delta t/2)]/\gamma + \exp(-\gamma\Delta t) \\ & \times [X_{n-1/2}(-\Delta t/2)] - X_{n-1/2}(\Delta t/2)]. \end{aligned} \tag{5.4}$$

This expression is up to third power of Δt inclusive, equal to (5.1), as can be seen by expanding the factor within accolades in the third term in powers of $\gamma\Delta t$. The first three stochastic integral terms are equal to $X_n(\Delta t)$ and the last three are equal to $\exp(-\gamma\Delta t)X_n(-\Delta t)$. This means that the leap-frog SD algorithm (3.6, 3.10) is indeed a third-order algorithm, as was suggested in section. 3.

As far as storage of intermediate vectors is concerned, the leap-frog and Verlet algorithms require less computer memory than the Beeman algorithm does. In the usual mode of operation, the last stores four vectors: $x(t_n)$, $v(t_n)$, $F(t_n)$ and $F(t_{n-1})$ [12]. The Verlet and leap-frog algorithms discussed here require storage of only three vectors: $x(t_n)$, $x(t_{n-1})$ and $X_{n-1}(\Delta t)$ for the former and $x(t_n)$, $v(t_{n-1/2})$ and $X_{n-1/2}(\Delta t/2)$ for the latter.

In [17] we have discussed and have given references to various applications of SD algorithms such as the one derived here. Vesely [20] has given an algorithm for the more general case of non-white random forces, and Ciccotti and Ryckaert also integrated the generalized Langevin equations [21].

Finally we would like to mention a few promising applications of stochastic dynamics techniques.

1. Stochastic dynamics can be used instead of molecular dynamics or energy minimization techniques in order to search the conformational space of a molecule for low energy conformations.

2. When an appropriate potential of mean force can be derived, describing the average effect of the solvent molecules on a solute molecule, the latter can be simulated by stochastic dynamics. The omission of solvent molecules would make an SD simulation more efficient by a factor of 10–50 than a full MD simulation treating all solvent molecules with periodic boundary conditions.
3. Application of stochastic dynamics allows one to vary the strength of the coupling of the atoms (through the friction coefficients γ) to the heat bath. This yields the possibility of studying the process of transfer of energy through a molecule.

APPENDIX

For small values of $x = \gamma\Delta t$ or $y = \gamma\Delta t/2$, the coefficients of the algorithm should be evaluated using power series expansions. These are given below.

$$[1 - \exp(-x) = x - 1/2x^2 + 1/6x^3 - 1/24x^4 + 1/120x^5 - 1/720x^6 + 1/5040x^7 - O[x^8] \quad (A.1)$$

$$[\exp(y) - \exp(-y)] = 2y + 1/3y^3 + 1/60y^5 + 1/2520y^7 + O[y^9] \quad (A.2)$$

$$[1 - \exp(-2y)] = 2y - 2y^2 + 4/3y^3 - 2/3y^4 + 4/15y^5 - 4/45y^6 + 8/315y^7 - O[y^8] \quad (A.3)$$

$$B(y) = 1/3y^4 + 1/3y^5 + 17/90y^6 + 7/90y^7 + 43/1680y^8 + 107/15120y^9 + 769/453600y^{10} + O[y^{11}] \quad (A.4)$$

$$C(y) = 2/3y^3 - 1/2y^4 + 7/30y^5 - 1/12y^6 + 31/1260y^7 - 1/160y^8 + 127/90720y^9 - O[y^{10}] \quad (A.5)$$

$$D(y) = -y^2 - 1/12y^4 - 1/360y^6 - 1/20160y^8 - O[y^{10}] \quad (A.6)$$

$$B(y)/C(y) = 1/2y + 7/8y^2 + 367/480y^3 + 857/1920y^4 + 52813/268800y^5 + 224881/3225600y^6 + 1341523/64512000y^7 + O[y^8] \quad (A.7)$$

$$yD(y))/C(y) = -3/2 - 9/8y - 71/160y^2 - 81/640y^3 - 7807/268800y^4 - 1971/358400y^5 - 56417/64512000y^6 - O[y^7] \quad (A.8)$$

$$B(-y)/[\exp(-2y) - 1] = -1/6y^3 + 1/60y^5 - 17/10080y^7 + 31/181440y^9 - O[y^{11}] \quad (A.9)$$

$$D(-y)[\exp(-2y) - 1] = 1/2y + 1/2y^2 + 5/24y^3 + 1/24y^4 + 1/240y^5 + 1/720y^6 + 5/8064y^7 + O[y^8] \quad (A.10)$$

For a 48-bit mantissa accuracy the analytical formulas of section 3 are accurate to better than 1:10^6 when $x = 2y = \gamma\Delta t > 0.05$. For $x < 0.05$ this accuracy will be maintained if the expansions given above are used.

References

[1] G. Cicotti and W.G. Hoover, eds, *Molecular Dynamics Simulation of Statistical Mechanical Systems*, Proceedings of the International School of Physics "Enrico Fermi", course 97, North-Holland, Amsterdam, 1986.

[2] R. Kubo, "The fluctuation–dissipation theorem", *Rep. Progr. Phys.*, **29**, 255 (1966).

[3] G. Ciccotti and J.P. Ryckaert, "On the derivation of the generalized Langevin equation for interacting Brownian particles", *J. Statist. Phys.*, **26**, 73 (1981).

[4] G. Bossis, B. Quentrec and J.P. Boon, "Brownian dynamics and the fluctuation–dissipation theorem", *Molec. Phys.*, **45**, 191 (1982).

[5] W.F. van Gunsteren and H.J.C. Berendsen, "On the fluctuation–dissipation theorem for interacting brownian particles", *Molec. Phys.*, **47**, 721 (1982)

[6] W.F. van Gunsteren and H.J.C. Berendsen, "Algorithms for macromolecular dynamics and constraint dynamics", *Molec. Phys.*, **34**, 1311 (1977).

[7] H.J.C. Berendsen and W.F. van Gunsteren, "Practical algorithms for dynamic simulations", in [1], pp. 43–65.

[8] L. Verlet, "Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules", *Phys. Rev.*, **159**, 98 (1967).

[9] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, McGraw-Hill, New York, 1981.

[10] D. Beeman, "Some multistep methods for use in molecular dynamics calculations", *J. Comput. Phys.*, **20**, 130 (1976).

[11] W.F. van Gunsteren and H.J.C. Berendsen, "Algorithms for Brownian dynamics", *Molec. Phys.*, **45**, 637 (1982).

[12] M.P. Allen, "Algorithms for Brownian dynamics", *Molec. Phys.*, **47**, 599 (1982).

[13] J. Åqvist, W.F. van Gunsteren, M. Leijonmarck and O. Tapia, "A molecular dynamics study of the C-terminal fragment of the L7/L12 ribosomal protein: secondary structure motion in a 150 ps trajectory", *J. Molec. Biol.*, **183**, 461 (1985).

[14] D.L. Ermak and H. Buckholz, "Numerical integration of the Langevin equation: Monte-Carlo simulation", *J. Comput. Phys.*, **35**, 169 (1980).

[15] M.P. Allen, "Brownian dynamics simulation of a chemical reaction in solution", *Molec. Phys.*, **40**, 1073 (1980).

[16] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, 1965.

[17] W.F. Van Gunsteren, H.J.C. Berendsen and J.A.C. Rullmann, "Stochastic dynamics for molecules with constraints: Brownian dynamics of n-alkanes", *Molec. Phys.*, **44**, 69 (1981).

[18] J.P. Ryckaert, G. Ciccotti and H.J.C. Berendsen, "Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes", *J. Comput. Phys.*, **23**, 327 (1977).

[19] H.J.C. Berendsen and W.F. van Gunsteren, "Molecular dynamics simulations: techniques and approaches", in *Molecular Liquids-Dynamics and Interactions*, A.J. Barnes *et al.*, eds, NATO ASI Series *C135*, Reidel, 1984, pp. 475–500.

[20] F.J. Vesely, "Algorithms for Mori chain dynamics: a unified treatment", *Molec. Phys.*, **53**, 505 (1984).

[21] G. Ciccotti, M. Ferrario and J.P. Ryckaert, "Computer simulation of the generalized Brownian motion II. An argon particle in argon fluid", *Molec. Phys.*, **46**, 875 (1982).